Fitness Distance Correlation of Neural Network Error Surfaces: A Scalable, Continuous Optimization Problem

Marcus Gallagher

School of Computer Science and Electrical Engineering
University of Queensland, QLD 4072, Australia
`marcusg@csee.uq.edu.au`

Abstract. This paper investigates neural network training as a potential source of problems for benchmarking continuous, heuristic optimization algorithms. Through the use of a student-teacher learning paradigm, the error surfaces of several neural networks are examined using so-called fitness distance correlation, which has previously been applied to discrete, combinatorial optimization problems. The results suggest that the neural network training tasks offer a number of desirable properties for algorithm benchmarking, including the ability to scale-up to provide challenging problems in high-dimensional spaces.

1 Introduction

Heuristic optimization algorithms such as Evolutionary Algorithms (EAs) are expected to be capable of solving high-dimensional, complex problems, requiring only the ability to be able to evaluate the objective function at any feasible point in the search space. These kinds of optimization problems arise frequently in many areas, and are often a fundamental part of methods used for data-driven tasks such as supervised and unsupervised learning. In recent years, heuristic optimization algorithms have been the focus of a very large amount of research, as well as being successfully applied to many challenging real-world problems.

Researchers face several difficulties when trying to evaluate the effectiveness of a given algorithm at solving problems. In particular, a set of benchmark optimization problems which can be usefully scaled to high-dimensions is an important requirement. Unfortunately, commonly used test functions have a number of short-comings in meeting this need. This paper investigates a particular form of a well-known task: training a feedforward neural network, as a possible source of test optimization problems. The evaluation of continuous heuristic optimization algorithms is discussed further in the following section. Section 3 considers the exploration of continuous optimization problems using correlation statistics and scatterplots. The student-teacher neural network training problem is then described in Section 4, and experimental results for a set of such problems is presented in Section 5. Section 6 concludes and summarizes these results. Note that here we are concerned mainly with solving continuous multivariate optimization problems, using algorithms which work directly with a continuous representation of solution vectors.

2 Heuristic Algorithms and Continuous Optimization

The standard multidimensional optimization problem involves a parameter vector $\mathbf{w} = (w_1, \dots, w_n)$, with the quality of a candidate solution being measurable by an objective function $f(\mathbf{w})$. The goal of optimization can then be simply to find a solution vector $\mathbf{w}^*$, which yields the best possible objective function value, subject to given resources (e.g, computation time). In some fields (e.g, evolutionary computation), optimization is normally defined as maximization of $f()$ (a fitness function) while in others (e.g, neural networks) the problem is equivalently framed as minimization of a cost (e.g, error) function. To describe and visualize an optimization algorithm (which performs an iterative, guided search in the space of possible solutions), the ideas of hill-climbing on a fitness landscape [5] and searching for the minimum of an error surface [10] have become well-known.

The focus of this paper is on continuous optimization problems (i.e, $\mathbf{w} \in \mathbb{R}^n$). Although heuristic algorithms have been successfully applied to many real-world optimization problems, interest in quantifying what kind of problems are easy or hard for algorithms such as EA's to solve has increased. This interest is partly due to the No Free Lunch (NFL) theorems, which indicate that there is no algorithm that outperforms all other algorithms in any general sense [12].

Researchers seek to understand and evaluate heuristic algorithms by empirical testing on sets of benchmark problems. Many of the test functions that have been used are analytical functions of continuous variables. Such functions can be easily visualized in one or two dimensions, and the locations of local and globally optimal points are often known precisely. Unfortunately, commonly used functions have a number of shortcomings [11]. Desirable properties for test functions include [4,7,11]:

- P1. difficult to solve using simple methods such as hill-climbing
- P2. nonlinear, nonseparable and nonsymmetric
- P3. scalable in terms of problem dimensionality
- P4. scalable in terms of time to evaluate the cost function
- P5. tunable by a small number of user parameters
- P6. can be generated at random and are difficult to reverse engineer
- P7. exhibit an array of landscape-like features.

