Analysis of Error Landscapes in Multi-layered Neural Networks for Classification

Anna Rakitianskaia*, Eduan Bekker*, Katherine M. Malan* and Andries Engelbrecht*
*Department of Computer Science
University of Pretoria, Pretoria, South Africa
Email: see http://cs.up.ac.za/staff

***Abstract*—Artificial neural networks are inherently high-dimensional, which limits our ability to visualise and understand their inner workings. Neural network architecture and training algorithm parameters are usually optimised on an ad hoc basis, with very limited insight into the nature of the objective function landscape. This study proposes using fitness landscape analysis to quantify topological properties of neural network error landscapes. Five techniques from the fitness landscape analysis field are adapted to work with neural network error landscapes. These techniques are then used to analyse how the error landscape changes under different error measurements and different number of hidden layers. The results show that fitness landscape analysis provides valuable insight into neural network error landscapes, and could be used for architecture selection.**

I. Introduction

Artificial neural networks (ANNs) have been used since the 1970's for tasks such as classification [1], time series prediction [2], computer vision [3], and speech recognition [4]. An ANN consists of interconnected neurons, where every neuron is a simple non-linear function, and every neuron-to-neuron connection bears a weight [5]. An ANN is capable of discovering patterns in data by means of training. The training is achieved by feeding data through the ANN, and adjusting the connection weights such that the error is minimised [6].

ANNs are often referred to as "black boxes", because they provide little insight into the relative influence of the independent variables on the prediction. Not only is the prediction model therefore not well understood [7], but the practice of constructing the best model is also a challenge. For example, although it is known that the number of hidden layers and neurons per layer does have an impact on the ANN performance [8], [9], due to our lack of understanding of these "black boxes" the problem of choosing a suitable configuration is harder than it seems. Up to now the most common method of architecture selection is trial and error.

Some insight into the problem can be gained through fitness landscape analysis (FLA). FLA is a young and evolving field of computational intelligence, first applied in evolutionary computation for algorithm performance prediction [10], [11]. The aim of FLA is to estimate and quantify topological properties of the given objective function surface in order to better understand the problem. Objective functions are studied through their fitness landscapes: i.e. the search space of all possible solutions with associated fitness values. In the context of ANNs, the search space is made up of all possible weight combinations, and the ANN error measures corresponding to these weight combinations make up the error landscape.

The ANN error landscape depends on the error measurement chosen. This study compares the error landscapes generated by the mean squared error and the classification error. It is discovered that classification error generates a significantly less informative error landscape than the mean squared error.

Recent successes of deep learning showed that multi-layered ANN architectures have greater information capacity than single hidden layer ANNs [12], [13]. Current attempts at training deep ANNs would benefit from better understanding how the addition of extra hidden layers change the search space. In this paper, the effect of progressively adding non-linear hidden layers to the ANN is analysed on a selection of problems.

The rest of the paper is structured as follows: Section II discusses FLA, and explains the FLA measurements used in this study. Section III highlights the fitness landscape characteristics specific to ANNs. Section IV describes the experiments conducted. Section V discusses the empirical results obtained. Finally, Section VI summarises the observations made, and concludes the paper.

II. Fitness Landscape Analysis

A fitness landscape is a representation of the search space with regards to the objective function fitness values [10], [14]. The term was coined in the evolutionary optimisation community, but is applicable to any optimisation problem with a well-defined objective function: the objective function values as calculated across the search space form a hypersurface that the optimisation algorithm either minimises or maximises. The aim of fitness landscape analysis (FLA) is to estimate landscape features of the objective function landscape in order to better understand why a certain algorithm performs well or fails. Estimation of fitness landscape characteristics aids problem understanding and optimisation algorithm selection.

Fitness landscape characteristics are estimated by taking random samples of the search space, calculating the objective function value for every point in each sample, and analysing the relationship between the spatial and the qualitative characteristics of the sample points. There are many techniques to analyse fitness landscapes proposed in literature, but most of these only apply to discrete landscapes [14]. In this study, five metrics for continuous fitness landscape analysis were

used: the fitness distance correlation measure adapted to work without knowledge of the global optima, micro and macro ruggedness, and two gradient measures, G_{avg} and G_{dev}. These metrics are described below.

A. Fitness distance correlation

The fitness-distance correlation (FDC) metric was proposed by Jones and Forest [15] as a measure of global problem hardness. FDC gives an indication of the global shape of the landscape to be searched. FDC measures the covariance between the fitness of a solution and its distance to the nearest optimum.

Malan and Engelbrecht [16] proposed a new measure, FDC_s, that is based on a sample of solutions without known optima. Applied to ANN error landscapes, 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 it 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.

The range of the FDC_s measurement is bounded to $[-1, 1]$. For minimisation problems, a value close to 1 indicates a highly searchable landscape, a value close to 0 shows a lack of information in the landscape, and a negative value indicates a deceptive search landscape.

FDC has been applied to ANNs before. Gallagher [17] used the FDC measurement to examine the error surfaces of ANNs. The aim was to measure the difficulty of training ANNs. The weights obtained from each training epoch were used as sampling points, and the optima found by training was used as the global optima. Gallagher concluded that ANNs can indeed be used as scalable optimisation benchmark problems. This study applies FDC_s to characterise ANN error landscapes outside of the training algorithm context.

