ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Training Multi-Layer Perceptron with Multi-Objective Optimization and Spherical Weights Representation

Honovan P. Rocha and Marcelo A. Costa and Antônio P. Braga *

Graduate Program in Electrical Engineering - Federal University of Minas Gerais - Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil

Abstract. This paper proposes a novel representation of the parameters of neural networks in which the weights are projected into a new space defined by a radius r and a vector of angles Θ. This spherical representation further simplifies the multi-objective learning problem in which error and norm functions are optimized to generate Pareto sets. Using spherical weights the error is minimized using a mono-objective problem to the vector of angles whereas the radius (or norm) is fixed. Results indicate that spherical weights generate more reliable and accurate Pareto set estimates as compared to standard multi-objective approach.

1 Introduction

It has been well established in the literature that the general problem of learning from data has a multi-objective nature [1, 2]. Artificial Neural Networks (ANNs) learning should be accomplished by minimizing both the empirical risk $R_{emp}(\mathbf{w})$ and the model capacity h [1, 2]. Since $R_{emp}(\mathbf{w})$ and h have a conflicting behavior it is not possible to jointly minimize them, so a trade-off is required. According to this formulation, ANNs model selection should be accomplished amongst the Pareto set solutions of $R_{emp}(\mathbf{w})$ and h. From this perspective, the general problem of model induction from data can be regarded as a trade-off problem, however, the candidate solutions should be generated prior to model selection.

In order to solve the bi-objective problem it is usual to adopt a constrained optimization approach by minimizing norm with constrained error [3] or by minimizing error with constrained norm [2]. Constrained optimization of non-linear objective functions, however, has well-known numerical and convergence-related difficulties. With the goal of overcoming some of them, in this paper ANNs multi-objective learning is reformulated and presented as an unconstrained optimization problem.

The principle of the proposed method is based on a spherical representation of Multi-Layer Perceptrons (MLP), so that error minimization can be accomplished for different radius r_j of the n-dimensional circle $w_1^2 + w_2^2 + \cdots + w_n^2 = r_j^2$. The geometrical concept is intuitive and can be observed in the schematic example of Figure 1. Inequality constraint methods such as the ϵ-constrained [2] minimize the error within the disc $||\mathbf{w}||^2 \leq r_j^2$. Equality constraint approaches [4] minimize the error in the circle $||\mathbf{w}||^2 = r_j^2$. In both approaches, the optimization problem

*This work has been supported by the Brazilian agencies CAPES and CNPQ.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

is then solved for pre-established values of r_j so that a portion of the Pareto set is generated. Once the set of Pareto set solutions is obtained, a decision making procedure picks-up one of them according to a selection criteria. The method proposed here has no constraints since the weights are represented as spherical coordinates and the independent variables to be optimized are actually the angles of the spherical representation for a given radius r_j.

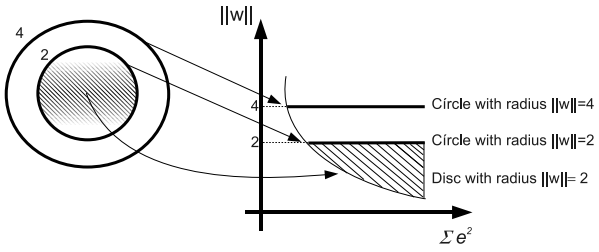

Fig. 1: Mapping from parameter's space to objective's space by constraining the solutions to the region $||\mathbf{w}|| \leq r$.

The structure of the present paper is as follows. The spherical representation is presented in section 2. Section 3 presents the backpropagation algorithm using spherical weights. Section 4 presents results, and discussion and conclusion are presented in section 5.

2 Spherical weigths

We consider a multi-layer perceptron with one hidden layer and one linear output neuron. Thus, the neural network output equation given an arbitrary input vector $\mathbf{x}$ of dimension N, $\mathbf{x}^T = [x_1, \ldots, x_N]$ is given by:

$$y_{(\mathbf{x})} = \sum_{i=1}^{H+1} \varphi_i \cdot f_i^h \left[\sum_{j=1}^{N+1} (\psi_{ji} \cdot x_j) \right] \tag{1}$$

where H is the number of hidden neurons, N is the number of inputs, ψ_{ji} is the input layer weights, φ_i is the output layer weights and f^h is the activation function of the hidden layer neurons. The bias components were augmented to the neurons inputs and weights as: $x_{N+1} = 1$, $f_{H+1}^h = 1$, $\dot{f}_{H+1}^h = 0$ where $\dot{f}^h$ is the hidden function derivative.

