Neural Processing Letters manuscript No.
(will be inserted by the editor)

Fitness Landscape Analysis of Weight-Elimination Neural Networks

Anna Bosman · Andries Engelbrecht · Mardé Helbig

Received: date / Accepted: date

Abstract Neural network architectures can be regularised by adding a penalty term to the objective function, thus minimising network complexity in addition to the error. However, adding a term to the objective function inevitably changes the surface of the objective function. This study investigates the landscape changes induced by the weight elimination penalty function under various parameter settings. Fitness landscape metrics are used to quantify and visualise the induced landscape changes, as well as to propose sensible ranges for the regularisation parameters. Fitness landscape metrics are shown to be a viable tool for neural network objective function landscape analysis and visualisation.

Keywords Neural networks · Fitness landscapes · Regularisation · Weight elimination

1 Introduction

Despite being studied for decades, and successfully applied in numerous areas [1, 8], neural networks (NNs) remain to this day black box models, inner workings of which are hard to characterise and visualise. In particular, the shape of the objective functions associated with supervised NN training is poorly understood [5]. Certain landscape properties of the NN objective functions, such as the presence

A. S. Bosman (Rakitianskaia)
Department of Computer Science, University of Pretoria,
Pretoria, South Africa
E-mail: annar@cs.up.ac.za
orcid.org/0000-0003-3546-1467

A. P. Engelbrecht
Department of Computer Science, University of Pretoria,
Pretoria, South Africa
E-mail: engel@cs.up.ac.za

M. Helbig
Department of Computer Science, University of Pretoria,
Pretoria, South Africa
E-mail: mhelbig@cs.up.ac.za

of saddle points [7, 17], plateaus, and narrow ridges [11, 19], have been established, but the relationship between these landscape features and corresponding NN parameters, such as the number of neurons and hidden layers, or the activation functions employed, remains unclear [22].

Empirical studies of the link between the objective function landscape characteristics and the different NN parameters can be performed using fitness landscape analysis (FLA). FLA is a relatively recent field of computational intelligence, applied for the first time in evolutionary computation for algorithm performance prediction [20, 32]. FLA estimates and quantifies topographical properties of an objective function landscape, such as ruggedness, neutrality, and searchability. The obtained metrics can be subsequently used to better understand the given optimisation problem, and make an intelligent algorithm choice [24, 27, 37]. The properties of fitness landscapes are estimated by taking multiple random samples of the search space, calculating the fitness value for every sampling point, and quantifying the relationship between the spatial characteristics of the sample points and the corresponding fitness values [27, 37]. Sample analysis makes no assumptions regarding the problem at hand, and can easily be applied to "black box" optimisation problems such as NNs.

The ability of a NN to correctly predict the outputs of input patterns not seen during training is known as the generalisation ability. A model that cannot generalise has no practical use, therefore maximising the generalisation potential of a NN is a major goal of NN training. A simple, yet effective way to improve the generalisation ability of a NN is to add a weight regularisation term to the objective function [33, 34, 42]. Weight regularisation aims to penalise network complexity by decreasing the rate of weight growth, as well as by driving irrelevant weights to zero. Regularisation was shown to be beneficial in practical NN applications [30, 36, 42]. Therefore, investigating the effect of regularisation on the NN training problem is important.

It is easy to understand the regularisation process intuitively: if large and irrelevant weights are penalised, the final model will be more compact. It is, however, harder to imagine the surface of the objective function after a penalty term has been added to it – will the penalty term introduce new optima, or make the function smoother? Will the chosen training algorithm find the problem easier or harder to optimise?

The relationship between the regularisation term and the resulting error surface is far from trivial [12], especially given the fact that regularisation parameters typically have to be empirically tuned before an improvement in generalisation performance is observed. One way to investigate the relationship between the regularisation parameters and the resulting error surface is to use FLA techniques. FLA provides an easy and convenient method to quantify and visualise the correlation between the error landscape changes and the chosen regularisation scheme. This study applies selected FLA metrics to study the NN error surfaces under the weight elimination regularisation scheme. The obtained results provide interesting insights into the nature of regularised NN error surfaces, give some guidance for the corresponding parameter tuning, and set the path for future applications of FLA in the NN context.

The rest of the paper is structured as follows: Section 2 discusses weight elimination in NNs. Section 3 describes the FLA metrics used in this study, and the applicability of FLA in the NN context. Section 4 describes the experimental proce-

dure. Section 5 presents the empirical study of the effects of the weight elimination term on the NN error surfaces. Section 6 concludes the paper and lists potential topics for future research.

2 Neural Network Weight Elimination

The sum squared error (SSE) is one of the most commonly used NN objective functions:

$$E_{sse} = \sum_{p=1}^{P} \sum_{k=1}^{K} (t_{k,p} - o_{k,p})^2 \tag{1}$$

where P is the total number of training patterns, K is the total number of output units, $t_{k,p}$ is the k-th target value for pattern p, and $o_{k,p}$ is the k-th output obtained for pattern p. Minimisation of the SSE minimises the overall NN error.

Weight regularisation is applied to minimise both NN error and NN complexity. If E_p is a penalty function that quantifies the complexity of a NN, the objective function can be modified as follows:

$$E_{nn} = E_{sse} + \lambda E_p \tag{2}$$

where λ is a hyperparameter controlling the "strength" of regularisation. If λ is too small, the value of the penalty function will be much smaller than the error value, and the error is likely to "overshadow" the penalty, thus causing the penalty to be disregarded. On the other hand, if λ is too big, the penalty contribution to the objective function will become larger than the error term contribution, and the algorithm will focus on minimising the NN complexity instead of minimising the error. In practice, λ is chosen empirically per problem and per penalty function E_p.

The complexity of a NN can be expressed by the overall number of NN weights. Simplistic architectures with too few weights may be incapable of learning a complex problem representation. Excessive architectures with too many weights, on the other hand, may promote overfitting. Thus, penalty functions are usually designed to optimise the total number of NN weights.

A well-known L2 (i.e. quadratic) penalty function proposed in the literature is weight decay [18], given by

$$E_p = \frac{1}{2} \sum_{l=1}^{W} w_l^2 \tag{3}$$

where W is the total number of weights in the NN, and w_l is the l-th weight. The weight decay penalty essentially calculates the magnitude of the weight vector. The larger the magnitude, the more the NN will be penalised. Limiting the weight growth tends to improve NN generalisation [34], since the relevant weights are reinforced by the training algorithm at every iteration, while the irrelevant ones decay towards zero over time.

A disadvantage of weight decay is that no differentiation between relevant and irrelevant weights is explicitly made, thus both large and small weights are penalised with the same rigour. Weigend *et al* [46] introduced an alternative L2

penalty function, which uses an extra parameter w_0 to specify the threshold that separates relevant weights from irrelevant weights:

$$E_p = \sum_{l=1}^{W} \frac{w_l^2/w_0^2}{1 + w_l^2/w_0^2} \tag{4}$$

This penalty function is known as weight elimination. Parameter w_0 defines a threshold that distinguishes between significantly and insignificantly large weights. Weights with $|w| >> w_0$ yield a complexity cost close to 1, and contribute towards the penalty term in Equation 4. Thus, weights with $|w| >> w_0$ are seen as "too large" and in need of regularisation. Weights with $|w| << w_0$ yield a complexity cost close to zero, and contribute very little to the weight elimination term. Thus, weights with $|w| << w_0$ are not penalised. A small w_0 value will result in more weights being penalised, thus only the most persistent weights will survive, yielding an architecture comprised of few larger weights. On the other hand, for a large w_0 value, small weights will not be subject to the penalty, resulting in an architecture made up of many small weights.

The preference of a few large weights or many small weights is problem-dependent, although it should be noted that large weights may cause the NN to saturate. Saturation occurs when the hidden neurons of a NN predominantly output values close to the asymptotic ends of the activation function range. The output of the hidden unit is determined by the magnitude of the weighted sum of inputs, or, in other words, the "strength" of the input signal. Very large weights increase the signal strength, causing the bounded activation functions to output near-asymptotic values. Saturated neurons are undesirable, because derivatives are very small near the asymptotes, which cause a significant slow down in gradient descent learning [13]. Non-gradient learning, such as particle swarm optimisation, can also be hindered by NN saturation [40].

