Machine learning: statistical physics based theory and smart industry applications

Straat, Michiel

DOI:
10.33612/diss.255731774

university of groningen

Machine learning: statistical physics based theory and smart industry applications

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. C. Wijmenga
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Tuesday 6 December 2022 at 12.45 hours

by

Michiel Joost Christiaan Straat

born on 30 December 1994
in Tytsjerksteradiel

Supervisors
Prof. M. Biehl
Prof. K. Bunte
Prof. N. Petkov

Assessment Committee
Prof. G. Gaydadjiev
Prof. O. Winther
Prof. B. Rosenow

Abstract

The contributions in this thesis are divided into two main parts: 1) a theoretical analysis of learning in neural networks and Learning Vector Quantization (LVQ) in model situations using statistical physics techniques and 2) the application of machine learning to smart industry settings.

In the first part we address highly relevant situations and questions for current machine learning practice: using tools from statistical physics we analyse the learning behaviour in Rectified Linear Unit (ReLU) neural networks and compare it to sigmoidal neural networks in both on-line and off-line supervised learning settings, in order to contribute to the much needed theoretical insight into the properties of the use of the ReLU function, the most popular type of activation function in deep neural networks that are used in many machine learning tasks. Secondly, we analyse neural networks and LVQ under real and virtual concept drift processes that affect many applications of machine learning systems. Our analyses reveal several significant effects, which are, among others: ReLU networks handle overparameterization differently than sigmoidal networks, ReLU networks exhibit favourable second order phase transitions towards hidden unit specialization instead of the first order phase transitions observed for sigmoidal networks and applying weight decay in concept drift scenarios is more effective for ReLU neural networks, in which it significantly accelerates the onset of specialization. For LVQ we find non-trivial dependence of the generalization performance on the learning rate in concept drift situations. Moreover, it is shown that an appropriate amount of weight decay can be beneficial to the performance in the real drift settings. In contrast, the resulting limited flexibility of the prototypes decreases the performance under virtual drift.

In the second part of the thesis we focus on the use of computational intelligence approaches in applications in industry. First, we perform a typical Industry 4.0 case study in collaboration with industry that concerns the development of real-time material quality control in a high-throughput production line of steel-based products. In this case study, material measurements

taken with a fast non-invasive sensor are related to material properties measured by tensile tests. Due to significant correlations between the two types of measurements, we successfully fit and evaluate a model that can estimate material properties in real-time. Furthermore, it is shown on 108 kilometres of processed steel that the model is able to prevent expensive production problems and that it can indicate a risk of the occurrence of production faults.

Additionally, we propose a methodology for time series classification that combines Generalized Matrix Learning Vector Quantization (GMLVQ) formulated for complex-valued data with complex-valued Fourier and wavelet features. On several benchmark datasets and a heart beat classification task, learning in the space of complex-valued coefficients is found to yield better classification accuracy with fewer adaptive parameters compared to time domain classification. Moreover, we formulate a back-transformation of prototypes and relevance values that facilitates the interpretation of the classifier in both the transform- and the time domain.

Acknowledgements

This thesis marks the end of my time as a PhD student at the University of Groningen. As part of the Intelligent Systems research group, I learnt from and collaborated with great professors and colleagues. In particular, I owe a great debt of gratitude to my supervisors Prof. Michael Biehl, Prof. Kerstin Bunte and Prof. Nicolai Petkov. Having the opportunity to collaborate with them has been crucial to my development and through their supervision I got inspired to try new ideas and shape the direction of this research. I also thank Prof. Michael Wilkinson for inspiring conversations.

I would like to express my sincere thanks to my paranymphs Abolfazl Taghribi and Elisa Oostwal and also to my other office mates Mohammad Mohammadi and Mohammad Babai. I am grateful for our fruitful discussions about research and general topics. Fellow PhD students from the research group and the institute have been important to me in many ways: from helpful discussions to playing a good game of squash.

As part of my PhD project, I collaborated with Philips in Drachten to solve a problem in the factory using machine learning techniques. It provided me with highly valuable experience and I am grateful for the excellent collaboration at Philips with Nick Goet and Kevin Koster.

I would like to express my gratitude to the assessment committee for reviewing the thesis and I thank the reviewers of the papers.

Lastly, I express special thanks and appreciation to my parents, my brother and my friends. They have always supported me.

Michiel Straat
Groningen
November 13, 2022

Contents

List of Figures

Chapter 1

Introduction

Machine Learning algorithms play an increasingly important role in science and applications. Although the origins of machine learning go back to at least the second half of the 20th century and arguably earlier due to its strong dependence on statistical theory, the advances in several fields are currently revealing more of the potential of machine learning methods. Most prominently, the prevalence of advanced sensors, cameras, web and mobile applications has increased significantly. Combined with developments in communication technology and cloud computing, it has become possible to measure, transfer and store large amounts of data. Simultaneously, the ever-increasing speed and availability of computational devices have facilitated the training of large artificial neural network architectures on big datasets to perform remarkably complex tasks. Some tasks that neural networks currently perform had been thought earlier to belong to the realm of human intelligence mainly and particularly difficult for computers to carry out. Examples can be found, among others, in advanced speech processing (Khalil et al. 2019, Wang and Chen 2018), self-driving cars (Gupta et al. 2021), advanced recommendation systems (Fessahaye et al. 2019) and medical applications (Shen et al. 2017, Singh et al. 2020).

The increased capabilities of neural networks and the accessibility of tools that facilitate their implementation have come at a cost: the growth in model size has inevitably complicated the analysis of the model's inner workings and overall behaviour. The black box character of the large networks makes it extremely challenging to give guarantees about the model's performance on new data beyond the set on which it was trained and evaluated. This causes various vulnerabilities in the models, see for instance (Göpfert et al. 2020) for a study related to so-called adversarial examples. These are inputs altered in a way humans can hardly perceive, but that cause neural networks to make drastically different and unexpected predictions. Whereas failing models merely cause annoyances in some recommendation systems on the web, in safety-critical autonomous systems such as self-driving cars the consequences can be life-threatening (Tu et al. 2020).

Although the increase in model and data scale explains the recent advances to a large extent, it has certainly not been the only driver of progress. The intricate field of neural networks, and more generally machine learning, can broadly be described by

three interdependent main components: model architecture, learning algorithm and data (Zdeborová 2020). The interplay of these three components determines crucial performance characteristics, such as the learning speed of neural networks and their generalization ability beyond the training and evaluation data. Neural network architectures have been proposed that are highly efficient for certain tasks, such as convolutional neural networks for computer vision tasks (LeCun 1989). In the neural network learning algorithms and optimization domain prominent recent studies have adapted and extended the default gradient descent formulation (Kingma and Ba 2015). Other examples are *active learning* algorithms which decide for which data points or parts of the data space a target label would be most informative (Settles 2009).

Zooming in on model architecture, studies have focused on a key component: the non-linear activation function that is used in neurons in the network to map input- to output activations. The choice of activation function defines the non-trivial functional representations within neural networks and therefore plays an important role in learning. The so-called Rectified Linear Unit (ReLU) function has been found to result in faster convergence and better generalization error in a variety of tasks implemented by deep neural networks (Jarrett et al. 2009, Nair and Hinton 2010). It is hypothesized that this is partly due to the resulting sparse activity in the networks and the prevention of the *vanishing gradient problem* that hampers learning in deep sigmoidal neural networks. However, the properties and characteristics of the use of ReLU in various learning scenarios and architectures are not understood thoroughly and hence there is a need for more theoretical investigations.

A second topic of great interest in machine learning is the frequently occurring phenomenon of *concept drift* (Zliobaite et al. 2016, Faria et al. 2016). This refers to situations that contain a changing input data distribution over time, known as virtual drift, and/or a change in the target task, which is referred to as real drift. The appropriate handling of the various types of concept drift requires a combination of different approaches that depend on the application and machine learning models. Theoretical approaches that study the behaviour of machine learning models and algorithms under different types of concept drift are necessary to contribute to the design of effective methods that retain the performance of these systems in applications exhibiting concept drift.

The need for theoretical analyses and understanding of these neural network and machine learning settings could be addressed by *the statistical physics of learning*, as has recently been argued in (Zdeborová 2020), among others. The general idea is to use methods of statistical physics to study the average-case properties of machine learning models that exhibit many parameters. These types of analyses complement other theoretical investigations of deep learning settings.

Starting from at least the 1980s, statistical physics techniques have been used extensively for the analysis of various neural network- and machine learning architectures. For instance, the Hopfield network that realizes an associative memory strongly resembles an Ising model of magnetic spins with the same energy function associated with the state of the network. A Hebbian update rule minimizes the energy and the network dynamics are attracted by stable states corresponding to the stored patterns (Hopfield 1982). A quantity of interest in the Hopfield model and several related theoretical studies of neural networks is the *storage capacity*, which is the ratio of patterns to neurons that can be reproduced by the network, see also (Hopfield 1982, Gardner and Derrida 1988, Gardner 1988, Baldassi et al. 2019, Zavatone-Veth and Pehlevan 2021). In many subsequent studies statistical physics methods and analogies have also been used in the analysis of, among others, supervised regression and classification problems implemented by neural networks.

The so-called *student teacher* setting has frequently been used in the formulation of such learning scenarios (Engel and van den Broeck 2001): the teacher is a model instance that defines the target regression or classification rule and thus provides the target labels for the input data. The student is a model instance that represents a data-driven hypothesis about the rule. In the general treatment, the many degrees of freedom on the microscopic level - the weights of the networks - are summarized by a few macroscopic *order parameters*, which describe both student and teacher models and their relationship. To facilitate the analytical treatment, the input data is assumed to be independently and identically distributed. Both on-line and off-line learning scenarios have been analysed using these ideas (Engel and van den Broeck 2001).

In the statistical physics analysis of on-line learning, a machine learning system is modelled that learns in an incremental fashion from a stream of independently generated data points, i.e. in these systems the data points are used once for model optimization with respect to the cost function and are discarded afterwards. Hence, in the modelling of the dynamics of the order parameters it becomes possible to compute averages of the learning rule over the data distribution with respect to the latest input. For an increasing number of input dimensions of the system the variance of the resulting dynamics decreases, which is due to the *self-averaging* property of the order parameters. Hence by computing averages of the learning rules for updating the order parameters, closed form equations are obtained for the average dynamics which describe the dynamics for finite systems with higher accuracy for increasing system size. In the limit of the number of input dimensions to infinity, the dynamics of the order parameters coincide with the average dynamics.

In the modelling of off-line learning, the statistical physics-based modelling describes a machine learning system that is optimized with respect to datasets of a fixed size. In off-line machine learning settings, the stochastic optimization of the cost

function over all training points with respect to the weights of the model requires the repeated use of the data points, which contrasts the on-line learning setting and therefore asks for a different modelling approach: in the modelling the cost function is regarded as an energy which is then analysed using techniques from equilibrium statistical mechanics. Energy minimization is modelled as an optimization process that exhibits noise that is dependent on a formal temperature parameter. In this setting the equilibrium properties are determined by the minimization of the so-called *free energy*: the minimization reflects the existence of a competition between a few highly probable low energy states and a large number of less probable states with a higher energy. The latter is the *entropy* that is defined by the volume of states that exhibit a particular energy. An additional average over the data distribution provides typical equilibrium results with respect to the order parameters independent of an exact realization of a dataset.

For a detailed overview of the concepts in the statistical physics of learning including many examples, see (Engel and van den Broeck 2001, Watkin et al. 1993, Opper and Kinzel 1996, Saitta et al. 2011), among others. In the next sections we detail the research aims relating to the ReLU function and concept drift that we address using methods from the statistical physics of learning.

A second topic in this thesis concerns the application of machine learning techniques to the optimization of manufacturing processes. The potential of machine learning techniques in combination with the aforementioned advances in, among others, communication and sensor technologies, has been recognized to be particularly powerful in the realization of highly optimized manufacturing processes. The vision of manufacturing of the future, which consists of densely interconnected networks of sensors, machines, and entire factories in production chains that perform optimization of production processes in real-time, is regarded as the next major step in the development of manufacturing and has therefore been called *Industry 4.0* (Schwab 2017). The implementation of Industry 4.0 is a challenging process that relies on research and solutions from diverse fields.

In order to contribute to these efforts, in collaboration with a manufacturer of steel-based products we perform a typical Industry 4.0 case study. In steel-based mass production settings it is crucial that the machinery operates on material that is within specifications, in order to prevent costly damage to the production tooling (Vaidya et al. 2018). Hence, in the steel-based manufacturing industry there is a need for real-time material quality control approaches to ensure all material conforms to the specifications before entering the production process. To this end, the use of in-line sensors that perform fast contactless measurements has become common (García-Martín et al. 2011). However, connecting these sensor measurements to production control systems and preventive intervention systems is highly non-trivial

and application dependent. Our case study aims at the prediction of material properties from contactless Eddy Current measurements, in order to obtain a real-time quality control solution. A second important aim is to relate the model predictions to faults that occurred in production. With the latter analyses we study the ability of the system to prevent faults in the future.

1.1 Scope of the thesis and research questions

Part I of this thesis addresses the need for more theoretical understanding of neural networks, the role and properties of the activation function and learning under concept drift. To this end, we analyse learning processes in model settings with tools from the statistical physics of learning. These methods yield typical and average case results, which complement other empirical and theoretical studies well to provide a more thorough understanding. In particular, we analyse learning and optimization processes in ReLU neural networks on the level of order parameters for a large number of input dimensions. Furthermore, learning under various types of concept drift is studied for Learning Vector Quantization (LVQ) and neural networks. Specifically, we investigate the following research topics in the chapters of the thesis:

- We formulate a modelling framework for analysing on-line gradient descent learning scenarios based on the student-teacher setup and averaging over a high-dimensional density of uncorrelated inputs. Using the modelling framework we put forward the typical dynamics of order parameters in two-layer neural networks with ReLU activation. The obtained system of ordinary differential equations describes the dynamics of the order parameters in the limit of an infinitely large number of input dimensions. The average error over the input distribution is obtained analytically in terms of the order parameters of the model.

- With the obtained modelling framework, we address the following research question: **what are key differences between learning in ReLU and sigmoidal neural networks in relevant on-line learning settings?** We address this research question by solving the dynamical system numerically and analysing the obtained learning curves for three highly relevant settings: 1) matching student and teacher complexity, 2) an overparameterized student and 3) an unlearnable target rule. A qualitative comparison is made for these three settings between the learning curves obtained for ReLU neural networks and for sigmoidal neural networks.

- **How does concept drift affect the learning performance of neural networks**

and LVQ? To address this question, we first formulate the on-line learning dynamics of LVQ and then extend the modelling framework to include various forms of drift. In particular, in case of LVQ, we include virtual drift of class biases and real drift using random displacements of cluster centers of the data generating distribution. In the case of two-layer networks, real drift processes are considered that are modelled by random displacements of the weight vectors in the teacher network. Here the comparison between ReLU- and sigmoidal networks is also in the center of interest. **Does weight decay improve learning performance in concept drift situations?** For both LVQ and neural networks, a weight decay is included as a forgetting mechanism. Besides the analysis of the effect of the different types of drift on the two models, we also study the properties of applying weight decay in these concept drift situations.

- **How does off-line learning behaviour in ReLU networks differ from off-line learning in sigmoidal networks?** Using a different modelling framework based on equilibrium statistical physics we analyse off-line learning in both two-layer ReLU- and sigmoidal neural networks: for datasets of uncorrelated high-dimensional inputs we formulate the quenched free energy function in terms of order parameters of a canonical ensemble of networks in the simplifying limit of high formal training temperatures. The minimization of the free energy function yields the typical result of stochastic optimization in terms of the order parameters of the system for increasing values of the dataset size.

In the application in collaboration with industry we contribute to the Industry 4.0 effort by performing a case study of a production line in a typical smart industry setting. We address the need for advanced real-time quality control systems in steel-based mass production and devise a methodology for the early detection of production faults. The research questions and scope are as follows:

- The case study addresses the prediction of material properties from fast Eddy Current sensor measurements. In particular, we address the question: **are material properties of steel that are obtained by tensile testing predictable from fast contactless Eddy Current sensor measurements?** We consider the real-time estimation of material properties and quality in a mass production setting of steel-based products. We explore the predictability of material properties obtained by tensile tests from measurements performed by an Eddy Current sensor.

- **1) Can the Eddy Current sensor measurements prevent material of insufficient quality from entering the production? 2) Can Eddy Current measure-**

ments be used to prevent production faults? We study whether the real-time estimation of material properties can prevent material of insufficient quality or material exceeding the specifications from entering production. Secondly, we test using sensor measurements and logbooks of previous production data whether faults that occurred in production could have been prevented. This study is performed using sensor measurement and logbook data covering 108 km of strip steel that was used for production.

Lastly, we consider the highly relevant task of time series classification that arises in many domains including smart industry settings. These types of data often have a continuous functional nature that should be taken advantage of.

- **To which extent is it possible to increase classification accuracy and reduce the number of dimensions by exploiting the functional nature of time series data in combination with complex-valued GMLVQ?** We propose a method that uses complex-valued feature representations of time series as obtained by the complex-valued Fourier or wavelet transform in combination with an extension of Generalized Matrix Learning Vector Quantization (GMLVQ) to the complex domain. Furthermore, a backtransformation of the classifier is formulated in order to obtain additionally a time domain interpretation of relevance values and prototypes. The method is exemplified on a variety of time series datasets and on a heart beat classification task. Central to our investigation is the analysis of the classification accuracy in comparison to time domain training and to the standard approach of handling complex values by the concatenation of real and imaginary parts in a real-valued vector. Simultaneously, the potential of reducing the number of dimensions in the classification tasks is analysed.

Although this thesis contains proposed methods and analyses from different settings and perspectives, the relevance between the topics is emphasized. As one example: an increased understanding of machine learning under concept drift as addressed in Part I is highly relevant to processes in smart factory settings such as in Part II.

1.2 Outline of the thesis chapters

This thesis is divided into two parts. Part I concerns the theoretical analyses of model scenarios in machine learning and Part II consists of the Industry 4.0 case study and the classification of time series data. Here we provide how the various analyses and aims of the thesis as discussed in the previous sections are arranged in the chapters.

1.2.1 Part I

Part I consists of three chapters. In Chapter 2 we introduce a generic modelling framework based on the student-teacher concept to analyse on-line gradient descent learning in machine learning models. The order parameters of the system, the input data distribution and the behaviour in the limit of a large number of input dimensions are introduced and discussed. Using the modelling framework, we perform the required calculations to formulate Ordinary Differential Equations (ODE) that describe the exact large input dimension evolution of order parameters in on-line learning of ReLU two-layer neural networks in a number of scenarios: matching student and teacher, an overparameterized student and an unlearnable rule. Subsequently, the generalization error is formulated analytically in terms of the order parameters. In these settings, we focus on the comparison of the learning behaviour in ReLU networks with sigmoidal networks.

In Chapter 3 we extend the analysis to incorporate concept drift. First, using the generic modelling framework introduced in Chapter 2, we formulate macroscopic learning dynamics of LVQ for input data that is generated by a clustered density. In this setting, the cluster membership defines the target label. We extend the modelling framework to incorporate a variety of virtual and real drift processes: for LVQ, we consider changing class biases in the input stream and the random displacement of cluster centers in the input space. For two-layer networks, a real drift process is introduced in the modelling by random displacements of the teacher weight vectors. Moreover, weight decay is introduced in the modelling framework as a mechanism of forgetting. The second part of the chapter contains the results of the learning behaviour of the models under the various drift settings. In concept drift settings, the role of the learning rate and weight decay in LVQ is analysed and the use of weight decay in ReLU and sigmoidal neural networks is discussed in detail.

Chapter 4 concerns the analysis of off-line learning in two-layer ReLU and sigmoidal networks. Following previous work, we formulate the free energy of a canonical ensemble of networks in the simplifying high temperature limit. The quenched free energy with respect to the input data distribution depends on the order parameters, which fully define its two components in the considered limit: the generalization error and the entropy. Hence, in order to formulate the free energy for ReLU networks we use the generalization error derived in Chapter 2 and combine it with the result of previous work for the entropy of the system. Typical results of stochastic optimization are obtained by numerically minimizing the free energy with respect to the order parameters for an increasing control parameter, interpreted as dataset size. In previous work, this type of analysis was done for sigmoidal networks. We reproduce those results and compare them to the results obtained for ReLU net-

works. The systems exhibit phase transitions and their study and comparison is of central interest in the discussion in the chapter.

Although the aim is to keep the thesis' chapters as self-contained as possible with separate introductions to each chapter, in this part of the thesis later chapters contain some references to discussions and equations in earlier chapters. Chapter 2 in particular introduces concepts and equations that are used in Chapters 3 and 4. For instance, Chapter 3 relies on the discussion of the on-line gradient descent setting in the limit of a large number of input dimensions. Therefore the discussion and a few equations describing the modelling framework in Chapter 2 are referenced. Furthermore, Chapters 3 and 4 refer to the definition of the student teacher setting, the order parameters and the generalization error in Chapter 2.

1.2.2 Part II

Chapters 5 and 6 constitute Part II of the thesis. Chapter 5 discusses our work in collaboration with industry. We describe the Industry 4.0 vision and introduce the typical case study. The case study concerns real-time quality control of steel in a high-throughput production line, in which it is crucial that the material conforms to the specifications in order to prevent costly damage to the main production press in the line. The real-time quality control is based on Eddy Current sensor measurements that are performed in a non-invasive manner. We analyse the relationship between the sensor measurements and the manual destructive tests that are usually taken to measure properties of the material. Based on this analysis, a model is proposed and evaluated that estimates two key material properties given the sensor measurements. Furthermore, we analyse on previous production data whether the model could have prevented various types of production faults that occurred, in order to test the model's ability to prevent faults in the future.

Chapter 6 concerns the classification of time series data. The problem is highly relevant to various settings: industrial, natural, financial and medical settings contain many problems that revolve around the classification of time series. Often, time series have a functional continuous nature that can be taken advantage of in the machine learning task by using functional decompositions. In particular, we propose a methodology that uses complex-valued Fourier- and wavelet features in combination with an adaptation of GMLVQ that works directly with complex-valued data. The benefits of the approach and the appropriateness of the method are discussed by showing its performance on several benchmark datasets. In a larger experiment, we consider the classification of heartbeats using only complex-valued wavelet features. In all experiments, we test whether dimensionality reduction by means of truncation of the complex-valued feature vectors is a promising strategy

for reducing the number of adaptive parameters and thereby over-fitting effects. A back-transformation of prototypes and relevance values is formulated to obtain the time domain interpretation of the classifier besides the transform interpretation.

In Chapter 7 the thesis and the main results are summarized. The thesis is concluded by a discussion about possible future directions of research.

Part I

Model scenarios of machine learning

Published as:

M. Straat, M. Biehl – *"On-line learning dynamics of ReLU neural networks using statistical physics techniques,"*, in M. Verleysen (ed.), 27th Europ. Symp. on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 517–522, 2019.

Chapter 2

Dynamics of online gradient descent in ReLU networks

Abstract

In this chapter we study learning behavior in ReLU neural networks using a previously established framework based on statistical physics techniques. It uses the concept of a student- and teacher model, in which the student model learns from data with target labels that are provided by the teacher model. For the limit of large inputs $N \to \infty$ and independent examples, it is possible to derive exact learning dynamics in terms of a system of differential equations that describe the evolution of macroscopic parameters that aggregate the many weights of the network. We focus on the gradient descent learning algorithm and our formulation is generic in the number of hidden units in the student and teacher model and the activation function. By performing the required calculations for the ReLU activation function, we obtain the Ordinary Differential Equations (ODE) of the learning dynamics for ReLU networks. Using numerical integration of the ODE we study an overparameterized student and an unlearnable rule, and we find significant differences of the learning dynamics of ReLU neural networks compared to the dynamics of sigmoidal neural networks. In some of the experiments, we perform Monte Carlo simulations which are in excellent qualitative agreement with the theoretical results. We emphasize the broad applicability of the modelling framework in the analysis of typical learning dynamics in a variety of machine learning situations. For instance, in Chapter 3, the modelling framework is extended to analyse Learning Vector Quantization (LVQ) and neural networks in non-stationary situations.

2.1 Introduction

The many challenges of modern data science call for the design and putting forward of efficient methods for automated analysis. Machine learning techniques play a key role in this context (Hastie et al. 2001, Bishop 2006, Goodfellow et al. 2016). The subfield of artificial neural networks within machine learning has especially seen

a tremendous rise in popularity, which is largely due to the successful application of so-called *deep learning* in a number of practical contexts, see e.g. (Goodfellow et al. 2016, LeCun et al. 2015, Angelov and Sperduti 2016) for reviews and further references. The successful training of these powerful, multi-layered deep networks has become feasible for a number of reasons, including the automated acquisition of large amounts of training data in various domains, the use of modified and optimized architectures, e.g. convolutional neural networks for image processing, and the ever-increasing availability of computational power needed for the implementation of efficient training (Goodfellow et al. 2016).

One particularly important modification of earlier models is the use of alternative activation functions (Goodfellow et al. 2016, Ramachandran et al. 2017, Eger et al. 2018). Arguably, the so-called Rectified Linear Unit (ReLU) constitutes the most popular choice in Deep Neural Networks (Hahnloser et al. 2000, Krizhevsky et al. 2012, Goodfellow et al. 2016, Ramachandran et al. 2017, Eger et al. 2018, Maas et al. 2013). Compared to more traditional activation functions, the simple ReLU and recently suggested modifications warrant computational ease and appear to speed up the training, see for instance (Nair and Hinton 2010, Maas et al. 2013, Villmann et al. 2019). The one-sided ReLU function is found to yield sparse activity in large networks, a feature which is frequently perceived as favorable and biologically plausible (Hahnloser et al. 2000, Goodfellow et al. 2016, Glorot et al. 2011). In addition, the problem of *vanishing gradients*, which arises when applying the chain rule in layered networks of sigmoidal units, is avoided (Goodfellow et al. 2016). Moreover, networks of rectified linear units have displayed favorable generalization behavior in several practical applications and benchmark tests, e.g. (Ramachandran et al. 2017, Hahnloser et al. 2000, Krizhevsky et al. 2012, Eger et al. 2018, Maas et al. 2013).

In this chapter we investigate the on-line learning behavior of two-layer neural networks with ReLU activation. To this end, we develop a modelling framework based on statistical physics techniques, in which to obtain general insights into practically relevant phenomena. This is instrumental in order to achieve the necessary theoretical understanding.

Analytical and computational approaches that come from or are related to statistical physics (Hertz et al. 1991, Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993, Biehl and Caticha 2003, Biehl et al. 2009) have played an important role in this field and continue to do so. These methods are used to analyse the typical, average-case, learning behavior of various learning systems in model scenarios. Examples of the successful application of these techniques is the study of the on-line gradient descent learning dynamics of neural networks (Biehl and Schwarze 1995, Saad and Solla 1995b, Straat et al. 2018, Vicente and Caticha 1997, Inoue et al. 2003) and

prototype-based models (Biehl et al. 2007, Straat et al. 2018), the analysis of learning in non-stationary environments (Straat et al. 2018), and the analysis of off-line batch learning (Biehl, Schlösser and Ahr 1998).

The particularly successful analysis of on-line learning is based on the assumption that a sequence of independently generated random N-dimensional examples is presented to the learning system (Saad 1999, Biehl and Caticha 2003, Biehl et al. 2009). Macroscopic quantities, the so-called order parameters of the system, aggregate and summarize the usually large number of individual parameters of the machine learning system. Further simplifying assumptions and the consideration of the so-called thermodynamic limit $N \to \infty$ in combination with the Central Limit Theorem facilitate the exact mathematical description of typical macroscopic *learning curves* in terms of Ordinary Differential Equations (ODE). These equations provide a useful tool to study the behavior of learning theoretically, in order to gain a deeper understanding of the learning process, which could potentially be used to improve algorithms used in practical scenarios. Various reviews, article collections and monographs present and discuss the approach with respect to supervised learning in simple perceptrons and multilayered neural networks, see e.g. (Saad 1999, Hertz et al. 1991, Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993, Biehl and Caticha 2003, Biehl et al. 2009) and references therein. The Soft Committee Machine (SCM) in stationary environments has been studied extensively from the statistical physics perspective. Practically relevant phenomena, such as the occurrence of quasi-stationary plateau states have been investigated in great detail, see (Biehl and Schwarze 1995, Saad and Solla 1995a, Saad and Solla 1995b, Riegler and Biehl 1995, Biehl et al. 1996, Vicente and Caticha 1997, Inoue et al. 2003, Saad 1999) for examples and further references. Similarly, the dynamics of unsupervised learning has been studied, including prototype-based competitive learning, Principal Component Analysis and related schemes (Biehl et al. 1997, Biehl, Freking, Reents and Schlösser 1998, Biehl and Schlösser 1998). For detailed discussions of the limitations of the approach as well as extensions that allow to overcome them, see several contributions in (Saad 1999) and, for instance, (Biehl et al. 2007).

Our investigation in the dynamics of ReLU neural networks starts with a brief revisit of the analytical treatment of on-line gradient-based learning in stationary environments. We present a so-called *student-teacher scenario* (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993) for the learning of a regression scheme with shallow, layered neural networks of the feedforward type. In this case, the discussion is restricted to the SCM: A two-layer neural network with non-linear activations in the hidden layer, fixed second layer weights and linear output.

The macroscopic dynamics are first given generically in a unified description and modelling framework. This modelling framework can be used to analyse the

training dynamics in various classification and regression systems. In Chapter 3, we use and extend the modelling framework of this chapter to formulate the on-line training dynamics for the Learning Vector Quantization classifier and the SCM in non-stationary situations.

The main result in this chapter is the formulation of the macroscopic learning dynamics for the special case of two-layer ReLU neural networks. These learning dynamics are studied in various model situations that are of great practical interest. In one setting the teacher has more hidden units than the students, which represents an *unrealizable* rule. In another setting the student is over-parameterized, which represents an *overrealizable* rule. Specific focus is on the comparison of the results obtained in these settings for ReLU networks to the results obtained earlier for traditional sigmoidal activation (Biehl and Schwarze 1995, Saad and Solla 1995a, Saad and Solla 1995b, Riegler and Biehl 1995, Biehl et al. 1996).

2.2 Definitions and Methods

2.2.1 Soft Committee Machines

The term Soft Committee Machine (SCM) has been coined for feedforward neural networks with sigmoidal activations in a single hidden layer and a linear output unit, see for instance (Biehl and Schwarze 1995, Saad and Solla 1995a, Saad and Solla 1995b, Riegler and Biehl 1995, Biehl et al. 1996, Vicente and Caticha 1997, Inoue et al. 2003, Biehl, Schlösser and Ahr 1998, Ahr et al. 1999). Its structure resembles that of a *(crisp)* committee machine with binary threshold hidden units, where the network's response is given by their *majority vote*, see (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993) and references therein.

Network Definition

The output of an SCM with K hidden units and fixed hidden-to-output weights is of the form

$$y(\boldsymbol{\xi}) = \sum_{k=1}^{K} g(\boldsymbol{w}_k \cdot \boldsymbol{\xi}) \tag{2.1}$$

where $\boldsymbol{w}_k \in \mathbb{R}^N$ denotes the weight vector connecting the N-dimensional input layer with the k-th hidden unit. A non-linear transfer function or activation function $g(\cdots)$ defines the hidden unit states and the final output is given as their sum. Traditionally, sigmoidal-shaped activation functions have been used. Here, we consider specifically

$$g(x) = \mathrm{erf}\left(x/\sqrt{2}\right) \text{ with the derivative } g'(x) = \sqrt{\frac{2}{\pi}}\, e^{-x^2/2}\,. \tag{2.2}$$

The activation resembles closely other sigmoidal functions, e.g. the popular $\tanh(x)$, but offers great mathematical ease in the analytical treatment as exploited in (Biehl and Schwarze 1995), originally. The ReLU function is defined as:

$$g(x) = \max\{0, x\} = x\,\Theta(x) = \begin{cases} 0 & \text{for } x \leqslant 0 \\ x & \text{for } x > 0 \end{cases} \text{ with the derivative } g'(x) = \Theta(x)\,, \tag{2.3}$$

where $\Theta(x)$ is the Heaviside step function defined as:

$$\Theta(x) = \begin{cases} 0 & \text{for } x \leqslant 0 \\ 1 & \text{for } x > 0 \end{cases}\,. \tag{2.4}$$

Note that the ReLU function is only semi-differentiable in $x = 0$; In this point the left derivative is zero and the right derivative is one. This mathematical subtlety is considered irrelevant in practice (Goodfellow et al. 2016) and one is free to choose a value in $[0, 1]$. However, a recent study shows empirical evidence for better training and generalization performance in neural networks with lower numerical precision using $\mathrm{ReLU}'(0) = \Theta(0) = 0$ compared to other choices for $\mathrm{ReLU}'(0)$ (Bertoin et al. 2021). In our simulations we use $\Theta(0) = 0$ as defined in Eq. (2.4). In the theoretical modelling framework that is introduced in Section 2.2.4, which relies on the integration of expressions involving the Heaviside functions $\Theta(\cdot)$ in case of ReLU activation, the choice of value for $\Theta(0)$ is irrelevant.

In the rest of the discussion, we will refer to an SCM that uses erf activation as Erf-SCM and to an SCM that uses ReLU activation as ReLU-SCM. Note that an SCM, cf. Eq. (2.1), is not quite representing a *universal approximator*. However, this property could be achieved by introducing hidden-to-output weights and adaptive local thresholds $\vartheta_i \in \mathbb{R}$ in hidden unit activations of the form $g\left(\boldsymbol{w}_i \cdot \boldsymbol{\xi} - \vartheta_i\right)$, see (Cybenko 1989, Hornik 1991, Hanin 2017) for a proof. Adaptive hidden-to-output weights have been studied in, for instance, (Riegler and Biehl 1995) from a statistical physics perspective. However, we restrict ourselves to the simpler model defined above and focus on the basic dynamical effects and potential differences between the ReLU- and Erf-SCM.

2.2.2 Regression Scheme and On-Line Gradient Descent

In the context of continuous regression, the training of neural networks with output $y(\boldsymbol{\xi}) \in \mathbb{R}$ based on examples $\left\{\boldsymbol{\xi}^\mu \in \mathbb{R}^N, \tau^\mu \in \mathbb{R}\right\}$, where τ^μ is the target value, is

frequently guided by the quadratic deviation of the network output from the target values (Hastie et al. 2001, Bishop 2006, Goodfellow et al. 2016). It serves as a cost function which evaluates the network performance with respect to a single example as

$$e^{\mu}\left(\{\boldsymbol{w}_k\}_{k=1}^{K}\right) = \frac{1}{2}\big(y^{\mu} - \tau^{\mu}\big)^2 \quad \text{with the shorthand } \; y^{\mu} = y(\boldsymbol{\xi}^{\mu}). \tag{2.5}$$

In stochastic or on-line gradient descent, updates of the weight vectors are based on the sequential presentation of single examples:

$$\boldsymbol{w}_k^{\mu} = \boldsymbol{w}_k^{\mu-1} + \frac{\eta}{N}\Delta\boldsymbol{w}_k^{\mu} \;\text{ with }\; \Delta\boldsymbol{w}_k^{\mu} = -\frac{\partial e^{\mu}}{\partial \boldsymbol{w}_k} = -\big(y^{\mu} - \tau^{\mu}\big)\frac{\partial y^{\mu}}{\partial \boldsymbol{w}_k} \tag{2.6}$$

where $\eta > 0$ is the learning rate and N is the number of dimensions of the input and weight vectors. The gradient is evaluated in $\{\boldsymbol{w}_k^{\mu-1}\}_{k=1}^{K}$. For the SCM architecture specified above we have

$$\frac{\partial y^{\mu}}{\partial \boldsymbol{w}_k} = g'\left(h_k^{\mu}\right)\boldsymbol{\xi}^{\mu}\,. \tag{2.7}$$

Hence, for the $\mathrm{erf}(\cdot)$ activation function the update applied to the weight vectors is proportional to:

$$\Delta\boldsymbol{w}_k^{\mu} = -\left(\sum_{i=1}^{K}\mathrm{erf}\left[\frac{1}{\sqrt{2}}h_i^{\mu}\right] - \tau^{\mu}\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(h_k^{\mu}\right)^2\right]\boldsymbol{\xi}^{\mu} \tag{2.8}$$

with the inner products $h_i^{\mu} = \boldsymbol{w}_i^{\mu-1}\cdot\boldsymbol{\xi}^{\mu}$ of the current weight vectors with the next example input in the stream. For ReLU activation the update applied to the weight vectors is proportional to:

$$\Delta\boldsymbol{w}_k^{\mu} = -\left(\sum_{i=1}^{K} h_i^{\mu}\Theta(h_i^{\mu}) - \tau^{\mu}\right)\Theta(h_k^{\mu})\,\boldsymbol{\xi}^{\mu}\,. \tag{2.9}$$

Note that the change of weight vectors is proportional to $\boldsymbol{\xi}^{\mu}$ and can be interpreted as a form of *Hebbian Learning* (Hastie et al. 2001, Bishop 2006, Goodfellow et al. 2016).

2.2.3 Student-Teacher Scenario and Model Data

In order to define and model meaningful learning situations we resort to the consideration of student-teacher scenarios (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993, Biehl and Caticha 2003). We assume that the regression

can be defined in terms of an SCM with a number M of hidden units and a specific set of weights $\left\{\boldsymbol{B}_m \in \mathbb{R}^N\right\}_{m=1}^M$:

$$\tau(\boldsymbol{\xi}) = \sum_{m=1}^{M} g(\boldsymbol{B}_m \cdot \boldsymbol{\xi}) \quad \text{and } \tau^\mu = \tau(\boldsymbol{\xi}^\mu) = \sum_{m=1}^{M} g(b_m^\mu) \tag{2.10}$$

with $b_m^\mu = \boldsymbol{B}_m \cdot \boldsymbol{\xi}^\mu$. This so-called teacher network can be equipped with $M > K$ hidden units in order to model regression schemes which cannot be learnt by an SCM student of the form (2.1); this learning scenario is therefore called *unrealizable*. On the contrary, $K > M$ would correspond to an *over-learnable* target or *over-sophisticated* student. For the discussion of these highly interesting cases in the context of the SCM with sigmoidal erf activation, see for instance (Biehl and Schwarze 1995, Saad and Solla 1995a, Saad and Solla 1995b, Riegler and Biehl 1995, Biehl et al. 1996). In a student-teacher scenario with K and M hidden units, the update of the student weight vectors by on-line gradient descent reads:

$$\boldsymbol{w}_k^\mu = \boldsymbol{w}_k^{\mu-1} - \frac{\eta}{N}\,\rho_k^\mu\,\boldsymbol{\xi}^\mu \;\text{ where }\; \rho_k^\mu = \left(\sum_{i=1}^{K} g(h_i^\mu) - \sum_{m=1}^{M} g(b_m^\mu)\right) g'(h_k^\mu) \tag{2.11}$$

with the quantities $b_m^\mu = \boldsymbol{B}_m \cdot \boldsymbol{\xi}^\mu$ and $h_k^\mu = \boldsymbol{w}_k^{\mu-1} \cdot \boldsymbol{\xi}^\mu$. In the equation above, the teacher output for the example $\boldsymbol{\xi}^\mu$ is used as the target. We will perform the analysis of the matching scenario $K = M$ as well as the $K \neq M$ cases. The equations for the erf and ReLU activation function are obtained by substituting in the equation above the definitions from Eq. (2.2) and Eq. (2.3), respectively.

The vectors $\boldsymbol{B}_m$ define the target output $\tau^\mu = \tau(\boldsymbol{\xi}^\mu)$ explicitly via the teacher network of Eq. (2.10) for any input vector. While clustered input densities can also be studied for feedforward networks as in (Meir 1995, Marangi et al. 1995), we assume here that the actual input vectors are uncorrelated with the teacher vectors $\boldsymbol{B}_m$. Consequently, we can resort to a simpler model density and consider vectors $\boldsymbol{\xi}$ of independent, zero mean, unit variance components with, e.g.,

$$P(\boldsymbol{\xi}) = \frac{1}{(2\,\pi)^{N/2}}\,\exp\left[-\frac{1}{2}\,\boldsymbol{\xi}^2\right]. \tag{2.12}$$

Note that the student-teacher scenario considered here is different from an equally named concept used in studies of *knowledge distillation*, see (Wang and Yoon 2021) and references therein. In the context of distillation, a teacher network is first trained on a given data set to approximate the target function. Thereafter a student network, frequently of a simpler architecture, distills the knowledge in a subsequent training process. In our work, as in most statistical physics based studies (Barkai et al. 1993, Engel and van den Broeck 2001, Watkin et al. 1993), the teacher network is taken

to directly define the true target function. A particular architecture is chosen and, together with its fixed weights, it controls the complexity of the task. The teacher network provides correct target outputs to all input data that are generated according to the distribution in Eq. (2.12). In the actual training process, a sequence of such input vectors and teacher-generated labels is presented to the student network.

2.2.4 Mathematical analysis of the on-line training dynamics

In the following we sketch the successful theory of on-line learning (Saad 1999, Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993, Biehl and Caticha 2003) as, for instance, applied to the dynamics of on-line gradient descent in the SCM in (Biehl and Schwarze 1995, Saad and Solla 1995a, Saad and Solla 1995b, Biehl et al. 1996, Riegler and Biehl 1995, Vicente and Caticha 1997, Inoue et al. 2003) or applied to the dynamics of LVQ algorithms in (Biehl et al. 2007, Biehl et al. 2005, Ghosh et al. 2005, Ghosh et al. 2006). The reader is referred to the original publications for details.

We consider learning systems with adaptive vectors $\boldsymbol{w}_i \in \mathbb{R}^N$ while the characteristic vectors $\boldsymbol{B}_j \in \mathbb{R}^N$ specify the target task that the learning system must implement. For the SCM, the adaptive vectors are student weight vectors and the characteristic vectors are teacher weight vectors. In Chapter 3, we treat the LVQ classifier in which the adaptive weights are the prototypes and the characteristic vectors are cluster centers of the input density.

The consideration of the *thermodynamic limit* $N \to \infty$ is instrumental for the theoretical treatment. The limit facilitates the following key steps which, eventually, yield an exact mathematical description of the training dynamics in terms of ordinary differential equations (ODE):

a) *Order parameters*
The many degrees of freedom, i.e. the components of the adaptive vectors, can be characterized in terms of only very few quantities. The definition of meaningful so-called *order parameters* follows naturally from the specific mathematical structure of the model. After presentation of a number μ of examples, as indicated by corresponding superscripts, we describe the system by the projections

$$R_{im}^{\mu} = \boldsymbol{w}_i^{\mu} \cdot \boldsymbol{B}_m \quad \text{and} \quad Q_{ik}^{\mu} = \boldsymbol{w}_i^{\mu} \cdot \boldsymbol{w}_k^{\mu} \quad \text{with } i,k \in \{1,...,K\}\,,\; m \in \{1,...,M\}. \tag{2.13}$$

Obviously, the K order parameters Q_{ii}^{μ} relate to the norms of the adaptive vectors and the $K(K-1)/2$ order parameters $Q_{ik}^{\mu} = Q_{ki}^{\mu}$ relate to the mutual

overlaps of the adaptive vectors. The KM quantities R_{im} specify their projections into the linear subspace defined by the characteristic vectors $\{\boldsymbol{B}_m\}_{m=1}^{M}$, respectively. The projections

$$T_{mn} = \boldsymbol{B}_m \cdot \boldsymbol{B}_n \tag{2.14}$$

relate to the norms and mutual overlaps of the teacher vectors that are characteristic for the target rule. In this chapter these vectors are fixed to $T_{mn} = \delta_{mn}$.

b) *Recursions*
For the order parameters in Eq. (2.13), recursion relations can be derived directly given the learning algorithm of the machine learning model which dictates the updates applied to the adaptive weight vectors. It is of the generic form:

$$\boldsymbol{w}_k^{\mu} = \boldsymbol{w}_k^{\mu-1} + \frac{\eta}{N}\Delta\boldsymbol{w}_k^{\mu} .$$

The corresponding inner products yield

$$\begin{aligned} \frac{R_{im}^{\mu} - R_{im}^{\mu-1}}{1/N} &= \eta\, \Delta\boldsymbol{w}_i^{\mu} \cdot \boldsymbol{B}_m \\ \frac{Q_{ik}^{\mu} - Q_{ik}^{\mu-1}}{1/N} &= \eta \left(\boldsymbol{w}_i^{\mu-1} \cdot \Delta\boldsymbol{w}_k^{\mu} + \boldsymbol{w}_k^{\mu-1} \cdot \Delta\boldsymbol{w}_i^{\mu} \right) + \frac{\eta^2}{N}\, \Delta\boldsymbol{w}_i^{\mu} \cdot \Delta\boldsymbol{w}_k^{\mu}. \end{aligned} \tag{2.15}$$

Terms of order $\mathcal{O}(1/N)$ on the r.h.s. of Eq. (2.15) will be neglected in the following. Note however that the inner products $\Delta\boldsymbol{w}_i^{\mu} \cdot \Delta\boldsymbol{w}_k^{\mu}$ comprise contributions of order $||\boldsymbol{\xi}||^2 \propto N$. Furthermore, in the limit of small learning rates $\eta \to 0$, the terms of order $\mathcal{O}(\eta^2)$ become negligible in comparison to the terms of order $\mathcal{O}(\eta)$.

c) *Averages over the Model Data*
Applying the central limit theorem (CLT) we can perform an average over the random sequence of independent examples. Note that $\Delta\boldsymbol{w}_k^{\mu} \propto \boldsymbol{\xi}^{\mu}$. Consequently, the current input $\boldsymbol{\xi}^{\mu}$ enters the r.h.s. of Eq. (2.15) only through its norm $||\boldsymbol{\xi}||^2 = \mathcal{O}(N)$ and the quantities

$$h_i^{\mu} = \boldsymbol{w}_i^{\mu-1} \cdot \boldsymbol{\xi}^{\mu} \text{ and } b_m^{\mu} = \boldsymbol{B}_m \cdot \boldsymbol{\xi}^{\mu} . \tag{2.16}$$

These inner products of the student- and teacher weight vectors with the input are commonly referred to as *pre-activations*, since the activation function of a unit receives these values as input. Since these inner products correspond to sums of many independent random quantities in our model, the CLT implies

that the projections in Eq. (2.16) are correlated Gaussian quantities for large N. Hence, their joint density $P(\boldsymbol{h}^\mu, \boldsymbol{b}^\mu)$, with $\boldsymbol{h}^\mu = \{h_i^\mu\}_{i=1}^K$ and $\boldsymbol{b}^\mu = \{b_m^\mu\}_{m=1}^M$, is given completely by first and second moments.

SCM: In the case of the isotropic, spherical input density of Eq. (2.12) the moments are

$$\begin{aligned} &\langle h_i^\mu \rangle = 0, \quad \langle b_m^\mu \rangle = 0, && \langle h_i^\mu h_k^\mu \rangle - \langle h_i^\mu \rangle \langle h_k^\mu \rangle = Q_{ik}^{\mu-1} \\ &\langle h_i^\mu b_m^\mu \rangle - \langle h_i^\mu \rangle \langle b_m^\mu \rangle = R_{im}^{\mu-1}, && \langle b_m^\mu b_n^\mu \rangle - \langle b_m^\mu \rangle \langle b_n^\mu \rangle = T_{mn} = \delta_{mn}\,. \end{aligned} \tag{2.17}$$

Hence, the mean of $(\boldsymbol{h}^\mu, \boldsymbol{b}^\mu)^T$ is the $(K+M)$-dimensional vector of zeroes and the covariance matrix is given as:

$$\mathcal{C} = \begin{bmatrix} \boldsymbol{Q}^{\mu-1} & \boldsymbol{R}^{\mu-1} \\ (\boldsymbol{R}^{\mu-1})^T & \boldsymbol{T} \end{bmatrix} \in \mathbb{R}^{(K+M)\times(K+M)}\,. \tag{2.18}$$

Subsequently, the joint density $P(\boldsymbol{h}^\mu, \boldsymbol{b}^\mu)$ is defined as:

$$P(\boldsymbol{h}^\mu, \boldsymbol{b}^\mu) = \frac{1}{\sqrt{(2\pi)^{K+M}|\mathcal{C}|}} \exp\left[-\frac{1}{2}(\boldsymbol{h}^\mu, \boldsymbol{b}^\mu)^T \mathcal{C}^{-1} (\boldsymbol{h}^\mu, \boldsymbol{b}^\mu)\right]. \tag{2.19}$$

Hence, the joint density $P(\boldsymbol{h}^\mu, \boldsymbol{b}^\mu)$ is fully specified by the values of the order parameters in the previous time step and the parameters of the model density. This important result enables us to perform an average of the recursion relations (2.15) over the latest training example in terms of Gaussian integrals over the density $P(\boldsymbol{h}^\mu, \boldsymbol{b}^\mu)$. Moreover, the resulting r.h.s. can be expressed in closed form in $\{R_{im}^{\mu-1}, Q_{ik}^{\mu-1}, T_{mn}\}$. Obviously, the precise form depends on the details of the algorithm and model setup.

d) *Self-Averaging Properties*
The self-averaging property of order parameters makes it possible to restrict the description to their mean values: Fluctuations of the stochastic dynamics can be neglected in the limit $N \to \infty$. This concept has been borrowed from the statistical physics of disordered materials and has been applied frequently in the study of neural network models and learning processes (Hertz et al. 1991, Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993). For a detailed mathematical discussion in the context of sequential on-line learning see (Reents and Urbanczik 1998). As a consequence, we can interpret the averaged equations (2.15) directly as deterministic recursions for the actual values of $\{R_{im}^\mu, Q_{ik}^\mu\}$, which coincide with their disorder average in the thermodynamic limit.

e) *Continuous Time Limit and ODE*
In the thermodynamic limit $N \to \infty$, ratios of the form $(\ldots)/(1/N)$ on the left hand sides of Eq. (2.15) can be interpreted as derivatives with respect to a continuous learning time α defined by

$$\alpha \;=\; \mu/N \text{ with } d\alpha \;\sim\; 1/N. \tag{2.20}$$

This scaling corresponds to the natural assumption that the number of examples should be proportional to the number of adaptive quantities in the system.

Averages are performed over the joint density $P(\boldsymbol{h}^\mu, \boldsymbol{b}^\mu)$ defined in Eq. (2.19) corresponding to the latest, independently drawn input vector. For simplicity, we omit indices μ in the following. The resulting sets of coupled ODE obtained from Eq. (2.15) are of the generic form:

$$\frac{dR_{im}}{d\alpha} = \eta F_{im} \quad \text{and} \quad \frac{dQ_{ik}}{d\alpha} = \eta G^{(1)}_{ik} + \eta^2 G^{(2)}_{ik}. \tag{2.21}$$

As mentioned before, when the dynamics are studied for the limit of small learning rate $\eta \to 0$, the term $\eta^2 G^{(2)}_{ik}$ can be neglected in Eq. (2.21). In order to retain non-trivial performance, the small step size has to be compensated for by training with a large number of examples that diverges like N/η. Formally, we introduce the quantity $\widetilde{\alpha}$ in the simultaneous limit

$$\widetilde{\alpha} \;=\; \lim_{\eta\to 0}\lim_{\alpha\to\infty}(\eta\alpha), \tag{2.22}$$

which leads to a simplified system of ODE

$$\frac{dR_{im}}{d\widetilde{\alpha}} = F_{im}\,;\; \frac{dQ_{ik}}{d\widetilde{\alpha}} = G^{(1)}_{ik} \tag{2.23}$$

in rescaled continuous time $\widetilde{\alpha}$ for $\eta \to 0$.

SCM: In the modelling of non-linear regression in a student-teacher scenario we obtain

$$F_{im} = \langle \rho_i b_m \rangle, \quad G^{(1)}_{ik} = \Big\langle\, (\rho_i h_k + \rho_k h_i) \,\Big\rangle \text{ and } G^{(2)}_{ik} = \langle \rho_i \rho_k \rangle \tag{2.24}$$

where the quantities ρ_i are defined in Eq. (2.11) for the latest input vector. The averages F_{im} and $G^{(1)}_{ik}$ consist of at most three-dimensional averages of the form:

$$I_3 = \langle\, g'(u)\, v\, g(w) \,\rangle\,. \tag{2.25}$$

The quantities $G^{(2)}_{ik}$ consist of at most four-dimensional averages of the form:

$$I_4 = \langle\, g'(u)\, g'(v)\, g(w)\, g(z) \,\rangle\,. \tag{2.26}$$

These averages are performed over the marginal distributions $P(u, v, w)$ and $P(u, v, w, z)$ of the joint density $P(\boldsymbol{h}, \boldsymbol{b})$ as defined in Eq. (2.19). For the Erf-SCM, i.e. $g(x) = \mathrm{erf}(x/\sqrt{2})$, the closed-form expressions for the averages of the forms (2.25) and (2.26) can be found in (Saad and Solla 1995b). In case of the ReLU-SCM with $g(x) = x\,\Theta(x)$, a closed-form expression for the averages of the form (2.25) reads:

$$\langle \Theta(u)\, v\, w\, \Theta(w) \rangle = \frac{\widehat{\mathcal{C}}_{12}\sqrt{\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{33} - \widehat{\mathcal{C}}_{13}^2}}{2\pi\widehat{\mathcal{C}}_{11}} + \frac{\widehat{\mathcal{C}}_{23}\sin^{-1}\left(\frac{\widehat{\mathcal{C}}_{13}}{\sqrt{\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{33}}}\right)}{2\pi} + \frac{\widehat{\mathcal{C}}_{23}}{4}\,, \quad (2.27)$$

where $\widehat{\mathcal{C}} \in \mathbb{R}^{3\times 3}$ is the covariance matrix of the marginal distribution $P(u, v, w)$. The derivation of the average can be approached in several ways, but perhaps the most succinct derivation is found in (Yoshida et al. 2017), where the integration is re-written to a three-dimensional orthant probability that has a known closed form solution. The averages of the form Eq. (2.26) posed mathematical difficulties for ReLU activation and can also not be approached in a similar way as in (Yoshida et al. 2017), because a closed form solution for the orthant probability in four dimensions does not exist. Hence, we resort to the ODE in Eq. (2.23) that is valid for the limit of small learning rates $\eta \to 0$.

f) *Generalization error*

After training, the success of learning is quantified in terms of the generalization error ϵ_g, which is also given as a function of the macroscopic order parameters.

