Incorporating a priori Knowledge into Initialized Weights for Neural Classifier

Zhe Chen [1], Tian-Jin Feng [1], Zweitze Houkes [2]

1 Department of Electrical Engineering, Ocean University of Qingdao, Qingdao 266003, P.R. China
email: **sage0616@ouqd.edu.cn, tjfeng@ouqd.edu.cn** , fax: 86-532-2032799
2 Department of Electrical Engineering, Twente University, 7500 AE Enschede, The Netherlands
email: Z.Houkes@el.utwente.nl, fax: 31-53-4891067

Abstract

Artificial neural networks (ANN), esp. multilayer perceptrons (MLP) have been widely used in pattern recognition and classification. Nevertheless, how to incorporate *a priori* knowledge to design ANN is still an open problem. This paper tries to give some insightful discussions on this topic emphasizing weight initialization from three perspectives. Theoretical analyses and simulations are offered for validation.

1 Introduction

Multilayer perceptron (MLP) with sigmoidal nonlinearity is one of most popular neural classifier due to its powerful ability in functional approximation and generalization, however, it often encountered some drawbacks in practice such as slow convergence speed and local minima. Therefore, improving MLP performance is of very importance to its wider applications. Among many efforts, incorporating knowledge known *a priori* to the neural classifier attracted much attention from many researchers [1], e.g. network structure [2], activation function [15], objective function [12], and weight initialization [3,5,6]. This paper will give a brief discussion on designing initialized weights from three perspectives for multilayer neural classifier. Geometric and theoretical analyses are given to design weight initialization strategy for specific classification problems. The simulation results demonstrated the initialized weights embedding *a priori* knowledge could lead to a better performance in convergence and generalization than the *ad hoc* ones.

2 Notation and background

Without loss of generality, two-layer (single hidden layer) MLP is discussed here. Denoting $\mathbf{W} = \{W_{jk}\}_{H\times N}$ and $\mathbf{V} = \{V_{ij}\}_{M\times H}$ as the weight matrix of input-to-hidden and hidden-to-output interconnection respectively, thus the network architecture can be described as

$$O_i = f\left(\sum_{j=1}^{H} V_{ij} H_j - b_i\right) \tag{1}$$

where $H_j = f(net_j)$ and $net_j = \sum_{k=1}^{N} W_{jk} x_k - b_j$. $f(\cdot)$ is sigmoid activation function $1/(1+e^{-x})$, which can be regarded as an approximately piecewise linear function.

Suppose MLP classifier is trained to separate the C-classes $\{\omega_1, \omega_2, \cdots, \omega_C\}$ among the given training samples, in geometric interpretation, the hidden units are trained to form a set of hyperplanes (or hyperspheres) in N-dimensional input space, on which the final decision boundary are constructed. The extreme value (0 / 1) of sigmoid output is an indication to the direction of half hyperplane (hypershere), on either side of which containing the input vector.

3. Examples and Simulations

I Geometric Perspective

3.1 *Circle-decision-area-forming (CDAF) problem*

The CDAF problem is to form a decision boundary for separating the two-class patterns, which are located inside or outside a circle (see Fig.1). By observing the 16 bi-category pattern pairs, the decision boundary between them is generally a circle in Bayesian sense, and it is such geometric constraint that is conducive to exploring a fast solution. Keep in mind that sigmoid function can be regarded as an approximately linear piecewise function, hence the decision circle can be approximated with 16 piecewise lines (shown in Fig.1). Supposing each hidden unit is to perform a

0-7695-0619-4/00 $10.00

linear partition between each pattern pair, the hyperplane spanned by the each hidden unit is characterized by

$$net_j = \sum_{k=1}^{N} W_{jk} x_i - b_j = 0, \tag{2}$$

and the decision-rule is generated as follows

if $(W_{j1}x_1 + W_{j2}x_2 - b_j > 0)$, then $\mathbf{X} \in \omega_1$, if $(W_{j1}x_1 + W_{j2}x_2 - b_j < 0)$, then $\mathbf{X} \in \omega_2$,

where $\mathbf{X}=[x_1, x_2]^T$ in two-class classification case. Henceforth, there must exist a set of points (x_1', x_2') satisfying

$$W_{j1}x_1' + W_{j2}x_2' - b_j = 0. \tag{3}$$

We draw a tangent line at a given arbitrary point $P'(x_1', x_2')$ of decision circle, which cross the abscissa and ordinate at $M(b_j/W_{j1}, 0)$ and $N(0, b_j/W_{j2})$, respectively (shown in Fig. 2). Observing that $x_1' = r\cos\alpha$, $x_2' = r\sin\alpha$, (r is the radius of the decision circle) and noticing $\Delta OP'P \sim \Delta NMO'$, we have

$$\frac{x_2'}{x_1'} = \frac{b/W_{j1}}{b/W_{j2}} = \frac{W_{j2}}{W_{j1}} = \frac{\sin\alpha}{\cos\alpha}. \tag{4}$$

Therefore, the weight pairs $\{W_{j1}, W_{j2}\}$ form a circle equation.