Despite the large amount of research in developing new algorithms, very few attempts have been made to identify problems that possess many of these properties. Whitley et al. [11] and Michalewicz et al. [7] propose methods for generating analytical test functions for unconstrained and constrained optimization respectively.

3 Correlation Measures of Landscapes

A number of researchers (see, e.g. [9] and the references therein) have considered examining the correlation between fitness and distances between points on discrete fitness landscapes. These studies include the cost versus average distance

of an optimum point to all other points within a sample of local optima [2,8] and cost versus distance of local optima from the best optima found [6]. These studies on artificial and combinatorial optimization landscapes have indicated that a “Massif Central” [6] or “big valley” [2] structure seems to exist in many landscapes. That is, perhaps not surprisingly, cost in general seems to increase with distance from the best minimum, providing an intuitive picture of the landscape as a big bowl-like structure with smaller ridges, valleys and other structure imposed on it. These approaches require collecting a sample of local optimum points, which is infeasible in the continuous case.

One correlation measure which can be adapted to continuous error surfaces is the Fitness Distance Correlation (FDC) of Jones [5]. FDC is intended to provide a measure of the global structure of the surface in question, by examining how the value of the cost (fitness) function varies with the distance between a given point and a global optimum. A sample of random points on the surface P is chosen, and the standard sample correlation coefficient is calculated:

$$r = \frac{Cov(D, E)}{\sigma_D \sigma_E} \quad (1)$$

$$= \frac{n \sum^n DE - (\sum^n D)(\sum^n E)}{\sqrt{[n \sum^n D^2 - (\sum^n D)^2][n \sum^n E^2 - (\sum^n E)^2]}} \quad (2)$$

where D is a set of distances for a sample of points from the global optimum, E is the corresponding set of fitness function values for the sample and $Cov(D, E)$ is the covariance of D and E . σ_X is the standard deviation of X . The value of r is in the range $[-1, +1]$, with a value of $+1$ suggesting perfect, positive linear correlation between the objective function and the distance from the global optimum, and a value of -1 suggesting perfect negative correlation between these quantities.

FDC was proposed as an indicator of the difficulty of a search space for a Genetic Algorithm (GA) to solve [5]. The results support this measure, indicating that problems with a low FDC coefficient are hard for a GA to solve, whereas those with a high FDC coefficient value are GA-easy. While it has been shown that this correlation coefficient is not always successful at predicting the “GA-difficulty” of a problem [1,9], Jones suggests that FDC scatterplots provide a useful visualization of this relationship, even when the FDC coefficient does not summarize the relationship well [5].

4 The Neural Network Training Task

The Multi-layer Perceptron (MLP) [10] is the standard feedforward neural network model used in supervised learning problems. The model consists of a layer of N_i inputs, connected to one or more layers of “hidden” units, which are themselves connected to a final layer of N_o output units. Input signals propagate through the units in the network, via weighted connections, to the outputs.

The network mapping for a single hidden layer is given by

$$\mathbf{y} = \Psi(\mathbf{x}, \mathbf{u}, \mathbf{v}) \quad (3)$$

$$y_i = g_o\left(\sum_j u_{ij}h_j\right) \quad (4)$$

$$h_j = g_h\left(\sum_k v_{jk}x_k\right) \quad (5)$$

where $\mathbf{x}$ is an input vector, $\mathbf{u}$ and $\mathbf{v}$ are vectors of weights on the input connections to the hidden and output layers respectively, $\mathbf{y} = (y_1, \dots, y_{N_o})$ is an output vector, g_h is a sigmoidal activation function and g_o is typically either an identity function or another sigmoid. The latter is employed if the outputs are to be constrained to an interval (e.g. $(-1, 1)$). A linear bias term is also normally included as an input and is connected directly to the hidden and output layers, by setting $x_0 = \pm 1 \forall \mathbf{x} = (x_0, x_1, \dots, x_{N_i})$. This network architecture is known to be a universal approximator (with a sufficient number of hidden units) by appropriate choice of connection weight values $u_{ij}, v_{jk} \in \mathbb{R}$ [10].

