The University of Nottingham

School of Computer Science

Deep Learning Models of Biological Visual Information Processing

Diána Turcsány

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy

July, 2016

Abstract

Improved computational models of biological vision can shed light on key processes contributing to the high accuracy of the human visual system. Deep learning models, which extract multiple layers of increasingly complex features from data, achieved recent breakthroughs on visual tasks. This thesis proposes such flexible data-driven models of biological vision and also shows how insights regarding biological visual processing can lead to advances within deep learning.

To harness the potential of deep learning for modelling the retina and early vision, this work introduces a new dataset and a task simulating an early visual processing function and evaluates deep belief networks (DBNs) and deep neural networks (DNNs) on this input. The models are shown to learn feature detectors similar to retinal ganglion and V1 simple cells and execute early vision tasks.

To model high-level visual information processing, this thesis proposes novel deep learning architectures and training methods. Biologically inspired Gaussian receptive field constraints are imposed on restricted Boltzmann machines (RBMs) to improve the fidelity of the data representation to encodings extracted by visual processing neurons. Moreover, concurrently with learning local features, the proposed local receptive field constrained RBMs (LRF-RBMs) automatically discover advantageous non-uniform feature detector placements from data.

Following the hierarchical organisation of the visual cortex, novel LRF-DBN and LRF-DNN models are constructed using LRF-RBMs with gradually increasing receptive field sizes to extract consecutive layers of features. On a challenging face dataset, unlike DBNs, LRF-DBNs learn a feature hierarchy exhibiting hierarchical part-based composition. Also, the proposed deep models outperform DBNs and DNNs on face completion and dimensionality reduction, thereby demonstrating the strength of methods inspired by biological visual processing.

List of Publications

Research presented in this thesis has resulted in the following publications:

Turcsany, D., Bargiela, A., and Maul, T. (2016). Local receptive field constrained deep networks. *Information Sciences*, 349–350:229–247. doi: 10.1016/j.ins.2016.02.034

Turcsany, D. and Bargiela, A. (2014). Learning local receptive fields in deep belief networks for visual feature detection. In *Neural Information Processing*, volume 8834 of *Lecture Notes in Computer Science*, pages 462–470. Springer International Publishing. doi: 10.1007/978-3-319-12637-1_58

Turcsany, D., Bargiela, A., and Maul, T. (2014). Modelling retinal feature detection with deep belief networks in a simulated environment. In *Proceedings of the European Conference on Modelling and Simulation*, pages 364–370. doi: 10.7148/2014-0364

Acknowledgements

First and foremost, I would like to gratefully thank my supervisor, Prof. Andrzej Bargiela, for supporting my research directions—which allowed me to explore ideas and a research topic I greatly enjoyed—as well as for extending help, encouragement, and motivation throughout the years of this PhD. I also wish to convey my gratitude to Dr Tomas Maul, my second supervisor, for providing support, encouragement, and invaluable comments on my work. Furthermore, I would like to kindly thank Prof. Tony Pridmore who offered supervision and help in the final year of my PhD. Thanks also go to Andy for insightful comments on my annual reports. Likewise, it is a pleasure to express my appreciation to friends and colleagues from the department as well as to the University for providing an excellent environment for research, including a HPC service, which facilitated my experiments.

I would like to gratefully acknowledge my friends and previous lecturers who helped and encouraged my choice to pursue a PhD. Also, I wish to deeply thank all my family members for their assistance, with special thanks to my mother, Katalin, whose relentless support and dedication has been an invaluable help. My sincerest gratitude is also greatly deserved by my partner, James, for years of support, care, and inspiration, which made my work and this thesis possible.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

CD	contrastive divergence learning
CD_1	single-step contrastive divergence learning
CD_n	n-step contrastive divergence learning
ConvNet	convolutional neural network
DAG	directed acyclic graph
DBN	deep belief network
DNN	deep neural network
DoG	difference-of-Gaussians
EM	expectation-maximisation
FFA	fusiform face area
GPU	graphics processing unit
IBP	Indian buffet process
LFW	Labeled Faces in the Wild
LGN	lateral geniculate nucleus
LN	linear-nonlinear model
LRF-DBN	local receptive field constrained deep belief network
LRF-DNN	local receptive field constrained deep neural network
LRF-RBM	local receptive field constrained restricted Boltzmann machine

OFA	occipital face area
OMS	object motion sensing
PCA	principal component analysis
RBM	restricted Boltzmann machine
SD	standard deviation
SPI	simulated photoreceptor input
SRE	squared reconstruction error
SVM	support vector machine

Notation

$< . >_{\phi}$	expectation under the distribution ϕ
$.(L)$	architecture of hidden layer trained by an LRF-RBM
$\boldsymbol{a}$	visible node biases in an RBM
$\boldsymbol{b}$	hidden node biases in an RBM
C	recall
$E(\boldsymbol{v}, \boldsymbol{h})$	energy given visible and hidden node states
ϵ	learning rate
F	F-measure score
fn	number of false negatives
fp	number of false positives
$G(\sigma^{RF}, k)$	Gaussian filter with SD σ^{RF} and filter size k
$\boldsymbol{h}$	hidden node states in an RBM
H	hidden layer in an RBM
$H(.,.)$	cross-entropy error
H_i	i^{th} hidden layer in a DBN or DNN
I^i	weight image of hidden node h_i in an LRF-RBM

k	filter size used during LRF-RBM training
$\boldsymbol{k}$	filter sizes used for training consecutive layers of an LRF-DBN
n	number of visible nodes in an RBM
N	element-wise transformation used during LRF-RBM training
m	number of hidden nodes in an RBM
$\boldsymbol{p}$	ground-truth label
$\hat{\boldsymbol{p}}$	predicted probability of classes
P	precision
$p(.)$	probability
R	receptive field masks in an LRF-RBM
$\boldsymbol{\sigma}$	SDs of Gaussian visible nodes in an RBM
σ^{RF}	SD of Gaussian receptive field constraints in an LRF-RBM
$\boldsymbol{\sigma^{RF}}$	SDs of Gaussian receptive field constraints on consecutive layers of an LRF-DBN
tn	number of true negatives
tp	number of true positives
$\boldsymbol{v}$	visible node states in an RBM
V	visible layer in an RBM or DBN
W	weights between visible and hidden nodes in an RBM

1 Introduction

In this chapter, I provide the background and motivation behind the research described in this thesis, introduce the approach taken, and summarise the main goals and contributions of the presented work.

1.1 Motivation

Research described in this thesis is motivated by the importance of understanding more about neural processing of visual information, the challenge of improving the performance of algorithms for high-level interpretation of visual data, and the widespread potential advantages of developing better computational models of information processing units within the visual pathway.

1.1.1 Computational Vision Systems

The design of computational methods for automatically interpreting visual data holds huge commercial and academic value but remains a highly challenging task. Advancements in this area can generate great impact within technology, security, and medicine with potential applications ranging from face recognition, intelligent image search engines and advanced modes of human-computer interaction to the design of navigation aids and devices for visually impaired people.

Fuelled by such interest, within the field of machine learning and computer vision much research has focused on the development of learning algorithms for understanding high-level semantics in visual data. Even with recent advancements, artificial vision systems still lag behind the highly efficient and accurate human visual system. Therefore, it is highly plausible that gaining more insight into how visual recognition is implemented within the brain can be the key to lift the performance of computational vision systems to a greater level.

1.1.2 Biological Visual Information Processing

The highly accurate vision of humans and other biological systems has long been within the centre of interest in neuroscience and biology. This great effort has brought about important discoveries regarding the morphology and functionality of neural cells and networks. However, our knowledge of visual information processing circuits and the specific roles of cells within the visual pathway is still far from complete.

Although numerous experiments have been conducted on different parts of the visual pathway, due to the high number, diverse functionality and complex connection patterns of neurons, understanding mechanisms behind biological visual processing remains a challenging open problem. Even well studied processing units, such as retinal neural networks contain a number of mysterious cell types whose functionality is yet unrevealed (Masland, 2012).

1.1.3 Computational Modelling of Biological Vision

Extending our knowledge regarding neural processing of visual information is not only important for neuroscientific and medical research but is a key challenge, underpinning many areas of machine learning, cognitive science, and intelligent

systems research.

Designing computational models of circuits within the visual pathway can greatly improve our understanding of biological visual information processing and can, hence, provide a more informed background for the design of visual data processing units, such as retinal implants. Furthermore, modelling mechanisms of biological vision can yield important practical benefits for algorithm design in machine learning, image processing, and computer vision.

A key motivation behind the modelling of biological vision is, therefore, to answer questions about the way biological systems process visual information and transfer this knowledge into the production of artificial vision systems, such as machine learning and computer vision algorithms or retinal implant designs.

1.2 The Proposed Approach

Modelling studies have traditionally focused on directly implementing functionalities of visual information processing cells and circuits based on our current knowledge, obtained from biological studies, regarding the workings of the visual system. Nowadays however, with the constant advancement in experimental equipment and methodology, this 'current knowledge' changes ever so rapidly. Due to incomplete understanding of visual processing mechanisms in biological systems, designing robust computational models of these processes prompts one to account for uncertainty and unknown details.

Consequently, this thesis proposes a flexible data-driven approach that offers great potential for modelling in this uncertain environment. To model visual information processing mechanisms, I investigate existing deep learning methods and probabilistic models, e.g. deep belief networks (DBNs) (Hinton et al., 2006), and propose novel deep learning algorithms.

Deep learning methods, such as DBNs, deep neural networks (DNNs), and convolutional neural networks (ConvNets) (LeCun et al., 1998), extract multiple layers of feature detectors from data, where the consecutive layers correspond to increasingly abstract representations of information. DBNs can be trained efficiently by utilising a layer-wise unsupervised pretraining phase using restricted Boltzmann machines (RBMs) (Smolensky, 1986) on each consecutive layer, which can be followed by a supervised fine-tuning with backpropagation resulting in a DNN (Hinton et al., 2006). Deep learning methods have gained outstanding popularity in recent years owing to their state-of-the-art results achieved on a number of challenging machine learning and computer vision benchmarks.

The retina and the visual cortex exhibit an inherently multi-layered structure which lends itself naturally to modelling with deep learning models. The first part of this thesis investigates deep networks for modelling the earliest stages of the visual pathway, such as mechanisms implemented by the retina. Subsequently, novel deep learning methods are proposed for solving complex visual tasks, which require higher-level understanding of visual information. An example of such a task investigated here is face completion, which involves the *occipital face area* (OFA) and the *fusiform face area* (FFA) of the visual cortex (Chen et al., 2010).

1.2.1 Early Visual Processing

Modelling biological information processing within early stages of the visual pathway, such as the retina, the thalamic lateral geniculate nucleus (LGN), and visual area V1, has key potential for advancing the field of retinal implant engineering and improving image processing and computer vision algorithms.

As opposed to strict traditional methods, I advocate a data-driven approach for modelling the retina and early visual processing. Instead of making assump-

tions regarding the morphology of retinal circuits, I strive to construct models which can automatically learn functional properties of the retinal network. To this end, I train deep learning models, DBNs and DNNs, on my specially devised dataset which simulates input reaching a small area of the retina.

Although DBNs have been used for modelling functionality of certain V1 and V2 cells (Lee et al., 2008), so far, less attention has been given to utilising the great potential of DBNs for modelling lower-level mechanisms, such as retinal processes, in detail.

Here, I address this issue by showing the proposed models are capable of discovering retinal functionality by learning from simulated retinal photoreceptor input data in an unsupervised setting. Among other features of early visual processing, the networks learn feature detectors similar to retinal ganglion cells. Additionally, classification performance is measured on a circular spot detection task, which approximates an early visual processing functionality, and results confirm the advantage of multi-layered network models.

The experiments, thereby, demonstrate that DBNs and DNNs have great potential for modelling feature detection implemented by the retina. Algorithms that can learn to replicate retinal functionality may provide important benefits for retinal prosthesis design. Such methods can increase the fidelity of visual information processing in a prosthetic device to real retinal functions, resulting in better vision restoration.

The photoreceptor input dataset and the analysis of experiments with DBN- and DNN-based retinal models trained on these data were previously described by Turcsany et al. (2014).

1.2.2 High-Level Visual Information Processing

In this thesis, I introduce novel deep learning methods, and using challenging visual tasks inspired by cortical functions, I demonstrate the methods' great capability for modelling high-level visual information processing.

Despite successful attempts in applying deep learning methods to model visual cortical areas (Lee et al., 2008), primary emphasis in the literature has been given to improving the performance of deep learning on visual recognition tasks, rather than increasing the fidelity of deep architectures to neural circuits of the visual pathway. Consequently, a main goal of the research presented here is to introduce deep learning architectures which more closely resemble biological neural networks of the visual pathway, while retaining the models' flexibility and great performance on visual recognition tasks.

To address this, I investigate the introduction of biologically inspired structural constraints into deep learning architectures and training methods, and show how such architectural changes can improve RBM, DBN, and DNN models. Also, I propose a novel RBM training method which improves the capability of the model to adapt its structure according to particularities of the input data. Utilising this method during deep network training results in models with increased flexibility.

1.2.3 Novel Methods

This thesis proposes a novel unsupervised RBM-based model, the *local receptive field constrained RBM* (LRF-RBM), which imposes Gaussian constraints on feature detector receptive fields, thereby limiting the spatial scope of detectors. Concurrently with learning the appearance of features, LRF-RBMs can discover an advantageous non-uniform placement of feature detectors automatically from

data. This way, the LRF-RBM training encourages the emergence of local feature detectors and, at the same time, improves feature detector coverage over key areas of the visual space.

Furthermore, this thesis introduces novel biologically inspired deep network architectures, the *local receptive field constrained DBN and DNN* (LRF-DBN and LRF-DNN). Training of both models starts by an unsupervised phase using LRF-RBMs with increasing receptive field sizes to learn consecutive layers. For LRF-DNNs, fine-tuning of the network weights follows, which in the experiments presented here comprises of training autoencoder networks with backpropagation on an image reconstruction task.

On the challenging *'Labeled Faces in the Wild'* (LFW) face dataset, I show how LRF-RBM feature detectors automatically converge to important areas within face images, e.g., eyes and mouth, forming feature hubs. Also, a comparison of generative models confirm that LRF-DBNs, trained layer-wise with LRF-RBMs, perform face completion tasks better than DBNs. Moreover, unlike DBNs, LRF-DBNs learn a feature hierarchy which exhibits part-based compositionality. On the same dataset, dimensionality reduction experiments demonstrate the superiority of LRF-DNN autoencoders compared to DNNs for reconstructing previously unseen face images with a limited number of nodes. In addition to obtaining lower reconstruction errors, LRF-DNN reconstructions better retain fine details of faces, such as mouth and nose shapes, directions of gaze, and facial expressions.

LRF-RBM, LRF-DBN, and LRF-DNN models, together with the analysis of the conducted experiments, were introduced by Turcsany and Bargiela (2014) and Turcsany et al. (2016).

1.3 Goals and Contributions

Work in this thesis was set out to show important benefits of forging a stronger connection between deep learning and the computational modelling of biological vision. Advantages, I argue, are twofold: on the one hand, flexible deep learning methods can reduce the need for handcrafted circuits by learning powerful models of visual information processing units automatically from data. On the other hand, improving the similarity of deep learning models to neural networks of the visual pathway can boost the performance of deep learning on challenging visual tasks.

The key contributions of this thesis are as follows:

(i) deep learning (DBN and DNN) models of the retina and early vision, which comply with a novel data-driven approach to retinal modelling;

(ii) a new simulated photoreceptor input dataset and a classification task approximating early visual processing functionalities;

(iii) experiments demonstrating the great generative, reconstruction, and classification capabilities of the proposed retinal models;

(iv) experiments confirming the models' capability to learn feature detectors similar to difference-of-Gaussians and Gabor filters, common models of retinal ganglion, LGN, and V1 simple cells, and thereby demonstrating the suitability of DBNs and DNNs for modelling functions of the retina and early vision;

(v) development of a novel local receptive field constrained RBM (LRF-RBM), which uses biologically inspired receptive field constraints to encourage the

emergence of local features;

(vi) an adaptation of contrastive divergence learning (Hinton, 2002) which includes receptive field constraints and

(vii) a mechanism which, concurrently with the learning of LRF-RBM features, can discover advantageous non-uniform feature detector placements from data by automatically identifying important locations within the visual space;

(viii) the development of LRF-DBN and LRF-DNN deep learning models, where LRF-RBMs are utilised during training to obtain architectures inspired by visual cortical organisation;

(ix) experiments on the 'Labeled Faces in the Wild' face image dataset demonstrating LRF-DBNs can learn a feature hierarchy which exhibits part-based composition;

(x) experiments showing LRF-DBN generative models perform face completion superior to DBNs and

(xi) LRF-DNN autoencoders obtain better dimensionality reduction results than DNNs, thereby demonstrating LRF-DBNs and LRF-DNNs can extract highly compact representations of visual information and can constitute powerful models of high-level visual information processing.

This thesis presents some novel methods and experimental results which were first described by Turcsany and Bargiela (2014) and Turcsany et al. (2014, 2016).

1.4 Overview

This thesis is organised as follows. Chapter 2 introduces the concept of deep learning and training algorithms for deep networks. Chapter 3 presents my approach to retinal modelling and describes findings and open questions regarding the workings of the retina. My proposed models of the retina and early vision are evaluated in Chapter 4. Chapter 5 introduces my novel deep learning architectures and training algorithms. Experiments evaluating these novel methods and proposed models of high-level visual information processing are presented in Chapter 6. Chapter 7 summarises the findings of this work and concludes the thesis. Finally, possible avenues for future work are suggested in Chapter 8.

2 Deep Learning

This chapter introduces the concept of deep leaning and describes the learning algorithms behind some established deep learning methods, which will be referred to in Chapters 4 and 5.

2.1 The Deep Learning Shift

Deep learning has emerged as a highly popular direction within machine learning research with the aim of developing methods which can extract multiple layers of features automatically from data. Deep learning algorithms create multi-layer representations of patterns present in the data, where each consecutive layer executes the detection of increasingly complex features. Higher-level feature detectors are constructed using features from lower layers; hence, this representation scheme exhibits increased abstraction levels on each consecutive layer.

A strong analogy with biological neural information processing can be noticed, as it is widely accepted that brains of humans and other biological entities also utilise multi-level representation schemes for the efficient extraction of information from sensory input (Coogan and Burkhalter, 1993; Felleman and Van Essen, 1991; Hinton, 2007; Konen and Kastner, 2008; Van Essen and Maunsell, 1983; Wessinger et al., 2001). Such an organisation with multiple consecutive

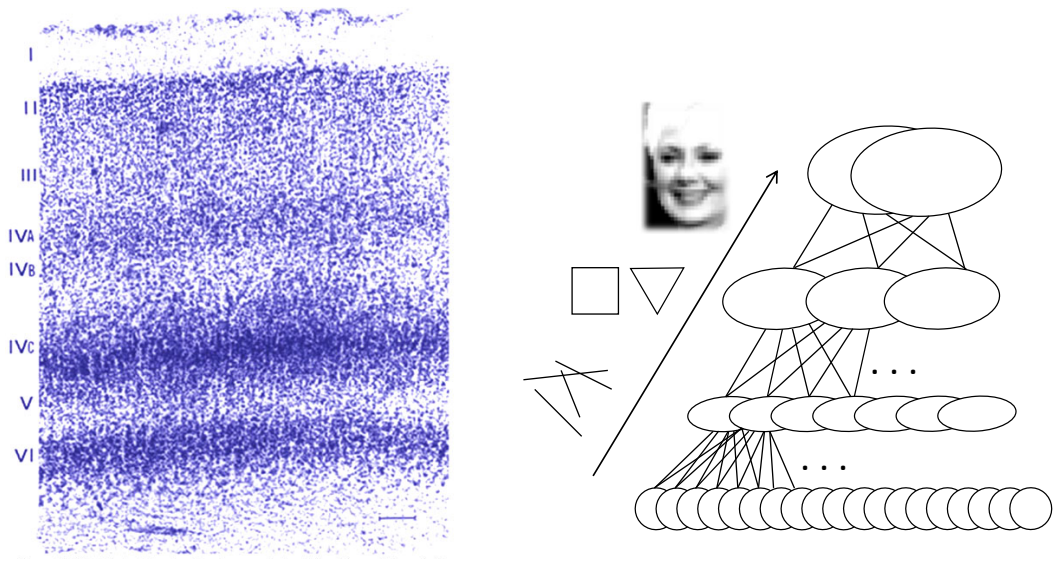


Figure 2.1 (a) Layers I-VI of the visual cortex are shown in a cross-section, illustrating the hierarchical organisation of the visual cortex. Increasing numbers correspond to higher-level processing areas. Original image source: Schmolesky (2000). (b) The concept of deep learning, whereby consecutive layers of representation correspond to features of increasing complexity, is illustrated in the context of visual processing. Lower layers detect simple features, such as edges and shapes, while higher layers correspond to complex features, such as faces and objects.

processing layers can be observed in, for example, the retina and the cortex. Figure 2.1(a) demonstrates the multi-layer structure of the visual cortex in a cross-section, while Figure 3.1(a)–(b) show the numerous processing layers of the retina. Figure 2.1(b) illustrates the concept of deep learning and the similarity between cortical processing and deep learning.

A highly advantageous property of deep learning algorithms is their ability to learn features automatically from data, which eliminates the need for hand-crafted feature detectors. These flexible models have been shown to outperform traditional approaches on numerous information processing tasks, leading to their application not only in academic research but in a variety of industrial sectors. The explored application domains range from computer vision (see Section 2.1.1),

speech recognition (Collobert and Weston, 2008; Graves et al., 2013), text analysis (Salakhutdinov and Hinton, 2009b), reinforcement learning in games (Mnih et al., 2013, 2015), robot navigation (Giusti et al., 2016), and time-series prediction (Kuremoto et al., 2014; Prasad and Prasad, 2014) to even art (Gatys et al., 2015; Mordvintsev et al., 2015). An area where deep learning has had a highly remarkable influence is the visual information processing field.

2.1.1 Impact in Computer Vision

Many areas of visual information processing research, such as computer vision, have traditionally relied on the handcrafted SIFT (Lowe, 2004), SURF (Bay et al., 2008) or similar local feature detectors (Li et al., 2014c; Rublee et al., 2011), often followed by bag-of-words type quantisation techniques (Sivic and Zisserman, 2003; Tao et al., 2014; Turcsany et al., 2013), for the successful design of accurate systems (e.g. Battiato et al., 2007; Lin and Lin, 2009; Zhang et al., 2015).

With the emergence of deep learning, however, focus has shifted towards automatically learning a hierarchy of relevant features directly from input data instead of applying generic, handcrafted feature detectors. Such feature hierarchies can be well captured by multi-layer network models, such as DBNs, DNNs or ConvNets. When applied to visual recognition, multi-level representations are especially suitable for modelling complex relationships of visual features, and the hierarchical structure allows one to represent principles such as composition and part-sharing in an intuitive way. Consequently, deep architectures possess great potential for solving visual tasks, and in recent years such methods have demonstrated exceptional improvements over the existing state of the art, reaching and in some cases surpassing human-level performance (e.g. Cireşan et al., 2012;

Giusti et al., 2016; He et al., 2015b; Mnih et al., 2015; Sun et al., 2015). Deep learning methods routinely achieve state-of-the-art results on computer vision benchmarks, including datasets for handwritten digit recognition (Cireşan et al., 2012, 2011b), object recognition and classification (Cireşan et al., 2011b; He et al., 2015a,b; Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Szegedy et al., 2014), face verification (Ding and Tao, 2015; Schroff et al., 2015; Sun et al., 2014), and human pose estimation (Charles et al., 2016; Pfister et al., 2015; Tompson et al., 2014; Toshev and Szegedy, 2014).

2.1.2 Deep Network Models

Deep neural networks and deep learning models have been investigated since the late 1960s (see, e.g., Ivakhnenko, 1971; Ivakhnenko and Lapa, 1965); however, for a number of years, interest in such models has largely been overshadowed by the success of 'shallow' models, such as support vector machines (SVMs) (Cortes and Vapnik, 1995; Vapnik, 1995).

A recent, remarkable surge in the popularity of deep learning has been triggered by the work of Hinton et al. (2006), where a highly efficient DBN model and training algorithm was introduced. This has been followed by the applications of the method for image classification, dimensionality reduction, and for document clustering and retrieval (Hinton and Salakhutdinov, 2006a; Salakhutdinov and Hinton, 2009b).

To learn a multi-layer generative model of the data where each higher layer corresponds to a more abstract representation of information, Hinton et al. (2006) train a DBN first layer by layer using RBMs. The network parameters learnt during this unsupervised pretraining phase are subsequently fine-tuned in a supervised manner with backpropagation (Linnainmaa, 1970, 1976; Werbos, 1982),

resulting in a DNN. Such models were shown to be superior to principal component analysis (PCA) for dimensionality reduction (Hinton and Salakhutdinov, 2006a).

Since the introduction of this efficient training method for deep networks, deep learning research has gained increased interest, resulting in a wide-ranging academic and commercial adoption of deep learning techniques. A number of successful applications of DBNs, DNNs, and other deep architectures, for example convolutional DBNs (Lee et al., 2009a), ConvNets (LeCun et al., 1998), deep recurrent neural networks (Fernández et al., 2007; Graves et al., 2013), sum-product networks (Poon and Domingos, 2011), and deep Boltzmann machines (Salakhutdinov and Hinton, 2009a), have been presented. Within computer vision, the potential of these and similar methods for automatically learning meaningful features from input data and thereby providing improved models has been demonstrated on:

(i) object recognition (He et al., 2015a,b; Kavukcuoglu et al., 2010; Krizhevsky et al., 2012; Le et al., 2012; Nair and Hinton, 2010; Rozantsev et al., 2015; Simonyan and Zisserman, 2015; Szegedy et al., 2014; Xia et al., 2015),

(ii) image classification (Cireşan et al., 2011a, 2012, 2011b; Giusti et al., 2016; Hinton and Salakhutdinov, 2006a; Jaderberg et al., 2015; Larochelle et al., 2007; Ranzato et al., 2006; Salakhutdinov and Hinton, 2007),

(iii) face image analysis (Ding and Tao, 2015; Huang et al., 2012; Nair and Hinton, 2010; Ranzato et al., 2011; Schroff et al., 2015; Sun et al., 2015, 2014; Turcsany and Bargiela, 2014; Turcsany et al., 2016; Zhu et al., 2013),

(iv) human pose estimation (Charles et al., 2016; Pfister et al., 2015, 2014; Tompson et al., 2014; Toshev and Szegedy, 2014), and on

(v) further visual tasks (Dosovitskiy et al., 2015; Eslami et al., 2012; Gatys et al., 2015; Mnih et al., 2013, 2015; Su et al., 2015; Turcsany et al., 2014).

Besides the analysis of visual data, deep learning methods have been used with great success on problems concerning the analysis of

(i) text (Hinton and Salakhutdinov, 2006a; Salakhutdinov and Hinton, 2009b),

(ii) speech (Amodei et al., 2015; Collobert and Weston, 2008; Fernández et al., 2007; Graves et al., 2013; Lee et al., 2009b),

(iii) further types of audio data (Lee et al., 2009b; Maniak et al., 2015), and

(iv) time-series data (Kuremoto et al., 2014; Prasad and Prasad, 2014).

Through distributed implementations (Le et al., 2012) or with the application of graphics processing units (GPUs) (Cireşan et al., 2011a; Krizhevsky et al., 2012), deep learning methods have been shown to scale up and provide excellent performance even on large-scale problems.

2.2 Multi-layer Representations in the Brain

As mentioned in Section 2.1, the concept of deep learning shows similarity to neural processing, since brains also represent information extracted from sensory input on multiple abstraction levels (see Figure 2.1). Furthermore, deep network models display structural resemblance to biological visual information processing units. For example, as Chapter 3 will demonstrate, the structure of mammalian retinae is inherently multi-layered: the different cell types (i.e. rods, cones, horizontal cells, bipolar cells, amacrine cells, and ganglion cells) are organised into multiple consecutive processing layers such as the photoreceptor, outer plexiform, and inner plexiform layers, which implement increasingly complex functions. On

a larger scale, visual information processing is executed by consecutive areas of the visual pathway (i.e. retina, LGN, V1, V2, etc.) through the extraction of more and more abstract representations of patterns.

Due to the structural analogy, it is not surprising deep learning methods have been used successfully to model certain neural information processing units. For example, Lee et al. (2008) have shown the suitability of DBNs for modelling feature detection in the V1 and V2 areas of the visual cortex. While their study did not focus on the exact replication of neural connectivity patterns in visual cortical areas, features automatically learnt by the network show similarity to typical V1 and V2 feature detectors. These results indicate that such models can successfully learn functionalities implemented by neural networks of the visual pathway and can therefore be applied to model biological visual information processing on a more abstract level.

Despite this achievement in neural modelling, primary emphasis in prior works has been assigned to improving the performance of deep learning on visual recognition tasks, rather than increasing the fidelity of deep architectures to real neural circuits of the visual pathway. A key contribution of my work has been to take a step towards filling this gap by proposing deep network structures that more closely resemble biological neural networks of the visual pathway. The proposed models not only retain the characteristic flexibility of deep networks but further increase their performance on visual recognition tasks. Such architectures possess high potential for learning improved computational models of visual information processing in the retina and the visual cortex.

Chapter 4 introduces my DBN-based retinal model, which provides a further proof for the suitability of deep learning methods for modelling neural information processing units. This model has been shown to successfully learn ganglion

cell functionality from simulated photoreceptor input data and execute an early vision task (Turcsany et al., 2014).

Chapter 5 describes my LRF-DBN and LRF-DNN models (Turcsany and Bargiela, 2014; Turcsany et al., 2016), which have been designed with the aim to increase the structural similarity of deep network models to neural networks of the visual pathway. Chapter 6 demonstrates that this novel deep learning model can successfully implement face completion tasks, known to be executed in high-level processing areas of the human visual cortex (Chen et al., 2010).

Chapters 4 to 6 build on certain concepts of DBN training, which will be introduced in the following sections. These sections describe the steps of RBM and DBN training and introduce common DBN-based models, such as autoencoders.

2.3 Restricted Boltzmann Machines

The first, unsupervised phase of Hinton et al.'s (2006) DBN training utilises RBMs for learning each layer of the representation.

RBMs (Smolensky, 1986) are probabilistic graphical models which originate from Boltzmann machines (Hinton and Sejnowski, 1983). The energy-based Boltzmann machine model contains visible nodes, whose states are observed, and unobserved hidden nodes and has no constraints on which types of nodes can be connected.

RBMs, on the other hand, do not contain links between nodes of the same type, hence the 'restriction' (see Figure 2.2 for an illustration). RBMs are therefore bi-partite graphs, consisting of a visible and a hidden layer, where symmetric connections exist between layers but not within layers. As a result, they are significantly quicker to train than Boltzmann machines due to the conditional independence of hidden nodes given visible nodes and vice versa.

This efficiency in training has resulted in widespread use of RBMs and their variants within machine learning, including applications to:

(i) document retrieval (Hinton and Salakhutdinov, 2006a; Xie et al., 2015),

(ii) collaborative filtering (Georgiev and Nakov, 2013; Salakhutdinov et al., 2007),

(iii) link prediction (Bartusiak et al., 2016; Li et al., 2014a,b; Liu et al., 2013),

(iv) multi-objective optimisation problems (Shim et al., 2013),

(v) visual tasks (Elfwing et al., 2015; Eslami et al., 2012; Hinton and Salakhutdinov, 2006a; Kae et al., 2013; Memisevic and Hinton, 2007; Nair and Hinton, 2010; Nie et al., 2015; Ranzato et al., 2011; Salakhutdinov and Hinton, 2009a; Sutskever et al., 2008; Turcsany and Bargiela, 2014; Turcsany et al., 2014, 2016; Wu et al., 2013; Xu et al., 2015; Zhao et al., 2016),

(vi) modelling of motion capture data (Mittelman et al., 2014; Sutskever et al., 2008; Taylor and Hinton, 2009; Taylor et al., 2006) as well as

(vii) speech and audio data analysis (Dahl et al., 2010; Lee et al., 2009b; Maniak et al., 2015).

In visual recognition tasks visible nodes usually correspond to visual input coordinates (e.g. image pixels), while hidden nodes represent image feature detectors and can, thus, be seen as models of neurons in the visual pathway.

The following sections will first discuss how RBMs with binary visible and hidden nodes are trained, then show how Gaussian visible nodes can be used to model real-valued data. For more details regarding the training of RBMs, refer to (Hinton, 2002, 2012; Hinton et al., 2006; Nair and Hinton, 2010).

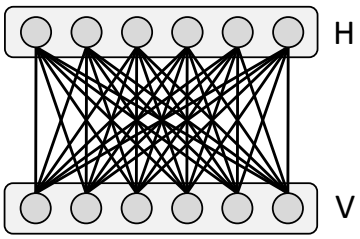


Figure 2.2 Diagram showing a restricted Boltzmann machine. The model consists of a visible (V) and a hidden layer (H), where connections only exist between layers but not within layers. V corresponds to the input data coordinates, while hidden nodes in H are used for learning features from the input data.

2.3.1 The RBM Model

The probability that an RBM model assigns to a configuration $(\boldsymbol{v}, \boldsymbol{h})$ of visible and hidden nodes can be calculated using the energy function (Hopfield, 1982) of the model. In the case of binary visible and hidden nodes, the RBM's energy function takes the form:

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{a}^T \boldsymbol{v} - \boldsymbol{b}^T \boldsymbol{h} - \boldsymbol{v}^T W \boldsymbol{h} \,, \tag{2.1}$$

where $\boldsymbol{v} \in \{0,1\}^n$ and $\boldsymbol{h} \in \{0,1\}^m$ describe the binary states of the visible and the hidden nodes, respectively, $W \in \mathbb{R}^{n \times m}$ is the weight matrix defining the symmetric connections between visible and hidden nodes, while $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^m$ provide the biases of visible and hidden nodes, respectively.

The probability of a joint configuration $(\boldsymbol{v}, \boldsymbol{h})$ is then given by:

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{e^{-E(\boldsymbol{v}, \boldsymbol{h})}}{\sum_{\boldsymbol{\eta}, \boldsymbol{\mu}} e^{-E(\boldsymbol{\eta}, \boldsymbol{\mu})}} \,. \tag{2.2}$$

The probability an RBM assigns to a given configuration $\boldsymbol{v}$ of visible nodes can

be obtained after marginalising out $\boldsymbol{h}$:

$$p(\boldsymbol{v}) = \frac{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{\eta},\boldsymbol{\mu}} e^{-E(\boldsymbol{\eta},\boldsymbol{\mu})}} \, . \tag{2.3}$$

Unsupervised learning in an RBM aims at increasing the log probability of the training data, which is equivalent to reducing the energy of the training data. During the training phase, the probability of a given training example can be increased (the energy reduced) by altering the weights and biases. The following learning rule can be applied to maximise the log probability of the training data by stochastic steepest ascent:

$$\Delta w_{ij} = \epsilon(< v_i h_j >_{data} - < v_i h_j >_{model}) \, , \tag{2.4}$$

where ϵ is the learning rate and $< \, . \, >_{\phi}$ is used for denoting expectations under the distribution ϕ.

Due to the conditional independence properties in RBMs, sampling for $v_i h_j$ according to the distribution given by the data is simple. In the case of an RBM with binary visible and hidden nodes, the probability of a hidden node h_j turning on given a randomly chosen training example $\boldsymbol{v}$ is:

$$p(h_j = 1|\boldsymbol{v}) = \frac{1}{1 + exp(-b_j - \sum_i v_i w_{ij})} \, ; \tag{2.5}$$

that is the logistic sigmoid function applied to the total input of the hidden node. An example of an unbiased sample is then given by $v_i h_j$. Sampling for the visible nodes is similarly easy, i.e. the probability of a visible node v_i being 1 given the

states $\boldsymbol{h}$ of the hidden nodes is:

$$p(v_i = 1|\boldsymbol{h}) = \frac{1}{1 + exp(-a_i - \sum_j h_j w_{ij})} \,. \tag{2.6}$$

On the other hand, calculating an unbiased sample of $< v_i h_j >_{model}$ would require a long sequence of alternating Gibbs sampling between the visible and hidden layers; therefore, in practice approximations are normally applied.

2.3.2 Contrastive Divergence Learning

Contrastive divergence learning (CD) (Hinton, 2002) is an efficient approximate training method for RBMs. Even though CD only broadly approximates the gradient of the log probability of the training data (Hinton, 2002, 2012), in practice CD has been found to produce good models. RBMs trained efficiently using CD and also 'stacked' RBMs, used for pretraining a DBN, are powerful tools for learning generative models of visual, text, and further types of complex data.

In the single-step version (CD_1) of contrastive divergence learning, each training step corresponds to one step of alternating Gibbs sampling between the visible and hidden layers starting from a training example. The algorithm proceeds as follows:

(i) first, the visible states are initialised to a training example;

(ii) then, binary hidden states can be sampled in parallel[1] according to Equation (2.5);

(iii) followed by the reconstruction phase, where visible states are sampled using Equation (2.6);

[1]Due to conditional independence of hidden nodes given visible nodes.

(iv) finally, the weights are updated according to:

$$\Delta w_{ij} = \epsilon(< v_i h_j >_{data} - < v_i h_j >_{reconst}) \,, \tag{2.7}$$

where ϵ is the learning rate, the correlation between the activations of visible node v_i and hidden node h_j measured after (ii) gives $< v_i h_j >_{data}$, while the correlation after the reconstruction phase (iii) provides $< v_i h_j >_{reconst}$[2].

In order to obtain an improved model, the sampling stage in each step can be continued for multiple iterations, resulting in the general form of the CD algorithm: CD_n, where n denotes the number of alternating Gibbs sampling iterations used in a training step.

2.3.3 RBMs with Gaussian Visible Nodes

While RBMs with binary visible nodes are generally easier to train than RBMs with Gaussian visible nodes, the latter can learn better models of real-valued data, such as the simulated photoreceptor input data in Chapter 4 or the face images studied in Chapter 6.

In the case of Gaussian visible nodes the energy function changes to:

$$E(\boldsymbol{v}, \boldsymbol{h}) = \sum_i \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_j b_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij} \,, \tag{2.8}$$

where σ_i is the standard deviation corresponding to visible node v_i. The probability of hidden node activation becomes:

$$p(h_j = 1|\boldsymbol{v}) = \frac{1}{1 + exp(-b_j - \sum_i (v_i/\sigma_i) w_{ij})} \,, \tag{2.9}$$

[2] A similar learning rule is applied to the biases.

and the expected value of a Gaussian visible node (i.e. the reconstructed value) is given by:

$$< v_i >_{reconst} = a_i + \sigma_i \sum_j h_j w_{ij} \, . \tag{2.10}$$

When Gaussian visible nodes are used instead of binary visible nodes, certain learning parameters may need to be adjusted (e.g. the learning rate generally has to be lowered).

2.4 Deep Belief Networks

Deep belief networks are probabilistic graphical models capable of learning a generative model of the input data in the form of a multi-layer network. Within these models, multiple consecutive layers facilitate the extraction of highly non-linear features, making DBNs capable of learning powerful models of even complex data distributions. It has been shown DBNs, given certain architectural constraints, are universal approximators (Sutskever and Hinton, 2008). Furthermore, the trained models define a generative process, which enables the construction of new datapoints that fit the modelled input data distribution. Such a generative DBN is illustrated in Figure 2.3.

Deep belief networks can be trained efficiently according to the method of Hinton et al. (2006) by first pretraining the network in a 'greedy' layer-by-layer fashion with unsupervised RBMs in order to learn a favourable initialisation of the network weights. In this process, after using an RBM to train a hidden layer, the weights become fixed, and the activations on this hidden layer provide the input to the next RBM.

Once pretraining has been completed, the multi-layer network can be fine-tuned either as a generative model or discriminatively, to solve a specific task.

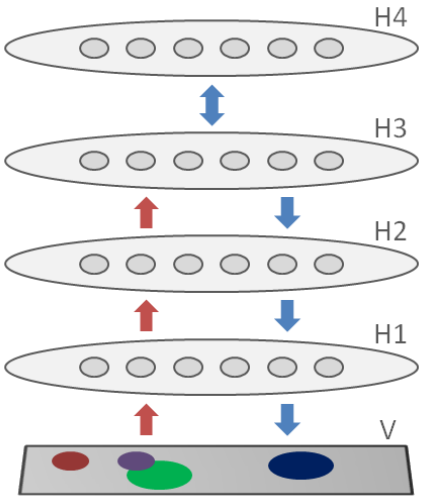


Figure 2.3 Diagram of a DBN with 4 hidden layers trained on an image dataset. V denotes the visible and H_1–H_4 the hidden layers. The blue arrows correspond to the generative model, while the upward-pointing red arrows indicate the direction of recognition. In the generative model, the top two layers form an associative memory.

Fine-tuning can be conducted by, e.g., backpropagation or a variant of the 'wake-sleep' algorithm (Hinton, 2007; Hinton et al., 2006; Hinton and Salakhutdinov, 2006a).

In the following, common use cases and applications of DBN models are introduced.

2.4.1 Generative Model

Hinton et al.'s (2006) DBN generative model is composed of multiple consecutive layers, where the lower layers contain directed connections, while the top two layers have undirected connections and form an associative memory (Hopfield, 1982). If certain conditions are met, it can be shown that adding a further hidden layer to a DBN produces a better generative model of the data (Hinton et al., 2006). Fine-tuning of a DBN with the aim of obtaining an improved generative model can be conducted using the 'wake-sleep' algorithm (Hinton et al., 1995,

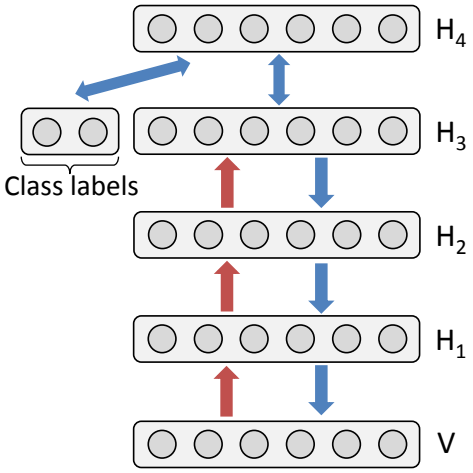


Figure 2.4 Schematic of a DBN generative model containing 4 hidden layers (H_1–H_4) used for generating images together with their class labels. V denotes the visible and H_1–H_4 the hidden layers. The blue arrows correspond to the generative model, while the upward-pointing red arrows indicate the direction of recognition. H_3, extended with the class label nodes, and H_4 form an associative memory.

2006) after the greedy layer-wise pretraining phase.

A DBN generative model can either be trained completely unsupervised or, alternatively, labels can be introduced. The model proposed by (Hinton, 2007; Hinton et al., 2006) learns to generate new data together with the appropriate class labels. To this end, the penultimate layer of a DBN is extended with nodes corresponding to class labels, and the top RBM learns to model the data jointly with the labels (see diagram in Figure 2.4). It is possible to generate new images from a given class by executing alternating Gibbs sampling between the top two layers while appropriately clamping the value of the class labels, then calculating top-down activations. The capability of such a DBN to learn a generative model of handwritten digits was demonstrated on the MNIST dataset (Hinton et al., 2006).