SCM: In the regression scenario, the generalization error is defined as an average $\langle \cdots \rangle$ of the quadratic deviation between student and teacher output over the isotropic density, cf. Eq. (2.12):

$$\epsilon_g = \frac{1}{2}\left\langle \left[\sum_{k=1}^{K} g\left(h_k\right) - \sum_{m=1}^{M} g\left(b_m\right) \right]^2 \right\rangle . \quad (2.28)$$

In order to compute the full average of the r.h.s. of Eq. (2.28), averages of the form

$$I_2 = \langle g(u)\, g(v) \rangle \quad (2.29)$$

need to be evaluated. Again, these are evaluated over the marginal distributions $P(u, v)$ of the joint density $P(\boldsymbol{h}, \boldsymbol{b})$ defined in Eq. (2.19). For the Erf-SCM, the full form of the generalization error for arbitrary K and M can be found in (Saad and Solla 1995a, Saad and Solla 1995b). For the ReLU-SCM a closed-form

expression for the average in Eq. (2.29) reads:

$$\langle u\,\Theta(u)\,v\,\Theta(v)\rangle = \frac{\widehat{\mathcal{C}}_{12}}{4} + \frac{\sqrt{\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{22} - \widehat{\mathcal{C}}_{12}^2}}{2\pi} + \frac{\widehat{\mathcal{C}}_{12}}{2\pi}\sin^{-1}\left(\frac{\widehat{\mathcal{C}}_{12}}{\sqrt{\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{22}}}\right), \quad (2.30)$$

where $\widehat{\mathcal{C}} \in \mathbb{R}^{2\times 2}$ is the covariance matrix of the marginal distribution $P(u, v)$. The derivation of the above result can be found easily using the elegant formulation used in (Yoshida et al. 2017). Using the closed-form expression (2.30), the full form of the generalization error of the ReLU-SCM for arbitrary K and M is in Eq. (A.5). For a teacher SCM with orthonormal weight vectors as considered here, i.e. $T_{mn} = \delta_{mn}$, the full expression of the generalization error reduces to:

$$\begin{aligned}\epsilon_g = \frac{1}{2}\Bigg[&\sum_{i=1}^{K}\sum_{j=1}^{K}\left(\frac{Q_{ij}}{4} + \frac{\sqrt{Q_{ii}Q_{jj} - Q_{ij}^2}}{2\pi} + \frac{Q_{ij}\sin^{-1}\left(\frac{Q_{ij}}{\sqrt{Q_{ii}Q_{jj}}}\right)}{2\pi}\right) \\ &- 2\sum_{i=1}^{K}\sum_{m=1}^{M}\left(\frac{R_{im}}{4} + \frac{\sqrt{Q_{ii} - R_{im}^2}}{2\pi} + \frac{R_{im}\sin^{-1}\left(\frac{R_{im}}{\sqrt{Q_{ii}}}\right)}{2\pi}\right) \\ &+ \frac{M}{2} + \frac{(M-1)M}{2\pi}\Bigg].\end{aligned}$$

Learning curves

The (numerical) integration of the ODE for a given particular training algorithm, model density and specific initial conditions $\{R_{im}(0), Q_{ik}(0)\}$ yields the temporal evolution of order parameters in the course of training.

Exploiting the self-averaging properties of order parameters once more, we can obtain the learning curves $\epsilon_g(\alpha) = \epsilon_g\left(\{R_{im}(\alpha), Q_{ik}(\alpha)\}\right)$, i.e. the typical generalization error after on-line training with $(\alpha\, N)$ random examples.

2.2.5 Intial conditions in the ODE and simulations

In previous research it has been shown that the learning process in the SCM is characterized by the occurrence of quasi-stationary plateau states in which the student vectors are equally specialized to the teacher vectors (Saad and Solla 1995b, Biehl et al. 1996). These plateau states are caused by a weakly repulsive fixed points of the ODE. In Chapter 3 the plateau states are discussed in more detail and their occurrence in the presence of *concept drift* is studied. For the discussion here, we note that one of the factors that determines the length of the plateau is the amount of initial *student*

specialization towards the teacher vectors. We define the student specialization of student vector $\boldsymbol{w}_i$ as:

$$S_i = |R_{i1} - R_{i2}| \ . \tag{2.31}$$

In (Biehl et al. 1996) it was found that the length of the plateau is proportional to $-\log(S_i(0))$, where $S_i(0)$ is the initial student specialization. For $S_i(0) \to 0$, the repulsion of the fixed point decreases and for $S_i(0) = 0$ the fixed point is attractive. Hence, for initial conditions that satisfy $S_i(0) = 0$ for all $1 \leqslant i \leqslant K$ in the ODE, the system is not able to specialize. In the experiments concerning the SCM with multiple teacher units, it is therefore necessary to start with non-zero student specialization $S_i > 0$.

In order to compare the learning dynamics for $N \to \infty$ with finite N Monte Carlo simulations of the training process, it is useful for the simulations to initialize the student weight vectors $\{\boldsymbol{w}_i\}_{i=1}^K$ and the teacher weight vectors $\{\boldsymbol{B}_n\}_{n=1}^M$ to satisfy specific values of the order parameters R_{im}, Q_{ik} and T_{mn}. The algorithm provided in Algorithm 2.1 initializes vectors one-by-one, ensuring that each initialized vector satisfies given overlaps with the previously initialized vectors. This is similar to the Gram-Schmidt procedure for the orthogonalization of a set of linearly independent vectors, with the difference that with the provided algorithm the specified mutual overlaps between the vectors can be general instead of only orthogonal.

2.3 Experiments

We studied the macroscopic on-line learning dynamics for networks with ReLU activation in different settings:

1. $K = M = 1$: This setting corresponds to a student perceptron learning a rule defined by a teacher perceptron. Hence, the order parameters are R and Q describing the overlap between the student and teacher vector and the student norm, respectively. Since for this case the averages of the form (2.26) are at most two-dimensional (there are only two Gaussian variables h and b), they could be obtained analytically. Hence, the ODE were obtained for general learning rate η from Eq. (2.21) for $K = M = 1$.

   We studied the learning dynamics by numerically integrating the ODE for several learning rates $\eta = [0.1, 0.5, 0.8, 2.2]$ starting from initial conditions $(R(0), Q(0)) = (0, 0.25)$, which correspond to a random initialization of the student weights. Monte Carlo simulations of the training process for $N = 1000$ were also performed and compared to the results of the theoretical learning

Algorithm 2.1 Generalized Gram-Schmidt initialization

Input: Number of dimensions N, number of student vectors K, number of teacher vectors M.
Input: Order parameters R_{im}, Q_{ik} and T_{mn}, for $1 \leqslant i, k \leqslant K$ and $1 \leqslant m, n \leqslant M$.
Output: Vectors $\{\boldsymbol{w}_i \in \mathbb{R}^N\}_{i=1}^K$ and $\{\boldsymbol{B}_n \in \mathbb{R}^N\}_{n=1}^M$ satisfying $\boldsymbol{w}_i \cdot \boldsymbol{B}_m = R_{im}$, $\boldsymbol{w}_i \cdot \boldsymbol{w}_k = Q_{ik}$ and $\boldsymbol{B}_m \cdot \boldsymbol{B}_n = T_{mn}$, for $1 \leqslant i, k \leqslant K$ and $1 \leqslant m, n \leqslant M$.
for $i = 1, \ldots, K + M$ **do**
  $\boldsymbol{r}_i \leftarrow$ randomly generated vector from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \in \mathbb{R}^N$
  $\boldsymbol{r}_i \leftarrow \boldsymbol{r}_i / ||\boldsymbol{r}_i||$
end for
$\boldsymbol{B}_1 \leftarrow \sqrt{T_{11}}\boldsymbol{r}_1$
for $n = 2, \ldots, M$ **do**
  $\boldsymbol{c} \leftarrow$ vector with symbolic coefficients $\{c_i\}_{i=1}^{n-1}$
  $\hat{\boldsymbol{P}} \leftarrow$ matrix of previously initialized vectors, i.e., $(\{\boldsymbol{B}_m\}_{m=1}^{n-1}) \in \mathbb{R}^{N \times (n-1)}$
  $\boldsymbol{B}_n \leftarrow \hat{\boldsymbol{P}}\boldsymbol{c} + c_n \boldsymbol{r}_n$
  eqns $\leftarrow$ system of equations consisting of $T_{nn} = \boldsymbol{B}_n \cdot \boldsymbol{B}_n$ and $\{T_{mn} = \boldsymbol{B}_m \cdot \boldsymbol{B}_n\}_{m=1}^{n-1}$
  Solve eqns numerically for the coefficients $\boldsymbol{c}$ and c_n.
end for
for $j = 1, \ldots, K$ **do**
  $\boldsymbol{c} \leftarrow$ vector with symbolic coefficients $\{c_i\}_{i=1}^{M+j-1}$
  $\hat{\boldsymbol{P}} \leftarrow$ matrix of previously initialized vectors, i.e., $(\{\boldsymbol{B}_m\}_{m=1}^{M}, \{\boldsymbol{w}_k\}_{k=1}^{j-1},) \in \mathbb{R}^{N \times (M+j-1)}$
  $\boldsymbol{w}_j \leftarrow \hat{\boldsymbol{P}}\boldsymbol{c} + c_{M+j}\boldsymbol{r}_{M+j}$
  eqns $\leftarrow$ system of equations consisting of $Q_{jj} = \boldsymbol{w}_j \cdot \boldsymbol{w}_j$, $\{Q_{ij} = \boldsymbol{w}_i \cdot \boldsymbol{w}_j\}_{i=1}^{j-1}$ and $\{R_{jm} = \boldsymbol{w}_j \cdot \boldsymbol{B}_m\}_{m=1}^{M}$
  Solve eqns numerically for the coefficients $\boldsymbol{c}$ and c_{M+j}.
end for

dynamics for $N \to \infty$ obtained from the ODE. The weight vectors in the simulations were initialized using Algorithm 2.1 to satisfy the same initial conditions of the order parameters as the ones used in the ODE.

2. $K = M = 2$: In this setting, the student and the teacher SCM both have two hidden units and are therefore of matching complexity. The formulation of the dynamics for this learning scenario requires the evaluation of four-dimensional averages (2.26) for the Gaussian variables (h_1, h_2, b_1, b_2). Since we did not obtain this average analytically, the ODE describing the evolution for the seven

order parameters

$$R_{11},\, R_{12},\, R_{21},\, R_{22},\, Q_{11},\, Q_{12},\, Q_{22}\,, \tag{2.32}$$

was obtained for $\eta \to 0$ with Eq. (2.23). This was also necessary in settings 3 and 4. We numerically integrated the ODE starting from initial conditions

$$\begin{aligned} &R_{11}(0) = 10^{-3},\, R_{12}(0) = 0,\, R_{21}(0) = 0,\, R_{22}(0) = 10^{-3}, \\ &Q_{11}(0) = 0.2,\, Q_{12}(0) = 0,\, Q_{22}(0) = 0.3\,. \end{aligned} \tag{2.33}$$

Monte Carlo simulations of the training process for $N = 10000$ were also performed and compared to the results of the theoretical learning dynamics for $N \to \infty$. Since the $N \to \infty$ dynamics are formulated for learning rate $\eta \to 0$, we used a reasonably small learning rate $\eta = 0.1$ in the simulations to approximate this limit. The weight vectors were initialized using Algorithm 2.1 to satisfy the above initial conditions of the order parameters.

3. $K = 3, M = 2$: In this setting, the student SCM with three hidden units is over-parameterized compared to the teacher SCM with two hidden units. The equations of motion describing the evolution of the $KM + K(K+1)/2 = 12$ order parameters were obtained for $\eta \to 0$ using Eq. (2.23). The evolution of the order parameters was obtained by numerically integrating the equations of motion starting from zero-valued initial conditions except for:

$$R_{11}(0) = 10^{-3},\, Q_{11}(0) = 0.2,\, Q_{22}(0) = 0.3,\, Q_{33}(0) = 0.25\,. \tag{2.34}$$

4. $K = 2, M = 3$: In this setting the teacher SCM with three hidden units is more complex than the student SCM with two hidden units. The equations of motion describing the evolution of the $KM + K(K+1)/2 = 9$ order parameters were obtained for $\eta \to 0$ using Eq. (2.23). The evolution of the order parameters was obtained by numerically integrating the equations of motion starting from zero-valued initial conditions except for:

$$R_{11}(0) = 10^{-3},\, Q_{11}(0) = 0.2,\, Q_{22}(0) = 0.2,\, Q_{33}(0) = 0.2\,. \tag{2.35}$$

In all experiments the teacher vectors are fixed to

$$T_{mn} = \delta_{mn} \;\text{ for }\; 1 \leqslant m, n \leqslant M\,. \tag{2.36}$$

2.4 Results and Discussion

We first consider perceptron learning as described in setting 1. A numerical solution

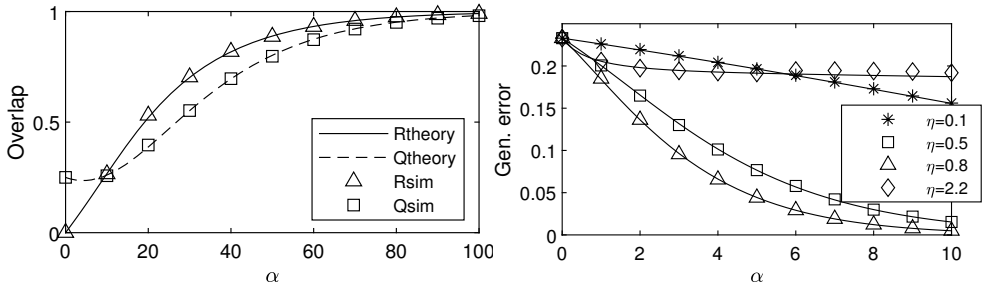


Figure 2.1: *Left panel*: Evolution of the order parameters R and Q for ReLU perceptron learning with gradient descent using a learning rate $\eta = 0.1$, starting from initial conditions $R(0) = 0$ and $Q(0) = 0.25$. *Right panel*: Evolution of the generalization error ϵ_g at the start of learning for different learning rates η. In both panels solid and dashed lines show the theoretical $N \to \infty$ results obtained from numerical integration of the ODE and symbols show Monte Carlo simulation results for $N = 1000$.

to the ODE system is shown in the left panel of Figure 2.1 for learning rate $\eta = 0.1$. It can be seen that the results obtained from Monte Carlo simulations had excellent agreement with the ODE results. In the initial stage of learning from $\alpha = 0$ to $\alpha \approx 10$, the process is characterized by a steep increase of the student-teacher overlap R while the student vector's norm hardly changes and shows a small dip. Hence, the initial increase in R is only caused by the decrease of the student vector's angle with the teacher vector. After $\alpha \approx 10$, the increase in Q indicates that the norm of the student vector increases in addition to the alignment of the student and teacher vector. For $\alpha \to \infty$ both R and Q increase towards $(R, Q) = (1, 1)$. This state corresponds to perfect learning of the rule in which the student vector is equal to the teacher vector, i.e., $\boldsymbol{w} = \boldsymbol{B}$. The point $(R, Q) = (1, 1)$ is a fixed point of the ODE for all meaningful learning rates, i.e. $\eta > 0$. Defining $(r, q) = (R - 1, Q - 1)$, the linearized dynamics dynamics around the fixed point read:

$$\begin{bmatrix} \frac{\partial r}{\partial \alpha} \\ \frac{\partial q}{\partial \alpha} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial}{\partial R}\frac{\partial R}{\partial \alpha} & \frac{\partial}{\partial Q}\frac{\partial R}{\partial \alpha} \\ \frac{\partial}{\partial R}\frac{\partial Q}{\partial \alpha} & \frac{\partial}{\partial Q}\frac{\partial Q}{\partial \alpha} \end{bmatrix}}_{\boldsymbol{A}(R=1,Q=1)} \begin{bmatrix} r \\ q \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{\eta}{2} & 0 \\ -\eta(\eta-1) & \eta\left(\frac{1}{2}\eta - 1\right) \end{bmatrix}}_{\boldsymbol{A}(R=1,Q=1)} \begin{bmatrix} r \\ q \end{bmatrix}, \quad (2.37)$$

where $\boldsymbol{A}(R = 1, Q = 1)$ is the Jacobian matrix of first derivatives of the non-linear differential equations evaluated in the fixed point. The eigenvalues of $\boldsymbol{A}(R = 1, Q =$

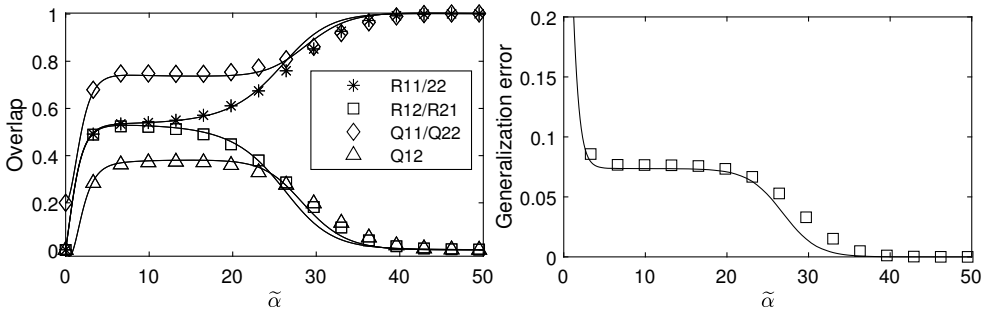


Figure 2.2: *Left*: Evolution of the order parameters for a student ReLU SCM with $K = 2$ hidden units learning a rule defined by a teacher ReLU SCM with $M = 2$ hidden units using gradient descent. *Right*: Evolution of the generalization error corresponding to the order parameters in the left panel. In both panels solid lines show the result obtained by numerically integrating the ODE and symbols show Monte Carlo simulation results for $N = 10000$ and $\eta = 0.1$.

1) determine the stability of the fixed point and are given by:

$$\lambda_1 = -\frac{\eta}{2}\,, \quad \lambda_2 = \eta\left(\frac{1}{2}\eta - 1\right)\,. \tag{2.38}$$

The eigenvalue λ_1 is always negative for $\eta > 0$ and λ_2 is negative for $0 < \eta < 2$. Therefore the fixed point is asymptotically stable for learning rates $\eta < 2$. For $\eta > 2$, λ_2 is positive and the fixed point becomes unstable. Experimentation reveals that there exists other stable fixed points of the dynamics which are approached for $\eta > 2$ with $\eta_g > 0$. At an even larger value of η, the norm of the student vector grows indefinitely.

We define the critical value of η where the transition of the fixed point stability from stable to unstable occurs as $\eta_c = 2$. Figure 2.1 (right) shows the evolution of ϵ_g for several values of η. Convergence is slow for $\eta << \eta_c$ but also for $\eta \approx \eta_c$. The Monte Carlo simulations agree with the theoretical results very well for all considered values of eta.

Figure 2.2 shows the results for setting 2: The dynamics for the ReLU network with $K = M = 2$. Monte Carlo simulation agree well with the theoretical results already for a reasonably small learning rate of $\eta = 0.1$. The learning process is characterized by a suboptimal plateau during which the student vectors are not specialized towards specific teacher vectors and therefore the specialization quantity defined in Eq. (2.31) evaluates to $S_1 \approx 0$ and $S_2 \approx 0$. The fixed point in the ODE that

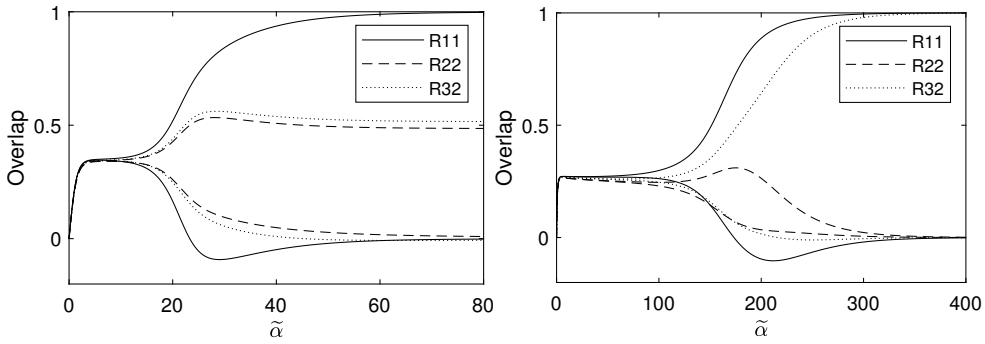


Figure 2.3: Evolution of student-teacher order parameters for overrealizable case with $K = 3$ hidden units in the student SCM and $M = 2$ hidden units in the teacher SCM. *Left*: ReLU-SCM. *Right*: Erf-SCM. The correlation of each student unit to the teacher units is shown shown by the style of the curves. A pair of the same style of curves shows the correlation of the student unit to each of the two teacher units. The legends point to the upper curve of the pair.

causes the observed plateau is found numerically:

$$R_{in} \approx 0.52 \text{ for } 1 \leqslant i \leqslant 2,\ 1 \leqslant n \leqslant 2\,, \tag{2.39}$$

$$Q_{11} \approx 0.72\,, Q_{22} \approx 0.72\,, Q_{12} \approx 0.38\,. \tag{2.40}$$

In contrast to the results for the erf SCM (Biehl and Schwarze 1995, Saad and Solla 1995b), the student vectors are not identical during the plateau. Similar to the perceptron case, we formulate the Jacobian and find its eigenvalues numerically to determine the stability of the fixed point. One positive eigenvalue is positive and guides the repulsion away from the fixed point:

$$\lambda_5 \approx 0.24 \text{ with eigenvector } \boldsymbol{u}_5 = (0.5, -0.5, -0.5, 0.5, 0, 0, 0)^T\,. \tag{2.41}$$

It indicates symmetry breaking: The increase of R_{11} and R_{22} and the decrease of R_{12} and R_{21}. Therefore, at the end of the plateau the learning algorithm starts to position student vector $\boldsymbol{w}_1$ into the direction of teacher vector $\boldsymbol{B}_1$ and student vector $\boldsymbol{w}_2$ into the direction of teacher vector $\boldsymbol{B}_2$. Shortly after the symmetry breaking, it can be observed that the norms of the student vectors increase further. The onset of specialization is associated with a decrease in generalization error, see Figure 2.2 (right).

In Figure 2.3, numerical integration results for the overparameterized student SCM of setting 3 are shown for ReLU activation (left) and sigmoidal Erf activation (right). In both cases, student vector $\boldsymbol{w}_1$ specializes to teacher vector $\boldsymbol{B}_1$ and align

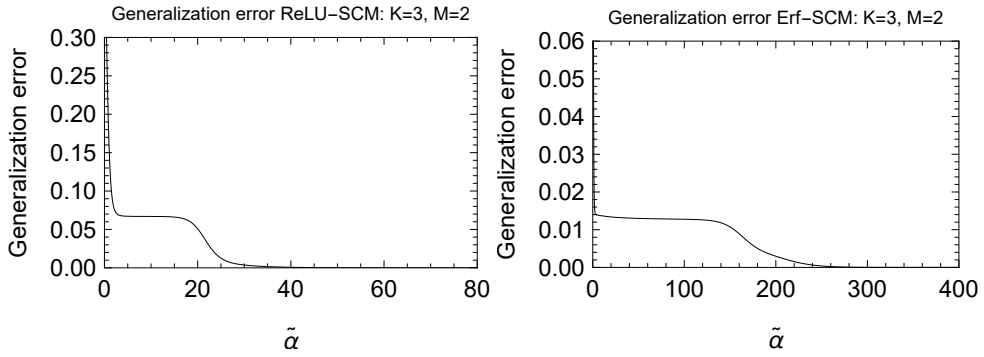


Figure 2.4: Evolution of the generalization error obtained in the overrealizable setting with $K = 3$ hidden neurons in the student SCM and $K = 2$ hidden neurons in the teacher SCM. *Left panel*: ReLU-SCM. *Right panel*: Erf-SCM.

	$Q_{11}(\infty)$	$Q_{12}(\infty)$	$Q_{13}(\infty)$	$Q_{22}(\infty)$	$Q_{23}(\infty)$	$Q_{33}(\infty)$
ReLU	1.00	0.00	0.00	0.24	0.25	0.27
Erf	1.00	0.00	0.00	0.00	0.00	1.00

Table 2.1: Asymptotic values of the student-student overlaps Q_{ik} for the over-parameterized scenario corresponding to the same setting and ODE integration as the results in Figure 2.3.

fully with it, which is caused by the specialization given in the initial conditions. In the ReLU case, student vectors $\boldsymbol{w}_2$ and $\boldsymbol{w}_3$ achieve a similar overlap with teacher vector $\boldsymbol{B}_2$. From the values of Q_{22}, Q_{23} and Q_{33} obtained for large $\tilde{\alpha}$ as shown in Table 2.1, it follows that the student vectors $\boldsymbol{w}_2$ and $\boldsymbol{w}_3$ have become nearly identical. Moreover, both vectors have become practically fully aligned with teacher vector $\boldsymbol{B}_2$, which follows from the numerical values of the order parameters at $\tilde{\alpha} = 80$:

$$\frac{R_{22}}{\sqrt{Q_{22}}} = \cos\left(\phi(\boldsymbol{w}_2, \boldsymbol{B}_2)\right) \approx 1\,, \quad \frac{R_{32}}{\sqrt{Q_{33}}} = \cos\left(\phi(\boldsymbol{w}_3, \boldsymbol{B}_2)\right) \approx 1\,. \tag{2.42}$$

The generalization error that corresponds to the order parameters of Fig. 2.1 is shown in Fig. 2.4, left panel. For large α, the generalization error approaches zero. The reason that this configuration of order parameters results in perfect learning of the rule is that the ReLU is a piece-wise linear function, for which we have:

$$\mathrm{ReLU}(a\,u) = \mathrm{ReLU}(b\,u) + \mathrm{ReLU}(c\,u)\,, \quad \text{for } \; a = b + c \;\text{and}\; \operatorname{sign} a = \operatorname{sign} b = \operatorname{sign} c\,, \tag{2.43}$$

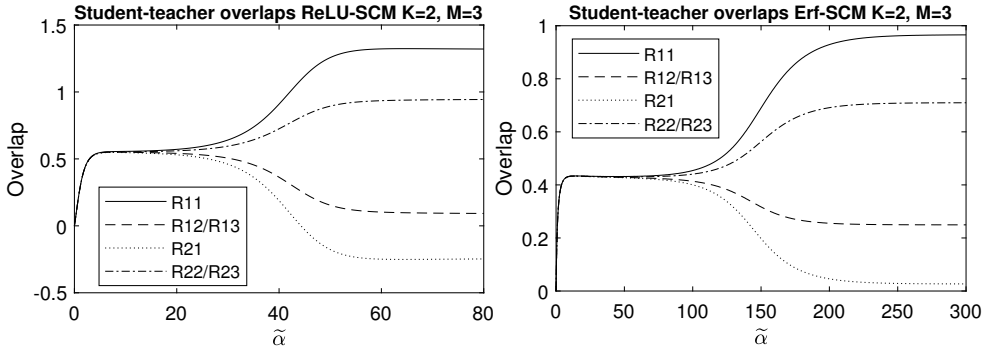


Figure 2.5: Evolution of student-teacher order parameters for the unrealizable case with $K = 2$ hidden units in the student SCM and $M = 3$ hidden units in the teacher SCM. *Left*: ReLU-SCM. *Right*: Erf-SCM.

and equivalently for the inner field inputs to the units in our case:

$$\begin{aligned} \mathrm{ReLU}(\boldsymbol{B}_2 \cdot \boldsymbol{\xi}) = \mathrm{ReLU}(\boldsymbol{w}_2 \cdot \boldsymbol{\xi}) + \mathrm{ReLU}(\boldsymbol{w}_3 \cdot \boldsymbol{\xi}) \\ \iff \\ (\boldsymbol{w}_2 = a\boldsymbol{B}_2) \wedge (\boldsymbol{w}_3 = b\boldsymbol{B}_2) \wedge (a + b = 1) \wedge a \geqslant 0 \wedge b \geqslant 0\,. \end{aligned} \tag{2.44}$$

Therefore, during training, two units of the ReLU student SCM aligned to one unit of the teacher SCM. These two student vectors were scaled to exactly reproduce the teacher vector. In this case all non-negative scale factors a and b with $a + b = 1$ would yield a student SCM that perfectly represents the rule. Which particular scale factors are achieved is dependent on the initial conditions.

Different behavior is observed for this training setting with the Erf-SCM as shown in the right panel of Fig. 2.3. Similar to the ReLU-SCM, after the plateau the second and third student unit compete for the specialization in the second teacher unit. However, in this case the third student unit specializes more quickly and eventually approaches full specialization $S_3 \to 1$. The second student unit loses its specialization and its weights approach the zero-vector, i.e. $\boldsymbol{w}_2 \to \boldsymbol{0}$, as indicated by $Q_{22}(\alpha \to \infty) = 0$ shown in Table. 2.1. The second student unit is therefore effectively removed from the network and the generalization error approaches zero for large $\widetilde{\alpha}$, as shown in the right panel of Fig. 2.4. There is a clear explanation for this behavior: It's not possible for the non-linear erf units' weight vectors to linearly combine and thereby fully represent a teacher unit. The superfluous complexity in the student Erf-SCM has to be removed from the network by shrinking the redundant weight vectors to zero.

In Figure 2.5 and Table 2.2, the results of setting 4 are shown: The unrealizable setting with $K = 2$ hidden units in the student and $M = 3$ hidden units in the teacher.

	$Q_{11}(\infty)$	$Q_{12}(\infty)$	$Q_{22}(\infty)$
ReLU	1.76	-0.15	1.84
Erf	1.06	0.38	1.01

Table 2.2: Asymptotic values of the student-student overlaps Q_{ik} for the unrealizable setting. The results correspond to the ODE integration of Figure 2.5.

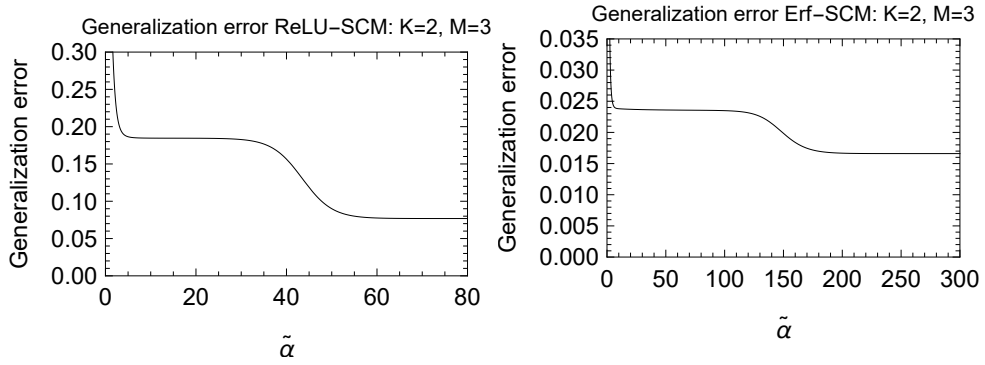


Figure 2.6: Evolution of the generalization error obtained in the unrealizable setting with $K = 2$ hidden neurons in the student SCM and $K = 3$ hidden neurons in the teacher SCM. *Left panel*: ReLU-SCM. *Right panel*: Erf-SCM.

For the ReLU (left panel), student vector $\boldsymbol{w}_1$ mainly specializes to teacher vector $\boldsymbol{B}_1$, almost achieving full alignment. A small overlap with teacher vector $\boldsymbol{B}_2$ and $\boldsymbol{B}_3$ remains. Student vector $\boldsymbol{w}_2$ achieves significant overlap with the second and third student vector $\boldsymbol{B}_2$ and $\boldsymbol{B}_3$. In fact, from the numerical values of the order parameters at $\widetilde{\alpha} = 80$, we obtain:

$$\frac{R_{22}}{\sqrt{Q_{22}}} = \cos\left(\phi(\boldsymbol{w}_2, \boldsymbol{B}_2)\right) \approx \frac{\pi}{4}\,, \quad \frac{R_{23}}{\sqrt{Q_{22}}} = \cos\left(\phi(\boldsymbol{w}_2, \boldsymbol{B}_3)\right) \approx \frac{\pi}{4}\,. \tag{2.45}$$

Therefore the second student unit approximates a superposition of the second and the third teacher units. Since the student SCM cannot not realize the rule, the generalization error does not approach zero, as shown in the left panel of Fig. 2.6.

The learning behavior of the Erf-SCM is very similar in this case; the student vectors also represent a combination of the teacher vectors. The generalization error is shown in the right panel of Figure 2.6.

2.5 Conclusion and Future Work

In this chapter we first introduced a generic modelling framework for the analysis of learning dynamics based on techniques from statistical physics that are already successfully used in previous studies. The modelling framework uses the concept of a student- and teacher model, in which a learning algorithm adapts the student model based on the available data and corresponding target labels provided by the teacher model. A few order parameters can be defined and quantify the state of learning. For a data distribution with an infinitely large number of independent dimensions, the evolution of the order parameters becomes deterministic. By averaging the learning equations, a system of ordinary differential equations can be defined.

Using the modelling framework, we studied gradient descent learning of the Soft Committee Machine. The equations are formulated for a student-teacher scenario with an arbitrary number of hidden units in the student and the teacher model and an arbitrary choice of activation function in the hidden units. In order to study the learning dynamics for the popular ReLU function, we derived closed-form expressions for the required averages in the case of small learning rates. This allows the formulation of various on-line gradient descent learning settings of the ReLU-SCM model.

Using the obtained formulation for the ReLU-SCM and the formulation for the Erf-SCM from the literature, we studied the learning dynamics in a number of settings: 1) For the ReLU perceptron learning we showed the learning behavior for different values of the learning rate and analytically derived the maximum learning rate for which perfect learning is achieved. 2) In a matching scenario of SCM learning where the student and teacher both had two hidden units, the characteristic plateau was studied analytically and the specialization of student units was discussed. In contrast to the Erf-SCM, in the ReLU-SCM a plateau is favored that corresponds to an unspecialized configuration in which the student vectors are different. In both 1) and 2), Monte Carlo simulations had good correspondence with the theory. 3) We showed the existence of a qualitative difference between the ReLU-SCM and the Erf-SCM in an overrealizable setting. The learning algorithm can combine ReLU units by aligning the directions of the weight vectors and scaling. The extra complexity of the Erf-SCM has to be eliminated by shrinking the redundant weights to zero. In both settings, perfect learning is achieved. In the unrealizable case, both SCM approximate the rule by correlating with multiple weight vectors.

Future works should address the learning dynamics of ReLU networks for larger learning rates and learning rate adaptation schemes. This can be done by finding a closed-form expression for the required four-dimensional averages. If a closed-form expression cannot be obtained, an approximation of the ReLU function may be used.

One possibility is to consider a generalization of the so-called Gaussian Error Linear Unit (GELU) activation function (Hendrycks and Gimpel 2016)

$$\begin{aligned} &\mathrm{GELU}(x,\gamma) = x\Phi(\gamma x) = \frac{1}{2}x\left(1+\mathrm{erf}\left(\frac{\gamma x}{\sqrt{2}}\right)\right)\,, \\ &\lim_{\gamma\to\infty}\mathrm{GELU}(x,\gamma) = \lim_{\gamma\to\infty}\left(x\Phi(\gamma x)\right) = x\Theta(x) = \mathrm{ReLU}(x)\,, \end{aligned} \tag{2.46}$$

where $\Phi(\cdot)$ is the standard Gaussian's CDF and γ a scaling factor.

Besides the potential of simplifying the derivation of the equations for the ReLU, studying the $\mathrm{GELU}(x,\gamma)$ for $\gamma = O(1)$ is highly relevant, as the activation function and the similarly shaped version $g(x) = x\sigma(\gamma x)$ has been found in (Ramachandran et al. 2017, Eger et al. 2018) to perform well on a variety of tasks and has gained in popularity in deep learning applications.

To further simplify the derivations, the error function could be closely approximated by (Tsay et al. 2013):

$$\mathrm{erf}(x) = \begin{cases} 1-\exp(-c_1x-c_2x^2) & \text{for } x \geqslant 0 \\ -1+\exp(c_1x-c_2x^2) & \text{for } x < 0 \end{cases}\,. \tag{2.47}$$

New activation functions with desirable properties could be designed using this approach.

Another important direction for future studies is the extension of the theory to more hidden layers. A starting point could be the consideration of deep tree-like neural networks. In these networks the receptive field of each neuron in the receiving layer is a dedicated non-overlapping part of the previous layer's activations. This property is maintained throughout the network. Hence, for independent inputs the units in every layer of the network are mutually independent, so that the central limit theorem applies to the pre-activations in the subsequent layer.

Furthermore, we emphasize the broad applicability of the presented modelling framework in the analysis of a variety of machine learning situations. As one example, in the next chapter the modelling framework is extended for the analysis of typical learning behaviour of learning vector quantization and neural networks in non-stationary learning settings.

Published as:

M. Straat, F. Abadi, C. Göpfert, B. Hammer and M. Biehl – *"Statistical Mechanics of On-line Learning Under Concept Drift"*, Entropy, vol. 20, no. 10, art. no. 775, 2018.

M. Straat, F. Abadi, Z. Kan, C. Göpfert, B. Hammer and M. Biehl – *"Supervised learning in the presence of concept drift: a modelling framework"*, Neural Computing and Applications (2022) 34:101-118, DOI: 10.1007/s00521-021-06035-1, 2021.

Chapter 3

Learning under concept drift

Abstract

In this chapter we extend the modelling framework of the previous chapter to incorporate supervised learning in non-stationary environments. Specifically, we model two example types of learning systems: prototype-based Learning Vector Quantization (LVQ) for classification and shallow, layered neural networks for regression tasks. We investigate so-called student teacher scenarios in which the systems are trained from a stream of high-dimensional, labeled data. Properties of the target task are considered to be non-stationary due to drift processes while the training is performed. Different types of concept drift are studied, which affect the density of example inputs only (virtual drift), the target rule itself (real drift), or both. Furthermore, we introduce weight decay as an explicit mechanism of forgetting. The required extensions of the modelling framework to incorporate these effects are shown. We then obtain the systems of ODE that describe the typical learning behavior of the two models under the different types of drift and weight decay. Our results show that standard LVQ algorithms are already suitable for the training in non-stationary environments to a certain extent. We show that the application of weight decay is effective for increasing the performance under real drift processes. On the other hand, a clear benefit of weight decay can not be confirmed under drifting class biases. In the investigation of gradient-based training of layered neural networks, we focus on the comparison of the use of sigmoidal- and Rectified Linear Unit (ReLU) activation functions. It is shown that concept drift can cause the persistence of sub-optimal plateau states in the evolution of the weights in the networks. Furthermore, we show that the sensitivity to concept drift and the effectiveness of weight decay differs remarkably between the two types of activation function: For instance, we find that the plateau lengths in the learning curves of ReLU networks can be significantly shortened by the weight decay.

3.1 Introduction

In this chapter, we address a topic which is currently attracting increasing interest in the scientific community: the efficient training of machine learning systems in a non-stationary environment, where the target task or the statistical properties of the example data vary with time, see for instance (Zliobaite et al. 2016, Losing et al. 2017, Ditzler et al. 2015, Joshi and Kulkarni 2012, Ade and Desmukh 2013, Morales and Bifet 2015) and references therein. Terms like *continual learning* or *lifelong learning* have been coined in this context.

Frequently, the set-up of machine learning processes comprises two different stages, see for instance (Hastie et al. 2001, Bishop 2006, Goodfellow et al. 2016): In the *training phase,* a given set of example data is analyzed, information is extracted and a corresponding hypothesis is parameterized in terms of, say, a classifier or regression system. In the subsequent *working phase,* this hypothesis is applied to novel data. Implicitly, one assumes that the training set is representative of the problem and that statistical properties of the data and the actual target task do not change after training.

For many practical applications of machine learning the assumption of stationarity may be well justified. However, the conceptual and temporal separation of training and working phase is not very plausible in human and other biological learning processes (Grandinetti et al. 2014, Amunts et al. 2014), in which learning and applying are continuously intertwined. As an example, in a *predator and prey* system, strategies can change continuously with species trying to adapt to their adversaries' behavior. Also in many technical applications of machine learning the separation becomes inappropriate if the actual task of learning, e.g. the target classification, changes over time (Zliobaite et al. 2016). Moreover, very frequently the training samples become available in the form of a non-stationary stream of data, e.g. (Losing et al. 2017, Ditzler et al. 2015, Joshi and Kulkarni 2012, Ade and Desmukh 2013). In such situations, the learning system must be able to detect and track concept drift, i.e. forget irrelevant, older information while continuously adapting to more recent inputs. Examples for this situation can be found, for instance, in robotics. Other problems like the filtering of *spam messages* in e-mail communication, resemble the *predator prey* example in that the learning systems try to adapt to changing strategies of their *opponents*. Further applications range from fraud detection, quality control and customer segments management to drop out prediction for e-learning and gaming (Zliobaite et al. 2016). Overviews of earlier work and recent developments in the context of machine learning in non-stationary environments are provided in (Zliobaite et al. 2016, Losing et al. 2017, Ditzler et al. 2015, Joshi and Kulkarni 2012, Ade and Desmukh 2013, Morales and Bifet 2015), for instance. While drift can occur

in any learning scenario, in this contribution, we will focus on supervised learning.

In the literature, two major types of non-stationary environments have been discussed (Zliobaite et al. 2016, Losing et al. 2017, Ditzler et al. 2015, Joshi and Kulkarni 2012, Ade and Desmukh 2013, Morales and Bifet 2015): In so-called *virtual drifts*, the statistical properties of the available example data change with time, while the actual target task, e.g. the classification or regression scheme, remains unaltered. The term *real drift* has been coined for situations in which the target itself is time-dependent. Frequently, both effects coincide and a clear distinction of the two cases becomes difficult.

3.1.1 Models of On-Line Learning Under Concept Drift

There exists a large variety of technologies which address learning in the context of drift, see (Zliobaite et al. 2016, Losing et al. 2017, Ditzler et al. 2015, Joshi and Kulkarni 2012, Ade and Desmukh 2013) for overviews of earlier work and more recent developments in the context of non-stationary learning. On a global level, one often differentiates so-called *active* methods, which aim for an explicit detection of drift and according action of the learning system, and *passive* methods, which can implicitly react to drift by their design.

Popular active methods combine statistical tests for novelty detection (Faria et al. 2016) with a rearrangement or retraining of the system to account for the observed drift. The latter is particularly efficient if, for instance, ensemble methods are used (Krawczyk et al. 2017, Gomes et al. 2017). The need for explicit drift detection often has the consequence that only specific types of drift can be dealt with (one exception being e.g. (Gomes et al. 2017)). In particular, small gradual drifts are notoriously difficult to detect (Losing et al. 2018).

Passive methods continuously adapt the model according to the given data. Thus, they automatically react to all types of drift which are present in the training data. The presence of drift requires some form of *forgetting* of dated information while the system is adapted to more recent observations. Yet, these passive methods face the classical stability-plasticity dilemma: relevant novel information has to be dealt with while preserving already learned signals. Local or hybrid schemes have been particularly successful in the past years, see e.g. (Losing et al. 2018, Loeffel et al. 2015). Other popular passive technologies rely on online learning schemes, in particular online gradient descent, which has been incorporated into drift learning strategies for the simple perceptron, neural networks, or extreme learning machines, as an example (Benczúr et al. 2018, Janakiraman et al. 2016). The behavior of such models varies extensively across different learning scenarios (Losing et al. 2017).

In this contribution, we study two basic scenarios of on-line learning in non-

stationary environments, addressing binary classification and continuous regression problems. We present a mathematical model of drifting concepts in on-line training from high-dimensional data. The design of useful, forgetful training schemes hinges on an adequate theoretical understanding of the relevant phenomena. To this end, the development of a suitable modelling framework is instrumental. Methods borrowed from statistical physics facilitate the study of the typical learning dynamics for different training scenarios and strategies. Here we extend the statistical physics based modelling framework of on-line learning in stationary settings as given in Chapter 2 to cover non-stationary learning contexts and weight decay as a mechanism of forgetting. Specifically, we use the modelling framework to analyze gradient based learning in non-stationary situations in prototype-based binary classification and continuous regression with feedforward neural networks. Both virtual and real drift processes are addressed.

Learning Vector Quantization (LVQ) is a prototype-based learning system originally suggested by Kohonen (Kohonen et al. 1988, Kohonen 2001, Kohonen 1990, Nova and Estevez 2014, Biehl et al. 2016). LVQ training is most frequently performed in an on-line setting by presenting a sequence of single examples which are used to improve the system iteratively (Nova and Estevez 2014, Biehl et al. 2016). Therefore, LVQ should constitute a promising framework for incremental learning in the presence of concept drift.

Layered neural networks with sigmoidal- and Rectified Linear Unit (ReLU) activation functions serve as example systems in the context of regression. Specifically, we consider the so-called *Soft Committee Machine* (SCM) as formulated and studied in stationary settings in Chapter 2. This *shallow* architecture can be trained by means of on-line (stochastic) gradient descent (Biehl and Schwarze 1995, Saad and Solla 1995a, Saad and Solla 1995b, Biehl et al. 1996, Riegler and Biehl 1995, Vicente and Caticha 1997, Inoue et al. 2003). Gradient based techniques are widely used also for multi-layered *deep* architectures and their suitability for the learning of non-stationary targets is a question of significant relevance (Goodfellow et al. 2016, Marcus 2018). For the SCM model in the context of concept drift, the emphasis is on the comparison of sigmoidal and ReLU activation with respect to the sensitivity to drift and the effect of weight decay.

The reason for the selection of these two learning systems is that they are representatives of important paradigms in machine learning. Therefore, the analysis of these systems also provides an example of the use of the workshop in which to develop modelling techniques and analytical approaches that will facilitate the study of other learning systems and setups in the future.

3.1.2 Relation to earlier Work

Stationary model densities of clustered data, similar to the ones considered here for LVQ, have been studied with respect to several unsupervised and supervised training schemes, see (Biehl et al. 1997, Biehl, Freking, Reents and Schlösser 1998, Biehl and Schlösser 1998, Barkai et al. 1993, Marangi et al. 1995, Meir 1995) for examples and further references. Supervised LVQ training was considered more recently in the framework of simplifying model situations in (Biehl et al. 2007, Ghosh et al. 2006, Biehl et al. 2005, Ghosh et al. 2005, Witoelar et al. 2010).

The presence of concept drift has also been addressed within the statistical physics of on-line learning. In particular, the learning of time-dependent, linearly separable rules served as a model system in (Biehl and Schwarze 1992, Biehl and Schwarze 1993, Kinouchi and Caticha 1993, Vicente and Caticha 1998). Note, that the assumption of statistically independent examples in the stream of data does not hinder the study of meaningful drift scenarios. It is, for instance, well possible to consider settings in which the characteristics of the generating density or the target itself depends, implicitly, on the previous training. As an example, *adversarial drifts* have been considered in (Biehl and Schwarze 1992, Biehl and Schwarze 1993, Kinouchi and Caticha 1993, Vicente and Caticha 1998) for the simple perceptron.

To the best of our knowledge, we present here the first statistical mechanics analysis of on-line learning under concept drift in prototype-based classification and layered neural networks for regression.

3.1.3 Outline

This chapter is organized as follows:

In Section 3.2 the LVQ learning system is introduced briefly, together with the model update rule and the assumed data density that we use in the modelling. In this section, we also provide the expressions of the quantities that are required for the formulation of the macroscopic learning dynamics and the generalization error according to the modelling framework. The same order of steps in the derivation is used as in the derivation of the dynamics for the SCM in Chapter 2.

Section 3.3 discusses the mathematical modelling of the considered drifts and the incorporation of weight decay as an explicit mechanism of *forgetting*. We then discuss the extension of the modelling framework in order to include the drifts and weight decay, which is identical for both learning systems.

The results of our analyses are presented in Sec. 3.4, which exemplify and demonstrate the usefulness of the methodological approach: We obtain insights into the ability of prototype-based systems to track a time-varying classification scheme and

changing class-wise prior probabilities. Furthermore, in this section we show the results of real concept drift on regression systems trained by gradient-based methods.

In Sec. 3.5 we conclude with a general discussion and outlook on future work.

3.2 Model and Methods

We first introduce Learning Vector Quantization for classification tasks with emphasis on the well-established heuristic LVQ1 training scheme. We furthermore introduce a suitable, clustered density of input data which is taken to define the target task in the model and formulate the typical learning dynamics.

3.2.1 Learning Vector Quantization

Learning Vector Quantization constitutes a family of prototype-based algorithms which are used in a wide variety of practical classification problems (Kohonen et al. 1988, Kohonen 1990, Nova and Estevez 2014, Biehl et al. 2016). The popularity of the approach is due to a number of attractive features: LVQ procedures are easy to implement, very flexible and intuitive. Moreover, it constitutes a natural tool for multi-class problems. The actual classification scheme is very often based on Euclidean metrics or other simple measures, which quantify the distance of inputs or feature vectors from the class-specific prototypes. In contrast to the *black-box* character of many less transparent methods, LVQ facilitates straightforward interpretations of the classifier since the prototype vectors are embedded in the actual feature space of the data and directly parameterize the classifier (Nova and Estevez 2014, Biehl et al. 2016). The approach is based on the idea of representing classes by more or less typical representatives of the classes among the training instances.

3.2.2 Nearest Prototype Classifier and generic training rule

In general, several prototypes can be employed to represent each class. However, we restrict the analysis to the simple case of only one prototype per class in binary classification problems. Hence we consider two prototypes $\boldsymbol{w}_k \in \mathbb{R}^N$ in total, where prototype k is supposed to represent the data from class $k \in \{1, 2\}$. Together with a distance measure $d(\boldsymbol{w}, \boldsymbol{\xi})$, the system parameterizes a Nearest Prototype Classification (NPC) scheme: Any given input $\boldsymbol{\xi} \in \mathbb{R}^N$ is assigned to the class of the closest prototype, i.e. it is assigned to class 1 if $d(\boldsymbol{w}_1, \boldsymbol{\xi}) < d(\boldsymbol{w}_2, \boldsymbol{\xi})$ and to class 2, otherwise. In practice, ties can be broken arbitrarily.

A variety of distance measures have been used in LVQ, enhancing the flexibility of the approach even further (Biehl et al. 2016, Nova and Estevez 2014). This includes

the conceptually interesting use of adaptive metrics in *relevance learning*, see (Biehl et al. 2016, Biehl et al. 2014) and references therein. Throughout the following, we restrict our analysis to the simple (squared) Euclidean measure

$$d(\boldsymbol{w}, \boldsymbol{\xi}) = (\boldsymbol{w} - \boldsymbol{\xi})^2. \tag{3.1}$$

We assume that in the training process, a sequence of single examples is presented to the system (Biehl and Caticha 2003): At time step $\mu = 1, 2, \ldots,$ the data point $\boldsymbol{\xi}^\mu$ is presented, together with its class label $\sigma^\mu = 1, 2$. Iterative on-line LVQ updates are of the general form (Biehl et al. 2007, Witoelar et al. 2007, Ghosh et al. 2006)

$$\boldsymbol{w}_k^\mu = \boldsymbol{w}_k^{\mu-1} + \frac{\eta}{N}\Delta\boldsymbol{w}_k^\mu \text{ with } \Delta\boldsymbol{w}_k^\mu = f_k\left[d_1^\mu, d_2^\mu, \sigma^\mu, \ldots\right]\left(\boldsymbol{\xi}^\mu - \boldsymbol{w}_k^{\mu-1}\right) \tag{3.2}$$

where $d_i^\mu = d(\boldsymbol{w}_i^{\mu-1}, \boldsymbol{\xi}^\mu)$ is the distance between the data point and the prototype of class i. The learning rate η is scaled with the input dimension N. The precise algorithm is specified by the choice of the *modulation function* $f_k[\ldots]$, which depends typically on the Euclidean distances of the data point from the current prototype positions and on the labels $k, \sigma^\mu = 1, 2$ of the prototype and the training example, respectively.

3.2.3 The LVQ1 training algorithm

A popular and intuitive LVQ training scheme was already suggested by Kohonen and is known as LVQ1 (Kohonen et al. 1988, Kohonen 2001). Following the NPC concept, it updates only the currently closest prototype according to a so-called *Winner-Takes-All* (WTA) scheme. Formally, the LVQ1 prescription for a system with two competing prototypes is given by Eq. (3.2) with

$$f_k[d_1^\mu, d_2^\mu, \sigma^\mu] = \Theta\left(d_{\widehat{k}}^\mu - d_k^\mu\right)\Psi(k, \sigma^\mu), \quad \text{where } \widehat{k} = \begin{cases} 2 & \text{if } k = 1 \\ 1 & \text{if } k = 2, \end{cases} \tag{3.3}$$

$$\Theta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{else} \end{cases}, \text{ and } \Psi(k, \sigma) = \begin{cases} +1 & \text{if } k = \sigma \\ -1 & \text{else.} \end{cases}.$$

Here, the Heaviside function $\Theta(\ldots)$ singles out the winning prototype and the factor $\Psi(k, \sigma^\mu)$ determines the sign of the update: The WTA update according to Eq. (3.3) moves the prototype towards the presented feature vector if it carries the same class label $k = \sigma^\mu$. On the contrary, if the prototype is meant to present a different class, its distance from the data point is increased. Note that LVQ1 cannot be interpreted as a gradient descent procedure of a suitable cost function in a straightforward way due to discontinuities at the class boundaries, see (Biehl et al. 2007) for a discussion and references.

Numerous variants and modifications of LVQ have been presented in the literature, including heuristically motivated extensions of LVQ1, cost function based schemes and variants employing unconventional or adaptive distance measures (Kohonen et al. 1988, Kohonen 2001, Kohonen 1990, Nova and Estevez 2014, Biehl et al. 2016, Biehl et al. 2014). These variants usually aim at better convergence or classification performance. Most of these modifications, however, retain the basic idea of attraction and repulsion of the winning prototypes.

3.2.4 Clustered Model Data

LVQ algorithms are most suitable for classification problems which reflect a given cluster structure in the data. In the modelling, we therefore consider a stream of random input vectors $\boldsymbol{\xi} \in \mathbb{R}^N$ which are generated independently according to a mixture of two Gaussians (Biehl et al. 2007, Witoelar et al. 2007, Ghosh et al. 2006)

$$P(\boldsymbol{\xi}) = \sum_{m=1,2} p_m \, P(\boldsymbol{\xi} \mid m) \;\; \text{with} \;\; P(\boldsymbol{\xi} \mid m) = \frac{1}{(2\,\pi\, v_m)^{N/2}} \, \exp\left[-\frac{1}{2\, v_m} \, (\boldsymbol{\xi} - \lambda \boldsymbol{B}_m)^2 \right]. \tag{3.4}$$

The target classification is taken to coincide with the cluster membership, i.e. $\sigma := m$ in Eq. (3.3). The class-conditional densities $P(\boldsymbol{\xi} \mid m = 1, 2)$ correspond to isotropic, spherical Gaussians with variance v_m and mean $\lambda \, \boldsymbol{B}_m$. Prior weights of the clusters are denoted as p_m and satisfy $p_1 + p_2 = 1$. We assume that the vectors $\boldsymbol{B}_m$ are orthonormal with $\boldsymbol{B}_1^2 = \boldsymbol{B}_2^2 = 1$ and $\boldsymbol{B}_1 \cdot \boldsymbol{B}_2 = 0$. Obviously, the classes $m = 1, 2$ are not perfectly separable due to the overlap of the clusters. As an illustration, Fig. (3.1) displays data in $N = 200$ dimensions, generated according to a density of the form (3.4). While the clusters are clearly visible in the subspace given by $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, projections into a randomly chosen plane completely overlap.

We denote conditional averages over $P(\boldsymbol{\xi} \mid m)$ by $\langle \cdots \rangle_m$, whereas mean values $\langle \cdots \rangle = \sum_{m=1,2} p_m \, \langle \cdots \rangle_m$ are defined with respect to the full density (3.4). One obtains, for instance, the conditional and full averages

$$\langle \boldsymbol{\xi} \rangle_m = \lambda \, \boldsymbol{B}_m, \;\; \langle \boldsymbol{\xi}^{\,2} \rangle_m = v_m \, N + \lambda^2 \;\; \text{and} \;\; \langle \boldsymbol{\xi}^{\,2} \rangle = (p_1 v_1 + p_2 v_2) \; N + \lambda^2. \tag{3.5}$$

In the thermodynamic limit $N \to \infty$, which will be considered later, λ^2 can be neglected in comparison to the terms of $\mathcal{O}(N)$ in Eq. (3.5).

Similar clustered densities have been studied in the context of unsupervised learning and supervised perceptron training, see e.g. (Barkai et al. 1993, Biehl et al. 1997, Marangi et al. 1995). Also, online LVQ in stationary situations was analysed in e.g. (Biehl et al. 2007).