Upon above geometric analysis, one can incorporate the *a priori* knowledge into weight initialization: provided $W_j = \{W_{j1}, W_{j2}\}$ (j=1,2,...,H) were initialized randomly among a "weight circle", V_j were initialized randomly in the interval [-1,1], the network achieved a considerable convergence improvement over the traditional weight initialization strategy (see Table 1). On the other hand, one can particularly write an algorithm to learn the given task, in which the W_j are preset as constant. In the experiments, we found that, if one keep initialized weights $\{W_{j1}, W_{j2}\}$ as a circle distribution and unchanged during the training phase, the network can also achieved convergence with even slightly fewer learning epochs. The advantage of such strategy lies in decreasing the unknown parameters of network and thereby computational and storage requirements, it can also be optimistically estimated that the generalization ability will be improved. In addition, this idea can be extended to other classification problems whose patterns satisfy the circle distribution, such as the circle-in-the-square problem.

3.2 ***Two-spiral problem***

The two-spiral benchmark was considered as one of the most difficult problems in two-class pattern classification field due to the complicated decision boundary [10]. It is extremely hard to solve using MLP models trained with various BP algorithms [5,8,12]. In the following, we will prove that the weight pairs $\{W_{j1}, W_{j2}\}$ of nested spiral problem also satisfy the circle equation, based on which a similar weight initialization strategy can be derived. In Cartesian coordinate, the points on each spiral can be expressed as

$$x_1 = r\theta\cos\theta, \qquad x_2 = r\theta\sin\theta. \tag{5}$$

As shown in Fig. 3, suppose a piecewise linear decision boundary between the nested spirals, (noting that the decision boundary is also a spiral trajectory), thus the above-mentioned decision-rule still holds. Given an arbitrary point P' (x_1', x_2') in the decision spiral and drawing a tangent line NM passing P', one can also find a *circle of curvature* passing P'. By linking the tangent point P' and the circle center $C(\xi, \eta)$, one can have $C'P' \perp NM$ and two similar triangles $\Delta NMO \sim \Delta CP'D$, and it follows that

$$\frac{b_j/W_{j1}}{b_j/W_{j2}} = \frac{W_{j2}}{W_{j1}} = \frac{x_2' - \eta}{x_1' - \xi}. \tag{6}$$

And one can further obtain the *center of curvature* $C(\xi, \eta)$ as

$$\xi = x_1' - \frac{(x_2')' \cdot (x_1' - x_2'\theta) \cdot (1+\theta^2)}{\theta \cdot (2+\theta^2)}, \qquad \eta = x_2' - \frac{(x_1' - x_2'\theta) \cdot (1+\theta^2)}{\theta \cdot (2+\theta^2)}. \tag{7}$$

According to (6) and recalling (5), we have

$$\frac{W_{j2}}{W_{j1}}=\frac{x_2'\theta-x_1'}{x_2'+x_1'\theta}=\frac{a\theta^2\sin\theta-a\theta\cos\theta}{a\theta^2\sin\theta+a\theta\cos\theta}=\frac{\theta\sin\theta-\cos\theta}{\sin\theta+\theta\cos\theta}=\frac{\sin(\theta-\beta)}{\cos(\theta-\beta)}. \quad (8)$$

where $\beta=arcctg(\theta)$ and the proof is completed. For mathematical background on *circle of curvature* and detailed derivation, see [3]. The simulations also confirmed our proposition, all of networks initialized with "weight circle" have excellent classification accuracy and generalization (see Fig.4), the results are summarized in Table 2. It is noteworthy to point out that, there are only 151 and 181 unknown parameters for NMLP with 50 and 60 hidden units, fewer than the average 181.75 connections reported in Cascade-Correlation algorithm [7], and the required CPU time for convergence was also better than many previously reported results (for instance, [5],[8],[12]). Similarly, the above conclusion can be generalized to four-spiral problem (Fig.5), the experimental results also showed the considerable improvement in convergence speed with proposed method while dealing with that more complicated problem.

