On Some Factors Influencing MLP Error Surface

Mirosław Kordos[1] and Włodzisław Duch[2,3]

[1] Faculty of Automatic Control, Electronics and Computer Science, The Silesian University of Technology, Gliwice, Poland.
[2] Department of Informatics, Nicholas Copernicus University, Toruń, Poland, http://www.phys.uni.torun.pl/kmk
[3] School of Computer Engineering, Nanyang Technological University, Singapore.

Abstract. Visualization of MLP error surfaces helps to understand the influence of network structure and training data on neural learning dynamics. PCA is used to determine two orthogonal directions that capture almost all variance in the weight space. 3-dimensional plots show many aspects of the original error surfaces.

1 Introduction

Multi-layer perceptron (MLP) error surface (ES) $E(\mathbf{W}) = \sum_{\mathbf{X}} ||\mathbf{Y} - M(\mathbf{X};\mathbf{W})||$ is defined in the weight space $\mathbf{W}$ (including biases as W_0 weights) for a given training data $\mathbf{X}$, desired output vector $\mathbf{Y}$ and structure of network mapping $M(\mathbf{X};\mathbf{W})$. Only mean-square error functions are considered here, so $||\cdot||$ is Euclidean norm and $E(\mathbf{W}) = \sum_{\mathbf{X}} ||\mathbf{Y} - M(\mathbf{X};\mathbf{W})||^2$. Learning processes are trajectories that lie on the hyper-surface $E(\mathbf{W})$ in the weight space $\mathbf{W}$. To understand learning dynamics error surface can be visualized using projections of the original space onto a three-dimensional subspace. In all plots presented here we use sigmoidal transfer functions, but ES projections obtained with hyperbolic tangent do not differ significantly.

It is beneficial to choose the projection directions which preserve most information about the original surface character. PCA (Principal Component Analysis) proved a good method of determining the directions. The network was trained using several algorithms, but the results were algorithm independent. The used algorithms included standard backpropagation (BP) [3], numerical gradient (NG) [1] and search-based methods (SM) [2]. Weight vectors $W(t)$ after each training epoch t were collected into the weight matrix. The training was stopped when the error begun decreasing very slowly (close to convergence). Singular Value Decomposition (SVD) was performed either on the weight matrix, or on the weight covariance matrix to determine principal components (all results here are for covariance matrices).

Typically the first and second PCA directions contain together about 95% of the total variance and therefore the plots reflect ES properties very well. The ES character is determined by the dataset and network structure but not by the training method and starting point. Several training methods (various versions of NG, SM and BP) have been used for the same network structure and training set. The training has been repeated several times for a given method with various random initial weights. Neither the random weight distribution, nor the training method, nor the number of training cycles for which PCA is calculated has significant influence on the ES presented in the space of two main PCA components. The plots may differ slightly, especially those obtained with BP, because BP depends more on initialization and produces ES projections that are not so uniform. The surface may rotate from one plot to another, its fragments may be a bit higher or lower, but the overall structure is well preserved. Experiments with over 20 datasets, most of them from the UCI dataset repository [4], have been made. Due to the limited space only a few ES are shown here. The name of a dataset in figure labels is followed by numbers of neurons in the successive layers; for example, in Fig. 1 Iris 4-4-3 means that the network trained on Iris data had 4 input, 4 hidden and 3 output neurons.

At the final stage of the training weights of output neurons tend to grow quicker then those of hidden neurons, but since the training is stopped before convergence weights of each layer have comparable contributions in determining PCA directions. Vertical axis in the plots shows relative error $E_r(\mathbf{W}) = E(\mathbf{W})/N_v N_c$, where N_v is the number of vectors and N_c is the number of classes in the training set. For all error functions based on Minkovsky's metric $||\cdot||_\alpha$ the error function is bounded from above by $N_v N_c$, thus the relative error is bounded by 1. Horizontal axes show distances in the weight space in c_1 and c_2 PCA directions corresponding to the first and second eigenvector of the weight covariance matrix.