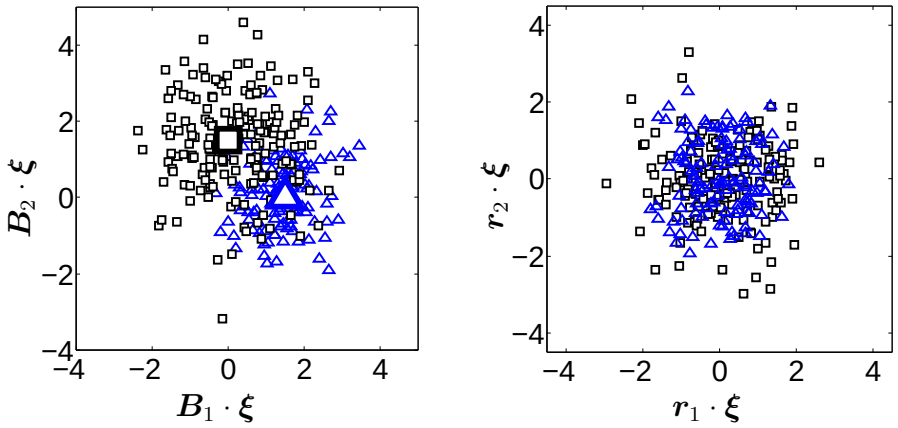


Figure 3.1: *Clustered Model Density*
Illustration of the clustered density, Eq. (3.4), in $N = 200$ dimensions, here with $p_1 = 0.4, p_2 = 0.6$ and $v_1 = 0.64, v_2 = 1.44$. Blue triangles (black squares) represent $120\,(180)$ vectors $\boldsymbol{\xi}$ from the clusters centered at $\lambda \boldsymbol{B}_1$ ($\lambda \boldsymbol{B}_2$) with $\lambda = 1.5$, respectively. **Left panel:** Projections $\boldsymbol{B}_{1,2} \cdot \boldsymbol{\xi}$ of the data. The cluster centers are marked by larger symbols. **Right panel:** Projections $\boldsymbol{r}_{1,2} \cdot \boldsymbol{\xi}$ on two randomly chosen orthonormal vectors $\boldsymbol{r}_{1,2}$.

Here we focus on the question whether LVQ learning schemes are able to cope with drift in characteristic model situations and whether extensions like weight decay can improve the performance in such settings.

Note that the density used in the analysis of the Soft Committee Machine (SCM) in Eq. (2.12) is recovered formally from Eq. (3.4) by setting $\lambda = 0$ and $v_1 = v_2 = 1$, for which both clusters in Eq. (3.4) coincide in the origin and the parameters $p_{1,2}$ become irrelevant.

3.2.5 Macroscopic learning dynamics of LVQ

For convenience, the definitions of the order parameters, their stochastic updates and their averaged dynamics for $N \to \infty$ as derived in Chapter 2 are repeated here:

$$R_{im}^{\mu} = \boldsymbol{w}_i^{\mu} \cdot \boldsymbol{B}_m \quad \text{and} \quad Q_{ik}^{\mu} = \boldsymbol{w}_i^{\mu} \cdot \boldsymbol{w}_k^{\mu}\,, \tag{3.6}$$

$$T_{mn} = \boldsymbol{B}_m \cdot \boldsymbol{B}_n = \delta_{mn} \quad \text{with } i,k \in \{1,...,K\}\,,\ m,n \in \{1,...,M\}\,. \tag{3.7}$$

$$\frac{R_{im}^{\mu} - R_{im}^{\mu-1}}{1/N} = \eta\, \Delta \boldsymbol{w}_i^{\mu} \cdot \boldsymbol{B}_m$$
$$\frac{Q_{ik}^{\mu} - Q_{ik}^{\mu-1}}{1/N} = \eta \left(\boldsymbol{w}_i^{\mu-1} \cdot \Delta \boldsymbol{w}_k^{\mu} + \boldsymbol{w}_k^{\mu-1} \cdot \Delta \boldsymbol{w}_i^{\mu} \right) + \frac{\eta^2}{N} \Delta \boldsymbol{w}_i^{\mu} \cdot \Delta \boldsymbol{w}_k^{\mu}. \tag{3.8}$$

$$\frac{dR_{im}}{d\alpha} = \eta F_{im} \quad \text{and} \quad \frac{dQ_{ik}}{d\alpha} = \eta G_{ik}^{(1)} + \eta^2 G_{ik}^{(2)}\,, \tag{3.9}$$

where in the case of LVQ the adaptive vectors $\{\boldsymbol{w}_i\}_{i=1}^K$ are prototype vectors and $\{\boldsymbol{B}_n\}_{n=1}^M$ are cluster centers. To formulate the macroscopic learning dynamics, we substitute the definition of the LVQ1 updates $\Delta \boldsymbol{w}_i^{\mu} \propto \left(\boldsymbol{\xi}^{\mu} - \boldsymbol{w}_i^{\mu-1} \right)$ from Eq. (3.2) in the generic Eq. (3.8), which then becomes an equation in the projections $\{h_i^{\mu} = \boldsymbol{w}_i^{\mu-1} \cdot \boldsymbol{\xi}^{\mu}\}_{i=1}^K$ and $\{b_n^{\mu} = \boldsymbol{B}_n \cdot \boldsymbol{\xi}^{\mu}\}_{n=1}^M$. Note that the current discussion focuses on settings with two prototypes and two cluster centers, i.e. $K = 2, M = 2$, but we define the modelling framework for settings with general K and M. The class-conditional moments of the projections for the clustered input density (3.4) are:

$$\begin{aligned}
&\langle h_i^{\mu} \rangle_m = \lambda R_{im}^{\mu-1}, \quad \langle b_m^{\mu} \rangle_n = \lambda \delta_{mn}\,, \\
&\langle h_i^{\mu} h_k^{\mu} \rangle_m - \langle h_i^{\mu} \rangle_m \langle h_k^{\mu} \rangle_m = v_m\, Q_{ik}^{\mu-1}\,, \\
&\langle h_i^{\mu} b_n^{\mu} \rangle_m - \langle h_i^{\mu} \rangle_m \langle b_n^{\mu} \rangle_m = v_m\, R_{in}^{\mu-1}\,, \\
&\langle b_l^{\mu} b_n^{\mu} \rangle_m - \langle b_l^{\mu} \rangle_m \langle b_n^{\mu} \rangle_m = v_m\, \delta_{ln}\,,
\end{aligned} \tag{3.10}$$

with $i, k, l, m, n \in \{1, 2\}$ and the Kronecker-Delta $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ else.

For the limit $N \to \infty$, the dynamics can be found by defining the LVQ expressions for F_{im}, $G_{ik}^{(1)}$ and $G_{ik}^{(2)}$:

$$\begin{aligned}
F_{im} &= \left(\langle b_m f_i \rangle - R_{im} \langle f_i \rangle \right), \\
G_{ik}^{(1)} &= \left(\langle h_i f_k + h_k f_i \rangle - Q_{ik} \langle f_i + f_k \rangle \right) \text{ and } G_{ik}^{(2)} = \sum_{m=1,2} v_m p_m \langle f_i f_k \rangle_m
\end{aligned} \tag{3.11}$$

with the LVQ1 modulation functions f_i from Eq. (3.3) and (conditional) averages with respect to the density (3.4), which are computed using the joint Gaussian of the projections with moments given in Eq. (3.10).

The expressions (3.11) required for the formulation of the learning dynamics can be expressed in terms of elementary functions of order parameters and can be substituted in Eq. (3.9) to formulate the ODE. For the straightforward yet lengthy results, we refer the reader to the original literature (Biehl et al. 2007, Ghosh et al. 2006).

After training, the success of learning is quantified in terms of the generalization error ϵ_g, which can also be expressed as a function of order parameters as was also

done for the SCM. For the LVQ model ϵ_g is given as the probability of misclassifying a novel, randomly drawn input vector. The class-specific errors corresponding to data from clusters $k = 1, 2$ in Eq. (3.4) can be considered separately:

$$\epsilon_g \;=\; p_1\,\epsilon_g^1 \;+\; p_2\,\epsilon_g^2, \quad \text{where } \epsilon_g^k \;=\; \left\langle \Theta\left(d_k - d_{\widehat{k}}\right) \right\rangle_k \tag{3.12}$$

is the class-specific misclassification rate, i.e. the probability for an example drawn from a cluster k to be assigned to $\widehat{k} \neq k$ with $d_k > d_{\widehat{k}}$. For the derivation of the class-wise and total generalization error for systems with two prototypes as functions of the order parameters we also refer to (Biehl et al. 2007). One obtains

$$\epsilon_g^k \;=\; \Phi\left(\frac{Q_{kk} - Q_{\widehat{k}\widehat{k}} - 2\lambda(R_{kk} - R_{\widehat{k}k})}{2\sqrt{v_k}\sqrt{Q_{11} - 2Q_{12} + Q_{22}}}\right) \;\text{ where }\; \Phi(z) = \int_{-\infty}^{z} dx \frac{e^{-x^2/2}}{\sqrt{2\pi}}. \tag{3.13}$$

The (numerical) integration of the ODE (3.9) starting from initial conditions

$$\{R_{im}(0), Q_{ik}(0)\}$$

yields the temporal evolution of order parameters in the course of training. By substitution of the solutions in Eq. (3.12) and Eq. (3.13), we can obtain the general learning curve $\epsilon_g(\alpha) = \epsilon_g\left(\{R_{im}(\alpha), Q_{ik}(\alpha)\}\right)$ and the class-wise learning curves $\epsilon_g^k(\alpha)$, respectively. Hence, we determine the typical generalization error after on-line training with $(\alpha\, N)$ random examples.

3.3 The Learning Dynamics Under Concept Drift

The analysis summarized in the previous section and in Chapter 2 concerns learning in the presence of a stationary concept, i.e. for a density of the form (3.4) or (2.12) which does not change in the course of training. Here, we introduce the effect of concept drift to the modelling framework and consider weight decay as an example mechanism for explicit *forgetting*.

3.3.1 Virtual Drift

Virtual drifts affect statistical properties of the observed example data while the actual target function remains unchanged. A variety of virtual drift processes can be addressed in our modelling framework. As one example, time-varying *label noise* in classification or regression could be incorporated in a straightforward way (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993). Similarly,

non-stationary cluster variances in the input density, cf. Eq. (3.4), can be introduced through explicitly time-dependent $v_\sigma(\alpha)$ into Eq. (3.9) for the LVQ system. One part of our analysis of the LVQ system focuses on the particularly relevant case in the context of classification tasks, namely, a varying fraction of examples of the classes in the data stream. We consider non-stationary prior probabilities $p_1(\alpha) = 1 - p_2(\alpha)$ in the mixture density (3.4). In practical situations, varying class bias can complicate the training significantly and lead to inferior performance (Wang et al. 2017). Specifically, we distinguish the following scenarios:

(A) Drift in the training data only
Here we assume that the true target classification is defined by a fixed *reference density* of data. As a simple example we consider equal priors $p_1 = p_2 = 1/2$ in a symmetric reference density (3.4) with $v_1 = v_2$. On the contrary, the characteristics of the observed training data are assumed to be time-dependent. In particular, we study the effect of non-stationary class priors $p_m(\alpha)$ and weight decay on the learning dynamics. Given the order parameters of the learning systems in the course of training, the corresponding *reference generalization error*

$$\epsilon_{ref}(\alpha) = \left(\epsilon_g^1 + \epsilon_g^2\right)/2 \tag{3.14}$$

is obtained by setting $p_1 = p_2 = 1/2$ in Eq. (3.12), but inserting $R_{im}(\alpha)$ and $Q_{ik}(\alpha)$ as obtained from the integration of the corresponding ODE with time dependent $p_1(\alpha) = 1 - p_2(\alpha)$ in the training process.

(B) Drift in training and test data
In the second interpretation we assume that the variation of $p_m(\alpha)$ affects training and test data in the same way. Hence, the change of the statistical properties of the data is inevitably accompanied by a modification of the target classification: For instance, the Bayes optimal classifier and its best linear approximation depend explicitly on the actual priors (Biehl et al. 2007).

The learning system is supposed to track the actual drifting concept and we refer to the corresponding generalization error as the *tracking error*

$$\epsilon_{track} = p_1(\alpha)\,\epsilon_g^1 \;+\; p_2(\alpha)\,\epsilon_g^2. \tag{3.15}$$

In terms of modelling the training dynamics, both scenarios, (A) and (B), require the same straightforward modification of the ODE system: the explicit introduction of α-dependent quantities $p_\sigma(\alpha)$ in Eq. (3.4). The obtained temporal evolution of order parameters yields the reference error $\epsilon_{ref}(\alpha)$ for the case of drift in the training data (A) and $\epsilon_{track}(\alpha)$ in interpretation (B).

Note that in both interpretations, we consider the very same drift processes affecting the training data. However, the interpretation of the relevant performance measure is different. In (A) only the training data is subject to the drift, but the classifier is evaluated with respect to an idealized static situation representing a fixed target. On the contrary, the tracking error in (B) is thought to be computed with respect to test data available from the stream, at the given time. Alternatively, one could interpret (B) as an example of real drift with a non-stationary target, where ϵ_{track} represents the corresponding generalization error. However, we will refer to (A) and (B) as virtual drift throughout the following.

3.3.2 Real Drift

In the presented framework, a real drift can be modelled as a process which displaces the characteristic vectors $\boldsymbol{B}_{1,2}$, i.e. cluster centers in LVQ or the teacher weight vectors in the SCM, in the N- dimensional feature space.

A variety of time-dependences could be considered in the model. We restrict ourselves to the analysis of diffusion-like random displacements of vectors $\boldsymbol{B}_{1,2}(\mu)$ at each time step. Upon presentation of example μ, we assume that random vectors $\boldsymbol{B}_{1,2}(\mu)$ are generated which satisfy the conditions

$$\begin{aligned} &\boldsymbol{B}_1(\mu)\cdot\boldsymbol{B}_1(\mu-1)=\boldsymbol{B}_2(\mu)\cdot\boldsymbol{B}_2(\mu-1)=\left(1-\frac{\delta}{N}\right)\\ &\boldsymbol{B}_1(\mu)\cdot\boldsymbol{B}_2(\mu)=0 \text{ and } \mid\boldsymbol{B}_1(\mu)\mid^2=\mid\boldsymbol{B}_2(\mu)\mid^2=1\,. \end{aligned} \tag{3.16}$$

Here δ quantifies the strength of the drift process. The displacement of the characteristic vectors is very small in an individual training step. We assume for simplicity that the orthonormality of the characteristic vectors is preserved in the drift. In terms of the continuous time $\alpha=\mu/N$, the drift parameter defines a characteristic scale $1/\delta$ on which the overlap of the current teacher vectors with their initial positions decay:

$$\boldsymbol{B}_m(\mu)\cdot\boldsymbol{B}_m(0)\;=\exp[-\delta\mu/N]\,. \tag{3.17}$$

The effect of such a drift process is easily taken into account in the formalism: For a given adaptive vector $\boldsymbol{w}_i\in\mathbb{R}^N$ we obtain (Biehl and Schwarze 1992, Biehl and Schwarze 1993, Kinouchi and Caticha 1993, Vicente and Caticha 1998)

$$[\boldsymbol{w}_i\cdot\boldsymbol{B}_k(\mu)]=\left(1-\frac{\delta}{N}\right)\;[\boldsymbol{w}_i\cdot\boldsymbol{B}_k(\mu-1)]\quad\text{for } k=1,2 \tag{3.18}$$

under the above specified small displacement in discrete learning time. Hence, the drift tends to decrease the quantities R_{ik} which clearly deteriorates the success of

training compared with the stationary case. The resulting ODE for the training dynamics in the limit $N \to \infty$ under the drift process (3.16) read

$$\left[\frac{dR_{im}}{d\alpha}\right]_{drift} = \left[\frac{dR_{im}}{d\alpha}\right]_{stat} - \delta\, R_{im} \quad \text{and} \quad \left[\frac{dQ_{ik}}{d\alpha}\right]_{drift} = \left[\frac{dQ_{ik}}{d\alpha}\right]_{stat} \tag{3.19}$$

with the terms $[\cdots]_{stat}$ denoting the dynamics of the stationary environments defined in Eq. (2.21) and Eq. (3.9). Note that now order parameters $R_{im}(\alpha)$ correspond to the inner products $\boldsymbol{w}_i(\alpha) \cdot \boldsymbol{B}_m(\alpha)$, as the characteristics vectors themselves are also time-dependent.

3.3.3 Weight Decay

Possible motivations for the introduction of so-called *weight decay* in machine learning systems range from *regularization* as to reduce the risk of *over-fitting* in regression and classification (Hastie et al. 2001, Bishop 2006, Goodfellow et al. 2016) to the modelling of *forgetful memories* in attractor neural networks (Mezard et al. 1986, van Hemmen et al. 1987).

Here we introduce weight decay as to enforce *explicit forgetting* and to potentially improve the performance of the systems in the presence of concept drift. We consider the multiplication of all adaptive vectors by a factor $(1 - \gamma/N)$ before the generic learning step given by $\Delta\boldsymbol{w}_i^\mu$ in Eq. (3.2) is performed:

$$\boldsymbol{w}_i^\mu \;=\; \left(1 - \frac{\gamma}{N}\right)\, \boldsymbol{w}_i^{\mu-1} \;+\; \frac{\eta}{N}\Delta\boldsymbol{w}_i^\mu \,. \tag{3.20}$$

Since the multiplications with $(1 - \gamma/N)$ accumulate in the course of training, weight decay enforces an increased influence of the most recent training data as compared to *earlier* examples. Note that analogous modifications of perceptron training under concept drift were discussed in (Biehl and Schwarze 1992, Biehl and Schwarze 1993, Kinouchi and Caticha 1993, Vicente and Caticha 1998).

In the thermodynamic limit $N \to \infty$, the modified ODE for training under real drift, cf. Eq. (3.16), and weight decay, Eq. (3.20), are obtained as

$$\begin{aligned} \left[\frac{dR_{im}}{d\alpha}\right]_{decay} &= \left[\frac{dR_{im}}{d\alpha}\right]_{stat} - (\delta + \gamma) R_{im} \,, \\ \left[\frac{dQ_{ik}}{d\alpha}\right]_{decay} &= \left[\frac{dQ_{ik}}{d\alpha}\right]_{stat} - 2\,\gamma\, Q_{ik} \,. \end{aligned} \tag{3.21}$$

3.4 Results and Discussion

Here we present and discuss our results obtained by integrating the systems of ODE with and without weight decay under different time-dependent drifts. For compari-

son, averaged learning curves obtained by means of Monte Carlo simulations are also shown. These simulations of the actual training process provide an independent confirmation of the ODE-based description and demonstrate the relevance of results obtained in the thermodynamic limit $N \to \infty$ for relatively small, finite systems.

We present the results in the following order:

- **Real drift in LVQ training**: In Sec. 3.4.1 we present the results that were obtained for the learning dynamics of LVQ under a random displacement of the cluster centers of the Gaussian mixture that defines the classification, shifting the optimal decision boundary.
- **Virtual drift in LVQ training**: In Sec. 3.4.2 we present the results of the virtual drift that consists of changing class biases.
- **Real drift in the Erf-SCM for general learning rate**: Sec. 3.4.3 contains a discussion of results obtained for the learning dynamics of the Erf-SCM under a random drift of the teacher vectors. These results are obtained using the full system of ODE (2.21) for general learning rate η.
- **Real drift in the Erf-SCM and the ReLU-SCM**: In Sec. 3.4.4 the analysis of the Erf-SCM in Sec. 3.4.3 is extended significantly: The results are compared with the learning behavior in the ReLU-SCM under real drift and we analyse the sensitivity to drift and weight decay of the two systems in terms of the obtained generalization performance and plateau length in the learning curves.

3.4.1 Learning Vector Quantization in the Presence of Real Concept Drift

We study the typical behavior of LVQ1 under real concept drift as defined in Sec. 3.3.2. Throughout the following we consider prototypes initialized as independent, normalized random vectors with no prior knowledge of the cluster structure, which corresponds to

$$Q_{11}(0) = Q_{22}(0) = 1, \;\; Q_{12}(0) = 0 \;\text{ and }\; R_{im}(0) = 0 \;\text{ for }\; i, m \in \{1, 2\}. \tag{3.22}$$

Fig. 3.2 (left panel) displays example learning curves $\epsilon_g(\alpha)$ for a drift with $\delta = 1$ for different learning rates, see the caption for other model parameters. Details of the initial phase of training depend on the interplay of initial values $Q_{ii}(0)$ and the learning rate. Note that a non-monotonic behavior of $\epsilon_g(\alpha)$ can be observed for some settings.

Monte Carlo simulations show excellent agreement with the $(N \to \infty)$ theoretical predictions already for relatively small systems. This parallels the findings presented

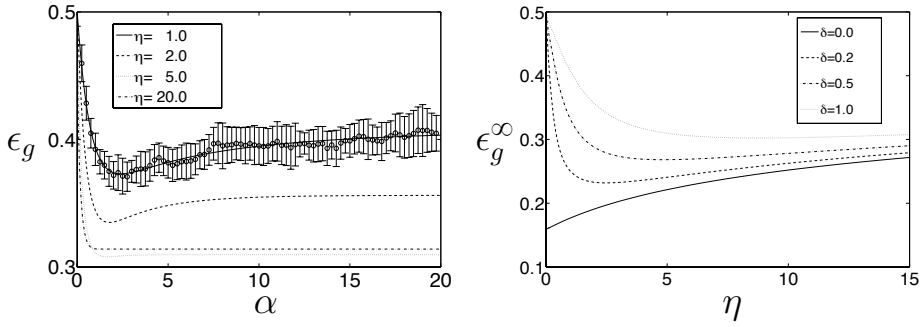


Figure 3.2: *LVQ under Concept Drift: Learning Curves and the Role of the Learning Rate* LVQ1 training from data according to the model density (3.4) with $\lambda = 1, p_1 = p_2 = 0.5$ and $v_1 = v_2 = 0.5$ in the presence of real concept drift. **Left panel:** Learning curves $\epsilon_g(\alpha)$ for $\delta = 1$ and various learning rates η. Symbols and error bars mark the mean results and standard deviations observed in 25 randomized simulations for $N = 1000$ with $\eta = 1$ as an example. **Right panel:** Asymptotic $(\alpha \to \infty)$ generalization error as a function of the learning rate η for different drift parameters δ and in the stationary environment with $\delta = 0$.

in (Biehl et al. 2007, Ghosh et al. 2006) for stationary environments. As just one example, Fig. 3.2 (left) also shows the mean and standard deviation of ϵ_g over 25 randomized runs of the training for $\eta = 1$ and $N = 1000$.

The results for large α show that the success of learning, i.e. the degree to which the drifting concept can be tracked by LVQ1, depends on the learning rate in a non-trivial way. In contrast to learning in stationary environments, the use of very small learning rates obviously fails to maintain the ability to generalize in the presence of a significant real drift. On the other hand, too large learning rates result in inferior performance as well.

After presenting many examples, i.e. in the limit $\alpha \to \infty$, the system approaches a quasi-stationary state in which the LVQ prototypes track the drifting center vectors $\boldsymbol{B}_{1,2}$ with constant overlap parameters R_{im}, Q_{ik}. The configuration corresponds to the stationarity conditions

$$\left[\frac{dR_{im}}{d\alpha}\right]_{drift} = 0 \quad \text{and} \quad \left[\frac{dQ_{ik}}{d\alpha}\right]_{drift} = 0\,. \tag{3.23}$$

Fig. 3.2 (right panel) shows the $\alpha \to \infty$ asymptotic generalization error $\epsilon_g^\infty = \lim_{\alpha\to\infty} \epsilon_g(\alpha)$ as a function of the learning rate η. Only in absence of drift, i.e. for $\delta = 0$, the best possible generalization ability of LVQ1 is obtained in the limit $\eta \to 0$.

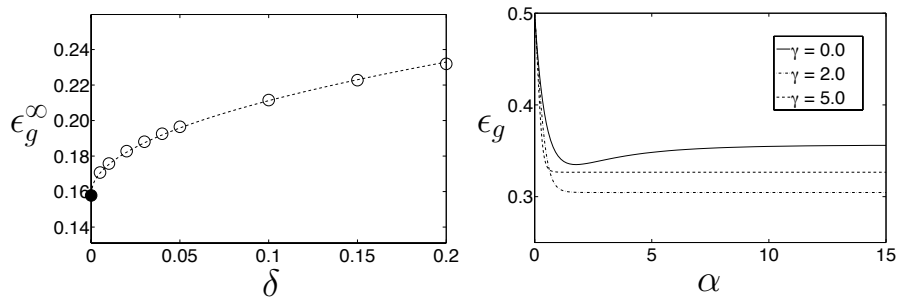


Figure 3.3: *LVQ under Concept Drift: Asympotic Generalization and the Influence of Weight Decay*
LVQ1 in the presence of a real drift with model parameters $\lambda = 1, v_1 = v_2 = 0.5, p_1 = p_2 = 0.5$. **Left panel:** The $(\alpha \to \infty)$ asymptotic generalization error of LVQ1 as obtained with an optimized constant learning rate. Empty circles correspond to numerical results for different drift parameters, the filled circle represents stationary data, for which $\epsilon_g^\infty(\delta = 0) \approx 0.158$. The dashed line corresponds to a fit of the form $\epsilon_g^\infty(\delta = 0) + 0.166\,\delta^{1/2}$. **Right panel:** Learning curves in the model with learning rate $\eta = 2.0$ and drift parameter $\delta = 1.0$. The three curves correspond to learning without weight decay (upper, solid line), with $\gamma = 2$ (lower, dash-dotted line) and $\gamma = 5$ (middle, dashed line) respectively.

We refer the reader to (Biehl et al. 2007, Ghosh et al. 2006) for a detailed discussion of ϵ_g^∞ and its dependence on the model parameters $\lambda, p_\pm$ and $v_\pm$. For $\delta > 0$, the limit $\eta \to 0$ results in trivial asymptotic behavior corresponding to random guesses, with $\epsilon_g^\infty = 1/2$ for the symmetric input density with $p_1 = p_2$ and $v_1 = v_2$, for instance.

Given the drift parameter δ, an optimal constant learning rate can be identified with respect to the generalization ability in the quasi-stationary state. The use of this learning rate yields, for $\alpha \to \infty$, the best ϵ_g^∞ achievable under drift. It is displayed in Fig. 3.3 (left panel) as a function of δ for small values of the drift parameter. The optimal quasi-stationary generalization error under concept drift scales like

$$\left[\epsilon_g^\infty(\delta) - \epsilon_g^\infty(0)\right] \propto \delta^{1/2} \text{ for small } \delta\,. \tag{3.24}$$

As expected, the drift impedes the learning process. However, our results show that already the simplest LVQ scheme is capable of tracking randomly drifting clusters and to maintain a significant generalization ability, even in very high-dimensional spaces.

We have also studied the effect of weight decay in the presence of the above discussed real concept drift. Fig. 3.3 (right panel) displays example learning curves for

LVQ1 training under a random drift with $\delta = 1$ with various weight decay parameters γ for a given learning rate $\eta = 2$. As these examples show, the implementation of weight decay has the potential to improve the generalization behavior significantly when tracking a drifting concept. The simultaneous optimization of learning rate and weight decay $\{\eta, \gamma\}$ with respect to the success of training in the *tracking state* will be addressed in forthcoming studies.

3.4.2 Virtual Drift in LVQ training

All results presented in the following are for constant learning rate $\eta = 1$ in the LVQ training. The results remain qualitatively the same for a range of learning rates. LVQ prototypes were initialized as normalized independent random vectors without prior knowledge:

$$Q_{11}(0) = Q_{22}(0) = 1,\ Q_{12}(0) = 0,\ \text{and}\ R_{ik}(0) = 0\,. \tag{3.25}$$

We study three specific scenarios for the time-dependence $p_1(\alpha) = 1 - p_2(\alpha)$ as detailed in the following.

Linear increase of the bias

Here we consider a time-dependent bias of the form $p_1(\alpha) = 1/2$ for $\alpha < \alpha_o$ and

$$p_1(\alpha) = \frac{1}{2} + \frac{(p_{max} - 1/2)\,(\alpha - \alpha_o)}{(\alpha_{end} - \alpha_o)} \text{ for } \alpha \geqslant \alpha_o. \tag{3.26}$$

where the maximum class weight $p_1 = p_{max}$ is reached at learning time α_{end}.

Fig. 3.4 shows the learning curves as obtained by numerical integration of the ODE together with Monte Carlo simulation results for $(N = 100)$-dimensional inputs and prototype vectors. As an example we set the parameters to $\alpha_o = 25, p_{max} = 0.8, \alpha_{end} = 200$. We set $v_{1,2} = 0.4$ and $\lambda = 1$ in the density 3.4. The learning curves are displayed for LVQ1 without weight decay (upper) and with $\gamma = 0.05$ (lower panel). Simulations show excellent agreement with the ODE results.

The system adapts to the increasing imbalance of the training data, as reflected by a decrease (increase) of the class-wise error for the over-represented (under-represented) class, respectively. The weighted over-all error ϵ_{track} also decreases, i.e. the presence of class bias facilitates smaller total generalization error, see (Biehl et al. 2007). The performance with respect to unbiased reference data deteriorates slightly, i.e. ϵ_{ref} grows with increasing class bias as the training data represents the target less faithfully.

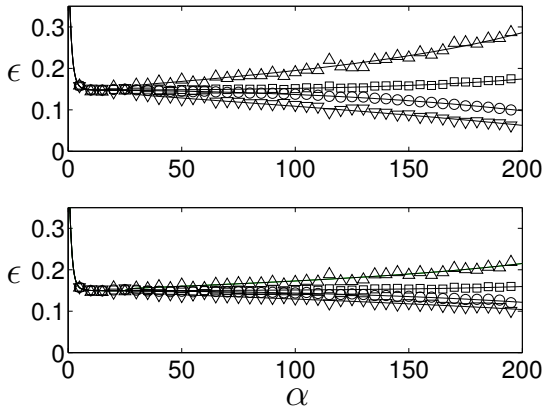


Figure 3.4: LVQ1 in the presence of a concept drift with linearly increasing $p_1(\alpha)$ given by $\alpha_o = 20$, $\alpha_{end} = 200$, $p_{max} = 0.8$ in (3.26). Solid lines correspond to the integration of ODE with initialization as in Eq. (3.25). We set $v_{1,2} = 0.4$ and $\lambda = 1$ in the density (3.4). The upper graph corresponds to LVQ1 without weight decay, the lower graph displays results for $\gamma = 0.05$ in Eq. (3.20). In addition, Monte Carlo results for $N = 100$ are shown: class-wise errors $\epsilon^{1,2}(\alpha)$ are displayed as downward (upward) triangles, respectively; squares mark the reference error $\epsilon_{ref}(\alpha)$; circles correspond to $\epsilon_{track}(\alpha)$, cf. Eqs. (3.14,3.15).

Sudden change of the class bias

Here we consider an instantaneous switch from low bias $p_1(\alpha) = 1 - p_{max}$ for $\alpha \leqslant \alpha_o$ to high bias

$$p_1(\alpha) = \begin{cases} 1 - p_{max} & \text{for } \alpha \leqslant \alpha_o. \\ p_{max} > 1/2 & \text{for } \alpha > \alpha_o. \end{cases} \tag{3.27}$$

We consider $p_{max} = 0.75$ as an example, the corresponding results from the integration of ODE and Monte Carlo simulations are shown in Fig. 3.5 for training without weight decay (upper) and for $\gamma = 0.05$ (lower panel).

We observe similar effects as for the slow, linear time-dependence: The system reacts rapidly with respect to the class-wise errors and the tracking error ϵ_{track} maintains a relatively low value. Also, the reference error ϵ_{ref} displays robustness with respect to the sudden change of p_1. Weight decay, as can be seen in the lower panel of Fig. 3.5 reduces the over-all sensitivity to the bias and its change: Class-wise errors are more balanced and the weighted ϵ_{track} slightly increases compared to the setting with $\gamma = 0$.

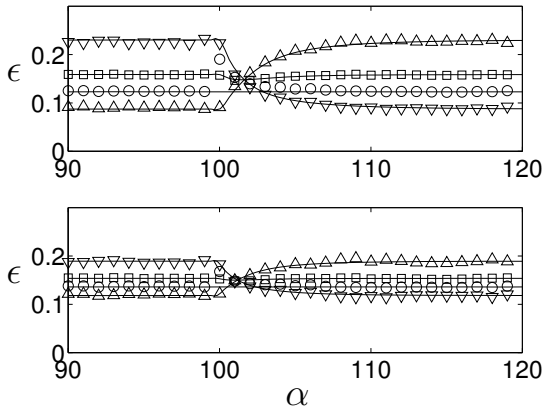


Figure 3.5: LVQ1 in the presence of a concept drift with a sudden change of class weights according to Eq. (3.27) with $\alpha_o = 100$ and $p_{max} = 0.75$. Only the α-range close to α_o is shown. All other details are provided in Fig. 3.4.

Periodic time dependence

As a third scenario we consider oscillatory modulations of the class weights during training:

$$p_1(\alpha) = 1/2 + (p_{max} - 1/2)\, \cos\left(2\pi\, \alpha/T\right) \tag{3.28}$$

with periodicity T on α-scale and maximum amplitude $p_{max} < 1$. Example results are shown in Fig. 3.6 for $T = 50$ and $p_{max} = 0.8$. Monte Carlo results for $N = 100$ are only displayed for the class-wise errors, for the sake of clarity. They show excellent agreement with the numerical integration of the ODE for training without weight decay (upper panel) and for $\gamma = 0.05$ (lower panel). These results confirm our findings for slow and sudden changes of the prior weights: Weight decay limits the flexibility of the LVQ system to react to the presence of a bias and its time-dependence.

Discussion: LVQ under virtual drift

Our results for the different realizations of time-dependent class weights show that Learning Vector quantization can cope with this form of drift to a certain effect. By design, standard incremental updates like the classical LVQ1 allow the prototypes to adjust to the changing statistics of the data. This has been shown in Sec. 3.4.1 for the actual drift of the cluster centers in the model density. Here we show that LVQ1 can also cope with the virtual drift processes.

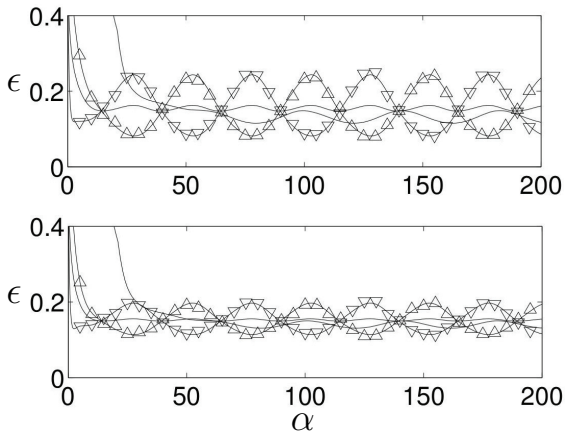


Figure 3.6: LVQ1 in the presence of oscillating class weights according to Eq. (3.28) with parameters $T = 50$ and $p_{max} = 0.8$, without weight decay $\gamma = 0$ (upper graph) and for $\gamma = 0.05$ (lower). For clarity, Monte Carlo results are only shown for the class-conditional errors ϵ^1 (downward) and ϵ^2 (upward triangles). All other details are given in Fig. 3.4.

In analogy to our findings in Sec. 3.4.1, one might have expected improved performance when introducing weight decay as a mechanism of *forgetting*. As we demonstrate, however, weight decay does not have a very strong effect on the system's reaction to changing prior class weights. Essentially, weight decay limits the prototype norms and hinders shifts of the decision boundary by prototype displacement. The overall influence of class bias and its time-dependence is reduced in the presence of weight decay. As a consequence, the tracking error slightly increases for $\gamma > 0$, in general. On the contrary, the error ϵ_{ref} with respect to the reference density decreases compared to the training without weight decay.

A clear beneficial effect of *forgetting* previous information in favor of the most recent examples cannot be confirmed in this case. The reaction of the learning system to sudden (B) or oscillatory changes of the priors (C) remains also largely unchanged and similar to (A) when introducing weight decay.

3.4.3 SCM Regression in the Presence of Real Concept Drift

Here we present results concerning the SCM student-teacher scenario with $K = M = 2$. Already in this simple setting and in the absence of concept drift, the learning dynamics display non-trivial effects which have been shown and studied in detail in, among others, (Saad and Solla 1995a, Saad and Solla 1995b, Biehl et al. 1996) for

erf activation and in Chapter 2 for ReLU activation. Perhaps the most thoroughly studied phenomenon in the SCM training process is the existence of quasi-stationary plateaus in the evolution of the order parameters and the generalization error. In the most clear-cut cases, they correspond to approximately symmetric configurations of the student network with respect to the teacher network, i.e. $R_{im} \approx R$ for all i, m. In such a state, all student units have acquired the same, limited knowledge of the target rule. Hence, the generalization error in the plateau is sub-optimal. In terms of Eqs. (2.21), plateaus correspond to weakly repulsive fixed points of the ODE system. One can show in case of orthonormal teacher units and for small learning rates that a symmetric fixed point with $R_{im} = R$ and the associated plateau state always exists, see e.g. (Saad and Solla 1995b). In order to achieve a further decrease of the generalization error, the symmetry of the student with respect to the teacher units has to be broken by *specialization*: Each student weight vector $\boldsymbol{w}_{1,2}$ has to represent a specific teacher unit and $R_{i1} \neq R_{i2}$, i.e., the student specialization $S_i > 0$, is required for successful learning.

Note that in general, more complex fixed point configurations with different degrees of (partial) specialization can be found. The number of observable plateaus depends on the learning rate and increases for larger K and M, see (Biehl et al. 1996) for a detailed discussion in the absence of drift.

The problem of delayed learning due to saddle points and related effects in gradient-based training is obviously also of interest in the context of *Deep Learning*, see (Goodfellow et al. 2016, Marcus 2018, Dauphin et al. 2014, Tishby and Zaslavsky 2015) for recent investigations and further references.

In practice, one expects $R_{im}(0) \approx 0$ for all i, m unless prior knowledge is available about the target. Hence, also the *student specialization* $S_i(0) = |R_{i1}(0) - R_{i2}(0)|$ is expected to be small, initially. A nearly unspecialized configuration with $S_i(\alpha) \approx 0$ persists in a transient phase of learning, which can extend over large values of α. The actual shape and length of the plateau depends on the precise initialization and the repulsive properties of the corresponding fixed point of the dynamics, see (Biehl et al. 1996) for a detailed discussion, which also addresses the effect of finite N in Monte Carlo simulations.

Fig. 3.7 (left panel) shows an example (lowest curve) of a pronounced plateau state in on-line gradient descent for initial conditions

$$R_{im} = R_o + U(10^{-5}) \text{ with } R_o = 0.01, \;\; Q_{11} = Q_{22} = 0.5, Q_{12} = 0.49. \tag{3.29}$$

Here $U(X)$ denotes a random number drawn from the interval $(0, X]$ with uniform probability, hence also $S_i(0) = \mathcal{O}(X)$. The initialization corresponds to nearly identical student vectors with little prior knowledge. It is inspired by the analyses in (Saad and Solla 1995b, Biehl et al. 1996) which showed that the actual value of R_o is largely

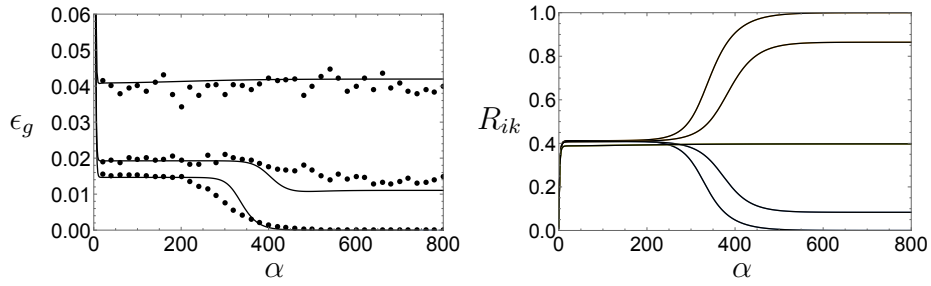


Figure 3.7: *Regression under Concept Drift: Learning Curves*
Gradient-based training of the Soft Committee Machine with $K = M = 2$ and orthogonal teacher vectors in the presence of real target drift. Erf activation is used in the hidden units and, the learning rate is set to $\eta = 0.5$ and initial conditions are as specified in Eq. (3.29). **Left panel:** Learning curves for the stationary case with $\delta = 0$ (lower line), for weak drift with $\delta = 0.005$ (middle) and for strong drift with $\delta = 0.03$ (upper line). Symbols represent the result of single Monte Carlo simulation runs for system size $N = 500$. **Right panel:** The corresponding evolution of the student-teacher overlaps $R_{11} = R_{22}$ and $R_{12} = R_{21}$ vs. α for the stationary case with $\delta = 0$ (lower and upper line), for weak drift with $\delta = 0.005$ (intermediate) and strong drift with $\delta = 0.03$ (center, all overlaps equal).

irrelevant for the observed plateau length, while it depends logarithmically on X (Biehl et al. 1996). Corresponding Monte Carlo simulations are shown in the left panel of Fig. 3.7 for $N = 500$ and randomly drawn initial student vectors, resulting in $R_{im}(0) = \mathcal{O}(1/\sqrt{N})$, with $Q_{ik}(0)$ fixed according to Eq. (3.29). Simulations confirm the theoretical predictions very well, qualitatively.

For very slow drifts of the target concept, the behavior is still similar to the stationary case. For an example with $\delta = 0.005$, Fig. 3.7 (left panel) shows the $N \to \infty$ theoretical learning curve and Monte Carlo simulations: After a rapid, initial decrease of the generalization error, a quasi-stationary, unspecialized plateau is reached. Eventually, the symmetry is broken and the system approaches its $\alpha \to \infty$ asymptotic state, in which a smaller but non-zero $\epsilon_g^\infty(\delta)$ is achieved. Obviously, on-line gradient descent training enables the SCM to track the drifting target to a reasonable degree and maintains a specialized hidden unit configuration.

The behavior changes significantly in the presence of stronger concept drifts: The SCM remains unspecialized even for $\alpha \to \infty$ and, consequently, the achievable generalization ability is relatively poor. Fig. 3.7 (left panel) displays the corresponding learning curve for $\delta = 0.03$ as an example, together with the result of a single Monte Carlo simulation.

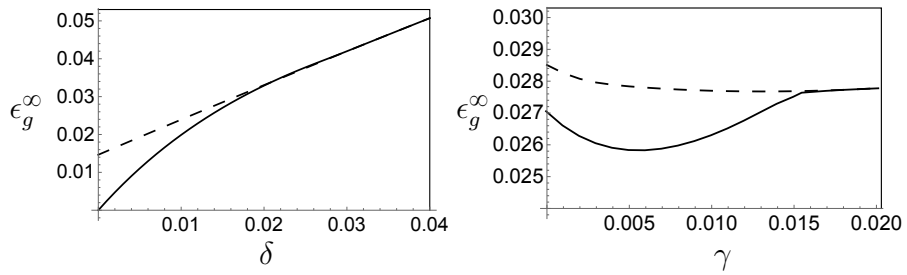


Figure 3.8: *Regression under Concept Drift: Plateaus and Specialized States*
Soft Committee Machine with Erf activation, regression in the presence of real target drift, learning rate and model parameters as in Fig. 3.7. **Left panel:** The generalization error vs. the drift parameter δ for weight decay strength $\gamma = 0$, in the symmetric plateau state with $R_{11} = R_{22} = R_{12} = R_{21}$ (dashed line) and in the $\alpha \to \infty$ stationary state (solid). **Right panel:** The influence of weight decay: For a given drift with $\delta = 0.015$, the $\alpha \to \infty$ asymptotic generalization error is displayed as a function of the weight decay parameter γ. In addition, the dashed line marks ϵ_g in the unspecialized plateau state.

Fig. 3.7 (right panel) shows the evolution of the overlap parameters $R_{im}(\alpha)$ corresponding to the learning curves displayed in the left panel. While for $\delta = 0.005$ the student units still specialize, the unspecialized plateau state with $R_{im} \approx R$ for all i, m persists for $\delta = 0.03$.

In the left panel of Fig. 3.8, this is illustrated in terms of the (quasi-) stationary values of ϵ_g: The system can benefit from the specialization in terms of a low $\alpha \to \infty$ asymptotic generalization error (solid line). For $\delta \approx 0$, the achievable generalization error increases linearly with the drift parameter: $\epsilon_g^\infty(\delta) \propto \delta$. Note that $\epsilon_g^\infty(\delta = 0) = 0$ in the perfectly learnable scenario with $K = M$ considered here. On the contrary, for larger δ, the only stable fixed point of the system coincides with an unspecialized configuration (dashed line). The generalization error of the latter also displays a linear dependence on δ for slow drifts.

Weight decay can improve the performance slightly in the presence of weak concept drifts. As displayed in Fig. 3.8 (right panel) for an example drift of $\delta = 0.015$, the parameter γ in Eq. (3.3.3) can be tuned to decrease the achievable generalization error in the unspecialized plateau (dashed line) and, more importantly, in the final quasi-stationary *tracking state* (solid line). Specialization cannot be achieved if the weight decay parameter is set too large.

3.4.4 SCM regression under real concept drift: Erf vs. ReLU in case of small learning rates

Here we present the results concerning the SCM student teacher scenario with $K = M = 2$ under real concept drift, i.e. random displacements of the teacher vectors as introduced in Sec. 3.3.2. We extend the analysis in the previous section significantly and we focus specifically on the comparison of the Erf-SCM with the ReLU-SCM. For the comparison of Erf-SCM and ReLU-SCM, in absence of concept drift, see Chapter 2 in which interesting distances between the two activation functions are revealed. Unlike LVQ for classification, gradient descent based training of a regression system is expected to be much more sensitive to the choice of the learning rate. Here, we restricted the discussion to the well-defined limit of small learning rates, $\eta \to 0$ and $\alpha \to \infty$ with $\widetilde{\alpha} = \eta\alpha = \mathcal{O}(1)$, see the discussion before Eq. (2.22) in Chapter 2.

ODE and Monte Carlo simulations

Here, we investigate and compare the learning dynamics of networks with Erf- and ReLU-activation under concept drift and in the presence of weight decay. To this end we study the models by numerical integration of the corresponding ODE and, in addition, by Monte Carlo simulations.

Similar to the previous section, we study training processes in absence of prior knowledge in the student. In the following we consider exemplary initial conditions with

$$R_{im}(0) = 0,\ Q_{11}(0) = Q_{22}(0) = 0.5,\ Q_{12}(0) = 0.49 \tag{3.30}$$

which correspond to almost identical student weight vectors $\mathbf{w}_1(0)$ and $\mathbf{w}_2(0)$, which are both orthogonal to the teacher vectors. Note that the initial norm of the student vectors and their mutual *overlap* $Q_{12}(0)$ can be set arbitrarily in practice.

For the networks with two hidden units we recall the definition of the quantity $S_i(\alpha) = |R_{i1}(\alpha) - R_{i2}(\alpha)|$ as the specialization of student units $i = 1, 2$. In the plateau state, $S_i(\alpha) \approx 0$ for an extended amount of training time, while an increasing value of $S_i(\alpha)$ indicates the specialization of the unit. In practice, one expects that initially $R_{im}(0) \approx 0$ for all i, m if no prior information is available about the target rule. Hence, the student specialization $S_i(0) = |R_{i1}(0) - R_{i2}(0)|$ is also small, initially.

We noted in the previous section that the unspecialized plateau can dominate the learning process and, consequently, its length is a quantity of significant interest. Quite generally, the length is governed by the repulsive properties of the relevant fixed point of the ODE system and depends logarithmically on the the magnitude of the initial specialization $S_i(0)$, see (Biehl et al. 1996) for a detailed discussion. In

simulations for large N, a random initialization of student vectors would result in overlaps $R_{im}(0) = \mathcal{O}(1/\sqrt{N})$ with the teacher vectors which also implies that $S_i(0) = \mathcal{O}(1/\sqrt{N})$. The accurate extrapolation of simulation results for $N \to \infty$ is complicated by this interplay of finite size effects and initial specialization which governs the escape from the plateau states (Biehl et al. 1996). Due to fluctuations in a finite system, plateaus are typically left earlier than predicted by the theoretical prediction for $N \to \infty$. Here we focus on the performance achieved in the plateau states and resort to a simpler strategy: The values of the order parameters observed at $\widetilde{\alpha} = 0.05$ in the Monte Carlo simulation are used as initial values for the numerical integration of the ODE. This does not necessarily warrant a one-to-one correspondence of the precise shape and length of the plateau states. However, the comparison shows excellent qualitative agreement and allows for the quantitative comparison of the performance in the quasi-stationary states.

We have studied the Erf-SCM and the ReLU-SCM under concept drift, Eq. (3.16), and weight decay, Eq. (3.20), in the limit of small learning rates $\eta \to 0$. We resorted to this simplifying limit as the term $G_{ik}^{(2)}$ in Eq. 2.24 could not be obtained analytically for the ReLU-SCM. However, non-trivial results can be achieved in terms of the rescaled training time $\widetilde{\alpha}$ in the limit (2.22). Hence we integrate the ODE provided in Eq. (2.23), combined with the drift and weight decay terms from Eqs. (3.19) and (3.21) that also have to be scaled with η in this case: $\widetilde{\delta} = \eta\delta$, $\widetilde{\gamma} = \eta\gamma$. For completeness, the full ODE then reads:

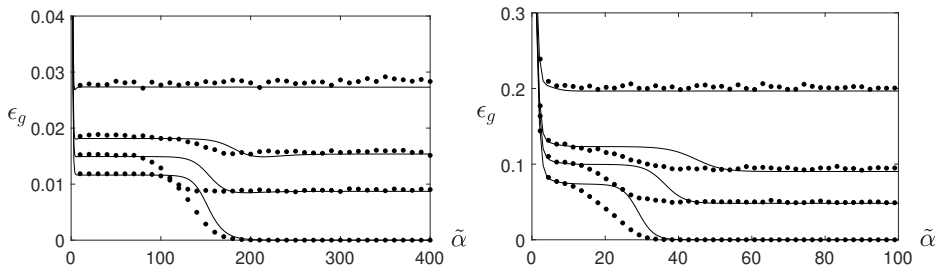
$$\begin{aligned} \left[\frac{dR_{im}}{d\widetilde{\alpha}}\right]_{decay} &= \left[\frac{dR_{im}}{d\widetilde{\alpha}}\right]_{stat} - (\widetilde{\delta} + \widetilde{\gamma})R_{im}\,, \\ \left[\frac{dQ_{ik}}{d\widetilde{\alpha}}\right]_{decay} &= \left[\frac{dQ_{ik}}{d\widetilde{\alpha}}\right]_{stat} - 2\,\widetilde{\gamma}\,Q_{ik}\,. \end{aligned} \tag{3.31}$$

In addition to the numerical integration of the above ODE we have performed and averaged over 10 independent runs of Monte Carlo simulations with system size $N = 500$ and small but finite learning rate $\eta = 0.05$.

Learning curves under concept drift

Fig. 3.9 shows the learning curves $\epsilon_g(\widetilde{\alpha})$ as results of the averaged Monte Carlo simulations and the ODE integration for different strengths $\widetilde{\delta}$ of concept drift with no weight decay ($\widetilde{\gamma} = 0$). The left and right panel correspond to Erf- and ReLU-SCM, respectively.

Apart from deviations in terms of the plateau lengths, simulations and the numerical integration of the ODE show very good agreement. In particular, the generalization error in the plateau and final states nearly coincides. As outlined in Sec. 3.4.3, the



(a) Erf activation (b) ReLU activation

Figure 3.9: The learning performance under concept drift in terms of generalization error as a function of the learning time $\tilde{\alpha}$. Dots correspond to the average over 10 runs of Monte Carlo simulations with $N = 500$, $\eta = 0.05$ with initial conditions as in Eq. (3.30). Solid lines show ODE integrations. (a): Erf SCM. From bottom to top, the curves correspond to the levels of target drift $\widetilde{\delta} = \{0, 0.01, 0.02, 0.05\}$. (b): ReLU SCM. From bottom to top, the levels of target drift are: $\widetilde{\delta} = \{0, 0.05, 0.1, 0.3\}$.

actual length of plateaus in simulations depends on subtle details (Biehl et al. 1996) which were not addressed here.

Note also that a direct, quantitative comparison of Erf- and ReLU-SCM in terms of training times $\tilde{\alpha}$ is not meaningful. For instance, it seems tempting to conclude that the ReLU-SCM exhibit shorter plateau states for the same network size and training conditions. However, one has to take into account that the activation functions influence the complexity of the input output relation of the network in a non-trivial way.

From the behavior of the learning curves for increasing strengths $\widetilde{\delta}$ as shown in Fig. 3.9, several impeding effects of the drift can be identified: The generalization errors in the unspecialized plateau and in the final state for large $\tilde{\alpha}$ increase with $\widetilde{\delta}$. At the same time, the plateau lengths increase. These effects are observed for both types of activation function. More specifically, the behavior for small $\widetilde{\delta}$ is close to the stationary setting with $\widetilde{\delta} = 0$: A rapid initial decrease of the generalization error is followed by the quasi-stationary plateau state that persists for a relatively long training time. Eventually, the system escapes from the plateau and improved generalization performance becomes possible. Despite the matching complexity of student and teacher, perfect generalization cannot be achieved in the presence of on-going concept drift. In the corresponding Monte Carlo simulations, cf. Figs. 3.9a and 3.9b, we employed a reasonably small learning rate $\eta = 0.05$ which yielded very good agreement.

We note that the stronger the drift, the smaller is the difference between the

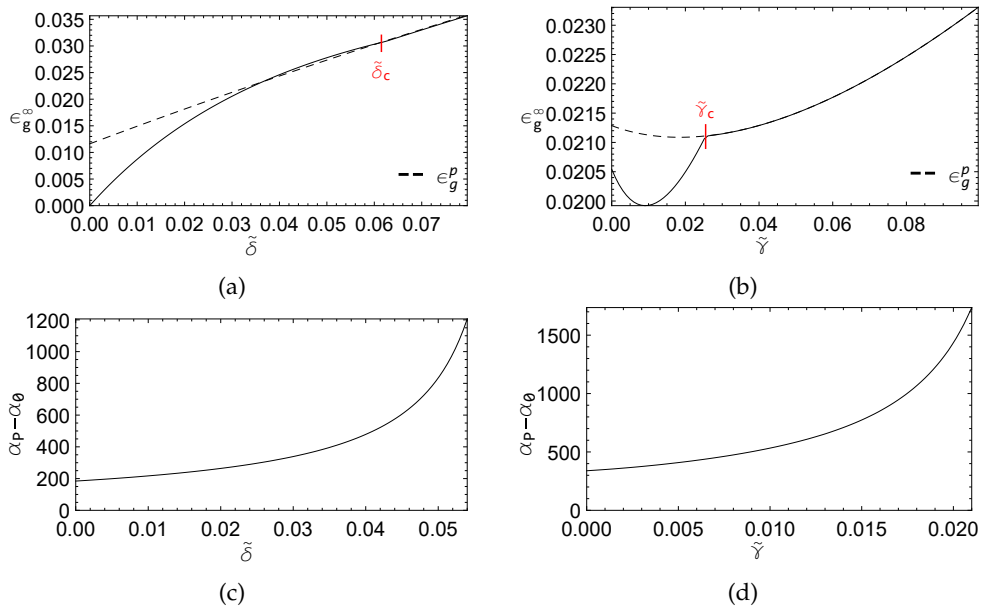


Figure 3.10: Erf-SCM: Generalization error under concept drift in unspecialized plateau states (dashed lines) and final states (solid) of the learning process. 3.10a: Plateau- and final generalization error for an increasing strength $\widetilde{\delta}$ of the target drift. Here, weight decay is not applied: $\widetilde{\gamma} = 0$. For $\widetilde{\delta} > \widetilde{\delta}_c$ as marked by the vertical line, the curves merge. 3.10b: The plateau- and final generalization error as a function of the weight decay parameter $\widetilde{\gamma}$ for a fixed level of real target drift, here: $\widetilde{\delta} = 0.03$. The curves merge for $\widetilde{\gamma} > \widetilde{\gamma}_c$, as marked by the vertical line. The lower panels show the observed plateau lengths as a function of $\widetilde{\delta}$ for $\widetilde{\gamma} = 0$ (5c) and as a function of $\widetilde{\gamma}$ for fixed $\widetilde{\delta} = 0.03$ (5d), respectively.

performance in the plateau and the final state. For very large values of $\widetilde{\delta}$ both versions of the SCM cannot escape the plateau state anymore as it corresponds to a stable fixed point of the ODE.