Weight elimination allows a refined, problem-specific approach to NN regularisation. A recent study by Wang et al [43] provided a theoretical analysis of boundedness and convergence of the weight-elimination NNs, and confirmed good generalisation and pruning capabilities of weight elimination. However, there are two parameters that need to be tuned: λ and w_0. This study analyses the relationship between different settings of these two parameters and the corresponding NN error landscapes. A sensible parameter optimisation range for λ and w_0 is proposed. It should be noted that weight elimination was chosen based on its relative simplicity; even though other more complex regularisation schemes have also been proposed in the literature, an investigation of the relationship between the penalty function and the objective function must begin at the most interpretable point. While weight decay is perhaps too trivial to provide interesting insight, and simply imposes a quadratic convex shape on the original objective function, weight elimination, with its two tunable parameters, is harder to visualise intuitively [43]. It was also shown that modern regularisation techniques benefit when combined with simpler L2 penalty functions [42]. Thus, the results of this study can be extended and applied to numerous recently proposed regularisation schemes.

3 Fitness Landscape Analysis

A fitness landscape refers to the hypersurface formed by the objective function values as calculated across the search space. The goal of fitness landscape analysis (FLA) is to estimate and quantify various features of the objective function hypersurface, such as ruggedness, neutrality, and searchability, and to discover correlations between landscape features and algorithm performance [24, 37]. The term "fitness landscape" was defined in the evolutionary optimisation community, and the FLA techniques were originally developed for discrete binary search spaces. However, the notion of fitness landscapes was soon extended to continuous spaces [27, 37]. Any optimisation problem with a well-defined objective function can be studied from the FLA perspective.

To estimate fitness landscape characteristics, multiple samples of the search space are taken. The objective function value for every point in each sample is then calculated. FLA metrics provide different ways of quantifying the relationship between the spatial and the qualitative characteristics of the sample points. The sample-based conclusions provide useful estimates of the fitness landscape properties for the given optimisation problem. FLA metrics were shown to be descriptive on a wide selection of continuous benchmark functions [27, 31]. FLA metrics were also successfully used for algorithm performance prediction [24] and algorithm selection [35] on continuous benchmark functions of up to 30 dimensions.

This section presents an overview of FLA in the NN context, and describes the FLA metrics used in this study. Section 3.1 describes the existing applications of FLA to NNs. Section 3.2 discusses the gradient estimate metrics. Section 3.3 discusses the ruggedness metric based on information entropy. Section 3.4 discusses the searchability metric.

3.1 Error Landscapes of Neural Networks

NN training is the process of finding the best possible combination of weights that connect the neurons between layers. Each unique combination of weights can be treated as a candidate solution that represents the mapping between the inputs and the outputs. Thus, given m weights and biases, the search space is a continuous m-dimensional space of all possible weight combinations. The complete search space of all possible NN weight vectors with corresponding error values constitutes the "error landscape" of a NN.

Error landscapes of NNs can thus be treated as fitness landscapes of continuous optimisation problems. An important property specific to NNs is that NN error landscapes are unbounded, since each weight is defined as any number in $\mathbb{R}$. This poses a problem to sampling algorithms, as no amount of sampling is guaranteed to adequately cover the space between minus infinity and plus infinity. The problem is solved by focusing on the areas of the search space that the algorithms actually explore, and where acceptable solutions can in fact be found [2].

Error landscapes have been studied before in an attempt to understand the inner workings of NNs. Gallagher [11] used principal component analysis (PCA) to simplify the weight space in order to visualise NN training trajectories. Gallagher [11] found that error landscapes have many flat areas with sudden cliffs or ravines. PCA was further used in [22] to identify the factors that influence the NN error

surfaces. It was also shown that NN error landscapes exhibit more saddle points than local minima, and that the number of local minima diminishes exponentially as the dimensionality of the problem increases [7]. Malan and Engelbrecht's FLA metrics have been successfully applied to investigate the amount of information contained in SSE landscapes and classification error landscapes, and to study the effect of multiple hidden layers on the error landscape properties [41]. In the recent paper by Gonçalves et al [14], single-output NNs were constructed using geometric semantic genetic programming such that the resulting error landscape was guaranteed to be unimodal.

Even though some properties of the NN error landscapes have been identified, the relationship between various NN parameters and the resulting error surface remains an open problem [5]. This study contributes to the existing body of knowledge by applying a selection of FLA measures for the first time to investigate the relationship between NN regularisation parameters and the resulting error landscapes. The insights gained are used to propose sensible regions for the regularisation parameters involved. One of the goals of this paper is to illustrate that FLA is a powerful visualisation and analysis tool, deserving acknowledgement in the NN community.

3.2 Gradients

An important property of a fitness landscape is the magnitude of fitness changes, i.e. the gradient information available to a training algorithm. To quantify the fitness change magnitudes, Malan and Engelbrecht [26] proposed two gradient measures, i.e. the average estimated gradient G_{avg}, and the standard deviation of the gradient G_{dev}. G_{avg} and G_{dev} are calculated based on Manhattan progressive random walk [27] samples of the search space. G_{avg} is defined as:

$$G_{avg} = \frac{\sum_{t=0}^{T-1} |g(t)|}{T}$$

where T is the number of steps in the Manhattan progressive random walk, and $g(t)$ is defined as:

$$g(t) = \frac{\Delta e_t}{d(\mathbf{x}_t, \mathbf{x}_{t+1})}$$

where Δe_t is the difference between the error values of the weight vectors $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$, which define step t of the random walk, and $d(\mathbf{x}_t, \mathbf{x}_{t+1})$ is the Euclidean distance between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$. The absolute value of $g(t)$ is used in the G_{avg} calculation, since Δe_t can be either positive or negative (the error can increase or decrease). The aim of G_{avg} is to quantify the magnitude of changes rather than their direction, thus the sign of $g(t)$ is of no consequence. Positive values of $g(t)$ are also required to ensure that the negative error slopes do not cancel out the positive error slopes.

The standard deviation of the gradient G_{dev} is defined as:

$$G_{dev} = \sqrt{\frac{\sum_{t=0}^{T-1} (G_{avg} - |g(t)|)^2}{T-1}}$$

The G_{avg} metric quantifies the mean magnitude of change in fitness values, while G_{dev} represents the corresponding standard deviation. In the NN context, the presence of gradients is a desired characteristic, because gradient descent-based algorithms rely on gradient information. However, very steep gradients may yield abrupt changes in the weight vector, potentially causing the algorithm to "overshoot" regions with good optima.

In this study, Manhattan progressive random walks of 1000 steps were used, where the size of each step was equal to 1% of the search space. Pseudocode for the Manhattan progressive random walk, as well as G_{avg} and G_{dev} calculations, can be found in [24].