II Pattern Perspective

3.3 *Symmetry test problem*

Symmetry test problem, originally studied in [13], was studied here to show how the prior knowledge can be utilized to find a simple solution. Let's denote $\mathbf{X}=[x_1,\cdots,x_{N/2},x_{N/2+1},\cdots,x_N]$, $x_k=\{-1,1\}$, the network is trained to accomplish the task to produce the output as the following rule:

$$O=\begin{cases}1 & x_k=x_{N-k+1}\quad (k=1,\cdots N)\\ 0 & otherwise\end{cases}. \quad (9)$$

Observing the characteristic of binary code of input patterns, we rewrite the following expression:

$$net=\sum_{k=1}^{N}W_kx_k=\sum_{k=1}^{N/2}(W_kx_k+W_{N-k+1}x_{N-k+1})\xrightarrow{W_k=W_{N-k+1}}\sum_{k=1}^{N/2}W_k(x_k+x_{N-k+1}). \quad (10)$$

Hence, $\mathbf{X}$ is symmetric if $x_k=-x_{N-k+1}$ for all k (i.e., net≡0), and net=±2 / 0 otherwise. If we let $W_k=C^k$, (C is any positive integer C>2), s.t. $\pm\sum_{k=1}^{k=N/2}W_k\neq 0$ and net≠0 in unsymmetrical case. By incorporating such simple constrained information into initialized weights, the problem is easily solved by one-layer neural classifier. Similar ideas can be extended to solve the N-parity problem [15].

III Optimum Parameter Perspective

3.4 *Gaussian mixture discrimination problem*

The Gaussian Mixture classification problem, originally studied by Kohonen [9] is also discussed here. The input samples to be classified are taken from two symmetrical two-dimensional Gaussian distributions having the same mean 0 and variance 1 and 2 respectively. The difficulty of problem is the overlapping class distribution between bi-category patterns (denoting X^{p1} and X^{p2}). The training and testing set was generated independently in 50 trials, each consisting 100 samples (50 for each class), one of training sets are shown in Fig.6. It was well known that MLP is very sensitive to initial condition [11], improperly chosen weights may cause the network paralysis or saturation, the optimum weight distribution parameters are recommended to preset before the training. In this example, the only known knowledge the statistical number (mean and variance) of two-class pattern, thus we can firstly suppose the entries of matrix $\mathbf{W}$ are taken uniformly among a small interval [-θ, θ] and further derive a optimum parameter for θ. Due to the independence of the input vectors, the mean and variance (E(·) and D(·) respectively) of the whole input samples are calculated as $E(X)=E(X^{p1}+X^{p2})=E(X^{p1})+E(X^{p2})=0$ and $D(X)=D(X^{p1})+D(X^{p2})=3$. Because the summation of Gaussian distribution is also Gaussian, thus the expected value and variance of hidden output is calculated as

$$E(net_j)=E(\sum_{k=1}^{2}W_{jk}x_k)=\sum_{k=1}^{2}W_{jk}E(x_k)=0 \text{ and } D(net_j)=\sum_{k=1}^{2}W_{jk}{}^2D(x_k)=3\cdot\frac{2}{3}\cdot\theta^2=2\theta^2. \quad (11)$$

On the other hand, a very small number δ satisfying $\delta\le f(net_j)\le 1-\delta$ (say δ=0.01), is usually chosen in order to avoid the saturation region of sigmoid function, thus the activation region is calculated as [-4.6, 4.6]. In addition, according to the $3\sigma^2$-Gaussian property ($\sigma^2=2\theta^2$), it follows that $3\cdot D(net_j)\le 4.6$ and $\theta\le 4.6/3\sqrt{2}\approx 1.084$. Since the weight update ΔW_{jk} has connection to the first derivative of sigmoid function (the maximum value is 0.25 due to $f'(x)=f(x)(1-f(x))$ expression), it can be estimated the expected value of the first derivative of $f'(net_j)$. From the

statistical characteristic of net_j shown in (11), we follow that

$$E[f'(net_j)] = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty}\frac{e^{-net_j}}{(1+e^{-net_j})^2}\cdot e^{-\frac{net_j^2}{\sigma^2}}d(net_j) = \frac{1}{\sqrt{2\pi}\cdot\sqrt{2\theta^2}}\int_{-\infty}^{+\infty}\frac{e^{-t}}{(1+e^{-t})^2}\cdot e^{-\frac{t^2}{2\theta^2}}dt \ . \qquad (12)$$

The numerical estimate of (12) is about 0.1273.