In the following we discuss for both activation functions the effect of concept drift on the plateau- and final generalization error in greater detail. The influence of weight decay on the dynamics is also presented.

Erf-SCM under drift and weight decay

Fig. 3.10a displays the effect of the drift strength $\widetilde{\delta}$ on the generalization error in the unspecialized plateau state and in the final state for $\widetilde{\alpha} \to \infty$, denoted with $\epsilon_g^p(\widetilde{\delta})$

and $\epsilon_g^\infty(\widetilde{\delta})$, respectively. As mentioned above, weak drifts still allow for student specialization with improved performance in the final state for large $\widetilde{\alpha}$. However, increasing the drift strength results in a decrease of the difference $|\epsilon_g^\infty(\widetilde{\delta}) - \epsilon_g^p(\widetilde{\delta})|$. We have marked the value of $\widetilde{\delta}$, above which the plateau becomes the stable final state for $\widetilde{\alpha} \to \infty$ in the figure and refer to it as $\widetilde{\delta}_c$.

Interestingly, in a small range of the drift parameter, $0.036 < \widetilde{\delta} < 0.061$, the final performance is worse than in the plateau, i.e. $\epsilon_g^\infty(\widetilde{\delta}) > \epsilon_g^p(\widetilde{\delta})$. Since ϵ_g depends explicitly also on the Q_{ik}, it is possible for an unspecialized state with $R_{im} = R$ to generalize better than a slightly specialized configuration with unfavorable values of the student norms and mutual overlaps.

Fig. 3.10c shows the effect of the drift on the plateau length. The start and end of the plateau are defined as

$$\begin{aligned} \widetilde{\alpha}_0 &= \min\{\widetilde{\alpha} \,|\, \epsilon_g^p - 10^{-4} < \epsilon_g(\widetilde{\alpha}) < \epsilon_g^p + 10^{-4}\} \\ \widetilde{\alpha}_P &= \min\{\widetilde{\alpha} \,|\, S_i(\widetilde{\alpha}) \geqslant 0.2\, S_i(\widetilde{\alpha} \to \infty)\}\,. \end{aligned} \tag{3.32}$$

Here, $S_i(\widetilde{\alpha} \to \infty)$ represents the final specialization that is achieved by the system for large training times. The value $(\widetilde{\alpha}_P - \widetilde{\alpha}_0)$ is used as a measure of the plateau length.

In the weak drift regime, the plateau length increases slowly with $\widetilde{\delta}$ as shown in panel (c) for $\widetilde{\gamma} = 0$. It eventually diverges as $\widetilde{\delta}$ approaches $\widetilde{\delta}_c$ from Fig. 3.10a.

The dependence of ϵ_g^p and ϵ_g^∞ on the weight decay parameter $\widetilde{\gamma}$ is shown in Fig. 3.10b. We observe improved performance for a small amount of weight decay compared to absence of weight decay ($\widetilde{\gamma} = 0$), similar to the result in Fig. 3.8 for larger learning rate. However, the system is quite sensitive to the actual setting of $\widetilde{\gamma}$: Values slightly larger than the optimum quickly deteriorate the ability for improvement from the plateau generalization error. The value of $\widetilde{\gamma}$, above which the plateau- and final generalization error coincide has been marked in the figure and we refer to it as $\widetilde{\gamma}_c$.

Fig. 3.10d shows the effect of the weight decay on the plateau length in the same setting as in Fig. 3.10b. Introducing a weight decay always extends the plateau length. For small $\widetilde{\gamma}$ the plateau length grows slowly and diverges as $\widetilde{\gamma}$ approaches $\widetilde{\gamma}_c$ from Fig. 3.10b.

ReLU-SCM under drift and weight decay

For the ReLU-SCM, part of the results are similar and other results are significantly different from the results of the Erf-SCM.

The effect of the strength of the drift on the generalization error in the unspecialized plateau state and in the final state is shown in Fig. 3.11a. Here, the picture is similar to the Erf-SCM: an increase in the drift strength causes an increase in the

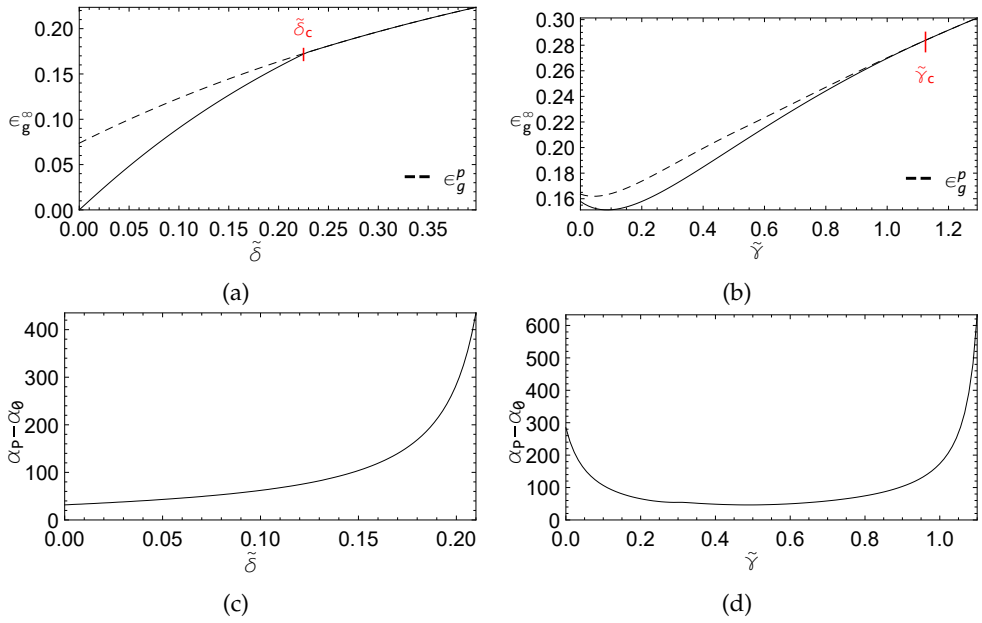


Figure 3.11: ReLU-SCM: Generalization error under concept drift in unspecialized plateau states (dashed lines) and final states (solid), as a function of the drift strength (6a) and weight decay (6b). In (6b), $\widetilde{\delta} = 0.2$. The drift strength $\widetilde{\delta}_c$ above which the curves merge is marked in (6a) and similar for weight decay $\widetilde{\gamma}_c$ in (6b). The lower panels show the observed plateau lengths as a function of $\widetilde{\delta}$ for $\widetilde{\gamma} = 0$ (6c) and as a function of $\widetilde{\gamma}$ for fixed $\widetilde{\delta} = 0.2$ (6d), respectively.

plateau- and final generalization error. We have marked in the figure the drift strength $\widetilde{\delta}_c$ at which there is no further change in performance from the plateau. In contrast to the Erf-SCM, there is no range of $\widetilde{\delta}$ for which the ReLU-SCM generalization error increases after leaving the plateau.

Fig. 3.11c shows the effect of the strength of the drift on the plateau length. Here too, a similar dependence is observed as for the Erf-SCM: For the range of weaker drifts, the plateau length grows slowly and diverges for strong drifts up to the drift strength $\widetilde{\delta}_c$ from Fig. 3.11a.

Fig. 3.11b shows the effect of the amount of weight decay on the plateau- and final generalization error in a concept drift situation. A small amount of weight decay can improve the generalization error compared to no weight decay ($\widetilde{\gamma} = 0$). The effect weight decay has on the ReLU-SCM shows a much greater robustness compared to the Erf-SCM in terms of the ability to improve from the plateau value: For high

amounts of weight decay, an escape from the plateau to better performance can still be observed. The value $\widetilde{\gamma}_c$, above which the plateau- and final generalization error coincide has been marked in the figure.

Fig. 3.11d shows the effect of the amount of weight decay on the plateau length in the same concept drift situation as in Fig. 3.11b. In contrast to the result for the Erf-SCM, it shows that for the ReLU-SCM the plateau is shortened significantly in the smaller range of weight decay, the same range that also improves the final generalization error as observed in Fig. 3.11b. The plateau length increases again for very high levels of weight decay and diverges as $\widetilde{\gamma}$ approaches the $\widetilde{\gamma}_c$ from Fig. 3.11b.

3.4.5 Discussion: SCM regression under real drift

As was already discussed, the symmetric plateau corresponds to states where the student units have all learned the same limited and general knowledge about the teacher units, i.e. $R_{im} \approx R$ and therefore the specialization of each student unit i is small: $S_i(\widetilde{\alpha}) = |R_{i1}(\widetilde{\alpha}) - R_{i2}(\widetilde{\alpha})| \approx 0$. Eventually, the symmetry is broken by the start of specialization, when $S_i(\widetilde{\alpha})$ increases for each student unit i. For stationary learnable situations with $K = M$, throughout learning the students units will acquire a full overlap to the teacher units: $S_i = 1$ for all student units i. In this configuration, the target rule has been fully learned which corresponds to perfect generalization error. In our model of real concept drift, the teacher vectors are changing continuously. This reduces the overlaps the student units can achieve with the teacher units, which increases the generalization error in the plateau state and the final state.

Identifying the specific teacher vectors is more difficult than learning the general structure of the teacher: Hence, increasing the drift causes the final generalization error to deteriorate faster than the plateau generalization error. For very strong target drift, the teacher vectors are changing too fast for specialization to be possible. We have identified the strength of the drift above which any kind of specialization is impossible for both SCM by studying the properties of the fixed point in the ODE. In stationary situations, one eigenvalue of the matrix defining the linearized dynamics near the fixed point is positive and causes the repulsion away from the fixed point to specialization, as discussed for the ReLU-SCM in Chapter 2. We refer to this positive eigenvalue as λ_s. We found that the eigenvalue decreases linearly with the drift parameter $\widetilde{\delta}$: For small $\widetilde{\delta}$, the eigenvalue λ_s is still positive and and the plateau can be escaped. However, the eigenvalue λ_s is negative for $\widetilde{\delta} > \widetilde{\delta}_c$, so that the symmetric fixed point is stable (attractor) and specialization becomes impossible. For the Erf-SCM, $\widetilde{\delta}_c \approx 0.0615$ and for the ReLU-SCM $\widetilde{\delta}_c \approx 0.225$. The weaker repulsion of the fixed point for stronger drift causes the plateau length to grow for $\widetilde{\delta} \to \widetilde{\delta}_c$. In practice, this implies that higher training effort is necessary the stronger the concept

drift is.

In the $\widetilde{\alpha} \to \infty$ final state, the student tracks the drifting target rule. For $\widetilde{\delta} \ll \widetilde{\delta}_c$, the student can achieve highly specialized states while tracking the teacher. The closer the drift strength is to $\widetilde{\delta}_c$, the weaker is the specialization that can be achieved by the student while following the rapidly moving teacher vectors. For $\widetilde{\delta} > \widetilde{\delta}_c$, the unspecialized student can only track the rule in terms of a simple approximation.

In the results of the Erf-SCM, a range of drift strength $0.036 < \widetilde{\delta} < \widetilde{\delta}_c$ was observed for which the final generalization error in the tracking state is worse than the plateau generalization error. Upon further inspection, this is due to the large values of Q_{11} and Q_{22} of the student vectors in the specialized regime. Hence, the effect can be prevented by introducing an appropriate weight decay.

Erf SCM vs. ReLU SCM: Weight decay in concept drift situations

Our results show that weight decay can improve the final generalization error in the specialized tracking state for both SCM. The suppression of the contributions of older and thus less representative data shows benefits in both systems.

However, from the result in Fig. 3.10b, we find that it is particularly important to tune the weight decay parameter for the Erf-SCM, since the specialization ability deteriorates quickly for values slightly off the optimum, as shown in the figure by the rapid increase in ϵ_g^∞. This reflects a steep decrease of the largest eigenvalue λ_s in the ODE for the Erf-SCM with increasing $\widetilde{\gamma}$, which also causes the increase of the plateau length as observed in Fig. 3.10d. Already from $\widetilde{\gamma}_c \approx 0.0255$, the eigenvalue λ_s becomes negative, and therefore the fixed point becomes an attractor.

We found a very different effect of weight decay on the performance of the ReLU-SCM. Not only is it able to improve the final generalization error in the tracking state as shown in Fig. 3.11b, it also significantly reduces the plateau length in the lower range of weight decay. This reflects the increase of λ_s with increasing weight decay in the fixed point of the ODE, which increases the repulsion from the unspecialized fixed point towards specialization. Clearly, suppressing the contribution of older data is beneficial for the specialization ability of the ReLU-SCM. For larger values of $\widetilde{\gamma}$, the plateau length increases, reflecting a decrease of λ_s. However, specialization remains possible up to a rather high value of weight decay: $\widetilde{\gamma}_c \approx 1.125$. The greater robustness to weight decay with respect to specialization as shown in Fig. 3.11b is likely related to our previous findings in (Straat and Biehl 2019) and Chapter 2, which show that the ReLU student-teacher setup needs fewer examples to reach specialization. We hypothesize that the simple linear nature of the function makes it easier for the student to learn features of the target rule. Hence a relatively small window of recent examples can already facilitate a degree of specialization.

3.5 Conclusions

Here we conclude with a brief summary, provide an outlook on potential follow-up studies and discuss challenges and open questions.

3.5.1 Brief Summary

In this contribution we presented a modelling framework which facilitates the systematic study and exact mathematical description of on-line learning in the presence of concept drift. We exemplified the versatile framework in terms of models for training of prototype-based classifiers (LVQ) and shallow neural networks for regression, respectively. Virtual drift in terms of changing class priors was formulated and studied for LVQ. Real drift, where the target classification or regression scheme was subject to a randomized drift process, was studied for both LVQ and the soft committee machine.

Most importantly, we demonstrated that the presented framework is suitable for the mathematical analysis of a variety of learning and drift scenarios, including weight decay as a possible mechanism of explicit *forgetting*.

3.5.2 LVQ for classification under drift and weight decay

In the real drift setting with a random displacement of cluster centers, we observed that the simple LVQ1 prescription is indeed capable of tracking time-dependent classification schemes in high-dimensional input space. Furthermore, we showed that weight decay has the potential to improve the generalization behavior under real drift in the quasi-stationary tracking state.

In the virtual drift setting, we observed that simple LVQ training can track the time-varying class bias to a non-trivial extent: In the interpretation of these results in terms of real drift, the class-conditional performance and the tracking error $\epsilon_{track}(\alpha)$ clearly reflect the time-dependence of the prior weights. In general, the reference error $\epsilon_{ref}(\alpha)$ with respect to class-balanced test data, displays only little deterioration due to the drift in the training data. In this case, the main effect of introducing weight decay is a reduced overall sensitivity to bias in the training data: Figs. 3.4-3.6 display a decreased difference between the class-wise errors $\epsilon^{1,2}$ for $\gamma > 0$. Naïvely, one might have expected an improved tracking of the drift due to the imposed *forgetting*, resulting in, for instance, a more rapid reaction to the sudden change of bias in Eq. (3.27). However, such an improvement cannot be confirmed. This finding is in contrast to the results obtained for the real drift with the randomized displacement of cluster centers, in which we observed increased performance by the use of weight decay.

The precise influence of weight decay clearly depends on the geometry and relative position of the clusters. Its dominant effect, however, is the regularization of the LVQ system by reducing the norms of the prototype vectors. Consequently, the NPC classifier is less flexible to reflect class bias which would require significant offset of the prototypes and decision boundary from the origin. This mildens the influence of the bias and its time-dependence and it results in a more robust behavior of the employed error measures.

3.5.3 SCM for regression under drift and weight decay

On-line gradient descent learning in the SCM has proven able to cope with drifting concepts in regression: For weak drifts, the SCM still achieves significant specialization with respect to the drifting teacher vectors, although the required learning time increases with the strength of the drift. In practice, this results in higher training effort to reach beneficial states in the network. The drift continuously reduces the overlaps with the teacher vectors which deteriorates the generalization performance. After reaching a specialized state, the network efficiently tracks the drifting target. However, in the presence of very strong drift, both versions of the SCM (with Erf- and ReLU-activation) lose their ability to specialize and as a consequence their generalization behavior remains poor.

We showed that weight decay can improve the generalization performance in the plateau and in the final tracking state. For the Erf-SCM, we found that there is a small range of values of the weight decay parameter in which favorable performance is achieved. Outside of this range, the network quickly loses the specialization ability. Therefore, in practice a careful tuning of the weight decay parameter would be required. The ReLU network shows greater robustness to the value of the weight decay parameter and displays a stronger tendency to specialize. Most importantly, weight decay reduces the plateau length significantly in the training of ReLU SCM. Hence, weight decay could speed up the training of ReLU networks in practical concept drift situations, achieving favorable weight configurations more quickly. Furthermore, the network performs well with a larger range of values of the weight decay parameter and does not require the careful tuning that is necessary in the case of the Erf-SCM.

3.5.4 Future work

The presented modelling framework offers the possibility to extend the scope of our studies in several relevant directions. For instance, the formalism facilitates the consideration of more complex model scenarios. Greater values of K and M should

be studied in, both, classification and regression. While we expect key results to carry over from $K = M = 2$, the greater complexity of the systems should result in richer dynamical behavior in detail. We will study if and how a mismatched number of prototypes further impedes the ability of LVQ systems to react appropriately to the presence of the virtual concept drift with changing class biases.

The training of an SCM with $K \neq M$ should be of considerable interest and will also be addressed in forthcoming studies. One might speculate that concept drift could enhance overfitting effects in over-sophisticated SCM with $K > M$ hidden units. Ultimately, the characteristic robustness and benefits of the ReLU activation function in the application of weight decay that was found should be studied in practical situations. Qualitative results are likely to carry over to similarly shaped activation functions, which will be verified in future work.

In a sense, the considered sigmoidal and ReLU activation functions are prototypical representatives of the most popular choices in machine learning practice. The extension to various modifications or significantly different transfer functions (Eger et al. 2018, Goodfellow et al. 2016) should provide additional valuable insights of practical relevance. Exact solutions to the averages that are required for the formulation of the learning dynamics in the thermodynamic limit may not be available for all activation functions. In such cases we can resort to approximations, for instance as covered in Chapter 2, and simulations.

The consideration of more complex input densities will also shed light on the practical relevance of our theoretical investigations. Recent work (Goldt et al. 2020, Loureiro et al. 2021) shows that the statistical physics based investigation of machine learning processes can take into account realistic input densities, bridging the gap between the theoretical models and practical applications.

Our modelling framework can also be applied in the analysis of other types of drift or combinations thereof. Several virtual processes could readily be implemented in the model of LVQ training: time-dependent characteristics of the input density could include the variances of the clusters or their relative position in feature space. A number of extensions is also possible in the regression model. For instance, teacher networks with time-dependent complexity could be studied by varying the mutual teacher overlaps $\boldsymbol{B}_m \cdot \boldsymbol{B}_n$ in the course of training.

Alternative mechanisms of *forgetting* beyond weight decay should be considered, which do not limit the flexibility of the trained systems as drastically. As one example strategy we intend to investigate the accumulation of additive noise in the training processes. We will also explore the parameter space of the model density in greater depth and study the influence of the learning rate systematically.

One of the major challenges in the field is the reliable detection of concept drift in a stream of data. Learning systems should be able to discriminate drift from static

noise in the data and infer also the type of drift, e.g. virtual vs. real. Moreover, the strength of the drift has to be estimated reliably in order to adjust the training prescription accordingly. It could be highly beneficial to extend our framework towards efficient drift detection and estimation procedures.

Other considerations for future work include:

- Alternative LVQ prescriptions, as studied in (Biehl et al. 2007, Biehl et al. 2005, Ghosh et al. 2005, Ghosh et al. 2006) for stationary data, can be systematically compared in terms of their potential to deal with concept drift.
- Similarly, modifications of the basic gradient descent scheme can be considered under concept drift in the SCM student-teacher scenario, see for instance (Saad 1999, Vicente and Caticha 1997, Inoue et al. 2003).
- Deterministic concept drifts, similar to the processes studied in the context of perceptron training in (Biehl and Schwarze 1992, Biehl and Schwarze 1993, Kinouchi and Caticha 1993, Vicente and Caticha 1998), can be considered as well. This way, learning from an *adversary* can be modelled, where the modification of the target depends explicitly on the actual student configuration.
- A systematic comparison and discussion of the N-dependence in computer experiments of LVQ under concept drift. For the SCM, the precise influence of finite size effects on the shape and length of plateau in Monte Carlo simulations.
- For LVQ and SCM: The simultaneous optimization of learning rate and weight decay $\{\eta, \gamma\}$ with respect to the success of training in the *tracking state*.

3.5.5 Perspectives and Challenges

We have demonstrated that the presented modelling framework bears the promise to provide valuable insights into the effects of concept drift in a variety of learning scenarios. Ultimately, a better understanding of relevant phenomena should facilitate the development and optimization of robust, efficient training algorithms for lifelong machine learning. Variational approaches as discussed in, for instance, (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993, Biehl and Caticha 2003, Kinouchi and Caticha 1993, Vicente and Caticha 1998, Vicente and Caticha 1997) could play an important role in this context.

Recently suggested strategies for continual learning include so-called *Dedicated Memory Models* and the appropriate combination of off-line and on-line learning (Losing et al. 2018, Fischer et al. 2015, Fischer et al. 2016). Suitable rejection mechanisms for the mitigation of concept drift were recently considered in (Göpfert

et al. 2018). Extensions of our modelling approach in these directions would be highly desirable.

Published as:

E. Oostwal, M. Straat, M. Biehl – *"Hidden unit specialization in layered neural networks: ReLU vs. sigmoidal activation"*, Physica A: Statistical Mechanics and its Applications, vol. 564, art. no. 125517, 2021.

Chapter 4

Off-line Learning in Layered Networks: ReLU vs. Sigmoidal Activation

Abstract

We study layered neural networks of rectified linear units (ReLU) in a modelling framework for stochastic training processes on datasets of a fixed training set size (off-line learning). The comparison with sigmoidal activation functions is in the center of interest. We compute typical learning curves for shallow networks with K hidden units in matching student teacher scenarios. As confirmed qualitatively by Monte Carlo simulations, the systems exhibit sudden changes of the generalization performance via the process of hidden unit specialization at critical sizes of the training set. Surprisingly, our results show that the training behavior of ReLU networks is qualitatively different from that of networks with sigmoidal activations. In networks with $K \geqslant 3$ sigmoidal hidden units, the transition is discontinuous: Specialized network configurations co-exist and compete with states of poor performance even for very large training sets. On the contrary, the use of ReLU activations results in continuous transitions for all K. For large enough training sets, two competing, differently specialized states display similar generalization abilities, which coincide exactly for large networks in the limit $K \to \infty$.

4.1 Introduction

The statistical physics of learning has been applied with great success in the context of neural networks and machine learning in general, e.g. (Hertz et al. 1991, Seung et al. 1992, Watkin et al. 1993, Kinzel 1998, Opper 1994, Biehl and Caticha 2003, Engel and van den Broeck 2001) and the analyses in the previous chapters. The statistical physics approach complements other theoretical frameworks in that it studies the *typical behavior* of large learning systems in model scenarios.

Currently, the statistical physics of learning is being revisited extensively in order to investigate relevant phenomena in deep neural networks and other learning paradigms, see (Cocco et al. 2018, Kadmon and Sompolinsky 2016, Pankaj et al. 2014,

Sohl-Dickstein et al. 2016, Caticha et al. 2016, Biehl et al. 2019, Baldassi et al. 2019, Goldt et al. 2020) for recent examples and further references.

The aim of this chapter is to contribute to a better theoretical understanding of how the use of ReLU activations influences and potentially improves the training behavior of layered neural networks. In contrast to our previous analyses, here we focus on the comparison of ReLU activation with traditional sigmoidal functions in non-trivial model situations of *off-line learning*. A topic of particular interest for this work is the analysis of phase transitions in learning processes, i.e. sudden changes of the expected performance with the training set size or other control parameters, see (Kinzel 1998, Opper 1994, Herschkowitz and Opper 2001, Kang et al. 1993, Biehl et al. 2000, Biehl, Schlösser and Ahr 1998, Ahr et al. 1999, Saitta et al. 2011) for examples and further references. We systematically study the training of layered networks in student teacher settings similar to the settings in Chapter 2 and 3, see also e.g. (Seung et al. 1992, Watkin et al. 1993, Opper 1994, Engel and van den Broeck 2001). We consider idealized, yet non-trivial scenarios of matching student and teacher complexity.

Our findings demonstrate that ReLU networks display training and generalization behavior which differs significantly from their counterparts composed of sigmoidal units. Both network types display sudden changes of their performance with the number of available examples. In statistical physics terminology, the systems undergo phase transitions at a critical training set size. The underlying process of hidden unit specialization and the existence of saddle points in the objective function have recently attracted attention also in the context of Deep Learning (Kadmon and Sompolinsky 2016, Dauphin et al. 2014, Saxe et al. 2014).

Before analysing ReLU networks, we confirm earlier theoretical results which indicate that the transition for large networks of sigmoidal units is discontinuous *(first order)*: For small training sets, a poorly generalizing state is observed, in which all hidden units approximate the target to some extent and essentially perform the same task. At a critical size of the training set, a favorable configuration with specialized hidden units appears. However, a poorly performing state remains metastable and the specialization required for successful learning can delay the training process significantly (Kang et al. 1993, Biehl, Schlösser and Ahr 1998, Ahr et al. 1999, Biehl et al. 2000).

In contrast we find that, surprisingly, the corresponding phase transition in ReLU networks is always continuous *(second order)*. At the transition, the unspecialized state is replaced by two competing configurations with very similar generalization ability. In large networks, their performance is nearly identical and it coincides exactly in the limit $K \to \infty$.

In the next section we detail the considered models and outline the theoretical

approach. In Sec. 3.4 our results are presented and discussed. In addition, results of supporting Monte Carlo simulations are presented. We conclude with a summary and outlook on future extensions of this work.

4.2 Model and Analysis

Here we introduce the modelling framework, i.e. the considered student teacher scenarios. Moreover, we outline their analysis by means of statistical physics methods and discuss the simplifying assumption of training at high (formal) temperatures.

4.2.1 Network architecture and activation functions

We consider feed-forward neural networks where N input nodes represent feature vectors $\boldsymbol{\xi} \in \mathbb{R}^N$. A single layer of K hidden units is connected to the input through adaptive weights $\underline{W} = \left\{\mathbf{w}_k \in \mathbb{R}^N\right\}_{k=1}^K$. The total real-valued output reads

$$y(\boldsymbol{\xi}) = \frac{1}{\sqrt{K}} \sum_{k=1}^{K} g\left(h_k\right) \quad \text{with} \quad h_k = \frac{1}{\sqrt{N}} \mathbf{w}_k \cdot \boldsymbol{\xi}. \tag{4.1}$$

The quantity h_k is referred to as the *local potential* of the hidden unit. The resulting activation is specified by the function $g(x)$. In contrast to the definition used in Chapters 2 and 3, the hidden to output weights are fixed to $1/\sqrt{K}$ instead of one. This is done to make the variance of y independent from the number of hidden units K, facilitating the analyses for large K in this chapter. Additionally, in this chapter it is assumed that the weight vectors have a norm $O(\sqrt{N})$, which requires the scaling of the inner products in Eq. (4.1) by $\sqrt{N}$. Figure 4.1 (left panel) illustrates the network architecture together with indications of the key quantities.

This type of network has been termed the *Soft Committee Machine* (SCM) in the literature due to its vague similarity to the committee machine for binary classification, e.g. (Engel and van den Broeck 2001, Watkin et al. 1993, Urbanczik 1997, Schwarze and Hertz 1993, Opper 1994, Herschkowitz and Opper 2001, Baldassi et al. 2019). There, the discrete output is determined by the majority of threshold units in the hidden layer, while the SCM is suitable for regression tasks.

We will consider two popular types of transfer functions:

a) **Sigmoidal activation**

Frequently, S-shaped transfer functions $g(x)$ have been employed, which increase monotonically from *zero* at large negative arguments and saturate at a finite maximum for $x \to \infty$. Popular examples are based on $\tanh(x)$ or the

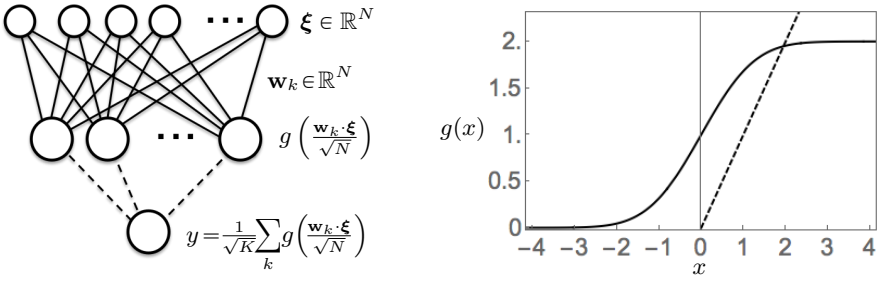


Figure 4.1: **Left panel:** Illustration of the network architecture with an N-dim. input layer, a set of adaptive weight vectors $\mathbf{w}_k$ with $k = 1, \ldots, K$ (represented by solid lines) and total output y given by the sum of hidden unit activations with fixed weights (dashed lines). **Right panel:** The considered activation functions: the sigmoidal $g(x) = \left(1 + \text{erf}[x/\sqrt{2}]\right)$ (solid line) and the ReLU activation $g(x) = \max\{0, x\}$ (dashed line).

sigmoid $(1 + e^{-x})^{-1}$, often with an additional threshold θ as in $g(x - \theta)$, or a steepness parameter controlling the magnitude of the derivative g'.

We study the particular choice

$$g(x) = \left(1 + \text{erf}\left[\frac{x}{\sqrt{2}}\right]\right) = 2 \int_{-\infty}^{x} dz \frac{e^{-z^2/2}}{\sqrt{2\pi}} \tag{4.2}$$

with $0 \leqslant g(x) \leqslant 2$, which is displayed in the right panel of Fig. 4.1. The relation to an integrated Gaussian facilitates significant mathematical ease, which has been exploited in numerous studies of machine learning models, e.g. (Biehl and Schwarze 1995, Saad and Solla 1995a, Saad and Solla 1995b). Here, the function (4.2) serves as a generic example of a sigmoidal and its specific form is not expected to influence our findings crucially. As we argue below, the choice of limiting values 0 and 2 for small and large arguments, respectively, is also arbitrary and irrelevant for the qualitative results of our analyses.

b) **Rectified Linear Unit (ReLU) activation**

This particularly simple, piece-wise linear transfer function has attracted considerable attention in the context of multi-layered neural networks. It is given by

$$g(x) = \max\{0, x\} = \begin{cases} 0 & \text{for } x \leqslant 0 \\ x & \text{for } x > 0 \end{cases} \tag{4.3}$$

which is illustrated in Fig. 4.1 (right panel). In contrast to sigmoidal activations, the response of the unit is unbounded for $x \to \infty$. The function (2.3) is obviously not differentiable in $x = 0$, as was already discussed in Chapter 2. Note that our theoretical investigation in Sec. 4.2.4 does not relate to a particular realization of gradient-based training.

It is important to realize that replacing the above functions by

$$g(x) = \gamma \left(1 + \mathrm{erf}[x/\sqrt{2}]\right) \quad \text{or by} \quad g(x) = \max\{0, \gamma\, x\} = \gamma \max\{0, x\} \tag{4.4}$$

where $\gamma > 0$ is an arbitrary factor, would be equivalent to setting the hidden unit to output weights to $\gamma/\sqrt{K}$ in Eq. (4.1). Alternatively, we could incorporate the factor γ in the effective temperature parameter α of the theoretical analysis in Sec. 4.2.4. Apart from this trivial re-scaling, our results would not be affected qualitatively.

4.2.2 Student and teacher scenario

We investigate the training and generalization behavior of the layered networks introduced above in a setup that models the learning of a regression scheme from example data. Assume that a given training set

$$\mathbb{D} = \{\boldsymbol{\xi}^\mu, \tau^\mu\}_{\mu=1}^P \tag{4.5}$$

comprises P input output pairs which reflect the target task. In order to facilitate successful learning, P should be proportional to the number of adaptive weights in the trained system. Similar to the previous chapter, in our specific model scenario the labels $\tau^\mu = \tau(\boldsymbol{\xi}^\mu)$ are thought to be provided by a teacher SCM, representing the target input output relation

$$\tau(\boldsymbol{\xi}) = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} g\left(b_m\right) \quad \text{with} \quad b_m = \frac{1}{\sqrt{N}} \boldsymbol{B}_m \cdot \boldsymbol{\xi}\,. \tag{4.6}$$

The response is specified in terms of the set of teacher weight vectors

$$\underline{B} = \left\{\boldsymbol{B}_m \in \mathbb{R}^N\right\}_{m=1}^M \tag{4.7}$$

and defines the correct target output for every possible feature vector $\boldsymbol{\xi}$. For simplicity, we will focus on settings with orthonormal teacher weight vectors and restrict the adaptive student configuration to normalized weights:

$$\boldsymbol{B}_m \cdot \boldsymbol{B}_n/N = \delta_{mn} \quad \text{and} \quad |\boldsymbol{w}_j|^2 = N \tag{4.8}$$

with the Kronecker-Delta $\delta_{mn} = 0$ if $m \neq n$ and $\delta_{mm} = 1$.

Throughout the following, the evaluation of the student network will be based on a simple quadratic error measure that compares student output and target value. Accordingly, the selection of student weights $\underline{W}$ in the training process is guided by a cost or loss function which is given by the corresponding sum over all available data in the training set $\mathbb{D}$:

$$E = \sum_{\mu=1}^{P} \epsilon(\boldsymbol{\xi}^\mu) \;\text{ with }\; \epsilon(\boldsymbol{\xi}) = \frac{1}{2}\Big(y(\boldsymbol{\xi}) - \tau(\boldsymbol{\xi})\Big)^2 . \tag{4.9}$$

By choosing the parameters K and M, a variety of situations can be modelled, as we discussed and studied in the modelling of on-line learning in Chapter 2. There, we studied the learning of unrealizable rules ($K < M$) and training of over-sophisticated students with $K > M$. In the current chapter that analyses off-line learning, we restrict ourselves to the idealized, yet non-trivial case of perfectly matching student and teacher complexity, i.e. $K = M$, which makes it possible to achieve $\epsilon(\boldsymbol{\xi}) = 0$ for all input vectors.

4.2.3 Generalization error and order parameters

Here we briefly give the definitions of the data distribution, the order parameters of the SCM and the generalization error, which are similar to the definitions as given in Chapter 2 besides different scalings and the applicability to the off-line learning setting. We also introduce the assumption of *site symmetry* and give the equations of the generalization error for Erf-SCM and ReLU-SCM under this assumption.

Throughout the following we consider feature vectors $\boldsymbol{\xi}^\mu$ in the training set with uncorrelated i.i.d. random components of zero mean and unit variance. Likewise, arbitrary input vectors $\boldsymbol{\xi} \notin \mathbb{D}$ are assumed to follow the same statistics:

$$\left\langle \xi_j^\mu \right\rangle = 0, \left\langle \xi_j^\mu \xi_k^\nu \right\rangle = \delta_{j,k}\delta_{\mu,\nu}, \left\langle \xi_j \right\rangle = 0 \text{ and } \left\langle \xi_j \xi_k \right\rangle = \delta_{j,k} .$$

As a consequence of this assumption, the Central Limit Theorem applies to the local potentials

$$h_j = \boldsymbol{w}_j \cdot \boldsymbol{\xi}/\sqrt{N} , \quad b_m = \boldsymbol{B}_m \cdot \boldsymbol{\xi}/\sqrt{N} , \qquad \text{for } 1 \leqslant j, m \leqslant K$$

which become correlated Gaussian random variables of order $\mathcal{O}(1)$. It is straightforward to work out the characteristic averages $\langle \ldots \rangle$ and (co-)variances:

$$\begin{aligned} &\langle h_k \rangle = \langle b_k \rangle = 0, \; \langle h_j h_k \rangle = \mathbf{w}_j \cdot \mathbf{w}_k/N \equiv Q_{jk}, \\ &\langle b_j b_k \rangle = \mathbf{B}_j \cdot \mathbf{B}_k/N = \delta_{jk} \text{ and } \langle h_j b_k \rangle = \mathbf{w}_j \cdot \mathbf{B}_k/N \equiv R_{jk}, \end{aligned} \tag{4.10}$$

which fully specify the joint density $P(\{h_i, b_i\})$. The so-called order parameters R_{ij} and Q_{ij} for $(i, j = 1, 2, \dots K)$ serve as macroscopic characteristics of the student configuration. The norms $Q_{ii} = 1$ are fixed according to Eq. (4.8), while the symmetric $Q_{ij} = Q_{ji}$ quantify the $K(K-1)/2$ pairwise alignments of student weight vectors. The similarity of the student weights to their counterparts in the teacher network are measured in terms of the K^2 quantities R_{ij}. Due to the assumed normalizations, the relations $-1 \leqslant Q_{ij}, R_{ij} \leqslant 1$ are obviously satisfied.

Now we can work out the generalization error, i.e. the expected deviation of student and teacher output for a random input vector, given specific weight configurations $\underline{W}$ and $\underline{B}$. Note that SCMs with $g(x) = \mathrm{erf}[x/\sqrt{2}]$ have been treated in (Saad and Solla 1995a, Saad and Solla 1995b) for general K, M. Here, we resort to the special case of matching network sizes, $K = M$, with

$$\epsilon_g \;=\; \frac{1}{2K} \left\langle \left(\sum_{i=1}^{K} g(h_i) - \sum_{j=1}^{K} g(b_j) \right)^2 \right\rangle . \tag{4.11}$$

We note here that matching additive constants in the student and teacher activations would leave ϵ_g unaltered. As detailed in the thesis' Appendix, all averages in Eq. (4.11) can be computed analytically for both choices of the activation function $g(x)$ in student and teacher network. Eventually, the generalization error is expressed in terms of very few macroscopic order parameters, instead of explicitly taking into account KN individual weights. The concept is characteristic for the statistical physics approach to systems with many degrees of freedom.

Site-symmetry assumption

In the following, we restrict the analysis to student configurations which are site-symmetric with respect to the hidden units:

$$R_{ij} = \begin{cases} R & \text{for } i = j \\ S & \text{otherwise}\,, \end{cases} \qquad Q_{ij} = \begin{cases} 1 & \text{for } i = j \\ C & \text{otherwise} \end{cases} . \tag{4.12}$$

Obviously, the system is invariant under permutations, so we can restrict ourselves to one specific case with matching indices $i = j$ in Eq. (4.12). While this assumption reflects the symmetries of the student teacher scenario, it allows for the *specialization* of hidden units: For $R = S$ all student units display the same overlap with all teacher units. In specialized configurations with $R \neq S$, however, each student weight vector has achieved a distinct overlap with exactly one of the teacher units. Our analysis shows that states with both positive ($R > S$) and negative specialization ($R < S$) can play a significant role in the training process.

Under the above assumption of site-symmetry (4.12) and applying the normalization (4.8), the generalization error (4.11), see also Eqs. (A.3) and (A.5), becomes

a) for $g(x) = \left(1+\mathrm{erf}\left[x/\sqrt{2}\right]\right)$ in student and teacher (Saad and Solla 1995a):

$$\epsilon_g = \frac{1}{K}\left\{\frac{1}{3} + \frac{K-1}{\pi}\left[\sin^{-1}\left(\frac{C}{2}\right) - 2\sin^{-1}\left(\frac{S}{2}\right)\right] - \frac{2}{\pi}\sin^{-1}\left(\frac{R}{2}\right)\right\}, \tag{4.13}$$

Note that additional pre-factors in the sigmoidal activation function (4.2), for instance as to achieve $0 < g(x) < 1$, would only yield a factor in ϵ_g and could be compensated for by re-scaling the number of examples, accordingly, see Section 4.2.4.

b) for ReLU $g(x) = \max\{0, x\}$:

$$\begin{aligned}\epsilon_g \;=\; & \frac{1}{2K}\left\{K + \frac{K(K-1)}{2\pi} - 2K\left(\frac{R}{4} + \frac{\sqrt{1-R^2}}{2\pi} + \frac{R\sin^{-1}(R)}{2\pi}\right)\right. \\ & + K(K-1)\left(\frac{C}{4} + \frac{\sqrt{1-C^2}}{2\pi} + \frac{C\sin^{-1}(C)}{2\pi}\right) \\ & \left. - 2K(K-1)\left(\frac{S}{4} + \frac{\sqrt{1-S^2}}{2\pi} + \frac{S\sin^{-1}(S)}{2\pi}\right)\right\}. \end{aligned} \tag{4.14}$$

In both settings, perfect agreement of student and teacher with $\epsilon_g = 0$ is achieved for $C = S = 0$ and $R = 1$. The scaling of outputs with hidden to output weights $1/\sqrt{K}$ in Eq. (4.1) results in a generalization error which is not explicitly K-dependent for uncorrelated random students: A configuration with $R = C = S = 0$ yields $\epsilon_g = 1/3$ in the case of sigmoidal activations (a), whereas $\epsilon_g = \frac{1}{2} - \frac{1}{2\pi} \approx 0.341$ for ReLU student and teacher.

4.2.4 Thermal equilibrium and the high-temperature limit

In order to analyse the expected outcome of training from a set of examples $\mathbb{D}$, we follow the well-established statistical physics approach and analyse an *ensemble* of networks in a formal *thermal equilibrium* situation. In this framework, the cost function E is interpreted as the *energy* of the system and the density of observed network states is given by the so-called Gibbs-Boltzmann density

$$\frac{\exp[-\beta E]}{Z} \quad \text{with} \quad Z = \int d\mu(\underline{W}) \exp[-\beta E], \tag{4.15}$$

where the measure $d\mu(\underline{W})$ incorporates potential restrictions of the integration over all possible configurations of $\underline{W} = \{\mathbf{w}_i\}_{i=1}^{K}$, for instance the normalization $\mathbf{w}_k^2 = N$ for all k. This equilibrium density would, for example, result from a Langevin type of training dynamics

$$\frac{\partial \underline{W}}{\partial t} \;=\; -\nabla_W E(\underline{W}) \;+ \underline{\eta},$$

where ∇_W denotes the gradient with respect to all KN degrees of freedom in the student network. Here, the minimization of E is performed in the presence of a δ-correlated, zero mean noise term $\underline{\eta}(t) \in \mathbb{R}^{KN}$ with

$$\langle \eta_i(t) \rangle = 0 \text{ and } \langle \eta_i(t)\eta_j(t') \rangle = \frac{2}{\beta}\delta_{ij}\delta(t-t') \,,$$

where $\delta(\ldots)$ denotes the Dirac delta-function. The parameter $\beta = 1/T$ controls the strength of the *thermal noise* in the gradient-based minimization of E.

According to the, by now, standard statistical physics approach to off-line learning (Hertz et al. 1991, Engel and van den Broeck 2001, Watkin et al. 1993, Seung et al. 1992) typical properties of the system are governed by the so-called *quenched free energy*

$$f = -\frac{1}{N} \langle \ln Z \rangle_{\mathbb{D}} / \beta \tag{4.16}$$

where $\langle \ldots \rangle_{\mathbb{D}}$ denotes the average over the random realization of the training set. In general, the evaluation of the quenched average $\langle \ln Z \rangle_{\mathbb{D}}$ is technically involved and requires, for instance, the application of the replica trick (Hertz et al. 1991, Engel and van den Broeck 2001, Watkin et al. 1993). Here, we resort to the simplifying limit of training at high temperature $T \to \infty, \beta \to 0$, which has proven useful in the qualitative investigation of various learning scenarios (Seung et al. 1992). In the limit $\beta \to 0$ the so-called annealed approximation (Seung et al. 1992, Watkin et al. 1993, Engel and van den Broeck 2001) $\langle \ln Z \rangle_{\mathbb{D}} \approx \ln \langle Z \rangle_{\mathbb{D}}$ becomes exact. Moreover, we have

$$\langle Z \rangle_{\mathbb{D}} = \left\langle \int d\mu(\underline{W}) e^{-\beta E} \right\rangle_{\mathbb{D}} \approx \int d\mu(\underline{W}) e^{-\beta \langle E \rangle_{\mathbb{D}}} . \tag{4.17}$$

Here, P is the number of statistically independent examples in $\mathbb{D}$ and $\langle E \rangle_{\mathbb{D}} = P \langle \epsilon(\boldsymbol{\xi}) \rangle_{\xi} = P\epsilon_g$. As the exponent grows linearly with $P \propto N$, the integral is dominated by the maximum of the integrand. By means of a saddle-point integration for $N \to \infty$ we obtain

$$-\frac{1}{N} \ln \langle Z \rangle_{\mathbb{D}} = \beta f(\{R_{ij}, Q_{ij}\}) \approx \frac{\beta P}{N}\epsilon_g - s. \tag{4.18}$$

Here, the right hand side has to be minimized with respect to the arguments, i.e. the order parameters $\{R_{ij}, Q_{ij}\}$. In Eq. (4.18) we have introduced the entropy term

$$s = \tfrac{1}{N} \ln \textstyle\int d\mu(\underline{W}) \prod_{i,j} [\delta(NR_{ij} - \mathbf{w}_i \cdot \mathbf{B}_j)\delta(NQ_{ij} - \mathbf{w}_i \cdot \mathbf{w}_j)] \,. \tag{4.19}$$

The quantity e^{Ns} corresponds to the volume in weight space that is consistent with a given configuration of order parameters. Independent of the activation functions or

other details of the learning problem, one obtains for large N (Biehl, Schlösser and Ahr 1998, Ahr et al. 1999)

$$s(\{R_{ij}, Q_{ij}\}) = \frac{1}{2} \ln \det (\mathcal{C}) + \text{const}\,. \tag{4.20}$$

where $\mathcal{C}$ is the $(2K \times 2K)$-dimensional matrix of all pair-wise and self-overlaps of the vectors $\{\mathbf{w}_i, \mathbf{B}_i\}_{i=1}^{K}$, i.e. the matrix of all $\{R_{ij}, Q_{ij}, T_{ij}\}$, see also Eq. (A.1) in the Appendix. The constant term is independent of the order parameters and, hence, irrelevant for the minimization of Eq. (4.18). A compact derivation of (4.20) is provided in, e.g., (Ahr et al. 1999).

Omitting additive constants and assuming the normalization (4.8) and site-symmetry (4.12), the entropy term reads (Biehl, Schlösser and Ahr 1998, Ahr et al. 1999)

$$\begin{aligned} s \;\; = \;\; & \frac{1}{2} \ln \Big[1 + (K-1)C - \big((R-S) + KS\big)^2 \Big] \\ & + \tfrac{K-1}{2} \ln \big[1 - C - (R-S)^2 \big] \,. \end{aligned} \tag{4.21}$$

In order to facilitate the successful adaptation of KN weights in the student network we have to assume that the number of examples also scales like $P = \tilde{\alpha}\, K\, N$. Training at high temperature additionally requires that $\alpha = \tilde{\alpha}\beta = \mathcal{O}(1)$ for $\tilde{\alpha} \to \infty, \beta \to 0$, which yields a free energy of the form

$$\beta\, f(R, S, C) \;=\; \alpha\, K\, \epsilon_g(R, S, C) \;-\; s(R, S, C)\,. \tag{4.22}$$

The quantity $\alpha = \beta P/(KN)$ can be interpreted as an effective temperature parameter or, likewise, as the properly scaled training set size. The high temperature has to be compensated by a very large number of training examples in order to facilitate non-trivial outcome. As a consequence, the energy of the system is proportional to ϵ_g, which implies that training and generalization error are effectively identical in the simplifying limit.

4.3 Results and Discussion

In the following, we present and discuss our findings for the considered student teacher scenarios and activation functions.

In order to obtain the equilibrium states of the model for given values of α and K, we have minimized the scaled free energy (4.22) with respect to the site-symmetric order parameters. Potential (local) minima satisfy the necessary conditions

$$\frac{\partial(\beta f)}{\partial R} = \frac{\partial(\beta f)}{\partial C} = \frac{\partial(\beta f)}{\partial S} = 0\,. \tag{4.23}$$

In addition, the corresponding Hesse matrix $\mathcal{H}$ of second derivatives w.r.t. R, S, and C has to be positive definite. This constitutes a sufficient condition for the presence of a local minimum in the site-symmetric order parameter space. Furthermore, we have confirmed the stability of the local minima against potential deviations from site-symmetry by inspecting the full matrix of second derivatives involving the $(K^2+K(K-1)/2)$ individual quantities $\{R_{ij}, Q_{ij} = Q_{ji}\}$.

4.3.1 Sigmoidal units re-visited

The investigation of SCMs with sigmoidal $g(x) = \mathrm{erf}[x/\sqrt{2}]$ with $-1 < g(x) < 1$ along the lines of the previous section has already been presented in (Biehl, Schlösser and Ahr 1998). A corresponding model with discrete binary weights was studied in (Kang et al. 1993).

As argued above, for $g(x) = (1 + \mathrm{erf}[x/\sqrt{2}])$, the mathematical form of the generalization error, Eqs. (4.13, A.3), and the free energy (βf) are the same as for the activation $\mathrm{erf}[x/\sqrt{2}]$. Hence, the results of (Biehl, Schlösser and Ahr 1998) carry over without modification. The following summarizes the key findings of the previous study, which we reproduce here for comparison.

For $K = 2$ we observe that $R = S$ in thermal equilibrium for small α, see the upper row of graphs in Fig. 4.2. Both hidden units perform essentially the same task and acquire equal overlap with both teacher vectors, when trained from relatively small data sets. At a *critical* value $\alpha_c(2) \approx 23.7$, the system undergoes a transition to a specialized state with $R > S$ or $R < S$ in which each hidden unit aligns with one specific teacher unit. Both configurations are fully equivalent due to the invariance of the student output under exchange of the student weights $\mathbf{w}_1$ and $\mathbf{w}_2$ for $K = 2$. The specialization process is continuous with the quantity $|R-S|$ increasing proportional to $(\alpha - \alpha_c(K))^{1/2}$ near the transition. This results in a *kink* in the continuous learning curve $\epsilon_g(\alpha)$ at α_c, as displayed in the upper right panel of Fig. 4.2.

Interestingly, a different behavior is found for all $K \geqslant 3$. The following regimes can be distinguished:

(a) $0 \leqslant \alpha < \alpha_s(K)$: For small α, the only minimum of βf corresponds to unspecialized networks with $R = S$. Within this subspace, a rapid initial decrease of ϵ_g with α is achieved.

(b) $\alpha_s(K) \leqslant \alpha < \alpha_c(K)$: In $\alpha_s(K)$, a specialized configuration with $R > S$ appears as a local minimum of the free energy. The $R = S$ configuration corresponds to the global minimum up to $\alpha_c(K)$. At this K-dependent critical value, the free energies of the competing minima coincide.

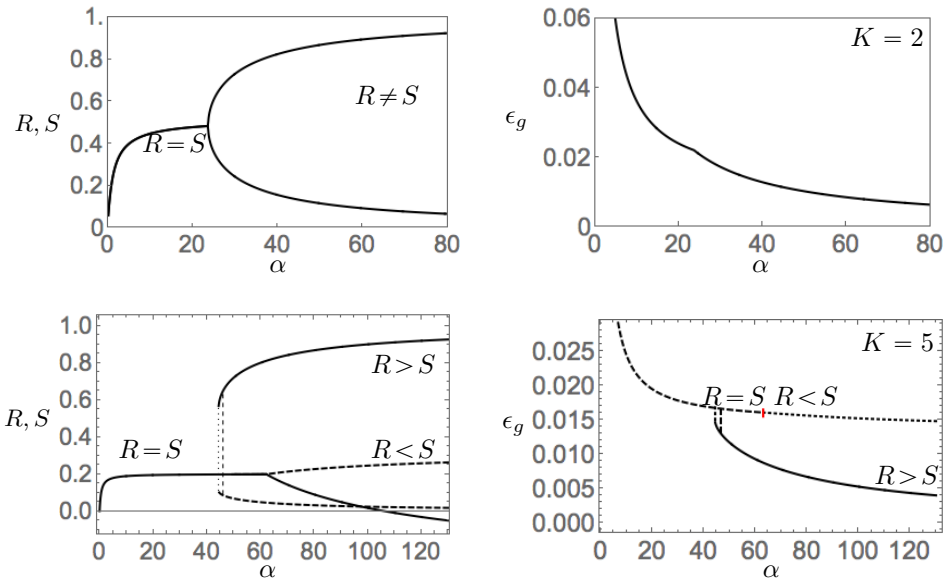


Figure 4.2: **Sigmoidal activation:** Learning curves of perfectly matching student teacher scenarios. The **upper row** of graphs shows the results for $K = 2$: The upper left panel displays the order parameters R and S as functions of $\alpha = \beta P/(KN)$. At the critical value $\alpha_c(2) \approx 23.7$, a continuous phase transition occurs with $|R - S| > 0$ for greater values of α. The upper right panel shows the corresponding learning curve $\epsilon_g(\alpha)$ which displays a *kink* at the transition. In the **lower row** the corresponding results are shown for $K = 5$. Order parameters are displayed in the lower left panel as functions of α. The transition is discontinuous with $\alpha_s(5) \approx 44.3$ (vertical dotted line) and $\alpha_c(5) \approx 46.6$ (vertical dashed line). In addition, the local minimum of the free energy with $R = S$ is replaced by a configuration with $R < S$ in $\alpha_d(5) \approx 62.8$. The lower right panel displays the corresponding $\epsilon_g(\alpha)$. Here, the solid line represents the specialized state and the transition from the unspecialized configuration (dashed curve) to the state with $R < S$ (dotted curve) is marked by the short line in α_d. The dashed vertical line marks the critical α_c, where the specialized $(R > S)$ solution becomes the global minimum of (βf).

(c) $\alpha > \alpha_c(K)$: Above α_c, the configuration with $R > S$ constitutes the global minimum of the free energy and, thus, the thermodynamically stable state of the system. Note that the transition from the unspecialized to the specialized configuration is associated with a discontinuous change of ϵ_g, cf. Fig. 4.2 (lower right panel). The $(R > S)$ specialized state facilitates perfect generalization in

the limit $\alpha \to \infty$.