3.3 First entropic measure of ruggedness

The first entropic measure of ruggedness (FEM), proposed in [25], quantifies the level of ruggedness observed in a fitness landscape. A progressive random walk [29] through the search space is taken, generating a time series of fitness values $\{f_t\}_{t=0}^{n}$. A symbol sequence, $S(\epsilon) = s_1 s_2 ... s_n$, is generated from $\{f_t\}_{t=0}^{n}$, where $s_i \in \{\bar{1}, 0, 1\}$ is given by

$$s_i = \Psi_{f_t}(i, \epsilon) = \begin{cases} \bar{1} & \text{if } f_i - f_{i-1} < -\epsilon \\ 0 & \text{if } |f_i - f_{i-1}| \leq \epsilon \\ 1 & \text{if } f_i - f_{i-1} > \epsilon \end{cases}$$

where ϵ is the chosen sensitivity threshold. An entropic measure $H(\epsilon)$ is now defined as

$$H(\epsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]}$$

where $p, q \in \{\bar{1}, 0, 1\}$, and $P_{[pq]}$ is given by

$$P_{[pq]} = \frac{n_{[pq]}}{n}$$

where $n_{[pq]}$ is the number of sub-blocks pq in $S(\epsilon)$. Note that $p \neq q$, thus the total number of unique pq value combinations is 6. The value of $H(\epsilon)$ depends on the chosen value for ϵ. It was shown in [24, 25] that for a certain ϵ^*, $H(\epsilon^*)$ converges on the value of 0 for any $\{f_t\}_{t=0}^{n}$. The value of ϵ^* is defined as the smallest value of ϵ for which the landscape becomes flat. The first entropic measure of ruggedness (FEM) is calculated as follows:

$$FEM = \max_{\forall \epsilon \in [0, \epsilon^*]} \{H(\epsilon)\}$$

Two FEM measures are usually used to describe a fitness landscape: micro-ruggedness $FEM_{0.01}$, where the maximum size of the random walk step is equal to 1% of the objective function domain, and macro-ruggedness $FEM_{0.1}$, where the maximum size of the random walk step is equal to 10% of the objective function domain. Both were considered in this study. Each random walk consisted of 1000 steps. Pseudocode for the progressive random walk, as well as FEM calculations, can be found in [24].

The value of FEM is continuous and falls in the $[0, 1]$ range, where 0 indicates a smooth landscape (no entropy), and 1 indicates maximal ruggedness (highest entropy). In the NN context, a smooth landscape would be easier to search than a rugged one, provided that the smooth surface is inclined, i.e. contains enough gradient information to guide the search.

3.4 Fitness distance correlation

The fitness distance correlation (FDC) metric, proposed by Jones and Forrest [21], is designed to quantify global problem hardness. FDC estimates the global shape of the fitness landscape by calculating the covariance between the fitness of a solution and its distance to the nearest optimum.

FDC_s, proposed in [28], is an adaptation of FDC for continuous landscapes without known optima. FDC_s is defined as:

$$\text{FDC}_s = \frac{\sum_{i=1}^{n}(e_i - \overline{e})(d_i - \overline{d})}{\sqrt{\sum_{i=1}^{n}(e_i - \overline{e})^2}\sqrt{\sum_{i=1}^{n}(d_i - \overline{d})^2}}$$

where n is the size of a uniform sample of weight vectors, $\mathbf{W} = \{\mathbf{w}_1, ..., \mathbf{w}_n\}$, with associated error values $E = \{e_1, ..., e_n\}$; $\overline{e}$ is the mean of E, d_i is the Euclidean distance from $\mathbf{w}_i$ to the weight vector in the sample with the lowest error value, and $\overline{d}$ is the mean of all d_i.

FDC_s generates values in the range $[-1, 1]$. For minimisation problems, a value close to 1 indicates a highly searchable landscape, i.e., the closer the sample points are to the fittest point, the higher is their fitness. A value close to 0 indicates a lack of information in the landscape, i.e., points both far from the fittest point and close to the fittest point may have similar fitness values. A negative FDC_s value indicates a "deceptive" search landscape, i.e., approaching the fittest point in the sample may produce points of increasingly worse fitness.

Previous application of FDC in the NN context was done by Gallagher [10]. The aim of the study was to estimate the difficulty of training NNs, thus the weights obtained from each training epoch rather than a random sample were used as sample points. The optima found by the training algorithm was used in place of the global optima. Since a training algorithm was used for sampling, the samples were biased towards regions of the search space with higher gradients.

Note that FDC_s uses random uniform samples of the search space rather than samples gathered during the training. The main advantage of random samples and random walk samples over samples gathered along the training trajectory is that the random samples are independent of the training algorithm, thus the information gathered from the randomised samples is more objective, and provides a more general view of the error landscape characteristics.

4 Experimental Procedure

The aim of the experiments was to apply FLA metrics to regularised NN error landscapes, and to observe the influence of regularisation parameters on both error landscape characteristics and training algorithm performance. Thus, insight into

the regularised NN error landscapes can be gained, and the expressiveness of FLA metrics in the NN context can be evaluated.

The rest of the section is structured as follows: Section 4.1 outlines the benchmark problems used in this study, Section 4.2 describes the corresponding NN architectures, Section 4.3 describes the chosen search space boundaries, Section 4.4 lists the regularisation parameter settings investigated in this study, Section 4.5 describes the NN training algorithm used, and Section 4.6 lists the NN training algorithm parameters.

4.1 Benchmark problems

The three classification benchmark problems used in this study are outlined in Table 1. Well-known benchmarks were chosen for their relative interpretability. The NN architectures were adopted from the listed sources.

Table 1: Benchmark Problems

Problem	In	Hidden	Out	**Source**	Dimensionality
Iris	4	4	3	[16]	35
Diabetes	8	6	2	[3]	68
Glass	9	9	6	[16]	150

4.1.1 Iris

The Iris flower data set [9] contains 50 examples belonging to the three species of Iris flowers: *Iris setosa*, *Iris versicolor*, and *Iris virginica*. Four input variables are defined: sepal length, sepal width, petal length, and petal width. The iris data set, even though relatively low-dimensional and simple, is not altogether trivial, as two of the three output classes significantly overlap across two of the four input variables, and two inputs have low correlation with the class labels [15].

4.1.2 Diabetes

The diabetes data set [38] contains 768 patterns describing Pima Indian patients. Personal patient data is recorded, such as age and the results of medical examinations, e.g. blood pressure, body mass index, and glucose tolerance test result, among others. The patterns are divided into two classes: diabetes positive or diabetes negative. All inputs are continuous, and 65.1% of the examples are diabetes negative. The data set contains noise [38].

4.1.3 Glass

The glass data set [38] describes 6 glass types (float processed or non-float processed building windows, vehicle windows, containers, tableware, or head lamps) based on chemical analysis represented by the percentage content of eight different

elements. Recognizing the type of glass from the glass shard analysis has a practical application in forensic investigations. The data set consists of 214 examples, all inputs are continuous, and two of the inputs have very low correlation with the class labels. The frequency of the 6 classes are 70, 76, 17, 13, 9, and 29 instances, respectively.

4.2 Neural network architecture

This study considered feed-forward NNs with a single hidden layer. The input layer employed the identity (linear) activation function. The hidden and output layers employed the sigmoid activation function, given by $f_{NN}(net) = 1/(1 + e^{-net})$, where net is the weighted sum of inputs. The inputs were scaled to $[-1, 1]$ for all experiments, and the binary target values were scaled to $t_k \in \{0.1, 0.9\}$. The chosen scaling corresponds with the active domain and range of the sigmoid activation function.

4.3 Search space boundaries

NN weights are defined to be any numbers in $\mathbb{R}$, thus sensible boundaries have to be chosen for the sampling to take place. As discussed in Section 3.1, the search space should be sampled in the areas explored by the search algorithms, where acceptable solutions are likely to be found. Three search space boundary settings were considered in this study: $[-0.5, 0.5], [-1, 1], [-5, 5]$. These regions correspond to the typical weight initialisation area [23], as well as the areas where a search algorithm may find an acceptable solution [2]. The three different boundary settings yielded similar FLA results, therefore only $[-1, 1]$ results are reported. An investigation of the relationship between the NN search space boundaries and the corresponding FLA metrics can be found in [2].

4.4 Regularisation parameters

The success of weight elimination is heavily dependent on the regularisation parameters λ and w_0 (see Equations (2) and (3)), which are usually chosen empirically [46]. FLA offers an intuitive way to visualise the effects that the regularisation parameters have on the resulting error surface. To study these effects, different combinations of λ and w_0 must be considered. Previous studies have shown that w_0 of order unity is usually a good choice [45], and that small values of λ tend to give better results [39], because λ significantly larger than 1 causes the error to be dominated by the penalty function. This study considered all combinations of $\lambda \in \{1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-2}, 0.1, 0.5, 1\}$ and $w_0 \in \{1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-2}, 0.1, 0.5, 1, 2, 5\}$ for each problem.