B. Ruggedness

Malan and Engelbrecht proposed two ruggedness measures based on Vassilev's [18] first entropic measure (FEM). These measures are based on a random walk through the search space, and quantify the change in fitness values based on entropy [19]. This study uses Malan and Engelbrecht's random walk for continuous spaces [20]. The two measures used are:

- $FEM_{0.01}$ – Micro ruggedness, which is based on a random walk with a maximum step size of 1% of the bounded search space.
- $FEM_{0.1}$ – Macro ruggedness, which is based on a random walk with a maximum step size of 10% of the bounded search space.

The value of FEM is continuous and ranges between 0 and 1, where 0 indicates a flat landscape, and 1 indicates maximal ruggedness.

C. Gradients

Although ruggedness quantifies the presence of fitness changes in the landscape, it does not quantify the magnitude of the jumps. Malan and Engelbrecht [21] proposed two gradient measures: average estimated gradient G_{avg} and the standard deviation of gradient G_{dev}, calculated based on Manhattan random walk [14] through the search space. G_{avg} and G_{dev} are both indicative of the variance magnitude within the fitness landscape. Applied to ANN error landscapes, G_{avg} can be defined as:

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

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

$$g(t) = \frac{\Delta e_t}{s}$$

where Δe_t is the difference between the error values of the weight vectors defining step t of the random walk. Similarly, G_{dev} is defined as:

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

The G_{avg} measurement is essentially the mean magnitude of change in fitness values, while G_{dev} is the corresponding standard deviation. Lower values for G_{avg} and G_{dev} typically indicate a simpler landscape that is easier to search. However, a lack of gradients can also mean high neutrality (plateaus) in the landscape [22], which may prove challenging to optimisation algorithms.

III. Error Landscapes of Artificial Neural Networks

Feed forward ANNs typically have an input layer of neurons, one or more hidden layers of neurons, and a final layer of output neurons [23]. In fully-connected ANN architectures, each neuron in a layer is connected to every neuron in the next layer, and each connection bears a weight. Given m weights, the solution space of all possible classifiers for an ANN is an m-dimensional space of all possible weight combinations. Some of these weight combinations may have a poor measure of error, while some may be good. The complete search space of all possible ANN weights with associated error values is referred to as the "error landscape" in this study.

Error landscapes of ANNs are in many ways similar to fitness landscapes of continuous optimisation problems. There are, however, a few characteristics specific to ANNs that should be considered:

- ANNs are often used for the purpose of classification, and a classification error indicative of the percentage of patterns correctly classified is often used to measure ANN performance. In practice, classification error is a discrete value, since the total number of patterns in a data set is discrete. Considering classification error surface, the discrete nature of this error measurement has the potential of introducing perfectly flat areas into the landscape.

- Due to the inherent high dimensionality of ANNs, error landscapes are hard to visualise and analyse for practical problems. Even the smallest non-linear ANN modelling the XOR gate requires two inputs, two hidden units, and one output, instantly leaving the interpretable three-dimensional space behind.
- ANN error landscapes are unbounded, whereas optimisation problems usually have bounded decision variables. ANN weights do not have any meaning by themselves, and can be any numbers in $\mathbb{R}^m$, as opposed to decision variables in optimisation problems that relate to some limited resource in the real world.

Exploring error landscapes in an attempt to understand the inner workings of ANNs is not an entirely new concept: Gallagher [22] used techniques such as principal component analysis to simplify error landscape representation in order to visualise ANN error landscapes. It was determined that error landscapes have a lot of flat areas with sudden cliffs or ravines. It was also theoretically proved using random matrix theory that ANN 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 [24], [25]. This study attempts to add to this body of knowledge by applying Malan and Engelbrecht's FLA measures [26] described in Section II to ANN error surfaces. The details of the experiments conducted are provided in the section below.

IV. Experimentation

The aim of the experiments was to apply Malan and Engelbrecht's FLA metrics outlined in Section II to ANN error landscapes, and to study the effect of different error measurements and the number of non-linear hidden layers on the ANN error landscapes. The rest of the section is structured as follows: Section IV-A outlines the benchmarks used, Section IV-B describes the artificial boundaries placed on the search spaces, and Section IV-C lists the error measurements used to generate the error landscapes.

A. Benchmark problems

Three classification benchmark problems outlined in Table I were used in this study. The aim of the study was not to compare error landscapes of the different problems, but rather to compare different ANN architectures for a given problem. The exploratory nature of this study suggested using simple, well-known benchmarks to aid the interpretation of results.