MLP networks are typically trained to perform supervised learning tasks from a set of training data. As mentioned earlier, training is usually formulated as the minimization of an appropriate error function. The error surface of MLP networks has an interesting structure, which is dependent on a number of factors, such as the number of nodes in each layer, and the given training set. Although precise results concerning the structure of error surfaces are limited, they are typically considered to consist of a number of wide, flat plateaus, narrow ravines, symmetries due to the network topology and often a small number of unique non-global minima [3].

In this set of experiments, an artificial learning task is used, which is sometimes referred to as the *student-teacher* learning model (see [3]). In this problem, two networks are created. One network, the teacher, is initialized in some way, and then represents the “oracle” for the learning task. The training set is produced by generating random input vectors and passing these inputs to the teacher network, producing the desired output vectors. The other network, the student, is then trained using this data.

The main advantage, from an optimization viewpoint, of using this student-teacher model of learning is that it guarantees the presence of a known globally minimum point (ie, a parameter vector identical to the teacher weight configuration), where the value of the error function is zero. Symmetries in neural network error surfaces mean that there are $2^{N_h} N_h!$ functionally equivalent weight vectors to the teacher network, which are also global minima. Knowledge of the global minimum is required for calculating the FDC, which is normally not available on neural network training tasks derived from simple test problems like XOR, or from real world supervised learning datasets [10].

The error surface of an student-teacher MLP training task can be seen as having many of the properties listed in Section 2. Error surfaces are not easy to optimize over using simple methods like gradient descent [10] (P1). They are clearly nonlinear and nonseparable (part of (P2)), and scalable in problem dimensionality (P3) (furthermore, their properties can be changed by varying

user-parameters such as the number of training patterns and units in the network (P5)). The time to evaluate the cost function scales linearly with these user-parameters (P4), and networks can be generated randomly as desired (part of (P6)). This paper explores the possible features of these error surfaces (P7), in relation to varying three network parameters (P5), through a series of experiments and FDC-scatterplots¹.

In the following experiments, the student-teacher model is used to generate FDC results for MLP's. Throughout, mean-squared error is used as the error function and Euclidean distance is the distance measure. Teacher networks are generated by choosing their weights from a Normal distribution (with mean zero, standard deviation 5), in an attempt to generate networks with realistic weights (i.e. some large weights leading to some units saturating their outputs [3]). The "student" networks' weights are chosen from a Normal distribution (with mean $\mu_{teacher}$, standard deviation 1) distribution, and the weight vector is then scaled to a length chosen uniformly between 0 and 100. 5000 points are used to calculate the FDC coefficients, while for clarity only 2000 are used for the fitness-distance scatterplots. The number of input units, number of hidden units and the number of training patterns are varied in the experiments. All networks have a single output unit. The hidden and output units use the *tanh* activation function (i.e. g_h and g_o). Each unique training instance is run 10 times with different random initializations of teacher networks.

5 FDC Results for Error Surfaces

The FDC coefficient results are shown in Table 1. Each table entry reports the mean and standard deviation of the 10 different experiments for each training instance. Firstly, all coefficient values are positive, confirming intuition that moving away from a global minimum cannot lead to a *decrease* in error. At worst ($r \approx 0$), there is basically no correlation between the error value and the distance to the global minimum - the current location of a search or the trajectory up to some point in the search yields no information regarding the location of the global minimum. A general trend in this table is from relatively low r values for small numbers of inputs, hidden units and training patterns, to high values as these variables are increased. Standard deviations can be reasonably high for small values of the variables, and remain high even for large training sets when the network size is small.

From these r values alone, the indication is that for networks with more inputs and hidden units, distance from the global minimum and the current error value are related - implying that algorithms that are able to utilize this property will perform well in this situation. To allow a more detailed insight into the nature of these error surfaces however, the scatterplots of the experiments must be examined.