4.5 Neural Network Training

Characterisation of error landscapes is not very useful unless insight into the nature of the problem is provided that can aid the training process. In addition to studying the error landscapes of regularised NNs, this study investigates the relationship between NN training algorithm performance and the FLA characteristics of the problem.

Backpropagation (BP) is one of the most popular NN training algorithms, first applied to NN training in 1974 by Werbos [47]. BP uses gradient descent to iteratively adjust NN weights and biases in the direction of the negative gradient of the objective function. Weight regularisation is applied to a NN by incorporating the desired penalty term in the gradient calculations.

4.6 Training algorithm parameters

To investigate the relationship between NN training algorithm performance and the FLA characteristics of the problem, the corresponding training algorithm parameters had to be optimised to ensure that the algorithm performed adequately. An iterative approach to algorithm parameter optimisation was used. Algorithm parameters were optimised one at a time. For each parameter, the algorithm was tested under a selected range of possible values for this parameter, while the other parameters remained fixed. In order to keep the optimisation process statistically sound, 30 independent runs were conducted for every value in the chosen discrete range. The parameter value yielding the lowest average generalisation error for the current parameter optimised was subsequently chosen, and optimisation proceeded to the next parameter. For optimisation of the remaining parameters, all the parameters already optimised were fixed to their best values.

For stochastic backpropagation, the learning rate, η, and momentum, α, had to be optimised. Values considered during the optimisation process are listed in Table 2. Final parameter values used in the experiments are listed in Table 3.

Table 2: BP Parameter Values Considered in the Optimisation Process

	Discrete value range
Learning rate η	$\{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5\}$
Momentum α	$\{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5\}$

Table 3: Optimised BP Parameter Values

	Iris	Glass	Diabetes
Learning rate η	0.1	0.4	0.3
Momentum α	0.9	0.9	0.8

It should be noted that the focus of the study was on investigating the relationship between the error landscape characteristics and regularisation parame-

ters, rather than algorithm performance. Thus, adequate performance rather than optimal performance was sufficient for the purpose of this study.

Each reported result is an average over 30 independent simulations that ran for 1000 iterations. Data sets were divided into a training set and a generalisation set; 80% of the patterns were randomly chosen to form the training set, and the remaining 20% were used for testing. Test data used to calculate the generalisation errors was never used for parameter optimisation.

5 Experimental Results

The purpose of the experiments was twofold: first, to investigate the influence of the regularisation term on the NN error landscapes under different regularisation parameter settings (in Section 5.1); secondly, to observe the training algorithm response to the error landscape changes induced by weight elimination (in Section 5.2).

5.1 Characterising regularised NN error landscapes

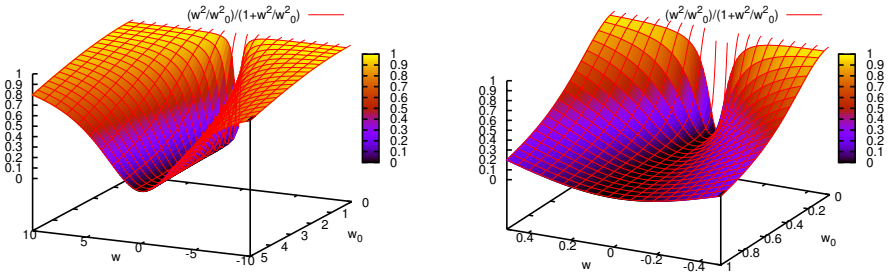
This section presents an analysis of the relationship between the regularisation term and the NN error landscapes. Section 5.1.1 investigates the effect of λ and w_0 on the average gradients observed in the landscapes. Section 5.1.2 looks at landscape ruggedness under different λ and w_0. Section 5.1.3 investigates the "searchability" of the NN error landscapes under various λ and w_0.

5.1.1 Gradients

To understand the impact of the penalty term, consider the weight elimination penalty for a single weight over various values of w_0, illustrated in Figure 1. As can be seen from Figure 1, the weight elimination term has a clear minimum in one dimension: a weight of 0 yields no penalty. The value of w_0 controls the "sharpness" of the minimum. It can be hypothesised that an increase in the value of λ increases the contribution of the penalty term to the objective function, thus "simplifying" the objective function by adding a global attractor in the form of a global minimum imposed by the penalty term.

Figure 2 shows the average values of G_{avg} and G_{dev} associated with different combination of λ and w_0. For interpretability, every scatter plot in Figure 2 includes a LOESS curve [6], representing local polynomial regression. Across all problems considered, an overall downward trend in G_{avg} is associated with an increase in λ. Indeed, the surface of the penalty function only has one minimum and is otherwise rather smooth. Therefore, it can be hypothesised that increasing the contribution of the penalty term to the objective function smoothes the error landscape. However, the effect of the penalty strongly depends on the chosen value of w_0: as shown in Figure 2, larger values of w_0 indeed yield smaller G_{avg} gradients. According to Figure 1, larger w_0 implies that the imposed minimum is less sharp, thus smaller gradients are to be expected.

In Figure 2, the problems are presented in ascending order of dimensionality. Figure 2 shows that the average magnitudes of the gradients increase with an



(a) $w \in [-10, 10]$ (b) $w \in [-0.5, 0.5]$

Fig. 1: Weight elimination function for a single weight, w, and weight elimination threshold, w_0

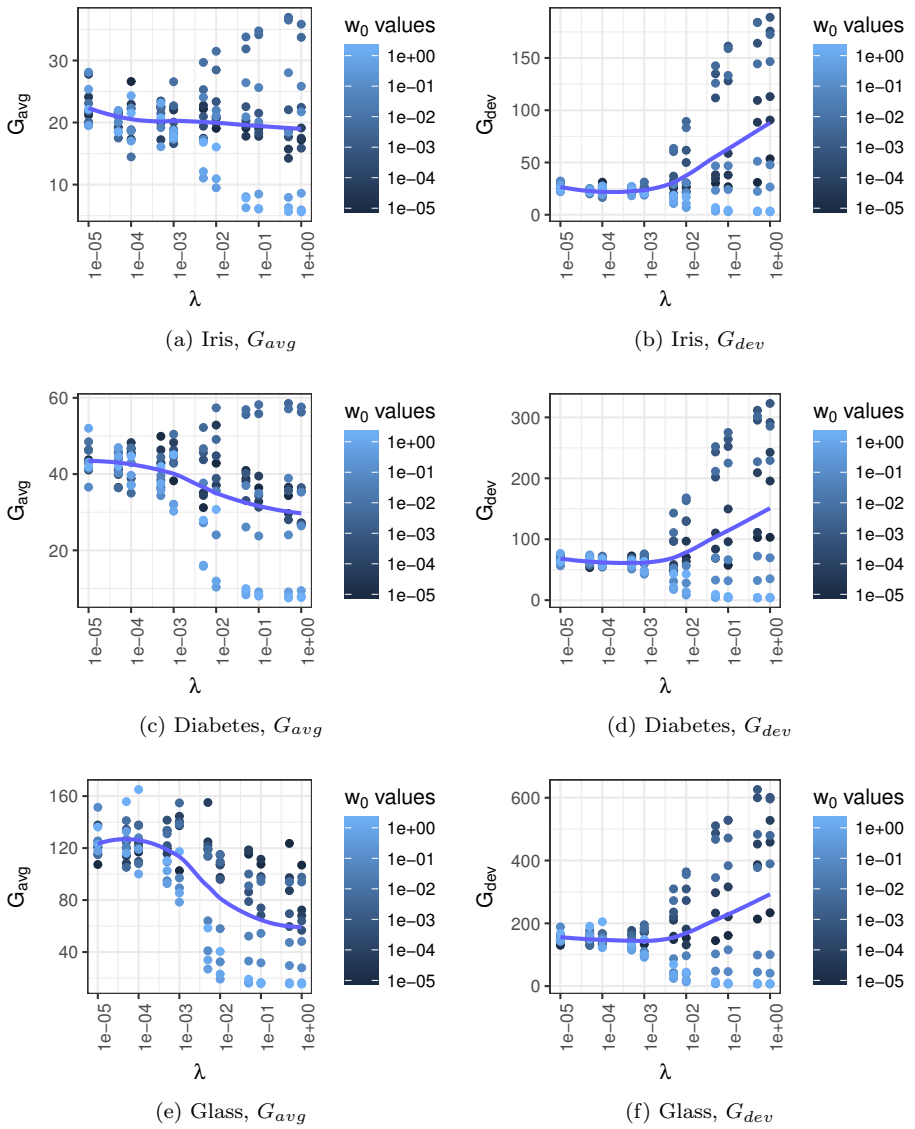
increase in problem dimensionality, which seems to be an inherent property of NN error landscapes. The downward trend in G_{avg} associated with the penalty term contribution becomes more definite with an increase in dimensionality. A high-dimensional fully-connected NN architecture is more likely to have redundant free parameters than a low-dimensional architecture, thus the effect of the penalty term becomes more pronounced in high dimensions.