TABLE I
Benchmark Problems

Problem	# Inputs	# Outputs	Source
XOR	2	1	Eberhart et al. [27]
Iris	4	3	Gupta and Lam [28]
Diabetes	8	1	Carvalho and Ludermir [29]

To study the effect of different number of hidden layers on the ANN error landscapes, a number of ANN architecture set-ups, listed in Table II, were chosen for each problem. Since each problem has a different number of inputs and outputs, each problem had its own set of ANN architectures used for benchmarking. Note that each input and hidden layer had a bias unit, set to -1. In Table II, ANN1 are optimised ANN architectures, according to the sources listed in Table I. ANN2 doubles the hidden layer size of ANN1, ANN3 adds a single extra hidden layer, and ANN4 adds two extra hidden layers. The total dimensionality of each ANN set-up is shown in Table II in parenthesis. All ANNs employed the sigmoid activation function in the hidden and output layers, defined as $f(x) = 1/(1 + e^{-x})$. The output of the sigmoid function is in the range $(0, 1)$.

TABLE II
Architectures per problem (number of weights in parenthesis)

Problem	ANN1	ANN2	ANN3	ANN4
XOR	2-2-1 (9)	2-4-1 (17)	2-2-2-1 (15)	2-2-2-2-1 (21)
Iris	4-4-3 (35)	4-8-3 (67)	4-2-2-3 (25)	4-2-2-2-3 (31)
Diabetes	8-6-1 (61)	8-12-1 (121)	8-4-4-1 (61)	8-4-4-4-1 (81)

B. Search space bounds

As outlined in Section III, ANN weights are defined to be any real numbers in $\mathbb{R}^m$. This poses a problem, since all sampling algorithms require a knowledge of the minimum and maximum values for each dimension, which are not defined in case of ANNs. It is known, however, that ANN weights are usually initialised in a small range around 0 to avoid saturation [30]. Therefore, it is not unreasonable to study the ANN error landscapes on a small area around the origin, since that is exactly where the search for a solution begins. This study considered ANN error landscapes on $[-1, 1]$ interval. Further studies will analyse the effect of different search space bounds on the way the error landscapes are perceived through the FLA measurements.

C. Error measurements

The nature of ANN error landscapes depends on the error measurement chosen. This study considered two widely-used error measurements: classification error and mean squared error.

1) Classification error: Classification error, E_{ce}, is the simplest method of evaluating the performance of a classifier. To calculate E_{ce}, all output values from the activation function, $f(x)$, in output neuron o_k for patten z_p are discretised to 0 or 1 as follows:

$$o_k = \begin{cases} 0, & \text{if } f(x) < 0.5 \\ 1, & \text{otherwise} \end{cases}$$

Pattern z_p is only correctly classified if $\forall k, o_{kp} = t_{kp}$, where t_{kp} is the target output for output neuron k for patten z_p. The E_{ce} is then given by:

$$E_{ce} = \frac{P_i}{P}$$

where P is the total number of patterns, and P_i is the number of incorrectly classified patterns. E_{ce} is in the range $[0, 1]$, where 0 means that all patterns are correctly classified, and 1 indicates that no patterns were correctly classified. In practice, E_{ce} is a floating point value with a discrete number of outcomes. The total number of possible E_{ce} outcomes is equal to the number of patterns in the data set. For classification problems, the aim is to minimise E_{ce}.

2) Mean squared error: Mean squared error (MSE) is a more fine-grained, continuous error measurement, given by:

$$E_{mse} = \frac{\sum_{p=1}^{P} \sum_{k=1}^{K} (t_{kp} - o_{kp})^2}{PK}$$

where K is the total number of outputs per pattern, and P is the total number of patterns. MSE quantifies the magnitude of the distance between the generated outputs, o_{kp}, and target outputs, t_{kp}. The aim of training algorithms is to minimise MSE, i.e. minimise the distance between the outputs and the targets.

V. Experimental Results

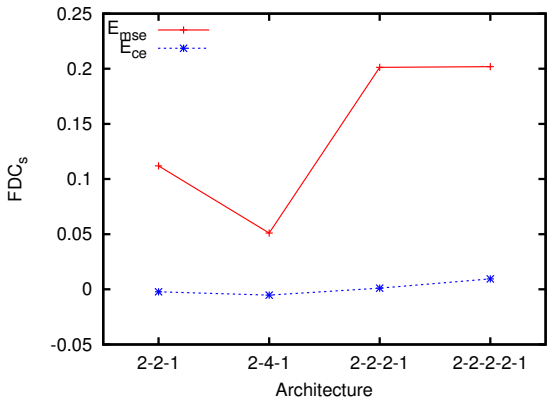
Experimental results obtained on various ANN architectures for the three problems considered are presented in this section. The five FLA metrics mentioned in Section II were used to analyse the ANN error landscapes with respect to E_{ce} and E_{mse} (Section V-A), for different number of hidden layers (Section V-B). All reported results are averages over 30 independent runs. All results together with corresponding standard deviations are summarised in Table III.

A. E_{ce} versus E_{mse}

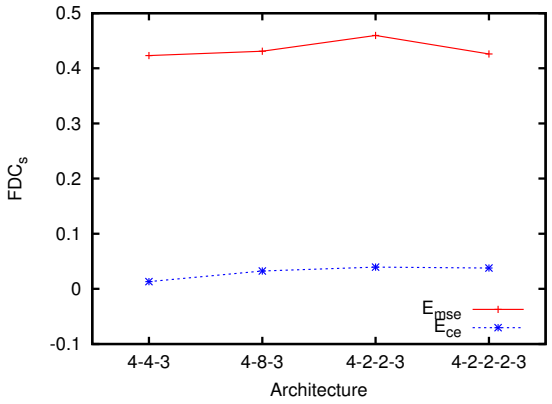
The term "error landscape" implies that the nature of such a landscape would depend on the error function chosen. Quantifying the properties of error landscapes corresponding to different error functions can indicate which error functions are more informative, and should thus be preferred for ANN training.