The trained model can also be used for classification. In this case, bottom-up (i.e. recognition) activations are calculated after feeding in an example; then,

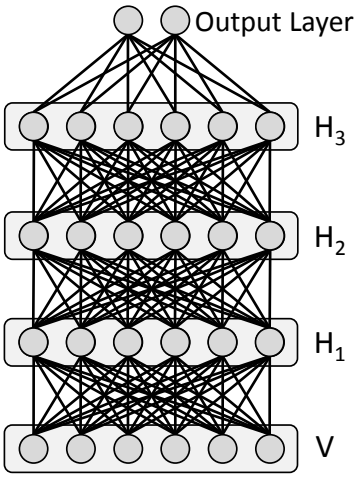


Figure 2.5 Schematic of a DNN used for classification. The network contains a visible layer V corresponding to the input, 3 hidden layers (H_1–H_3), and an output layer with nodes corresponding to the classes.

while the bottom-up activations on the penultimate layer are kept fixed, alternating Gibbs sampling is conducted between the top two layers, and the probability of each class label turning on is compared.

2.4.2 Deep Neural Networks and Classification

Using backpropagation, a pretrained DBN can be explicitly tuned to solve a classification task. After the network has been pretrained layer by layer as a generative model, an output layer is added, where the nodes correspond to class labels. Then, backpropagation is performed on the training set to minimise classification error, which can, for example, be measured using the cross-entropy error:

$$H(\boldsymbol{p}, \hat{\boldsymbol{p}}) = -\sum_i p_i \ln \hat{p}_i \,, \tag{2.11}$$

where, given a training example, the true label defines $\boldsymbol{p}$, while $\hat{p}_i$ denotes the probability of the example belonging to category i according to the prediction of the model.

When backpropagation is used for fine-tuning, the resulting model is a deep neural network (DNN)[3]. A schematic diagram of such a classifier DNN is shown in Figure 2.5. These discriminatively fine-tuned models generally provide better performance on classification tasks than the generative classifier described in Section 2.4.1 (see Hinton, 2007). Such DNN models have achieved excellent results on challenging datasets, including the MNIST handwritten digit classification problem (Hinton et al., 2006).

In the context of retinal modelling, Chapter 4 analyses the performance of DNNs on a discriminative circle detection task and confirms the importance of pretraining.

2.4.3 Autoencoders

Autoencoder DNNs are used for dimensionality reduction of input data. Their multi-layer network architecture consists of an encoder and a decoder part, where the encoder part is trained to generate reduced-dimensional codes for input datapoints. From such an encoding, the decoder part is capable of calculating a reconstruction which approximates the original input example. A diagram of an autoencoder is shown in Figure 2.6.

Autoencoder DNNs are pretrained as DBNs where the top layer has fewer hidden nodes than the input dimension. Representations obtained on this layer therefore provide a lower-dimensional encoding of the input. This pretraining constitutes an initialisation for the encoder part of the deep autoencoder, while to obtain the decoder part, the pretrained layers are 'unrolled' by transposing the weights on each layer. The weight initialisations obtained for the encoder and decoder layers provide a starting point for backpropagation, whereby the encoder

[3]Although, in some of the literature the resulting DNNs are also referred to as DBNs.

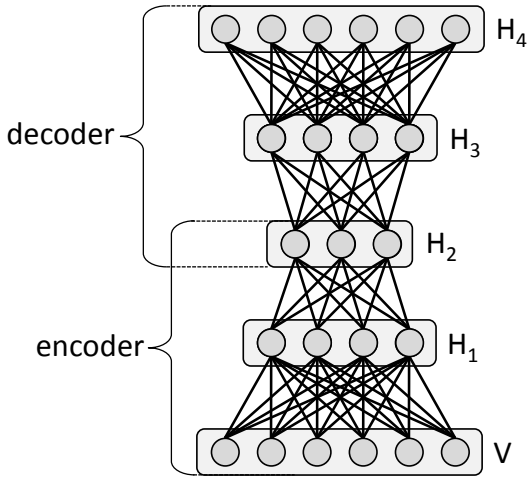


Figure 2.6 Schematic of a deep autoencoder neural network used for dimensionality reduction. The network consists of an encoder part, which calculates reduced-length codes (H_2) for input examples, and a decoder part that can obtain a reconstruction (H_4) from such an encoding.

and decoder are fine-tuned as one multi-layer network. During optimisation, the squared reconstruction error (SRE) is minimised, which is defined as the squared distance between the original data and its reconstruction. Salakhutdinov and Hinton (2009b) apply such an autoencoder to produce compact binary codes for documents in a text retrieval task.

Chapter 5 introduces a new type of deep autoencoder network, the LRF-DNN autoencoder, and compares results with the traditional method on a face image reconstruction task. A schematic of deep autoencoder training on this face dataset is shown in Figure 5.3(b) of Chapter 5.

A more detailed description of autoencoder training is provided by Hinton and Salakhutdinov (2006a), where fine-tuning of DBNs as classifiers is also described.

2.5 Learning the Structure of Deep Networks

Structure learning can improve the flexibility of a modelling approach by identifying the best structural parameters of a model automatically during training. In the case of DBNs, such parameters describe the network architecture, i.e. the number of layers and the number of nodes in each layer.

A variety of approaches have been proposed in the literature for learning the structure of belief networks or the architectural parameters of a neural network. However, due to scalability issues, the use of structure learning methods on complex real-world problems is currently limited. In the following, a few promising approaches to structure learning are introduced with a primary focus on Bayesian non-parametric methods.

For learning the structure of belief networks, heuristic procedures (Heckerman et al., 1995) and evolutionary algorithms (Pelikan et al., 2001) have been explored in the literature, along with a structure learning extension of the expectation-maximisation (EM) algorithm (Dempster et al., 1977), called model selection EM, introduced by Friedman (1997).

Efficient structure learning is considerably easier to implement in certain restricted types of models, such as mixtures-of-trees, for which Meila and Jordan (2001) proposed an EM-based algorithm. In a more general setting, a method based on Mercer kernels was introduced by Bach and Jordan (2002) for learning the structure of graphical models. For a specific class of neural networks, Angelov and Filev (2004) developed a method that can adapt the networks' structure in an online manner, thereby providing improved flexibility.

A popular approach is the use of Bayesian non-parametric methods for inferring structure (Ghahramani, 2012), which provides a principled way of learning

the structural parameters of a probabilistic model from data.

2.5.1 Bayesian Non-parametrics

Bayesian non-parametric methods allow for the introduction of suitable prior probability distributions over structural components of a model. Using standard probabilistic inference methods, the model structure can be inferred together with the model parameters. Commonly used priors include the Chinese restaurant process (Aldous, 1985; Pitman, 2002), applied in the infinite mixture model, and the Indian buffet process (Griffiths and Ghahramani, 2005, 2011) for the infinite latent feature model.

In a mixture model, each example belongs to a category, whereas a latent feature model represents objects by their sets of features. Each object can possess multiple features, and in the infinite latent feature model, the total number of available features is infinite. Fitting this model can be formalised as a problem to learn a sparse binary matrix, where rows correspond to objects, columns correspond to features, and elements of the matrix describe whether the given object possesses the given feature.

In the infinite model, the number of objects (rows) is finite and the number of features (columns) is infinite. To regularise the effective number of features in the model, i.e. features that are actually possessed by the objects, a suitable prior distribution has to be used. The Indian buffet process (IBP) defines a distribution on infinite sparse binary matrices (finite row, infinite column) and therefore provides a suitable prior. Using the IBP the expected number of features per object will follow a Poisson distribution.

As demonstrated by Adams et al. (2010), the cascading (stacked) IBP is suitable for learning the structure of a sparse DBN from data, including the

number of layers, number of nodes per layer, and the type of each node: either binary or continuous valued. In this work, a flexible 2-parameter version of the IBP is used and inference is conducted using sampling. An IBP prior is also utilised in the work of Feng and Darrell (2015), where the structure of a deep convolutional network is learnt from data.

Non-parametric approaches are highly promising for learning the structure of networks; however, for some distributions, variational inference methods are not available and inference using sampling can take an extensive amount of time on complex models, which limits current use.

2.5.2 Hyperparameter Learning

In the case of neural networks, practical application of the methods is not only complicated by the difficulty of identifying a suitable network structure, but successful training also often requires extensive tuning of certain hyperparameters, such as the learning rate and momentum. By calculating the gradients of cross-validation performance with respect to the hyperparameters, Maclaurin et al. (2015) propose a method for automatically optimising any type of hyperparameter (e.g. network architecture, learning rate, and the distribution used for weight initialisation) in neural networks trained using stochastic gradient descent with momentum.

2.6 Feature Visualisation

In order to gain a deeper understanding of the internal representations used by a DBN to encode input data or the process a classifier DNN follows to make predictions, it would be highly beneficial to examine the feature detectors learnt on each layer of the network. In the following, I will describe how feature detectors

on each layer of a deep network can be visualised when the network has been trained on image data.

As the weights of nodes on the first hidden layer correspond to image locations, a visualisation of these features can be obtained in a straightforward manner by displaying the hidden nodes' weight vectors in the shape of the input image data.

Visualisation of features on consecutive layers in a DBN or DNN is complicated by the presence of non-linearities on higher layers. However, by applying an approximate method used in prior work (Erhan et al., 2009; Lee et al., 2008, 2009a), a useful visualisation can still be obtained. For each hidden node on a higher layer, a feature visualisation can be constructed by using the connection weights to calculate a linear combination of those features from the previous layer that exhibited the strongest connections to the given higher-layer node. This way, feature visualisations can be calculated in a layer-by-layer manner.

Chapters 4 and 6 will use the above described method to show features of DBNs and DNNs trained for modelling the retina and face images.

2.7 Summary

This chapter has introduced the concept of deep learning together with recent advances within this line of machine learning research. Key steps of DBN and DNN training have been provided, including a description of the RBM model and the contrastive divergence learning algorithm.

As discussed, in recent years deep learning algorithms have obtained state-of-the-art status on a large number of machine learning and computer vision benchmarks. It is, however, not surprising given the recency of deep learning developments that various theoretical and practical questions in this area have

not yet been answered. It is still unclear what architectures are the most powerful for learning multi-layer representations and what training methods guarantee the most reliable learning performance alongside efficient computation. This applies especially to problems where the type of data or the learning task differs from the conventional cases. From a practical point of view, application of deep networks to challenging 'real-world' problems is non-trivial and the process often involves extensive parameter tuning.

Such challenges and the potential great benefits make investigations concerning deep architectures and suitable training algorithms highly important for the advancement of machine learning research and related applications.

3 Modelling the Retina

This chapter describes the motivation behind designing computational models of the retina, summarises key knowledge concerning retinal cells and circuits, introduces previous computational models, and contrasts these methods with my novel approach to retinal modelling. I will also give examples of open questions regarding the workings of the retina and provide arguments for the suitability of my retinal modelling approach for the study of these questions.

3.1 Motivation for Retinal Modelling

Extending our knowledge of information processing mechanisms within the retina could lead to major advances in the design of learning algorithms for artificial retinae. Such algorithms are much sought after, as building better computational models of the retina is crucial for advancing the field of retinal implant engineering and can also be beneficial for improving image processing and computer vision algorithms.

Despite insights into the anatomy, physiology, and morphology of neural structures, gained through experimental studies, understanding biological vision systems still remains an open problem due to the abundance, diversity, and complex interconnectivity of neural cells. Retinal cells and networks are not exempt

from this rule even though they are relatively well studied. For example, recent studies have revealed a lot more complexity in the functionality implemented by retinal cells than what was traditionally believed (Gollisch and Meister, 2010; Moritoh et al., 2013).

As a consequence, currently, designing computational models of the retina involves dealing with a great amount of uncertainty and only partially known information. In such circumstances, a competitive modelling approach has to produce computational models which are flexible, in order to enable the discovery of yet unknown retinal features, and are easily adaptable in case new information becomes available regarding the retina. Consequently, my aim is to utilise flexible modelling approaches and develop computational models which exhibit substantial fidelity to the retina's currently known neural structure, whilst being highly adaptable. My research towards the construction of these flexible models builds on deep learning and probabilistic modelling techniques within machine learning, such as DBNs.

The following section provides an introduction to cell types and circuits identified in mammalian retinae and summarises key information that influenced the modelling approach proposed in this thesis. The description is based on review papers by Kolb et al. (1995), Kolb (2003), Troy and Shou (2002), Wässle (2004), Field and Chichilnisky (2007), Masland (1996, 2012), and Euler et al. (2014). For more in depth discussions, please refer to the original papers.

3.2 Anatomy and Physiology of the Retina

Mammalian retinae contain a variety of cell types, which are organised into consecutive layers. The 6 main types of retinal cells are: rods, cones, horizontal cells, bipolar cells, amacrine cells, and ganglion cells. Within the main cell cat-

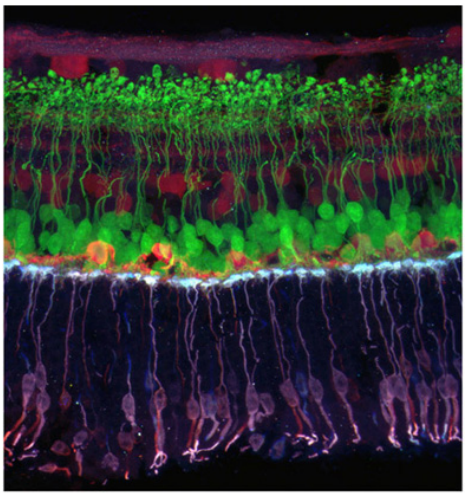
egories, there also exist a number of variants, which can exhibit differences in their receptive fields (i.e. the area of the visual space in which a stimulus can result in response of the neuron) and have diverse morphological, electrophysiological, and functional properties. The exact number of these variant types can differ between species and in many cases the catalogue is not yet thought to be complete (Masland, 2012). There is also an on-going debate in the literature regarding what constitutes a unique type. Therefore, in many cases the following description provides a range or an approximate value for the number of cell types.

Figure 3.1 illustrates the spatial relationship between different retinal cells and the stratification of the retina. In Figure 3.1(b)–(c) the layers of retinal cells are shown in schematic diagrams, while a cross-section of a mouse retina is provided in Figure 3.1(a), showing immunostained retinal cells.

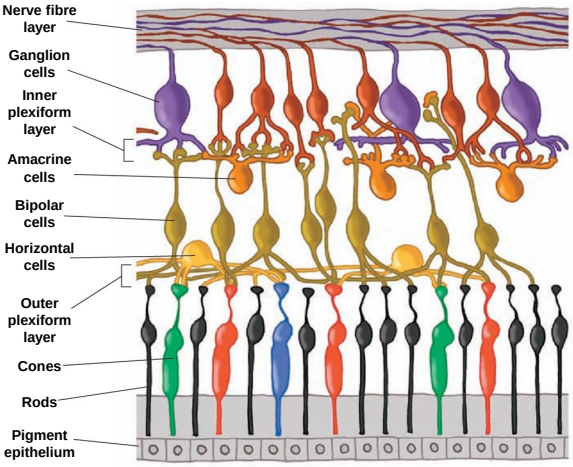
3.2.1 Photoreceptors

Visual information processing in the retina starts in the layer of photoreceptors by the detection of light. Photoreceptors consist of rods and cones, and the cell bodies of these cells form the *outer nuclear layer* of the retina. Cones provide the first processing step in colour vision, predominantly utilised during daytime, whereas rods are capable of sensing light in dark environments and thus are essential for night vision.

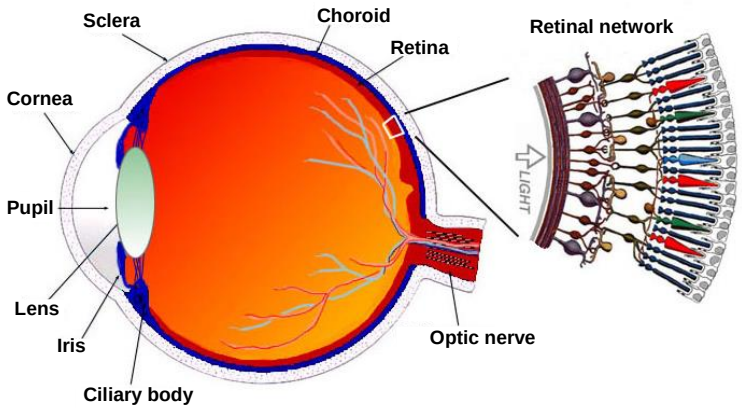
Mammalian retinae contain 2 to 3 types of cones, which are capable of sensing different wavelength light. Colour vision is made possible by comparing the response obtained from different wavelength cones. The retina of humans and that of other trichromatic primates contain 3 types of cones: blue cones sense short wavelength (i.e. blue) light, green cones medium, and red cones long wavelength



(a) Retinal cross-section



(b) Diagram of the retina



(c) Diagram of the eye

Figure 3.1 Neural structure of the retina. (a) Retinal cells stained different colours are shown in a cross-section of immunostained mouse retina. The bottom layer consisting of purple and blue coloured cells corresponds to the photoreceptors. Bipolar cells are shown in green, while amacrine and ganglion cells are stained red. Original image source: Morgan and Wong (2007). (b) A schematic diagram illustrates the layered structure of the retina. The layer of photoreceptors on the bottom, consisting of rods and cones, is the first unit in the visual information processing pipeline. This is followed by the outer plexiform layer featuring synaptic connections between photoreceptors, horizontal cells, and bipolar cells. While the subsequent inner plexiform layer corresponds to the connections between bipolar cells, amacrine cells, and ganglion cells. Original image source: Kolb (2003, p. 30), permission for use granted. (c) Schematic diagram showing the location of the retina on the back of the eye and a close up of the retinal network. Original image source: Kolb (1995).

light. A dichromat's retina, on the other hand, has not evolved to differentiate between medium and long wavelength light and consequently only contains 2 types of cones: a short wavelength blue cone and a long wavelength cone, termed a green cone.

3.2.2 Horizontal Cells

The synaptic connections between photoreceptors, horizontal cells, and bipolar cells are referred to as the *outer plexiform layer* of the retina. With their large receptive fields, horizontal cells connect to a high number of photoreceptors. Moreover, they can connect to neighbouring horizontal cells through gap junctions, thereby further extending the area they receive information from.

The main role of horizontal cells is local gain control, whereby a horizontal cell measures the average illumination within its receptive field and sends inhibitory feedback to photoreceptors. Furthermore, evidence strongly suggests bipolar cells also receive horizontal cell feedback (Euler et al., 2014; Masland, 2012). Through such feedback mechanisms, the photoreceptor–horizontal cell circuit guarantees that the input signal reaching further processing units of the retina is kept within the appropriate range. Additionally, this method also provides some degree of edge enhancement.

3.2.3 Bipolar Cells

Bipolar cells receive input from photoreceptor cells and send their output to ganglion and amacrine cells. In addition, they can receive feedback from different types of amacrine cells.

Bipolar cells can be divided into two main categories based on whether the cells connect to the ON or OFF pathways. The ON pathway is responsible for

detecting light objects on dark background, whereas the OFF pathway detects dark objects on light background. The temporal response properties to light stimulus can also be different among bipolar cells. Some of these cells show transient response at the onset of light, while other bipolar cells express sustained responses, yet others exhibit more complex temporal patterns. Currently, it appears that all (or most) anatomical types of bipolar cells have been discovered, and recent lists in the literature suggest the existence of around 12–13 distinct bipolar cell types (Euler et al., 2014; Masland, 2012). Among these, one type of bipolar cell is rod specific, whereas the others connect to cones.

A characteristic trait of the bottom-up connections is that information is processed through multiple parallel pathways. This phenomena can be observed even at the earliest stages of visual information processing, as the output of a single cone is sent to multiple bipolar cells which represent all suitable bipolar cell types. As for the top-down connection patterns, bipolar cells generally exhibit non-selective connection patterns and connect to all cones within the region covered by their dendrites. Exceptions are a type of cone bipolar cell that connects only to blue cones and a few other types which ignore blue cones.

Some bipolar cells are non-chromatic and have the putative role of conveying brightness information. Other, wavelength specific, bipolar cells implement a step towards colour vision by sending their output to ganglion cells that compare short and long wavelength excitations. In a dichromatic retina, for example, the blue cone specific bipolar cell connects to the *'blue ON/green OFF'* ganglion cell, which can detect the presence of a blue blob in the centre of its receptive field. Recently, it has been demonstrated bipolar cells implement more versatile functions, e.g., they even have roles in contrast adaptation (Brown and Masland, 2001).

3.2.4 Amacrine Cells

The cell bodies of bipolar, horizontal, and amacrine cells form the *inner nuclear layer* of the retina. In addition, there exist a few so-called *displaced amacrine cells* (Pérez De Sevilla Müller et al., 2007), whose cell bodies are located in the ganglion cell layer. Amacrine cells receive input from bipolar cells and, depending on their type, can either send their output to ganglion cells or other types of amacrine cells or provide feedback to bipolar cells.

The approximately 30–42 different types of amacrine cells implement a variety of diverse tasks within the retina (Euler et al., 2014; Masland, 2012). For example, a polyaxonal, wide-field amacrine cell has a crucial role in sensing differential motion of an object and its background (Ölveczky et al., 2003). The process involves a specific, *object motion sensing* (OMS) ganglion cell, whose role is to determine if an object in the ganglion cell's receptive field centre moves with a different speed than the rest of the receptive field area, which corresponds to the background. Central motion of an object results in the OMS ganglion cell receiving excitatory signals from bipolar cells located within the centre of its receptive field. The motion of the larger background area fuels polyaxonal amacrine cells, which subsequently send inhibitory signals to the OMS ganglion cell. If the speed of the background and the centre is the same, the inhibitory signals cancel the excitatory signals before they can reach the ganglion cell. If, however, the object's speed does not agree with the background's speed, the ganglion cell can sense the difference in object and background motion.

Other than a few amacrine cell types that have been examined in such detail, there still exists a number of amacrine cells whose functionality is not yet understood.

3.2.5 Ganglion Cells

The highest level of information processing within the retina is implemented by ganglion cells. These cells receive input from a number of bipolar cells, which can represent a mixture of different types, and occasionally from amacrine cells, like in the case of the above described OMS ganglion cell. The synapses between bipolar cells, amacrine cells, and ganglion cells constitute the *inner plexiform layer* of the retina.

Different ganglion cell types detect specific visual features. Subsequently, information extracted by these cells is transmitted, through the optic nerve and the LGN, towards higher visual processing areas located in the visual cortex. According to current knowledge, between 11–22 types of ganglion cells can be present in mammalian retinae, but the list of existing types is still likely to be incomplete (Masland, 2012). Also, the visual information processing mechanisms implemented by some ganglion cells are still not fully understood. For example, half of the ganglion cell types in the rabbit retina execute yet undiscovered functionalities. Nevertheless, extensive knowledge has been obtained regarding the operational mechanisms of the more common ganglion cell types.

The most well-known types of ganglion cells play roles in local edge detection and have receptive fields which exhibit centre-surround antagonism (Kuffler, 1953): containing either an ON or OFF type centre with the opposite type surround. ON-centre cells receive excitatory signals when light appears in their receptive field centre and inhibitory signals resulting from light in the surround, while the opposite signalling pattern characterises OFF-centre ganglion cells. The receptive fields of these ganglion cells are most often modelled as difference-of-Gaussians (DoG) (see, e.g., Chen et al., 2014; Crook et al., 2011; Dacey et al., 2000; Enroth-Cugell and Robson, 1966; McMahon et al., 2004; Rodieck, 1965).

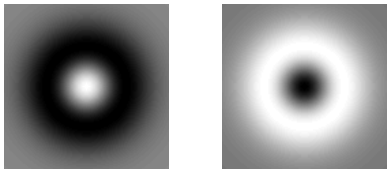


Figure 3.2 Difference-of-Gaussians filters: common models of ON (left) and OFF (right) type ganglion cell receptive fields.

Figure 3.2 shows an ON and an OFF type DoG filter.

In recent years, ganglion cells exhibiting more complex behaviours have been identified in mammalian retinae. Most of these cells show interesting dynamic properties and execute functionalities related to the detection of specific motion patterns in the visual input. The response of these cell types can depend heavily on context, for example, as seen before, the OMS ganglion cell's response depends not only on the motion within the receptive field centre but also on motion patterns of the background. Other examples of ganglion cell functionalities related to specific motion patterns in the input include direction selectivity (Barlow et al., 1964; Vaney et al., 2012), saccadic suppression (Roska and Werblin, 2003), the anticipation of motion direction (Hosoya et al., 2005), and the detection of approaching stimuli (Münch et al., 2009). Additionally, the collective behaviour of ganglion cells can also implement specific tasks. For example, synchronised firing of a ganglion cell population in the tiger salamander retina can signal the reversal of motion (Chen et al., 2014; Schwartz et al., 2007).

Furthermore, a surprising finding has been the discovery of intrinsically photosensitive retinal ganglion cells (Berson et al., 2002), which are capable of directly sensing light through melanopsin, thereby constituting a distinct type of photoreceptor. These ganglion cells have important roles in the regulation of circadian rhythms by signalling extended, constant occurrences of light.

An interesting property of primate and human retinae is the existence of a small central area, the *fovea*, which contains an abundance of cones and exhibits near one-to-one connections between cones, bipolar cells, and ganglion cells. The existence of these so-called *midget* ganglion cells is essential for achieving high resolution colour vision in the centre of the visual field.

Similarly, it is usual to find differences between the centre and the periphery of vertebrate retinae with respect to the density of ganglion cell coverage. For example, in the cat retina, Wässle et al. (1981) show the alpha ganglion cells populate the centre, called the *area centralis*, more densely than the periphery. Peripheral alpha cells, in return, have larger dendritic arbours and hence cover larger areas. As a typical property of the retina, the layout of ganglion cells exhibits a 'tiling' organisation, whereby cells are placed close enough to provide good coverage but far enough to avoid much overlap. Accordingly, the increased spread of peripheral ganglion cell dendritic arbours is necessary for maintaining an almost perfect coverage in all areas of the retina.

3.3 Retinal Modelling on Different Scales

Neural computation processes can be modelled on different abstraction levels, ranging from detailed models of single neurons, through models of smaller neural circuits, to large-scale neural networks. This is no different in the case of retinal modelling, where the literature has investigated:

(i) models of single cells (Fohlmeister et al., 1990; Fohlmeister and Miller, 1997; Kameneva et al., 2011; Liu and Kourennyi, 2004; Miller et al., 2006; Tsai et al., 2012; Usui et al., 1996a,b; Vallerga et al., 1980; Velte and Miller, 1997),

(ii) populations of neurons (Field et al., 2010; Lennie et al., 1991; Nirenberg and Pandarinath, 2012; Pillow et al., 2008; Shlens et al., 2008; Tsai et al., 2012; Werblin and Jacobs, 1996), and

(iii) networks of interacting retinal cell types (Chen et al., 2014; Cottaris and Elfar, 2005; Gaudiano, 1992; Gollisch and Meister, 2010; Hennig and Funke, 2001; Kien et al., 2012b; Maul et al., 2011; Publio et al., 2009; Smith and Vardi, 1995; Teeters et al., 1997, 1991; Werblin, 1991; Zaghloul and Boahen, 2006).

In the works of Fohlmeister et al. (1990), Fohlmeister and Miller (1997), and Kameneva et al. (2011) models capturing single cell firing patterns of ganglion cells are proposed and validated against experimental data obtained from intracellular recordings. Liu and Kourennyi (2004) propose a complete computational model of the rod and show that the model can reproduce response properties of rods measured by patch clamp experiments.

To study the implementation of colour opponency in the retina, Lennie et al. (1991) use a simulation of large-scale ganglion cell populations to aid with the examination of hypotheses regarding their wiring patterns. (Werblin and Jacobs, 1996) develop a cellular neural network model of a retinal bipolar cell population, resulting in the discovery of a new edge enhancement mechanism. Still on the scale of neural populations, Pillow et al. (2008) show that synchronised firing patterns of a retinal ganglion cell population can be described accurately by a model which takes into account the dependence of neighbouring ganglion cells.

The literature also contains a number of models which describe interactions between different retinal cell types. Publio et al. (2009) model circuitries of the rod pathway, while the cone pathway is studied by Teeters et al. (1997) and Cottaris and Elfar (2005). Teeters et al. (1991) provide a simulation to analyse

a potential circuitry responsible for the inhibition of ganglion cell responses to stimuli in the receptive field centre at times when movement occurs in the surround. On a larger scale, each of the main retinal cell types are modelled using predefined low-pass filters in a retinal simulation study by Hennig and Funke (2001).

Motion reversal detection is known to be implemented by populations of ganglion cells in the salamander retina; however, commonly used models of ganglion cells, such as a linear-nonlinear model (LN) or an LN with gain control, cannot reproduce this response (Chen et al., 2014; Schwartz et al., 2007). To this end, Chen et al. (2014) propose the more complex, adaptive cascade model, which produces results consistent with experiments.

As a model of the outer plexiform layer, Maul et al. (2011) propose a recurrent neural network, where the parameters are learnt by evolutionary optimisation and possible functions of the outer plexiform layer are investigated. In prior art, neural network models implementing different types of interacting retinal cells are predominantly of limited complexity and scale; also, the networks typically do not contain more than a single (in some cases recurrent) hidden layer.

With respect to the modelling of early vision, this thesis focuses on a higher level description of retinal circuits and mechanisms. Consequently, instead of requiring fidelity to the characteristics of single neuron behaviour, the primary goal is to develop methods which can identify and execute higher-level functionalities implemented by retinal circuits. Such functionalities include, for example, preprocessing mechanisms employed by the outer plexiform layer and features detected by ganglion cells.

In the centre of my interest stands the assessment of large-scale deep network models for retinal modelling and the identification of such training algorithms

that have the potential of automatically learning functionalities implemented by the retina. In the forthcoming chapter, I will investigate how the use of deep belief and deep neural networks, equipped with a high number of layers and nodes, can provide a powerful new way of modelling the retina.

3.4 A New Retinal Modelling Approach

The following sections compare two different approaches towards modelling the retina. Section 3.4.1 describes a common approach, often found in the computational neuroscience, neuroengineering, and biological modelling literature, based on strict implementation of known mechanisms in the retina. Subsequently, my novel approach to retinal modelling, which proposes the use of flexible probabilistic models and structure learning, is introduced in Section 3.4.2.

To highlight areas where this novel approach can be advantageous, in Section 3.5 some intriguing open questions are presented regarding the retina, and the feasibility of their investigation using the proposed retinal modelling approach is evaluated.

3.4.1 Traditional Approach: *Modelling the retina to the 'best of our knowledge'*

One straightforward way of designing computational retinal models starts with an assembly of readily available information regarding neuronal structures and processes in the retina, followed by the replication of these mechanisms with computer simulation. Such carefully 'handcrafted' retinal models can be used to validate and better understand experimental findings. Also, this approach provides a viable option for retinal prosthesis design, and examples of such systems are numerous in the literature (see, e.g., Gollisch and Meister, 2010; Hallum et al., 2004; Humayun et al., 1996; Maul et al., 2011; Zaghloul and Boahen, 2006).

When following this modelling approach, expert knowledge regarding neural structures and processes previously discovered in the retina can be collected from recent neuroscience studies (e.g. Field et al., 2010; Gollisch and Meister, 2010; Kolb, 2003; Masland, 2012; Wässle, 2004). Relevant findings include information on connection patterns of retinal cells or the functionalities associated with cells or cell populations, e.g. contrast gain control, object motion detection (Ölveczky et al., 2003), or the detection of an approaching object (Münch et al., 2009).

Based on such findings a model can be formulated which describes how cells are thought to co-operate. The behaviour of the simulated retina can then be tested when, for example, new input is used or changes are introduced to the neural structures.

Within a similar framework, it is also possible to propose and investigate the feasibility of some hypothesised functionality of the retina. In Maul et al.'s (2011) work, for example, candidate features are selected first, then experiments are conducted to determine which one of these features can be implemented by circuitries of their proposed retinal model.

3.4.1.1 Critique of the Traditional Approach

Although it seems straightforward to construct a computational model of the retina using hard-wired connection patterns, such a modelling approach has two main shortcomings.

Firstly, a hard-wired model can only be as close to a real retinal neural network as the available information allows. The problem is not only the fact that our knowledge regarding the working of the retina is limited but, also, that discoveries and theories in neuroscience sometimes get proven wrong. A number of theories about the retina were believed to be true until someone with advanced

equipment or new testing protocol has proven the opposite. For example, it is only recently that scientists started to realise the retina is more than just a simple filtering mechanism using difference-of-Gaussians filters. This is due to ganglion cells with more complex functionality being harder to identify through standard experiments. As an example, object motion sensitive ganglion cells were not discovered until the right type of input was presented to the photoreceptors (Gollisch and Meister, 2010; Ölveczky et al., 2003).

A computational model built by the 'traditional' approach therefore may need to be re-designed at a later stage when new findings are obtained from clinical experiments. Also, when proposing novel candidate functionalities of retinal cells using this type of model, one would need to incorporate the uncertainty of any previous findings upon which the model is built.

A second concern regarding the traditional modelling approach is its limited use for identifying potential, yet undiscovered, functionalities implemented by the retina. This is due to the strict hand-selected architecture and, more generally, a modelling approach driven by expert knowledge instead of data.

Even in the case when such models are used for testing the feasibility of a hypothesised retinal functionality, selection of the candidate functionality depends entirely on the choice of the modeller. Such a choice or intuition is influenced by our current knowledge of visual processing; therefore, more complex functionalities, not yet thought of, might get missed.

3.4.2 Novel Approach: *Discovering the neural structure of the retina from data*

With the goal of this work being the design of computational models which enable the discovery of yet unknown retinal features, instead of the traditional route, I propose a data-driven approach towards retinal modelling. Such a modelling

approach repositions the main focus from what is known about the retinal neural network to how and why such a network has developed. Instead of simulating known neural processes in the retina by strict implementation, this data-driven approach strives to discover retinal functionalities automatically from data.

From the biological modelling point of view, the goal of this approach can be interpreted as inferring the whys and wherefores of retinal evolution and development, albeit without strictly replicating such processes. If one considers that a key component of visual information processing is the encoding of patterns present in input data, focusing on methods which can learn such encodings efficiently shows high potential for retinal modelling and can provide insights into the reasons behind the development of certain retinal functions.

Learning a computational model of the retina from data encompasses multiple aspects, such as ensuring the functional and the structural fidelity of the model to a retina. These two aspects are distinct: a model could learn functionalities executed by a retina (e.g. contrast gain control, DoG filtering), thereby demonstrating functional similarities, without adopting the retina's neural network structure. The second aspect of modelling would comprise of approximating the neural network structure of the retina by automatically learning the structure from data. This structure has been optimised through evolution to provide fast computation of various informative features as a first step towards abstract information encoding for visual recognition. Examples of such feature detections implemented by the retina include the presence of dark blobs on light background and vice-versa, the direction of motion (Barlow et al., 1964), detection of an approaching dark object (Münch et al., 2009), and motion reversal detection (Chen et al., 2014; Schwartz et al., 2007). A method capable of learning such a network structure from data could be used for simulating how the neural network of the

retina has evolved and developed to solve feature detection tasks.

While both the traditional and the novel data-driven approach can discover useful clues about the routines followed by retinal cells, the latter has much greater potential for identifying yet undiscovered functionalities in the retina. This thesis therefore proposes a data-driven approach for modelling the retina and the following chapters introduce machine learning algorithms which support such an approach. Furthermore, experimental analysis is provided to assess the ability of these methods for identifying the types of encodings that are extracted by the retina and subsequently conveyed towards higher cortical areas.

3.4.3 Considerations

To learn retinal models in a data-driven way, both supervised and unsupervised learning protocols can be applied, and Chapter 4 investigates the suitability of both types of methods.

The evaluation focuses on deep learning methods which are highly efficient at identifying complex patterns in input data and thereby learning multi-layered models with data-specific feature detectors. Although model architectures examined in Chapter 4 are predefined prior to training, the high number of parameters learnt by the model from data greatly improves flexibility and allows for the encoding of key features even in the case of challenging datasets.

It is also possible to design an algorithm for learning the architecture of the model directly from data. For this, as described in Section 2.5 of Chapter 2, one can build on previous work in the areas of evolutionary algorithms (Pelikan et al., 2001), evolving neural networks (Angelov and Filev, 2004), Bayesian non-parametrics (Adams et al., 2010), structural extensions of EM (Friedman, 1997; Meila and Jordan, 2001), and kernel-based approaches (Bach and Jordan, 2002).

A description of Bayesian non-parametric approaches for structure learning was provided in Section 2.5.1 of Chapter 2, while my proposed deep learning method that incorporates a novel structure learning element is introduced in Chapters 5 and 6.

3.4.3.1 Specifics of Retinal Modelling

In the literature, standard methods for learning the structure of a neural network have mainly been applied in the case of some restricted types of structures, e.g. directed acyclic graphs (DAGs). Therefore, to be able to learn the architecture of the retina, these methods would need to be adapted in order to incorporate the learning of more complex neural structures, such as feedback circuits, which are prevalent within the retina.

When designing a computational model of the retina, it is also important to take into account the following specifics of retinal information processing:

(i) the characteristics of retinal processing can change with time,

(ii) the firing of neurons has an effect on other nearby neurons, and

(iii) the temporal characteristics of neuronal firing is important for information encoding.

The significance of accounting for the effects of neighbouring neurons on each other's firing patterns has been demonstrated by Pillow et al. (2008).

To give an example which emphasises the importance of considering temporal effects, as shown by Ölveczky et al. (2007), the retina is most sensitive to the onset of object motion and thereafter adapts its behaviour by lowering the firing rate in response to the same stimulus. A fully functional model of the retina should, therefore, not be static over time but, rather, be a dynamic model that

can alternate between different states. The model should also be able to adapt to long and short term changes within the environment and account for the complex dynamics of neuronal populations.

3.4.4 Training and Evaluation Protocol

When using a data-driven modelling approach, it is especially important to carefully construct the training dataset. For example, in the case of a supervised task, one has to make sure the training data and the corresponding labels show a high level of similarity to the type of input and output correspondences the retinal neural network would produce. If it was known exactly what input-output correspondences were present during retinal information processing, the evaluation protocol could be constructed by simulating the type of input the retina would receive and comparing the model's output to the required output. However, when modelling the retina with the intention of facilitating the discovery of novel functionalities, the required output is not completely known and inferring it is indeed our task.

3.4.4.1 Experimental Data

Nirenberg and Pandarinath (2012) implement a similar approach by measuring firing patterns of a group of mouse retinal ganglion cells to known input using multielectrode recordings, followed by an approximation of each ganglion cell's processing function by a linear-nonlinear cascade. Shortcomings of the method include the need for laboratory experiments, the necessity of hand-selecting and identifying a set of retinal cells which the modeller conducts the experiments with, and the limitation on the amount of data that is possible to be collected with such an experiment. Also, the use of the LN cascade may be suboptimal for

approximating ganglion cell functionality as such shallow models do not account for the multi-layer structure of the retina, thereby ignoring important processing elements, such as bipolar and amacrine cells.

In summary, acquiring labelled data which describes input-output correspondences of selected neural circuits in the retina is possible with experimental techniques, e.g. multielectrode recordings; however, with such an approach, the modeller's choices regarding the experimental set-up pose inherent constraints on what cell and circuit functions can be studied and the obtainable data are highly limited.

3.4.4.2 Output of Higher-Level Processes

One of the complementary approaches proposed in this thesis uses the output of higher-level processes to help infer certain lower-level functionalities. In the human brain the final output of visual processing is, in some way, easier to estimate than the output of lower processing layers, because humans can interpret and directly use this output at every moment in visual scene understanding. This higher-level representation of information describes, for example, objects which can be recognised in the scene and the direction of their motion, faces and their emotions, or the activities conducted in visual scenes. Furthermore, the output can have components utilised during different classification processes (e.g. determining whether the environment is dangerous), the initiation of eye-movements (e.g. calculating the coordinates of the next saccadic eye movement), or decision making (e.g. after recognising the approach of an object, making the decision to move away). Additionally, by building on knowledge from neuroscience experiments, some assumptions can also be made regarding intermediate level tasks the visual pathway executes.

These observations lead to a complementary way of evaluating the performance of a retinal model by making use of knowledge concerning the output of more complex, higher-level encoding processes. In Chapters 4 and 6, I will give examples of how intermediate and higher-level tasks can be used for guiding the optimisation, and I will show the proposed models can discover feature detectors analogous to the ones implemented by certain retinal neural circuits.

An apparent caveat of this approach is the increased complexity of the tasks used for learning a computational model of the retina. Nonetheless, by following this method, information can also be discovered about functionalities of higher-level processing layers in the visual cortex. It becomes somewhat harder to determine which of the learnt features are plausible functionalities implemented by the retina. However, if there exists some well-known retinal feature detectors among the detectors automatically learnt from data, other features identified by the model could also be considered as putative functions of the retinal network.

Further guarantee can be achieved through the addition of constraints and validation mechanisms to the learning algorithm. For example, receptive field constraints introduced in Chapter 5 can be beneficial for improving the structural similarity of the model to a retina.

3.4.4.3 Unsupervised Methods

A third approach, which I will also investigate through experiments in Chapter 4, utilises unsupervised methods for automatically learning feature detectors from adequate training data. Such detectors are then treated as possible functions implemented by the retinal network or the visual cortex. This method builds on the view that the evolution of a large percentage of visual information processing functionalities within the visual pathway (or, at least, in the earlier parts of the

visual pathway) was driven by the goal to detect and efficiently encode patterns present in input data (see the *efficient coding hypothesis* investigated by Atick and Redlich, 1992; Attneave, 1954; Barlow, 1961; Dong and Atick, 1995; Graham and Field, 2009; Simoncelli and Olshausen, 2001).

Following the 'unsupervised' approach, one can bypass the difficulty of selecting suitable classification tasks, acquiring labelled data, and problems that can arise from noisy classification labels. In this case, it is still important to assemble a suitable training dataset which reflects well the particularities of the input processed by retinal cells. For example, if the input was provided in the form of a colour video stream of natural scenes, one would have to make sure typical image preprocessing techniques, such as noise removal or contrast adjustment, are only applied in moderation if at all. Within the visual pathway, the implementation of such preprocessing functions is mainly the task of the retina; therefore, a complete model of the retina should, indeed, incorporate the execution of these preprocessing tasks and should not rely on prior image processing.

3.4.5 Advantages of the Proposed Approach

Instead of building strict handcrafted retinal models, I propose a data-driven approach for learning a computational model of the retina. Such an approach is advantageous, as it provides a suitable framework for the discovery of yet unknown potential functionalities of retinal cells and circuits. There still exist a number of ganglion cell types whose functions are unknown or not fully understood (Masland, 2012). Without having to hypothesise what processing mechanisms can be implemented in the retina, methods complying with the proposed data-driven approach strive to discover these features automatically from data.