Figure 2 also shows that an increase in λ is associated with an overall upward trend in G_{dev}. In other words, a stronger contribution of the penalty term to the objective function yields smaller gradients of higher variability. As Figure 1 illustrates, weight elimination introduces sharp minima, surrounded by a plateau-like surface. On the plateau, the gradients will be small. Around the minima, however, the fitness value will change rapidly. Thus, high G_{dev} likely resulted due to the contrast between the plateaus and the sudden minima. Indeed, a large difference between G_{avg} and G_{dev} is indicative of a step-like landscape with sudden jumps, according to [24]. The hypothesis is further confirmed by observing that the larger values of w_0, as illustrated in Figure 2, are not associated with an increase in G_{dev}: higher values of w_0 decrease the sharpness of the introduced optima.

Thus, introduction of the weight elimination term decreases the overall gradients of the error surface, but adds sharp, narrow optima that may not be very easy to find.

5.1.2 First Entropic Measure of Ruggedness

The first entropic measure of ruggedness, FEM, quantifies the change in fitness values based on entropy. Figure 3 illustrates how the micro- and macro-ruggedness of the regularised error landscape change in relation to different values of λ and w_0. Variation in ruggedness for different values of w_0 is only observed for larger values of λ. If λ is too small, the contribution of the penalty term may become negligible. As the value of w_0 increases, so does the ruggedness: small w_0 yields sharp narrow optima that alter a small part of the search space; increasing w_0 widens the "diameter" of the penalty-induced optima, thus influencing a larger



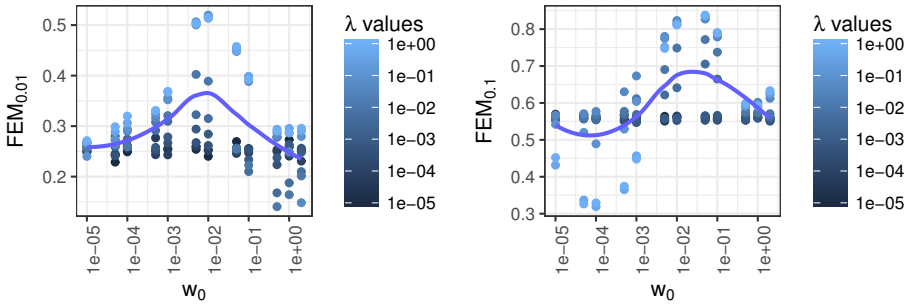
(a) Iris, G_{avg} (b) Iris, G_{dev}

(c) Diabetes, G_{avg} (d) Diabetes, G_{dev}

(e) Glass, G_{avg} (f) Glass, G_{dev}

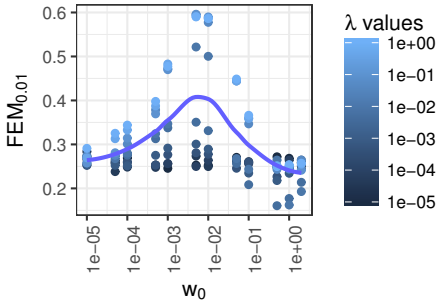
Fig. 2: G_{avg} and G_{dev} results obtained for different combinations of λ and w_0 on the $[-1, 1]$ interval

part of the error landscape. The ruggedness begins to drop again as w_0 becomes larger than 0.01: the induced optima gradually "flattens" and is lost among other error landscape fluctuations. When w_0 becomes larger than the chosen error landscape boundaries, no sampled weights are deemed large enough to be penalised, thus the contribution of the penalty term vanishes altogether.

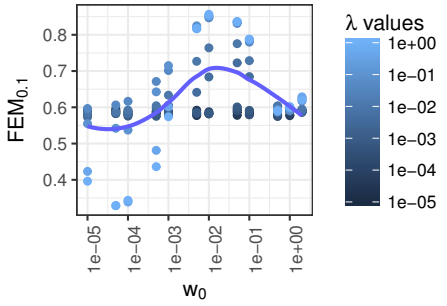


(a) Iris, $FEM_{0.01}$

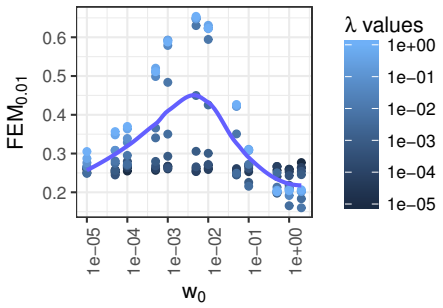
(b) Iris, $FEM_{0.1}$



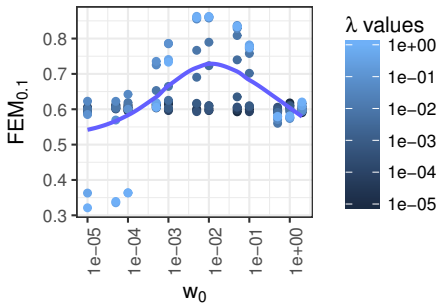
(c) Diabetes, $FEM_{0.01}$



(d) Diabetes, $FEM_{0.1}$



(e) Glass, $FEM_{0.01}$



(f) Glass, $FEM_{0.1}$

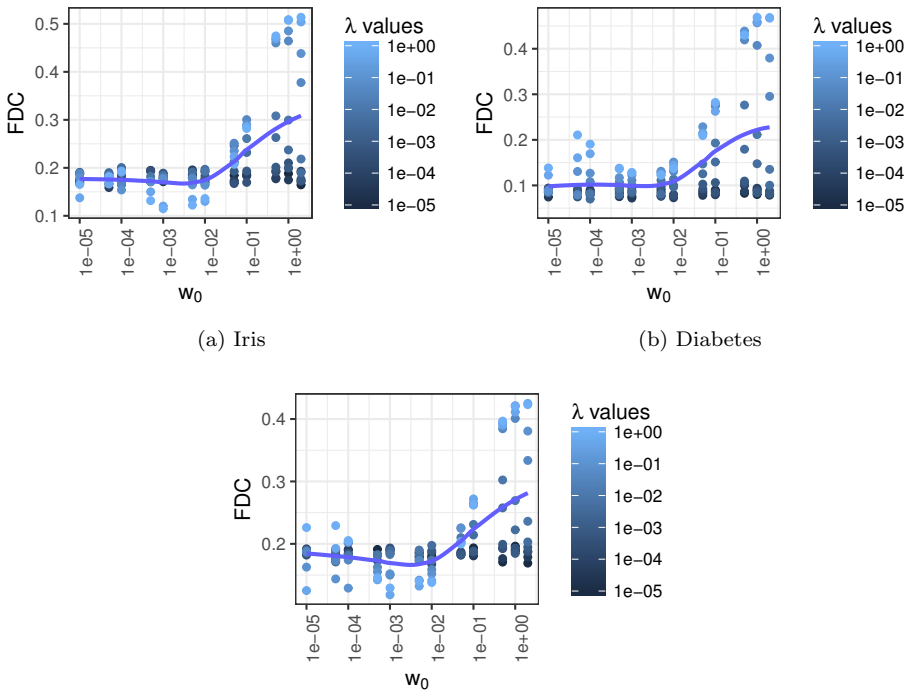
Fig. 3: $FEM_{0.01}$ and $FEM_{0.1}$ results obtained for different combinations of λ and w_0 on the $[-1, 1]$ interval

Macro-ruggedness results, also shown in Figure 3, illustrate the same trends as micro-ruggedness, but in a more pronounced manner. Average values of $FEM_{0.1}$ exceed the corresponding $FEM_{0.01}$ values, indicating that larger step sizes experience more variation in the landscape. Once again, values of w_0 close to 0.01 induce the most ruggedness across all problems considered.