(d) $\alpha \geqslant \alpha_d(K)$: In addition, at another characteristic value α_d, the $(R = S)$ local minimum disappears and is replaced by a negatively specialized state with $R < S$. Note that the existence of this local minimum of the free energy was not reported in (Biehl, Schlösser and Ahr 1998). The observed specialization $(S-R)$ increases linearly with $(\alpha - \alpha_d)$ for $\alpha \approx \alpha_d$. This smooth transition does not yield a *kink* in $\epsilon_g(\alpha)$. A careful analysis of the associated Hesse matrix shows that the $R < S$ state of poor generalization persists for all $\alpha > \alpha_d$, indeed.

The limit $K \to \infty$ with $K \ll N$ has also been considered in (Biehl, Schlösser and Ahr 1998): The discontinuous transition is found to occur at $\alpha_s(K \to \infty) \approx 60.99$ and $\alpha_c(K \to \infty) \approx 69.09$. Interestingly, the characteristic value α_d diverges as $\alpha_d(K) = 4\pi K$ for large K (Biehl, Schlösser and Ahr 1998). Hence, the additional transition from $R = S$ to $R < S$ cannot be observed for data sets of size $P \propto KN/\beta$. On this scale, the unspecialized configuration persists for $\alpha \to \infty$. It displays site-symmetric order parameters $R = S = \mathcal{O}(1/K)$ with $R, S > 0$ and $C = \mathcal{O}(1/K^2)$, see (Biehl, Schlösser and Ahr 1998) for details. Asymptotically, for $\alpha \to \infty$, they approach the values $R = S = 1/K$ and $C = 0$ which yields the non-zero generalization error $\epsilon_g(\alpha \to \infty) = 1/3 - 1/\pi \approx 0.0150$. On the contrary, the $R > S$ specialized configuration achieves $\epsilon_g \to 0$, i.e. perfect generalization, asymptotically.

The presence of a discontinuous specialization process for sigmoidal activations with $K \geqslant 3$ suggests that – in practical training situations – the network will very likely be trapped in an unfavorable configuration unless prior knowledge about the target is available. The escape from the poorly generalizing metastable state with $R = S$ or $R < S$ requires considerable effort in high-dimensional weight space. Therefore, the success of training will be delayed significantly.

4.3.2 Rectified linear units

In comparison with the previously studied case of sigmoidal activations, we find a surprisingly different behavior in ReLU networks with $K \geqslant 3$.

For $K = 2$, our findings parallel the results for networks with sigmoidal units: The network configuration is characterized by $R = S$ for $\alpha < \alpha_c(K)$ and specialization increases like

$$|R - S| \propto (\alpha - \alpha_c(K))^{1/2} \tag{4.24}$$

near the transition. This results in a *kink* in the learning curve $\epsilon_g(\alpha)$ at $\alpha = \alpha_c(K)$ as displayed in Fig. 4.3 (upper row) for $K = 2$ with $\alpha_c(2) \approx 6.1$.

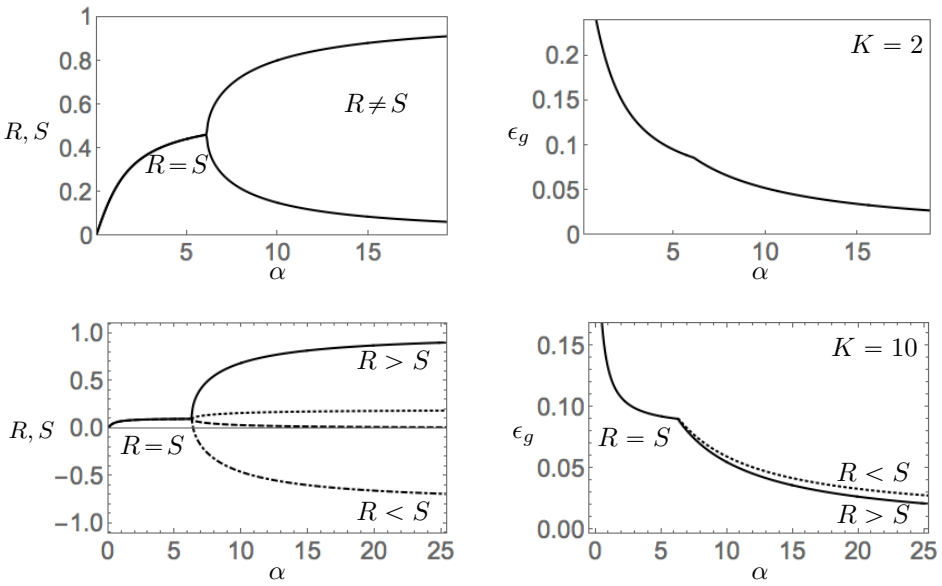


Figure 4.3: **ReLU activation:** Learning curves of perfectly matching student teacher scenarios. The **upper row** of graphs shows the results for $K = 2$. The upper left panel displays the order parameters R and S as a function of $\alpha = \beta P/(KN)$. A continuous transition occurs at $\alpha_c(2) \approx 6.1$, the right panel shows the resulting $\epsilon_g(\alpha)$. The **lower row** corresponds to the ReLU network with $K = 10$. The transition is also continuous and occurs at $\alpha_c(10) \approx 6.2$. The specialized solution with $R > S$ is represented by the solid (R) and the dashed line (S) in the lower left panel. The dotted line (S) and the chain line (R) represent the local minimum of βf with $R < S$. The corresponding generalization errors is displayed in the lower right panel, where the dotted line represents the suboptimal configuration with $R < S$.

However, in ReLU networks the transition is also continuous for $K \geqslant 3$. Figure 4.3 (lower row of graphs) displays the results for the example case $K = 10$ with $\alpha_c(10) \approx 6.2$ (lower row).

The student output is invariant under exchange of the hidden unit weight vectors, consistent with an $R = S$ unspecialized state for small α. At a critical value $\alpha_c(K)$ the unspecialized $(R = S)$ configuration is replaced by two minima of βf: in the global minimum we have $R > S$, while the competing local minimum corresponds to configurations with $R < S$. The former facilitates perfect generalization with $R \to 1, S \to 0$ in the limit $\alpha \to \infty$. In both competing minima the emerging specialization follows Eq. (4.24) with *critical exponent* $1/2$.

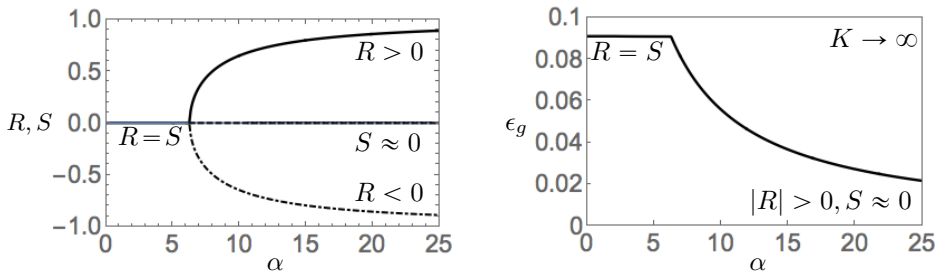


Figure 4.4: **ReLU activation:** Learning curves of the perfectly matching student teacher scenario for $K \to \infty$. In this limit, the continuous transition occurs at $\alpha_c = 2\pi$. In the **left panel**, the solid line represents the specialized solution with $R(\alpha) > 0$, while the chain line marks the solution with $R(\alpha) < 0$. In the former, $S \to 0$ for large α, while in the latter, S remains positive with $S = \mathcal{O}(1/K)$ for large K. The learning curves $\epsilon_g(\alpha)$ for the competing minima of βf coincide for $K \to \infty$ as displayed in the **right panel**. It approaches perfect generalization, i.e. $\epsilon_g \to 0$ for $\alpha \to \infty$

In contrast to the case of sigmoidal activation, both competing configurations of the ReLU system display very similar generalization behavior. While, in general, only states with $R > 0$ can perfectly reproduce the teacher output, the student configurations with $S > 0$ and $R < 0$ also achieve relatively low generalization error for large α, see Fig. 4.3 (lower row) for an example.

The limiting case of large networks with $K \to \infty$ can be considered explicitly. We find for large ReLU networks that the continuous specialization transition occurs at

$$\alpha_c(K \to \infty) = 2\pi \approx 6.28\,.$$

The generalization error decreases very rapidly (instantaneously on α-scale) from the initial value of $\epsilon_g(0) \approx 0.341$ with $R = S = C = 0$ to a constant plateau-like state with

$$\epsilon_g(\alpha) = \tfrac{1}{4} - \tfrac{1}{2\pi} \approx 0.091 \text{ for } 0 < \alpha < 2\pi$$

where $R = S = 1/K$ and $C = \mathcal{O}(1/K^2)$. For $\alpha > \alpha_c$, the order parameter R either increases or decreases with α, approaching the values $R \to \pm 1$ asymptotically, while $S(\alpha) = 0$ in both branches for $K \to \infty$.

Surprisingly, both solutions display the exact same generalization error, see Fig. 4.4 (right panel). Consequently, the free energies βf of the competing minima also coincide in the limit $K \to \infty$ since the entropy (4.19) satisfies $S(-R, 0, 0) = S(R, 0, 0)$. In the configuration with $R < 0$ the order parameters display the scaling behavior

$$S = \mathcal{O}\left(1/K\right) \text{ and } C = \mathcal{O}\left(1/K^2\right) \tag{4.25}$$

for large K. In Appendix 4.5.2 we show how a single teacher ReLU with activation $\max(0, b_i)$ can be approximated by $(K-1)$ weakly aligned units in combination with one anti-correlated student node. While the former effectively approximates a linear response of the form $const. + b_i$, the unit with $R = -1$ implements $\max(0, -b_i)$. Since $\max(0, b_i) = \max(0, -b_i) + b_i$ the student can approximate the teacher output very well, see also the appendix for details. In the limit $K \to \infty$, the correspondence becomes exact and facilitates perfect generalization for $\alpha \to \infty$.

Note that a similar argument does not hold for student teacher scenarios with sigmoidal activation functions which do not display the partial linearity of the ReLU.

4.3.3 Student-student overlaps

It is also instructive to inspect the behavior of the order parameter C which quantifies the mutual overlap of student weight vectors. In the ReLU system with large finite K, we observe $C(\alpha) = \mathcal{O}(1/K^2) > 0$ before the transition. It reaches a maximum value at the phase transition and decreases with increasing $\alpha > \alpha_c$. In the positively specialized configuration it approaches the limiting value $C(\alpha \to \infty) = 0$ from above, while it assumes negative values on the order $\mathcal{O}(1/K^2)$ in the configuration with $R < S$.

This is in contrast to networks of sigmoidal units, where $C < 0$ before the discontinuous transition and in the specialized $(R > S)$ state, see (Biehl, Schlösser and Ahr 1998, Ahr et al. 1999) for details. Interestingly, the characteristic value α_d coincides with the point where C becomes positive in the suboptimal local minimum of βf.

Figure 4.5 displays $C(\alpha)$ for sigmoidal (left panel) and ReLU activation (right panel) for $K = 5$ as an example. Apparently the ReLU system tends to favor correlated hidden units in most of the training process.

4.3.4 Monte Carlo simulations

In order to demonstrate the qualitative validity of our theoretical results also in finite systems and beyond the high-temperature limit, we performed Monte Carlo simulations of the training processes.

We have implemented the student teacher scenarios in relatively small systems with $N = 50$ and $K = 4$ hidden units. The systems were trained according to a Metropolis-like scheme with continuous changes of the student weights. In an individual Monte Carlo step (MCS), all adaptive weights in the student network were subject to independent, zero mean additive Gaussian noise with subsequent normalization to maintain $\mathbf{w}_j^2 = N$ for all j. The associated change ΔE of the training

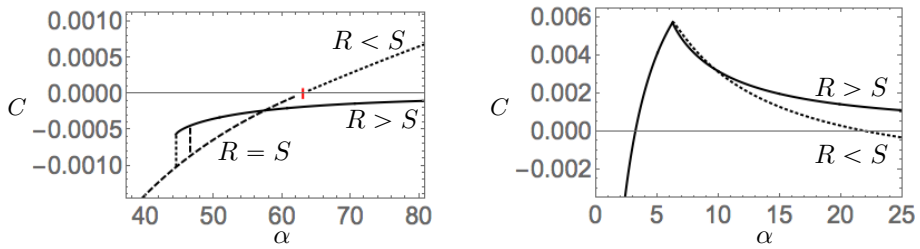


Figure 4.5: **Student cross overlap** C in a perfectly matching student teacher scenario. **Left panel:** For sigmoidal activations, here $K = 5$, the order parameter C is negative in the $(R > S)$ specialized and in the unspecialized configurations with $R = S$. The values $\alpha_s, \alpha_c, \alpha_d$ are marked as in Fig. 4.2 (lower left panel). It remains negative in the $R > S$ specialized state (solid line) for all α, while it becomes positive in α_d where the configuration with $R = S$ (dashed line) is replaced by a state with $R < S$ (dotted line). For better visibility of the behavior near α_c, only a small range of α is shown. **Right panel:** In the ReLU system with, e.g., $K = 10$, C becomes positive before the continuous transition occurs, it reaches a maximum in α_c and approaches *zero* from above for $\alpha \to \infty$ in the specialized configuration with $R > S$ (solid line). In the local minimum of βf with $R < S$, C becomes negative for large α as marked by the dotted line.

energy E, Eq. (4.9), was computed and the randomized modification was accepted with probability $\min\{1, e^{-\beta \Delta E}\}$. A constant variance of the Gaussian noise was selected as to maintain acceptance rates in the vicinity of 0.5 in each setting.

All simulations were performed with $\beta = 1$, which corresponds to a relatively low training temperature, and with training set sizes $P = \tilde{\alpha} K N$ that could be handled with moderate computational effort.

In principle, all stable and metastable states, i.e. local and global minima of the associated free energy, would be visited from random initializations by the finite system, eventually. However, this would require very large equilibration and observation times. Therefore we followed an alternative strategy by preparing initial states which slightly favored one of the competing configurations and observed the quasi-stationary behavior of the system at intermediate training times. In all training processes, quasistationary states could be observed after $\mathcal{O}(10^4)$ elementary MCS. Averages and standard deviations were determined over the last 1000 MCS in 20 independent runs for each considered setting.

Fig. 4.6 shows example learning curves in the ReLU system with $K = 4$ for $\tilde{\alpha} = 24$. The last 1000 MCS are marked by solid lines in the upper panel. The simulations

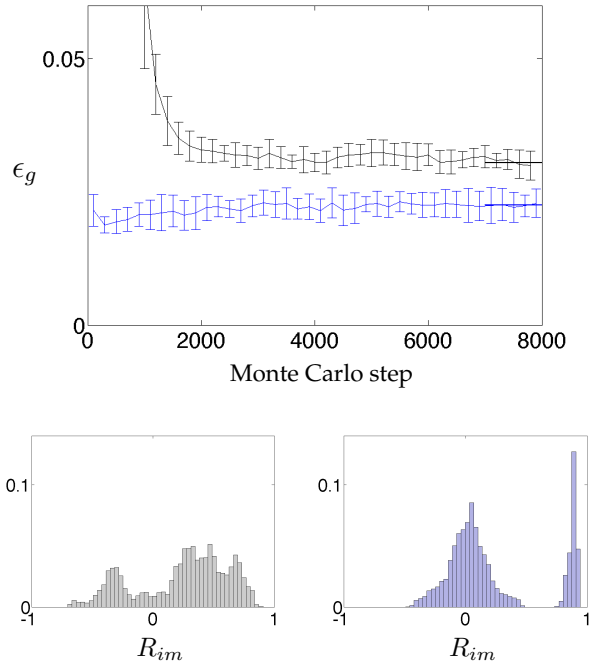


Figure 4.6: **Monte Carlo simulations** of the ReLU system. **Upper panel:** the generalization error as observed with $N = 50, \beta = 1, K = 4$ for $\tilde{\alpha} = 24$ on average over 20 independent runs, error bars represent the standard deviations in a subsets of time steps. **Lower panels:** histograms of the relative frequency of values R_{im} observed over the last 1000 elementary Monte Carlo steps as marked by the solid lines at the end of the curves in the upper panel. **Colors:** The *grey* curve and histogram correspond to initializations of the systems in slightly anti-specialized states. For the *blue* curve and histogram the systems were initialized with a weak positive specialization, see Sec. 4.3.4.

confirm the existence of two competing quasistationary states. Histograms of the observed order parameters R_{im} show that they correspond to a specialized state with few, large positive student teacher overlaps (lower right panel). The anti-specialized state is characterized by a considerable fraction of values $R_{im} < 0$, see the lower left panel of Fig. 4.6. We also obtained results for the system with sigmoidal activation, confirming the competition of a specialized state with unspecialized configurations, which are not displayed here. Similar findings, including histograms of the observed

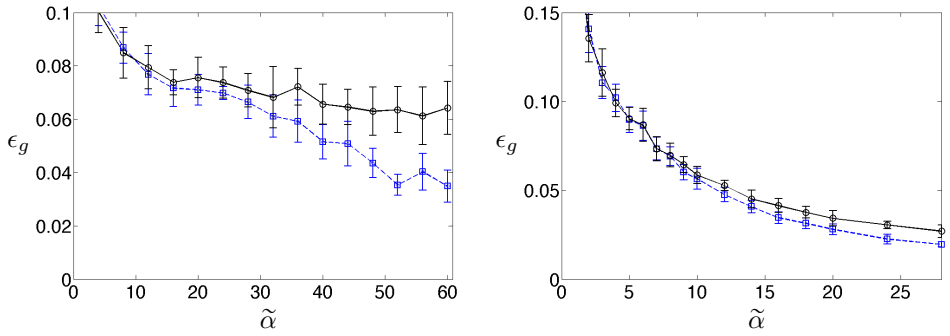


Figure 4.7: **Monte Carlo simulations** of the student teacher scenarios. The generalization error as observed for systems with $N = 50, \beta = 1, K = 4$ as a function of $\widetilde{\alpha}$; discrete points are connected for clarity. Averages and standard deviations (20 independent simulation runs) of ϵ_g were determined over the last 1000 Monte Carlo steps. **Left panel:** networks with sigmoidal activation in unspecialized (grey curve) and specialized configurations (blue curve). **Right panel:** systems with ReLU hidden layer in anti-specialized (grey curve) and specialized (blue curve) quasistationary states.

R_{im} had been published in (Biehl, Schlösser and Ahr 1998) for sigmoidal units only. There, the authors also present simulation results for $K = 2$.

We determined the average generalization error from the order parameter values as observed in the competing quasistationary states of training in the last 1000 MCS. Figure 4.7 displays the corresponding generalization error as a function of $\widetilde{\alpha}$ for sigmoidal activation in the left panel and for a ReLU hidden layer in the right panel. The observed behavior is consistent with the predicted first order and continuous phase transition, respectively. In particular we note that the competing configurations in the ReLU system (right panel) display very similar generalization errors. In contrast, the difference between specialized and unspecialized sigmoidal networks is much more pronounced, see the left panel of Fig. 4.7.

While the simulations were performed in fairly small systems with $\beta = 1$, the results are in very good qualitative agreement with the theoretical predictions obtained in the limits $N \to \infty$ and $\beta \to 0$. Note that at low temperature, training and generalization error are not identical. As expected E/P is found to be systematically lower than ϵ_g. However, we observed that generalization and training error evolve in parallel with the training time (MCS) and display analogous dependencies on the training set size $\widetilde{\alpha}$.

4.3.5 Practical relevance

It is important to realize that a quantitative comparison of the two scenarios, for instance w.r.t. the critical values α_c, is not sensible. The complexities of sigmoidal and ReLU networks with K units do not necessarily correspond to each other. Moreover, the actual α-scale is trivially related to a potential scaling of the activation functions.

However, our results provide valuable qualitative insight: The continuous nature of the transition suggests that ReLU systems should display favorable training behavior in comparison to systems of sigmoidal units. In particular, the suboptimal competing state displays very good performance, comparable to that of the properly specialized configuration. Their generalization abilities even coincide in large networks of many hidden units.

On the contrary, the achievement of good generalization in networks of sigmoidal units will be delayed significantly due to the discontinuous specialization transition which involves a poorly generalizing metastable state.

4.4 Conclusion and Outlook

We have investigated the training of shallow, layered neural networks in student teacher scenarios of matching complexity. Large, adaptive networks have been studied by employing modelling concepts and analytical tools borrowed from the statistical physics of learning. Specifically, stochastic training processes at high formal *temperature* were studied and learning curves were obtained for two popular types of hidden unit activation.

To the best of our knowledge, this work constitutes the first theoretical, model-based comparison of sigmoidal hidden unit activations and rectified linear units in feed-forward neural networks.

Our results confirm that networks with $K \geqslant 3$ sigmoidal hidden units undergo a discontinuous transition: A critical training set size is required to facilitate the differentiation, i.e. specialization of hidden units. However, a poorly performing state of the network persists as a locally stable configuration for all sizes of the training set. The presence of such an unfavorable local minimum will delay successful learning in practice, unless prior knowledge of the target rule allows for non-zero initial specialization.

On the contrary, the specialization transition is always continuous in ReLU networks. We show that above a weakly K-dependent critical value of the re-scaled training set size α, two competing specialized configurations can be assumed. Only one of them displays positive specialization $R > S$ and facilitates perfect generalization from large training sets for finite K. However, the competing configuration with

negative specialization $R < 0, S > 0$ realizes similar performance which is nearly identical for networks with many hidden units and coincides exactly in the limit $K \to \infty$.

As a consequence, the problem of *retarded learning* associated with the existence of metastable configurations is expected to be much less pronounced in ReLU networks than in their counterparts with sigmoidal activation.

Clearly, our approach is subject to several limitations which will be addressed in future studies.

Probably the most straightforward, relevant extension of our work would be the consideration of further activation functions, for instance modifications of the ReLU such as the *leaky* or *noisy* ReLU or alternatives like *swish* and *max-out* (Eger et al. 2018, Ramachandran et al. 2017).

Within the site-symmetric space of configurations, cf. Eq. (4.12), only the specialization of single units with respect to one of the teacher units can be considered. In large networks, one would expect partially specialized states, where subsets of hidden units achieve different alignment with specific teacher units. Their study requires the extension of the analysis beyond the assumption of site-symmetry.

Training at low formal temperatures can be studied along the lines of (Ahr et al. 1999) where the replica formalism was already applied to networks with sigmoidal activation. Alternatively, the simpler annealed approximation could be used (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993). Both approaches allow to vary the control parameter β of the training process and the scaled example set size $\widetilde{\alpha} = P/(KN)$ independently, as it is the case in more realistic settings. Note that the findings reported in (Ahr et al. 1999) for sigmoidal activation displayed excellent qualitative agreement with the results of the much simpler high-temperature analysis in (Biehl, Schlösser and Ahr 1998).

The dynamics of non-equilibrium on-line training by gradient descent has been studied extensively for soft-committee-machines with sigmoidal activation, e.g. (Saad and Solla 1995a, Saad and Solla 1995b, Biehl et al. 1996, Vicente and Caticha 1997) and with ReLU activation in Chapters 2 and 3. There, quasi-stationary plateau states in the learning dynamics are the counterparts of the phase transitions observed in thermal equilibrium situations. The results for ReLU networks discussed in the previous chapters should be extended in order to identify and understand the influence of the activation function on the training dynamics in greater detail.

Model scenarios with mismatched student and teacher complexity will provide further insight into the role of the activation function for the learnability of a given task. It should be interesting to investigate specialization transitions in practically relevant settings of off-line learning in which either the task is unlearnable ($K < M$) or the student architecture is over-sophisticated for the problem at hand ($K > M$). In

addition, student and teacher systems with mismatched activation functions should constitute interesting model systems.

The complexity of the considered networks can be increased in various directions. If the simple shallow architecture of Eq. (4.1) is extended by local thresholds and hidden to output weights that are both adaptive, it parameterizes a *universal approximator*, see e.g. (Cybenko 1989, Hornik 1991, Hanin 2017). Decoupling the selection of these few additional parameters from the training of the input to hidden weights should be possible following the ideas presented in (Endres and Riegler 1999).

Ultimately, *deep* layered architectures should be investigated along the same lines. As a starting point, simplifying tree-like architectures could be considered as in e.g. (Herschkowitz and Opper 2001, Baldassi et al. 2019).

Our modelling approach and theoretical analysis goes beyond the empirical investigation of data set specific performance. The suggested extensions bear the promise to contribute to a better, fundamental understanding of layered neural networks and their training behavior.

4.5 Chapter Appendix

4.5.1 Single unit student and teacher

In the simple case $K = 1$ with a single unit as student and teacher network, we have to consider only one order parameter $R = \mathbf{w}\cdot\mathbf{B}/N$. Assuming $\mathbf{w}\cdot\mathbf{w}/N = \mathbf{B}\cdot\mathbf{B}/N = 1$, we obtain the free energy $(\beta f) = \alpha\,\epsilon_g - s$ with

$$s = \frac{1}{2}\ln[1-R^2] + const. \tag{4.26}$$

$$\epsilon_g = \frac{1}{3} - \frac{2}{\pi}\sin^{-1}\big[R/2\big] \qquad \text{(sigmoidal)} \tag{4.27}$$

$$\epsilon_g = \frac{2-R}{4} - \frac{\sqrt{1-R^2}+R\sin^{-1}[R]}{2\pi} \quad \text{(ReLU)}. \tag{4.28}$$

The necessary condition $\partial(\beta f)/\partial R{=}0$ becomes

$$\alpha = \frac{\pi R\sqrt{4-R^2}}{2(1-R^2)} \qquad \text{(sigmoidal)} \tag{4.29}$$

$$\alpha = \frac{4\pi R}{(1-R^2)(\pi + 2\sin^{-1}[R])} \qquad \text{(ReLU)}\,. \tag{4.30}$$

In both cases, the student teacher overlap increases smoothly from *zero* to $R = 1$. A Taylor expansion of $1/\alpha$ for $R \approx 1$ yields the asymptotic behavior

$$R(\alpha) = 1 - \frac{const.}{\alpha} \;\text{ and }\; \epsilon_g(\alpha) = \frac{1}{2\alpha} \;\text{ for }\; \alpha \to \infty$$

for both types of activation. This basic large-α behavior with $\epsilon_g \propto \alpha^{-1}$ carries over to student teacher scenarios with general $K = M$ in configurations with positive specialization.

4.5.2 Weak and negative alignment

Here we consider a particular teacher unit which realizes a ReLU response

$$\max(0, b) \text{ with } b = \frac{\mathbf{B}\cdot\boldsymbol{\xi}}{\sqrt{N}}\,.$$

A set of K hidden units in the student network can obviously reproduce the response by aligning one of the units perfectly with, e.g., $R{=}\mathbf{w}_1\cdot\mathbf{B}/N{=}1$ and $S{=}\mathbf{w}_j\cdot\mathbf{B}/N{=}0$ for $j > 1$. Similarly, we obtain for $R = -1$ that $h_1 = -\mathbf{w}^*\cdot\boldsymbol{\xi}$ and $\max(0, h_1) = \max(0, -b)$.

Now consider the mean response of a student unit with small positive overlap $S = \mathbf{w}_j \cdot \mathbf{B}/N$, given the teacher unit response b. It corresponds to the average $\langle \max(0, h_j) \rangle_b$ over the conditional density $P(h_j|b) = P(h_j, b)/P(b)$. One obtains

$$\langle \max(0, h_j) \rangle_b = \frac{1}{\sqrt{2\pi}} + \frac{b}{2} S + \mathcal{O}(S^2)$$

by means of a Taylor expansion for $S \approx 0$. As a special case, the mean response of an orthogonal unit with $S = 0$ is $1/\sqrt{2\pi}$, independent of b.

It is straightforward to work out the conditional average of the total student response for a particular order parameter configuration with $R = -1$ and $S = 2/(K-1)$. Apart from the prefactor $1/\sqrt{K}$ it is given by

$$\max(-b, 0) + b + \tfrac{K-1}{\sqrt{2\pi}} = \max(0, b) + \tfrac{K-1}{\sqrt{2\pi}} \,,$$

where the right hand side coincides with the expected output for $R = 1$ and $S = 0$. Hence, the average response agrees with the teacher output for large K. Moreover, the correspondence becomes exact in the limit $K \to \infty$, which facilitates perfect generalization in the negatively specialized state with $S > 0, R < 0$ discussed in Sec. 4.3.

Part II

Machine learning applications in smart industry and functional data classification

Published as:

M. Straat, K. Koster, N. Goet, K. Bunte – *'An Industry 4.0 example: real-time quality control for steel-based mass production using Machine Learning on non-invasive sensor data"*, International Joint Conference on Neural Networks (IJCNN), 2022.

Chapter 5

An Industry 4.0 case study: real-time quality control for steel-based mass production using Machine Learning on non-invasive sensor data

Abstract

Insufficient steel quality in mass production can cause extremely costly damage to the production tooling, production downtimes and low quality products. Automatic, fast and cheap strategies to estimate essential material properties for quality control, risk mitigation and the prediction of faults are highly desirable. In this work we analyse a high throughput production line of steel-based products. Currently, the material quality in the line is checked using manual destructive testing, which is slow, wasteful and covers only a tiny fraction of the material. To achieve complete testing coverage our industrial collaborator developed a contactless, non-invasive, electromagnetic sensor to measure all material during production in real-time. Our contribution is three-fold: 1) We show in a controlled experiment that the sensor can distinguish steel with deliberately altered properties. 2) During several months of production 48 steel coils were fully measured non-invasively and additional destructive tests were conducted on samples taken from them to serve as ground truth. A linear model is fitted to predict from the non-invasive measurements two key material properties (yield strength and tensile strength) that normally have to be obtained by destructive tests. The performance is evaluated in leave-one-coil-out cross-validation. 3) The resulting model is used to analyse real production data of approximately 108 km of processed material measured with the non-invasive sensor and the relationship with recorded product faults. The model achieves an excellent performance (F3-score of 0.95) predicting material running out of specifications for the tensile strength. In a second controlled experiment one coil suspected of material faults was sampled 18 times over its full length and repeated non-invasive as well as destructive testing was performed to analyse the relationship between both measurement types in a situation where also product faults and problems during production are expected to occur. On this coil the model predictions demonstrate that material properties are indeed out of specification near the point for which the products made from the neighboring coil exhibited faults during production. The combination of model predictions and logged

product faults shows that if a significant percentage ($> 30\%$) of estimated yield stress measurements is out of specification, the risk of product faults is high. Our analysis demonstrates promising directions for real-time quality control, risk monitoring and fault detection.

5.1 Introduction

The terms "Smart Industry", "Industry 4.0", or "Fourth Industrial Revolution" (Schwab 2017) have been coined to describe a vision that includes a wide range of emerging technologies that, when used collaboratively, have the potential to contribute to highly optimized production processes (Lasi et al. 2014, Vaidya et al. 2018). Several of these fields that are of central importance in the development of the so-called *smart factory* are sensor technology, Cyber Physical Systems, the Internet of Things, advanced communication technology, big data analytics, Machine Learning (ML), Artificial Intelligence (AI) and cloud computing (Chen et al. 2017, Castelo-Branco et al. 2019). Various examples exist in which the successful implementation of a combination of these technologies results in higher production efficiency, better human decision making and less waste (Vaidya et al. 2018). Realizing the large potential of Smart Industry has been recognized as a key factor by governments and industries for ensuring economic competitiveness and sustainability in the next decades.

In this chapter we develop a typical Industry 4.0 solution: A real-time quality control and fault detection system for a high-throughput production line of products made from strip steel. The main machinery in the production line under study is a high-speed stamping press operating on the strip steel at frequencies of up to 180 strokes per minute. In general, if the material quality of the steel is insufficient, it may have a variety of serious consequences, such as poor quality of the final products, expensive damage to the production machinery and resulting production downtime.

More specifically, the manufacturer frequently encounters a specific fault in the production process that causes production downtimes and potentially expensive damage to the machinery if the production is not stopped immediately. During such an event, a crack arises in the product in the stamping process. It is hypothesized that insufficient material quality may be one of the causes of the fault. Hence, it is crucial that all strip steel that enters the production process is of sufficient quality in order to prevent this type of fault.

The current material quality requirements specify the upper limit of stress [MPa] for the properties yield strength and tensile strength, which is known as the Upper Specification Limit (USL). Currently, destructive tests on samples of the steel are performed to check whether the material conforms to the specifications. By means

of a tensile test on the sample, the yield stress and tensile strength of the sample are determined, together with several other material properties not considered in this work. Although the material properties of the sampled steel can be measured reliably using these methods, the process is manual, therefore slow and it produces material waste. For these reasons, it is only feasible to test a tiny fraction of the material in the production line. If the test results indicate insufficient material quality, a potentially large batch of products that were already produced may not be of sufficient quality and must be thrown away, which increases the waste. On the other hand, if the test shows sufficient material quality, it can still be the case that undetected material changes have occurred, resulting in lower quality products. Hence, tensile testing cannot be a solution for continuous quality control and material quality guarantees. A continuous quality control solution should be able to achieve full test coverage of the production material, detect highly local material faults and sudden changes in material properties.

In implementations of so-called soft sensors, easily obtainable process variables are measured inline in real-time. These measurements are simultaneously converted using a statistical or machine learning model to quantities that otherwise have to be measured in expensive, time-consuming lab tests (Jiang et al. 2021). An important component of soft sensing in smart industry is Nondestructive Testing (NDT) (Sophian 2020). In the steel-based manufacturing industry, NDT sensors perform contactless and non-destructive measurements on the steel in real-time and can therefore be used in a high-throughput production line to measure all strip steel that enters the process (García-Martín et al. 2011). By combining the real-time stream of measurements with appropriate machine learning models, advanced online fault detection and quality control systems can be developed.

Previously, Long Short-Term Memory and Gaussian Processes were used in industrial settings where temporal patterns are relevant (Malhotra et al. 2015, Berns et al. 2020). Although these techniques are able to represent complex temporal relationships, they are computationally expensive and have large data requirements. Latent variable models such as supervised factor analysis and Partial Least Squares have also been used in industrial settings (Ge 2016, Rosipal and Krämer 2005), which generally require less computational effort. A successful implementation of a real-time quality control system leads to fewer defects in products, improved quality, less production downtime and less material waste. Furthermore, the real-time model estimation of material quality from the inline measurements can be used in the active control of production parameters, which aims to adjust the machinery settings in real-time towards optimal parameter values with respect to the specifics of the measured material (Heingärtner et al. 2010, Ge 2016, Jiang et al. 2021).

Our industrial collaborator developed a soft sensor based on Eddy Currents

(García-Martín et al. 2011). The sensor is located at the start of the production line and it measures the strip steel exactly at the locations where the press operates further in the line. Our main contribution is the development of a real-time quality control and fault detection solution for the steel in the production line based on the sensor measurements. Our contributions that are necessary to realize the system are three-fold: 1) A model is developed for estimating material properties in real-time from the inline contactless sensor measurements. We use the ground truth material properties of several production coils to fit the model. 2) The model is used for the early detection of insufficient material quality. Specifically, the model is used for predicting the material properties of a steel coil which comes from a batch of coils that had caused product faults earlier. For this case the model estimation of the material properties is able to detect material of insufficient quality and thereby prevent product faults. 3) We study the model estimations on 108 km of processed strip steel in the line and we link the model estimations to reported product faults during production. It is demonstrated that the model estimations can be used to assess the risk of the occurrence of product faults.

The paper is organized as follows: In Sec. 5.2 the relevant details of the new industrial datasets are introduced. Subsequently, in Sec. 5.3 the methods used for the analysis of the data and for the estimation of material properties are discussed. In Sec. 5.4 we present the results of our experiments and discuss them in Sec. 5.5. Lastly, the work is summarized in Sec. 5.6 and an outlook for future work is provided.

5.2 Data description and analysis

The production coils of small strip steel have an associated `Heat` number, which identifies the specific elements used in the steel production batch. The NDT sensor measurements based on Eddy current testing (García-Martín et al. 2011) are performed at 10 test frequencies. We denote the measurements with $\boldsymbol{x}_i \in \mathbb{R}^{20}$. The first half of the components of $\boldsymbol{x}$ are the amplitude gains and the second half the phase shifts of each frequency. Hence, for measurement number i, the amplitude gain and phase shift of test frequency j are stored in $x[i, j]$ and $x[i, j + 10]$, respectively. In some parts of the text and figures, we will use the shorthand name SV i to denote the sensor variables.

5.2.1 Controlled experiment: measuring modified steel samples

In order to study the sensitivity of the sensor and to establish an expected lower and upper bound for each sensor variable, steel modified to extreme material properties was measured with the contactless sensor and compared to the reference steel. A

selection of nine steel strips was divided into three groups of three steel strips. One group was left unmodified and serves as reference material. For the other two groups, one group was modified to be "harder" and the other group to be "softer": Although the properties of the resulting steel were not measured, after the modification the harder steel should have yield- and tensile strength properties much larger than the reference material and these properties for the resulting softer steel should be much smaller than the reference. Subsequently, each metal strip was measured with the sensor at the start, middle and end of the strip. At each location 100 sensor measurements were made.

We transformed the original sensor measurements $\boldsymbol{x}_i \in \mathbb{R}^{20}$ as follows:

$$\boldsymbol{x}_i := \frac{\boldsymbol{x}_i - \boldsymbol{P}^H_{10\%}}{\boldsymbol{P}^S_{90\%} - \boldsymbol{P}^H_{10\%}}\,, \tag{5.1}$$

where $\boldsymbol{P}^H_{10\%} \in \mathbb{R}^{20}$ are the 10th percentiles of the variables only considering the measurements of the **H**ard group and $\boldsymbol{P}^S_{90\%} \in \mathbb{R}^{20}$ are the 90th percentiles of the variables only taking into account the measurements of the **S**oft group. Note that since $\boldsymbol{P}^H_{10\%}[j] < \boldsymbol{P}^S_{90\%}[j]$ for $j = [1, ..., 20]$, the applied transformation in Eq. (5.1) is effectively min-max normalization, with the percentiles being estimates of the min and max. After the transformation of Eq. (5.1), the resulting values were shifted according to

$$\boldsymbol{x}_i := \boldsymbol{x}_i - \boldsymbol{\mu}^R\,, \tag{5.2}$$

where $\boldsymbol{\mu}^R \in \mathbb{R}^{20}$ are the means of the variables computed over the **R**eference group after the transformation of Eq. (5.1). The shift effectively moves the values for the reference material close to zero.

We denote a measurement at a location on a strip in one of the categories using its category letter [S, R, H], a digit in [1, 2, 3] to denote the strip and another digit in [1, 2, 3] to denote the measurement location on the strip. For instance, "H31" denotes the sensor measurements of strip number 3 in category "H" at location "1".

Additionally, two weeks after the measurements, the same locations were measured a second time. We denote the time of the first measurements by t_1, the time of the second measurements by t_2 and $t_2 - t_1 \approx 14$ days.

In Fig. 5.1 the left- and right figure shows for the two measurement times, the obtained distributions of measurements taken at the locations S12 and H13, respectively, for SV 17. Clearly, the sample means at the two measurement times for location S12 were different. In contrast, the sample means for location H13 were approximately equal.

We performed two-tailed dependent samples t-tests for all measurement locations comparing for each individual sensor variable the samples measured at time t_1 to the

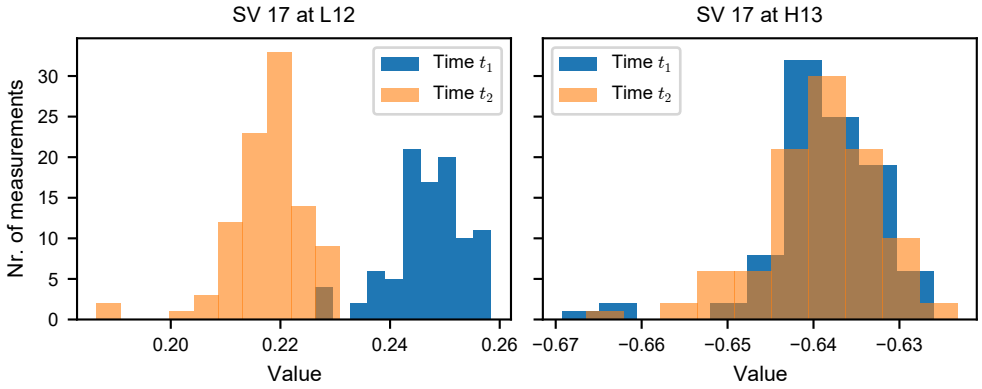


Figure 5.1: Distribution of the approximately 100 sensor measurements at the locations S12 (left) and H13 (right). t_1 is the time of the first set of measurements and t_2 the time of the second set of measurements, with $t_2 - t_1 \approx 14$ days.

samples measured at time t_2. The null-hypothesis H_0 was that the two distributions have equal population means. For each comparison the p-value that is computed from the resulting t-statistic is shown in Fig. 5.2. For a large part, the p-values were below a value of 0.05. Hence, using a significance value of $p = 0.05$, for these cases the null hypothesis of equal population means could be rejected. The null hypothesis could frequently not be rejected for the steel strips in the **H**ard category, especially for the 10 sensor variables corresponding to the phases. Not surprisingly, for the S12 samples shown in Fig. 5.1, the p-value was small whereas for the H13 samples the p-value was large.

Fig. 5.3a shows a visualization of the Pearson correlation coefficients between the sensor variables. In general the sensor variables are strongly correlated. The variables SV 4 and SV 11 each have small correlation values with all other variables. In the figure SV 17 is plotted against the SV 11 where the lack of a clear correlation can be seen. Although the variable `svar1` has low discriminatory power in separating the classes in this experiment, it can still be an important variable that holds information about the quality of the steel. For instance, in this case it separates in the "H" class two groups of measurements from the other six groups. This could indicate that the corresponding two locations have differing material properties than the other groups.

From the plot of SV 17 against SV 18, the almost perfect correlation can be observed. In this case, the **S**oft, **R**eference and **H**ard classes of steel can clearly be distinguished by the value of both variables. A group of measurements of the hard material had similar values to the measurements from the reference and soft material.

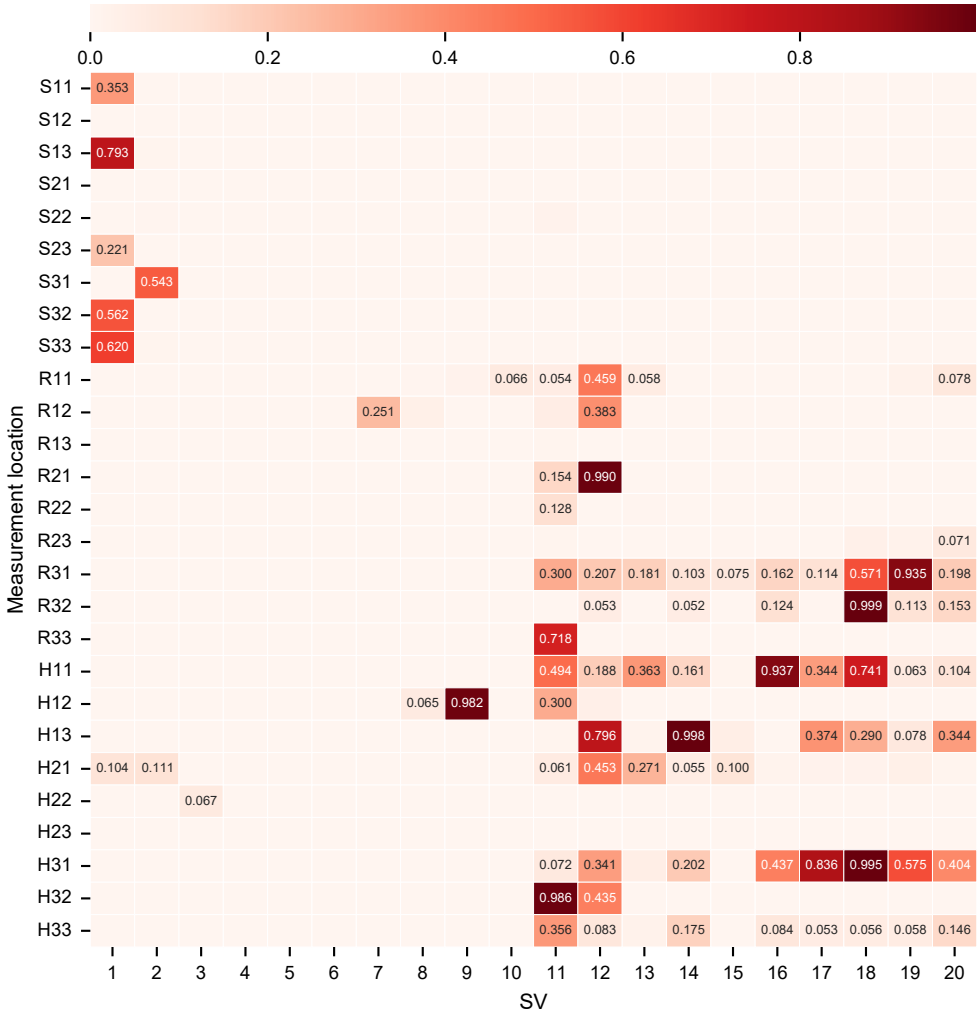
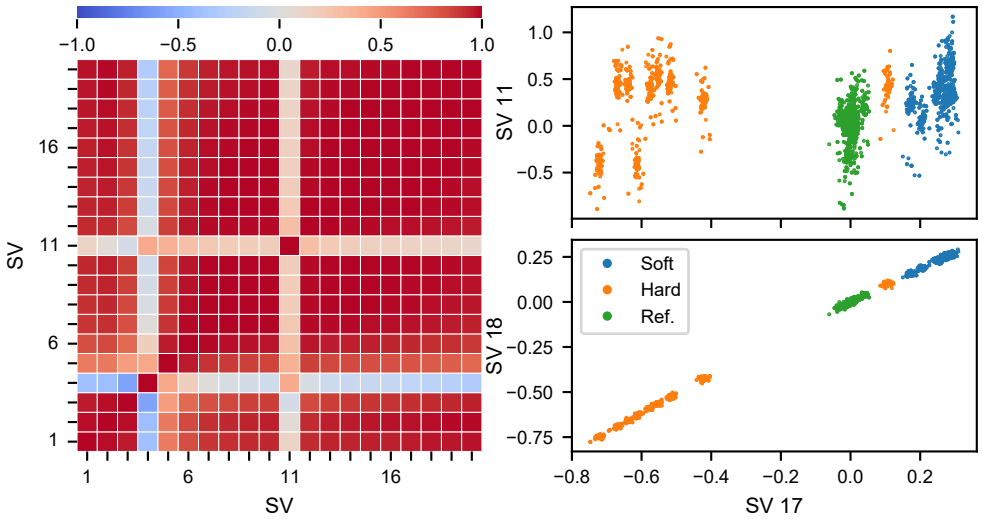


Figure 5.2: Heatmap of p-values obtained from paired t-tests of the sensor measurements taken on the same steel samples at times t_1 and t_2. Sensor variables are on the x-axis and measurement locations on the y-axis. The S, R or H in the location identifier denotes steel with the **S**oft, **R**eference or **H**ard material property, respectively. The next digit denotes the number of the strip in the category and the last digit the measurement location on the strip. The p-value is only shown if it is greater than the significance value of 0.05.



(a) (b)

Figure 5.3: *Left*: Pearson correlation coefficient between pairs of sensor variables computed over all measurements in the modified steel experiment. *Right*: SV 17 plotted against SV 1 (top) and SV 18 (bottom), colored by the category of the steel property. Note that we only plotted the data measured at time t_1 here.

For further investigation, the corresponding strip which is from the location H23 was subjected to a tensile test. The tensile test showed that the material had material properties that were similar to the reference material. Hence, it is likely that the steel modification failed at this location and the obtained sensor measurements should be regarded as the reference group.

Fig. 5.4 shows the sensor variable loadings on the first two principal components computed on the data which was z-score standardized for the purpose of performing the PCA. Due to the large mutual positive correlations between the variables, the majority of the variables has a large loading on the first principal component (PC 1). PC 1 explains already 86% of the variance in the data. Together with the second principal component (PC 2), which has large loadings for SV 11 and SV 4, 96% of the total variance is explained. Fig. 5.5 shows the projection of all data points on the first two principal components. In general, it can be seen that the groups of measurements have characteristic scores on the first principal component. That is, the different material properties can be distinguished clearly by the scores on the first principal component. The scores on the second principal component separate

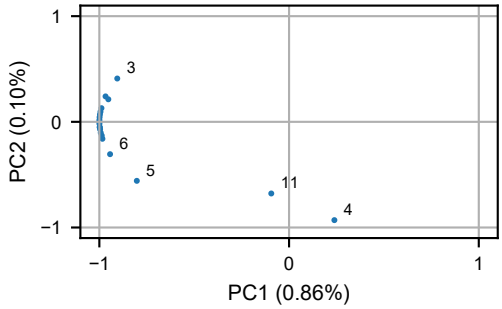


Figure 5.4: Sensor variable loadings on the first two Principal Component Analysis (PCA) components computed on the standardized hard, soft and reference measurements. Only outlier variables are labeled.

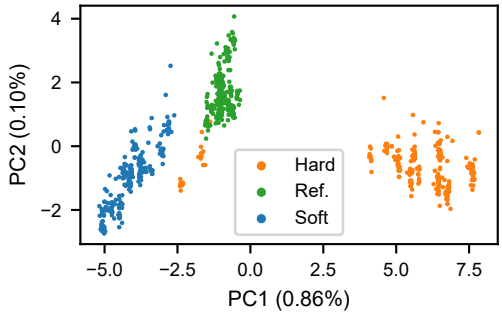


Figure 5.5: Sensor measurements on steel with different material properties projected on the the first two PCA components of the standardized data. 15% of the total number of measurements is shown, uniform randomly chosen.

the soft and hard groups from the reference group.

5.2.2 Production setting: continuous measurements in the line

In this section we discuss the dataset that was obtained to investigate the relationship of the non-invasive 20 dimensional sensor measurements to material properties confirmed by destructive testing.

Sensor data during production

The sensor was installed at the start of the production line to continuously measure the steel coils that were used for production. This produced a stream of sensor

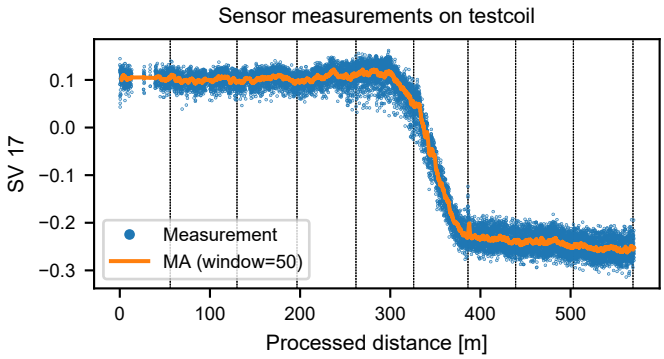


Figure 5.6: *Blue points*: Sensor variable 17 measurements taken on the testcoil. *Solid orange line*: Moving average over 50 measurements. *Dashed black lines*: Locations of the destructive test samples.

measurements $\boldsymbol{x}_i \in \mathbb{R}^{20}$ together with timestamp metadata and the current steel coil identification. From each coil a variable number of products is made; the range of measurements varies from a few hundred to tens of thousands of final products. In some instances a production stop caused the sensor to produce physically impossible values (e.g. negative values) or no values at all. These faulty records were removed from the dataset.

Destructive Tensile tests

From 47 selected coils a sample at the start of the coil was taken to measure material properties using a tensile test. Three tensile tests were performed on the sample to measure yield strength and tensile strength. In the following, the tensile strength and yield strength are denoted by "t1" and "t2", respectively. During the period of the experiment, one production coil resulted in many instances of products with cracks, hypothesized to be caused by insufficient steel quality. Hence, it was decided that a related coil from the same heat should be rejected for production and instead be fully measured by the non-invasive sensor as well as frequently sampled for tensile testing. At each of nine locations distributed over the full length of the coil two samples were taken for tensile testing. We label this particular coil as "Testcoil" to distinguish it from the rest of the 47 production coils.

Fig. 5.6 shows the value of sensor variable 17 as measured over the full length of the coil, along with the nine locations at which two tensile test samples were taken. The signal is characterized by a band of values around the moving average and exhibits a large transition in the middle of the coil. Because of high redundancy of

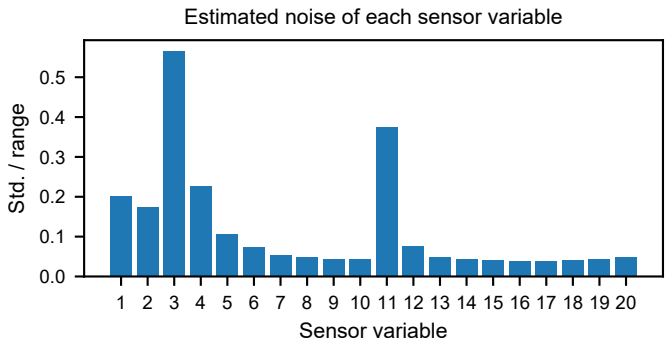


Figure 5.7: The fraction of the standard deviation with respect to the transition difference, as an estimation of the measurement noise.

the variables we made a ranking of the quality of the variables by estimating the measurement noise. The standard deviation of the signal between sample number 2000 and 4000 was computed and divided by the total transition difference of the signal, i.e. for each sensor variable the difference between the first and last value of the moving average from Fig. 5.6. In other words, the standard deviation of the band around the moving average of the signal is expressed as a fraction of the total transition difference. For each sensor variable, the value of this fraction is shown in Fig. 5.7. Sensor variable 17 has one of the lowest estimated noise values. It is also clear that SV 3 and 11 have high estimated noise values.

Logs of production faults

Instances of cracks appearing in products were logged during the months of the experiment. Since the fault has to be detected and handled by an operator, logging the faults is a manual process which was done at two levels of detail.

At the highest level of detail, the faulty product code was logged together with the identification code of the sensor measurement made on the corresponding steel. Hence, the sensor measurement of the steel is available that was used for a product which had a fault later on in the line. Due to the fast production process and the current system facilities, it is highly complex to couple the product code to the correct corresponding sensor measurement identification. Therefore, only 17 of these instances are available in the logs.

At the lowest detail level, fault instances were written down in a logbook. These entries specify the hour during which the product fault occurred and they do not contain the product code itself. In the dataset there are 25 of these entries that span

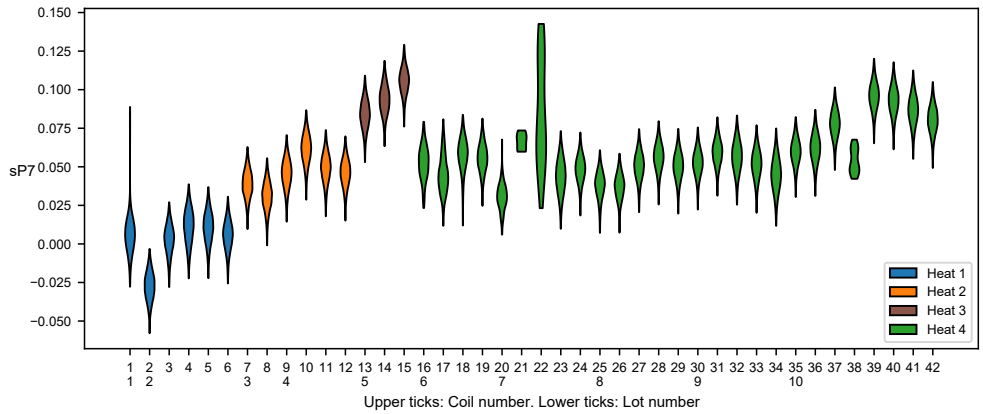


Figure 5.8: The distribution of sensor variable SV 17 (sP7) for each production coil in the dataset. Measurements outside of the 0.5 and 99.5 percentiles are not shown.

six production coils.