For this purpose, E_{ce} and E_{mse} error landscapes were first considered in terms of FDC$_s$, a measure of searchability. Figures 1a, 1b, and 1c show FDC$_s$ values obtained for the XOR, iris, and diabetes classification problems, respectively. It is clear from these figures that E_{ce} and E_{mse} produced very different FDC$_s$ values indeed. The FDC$_s$ values for E_{mse} were always above zero, thus showing a positive correlation between the error values and the distance to the optima. The iris problem in particular showed fairly high FDC$_s$ values (around 0.45), which indicates that the error landscape produced by E_{mse} contained enough information to guide a search towards the optima. FDC$_s$ values produced for the XOR problem using E_{mse} were below 0.2, indicating a weaker correlation, but still a positive one.

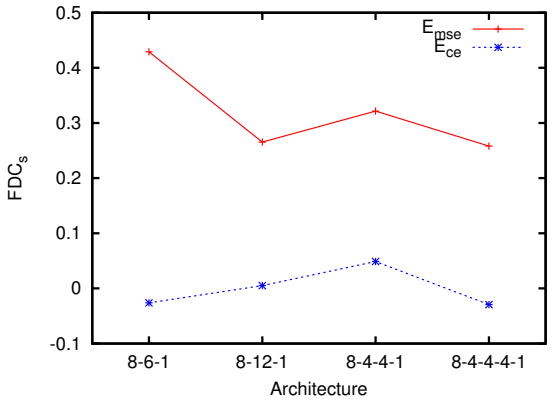
FDC$_s$ values for E_{ce}, on the other hand, were always lower than that for E_{mse}. According to the definition of FDC$_s$ as a searchability measure, this observation can only mean that the E_{ce} landscape contained less information to guide the search. E_{ce} values hovered around 0 for all problems



(a) XOR



(b) Iris



(c) Diabetes

Fig. 1. FDC_s values obtained for $[-1, 1]$ interval

considered, indicating a lack of information in the landscape. Therefore, using E_{ce} as a guide in a search for an optimum can be extremely inefficient. A search that uses E_{ce} for gradient information would be as effective as a random search. For some ANN architectures for XOR and diabetes, FDC$_s$ values for E_{ce} were negative. Thus, the corresponding error landscape was not only uninformatie, but even deceptive, and could have potentially guided the search in a wrong direction.

E_{ce} and E_{mse} landscapes can also be compared in terms of

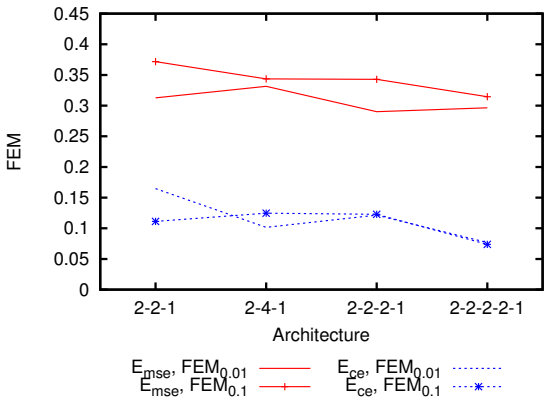
ruggedness, quantified by $FEM_{0.01}$ and $FEM_{0.1}$. Figures 2a, 2b, and 2c show $FEM_{0.01}$ and $FEM_{0.1}$ values obtained for XOR, iris, and diabetes classification problems, respectively. On all problems considered, using E_{mse} as the error measure produced much higher $FEM_{0.01}$ and $FEM_{0.1}$ values than E_{ce}. Higher values for $FEM_{0.01}$ and $FEM_{0.1}$ indicate more change in the landscape, therefore a more rugged overall landscape. Lower ruggedness observed for E_{ce} supports the hypothesis that E_{ce} generates a flatter error landscape due to only having a discrete number of possible outcomes, as discussed in Section III.

Table III shows that on all problems, FEM of E_{mse} error landscapes always had lower standard deviation than FEM of E_{ce}. This indicates that E_{mse} error landscapes were more consistent and predictable, thus potentially easier to search.