Entropy is used to estimate the level of ruggedness in fitness landscapes. From the information theory perspective, the amount of entropy can be interpreted as the amount of "information", or variability. Clearly, specific values of w_0 and λ maximise the amount of variability present in the error landscape. The question that remains to be answered is whether this "information" is indeed useful to the training algorithms, and whether the penalty term makes the error landscape easier to search.

5.1.3 Fitness Distance Correlation

FDC_s results for different values of λ and w_0 are shown in Figure 4. Once again, the effect of the penalty term on the error landscape only becomes noticeable for larger values of λ. It becomes evident from Figure 4 that FDC_s tends to decrease as the value of w_0 increases. It was observed in Section 5.1.2 that increasing the value of w_0 results in increased ruggedness. Ruggedness implies that the fitness value fluctuates instead of persistently going up or down as the landscape is traversed by an algorithm. Increased fluctuations are thus labelled as "less searchable".



(a) Iris

(b) Diabetes

(c) Glass

Fig. 4: FDC_S results obtained for different combinations of λ and w_0 on the $[-1, 1]$ interval

After the highest peak of ruggedness is reached at $w_0 \approx 0.01$, and the ruggedness begins to decline with an increase in w_0, the landscape is progressively perceived as more and more searchable. Highest values of w_0, combined with the highest values of λ, yield the higest searchability on all problems considered. Thus, only large weights are penalised, and the applied penalty is strong. It was hypothesised in Section 5.1.1 that the application of penalty simplifies the landscape, and the FDC_s results support the hypothesis. High values of w_0 combined with high values of λ have also been shown to yield error landscapes with low and consistent gradients in Section 5.1.1, which once again confirms the landscape simplification via regularisation.

5.1.4 Summary

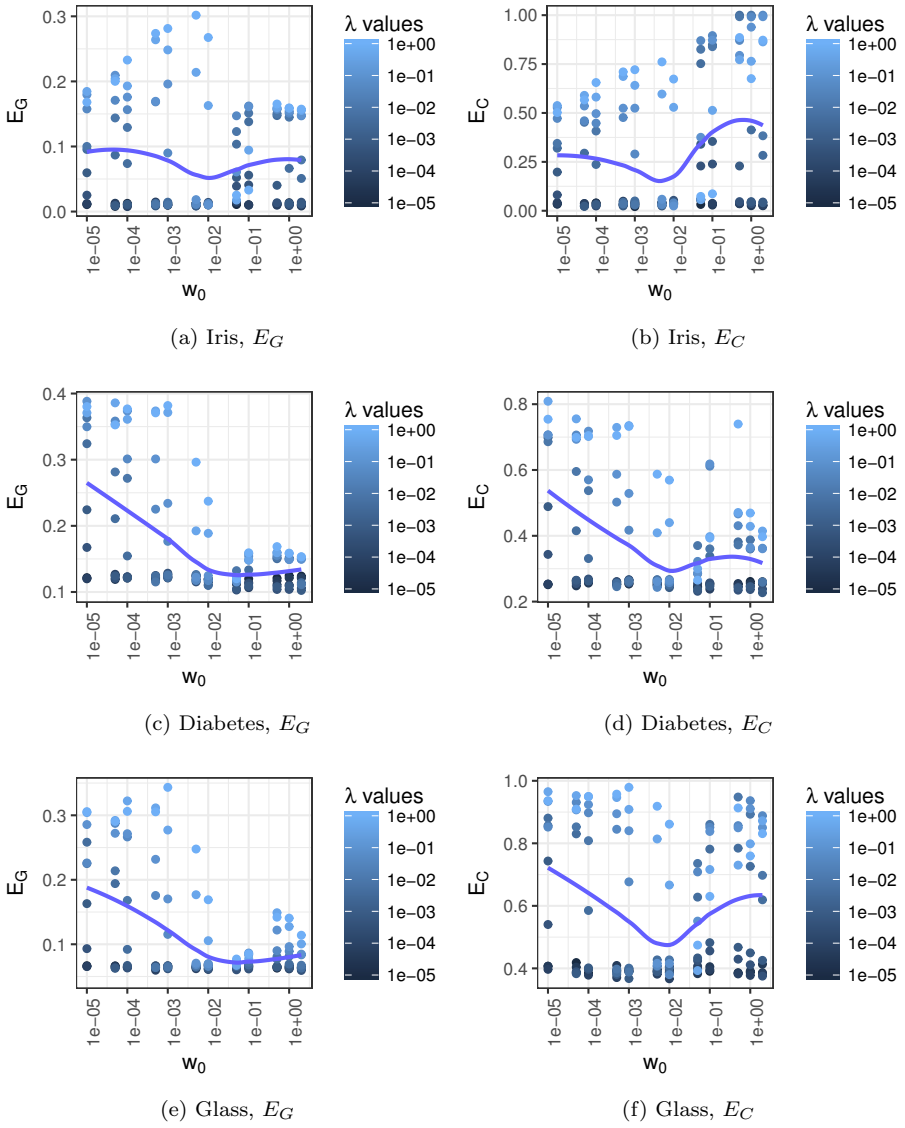
Regularised NN error landscapes were considered in terms of approximate gradients, ruggedness, and searchability. It was observed that the addition of a penalty term has a visible impact on the resulting error landscape, and that such properties of the error surface as gradients and ruggedness can be controlled by tuning the regularisation parameters. The next section puts these observations in the context of NN training.

5.2 Fitness landscape analysis and neural network training

Now that it has been established how the penalty term changes the NN error landscape, it is important to understand whether the induced changes make the landscape easier or harder to search for the NN training algorithms. This study considers the classical BP algorithm for NN training, as outlined in Section 4.5. No search space boundaries were enforced during training, since NN weights are defined to be any real numbers in $\mathbb{R}$. The goal of the study was to execute an instance of an algorithm on a problem, and to observe any difference in algorithm performance induced by the various combinations of λ and w_0 values. All NN weights were randomly initialised in the $[-0.5, 0.5]$ interval. Algorithm performance was evaluated in terms of the mean squared training error, E_T, the mean squared generalisation error, E_G, and the mean classification error, E_C. Both E_G and E_C were calculated on the test set, which constituted a randomly selected 20% of the data set not used during training or parameter optimisation. If at least one value in the output vector differed from the corresponding target value by more than 0.5, the pattern was labelled as incorrectly classified.

Figure 5 summarises the average E_G and E_C values obtained for different values of λ and w_0. Across all problems and both error metrics, high values of λ tend to result in inferior performance. An increase in λ implies that the contribution of the penalty term to the objective function becomes stronger. Indeed, if the training algorithm focuses on eliminating the weights rather than minimising the error, the training will produce a minimal architecture that is utterly useless.