We normalized the sensor measurements using Eq. (5.1), so that values close to zero indicate measurements similar to the reference material of the experiment in Sec. 5.2.1. Furthermore, negative values are closer to the measurements of the hard material while positive values are closer to the measurements of the soft material.

We standardized both material properties t1 and t2 that were obtained by the tensile tests on the 48 coils. To relate the tensile tests to the sensor measurements, the mean and standard deviation were computed from the first 200 NDT sensor measurements performed on the 47 production coils. Coils with less than 200 measurements were dropped from the data. This left 42 production coils in the dataset. Fig. 5.8 shows the distribution of the NDT sensor variable SV 17 for each of these 42 coils used in the production process. For the 18 tensile test samples taken over the full length of the testcoil, we computed the mean and standard deviation of the five NDT sensor measurements that were closest to the center of the sample. Fig. 5.9 exemplifies the resulting values of the tensile tested material properties against non-invasive sensor variable 17 for the 43 coils (42 production coils + testcoil). The current USL for the respective material properties is marked in both figures. As can be seen, several points measured using tensile tests on the testcoil had material properties far exceeding both USL. The corresponding values for SV 17 were also very different from the rest. The material properties of some production coils were slightly exceeding the USL as well. We observe a negative, approximately linear correlation between material properties and sensor measurements. In general, coils from the same heat exhibit similar material properties and sensor measurements.

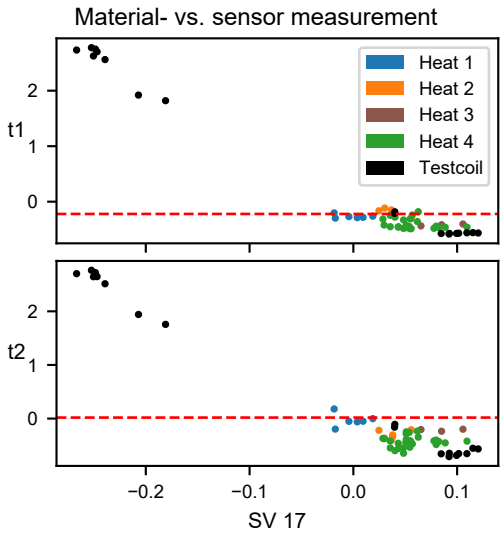


Figure 5.9: Material properties t1 and t2 against sensor variable SV 17 for the 42 production coils and the testcoil. Values for t1 and t2 represent the mean of three tensile tests. Values of SV 17 represent the mean of the first 200 NDT sensor measurements for the production coils and for the testcoil the mean around the 18 samples. Standard deviations are on average about two times the size of the markers. The dashed red line denotes the USL of the respective material properties.

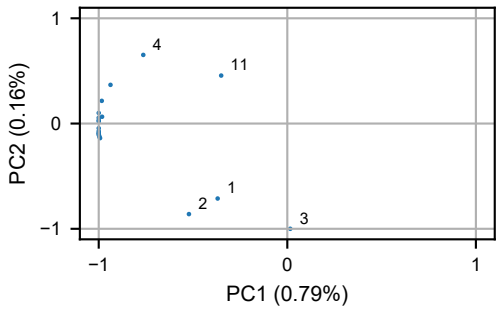


Figure 5.10: Sensor variable loadings on the first two PCA components computed on the standardized production coil and test coil dataset. Only outlier variables are labeled.

For this dataset, Fig. 5.10 shows the sensor variable loadings on the first two principal components obtained by a PCA on the full 20-dimensional sensor measure-

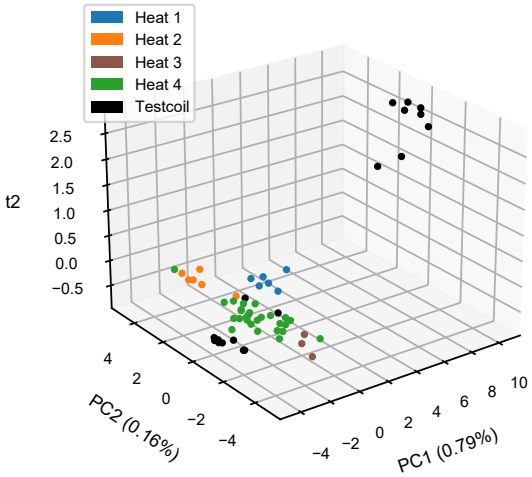


Figure 5.11: The production coil and testcoil dataset: Material property t2 against the scores on the first two principal components computed by a PCA on the standardized NDT sensor measurements.

ments. In general, the variable loadings and the variations explained by the principal components were similar to those of the controlled experiment of Fig. 5.4. Fig. 5.11 shows the measured material property t2 against the projections of the sensor datapoints on the first two principal components. Note that the points with outlier t2 measurements are only separated from the rest of the sensor measurements by the scores on the first principal component. The highly different material properties of the datapoints are not expressed in the scores on the second principal component. However, the different heats have characteristic scores on this component. Table 5.1 contains the Pearson correlation for this dataset computed with and without considering the testcoil points. Excluding the test points, the correlation with the principal components is much smaller, but still significant.

5.3 Methods

The change in material properties is not considered to result from periodic time variations, but rather local fluctuations in the production of the steel. The analysis in previous sections demonstrates linear correlations and relationships in our datasets and hence a linear model is considered for estimating the material properties and fault detection.

Table 5.1: Correlation matrix: Principal Components and material properties without (left) and including testcoil points (right)

	t1	**t2**
PC1	0.31	0.42
PC2	0.45	0.04
t1	1.00	0.38

	t1	**t2**
PC1	0.97	0.97
PC2	0.03	-0.01
t1	1.00	0.99

The model that we employ is a Partial Least Squares (PLS) regression model. For PLS regression it is assumed that the data is generated by a smaller number of latent variables than the number of observed variables. Let n be the number of data points, m the number of observed variables and o the number of target variables. Then for the predictor matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$, target matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times o}$ and selecting k number of latent variables, the PLS assumption can be written as follows:

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E} \ , \\ \boldsymbol{Y} &= \boldsymbol{U}\boldsymbol{Q}^T + \boldsymbol{F} \ , \end{aligned} \tag{5.3}$$

where $\boldsymbol{T} \in \mathbb{R}^{n \times k}$ and $\boldsymbol{U} \in \mathbb{R}^{n \times k}$ are the score matrices containing the scores on the k latent variables for each datapoint's input and target, respectively. The matrix $\boldsymbol{P} \in \mathbb{R}^{m \times k}$ contains the original input variable loadings on the k latent input variables and the matrix $\boldsymbol{Q} \in \mathbb{R}^{o \times k}$ contains the original target variable loadings on the k latent target variables. Lastly, $\boldsymbol{E} \in \mathbb{R}^{n \times m}$ and $\boldsymbol{F} \in \mathbb{R}^{n \times o}$ are the residuals.

The optimization procedure finds the k latent variables in $\boldsymbol{X}$ and $\boldsymbol{Y}$ that have maximal covariance. For an overview of the different variants of PLS and optimization procedures, see (Rosipal and Krämer 2005).

Here, we used the `PLSRegression` implementation of (Pedregosa et al. 2011) using the NIPALS algorithm with default optimization parameters. The sensor measurements $\boldsymbol{x}_i \in \mathbb{R}^{20}$ were used as inputs and the material properties $\boldsymbol{y}_i \in \mathbb{R}^2$ as targets. Note that the number of latent variables is restricted to $\min(n, m)$ in this implementation.

We used cross-validation to determine the optimal number of latent variables k with respect to average validation set Root-Mean-Square Error (RMSE) between model predictions $\hat{\boldsymbol{y}}_i \in \mathbb{R}^2$ and targets $\boldsymbol{y}_i \in \mathbb{R}^2$. In each fold of the cross-validation, the model was fitted on the measurements of all coils except one. The left out coil was used for validating the model. Hence, in each fold there was one measurement in the validation set (Leave One Out cross-validation), except for the fold in which the testcoil served as validation, which had 18 measurements.

Furthermore, we evaluated the accuracy of a binary classifier that was based on the estimated material properties. Depending on the importance of the material properties to the prevention of faults, it is possible to reject material based on the material properties individually or on a combination of the material properties. We distinguished three types of rules for the classification of measurement i: According to the first rule, the material was rejected if the estimation of t1 was above its USL, i.e. $\hat{y}_{i1} > USL(t1)$. Similarly, in the second rule the material was rejected if the estimation of $t2$ was above the USL,i.e. $\hat{y}_{i2} > USL(t2)$. According to the last rule, the material was rejected if either of t1 or t2 was above their respective USL, i.e. $(\hat{y}_{i1} > USL(t1)) \,|\, (\hat{y}_{i2} > USL(t2))$. The target labels were obtained by applying the same rules on the ground truth material properties $\boldsymbol{y}$ that were measured with the tensile tests. Based on the number of true positives, false positives, false negatives and true negatives we computed the precision and recall. We also computed F_1 and F_3 scores. The F_3 score assigns three times more importance to recall over precision, which is more appropriate for our case: A missed material fault means that material that is out of specification goes into the production process, which may then cause extremely costly damage to the machinery or final products of lower quality. In contrast, a false alarm results in minor cost in the form of wasted material that would have been suitable for production and potentially minor production delays when the material is removed from the coil. It should be understood that the precise assessment and adjustment of the classifier should be based on a thorough cost analysis of the classifiers' decisions in practice. Our choice for the F_3 score is a first estimation of the importance of recall with respect to precision.

5.4 Results

5.4.1 Dataset/Production coils

Fig. 5.12 shows the average RMSE for both material properties obtained by the PLS model in the one-coil-out cross-validation for increasing number of latent variables k. Upon further inspection we note that excluding the testcoil points from the training set results in by far the highest RMSE, which represents an outlier that is not shown in Fig. 5.12. Thus, one needs to ensure that the full range of variation that is potentially seen in production is included in model fitting, which might require deliberate creation of undesirable material. Furthermore, it can be seen that the RMSE does not decrease significantly by introducing more than one component. Hence, PLS determined one component of the sensor measurements $\boldsymbol{X}$ and one component of the material properties $\boldsymbol{Y}$ which, due to significant covariance, could be exploited in the regression. Extracting more than one latent variable did not increase the performance

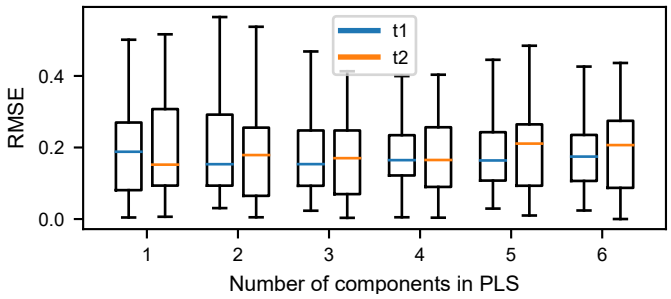


Figure 5.12: RMSE computed as the mean of the RMSE obtained on the validation sets in leave-one-coil-out cross-validation vs. the number of components/latent variables in PLS.

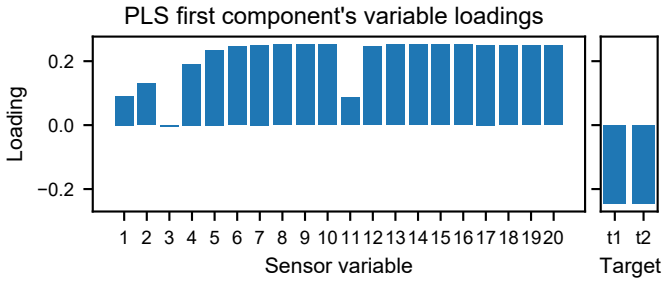


Figure 5.13: Loadings on the component extracted by PLS of the sensor variables in $\boldsymbol{X}$ (*Left*) and the destructively tested material properties in $\boldsymbol{Y}$ (*Right*).

considerably.

Fig. 5.13 shows the loadings of the variables on the first PLS component, for both the sensor variables $\boldsymbol{X}$ and the material properties $\boldsymbol{Y}$. These loadings were obtained by a PLS fit on the entire dataset (42+18 datapoints). As can be seen, the sensor variables 5 to 10 and 12 to 20 had nearly identical loadings on the first component. Upon further inspection we found that these loadings were nearly identical to the first principal component from the PCA of Sec. 5.2. The component extracted from Y had equal loadings for both material properties. Since the non-invasive sensor measurements are strongly correlated and one PLS component is sufficient for the regression task, the question arises if similar performance can be achieved by regression on individual sensor variables.

Fig. 5.14 shows the cross-validation RMSE for linear regressions on the individual sensor variables as predictor along with the RMSE obtained by PLS. Linear regressions using one of the higher loaded variables from Fig. 5.13 had similar performance

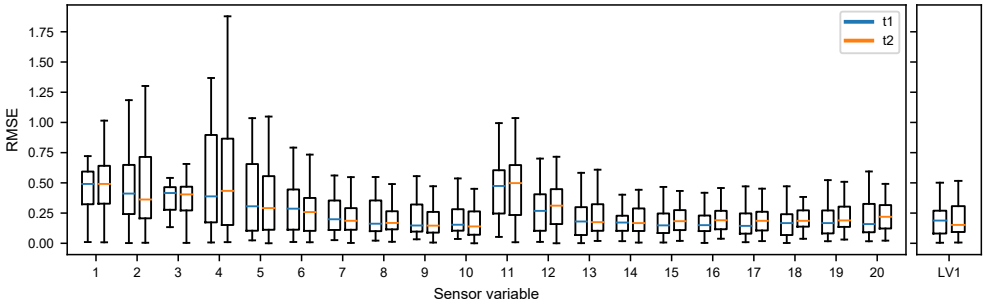


Figure 5.14: *Left*: Cross-validation RMSE of OLS linear regression for each sensor variable as predictor of the material properties t1 and t2. *Right*: Cross-validation RMSE of PLS with number of components $k = 1$. Outliers are not shown.

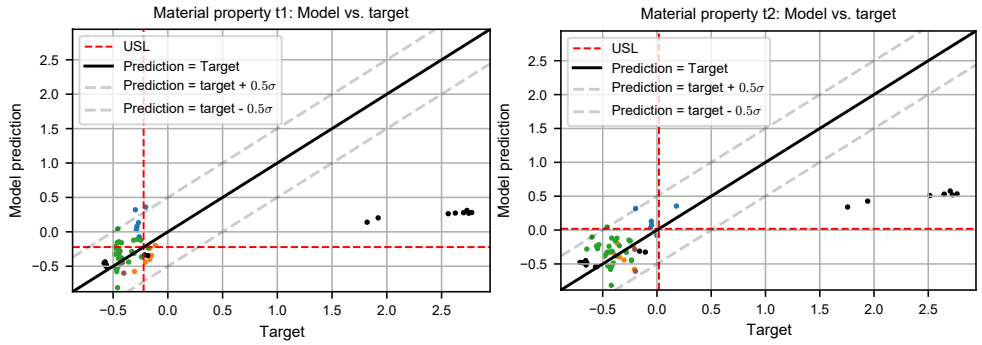


Figure 5.15: One-coil-out cross-validation prediction results of material properties t1 (left panel) and t2 (right panel) using the PLS model with number of components $k = 1$.

as the PLS model. Although differences were small, the predictions of material property t2 were most accurate when based on SV 10 and the predictions of material property t1 were most accurate when based on SV 17. This can be explained by the fact that these sensor variables had low estimated measurement noise, as was discussed in Sec. 5.2 in combination with Fig. 5.7.

We continued with the PLS model, as the latent variable is more robust against sudden changes of the noise pattern in the variables. The coils vary largely in their material properties and the predictions are negatively affected if the full range is not observed. Fig. 5.15 shows for both material properties t1 (left) and t2 (right) the PLS predictions made in the one-coil-out cross-validation. Hence, all predictions shown in the figures are from the folds in which the data point was in the validation

Table 5.2: Fault classification results based on the cross-validation predictions of the PLS model

	TP	**FN**	**FP**	**TN**	**Precision**	**Recall**	F_1**-score**	F_3**-score**
Based on t1	10	7	13	30	0.43	0.59	0.50	0.57
Based on t2	9	0	5	46	0.64	1.00	0.78	0.95
t1 and t2	10	7	13	30	0.43	0.59	0.50	0.57

set. The predictions are mostly within 0.5σ of the target. The testcoil predictions are poor, resulting from a lack of data points in the training set in the range of similar material properties. However, the indicated USL on both axes divides the space into quadrants that are still mostly correctly predicted despite the extreme setting: The bottom-left quadrant corresponds to true negative-, the bottom-right to false negative-, top-right to true positive- and top-left to false positive material out-of-specification classifications. For the three variants of material fault classifications explained in Sec. 5.3, the number of true-positives, false-negatives, false-positives and true-negatives based on the predictions in cross-validation is shown in Table 5.2, along with the corresponding precision, recall, F_1-score and F_3-score. The fault classification based on t2 achieved a perfect recall of 1.00 and a precision of 0.64. Indeed, as can be seen in the right panel of Fig. 5.15, the fault classifier did not miss any material faults and it classified some samples that were close to the USL as faults, being extra careful. The corresponding F_3 score was excellent: 0.95. The recall of the fault classification based on t1 and the combination of t1 and t2 was only 0.59 and the precision 0.43. The corresponding F_3-score was 0.57. The results for these cases were identical because all predictions and measurements that were out of specification with respect to t2 were also out of specification with respect to t1, but not vice versa. Hence, the classifications and target labels based on only t1 and the combined case were equal. Based on the predictions and measurements, it can furthermore be hypothesized that the USL on material property t1 is tighter than the USL on material property t2.

Fig. 5.16 shows the training fit of the PLS model when all data was included in the training set; this was the same fit of which the variable loadings are shown in Fig. 5.13. This linear model fitted the testcoil data points much more accurately than the fits where these data points were not included in the training set, as is shown by comparing Fig. 5.16 to Fig. 5.15. Hence, these outlier data points had a large influence in the model fitting, which can be explained by the sensitivity of least squares optimization to outliers. The out-of-specification classifications were largely

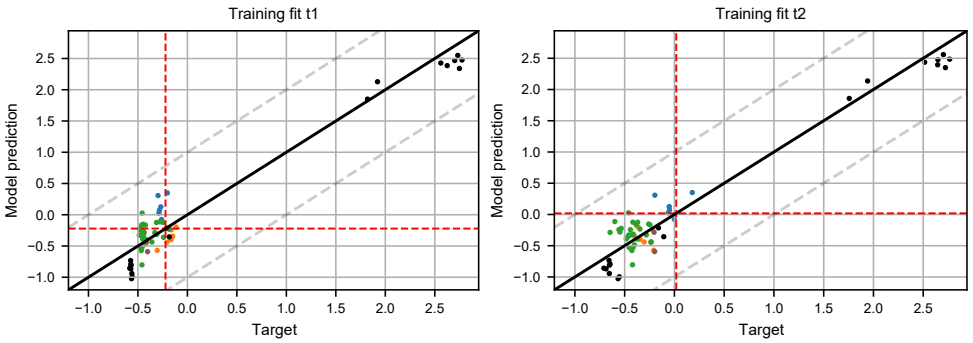


Figure 5.16: Training fit (model vs. target) of PLS with number of PLS components $k = 1$ on all production samples. *Left*: Model vs. target for material property t1. *Right*: Model vs. target for material property t2.

the same for this model. The only difference was a slight increase in precision and recall for the classifications based on t1 and the combination of t1 and t2: 0.65 and 0.48 respectively, with a corresponding F_3 score of 0.62.

The weaker linear relationship observed when excluding the testcoil points was likely related to the fact that these points were only in a small range of material properties and had additional noise caused by the distance between the tensile test and the sensor measurement. We assumed in the rest of the discussion that the linear relationship as observed for the entire dataset generalizes, see Sec. 5.5. Hence, in the rest of the discussion we used the PLS model as fitted on all available data to estimate material properties in production in real-time.

5.4.2 Relation of material properties to known production faults

Besides predicting if the material is out of specification bounds based on non-invasive sensor measurements we are interested whether such measures can be directly related to the occurrence of faulty products recorded during production. The PLS model fitted on all available data points was used to estimate material properties from the sensor measurements that were made in production. The estimations were compared to the logged production faults. As an example of a problematic case of what could be encountered in production, we first show the result of the model predictions made on the entire suspicious testcoil and then consider the other product faults that were logged in the rest of the production.

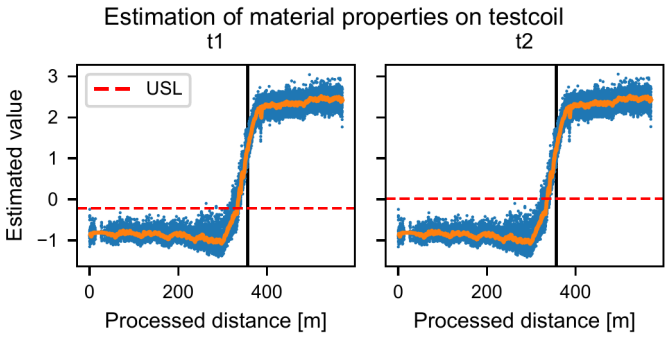


Figure 5.17: Estimation of the material properties t1 (*left*) and t2 (*right*) based on the sensor measurements taken on the testcoil. *Solid orange line*: Moving average over 50 values. *Solid black line*: Marks the point at which the related production coil was removed from the production line.

Testcoil results

In Fig. 5.17 the model estimations of the two material properties are shown for the testcoil. Halfway the coil, the estimated material properties drifted out of their respective specifications. The point at which production with the related production coil was stopped due to cracks that occurred in the products while in the press is marked in the figure. As can be seen, this was shortly after the material properties exceeded the USL. The area to the left of the solid black line, which is enclosed by that line as well as the the moving average line and the USL line, is slightly larger in the left panel. Hence, t1 went out of specification at an earlier stage than t2, which corresponds to the observation made in Sec. 5.4.1 that the specification on t1 is tighter.

Production data

We got 17 logged measurement identifiers of strip steel locations that were linked to product faults later in production and we obtained 25 instances of product faults of which the hour of occurrence was logged, see Sec. 5.2.

Fig. 5.18 shows the model estimations of the material properties for the 17 logged measurement identifiers that were linked to product faults. From the 17 sensor measurements, only one had estimated material properties that were within the specifications. Twelve resulted in estimated material properties that exceeded the USL for both t1 and t2. The remaining four measurements had estimated material properties that exceeded the USL with respect to t1 only. Fig. 5.19 shows the model predictions of property t1 for two full production days. Although the result from

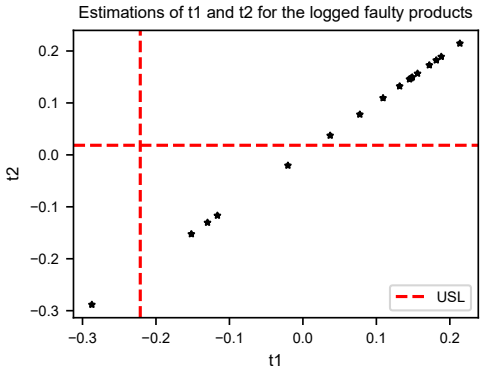


Figure 5.18: Model estimation of material properties for the sensor measurements on the locations of the material that were linked to product faults.

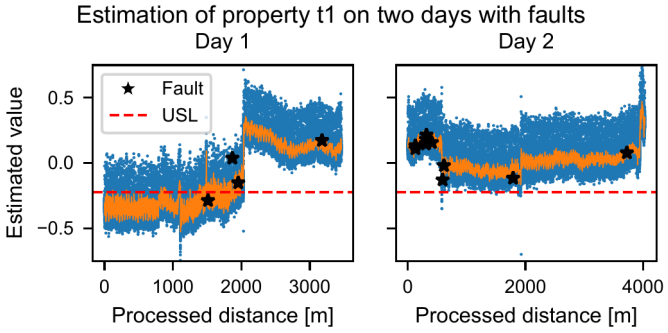


Figure 5.19: Model estimation of material property t1 for two full production days. *Black stars* indicate the model predictions made using the sensor measurements that were linked to product faults. *Solid orange line*: moving average over 50 values.

Fig. 5.18 suggests that a large fraction of the sensor measurements that were linked to product faults had estimated material properties that were out of specification, it is also the case that a large fraction of the estimated material properties in these coils were out of specification but not labeled as faults in production. Therefore the question arises whether the sensor measurements corresponding to predicted faulty material that was not connected to reported product faults can be distinguished from those related to product faults. If this was true a classifier that works on small sample sizes should distinguish those cases. In order to test this hypothesis we trained a supervised model, Generalized Matrix Learning Vector Quantization (GMLVQ) (Schneider et al. 2009), using the implementation from (van Veen et al. 2021). We

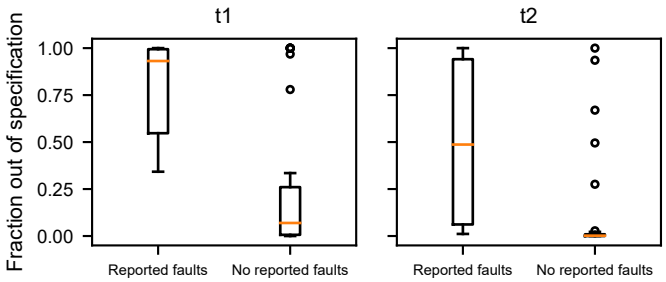


Figure 5.20: Fraction of out of specification model estimations of the material properties t1 and t2 for coils with reported faults and without reported faults.

selected as the positive class the 16 labeled sensor measurements that were linked to product faults and had estimated material properties that were out of specification. As negative class, 16 randomly chosen sensor measurements were selected with estimated out of specification material properties but not linked to product faults. Out of 100 random cross-validation splits with 8 samples validation set size and training with early stopping, the mean validation area under the ROC curve was 0.58, which is barely above random indicating that the classes could not be well distinguished. This suggests that the prediction of undesirable material properties does not necessarily cause a fault every time, but rather increases the risk of a production fault.

As indication of the risk for a product fault we computed the fraction of estimated material properties that were out of specification for each of the production coils with at least 2000 sensor measurements (40 coils). Fig. 5.20 shows that for the six coils with reported faults, the fraction of estimated out of specification material properties was significantly larger than for coils without reported faults. For t1, all coils with reported product faults had a substantial fraction of estimations that exceeded the USL. In the group of 34 coils without reported product faults, this fraction was much lower in general with only a few outliers that had large fractions. Due to the tighter USL of t1, the fractions computed for t1 were always larger than those computed for t2. For this reason, there were some coils with reported faults that exhibited a low fraction for t2 and a high fraction for t1. The results suggest that the occurrence of a high fraction of model estimations for t1 that exceed the USL indicates an increased risk for product faults.

5.5 Discussion

From the cross-validated PLS performance, we found evidence that the relevant information concerning the material properties is mainly contained in the higher frequency sensor variables. This was also initially indicated by the PCA on the modified steel samples, where the different material properties could be well distinguished by the first principal component which mainly consisted of the high frequency components. The latent variables of the sensor measurements and the targets are linearly correlated. We demonstrated that the sensor variables have different levels of measurement noise and that using linear regression with one of the least noisy variables of the higher test frequencies yielded similar estimation performance as the PLS model. Hence, the results are robust with a comparably wide range of higher test frequencies of the sensor.

The model fitting was heavily influenced by the suspicious testcoil measurements which covered a larger variety of material properties than the other coils. Coils with material properties close to the USL also conformed to the linear relationship between sensor measurements and material properties. Given that the material properties of the production coils were in a small normal range of material properties, it is important that these measurements were done as accurately as possible. The distance between the tensile test and the sensor measurement caused small deviations in the sensor measurement. We confirmed this by studying the signal and in a number of cases the sensor measurements showed a large variation, which added uncertainty to the true value of the sensor measurement at the location of the tensile test. Moreover, the administration of new coils was not always exact, such that in a few cases the closest sensor measurements to the location of the tensile test could not be determined and the averaging was done over a suboptimal sample. The accuracy of the current PLS model could be further verified by taking additional tensile test samples from the coils in production and comparing the model estimation with the result from the tensile test. By doing this over a large time span, this would include more material that is out of specification and that results in better estimates of the recall and the precision of the material fault detector.

In the cross-validation the estimations of material property t2 were slightly better than the estimations of t1. Likewise, the material fault classification based on thresholding with respect to the USL had a much better recall for t2 than for t1, which is a crucial performance indicator in mass production settings. However, when relating the material specification predictions to actual reported faults during production, t2 appears very conservative associating a high risk with a lot of the material, while t1 exhibits a better proportion between risk and actual fault. For coils with reported faults, this fraction for property t1 was always large. For the large group of coils

without reported faults, only relatively few coils also had a large fraction of t1 predictions that were out of the specification and the majority had a low fraction. Since logging was manually done by different operators, it could be that faults were not always reported. However, these first results already indicate that a large fraction of out of specification material property estimations is associated with an increased risk of product faults. In scenarios with a clear drift in material properties, such as the one of the test coil, the estimation of material properties from the inline sensor measurements can prevent material that is far out of specification from entering the production line in the future. In these situations the insufficient material quality is most likely the culprit for production faults. For more subtle scenarios, where the estimated material properties slightly exceed the USL, the production of the great majority of products did not result in reported faults. Hence, in order to prevent faults in these situations, it may be crucial to estimate a risk value for faults given the sensor measurements and raise an alert or adjust the parameters of the production machinery suitable for the encountered material.

5.6 Conclusion and Outlook

This contribution discusses an exemplary industry 4.0 case: The real-time fault detection and quality control in a mass production line. Material measurements gathered by an NDT soft sensor were analysed in three scenarios:

Firstly, we analysed sensor measurements made on deliberately altered material and we showed that the sensor was able to detect these modifications.

Secondly, a PLS model was fitted on a dataset that included sensor measurements and tensile tests results of steel from two settings: Steel from production and steel from a selected coil, called testcoil, that contained insufficient material quality. The testcoil was closely related to a coil that had to be removed from production because the products made from it contained faults. This model was initially used to estimate the rest of the material properties of the testcoil. The results obtained in these experiments showed the potential of the strategy for achieving full real-time test coverage and for the early detection of insufficient material quality, preventing it from entering the production line. Hence, in the future, the detection and prevention of material faults in production could save extremely high costs due to less damage to the production machinery and fewer thrown away products.

Lastly, the model was employed for an analysis of 108 km of strip steel coil encountered during the full run of the experiment. We demonstrated evidence in preventing the more subtle faults, by revealing the relationship between large fraction of out of specification estimations and reported faults. Furthermore, the material

specification may not always directly lead to faults, but could have a direct influence on the durability of tooling.

A future direction is to combine the model estimations and risk determination with machine parameters, to identify optimal settings for the specific properties of the material, which has the potential to widen the specification limits of the material. In addition, a cost analysis should be made in order to adjust the sensitivity of the material fault classifier to optimize with respect to the ratio of the costs of false positives to false negatives. One could start with a sensitive setting (decreasing the USL) and gradually decrease the sensitivity while monitoring the number of product faults. This would optimize the USL with respect to the cost of faults. Our findings on the sensor's consistency on material with different properties are further investigated by the company. Other future investigations will incorporate process knowledge, the physics of the sensor, other inline measurements and the interplay of the tooling with certain material properties for the prevention of faults.

Published as:

M. Straat, M. Kaden, M. Gay, T. Villmann, A. Lampe, U. Seiffert, M. Biehl and F. Melchert – *"Learning vector quantization and relevances in complex coefficient space"*, Neural Computing and Applications, vol. 32, no. 24, pp. 18085-18099, Springer, 2020.

M. Münch, M. Straat, M. Biehl and F.M. Schleif – *" Complex-Valued Embeddings of Generic Proximity Data"*, Structural, Syntactic, and Statistical Pattern Recognition. S+SSPR 2021. Lecture Notes in Computer Science, vol. 12644, Springer, Cham, 2021.

Chapter 6

Prototypes and Matrix Relevance Learning in Complex Coefficient Space

Abstract

We consider the classification of time-series and similar functional data which can be represented in complex Fourier- and wavelet coefficient space. We apply versions of Learning Vector Quantization (LVQ) which are suitable for complex-valued data, based on the so-called Wirtinger calculus. It makes possible the formulation of gradient based update rules in the framework of cost-function based Generalized Matrix Learning Vector Quantization (GMLVQ). Alternatively, we consider the concatenation of real and imaginary parts of Fourier coefficients in a real-valued feature vector and the classification of time domain representations by means of conventional GMLVQ. In addition, we consider the application of the method in combination with wavelet-space features to the classification of heartbeats in electrocardiogram (ECG) data. Besides the interpretation of the LVQ classifier in the domain of the transform, time-domain interpretability is retained by transforming the prototypes and relevance matrix using inverse transformations appropriately. With feature relevance information and prototypes in both the time and the transform domain, our approach provides rich insight into the classification problem.

6.1 Introduction

Time series constitute an important example of *functional data* (Ramsay and Silverman 2006): Their time-domain discretized vector representations comprise components which reflect the temporal order and are often highly-correlated over characteristic times. This is in contrast to more general datasets, where the feature vectors are concatenations of more or less independent quantities and without any meaningful interpretation of their order.

The machine learning analysis of time series data, e.g. for the purpose of classification, should take into account their functional nature. Recently, prototype-based systems have been put forward, which employ the representation of data and prototypes in terms of suitable basis functions (Melchert et al. 2016a, Melchert et al. 2016b). In addition, corresponding adaptive distance measures can be defined and trained in the space of expansion coefficients (Kästner et al. 2011, Biehl et al. 2014, Biehl et al. 2016). Hence, the functional nature of data is taken advantage of, explicitly. At the same time, it is possible to compress high-dimensional data by functional approximations, thus reducing computational effort and - potentially - the risk of over-fitting. Examples of the basic approach include the application of wavelet representations of mass spectra (Schneider, Biehl, Schleif and Hammer 2007) or hyperspectral images (Mendenhall and Merenyi 2006), and also polynomial expansions of smooth functional data (Melchert et al. 2016a, Melchert et al. 2016b).

In the context of signal processing, the Discrete Fourier Transform (DFT) to the frequency domain is a popular tool which can be applied to time series or more general, sequential data. In the following the discussion is presented mostly in terms of actual time series, but it is understood that methods and results would carry over to suitable sequential data from other contexts. The standard formulation of the DFT resorts to the determination of complex coefficients, conveniently. Hence, we suggest and study the combination of DFT functional representations with the extension of Generalized Matrix Learning Vector Quantization (GMLVQ) (Schneider, Biehl and Hammer 2007), (Schneider et al. 2009) to complex feature space (Gay et al. 2016). We present furthermore the formalism to back-transform the resulting prototypes and relevance matrix to the time domain, thus retaining the intuitive interpretability of the LVQ approach. With the prototypes and relevance matrix in both the time domain and the transform domain, the methodology provides a multi-perspective insight into the classification problem. We apply the suggested framework to a number of benchmark datasets (Chen et al. 2015) and study, among other aspects, the dependence of the performance on the approximation quality, i.e. the number of coefficients considered. In addition, we compare performance with an approach that resorts to the concatenation of the imaginary and real parts of coefficients in a real-valued feature vector. The application of conventional GMLVQ classification in the time domain serves as an important and intuitive baseline for comparison of performances and for the interpretation of the obtained relevance matrices.

The Discrete Wavelet Transform (DWT) is another frequently used tool for signal analysis (Mallat 2008). In contrast to the standard DFT, the DWT also captures time information in addition to frequency information. Additionally, the more recently proposed Dual-Tree Complex Wavelet Transform (DTCWT) provides approximate shift invariance and directionally selective filters (Kingsbury 2001, Selesnick et al.

2005). We describe the combination of complex wavelet representations with the extension of GMLVQ to complex feature space and discuss the interpretation of the classifiers in wavelet space and in the time domain after a signal reconstruction of the prototypes. As an application example, the methodology is used for the purpose of classifying heartbeats in electrocardiogram (ECG) data from the MIT-BIH dataset (Moody and Mark 2001). We consider a general classifier and a patient-specific classifier. The classification performance is assessed for full wavelet representations and compressed representations.

6.2 The mathematical framework

In this section we present the mathematical framework of our methodology. After a short description of the DFT and the DTCWT, we discuss the adaptation of the GMLVQ machine learning algorithm that is necessary for correct functioning on complex-valued data. Lastly, the back-transformation of the prototypes and relevance matrix to the original time domain of the data is discussed for both transforms.

6.2.1 Discrete Fourier Transform

The uniform sampling of a continuous process $f(t)$ with a sampling interval of ΔT seconds results in a potentially high-dimensional feature vector $\boldsymbol{x} \in \mathbb{R}^N$ containing the values of $f(t)$ at the sampling times:

$$x[p] = f(p\Delta T)\,, \quad p = 0, 1, 2, ..., N-1\,, \tag{6.1}$$

covering $\Delta T(N-1)$ seconds of the continuous process. The time domain vector $\boldsymbol{x} \in \mathbb{R}^N$ can also be represented as a linear combination of N sampled complex sinusoids:

$$x[p] = \sum_{k=0}^{N-1} x_k^{(f)} g_k[p]\,, \quad p = 0, 1, 2, ..., N-1\,, \tag{6.2}$$

where

$$g_k[p] = e^{-i2\pi pk/N} = \cos\left(\frac{2\pi k}{N}p\right) - i\sin\left(\frac{2\pi k}{N}p\right) \tag{6.3}$$

is the complex sinusoid of radial frequency $2\pi k/N$ and the coefficients $x_k^{(f)} \in \mathbb{C}$ are the DFT coefficients (Brigham 1974):

$$x_k^{(f)} = \sum_{p=0}^{N-1} x[p] g_k[p]\,, \quad k = 0, 1, 2, ..., N-1\,. \tag{6.4}$$

As in Eq. (6.2) and Eq. (6.4), for the rest of the discussion the superscript (f) is used to denote a vector of complex-valued Fourier coefficients or a matrix of complex-valued feature relevance values. It should be noted that the coefficients of $\boldsymbol{x}^{(f)} \in \mathbb{C}^N$ are conjugate symmetric for real-valued signals, so that the non-redundant information is contained in the first $\lfloor N/2 \rfloor + 1$ coefficients: $x_k^{(f)}$, $k = 0, 1, ..., \lfloor N/2 \rfloor$. By truncating the vector further to a number $n < \lfloor N/2 \rfloor + 1$ of lower frequency coefficients, the resulting vector $\hat{\boldsymbol{x}}^{(f)} \in \mathbb{C}^n$ represents a smooth low-pass approximation $\hat{\boldsymbol{x}} \in \mathbb{R}^N$ of the original vector $\boldsymbol{x} \in \mathbb{R}^N$, obtained by Eq. (6.2) for the truncated vector. Note that in some classification problems, the classes in the dataset can be differentiated by the values of the higher frequency coefficients. In such cases, classification performance may decrease when those frequencies are omitted from the signals.

As the Fourier coefficients are defined as dot products between the time domain signal $\boldsymbol{x}$ and the corresponding sampled complex sinusoids according to Eq. (6.4), it is possible to write the full transformation as a matrix equation:

$$\boldsymbol{x}^{(f)} = \boldsymbol{F}\boldsymbol{x}\,, \tag{6.5}$$

where $\boldsymbol{F} \in \mathbb{C}^{N \times N}$ is the transformation matrix containing the complex sinusoids in the rows of the matrix. The multiplication with $\boldsymbol{F}$ in Eq. (6.5) has a computational cost of $O(N^2)$. It is usually efficiently computed using the Fast Fourier Transform that reduces the computational cost to $O(N \log N)$.

6.2.2 Dual-Tree Complex Wavelet Transform

A shortcoming of the standard formulation of the DFT is the lack of time localization of the frequency content, i.e., there is no mapping between time and the prevalence of the frequencies. An advantage of the wavelet transform is that it provides time localization of the activity of the basis functions.

The one-dimensional Continuous Wavelet Transform (CWT) (Mertins and Mertins 1999) is defined as:

$$W(\tau, s)_x^{\psi} = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} x(t)\psi^* \left(\frac{t-\tau}{s} \right) dt\,. \tag{6.6}$$

In the above definition, $\psi(\cdot)$ is the mother-wavelet that characterizes the shape of the wavelet, $s \in \mathbb{R}^+$ is the scale of the wavelet, $\tau \in \mathbb{R}$ is the translation of the wavelet. In other words, the mother-wavelet $\psi(\cdot)$ is the main function shape from which the specific scaled and translated wavelets $\psi((t-\tau)/s)$ are derived (Mallat 2008).

The more compressed wavelets that are obtained for larger values of s results provide more time resolution than the more dilated functions corresponding to smaller values of s.

The discrete version of the CWT, the DWT, is efficiently implemented as a repeated filtering process referred to as *sub-band coding* (Akansu and Haddad 1992): A high- and low-pass filter h and l are repeatedly applied to the discrete signal $\boldsymbol{x}$ in each level of the decomposition upto the highest level j. For each level $1 \leqslant i \leqslant j$, detail coefficients $\boldsymbol{d}_i$ and approximation coefficients $\boldsymbol{a}_i$ are obtained from the high- and low-pass filter, respectively:

$$\boldsymbol{d}_i[k] = \sum_p x[p] \cdot h[2k - p] \,. \tag{6.7}$$

For $i = 1$, we obtain $\boldsymbol{d}_1$ following Equation 6.7 and $\boldsymbol{a}_1$ by an application of the low-pass filter l:

$$\boldsymbol{a}_i[k] = \sum_p x[p] \cdot l[2k - p] \,, \tag{6.8}$$

for $i = 1$. In the next level $i = 2$, h and l are applied on $\boldsymbol{a}_1$, reducing the analyzed frequency window by a factor two in each step. The output of the DWT is the concatenation of all detail coefficients $\{\boldsymbol{d}_i\}$ for $1 \leqslant i \leqslant j$ and the approximation coefficients of the last level $\boldsymbol{a}_j$:

$$\boldsymbol{x}^{(w)} = [\{\boldsymbol{d}_i\}, \boldsymbol{a}_j] \in \mathbb{R}^N \,, \quad 1 \leqslant i \leqslant j \,. \tag{6.9}$$

As in (6.9), in the rest of the discussion the subscript w is used to denote a vector of wavelet coefficients or a matrix of relevance values of the wavelet coefficients.

The original DWT is not shift-invariant. In (Kingsbury 2001), a version of the discrete wavelet transform was proposed which attains *approximate* shift-invariance: The Dual-Tree Complex Wavelet Transform (DTCWT). This transform uses an additional filter tree: One tree produces the real parts and the other tree produces the imaginary parts of the coefficients of the DTCWT. Therefore the application of the DTCWT up to level j yields vectors

$$\boldsymbol{x}^{(w)} = [\{\boldsymbol{d}_i\}, \boldsymbol{a}_j] \in \mathbb{C}^N \,, \quad 1 \leqslant i \leqslant j \,. \tag{6.10}$$

6.2.3 Formulation of GMLVQ using Wirtinger calculus

For the purpose of classifying complex-valued feature vectors, we consider an appropriate adaptation of the GMLVQ algorithm similar to (Gay et al. 2016, Bunte et al. 2012). Following the general prescription outlined in (Biehl et al. 2016), we consider a dataset of complex-valued feature vectors and corresponding class labels:

$$\mathcal{D} = \{\boldsymbol{x}_i, \, y_i\}_{i=1}^P \,, \quad \boldsymbol{x}_i \in \mathbb{C}^N \,, \quad y_i \in [1, 2, ..., C] \,, \tag{6.11}$$

where C is the number of classes in the dataset, P is the number of data points in the dataset and N is the number of complex-valued features of the data points. In the GMLVQ algorithm, K prototypes are defined with corresponding class labels:

$$\boldsymbol{W} = \{\boldsymbol{w}_i, c_i\}_{i=1}^{K}\,, \quad \boldsymbol{w}_i \in \mathbb{C}^N\,, \quad c_i \in [1, 2, ..., C]\,. \tag{6.12}$$

As each of the C classes must have at least one prototype representing it, the number of prototypes must satisfy $K \geqslant C$.

The similarity between a given data point i and prototype j is computed with a quadratic distance measure:

$$d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_j] = (\boldsymbol{x}_i - \boldsymbol{w}_j)^H \boldsymbol{\Omega}^H \boldsymbol{\Omega} (\boldsymbol{x}_i - \boldsymbol{w}_j) \in \mathbb{R}\,, \quad 1 \leqslant i \leqslant P\,, \quad 1 \leqslant j \leqslant K\,, \tag{6.13}$$

in which matrix $\boldsymbol{\Omega} \in \mathbb{C}^{N \times N}$ represents a linear transform and the superscript H denotes the Hermitian transpose of complex-valued vectors and matrices. In the GMLVQ system, a data point $\boldsymbol{x}_i$ with unknown label is classified according to the class label c_j of the closest prototype:

$$c_j = \arg\min_j d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_j]\,. \tag{6.14}$$

Learning amounts to the adaptation of the prototypes $\boldsymbol{W}$ and the linear map $\boldsymbol{\Omega}$ guided by the dataset $\mathcal{D}$ of labeled data points, to the best-possible expected classification performance of new data points. Usually, the training is performed on subset of P_T number of data points from $\mathcal{D}$ and the remaining P_V number of data points are used for monitoring the classification performance on data points that the system does not used for learning. In the generalized versions of LVQ, learning is guided by a cost function (Sato and Yamada 1995):

$$E = \sum_{i=1}^{P_T} \Phi(e_i)\,, \quad e_i = \frac{d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_J] - d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_K]}{d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_J] + d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_K]} \in [-1, 1]\,, \tag{6.15}$$

where $\boldsymbol{w}_J$ is the closest prototype with a matching label $y_i = c_J$ and $\boldsymbol{w}_K$ is the closest prototype with a different label $y_i \neq c_K$:

$$J = \arg\min_j d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_j] \text{ such that } c_j = y_i\,,$$
$$K = \arg\min_k d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_k] \text{ such that } c_k \neq y_i\,.$$

In Eq. (6.15), E is the total cost associated with the GMLVQ configuration and e_i is the cost of data point $\boldsymbol{x}_i$. The modulation function $\Phi : \mathbb{R} \to \mathbb{R}$ determines learning behavior in gradient-based learning. Here we use $\Phi(x) = x$. To minimize the total

cost E with respect to the prototypes $\boldsymbol{W}$ and matrix $\boldsymbol{\Omega}$, gradient descent is used. For the computation of the gradients, we use the formalism of Wirtinger calculus (Wirtinger 1927) for taking derivatives and gradients of functions with respect to complex-valued variables, as proposed in (Gay et al. 2016) for GMLVQ. The required complex-valued derivatives are then given as follows:

$$\frac{\partial d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_j]}{\partial \boldsymbol{w}_j^*} = -\boldsymbol{\Omega}^H \boldsymbol{\Omega}(\boldsymbol{x}_i - \boldsymbol{w}_j)\,, \tag{6.16}$$

$$\frac{\partial d_{\boldsymbol{\Omega}}[\boldsymbol{x}_i, \boldsymbol{w}_j]}{\partial \boldsymbol{\Omega}^*} = \boldsymbol{\Omega}(\boldsymbol{x}_i - \boldsymbol{w}_j)(\boldsymbol{x}_i - \boldsymbol{w}_j)^H\,. \tag{6.17}$$

Then for a data point $\boldsymbol{x}_i$, the gradient of the data point's cost e_i with respect to prototypes $\boldsymbol{w}_J$, $\boldsymbol{w}_K$ and transformation matrix $\boldsymbol{\Omega}$ is:

$$\begin{aligned}
\frac{\partial e_i}{\partial \boldsymbol{w}_J} &= -2\frac{d(\boldsymbol{x}_i, \boldsymbol{w}_K)}{[d(\boldsymbol{x}_i, \boldsymbol{w}_J) + d(\boldsymbol{x}_i, \boldsymbol{w}_K)]^2}\boldsymbol{\Omega}^H\boldsymbol{\Omega}(\boldsymbol{x}_i - \boldsymbol{w}_J)\,,\\
\frac{\partial e_i}{\partial \boldsymbol{w}_K} &= 2\frac{d(\boldsymbol{x}_i, \boldsymbol{w}_J)}{[d(\boldsymbol{x}_i, \boldsymbol{w}_J) + d(\boldsymbol{x}_i, \boldsymbol{w}_K)]^2}\boldsymbol{\Omega}^H\boldsymbol{\Omega}(\boldsymbol{x}_i - \boldsymbol{w}_K)\,,\\
\frac{\partial e_i}{\partial \boldsymbol{\Omega}} &= 2\frac{d(\boldsymbol{x}_i, \boldsymbol{w}_K)}{[d(\boldsymbol{x}_i, \boldsymbol{w}_J) + d(\boldsymbol{x}_i, \boldsymbol{w}_K)]^2}\boldsymbol{\Omega}(\boldsymbol{x}_i - \boldsymbol{w}_J)(\boldsymbol{x}_i - \boldsymbol{w}_J)^H\\
&\quad - 2\frac{d(\boldsymbol{x}_i, \boldsymbol{w}_J)}{[d(\boldsymbol{x}_i, \boldsymbol{w}_J) + d(\boldsymbol{x}_i, \boldsymbol{w}_K)]^2}\boldsymbol{\Omega}(\boldsymbol{x}_i - \boldsymbol{w}_K)(\boldsymbol{x}_i - \boldsymbol{w}_K)^H\,.
\end{aligned} \tag{6.18}$$

A comparison of the above gradients for complex-valued data with the gradients for real-valued data, found for instance in (Schneider, Biehl and Hammer 2007, Schneider et al. 2009), that the two equations are highly similar in form. To obtain the true gradient of E with respect to the prototypes $\boldsymbol{W}$ and transformation matrix $\boldsymbol{\Omega}$, the gradients in (6.18) over all P_T training examples should be accumulated before applying the gradient descent update. These are *steepest descent* updates of the cost function E. Alternatively, one could accumulate gradients over smaller subsets of the training set, which amounts to *stochastic gradient descent*. In case the gradient of only one randomly selected data point is used, the update for each randomly selected data point $\boldsymbol{x}_i$ is:

$$\begin{aligned}
\boldsymbol{w}_J &:= \boldsymbol{w}_J - \alpha\frac{\partial e_i}{\partial \boldsymbol{w}_J}\,,\\
\boldsymbol{w}_K &:= \boldsymbol{w}_K - \alpha\frac{\partial e_i}{\partial \boldsymbol{w}_K}\,,\\
\boldsymbol{\Omega} &:= \boldsymbol{\Omega} - \beta\frac{\partial e_i}{\partial \boldsymbol{\Omega}}\,,
\end{aligned}$$

where α and β are the learning rates for the prototypes and transformation matrix, respectively.

The effect of the gradient-based minimization of the cost function E is increasing the number of correct classifications in the classification scheme (Eq. (6.14)) and the classification margins to improve the differentiation of the classifications. The transformation matrix $\mathbf{\Omega}$ is adapted to achieve better discrimination of the classifications in the transformed space. Simultaneously, in the transformed space the prototypes are adapted in order to be closer to data points of matching class label, while moving further from data points with a different class label. The symmetric matrix $\mathbf{\Lambda} = \mathbf{\Omega}^H\mathbf{\Omega} \in \mathbb{C}^{N\times N}$ is called the *relevance matrix*. The diagonal elements $\Lambda_{ii} \in \mathbb{R}$ are the squared norms $||\mathbf{\Omega}_i||^2$ of the column vectors of the transformation matrix, and therefore represent the learned importance of the corresponding features in the classification problem. The off-diagonal elements $\Lambda_{ij} \in \mathbb{C}$ for $i \neq j$ respresent

After an update step, the transformation matrix $\mathbf{\Omega}$ is normalized so that for the relevance matrix $\mathbf{\Lambda}$ it holds that $\mathrm{tr}(\mathbf{\Lambda}) = 1$:

$$\mathbf{\Omega} := \frac{\mathbf{\Omega}}{\sqrt{\sum_{i=1}^{N} \mathbf{\Omega}_i \cdot \mathbf{\Omega}_i}} \,. \tag{6.19}$$

6.2.4 Back-transformation

Using the adaptation of GMLVQ as described in the previous section it is possible to train a GMLVQ classifier on complex-valued data, as obtained from the DFT or the DTCWT for instance. The GMLVQ algorithm that operates on the data in the space of the transform learns prototypes in complex-valued coefficient space and a matrix of feature relevance values of the coefficients. The classifier is therefore interpretable in the space of the transform. In this section, we discuss back-transformations to the time domain in order to also obtain a time domain interpretation of the classifier.

Fourier space

The K prototypes $\{\boldsymbol{w}_j^{(f)}, c_j\}_{j=1}^{K}$ as obtained from training in Fourier space can be interpreted as class-typical Fourier space representations, i.e., the class-typical contributions of the sinusoidal components. The diagonal $\mathrm{diag}(\mathbf{\Lambda}^{(f)})$ of the relevance matrix indicates the importance of each of the sinusoidal components in differentiating between the classes.

The Fourier space prototypes can be transformed to the time-domain using the Inverse Discrete Fourier Transform (IDFT):

$$w_j[p] = \frac{1}{N} \sum_{k=0}^{N-1} w_{jk}^{(f)} e^{i2\pi pk/N} \,, \quad p = 0, 1, 2, ..., N-1 \,, \quad j = 1, 2, ..., K \,. \tag{6.20}$$

As it is possible to write the complex-valued data points and prototypes as time-domain vectors transformed with the Fourier transformation matrix $\boldsymbol{F}$, the distance measure can be written as:

$$\begin{aligned} d[\boldsymbol{x}_i^{(f)}, \boldsymbol{w}_j^{(f)}] &= (\boldsymbol{x}_i^{(f)} - \boldsymbol{w}_j^{(f)})^H \boldsymbol{\Lambda}^{(f)} (\boldsymbol{x}_i^{(f)} - \boldsymbol{w}_j^{(f)}) \\ &= (\boldsymbol{x}_i - \boldsymbol{w}_j)^T \boldsymbol{F}^H \boldsymbol{\Lambda}^{(f)} \boldsymbol{F} (\boldsymbol{x}_i - \boldsymbol{w}_j) \\ &= (\boldsymbol{x}_i - \boldsymbol{w}_j)^T \boldsymbol{\Lambda} (\boldsymbol{x}_i - \boldsymbol{w}_j)\,, \end{aligned} \tag{6.21}$$

where $\boldsymbol{x}_i \in \mathbb{R}^N$ and $\boldsymbol{w}_j \in \mathbb{R}^N$ are the time-domain representations. Eq. (6.21) shows that the matrix $\boldsymbol{\Lambda} = \boldsymbol{F}^H \boldsymbol{\Lambda}^{(f)} \boldsymbol{F} \in \mathbb{R}^{N \times N}$ are the coefficients for the quadratic distance measure in the time-domain, which can be interpreted as time domain feature relevance values.

GMLVQ could also be used directly on the data in the time-domain. Besides the potential of improving classification accuracy, we must note that our approach also has the ability to reduce the number of parameters in GMLVQ ($n < N$) and at the same time keep the time-domain intuitiveness. Since the number of parameters in GMLVQ scales quadratically with the number of dimensions, the computational effort in the training process is considerably reduced.

Wavelet space

After training in the space of complex-valued wavelet coefficients, each prototype $(\boldsymbol{w}_j^{(w)}, c_j)$ can be interpreted as a class-typical wavelet space representation for class c_j. In our example application the prototypes are the typical wavelet representations of various classes of heartbeats. The real-valued diagonal $\mathrm{diag}(\boldsymbol{\Lambda}^{(w)}) \in \mathbb{R}^N$ of the relevance matrix will reflect the importance of the wavelet coefficients of the different scales of wavelets. The in general complex-valued off-diagonal elements reflect the relevance of correlations between wavelet space coefficients.