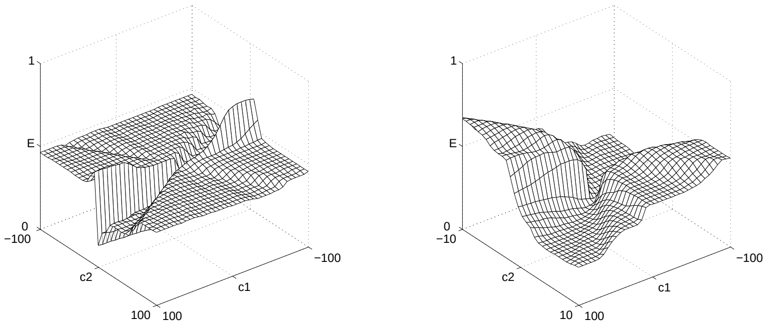

Fig. 1. The same error surface of a 3-layer network (Iris 4-4-3). Left: in original proportions, right: the scale of c2 axis multiplied by e_2/e_1 (this scaling is used for all drawings).

Usually the first PCA eigenvalue e_1 is an order of magnitude larger than the second one e_2. For that reason the plots are easier to interpret if unequal scales are used on horizontal exes (Fig. 1, right). For this purpose projections on c_2 are rescaled by the ratio e_2/e_1 of the second to the first eigenvalue of the weight covariance matrix. But it should be taken into consideration that in the rescaled plots the ill-conditioning and narrowness of the ravines are not so well visible as in pictures made in original proportions (Fig. 1, left).

2 Network Structure Influence on Error Surface

A network without a hidden layer has a very simple ES consisting only of two or four horizontal or slightly inclined half-planes, situated on various heights, with slopes connecting them (Fig. 2, left). ES of networks with hidden layers has a “starfish” structure. A vivid depiction of such ES was given by Denker et. al [5] “$E(\mathbf{W})$ surface resembles a sombrero or a phono record that has been warped in certain symmetric ways: near the middle ($\mathbf{W}$=0) all configurations have moderately bad E values. Radiating out from the center are a great number of ridges and valleys. The valleys get deeper as they go out, but asymptotically level out. In the best valleys, E is exactly or asymptotically zero, other valleys have higher floors”. The pictures presented in this paper confirm that global minima rarely create craters but frequently ravines reaching their minimum in infinity. This corresponds to the infinite growth of (usually output layer) weights when the training is continued for a sufficiently long time.

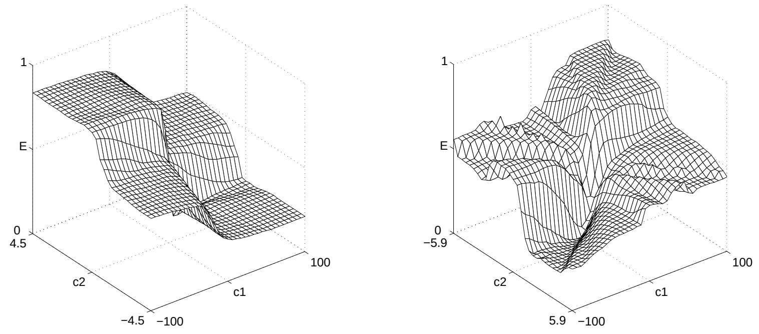

Fig. 2. Left: ES of 2-layer network (Iris 4-3); right: ES of 4-layer network (Iris 4-4-4-3).

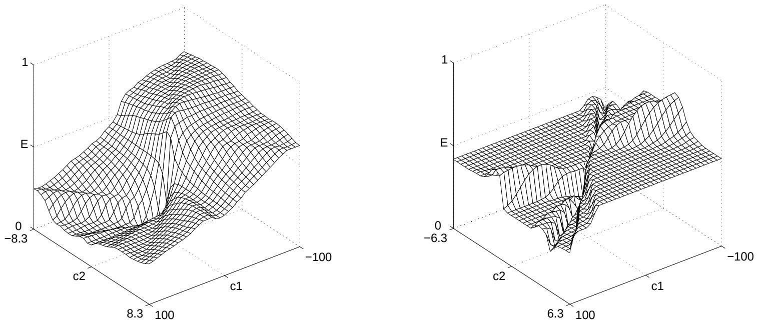

Fig. 3. Left: ES of 3-layer network with crossover connections (Iris 4-4-3); right: ES of 3-layer network with too many hidden neurons (Iris 4-100-3)