Let vector $\mathbf{w}$ be composed of all input and hidden layer weights, $\mathbf{w}^T = [\varphi_1, \ldots, \varphi_{H+1}, \psi_{1,1}, \ldots, \psi_{N+1,H+1},]$. Hereafter, each element of weight vector is represented by w_i. Therefore, the norm of the weights is written as $||\mathbf{w}||^2 = \sum_i w_i^2$. Using spherical weights we refer to the norm of the weights as the radius r, where $r^2 = ||\mathbf{w}||^2$. For instance, if the weight vector is of dimension 2, i.e., $\mathbf{w}^T = [w_1, w_2]$, then it can be represented using spherical weights by radius r and angle θ as $\mathbf{w}^T = [r \sin\theta, r \cos\theta]$. In this case, the parameter of interest is θ

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

which can be adjusted to minimize the error function. In general, the elements of vector $\mathbf{w}$, of size n, can be written as functions of $n-1$ angles which define vetor $\Theta^T = [\theta_1, \ldots, \theta_{n-1}]$, and the radius r as shown in Equation 2.

$$w_i = \begin{cases} r\ sin(\theta_1) & i = 1. \\ r\ \prod_{k=1}^{i-1} cos(\theta_k) sin(\theta_i) & i = 1, 2, \ldots, n-1. \\ r\ \prod_{k=1}^{i-1} cos(\theta_k) & i = n. \end{cases} \tag{2}$$

Recall that n is the dimension of the weight vector $\mathbf{w}$, i.e., the sum of the weights in both hidden and output layers. From Equation 2 the estimates of the angles θ_i ($0 \leq \theta_i < 2\pi$) can be calculated from the weight values, as shown in Equation3

$$\theta_i = \begin{cases} tan^{-1}\left(\frac{w_i}{w_{i+1}} sin(\theta_{i+1})\right) & 1 \leq i < n-1. \\ tan^{-1}\left(\frac{w_{n-1}}{w_n}\right) & i = n-1. \end{cases} \tag{3}$$

The use of spherical weights creates a new subset of parameters, Θ, of dimension $n-1$ which can be optimized in order to find the optimal weights of the neural network with minimum error. In this new space of parameters, the norm of the weights, or radius, is fixed. Therefore, the minimization of the error function, with respect to the vector of angles Θ can be seen as a mono-objective optimization problem without any constraint. On the other hand, the resulting weights have a fixed norm which, in fact, represents, a constrained optimization solution in the original space ($\mathbf{w}$).

3 The backpropagation algorithm using spherical weights

The standard *backpropagation* algorithm aims at minimizing the error function by updating the vector of weights in the opposite direction of the gradient at point $\mathbf{w}_0$, where $\mathbf{w}_0$ is the initial weight vector. The weight update equation is shown in Equation 7.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha \cdot \left.\frac{\partial \sum e_i^2}{\partial \mathbf{w}}\right|_{\mathbf{w}_k} \tag{4}$$

where $\sum e_i^2$ is the error function and α is the learning rate parameter. In order to update the vector of angles we simply apply the standard chain rule with respect to the gradient: $\frac{d\sum e_i^2}{d\theta} = \frac{d\sum e_i^2}{dw} \times \frac{dw}{d\theta}$. Using matrix notation we define matrix $\mathbf{T} = \frac{\partial \mathbf{w}}{\partial \Theta}$, as shown in Equation 5. A simpler approach to calculate the elements of matrix $\mathbf{T}$ using elements of the weight vector $\mathbf{w}$ is shown in Equation 6.

$$\mathbf{T} = \frac{\partial \mathbf{w}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial w_1}{\partial \theta_1} & \cdots & \frac{\partial w_n}{\partial \theta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial w_1}{\partial \theta_{n-1}} & \cdots & \frac{\partial w_n}{\partial \theta_{n-1}} \end{bmatrix} \tag{5}$$

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

$$[\mathbf{T}]_{ij} = \begin{cases} 0, & i = 2, \ldots, n;\ j = 1, \ldots, i-1 \\ \prod_{k=1}^{i} cos\theta_k & i = j \\ -w_j^* \frac{sen\theta_i}{cos\theta_i} & i = 1, \ldots, n-1;\ j = i+1, \ldots, n \\ -w_{n-1}^* & i = n-1;\ j = n \end{cases} \quad (6)$$

where $\mathbf{w}^* = \mathbf{w}/r$. Therefore, the *backpropagation* update equation using spherical weights is written as:

$$\Theta_{k+1} = \Theta_k - \alpha \cdot \mathbf{T} \times \mathbf{g}|_{\Theta_k} \quad (7)$$

where $\mathbf{g}$ is the gradient vector, $[\mathbf{g}]_j = \frac{\partial \sum e_i^2}{\partial w_j}$.