It is also possible to interpret the wavelet space prototypes in the original time domain, by using the inverse wavelet transform to back-transform the prototypes to the time domain. The inverse transform starts at the detail- and approximation coefficients at the highest level j and works backwards by repeatedly applying an up-sampling and an application of the reconstruction high-pass and low-pass filters on the analysis coefficients until the time domain signal after the lowest level is obtained. The reconstruction filters are simply the reverse of the analysis filters used in the forward transform (Mertins and Mertins 1999).

The back-transformation of the relevance matrix could be performed in a similar way: Working its way backwards by repeated up-sampling and application of the reconstruction filters starting from the highest level. After the first level, we obtain a matrix of relevance values in the time-domain. However, we will not back-transform

the relevance matrix obtained from learning in wavelet space, as the time localization of the wavelet transform already provides time domain interpretability.

6.3 Experiments: learning in Fourier space

In this section we describe the set-up of the experiments for studying the usefulness of the method in combination with Fourier space representations.

6.3.1 Workflows

For our investigation into the usefulness and performance of the proposed method, we compare and study the results for the following scenarios:

1. Train a GMLVQ system using the feature vectors $\boldsymbol{x} \in \mathbb{R}^N$ in the original time domain and evaluate the system on the test data. This serves as the baseline performance.

2. Transform the feature vectors $\boldsymbol{x}_i \in \mathbb{R}^N$ to complex Fourier space and truncate $n = [6, 11, ..., 51]$ number of Fourier coefficients, obtaining vectors $\boldsymbol{x}^{(f)} \in \mathbb{C}^n$ for each truncation level. On each of these representations a GMLVQ system is trained. The training results in a classifier defined by prototypes $\boldsymbol{w}^{(f)} \in \mathbb{C}^n$ and complex relevance matrix $\boldsymbol{\Lambda}^{(f)} \in \mathbb{C}^{n \times n}$. The classification accuracy of the classifiers is evaluated on validation sets, see Sec. 6.3.3.

3. Similar to the previous scenario 2, transform the feature vectors $\boldsymbol{x}_i \in \mathbb{R}^N$ to complex Fourier space and truncate at $n = [6, 11, ..., 51]$ coefficients obtaining vectors $\boldsymbol{x}^{(f)} \in \mathbb{C}^n$, but here the complex-valued coefficients are represented by the concatenation of the real and imaginary parts into a real-valued feature vector: $\boldsymbol{x}^{(f)} := \left[\Re(\boldsymbol{x}^{(f)}) \quad \Im(\boldsymbol{x}^{(f)})\right]^T \in \mathbb{R}^{2n}$. We train a GMLVQ system on each of these representations resulting in classifiers defined by prototypes $\boldsymbol{w}^{(f)} \in \mathbb{R}^{2n}$ and a real-valued relevance matrix $\Lambda^{(f)} \in \mathbb{R}^{2n \times 2n}$. The classification accuracy of the classifiers is evaluated on corresponding validation sets, see Sec. 6.3.3.

4. Transform the feature vectors $\boldsymbol{x}_i \in \mathbb{R}^N$ to Fourier space for the same numbers $n = [6, 11, ..., 51]$ of coefficients as in scenarios 2 and 3. The Fourier space representations $\boldsymbol{x}_i^{(f)} \in \mathbb{C}^n$ are then back-transformed to the time domain using the IDFT (Eq. (6.20). The obtained time domain vectors $\hat{\boldsymbol{x}}_i \in \mathbb{R}^N$ are low-pass smoothed versions of the original time domain vectors $\boldsymbol{x}_i \in \mathbb{R}^N$. The GMLVQ systems are now trained and evaluated on the smoothed time domain

feature vectors $\hat{\boldsymbol{x}}_i$. Observing the correlation and differences of the obtained results with the results in scenario 2 and 3 helps to explain how much of the performance changes in Fourier space can be attributed to the elimination of the high frequency signals and how much can be attributed to training in Fourier space.

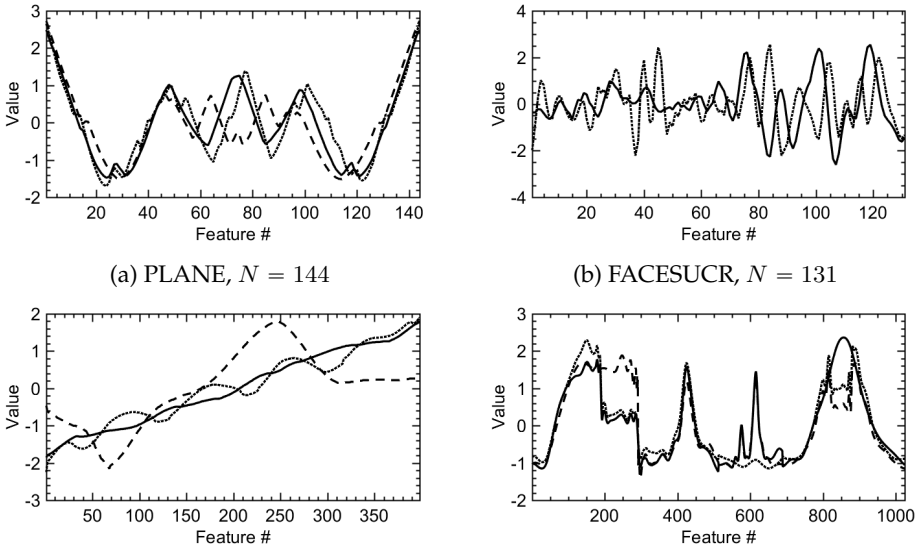
6.3.2 Training settings and parameter values

Prior to training, the training data were transformed such that all dimensions have zero mean and unit variance. The test data were transformed correspondingly using the mean and standard deviation of the features in the training set. This normalization is useful for the intuitive interpretation of the relevance matrix, since in this case the relevance matrix does not compensate for the different scales of the features. The relevance values will therefore be directly comparable. All systems used one prototype per class, which was initialized to a small random deviation from the corresponding class conditional mean. The relevance matrix was initialized proportional to the identity matrix. Furthermore, a batch gradient descent along the lines of (Papari et al. 2011) was applied as the optimization procedure using the default parameters from (Biehl 2018). All classification results were obtained from the model as it was trained after 300 batch training steps. Please note that the goal of the experiments is to gain insights into the properties and highlight potential advantages of the proposed method. The presented classification accuracies may be further improved through the implementation of early-stopping strategies or regularization methods.

6.3.3 Example Datasets

The suggested approach and the workflows as given in Sec. 6.3.1 were applied to four time series datasets from the UCR repository (Chen et al. 2015). The names of the datasets and their properties are in Table 6.1. The four datasets all contain time series with more or less periodic behavior. The repository does not provide any further details nor annotations about the origin and interpretation of the datasets. As an illustration Figure 6.1 shows a few data points per class for each dataset and provides an idea about their properties and and complexity.

Note that it is required that $\lfloor N/2 \rfloor + 1 \geqslant n_{max}$, where $n_{max} = 51$, the maximum number of coefficients we consider in the experiments (see scenario 2). As mentioned in Section 6.2.1, all information is contained in $\lfloor N/2 \rfloor + 1$ coefficients which is therefore the upper-bound for the number of approximation coefficients n. As can be seen in Table 6.1 all the considered datasets satisfy $\lfloor N/2 \rfloor + 1 \geqslant 51$.



(a) PLANE, $N = 144$ (b) FACESUCR, $N = 131$

(c) SYMBOLS, $N = 398$ (d) MALLAT, $N = 1024$

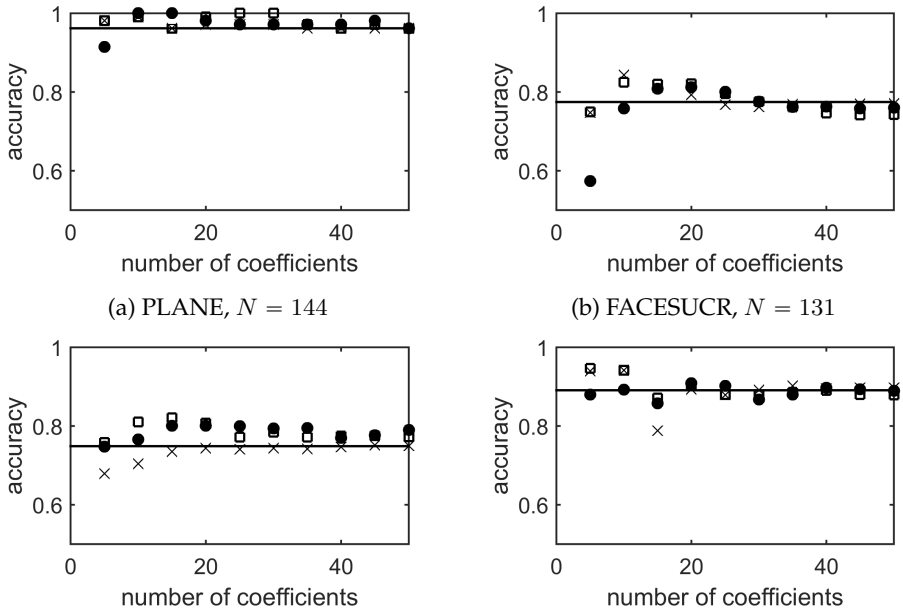
Figure 6.1: Example time series of each dataset. For the Plane, Symbols and MALLAT datasets, one example is shown from the first three classes in the dataset. For the FacesUCR dataset, one example is shown for the first two classes in the dataset.

Table 6.1: Time series datasets

Dataset name	classes	sampling points	samples	
			training	validation
PLANE	7	144	105	105
MALLAT	8	1024	55	2345
SYMBOLS	6	398	25	995
FACESUCR	14	131	200	2050

6.3.4 Performance evaluation

The performance for the different scenarios was evaluated by the classification accuracy, i.e. the fraction of correctly classified feature vectors of the validation set as indicated in Table 6.1. For scenario 1 this was one baseline validation classification accuracy. For the functional Fourier approximation scenarios, 2, 3 and 4, a validation



(a) PLANE, $N = 144$

(b) FACESUCR, $N = 131$

(c) SYMBOLS, $N = 398$

(d) MALLAT, $N = 1024$

Figure 6.2: Percentage of correctly classified vectors in the test sets for each dataset. The solid line represents the classification result in the original time domain of the data. *Filled circles* show the classification accuracy in the n-coefficient complex Fourier space of the data. *Empty squares* show the classification accuracy in the n-coefficient Fourier space where the real and imaginary parts of the complex features are concatenated yielding real feature vectors. *Crosses* show the classification accuracy on the smooth data in the original space that was obtained by an inverse transform of the Fourier representation. For each dataset the number of dimensions N of the original feature vectors is indicated.

classification accuracy was obtained for each level of approximation with number of Fourier coefficients n. The results will be compared with each other and discussed.

6.4 Results and Discussion

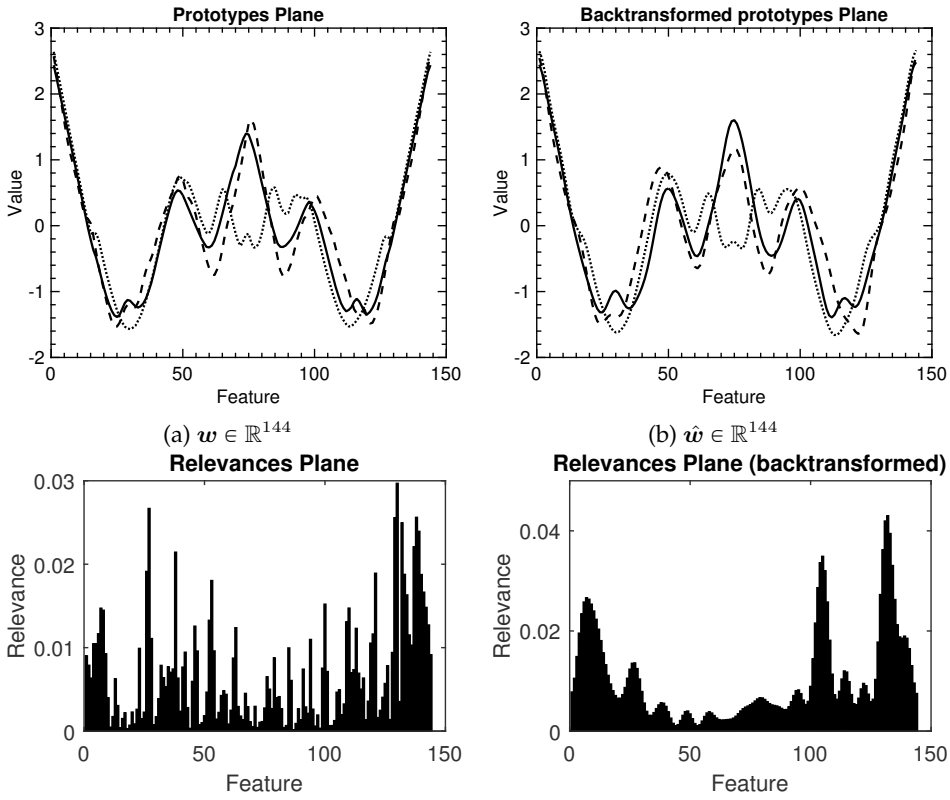
The results displayed in Figure 6.2 suggest that, in general, the classification results of functional data using a Fourier representation are comparable to or better than the baseline performance in the original time domain of the data.

The results on the PLANE dataset in Figure 6.2a show that for all numbers of complex Fourier coefficients $n > 5$ the classification accuracy is at least as good as the accuracy in the original 144-dimensional feature space. The obtained accuracies are robust with respect to n, as there are no large fluctuations in performance. For this particular dataset, a functional approximation with 15 or 20 complex Fourier coefficients already seems sufficient to accurately distinguish between the classes. The representation with concatenated Fourier coefficients of Scenario 3 achieves a similar accuracy as the complex representation. The performance for scenario 4, the smoothed time domain signals is highly similar to the performance obtained for the functional Fourier representations. The classification can already be done to high accuracy for a few Fourier coefficients. Hence, the functional approximations provide a highly effective dimensionality reduction for this classification problem.

In the results obtained for the FACESUCR dataset as shown in Figure 6.2b, the best performance is achieved for 20 Fourier coefficients. The performance in Fourier space is better than the performance in original space for $n = [15, 20, 25]$. For small number of coefficients, the performance quickly deteriorates due to the elimination of relevant frequency information. For a large number of coefficients, the performance decreases slowly, likely because the higher frequency information does not provide additional relevance for the classification and contains noise. The accuracy obtained for the smoothed time domain signals in scenario 4 is largely correlated and similar to the accuracy achieved for the functional Fourier approximation. This indicates that the increase in performance can mainly be attributed to the elimination of non-informative higher frequencies.

On the SYMBOLS dataset the functional Fourier representations always achieve a better performance than the baseline performance in the original 398-dimensional space, even with a number of coefficients as low as $n = 15$. The accuracies of the complex representation and the concatenated real representation of scenarios 2 and 3 are similar. In contrast to the results obtained for the other datasets, the accuracies achieved on the smoothed time series of Scenario 4 are systematically lower than the accuracies of the Fourier space representations and never exceed the accuracy of scenario 1. Hence, the performance increase observed for learning in Fourier space cannot be attributed to the smoothening of the time series. Instead, performing the training in Fourier space seems beneficial for the type of data in this classification problem.

For the further investigation of the performance of the method for even higher dimensional time series data, the dataset MALLAT is considered consisting of feature vectors with dimensions $N = 1024$. Figure 6.2d shows that the results in complex and concatenated Fourier space do not deviate significantly from the achieved accuracy in the original space. A functional Fourier approximation with 20 coefficients provides



(a) $\boldsymbol{w} \in \mathbb{R}^{144}$

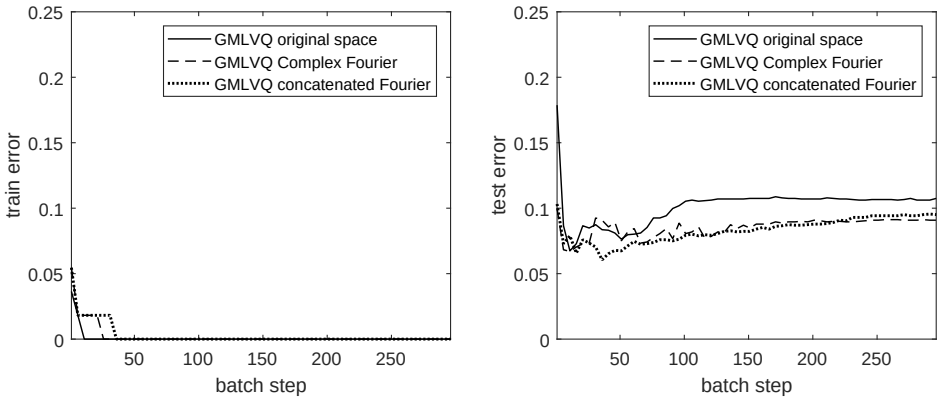
(b) $\hat{\boldsymbol{w}} \in \mathbb{R}^{144}$

(c) Feature relevance PLANE

(d) Back-transformed feature relevance of the PLANE dataset

Figure 6.3: In Figure 6.3a, the resulting class prototypes of the PLANE dataset are shown for training in the original 144-dimensional space. For clarity, only three of the seven prototypes are shown. The corresponding feature relevance values, which are the diagonal elements of the resulting relevance matrix for the PLANE dataset, are shown in Figure 6.3c. In Figure 6.3b the back-transformed prototypes obtained from training in 20-coefficient Fourier space are shown. Figure 6.3d shows the corresponding feature relevances, obtained from back-transforming the complex relevance matrix using the method discussed in Section 6.2.4.

similar classification accuracy as the accuracy obtained in the original space. Despite the result on this dataset showing no improvement in accuracy, the dimensionality in the classification problem was reduced by 98% without loss of classification accuracy, yielding a large computational advantage in the training- and classification stage.



(a) Train set error (b) Validation set error

Figure 6.4: Training and validation error for the MALLAT dataset in the course of training. The *dashed line* is the evolution of the error in 20-coefficient complex Fourier space. The *solid line* shows the error development in the original space of the data. The *dotted line* shows the error development in 20-coefficient concatenated Fourier space.

The prototypes that arise in the training process in complex Fourier coefficient space can be interpreted as class-specific contributions of the complex sinusoidal components of different frequencies in the corresponding classes. In Figure 6.3b, the back-transformation of the prototypes to the time domain as discussed in Sec. 6.2.4 was applied to the resulting complex-valued prototypes of the PLANE dataset in 21-coefficient Fourier space. A comparison with the prototypes resulting from training in the original time domain (Figure 6.3a) reveals that the back-transformed prototypes are smoother, but resemble the prototypes from training in the full original space closely. Correspondingly, Figure 6.3d shows the back-transformed relevance values. A comparison with the relevance values obtained in the original time domain shown in Figure 6.3c reveals that the two relevance profiles indicate similar regions of high relevance.

Figure 6.4 shows the error development computed on the training- and validation set of the MALLAT classification task in the original time domain(scenario 1) and in the 20-coefficient Fourier space for the representations of scenarios 2 and 3. The three methods all achieve zero training error before 50 training epochs. After 50 epochs the increased validation set error in the original space indicates an overfitting effect. Both Fourier representations, complex and concatenated real- and imaginary parts, are less affected by overfitting, as the error on the validation set for these representations

Table 6.2: Relations between absolute training time (in seconds) and dimensionality reduction.

Dataset	Original space	20-coefficient Fourier
MALLAT	2535	55
SYMBOLS	96	28

does not increase significantly. This confirms the conjecture that training in reduced Fourier coefficient space can help to alleviate overfitting effects. On this dataset, the complex Fourier representation seems to be affected the least by overfitting.

For all datasets that were considered, the difference in classification accuracy between the complex-valued Fourier representation and the concatenated Fourier representation was small. However, we believe that learning on complex-valued data directly, made possible by the adaptation of GMLVQ using Wirtinger calculus, is mathematically more sound since it treats a complex variable as one feature in the learning algorithm. Hence, it should be preferred over the concatenation of real- and imaginary parts.

Note that the number of adaptive parameters in GMLVQ scales as $O(N^2)$, where N is the number of features. Hence, the dimensionality reductions obtained here considerably reduce the computational effort in the training procedure. Table 6.2 contains the training times for the MALLAT and SYMBOLS classification tasks obtained by training the systems using a typical desktop computer on 20 Fourier coefficients and on the original data. Twenty Fourier coefficients is a dimension reduction of approx. 98% and 95% for the MALLAT and SYMBOLS datasets, respectively. The training times were reduced by approx. a factor 50 and 3.4, respectively.

6.5 Experiments: Learning in wavelet-space

In this part, we study the usefulness of the complex-valued extension of GMLVQ in combination with wavelet-space representations, by considering a heartbeat classification task from ECG data. The next sections describe the dataset, data preparation, feature extraction and the general training settings for the experiments.

6.5.1 Dataset and training set-up

We perform our experiments on the MIT-BIH Arrhythmia dataset (Moody and Mark 2001). The data was obtained from 4000 long-term Holter recordings (Moody

and Mark 2001). In total, 48 recordings selected from this set are available in the MIT-BIH database. Twenty-five recordings were selected based on a variety of anomalies and rare phenomena occurring in the heart rhythm and 23 recordings were chosen randomly from the total set. The signals were band-pass filtered using a passband from 0.1 to 100 Hz and then digitized with a sampling rate of 360 Hz. For each record, slightly over 30 minutes of ECG signal is selected. In principal, two leads are available for each recording. Usually the main lead is MLII, which is a modified limb lead that is obtained by the placement of the electrodes on the chest (Moody and Mark 2001).

A variety of works concerning the automatic classification of heartbeats exist in the literature. For instance, classification accuracies of 98% were achieved using a feed forward neural network and DTCWT features. However, in these settings the classifier is not validated on the data of new patients that were not in the training set. It is common to learn patient-specific classifiers, as was in (Ince* et al. 2009).

In contrast to the aforementioned studies, we did not include temporal features in the classification. Although these features are guided by expert knowledge and have great potential to improve classification accuracy, our aim is not to achieve top classification performance. In future research, the wavelet features should be combined with expert features to increase classification accuracy further.

Annotations

After the records had been selected and digitized, a simple QRS complex detector was applied on the signals (Moody and Mark 2001): The R-point is the central peak of the signal during a heartbeat, the Q-point is the dip that precedes the R-peak and the S-point the usually slightly larger dip after the R-peak. After the simple QRS complex detector was applied, two cardiologists independently annotated the detected heartbeats, indicating the heartbeat class for each beat. Additionally, annotations indicating heart rhythm, signal quality and additional comments are also available. The heart beat labels are denoted by symbols. The mapping from symbols to specific types of heart beats is found in (Moody and Mark 2001).

6.5.2 Data preparation and feature extraction

Using the heartbeat annotations, we extracted the heartbeats from the recorded ECG: For each annotated R-peak sample, a 128 length segment was extracted preceding the R-peak and a 127 length segment was extracted following the R-peak. Including the R-peak sample, this gives segments of $256 = 2^8$ samples in length. This length always segments the entire QRS complex including the preceding P wave and the succeeding T wave. A power of two was chosen for direct compatibility with the

DTCWT transform. The extracted segments are approximately $1/360 * 256 \approx 0.711$ seconds. Hence, for above average heart rates, segments may overlap.

From the segmented time domain heart beat vectors $\boldsymbol{x} \in \mathbb{R}^{256}$, we extracted wavelet features using the DTCWT up to level $j = 5$. For the first level, this gives $2^{8-1} = 2^7$ complex-valued detail coefficients, representing higher frequency wavelet correlations in the signal. The second level reduces the frequency window by a factor two and yields 2^6 complex-valued detail coefficients. This continues up to the highest level j, which yields 2^3 complex-valued detail coefficients and 2^3 complex-valued approximation coefficients. The approximation coefficients were obtained from the application of the low-pass filter at the highest level and therefore correspond to the lower frequencies in the signal. In summary, the transformation of the time domain signal $\boldsymbol{x} \in \mathbb{R}^{256}$ to wavelet-space produces feature vectors with $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^3 = 256$ complex-valued coefficients, $\boldsymbol{x}^{(w)} \in \mathbb{C}^{256}$.

6.5.3 Training settings and parameter values

In each experiment, the wavelet space feature vectors $\boldsymbol{x}^{(w)} \in \mathbb{C}^{256}$ obtained from a 5-level DTCWT as described in the previous section were considered. Additionally, we used truncated versions of the wavelet space feature vectors: $\hat{\boldsymbol{x}}^{(w)} \in \mathbb{C}^n$, where $n < 256$.

The wavelet space feature vectors were standardized using the z-score transformation and on the resulting vectors we applied the adapted complex-valued version of GMLVQ as explained in Sec. 6.2.3. In all training settings, one prototype per heartbeat class was used. The wavelet space prototypes $\boldsymbol{w}_i^{(w)} \in \mathbb{C}^n$ were initialized to small random deviations from the class-conditional means. The relevance matrix $\boldsymbol{\Lambda}^{(w)} \in \mathbb{C}^{n \times n}$ was initialized proportionally to the identity matrix. We used batch gradient descent along the lines of (Papari et al. 2011) in order to minimize the GMLVQ cost function from Equation 6.15, using the default parameters from (Biehl 2018).

6.6 Classification tasks

This section describes the specific experiment scenarios for studying the usefulness of the extension of GMLVQ in combination with wavelet representations for classifying heart beats.

6.6.1 General heartbeat classification

In this experiment we studied the classification performance of a classifier for recognizing heartbeats of multiple patients. The heartbeats for this experiment were

segmented from all available MIT-BIH records, hence spanning multiple patients. We considered the following classes of heartbeats for the classification: Normal beat (N), left bundle branch block beat (L), right bundle branch block beat (R), premature ventricular contraction (V) and paced beat (/). We performed the 5-level DTCWT on the labeled time domain beats and obtained wavelet space feature vectors $(\boldsymbol{x}_i^{(w)} \in \mathbb{C}^{256}, y_i)$, where y_i is a label from the set $C = \{N, L, R, V, /\}$. Subsequently, we randomly selected 100 examples from each class to be used as training data for the GMLVQ classifier. For validation, we randomly selected 150 other examples from each class. Sufficient training epochs were performed in order to let the GMLVQ cost function converge to the best performance obtained on the validation set.

The training on truncated wavelet space vectors $\boldsymbol{x}^{(w)} \in \mathbb{C}^{32}$ with 4th- and 5th level decomposition coefficients was performed and its validation performance was compared to the validation performance achieved on the full wavelet space representation. Note that as the number of parameters in GMLVQ increases quadratically with the number of input features, the GMLVQ model trained on only the 4th- and 5th-level coefficients has considerably less adaptive parameters. The training- and validation sets consisted of the same data points as were selected for the experiment on full wavelet space feature vectors.

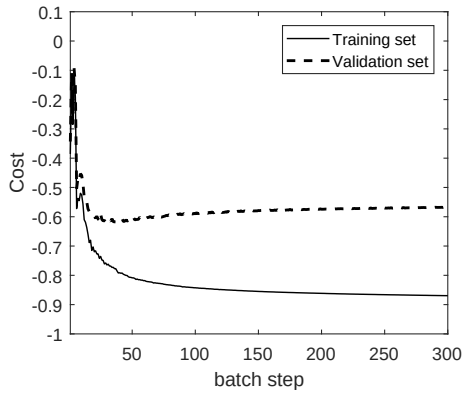
6.6.2 Patient-specific heartbeat classification

In the second experiment, we considered patient-specific classification. We followed a similar approach as in (Das and Ari 2014): A common training set from the MIT-BIH records 100 till 124 was selected and the patient-specific classification was performed on records 200 till 234. For each record in the latter group, the first 5 minutes of the record served as additional training data to the common heartbeats and the heartbeats occurring in the remaining 25 minutes, which the classifier had not seen during learning, was used for assessing the performance of the classifier.

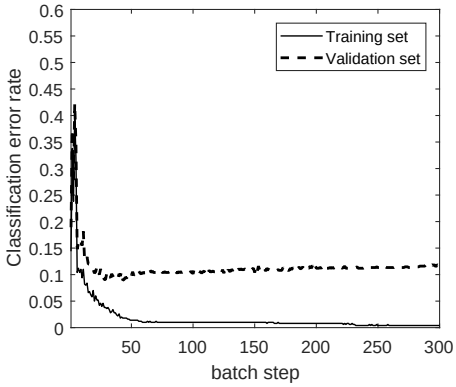
We trained on the full wavelet space vectors $\boldsymbol{x}^{(w)} \in \mathbb{C}^{256}$ obtained from the 5-level DTCWT. Additionally, training was performed in the patient-specific experiment using vectors containing only the 4th- and the 5th-level wavelet coefficients, $\boldsymbol{x}^{(w)} \in \mathbb{C}^{32}$.

6.7 Results and Discussion

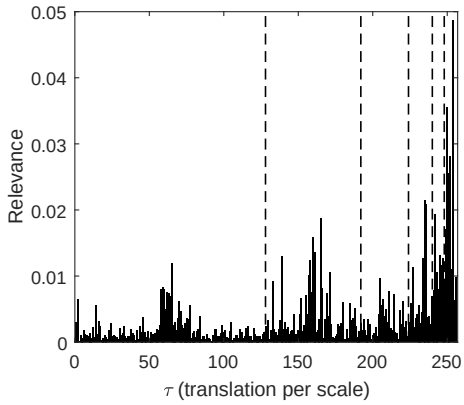
In this section, the results obtained from the experiments for general heartbeat classification and patient-specific heartbeat classification using the proposed methodology are shown and discussed.



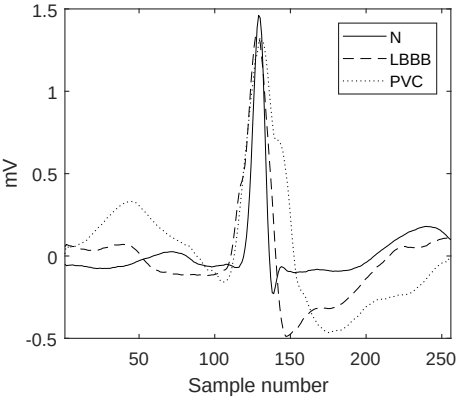
(a) Cost function vs. number of batch steps



(b) Classification error vs. number of batch steps



(c) Relevance wavelet features



(d) Three time domain prototypes learned in wavelet-space.

Figure 6.5: Wavelet space complex-valued GMLVQ training results for general heartbeat classification: training curves and classification error, relevance values and time-domain prototypes as back-transformed from wavelet-space.

6.7.1 General heartbeat classification

The results obtained from the first experiment are shown in Figure 6.5. Figure 6.5a and Figure 6.5b show the performance of the classifier throughout the learning process. In Figure 6.5a, the value of the cost function computed on the training data and computed on the validation data is shown for each learning step. The training set cost shows a stable converge towards a value of approximately -0.87. At the

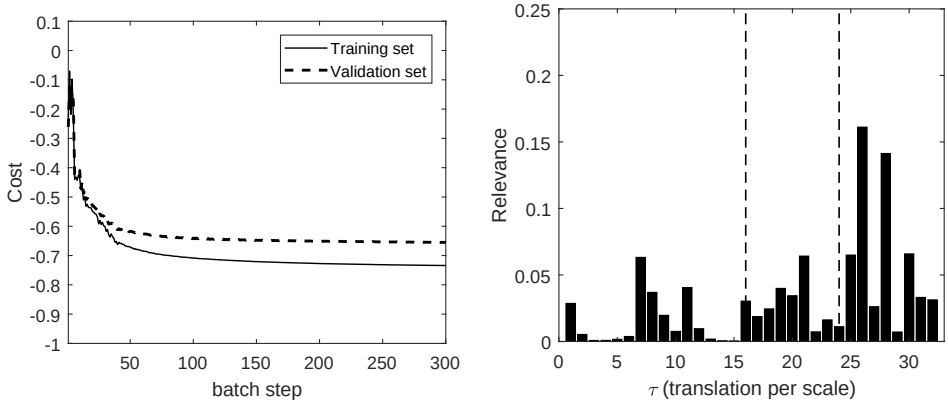
same time, the increase of the cost computed on the validation set shows signs of over-fitting, after its initial decrease. The lowest validation cost was approximately -0.62 and it was achieved at training epoch 38. After training epoch 300, the value of the validation cost increased to approximately -0.57. As expected, the classification error curves in Figure 6.5b are quite correlated with the cost-function curves. The classification error on the training set converged to approx. 0.004 (0.4%). The lowest achieved validation set classification error was approx. 0.089 (8.9%) in batch step 38, corresponding to the batch step of lowest validation cost. Due to the over-fitting, the validation set classification error increased after batch step 38. At batch step 300, the classification error was approx. 0.117 (11.7%). Finally, Figure 6.5c and Figure 6.5d show the interpretation of the classifier after 300 batch steps in terms of feature relevance values and back-transformed prototypes, respectively.

The feature relevance values that are shown in Figure 6.5c represent the importance of each wavelet coefficient in the classification problem. The highest values correspond to the most distinctive coefficients while low values correspond to features with which the classifier can not adequately differentiate between the classes. The dashed vertical lines mark the transition between different scales of the wavelet decomposition. Hence, the first dashed line appears at feature 128, indicating the border between the first level detail-coefficients of the higher frequencies and the second level detail-coefficients of the frequencies below. Inside each wavelet scale window, the horizontal axis indicates the translation τ of the wavelet. Due to the time-localization of the wavelets, relevance values are interpretable both in frequency/scale and time. For relevance in time, it can be seen that for most scales the highest relevance was found around the center of the segment in the region of the QRS-complex. For relevance in scale, wavelet coefficients of the second, fourth and fifth level of the decomposition were found to be highly discriminative.

In Figure 6.5d, the back-transformed time domain prototype is shown for the Normal, Left Bundle Branch Block and Premature Ventricular Contraction class. This allows the time domain interpretation of what the classifier learned as typical examples of the different beats in the classification problem.

Figure 6.6 shows the results of GMLVQ training on the truncated wavelet space feature vectors. From the training set and validation set cost curves as shown in Figure 6.6a, it can be seen that in this case there was no over-fitting effect. The final validation set cost was (-0.66), which is lower than the minimum cost achieved for GMLVQ learning on the full wavelet space vectors shown in Figure 6.5.

In Figure 6.6b, the relevance of the 4th- and 5th-level wavelet space coefficients is shown. The first window shows the 16 4th-level detail coefficients, the second window the eight 5th-level detail coefficients and the third window the eight 5th-level approximation coefficients. The most discriminative coefficient is an approximation



(a) Cost function vs. number of batch steps

(b) Relevance wavelet features

Figure 6.6: Wavelet space complex-valued GMLVQ results of general heartbeat classification where only the wavelet decomposition coefficients of the 4th- and 5th levels were used: training curves and relevance values.

Table 6.3: The class-wise validation set accuracy for GMLVQ trained on the truncated and full wavelet coefficient vectors in the general heartbeat classification task.

	Truncated	Full
Normal beat (N)	97.3%	85.0%
Left bundle branch block beat (L)	96.7%	95.9%
Right bundle branch block beat (R)	98.7%	94.9%
Premature ventricular contraction (V)	60.5%	68.8%
Paced beat (/)	97.0%	96.2%

coefficient of the fifth level. Higher values are obtained for wavelets with an activation corresponding to the center of the signal: A peak occurs around the center of the 4th- and 5th level detail coefficients, indicating a relevance of the wavelets active in the QRS-complex region on the two scales.

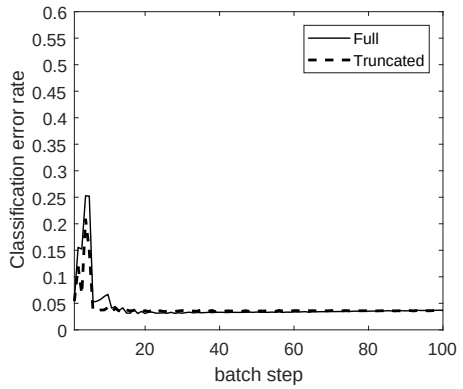
Table 6.3 shows that GMLVQ training on the truncated wavelet space vectors achieved a higher validation set accuracy than training on the full wavelet space vectors for all heartbeat classes except Premature ventricular contraction (V). As the classification of normal heartbeats was much less accurate in full wavelet space, this caused quite a substantial number of false positive anomalous heart beat detections in this case.

Although the GMLVQ classifier trained on the full wavelet space feature vectors has the advantage of full wavelet space interpretation and time domain prototype interpretation, only considering the 4th- and 5th level wavelet coefficients already seems to provide enough information to adequately discriminate between the classes and is substantially cheaper to train. On the other hand, provided with all wavelet coefficients, given enough training data and an appropriate training strategy including early stopping and regularization, the GMLVQ algorithm should be able to learn the most relevant features by adapting the relevance matrix. When such training is successful, the performance should be at least as good as the truncated versions. This can also be used to learn the most relevant features as a pre-processing step and then use GMLVQ classification trained on the most relevant wavelet coefficients.

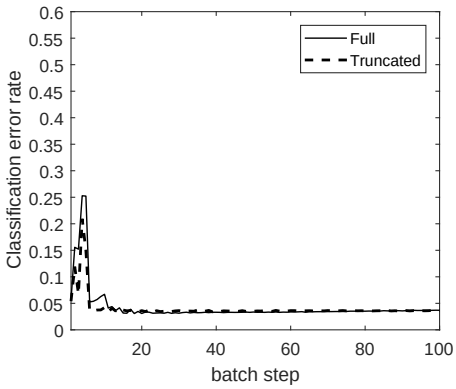
6.7.2 Patient-specific heartbeat classification

Patient-specific learning was applied for the last 25 records in the database as explained in Sec. 6.6.2. In Figure 6.7a, the average validation cost over the 25 patient-specific learning curves is shown for GMLVQ learning on both the full wavelet space vectors and the truncated vectors. Similar to the general heartbeat classification task, learning on the full wavelet space vectors was more prone to over-fitting, whereas over-fitting did not occur when using the truncated wavelet space vectors. The cost after batch step 100 was approx. -0.83 for learning on the truncated vectors; Perhaps not surprisingly, patient-specific classifiers were on average more accurate than the general heartbeat classifiers. The average validation set classification error for each batch step is shown in Figure 6.7b. In both scenarios, training quickly resulted in a validation error below 5%. In the classification scenario using truncated wavelet coefficients, the final error was approximately 3.6% whereas for the full wavelet space scenario, the final error was 3.7%.

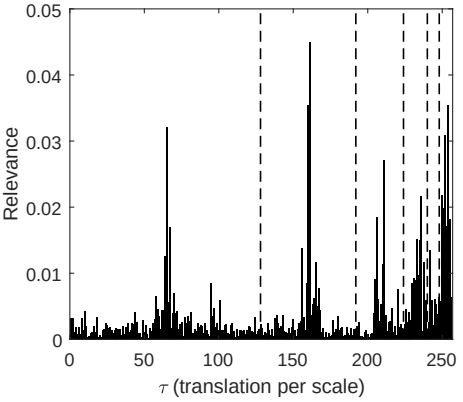
Figure 6.7c and Figure 6.7d show the interpretation of the resulting patient-specific classifier for record 217. This record belongs to a patient that experienced premature ventricular contractions and paced beats, besides the great majority of normal beats. The relevance profile in Figure 6.7c shows highly pronounced peaks around the center of each scale, the coefficients of the wavelets that were active in this area were much more informative for the classification than wavelets in other areas of the segment. The relative difference between these relevance values was significantly larger than the relevance values of the general heartbeat classifier, shown in Figure 6.5c. One potential reason for this is that this patient-specific classification problem has fewer classes. In the general heartbeat classification, the distinction between the classes was made using additional relevant features, so that the relevance values became more spread out. Figure 6.7d shows the time domain representation of the



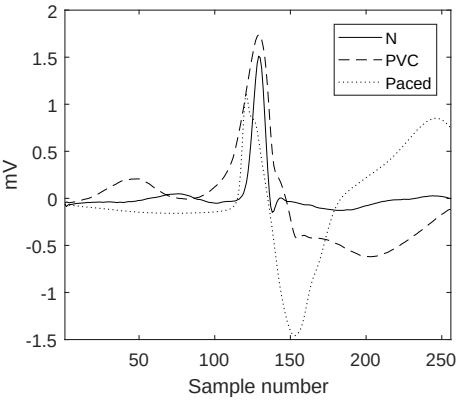
(a) Average validation cost vs. number of batch steps over all validation patients.



(b) Average classification error vs. number of batch steps over all validation patients.



(c) Relevance of wavelet features in patient-specific GMLVQ learning on MIT-BIH record 217.



(d) Time domain prototypes learned in wavelet-space for MIT-BIH record 217.

Figure 6.7: Wavelet space complex-valued GMLVQ results for the patient-specific classification task: training curves, classification error, relevance values and time-domain prototypes as back-transformed from wavelet-space.

back-transformed prototypes which were learned in wavelet space for the three heartbeat types that occur in this record.

In Table 6.4, the average class-wise validation set classification accuracy among the 25 patient-specific classifiers is shown for the two scenarios. The classification accuracies for patient-specific classifiers were better than the accuracies obtained for the general heartbeat classifier. Most notably, the normal beat classification accuracy

Table 6.4: Average class-wise validation accuracy in patient-specific learning on MIT-BIH records 200 till 234 for both truncated and the full wavelet coefficient vectors.

	Truncated	Full
Normal beat (N)	98.7%	98.6%
Left bundle branch block beat (L)	98.3%	95.9%
Right bundle branch block beat (R)	99.6%	97.0%
Premature ventricular contraction (V)	75.4%	79.1%
Paced beat (/)	99.8%	99.6%

of the classifier that used the full wavelet representation was much better than the accuracy obtained with the general classifier, which reduces the false positive rate. Both the truncated and full wavelet space patient-specific classifiers achieved a significantly better classification accuracy for premature ventricular contractions.

6.8 Summary and Outlook

In this contribution we first discussed two popular tools in signal processing, the DFT and DWT, that provide a more natural and powerful representation of signals with periodicities and certain shape characteristics. Using these representations would be useful for improving the accuracy in classification tasks of these functional data. The standard formulations of the DFT and the DTCWT yields vectors of complex-valued coefficients. Although there are obvious ways of representing the complex-valued dimensions with real dimensions, such as treating the imaginary parts as separate real dimensions or using the amplitude and phase representation of a complex number, we discussed a version of GMLVQ that naturally handles the data in the complex-valued domain. We then used this version of GMLVQ with the combination of the DFT and DTCWT for several classification tasks of functional data.

In the experiments where the data was transformed using the DFT, the classification performance for a reasonably small number of coefficients ($n = 20$) was similar or better than the classification accuracy on the dataset in the original time domain, for all considered classification tasks. In these cases, our methodology provided a powerful dimensionality reduction: For the MALLAT classification task, the number of dimensions was reduced by 98%, while retaining similar classification accuracy. A robust increase in classification accuracy in the Fourier domain was observed for the SYMBOLS classification task. In this case, the classification based on the data in the Fourier domain was significantly and consistently better than the classification based

on the equivalent smoothed data in the time domain. Although in our classification tasks, the representation that treats the imaginary parts as additional real dimensions showed highly similar performance as the complex-valued representation, we believe that the more natural complex-valued GMLVQ should be preferred, due to the explicit treatment of complex-valued dimensions and complex-valued relevance values. It is a future topic of investigation to understand under which type of conditions the complex version, with its complex-valued matrix of relevance values, performs better than the real-valued workarounds. An additional benefit of the methodology is that the classifier trained in the space of the transform is relatively easily back-transformed to the time domain. Hence, the interpretability of the classification task is increased significantly, because the method provides prototypes and relevance information from both the time domain and the transformation perspective, while still keeping the benefits of training in reduced coefficient space.

Similar observations were made for heartbeat classification using wavelet features. For the general heartbeat classifier we obtained a better classification performance when training on only the 4th- and 5th level coefficients than when training on the full wavelet space representation. Especially the number of false positives, which is the incorrect classification of a normal beat as one of the anomalous beats, was significantly reduced by training on the truncated feature vectors. For all wavelet scales, the diagonal of the relevance matrix had the largest values at the center of the segment. The classification performance was better in the patient-specific classification tasks, most notably the classification performance of premature ventricular contractions was improved significantly. It is likely that there is a patient-specific component to the premature ventricular contractions and the other heartbeat types that could explain the need for patient-specific classification. There could also be several characteristic shapes associated with each heartbeat class, which could be addressed by using more than one prototype per class in GMLVQ. However, it should be noted that obtaining the best classification performance was beyond the scope of this work. For future work it should be investigated whether the combination of complex-valued wavelet coefficients and in combination with additional features guided by expert knowledge, as well as hyperparameter tuning in GMLVQ, can improve the classification results in both the general and patient-specific classification. These results should then also be compared with results in the literature.

The optimal number of coefficients is dependent on the properties of the dataset and the classification problem. For future study, an automatic method could be devised that suggests a number of coefficients based on the available training and validation data according to a criterion of optimality, which seeks the best balance between accuracy and the number of coefficients.

In conclusion, our work demonstrates that the combination of appropriate data

transforms with GMLVQ constitutes a versatile framework which offers the potential to improve significantly the performance, computational workload and interpretability of prototype-based relevance learning.

Chapter 7

Conclusions and Future Work

In this thesis we considered model scenarios of machine learning in Part I and we addressed applications in smart industry and relevant methodology concerning time series classification in Part II.

To contribute to the need of an increased theoretical understanding of the use of Rectified Linear Unit (ReLU) activation in artificial neural networks, in our theoretical investigations the central aim was to reveal learning characteristics of neural networks with ReLU activation function. In particular, we aimed at a comparison with the learning characteristics of traditional sigmoidal activation in a variety of highly relevant model scenarios of machine learning. In order to address this aim, techniques from statistical physics were employed to obtain exact results that describe the typical learning behaviour in three main learning scenarios: on-line learning from a stream of independently generated examples (Chapter 2), learning under concept drift (Chapter 3) and off-line learning from a fixed set of examples (Chapter 4). In the analysis of learning under concept drift, our additional aim was to characterize the learning behaviour of the Learning Vector Quantization (LVQ) model in such situations.

We established a generic on-line learning modelling framework in Chapter 2 suitable for the formulation of on-line learning for a variety of learning rules and models in student-teacher settings. The formulation consists of a system of differential equations that describes the exact evolution of high-level order parameters of the model in the limit of the number of input dimensions to infinity. Using the framework, we formulated the system of differential equations describing typical gradient descent learning dynamics of two-layer neural networks with ReLU activation in the hidden units in three learning settings: matching student and teacher complexity, overparameterized student networks and unlearnable target rules. Numerically integrating the obtained system of differential equations and also the equations for gradient descent learning in sigmoidal networks obtained from the literature yielded the typical results for both cases. This allowed a comparison of ReLU and sigmoidal networks with regards to their typical learning behaviour. For the overparameterized setting, it was shown that the learning algorithm can combine ReLU units in order to limit the number of effective parameters, which is possible due to the activation's

piecewise linearity. In contrast, in overparameterized sigmoidal student networks the number of effective parameters is mainly reduced by removing units from the network through convergence of the weight vectors of superfluous units to zero.

In Chapter 3 we introduced concept drift to the modelling framework. In the neural network settings, we studied real drift by including a random drift of the teacher vectors. In addition, we also modelled the effect of weight decay. Significant differences between ReLU- and sigmoidal networks were found concerning their sensitivity to the drift and the effectiveness of weight decay in these situations. For both ReLU and sigmoidal networks the characteristic plateau state is prolonged by the introduction of the drift, delaying the onset of hidden unit specialization. The analysis of sigmoidal networks showed that weight decay can improve the generalization performance under concept drift. However, the network is highly sensitive to the setting of the weight decay parameter. Values slightly different from the optimal deteriorate the network's capacity for hidden unit specialization. In contrast, for ReLU networks, besides also showing improvement in generalization error by introducing weight decay, the analysis showed that the network is robust to the setting of the weight decay parameter; hidden unit specialization is possible for a wide range of values. Moreover, the analysis of the properties of the unspecialized plateau state indicated that introducing weight decay accelerates hidden unit specialization for ReLU networks for a wide range of parameter values. In contrast, for sigmoidal networks the introduction of weight decay delays the onset of specialization. Hence, we observed that for ReLU networks the length of the plateau is shortened significantly by introducing weight decay while the plateau is prolonged in case of sigmoidal networks.

Chapter 3 also addressed the LVQ machine learning model which was studied in the modelling framework for two concept drift situations: a random drift of cluster centres of the input density and a drift of class biases in the input stream. Our analyses showed that LVQ learning is able to deal with the considered drifts to some extent by default. It was demonstrated that the drift of cluster centres impedes the learning process and the dependence of the lowest achievable generalization error with respect to the learning rate was shown to be non-trivial. Furthermore, we demonstrated that in this setting weight decay can improve the generalization performance significantly. In contrast, in the case of drifting class biases, weight decay is not suitable. In this case, the restrictive effect of weight decay on the norms of the prototypes limits the flexibility of the system and hence decreases its ability to react appropriately to changing class biases.

In Chapter 4 we analysed and compared off-line learning in ReLU- and sigmoidal neural networks. The model considers a canonical ensemble of networks in formal thermal equilibrium. Here the density of states is given by the Gibbs distribution

which is characterized by the formal temperature. The quenched free energy, that involves an additional average over datasets of fixed size, determines typical properties of a stochastic optimization process for the energy of the system. Since the data average is technically involved, we considered the simplifying limit of high temperature, which had already been exploited in previous works for obtaining useful qualitative insights into learning scenarios. In this case the free energy is a combination of the generalization error and the system's entropy. For increasing dataset size, minimization of the free energies corresponding to ReLU and sigmoidal networks revealed highly interesting differences: most importantly, in case of networks with a number of hidden units $K > 2$, the phase transition associated with hidden unit specialization in ReLU networks is second order. In contrast, previous research found a first order transition in case of sigmoidal networks. After the continuous phase transition in ReLU networks, two minima of the free energy arise that have very similar generalization performance. For the number of hidden units $K \to \infty$, we showed that the generalization performance corresponding to these minima is identical. In contrast, for sigmoidal networks the competing state after the first order phase transition has significantly worse performance compared to the specialized state.

The second part of this thesis addressed computational intelligence applications in smart industry. In Chapter 5 we addressed the need for continuous quality control in steel-based manufacturing by performing a typical Industry 4.0 case study on a particular mass production line in collaboration with industry. This high-throughput production line contains a stamping press as the main machinery that produces the steel-based products. The aim was to develop a real-time material property identification and quality control system based on sensor measurements of Eddy Current. Secondly, we studied whether such a system could be helpful in the prevention of production faults. The dataset consisted of sensor measurements of previously measured strip steel that was used for production and the results of corresponding destructive tensile tests. Various analyses of the data using Principal Component Analysis showed that the variation of the material properties was included in the sensor data and further exploratory data analysis revealed linear correlations between the sensor measurements and the material properties. Furthermore, the Eddy Current input variables were mutually highly correlated. Hence, we fitted a Partial Least Squares model to the data and showed that the model estimations of the material properties can prevent material that does not conform to the specifications from entering the press in real production scenarios. Furthermore, using Eddy Current sensor data measured on previous production days in combination with logs of product faults, we demonstrated that the model estimations provide a risk value of the occurrence of product faults.

The frequently arising problem of time series classification was addressed in Chapter 6. Here, we proposed the idea of using complex-valued representations of time series as obtained by the complex-valued Fourier or wavelet transform in combination with the interpretable Generalized Matrix Learning Vector Quantization (GMLVQ) machine learning model. This approach adapts relevance values and prototypes in the space of the transform, which can be highly beneficial for functional time series data, especially when the data exhibits properties that are appropriate for the considered transform. In addition, a back-transformation of the relevance matrix was formulated, which yields time domain feature relevance values besides the transform-domain feature relevance values, broadening the interpretability of the classifier. Due to the appropriateness of the representations of the considered time series datasets, the approach always showed better classification performance compared to classification in the time domain. Moreover, large dimensionality reductions could be achieved in the classification problems which mitigated overfitting and improved training efficiency. In a larger experiment, we considered the classification of heart beats from ECG data by using complex-valued wavelet coefficients. In this case, it was shown that the dimensionality could be reduced significantly while maintaining classification performance. In summary, for a variety of situations we demonstrated that the method is highly efficient as a dimensionality reduction technique, maintains good classification performance and yields a broad interpretation of the classifier.

7.1 Future works

Here we emphasize a few of the possible directions for future research.

The learning behaviour of other activation functions could be studied using the promising tools from the statistical physics of learning. In the analysis of off-line learning, recent work investigated several activation functions and the corresponding networks' type of phase transition to specialized hidden units (Oostwal 2020): networks with leaky ReLU activation exhibit a continuous phase transition while networks with Piecewise Linear Unit activation have a first order phase transition. Moreover, in our recent initial investigations of the GELU type of activation, we found that the phase transition remains continuous for all arguments γ of Eq. (2.46). The GELU function is highly similar to the Swish function, which has seen a rise in popularity recently due to empirical success in applications. We provide the derivation of the generalization error for GELU activation in Eq. (A.8). Furthermore, it is still an open question which properties of the activation function determine the type of phase transition to specialized hidden units.

In the on-line learning setting, the effect of the GELU and other activation functions could be studied in concept drift situations together with methods such as weight decay. It should be noted that the statistical physics of learning approaches can also be used to design new activation functions with favourable properties, such as a combination of a second order phase transition towards specialized hidden units as we showed for the ReLU, fast escape from the plateau states in on-line learning and a favourable interplay with concept drift and weight decay. In the concept drift scenarios, we studied neural network student teacher settings of matching complexity. In future works, a particularly interesting research direction is the study of concept drift situations in settings with an overparameterized student.

Regarding the comparison of activation functions, a limitation of the approach is that the results are not directly quantitatively comparable. Quantitative comparisons between activation functions such as plateau lengths or the location of phase transitions cannot be readily made, because the teacher differs non-trivially in complexity in both cases. A possible idea to allow quantitative comparisons is to define student teacher scenarios with mismatched activation between student and teacher. One could then compare two activation functions that learn a target rule defined by a teacher that uses a third activation function. More extensive overparameterized settings possibly in combination with the idea of mismatched activations should provide additional highly relevant modelling scenarios. The extension of the methods to more hidden layers is another relevant direction for future research. Our initial work has focused on tree-like structures. First results indicate a first order phase transition towards specialized hidden layers, but further investigation is necessary.

For the material fault detection, a next step is to validate the model on more data from coils that exhibit diverse material properties. This may require the deliberate modification of material to desired properties in order to increase the confidence in the relationship between the material properties and sensor measurements. Secondly, the current specification limits of the material properties could be re-evaluated guided by a cost analysis to efficiently adjust the fault classification. Lastly, the incorporation of production parameters, machine settings and product quality measurements in the model could be used for the optimization of the production settings with respect to the specifics of the material in order to obtain the highest quality products and the least number of production faults.

In the evaluation of time series classification using complex-valued coefficients, we studied the benefits of the method in terms of its interpretability, dimensionality reduction and classification performance. In future work, one possible interesting research direction is to include expert features in the coefficient vectors and compare the resulting classification performance to the results found in the literature. For instance, the wavelet features of the heart beats should be complemented by expert

features. Cross-validation can then be applied to find optimal hyper parameters for GMLVQ, and the top results should then be compared to the performance found in the literature for other automatic classification approaches.