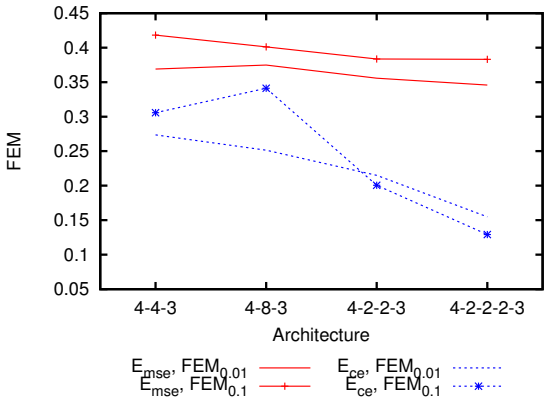
For all problems and ANN configurations, both error measurements resulted in a ruggedness value of less than 0.5, which is more flat than rugged. Even though E_{mse} always exhibited higher ruggedness than E_{ce}, the obtained FEM values were still fairly low compared to other optimisation problems as reported in the literature [26]. For example, the step function [26] has a $FEM_{0.1}$ of 0.66, and $FEM_{0.01}$ of 0.81 in 5 dimensions only. Thus, the ruggedness measures obtained on ANNs support the previously made observations that ANN error landscapes are prone to neutralities, or plateaus in the search space [17].

The last two measures to consider are G_{avg} and G_{dev}, which quantify the magnitude of variance in the error landscape. Figures 3a, 3b, and 3c illustrate that on all problems considered, the difference between G_{avg} and G_{dev} for E_{ce} was greater than for E_{mse}. According to [26], a large difference between G_{avg} and G_{dev} is indicative of a step-like landscape with sudden jumps. Both error measures seem susceptible to this phenomena, but E_{ce} clearly exhibits landscape "jumps" of higher severity. A small change to one of the weight values could have a dramatic effect on the E_{ce} value, indicating that optimization algorithms could get stuck oscillating between high and low E_{ce} values. The reason behind such behaviour can once again be attributed to the discrete nature of E_{ce}.

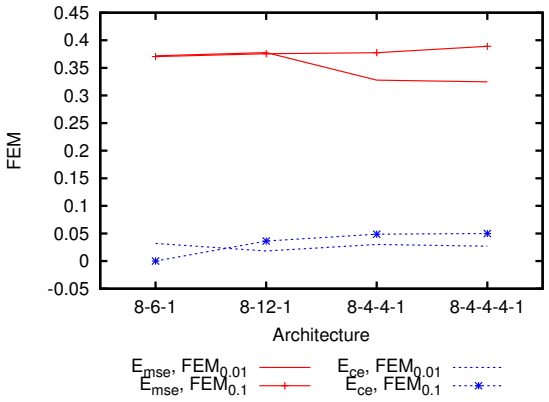
Table III shows that both G_{avg} and G_{dev} values on all problems considered were fairly low. Low gradient values are attributed to the fact that the search space bounds were also quite small, and the observed gradient magnitudes were proportional to the chosen bounds.



(a) XOR



(b) Iris



(c) Diabetes

Fig. 2. $FEM_{0.01}$ and $FEM_{0.1}$ values obtained for $[-1, 1]$ interval

B. Effect of multiple layers

One of the goals of this study was to see how the ANN error landscape features change with the change in ANN architecture. In this section, the ANN1 architecture for each problem, outlined in Table II and referred to as the optimal architecture, serves as the benchmark to compare against.

Figures 1a and 1c illustrate that simply increasing the dimensionality of the hidden layer to double the optimal size resulted in poorer searchability (FDS_s) for the XOR and the diabetes problems in the context of E_{mse}. Indeed, a redundant architecture will have irrelevant weights, introducing extra dimensions to the search space. The extra dimensions do not add any extra information in this case, and only divert the search, thus making the landscape less searchable. E_{ce}, already providing very little useful information, is not strongly affected by dimensionality change. The iris problem's FDS_s, depicted in Figure 1b, does not exhibit a drop with the increase in dimensionality from ANN1 to ANN2: the difference between iris and the other two data sets is that iris has three outputs per pattern, whereas XOR and diabetes only have one output each.

TABLE III
FLA MEASURES OBTAINED ON XOR, IRIS, AND DIABETES PROBLEMS (STANDARD DEVIATION IN PARENTHESIS)