¹ The FDC acronym is used in this paper, although error (minimization) rather than fitness (maximization) is being considered.

Table 1. Fitness-Distance correlation values for student-teacher experiments.

No. Patterns	No. Inputs (y)/No. Hidden (x)				
	1	5	10	100	
1	1	0.185 (0.069)	0.214 (0.074)	0.228 (0.101)	0.269 (0.097)
	5	0.245 (0.067)	0.189 (0.104)	0.273 (0.066)	0.248 (0.100)
	10	0.207 (0.088)	0.260 (0.064)	0.249 (0.094)	0.198 (0.130)
	100	0.322 (0.083)	0.283 (0.049)	0.183 (0.123)	0.113 (0.096)
5	1	0.176 (0.073)	0.285 (0.083)	0.378 (0.049)	0.423 (0.073)
	5	0.266 (0.078)	0.401 (0.075)	0.409 (0.400)	0.466 (0.043)
	10	0.304 (0.082)	0.466 (0.053)	0.493 (0.051)	0.480 (0.087)
	100	0.403 (0.091)	0.476 (0.089)	0.494 (0.137)	0.309 (0.148)
10	1	0.213 (0.069)	0.318 (0.073)	0.314 (0.059)	0.459 (0.059)
	5	0.305 (0.058)	0.469 (0.073)	0.494 (0.075)	0.588 (0.107)
	10	0.317 (0.076)	0.529 (0.054)	0.573 (0.086)	0.594 (0.067)
	100	0.454 (0.128)	0.548 (0.103)	0.612 (0.063)	0.523 (0.089)
100	1	0.194 (0.073)	0.320 (0.120)	0.401 (0.056)	0.529 (0.051)
	5	0.378 (0.121)	0.534 (0.072)	0.689 (0.064)	0.827 (0.037)
	10	0.383 (0.140)	0.670 (0.085)	0.744 (0.060)	0.880 (0.020)
	100	0.418 (0.108)	0.787 (0.117)	0.883 (0.056)	0.864 (0.026)
1000	1	0.180 (0.069)	0.280 (0.104)	0.382 (0.065)	0.488 (0.063)
	5	0.335 (0.152)	0.585 (0.059)	0.694 (0.052)	0.845 (0.024)
	10	0.377 (0.110)	0.631 (0.122)	0.798 (0.055)	0.926 (0.017)
	100	0.506 (0.161)	0.854 (0.050)	0.926 (0.044)	0.970 (0.005)

Figures 1-4 show a representative sample of the kinds of scatterplots which were observed from the experiments. The general nature of all of the experiments can be summarized by this sample.

Figure 1(a) shows an FDC scatterplot for a 10-5-1(#1) network (i.e. a 10 input, 5 hidden unit, 1 output MLP) with a single training pattern). Clearly, two error values dominate the sample, over the range of distances examined - one which is very close to the global minimum and one which is also the highest (worst) error value found ($E \approx 2$). This result indicates that this error surface is largely dominated by two plateaus, and the small number of points at intermediate error values suggests that the transition between these two levels is relatively sharp. Note also that the high plateau is not seen until a distance of roughly $\|\mathbf{w}\| \simeq 20$ from the global minimum. In this case, the area surrounding the global minimum is quite flat, which will cause problems for algorithms seeking to converge to the exact global minimum. In practice however, this may be less of a concern - any point with an error so close to zero would probably be sufficient to halt training.

In Figure 1(b) the number of training patterns has increased to 5 for a 10-5-1(#5) network. The observable effect is the appearance of a number of intermediate levels between the $E \approx 0$ and the $E \approx 2$ levels. In addition, the levels appear incrementally as the distance increases. This observation accounts for the higher correlation coefficient ($r = 0.5235$) compared to Figure 1(a). The levels are equidistant, separated by $E \approx 0.4$ in error. This result implies that the

Fig. 1. (a)FDC scatterplot; 10-5-1(#1) network; $r = 0.3846$. (b)FDC scatterplot; 10-5-1(#5) network; $r = 0.5235$.

number of levels are directly related to the number of training patterns in this experiment. Lastly, the number of intermediate points between the levels has increased. The overall impression is that the global minimum is now situated in a “smaller bowl”, in that walking a shorter distance from the global minimum leads to much worse error values. The number of points at a level close to the global minimum also drops off with increasing distance, again suggesting a unique global minimum and surrounding neighbourhood.

