# Revisiting the Problem of Weight Initialization for Multi-Layer Perceptrons Trained with Back Propagation

Stavros Adam[1], Dimitrios Alexios Karras[2], and Michael N. Vrahatis[3]

[1] Dept. Mathematics, University of Patras Artificial Intelligence Research Center (UPAIRC), GR-26110 Patras, Greece and TEI Hpeirou, Arta, Greece

[2] Dept. Automation , Chalkis Institute of Technology, Psachna, Evoia GR-34400 and Hellenic Open University, Greece
dakarras@{ieee.org,teihal.gr,usa.net}

[3] Dept. Mathematics, University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-26110 Patras, Greece

**Abstract.** One of the main reasons for the slow convergence and the suboptimal generalization results of MLP (Multilayer Perceptrons) based on gradient descent training is the lack of a proper initialization of the weights to be adjusted. Even sophisticated learning procedures are not able to compensate for bad initial values of weights, while good initial guess leads to fast convergence and or better generalization capability even with simple gradient-based error minimization techniques. Although initial weight space in MLPs seems so critical there is no study so far of its properties with regards to which regions lead to solutions or failures concerning generalization and convergence in real world problems. There exist only some preliminary studies for toy problems, like XOR. A data mining approach, based on Self Organizing Feature Maps (SOM), is involved in this paper to demonstrate that a complete analysis of the MLP weight space is possible. This is the main novelty of this paper. The conclusions drawn from this novel application of SOM algorithm in MLP analysis extend significantly previous preliminary results in the literature. MLP initialization procedures are overviewed along with all conclusions so far drawn in the literature and an extensive experimental study on more representative tasks, using our data mining approach, reveals important initial weight space properties of MLPs, extending previous knowledge and literature results.

## 1 Problem Statement and Previous Work

BP training suffers from been very sensitive to initial conditions. In general terms, the choice of the initial weight vector $\boldsymbol{w_0}$ may speed convergence of the learning process towards a global or a local minimum if it happens to be located within the attraction basin of that minimum. Conversely, if $\boldsymbol{w_0}$ starts the search in a relatively flat region of the error surface it will slow down adaptation of the connection weights.

Sensitivity of BP to initial weights, as well as to other learning parameters, was studied experimentally by Kolen and Pollack [1]. Using Monte Carlo simulations on feed forward networks trained with BP to learn the XOR function they discovered that convergence of these networks exhibits a complex fractal-like structure as a function of initial weights. On the other hand, analytical studies for the same problem were

M. Köppen et al. (Eds.): ICONIP 2008, Part II, LNCS 5507, pp. 308–315, 2009.

reported by Hamey [2] who reconsiders the XOR problem and provides a theoretical study of the error surface for the standard mean square error function. However, he notes the difficulty of having analytic solutions for the general pattern classification case as the study of the error surface is hampered by high dimensionality and because of the difficulty of theoretical analysis. In light of these results it seems that it is not possible, in general, to provide complete theoretical verification for a number of research results claiming to cope effectively with the problem of weight initialization. This is, partially, due to the fact that an exhaustive study of the error surface and of the learning dynamics is almost unfeasible for the general case of the pattern classification problem. On the other hand it is tempting to examine if the initial weight space possesses some kind of structure or if it is able to reveal features which may lead to an effective choice of initial weights. To this end, an effective means seems to be the analysis of the weight space of MLPs in different pattern classification problems. This also permits to gain significant evidence on the validity of different results having either a theoretical basis or proven by experiments.

In this paper we revisit the problem of weight initialization for neural networks trained with gradient descent based procedures. We verify, experimentally, a number of results reported by several researchers for the XOR-network and we extend these results to a well known problem, the IRIS classification problem. Our approach is based on clustering of the weight vectors after having trained an MLP with the BP procedure. Classification of the weight vectors into clusters is performed using unsupervised clustering of Kohonen's self organizing feature maps, or simply self-organizing maps (SOM). Results of our experiments not only reveal, as it was expected, the basins of attraction for the gradient descent learning algorithm, but also provide significant evidence that no inherent clustering exists for the initial weight space. Our approach consists in performing analysis of the weight space after having trained an MLP with the BP procedure for a significant number of weight vectors and for various different sets of training patterns. This approach has already been used by other researchers in the XOR problem, but what is new here is its application to a well known real life problem, the IRIS classification problem. Analysis of the weight space is done using a data clustering and visualization technique. We consider that this approach extends results obtained previously by other researchers. Main considerations of these previous researches are presented hereafter.