The situation looks quite different when observed from the perspective of w_0. On all problems, the smallest values of w_0 yield poor training and generalisation performance. It has been observed in Section 5.1 that low values of w_0 correspond to error landscapes of low ruggedness and drastic gradient changes due to the nature of the penalty term. BP, being a gradient-descent method, struggles to find



(a) Iris, E_G (b) Iris, E_C

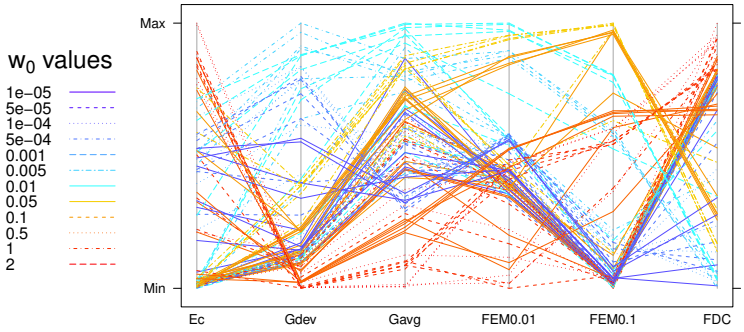
(c) Diabetes, E_G (d) Diabetes, E_C

(e) Glass, E_G (f) Glass, E_C

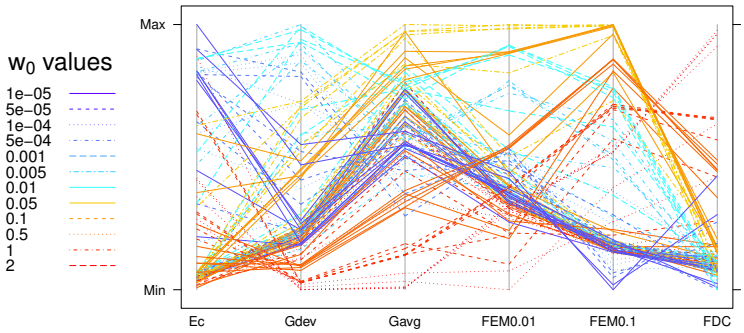
Fig. 5: Backpropagation results obtained for different combinations of λ and w_0

a way through the plateaus of the resulting staircase-like error surface. An increase in w_0, that corresponds to an increase in ruggedness, and a decrease in gradient variation, yields a predictable improvement in BP performance. As w_0 increases further, penalising fewer and fewer weights, the error begins to grow again, which is especially evident from the E_C values.

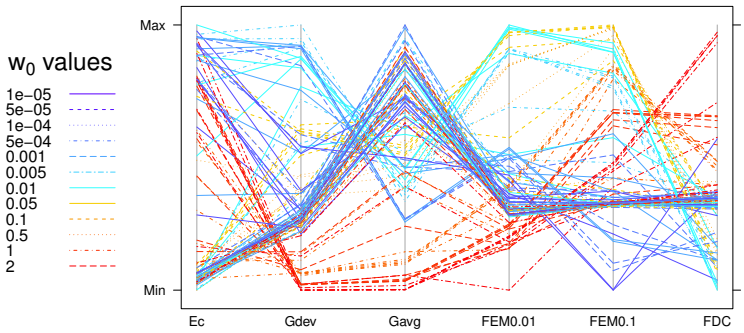
What parameter selection guidance can be induced from the above observations? First of all, w_0 in the range $[0.001, 0.01]$ generated the landscapes with most



(a) Iris



(b) Diabetes



(c) Glass

Fig. 6: Parallel coordinate plots for various FLA metrics obtained on the $[-1, 1]$ interval

variability, i.e. information, which resulted in the lowest average E_T, E_G, and E_C values. The authors suggest that values less or equal to 0.01 are considered for w_0 for low-dimensional problems, and values greater or equal to 0.01 are considered for higher-dimensional problems. Values around 0.01 are likely to produce a sensible result, thus 0.01 can be used as a starting point in the optimisation process.

Even though excessively large λ may hinder training by overshadowing the error by the penalty, a value of $\lambda < 0.001$ is not likely to influence the error landscape significantly. Therefore, λ in the range $[0.001, 0.1]$ is suggested to be considered in the parameter optimisation process.

It should also be noted that very low values of λ have yielded low error values in some scenarios, especially on lower-dimensional problems. Thus, regularised models should be compared to non-regularised models as a part of the optimisation process, to ensure that regularisation does indeed improve the generalisation performance.

To further visualise the relationship between the obtained FLA measures and the corresponding algorithm performance, parallel coordinate plots are presented in Figure 6. Parallel coordinate plots were first proposed by Wegman [44] as a way of visualising the relationships between various dimensions in high-dimensional spaces. In Figure 6, each FLA metric is represented as a parallel coordinate axis, and E_C is used as a metric representing BP performance. Each line represents a combination of averages over 30 simulations of each metric, for a given combination of λ and w_0 values.

Even though BP performance differs per problem (Figures 6a, 6b, and 6c), some general trends can be observed. For all problems considered, G_{avg} higher than G_{dev} is associated with lower E_C. Consistent but prominent gradients imply a more searchable and cohesive landscape, with enough gradient information to guide BP. Small G_{avg} and G_{dev}, indicative of a fairly flat landscape, yielded poor BP performance, which is to be expected. Mid-range G_{avg} with $G_{dev} >> G_{avg}$ also yielded poor performance, indicating that BP does not perform well on step-like error landscapes with abrupt fitness changes.

Figure 6 shows that high ruggedness is handled well by BP. Good BP performance on highly rugged surfaces indicates that BP may be much more resilient to local minima than previously suspected. These results correlate well with the recent theoretical findings showing that the NN error landscapes contain more saddle points than local minima [17], and that the number of local minima reduces exponentially as the dimensionality of the problem increases [4, 7].

Low micro-ruggedness, $FEM_{0.01}$, combined with higher macro-ruggedness, $FEM_{0.1}$, resulted in poor BP performance. Low $FEM_{0.01}$ and high $FEM_{0.1}$ also correspond to low G_{avg}. All of these properties combined describe landscapes with wide plateaus, with sudden changes observable only on the macro-level. Such landscapes are not very searchable from the gradient descent perspective.

Interestingly, the searchability measure FDC_s provided the least useful and the most misleading information: the highest FDC_s values corresponded to the flattest landscapes with low G_{avg} and G_{dev}. BP struggled to perform well on such landscapes for the lack of gradient information. Low values of FDC_s, on the other hand, corresponded to better BP performance. Perhaps NN error surfaces are too untrivial to be considered from a "global shape" perspective that FDC_s offers.

6 Conclusion

This study investigated the applicability of FLA metrics to regularised NN error landscapes. The influence of the weight elimination term on the NN error landscape characteristics was studied. It was observed that the addition of a weight elimination term to the objective function alters the error landscape, and does not necessarily make the error landscape easier to search. Five continuous FLA metrics were used to study the properties of the regularised error surfaces: gradient measures G_{avg} and G_{dev}, ruggedness measures $FEM_{0.01}$ and $FEM_{0.1}$, and the searchability measure FDC_s. Different combinations of regularisation parameters λ and w_0 were used, and the BP training algorithm was considered in the FLA context. FLA was shown to be a useful tool for visualising the properties of high-dimensional NN error landscapes.

The weight elimination term was shown to smooth the error landscape while introducing additional minima. Tuning of the w_0 parameter allows tuning of the sharpness of the introduced minima. Sharper minima result in more drastic, highly varied gradients. Values chosen from the $[0.001, 0.1]$ range for the w_0 parameter maximised the variability, or ruggedness of the landscape, and yielded the lowest average NN errors. It was shown that the BP algorithm is capable of efficiently searching very rugged landscapes. On the other hand, step-like landscapes with rare and sudden fitness changes render BP inefficient.

Very small λ values render regularisation insignificant, while excessively large values of λ overshadow the error by the penalty. Values chosen from the $[0.001, 0.1]$ range resulted in visible error landscape transformations and did not hinder training, provided that the w_0 value was sensible. The necessity to optimise λ can be eliminated by employing a multi-objective algorithm to optimise both the objective function and the weight elimination term separately, and find the suitable trade-off solution thereof. This is a topic of future research.

The searchability metric, FDC_s, evaluated rugged landscapes as less searchable, even though BP actually benefited from the variability in the landscape. Perhaps NN error surfaces are too complex for the crude "global shape" estimation that FDC_s provides. Out of the five metrics considered, FDC_s produced the least valuable results.