Increasing the number of training patterns further leads to an extension of these trends. Figure 2(a) shows the scatterplot for a 10-5-1(#10) network. A greater number of levels dominate the plot, though the levels themselves are becoming less defined as the number of points between them increases and as they become closer together. The rate at which higher error values appear moving away from the global minimum has also increased, indicating a reduced region around the global minimum with small error values (and producing a lower correlation value $r = 0.4330$). At a large distance, points with $E \approx 0$ are becoming increasingly rare.

Fig. 2. (a)FDC scatterplot; 10-5-1(#10) network; $r = 0.4330$. (b)FDC scatterplot; 1-1-1(#1000) network; $r = 0.1501$.

The next scatterplot is a 1-1-1(#1000) net, shown in Figure 2(b). This small network with large training set produces a plot dominated by two levels similar to Figure 1(a). In this case however, the number of intermediate points has increased substantially, and as distance increases the number and magnitude of points with worse error values increases more rapidly (leading to a low $r = 0.1501$).

Fig. 3. (a)FDC scatterplot; 1-100-1(#1000) network; $r = 0.5040$. (b)FDC scatterplot; 100-1-1(#1000) network; $r = 0.3064$.

In Figure 3(a) (1-100-1(#1000) net), the levels which have dominated the above examples are all but gone. A significant number of points exist at one level at a value slightly above the global minimum, but only for smaller distances. A wide range of different error values is spread over the majority of the distance values. Note however that the intuition of a bowl surrounding the global minimum is supported with the smooth increase in worse error values moving away from the global minimum. The value of r has increased significantly from Figure 2(b) to Figure 3(a), due mainly to the disappearance of the high plateau of points and the diminishing lower plateau for high values of distance from the global minimum.

A scatterplot produced by a 100-1-1(# 1000) experiment (Figure 3(b)) shows some similarity to Figure 2(b). Two levels of error dominate the scatterplot, but the level close to zero error is much more common, and appears even at large distances from the global minimum. A greater number of points appear between these levels than in Figure 2(b). Also, the higher error level has dropped to a smaller value ($E \approx 1.6$) than in previous figures.

Several of the experiments with larger networks (with moderate to large training sets) produced a scatterplot similar to that shown in Figure 4(a) (100-100-1(# 1000) network). Now the picture is of a unique global minimum, about which error steadily rises. The error values are concentrated into a tube, reinforcing the idea of a big-valley structure and suggesting an overall smoother surface where a multitude of different error values can be found within this tube.

Fig. 4. (a)FDC scatterplot; 100-100-1(#1000) network; $r = 0.9706$. (b)FDC scatterplot; 100-1-1(#1000) network; $r = 0.5748$.

6 Discussion

These results show that a number of different phenomena can be observed in the structure of MLP error surfaces. A small number of training patterns induces a number of distinct levels onto the error surface. For small values of inputs and hidden units, the number of levels was observed to have an upper bound of $(k+1)$, where k is the number of training patterns. As the number of training patterns increases, the effect becomes more like a smoothing of the error surface - as levels blur together and intersect in various different ways. Restricted numbers of hidden units and inputs (e.g. Figures 2(b) and 3(b)) seem to be factors in producing levels in the scatterplots. As all three variables (inputs, hidden units, number of training patterns) become large, the error surface becomes globally smoother, with a near-continuous range of error values at various distances from the global minimum.