Each of h hidden neurons may be labeled by an arbitrary and unique number from 1 to h. Renumerating the network parameters does not change the mapping implemented by the network thus giving $h!$ permutational symmetries. A neural activation function for which $f(-x) = -f(x) + const$ gives further 2^h sign-flip symmetries [6]. This gives together $2^h h!$ equivalent global minima. A training algorithm converges to that minimum, which is easiest to reach from the starting point. Only some of the minima are clearly visible in the PCA projections. Their number originally grows with the increase of hidden neurons number, but with too many hidden neurons big horizontal planes begin to appear Fig. 3, right). This effect caused by the weight redundancy is better perceptible in a two-weight coordinate system, where the projected ES is almost flat since many weights must be changed at the same time to change the error.

In 3-layer networks with crossover connections the output layer is connected directly to both; the input (as in 2-layer networks) and the hidden layer (as in 3-layer networks). Consequently their ES display features of 2-layer networks (asymmetry of ES) and 3-layers networks (complexity of ES) (Fig. 3, left). A network with too few neurons in any hidden layer cannot map all required information and as a result is unable to learn the task. Its ES consists of several horizontal planes, all placed relatively high, with some rough areas between them, but it does not show characteristic ravines leading to global minima (not shown here). Four-layer networks have more complex ES than the three-layer ones, even with fewer neurons. Thus they can map more complex data (Fig. 2, right).

3 Training Data Influence on Error Surface

In all the experiments presented in this section a similar network structure x-4-2 has been used for various datasets. More complex training data produces more complex ES, especially if the data is not linearly separable, as in the case of n-bit parity.

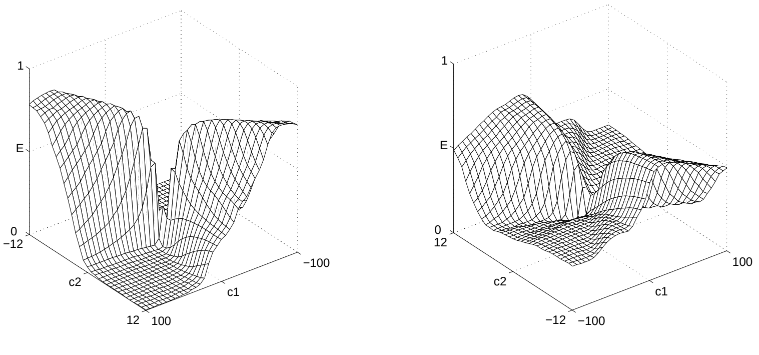

Fig. 4. Left: ES of Breast (10-4-3). Right: ES of Ionosphere (43-4-2)

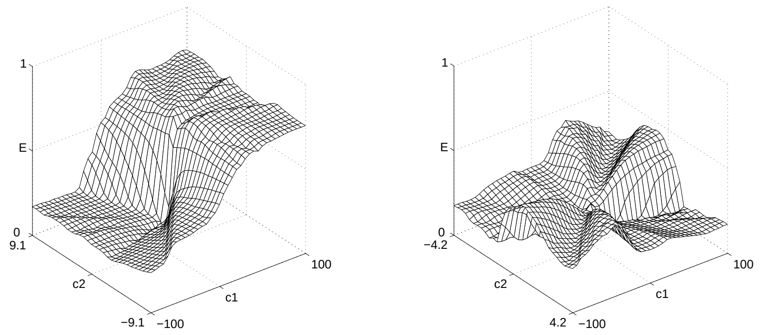

Fig. 5. Left: ES of entire Appendicitis dataset (12-4-3). Right: ES of Appendicitis dataset (12-4-3) with only 44 vectors - all 22 vectors of class 1 and randomly chosen 22 vectors of class 2.