## 2 Analyzing the Weight Space for MLP Using Kohonen's Self Organizing Feature Maps, as a Data Mining Tool for the Analysis

Data clustering and visualization of the clusters, in this paper is based on Kohonen's SOM. The SOM is a type of neural network which is based on unsupervised learning. Thus, unlike supervised learning methods, a SOM is able to perform clustering of data without any reference to the class membership of the input data.

Training the map is an iterative process. At each step a sample vector $x$ is randomly chosen from the input data set and distances between $x$ and all the codebook vectors are computed. Distances between codebook vectors and sample data correspond to similarities between input data and units of the SOM. The best matching unit (BMU), i.e. the most similar unit, is the map unit whose weight vector is closest to *x*. The training algorithm updates the weight vector of the BMU and of those of its neighborhood so as to get these units move closer to the input vector *x*, i.e. diminish their distance to the sample vector [3,4,5]. More details on SOM can be found in [3].

The SOM algorithm performs a mapping from the high dimensional input space onto map units. This mapping preserves topology, in the sense that, relative distances between data points in the input space are preserved by distances between map units. This means that data points lying near each other in the input space will be mapped onto neighboring map units. The SOM can thus serve as a clustering tool of high dimensional data. Compared to standard techniques (k-means, ISODATA, competitive learning etc) SOM not only performs better in terms of effectively clustering input data to unknown clusters but also it is computationally more effective [3], [4]. Other comparisons and studies on the data mining capabilities of SOM can be found in the literature. We should mention here the use of the SOM Toolbox for SOM training, data visualization, validation and interpretation. SOM Toolbox was developed at Helsinki University of Technology [5].

We considered two classical benchmarks, the XOR function and the Iris classification problem. The XOR function was studied with a 2-2-1 network while the IRIS classification problem was investigated with two different network architectures, one with 4-10-3 units and another one with 4-5-3 units. For all units the logistic sigmoid was used as an activation function. Experiments for both problems and for different network architectures were carried out according to the following steps:

1. MLPs were trained with the on line BP learning algorithm. All experiments were carried out with the same training parameters, that is interval for initial weights [-2.0, 2.0], learning rate 0.9, max number of epochs 30000 and error between target and actual network output less than 0.01.
2. A relatively large number of weight vectors, that is 5000, were chosen from the initial weight space. Weight vectors were randomly sampled in the interval [-2.0, +2.0] using uniform distribution. After training, the set of weight vectors was roughly divided into two distinct subsets, or categories, of weight vectors. One subset was made up from, those weight vectors for which both, training succeeded (the error goal was reached), and generalization performance was good, i.e. less than 20% of previously unseen patterns rejected per class. These vectors are called the ***successful*** weight vectors while those not meeting the above criteria are called the***failed*** weight vectors and they fall within the second category.
3. For each weight vector $w_i^0$ considered before training, the MLP was trained with the on-line gradient descent and a weight vector $w_i^*$ after training was obtained. Thus, gradient descent is considered mapping the weight space before training *W* onto the weight space after training *W'*. Given the high dimensionality of these spaces we then used SOMs and projected each one of them on the 2-dimensional space. This approach is graphically depicted in Figure 1.

$$\begin{tikzcd}
W^{(n)} \arrow[r, "gd"] \arrow[d, "som"'] & W'^{(n)} \arrow[d] \\
W^{(2)} & W'^{(2)}
\end{tikzcd}$$

**Fig. 1.** How SOM could be used as a data mining tool for clustering weight space

4. The 2-dimensional projections of *W* and *W'* thus obtained presented the clusters of weight vectors being discovered by the SOM. Visual inspection of the map representing *W'* permitted to draw some interesting qualitative information regarding the basins of attraction for the gradient descent procedure. Activation of the SOM units and visualization of the unified distance matrix (UM) to identify classification of weight vectors into different clusters. Details on these results are presented in the following section.
5. We, finally, used the possibility offered by the SOM Toolbox to identify the weight vectors for which a unit of the SOM is activated to verify density of *W* regarding convergence and generalization. Actually, given a SOM node in a cluster of successful weight vectors we identified one weight vector before training $w_i^0$ that gave after training a successful weight vector $w_i^*$. By injecting additive noise, with normal distribution $N(0,\sigma^2)$, on $w_i^0$, we took a number of weight vectors in the vicinity of $w_i^0$. Retraining the MLP with the same BP procedure and mapping the weight vectors after training on the SOM we discovered that even for very small variance many of the noisy weight vectors did not behave the same way as $w_i^0$.