As opposed to the traditional approach driven by expert knowledge regarding the retina, with a data-driven method, it becomes harder to guarantee that the resulting models are plausible neural networks for retinal modelling. However, the flexibility of learning retinal functionality automatically from data has great advantage when the expert knowledge is incomplete or inaccurate. Also, the novel approach possesses much greater potential for furthering our knowledge regarding the workings of the retina and biological visual information processing in general.

As a validation, the discovery of some well-known features of the retina can indicate that other features identified by the model could also be implemented in the retina. Furthermore, I will show how structural constraints can be utilised in the learning algorithm for improving the fidelity of the resulting model to biological neural networks of the visual pathway.

3.5 Open Questions in Retinal Research

In the rest of this chapter I will describe some exciting research directions and open questions within retinal research and modelling, while also discussing whether the proposed data-driven approach and, particularly, methods for network structure learning are suitable for investigating these questions.

3.5.1 Eye Movements

In psychology, neurophysiology, and related areas, remarkable attention has been paid towards understanding eye motion control and the relationship between attention and eye movements (Duhamel et al., 1992; Groner and Groner, 1989; Hoffman and Subramaniam, 1995; Kustov and Robinson, 1996; Liversedge and Findlay, 2000; Shepherd et al., 1986). Although, much of eye motion control

is conducted from higher processing areas in the brain, such as the *superior colliculus* (Kustov and Robinson, 1996; Robinson, 1972; Wurtz and Goldberg, 1972), retinal cells have important roles in relation to the guidance of attention and the execution of eye movements. For example, Roska and Werblin (2003) have found evidence of saccadic suppression in the rabbit retina, i.e. an inhibiting mechanism acting upon certain ganglion cells during fast eye movements, called saccades.

A straightforward way to investigate this area is through the analysis of eye motion data. If such data are available, the control of eye movements can be used as a higher-level task to guide the optimisation within the data-driven framework, as described in Section 3.4.4. This could enable the method to discover features at the retina level which have important roles in the control mechanisms of eye movements.

3.5.2 Retinal Development

Exciting questions accompany the development of neural circuits in the retina. Experiments with light-deprived developing mice have shown a deviation in ganglion cell dendritic distribution (Tian, 2008), which indicates that changes in the environment can have a strong effect on the development and maturation of retinal circuits. A structure learning approach has great potential for modelling circuit development and investigating the effects of sending abnormal input to the photoreceptors during the learning phase.

3.5.3 Colour Opponency

There still remain various details to be learnt regarding the way red-green colour opponency within the visual input is conveyed by the retina towards the brain in

trichromatic primates. The difference between the signals received from the centre and surround of a midget ganglion cell is crucial for channelling information about colour opponency.

According to the *'random wiring hypothesis'* (Paulus and Kröger-Paulus, 1983), the receptive field centre of a midget ganglion cell receives input originating selectively from one type of cone, whereas the surround receives mixed signals gained from both cone types. Here, cone type refers to the wavelength (i.e. red or green colour sensitivity) of the cone. Evidence supporting the random wiring hypothesis has grown significantly in recent years (Crook et al., 2011; Field et al., 2010; Jusuf et al., 2006), due in large part to the use of high resolution multielectrode recordings (Field et al., 2010).

Presently, the most likely hypothesis for the implementation of surround antagonism is based on horizontal cell negative feedback to cones (Crook et al., 2011). Such a finding would provide additional support for the random wiring hypothesis, as these horizontal cells, indeed, connect to both red and green cones non-selectively. In further agreement with the random wiring hypothesis, it is possible that during evolution trichromatic colour vision was implemented using pre-existing circuits, corresponding to the centre-surround midget ganglion cell receptive fields, without the need for rewiring.

On a lower processing level, it is interesting to examine the connection patterns between cones and bipolar cells. In general, midget bipolar cells have connections with only one type of cone (Field et al., 2010), but there exist some bipolar cells which have inter-chromatic connections and, thus, likely convey brightness information (Masland, 2012). From the notion that horizontal cells provide the surround of midget ganglion cells, it also follows that colour opponency can be observed as early as at the level of midget bipolar cells.

The proposed data-driven and structure learning approaches are highly suitable for investigating how these connection patterns, pathways, and mechanisms evolved for colour vision. By providing an opportunity to examine the dominant trends in connection patterns between cones and other cells in the learnt networks, these approaches allow for evaluating the feasibility and likelihood of different hypotheses regarding the implementation of colour vision in the retina.

3.5.4 Rod Pathway

As described above, colour opponency has likely been implemented in the retina by simply building on the existing centre-surround receptive field structure of midget ganglion cells, without major changes in the retinal circuits (Crook et al., 2011). This highlights a trend that is observable in retinal evolution: often, a new functionality is implemented by utilising pre-existing neural circuits instead of resorting to rewiring the neural structure of the retina. Another example is the evolution of the rod pathway, which utilises the previously evolved cone pathway. This phenomenon gives a hint about the particularities of retinal evolution, which can be incorporated within the optimisation algorithm of a structure learning method.

Circuits connecting rods to ganglion cells have been widely studied in the neuroscience literature (see, e.g., Deans et al., 2002; DeVries and Baylor, 1995; Field et al., 2005; Hack et al., 1999; Pahlberg and Sampath, 2011; Soucy et al., 1998; Tsukamoto et al., 2001). Creating computational models of this network is, on its own, a highly interesting task, and the problem lends itself well for applying data-driven modelling approaches. However, trying to learn the circuits of the rod pathway from data makes the task even more challenging and especially suitable for investigation with structure learning algorithms.

3.5.5 Photosensitive Ganglion Cells

As discussed in Section 3.2.5, melanopsin containing ganglion cells with intrinsic photoreceptive capabilities (Berson et al., 2002) were a remarkable addition to the line of rod and cone photoreceptors. Furthermore, their morphology and supposed roles turned out to be much more varied than once thought (Schmidt et al., 2011). Apart from enabling the execution of the pupillary light reflex and the photoentrainment of circadian rhythms (Golombek and Rosenstein, 2010; Wright et al., 2013), i.e. synchronisation of the body's biological clock with the 24 hour cycles of night and daytime, some of these ganglion cells may even have effects on image-forming vision (Schmidt et al., 2011).

Recently, the existence of a further type of photosensitive ganglion cell has been suggested, which, through the use of neuropsin, photoentrains the retina's local circadian clock without depending on the body's central biological clock (Buhr et al., 2015).

From the novelty of these findings follows that there remain a number of unanswered questions concerning the mechanisms involved with photosensitive ganglion cells. The modelling of these cells and circuits would involve extending the proposed data-driven framework with models of non-image-forming visual functions. While this may amount to a more complex task, data-driven approaches could play an important role in studying evolutionary processes behind the formation of these exceptional ganglion cells and the circuitries involved.

3.5.6 Digital Versus Analogue Computation

To understand the information processing mechanisms employed by a neuronal population in the brain, it is interesting to investigate if the cells conduct 'analogue' or 'digital' computation, i.e. whether signalling is executed through graded

potentials or action potentials (spikes), and the possible reasons behind either choice. Shu et al. (2006) show examples of both types of computation in the cerebral cortex. Furthermore, the retina has also been found to employ both graded and action potentials for information encoding (Baden et al., 2013).

Analogue systems can provide highly efficient computation, but the accumulation of noise is often a problem in large-scale systems. Sarpeshkar (1998) therefore hypothesises the brain works as a distributed hybrid system (i.e. conducting both digital and analogue computation) using a large number of noisy neurons and wires in order to ensure highly efficient computation. Averaging can play an important role in noise reduction, and it is also possible the brain implements some structure similar to an *Analogue/Digital/Analogue* adapter in electronics to mitigate noise (Sarpeshkar, 1998).

Retinal circuits implement a number of noise reduction mechanisms. For example, it is now accepted rod bipolar cells in the retina threshold the incoming signals from rods, and the most likely explanation for this is thought to be the improvement of the signal-to-noise ratio (Pahlberg and Sampath, 2011). To examine what encoding mechanisms can provide the best balance between efficient computation and noise-resistant coding in the retina, a data-driven approach could be highly advantageous. Here, both supervised and unsupervised methods, tailored to the architectural constraints and specifics of the retina, can be used to investigate efficient information encoding paradigms.

Due to a lack of large distances within the retinal network, which would warrant the digitalisation of signals, a large portion of information processing in the retina is conducted through graded potentials (Purves et al., 2001). While ganglion cells were known to send information towards the brain using noise-resistant digital signals in the form of action potentials, photoreceptors, horizontal cells,

and bipolar cells, on the other hand, were long thought to produce only graded potentials (Dowling, 1987; Kaneko, 1970; Masland, 2001; Werblin and Dowling, 1969). Recent findings (Baden et al., 2013, 2011; Dreosti et al., 2011; Protti et al., 2000), however, have weakened this theory by identifying examples of both spiking and graded responses recorded from bipolar cells.

It can be beneficial to investigate what reasons, apart from noise removal, exist for the digitalisation of signals in certain neural cells. Adams et al.'s (2010) method for learning the structure of deep belief networks incorporates Frey's (1997) continuous sigmoidal belief network training algorithm, thereby acquiring the ability to infer whether a given hidden node of the network is discrete or continuous. Also, Maul's (2013) neural diversity machine provides a way to learn a different activation function per node in a neural network through a type of evolutionary optimisation. The outcome of such learning methods can provide useful clues for deducing which stages of visual information processing can benefit from digital or analogue computation and, thereby, the reasons behind the existence of neurons with different types of signal encoding characteristics in the retina and the brain.

3.5.7 Retinal Prostheses

The importance of computational models for improving the design of retinal prostheses provides great motivation for research in retinal modelling. Dowling (2008) and Kien et al. (2012a) give detailed descriptions of various prosthesis design principles proposed in the literature.

Although to finalise the development of a retinal implant, access to clinical experimental equipment and test subjects is necessary, the theoretical foundations of prosthesis design can benefit greatly from data-driven modelling of retinal in-

formation encoding mechanisms. An example of a prosthesis system (Nirenberg and Pandarinath, 2012) benefiting from data-driven modelling of ganglion cell functions was introduced in Section 3.4.4.1. Unlike the works of Eckmiller et al. (2005), Hallum et al. (2004), Zaghloul and Boahen (2006), and Humayun et al. (1996), my research investigates theoretical rather than engineering aspects of retinal prosthesis design. As opposed to developing a carefully engineered system based on well-established knowledge regarding some retinal circuits, in this thesis, primary focus is given to methods suitable for automatically learning retinal functions and potentially discovering new information about the retina. While this approach is more indirect, the discovery of new putative features of the retina would inevitably prove highly useful for guiding retinal prosthesis design.

3.5.8 Population Codes

It has become a central topic in neuroscience to examine and model populations of neural cells (Schneidman et al., 2003; Stevenson et al., 2008), as opposed to modelling only single cells. The same trend exists within retinal research (Field et al., 2010; Pillow et al., 2008; Shlens et al., 2008), leading to the recognition of synchronised firing patterns in certain retinal cell populations. For example, synchronised firing has been observed by Pillow et al. (2008) within parasol ganglion cell populations of the peripheral macaque retina. As a computational model of ganglion cell firing, they propose a generalised linear model which takes into account dependencies between neighbouring cells using coupling filters. This model is shown to better fit the measured response patterns of the examined retinal ganglion cell population than a model assuming independence, suggesting correlations exist between the firing patterns of the cell population.

Such findings indicate that when designing a retinal model, it is important to

take into account the effects of synchronised firing among nearby neurons. Such dependencies can be built into a data-driven model in the form of constraints (similarly to the coupling filters used by Pillow et al., 2008). Alternatively, when applying a structure learning algorithm, these dependencies could be discovered automatically from data.

3.5.9 Neuronal Diversity

The presence of diversity within the neuronal responses to the same stimulus is typical in the cortex, with response properties of even nearby cortical cells being greatly varied (Chelaru and Dragoi, 2008; Hubel and Wiesel, 1962; Ringach et al., 2002; Soltesz, 2006). Chelaru and Dragoi (2008) found this inhomogeneity results in improved population codes.

Diversity in neural wiring among individuals is also strongly pronounced, which may have important evolutionary roles (Soltesz, 2006). Differences exist between connection patterns in any two brains (Freund et al., 2013); however, individuals still manage to process information in similar ways and execute the same basic tasks. This indicates, on more abstract levels, the information encodings extracted in the two brains are analogous.

Most structure learning frameworks are highly suitable for representing such differences in the connection patterns of individual neural networks. When evolutionary algorithms are used for structure learning, a whole population of networks is learnt, where key differences exist among the networks. With Bayesian non-parametric structure learning approaches, model structure is usually optimised through sampling, making it possible to assemble a diverse population of plausible models. Such methods result in a group of networks where the individual connection patterns are likely to differ but the majority of the population is able

to solve the designated task. It is possible for the population to contain some incapable networks; however, a good structure learning process should ensure that the probability of retaining such networks is low.

3.5.10 Plasticity of the Brain

The brain is capable of displaying a great degree of plasticity, i.e. adaptation in response to changes and faults in neuronal structures or in the type of input received by the sensory systems (Freund et al., 2013; Kolb and Whishaw, 1998; Pascual-Leone et al., 2005, 2011). A remarkable neuroplasticity has been demonstrated by experiments with adult dichromatic monkeys (Mancuso et al., 2009), where the monkeys acquired trichromatic colour vision after an L-cone photopigment was added to their retinae.

Also, the brain's plasticity has key importance for current retinal prosthesis designs. Signals produced by these prostheses cannot replicate the exact input signals received by ganglion cells in an intact retina; however, the brain can still learn to make sense of the new type of input, at least, to a certain degree (Nirenberg and Pandarinath, 2012).

It is an engaging topic to study how this adaptability is ensured through the use of existing neural pathways and possible rewiring. Data-driven methods can model the effects of changing the normal type of sensory input, and structure learning algorithms are highly suitable for modelling the potential rewiring executed by the brain. Such methods can be especially useful when data from clinical experiments are available. Additionally, using data-driven and structure learning methods, one can evaluate the suitability of different network structures and training algorithms for the implementation of plasticity.

3.6 Summary

In summary, I propose to follow a data-driven approach towards modelling the retina and biological visual information processing in general. I argue that data-driven methods suit the ultimate goal of identifying new features useful for visual recognition. Such informative features may also be computed by retinal cells; therefore, this approach can facilitate the discovery of yet unknown retinal functions. The proposed data-driven approach is highly suitable for investigating a large spectrum of the research questions described in this chapter, and the following chapters will show examples of investigations.

4 Modelling the Retina with Deep Networks

As described in Chapter 3, I advocate a novel data-driven approach towards retinal modelling. This chapter evaluates my proposed retinal model, introduced by Turcsany et al. (2014), which uses DBNs and discriminatively fine-tuned DNN classifiers to learn early vision functionalities automatically. I will also introduce a new dataset that simulates photoreceptor input, together with a task suitable for supervised fine-tuning of the proposed retinal models.

4.1 Experiments

Chapter 3 discussed facts, open questions, and current theories regarding the working of the retina. It has been emphasised that—due to the high number, diverse functions, and complex connection patterns of neurons—our factual knowledge of the retinal network is incomplete and surrounded by a substantial level of uncertainty. In such an environment, I propose to develop data-driven modelling techniques and computational models which exhibit substantial fidelity to the retina's currently known neural structure, while also being flexible and highly adaptable.

As seen in Chapter 2, recent research has demonstrated the capability of flexible, data-driven deep learning methods, such as DBNs and DNNs, for solving

a variety of visual recognition tasks. Promising results have been obtained with RBMs and DBNs in the context of modelling elements of biological vision. However, primary focus has been directed towards higher-level visual features and later stages of visual processing found in the brain, e.g. certain V1 and V2 functionalities (Lee et al., 2008). So far, the great potential of DBNs has not been explored for modelling lower-level processes in detail, such as those implemented in the retina. Work described here addresses this issue and demonstrates how the retina's inherently multi-layered structure lends itself naturally to modelling with deep networks.

My experiments are conducted using a newly constructed dataset and a corresponding detection task, which simulates an early vision functionality. Using this input, DBNs are capable of learning feature detectors similar to retinal ganglion cells, and, after fine-tuning, the resulting DNNs achieve high accuracy on the corresponding detection task. These experiments thereby demonstrate the potential of deep networks for modelling the earliest stages of visual processing.

4.2 Modelling the Retina

Mammalian retinae contain 6 main cell types: rods, cones, horizontal cells, bipolar cells, amacrine cells, and ganglion cells, which are organised into consecutive layers as illustrated in Figure 4.1(a) and in Figure 3.1 of Chapter 3.

Visual information processing starts in the layer of light-sensitive photoreceptors, which consist of rods and cones. Rods are essential for sensing light in darker, low-light environments, whereas cones are mainly utilised during daylight and provide the first step towards the implementation of colour vision. Retinal models described in this chapter focus on early-stage processing units underpinning visual processing in the brain in daylight conditions, with primary

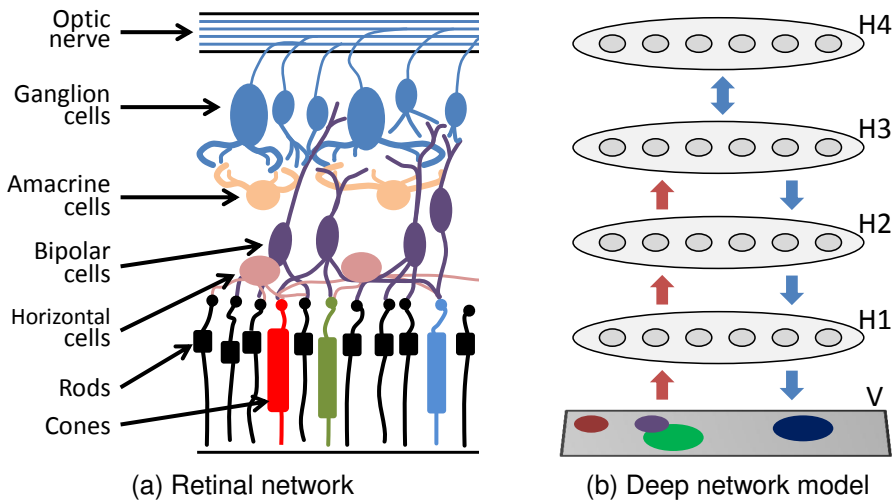


Figure 4.1 Diagrams illustrating the structural resemblance of the retinal network and a DBN. (a) Schematic of interacting cell types in the multi-layer retinal network. (b) Diagram showing an example of the proposed retinal models: a 4-hidden-layer DBN that provides a generative model of the input data. V denotes the visible and H_1–H_4 the hidden layers. Blue arrows correspond to the generative model, while the upward pointing red arrows indicate the direction of recognition (i.e. class label prediction).

emphasis on the pathway between cones and ganglion cells.

Neural computation units in the retina have been modelled at different abstraction levels, ranging from single neuron models to networks. My work focuses on a higher level description of retinal circuits and the development of large-scale network models. The primary goal of this study was to assess the capability of DBN and DNN models to learn functionalities of retinal cells automatically from data without resorting to hard-wired circuitries. Special attention was assigned to learning models of top-layer units, corresponding to the inner plexiform layer of the retina (see Figure 3.1(b) in Chapter 3), such as retinal ganglion cells.

The majority of ganglion cells and bipolar cells exhibit centre-surround receptive field organisation and either join onto the ON or the OFF pathway, which are respectively responsible for detecting light objects on a dark background and vice versa. Ganglion cells implement the highest level of information processing

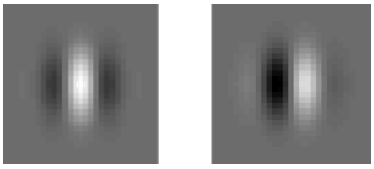


Figure 4.2 Examples of Gabor filters. Such functions are commonly used models of V1 simple cell receptive fields.

in the retina. They detect specific visual features and transmit the extracted information through the optic nerve and the LGN towards higher processing areas located in the visual cortex. Common ganglion cell feature detectors include the 'ON-centre' and the 'OFF-centre' ganglion cells. The former receives excitatory signals when light appears in its receptive field centre and inhibitory signals resulting from light in the surround, while the opposite is true for OFF-centre ganglion cells. The receptive fields of these cells are most often modelled as difference-of-Gaussians (DoG) filters (see Figure 3.2 in Chapter 3).

Centre-surround organisation is also typical of most LGN cells (Piscopo et al., 2013), while V1 simple cell receptive fields are commonly modelled using Gabor functions (Lauritzen and Miller, 2003), shown in Figure 4.2.

The following sections will demonstrate that the proposed DBN- and DNN-based retinal models can learn feature detectors with centre-surround receptive fields, similar to DoG filters, and also some Gabor-like features automatically from data, thereby confirming the adequacy of deep learning algorithms for modelling the retina and early vision.

4.3 A Multi-layer Retinal Model

As discussed, the retina possesses a multi-layer structure, where each layer contains different cell types with specific functions. This distinctive structure can best be captured through a model which exhibits a similar deep architecture and

utilises multiple abstraction levels to encode visual features. This analogy in structure can be observed in the schematic diagrams in Figure 4.1, where the layered organisation of the retina is shown in Figure 4.1(a), while Figure 4.1(b) illustrates a generative DBN model with 4 hidden layers (H_1–H_4) trained on simulated photoreceptor input data.

Deep learning methods, such as DBNs and DNNs, constitute a powerful means for learning successive layers of representation from data, where features on each consecutive layer are of increasing complexity. Such methods that can automatically learn a hierarchy of distinctive features from data are highly suitable for constructing a data-driven retinal model, as proposed in Chapter 3.

Additionally, there exists some gaps in our understanding regarding the morphology and functionality of retinal cells and circuits. Probabilistic models, e.g. DBNs and RBMs, can provide the required flexibility for modelling in the presence of uncertainty.

The following experiments investigate my proposed DBN- and DNN-based models of early visual processing, which incorporate both the outer and inner plexiform layers of the retina.

The primary goals of the following experiments and analysis are to:

(i) design a dataset, which simulates input received by a small area of the retina;

(ii) on this dataset, set out a classification task that approximates a functionality of early visual processing circuits;

(iii) investigate the capability of unsupervised RBMs and DBNs to learn features similar to the ones detected in the retina and the brain during the early stages of visual processing;

(iv) analyse images produced automatically by these DBNs when used as generative models;

(v) evaluate the reconstruction performance of the proposed DBN models;

(vi) fine-tune DBNs in a supervised manner on the specified classification task and compare the performance of the resulting DNN classifiers to 'traditional' DNNs (i.e. DNNs trained with backpropagation without pretraining); and thereby

(vii) evaluate the suitability of the proposed DNNs and DBNs for modelling the retina and early vision in comparison with traditional models.

4.4 Simulated Photoreceptor Input Dataset

The retinal network implements the first stages of object recognition and visual information processing in general by performing a variety of image enhancement (e.g. contrast adjustment) and feature extraction routines (e.g. local edge detection). To test the suitability of DBNs and DNNs for learning retinal and early vision functionalities, I have designed a *simulated photoreceptor input (SPI) dataset and task*, which approximates the detection of light spots or 'blobs' in visual input received by photoreceptors. The detection or localisation of blobs, objects, and object parts in scenes constitutes an integral part of visual information processing both within the visual pathway and in computer vision systems.

Retinal ganglion cells which have centre-surround receptive fields are capable of signalling the presence of a small light spot in their receptive field centre. The specific response patterns in reaction to different coloured spots depend on the type of the ganglion cell. For example, in an ON-centre ganglion cell, the centre receives excitatory signals resulting from light, while the presence of

light in the antagonistic surround causes inhibition. Consequently, maximum excitation occurs when a light spot occupies the centre but not the surround, while the opposite scenario, when light covers the surround but not the centre, results in inhibition. Patterns in-between these extremal cases cause intermediate responses (e.g. moving edges can cause excitations somewhat weaker than the maximal excitatory response triggered by a light spot covering just the centre or, similarly, a large spot covering both the centre and parts of the surround can evoke weaker excitatory responses).

The SPI dataset was constructed with the aim of providing an easily controllable test-bed with large amounts of labelled data for evaluating methods on a circular spot detection task. The dataset contains circular spots of various colours and sizes superimposed on a different coloured, uniform background. To approximate different wavelength light input and obtain a challenging, varied dataset, images with multiple background colours are included, showing circular spots which can overlap each other and thereby build up complex arrangements and shapes.

In this simulated environment, there is no limit on the amount of data that can be generated, and classification labels for the detection of circular spots (or blobs) are readily available.

Similar input data, such as movies with flashing light spots of various sizes or moving greyscale spots, have been used in experimental studies, e.g., for measuring LGN neural responses in vivo (Piscopo et al., 2013). In our case, circular spots of different sizes and colours on a uniform background simulate the input reaching a small area of the retina, and automatically generated classification labels model responses of early visual processing neurons. Although greyscale image datasets are popular in image processing, computer vision, and machine

learning, experiments presented in this chapter use RGB images in order to keep stronger similarity to the photoreceptor input of trichromats.

The SPI dataset contains both video data and an image dataset assembled using the videos.

4.4.1 Video Dataset

First, a video dataset of simulated photoreceptor input was obtained by generating several videos consisting of moving circular spots in front of a background. Each video had a different background colour, which was unchanged throughout the video. Multiple groups of videos were produced exhibiting different statistics with respect to the average size and speed of the circles and the expected number of circles per frame.

This video dataset is intended for learning models of retinal circuits dedicated to the detection of motion-related features, such as circuits for differential motion detection (Ölveczky et al., 2003) or for the prediction of motion direction (Hosoya et al., 2005).

4.4.2 Image Dataset and Classification Task

Experiments presented in this chapter focus on learning the spatial receptive field structure of retinal cells automatically from suitable input data. To obtain this input, a 'static' image dataset was generated by extracting reduced-size subwindows from randomly sampled snapshots taken from a group of videos. The size of subwindows is similar to the diameter of the largest circles in the dataset. This image dataset was compiled in order to simulate input reaching a small area of the retina, approximately corresponding to the receptive field of a ganglion cell.

Some of the subwindows were chosen to be centred around a circular spot, while the rest did not have a circle positioned in the exact centre of the subwindow. This way, the resulting image data, consisting of the subwindows, can be categorised into two distinct classes. The classification task is thereby given as predicting whether the given image contains a circle centred in the middle, which defines a circle (or blob) detection task[1].

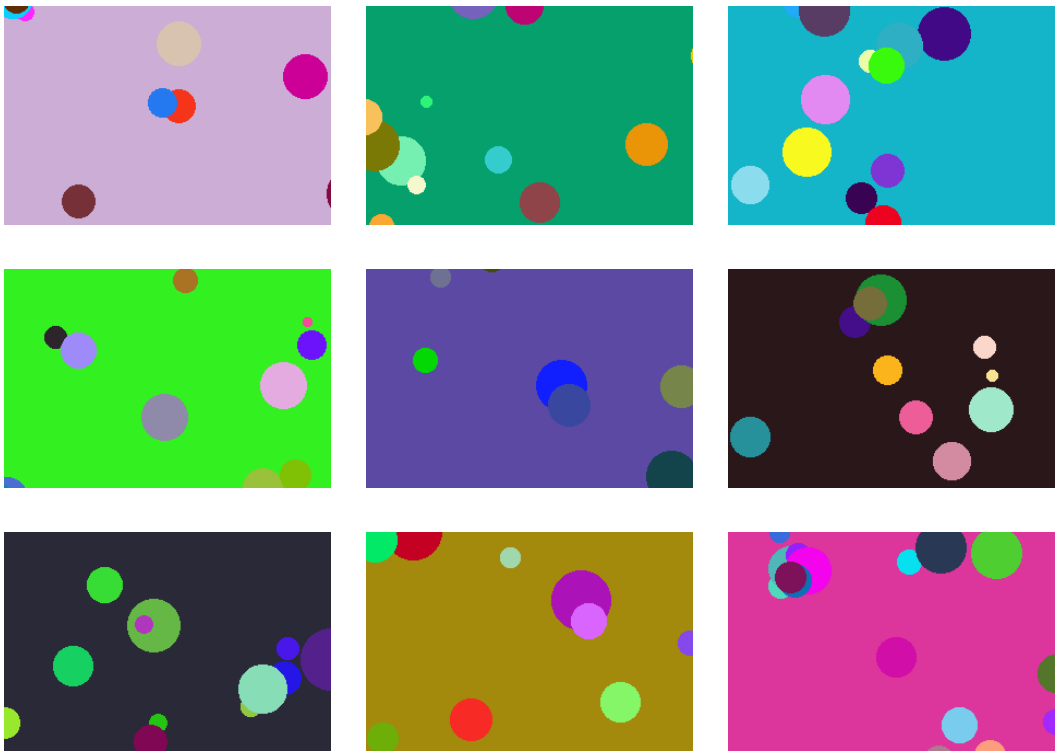
This task mimics how ganglion cells are capable of signalling when contrasting light patterns reach the centre and surround of their receptive fields. Results in Section 4.7.1 will show, in order to solve this task, DBNs develop feature detectors similar to those implemented in the earliest stages of visual processing, including analogues of retinal ganglion cells with centre-surround receptive fields.

4.4.3 Training and Test Set

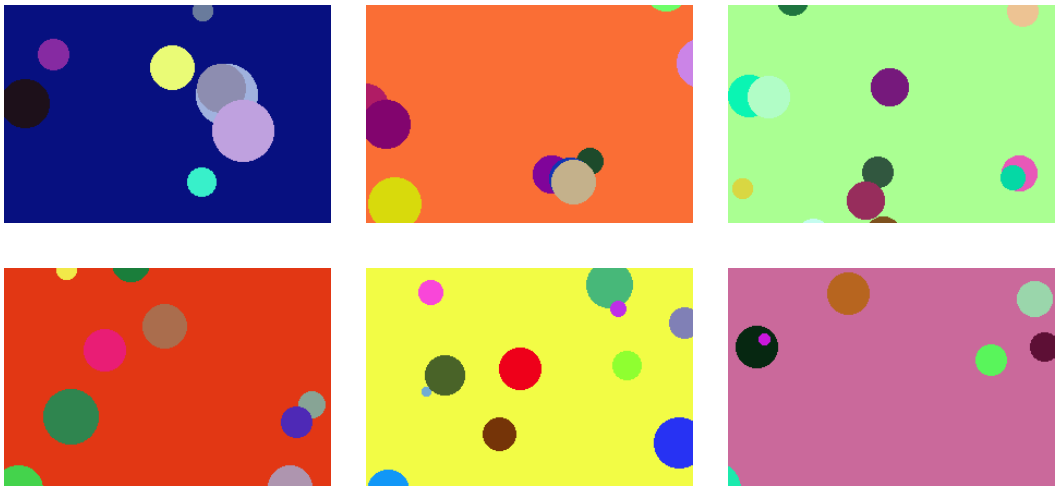
To generate the image dataset, a number of videos were randomly assigned to either the training or the test set. Example frames taken from the training and test set videos are shown in Figure 4.3(a) and (b), respectively, while Figure A.1 in Appendix A contains example frames from all the training and test set videos used in the experiments.

The training set of images was created from 10 000 subwindows sampled in equal numbers from 20 training videos, while the test set contains 5000 subwindows extracted from 10 test videos. The 20 background colours present in the training set and the 10 background colours of the test set are different in order to ensure substantial dissimilarity between training and test examples. The extracted subwindows were 64×64(×3) RGB images, resulting in an input data dimension of 12 288.

[1] In a different terminology, this task is called circle localisation.



(a) Training video frames



(b) Test video frames

Figure 4.3 Example frames of the (a) training and (b) test videos, from which sub-windows were extracted in order to generate (a) the training and (b) the test set of images. The snapshots show circles of different sizes and colours against a uniform background. Note the presence of overlapping circles. The sets of background colours used in the training and the test videos are mutually exclusive.

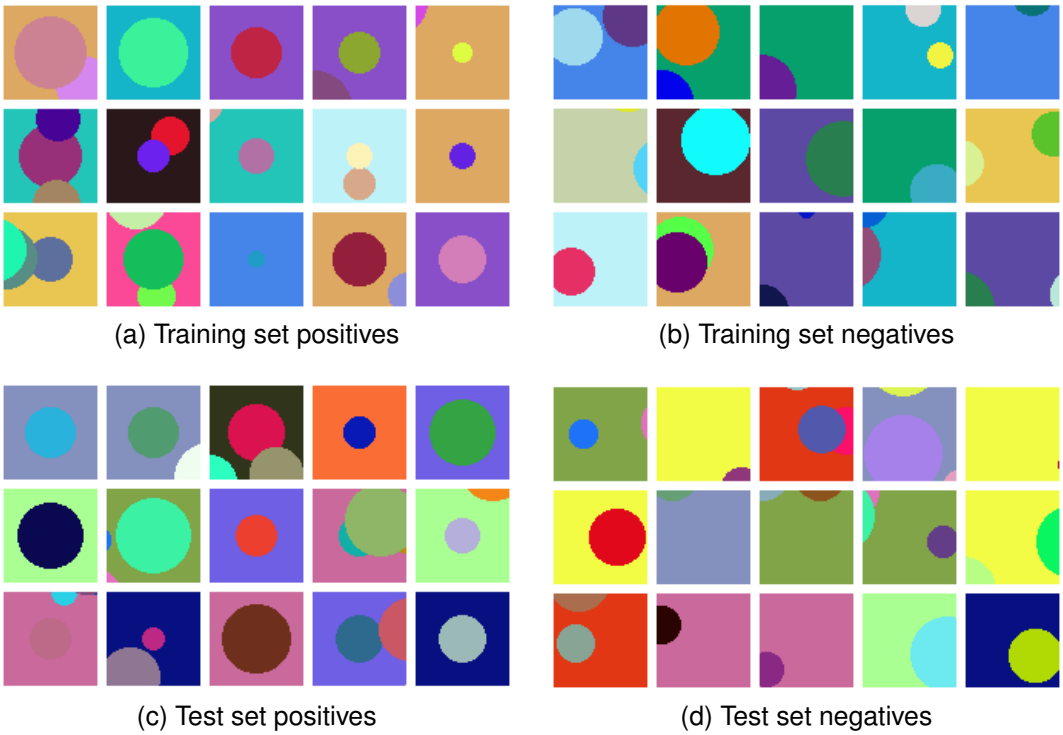


Figure 4.4 Examples of subwindows from the (a)–(b) training and (c)–(d) test set, representing the (a) and (c) positive and (b) and (d) negative classes of the circle detection task. As opposed to negative class images, examples of the positive class have a circle positioned in the exact centre of the image. Images can contain circles that overlap and ones that have highly similar colour to the background.

Examples of the positive class, which contains images with a circle positioned in the exact centre, are shown in Figures 4.4(a) and (c) for the training and test set, respectively, while negative class examples (i.e. images without a centre circle) of the training and test set are shown in Figures 4.4(b) and (d). Additional positive and negative class examples of the test set can be seen in Figure A.2 in Appendix A and in the first rows of Figures 4.9(a) and (b).

4.4.4 Advantages of Simulated Data

Experiments in this chapter utilise synthetic data and automatically generated labels to learn a model of the retina and early vision. As a different approach,

electrophysiological data can also be used for retinal modelling: see, e.g., Nirenberg and Pandarinath's (2012) method discussed in Section 3.4.4.1 of Chapter 3. Yet a different alternative is explored in Chapter 6, which presents my proposed hierarchical LRF-DNN model of visual information processing trained on images of faces taken with heterogeneous camera settings.

As opposed to camera-generated natural images or electrophysiological data from animal experiments, the simulated photoreceptor input and the corresponding class labels can be obtained in unlimited quantity with no cost and provide the advantage of having good control over the quality of the data.

The SPI data and task provides an easy-to-control simulated environment for comparing methods on a circle detection task, which approximates the detection of different coloured spots in retinal input received by photoreceptors. Consequently, the primary goal of the following experiments is to show that even when using a simulated environment during training, deep networks can learn feature detectors similar to those implemented by retinal ganglion cells and can execute key early visual processing tasks.

An alternative type of input could be constructed from natural images; however, in this case labels would not be readily available. Another advantage of utilising synthetic datasets is having better control over the data quality. If natural images were used, one would have to ensure that the recording conditions of the footage make it adequate for modelling retinal input. This is far from straightforward, as the kinds of noise and artefacts typically present in natural images are not necessary the same as those a retina has to deal with.

During the training of deep learning methods, particularly ConvNets, synthetic data are often used for complementing or replacing other data sources (see, e.g., Dosovitskiy et al., 2015; Rozantsev et al., 2015; Su et al., 2015). Deep

learning methods have been found to make great use of synthetic data and state-of-the-art computer vision systems routinely apply such datasets and data augmentation in the form of automatically generated or transformed images.

Most importantly, however, a key advantage of deep learning, especially unsupervised methods, is the ease of transferring the training algorithm to other types of data. As the features are learnt directly from input, the proposed models can be adapted in a straightforward manner with the aim of learning retinal features from suitable camera-generated images (e.g. patches of high resolution natural images) or even from substantially different types of data, such as recordings from clinical experiments.

4.5 Methods

As outlined in the previous sections, I propose to use DBNs and DNNs (pretrained as DBNs) for modelling the earliest stages of visual processing and test the performance of these models on a circular spot detection task using the SPI dataset.

Although DBNs with binary visible nodes are quicker to train and finding suitable learning parameters for them is easier, when the data are continuous valued, in most cases, binary visible nodes are inadequate and DBNs with Gaussian visible nodes can provide better models. As the SPI dataset is continuous valued, my proposed models use Gaussian visible nodes and the hidden nodes are binary.

In the following experiments, unsupervised DBN and classifier DNN models were trained as described in Section 2.4.1 and Section 2.4.2 of Chapter 2, respectively, according to the method of Hinton et al. (2006). RBMs were used to pretrain each consecutive layer of the networks, and classifier DNNs were sub-

sequently fine-tuned in a supervised manner on the circle detection task, a two class classification problem, using backpropagation. For a diagram illustrating a DBN generative model trained on SPI, see Figure 4.1(b).

4.6 Experimental Set-Up

The following sections summarise the details of training the proposed early vision models as well as the quantitative and qualitative evaluation measures used.

4.6.1 Training Protocol

Experiments were conducted on the 'static' image dataset of SPI, which contains subwindows extracted from the video dataset. Tests were conducted to evaluate a number of different DBN and DNN parameter choices, including different settings for the number of training iterations and the learning rate. From the trained models, the top performing ones were selected for discussion here. Settings used for training parameters which are not detailed below are the same as those provided by (Hinton and Salakhutdinov, 2006a,b).

DBN architectures with hidden node numbers of 100, 500, and 2000 were examined, and the network depth ranged between 1 to 5 hidden layers. The number of visible nodes on the first layer was given by the dimension of the input data, namely 12 288.

Deep networks pretrained as generative models usually require larger numbers of hidden nodes than models trained purely discriminatively (Bengio, 2012). In agreement with the architectural choices described above, Hinton and Salakhutdinov's (2006a) 3-hidden-layer DBN-based digit classifier also uses 500- and 2000-sized hidden layers on a dataset containing 60 000 training examples (6 times as many as SPI), albeit with lower dimensionality (784). Despite the larger dimen-

sion of SPI images, a need for greatly increased layer sizes was not expected as SPI images can often contain highly correlated pixels. As predicted, my experiments showed that networks with 500 nodes per hidden layer were already capable of learning good models of SPI images, and increasing the layer size required undue additional training time.

Pretraining DBNs were first trained unsupervised in a layer-by-layer manner using RBMs, in order to extract key features from the image data. RBMs were trained with CD_1 on mini-batches of size 100 for 10 000 iterations using the whole of the training set in each iteration. Training was conducted with a learning rate of 0.005 on the first hidden layer and 0.1 on consecutive layers. An initial momentum of 0.5 was used for 5 epoch, which was then changed to 0.9. On the first layer of each DBN, an RBM with Gaussian visible nodes and binary hidden nodes was used for pretraining, while on consecutive layers, RBMs with binary visible and hidden nodes were used.

Fine-Tuning The features learnt by each DBN were subsequently fine-tuned during the supervised training phase in order to classify images into two classes based on the presence of a centre circle. For this, first an output layer was added to enable class label predictions; then, the multi-layer network was fine-tuned using 200 iterations of backpropagation, minimising the cross-entropy error (as described in Section 2.4.2 in Chapter 2).

4.6.2 Evaluation Protocol

The following sections provide qualitative and quantitative analyses of the proposed DBN- and DNN-based retinal models. The DBN models are evaluated by visualising the feature hierarchy learnt by the multi-layer network and examin-

ing how the generative model can produce new input data as well as how it can encode and reconstruct test examples. The performance of fine-tuned classifier DNNs is measured by calculating the precision, recall, and F-measure scores on the classification task described above.

4.6.2.1 Learnt Features

When a deep network is trained on image data, visualising features learnt on the first hidden layer is straightforward as the in-going weights of a hidden node can be written in the form of the original input image data. Features present on consecutive layers of a deep network can be visualised by an approximate method described in Section 2.6 in Chapter 2 using linear combinations of features on the previous layer to compose the visualisation of a higher-layer feature.

Using these methods, visualisations of randomly sampled features will be shown for each layer of a DBN trained unsupervised on the SPI dataset. The process of obtaining visualisations for higher-layer features through composition of previous-layer features will also be demonstrated. To this end, randomly selected features from a higher layer will be displayed together with those lower-layer features that have the strongest connection in magnitude to the given higher-layer feature. Feature compositions on the output layer of a fine-tuned classifier DNN will be examined in order to determine which features are the most important for predicting class labels.

It is important to emphasise, due to the non-linearities on each hidden layer, the feature visualisation technique used on higher layers can only provide an approximation of the implemented function. Consequently, the visualisations shown in Section 4.7.1 cannot fully communicate the complex functionalities of non-linear features at the top layers.

4.6.2.2 Generative Model

DBNs trained in an unsupervised manner using RBMs provide a multi-layer generative model of the input data. Newly constructed input can be obtained by first running alternating Gibbs sampling between the top two layers of the network for a large number of steps. Subsequently, a down pass is performed through the network layers using the generative weights to calculate probabilities, according to Equation (2.6) on higher layers and Equation (2.10) on the bottom (visible) layer.

Alternatively, generation can be started from an initialisation of the network activations obtained by feeding in an input image from the dataset, where hidden node activations are calculated using Equation (2.9) on the first hidden layer and Equation (2.5) on consecutive layers. This is followed by alternating Gibbs sampling between the top two layers—with, in some cases, added noise (see, e.g., Taylor et al., 2006)—to gradually diverge from the example image.

Section 4.7.2 will examine examples of new input images generated by a DBN model trained on the SPI dataset.

4.6.2.3 Reconstruction

As a further means of analysis, the ability of unsupervised DBNs to encode and reconstruct previously unseen images will be evaluated qualitatively by examining the reconstruction of positive and negative class examples randomly selected from the test set.

A reduced-dimensional encoding of an input image can be obtained in the top layer by first initialising the visible node activations using the input, then calculating activations with a bottom-up pass through the network layers. From such an encoding, a reconstruction can be obtained in the visible layer by calculating

top-down activations through the network.