4 Results

The proposed spherical method was compared to the multi-objective (MOBJ) algorithm [2]. Both algorithms were applied to regression and classification problems that are listed in Table 1. For each regression problem, 100 samples were generated. A Gaussian noise with zero mean and $\sigma^2 = 0.2^2$ variance was included to the output. In addition, 12 classification problems were used. Three of them represent synthetic data, a four-class classification problem [2] and two obtained in R *package* clusterSim: two.moon and circles2. The other nine datasets are from real problems chosen from the UCI Repository [7] except banana dataset that was obtained from Kell repository. For each data set, a five-fold cross validation procedure was applied. Within the training set, i.e., the four blocks, 20% of the data was randomly chosen as validation set. One block was used as test set. Both MOBJ and spherical algorithms were applied to a multilayer perceptron with 10 hidden nodes, with hyperbolic activation functions. The Pareto set was estimated using a grid of 30 equally spaced norm values in the range 0 to 20 for the regression problems, and 0 to 50 for the classification problems. We evaluated the MOBJ and spherical results by comparing the Pareto sets generated by each method, and evaluating the mean squared error (mse) and accuracy (acc) with respect to the test set.

The MOBJ and spherical Pareto sets were compared using the dominated hypervolume or S-Metric statistic [6]. This statistic calculates the hypervolume between a multidimensional region, determined by the Pareto solutions, and a dominated solution used as the reference point. This metric was chosen by taking into consideration two important measures to compare Pareto sets: convergence and diversity.

Figure 2 illustrates the ability of the spherical method to generate improved estimates of the Pareto set as compared to MOBJ, using the bupa data set. The figure shows average values of the Pareto sets and the average values of selected solutions for each method, using the cross validation folds. The dominated solution used as reference point is also shown in the figure. It can be seen that the Pareto set generated using the spherical method has smaller errors than the Pareto set generated using MOBJ and, therefore, it is more accurate than

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

MOBJ. It can be seen that in the lower norm region the two methods have similar solutions. Nevertheless, the spherical Pareto set dominates the MOBJ Pareto set. It is worth noting that the selected Pareto solution using MOBJ share a similar error value as compared to the spherical solution. However, the spherical solution has a smaller norm value.

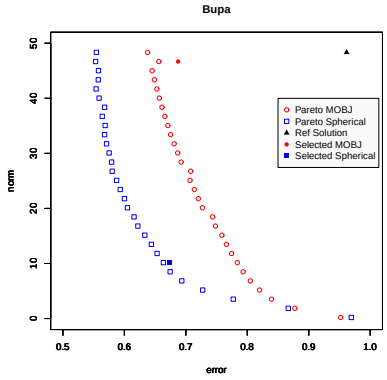

Fig. 2: Pareto sets generated by MOBJ and spherical weights for bupa data set.

Table 1 shows average values of the hypervolume statistic **(Hv)**, mean squared error **(mse)** for regression data sets, and test set accuracy **(Acc)** for classification data sets, using the cross validation procedures. Standard deviation values are shown in parenthesis. **N** is the sample size and **D** is the number of input variables.

Values of **Hv**, **mse** and **Acc** were used to compare the Pareto sets and the selected solutions. Furthermore, the Wilcoxon Signed-Rank Test was applied, as recommended in [5], to compare **Hv** and **Acc** results. The test evaluates whether the differences of the medians between MOBJ and spherical methods are statistically significant, at α-level. We used $\alpha = 0.01$. Large values of **Hv** and **Acc**, and smaller values of **mse** are shown in bold type in Table 1. It can be seen that the spherical method achieved the best performance in most regression and classification problems, and the best **Hv** values. P-values of the Wilcoxon test were 0.0003 and 0.35758 for **Hv** and **Acc**, respectively. Thus, the null hypothesis was rejected for the **Hv** statistic. Therefore, it can be concluded that our method generates more efficient Pareto sets than the MOBJ. Nevertheless, both methods generate solutions with similar accuracies (**Acc**).