Problem	Error	Configuration	FDC$_s$	$FEM_{0.01}$	$FEM_{0.1}$	G_{avg}	G_{dev}
XOR	E_{ce}	2-2-1	-0.00226 (0.01266)	0.31260 (0.05365)	0.11119 (0.02291)	0.07871 (0.01515)	0.68039 (0.04741)
		2-4-1	-0.00521 (0.01015)	0.33151 (0.02091)	0.12462 (0.02399)	0.02142 (0.01134)	0.33223 (0.04194)
		2-2-2-1	0.00111 (0.02090)	0.29014 (0.05365)	0.12283 (0.05051)	0.03769 (0.01933)	0.46210 (0.07979)
		2-2-2-2-1	0.00946 (0.04024)	0.29647 (0.01572)	0.07383 (0.01234)	0.02602 (0.01895)	0.37972 (0.12526)
	E_{mse}	2-2-1	0.11193 (0.08628)	0.16464 (0.00618)	0.37171 (0.00980)	0.01057 (0.00695)	0.01299 (0.00446)
		2-4-1	0.05099 (0.07846)	0.10162 (0.00787)	0.34368 (0.01217)	0.00308 (0.00792)	0.00863 (0.00465)
		2-2-2-1	0.20130 (0.07257)	0.12141 (0.01618)	0.34290 (0.00618)	0.00465 (0.00921)	0.00861 (0.01191)
		2-2-2-2-1	0.20194 (0.07275)	0.07677 (0.00941)	0.31452 (0.01406)	0.00341 (0.00767)	0.00801 (0.01190)
Iris	E_{ce}	4-4-3	0.01326 (0.02571)	0.27365 (0.03201)	0.30578 (0.03595)	0.06230 (0.03284)	0.37488 (0.11328)
		4-8-3	0.03237 (0.02188)	0.25143 (0.02417)	0.34119 (0.02199)	0.05335 (0.02973)	0.33929 (0.10957)
		4-2-2-3	0.03947 (0.03458)	0.21491 (0.06005)	0.20056 (0.04588)	0.07176 (0.03823)	0.74486 (0.21025)
		4-2-2-2-3	0.03786 (0.03798)	0.15489 (0.04727)	0.12925 (0.02980)	0.03864 (0.02562)	0.67496 (0.20871)
	E_{mse}	4-4-3	0.42313 (0.08198)	0.36893 (0.00312)	0.41828 (0.02980)	0.06689 (0.00399)	0.06821 (0.00369)
		4-8-3	0.43100 (0.07073)	0.37474 (0.00154)	0.40108 (0.01708)	0.06638 (0.00397)	0.06737 (0.00254)
		4-2-2-3	0.45957 (0.05124)	0.35579 (0.00952)	0.38363 (0.00713)	0.05315 (0.00580)	0.07039 (0.00859)
		4-2-2-2-3	0.42591 (0.06069)	0.34581 (0.01344)	0.38306 (0.01566)	0.04258 (0.00417)	0.06611 (0.00746)
Diabetes	E_{ce}	8-6-1	-0.02616 (0.03391)	0.03191 (0.04088)	0.11119 (0.00637)	0.01778 (0.01002)	0.07603 (0.05352)
		8-12-1	0.00513 (0.05117)	0.01824 (0.02121)	0.12462 (0.00413)	0.01386 (0.00944)	0.06758 (0.03321)
		8-4-4-1	0.04890 (0.05611)	0.03008 (0.02871)	0.12283 (0.00600)	0.01375 (0.01123)	0.12170 (0.06923)
		8-4-4-4-1	0.02941 (0.04855)	0.02696 (0.03211)	0.07383 (0.00459)	0.00483 (0.00862)	0.11061 (0.09967)
	E_{mse}	8-6-1	0.42919 (0.03626)	0.37205 (0.00252)	0.37171 (0.00325)	0.04280 (0.02388)	0.07482 (0.04552)
		8-12-1	0.26535 (0.04354)	0.37771 (0.00140)	0.34368 (0.00121)	0.03951 (0.01521)	0.07057 (0.03988)
		8-4-4-1	0.32163 (0.03211)	0.32793 (0.00115)	0.34290 (0.00364)	0.02883 (0.01472)	0.06842 (0.03216)
		8-4-4-4-1	0.25803 (0.04453)	0.32482 (0.00271)	0.31452 (0.00225)	0.02218 (0.01130)	0.05958 (0.02399)

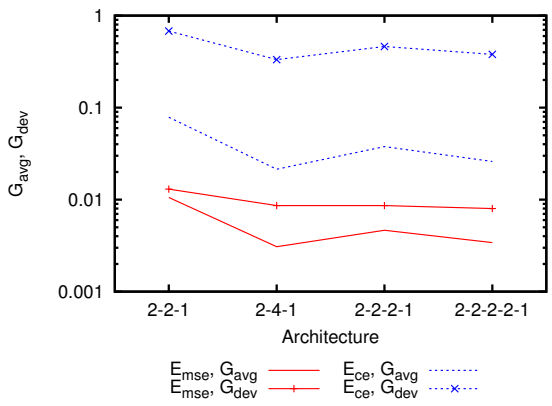
The iris data set, as simple as it may seem, is not the most trivial problem to solve, as two of the three flower types represented in the data set significantly overlap on two of the four input parameters [31]. Increasing the dimensionality of the ANN may have catered for those hidden complexities, making the error landscape slightly more coherent and searchable.

Adding extra layers, as shown in Figure 1, has a vastly different effect on the FDS$_s$ measure for the three problems considered. In case of XOR, FDS$_s$ visibly increases as an extra hidden layer is added. XOR is an extremely simple problem with only four binary training patterns. The ANN solution is essentially a simple non-linear bit-flipping function. Having two hidden layers instead of one adds robustness to the model and "slows down" signal propagation: a bit flipped in the first hidden layer can be reverted to its original state by the second hidden layer. This "resilience" of the final output to weight changes gives the ANN error landscape a simpler shape, making it more searchable. No difference in FDS$_s$ is observed between two and three hidden layer architectures. Table III shows that the maximum FDS$_s$ value observed on XOR was 0.2019, which is below moderately searchable (≈ 0.5). Low searchability levels can be attributed to the discrete nature of the logical XOR gate.