4.6.2.4 Classification

DNN classifiers, fine-tuned on the circle detection task in a supervised manner using backpropagation, will be evaluated based on the accuracy of classification on the unseen test set. The change in classification performance during the epochs (or iterations) of backpropagation is measured by calculating the precision (P), recall (C), and F-measure (F) scores on the test set. Let tp denote the number of true positives, fp false positives, tn true negatives, and fn false negatives, then the precision is defined as:

$$P = \frac{tp}{tp + fp}\,, \tag{4.1}$$

the recall is:

$$C = \frac{tp}{tp + fn}\,, \tag{4.2}$$

and the F-measure is given by:

$$F = \frac{2PC}{P + C}\,. \tag{4.3}$$

The precision provides the fraction of true positives out of all positive class predictions made by the classifier. The recall gives the fraction of positive class examples the classifier is able to retrieve. Finally, the F-measure is defined as the harmonic mean of the precision and recall values and provides a single measurement of the detection system's performance (with 1 constituting the best achievable score).

4.7 Results

Experiments presented in this chapter investigate the capability of DBNs and DNNs to discover features similar to retinal feature detectors and to learn retinal functions without resorting to hard-wired circuits. For this analysis, visualising feature detectors learnt by the deep network models and evaluating classification performance on the circle detection task are equally important.

4.7.1 Learnt Features

Figure 4.5 shows random samples of features learnt unsupervised by RBMs on consecutive layers of a DBN, which had 5 hidden layers with 500 nodes on each hidden layer. These feature visualisations reveal the success of the network in learning a variety of functions typical of early visual processing units within the retina, LGN, and the visual cortex. For example, DoG filters—common models of retinal ganglion cells and LGN cells (see Figure 3.2 in Chapter 3)—and Gabor filters (see Figure 4.2), which approximate V1 simple cell receptive fields, have their analogues within the learnt features (both monochrome and colour variants).

Fine-tuning uses backpropagation and therefore commits only minor changes to the weights in lower layers. Instead of making major alterations to feature detectors learnt during the unsupervised phase, the primary goal of fine-tuning is to optimise the weights of the output layer and thereby provide a way of combining high-level features to make classification predictions.

Figures 4.6 and 4.7 show features after the fine-tuning stage in a DNN with 3 hidden layers, each containing 500 nodes. In each plot, the first column shows features randomly chosen from a higher layer, while in the consecutive columns,

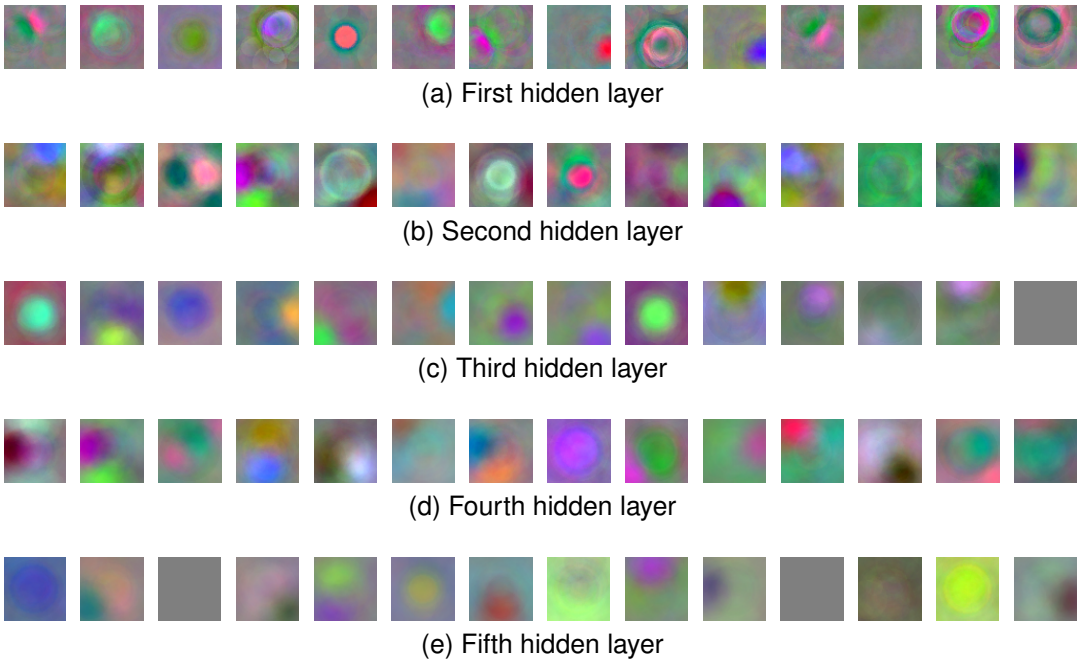
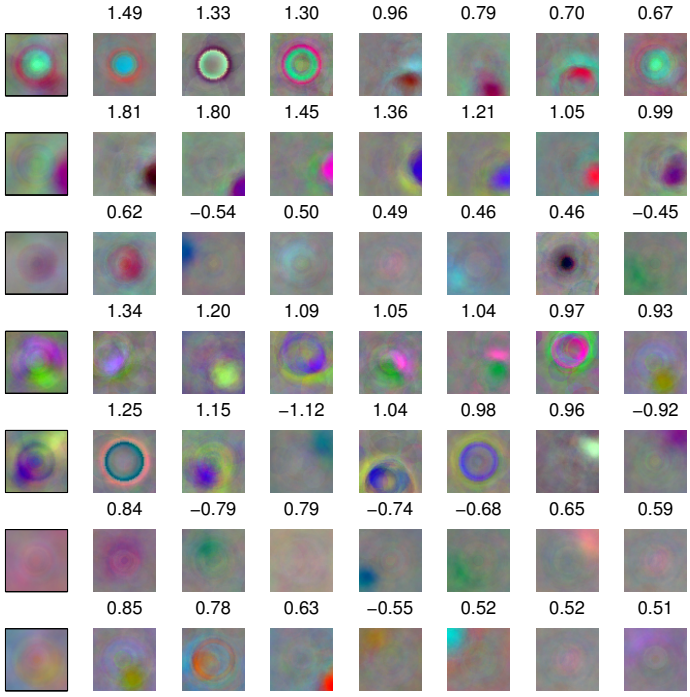


Figure 4.5 Random samples of features learnt on consecutive layers during the unsupervised pretraining phase in a DBN with 5 hidden layers (500 nodes per hidden layer). Features on higher layers are visualised by linear combinations of previous-layer features. The learnt features contain a number of DoG detectors, Gabor-like filters, and Gaussian derivatives.

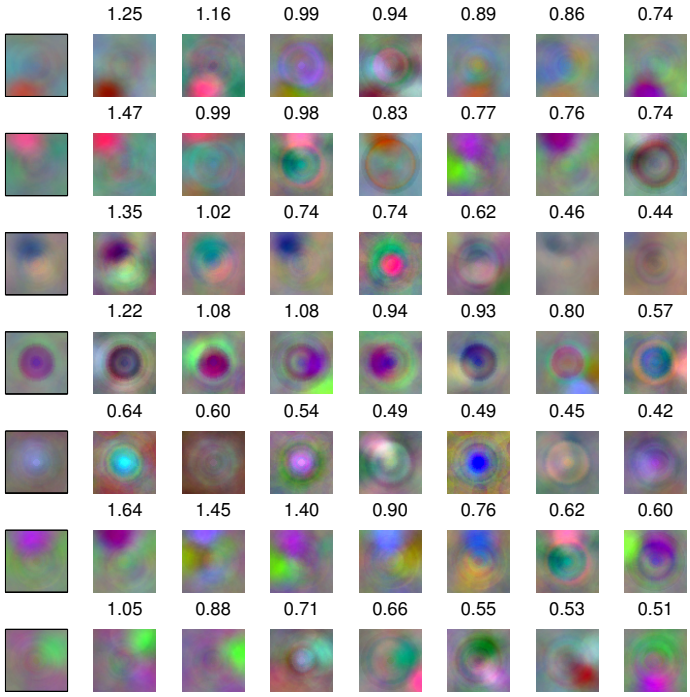
those features from the previous layer are displayed which have the strongest connections to the corresponding higher-level feature. The connection weight can be seen above each feature.

Consistent with the previous plot, the majority of feature detectors shown in Figure 4.6 exhibit similarity to retinal ganglion cells (and LGN cells) with easily recognisable centre-surround receptive fields, while the rest contains examples of Gaussian derivatives and detectors similar to Gabor filters.

Figure 4.7 shows those feature detectors from the highest hidden layer which possess the strongest connections to output nodes. As this ensemble consists of well-defined DoG and Gabor-like features, it can be concluded that the network assigns highest importance to these types of detectors in making classification



(a) Second and first hidden layers



(b) Third and second hidden layers

Figure 4.6 Visualisations show how (a) second- and (b) third-layer features in a DNN are composed of (a) first- and (b) second-layer features, respectively. The first column contains higher-layer features and consecutive images in each row show the strongest connected features from the previous layer together with their weights. Note RBMs on each layer often group features with similar appearance to form a higher-level feature.

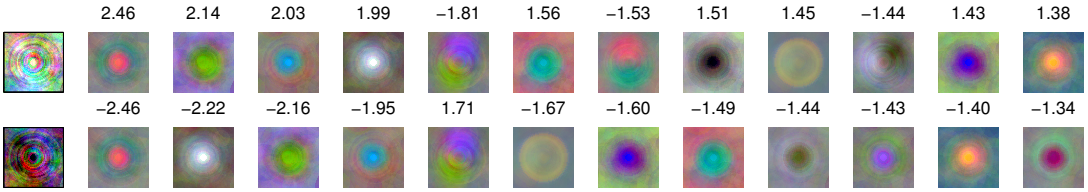


Figure 4.7 Visualisation of the output layer and its connections to the highest, third-layer features in a DNN classifier. The first image in each row shows an output layer node, with the top row corresponding to the positive class (images with circle centred in the middle) and the bottom row to the negative class. The consecutive images in each row visualise features from the previous layer that have the strongest connections to the given output node (weight shown above). The displayed features contribute strongly to the classifier's predictions. Note the dominance of various DoG filters and examples of Gabor-like filters.

predictions.

Results shown in Figures 4.6 and 4.7 thereby demonstrate that, on the SPI dataset and the classification task of circle detection, the network can successfully learn feature detectors which mimic functionalities implemented in the early visual system. It is important to note an algorithm would not necessarily have to apply DoG filters in order to solve this classification task, as circles can be detected using different types of features. The fact that the deep networks trained here learnt DoG filters to execute this task is therefore an important discovery.

Although the resulting networks implement similar functionalities to retinal ganglion cells, the underlying mechanisms, such as the network structure, may exhibit differences. This is because specific connections between retinal cells are not hard-coded into the algorithm but, instead, learnt in a primarily unsupervised fashion.

Chapters 5 and 6 will explore how the compositional behaviour of a deep network, whereby higher-level features have a more complex structure than lower-level ones, can be improved using a novel adaptation of the training method and architectural constraints.

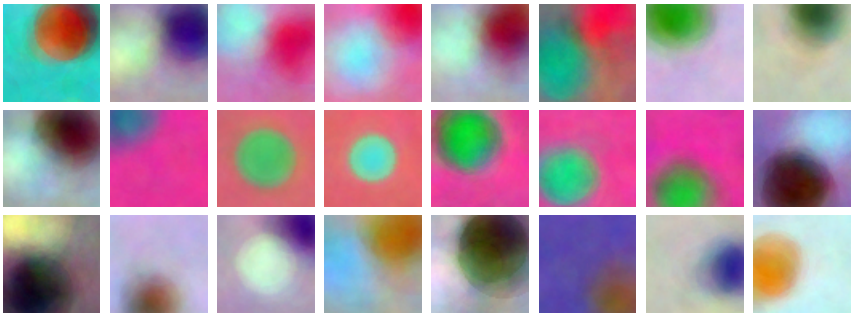


Figure 4.8 New instances of data produced by a DBN generative model trained on SPI images.

4.7.2 Generative Model

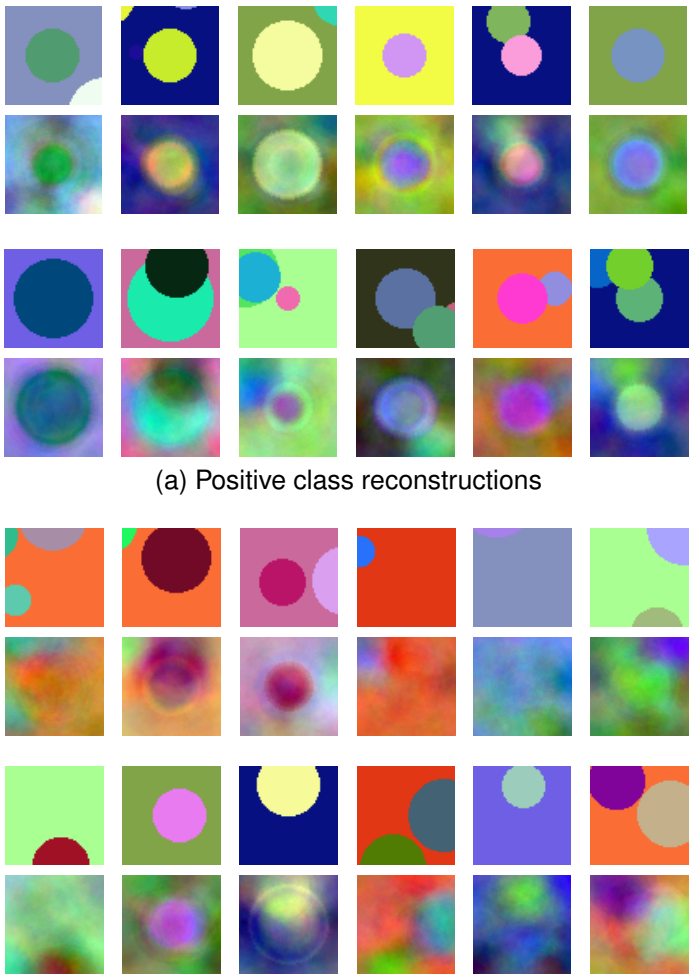
The generative capabilities of a DBN trained on SPI are analysed in Figure 4.8, which shows samples of new data generated automatically by the DBN after the unsupervised layer-wise pretraining phase. The network contained 5 hidden layers with 500 nodes per hidden layer.

These network 'fantasies' consist of images with different background colours typically showing one or two circles which can overlap. Both positive and negative class examples are represented among the generated data, since some of the images contain a circle positioned in the centre while others do not.

4.7.3 Reconstruction

An evaluation of the dimensionality reduction performance of a DBN trained on SPI is provided in Figure 4.9. To this end, randomly selected examples of the test set are displayed in the first row of each plot followed by their reconstructions in the consecutive row. Figure 4.9 (a) shows positive and (b) negative class images.

Reconstructions were generated by a 5-hidden-layer DBN with 500 nodes on each hidden layer after the unsupervised layer-wise pretraining phase. This means features of the input images were encoded with a 500-length code in the



(a) Positive class reconstructions

(b) Negative class reconstructions

Figure 4.9 Randomly selected examples of the test set are displayed in the top row of each plot, and their reconstructions generated by a DBN with 5 hidden layers is shown in the consecutive rows. Key features of images are retained, even though reconstructions were calculated from a limited-length (500) code. Reconstructions of positive (i.e. images with circle centred in the middle) and negative class examples (i.e. images without a circle centred in the middle) are shown in (a) and (b), respectively.

top layer, from which reconstructions were calculated by a top-down pass. It is important to emphasise these results were obtained without specifically fine-tuning the model for the optimisation of reconstruction performance.

As can be seen from the reconstructed images, the examined DBN model learnt to encode the input data with a limited number of nodes in such a way

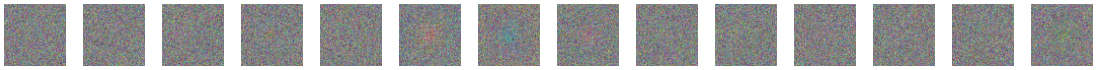


Figure 4.10 Example features of a neural network, containing a single hidden layer with 500 hidden nodes, are visualised. The network was trained with backpropagation for 200 iterations on the SPI dataset without pretraining. The features with the highest L2 norms are shown. Note that these features are noisy and less distinctive compared to the previously shown DBN and DNN features obtained with the use of pretraining.

that key features of even previously unseen test examples are retained in the reconstructions.

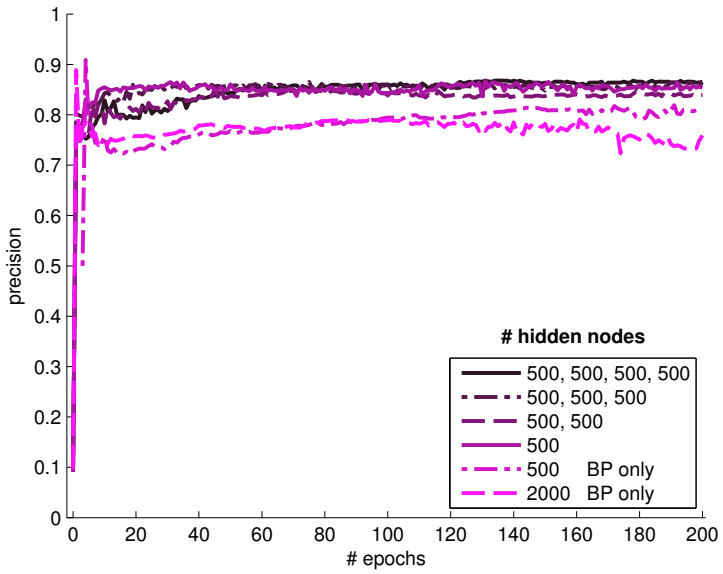
4.7.4 Classification

The precision, recall, and F-measure values achieved on the test set by DNNs with different numbers of hidden layers are shown in Figures 4.11 and 4.12. For comparison, results obtained by single-hidden-layer neural networks trained either with or without the use of pretraining are also displayed. The performance measures are shown as a function of the number of backpropagation epochs.

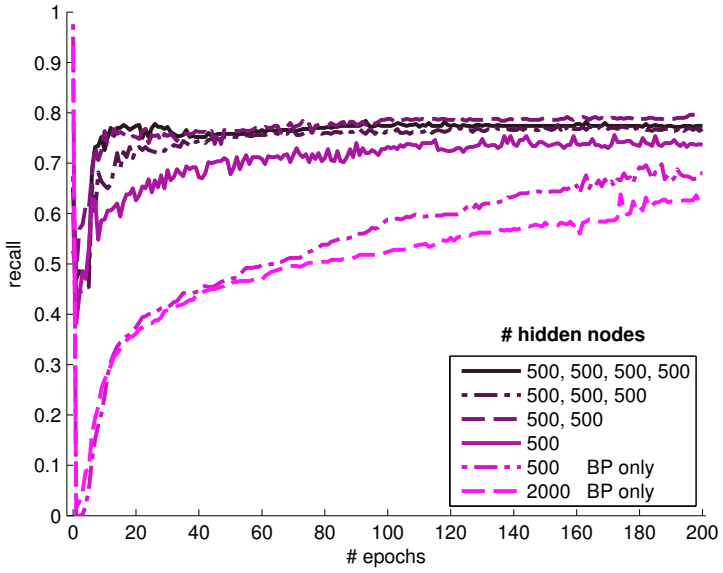
As can be seen, the two networks trained using backpropagation without pretraining perform poorly compared to any one of the pretrained networks. Also, the features learnt by networks without pretraining were noisy and much less distinctive. Example features are shown in Figure 4.10 (the rest are similar in nature). Consequently, this analysis has confirmed unsupervised layer-wise pretraining of neural networks can significantly improve classification results.

Furthermore, the classification experiments also contrasted the performance of single-hidden-layer neural networks with DNNs containing 2–4 hidden layers. As expected, superior results were obtained by multi-layered networks compared to shallow networks, with the 4-hidden-layer DNN showing the best overall performance and achieving the highest F-measure scores.

It has been shown, if certain conditions are met, additional hidden layers



(a) Precision



(b) Recall

Figure 4.11 Precision and recall. The change in (a) precision and (b) recall during the backpropagation epochs is shown on the unseen test set. Networks contained between 1–4 hidden layers, and each hidden layer had 500 hidden nodes, except for the last network containing 2000 nodes on a single hidden layer. Hidden node counts of network layers are shown in the graphs. In the first four cases, the networks were pretrained using RBMs before the start of backpropagation, as opposed to the last two cases (500 and 2000 BP only), where backpropagation alone was used for training. Pretrained DNNs significantly outperform single-hidden-layer networks.

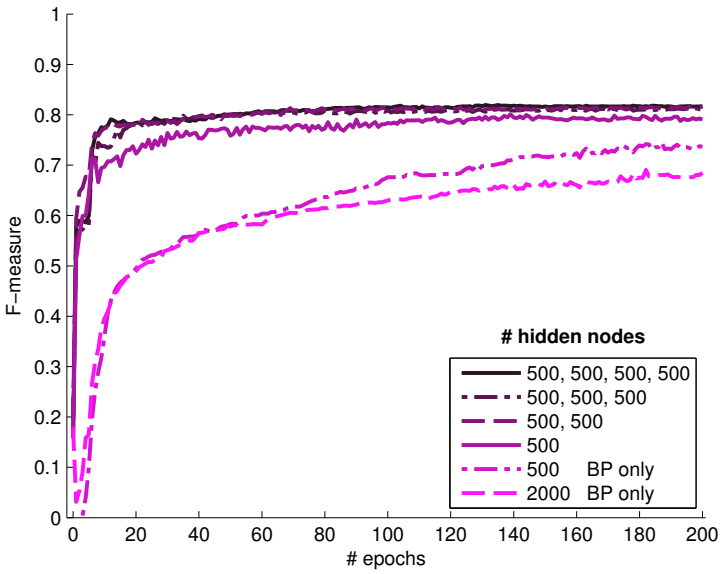


Figure 4.12 F-measure. The change in F-measure score during the backpropagation epochs is shown on the unseen test set. Networks contained between 1–4 hidden layers, and each hidden layer had 500 hidden nodes, except for the last network containing 2000 hidden nodes on a single hidden layer. In the first four cases, the networks had been pretrained using RBMs before the start of backpropagation, as opposed to the last two cases (500 and 2000 BP only), where only backpropagation was used for training. The F-measure provides a single value measurement of performance and confirms that pretrained DNNs outperform shallow networks, with the 4-hidden-layer DNN achieving the best scores.

improve a DBN generative model (Hinton et al., 2006). The strength of utilising multi-layered generative models to pretrain DNNs is revealed when examining the graphs in Figures 4.11 and 4.12: deep networks with at least 2 hidden layers can achieve high precision and recall after only a few backpropagation epochs. This shows the unsupervised pretraining phase initialises the weights of the network to a favourable range and therefore fewer epochs of backpropagation are sufficient to guarantee good classification performance.

4.8 Summary

In accordance with the flexible probabilistic and data-driven approaches outlined in Chapter 3, this chapter introduced novel DBN- and DNN-based models of the retina and early vision. These deep learning models do not rely on hard-wired circuitries to model the retinal network or the visual cortex; instead, learning of retinal and V1 functionalities is driven by data. Furthermore, it was emphasised that a remarkable structural resemblance exists between the inherently multi-layered retinal network and deep network models.

A new simulated photoreceptor input dataset and task have been constructed in order to provide an easy-to-control environment for the modelling of early vision functionalities. On this dataset, first unsupervised DBNs were trained and their generative and reconstruction capacities were analysed. Subsequently, supervised DNN models were evaluated with respect to their classification performance on a circle detection task inspired by early visual processing functions. Analysis revealed multi-layer models achieved superior classification results compared to shallow models, pretraining of networks was important for ensuring great performance, and the DNN containing the highest number of hidden layers produced the overall best results.

The analysis also examined features learnt automatically by the networks from simulated photoreceptor input in a primarily unsupervised fashion, with additional supervised fine-tuning. The models have successfully learnt a variety of feature detectors resembling DoG filters and also a number of Gabor-like filters. In the literature, DoG filters are the most commonly used models of retinal ganglion and LGN cells with centre-surround receptive fields. While Gabor functions are well-known models of V1 simple cell receptive fields.

The quantitative and qualitative analyses have thereby confirmed DBNs and DNNs do not only bear structural similarly to the retinal network but are capable of learning functionality of retinal, LGN, and V1 neural structures. Consequently, such methods are highly suitable for modelling early stages of visual processing.

Chapter 5 will introduce my proposed extensions of these models: the local receptive field constrained DBN and DNN. These models show improved compositional behaviour, whereby higher-level features have more complex structures and larger scopes than lower-level ones, and implement a mechanism for learning advantageous network structure from data.

5 Learning Local Receptive Fields in Deep Networks

This chapter introduces a new biologically inspired RBM model and training algorithm, the LRF-RBM. This unsupervised method is capable of simultaneously learning local features and an advantageous non-uniform feature detector placement. Furthermore, by utilising LRF-RBMs, novel deep network architectures are developed (LRF-DBNs and LRF-DNNs), which more closely resemble neural structures involved in visual information processing compared to traditional models. The LRF-RBM and the deep learning models described in this chapter have first been proposed and evaluated by Turcsany and Bargiela (2014) and Turcsany et al. (2016).

5.1 Motivation

As seen in Chapters 2 and 4, deep learning methods, such as DBNs and DNNs, are powerful tools for learning a hierarchy of feature detectors and thereby providing improved models of the input data. Deep learning systems have achieved state-of-the-art results on a number of challenging tasks, which underlines the importance of developing such flexible models.

Taking inspiration from neural information processing has proven highly beneficial for boosting the performance of machine learning methods. For example,

the very concept of deep learning has been inspired by the multi-layer organisation of the cortex.[1]

While the focus of deep learning research has not been the exact replication of neural connectivity patterns in the brain, DBNs have been shown suitable for modelling visual information processing systems on a more abstract level (Lee et al., 2008; Turcsany et al., 2014). Developing deep architectures that show greater similarity to biological neural networks, while retaining their flexibility and efficacy on visual tasks, is a crucial step towards improved computational modelling of information processing units within the visual pathway.

Consequently, increasing the fidelity of representations learnt by DBNs and DNNs to encodings extracted by visual processing neurons could provide benefits for numerous areas of visual information processing research, including both biological vision modelling and computer vision. My proposed models presented in this chapter increase the structural similarity of deep network models to neural networks of the visual pathway, which, I argue, represents a key step towards this goal.

Receptive Fields Visual neurons in consecutive processing layers typically only receive information from neurons in a local area of restricted size within the previous layer. The receptive field (Sherrington, 1906) of a neuron refers to the area of visual space within which a stimulus can result in a neural response. Receptive fields are highly local in the case of retinal cells, and receptive field size gradually increases from local to more global through different areas of the visual cortex. A key goal of my research was the extension of deep neural networks with local receptive fields in a way that the training process, the final architecture, and

[1]For a more detailed description of similarities between deep learning algorithms and information processing in the brain, refer to Section 2.2 in Chapter 2.

the encoding process, by which representations are calculated, closely resemble biological neural networks of the visual pathway.

5.2 The Proposed Methods

My proposed deep learning models and training methods implement local receptive fields within deep belief and deep neural network architectures. These methods have been described by (Turcsany and Bargiela, 2014) and (Turcsany et al., 2016).

LRF-RBM Turcsany and Bargiela (2014) introduced a new type of restricted Boltzmann machine[2] (RBM) model, the *local receptive field constrained RBM* (LRF-RBM), together with a suitable training algorithm. The LRF-RBM training procedure allows for seamless integration of local receptive field constraints into RBMs. Also proposed is a method for concurrently finding advantageous receptive field placement while training the LRF-RBM. This enables the network to utilise a non-uniform distribution of feature detectors over the visual space.

LRF-DBN & LRF-DNN In a subsequent work, Turcsany et al. (2016) described how deep network models can be constructed by stacking LRF-RBMs and proposed the *local receptive field constrained deep belief network* (LRF-DBN) and *local receptive field constrained deep neural network* (LRF-DNN) models. The training method makes use of the DBN training principles (Hinton et al., 2006) described in Chapter 2, i.e. greedy layer-by-layer pretraining, followed by fine-tuning; however, RBMs are replaced with LRF-RBMs. By utilising LRF-RBMs during pretraining, the proposed method results in a deep network which more

[2]For an introduction to RBMs, see Section 2.3 in Chapter 2

closely resembles biological processing and provides an improved encoding of image features.

5.2.1 Contributions

The rest of this chapter summarises related work and provides a detailed description of my proposed novel methods and contributions, which include:

(i) an adaptation of contrastive divergence learning (CD) (Hinton, 2002) that introduces local receptive field constraints for hidden nodes of RBMs, thereby enabling the training of LRF-RBMs;

(ii) a method for automatically identifying locations of high importance within the visual input space during LRF-RBM training, and utilising these locations as receptive field centres to obtain a compact, yet powerful, encoding of visual features;

(iii) showing that, by using LRF-RBMs with gradually larger Gaussian receptive field constraints for pretraining each consecutive DBN layer, a biologically inspired local receptive field constrained DBN can be constructed, where consecutive layers detect features of increasing size and complexity;

(iv) demonstrating the superior performance of this LRF-DBN generative model compared to DBNs on face completion, a task inspired by processing capabilities of higher areas in the visual cortex (Chen et al., 2010);

(v) furthermore, showing that supervised fine-tuning can be implemented for LRF-DBNs analogously to DBNs, and the resulting LRF-DNNs outperform traditional DNNs (fine-tuned DBNs) on dimensionality reduction of face images.

I have chosen biologically inspired Gaussian receptive field constraints to encourage the learning of local features. These constraints are implemented in the forms of Gaussian masks over the network weights, when the weights are written in the shape of the input image data. Training is conducted in such a way that the learnt features will not be restricted to have Gaussian shape; rather, the Gaussian constraints encourage the growth of localised features as opposed to global ones.

5.3 Related Work

The following sections introduce related methods and at the same time highlight some important distinguishing properties of LRF-RBMs and LRF-DNNs.

5.3.1 Convolutional Networks

Local receptive fields have been modelled in deep learning through convolutional neural networks (ConvNets) proposed by LeCun et al. (1998). Since then, ConvNets have been widely applied within machine learning and computer vision (see, e.g., Cireşan et al., 2011b; Huang et al., 2012; Kavukcuoglu et al., 2010; Krizhevsky et al., 2012; Lee et al., 2009a; Pfister et al., 2014).

Although the architecture of ConvNets was inspired by the organisation of the visual cortex, the prime objective has not been the modelling of biological visual processing. The main emphasis has been on improving the efficiency of learning in multi-layer networks on visual tasks through the use of convolution, thereby scaling up deep learning algorithms to high-dimensional problems.

In contrast with ConvNets, having fixed network architecture and hand-crafted feature detector organisation, my proposed adaptive network model supports the exploration of problem-specific feature detector placement. The layout

of local receptive fields and the connection patterns are learnt automatically from data, making it possible to exploit particularities of the task at hand.

5.3.1.1 Weight-Sharing

In ConvNets weights between visible and hidden layers are the same across all image locations, making the feature detection translation invariant. When weights are shared, the same feature detectors operate on each part of the image within a layer and the training procedure can therefore utilise convolution operations.

Translation invariant feature detection sometimes simplifies the learning task; however, spurious detections can often arise, and in some visual recognition tasks, translation invariance may not be advantageous. In many cases, image or video data have strong priors on the location of certain feature types within the visual space. For example, images of landmarks tend to capture sky in the top area; in portraits, eyes are located above the mouth; also, videos are often taken by a fixed camera (e.g. surveillance videos) and contain a large portion of static background in the peripheral areas. In such cases, keeping the feature detection translation invariant throws away important information particular to the given dataset and task. This can be detrimental to performance due to the possibility of false detections arising. For example, when face recognition is conducted on aligned images, the mouth always appears in the same area; therefore, a positive detection of a mouth elsewhere will be false.

In contrast, the proposed LRF-RBM model does not use weight-sharing and it is, therefore, capable of learning the most relevant feature detectors at any one image location. In this model, hidden nodes can only receive input from visible nodes which fall within their local receptive field, the shape of which is controlled by a Gaussian constraint.

5.3.2 Feature Detector Placement

Some deep architectures have been proposed in previous work where local receptive fields are used without weight-sharing; however, these systems typically apply a fixed grid layout for their rectangular receptive fields (Le et al., 2012; Ranzato et al., 2011; Zhu et al., 2013). Coates and Ng (2011) propose a method to select receptive fields for higher-layer nodes by grouping together similar features from the previous layer, which can be useful in the case of non-image data.

In contrast with these methods, the location of hidden node receptive fields in an LRF-RBM are learnt concurrently with learning the features themselves. Hidden nodes of an LRF-RBM move around the visual space during training to find the best location for their Gaussian receptive field centre. The LRF-DBN training procedure thereby leads to the construction of a deep architecture where the spatial distribution of feature detectors is non-uniform. Also, instead of resorting to a hard-wired feature detector placement, on each layer, an advantageous placement is learnt automatically from data.

5.3.2.1 Feature Hubs

It is often the case for visual data that some image regions express higher variability within the input with multiple distinctive features being present, while other areas are quasi-uniform, therefore, less interesting. By letting the detectors move to locations of interest, dense feature detector coverage can emerge in image regions where the training data have high variation, while more uniform areas will attract fewer detectors. The areas densely covered with feature detectors will be termed *'feature hubs'*.

When following this method of training, the structure of the network keeps morphing throughout the training procedure. This self-adaptation results in

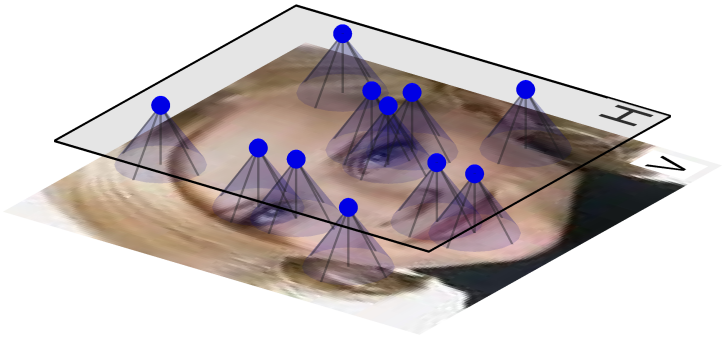


Figure 5.1 LRF-RBM training on face images. LRF-RBM model schematic, showing an input image in the visible layer V and local feature detector receptive fields (blue cones) in the hidden layer H. Feature hubs are located around the eyes and mouth.

a network architecture which is capable of extracting compact representations of visual information, enables very quick query time, and by combining local features as building blocks, the network is strong at reconstructing previously unseen images.

A diagram illustrating the learning of local receptive field locations on face images is shown in Figure 5.1. Furthermore, receptive field distributions in LRF-RBMs after training are provided in Figure 6.1(b) of Chapter 6 and examples of automatically learnt local features in Figure 6.1(c). Figure 6.5 analyses the result of LRF-RBM training on handwritten digit data by showing the spatial distributions of the most distinctive features per digit class.

It is important to note that a non-uniform distribution of feature detectors is also utilised in biological neural networks. As seen in Chapter 3, the structural organisation of the retinal network shows a distinct difference between the centre and the periphery of the visual space. Density of ganglion cells is highest in the centre (fovea), and these midget ganglion cells have very small receptive fields. On the other hand, the periphery contains ganglion cells which connect to a number of bipolar cells and therefore receive input from larger receptive fields (Kolb, 2003; Masland, 2012; Wässle, 2004). This non-uniform organisation

provides high resolution vision in the centre and much lower resolution in the periphery.

5.4 Local Receptive Field Constrained RBMs

Section 2.3 in Chapter 2 provided a description of the RBM model and the CD algorithm, an approximate but highly efficient training method for RBMs. Assuming familiarity with these methods, in the following sections I introduce my LRF-RBM model—a new version of RBMs—together with my proposed training algorithm, which can automatically identify advantageous receptive field centres while learning local feature detectors.

5.4.1 Training with Local Receptive Fields

A visual information processing neuron in early stages of the visual pathway typically only receives input from neurons emplaced within a highly local area of the previous processing layer. The area of the photoreceptor layer in which a stimulus can result in a neural response constitutes the receptive field of a visual processing neuron. Moving up the layers, the receptive fields of neurons become gradually larger and their structures show increasing complexity. For example, as discussed in Chapters 3 and 4, the receptive field structure of retinal ganglion cells can be modelled closely by difference-of-Gaussians, while receptive fields of V1 simple cells by Gabor filters.

5.4.1.1 Receptive Field Masks

In LRF-RBMs, receptive field constraints are applied to hidden nodes in order to outline the spatial area from which the hidden node is most likely to receive input. These constraints are given in the forms of *receptive field masks* (denoted

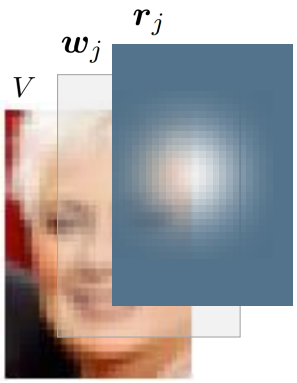


Figure 5.2 Diagram illustrating how the receptive field mask of hidden node h_j is applied in an LRF-RBM. An input image is shown in the visible layer V, the weight vector $\boldsymbol{w}_j$ of h_j is displayed in the shape of the input image, and $\boldsymbol{r}_j$ denotes the Gaussian receptive field mask of h_j. Receptive field mask $\boldsymbol{r}_j$ provides an additional weighting on top of $\boldsymbol{w}_j$. Darker areas in the mask contain near-zero elements, while transparent areas indicate values closer to one.

by R) that operate on the RBM weights W. Figure 5.2 shows an illustration of a receptive field mask when written in the shape of the input images.

Each mask has a centre location which corresponds to a hidden node's location in visual space. An element of the mask describes the likelihood of a connection being present between the underlying visible node and the hidden node, given the distance between the two nodes. The likelihood of a connection converges to 0 as the distance goes to infinity, which means visible nodes further away are less likely to have a connection to the given hidden node.

$R = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_m] \in [0, 1]^{n \times m}$, which denotes the collection of receptive field masks of all the hidden nodes, is of the same dimension as W, with r_{ij} being the receptive field constraint on the connection between visible node v_i and hidden node h_j.

The elements of R represent the likelihood of a given connection being present, implying the viability of a training method which samples connections in each training step. However, learning in this way would be prohibitive on a complex

task. Here, instead, the elements of R will be used as additional weights on top of W. R, thereby, narrows down the scope of hidden nodes to local neighbourhoods.

Note that from the biological modelling point of view, R only provides a constraint or regulariser on the receptive field structure; the actual receptive fields are specified by R and W together. After training, the combination of R and W can show significantly different structures compared to R alone. Still, to keep the description simple, here R will be referred to as the receptive fields. In the following, I will show how the training described in Section 2.3 in Chapter 2 can be adapted for LRF-RBMs.

I conducted experiments showing LRF-RBMs with disk- or square-shaped receptive fields are capable of discovering local features efficiently; however, LRF-RBMs with Gaussian receptive fields learn superior local feature detectors and provide smoother reconstructions with better detail. To keep the computation efficient, Gaussian receptive fields can be truncated. Also, when modelling biological neurons in the early stages of visual processing, Gaussian receptive field constraints are more adequate. Therefore, the description here will focus on the case of Gaussian receptive field constraints. It will also be assumed that a fixed standard deviation (SD) is used for each receptive field within a hidden layer, which will be denoted by σ^{RF}.

5.4.1.2 Modified Contrastive Divergence Learning

Compared to the RBM model, the energy function of an LRF-RBM incorporates the receptive field masks R, and a suitable training algorithm can be obtained by modifying the CD method described in Section 2.3.2 of Chapter 2.

Accordingly, the energy functions provided in Equations (2.1) and (2.8) in Sections 2.3.1 and 2.3.3 and, also, Equations (2.5), (2.6), (2.9) and (2.10) can be

adapted by substituting w_{ij} with $r_{ij}w_{ij}$, where r_{ij} is the receptive field constraint on the connection between v_i and h_j.

Therefore, provided both the visible and the hidden nodes are binary, the probability of a hidden node activation is given by:

$$p(h_j = 1|\boldsymbol{v}) = \frac{1}{1 + exp(-b_j - \sum_i v_i r_{ij} w_{ij})} , \tag{5.1}$$

and visible node states can be sampled according to:

$$p(v_i = 1|\boldsymbol{h}) = \frac{1}{1 + exp(-a_i - \sum_j h_j r_{ij} w_{ij})} . \tag{5.2}$$

In the case of LRF-RBMs with Gaussian visible and binary hidden nodes, the activation of a hidden node can be calculated according to:

$$p(h_j = 1|\boldsymbol{v}) = \frac{1}{1 + exp(-b_j - \sum_i (v_i/\sigma_i) r_{ij} w_{ij})} , \tag{5.3}$$

while the expected value of a visible node is given by:

$$< v_i >_{reconst} = a_i + \sigma_i \sum_j h_j r_{ij} w_{ij} . \tag{5.4}$$

Additionally, the receptive field masks are used to provide a weighting scheme for the weight updates: a connection w_{ij} will receive a stronger update if v_i and h_j are located closer to each other. The modified weight update equation takes the form:

$$\Delta w_{ij} = r_{ij} \epsilon (< v_i h_j >_{data} - < v_i h_j >_{reconst}) . \tag{5.5}$$

5.4.2 Automatically Learning Receptive Field Centres

In a network containing local receptive fields, the layout of hidden nodes can be allocated manually. For example, one could place receptive field centres of hidden nodes at uniform distances from each other in a fixed grid layout, such as in the works of Le et al. (2012), Ranzato et al. (2011) and Zhu et al. (2013). However, such a method does not allow the network architecture to adapt to specific properties of the input data.

Non-uniform feature detector distribution can assist in obtaining compact representations by exploiting spatial patterns in the dataset, such as that aligned faces have key facial features at specific locations and that most natural images have the centre of interest in the middle. When solving a task, some areas of the visual space may need to be represented at better resolution, using multiple different feature detectors, while other areas may not convey much information. In the human visual system, the retina also shows a non-uniform distribution of information processing cells between the centre and the periphery, the former being generally denser. This way, better resolution is obtained in the centre of the visual space.

Accordingly, the LRF-RBM model introduced in this chapter also utilises a non-uniform feature detector distribution. A training method is proposed that allows for the identification of areas in the visual input space which need a higher number of feature detectors in order to obtain a data representation with great descriptive power.

5.4.2.1 Update of Receptive Fields

In the following, my proposed method is described, which enables the learning of advantageous feature detector placements in LRF-RBMs alongside learning the

features themselves. When training the model with modified CD (as described in Section 5.4.1.2), after each complete pass through the training dataset, the receptive fields of hidden nodes undergo an updating process. For each hidden node, this update moves the receptive field centre to the local area which has the strongest connections to the given hidden node and therefore provides the most well-defined feature.