## 3 Main Results and Discussion

The tool for presenting results and analyzing them is the unified distance matrix (UM). UM represents the organization of the SOM units into groups, as uniform areas on the 2-dimensional grid.

**Result 1.** Clustering of the weight vectors after training, which is performed by the SOM without any class membership information, depicts uniform regions of unit activity corresponding to clusters of ***successful*** weight vectors and thin borderline areas for the ***failed*** weight vectors. Figures 2, a and b, visualize clustering of the weights for the 4-10-3 IRIS classification network, while Figures 2, c and d are representative for the 4-5-3 network.

The clusters formed by the SOM correspond to the various minima reached by the gradient descent throughout each experiment. These minima can be global or local. In this sense and together with the topology preservation mapping of the SOM it is straightforward to assume that clusters indicate basins of attraction for the dynamics of the learning procedure. This explains why the number of ***successful*** weight regions for the 4-5-3 IRIS network is less than the respective number for the 4-10-3 IRIS network. This constitutes an experimental confirmation that as the number of unit in the hidden layer increases the number of basins of attraction increases and therefore the study of the weight space becomes more difficult; see Kolen and Pollack [1].

**Result 2.** Execution of step 5, described above, for a number of different values of $\sigma^2$ demonstrated that even for very small variance many of the noisy weight vectors did not behave the same way as the initial vector $w_i^0$ , i.e. they did not result in successful training. Experiments showed that it is not possible to safely conclude on a minimum “size” for a neighbor of a ***successful*** weight vector in which gradient descent results in ***successful*** weight vectors.

Though important the above results are of practical importance in terms of weight initialization. In order to acquire a better idea on how to deal with this matter we proceeded in a number of experiments using the 4-10-3 MLP for the IRIS problem. During these experiments we used values for the synaptic weight randomly chosen from intervals $\left[-\alpha,+\alpha\right]$, with ***a*** varying from -6.0 up to +6.0, by a step of 0.20. Results of these experiments are stated hereafter.

**Result 3.** Training seems to be very sensitive to the choice of the training patterns. For the same interval of initial weight vectors and even the same weight vectors, learning curves and subsequent generalization of BP are clearly different.

However, during these trails we did not adopt some specific strategy on how to choose the training patterns and so it remains unclear what characteristic of the input space really biases the learning phase. A possible explanation relies on the inherent structure of the IRIS problem, where two classes are highly correlated. Finally, it seems that a good “strategy” to overcome this problem is to carry out training changing the set of training patterns every 50 or 100 initial weight vectors, these numbers chosen arbitrarily.

**Result 4.** Training tends to be more successful when the weight vectors are chosen in an interval $[-a,a]$ with $\alpha \approx \sigma_p{}^2$ , where $\sigma_p{}^2$ is the maximum variance of the variables of the input pattern space.

While this result is in the same line with some previous research outcomes, it seems that it more accurately reflects a good strategy for weight initialization than previous similar results in the literature. This paper shows that it is not possible to be more specific in the weight initialization range than the above result. More experiments, however, are needed to establish such an outcome.

**Result 5.** While training seems to be more successful for values of the initial weights within some interval $[-a,a]$ as described above, it is very likely for he BP to give a successful; learning curve for even greater values in intervals $[-ka-a/k,ka]\cup[ka,ka+a/k]$, where k a natural number.

Finally, figures 3,4 below demonstrate the validity of our results 4, 5 above by illustrating how generalization performance is affected by the initialization range when this increases. In the six samples below we see that up to a variance point as indicated by the results 4,5 of the initialization range, there are possibilities for obtaining better generalization than in all other cases. Incrementing this range we find points in the weight space where no solution can be granted, but afterwards, again, there are solutions but with less generalization capability than within the smaller range. This validates the view that even in larger ranges solutions exist, not so successful perhaps, but with less possibility than within the smaller initialization ranges.

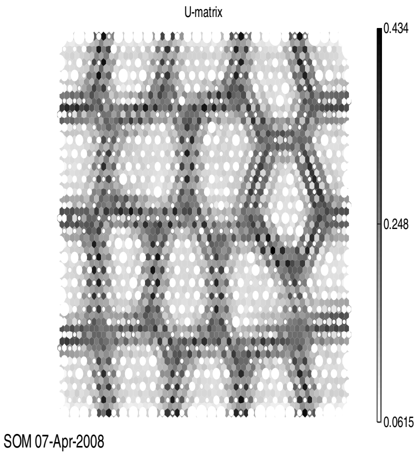

**Fig. 2a).** Mapping of weight vectors for the Iris network. Mapping of the successful weight vectors for the 10 hidden units Iris network.