Upon the above analysis and derivation, we can pre-determine the θ value to generate an optimum initial weight distribution. In experiments, the net 2-4-1 achieved average misclassification rate 0.15 with 500 iterations. It can be shown in Fig.7 that the performance is better than the other initial conditions (with different mean and variance σ^2), obviously, the estimated variance value is a good initial parameter. Likewise, the idea can be extended to the higher dimensional case of similar examples.

6 Concluding remarks

The aim of this paper is to give an insightful discussion on incorporating *a priori* knowledge into weight initialization for pattern classification. The main conclusions of this paper are led to:

1) Since the MLP is very sensitive to the initial weight condition, a proper initialization strategy seems important for improving the classifier performance (convergence and generalization). The idea behind incorporating prior knowledge into initialized weights lies in providing some hints or cues for optimization in the weight-space search process. In addition, certain weight constraints would be also helpful for finding a fast solution to specific problems.

2) How to efficiently incorporate the knowledge known *a priori*, constraints, rules, and expert information is still an open problem, albeit many relevant researches have shown promising results in improving the classifier performance, hence more attention and further studies are justified in the future. Particularly, we think further efforts should be devoted to comprehensive studies of prior knowledge (architecture, activation function, objective function, connection weights, and preprocessing strategy) influences to neural classifier and developing a formalized method for real-world classification problems.

References

[1] Anguita, D., Ridella, S., Rovetta, S., et al., 1995. Incorporating a priori knowledge into neural networks. *Electronics Letter.* 31(22), 1930-1931.

[2] Bruzzone, L., Roli, F., Serpico, S.B., 1998. Structured neural networks for signal classification. *Signal Processing.* Special Issue on Neural Networks. 64, 271-290.

[3] Chen, Z., Feng, T, Meng, Q., 1999. Geometric perspective: Finding solutions for specific pattern classification problems. Submitted to *Neural Processing Letters*

[4] Chen, Z., Feng, T. , 1999. On functional equivalence between CBP network and normalized MLP in pattern classification. Submitted to *IEEE Trans. Neural Networks.*

[5] Denoeux, T., Lengelle, R., 1993. Initializing back propagation networks with prototypes. *Neural Networks.* 6, 351-363.

[6] Drago, G., Ridella, S., 1992. Statistically controlled activation weight initialization (SCAWI). *IEEE Tran. Neural Networks.* 3, 627-631.

[7] Fahlmann, S.E. and Lebiere, C. 1989. The cascade-correlation learning architecture. *Advances in Neural Information Processing*, San Mateo, CA: Morgan Kaufmann, 2, 524-532.

[8] Karayiannis, B.N., 1996. Accelerating the training of feedforward neural networks using generalized Hebbian rule for initializing the internal representation. *IEEE Trans. Neural Networks.* 7(2), 419-425.

[9] Kohonen, T., Barna, G., Chrisley, R., 1988. Statistical pattern recognition with neural networks: benchmarking studies. *Proc. IEEE ICNN'88-San Diego.* Vol.1, 61-68.

[10] Lang, K., Witbrock, C., 1989. Learning to tell two spirals apart. *Proc. 1989 Connectionist Models Summer School.*

[11] Lee, Y., Oh, S.-H., Kim, W., 1993. An analysis of premature saturation in back propagation learning. *Neural Networks.* 6, 719-728.

[12] Lengelle, R., Denoeux, T., 1996. Training MLPs layer by layer using an objective function for internal representations. *Neural Networks.* 9(1), 83-97.

[13] Rumelhart, D.E, McClelland, J.L. and PDP groups. *Parallel Distributed Processing.* Cambridge: MIT press, 1986.

[14] Wessels, L.A., Barnard, E., 1992. Avoiding false local minimum by proper initialization of connections. *IEEE Trans. Neural Networks.* 3, 899-905.

[15] Wu, Y.S., 1996. How to choose an appropriate transfer function in designing a simplest ANN to solve specific problems. *Science in China (Series E).* 39(4), 369-374.