The updating process of receptive field $\boldsymbol{r}_j$, corresponding to hidden node h_j, proceeds as follows:

(i) the weights of the hidden node are written in the shape of the input image data (the resulting weight image is denoted by I^j);

(ii) an element-wise transformation N is applied to I^j;

(iii) values over channels are combined by taking the maximum: $\max_c(N(I^j)_c)$;

(iv) subsequently, the resulting image is filtered using a Gaussian filter $G(\sigma^{RF}, k)$ with SD σ^{RF} and filter size k;

(v) the location with maximum response is selected as the new centre, i.e. the updated location of h_j:

$$(p^j, q^j) = \arg\max_{rs}[\max_c(N(I^j)_c) * G(\sigma^{RF}, k)]_{rs}\ ; \tag{5.6}$$

(vi) finally, the values of the updated receptive field is given by a Gaussian distribution with SD σ^{RF} and mean (p^j, q^j).

I examined element-wise transformations including the identity, absolute value, and squared value, and found the latter two worked similarly well and were superior to identity. The results in Chapter 6 are shown with squared value.

5.5 Local Receptive Field Constrained Deep Networks

As described in Section 2.4 in Chapter 2, DBNs and DNNs can be trained efficiently according to Hinton et al.'s (2006) protocol of first pretraining the network in a greedy layer-by-layer fashion using RBMs, then fine-tuning the network weights by, e.g. backpropagation. Similarly, my proposed training procedure for LRF-DBNs and LRF-DNNs adopts a two-stage protocol.

5.5.1 Pretraining

In line with the DBN training procedure, one can train multiple layers of feature detectors on top of an LRF-RBM hidden layer using either RBMs or LRF-RBMs. This new type of deep learning model will be termed the local receptive field constrained DBN (LRF-DBN). As an example, the biologically inspired LRF-DBN architecture, with which the experiments in Chapter 6 were conducted, utilises LRF-RBMs with increasing receptive field sizes to pretrain consecutive layers of the network. Pretraining of an LRF-DBN starts by:

(i) training an LRF-RBM on the input image data to learn the first hidden layer;

(ii) subsequently, the 2D positions of hidden nodes become fixed, and the receptive field masks (R) will not change further;

(iii) hidden node activations are calculated according to Equation (5.1), in the case of binary visible and hidden nodes, or Equation (5.3), in the case of Gaussian visible and binary hidden nodes; and

(iv) these hidden node activations can subsequently be used as input data to train the consecutive hidden layer of the LRF-DBN.

When traditional RBMs are used for training higher layers of the network, from this point on, the training procedure is the same as in the DBN case described earlier.

If, on the other hand, LRF-RBMs are applied when training consecutive layers, the hidden node activations on the trained layer need to be arranged in 2D space to provide suitable input data to the next LRF-RBM. During this process the locations of hidden nodes, optimally chosen by the LRF-RBM, have to be preserved. The input used for training the next layer with an LRF-RBM is obtained by the following procedure:

(i) for each example image of the dataset, first a single channel zero-valued image of the same width and height as the original input data is constructed;

(ii) the activation of each hidden node is then calculated and added to the value at the fixed location of the given hidden node;

(iii) subsequently, an LRF-RBM with binary visible nodes is trained on this newly constructed input, in the same way as described in Section 5.4 but with the receptive field constraints of the higher layer being applied during training.

In the majority of my experiments presented in Chapter 6, receptive field constraints of consecutive layers are given by Gaussians with gradually increasing standard deviation. Such models are inspired by the hierarchical organisation of the visual cortex, where each consecutive processing layer utilises feature detectors with increasing scope. It is also intuitive to construct models where the lower layers are trained with LRF-RBMs, while the highest layers use traditional unconstrained RBMs. Chapter 6 summarises my experimental analysis with both types of models.

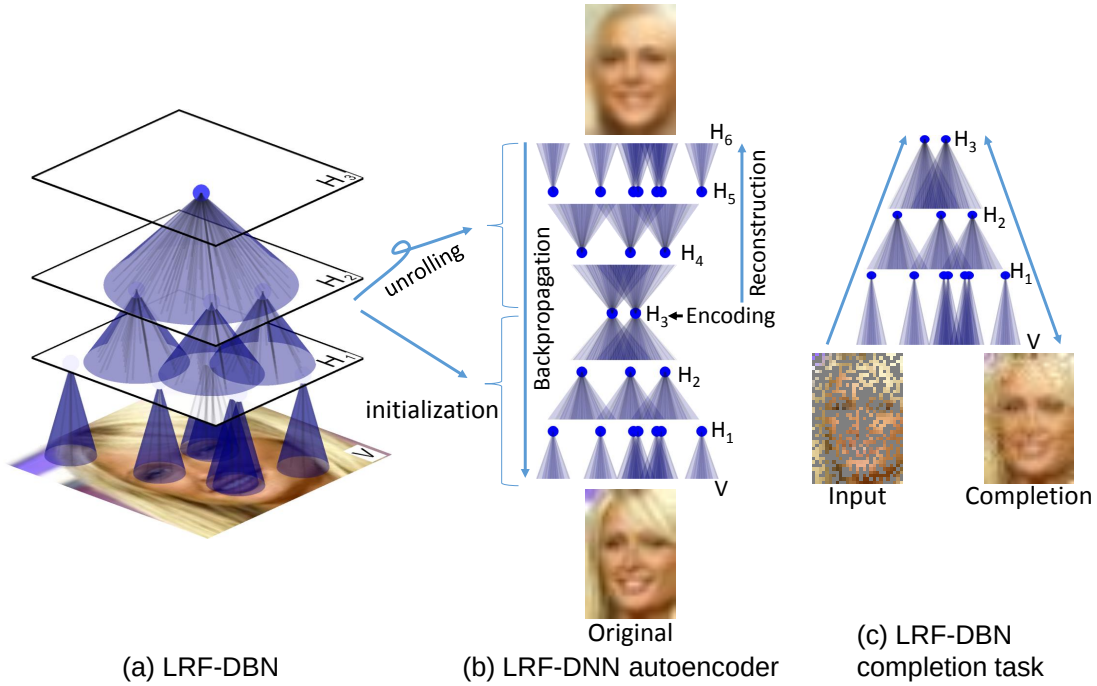


Figure 5.3 (a) Schematic of an LRF-DBN containing a visible (V) and 3 hidden layers (H_1–H_3) with receptive fields of increasing scope (blue cones). (b) Schematic of an LRF-DNN autoencoder used for dimensionality reduction of face images. Prior to supervised fine-tuning with backpropagation, the pretrained LRF-DBN layers are used for initialising the encoder (V–H_3) and, after unrolling, the decoder part (H_3–H_6) of the autoencoder. (c) LRF-DBN used for face completion, where missing pixels of the input are filled in by the generative model through up-down passes in the network.

A schematic diagram of an LRF-DBN model with receptive fields of increasing sizes is shown in Figure 5.3(a), while Figure 5.3(c) illustrates the application of an LRF-DBN generative model on a face completion task, where the generative model is used for filling in missing pixel values in an image.

5.5.2 Fine-Tuning

Once an LRF-DBN has been trained with either RBMs or LRF-RBMs on higher layers, the model can be fine-tuned analogously to DBNs. Multiple types of DBN and DNN models and suitable fine-tuning methods have been introduced in Chapter 2 (for details, refer to Section 2.4). When LRF-DBNs are fine-tuned dis-

criminatively with backpropagation, the resulting neural networks will be called local receptive field constrained DNNs (LRF-DNNs).

In Section 6.4.3 of Chapter 6, experiments conducted with autoencoder LRF-DNNs are evaluated. These results demonstrate the ability of LRF-DNNs to learn a compact reduced-dimensional encoding of image features and generate high-quality reconstructions from these codes.

A schematic diagram illustrating the construction and training procedure of LRF-DNN autoencoders is shown in Figure 5.3(b), where an LRF-DNN is applied to dimensionality reduction of face images. The encoder part of the network is initialised with the pretrained LRF-DBN layers, which are subsequently unrolled to obtain initialisation for the decoder part. Finally, backpropagation is used for minimising the squared reconstruction error.

5.5.3 Notation

In the following, a general notational guide for LRF-DBNs and LRF-DNNs is provided, which will be utilised in Chapter 6.

In the case of DBNs, the number of hidden nodes on consecutive layers will be referred to as the architecture of the DBN and these counts will be used as a notation, while the architecture of an LRF-DBN will be defined by the number of hidden nodes on consecutive layers and also the type of model that was used for training each layer. In the notation, LRF-RBM layers will have (L) attached.

For example, one would denote the architecture of an LRF-DBN that contains 2000, 1000, 500, and 100 hidden nodes on consecutive hidden layers and was trained using LRF-RBMs on the first two and traditional RBMs on the top two hidden layers as 2000(L)-1000(L)-500-100.

If an LRF-DBN or DBN is then fine-tuned as an autoencoder, technically,

the architecture includes the unrolled decoder layers too. However, to keep the description and notation concise, it is convenient to refrain from mentioning the hidden node counts of the decoder when referring to or denoting the architecture of an LRF-DNN or DNN autoencoder. For the same reason, the number of visible nodes will also be omitted from the notation in the following chapters.

5.6 Summary

This chapter introduced my proposed LRF-DBN and LRF-DNN multi-layered models, whose structure has been inspired by neural information processing units in the visual pathway. During LRF-DBN training, my novel LRF-RBM model is used for pretraining individual layers of the network, followed by fine-tuning of the entire network. To train LRF-RBMs unsupervised, a modified version of the CD method has been presented together with a mechanism for automatically learning advantageous feature detector placement from data. Using LRF-RBMs for pretraining enables the deep network to utilise a non-uniform spatial distribution of feature detectors and to learn local feature detectors of increasing scope on each consecutive layer.

6 Deep Network Models of Visual Processing

This chapter evaluates my novel deep learning models, introduced in Chapter 5, on tasks inspired by information processing capabilities of the human visual cortex. LRF-DBNs are tested on face completion, known to be executed by high-level processing areas of the visual cortex. The performance of LRF-DNN autoencoders is measured on dimensionality reduction of image data, as low-dimensional encoding of input data patterns is an essential requisite of efficient information processing. Analysis in this chapter is based on experiments described by Turcsany and Bargiela (2014) and Turcsany et al. (2016).

6.1 Experiments

The following experiments aim at testing the ability of the proposed LRF-RBM and LRF-DBN models to accumulate feature hubs in important areas of the visual space and learn a hierarchy of distinctive local feature detectors. Using the challenging *Labeled Faces in the Wild* (LFW) (Huang et al., 2007) face recognition dataset, I will demonstrate how the LRF-DBN feature hierarchy can be utilised to represent characteristic details of previously unseen face images.

In my first experiments, LRF-DBNs were trained on a training set, assembled from LFW, to learn generative models of the image data. The quality of the

generative models was evaluated on multiple face completion tasks, where the models were required to restore missing pixels in previously unseen test images. Face completion is a complex task implemented in the human brain that involves higher-level processes in the OFA and FFA of the visual cortex (Chen et al., 2010).

Subsequently, LRF-DNN autoencoders were trained to learn a compact low-dimensional encoding of image features, followed by the evaluation of reconstruction performance on an unseen test set. According to the efficient coding hypothesis (Barlow, 1961), the drive to develop compact encoding mechanisms is thought to underlie the evolution of numerous visual information processing functions in the visual pathway.

6.2 Dataset

6.2.1 The 'Labeled Faces in the Wild' Dataset

The LFW[1] dataset contains 13 233 RGB images of public figures, and it is currently one of the most popular face image datasets within computer vision and machine learning. The version of LFW used in my experiments consists of images automatically aligned by deep funnelling (Huang et al., 2012). The dataset has been constructed from an internet-based collection of real-life photos taken of public figures in unconstrained settings. Consequently, one can notice significant variations in appearance and camera settings among the images. Example LFW face images can be seen in the first rows of Figure 6.11 and Figure 6.19.

[1] Available at `http://vis-www.cs.umass.edu/lfw/`

6.2.2 Task

The dataset is commonly used for evaluating methods on face verification, which is predicting whether two face images are taken of the same person, without having previously seen any images of the person(s) during training. RBMs with rectified linear or binary hidden nodes, first trained in an unsupervised manner on single faces and subsequently fine-tuned in a supervised manner on pairs of images, have been shown to achieve good results on this face recognition task (Nair and Hinton, 2010).

Applying supervised fine-tuning methods on pairs of face images lies outside the scope of my work, which focuses on developing novel models of biological visual systems. Therefore, the study presented here will primarily concentrate on investigating the capability of the proposed LRF-RBM, LRF-DBN, and LRF-DNN models to:

(i) learn a feature hierarchy constructed from local feature detectors,

(ii) identify regions of high importance in LFW face images, and

(iii) utilise these as feature hubs to provide compact representations of visual information.

Accordingly, the analysis will compare:

(i) the quality and location of features learnt by LRF-RBMs and RBMs,

(ii) the generative capabilities of LRF-DBN and DBN models, and

(iii) the ability of LRF-DNN and DNN autoencoders to encode and reconstruct previously unseen face images.

6.2.3 Preprocessing

I applied similar preprocessing to LFW images as Nair and Hinton (2010) and trained RBMs with binary hidden nodes on single faces using their published parameter settings. These models were then compared with LRF-RBMs trained on the same data.

The preprocessing involved cropping the central 105×153 pixels part of the original 250×250 RGB images, thereby eliminating much of the background. These background pixels are known to unintentionally provide helpful context for recognition. The cropped images were then subsampled to 27×39(×3), resulting in an input data dimension of 3159.

Finally, the input data were standardised along each component to have zero mean and unit variance, which makes it possible to write $\sigma_i = 1$ in Equations (2.8) to (2.10) in Chapter 2 and Equations (5.3) and (5.4) in Chapter 5, thereby simplifying the training.

A training and test set were formed, with 4000 training and 1700 test examples. The test set did not contain any images of persons present in the training set.

6.2.4 The MNIST Digit Dataset

The MNIST database of handwritten digits (LeCun et al., 1998) is the most commonly used publicly available dataset for handwritten digit recognition. The dataset contains 60 000 training and 10 000 test set examples of the 10 digit classes. Example images are shown in the first row of Figure 6.5.

In addition to the experiments on LFW faces, I also trained LRF-RBMs on MNIST, in order to identify which areas in the handwritten digit images would attract feature detectors and thereby result in the emergence of feature hubs.

6.3 Experimental Set-Up

6.3.1 Training Protocol

My experiments aimed at comparing LRF-RBMs, LRF-DBN generative models, and LRF-DNN autoencoders with traditional RBMs, DBNs, and DNNs as well as evaluating different parameter choices for my proposed methods.

6.3.1.1 RBM and DBN Training

To establish baseline results, I trained fully connected RBM and DBN models on LFW without receptive field constraints according to the training procedure of Nair and Hinton (2010) and Hinton and Salakhutdinov (2006a) (described in Sections 2.3 and 2.4 of Chapter 2).

RBMs were trained with CD_1 on mini-batches of size 100 for 2000 iterations. An optimal learning rate (ϵ) of 0.001 was used during training. Higher learning rates failed. An initial momentum of 0.5 was applied for 5 epoch, which was then changed to 0.9. To extract the first representational layer of a DBN from input data, RBMs with Gaussian visible nodes and binary hidden nodes were used. Subsequent layers were trained with RBMs containing binary visible and hidden nodes. Hidden node numbers of 4000, 2000, 1000, 500, and 100 were tested, and each consecutive layer in a network had a smaller size than the previous layer. Included were all architectures that fitted these requirements.

6.3.1.2 LRF-RBM and LRF-DBN Training

LRF-DBNs were trained using the same settings, except for the learning rate, where $\epsilon = 0.1$ was found optimal for LRF-RBMs on the first layer with Gaussian visible nodes, while $\epsilon = 1$ worked best for LRF-RBMs on subsequent layers with

binary visible nodes. When top layers of an LRF-DBN were trained with RBMs, the learning rate was set to 0.001 on these layers. Both RBMs and LRF-RBMs were found to be able to learn good models within a few hundred iterations, after which performance only slightly improved. In the following, results are displayed for models trained for 2000 iterations.

The analysis focused on LRF-DBNs where higher layers had increasing receptive field sizes. The Gaussian receptive field constraints investigated had σ^{RF} between 0.5 and 13.0, and filter size k ranging from 1 to 19. I also studied LRF-DBNs where the top few layers were trained using RBMs without receptive field constraints. In Section 6.4, if not stated otherwise, the receptive field constraint parameters on n consecutive LRF-RBM layers are given by the first n elements of $\boldsymbol{\sigma^{RF}} = (3, 5, 7, 9, 11)$ and $\boldsymbol{k} = (5, 7, 9, 11, 13)$.

6.3.1.3 Autoencoder Training

Using the weights learnt by LRF-DBN and DBN models for initialisation, LRF-DNN and DNN autoencoder networks were trained in an analogue manner to Hinton and Salakhutdinov (2006a), as described in Sections 2.4.3 and 5.5.2 in Chapters 2 and 5. Backpropagation was conducted for 200 iterations to minimise the squared reconstruction error (SRE) (i.e. the squared distance between the original data and its reconstruction) on the training set. From this definition follows that the lowest achievable SRE is 0, where the reconstruction exactly matches the data, and, in general, there is no upper bound on the SRE. If the components of both the data and the model predictions follow a standard normal distribution, then 4 gives an upper bound on the average SRE per component.

On this dimensionality reduction task, all those architectures mentioned above were used where the size of the top hidden layer, which provides the reduced-

dimensional encoding, was either 100 or 500 nodes.

6.3.2 Evaluation Protocol

The evaluation concentrates on assessing the quality of the LRF-DBN and DBN generative models, contrasting the feature hierarchies learnt by the two types of models, and comparing the reconstruction performance of LRF-DNN and DNN autoencoders.

6.3.2.1 Learnt Features

The spatial distributions of feature detectors were examined and feature hubs were identified in LRF-RBMs used for training either the first or consecutive layers of DBNs. As a visualisation tool, *'centre maps'* were constructed using Parzen-window density estimation with a Gaussian kernel on 2D feature detector locations, in order to identify densely populated areas. Another means of evaluation was given by *'receptive field maps'*: heat maps generated by combining all receptive fields within an LRF-RBM through pixel-wise summation. (In the case of MNIST images, per class receptive field maps were also examined as described in Section 6.4.1.2.) When visualisations are shown (see, e.g., Figure 6.6), both in centre maps and in receptive field maps, darker red areas correspond to larger values, i.e. locations which are more densely covered by feature detectors, while blue areas indicate the presence of fewer detectors.

Feature detectors learnt by the different models were compared using the feature visualisation method described in Section 2.6 of Chapter 2. As the weights of a hidden node in an RBM correspond to pixels, the visualisation of RBM features can be obtained by showing the weight vectors of hidden nodes in the shape of the input images. For LRF-RBMs, the same visualisation method can

be used after the receptive field masks are applied to the weights by element-wise multiplication. Visualisation of higher-layer features in a DBN or an LRF-DBN is obtained by the approximate method described in Chapter 2, whereby these higher-level feature visualisations are calculated using linear combinations of feature visualisations from the previous layer.

By examining centre maps, receptive field maps, and feature visualisations, DBN and LRF-DBN features were compared based on the distinctiveness of their appearance and locations.

6.3.2.2 Face Completion Task

To evaluate the quality of LRF-DBN generative models, multiple face completion tasks were set out. In these problems, certain pixels of the input images are occluded and, with the location of the occlusion being given, the model is required to fill in the missing pixels. The occlusions and the infilling procedure were similar to the ones used by (Ranzato et al., 2011) and performance was evaluated on seven different face completion tasks:

(i) occlusion of the left,

(ii) right,

(iii) bottom, or

(iv) top half of the image (these tasks collectively will be referred to as the 'side completion' tasks),

(v) rectangular occlusion around the eyes or

(vi) mouth area, and

(vii) occlusion of 70% of the image pixels chosen at random.

The areas of occlusion can be seen in Figures 6.11 to 6.14. A schematic diagram in Figure 5.3(c) in Chapter 5 illustrates the use of an LRF-DBN generative model on a random area completion task.

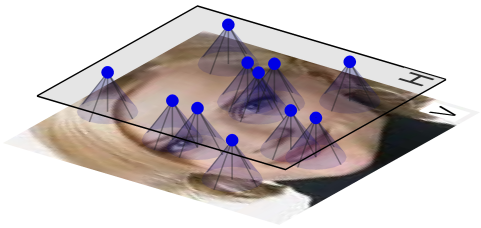
To solve a completion task with a generative model, when presented with an image, the occluded pixels were first set to 0 and the image was used as input to the model. Bottom-up propagation through the network layers was first performed, where hidden node probabilities were given by Equation (5.3) on the first and Equation (5.1) on the consecutive hidden layers. This was followed by a top-down pass through the network. When reaching the visible layer at the bottom, the states of visible nodes corresponding to uncorrupted pixels were left unchanged, while the missing pixels were filled in using Equation (5.4). Multiple iterations of bottom-up and top-down passes through the network layers were used to compute a restored image.

Face completion performance of LRF-DBN and DBN generative models were compared on the unseen test set quantitatively, by calculating the average SRE per missing pixel, and qualitatively, by evaluating example completions.

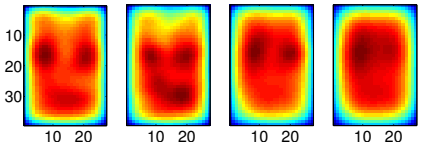
6.3.2.3 Reconstruction Task

LRF-RBM and RBM reconstructions were calculated on the test set during the iterations of CD using Equations (5.3) and (5.4) and Equations (2.9) and (2.10), respectively. The analysis involved comparing both the obtained SRE scores and examples of reconstructed images.

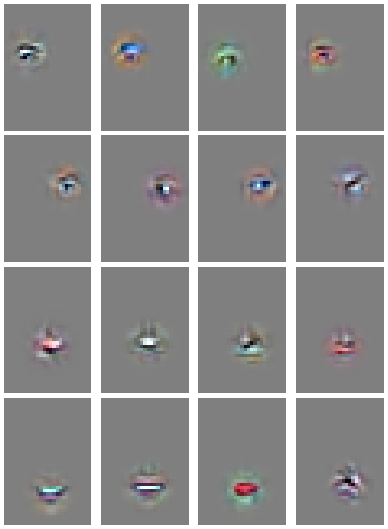
LRF-DNN and DNN image encodings and reconstructions can be obtained by computing the activations through the layers of the fine-tuned networks after feeding in an example image. The dimensionality reduction performance of LRF-DNN and DNN autoencoders were evaluated on the test set quantitatively, by



(a) LRF-RBM model



(b) LRF-RBM receptive field maps



(c) LRF-RBM features

Figure 6.1 LRF-RBM training on face images. (a) LRF-RBM model schematic, showing feature hubs around the eyes and mouth. (b) Receptive field maps of four LRF-RBM models, containing 4000 hidden nodes each, trained with different parameter settings. In each map, automatically learnt feature detector receptive fields are combined to show areas attracting more detectors. Darker red areas indicate higher feature density. Feature hubs emerged around average eye and mouth locations at pixels (9, 15), (20, 15), and (15, 30). (c) Distinctive-looking detectors located in feature hubs in a 4000-hidden-node LRF-RBM. From top to bottom row: Features, learnt unsupervised, detecting the persons' right eye, left eye, nose, and mouth can be seen. The second map in (b) belongs to the LRF-RBM which learnt the local features in (c).

measuring the average SRE per pixel, and qualitatively, by displaying example reconstructions.

6.4 Results

6.4.1 Learnt Features

Visualisation methods described in Section 6.3.2.1 will be used to identify the types of features LRF-RBMs on consecutive layers of an LRF-DBN extract from LFW face images and also to examine the spatial distribution of feature detectors learnt on LFW and MNIST.

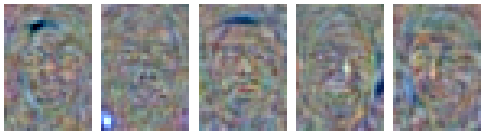


Figure 6.2 A sample of RBM features with high L2 norms, exhibiting a local peak around parts of the face contour.

6.4.1.1 First-Layer Features Learnt on LFW

Figures 6.1(c) and 6.7(a), columns 2–5 show local facial feature detectors learnt by LRF-RBMs, while RBM features can be seen in Figures 6.2 and 6.7(d), first row, columns 2–5.

All the traditional RBMs have learnt detectors similar in nature to the ones shown, with the majority exhibiting global structure and focusing on the main outline of a given type of face (as can be seen in Figure 6.7(d), first row, columns 2–5). While there exists a small number of detectors containing a local peak around the face contour, these are elementary in structure. Examples of such feature detectors are displayed in Figure 6.2. When examining RBM and DBN models, the lack of detectors focusing on local facial features is apparent. It is hard to find any well-defined local detector modelling face parts or a single facial feature, such as a dedicated eye or nose detector.

The LRF-RBMs, on the other hand, have attracted feature hubs around eye and mouth regions and, by focusing on these areas, have learnt a number of distinctive eye, mouth, and nose detectors. Receptive field maps in Figures 6.1(b) and 6.6(a) show spatial arrangement of detectors learnt by LRF-RBMs with different parameter settings. The layout of features, with the emergence of feature hubs around key areas in face images, demonstrates how LRF-RBMs can identify important regions within the input data, which need a higher density of feature detectors in order to represent their details. Figure 6.1(c) shows a set of

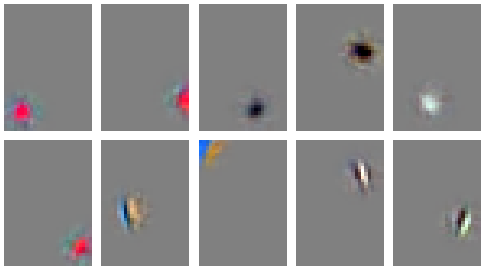


Figure 6.3 Examples of non-face-specific features learnt by an LRF-RBM with 4000 hidden nodes. DoG and Gabor filter shaped features are common among the detectors.

features sampled from areas around feature hubs in an LRF-RBM with 4000 hidden nodes. These visualisations show well-defined local features corresponding to distinctive-looking detectors of the eyes, mouth, and nose.

Alongside these face specific detectors, Gabor filters (illustrated in Figure 4.2 in Chapter 4) and DoG detectors (see Figure 3.2 in Chapter 3) were also common among the learnt features, especially in areas around the face contour. Examples of such features are shown in Figure 6.3, with additional examples provided in Figure B.1 in Appendix B. As seen in Chapters 3 and 4, in the literature, DoG filters are the most common models of retinal ganglion cells with centre-surround receptive fields (see Section 3.2.5). While Gabor functions are well-known models of V1 simple cells. The experiments, thereby, confirmed LRF-RBMs and LRF-DBNs do not only bear structural similarity to networks in the visual pathway but have the capability to learn functions of certain retinal and V1 cells from image data in an unsupervised manner.

If a second layer is trained with an RBM (without receptive field constraints) on top of the examined LRF-RBM, global face detectors emerge. These features are well-defined and varied-looking, as can be seen in Figure 6.4, which displays a sample of the second-layer features in a 4000(L)-1000 LRF-DBN.

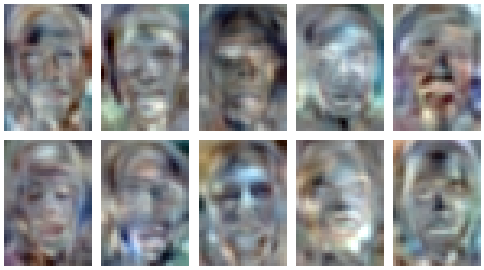


Figure 6.4 A sample of second-layer features learnt by a 4000(L)-1000 LRF-DBN, where the first layer was trained with an LRF-RBM and the second with a traditional RBM. The learnt features correspond to characteristic-looking faces.

6.4.1.2 Features Learnt on MNIST

On the MNIST digit dataset LRF-RBMs learnt feature detectors corresponding to local digit parts and also features of lower complexity, akin to DoG and Gabor filters.

LRF-RBMs learn hidden node locations in an unsupervised manner; however, for visualisation purposes on the MNIST dataset, one has the opportunity to use the available class labels. Receptive field maps in Figure 6.5 show, for each digit class, the location of those feature detectors which are the most selective for the given class. These maps give an insight into which image areas are most important for distinguishing between different digits.

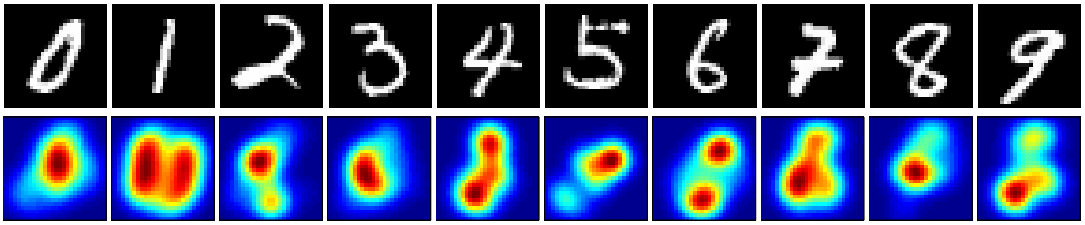


Figure 6.5 LRF-RBM receptive field maps learnt on the MNIST digit dataset. The top row shows example images of digits from 0 to 9. In the bottom row, for each digit class, a heat map shows locations of detectors which most selectively fire on images of the given digit.

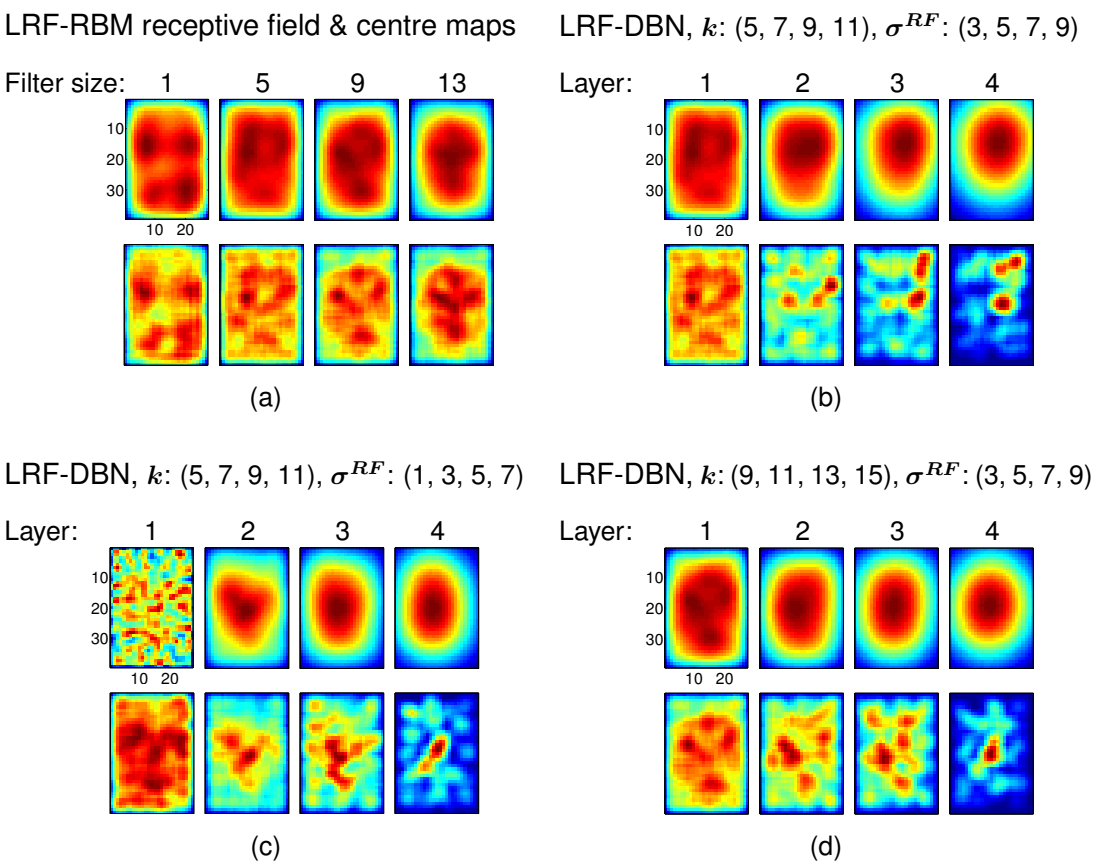


Figure 6.6 Receptive field and centre maps. (a) First-hidden-layer receptive field and centre maps shown in the top and bottom row, respectively, corresponding to different $\boldsymbol{k}$ values. (b)–(d) Receptive field maps in top rows and centre maps in bottom rows shown for consecutive layers from left to right in 2000(L)-1000(L)-500(L)-100(L) LRF-DBNs trained with different parameter choices for SD $\boldsymbol{\sigma^{RF}}$ and filter size $\boldsymbol{k}$.

6.4.1.3 Feature Hierarchy

Figure 6.6(b)–(d) shows receptive field and centre maps learnt by 2000(L)-1000(L)-500(L)-100(L) LRF-DBNs trained using different parameter settings. These maps reveal that feature hubs have typically emerged in the centre region, around average eye, mouth, and nose locations.

A sample of higher-layer features learnt in an unsupervised manner with a 2000(L)-1000(L)-500(L)-100 LRF-DBN is visualised in Figure 6.7(a)–(c). This feature hierarchy demonstrates how increasing the size of receptive field con-

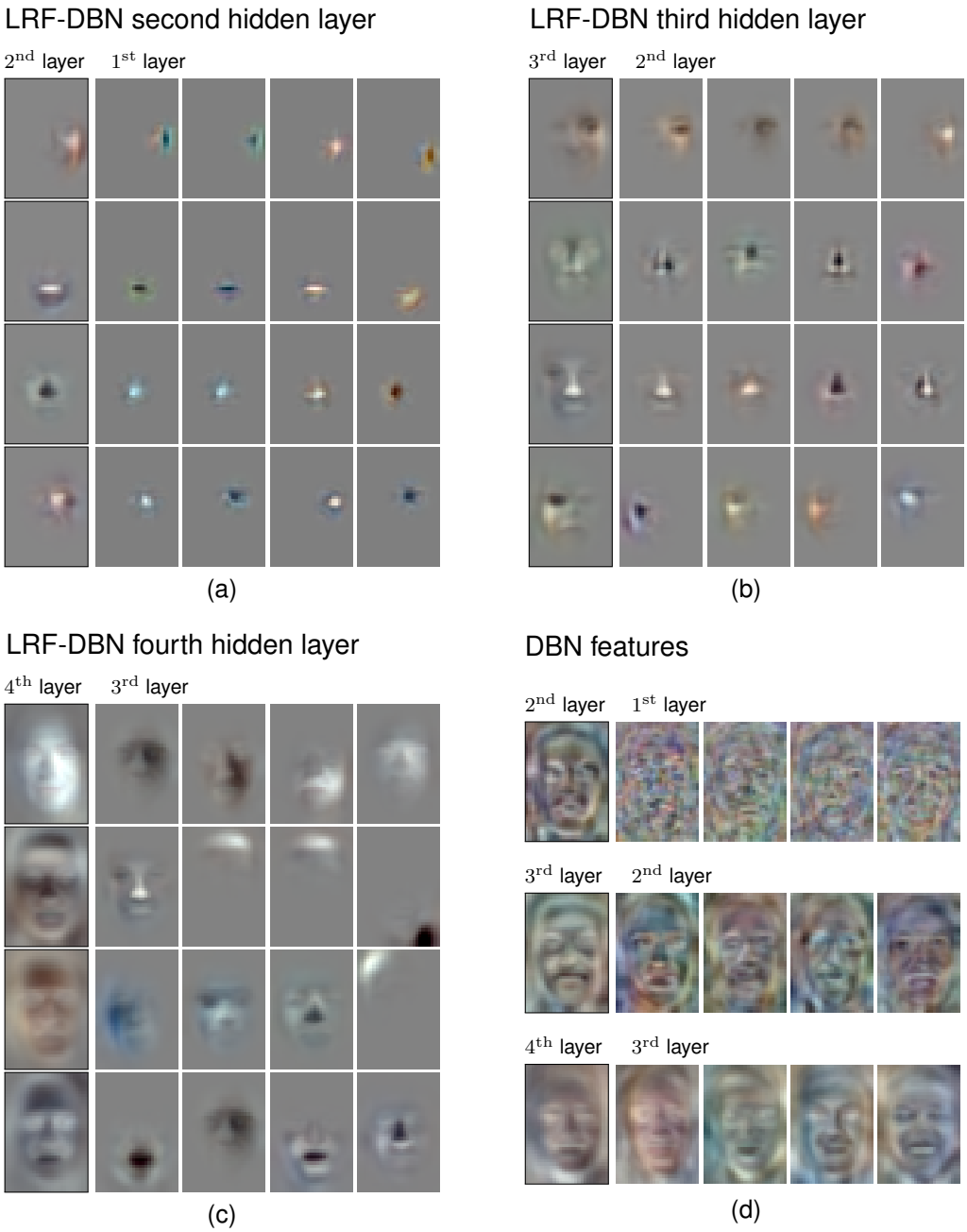


Figure 6.7 Feature hierarchy of (a)–(c) an LRF-DBN and (d) a DBN with 4 hidden layers. Visualisation of features demonstrates how consecutive layers are composed using features from the previous layer. The first image in each row in (a)–(d) visualises a feature from a higher layer, while consecutive images in the row illustrate those features from the previous layer which have the strongest connections to the given higher-layer feature. The LRF-DBN feature hierarchy in (a)–(c) is formed of features with gradually increasing size and demonstrates part-based composition.

straints between consecutive layers results in features of gradually increasing scope and complexity. Additional example features are provided in Figure B.2(a)–(c) in Appendix B. The receptive field and centre maps corresponding to this network can be seen in Figure 6.6(b), columns 1–3, with feature hubs around average eye and mouth locations and parts of the face contour.

First-layer features include highly local detectors focusing on parts of the eyes, mouth, and nose. Additionally, DoG and Gabor filter shaped detectors were also common (see, e.g., rows 2 and 10 in Figure B.2(a) in Appendix B). Among the second-layer features, one can easily notice well-defined eye, nose, and mouth detectors, while the third layer contains detectors of varying scope, including face part detectors and larger, near-global face detectors. Finally, Figure 6.7(c) demonstrates how the unconstrained RBM on the top layer is capable of combining different face part detectors to create global face detectors. Consequently, the experiments have shown that, from highly local features, the network is capable of building increasingly larger features and executes hierarchical part-based composition.

Features learnt by a traditional DBN of the matching 2000-1000-500-100 architecture are shown in Figure 6.7(d). A larger set of example features is provided in Figure B.2(d)–(f) in Appendix B. DBN features are mainly global, even on the first layer, and correspond to whole faces. Features on higher layers respond to clusters of faces with similar appearance by utilising the combination of lower-layer global face detectors. In contrast to our LRF-DBN feature detectors, DBN features do not show apparent part-based composition.

Architecture	Left	Right	Top	Bottom	Eyes	Mouth	Rand
DBN							
500	0.89	0.91	0.93	0.92	0.59	0.61	0.23
1000-500	0.89	0.91	0.92	0.90	0.56	0.58	0.26
2000-1000-500	0.90	0.92	0.94	0.92	0.50	0.52	0.27
4000-2000-1000-500	0.90	0.92	0.93	0.90	0.45	0.47	0.26
LRF-DBN							
500(L)	0.88	0.94	0.96	0.95	0.59	0.53	0.18
1000(L)-500(L)	0.78	0.83	0.85	0.82	0.37	0.39	**0.17**
2000(L)-1000(L)-500(L)	0.74	**0.76**	**0.76**	0.69	**0.33**	**0.36**	0.20
2000(L)-1000(L)-500	0.87	0.84	0.87	0.85	0.37	0.40	0.20
4000(L)-2000(L)-1000(L)-500(L)	**0.73**	**0.76**	0.77	**0.66**	0.34	0.37	0.22

Table 6.1 Average squared reconstruction error per missing pixel on the previously unseen test set shown for different generative models on the left, right, top, bottom, eyes, mouth, and random area completion tasks. The table compares multiple DBN and LRF-DBN architectures (also included are a 500 RBM and a 500(L) LRF-RBM). On all seven tasks, LRF-DBNs achieved the best scores.

6.4.2 Face Completion

Figures 6.8 and 6.9 and Table 6.1 show quantitative comparisons of RBM, DBN, LRF-RBM, and LRF-DBN generative models on the face completion tasks on the previously unseen test set, while a qualitative evaluation is provided through example completions in Figures 6.10 to 6.14.

6.4.2.1 Quantitative Comparison

Figures 6.8 and 6.9 and Table 6.1 compare SRE scores of an LRF-RBM, an RBM, and multiple LRF-DBN and DBN architectures achieved on the face completion tasks on the unseen test set. In Figures 6.8 and 6.9, SRE per missing pixel is shown as a function of the number of up-down passes that were performed through the network layers to infer missing values. On all the completion tasks,

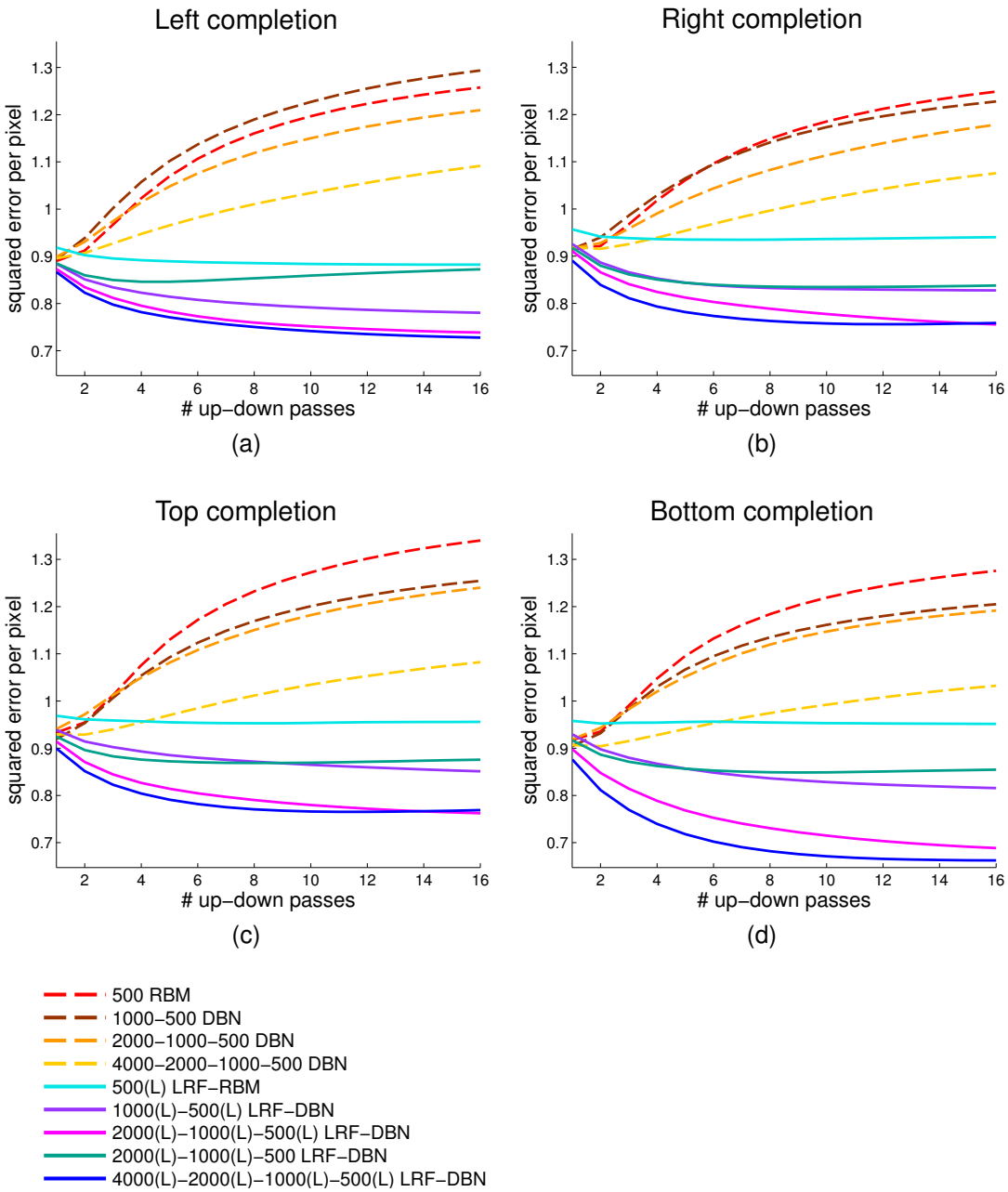


Figure 6.8 Average SRE per missing pixel on the test set shown on the (a) left, (b) right, (c) top, and (d) bottom face completion tasks as a function of the number of up-down passes. The comparison includes an LRF-RBM, an RBM, and multiple LRF-DBN and DBN architectures after the layer-wise pretraining (without fine-tuning). Trends on the different side completion tasks are similar, with LRF-DBNs outperforming DBNs.