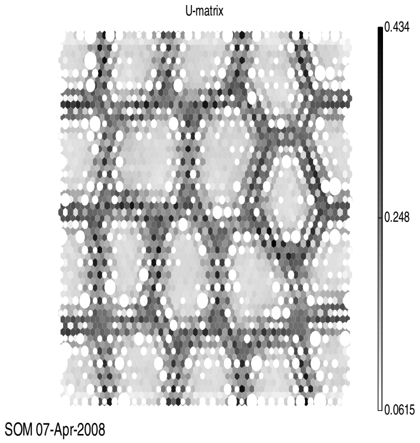

**Fig. 2b).** Mapping of weight vectors for the Iris network. Mapping of the failed weight vectors for the 10 hidden units the Iris network.

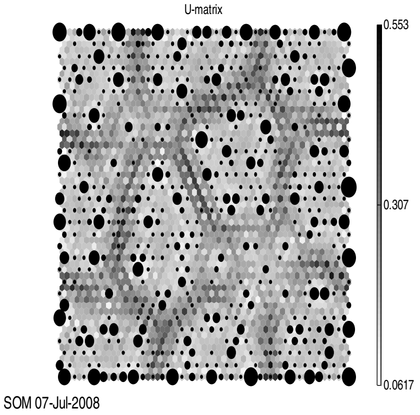

**Fig. 2c).** Mapping of weight vectors for the Iris network. Mapping of the successful weight vectors for the 5 hidden units Iris network.

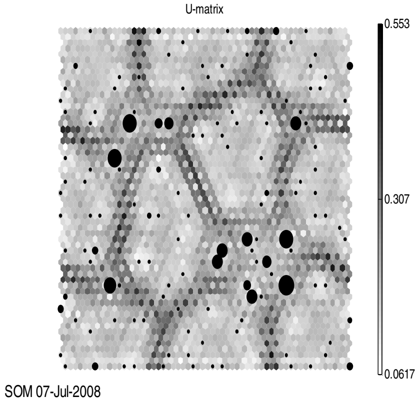

**Fig. 2d).** Mapping of weight vectors for the Iris network.Mapping of the failed weight vectors for the 5 hidden units Iris network.

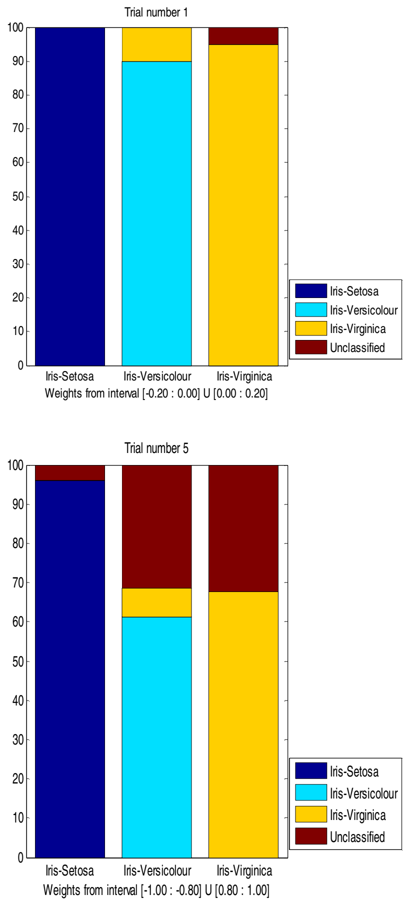

**Fig. 3a).** How MLP Generalization is affected by initial weights distribution for the Iris network. Misclassification results are shown for selection of the initial weights from the intervals [-0.2 0.0] U [0.0 0.2](upper slide) and [-1 -0.8] U [0.8 1] (lower slide).