Table 1. Classification performance comparison of two methods

Method 1 denotes the network is trained with random weights. Method 2 denotes the network is trained with "circle weight" distribution. All the networks are stopped until they achieve SSE=10^{-4} and 100% classification accuracy or the epoch number surpasses 200,000.

H (hidden units)	epochs		convergence rate (%)	
	Method 1	Method 2	Method 1	Method 2
6	193,560	5,450	75	100
8	151,011	2,023	80	100
10	124,203	4,230	85	100
12	101,342	1,188	90	100

Table 2. Convergence result of different network architectures (initialized with "weight circle") for two-spiral problem

All the networks achieved 100% classification accuracy on training set with SSE 0.001, the numbers were averaged on 10 runs under different initialized weight seeds.

no. of hidden units	50	60	64	70	80
epoch	36,172	33,467	22,668	20,846	17,606
conv. rate (%)	90	95	100	100	100
mis. rate of test set (%)	1.76	1.56	1.04	0.78	0.56
CPU time (s)	4224.9	3027.0	2812.0	2345.0	2534.2

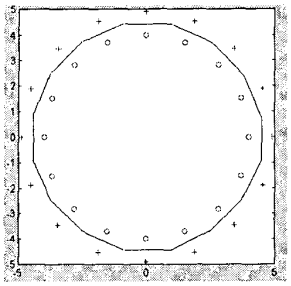

Fig. 1. The CDAF problem

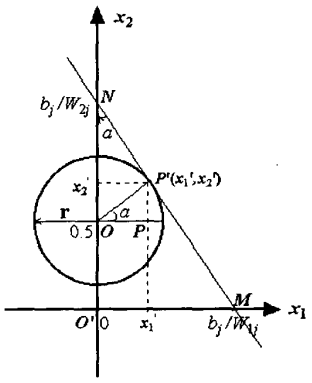

Fig. 2. The geometric proof of weight distribution

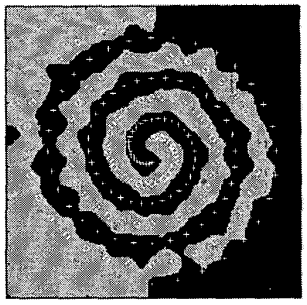

Fig. 4. Two-spiral problem solution

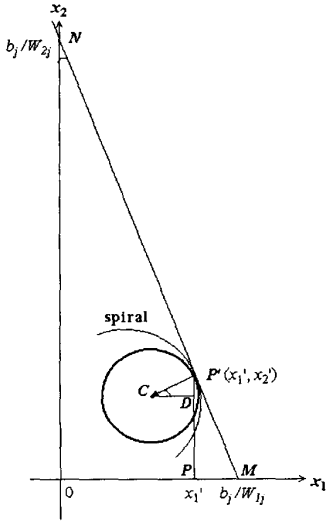

Fig. 3 Geometric proof of weight distribution of spiral problem

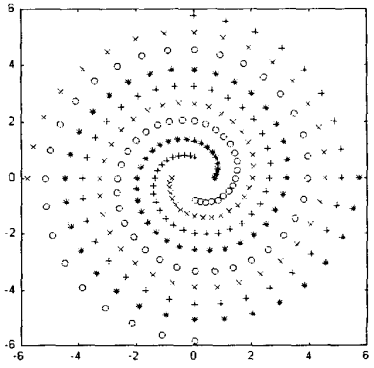

Fig. 5. Four-spiral problem

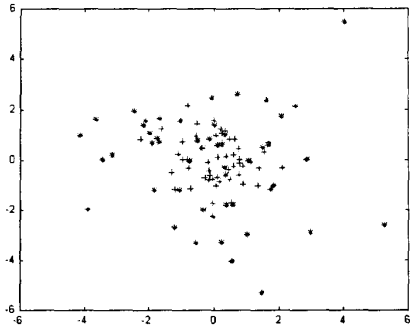

Fig. 6. Gaussian Mixture Discrimination problem

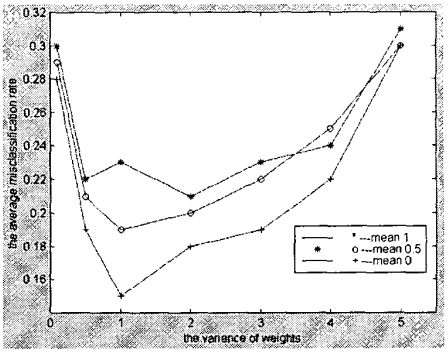

Fig. 7. The neural classifier performance under different conditions