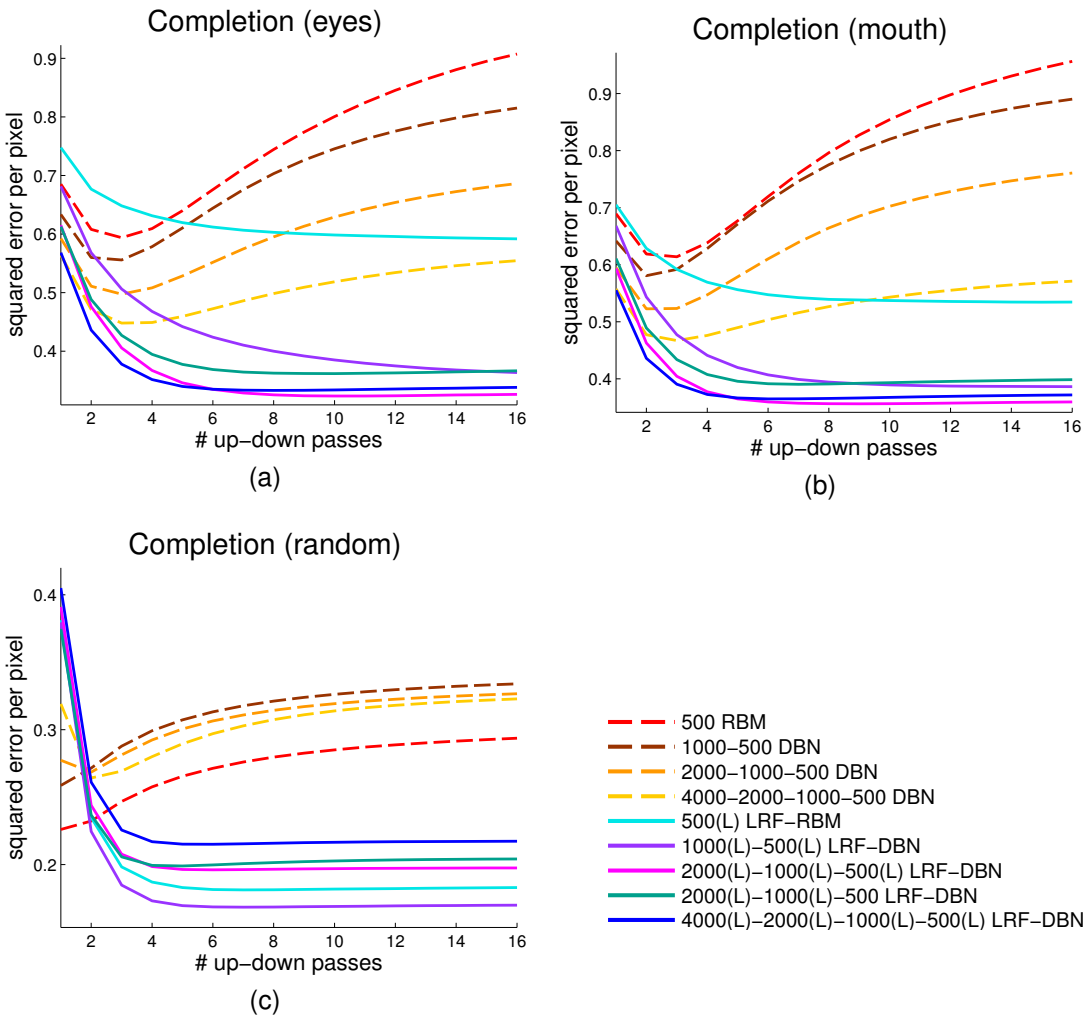


Figure 6.9 Average SRE per missing pixel shown on the test set as a function of the number of up-down passes on the (a) eyes, (b) mouth, and (c) random area face completion tasks. Networks are compared after layer-wise pretraining (without fine-tuning). LRF-DBNs show significant improvement with multiple up-down passes and outperform DBNs.

those LRF-DBNs which were trained using LRF-RBMs on each layer show gradual improvement with the number of passes and achieve lower SREs than DBNs. In contrast, after a few passes, the reconstruction error in DBNs steeply increases and, as the qualitative analysis also confirms, DBN completions can become very dissimilar to the original image.

Completions generated by a 2000-1000-500 DBN and the matching 2000(L)-

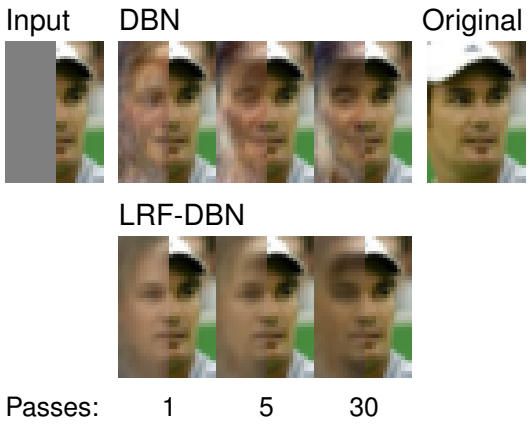


Figure 6.10 From a left-occluded input, shown in the first column, DBN (top row) and LRF-DBN (bottom row) completions are calculated after 1, 5, and 30 up-down passes (left to right). Unlike DBNs, LRF-DBNs generate completions which gradually become more similar to the original image (shown in last column).

1000(L)-500(L) LRF-DBN for an example test image with left-side occlusion are shown in Figure 6.10 after 1, 5, and 30 passes through the network. Visual inspection confirms, unlike the completions generated by the DBN, LRF-DBN completions gradually show more likeness to the original image as the number of infilling iterations increases.

Table 6.1 displays SRE scores of the networks examined in Figures 6.8 and 6.9. On each completion task, SREs of LRF-DBNs and the LRF-RBM are calculated after the 15^{th} pass, while in the case of DBNs and the RBM the best SRE value achieved during the first 15 passes is shown.[2] On each task, the top performing models were LRF-DBNs.

Face completion is, in general, a difficult task requiring higher-level knowledge of patterns in the data. In fact, face completion is implemented by higher processing areas in the human visual cortex. Therefore it is not surprising (as seen in Figures 6.8 and 6.9(a)–(b)) that LRF-RBMs, having highly local features,

[2]As a result, the displayed scores for LRF-DBNs are suboptimally chosen, whereas for each DBN the optimal, best-achieved scores are shown. This avoids giving any potential advantage to LRF-DBNs.

do not perform that well on these tasks. Through composition of local features, deeper LRF-DBN models, however, have the capability to extract higher complexity features from the input and thereby achieve superior results.

On most of the side completion tasks, lowest errors were achieved by the 4-hidden-layer LRF-DBN, while the 2000(L)-1000(L)-500(L) LRF-DBN performed best on eye and mouth area completion. The 2-hidden-layer LRF-DBN outperformed deeper networks on random occlusion infilling, which is likely due to this task being less reliant on global features.

6.4.2.2 Qualitative Comparison

Figures 6.11 to 6.14 qualitatively compare face completions generated by the 2000(L)-1000(L)-500(L) LRF-DBN and the 2000-1000-500 DBN on example images of the unseen test set (shown in the top rows).

Results on the left, right, top, and bottom completion tasks are displayed in Figures 6.11 and 6.12 in this order. For each task, the first row illustrates the area of missing pixels, followed by DBN and LRF-DBN completions in the subsequent rows. The number of up-down passes in each case was selected as described above for Table 6.1.

It can be concluded that the overall quality of LRF-DBN completions is superior: images look smoother, more face-like and show more similarity to the original faces. The LRF-DBN also managed to better recover small facial details, such as mouth shapes and smiles (see, e.g., column 6, left completion and column 10, right completion in Figure 6.11) as well as eye shapes and direction of gaze (see, e.g., last two columns, right completion in Figure 6.11).

The same tendency can be observed on the eyes, mouth, and random area completion tasks, for which the results are displayed in Figures 6.13 and 6.14.

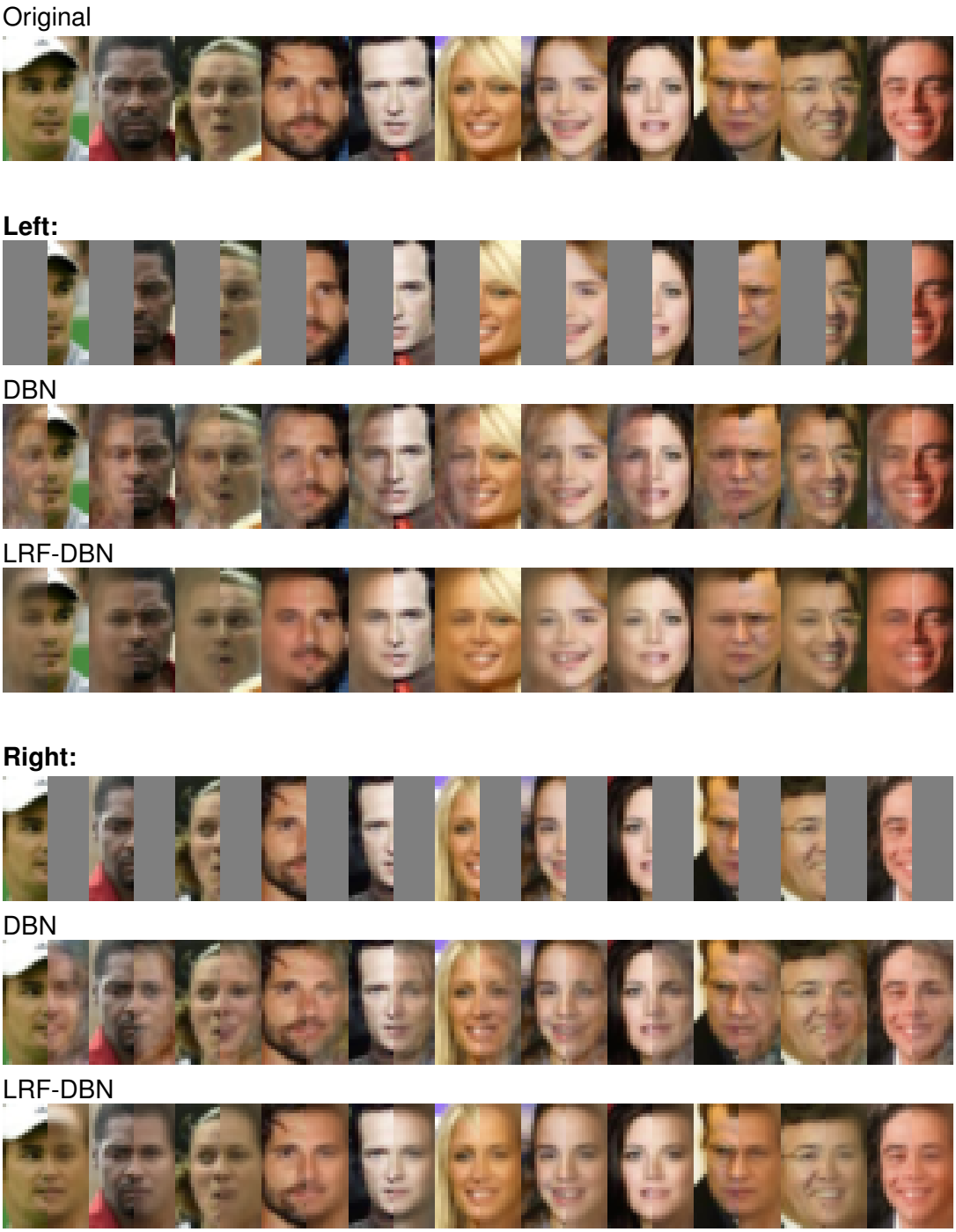


Figure 6.11 Example test images (first row) and their left and right completions (from top to bottom). For each task, the first row shows the occluded image, the second row images are completions generated by a DBN, and the third rows are computed by an LRF-DBN. LRF-DBN completions have better image quality, show more likeness to the original image, and better retain eye and mouth shapes and facial expressions.

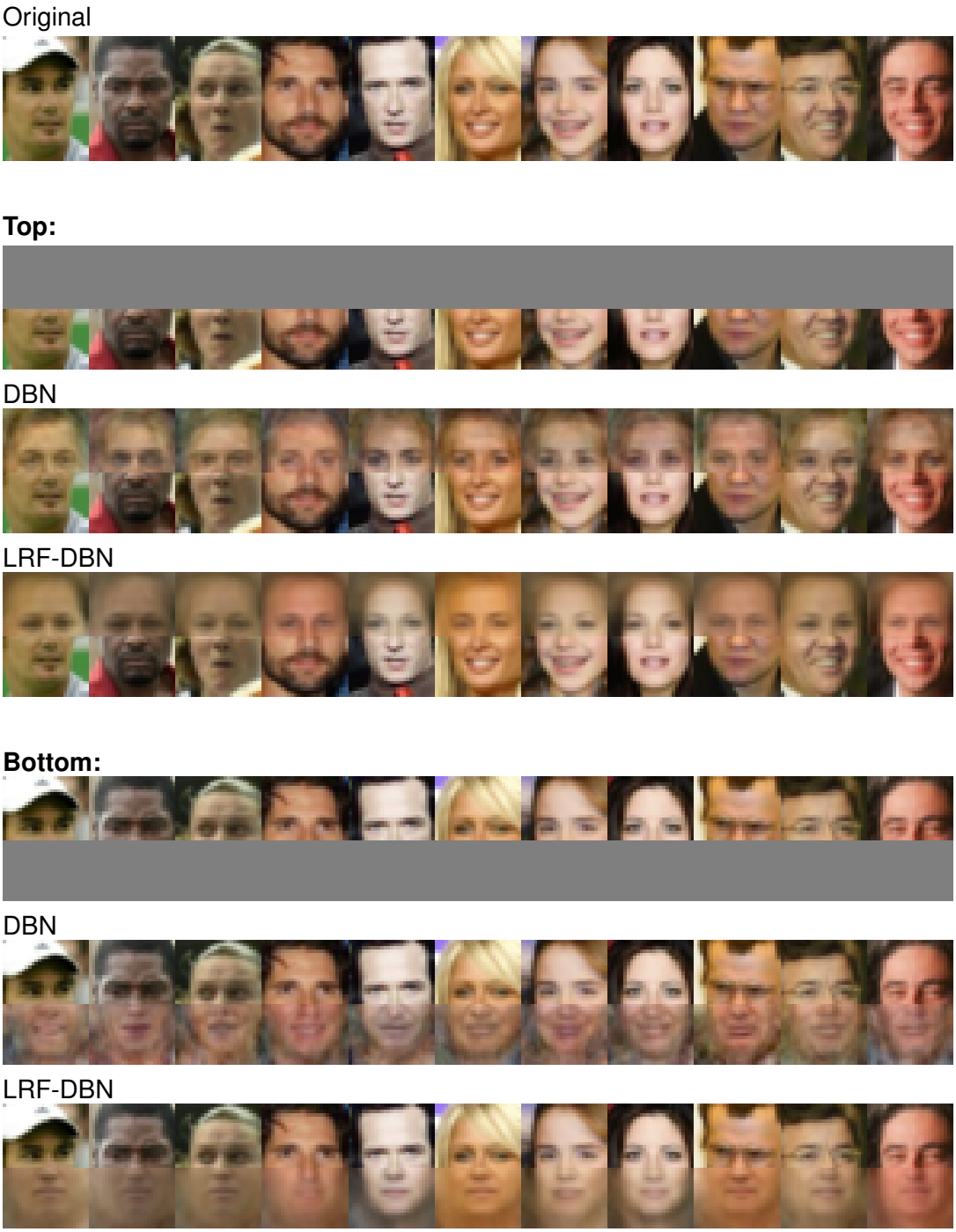


Figure 6.12 Example test images (first row) and their top and bottom completions (from top to bottom). For each task, the first row shows the occluded image, the second row images are completions generated by a DBN, and the third rows are computed by an LRF-DBN. LRF-DBN completions have better image quality: the images are smoother, more face-like, and show greater similarity to the original face images.

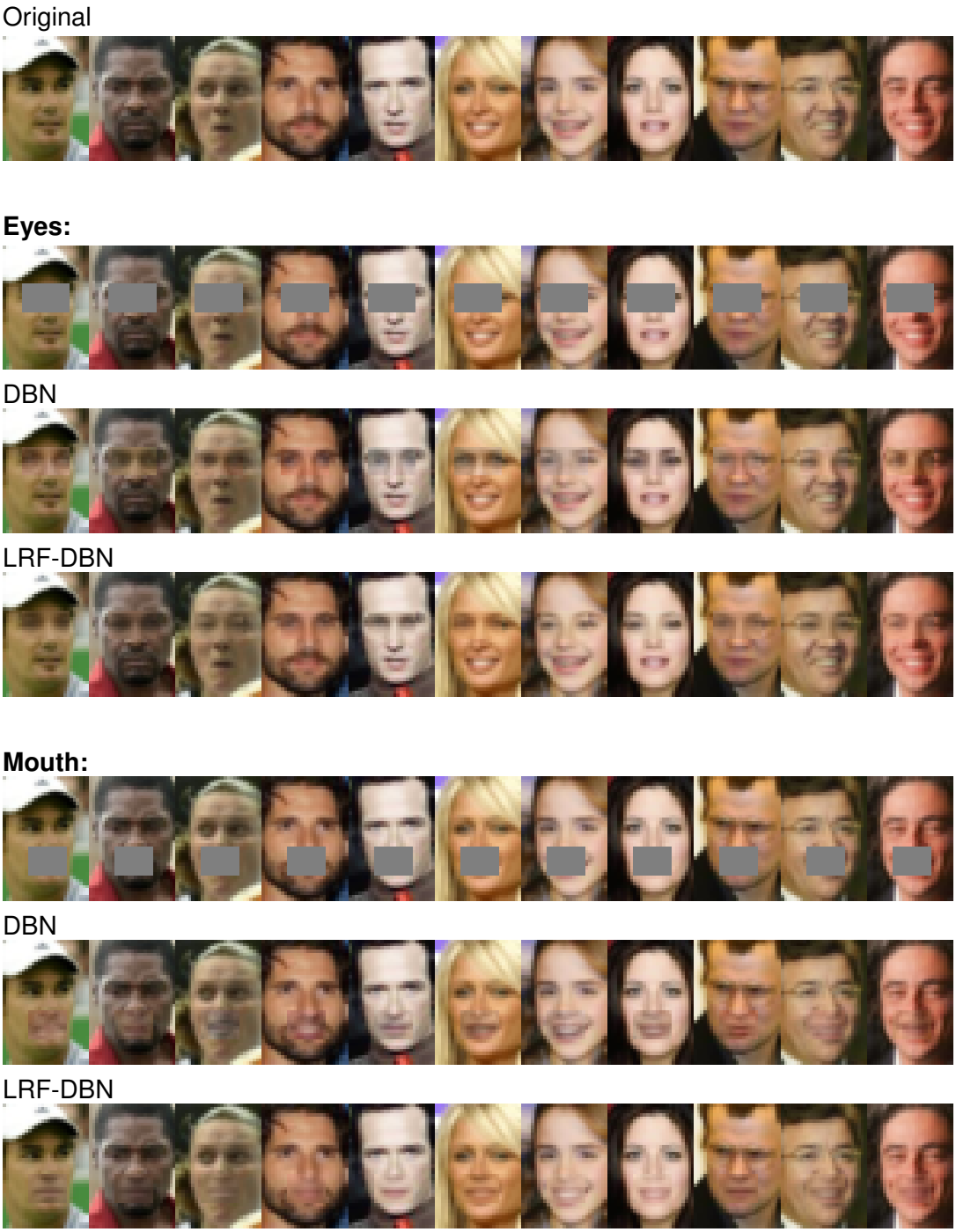


Figure 6.13 Example test images (first row) and their eye and mouth area completions. For each task, the first row shows the occluded image, the second row images are completions generated by a DBN, and the third rows are computed by an LRF-DBN. Facial expressions show greater similarity to the original images in the LRF-DBN completions, compared to DBNs, and an overall better image quality is achieved.

Figure 6.14 Example test images and their random area completions. Following the original images in the first row, the second row shows the input occlusion (where 70% of image pixels were occluded), while the third and fourth rows contain the DBN and LRF-DBN completions, respectively. LRF-DBN completions contain less noise and exhibit a high level of similarity to the original images.

The superior image quality and face similarity of LRF-DBN completions is especially pronounced on the mouth completion task, where mouth shapes, smiles, skin colours, and facial expressions are well retained by LRF-DBNs and the completed area blends in with the surrounding pixels.

6.4.3 Reconstruction

To provide a quantitative analysis, SRE scores of LRF-DNN and DNN autoencoders are compared on the unseen test data in Tables 6.2 and 6.3 and Figures 6.15 and 6.16, demonstrating the superior encoding and reconstruction capabilities of LRF-DNNs. This is confirmed by the qualitative evaluation of

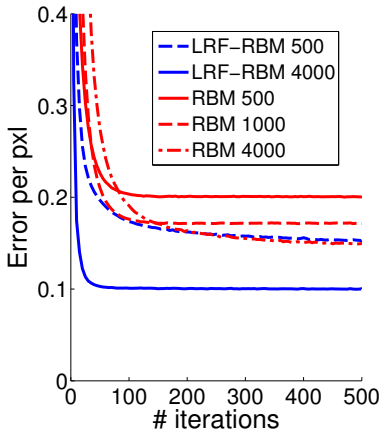


Figure 6.15 Average SRE per pixel shown on the test set as a function of the iterations of CD. Method names and hidden node counts constitute the labels in the graph. The LRF-RBMs were trained with $\sigma^{RF} = 3$ and $k = 5$. LRF-RBMs outperform RBMs of the matching architecture.

examples displayed in Figures 6.19 to 6.21. Furthermore, the following sections also examine different architectural and parameter choices for LRF-DNNs in Figures 6.17 and 6.18.

6.4.3.1 Quantitative Comparison

In Figure 6.15 SREs obtained on the unseen test set by LRF-RBMs and RBMs are shown as a function of the iterations of CD.[3] With the same node count, LRF-RBMs achieve significantly lower errors than RBMs. Moreover, a 500-node LRF-RBM performs similarly to a 4000-node RBM.

Figure 6.16 shows autoencoder results with a code length of 500 on the left and 100 on the right. For a given architecture, both the traditional and the LRF variant networks are included, and SREs are shown as a function of the number of backpropagation iterations. Figure 6.16(a)–(b) displays SREs for the complete set of LRF-DNN and DNN architectures described in Section 6.3.1, while Figure 6.16(c)–(d) compares different architectural choices in detail.

[3]Consequently, these models are not fine-tuned.

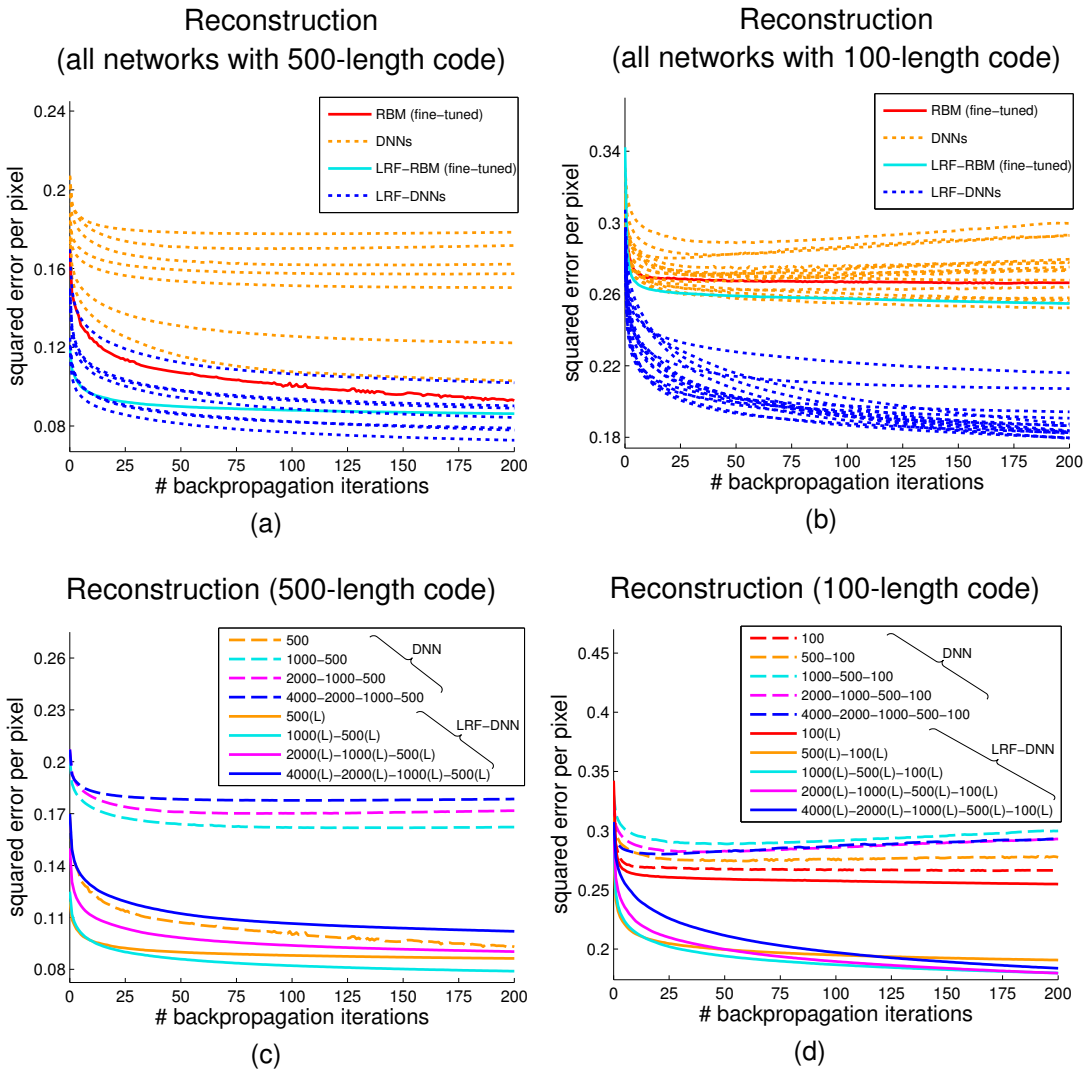


Figure 6.16 Average SRE per pixel on the unseen test set shown as a function of autoencoder fine-tuning iterations. The code length is 500 in (a) and (c) and 100 in (b) and (d). The 'fine-tuned RBM' and all DNNs were previously pretrained with RBMs on each layer, while LRF-DNNs and the 'fine-tuned LRF-RBM' were pre-trained with LRF-RBMs. The LRF-DNNs achieve lower SREs than DNNs.

Autoencoders in Figures 6.16(b) and (d) had a hidden layer with 100 nodes at the top; consequently, training aimed at reducing any 3159-dimensional input image of the dataset to a compact 100-length code, from which the image can be reconstructed. Results show all multi-layer LRF-DNN architectures compared favourably to any one of the DNNs, and even the single-layer LRF autoencoder,

Architecture	SRE
DNN	
500	0.093
1000-500	0.162
2000-1000-500	0.172
4000-2000-1000-500	0.178
LRF-DNN	
500(L)	0.086
1000(L)-500(L)	**0.079**
2000(L)-1000(L)-500(L)	0.090
2000(L)-1000(L)-500	0.103
4000(L)-2000(L)-1000(L)-500(L)	0.102

Table 6.2 SRE per pixel of DNN and LRF-DNN (including fine-tuned RBM and LRF-RBM) autoencoders with a code length of 500. Scores were measured on the test set after 200 backpropagation iterations. Superior results were obtained by LRF-DNNs.

the fine-tuned LRF-RBM, provided comparable reconstruction errors to the best performing DNN. As expected, multi-layer LRF-DNNs surpassed shallow architectures containing a single hidden layer. This is in contrast with DNNs, where most multi-layer architectures provided worse reconstructions than a shallow network.

In Figures 6.16(a) and (c), results with 500-length encodings are shown. Having a higher dimensional code layer is known to reduce the advantage of deep autoencoders compared to shallow networks (Hinton and Salakhutdinov, 2006a). In the case of DNN encoders, the advantage of deep models completely diminished on this task as none of the multi-layer DNNs performed better than the single-layer models, the fine-tuned RBM or LRF-RBM. On the other hand, a number of LRF-DNNs still retained superior performance compared to the shallow models. Akin to the 100-length encoding case, all multi-layer LRF-DNNs achieved better results than DNNs, and the LRF-RBM outperformed the RBM.

Architecture	SRE
DNN	
100	0.266
500-100	0.278
1000-500-100	0.300
2000-1000-500-100	0.293
4000-2000-1000-500-100	0.293
LRF-DNN	
100(L)	0.255
500(L)-100(L)	0.191
1000(L)-500(L)-100(L)	**0.180**
2000(L)-1000(L)-500(L)-100(L)	**0.180**
2000(L)-1000(L)-500(L)-100	0.222
4000(L)-2000(L)-1000(L)-500(L)-100(L)	0.184

Table 6.3 SRE per pixel of DNN and LRF-DNN (including fine-tuned RBM and LRF-RBM) autoencoders with a code length of 100. Scores were measured on the test set after 200 backpropagation iterations. Results confirm the superior performance of LRF-DNNs, with the 3- and 4-hidden-layer LRF-DNNs achieving best scores.

To further compare the architectures shown in Figure 6.16 (c)–(d), Tables 6.2 and 6.3 display SRE scores calculated at the end of fine-tuning, after 200 iterations of backpropagation. SREs in the case of 100-length codes (as shown in Table 6.3) were the lowest at 0.180, achieved by the LRF-DNNs with 3 and 4 hidden layers: 1000(L)-500(L)-100(L) and 2000(L)-1000(L)-500(L)-100(L), while the 2-hidden-layer 1000(L)-500(L) LRF-DNN produced the best 500-length codes with an SRE of 0.079 (see Table 6.2).

6.4.3.2 Architectural and Parameter Choices

Figures 6.17 and 6.18 examine different architectural and parameter choices for LRF-DNNs. Cases where RBMs were used for training some of the higher layers of an LRF-DNN are evaluated in Figure 6.17. As can be seen, these networks

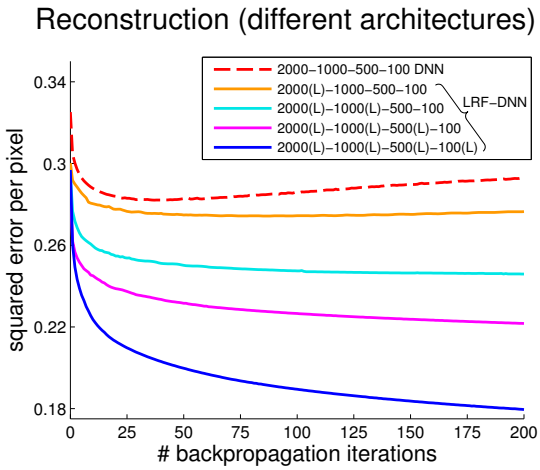


Figure 6.17 Comparison of SREs achieved by networks trained with different pretraining methods. All networks had 2000, 1000, 500, and 100 nodes on consecutive hidden layers. The architecture pretrained entirely with LRF-RBMs outperformed all other networks, while the DNN (pretrained with RBMs on all layers) obtained the highest errors.

provide inferior reconstruction scores compared to an LRF-DNN pretrained using only LRF-RBMs; however, they still compare favourably to a DNN.

The choice of SD $\boldsymbol{\sigma}^{\boldsymbol{RF}}$ is evaluated in Figure 6.18(a) in the case of a 2000(L)-1000(L)-500(L) network architecture and in Figure 6.18(b) for a 2000(L)-1000(L)-500(L)-100(L) architecture. Filter sizes were kept at $\boldsymbol{k} = (5, 7, 9)$ and $\boldsymbol{k} = (5, 7, 9, 11)$, respectively. The network with $\boldsymbol{\sigma}^{\boldsymbol{RF}} = (3, 5, 7)$ performed the best among the 3-hidden-layer networks, while in the case of the 4-hidden-layer architecture the network with $\boldsymbol{\sigma}^{\boldsymbol{RF}} = (5, 7, 9, 11)$ achieved the lowest SREs.

In the filter size comparison in Figures 6.18(c) and (d), $\boldsymbol{\sigma}^{\boldsymbol{RF}}$ was fixed at $(3, 5, 7)$ and $(3, 5, 7, 9)$, respectively, and the networks using $\boldsymbol{k} = (9, 11, 13)$ and $\boldsymbol{k} = (9, 11, 13, 15)$ on consecutive layers obtained the best scores within the two groups of networks.

Both the SD and filter size tuning experiments showed a good stability in performance within a large range of parameter values. With most settings, variation in SREs was small, especially when compared to the performance difference

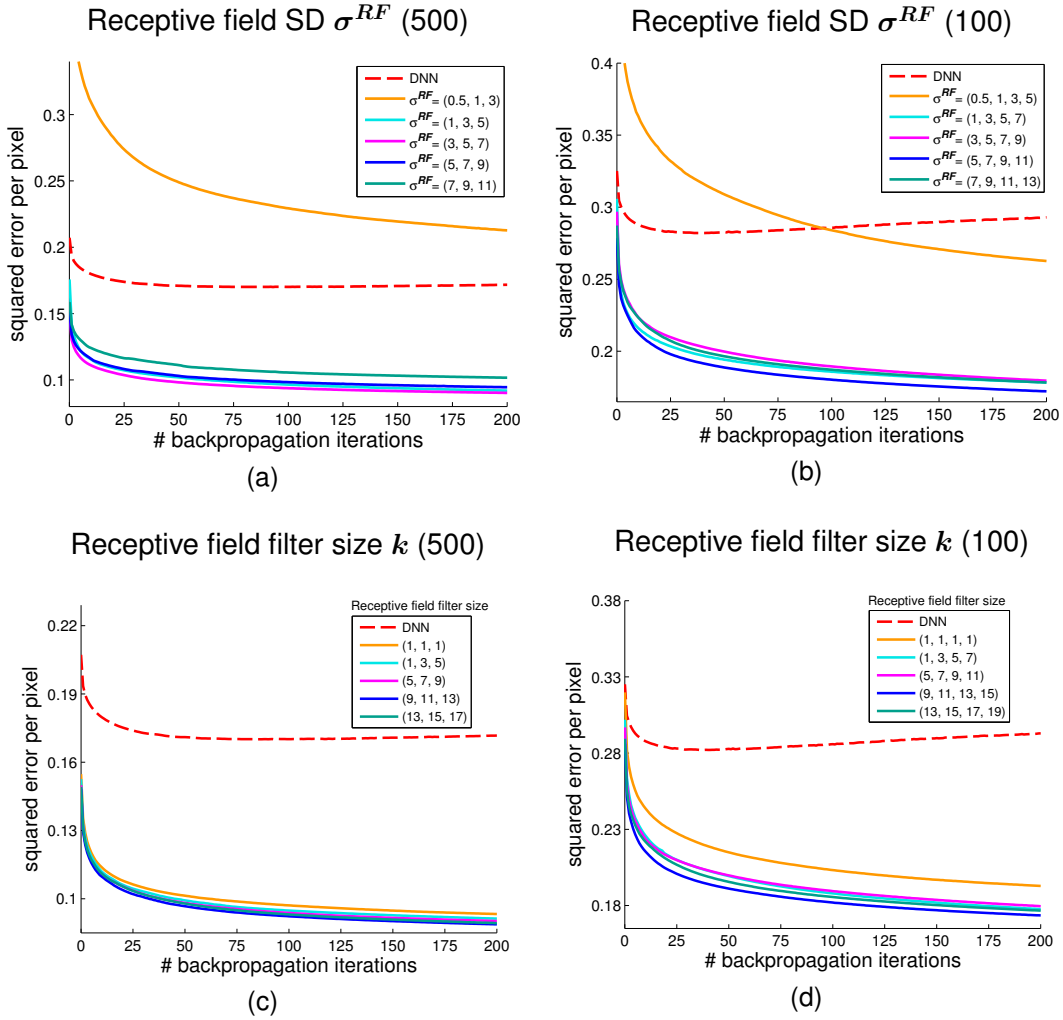


Figure 6.18 Comparison of different parameter choices for (a)–(b) SD $\boldsymbol{\sigma^{RF}}$ and (c)–(d) filter size $\boldsymbol{k}$. The architecture of the LRF-DNNs compared in (a) and (c) was 2000(L)-1000(L)-500(L), while the architecture was 2000(L)-1000(L)-500(L)-100(L) in (b) and (d). The graphs also include comparisons to the DNNs of the corresponding architecture. LRF-DNNs show good stability over different $\boldsymbol{\sigma^{RF}}$ and $\boldsymbol{k}$ values.

between LRF-DNNs and DNNs. This trend is confirmed for both the 500- and 100-length encoding tasks. Consequently, it can be concluded that reconstruction performance of LRF-DNNs is not too sensitive to the choice of $\boldsymbol{\sigma^{RF}}$ and $\boldsymbol{k}$.

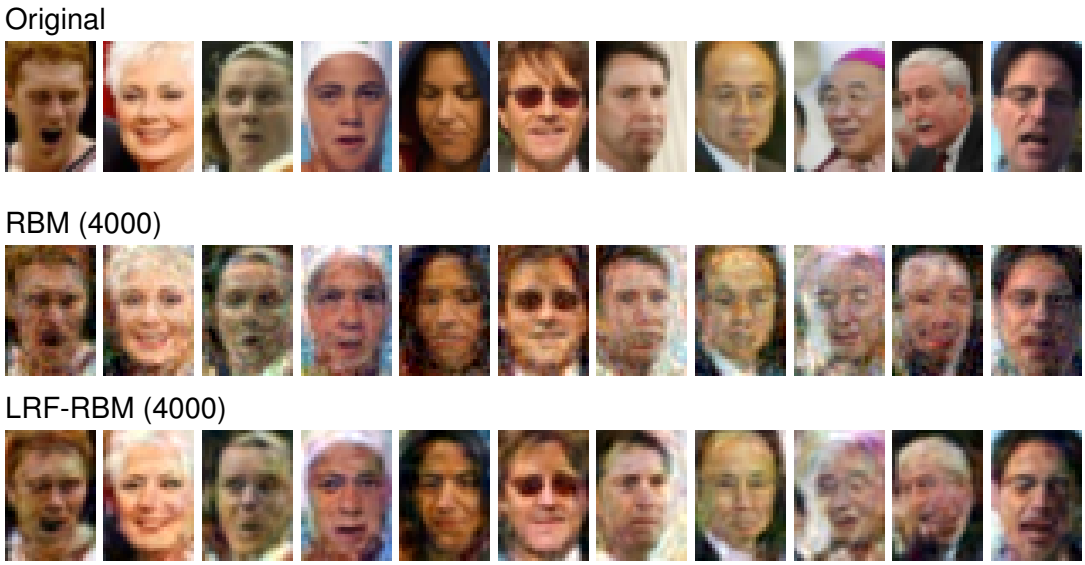


Figure 6.19 Example RBM and LRF-RBM reconstructions. To obtain reconstructions, the models were first presented with an unseen test image (shown in the first row) and the corresponding activations of the 4000 hidden nodes were measured, followed by the calculation of reconstructions in the visible layer. The second and third rows show reconstructions produced by an RBM and an LRF-RBM, respectively, trained without fine-tuning. Note how small details, e.g. eye and mouth shapes or the direction of gaze, are better retained with the LRF-RBM due to the number of specialised eye and mouth detectors. Note, also, how images of side-facing people can confuse the RBM but not the LRF-RBM (see columns 9–10).

6.4.3.3 Qualitative Comparison

Using unseen test images, Figure 6.19 compares the reconstruction quality of an LRF-RBM and an RBM, both containing 4000 binary hidden nodes. The models were not fine-tuned, and results are displayed after 2000 iterations of CD. Reconstructions are given by the top-down activations after feeding in a test image. As seen in Figure 6.15, LRF-RBMs obtain lower SREs. This superior reconstruction performance is confirmed by the qualitative analysis. LRF-RBMs better retain distinctive details of face images; moreover, the original facial expressions, direction of gaze, and emotions are easily recognisable in the reconstructed images.