5 Conclusions and Discussions

Theoretically, the Pareto set represents the region of the objectives space where non-dominated solutions are located. It also represents the best subset of solutions from which one final solution with maximum generalization ability, or minimum error and minimum complexity, is chosen. In practice, many algorithms

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

Table 1: Regression and classification results

			MOBJ		**Spherical**	
Dataset	**N**	**D**	**Hv (sd)**	**mse(sd)**	**Hv (sd)**	**mse(sd)**
$f1(x) = sin(x)$	100	1	8.6(0.4)	0.04(0.01)	**9.2(0.4)**	**0.03(0.01)**
$f2(x) = (x-2)(2x+1)/(1+x^2)$	100	1	31.9(0.7)	0.08(0.07)	**37.6(1.0)**	**0.06(0.04)**
$f3(x) = 4.26(e^{-x} - 4e^{-2x} + 3e^{-3x})$	100	1	3.6(0.5)	0.05(0.01)	**3.9(0.5)**	0.05(0.02)
$f4(x) = (e^{-0.2x}) + (2e^{-0.2x} \times sin(2\pi \cdot 0.2x - \pi/4) - 0.27)$	100	1	7.6(0.7)	0.12(0.03)	**10.5(0.7)**	**0.05(0.01)**
$f5(x) = sin(\pi x)/(\pi x)$	100	1	3.1(0.3)	0.60(0.03)	**6.6(8.1)**	**0.42(0.24)**
	N	**D**	**Hv (sd)**	**Acc (sd)**	**Hv (sd)**	**Acc (sd)**
gaussian 4 classes	200	2	51.3(5.4)	**84.5(4.8)**	**57.8(6.3)**	82.5(7.7)
circles2	200	2	37.8(0.8)	100(0.0)	**49.2(0.6)**	100(0.0)
two.moon	200	2	41.0(0.5)	**100(0.0)**	**43.3(0.5)**	98.5(2.2)
breast cancer	569	31	34.7(0.1)	96.5(1.1)	**36.2(0.1)**	**97.0(1.5)**
diabetes	768	8	17.1(0.5)	77.2(3.0)	**19.3(0.9)**	**77.3(3.4)**
sonar	208	60	42.9(0.4)	84.6(7.1)	**45.1(0.1)**	**85.1(4.8)**
bupa	345	6	11.1(1.2)	72.2(4.4)	**15.9(2.1)**	**74.2(4.9)**
vertebral column	310	6	20.8(0.4)	**86.1(4.4)**	**25.7(1.1)**	85.8(4.5)
heart diseases	270	13	34.9(0.8)	83.3(3.7)	**41.5(0.4)**	83.3(3.6)
blood transfusion	748	4	5.8(0.2)	77.8(5.3)	**6.3(0.3)**	**77.9(5.0)**
banana	5300	2	10.6(0.5)	81.9(2.1)	**24.5(1.5)**	**88.8(1.3)**
australian credit approval	690	14	30.0(0.9)	85.7(4.3)	**32.4(1.4)**	**86.2(4.4)**

try to estimate the Pareto set, and most algorithms are based on multi-objective approaches. We propose a transformation scheme in which the weights of neural networks are represented by angles. Results show that the optimization of the angles with respect to the error function generates improved Pareto estimates, i.e., solutions with smaller errors for fixed norm values as compared to standard multi-objetive solutions. Our results show that with respect to the error function a standard multi-objective algorithm (MOBJ) generates solutions which are very similar to the solutions generated using spherical weights. On the contrary, the spherical weights generates solutions with much smaller complexity, i.e., smaller norm values.

References

[1] Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag (1995)

[2] Teixeira, R.A., Braga, A.P., Takahashi, R.H.C., Saldanha, R.R.: Improving generalization of mlps with multi-objective optimization. Neurocomputing 35, 189-194 (2000)

[3] Boser, B., Guyon, I., Vapnik, V.: A Training algorithm for optimal margin classifiers. In: Fifth Annual Workshop on Computational Learning Theory, pp. 144–152, San Mateo, CA:Morgan Kaufmann (1992)

[4] Costa, M.A., Braga, A.P., Menezes, B.R.: Improving generalization of MLPs with Sliding Mode Control and the Levenberg-Marquadt algorithm. Neurocomputing 70, 1342–1347 (2007)

[5] Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Mach. Learn. Research 7, 1–30 (2006)

[6] Zitzler, E. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Application*, Ph.D. dissertation, Swiss Federal Int. Technology (ETH), Zurich, Switzerland, (1999)

[7] Blake, C. L. and Merz, C. J.: UCI Repository of machine learning databases. Irvine,CA: University of California, Dept. of Information and Computer Science, http://www.ics.uci.edu/ mlearn/MLRepository.html (1998)