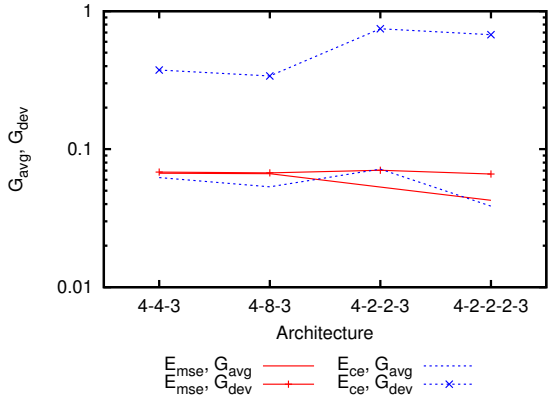
Figure 1b shows that iris benefited from adding the extra layers in terms of FDS$_s$. Extra hidden layers increase non-linearity and information capacity of the model, which proved useful in the context of local input parameter non-separability.

Diabetes, according to Figure 1c, became more searchable with more layers (ANN3 and ANN4) than with more neurons in a single hidden layer (ANN2). However, it is still the original optimal architecture that produced the highest average FDS$_s$ value. Out of the three problems considered, diabetes had the largest number of inputs. "Slower" propagation of the input signals through a deeper ANN and higher non-linearity must have made the landscape harder to search, increasing the chances of hidden unit saturation. The overall dimensionality of the problem also increased with the addition of extra layers, introducing more complexity into the landscape. Even though the extra complexity may be necessary for a more optimal classifier, training the additional free variables can indeed prove to be a difficult task.

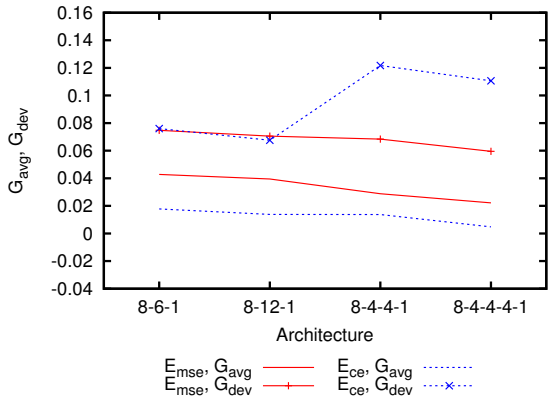
Figure 2 shows that for XOR and Iris, the macro-ruggedness measure $FEM_{0.1}$ decreased as the dimensionality of the ANN increased. In other words, adding extra hidden neurons and extra hidden layers made the error landscapes less rugged. The decrease in ruggedness can be attributed to the fact that as the dimensionality increases, the effect a single weight change has on the error value decreases, too. This makes sense: when there are more dimensions, the contribution of a single weight to the overall error is proportionally lessened. The $FEM_{0.1}$ for diabetes, however, shows a different trend and increases over dimensionality increase, as shown in Figure 2c. Micro-ruggedness $FEM_{0.01}$, however, disagrees with $FEM_{0.1}$, and shows the common decrease in ruggedness with increase



(a) XOR



(b) Iris



(c) Diabetes

Fig. 3. G_{avg} and G_{dev} values obtained for $[-1, 1]$ interval

in dimensionality for diabetes. The only difference between $FEM_{0.1}$ and $FEM_{0.01}$ is the maximum step size in the random walk, and it appears that in case of diabetes, the step size of $FEM_{0.1}$ is too large to pick up the landscape changes.

$FEM_{0.01}$ decreased for all problems as the dimensionality increased. This is the same expected behaviour as for $FEM_{0.1}$, but $FEM_{0.01}$ is fine-grained enough to reflect the landscape changes. The observations made on ANN error landscapes correspond to some benchmark functions such as Schwefel 2.22 and Quadric, which also showed a decrease in $FEM_{0.01}$ as the dimensionality increased [26].

Figure 3 shows that for E_{mse}, the G_{avg} value decreased as the ANN dimensionality increased. In case of the iris data set (Figure 3b), G_{avg} decreased only when more layers were added. Decrease in average gradient can be attributed to the same phenomena as that behind decreased ruggedness: a larger number of weights implies that the contribution of a single weight is diminished. In case of multiple layers, the chance of saturation is also increased, which implies wider plateaus in the landscape. The decrease in average gradient as a result of multiple hidden layers corresponds well to the known vanishing gradient phenomenon observed in deep neural networks [32]. Vanishing gradient is exactly what makes deep neural networks hard to train with a gradient descent-based approach.

According to Figure 3 and Table III, the average standard deviation of gradients, G_{dev}, did not decrease as much as G_{avg} as the dimensionality of the ANNs increased. Higher values for G_{dev} indicate the presence of sudden cliffs or ravines, which can be attributed to higher saturation in the deep ANN error landscapes. Cliffs and ravines can prove problematic to an optimisation algorithm, as it is easy to get trapped in them.

Figure 3 shows that E_{ce} exhibited different dynamics to E_{mse} in terms of gradients: on all problems except XOR, the G_{dev} values increased as the dimensionality of the problems increased. Such behaviour results from a combination of the discrete error measurement with increased ANN saturation: both introduce plateaus and sudden jumps into the landscape, resulting in a "staircase"-like error surface that is extremely hard to search.

VI. Conclusion

This paper applied continuous FLA metrics to ANN error landscapes. It was shown that the five FLA metrics chosen describe ANN error landscape characteristics accurately, and thus provide useful and interpretable information about the search space.