Side-facing people provide a challenge for the models as such head poses are

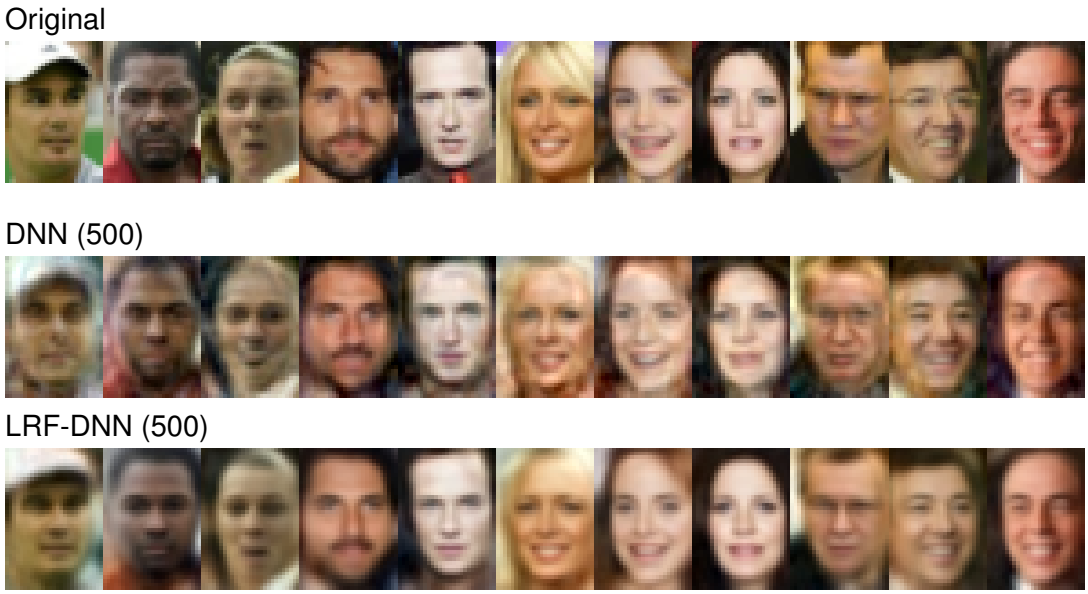


Figure 6.20 Example reconstructions from 500-length codes. A sample of the test data is shown in the first row. Reconstructions generated by a 2000-1000-500 DNN and a 2000(L)-1000(L)-500(L) LRF-DNN autoencoder are displayed in the second and third rows, respectively. Note how distinctive details, such as eye and mouth shapes (see, e.g., column 6) or direction of gaze (e.g. column 8), are better retained with an LRF-DNN due to the number of specialised eye and mouth detectors. LRF-DNN reconstructions also show an overall superior image quality.

less frequent in the dataset. However, unlike the RBM, the LRF-RBM can still provide good quality reconstructions that capture the side-facing pose of these faces (see columns 9–10).

A qualitative evaluation was also conducted on the deep autoencoder networks, which were trained to reduce the dimension of the input to either 500- or 100-length codes. Examination of reconstructed test images in Figures 6.20 and 6.21 confirms the superior performance of LRF-DNN autoencoders compared to DNNs. Example test images are shown in the first rows, while their reconstructions obtained from reduced-length codes with a DNN and an LRF-DNN of the matching architecture are given in the second and third rows, respectively. The dimension of the encoding was 500 in Figure 6.20 and 100 in Figure 6.21.

Close similarity of the original and the reconstructed images indicate LRF-

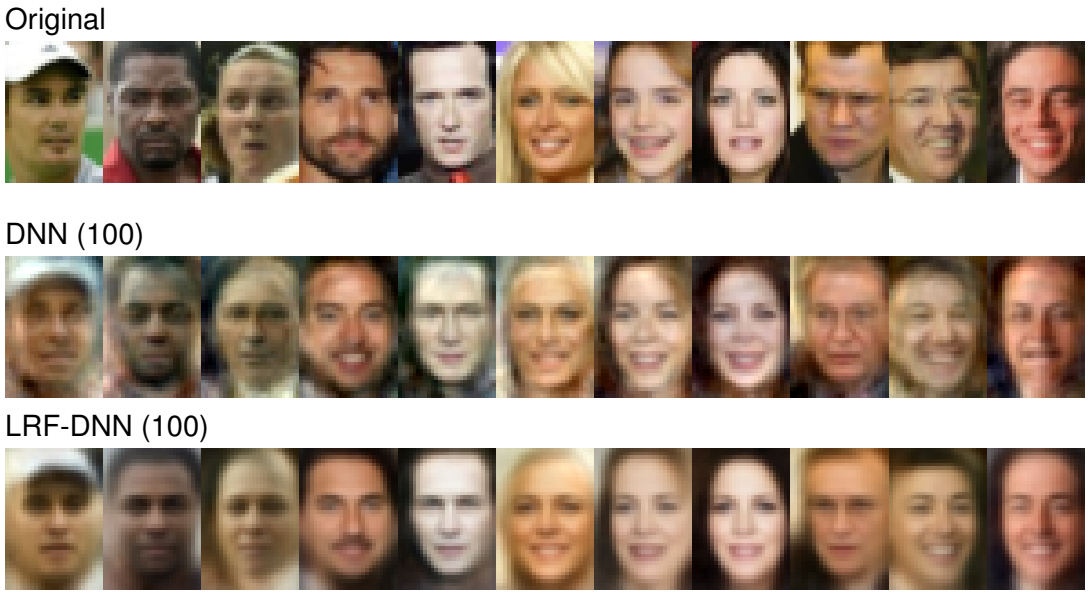


Figure 6.21 Example reconstructions from 100-length codes. A sample of the test data is shown in the first row. Reconstructions generated by a 2000-1000-500-100 DNN and a 2000(L)-1000(L)-500(L)-100(L) LRF-DNN autoencoder are displayed in the second and third rows, respectively. Note the superior image quality and how distinctive details, such as eye and mouth shapes (see, e.g., last column), are better preserved with an LRF-DNN due to the number of specialised eye and mouth detectors.

DNNs can encode and reconstruct key facial features and facial expressions of even the unseen test images using a code with a very limited length. DNN reconstructions show less likeness to the original faces and facial expressions. Characteristic details, especially around the eyes, nose, and mouth, are much better preserved with an LRF-DNN (see, e.g., third column in Figure 6.20 or the last column in Figure 6.21). Such facial features can be of crucial importance for distinguishing persons. The superior image quality of LRF-DNN reconstructions is also apparent, with the LRF-DNN providing conspicuously smoother and more natural-looking images than the DNN.

The quantitative and qualitative analyses thereby confirmed LRF-DNNs outperform DNNs (and, also, LRF-RBMs outperform RBMs) when reconstructing previously unseen data.

6.5 Summary

Section 6.4.1 has shown how LRF-RBMs are capable of identifying important regions in image data, termed feature hubs, which require a higher density of feature detectors to obtain an improved representation. Feature detectors learnt on consecutive layers in traditional DBNs and LRF-DBNs from the LFW face images were also analysed. Unlike DBNs, LRF-DBNs were found to learn feature hierarchies exhibiting part-based composition, ranging from DoG and Gabor filters and local feature detectors of the eyes, mouth, and nose to face part and whole face detectors.

As discussed in Sections 6.4.2 and 6.4.3, the LRF-DBN and LRF-DNN models achieved best scores on face completion and reconstruction tasks on the unseen test set. This superior performance underlines the importance of part-based compositionality and demonstrates the great generalisation capability of hierarchical models containing features of gradually increasing scale and complexity.

Furthermore, experiments presented here also confirmed that increasing the similarity of deep learning methods to information processing mechanisms in the visual pathway can lead to performance improvements on complex visual tasks.

7 Conclusions

The following chapter summarises the theoretical foundations and contributions of this thesis, presented in Chapters 2, 3 and 5, and discusses findings gathered through experimental analysis in Chapters 4 and 6. Both the advantages and shortcomings of the proposed methods are discussed, based on which, in Chapter 8, I suggest possible extensions of this work and attractive directions for related future research.

7.1 Overview

Chapter 2 discussed how the concept of deep learning gained remarkable popularity in recent years, leading to deep neural networks and convolutional neural networks largely replacing shallow models, such as SVMs, in machine learning and computer vision. This changeover was sparked by the development of an efficient training algorithm for neural networks with multiple hidden layers (Hinton et al., 2006; Hinton and Salakhutdinov, 2006a).

Chapter 2 provided a detailed description of this method, whereby multilayered network models are first pretrained layer by layer as a DBN using unsupervised RBMs. DBNs can serve as generative models, which allow for the production of new data according to the input data distribution. The pretrain-

ing phase is followed by fine-tuning of the entire network model to optimise performance on a specific task. For example, classifier or autoencoder deep neural networks can be obtained by fine-tuning with backpropagation.

Despite great uptake in computer vision, apart from a limited number of successful attempts, deep learning has not been widely applied within biological vision modelling. Additionally, the focus of those few studies has mainly been the modelling of higher-level processes in the visual cortex; consequently, the earliest stages of visual processing, implemented by the retina, have not been modelled in detail.

Chapter 3 emphasised that developing improved models of the retina can have widespread benefits across different domains, such as medical research, retinal prosthesis design, and also visual information processing and computer vision research. As described, there remains a number of uncertain details and open questions regarding the workings of the retina. Therefore, I argued that for modelling in this environment a flexible data-driven approach is most adequate. In agreement with such an approach, Chapter 4 has demonstrated the great suitability of deep learning methods for modelling the retina and early vision.

Still less knowledge is available about higher-order processing in the visual cortex. To address the modelling of such complex mechanisms, I proposed novel deep learning architectures and flexible training algorithms. Chapter 5 introduced my local receptive field constrained DBN and DNN models and Chapter 6 showed their great capability for executing high-level visual information processing tasks, such as face completion.

LRF-DBNs and LRF-DNNs are pretrained with my proposed novel version of an RBM, the LRF-RBM, which imposes biologically inspired Gaussian constraints on feature detector receptive fields. These constraints limit the spatial

scope of detectors, thereby encouraging the emergence of local features. Furthermore, concurrently with learning the appearance of local feature detectors, LRF-RBMs are able to identify areas of the visual space that require a higher number of feature detectors to represent their details. This way, during training an advantageous non-uniform feature detector placement is learnt, where feature hubs (i.e. areas densely covered by feature detectors) can emerge in key locations. Thereby, the LRF-RBM training procedure implements a form of structure learning.

Inspired by the hierarchical organisation of the visual pathway, training of LRF-DBNs and LRF-DNNs can utilise LRF-RBMs with increasing receptive field sizes as building blocks for constructing a network model, where each consecutive layer contains feature detectors of increasing scope and complexity.

Furthermore, experimental analysis in Chapters 4 and 6 confirmed the examined deep learning models not only show structural resemblance to the inherently multi-layered retinal network and visual cortex but are capable of learning information processing functions implemented in these areas automatically from data.

7.2 Key Findings and Contributions

For the modelling of visual information processing, this thesis proposed novel methods inspired by the hierarchical, multi-layered organisation of the retina and the visual cortex, whereby higher-layer features are composed of lower-layer ones and generally exhibit larger scope and higher complexity. The experimental analysis evaluated deep learning models on both early-stage and high-level visual tasks.

7.2.1 Modelling of Early Visual Processing

Experiments in Chapter 4 confirmed the suitability of a flexible data-driven approach and the use of deep learning methods and probabilistic models, such as DBNs, for modelling the retina and early vision.

A new dataset of simulated retinal input (SPI) was created, showing circular spots of various sizes and colours against different backgrounds. A circular blob detection task, approximating an early stage visual processing functionality, was set out on SPI. Using this dataset and task, DBN and DNN models were trained in order to obtain models of the retina and early vision.

Without the need for building in hard-wired circuits, DBNs trained in an unsupervised manner were capable of learning DoG and Gabor filters, common models of retinal ganglion cell and V1 simple cell receptive fields.

When training DNN classifiers on the detection task, pretraining of the network layers was found to be crucial for performance, and results confirmed the superiority of DNNs containing larger numbers of hidden layers. The accurate classification performance of DNNs demonstrates their great efficacy in learning typical early visual processing tasks.

These findings serve as a strong validation for following the proposed data-driven approach and harnessing the potential of deep learning for the modelling of biological visual information processing.

7.2.2 Modelling of High-Level Visual Processing

Development of the novel LRF-RBM model and training algorithm was driven by the goal to improve the compositional behaviour of DBNs and DNNs. By using LRF-RBMs for the pretraining of LRF-DBN and LRF-DNN models, one can obtain a network architecture which adopts a hierarchical organisation analogously

to the visual cortex. Experiments in Chapter 6 demonstrated the outstanding potential of these novel deep learning methods for modelling neural processing of visual information, from the early stages to higher-level mechanisms.

When LRF-RBMs were trained on an aligned version of the challenging LFW face dataset, feature hubs accumulated around key locations within the face images, such as average eye, mouth, and nose positions and areas around the face contour.

When multi-layer models were compared on LFW, unlike DBNs, LRF-DBNs were found to learn a hierarchy of local feature detectors which showed part-based composition. Detectors with increasing scope and complexity emerged—corresponding to progressively more global facial features—with lower layers showing DoG and Gabor filters as well as eye, mouth, and nose detectors and higher layers containing face part and whole face detectors. This analysis revealed LRF-RBMs on consecutive layers of an LRF-DBN can learn to combine local parts to obtain higher-level features, thereby building up a hierarchical model with remarkable generalisation capability.

The novel methods' performance was assessed on two challenging, high-level visual tasks. Firstly, LRF-DBNs were evaluated as generative models on a face completion problem, where missing pixel values in previously unseen test images had to be recovered. Secondly, in the context of a dimensionality reduction task, LRF-DNNs were fine-tuned as autoencoders to produce low-dimensional codes for input images, from which the original input can be reconstructed.

The face completion and reconstruction results showed a superior performance for LRF-DBN generative models and LRF-DNN autoencoders compared to DBNs and DNNs. In addition to obtaining lower reconstruction errors, my proposed models produced reconstructions which better retained fine details of

faces, such as eye, mouth, and nose shapes, direction of gaze, and facial expressions.

These results highlight the benefits of the models' part-based compositionality, the flexibility resulting from a structure learning mechanism, and the focus of attention to key areas in the visual space resulting in data-driven learning of specialised eye, mouth, and nose detectors.

Therefore, it can be concluded that LRF-DBNs and LRF-DNNs compute improved, highly compact representations of input data patterns, and, as a result, these novel deep learning methods can provide powerful models of visual information processing.

7.3 Summary

Work presented in this thesis aimed at strengthening the connection between deep learning and the modelling of biological visual information processing. In the Introduction, it was argued that such a connection could result in improvements within both areas. This claim has been verified through experiments in Chapters 4 and 6. Firstly, I have shown how DBNs and DNNs, when trained on simulated photoreceptor input data, can learn powerful models of early visual processing. Secondly, it has been demonstrated that LRF-DBNs and LRF-DNNs—my proposed novel deep learning architectures inspired by biological visual information processing—can provide improved performance compared to DBNs and DNNs on visual tasks.

8 Future Work

This chapter identifies possible ways of improving the proposed methods and provides a general overview of interesting avenues and intriguing research questions within the areas of deep learning and biological vision modelling.

8.1 Local Receptive Field Constrained Models

8.1.1 Extensions

In order to further improve the flexibility of the proposed models and to extend their structure learning capabilities, in future work I will test the effectiveness of certain modifications to the LRF-RBM training algorithm and the LRF-DBN and LRF-DNN architectures (introduced in Chapter 5).

These modifications include the learning of different receptive field constraints on each network layer automatically from data, which, in the case of Gaussian constraints, translates to learning different values for the variance of the Gaussian. This modification could result in networks where a layer is allowed to contain multiple types of receptive field constraints, which are suitably selected through data-driven learning. Also, instead of using receptive field masks, I will test the effect of sampling edges in each LRF-RBM training step according to the receptive field constraints, as described in Section 5.4.1.1 in Chapter 5.

Different types of hidden node activation functions—e.g. rectified linear (Nair and Hinton, 2010), softplus (Glorot et al., 2011), or maxout (Goodfellow et al., 2013)—can be incorporated into the model, which could likely result in improvements. To this end, I will investigate how local receptive field constraints can be applied to different types of hidden nodes. To provide increased flexibility, a mechanism can be developed to identify the most suitable activation function for each node (see, e.g., Frey, 1997; Maul, 2013), enabling the modelling of 'analogue' and 'digital' neural activations (refer to Section 3.5.6 in Chapter 3). These modifications would increase the flexibility of the proposed models by allowing for more variations in the structure of the learnt networks.

Another type of modification is directed towards the mechanism of automatically learning receptive field centre locations in the LRF-RBM. I will examine whether slowing down the movement of receptive field centres gradually over time improves learning. It is also interesting to test if limiting the length of movement allowed per feature in each training step—a constraint that would increase similarity to biological processing—would produce improved feature detectors.

Furthermore, the implementation of a mechanism, e.g. pooling (LeCun et al., 1998; Lee et al., 2009a), to address small differences in image alignment could benefit the methods' accuracy on benchmark tests. To achieve additional performance boosts, one could include the commonly used *'dropout'* regularisation technique (Srivastava et al., 2014) during training or a *'spatial transformer'* module (Jaderberg et al., 2015), which improves the invariance of the model.

Finally, my future work will also examine how local receptive field constraints can be applied within different learning algorithms.

8.1.2 Applications

I plan to conduct further experiments with LRF-DBNs and LRF-DNNs to evaluate the methods on multiple datasets and problems, such as further face datasets as well as object and scene classification problems. An important step would be to examine how the methods' strong performance shown on LFW can be transferred to video data, high resolution images, and large-scale object recognition datasets.

As seen in Chapter 3, the structural organisation of the retinal network shows a distinct difference between the centre and the periphery of the visual space, with the density of ganglion cells being highest in the centre and the cells there having small receptive fields, while the periphery being less populated and containing larger receptive fields. In LRF-RBMs, feature detectors can move to different areas of the visual space during training and could therefore adopt a similar pattern. Consequently, for retinal modelling, I will apply LRF-RBMs on such datasets and problems where different structural organisation between the centre and the periphery is likely to evolve.

8.2 Retinal and Biological Vision Modelling

8.2.1 Deep Learning of Retinal Models

It can be beneficial to extend the DBN retinal model, introduced in Chapter 4, with constraints and mechanisms aimed at improving the quality of the learnt features (e.g. by reducing features that are ill-defined or duplicated), the compactness and generalisation capability of the model, and the similarity to retinal or visual cortical circuits. Suggestions for such constraints are provided below.

Sparsity constraints are often applied in deep learning models during training

in order to guarantee that activations of the learnt features will be sparse. Models with sparse features can have a number of advantageous properties, e.g., they can model inhibition between feature detectors and have been shown to learn V1-like local features from natural image patches (Lee et al., 2006; Olshausen and Field, 1996). Sparse activations in DBNs are most often encouraged by a regularisation that penalises deviation from a predefined, low frequency of hidden node firing (refer to, e.g., Lee et al., 2008).

Chapters 5 and 6 showed how, with the help of LRF-RBMs, the size of receptive fields can be controlled in a deep model. In future work, I will investigate extending the structure learning capacities of the models and applying further architectural constraints within the networks (e.g. for the number of hidden nodes per layers). Such constraints will be used to encourage the learning of models that exhibit an even closer resemblance in structure to neural networks of the retina and the visual cortex.

Experiments in Chapter 6 demonstrated how LRF-DBNs can utilise part-based composition within their feature hierarchy leading to a superior generalisation capability. With this in mind, it is important to analyse mechanisms that can further improve the compositional behaviour of the proposed network models.

8.2.2 Datasets for Modelling Biological Visual Processing

Chapter 4 has shown that the proposed DBN retinal model can successfully discover DoG type feature detectors automatically from data in an unsupervised setting, demonstrating the model's capability to learn functionality similar to retinal ganglion cells. In future work, I will address further aspects of retinal modelling mentioned in Chapter 3, e.g. modelling the development of the retinal

network, eye motion analysis, or learning from biological experimental data. For the analysis of these problems, it will be necessary to identify and assemble suitable datasets.

Future work will extend the experiments with DBN retinal models (described in Chapter 4) and with LRF-DBN and LRF-DNN models of visual processing (introduced in Chapter 6) to further datasets and tasks. Different types of input can be examined (e.g. natural images, images with added noise, video data, and data from electrophysiological recordings), and a number of suitable tasks can be identified corresponding to retinal ganglion cell, early vision, or high-level visual information processing functionalities. Potential tasks involving lower-level processes include contrast gain control, differential motion detection, edge detection, and prediction of the direction or speed of movement, while higher-level reasoning tasks can focus on, e.g., scene, object, face, or facial expression recognition. Based on the success of experiments presented in this thesis, it can be anticipated the proposed models may develop retinal or visual cortical functionalities in order to solve these tasks when trained on adequate data.

The flexibility of the advocated data-driven modelling approach means the developed methods can easily be adapted for use on further visual datasets and even on different types of data. In their current form, LRF-DBNs are suitable for application on images, but it would also be possible to accommodate other datasets where a spatial proximity can be defined between input coordinates and the learning of local features is advantageous. Therefore, suitable data from electrophysiological experiments could also be utilised for improving the fidelity of the proposed models to biological visual information processing.

The SPI dataset, used in Chapter 4, has been constructed with the intention to provide an easily controllable environment for simulating an early vision func-

tionality. My experiments showed that, from these synthetic data, DBNs were capable of learning feature detectors exhibiting strong similarity to ganglion cells. This model could potentially be improved by further increasing the similarity of the approximation to input processed by retinal cells. Also, a different type of approximation could be provided by extraction of small image patches from natural images. Section 4.4.4 in Chapter 4 describes advantages and shortcomings of natural image datasets and synthetic data, and in future work both approaches will be explored.

Accordingly, both types of image datasets and also video data will be generated for the purpose of testing the proposed retinal models and the models of higher-order visual information processing on different visual tasks, including object detection, object motion anticipation, and the prediction of eye motion.

As described in Chapter 3, the retina contains a high number of motion-related feature detectors. Therefore, from the point of retinal modelling, adaption of the proposed methods for the detection of motion patterns in video data is an important task. Some temporal extensions of RBMs and DBNs have been proposed in the literature, e.g. recurrent temporal RBMs (Sutskever and Hinton, 2007; Sutskever et al., 2008), which can provide a starting point for development in this direction.

8.3 Deep Learning and Structure Learning

As it is not long ago that deep learning methods have achieved their current popularity, a number of practical and theoretical aspects of deep learning have not yet been fully investigated. There still exist questions regarding what architectures and training methods can guarantee the most reliable learning performance alongside efficient computation. This applies especially to large-scale problems

and those data types that are different from the commonly investigated vision, text, or speech datasets.

Also, in most cases, finding an optimal model architecture or parameter choice still involves excessive tuning. This burden could be reduced by the development of automatic structure learning methods, which strive to discover the optimal structural parameters of the model from data.

As seen in Section 2.5 in Chapter 2, a large percentage of earlier structure learning methods either impose strict restrictions on the model structure, e.g. mixtures-of-trees (Meila and Jordan, 2001), or do not guarantee efficient convergence: see, e.g., evolutionary (Pelikan et al., 2001) or heuristic algorithms (Heckerman et al., 1995). A different, commonly used approach, Bayesian non-parametric methods described in Section 2.5.1, provide a principled way for inferring model structure.

As mentioned, LRF-DBNs implement a form of structure learning by automatically discovering an advantageous feature detector placement from data. In future work, the flexibility of these models can be improved by learning more parameters of the model structure from data, including the number of nodes within layers or the number of layers, and automatically selecting suitable hidden node types, similarly to (Adams et al., 2010; Frey, 1997). Accordingly, the training of LRF-DBNs could be augmented with a Bayesian non-parametric structure learning routine to implement structure learning within a probabilistic framework.

8.3.1 Deep Learning and Biological Vision Modelling

Through experiments provided in this thesis, I have shown how combining elements of deep learning and biological vision modelling can bring mutual benefits to these two research areas.

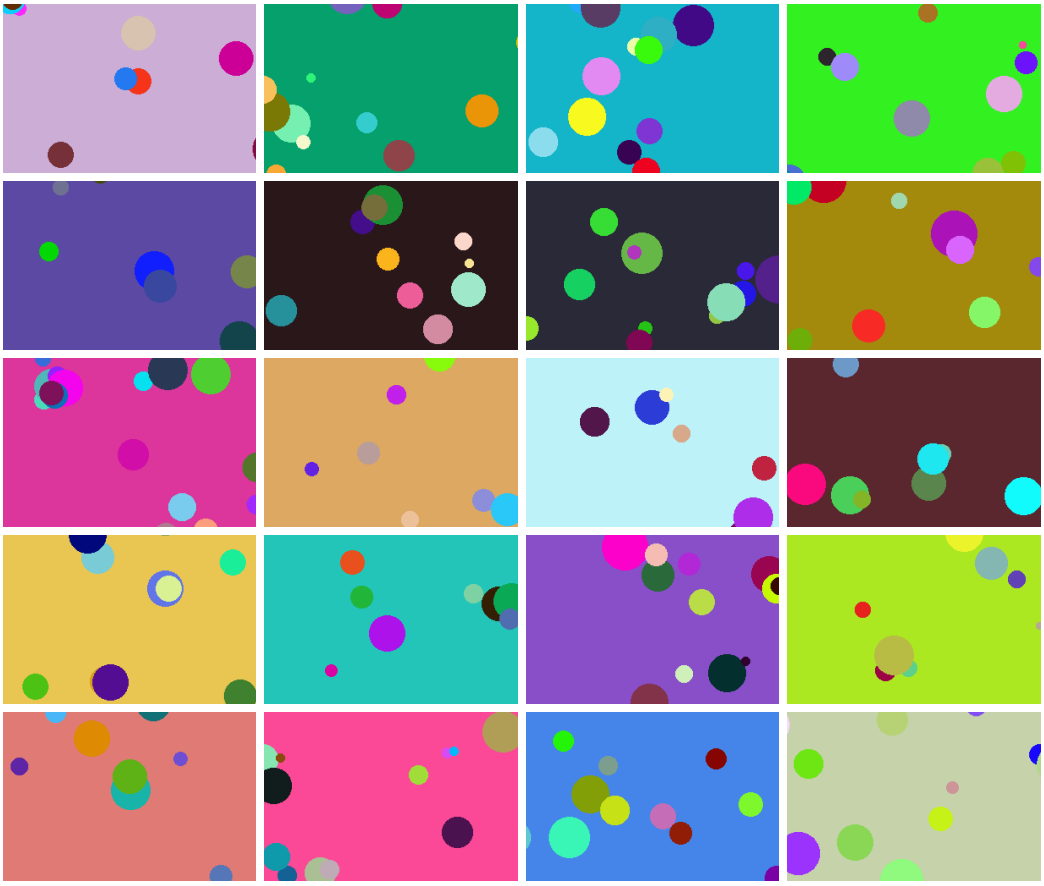
On the one hand, the use of deep learning methods for modelling biological vision introduces a data-driven approach to this domain and reduces the extensive reliance on knowledge-driven handcrafting of computational models.

On the other hand, increasing the similarity of deep learning methods to neural processing of visual information, by building on knowledge regarding these processes, not only increases the biological plausibility of deep learning models but can greatly improve their performance.

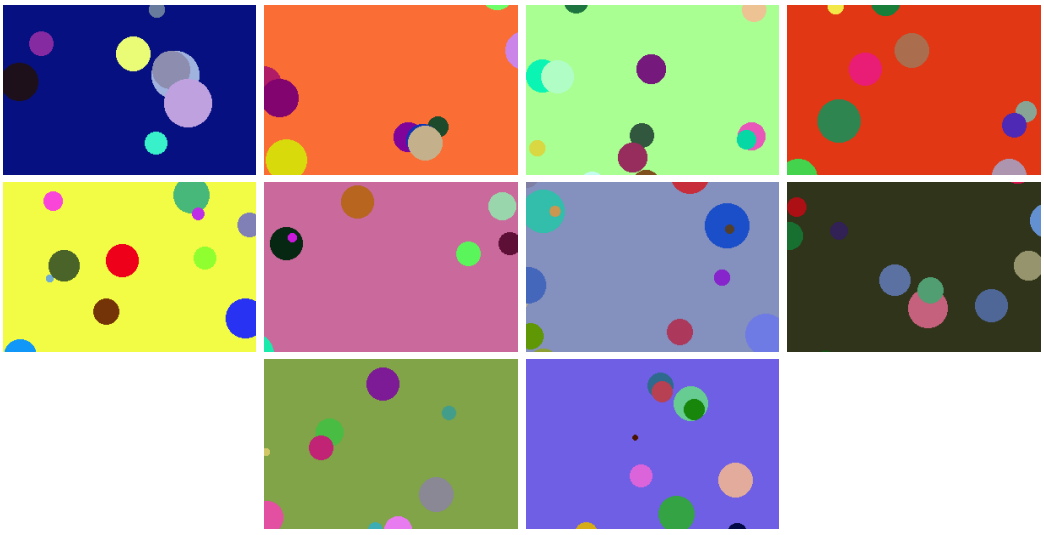
Therefore, I suggest, in future research such alternatives to current models may prove highly beneficial, popularising interdisciplinary approaches within these two areas.

Appendices

A Additional Figures from Chapter 4

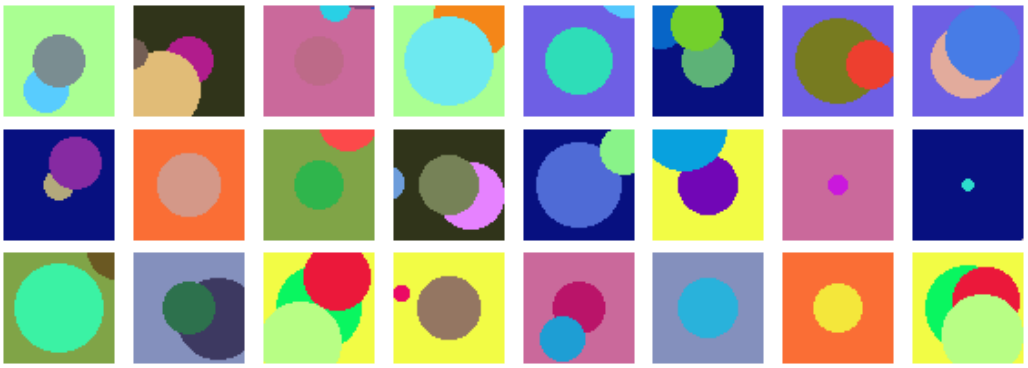


(a) Training video frames

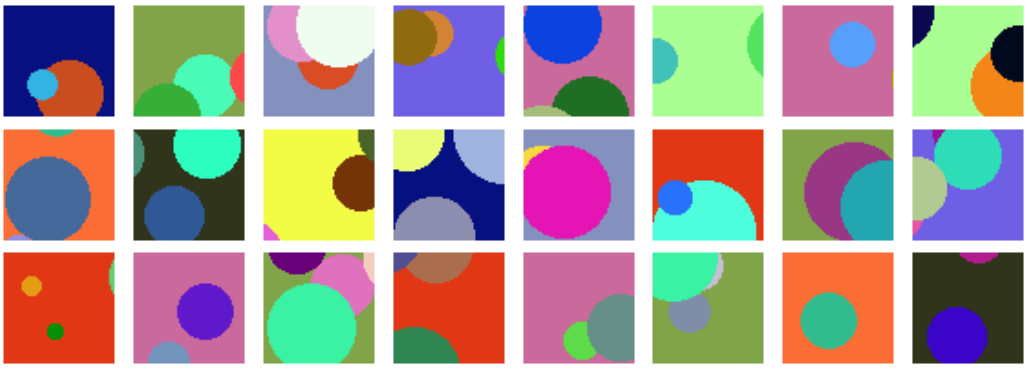


(b) Test video frames

Figure A.1 Example frames of the (a) training and (b) test videos showing circles of different size and colour against uniform background. Note the presence of overlapping circles. The background colours used in the training and the test set are different.



(a) Positives



(b) Negatives

Figure A.2 Examples of subwindows from the test set representing the (a) positive and (b) negative classes of the circle detection task. As opposed to negative class images, examples of the positive class have a circle positioned in the exact centre of the image. Images can contain circles that overlap and ones that have highly similar colour to the background.

B Additional Figures from Chapter 6

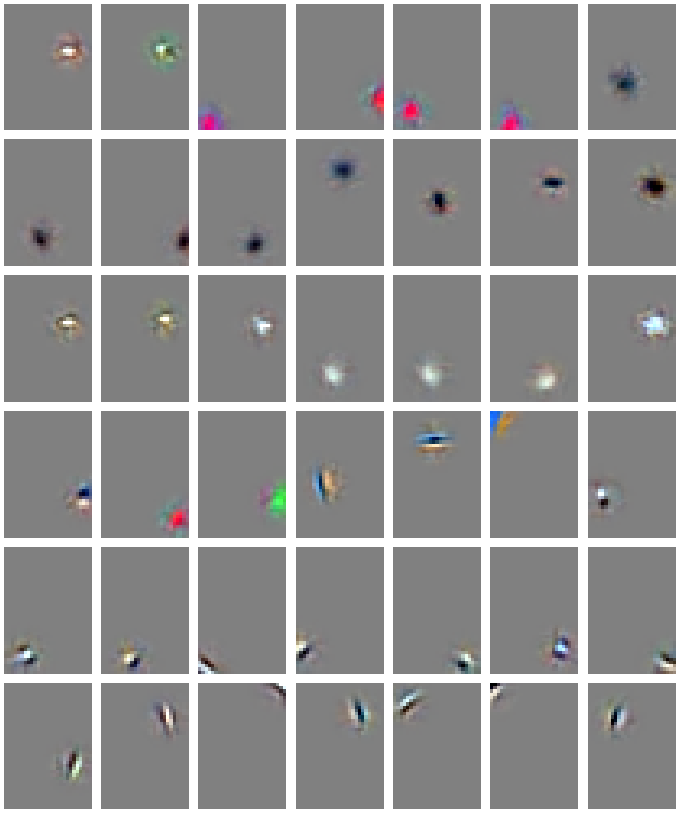
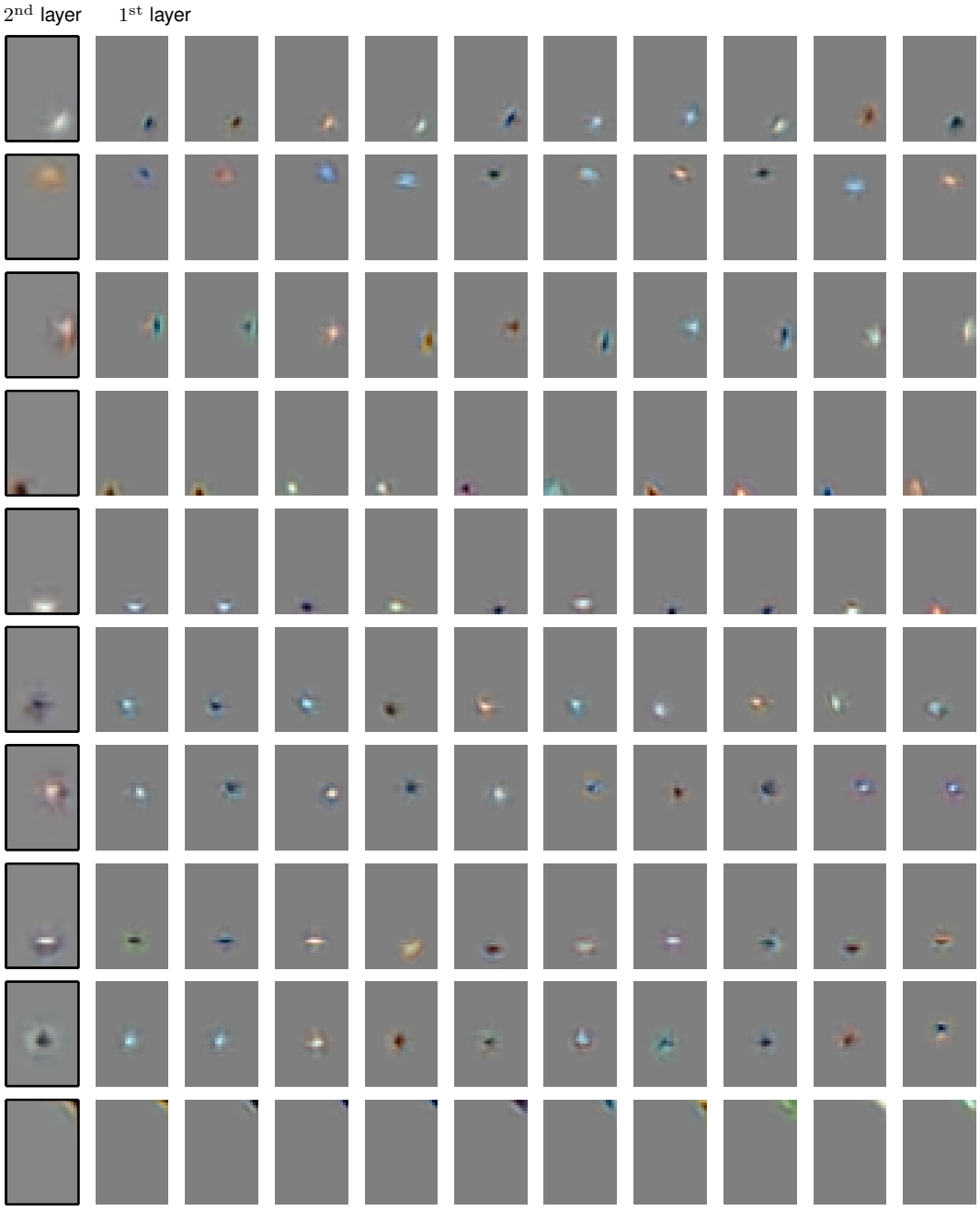
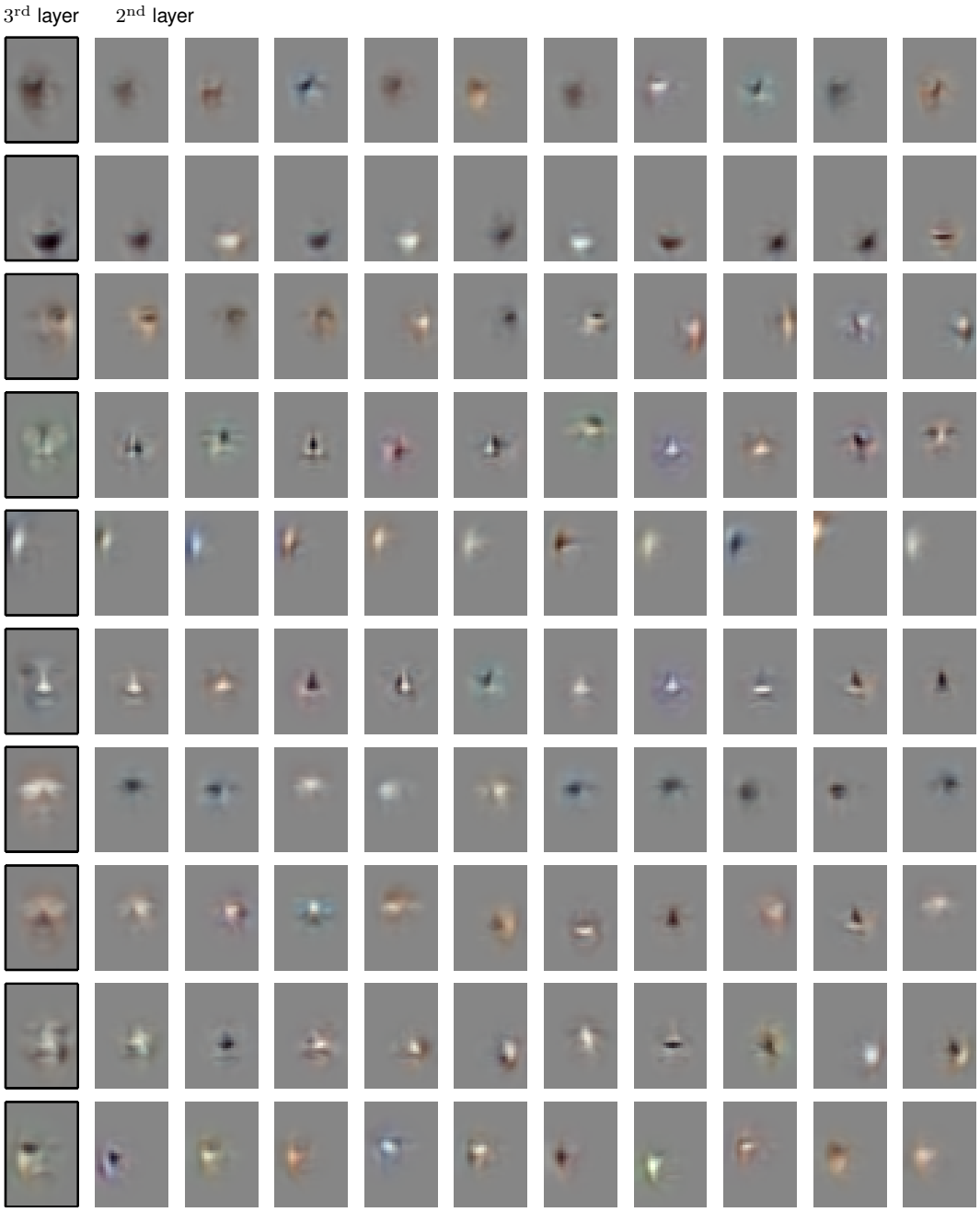


Figure B.1 Examples of non-face-specific features learnt by an LRF-RBM with 4000 hidden nodes. Many of the features show strong similarity to either DoG (see top three rows) or Gabor filters (see examples in the bottom three rows).