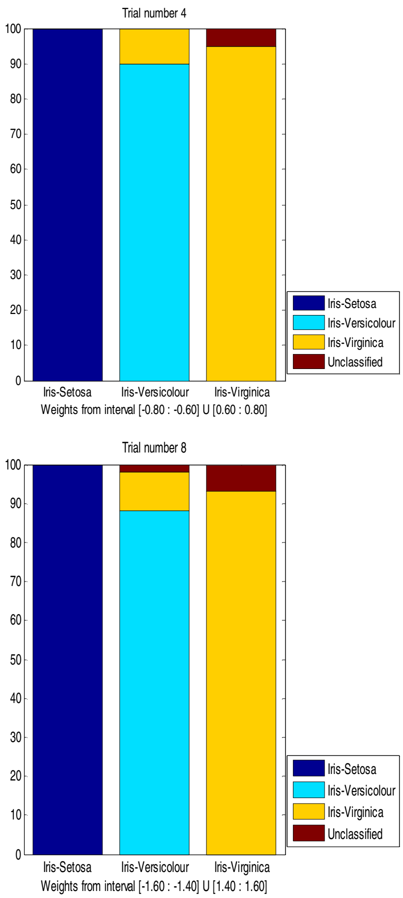

**Fig. 3b).** How MLP Generalization is affected by initial weights distribution for the Iris network. Misclassification results are shown for selection of the initial weights from the intervals [-0.8 0.6] U [0.6 0.8](upper slide) and [-1.6 -1.4] U [1.4 1.6] (lower slide).

## 4 Conclusions and Future Trends

This paper revisits MLP initialization problem in the case of BP training and extends literature results both in the description of the weight space as well as in the estimation of a good strategy for selecting weight initialization range. The analysis is performed on a complex classification task, like Iris problem, which is more representative of "real" world problems characteristics than benchmarks used so far in the literature. To this end, a data mining approach, based on Self Organizing Feature Maps (SOM), is involved in this paper. The conclusions drawn from this novel application of SOM algorithm in MLP analysis extend significantly previous preliminary results in the literature. More detailed analysis on real world benchmarks is needed to establish better these results and more elaborate specification of the weight initialization range than the ones of results 4, 5 in this study are needed not, however, too "accurate" as in previous studies. Previous studies have been misleading in this aspect not showing that the weight initialization space is not dense in solutions but it follows an almost fractal structure and, therefore, a probabilistic approach is more suitable in order to find out a good strategy for MLP weight initialization.

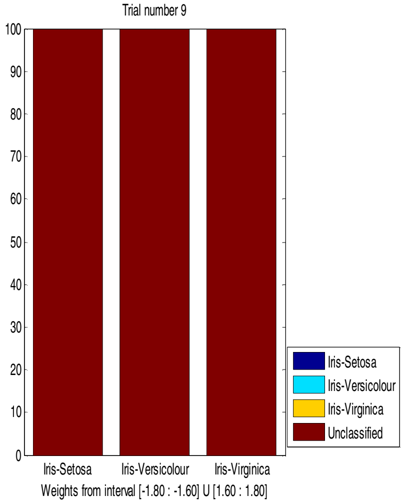

**Fig. 4a).** How MLP Generalization is affected by initial weights distribution for the Iris network. Misclassification results are shown for selection of the initial weights from the intervals [-1.8 -1.6] U [1.6 1.8].

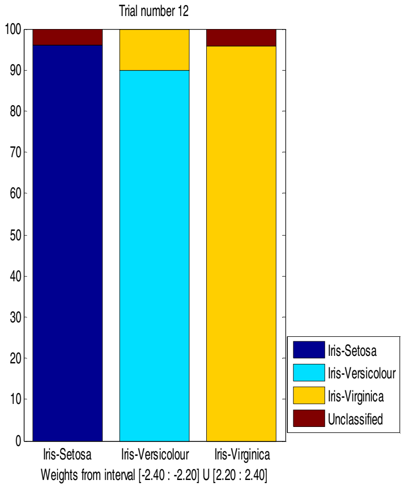

**Fig. 4b).** How MLP Generalization is affected by initial weights distribution for the Iris network. Misclassification results are shown for selection of the initial weights from the intervals [-2.4 -2.2] U [2.2 2.4].

## References

1. Kolen, J.F., Pollack, J.B.: Back propagation is sensitive to initial conditions. In: Advances in Neural Information Processing Systems 3, Denver (1991)
2. Hamey, L.: Analysis of the Error Surface of the XOR Network with Two Hidden Units. In: Proc. 7th Australian Conf. Artificial Neural Networks, pp. 179–183 (1996)
3. Kohonen, T.: Self-Organization and Associative Memory. Springer, Heidelberg (1989)
4. Olli Simula, O., Vesanto, J., Alhoniemi, E., Hollman, J.: Analysis and Modeling of Complex Systems Using the Self-Organizing Map. In: Neuro-Fuzzy Techniques for Intelligent Information Systems (1999)
5. Technical Report on SOM Toolbox 2.0, Helsinki University of Technology (April 2000), `http://www.cis.hut.fi/projects/somtoolbox/`