Adding hidden layers increased problem dimensionality, which resulted in less rugged, flatter landscapes with more treacherous cliffs and ravines. The observed results corresponded well to the vanishing gradient problem known to the deep learning community [32]. Further research can be done to see what step sizes for the FEM ruggedness measure provide the optimal level of granularity, and capture more relevant information about the changes in the landscape.

The measure of searchability, FDC_s, did not exhibit a general trend over all problems, capturing unique problem characteristics instead. Thus, FDC_s can potentially be used to determine which ANN architectures are more searchable than others for a particular problem. The FDC_s values showed that finding the optimal configuration could greatly increase the amount of information in the landscape.

It was also shown that E_{ce} error landscapes provide little to no information to the search, and E_{ce} should thus be avoided as a learning guide. On all problems considered,

E_{mse} provided a much more meaningful and searchable error landscape. Future research will investigate error landscapes produced by other error measurements, such as the cross-entropy error [33].

The effect of different search space boundaries on the error landscapes remains to be investigated. This paper made no comparisons of ANN activation functions: The influence of activation functions on ANN error landscapes is another very important topic to research in future.

REFERENCES

[1] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph.D. dissertation, Harvard University, 1974.

[2] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996.

[3] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[5] M. T. Hagan, H. B. Demuth, M. H. Beale *et al.*, *Neural network design*. Pws Pub. Boston, 1996.

[6] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Proceedings of the International Joint Conference on Neural Networks*. IEEE, 1989, pp. 593–605.

[7] J. D. Olden and D. A. Jackson, "Illuminating the black box: A randomization approach for understanding variable contributions in artificial neural networks," *Ecological modelling*, vol. 154, no. 1, pp. 135–150, 2002.

[8] S. Lawrence, C. L. Giles, and A. C. Tsoi, "What size neural network gives optimal generalization? convergence properties of backpropagation," Department of Electrical and Computer Engineering, University of Queensland, St. Lucia 4072, Australia, Tech. Rep., 1996.

[9] A. E. H. Paugam-Moisy, "Size of multilayer networks for exact learning: analytic approach," in *Proceedings of the 9th Conference on the Advances in Neural Information Processing Systems*, vol. 9. MIT Press, 1997, p. 162.

[10] T. Jones, "Evolutionary algorithms, fitness landscapes and search," Ph.D. dissertation, The University of New Mexico, 1995.

[11] P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 337–352, 2000.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[13] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[14] K. M. Malan and A. P. Engelbrecht, "A survey of techniques for characterising fitness landscapes and some possible ways forward," *Information Sciences*, vol. 241, pp. 148–163, 2013.

[15] T. Jones and S. Forrest, "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., 1995, pp. 184–192.

[16] K. Malan and A. Engelbrecht, "Characterising the searchability of continuous optimisation problems for PSO," *Swarm Intelligence*, vol. 8, no. 4, pp. 275–302, 2014.

[17] M. Gallagher, "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, 2001, pp. 157–166.

[18] V. K. Vassilev, T. C. Fogarty, and J. F. Miller, "Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application," in *Advances in evolutionary computing*. Springer, 2003, pp. 3–44.

[19] K. M. Malan and A. P. Engelbrecht, "Quantifying ruggedness of continuous landscapes using entropy," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 1440–1447.

[20] ——, "A progressive random walk algorithm for sampling continuous fitness landscapes," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2014, pp. 2507–2514.

[21] ——, "Ruggedness, funnels and gradients in fitness landscapes and the effect on PSO performance," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 963–970.

[22] M. Gallagher, "Multi-layer perceptron error surfaces: visualization, structure and modelling," Ph.D. dissertation, University of Queensland, 2000.

[23] S.-C. Wang, "Artificial neural network," in *Interdisciplinary Computing in Java Programming*. Springer, 2003, pp. 81–100.

[24] A. Choromanska, M. Henaff, M. Mathieu, G. Ben Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015, pp. 192–204.

[25] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in Neural Information Processing Systems*, 2014, pp. 2933–2941.

[26] K. M. Malan, "Characterising continuous optimisation problems for particle swarm optimisation performance prediction," Ph.D. dissertation, University of Pretoria, 2014.

[27] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1, New York, 1995, pp. 39–43.

[28] A. Gupta and S. M. Lam, "Weight decay backpropagation for noisy data," *Neural Networks*, vol. 11, no. 6, pp. 1127–1138, 1998.

[29] M. Carvalho and T. B. Ludermir, "Particle swarm optimization of feed-forward neural networks with weight decay," in *Proceedings of the Sixth International Conference on Hybrid Intelligent Systems*. IEEE, 2006, pp. 5–5.

[30] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[31] G. Grinstein, M. Trutschl, and U. Cvek, "High-dimensional visualizations," in *Proceedings of the Visual Data Mining Workshop*, 2001.

[32] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[33] D. M. Kline and V. L. Berardi, "Revisiting squared-error and cross-entropy functions for training neural network classifiers," *Neural Computing & Applications*, vol. 14, no. 4, pp. 310–318, 2005.