Equal distribution of examples among classes leads to a more symmetric ES [7]. Appendicitis (21 vectors of class 0 and 85 of class 1) gives a highly non-symmetric ES (Fig. 5, left). Selecting 42 vectors from the dataset, all of class 0 and 21 vectors randomly chosen from class 1, produces a quite symmetric error surface. Other datasets have approximately equal number of vectors in each class thus their ES are more symmetric. Breast dataset has two classes with a few overlapping vectors, and therefore its ES is quite simple (Fig. 4, left). Iris (Fig. 1, right) has 3 classes with little overlap, and ionosphere (Fig. 4, right) two classes with some more overlap, and they both give similar ES. XOR data is linearly non-separable and therefore has a complex ES (Fig. 6, left). 6-bit parity (Fig. 6, right) is linearly non-separable and has 32 clusters per class (XOR has only 2). ES for even-bit parity problems is highly intricate, however it is symmetric because of equal class distribution.

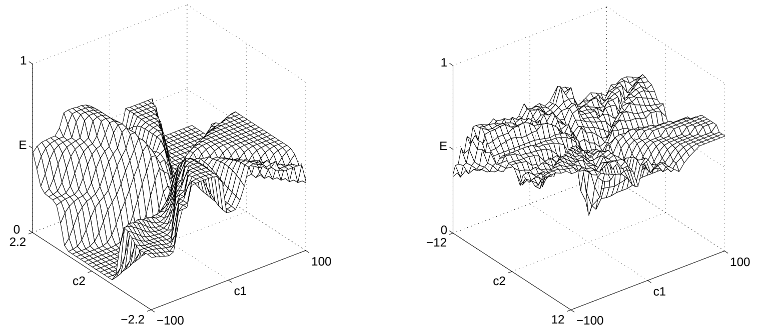

Fig. 6. Left: ES of xor (4-4-3). Right: ES of 6-bit parity (12-8-2).

4 Conclusions

Although it is impossible to see the error surface $E(\mathbf{W})$ without any distortions, displaying it in the first and second PCA component coordinate system gives good insight into many important ES properties (incomparably better than using any two weight system). Nevertheless due to the fact that local PCA directions are not constant in the entire weight space, such aspects of ES as ravine curvatures and slopes of their bottoms are not especially well reflected in the projections which use global PCA directions (the shape of ES projection is determined by weight changes in all training steps). For real-world data sets local minima in craters are very rare for networks with monotone transfer functions. Large plateaus accessible via narrow ravines, or ravines that lead to plateaus with larger error (due to poor network initialization) may cause many difficulties in neural training algorithms. The bigger is the difference between the first and the second eigenvalue, the more difficult and slower is the training procedure, because the training algorithm has to find proper direction very precisely. When the difference exceeds two orders of magnitude the training effectiveness may be severely affected.

ES depends on network structure, training data, transfer and error functions, but not on training methods. ES has greatest diversity close to its center. Far from the center flat horizontal planes occupy large areas. If the range of random initial weights is too broad then it is likely that the starting point lies somewhere on the flat area, and as a result the network cannot be trained with gradient-based or local search method. On the contrary, if all initial weights are zero the network can be successfully trained with search-based techniques [2]. Backpropagation methods cannot start from zero weights [3], but this is only due to the limitations of the algorithms, and not of the properties of the zero point on the error surface.

Perhaps an interesting suggestions from this study will be also to use PCA to re-duce the effective number of training parameters to a few.

References

1. M. Kordos, W. Duch, “Multilayer Perceptron Trained with Numerical Gradient." Int. Conf. on Artificial Neural Networks, Istanbul, June 2003, pp. 106-109
2. M. Kordos, W. Duch, “Search-based Training for Logical Rule Extraction by Multilayer Perceptron." Int. Conf. on Artificial Neural Networks, Istanbul, June 2003, pp. 86-89
3. S. Haykin, *Neural networks: a comprehensive foundations*. New York: MacMillian Publishing, 1994.
4. C.J. Mertz, P.M. Murphy, UCI repository of machine learning databases, http://www.ics.uci.edu/pub/machine-learning-data-bases.
5. J. Denker et. al. “Large automatic learning, rule extraction and generalization". Complex Systems, 1, pp. 887-922, 1987.
6. H.J. Sussmann, “Uniqueness of the weights for minimal feedforward nets with a given input- output map", Neural Networks, 5, pp. 589-593, 1992.
7. M.R. Gallagher, “Multi-layer Perceptron Error Surfaces: Visualization, Structure and Modeling", PhD Thesis, University of Queensland, 2000