This study only considered weight elimination. It will be interesting to compare weight elimination error surfaces to other regularised error surfaces. FLA can potentially be used to optimise the penalty parameters involved, as FLA metrics provide a handy visualisation tool for the corresponding error landscapes. The behaviour of FLA metrics on larger data sets and larger NN architectures needs to be investigated. Future research will include a broader investigation of NN error surfaces from the FLA perspective.

Acknowledgements The authors would like to thank the Centre for High Performance Computing (CHPC) (http://www.chpc.ac.za) for the use of their cluster to obtain the data for this study.

This work is based on the research supported by the National Research Foundation (NRF) of South Africa (Grant Number 46712). The opinions, findings and conclusions or recommendations expressed in this article is that of the author(s) alone, and not that of the NRF. The NRF accepts no liability whatsoever in this regard.

References

1. Bishop CM (1995) Neural Networks for Pattern Recognition. Oxford University Press, Oxford, UK
2. Bosman AS, Engelbrecht AP, Helbig M (2016) Search space boundaries in neural network error landscape analysis. In: Proceedings of the IEEE Symposium Series on Computational Intelligence, IEEE, Athens, Greece, pp 1–8
3. Carvalho M, Ludermir TB (2006) Particle swarm optimization of feed-forward neural networks with weight decay. In: Sixth International Conference on Hybrid Intelligent Systems (HIS'06), IEEE, pp 5–8
4. Choromanska A, Henaff M, Mathieu M, Ben Arous G, LeCun Y (2015) The loss surfaces of multilayer networks. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, pp 192–204
5. Choromanska A, LeCun Y, Arous GB (2015) Open problem: The landscape of the loss surfaces of multilayer networks. In: Proceedings of The 28th Conference on Learning Theory, pp 1756–1760
6. Cleveland WS, Devlin SJ (1988) Locally weighted regression: an approach to regression analysis by local fitting. Journal of the American statistical association 83(403):596–610
7. Dauphin YN, Pascanu R, Gulcehre C, Cho K, Ganguli S, Bengio Y (2014) Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In: Advances in Neural Information Processing Systems, pp 2933–2941
8. Dreyfus G (2005) Neural networks: methodology and applications. Springer, Berlin, Germany
9. Fisher RA (1936) The use of multiple measurements in taxonomic problems. Annals of eugenics 7(2):179–188
10. Gallagher M (2001) Fitness distance correlation of neural network error surfaces: A scalable, continuous optimization problem. In: Proceedings of the 12th European Conference on Machine Learning, Springer-Verlag, pp 157–166
11. Gallagher MR (2000) Multi-layer perceptron error surfaces: Visualization, structure and modelling. PhD thesis, University of Queensland, St Lucia 4072, Australia
12. Girosi F, Jones M, Poggio T (1995) Regularization theory and neural networks architectures. Neural computation 7(2):219–269
13. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: International Conference on Artificial Intelligence and Statistics, pp 249–256
14. Gonçalves I, Silva S, Fonseca CM (2015) Semantic learning machine: A feed-forward neural network construction algorithm inspired by geometric semantic genetic programming. In: Progress in Artificial Intelligence, Lecture Notes in Computer Science, vol 9273, Springer, pp 280–285
15. Grinstein G, Trutschl M, Cvek U (2001) High-dimensional visualizations. In: Proceedings of the Visual Data Mining Workshop, KDD, Citeseer
16. Gupta A, Lam SM (1998) Weight decay backpropagation for noisy data. Neural Networks 11(6):1127–1138
17. Hamey LG (1998) XOR has no local minima: A case study in neural network error surface analysis. Neural Networks 11(4):669–681

18. Hinton GE (1987) Learning translation invariant recognition in a massively parallel networks. In: PARLE Parallel Architectures and Languages Europe, Springer, pp 1–13
19. Hush DR, Horne B, Salas JM (1992) Error surfaces for multilayer perceptrons. IEEE Transactions on Systems, Man and Cybernetics 22(5):1152–1161
20. Jones T (1995) Evolutionary algorithms, fitness landscapes and search. PhD thesis, The University of New Mexico
21. Jones T, Forrest S (1995) Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., pp 184–192
22. Kordos M, Duch W (2004) A survey of factors influencing MLP error surface. Control and Cybernetics 33(4)
23. LeCun YA, Bottou L, Orr GB, Müller KR (2012) Efficient backprop. In: Neural networks: Tricks of the trade, Springer, pp 9–48
24. Malan KM (2014) Characterising continuous optimisation problems for particle swarm optimisation performance prediction. PhD thesis, University of Pretoria
25. Malan KM, Engelbrecht AP (2009) Quantifying ruggedness of continuous landscapes using entropy. In: IEEE Congress on Evolutionary Computation, IEEE, pp 1440–1447
26. Malan KM, Engelbrecht AP (2013) Ruggedness, funnels and gradients in fitness landscapes and the effect on PSO performance. In: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, pp 963–970
27. Malan KM, Engelbrecht AP (2013) A survey of techniques for characterising fitness landscapes and some possible ways forward. Information Sciences 241:148–163
28. Malan KM, Engelbrecht AP (2014) Characterising the searchability of continuous optimisation problems for PSO. Swarm Intelligence 8(4):275–302
29. Malan KM, Engelbrecht AP (2014) A progressive random walk algorithm for sampling continuous fitness landscapes. In: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, pp 2507–2514
30. Mc Loone S, Irwin G (2001) Improving neural network training solutions using regularisation. Neurocomputing 37(14):71–90
31. Mersmann O, Bischl B, Trautmann H, Preuss M, Weihs C, Rudolph G (2011) Exploratory landscape analysis. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation, ACM, pp 829–836
32. Merz P, Freisleben B (2000) Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. IEEE Transactions on Evolutionary Computation 4(4):337–352
33. Moody J, Hanson SJ, Lippmann RP (1992) The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. Advances in neural information processing systems 4:847–854
34. Moody J, Hanson S, Krogh A, Hertz JA (1995) A simple weight decay can improve generalization. Advances in neural information processing systems 4:950–957
35. Muñoz MA, Sun Y, Kirley M, Halgamuge SK (2015) Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. Information Sciences 317:224–245

36. Orr GB, Müller KR (2003) Neural networks: tricks of the trade. Springer
37. Pitzer E, Affenzeller M (2012) A comprehensive survey on fitness landscape analysis. In: Recent Advances in Intelligent Engineering Systems, Springer, pp 161–191
38. Prechelt L (1994) Proben1 – a set of neural network benchmark problems and benchmarking rules. Tech. rep., Universität Karlsruhe, Karlsruhe, Germany
39. Rakitianskaia A, Engelbrecht A (2014) Weight regularisation in particle swarm optimisation neural network training. In: Proceedings of the IEEE Symposium on Swarm Intelligence, IEEE, Florida, USA, pp 1–8
40. Rakitianskaia A, Engelbrecht A (2015) Saturation in PSO neural network training: Good or evil? In: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, Sendai, Japan, pp 125–132
41. Rakitianskaia A, Bekker E, Malan K, Engelbrecht A (2016) Analysis of error landscapes in multi-layered neural networks for classification. In: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, Vancouver, Canada, in press
42. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1):1929–1958
43. Wang J, Ye Z, Gao W, Zurada JM (2016) Boundedness and convergence analysis of weight elimination for cyclic training of neural networks. Neural Networks 82:49 – 61
44. Wegman EJ (1990) Hyperdimensional data analysis using parallel coordinates. Journal of the American Statistical Association 85(411):664–675
45. Weigend AS, Rumelhart DE, Huberman BA (1991) Generalization by weight-elimination applied to currency exchange rate prediction. In: Proceedings of the International Joint Conference on Neural Networks, IEEE, Seattle, vol 1, pp 837–841
46. Weigend AS, Rumelhart DE, Huberman BA (1991) Generalization by weight elimination with application to forecasting. Advances in Neural information processings systems 3
47. Werbos PJ (1974) Beyond regression: New tools for prediction and analysis in the behavioural sciences. PhD thesis, Harvard University, Boston, USA