Appendix A

Appendix

A.1 Covariance matrix and order parameters for the SCM

In this section we provide the full covariance matrix for the student-teacher settings of the SCM, which are studied in Chapters 2, 3 and 4. In particular, it is applicable to the mathematical analysis of the on-line training dynamics in Sec. 2.2.4, the student teacher scenario given in Sec. 4.2 and the definition of the entropy in Sec. 4.2.4, Eq. (4.19). The covariance matrix is defined in terms of the order parameters, which are also defined here.

For the SCM, the $(K+M)\times(K+M)$-dim. matrix of order parameters reads

$$\mathcal{C} = \begin{bmatrix} T & R \\ R^\top & Q \end{bmatrix} \text{ with submatrices } \begin{cases} T \in \mathbb{R}^{M\times M} \\ R \in \mathbb{R}^{K\times M} \\ Q \in \mathbb{R}^{K\times K} \end{cases} \tag{A.1}$$

that consist of the elements

$$T_{ij} = \langle b_i b_j \rangle = \mathbf{B}_i \cdot \mathbf{B}_j \,, \quad R_{ij} = \langle h_i b_j \rangle = \mathbf{w}_i \cdot \mathbf{B}_j \,, \quad Q_{ij} = \langle h_i h_j \rangle = \mathbf{w}_i \cdot \mathbf{w}_j \,.$$

Note that an additional pre-factor of $1/N$ is applied in the definitions of the order parameters in Chapter 4.

Furthermore, note that Eqs. (4.20) and (4.21) correspond to the special case of $K = M$ and exploit site-symmetry (4.12) and normalization (4.8).

A.2 Derivation of the generalization error of the SCM

Here we give a derivation of the generalization error in terms of the order parameters for sigmoidal, ReLU and GELU student and teacher. The equation for the generalization error in Eq. (2.28) and Eq. (4.11) is first used in the on-line learning analysis in Chapter 2 to obtain the learning curves of the generalization error by substitution of the solutions of the order parameters obtained by solving the ODE (2.23) numerically.

It is also used in the analysis of concept drift in Chapter 3. In the analysis of off-line learning in the high temperature limit in Chapter 4, the first term of the free energy function (4.22) contains the generalization error.

For general K and M we start by completing the square in the definition (2.28), obtaining:

$$\epsilon_g = \frac{1}{2}\left(\sum_{i,j=1}^{K} \langle g(h_i)g(h_j)\rangle - 2\sum_{i=1}^{K}\sum_{j=1}^{M}\langle g(h_i)g(b_j)\rangle + \sum_{i,j=1}^{M} \langle g(b_i)g(b_j)\rangle \right). \tag{A.2}$$

The above form reduces to Eq. (4.11) for $K = M$, where an additional pre-factor of $1/K$ is applied due to the scaling of the student and teacher by $1/\sqrt{K}$ in that model. To obtain ϵ_g for a particular choice of activation function g, expectation values of the form $\langle g(x)g(y)\rangle$ have to be evaluated over the joint Gaussian density of the hidden unit local potentials x and y, i.e. $P(x, y) = \mathcal{N}(\mathbf{0}, \widehat{\mathcal{C}})$ with the appropriate submatrix $\widehat{\mathcal{C}}$ of $\mathcal{C}$, cf. Eq. (A.1):

$$\widehat{\mathcal{C}} = \begin{bmatrix} \langle y^2\rangle & \langle xy\rangle \\ \langle xy\rangle & \langle x^2\rangle \end{bmatrix}.$$

In the resulting equations for the generalization error (A.3), (A.5) and (A.8) for the Erf, ReLU and GELU activation functions respectively, the additional pre-factor $1/K$ has to be applied to obtained the generalization error for the student teacher setting in the off-line analysis of Chapter 4.

A.2.1 Sigmoidal

For student and teacher with sigmoidal activation functions $g(x) = \mathrm{erf}\big[x/\sqrt{2}\big]$ or the shifted version $g(x) = \big(1 + \mathrm{erf}[x/\sqrt{2}]\big)$ that is used in Chapter 4, the generalization error has been derived in (Saad and Solla 1995b) and is given by:

$$\begin{aligned}\epsilon_g = \frac{1}{\pi}\Bigg\{ &\sum_{i,j=1}^{K} \sin^{-1}\frac{Q_{ij}}{\sqrt{1+Q_{ii}}\sqrt{1+Q_{jj}}} + \sum_{n,m=1}^{M} \sin^{-1}\frac{T_{nm}}{\sqrt{1+T_{nn}}\sqrt{1+T_{mm}}} \\ &- 2\sum_{i=1}^{K}\sum_{j=1}^{M} \sin^{-1}\frac{R_{ij}}{\sqrt{1+Q_{ii}}\sqrt{1+T_{jj}}}\Bigg\}.\end{aligned} \tag{A.3}$$

A.2.2 ReLU

For student and teacher with ReLU activations $g(x) = \max\{0, x\}$, applying the elegant formulation used in (Yoshida et al. 2017) gives an analytic expression for the

two-dimensional integrals:

$$\begin{aligned} I_2 = \langle g(x)g(y)\rangle = \langle \max\{0,x\}\max\{0,y\}\rangle &= \int_0^\infty\int_0^\infty xy\mathcal{N}(\mathbf{0},\widehat{\mathcal{C}})dxdy \\ &= \frac{\widehat{\mathcal{C}}_{12}}{4} + \frac{\sqrt{\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{22}-\widehat{\mathcal{C}}_{12}^2}}{2\pi} + \frac{\widehat{\mathcal{C}}_{12}}{2\pi}\sin^{-1}\left(\frac{\widehat{\mathcal{C}}_{12}}{\sqrt{\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{22}}}\right). \end{aligned} \tag{A.4}$$

Substituting the result from Eq. (A.4) in Eq. (A.2) for the corresponding covariance matrices gives the analytic expression for the generalization error in terms of the order parameters:

$$\begin{aligned} \epsilon_g = &\frac{1}{2}\sum_{i,j=1}^{K}\left(\frac{Q_{ij}}{4} + \frac{\sqrt{Q_{ii}Q_{jj}-Q_{ij}^2}+Q_{ij}\sin^{-1}\left[\frac{Q_{ij}}{\sqrt{Q_{ii}Q_{jj}}}\right]}{2\pi}\right) \\ &- \sum_{i=1}^{K}\sum_{j=1}^{M}\left(\frac{R_{ij}}{4} + \frac{\sqrt{Q_{ii}T_{jj}-R_{ij}^2}+R_{ij}\sin^{-1}\left[\frac{R_{ij}}{\sqrt{Q_{ii}T_{jj}}}\right]}{2\pi}\right) \\ &+ \frac{1}{2}\sum_{i,j=1}^{M}\left(\frac{T_{ij}}{4} + \frac{\sqrt{T_{ii}T_{jj}-T_{ij}^2}+T_{ij}\sin^{-1}\left[\frac{T_{ij}}{\sqrt{T_{ii}T_{jj}}}\right]}{2\pi}\right). \end{aligned} \tag{A.5}$$

A.2.3 GELU

For student and teacher with GELU activation, and using potentially different scale factors γ and κ in the sigmoidal component of student and teacher respectively, we derive the two-dimensional integrals:

$$\begin{aligned} I_2 = \langle g(x)g(y)\rangle &= \left\langle \frac{1}{4}xy\left(1+\operatorname{erf}\left(\frac{\gamma x}{\sqrt{2}}\right)\right)\left(1+\operatorname{erf}\left(\frac{\kappa y}{\sqrt{2}}\right)\right)\right\rangle \\ &= \frac{1}{4}\left(\langle xy\rangle + \langle xy\operatorname{erf}\left(\frac{\gamma x}{\sqrt{2}}\right)\operatorname{erf}\left(\frac{\kappa y}{\sqrt{2}}\right)\rangle + \underbrace{\langle xy\operatorname{erf}\left(\frac{\kappa y}{\sqrt{2}}\right)\rangle}_{0} + \underbrace{\langle xy\operatorname{erf}\left(\frac{\gamma x}{\sqrt{2}}\right)\rangle}_{0}\right). \end{aligned} \tag{A.6}$$

The integrands of the last two terms are anti-symmetric and therefore evaluate to zero. We continue with the integration of the remaining two terms:

$$\begin{aligned}
&\frac{1}{4}\left(\langle xy\rangle + \langle xy\,\mathrm{erf}\left(\frac{\gamma x}{\sqrt{2}}\right)\mathrm{erf}\left(\frac{\kappa y}{\sqrt{2}}\right)\rangle\right) \\
&= \frac{1}{4}\left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xy\mathcal{N}(\mathbf{0},\widehat{\mathcal{C}})dxdy + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xy\,\mathrm{erf}\left(\frac{\gamma x}{\sqrt{2}}\right)\mathrm{erf}\left(\frac{\kappa y}{\sqrt{2}}\right)\mathcal{N}(\mathbf{0},\widehat{\mathcal{C}})dxdy\right) \\
&= \frac{\widehat{\mathcal{C}}_{12}}{4} + \frac{1}{4}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xy\,\mathrm{erf}\left(\frac{\gamma x}{\sqrt{2}}\right)\mathrm{erf}\left(\frac{\kappa y}{\sqrt{2}}\right)\mathcal{N}(\mathbf{0},\widehat{\mathcal{C}})dxdy\,.
\end{aligned} \tag{A.7}$$

The last integration can for instance be performed by integrating with respect to x using Eq. (A.14). The final integrand is of the form (A.17). After standard re-writings, simplifications and substitution in Eq. A.7 above, we obtain the solution:

$$\begin{aligned}
I_2 = &\frac{\widehat{\mathcal{C}}_{12}}{4} + \frac{\widehat{\mathcal{C}}_{12}}{2\pi}\sin^{-1}\left(\frac{\gamma\kappa\widehat{\mathcal{C}}_{12}}{\sqrt{1+\gamma^2\widehat{\mathcal{C}}_{11}}\sqrt{1+\kappa^2\widehat{\mathcal{C}}_{22}}}\right) \\
&+ \frac{\gamma^3\kappa^3\widehat{\mathcal{C}}_{11}\left(\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{22}-\widehat{\mathcal{C}}_{12}^2\right)\widehat{\mathcal{C}}_{22} + \gamma\kappa\left(\gamma^2\widehat{\mathcal{C}}_{22}\widehat{\mathcal{C}}_{11}^2 + \kappa^2\widehat{\mathcal{C}}_{22}^2\widehat{\mathcal{C}}_{11} + \widehat{\mathcal{C}}_{22}\widehat{\mathcal{C}}_{11} + \widehat{\mathcal{C}}_{12}^2\right)}{2\pi\left(\gamma^2\widehat{\mathcal{C}}_{11}+1\right)\left(\kappa^2\widehat{\mathcal{C}}_{22}+1\right)\sqrt{\gamma^2\kappa^2\left(\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{22}-\widehat{\mathcal{C}}_{12}^2\right)+\gamma^2\widehat{\mathcal{C}}_{11}+\kappa^2\widehat{\mathcal{C}}_{22}+1}}\,.
\end{aligned} \tag{A.8}$$

Note that the GELU for the scaling factor in the sigmoidal $\gamma \to \infty$ is the ReLU as shown in Eq. (2.46). The above solution in the limits $\gamma \to \infty$ and $\kappa \to \infty$ reads:

$$\lim_{\gamma,\kappa\to\infty} I_2 = \frac{\widehat{\mathcal{C}}_{12}}{4} + \frac{\sqrt{\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{22}-\widehat{\mathcal{C}}_{12}^2}}{2\pi} + \frac{\widehat{\mathcal{C}}_{12}}{2\pi}\sin^{-1}\left(\frac{\widehat{\mathcal{C}}_{12}}{\sqrt{\widehat{\mathcal{C}}_{11}\widehat{\mathcal{C}}_{22}}}\right), \tag{A.9}$$

which is indeed equal to the solution found for I_2 in case of ReLU activation in Eq. (A.4).

A.3 Table of integrals

This section contains a table of integrals of frequently occurring forms in Gaussian integration. Some of these forms were used in the derivation of the integrals in Sec. A.2. The forms are also provided for future reference. In all integrations it is assumed that the real-valued coefficient $a > 0$.

$$\int_{-\infty}^{\infty} x e^{-\frac{1}{2}ax^2+bx} dx = \frac{\sqrt{2\pi} b e^{\frac{b^2}{2a}}}{a^{3/2}} . \tag{A.10}$$

$$\int_{-\infty}^{\infty} x^2 e^{-\frac{1}{2}ax^2+bx} dx = \frac{\sqrt{2\pi}\left(a+b^2\right) e^{\frac{b^2}{2a}}}{a^{5/2}} . \tag{A.11}$$

$$\int_{-\infty}^{\infty} e^{-\frac{1}{2}ax^2+bx} \operatorname{erf}(cx) dx = \sqrt{\frac{2\pi}{a}} e^{\frac{b^2}{2a}} \operatorname{erf}\left(\frac{bc}{\sqrt{a\left(a+2c^2\right)}}\right) . \tag{A.12}$$

$$\int_{-\infty}^{\infty} x e^{-\frac{1}{2}ax^2} \operatorname{erf}(bx) dx = \frac{2\sqrt{2} b}{a\sqrt{a+2b^2}} . \tag{A.13}$$

$$\begin{aligned}\int_{-\infty}^{\infty} x e^{-\frac{1}{2}ax^2+bx} \operatorname{erf}(cx) dx &= \frac{\sqrt{2}}{\sqrt{a^3\left(a+2c^2\right)}} e^{\frac{b^2}{2a+4c^2}} \\ &* \left(\sqrt{\pi} b \sqrt{a+2c^2} e^{\frac{b^2c^2}{a^2+2ac^2}} \operatorname{erf}\left(\frac{bc}{\sqrt{a\left(a+2c^2\right)}}\right) + 2\sqrt{a}c\right) .\end{aligned} \tag{A.14}$$

$$\int_{-\infty}^{\infty} x^3 e^{-\frac{1}{2}ax^2} \operatorname{erf}(bx) dx = \frac{2\sqrt{2}\left(3ab+4b^3\right)}{a^2\left(a+2b^2\right)^{3/2}} . \tag{A.15}$$

$$\int_{-\infty}^{\infty} e^{-\frac{1}{2}ax^2} \operatorname{erf}(bx) \operatorname{erf}(cx) dx = 2\sqrt{\frac{2}{a\pi}} \tan^{-1}\left(\frac{2bc}{\sqrt{a}\sqrt{a+2\left(b^2+c^2\right)}}\right) . \tag{A.16}$$

$$\begin{aligned}\int_{-\infty}^{\infty} x^2 e^{-\frac{1}{2}ax^2} \operatorname{erf}(bx) \operatorname{erf}(cx) dx &= 2\sqrt{\frac{2}{\pi}} \left(\frac{1}{a}\right)^{3/2} \tan^{-1}\left(\frac{2bc}{\sqrt{a}\sqrt{a+2\left(b^2+c^2\right)}}\right) \\ &+ 2\sqrt{\frac{2}{\pi}} \frac{4bc\left(a+b^2+c^2\right)}{a\left(a+2b^2\right)\left(a+2c^2\right)\sqrt{a+2\left(b^2+c^2\right)}} .\end{aligned} \tag{A.17}$$

Bibliography

Ade, R. and Desmukh, P.: 2013, Methods for incremental learning - a survey, *Int. J. Data Mining Knowl. Manag. Process.* **3**(4), 119–125.

Ahr, M., Biehl, M. and Urbanczik, R.: 1999, Statistical physics and practical training of soft-committee machines, *The European Physical Journal B-Condensed Matter and Complex Systems* **10**(3), 583–588.

Akansu, A. N. and Haddad, R. A.: 1992, *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*, Academic Press, Inc., Orlando, FL, USA.

Amunts, K., Grandinetti, L., Lippert, T. and Petkov, N. (eds): 2014, *Brain-Inspired Computing, Second International Workshop BrainComp 2015*, Vol. 10087 of *LNCS*, Springer.

Angelov, P. and Sperduti, A.: 2016, Challenges in Deep Learning, *in* M. Verleysen (ed.), *Proc. of the European Symposium on Artificial Neural Networks (ESANN)*, i6doc.com, pp. 489–495.

Baldassi, C., Malatesta, E. M. and Zecchina, R.: 2019, Properties of the geometry of solutions and capacity of multilayer neural networks with rectified linear unit activations, *Phys. Rev. Lett.* **123**, 170602.

Barkai, N., Seung, H. and Sompolinsky, H.: 1993, Scaling laws in learning of classification tasks, *Phys. Rev. Lett.* **70**(20), 3167–3170.

Benczúr, A. A., Kocsis, L. and Pálovics, R.: 2018, Online machine learning in big data streams: Overview, *in* S. Sakr and A. Zomaya (eds), *Encyclopedia of Big Data Technologies*, Springer International Publishing, Cham, pp. 1–11.

Berns, F., Lange-Hegermann, M. and Beecks, C.: 2020, Towards gaussian processes for automatic and interpretable anomaly detection in industry 4.0, *Proc. of the International Conference on Innovative Intelligent Industrial Production and Logistics - IN4PL*, pp. 87–92.

Bertoin, D., Bolte, J., Gerchinovitz, S. and Pauwels, E.: 2021, Numerical influence of relu'(0) on backpropagation, *in* M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang and J. W. Vaughan (eds), *Advances in Neural Information Processing Systems*, Vol. 34, Curran Associates, Inc., pp. 468–479.

Biehl, M.: 2018, A no-nonsense beginner's tool for GMLVQ.
URL: `http://www.cs.rug.nl/~biehl/gmlvq`

Biehl, M., Ahr, M. and Schlösser, E.: 2000, Statistical physics of learning: phase transitions in multilayered neural networks, *in* B. Kramer (ed.), *Advances in Solid State Physics*, Vol. 40, Vieweg, pp. 819–826.

Biehl, M. and Caticha, N.: 2003, The statistical mechanics of on-line learning and generalization, *in* M. Arbib (ed.), *The Handbook of Brain Theory and Neural Networks*, MIT Press, pp. 1095–1098.

Biehl, M., Caticha, N., Opper, M. and Villmann, T.: 2019, Statistical physics of learning and inference, *in* M. Verleysen (ed.), *27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, i6doc.com, pp. 501–509.

Biehl, M., Caticha, N. and Riegler, P.: 2009, Statistical mechanics of on-line learning, *in* M. Biehl, B. Hammer, M. Verleysen and T. Villmann (eds), *Similarity Based Clustering*, Vol. 5400, Springer, pp. 1–22.

Biehl, M., Freking, A. and Reents, G.: 1997, Dynamics of on-line competitive learning, *Europhys. Lett.* **38**, 73–78.

Biehl, M., Freking, A., Reents, G. and Schlösser, E.: 1998, Specialization processes in on-line unsupervised learning, *Philosophical Magazine B* **77**(5), 1487–1494.

Biehl, M., Ghosh, A. and Hammer, B.: 2005, The dynamics of learning vector quantization, *ESANN 2005, 13th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 27-29, 2005, Proceedings*, pp. 13–18.

Biehl, M., Ghosh, A. and Hammer, B.: 2007, Dynamics and generalization ability of LVQ algorithms, *Journal of Machine Learning Research* **8**, 323–360.

Biehl, M., Hammer, B. and Villmann, T.: 2014, Distance measures for prototype based classfication, *in* L. Grandinetti, N. Petkov and T. Lippert (eds), *BrainComp 2013, Proc. International Workshop on Brain-Inspired Computing, Cetraro/Italy, 2013*, Vol. 8603 of *Lecture Notes in Computer Science*, Springer, pp. 100–116.

Biehl, M., Hammer, B. and Villmann, T.: 2016, Prototype-based models in machine learning, *Wiley Interdisciplinary Reviews: Cognitive Science* **7**, 92–111.

Biehl, M., Riegler, P. and Wöhler, C.: 1996, Transient dynamics of on-line learning in two-layered neural networks, *Journal of Physics A: Mathematical and General* **29**(16), 4769–4780.

Biehl, M. and Schlösser, E.: 1998, The dynamics of on-line principal component analysis, *J. Phys. A: Math. Gen.* **31**, 97–103.

Biehl, M., Schlösser, E. and Ahr, M.: 1998, Phase transitions in soft-committee machines, *EPL (Europhysics Letters)* **44**(2), 261–266.

Biehl, M. and Schwarze, H.: 1992, On-line learning of a time-dependent rule, *Europhys. Lett.* **20**, 733–738.

Biehl, M. and Schwarze, H.: 1993, Learning drifting concepts with neural networks, *J. Phys. A: Math. and Gen.* **26**, 2651–2665.

Biehl, M. and Schwarze, H.: 1995, Learning by on-line gradient descent, *Journal of Physics A: Mathematical and General* **28**(3), 643–656.

Bishop, C.: 2006, *Pattern Recognition and Machine Learning*, Springer, Heidelberg, Germany.

Brigham, E. O.: 1974, The discrete fourier transform, *The Fast Fourier Transform* pp. 91–109.

Bunte, K., Schleif, F.-M. and Biehl, M.: 2012, Adaptive learning for complex valued data, *in* M. Verleysen (ed.), *20th European Symposium on Artificial Neural Networks, ESANN 2012*, d-side publishing, pp. 387–392.

Castelo-Branco, I., Cruz-Jesus, F. and Oliveira, T.: 2019, Assessing industry 4.0 readiness in manufacturing: Evidence for the european union, *Computers in Industry* **107**, 22–32.

Caticha, N., Calsaverini, R. and Vicente, R.: 2016, Phase transition from egalitarian to hierarchical societies driven between cognitive and social constraints, *arXiv repository* **1608.03637**.

Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M. and Yin, B.: 2017, Smart factory of industry 4.0: Key technologies, application case, and challenges, *IEEE Access* **6**, 6505–6519.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A. and Batista, G.: 2015, The UCR time series classification archive. `www.cs.ucr.edu/~eamonn/time_series_data/`.

Cocco, S., Monasson, R., Posani, L., Rosay, S. and Tubiana, J.: 2018, Statistical physics and representations in real and artificial neural networks, *Physica A: Stat. Mech. and its Applications* **504**, 45–76.

Cybenko, G.: 1989, Approximations by superpositions of sigmoidal functions, *Mathematics of Control, Signals, and Systems* **2**(4), 303–314.

Das, K. and Ari, S.: 2014, Patient-specific ECG beat classification technique, *Healthcare Technology Letters*.

Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S. and Bengio, Y.: 2014, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, *in* Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence and K. Weinberger (eds), *Advances in Neural Information Processing Systems (NIPS 27)*, Curran Assoc. Inc., pp. 2933–2941.

Ditzler, G., Roveri, M., Alippi, C. and Polikar, R.: 2015, Learning in nonstationary environment: a survey, *Comput. Intell. Mag.* **10**(4), 12–25.

Eger, S., Youssef, P. and Gurevych, I.: 2018, Is it Time to Swish? Comparing Deep Learning Activation Functions Across NLP tasks, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Brussels, Belgium, pp. 4415–4424.

Endres, D. and Riegler, P.: 1999, Learning dynamics on different timescales, *J. Phys. A: Math. and Gen.* **32**(49), 8655–8663.

Engel, A. and van den Broeck, C.: 2001, *The Statistical Mechanics of Learning*, Cambridge University Press.

Faria, E. R., Gonçalves, I. J. C. R., de Carvalho, A. C. P. L. F. and Gama, J.: 2016, Novelty detection in data streams, *Artif. Intell. Rev.* **45**(2), 235–269.

Fessahaye, F., Perez, L., Zhan, T., Zhang, R., Fossier, C., Markarian, R., Chiu, C., Zhan, J., Gewali, L. and Oh, P.: 2019, T-recsys: A novel music recommendation system using deep learning, *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6.

Fischer, L., Hammer, B. and Wersing, H.: 2015, Combining offline and online classifiers for life-long learning, *2015 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 1–8.

Fischer, L., Hammer, B. and Wersing, H.: 2016, Online metric learning for an adaptation to confidence drift, *2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 748–755.

García-Martín, J., Gómez-Gil, J. and Vázquez-Sánchez, E.: 2011, Non-destructive techniques based on eddy current testing, *Sensors* **11**(3), 2525–2565.

Gardner, E.: 1988, The space of interactions in neural network models, *Journal of Physics A: Mathematical and General* **21**(1), 257–270.

Gardner, E. and Derrida, B.: 1988, Optimal storage properties of neural network models, *Journal of Physics A: Mathematical and General* **21**(1), 271–284.

Gay, M., Kaden, M., Biehl, M., Lampe, A. and Villmann, T.: 2016, Complex variants of GLVQ based on Wirtinger's calculus, *in* E. Merényi, J. M. Mendenhall and P. O'Driscoll (eds), *Advances in Self-Organizing Maps and Learning Vector Quantization: Proc. of the 11th Intl. Workshop WSOM 2016, Houston, Texas, USA, January 6-8, 2016*, Springer, Cham, pp. 293–303.

Ge, Z.: 2016, Supervised latent factor analysis for process data regression modeling and soft sensor application, *IEEE Transactions on Control Systems Technology* **24**(3), 1004–1011.

Ghosh, A., Biehl, M. and Hammer, B.: 2005, Dynamical analysis of LVQ type learning rules, *in* M. Cottrell (ed.), *Proc. 5th Intl. Workshop on Self-Organising Maps (WSOM 2005)*, Univ. Paris I, pp. 587–594.

Ghosh, A., Biehl, M. and Hammer, B.: 2006, Performance analysis of LVQ algorithms: A statistical physics approach, *Neural Networks* **19**(6-7), 817–829.

Glorot, X., Bordes, A. and Bengio, Y.: 2011, Deep Sparse Rectifier Neural Networks, *in* G. Gordon, D. Dunson and M. Dudík (eds), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Vol. 15 of *Proceedings of Machine Learning Research*, PMLR, Fort Lauderdale, FL, USA, pp. 315–323.

Goldt, S., Mézard, M., Krzakala, F. and Zdeborová, L.: 2020, Modeling the Influence of Data Structure on Learning in Neural Networks: The Hidden Manifold Model, *Physical Review X* **10**(4), 041044.

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., Holmes, G. and Abdessalem, T.: 2017, Adaptive random forests for evolving data stream classification, *Mach. Learn.* **106**(9), 1469–1495.

Goodfellow, I., Bengio, Y. and Courville, A.: 2016, *Deep Learning*, MIT Press, Cambridge, MA, USA.

Göpfert, J. P., Hammer, B. and Wersing, H.: 2018, Mitigating concept drift via rejection, *in* V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis and I. Maglogiannis (eds), *Artificial Neural Networks and Machine Learning – ICANN 2018*, Springer International Publishing, Cham, pp. 456–467.

Grandinetti, L., Lippert, T. and Petkov, N. (eds): 2014, *Brain-Inspired Computing, International Workshop BrainComp 2013*, Vol. 8603 of *Lecture Notes in Computer Science*, Springer.

Gupta, A., Anpalagan, A., Guan, L. and Khwaja, A. S.: 2021, Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues, *Array* **10**, 100057.

Göpfert, J. P., Artelt, A., Wersing, H. and Hammer, B.: 2020, Adversarial attacks hidden in plain sight, *Symposium on Intelligent Data Analysis*.

Hahnloser, R., Sarpeshkar, R., Mahowald, M., Douglas, R. and Seung, S.: 2000, Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit, *Nature* **405**, 947–951.

Hanin, B.: 2017, Universal Function Approximation by Deep Neural Nets with Bounded Width and ReLU Activations, *arXiv e-print 1708.02691* .

Hastie, T., Tibshirani, R. and Friedman, J.: 2001, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer.

Heingärtner, J., Born, M. and Hora, P.: 2010, Online acquisition of mechanical material properties of sheet metal for the prediction of product quality by eddy current, *Proc. of the 10th European Conference on Non-Destructive Testing*.

Hendrycks, D. and Gimpel, K.: 2016, Gaussian error linear units (gelus). Unpublished manuscript.

Herschkowitz, D. and Opper, M.: 2001, Retarded Learning: Rigorous Results from Statistical Mechanics, *Phys. Rev. Lett.* **86**, 2174–2177.

Hertz, J., Krogh, A. and Palmer, R.: 1991, *Introduction To The Theory Of Neural Computation*, Addison-Wesley, Reading, MA, USA.

Hopfield, J. J.: 1982, Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences* **79**(8), 2554–2558.

Hornik, K.: 1991, Approximation Capabilities of Multilayer Feedforward Networks, *Neural Networks* **4**(2), 251–257.

Ince*, T., Kiranyaz, S. and Gabbouj, M.: 2009, A generic and robust system for automated patient-specific classification of ecg signals, *IEEE Transactions on Biomedical Engineering* **56**(5), 1415–1426.

Inoue, M., Park, H. and Okada, M.: 2003, On-Line Learning Theory of Soft Committee Machines with Correlated Hidden Units-Steepest Gradient Descent and Natural Gradient Descent, *Journal of the Physical Society of Japan* **72**(4), 805–810.

Janakiraman, V. M., Nguyen, X. and Assanis, D.: 2016, Stochastic gradient based extreme learning machines for stable online learning of advanced combustion engines, *Neurocomputing* **177**, 304–316.

Jarrett, K., Kavukcuoglu, K., Ranzato, M. and LeCun, Y.: 2009, What is the best multi-stage architecture for object recognition?, *2009 IEEE 12th International Conference on Computer Vision*, pp. 2146–2153.

Jiang, Y., Yin, S., Dong, J. and Kaynak, O.: 2021, A review on soft sensors for monitoring, control, and optimization of industrial processes, *IEEE Sensors Journal* **21**(11), 12868–12881.

Joshi, J. and Kulkarni, P.: 2012, Incremental learning: areas and methods - a survey, *Int. J. Data Mining Knowl. Manag. Process.* **2**(5), 43–51.

Kadmon, J. and Sompolinsky, H.: 2016, Optimal Architectures in a Solvable Model of Deep Networks, *Advances in Neural Information Processing Systems (NIPS 29)*, Curran Assoc. Inc., pp. 4781–4789.

Kang, K., Oh, J.-H., Kwon, C. and Park, Y.: 1993, Generalization in a two-layer neural network, *Phys. Rev. E* **48**, 4805–4809.

Kästner, M., Hammer, B., Biehl, M. and Villmann, T.: 2011, Generalized functional relevance learning vector quantization, *in* M. Verleysen (ed.), *Proc. Europ. Symp. on Artificial Neural Networks (ESANN)*, d-side, pp. 93–98.

Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H. and Alhussain, T.: 2019, Speech emotion recognition using deep learning techniques: A review, *IEEE Access* **7**, 117327–117345.

Kingma, D. P. and Ba, J.: 2015, Adam: A method for stochastic optimization, *International Conference on Learning Representations (ICLR)*.

Kingsbury, N.: 2001, Complex wavelets for shift invariant analysis and filtering of signals, *Applied and Computational Harmonic Analysis* **10**(3), 234–253.

Kinouchi, O. and Caticha, N.: 1993, Lower bounds on generalization errors for drifting rules, *J. Phys. A: Math. Gen.* **26**(22), 6161–6172.

Kinzel, W.: 1998, Phase transitions of neural networks, *Philosophical Magazine B* **77**(5), 1455–1477.

Kohonen, T.: 1990, Improved versions of learning vector quantization, *IJCNN International Joint Conference on Neural Networks*, Vol. 1, IEEE Computer Society Press, San Diego, pp. 545–550.

Kohonen, T.: 2001, *Self-Organizing Maps*, Vol. 30 of *Springer Series in Information Sciences*, Springer. (2nd edition).

Kohonen, T., Barna, G. and Chrisley, R.: 1988, Statistical pattern recognition with neural network: Benchmarking studies, *Proc. of the IEEE 2nd Intl. Conf. on Neural Networks, San Diego, USA, 1988*, IEEE, pp. 61–68.

Krawczyk, B., Minku, L., Gama, J., Stefanowski, J. and Wozniak, M.: 2017, Ensemble learning for data stream analysis: A survey.

Krizhevsky, A., Sutskever, I. and Hinton, G. E.: 2012, Imagenet classification with deep convolutional neural networks, *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS) - Volume 1*, Curran Assoc. Inc., USA, pp. 1097–1105.

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T. and Hoffmann, M.: 2014, Industry 4.0, *Business & Information Systems Engineering* **6**(4), 239–242.

LeCun, Y.: 1989, Generalization and network design strategies, *in* R. Pfeifer, Z. Schreter, F. Fogelman and L. Steels (eds), *Connectionism in Perspective*, Elsevier, Zurich, Switzerland.

LeCun, Y., Bengio, Y. and Hinton, G.: 2015, Deep learning, *Nature* **521**, 436–444.

Loeffel, P.-X., Marsala, C. and Detyniecki, M.: 2015, Classification with a reject option under concept drift: The droplets algorithm, *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–9.

Losing, V., Hammer, B. and Wersing, H.: 2017, Incremental on-line learning: A review and comparison of state of the art algorithms, *Neurocomputing* **275**, 1261–1274.

Losing, V., Hammer, B. and Wersing, H.: 2018, Tackling heterogeneous concept drift with the Self-Adjusting Memory (SAM), *Knowl. Inf. Syst.* **54**(1), 171–201.

Loureiro, B., Gerbelot, C., Cui, H., Goldt, S., Krzakala, F., Mézard, M. and Zdeborová, L.: 2021, Capturing the learning curves of generic features maps for realistic data sets with a teacher-student model.

Maas, A. L., Hannun, A. Y. and Ng, A. Y.: 2013, Rectifier nonlinearities improve neural network acoustic models, *Proc. 30th ICML Workshop on Deep Learning for Audio, Speech and Language Processing*.

Malhotra, P., Vig, L., Shroff, G. and Agarwal, P.: 2015, Long short term memory networks for anomaly detection in time series, *Proc. of the 2015 European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 89–94.

Mallat, S.: 2008, *A Wavelet Tour of Signal Processing: The Sparse Way*, Elsevier Science.

Marangi, C., Biehl, M. and Solla, S. A.: 1995, Supervised learning from clustered input examples, *Europhys.Lett.* **30**(2), 117–122.

Marcus, G.: 2018, Deep learning: A critical appraisal.

Meir, R.: 1995, Empirical risk minimization versus maximum-likelihood estimation: a case study, *Neural Comput.* **7**(1), 144–157.

Melchert, F., Seiffert, U. and Biehl, M.: 2016a, Functional Representation of Prototypes in LVQ and Relevance Learning, *in* E. Merényi, M. Mendenhall and P. O'Driscoll (eds), *Advances in Self-Organizing Maps and Learning Vector Quantization*, Vol. 428 of *Advances in Intelligent Systems and Computing*, Springer, pp. 317–327.

Melchert, F., Seiffert, U. and Biehl, M.: 2016b, Functional approximation for the classification of smooth time series, *in* B. Hammer, T. Martinetz and T. Villmann (eds), *Workshop New Challenges in Neural Computation 2016*, Machine Learning Reports, Univ. of Bielefeld, pp. 24–31.

Mendenhall, M. J. and Merenyi, E.: 2006, Relevance-based feature extraction from hyperspectral images in the complex wavelet domain, *2006 IEEE Mountain Workshop on Adaptive and Learning Systems*, pp. 24–29.

Mertins, A. and Mertins, D. A.: 1999, *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications*, John Wiley & Sons, Inc., New York, NY, USA.

Mezard, M., Nadal, J. and Toulouse, G.: 1986, Solvable models of working memories, *J. de Phys. (Paris)* **47**(9), 1457–1462.

Moody, G. and Mark, R.: 2001, The impact of the mit-bih arrhythmia database, *IEEE Engineering in Medicine and Biology Magazine* **20**(3), 45–50.

Morales, G. D. F. and Bifet, A.: 2015, Samoa: Scalable advanced massive online analysis, *Journal of Machine Learning Research* **16**(5), 149–153.

Nair, V. and Hinton, G.: 2010, Rectified linear units improve restricted Boltzmann machines, *Proc. 27th International Conference on Machine Learning (ICML)*, Omnipress, USA, pp. 807–814.

Nova, D. and Estevez, P.: 2014, A review of Learning Vector Quantization classifiers, *Neural Comput. Appl.* **25**(3-4), 511–524.

Oostwal, E.: 2020, Phase transitions in layered neural networks: The role of the activation function.
URL: `http://fse.studenttheses.ub.rug.nl/id/eprint/23691`

Opper, M.: 1994, Learning and generalization in a two-layer neural network: The role of the Vapnik-Chervonenkis dimension, *Phys. Rev. Lett.* **72**, 2113–2116.

Opper, M. and Kinzel, W.: 1996, Statistical Mechanics of Generalization, *in* E. Domany, J. L. van Hemmen and K. Schulten (eds), *Models of Neural Networks III: Association, Generalization, and Representation*, Physics of Neural Networks, Springer, New York, NY, pp. 151–209.

Pankaj, M., Lang, A. and Schwab, D.: 2014, An exact mapping from the Variational Renormalization Group to Deep Learning, *arXiv repository [stat.ML]* **1410.3831v1**.

Papari, G., Bunte, K. and Biehl, M.: 2011, Waypoint averaging and step size control in learning by gradient descent, *Machine Learning Reports* **MLR-06/2011**, 16.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E.: 2011, Scikit-learn: Machine learning in Python, *JMLR* **12**, 2825–2830.

Ramachandran, P., Zoph, B. and Le, Q. V.: 2017, Searching for activation functions, *ArXiv* **abs/1710.05941**. Presented at: Sixth Intl. Conf. on Learning Representations, ICLR 2018.

Ramsay, J. and Silverman, B.: 2006, *Functional Data Analysis*, Springer.

Reents, G. and Urbanczik, R.: 1998, Self-Averaging and On-Line Learning, *Phys. Rev. Lett.* **80**, 5445–5448.

Riegler, P. and Biehl, M.: 1995, On-Line backpropagation in two-layered neural networks, *J. Phys. A: Math. Gen.* **28**, L507–L513.

Rosipal, R. and Krämer, N.: 2005, *Overview and Recent Advances in Partial Least Squares*, Vol. 3940, Springer, pp. 34–51.

Saad, D. (ed.): 1999, *On-line learning in neural networks*, Cambridge Univ. Press.

Saad, D. and Solla, S. A.: 1995a, Exact solution for on-line learning in multilayer neural networks, *Phys. Rev. Lett.* **74**, 4337–4340.

Saad, D. and Solla, S. A.: 1995b, On-line learning in soft committee machines, *Phys. Rev. E* **52**, 4225–4243.

Saitta, L., Giordana, A. and Cornuéjols, A.: 2011, *Phase Transitions in Machine Learning*, Cambridge University Press.

Sato, A. and Yamada, K.: 1995, Generalized learning vector quantization, *in* G. Tesauro, D. Touretzky and T. Leen (eds), *Advances in Neural Information Processing Systems*, Vol. 7, MIT Press, pp. 423–429.

Saxe, A. M., McClelland, J. L. and Ganguli, S.: 2014, Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, *in* Y. Bengio and Y. LeCun (eds), *2nd International Conference on Learning Representations (ICLR), Conference Track Proceedings*.

Schneider, P., Biehl, M. and Hammer, B.: 2007, Relevance matrices in LVQ, *in* M. Verleysen (ed.), *Proc. European Symposium on Artificial Neural Networks*, d-side publishing, pp. 37–42.

Schneider, P., Biehl, M. and Hammer, B.: 2009, Adaptive relevance matrices in learning vector quantization, *Neural computation* **21**(12), 3532–3561.

Schneider, P., Biehl, M., Schleif, F.-M. and Hammer, B.: 2007, Advanced metric adaptation in Generalized LVQ for classification of mass spectrometry data, *Proc. 6th Intl. Workshop on Self-Organizing-Maps (WSOM)*, Bielefeld University. 5 pages.

Schwab, K.: 2017, *The fourth industrial revolution*, Crown Business.

Schwarze, H. and Hertz, J.: 1993, Generalization in fully connected committee machines, *Europhysics Letters* **21**(7), 785–790.

Selesnick, I., Baraniuk, R. and Kingsbury, N.: 2005, The dual-tree complex wavelet transform, *IEEE Signal Processing Magazine* **22**(6), 123–151.

Settles, B.: 2009, Active learning literature survey, *Computer Sciences Technical Report 1648*, University of Wisconsin–Madison.

Seung, H. S., Sompolinsky, H. and Tishby, N.: 1992, Statistical mechanics of learning from examples, *Phys. Rev. A* **45**, 6056–6091.

Shen, D., Wu, G. and Suk, H.-I.: 2017, Deep learning in medical image analysis, *Annual Review of Biomedical Engineering* **19**(1), 221–248.

Singh, S. P., Wang, L., Gupta, S., Goli, H., Padmanabhan, P. and Gulyás, B.: 2020, 3d deep learning on medical images: A review, *Sensors* **20**(18).

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. and Ganguli, S.: 2016, Deep unsupervised learning using non-equilibrium thermodynamics, *Proc. of Machine Learning Research* **37**, 2256–2265.

Sophian, A.: 2020, Non-destructive testing (ndt) in industry 4.0: A brief review, *Proc. of the International Conference on Science and Technology*.

Straat, M., Abadi, F., Göpfert, C., Hammer, B. and Biehl, M.: 2018, Statistical Mechanics of On-line Learning Under Concept Drift, *Entropy* **20**(10). Art. No. 775.

Straat, M. and Biehl, M.: 2019, On-line learning dynamics of ReLU neural networks using statistical physics techniques, *in* M. Verleysen (ed.), *27th Europ. Symp. on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, i6doc.com, pp. 517–522.

Tishby, N. and Zaslavsky, N.: 2015, Deep learning and the information bottleneck principle, *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5.

Tsay, W.-J., Huang, C. J., Fu, T.-T. and Ho, I.-L.: 2013, A simple closed-form approximation for the cumulative distribution function of the composite error of stochastic frontier models, *Journal of Productivity Analysis* **39**(3), 259–269.

Tu, J., Ren, M., Manivasagam, S., Liang, M., Yang, B., Du, R., Cheng, F. and Urtasun, R.: 2020, Physically realizable adversarial examples for lidar object detection, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Urbanczik, R.: 1997, Storage capacity of the fully-connected committee machine, *J. Phys. A: Mathematical and General* **30**(11), L387–L392.

Vaidya, S., Ambad, P. and Bhosle, S.: 2018, Industry 4.0 - a glimpse, *Procedia Manufacturing* **20**, 233–238.

van Hemmen, J., Keller, G. and Kühn, R.: 1987, Forgetful memories, *Europhysics Letters* **5**(7), 663–668.

van Veen, R., Biehl, M. and de Vries, G.-J.: 2021, sklvq: Scikit learning vector quantization, *JMLR* **22**(231), 1–6.

Vicente, R. and Caticha, N.: 1997, Functional optimization of online algorithms in multilayer neural networks, *Journal of Physics A: Mathematical and General* **30**(17), 599–605.

Vicente, R. and Caticha, N.: 1998, Statistical mechanics of on-line learning of drifting concepts: A variational approach, *Machine Learning* **32**(2), 179–201.

Villmann, T., Ravichandran, J., Villmann, A., Nebel, D. and Kaden, M.: 2019, Investigation of Activation Functions for Generalized Learning Vector Quantization, *in* A. Vellido, K. Gibert, C. Angulo and J. Martín Guerrero (eds), *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization, WSOM 2019*, Vol. 976 of *Advances in Intelligent Systems and Computing*, Springer, Cham, pp. 179–188.

Wang, D. and Chen, J.: 2018, Supervised speech separation based on deep learning: An overview, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **26**(10), 1702–1726.

Wang, L. and Yoon, K. J.: 2021, Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* . early access.

Wang, S., Minku, L. L. and Yao, X.: 2017, A systematic study of online class imbalance learning with concept drift, *CoRR* **abs/1703.06683**.
URL: `http://arxiv.org/abs/1703.06683`

Watkin, T. L. H., Rau, A. and Biehl, M.: 1993, The statistical mechanics of learning a rule, *Rev. Mod. Phys.* **65**(2), 499–556.

Wirtinger, W.: 1927, Zur formalen Theorie der Funktionen von mehr komplexen Veränderlichen, *Mathematische Annalen* **97**, 357–376.

Witoelar, A., Biehl, M. and Hammer, B.: 2007, Learning Vector Quantization: generalization ability and dynamics of competing prototypes, *Proc. 6th Intl. Workshop on Self-Organizing-Maps (WSOM 2007)*, Univ. Bielefeld, Germany. 6 pages.

Witoelar, A., Ghosh, A., de Vries, J., Hammer, B. and Biehl, M.: 2010, Window-based example selection in Learning Vector Quantization, *Neural Computation* **22**, 2924–2961.

Yoshida, Y., Karakida, R., Okada, M. and Amari, S.: 2017, Statistical Mechanical Analysis of Online Learning with Weight Normalization in Single Layer Perceptron, *Journal of the Physical Society of Japan* **86**(4), 044002.

Zavatone-Veth, J. A. and Pehlevan, C.: 2021, Activation function dependence of the storage capacity of treelike neural networks, *Phys. Rev. E* **103**, L020301.
URL: `https://link.aps.org/doi/10.1103/PhysRevE.103.L020301`

Zdeborová, L.: 2020, Understanding deep learning is also a job for physicists, *Nature Physics* **16**(6), 602–604.

Zliobaite, I., Pechenizkiy, M. and Gama, J.: 2016, An overview of concept drift applications, *Big Data Analysis: New Algorithms for a New Society*, Springer.

Samenvatting

In het eerste gedeelte van dit proefschrift beschouwden we modelscenario's van machine learning en in het tweede gedeelte behandelden we toepassingen in smart industry en relevante methodes omtrent de classificatie van tijdreeksen.

Om bij te dragen aan de behoefte om het theoretische begrip te verbeteren omtrent het gebruik van de Rectified Linear Unit (ReLU) activatiefunctie in kunstmatige neurale netwerken, was de hoofddoelstelling in onze theoretische onderzoeken om karakteristieken te onthullen van het gebruik van de ReLU activatiefunctie en deze te vergelijken met karakteristieken van sigmoïde activatiefuncties in diverse relevante modelscenario's van machine learning. Om dit doel te realiseren gebruikten we technieken uit de statistische natuurkunde om exacte resultaten te krijgen die het typerende leergedrag beschrijven in drie leerscenario's: on-line leren van een stroom van onafhankelijk gegenereerde voorbeelden (Hoofdstuk 2), leren in de aanwezigheid van een veranderende taak (Hoofdstuk 3) en off-line leren van een vaste dataset van voorbeelden (Hoofdstuk 4). Bij de analyse van het leren van een veranderende taak was een bijkomend doel het karakteriseren van het leergedrag van het Learning Vector Quantization (LVQ) model.

In hoofdstuk 2 ontwikkelden we een generiek modelraamwerk voor de formulering van on-line leerprocessen in student-leraar modelopstellingen voor verscheidene leerregels en modellen. De formulering bestaat uit een systeem van differentiaalvergelijkingen die de exacte evolutie beschrijven van order parameters van het model in het limiet van het aantal dimensies naar oneindig. Gebruikmakend van dit raamwerk formuleerden we voor drie verschillende leeropstellingen een systeem van differentiaalvergelijkingen dat de typerende leerdynamica beschrijft van gradiëntafdaling in een tweelaags neuraal netwerk met ReLU activatie in zijn verborgen neuronen: overeenkomende complexiteit van het student en het leraar model, een complexer studentmodel ten opzichte van het leraarmodel en taken

die het studentmodel niet leren kan. Het numeriek integreren van deze differentiaalvergelijkingen leverde de typerende resultaten op. Dit deden we ook voor de vergelijkingen die we vonden in de literatuur voor neurale netwerken met sigmoïde activatie. We vergeleken vervolgens de resultaten van ReLU netwerken met die van sigmoïde neurale netwerken. Uit deze vergelijking bleek dat het leeralgoritme de ReLU neuronen kan combineren om zo het aantal effectieve parameters van het model te beperken, wat mogelijk is door de stuksgewijs lineaire eigenschap van de functie. Daarentegen werd bij overgeparameteriseerde sigmoïde neurale netwerken het aantal effectieve parameters beperkt door het verwijderen van neuronen uit het netwerk. Dit laatste gebeurt middels convergentie van de gewichtsvectoren van de overbodige neuronen naar de nulvector.

In hoofdstuk 3 introduceerden we een veranderende taak en inputdistributie in het modelraamwerk. In de opstellingen met neurale netwerken modelleerden we een veranderende taak door de leraarvectoren willekeurig te veranderen. We introduceerden ook het effect van gewichtsverval in het model. Significante verschillen tussen ReLU- en sigmoïde neurale netwerken werden gevonden wat betreft hun gevoeligheid voor veranderende taken en de effectiviteit van gewichtsverval in dit soort situaties. In zowel ReLU- als sigmoïde neurale netwerken wordt de karakteristieke plateaufase verlengd door de veranderende taak, wat de aanvang van specialisatie vertraagt. De analyse van sigmoïde netwerken toonde dat gewichtsverval het generalisatievermogen kan verbeteren bij een veranderende taak. Het netwerk is echter zeer gevoelig voor de instelling van de sterkte van het gewichtsverval. Een kleine afwijking van de optimale waarde kan er al toe leiden dat het netwerk zich niet meer kan specialiseren. Daarentegen liet de analyse van ReLU netwerken zien dat het netwerk ongevoeliger is voor de instelling van de gewichtsverval parameter; specialisatie van de neuronen is mogelijk voor een groot bereik van waardes. Bovendien toonden we aan middels een analyse van de eigenschappen van de nog niet gespecialiseerde plateautoestand dat gewichtsverval de specialisatie van neuronen versnelt in een aanzienlijk bereik van waardes. Daarentegen stelt gewichtsverval in sigmoïde netwerken de aanvang van specialisatie uit. Dit verklaart de observatie dat gewichtsverval het plateau aanzienlijk verkort bij ReLU netwerken en verlengt in het geval van sigmoïde netwerken.

Hoofdstuk 3 behandelde ook het LVQ machine learning model dat werd bestudeerd in twee veranderende leersituaties: een willekeurige verandering van de cluster centra van de inputdistributie en een verandering van de proporties van de klassen in de datastroom. Onze analyses toonden dat het standaard LVQ leerproces tot op zekere hoogte in staat is om met deze soorten veranderingen om te gaan. We observeerden dat de verandering van de cluster centra het leerproces belemmert. Bovendien blijkt er een complex verband tussen de laagst haalbare generalisatiefout en de leersnelheid.

We lieten zien dat in deze situatie gewichtsverval het generalisatievermogen van het model sterk kan verbeteren. Gewichtsverval biedt echter geen verbetering bij veranderende proporties van klassen. Dit komt doordat gewichtsverval de normen van de LVQ prototype vectoren beperkt, waardoor het systeem minder flexibel wordt en hierdoor niet adequaat kan reageren op de veranderende proporties van de klassen.

In hoofdstuk 4 analyseerden we off-line leren en vergeleken we leerprocessen van ReLU- met sigmoïde neurale netwerken in deze situatie. Hiervoor gebruikten we een modellering die een canoniek ensemble van netwerken in een formele thermische evenwichtstoestand beschouwt. In de evenwichtssituatie is de verdeling van toestanden gegeven door de Gibbs functie die gekarakteriseerd wordt door de formele temperatuur. De gedoofde vrije energie bevat nog een gemiddelde over datasets van een vaste grootte en bepaalt typerende eigenschappen van stochastische optimalisatieprocessen voor de energie van het systeem. Omdat het berekenen van het gemiddelde over de data zeer complex is, namen we het vergemakkelijkende limiet van een hoge temperatuur. Dit was al gedaan in voorgaande werken voor het verkrijgen van bruikbare kwalitatieve inzichten in off-line leerprocessen. De vrije energie is een combinatie van de generalisatiefout en de entropie van het systeem. Voor verschillende dataset groottes minimaliseerden we de vrije energie van ReLU- en sigmoïde netwerken. De resultaten lieten zeer significante verschillen tussen beide netwerken zien: voor tweelaagsnetwerken met ReLU activatie die meer dan twee verborgen neuronen hebben is de faseovergang naar gespecialiseerde neuronen van tweede orde. Daarentegen is deze faseovergang bij sigmoïde netwerken van eerste orde. Na de continue faseovergang in ReLU netwerken ontstaan twee minima van de vrije energie die een vergelijkbaar generalisatievermogen hebben. In het limiet van een oneindig aantal verborgen neuronen toonden we aan dat de generalisatievermogens van deze minima identiek zijn. Bij sigmoïde neurale netwerken heeft daarentegen de concurrerende toestand na de eerste orde faseovergang een aanzienlijk slechtere prestatie vergeleken bij de gespecialiseerde toestand.

Het tweede gedeelte van dit proefschrift behandelde intelligente systemen in industriële toepassingen. In hoofdstuk 5 bespraken we de behoefte aan continue kwaliteitscontrole in op staal gebaseerde productielijnen. Vervolgens voerden we in samenwerking met de industrie een typische Industry 4.0 casestudy uit van een massaproductielijn. Deze productielijn bevat een pers die de producten uit het staal produceert. Het doel was om op basis van metingen gemaakt met een Eddy Current sensor een systeem te ontwikkelen voor de real-time kwaliteitscontrole en identificatie van materiaaleigenschappen. Ten tweede onderzochten we of dit systeem zou kunnen helpen bij het voorkomen van productiefouten. De dataset bestond uit sensormetingen die waren gemaakt op staal dat was gebruikt voor de productie. Op gedeelten van dit staal waren ook trektesten gedaan en deze

meetgegevens waren ook beschikbaar. Diverse analyses van de data met principale componentenanalyse lieten zien dat de variatie van de materiaaleigenschappen ook in de sensormetingen aanwezig was. Verdere verkennende analyse toonde lineaire correlaties aan tussen de sensormetingen en de materiaaleigenschappen. Tevens waren de Eddy Current variabelen onderling sterk gecorreleerd. Om deze redenen fitten we een Partial Least Squares model op de data. Vervolgens demonstreerden we dat de voorspellingen van het model kunnen voorkomen dat materiaal waarmee niet geproduceerd zou moeten worden toch het productieproces in gaat. Met behulp van sensordata gemeten op voorgaande productiedagen in combinatie met logboeken van productiefouten ontdekten we dat de voorspellingen van het model een risicowaarde geven voor het optreden van productiefouten.

Het veelvoorkomende probleem van het classificeren van tijdreeksen werd behandeld in hoofdstuk 6. In dit hoofdstuk stelden we voor om representaties van time series die bestaan uit complexe waarden, zoals verkregen uit de Fourier- of wavelet transformatie, te gebruiken in combinatie met het interpreteerbare Generalized Matrix Learning Vector Quantization machine learning model. Deze methode past relevantiewaarden en prototypes aan in de ruimte van de betreffende transformatie. Dit is gunstig voor bepaalde classificatieproblemen waarin de data geschikt is voor de transformatie. Verder formuleerden we een vergelijking voor de omzetting van de matrix die relevantiewaarden van de variabelen in de transformatieruimte geeft naar een matrix die relevantiewaarden van de variabelen in het tijdsdomein geeft. Dit levert dus naast de relevantiewaarden in de transformatieruimte tevens relevantiewaarden in het originele domein. Hierdoor verbreedt de interpretatie van de classificatiemethode. De voorgestelde methode liet altijd betere classificatieresultaten zien dan de classificatie in het tijdsdomein. Bovendien konden we sterke reducties van het aantal dimensies bewerkstelligen met minder over-fitting en verbeterde efficiëntie tijdens het trainen tot gevolg. In een groter experiment behandelden we de classificatie van hartslagen uit ECG data gebruikmakend van een wavelet transformatie die complexe waarden geeft. Voor dit geval lieten we zien dat het aantal dimensies sterk gereduceerd kon worden, zonder dat de classificatieresultaten hieronder leden. Samenvattend demonstreerden we dat de methode in diverse situaties zeer efficiënt kan zijn voor het reduceren van het aantal dimensies, goede prestaties heeft en een brede interpretatie van de classificatie geeft.