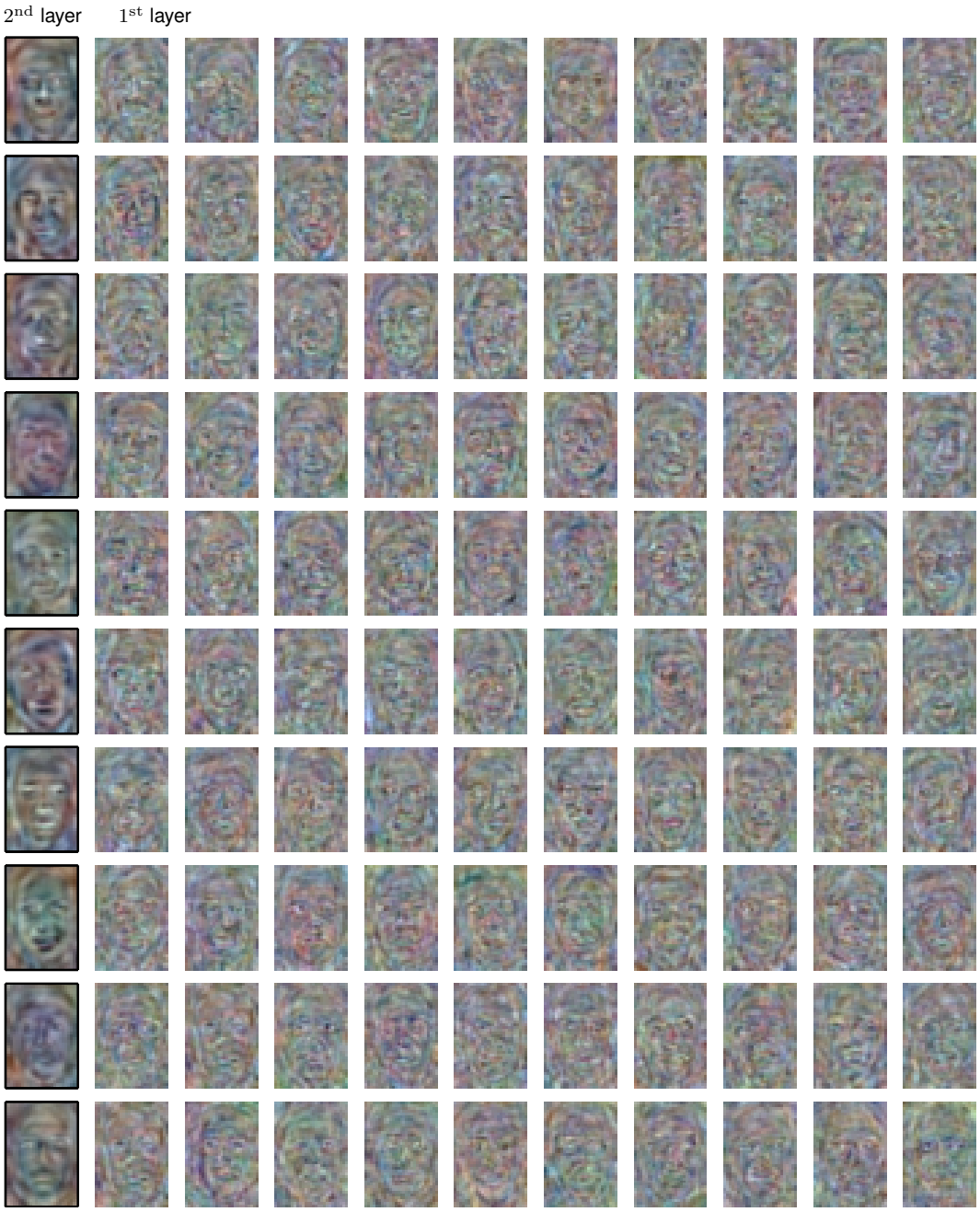


(a) Second-layer LRF-DBN features composed using first-layer features

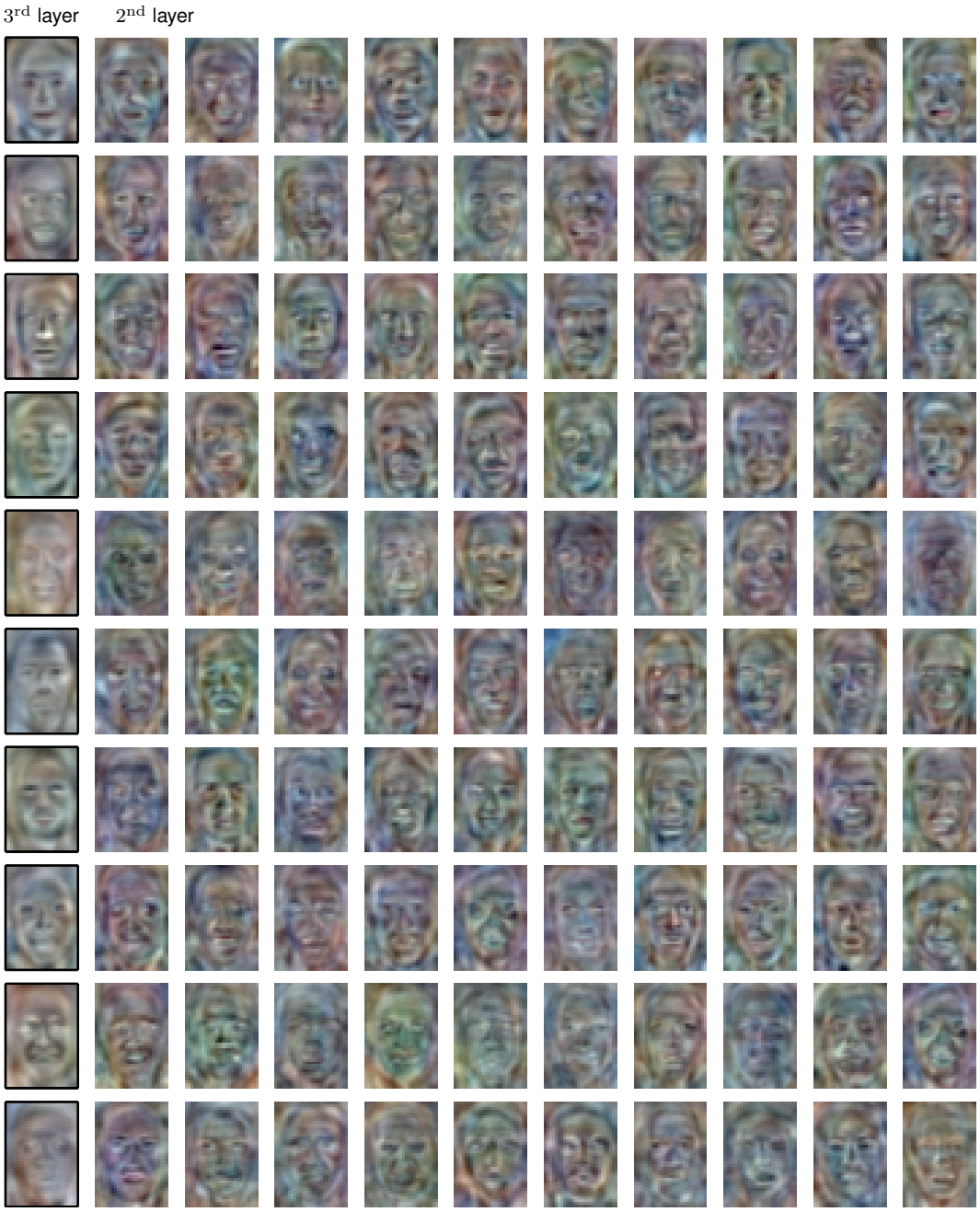


(b) Third-layer LRF-DBN features composed using second-layer features

(c) Fourth-layer LRF-DBN features composed using third-layer features



(d) Second-layer DBN features composed using first-layer features



(e) Third-layer DBN features composed using second-layer features

Figure B.2 Visualisation of (a)–(c) LRF-DBN and (d)–(f) DBN features demonstrates how consecutive layers are composed using features from the previous layer. The first image in each row in (a)–(f) shows a higher-layer feature, while consecutive images in the row illustrate those features from the previous layer which have the strongest connections to the given higher-layer feature. Unlike the DBN features, the LRF-DBN feature hierarchy in (a)–(c) demonstrates part-based composition.

References

Adams, R. P., Wallach, H. M., and Ghahramani, Z. (2010). Learning the structure of deep sparse graphical models. *Journal of Machine Learning Research: Workshop and Conference Proceedings (AISTATS)*, 9:1–8.

Aldous, D. (1985). Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII–1983*, pages 1–198.

Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B. C., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., Elsen, E., Engel, J., Fan, L., Fougner, C., Han, T., Hannun, A. Y., Jun, B., LeGresley, P., Lin, L., Narang, S., Ng, A. Y., Ozair, S., Prenger, R., Raiman, J., Satheesh, S., Seetapun, D., Sengupta, S., Wang, Y., Wang, Z., Wang, C., Xiao, B., Yogatama, D., Zhan, J., and Zhu, Z. (2015). Deep Speech 2: End-to-end speech recognition in English and Mandarin. *arXiv preprint arXiv:1512.02595*.

Angelov, P. P. and Filev, D. P. (2004). Flexible models with evolving structure. *International Journal of Intelligent Systems*, 19(4):327–340.

Atick, J. J. and Redlich, A. N. (1992). What does the retina know about natural scenes? *Neural Computation*, 4(2):196–210.

Attneave, F. (1954). Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193.

Bach, F. R. and Jordan, M. I. (2002). Learning graphical models with Mercer kernels. In *Advances in Neural Information Processing*.

Baden, T., Berens, P., Bethge, M., and Euler, T. (2013). Spikes in mammalian bipolar cells support temporal layering of the inner retina. *Current Biology*, 23(1):48–52.

Baden, T., Esposti, F., Nikolaev, A., and Lagnado, L. (2011). Spikes in retinal bipolar cells phase-lock to visual stimuli with millisecond precision. *Current Biology*, 21(22):1859–1869.

Barlow, H. B. (1961). Possible principles underlying the transformations of sensory messages. In Rosenblith, W. A., editor, *Sensory Communication*, chapter 13, pages 217–234. MIT Press, Cambridge, MA.

Barlow, H. B., Hill, R. M., and Levick, W. R. (1964). Retinal ganglion cells responding selectively to direction and speed of image motion in the rabbit. *The Journal of Physiology*, 173(3):377–407.

Bartusiak, R., Kajdanowicz, T., Wierzbicki, A., Bukowski, L., Jarczyk, O., and Pawlak, K. (2016). Cooperation prediction in GitHub developers network with restricted Boltzmann machine. In *Intelligent Information and Database Systems*, volume 9622 of *Lecture Notes in Computer Science*.

Battiato, S., Gallo, G., Puglisi, G., and Scellato, S. (2007). SIFT features tracking for video stabilization. In *Proceedings of the IEEE International Conference on Image Analysis and Processing*.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In Montavon, G., Orr, G. B., and Müller, K.-R., editors, *Neural Networks: Tricks of the Trade: Second Edition*, pages 437–478. Springer, Berlin, Heidelberg.

Berson, D. M., Dunn, F. A., and Takao, M. (2002). Phototransduction by retinal ganglion cells that set the circadian clock. *Science*, 295(5557):1070–1073.

Brown, S. P. and Masland, R. H. (2001). Spatial scale and cellular substrate of contrast adaptation by retinal ganglion cells. *Nature Neuroscience*, 4(1):44–51.

Buhr, E. D., Yue, W. W. S., Ren, X., Jiang, Z., Liao, H.-W. R., Mei, X., Vemaraju, S., Nguyen, M.-T., Reed, R. R., Lang, R. A., Yau, K.-W., and Gelder, R. N. V. (2015). Neuropsin (OPN5)-mediated photoentrainment of local circadian oscillators in mammalian retina and cornea. *Proceedings of the National Academy of Sciences of the United States of America*, 112(42):13093–

13098.

Charles, J., Pfister, T., Magee, D., and Hogg, D. Zisserman, A. (2016). Personalizing human video pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Chelaru, M. I. and Dragoi, V. (2008). Efficient coding in heterogeneous neuronal populations. *Proceedings of the National Academy of Sciences of the United States of America*, 105(42):16344–16349.

Chen, E. Y., Chou, J., Park, J., Schwartz, G., and Berry, M. J. (2014). The neural circuit mechanisms underlying the retinal response to motion reversal. *The Journal of Neuroscience*, 34(47):15557–15575.

Chen, J., Zhou, T., Yang, H., and Fang, F. (2010). Cortical dynamics underlying face completion in human visual system. *The Journal of Neuroscience*, 30(49):16692–16698.

Cireşan, D., Meier, U., Masci, J., and Schmidhuber, J. (2011a). A committee of neural networks for traffic sign classification. In *International Joint Conference on Neural Networks*.

Cireşan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. (2011b). Flexible, high performance convolutional neural networks for image classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Coates, A. and Ng, A. Y. (2011). Selecting receptive fields in deep networks. In *Advances in Neural Information Processing*.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning*.

Coogan, T. A. and Burkhalter, A. (1993). Hierarchical organization of areas in

rat visual cortex. *The Journal of Neuroscience*, 13:3749–3749.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Cottaris, N. P. and Elfar, S. D. (2005). How the retinal network reacts to epiretinal stimulation to form the prosthetic visual input to the cortex. *Journal of Neural Engineering*, 2(1):S74–S90.

Crook, J. D., Manookin, M. B., Packer, O. S., and Dacey, D. M. (2011). Horizontal cell feedback without cone type-selective inhibition mediates "red-green" color opponency in midget ganglion cells of the primate retina. *The Journal of Neuroscience*, 31(5):1762–1772.

Dacey, D. M., Diller, L. C., Verweij, J., and Williams, D. R. (2000). Physiology of L- and M-cone inputs to H1 horizontal cells in the primate retina. *Journal of the Optical Society of America A*, 17(3):589–596.

Dahl, G., Ranzato, M., Mohamed, A., and Hinton, G. E. (2010). Phone recognition with the mean-covariance restricted Boltzmann machine. In *Advances in Neural Information Processing*.

Deans, M. R., Völgyi, B., Goodenough, D. A., Bloomfield, S. A., and Paul, D. L. (2002). Connexin36 is essential for transmission of rod-mediated visual signals in the mammalian retina. *Neuron*, 36(4):703–712.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

DeVries, S. H. and Baylor, D. A. (1995). An alternative pathway for signal flow from rod photoreceptors to ganglion cells in mammalian retina. *Proceedings of the National Academy of Sciences of the United States of America*, 92(23):10658–10662.

Ding, C. and Tao, D. (2015). Robust face recognition via multimodal deep face representation. *arXiv preprint arXiv:1509.00244*.

Dong, D. W. and Atick, J. J. (1995). Temporal decorrelation: A theory of

lagged and nonlagged responses in the lateral geniculate nucleus. *Network: Computation in Neural Systems*, 6(2):159–178.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*.

Dowling, J. (2008). Current and future prospects for optoelectronic retinal prostheses. *Eye*, 23(10):1999–2005.

Dowling, J. E. (1987). *The Retina: An Approachable Part of the Brain*, chapter 4 Neuronal responses. Harvard University Press, Cambridge, MA.

Dreosti, E., Esposti, F., Baden, T., and Lagnado, L. (2011). In vivo evidence that retinal bipolar cells generate spikes modulated by light. *Nature Neuroscience*, 14(8):951–952.

Duhamel, J. R., Colby, C. L., and Goldberg, M. E. (1992). The updating of the representation of visual space in parietal cortex by intended eye movements. *Science*, 255(5040):90–92.

Eckmiller, R., Neumann, D., and Baruth, O. (2005). Tunable retina encoders for retina implants: why and how. *Journal of Neural Engineering*, 2(1):S91–S104.

Elfwing, S., Uchibe, E., and Doya, K. (2015). Expected energy-based restricted Boltzmann machine for classification. *Neural Networks*, 64:29–38.

Enroth-Cugell, C. and Robson, J. G. (1966). The contrast sensitivity of retinal ganglion cells of the cat. *The Journal of Physiology*, 187(3):517–552.

Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. Technical report, University of Montreal.

Eslami, S. M. A., Heess, N., and Winn, J. (2012). The shape Boltzmann machine: a strong model of object shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Euler, T., Haverkamp, S., Schubert, T., and Baden, T. (2014). Retinal bipo-

lar cells: elementary building blocks of vision. *Nature Reviews Neuroscience*, 15(8):507–519.

Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47.

Feng, J. and Darrell, T. (2015). Learning the structure of deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*.

Fernández, S., Graves, A., and Schmidhuber, J. (2007). Sequence labelling in structured domains with hierarchical recurrent neural networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Field, G. D. and Chichilnisky, E. J. (2007). Information processing in the primate retina: Circuitry and coding. *Annual Review of Neuroscience*, 30:1–30.

Field, G. D., Gauthier, J. L., Sher, A., Greschner, M., Machado, T. A., Jepson, L. H., Shlens, J., Gunning, D. E., Mathieson, K., Dabrowski, W., Paninski, L., Litke, A. M., and Chichilnisky, E. J. (2010). Functional connectivity in the retina at the resolution of photoreceptors. *Nature*, 467(7316):673–677.

Field, G. D., Sampath, A. P., and Rieke, F. (2005). Retinal processing near absolute threshold: from behavior to mechanism. *Annual Review of Physiology*, 67:491–514.

Fohlmeister, J. F., Coleman, P. A., and Miller, R. F. (1990). Modeling the repetitive firing of retinal ganglion cells. *Brain Research*, 510(2):343–345.

Fohlmeister, J. F. and Miller, R. F. (1997). Impulse encoding mechanisms of ganglion cells in the tiger salamander retina. *Journal of Neurophysiology*, 78(4):1935–1947.

Freund, J., Brandmaier, A. M., Lewejohann, L., Kirste, I., Kritzler, M., Krüger, A., Sachser, N., Lindenberger, U., and Kempermann, G. (2013). Emergence of individuality in genetically identical mice. *Science*, 340(6133):756–759.

Frey, B. J. (1997). Continuous sigmoidal belief networks trained using slice sampling. In *Advances in Neural Information Processing*.

Friedman, N. (1997). Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the International Conference on Machine Learning*.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Gaudiano, P. (1992). A unified neural network model of spatiotemporal processing in X and Y retinal ganglion cells. *Biological Cybernetics*, 67(1):23–34.

Georgiev, K. and Nakov, P. (2013). A non-IID framework for collaborative filtering with restricted Boltzmann machines. In *Proceedings of the International Conference on Machine Learning*.

Ghahramani, Z. (2012). Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110553.

Giusti, A., Guzzi, J., Ciresan, D., He, F.-L., Rodriguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., Scaramuzza, D., and Gambardella, L. (2016). A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters (in press)*.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Gollisch, T. and Meister, M. (2010). Eye smarter than scientists believed: Neural computations in circuits of the retina. *Neuron*, 65(2):150–164.

Golombek, D. A. and Rosenstein, R. E. (2010). Physiology of circadian entrainment. *Physiological Reviews*, 90(3):1063–1102.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. In *Proceedings of the International Conference on Machine Learning*.

Graham, D. J. and Field, D. J. (2009). Natural images: Coding efficiency.

Encyclopedia of Neuroscience, 6:19–27.

Graves, A., Mohamed, A., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Griffiths, T. L. and Ghahramani, Z. (2005). Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing*.

Griffiths, T. L. and Ghahramani, Z. (2011). The Indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224.

Groner, R. and Groner, M. T. (1989). Attention and eye movement control: An overview. *European Archives of Psychiatry and Neurological Sciences*, 239(1):9–16.

Hack, I., Peichl, L., and Brandstätter, J. H. (1999). An alternative pathway for rod signals in the rodent retina: Rod photoreceptors, cone bipolar cells, and the localization of glutamate receptors. *Proceedings of the National Academy of Sciences of the United States of America*, 96(24):14130–14135.

Hallum, L. E., Suaning, G. J., Taubman, D. S., and Lovell, N. H. (2004). Towards photosensor movement-adaptive image analysis in an electronic retinal prosthesis. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015a). Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015b). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

Hennig, M. H. and Funke, K. (2001). A biophysically realistic simulation of the

vertebrate retina. *Neurocomputing*, 38-40:659–665.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in Cognitive Sciences*, 11(10):428–434.

Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In Montavon, G., Orr, G. B., and Müller, K.-R., editors, *Neural Networks: Tricks of the Trade*, volume 7700 of *Lecture Notes in Computer Science*, pages 599–619. Springer Berlin Heidelberg.

Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.

Hinton, G. E. and Salakhutdinov, R. R. (2006a). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hinton, G. E. and Salakhutdinov, R. R. (2006b). Supporting online material for "Reducing the dimensionality of data with neural networks". *Science*, 313(5786):504–507.

Hinton, G. E. and Sejnowski, T. J. (1983). Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hoffman, J. E. and Subramaniam, B. (1995). The role of visual attention in saccadic eye movements. *Perception & Psychophysics*, 57(6):787–795.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.

Hosoya, T., Baccus, S. A., and Meister, M. (2005). Dynamic predictive coding by the retina. *Nature*, 436(7047):71–77.

Huang, G. B., Mattar, M. A., Lee, H., and Learned-Miller, E. (2012). Learning to align from scratch. In *Advances in Neural Information Processing*.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154.

Humayun, M. S., de Juan Jr, E., Dagnelie, G., Greenberg, R. J., Propst, R. H., and Phillips, D. H. (1996). Visual perception elicited by electrical stimulation of retina in blind humans. *Archives of Ophthalmology*, 114(1):40–46.

Ivakhnenko, A. G. (1971). Polynomial theory of complex systems. *IEEE Transactions on Systems, Man and Cybernetics*, 1(4):364–378.

Ivakhnenko, A. G. and Lapa, V. G. (1965). *Cybernetic Predicting Devices*. Purdue University School of Electrical Engineering.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. In *Advances in Neural Information Processing*.

Jusuf, P. R., Martin, P. R., and Grünert, U. (2006). Random wiring in the midget pathway of primate retina. *The Journal of Neuroscience*, 26(15):3908–3917.

Kae, A., Sohn, K., Lee, H., and Learned-Miller, E. (2013). Augmenting CRFs with Boltzmann machine shape priors for image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Kameneva, T., Meffin, H., and Burkitt, A. N. (2011). Modelling intrinsic electrophysiological properties of ON and OFF retinal ganglion cells. *Journal of Computational Neuroscience*, 31(3):547–561.

Kaneko, A. (1970). Physiological and morphological identification of horizontal, bipolar and amacrine cells in goldfish retina. *The Journal of Physiology*, 207(3):623–633.

Kavukcuoglu, K., Sermanet, P., Boureau, Y. L., Gregor, K., Mathieu, M., and LeCun, Y. (2010). Learning convolutional feature hierarchies for visual recognition. In *Advances in Neural Information Processing*.

Kien, T. T., Maul, T., and Bargiela, A. (2012a). A review of retinal prosthesis approaches. In *International Journal of Modern Physics: Conference Series*, volume 9.

Kien, T. T., Maul, T., Ren, L. J., and Bargiela, A. (2012b). Outer plexiform layer receptive fields as underlying factors of the Hermann grid illusion. In *Proceedings of the IEEE-EMBS International Conference on Biomedical Engineering and Sciences*.

Kolb, B. and Whishaw, I. Q. (1998). Brain plasticity and behavior. *Annual Review of Psychology*, 49(1):43–64.

Kolb, H. (1995). Simple anatomy of the retina. *Webvision: The Organization of the Retina and Visual System*. Retrieved Sep 10, 2013, from `http://webvision.med.utah.edu/`.

Kolb, H. (2003). How the retina works. *American Scientist*, 91(1):28–35.

Kolb, H., Nelson, R., Fernandez, E., and Jones, B., editors (Web, from 1995). *Webvision: The Organization of the Retina and Visual System*. Moran Eye Center. Retrieved Aug 22, 2015, from `http://webvision.med.utah.edu/`.

Konen, C. S. and Kastner, S. (2008). Two hierarchically organized neural systems for object information in human visual cortex. *Nature Neuroscience*, 11(2):224–231.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing*.

Kuffler, S. W. (1953). Discharge patterns and functional organization of mammalian retina. *Journal of Neurophysiology*, 16(1):37–68.

Kuremoto, T., Kimura, S., Kobayashi, K., and Obayashi, M. (2014). Time series forecasting using a deep belief network with restricted Boltzmann machines.

Neurocomputing, 137:47–56.

Kustov, A. A. and Robinson, D. L. (1996). Shared neural control of attentional shifts and eye movements. *Nature*, 384(6604):74–77.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the International Conference on Machine Learning*.

Lauritzen, T. Z. and Miller, K. D. (2003). Different roles for simple-cell and complex-cell inhibition in V1. *The Journal of Neuroscience*, 23(32):10201–10213.

Le, Q. V., Monga, R., Devin, M., Corrado, G., Chen, K., Ranzato, M. A., Dean, J., and Ng, A. Y. (2012). Building high-level features using large scale unsupervised learning. In *Proceedings of the International Conference on Machine Learning*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2006). Efficient sparse coding algorithms. In *Advances in Neural Information Processing*.

Lee, H., Ekanadham, C., and Ng, A. (2008). Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing*.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009a). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the International Conference on Machine Learning*.

Lee, H., Largman, Y., Pham, P., and Ng, A. Y. (2009b). Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing*.

Lennie, P., Haake, P. W., and Williams, D. R. (1991). The design of chromatically opponent receptive fields. In Landy, M. and Movshon, J. A., editors, *Computational Models of Visual Processing*, pages 71–82. MIT Press, Cam-

bridge, MA.

Li, K., Gao, J., Guo, S., Du, N., Li, X., and Zhang, A. (2014a). LRBM: A restricted Boltzmann machine based approach for representation learning on linked data. In *IEEE International Conference on Data Mining*.

Li, X., Du, N., Li, H., Li, K., Gao, J., and Zhang, A. (2014b). A deep learning approach to link prediction in dynamic networks. In *Proceedings of the SIAM International Conference on Data Mining*.

Li, Y., Liu, W., Li, X., Huang, Q., and Li, X. (2014c). GA-SIFT: A new scale invariant feature transform for multispectral image using geometric algebra. *Information Sciences*, 281:559–572.

Lin, T.-C. and Lin, C.-M. (2009). Wavelet-based copyright-protection scheme for digital images based on local features. *Information Sciences*, 179(19):3349–3358.

Linnainmaa, S. (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's thesis, University of Helsinki.

Linnainmaa, S. (1976). Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160.

Liu, F., Liu, B., Sun, C., Liu, M., and Wang, X. (2013). Deep learning approaches for link prediction in social network services. In *Neural Information Processing*, volume 8227 of *Lecture Notes in Computer Science*.

Liu, X.-D. and Kourennyi, D. E. (2004). Effects of tetraethylammonium on Kx channels and simulated light response in rod photoreceptors. *Annals of Biomedical Engineering*, 32(10):1428–1442.

Liversedge, S. P. and Findlay, J. M. (2000). Saccadic eye movements and cognition. *Trends in Cognitive Sciences*, 4(1):6–14.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Maclaurin, D., Duvenaud, D., and Adams, R. P. (2015). Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the International Conference on Machine Learning*.

Mancuso, K., Hauswirth, W. W., Li, Q., Connor, T. B., Kuchenbecker, J. A., Mauck, M. C., Neitz, J., and Neitz, M. (2009). Gene therapy for red–green colour blindness in adult primates. *Nature*, 461(7265):784–787.

Maniak, T., Jayne, C., Iqbal, R., and Doctor, F. (2015). Automated intelligent system for sound signalling device quality assurance. *Information Sciences*, 294(0):600–611.

Masland, R. H. (1996). Processing and encoding of visual information in the retina. *Current Opinion in Neurobiology*, 6(4):467–474.

Masland, R. H. (2001). The fundamental plan of the retina. *Nature Neuroscience*, 4(9):877–886.

Masland, R. H. (2012). The neuronal organization of the retina. *Neuron*, 76(2):266–280.

Maul, T. (2013). Early experiments with neural diversity machines. *Neurocomputing*, 113:36–48.

Maul, T. H., Bargiela, A., and Ren, L. J. (2011). Cybernetics of vision systems: Toward an understanding of putative functions of the outer retina. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 41(3):398–409.

McMahon, M. J., Packer, O. S., and Dacey, D. M. (2004). The classical receptive field surround of primate parasol ganglion cells is mediated primarily by a non-GABAergic pathway. *The Journal of Neuroscience*, 24(15):3736–3745.

Meila, M. and Jordan, M. I. (2001). Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48.

Memisevic, R. and Hinton, G. (2007). Unsupervised learning of image transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Miller, R. F., Staff, N. P., and Velte, T. J. (2006). Form and function of ON-OFF amacrine cells in the amphibian retina. *Journal of Neurophysiology*, 95(5):3171–3190.

Mittelman, R., Kuipers, B., Savarese, S., and Lee, H. (2014). Structured recurrent temporal restricted Boltzmann machines. In *Proceedings of the International Conference on Machine Learning*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Mordvintsev, A., Olah, C., and Tyka, M. (2015). Inceptionism: going deeper into neural networks. *Google Research Blog*, 20.

Morgan, J. and Wong, R. (2007). Development of cell types and synaptic connections in the retina. *Webvision: The Organization of the Retina and Visual System*. Retrieved Sep 10, 2013, from `http://webvision.med.utah.edu/`.

Moritoh, S., Komatsu, Y., Yamamori, T., and Koizumi, A. (2013). Diversity of retinal ganglion cells identified by transient GFP transfection in organotypic tissue culture of adult marmoset monkey retina. *PloS ONE*, 8(1):e54667.

Münch, T. A., da Silveira, R. A., Siegert, S., Viney, T. J., Awatramani, G. B., and Roska, B. (2009). Approach sensitivity in the retina processed by a multifunctional neural circuit. *Nature Neuroscience*, 12(10):1308–1316.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the International Conference on Machine Learning*.

Nie, S., Wang, Z., and Ji, Q. (2015). A generative restricted Boltzmann machine based method for high-dimensional motion data modeling. *Computer Vision and Image Understanding*, 136:14–22.

Nirenberg, S. and Pandarinath, C. (2012). Retinal prosthetic strategy with the capacity to restore normal vision. *Proceedings of the National Academy of Sciences*, 109(37):15012–15017.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Ölveczky, B. P., Baccus, S. A., and Meister, M. (2003). Segregation of object and background motion in the retina. *Nature*, 423(6938):401–408.

Ölveczky, B. P., Baccus, S. A., and Meister, M. (2007). Retinal adaptation to object motion. *Neuron*, 56(4):689–700.

Pahlberg, J. and Sampath, A. P. (2011). Visual threshold is set by linear and nonlinear mechanisms in the retina that mitigate noise: How neural circuits in the retina improve the signal-to-noise ratio of the single-photon response. *BioEssays*, 33(6):438–447.

Pascual-Leone, A., Amedi, A., Fregni, F., and Merabet, L. B. (2005). The plastic human brain cortex. *Annual Review of Neuroscience*, 28:377–401.

Pascual-Leone, A., Freitas, C., Oberman, L., Horvath, J. C., Halko, M., Eldaief, M., Bashir, S., Vernet, M., Shafi, M., Westover, B., Vahabzadeh-Hagh, A. M., and Rotenberg, A. (2011). Characterizing brain cortical plasticity and network dynamics across the age-span in health and disease with TMS-EEG and TMS-fMRI. *Brain Topography*, 24(3-4):302–315.

Paulus, W. and Kröger-Paulus, A. (1983). A new concept of retinal colour coding. *Vision Research*, 23(5):529–540.

Pelikan, M., Sastry, K., and Goldberg, D. E. (2001). Evolutionary algorithms + graphical models = scalable black-box optimization. Technical report, Illinois Genetic Algorithms Laboratory, Department of General Engineering, University of Illinois at Urbana-Champaign.

Pérez De Sevilla Müller, L., Shelley, J., and Weiler, R. (2007). Displaced amacrine cells of the mouse retina. *Journal of Comparative Neurology*, 505(2):177–189.

Pfister, T., Charles, J., and Zisserman, A. (2015). Flowing ConvNets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*.

Pfister, T., Simonyan, K., Charles, J., and Zisserman, A. (2014). Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Proceedings of the Asian Conference on Computer Vision*.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J., and Simoncelli, E. P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999.

Piscopo, D. M., El-Danaf, R. N., Huberman, A. D., and Niell, C. M. (2013). Diverse visual features encoded in mouse lateral geniculate nucleus. *The Journal of Neuroscience*, 33(11):4642–4656.

Pitman, J. (2002). Combinatorial stochastic processes. Technical report, Department of Statistics, University of California, Berkeley. Lecture notes for St. Flour course.

Poon, H. and Domingos, P. (2011). Sum-product networks: A new deep architecture. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Prasad, S. C. and Prasad, P. (2014). Deep recurrent neural networks for time series prediction. *arXiv preprint arXiv:1407.5949*.

Protti, D. A., Flores-Herr, N., and von Gersdorff, H. (2000). Light evokes Ca^{2+} spikes in the axon terminal of a retinal bipolar cell. *Neuron*, 25(1):215–227.

Publio, R., Oliveira, R. F., and Roque, A. C. (2009). A computational study on the role of gap junctions and rod Ih conductance in the enhancement of the dynamic range of the retina. *PloS ONE*, 4(9):e6970–e6970.

Purves, D., Augustine, G. J., Fitzpatrick, D., Katz, L. C., LaMantia, A. S., McNamara, J. O., and Williams, S. M., editors (2001). *Neuroscience, Second Edition*, chapter 11 Vision: The eye – Phototransduction. Sinauer Associates, Inc., Sunderland, MA.

Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. (2006). Efficient learning

of sparse representations with an energy-based model. In *Advances in Neural Information Processing*.

Ranzato, M., Susskind, J., Mnih, V., and Hinton, G. (2011). On deep generative models with applications to recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ringach, D. L., Shapley, R. M., and Hawken, M. J. (2002). Orientation selectivity in macaque V1: Diversity and laminar dependence. *The Journal of Neuroscience*, 22(13):5639–5651.

Robinson, D. A. (1972). Eye movements evoked by collicular stimulation in the alert monkey. *Vision Research*, 12(11):1795–1808.

Rodieck, R. W. (1965). Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*, 5(12):583–601.

Roska, B. and Werblin, F. (2003). Rapid global shifts in natural scenes block spiking in specific ganglion cell types. *Nature Neuroscience*, 6(6):600–608.

Rozantsev, A., Lepetit, V., and Fua, P. (2015). On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 137:24–37.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: an efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision*.

Salakhutdinov, R. R. and Hinton, G. E. (2007). Using deep belief nets to learn covariance kernels for Gaussian processes. In *Advances in Neural Information Processing*.

Salakhutdinov, R. R. and Hinton, G. E. (2009a). Deep Boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Salakhutdinov, R. R. and Hinton, G. E. (2009b). Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978.

Salakhutdinov, R. R., Mnih, A., and Hinton, G. E. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the International Conference on Machine Learning*.

Sarpeshkar, R. (1998). Analog versus digital: Extrapolating from electronics to neurobiology. *Neural Computation*, 10(7):1601–1638.

Schmidt, T. M., Chen, S.-K., and Hattar, S. (2011). Intrinsically photosensitive retinal ganglion cells: many subtypes, diverse functions. *Trends in Neurosciences*, 34(11):572–580.

Schmolesky, M. (2000). The primary visual cortex. *Webvision: The Organization of the Retina and Visual System*. Retrieved Oct 30, 2015, from `http://webvision.med.utah.edu/`.

Schneidman, E., Bialek, W., and Berry II, M. J. (2003). Synergy, redundancy, and independence in population codes. *The Journal of Neuroscience*, 23(37):11539–11553.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Schwartz, G., Taylor, S., Fisher, C., Harris, R., and Berry, M. J. (2007). Synchronized firing among retinal ganglion cells signals motion reversal. *Neuron*, 55(6):958–969.

Shepherd, M., Findlay, J. M., and Hockey, R. J. (1986). The relationship between eye movements and spatial attention. *The Quarterly Journal of Experimental Psychology Section A: Human Experimental Psychology*, 38(3):475–491.

Sherrington, C. S. (1906). Observations on the scratch-reflex in the spinal dog. *The Journal of Physiology*, 34(1-2):1–50.

Shim, V. A., Tan, K. C., Cheong, C. Y., and Chia, J. Y. (2013). Enhancing the scalability of multi-objective optimization via restricted Boltzmann machine-based estimation of distribution algorithm. *Information Sciences*, 248(0):191–213.

Shlens, J., Rieke, F., and Chichilnisky, E. J. (2008). Synchronized firing in the retina. *Current Opinion in Neurobiology*, 18(4):396–402.

Shu, Y., Hasenstaub, A., Duque, A., Yu, Y., and McCormick, D. A. (2006). Modulation of intracortical synaptic potentials by presynaptic somatic membrane potential. *Nature*, 441(7094):761–765.

Simoncelli, E. P. and Olshausen, B. A. (2001). Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*.

Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*.

Smith, R. G. and Vardi, N. (1995). Simulation of the Aii amacrine cell of mammalian retina: Functional consequences of electrical coupling and regenerative membrane properties. *Visual Neuroscience*, 12:851–860.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1: Foundations, pages 194–281. MIT Press, Cambridge, MA.

Soltesz, I. (2006). *Diversity in the neuronal machine: Order and variability in interneuronal microcircuits*. Oxford University Press.

Soucy, E., Wang, Y., Nirenberg, S., Nathans, J., and Meister, M. (1998). A novel signaling pathway from rod photoreceptors to ganglion cells in mammalian retina. *Neuron*, 21(3):481–493.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Stevenson, I. H., Rebesco, J. M., Miller, L. E., and Körding, K. P. (2008). Inferring functional connections between neurons. *Current Opinion in Neurobiology*, 18(6):582–588.

Su, H., Qi, C. R., Li, Y., and Guibas, L. J. (2015). Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. In *Proceedings of the IEEE International Conference on Computer Vision*.

Sun, Y., Liang, D., Wang, X., and Tang, X. (2015). DeepID3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*.

Sun, Y., Wang, X., and Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Sutskever, I. and Hinton, G. E. (2007). Learning multilevel distributed representations for high-dimensional sequences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Sutskever, I. and Hinton, G. E. (2008). Deep, narrow sigmoid belief networks are universal approximators. *Neural Computation*, 20(11):2629–2636.

Sutskever, I., Hinton, G. E., and Taylor, G. W. (2008). The recurrent temporal restricted Boltzmann machine. In *Advances in Neural Information Processing*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*.

Tao, W., Zhou, Y., Liu, L., Li, K., Sun, K., and Zhang, Z. (2014). Spatial adjacent bag of features with multiple superpixels for object segmentation and classification. *Information Sciences*, 281:373–385.

Taylor, G. W. and Hinton, G. E. (2009). Factored conditional restricted Boltzmann machines for modeling motion style. In *Proceedings of the International Conference on Machine Learning*.

Taylor, G. W., Hinton, G. E., and Roweis, S. T. (2006). Modeling human motion using binary latent variables. In *Advances in Neural Information Processing*.

Teeters, J., Jacobs, A., and Werblin, F. (1997). How neural interactions form neural responses in the salamander retina. *Journal of Computational Neuroscience*, 4(1):5–27.

Teeters, J. L., Eeckman, F. H., and Werblin, F. S. (1991). A four neuron circuit accounts for change sensitive inhibition in salamander retina. In *Advances in Neural Information Processing*.

Tian, N. (2008). Synaptic activity, visual experience and the maturation of retinal synaptic circuitry. *The Journal of Physiology*, 586(18):4347–4355.

Tompson, J. J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing*.

Toshev, A. and Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Troy, J. B. and Shou, T. (2002). The receptive fields of cat retinal ganglion cells in physiological and pathological states: Where we are after half a century of research. *Progress in Retinal and Eye Research*, 21(3):263–302.

Tsai, D., Chen, S., Protti, D. A., Morley, J. W., Suaning, G. J., and Lovell, N. H. (2012). Responses of retinal ganglion cells to extracellular electrical stimulation, from single cell to population: Model-based analysis. *PloS ONE*, 7(12):e53357.

Tsukamoto, Y., Morigiwa, K., Ueda, M., and Sterling, P. (2001). Microcircuits for night vision in mouse retina. *The Journal of Neuroscience*, 21(21):8616–8623.

Turcsany, D. and Bargiela, A. (2014). Learning local receptive fields in deep belief networks for visual feature detection. In *Neural Information Processing*, volume 8834 of *Lecture Notes in Computer Science*.

Turcsany, D., Bargiela, A., and Maul, T. (2014). Modelling retinal feature detection with deep belief networks in a simulated environment. In *Proceedings of the European Conference on Modelling and Simulation*.

Turcsany, D., Bargiela, A., and Maul, T. (2016). Local receptive field constrained deep networks. *Information Sciences*, 349–350:229–247.

Turcsany, D., Mouton, A., and Breckon, T. P. (2013). Improving feature-based object recognition for x-ray baggage security screening using primed visual words. In *Proceedings of the IEEE International Conference on Industrial Technology*.

Usui, S., Ishihaiza, A., Kamiyama, Y., and Ishii, H. (1996a). Ionic current model of bipolar cells in the lower vertebrate retina. *Vision Research*, 36(24):4069–4076.

Usui, S., Kamiyama, Y., Ishii, H., and Ikeno, H. (1996b). Reconstruction of retinal horizontal cell responses by the ionic current model. *Vision Research*, 36(12):1711–1719.

Vallerga, S., Covacci, R., and Pottala, E. W. (1980). Artificial cone responses: A computer-driven hardware model. *Vision Research*, 20(5):453–457.

Van Essen, D. C. and Maunsell, J. H. R. (1983). Hierarchical organization and functional streams in the visual cortex. *Trends in Neurosciences*, 6:370–375.

Vaney, D. I., Sivyer, B., and Taylor, W. R. (2012). Direction selectivity in the retina: symmetry and asymmetry in structure and function. *Nature Reviews Neuroscience*, 13(3):194–208.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY.

Velte, T. J. and Miller, R. F. (1997). Spiking and nonspiking models of starburst amacrine cells in the rabbit retina. *Visual Neuroscience*, 14(6):1073–1088.

Wässle, H. (2004). Parallel processing in the mammalian retina. *Nature Reviews Neuroscience*, 5(10):747–757.

Wässle, H., Peichl, L., and Boycott, B. B. (1981). Morphology and topography of on- and off-alpha cells in the cat retina. *Proceedings of the Royal Society of London B: Biological Sciences*, 212(1187):157–175.

Werblin, F. (1991). Synaptic connections, receptive fields, and patterns of activity in the tiger salamander retina. *Investigative Ophthalmology & Visual Science*, 32(3):459–483.

Werblin, F. and Jacobs, A. (1996). CNN-based retinal model uncovers a new form of edge enhancement in biological visual processing. In *Proceedings of the IEEE International Workshop on Cellular Neural Networks and their Applications*.

Werblin, F. S. and Dowling, J. E. (1969). Organization of the retina of the mudpuppy, Necturus maculosus. II. Intracellular recording. *Journal of Neurophysiology*, 32(3):339–355.

Werbos, P. J. (1982). Applications of advances in nonlinear sensitivity analysis. In *System Modeling and Optimization*, volume 38 of *Lecture Notes in Control and Information Sciences*.

Wessinger, C. M., VanMeter, J., Tian, B., Van Lare, J., Pekar, J., and Rauschecker, J. P. (2001). Hierarchical organization of the human auditory cortex revealed by functional magnetic resonance imaging. *Journal of Cognitive Neuroscience*, 13(1):1–7.

Wright, K. P., McHill, A. W., Birks, B. R., Griffin, B. R., Rusterholz, T., and Chinoy, E. D. (2013). Entrainment of the human circadian clock to the natural light-dark cycle. *Current Biology*, 23(16):1554–1558.

Wu, Y., Wang, Z., and Ji, Q. (2013). Facial feature tracking under varying facial expressions and face poses based on restricted Boltzmann machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Wurtz, R. H. and Goldberg, M. E. (1972). Activity of superior colliculus in behaving monkey. III. Cells discharging before eye movements. *Journal of Neurophysiology*, 35(4):575–586.

Xia, Y., Zhang, L., Xu, W., Shan, Z., and Liu, Y. (2015). Recognizing multiview objects with occlusions using a deep architecture. *Information Sciences*, 320:333–345.

Xie, P., Deng, Y., and Xing, E. (2015). Diversifying restricted Boltzmann machine for document modeling. In *Proceedings of the ACM SIGKDD Interna-*

tional Conference on Knowledge Discovery and Data Mining.

Xu, L., Li, Y., Wang, Y., and Chen, E. (2015). Temporally adaptive restricted Boltzmann machine for background modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zaghloul, K. A. and Boahen, K. (2006). A silicon retina that reproduces signals in the optic nerve. *Journal of Neural Engineering*, 3(4):257–267.

Zhang, C., Cheng, J., Zhang, Y., Liu, J., Liang, C., Pang, J., Huang, Q., and Tian, Q. (2015). Image classification using boosted local features with random orientation and location selection. *Information Sciences*, 310:118–129.

Zhao, F., Huang, Y., Wang, L., Xiang, T., and Tan, T. (2016). Learning relevance restricted Boltzmann machine for unstructured group activity and event understanding. *International Journal of Computer Vision (in press)*, pages 1–17.

Zhu, Z., Luo, P., Wang, X., and Tang, X. (2013). Deep learning identity-preserving face space. In *Proceedings of the IEEE International Conference on Computer Vision*.