The above experiments attempt to describe the fundamentally different features of scatterplots observed. Nevertheless, several experiments produced plots whose behaviour can be described as some kind of combination of such features. An example is shown in Figure 4(b), for a 100-1-1(# 100) experiment. The “tube”-like structure seen in Figure 4(a) is evident for lower values of distance and error, but a two-leveled structure becomes increasingly prominent as distance increases (cf. Figure 3(b)). Three of the ten 100-1-1(# 100) experiments produced scatterplots similar to Figure 4(b), while the other seven closely resembled Figure 3(b). A more detailed study would be required to explain the interaction of these effects with varying numbers of inputs, hidden units and numbers of training patterns.

Overall, the FDC experiments indicate that the error surface has a complex structure. Furthermore, the structure of the error surface is sensitive to factors that are commonly varied when MLP’s are used - the number of inputs, number of hidden units and the number of training patterns. This observation suggests that the performance of an optimization algorithm can be expected to vary widely across different values for these variables, and across different training tasks. The positive values in Table 1 support the idea of a global big

valley structure in these MLP error surfaces. However, this picture is an oversimplification, as can be seen by the variety of structure in the FDC scatterplots above.

This paper has examined the error surfaces of student-teacher MLP network as a possible source of useful test functions for evaluating heuristic continuous optimization algorithms. The problems have most of the properties listed in Section 2 as desirable for test functions, including the capacity to exhibit a variety of different structural features (P7) in response to varying a small number of user parameters (P5). Although the known symmetries of error surfaces are related to the network topology [3,11], it seems unlikely that this would reduce their effectiveness as test functions (though further work is required to examine this possibility).

Heuristic global optimization algorithms typically do not require any knowledge about a problem other than the ability to evaluate the objective function. Hence, it is straightforward to apply several of these algorithms to the student-teacher MLP training problem and compare their performance empirically, in relation to FDC scatterplots and the parameters of this problem domain. A large-scale application of the test problems to several optimization algorithms is an obvious, fundamental extension of the feasibility study presented in this paper.

References

- [1] ALTENBERG, L. Fitness distance correlation analysis: An instructive counterexample. In *Seventh International Conference on Genetic Algorithms (ICGA97)* (San Francisco, CA, 1997), T. Baeck, Ed., Morgan Kaufman, pp. 57–64.
- [2] BOESE, K. D. *Models for Iterative Global Optimization*. PhD thesis, University of California, Los Angeles, 1996.
- [3] GALLAGHER, M. *Multi-layer perceptron error surfaces: visualization, structure and modelling*. PhD thesis, Dept. Computer Science and Electrical Engineering, University of Queensland, 2000.
- [4] HOLLAND, J. H. Building blocks, cohort genetic algorithms, and hyperplane-defined functions. *Evolutionary Computation* 8, 4 (2000), 373–391.
- [5] JONES, T. *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, 1995.
- [6] KAUFFMAN, S. *The Origins of Order*. Oxford University Press, Oxford, 1993, ch. 2, pp. 33–67.
- [7] MICHALEWICZ, Z., DEB, K., SCHMIDT, M., AND STIDSEN, T. Test-case generator for nonlinear continuous parameter optimization techniques. *IEEE Transactions on Evolutionary Computation* 4, 3 (2000), 197–215.
- [8] MÜHLENBEIN, H., GORGES-SCHLEUTER, M., AND KRÄMER, O. Evolution algorithms in combinatorial optimization. *Parallel Computing* 7 (1988), 65–85.
- [9] NAUDTS, B., AND KALLEL, L. A comparison of predictive measures of problem difficulty in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 4, 1 (2000), 1–15.
- [10] REED, R. D., AND II, R. J. M. *Neural Smithing: supervised learning in feedforward artificial neural networks*. The MIT Press, Cambridge, MA, 1999.
- [11] WHITLEY, D., MATHIAS, K., RANA, S., AND DZUBERA, J. Evaluating evolutionary algorithms. *Artificial Intelligence* 85, 1-2 (1996), 245–276.
- [12] WOLPERT, D. H., AND MACREADY, W. G. No free lunch theorems for search. Tech. Rep. SFI-TR-95-02-010, Santa Fe Institute, February 1995.