Weight Initialization Methods for Multilayer Feedforward.

1
† Mercedes **Fernández-Redondo** - † Carlos **Hernández-Espinosa**.
† Universidad Jaume I, Campus de Riu Sec, Edificio TI, Departamento de Informática, 12080 Castellón, Spain. e-mail: redondo@inf.uji.es, espinosa@inf.uji.es

Abstract. In this paper, we present the results of an experimental comparison among seven different weight initialization methods in twelve different problems. The comparison is performed by measuring the speed of convergence, the generalization capability and the probability of successful convergence. It is not usual to find an evaluation of the three properties in the papers on weight initialization. The training algorithm was Backpropagation (BP) with a hyperbolic tangent transfer function. We found that the performance can be improved with respect to the usual initialization scheme.

1. Introduction.

It is well known that the training algorithm Backpropagation can be viewed as the optimization of the error with respect to the weights. A local optimization technique is almost always employed for training and as a consequence our training algorithm usually reaches a local minimum.

Furthermore, the particular local minimum will determine the quality of the neural network solution. On the one hand, if the minimum is close to the global one the performance will be acceptable and the training successful. On the other hand, there are minima that result in poorly trained networks and unsuccessful convergence.

The factors that determine the final local minimum are mainly the particular weight initialization and the training algorithm.

Furthermore, the weight initialization influences the speed of convergence, the probability of convergence and the generalization.

Under the point of speed of convergence, a particular initialization value can be closer or farther than another different value to the same final local minimum. So, the number of iterations of the training algorithm and the convergence time will vary depending on the weight initialization.

Considering the probability of successful convergence, it is clear from the initial discussion that a particular weight initialization value can lead the training algorithm to an acceptable local minimum or to a false local minimum. In one case, we will consider that the neural network converged successfully, and in the other that the neural network did not converge. The probability of successful convergence depends on the weight initialization scheme.

Finally, the third effect is the generalization performance of the neural network. Consider the case of two successful convergences, but two different local minima were reached. In this case, we have considered that the performance of both local minima is acceptable, but it can be different and therefore the generalization

1 Work funded by a project number GV-99-75-1-14 from Generalitat Valenciana.
ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 119-124

performance will also be different.

From the above discussion, we can conclude that the weight initialization is a very important issue. However, the usual way to initialize the weights is at random. This fact seems to be paradoxical because we leave an important topic at random.

In the bibliography, we can find several papers on weight initialization for the Multilayer Feedforward. In some of them a new weight initialization scheme is proposed.

We should also point out that in most of the papers the authors do not provide a full set of results to evaluate speed, generalization and probability of successful convergence. Results are usually focussed on one of these aspects and therefore the conclusions are rather limited.

The objective of this paper is to present an empirical comparison of weight initialization methods. We present measurements of the three aspects mentioned above.

2. Theory and Methodology.

In this section we will describe the bases and equations of the seven weight initialization techniques that will be compared. The transfer function in the neural

network was the hyperbolic tangent. So the original weight initialization methods are modified to deal with this transfer function when needed.

2.1. Method 1.

This first method is just the usual weight initialization: a uniform random initialization inside the interval [-0.05, 0.05]. This method was included to provide a reference.

2.2. Method 2.

In [1], it is proposed a minimum bound for the weight initialization, equation (1). The initialization is still random, but satisfying the minimum. In the equation, | □is the learning step used in the BP training after initialization.

In the reference it is not absolutely specified the initialization procedure because there is just a lower bound and not an upper one. So, we have finally used as upper bound 0.1 plus the lower bound for all the experiments.

2.3. Method 3.

Li et al. proposed another method [2] quite different to the usual initialization. First, the multilayer network is partitioned at the hidden layer into two simple perceptrons. After, the weights of the perceptrons are initialized with zero values and it is performed an independent pre-training of both perceptrons by using the Delta rule. For the pre-training we need to specify the inputs and targets for each of the two perceptrons; the details can be found in the reference.

We have to choose the learning step | □and the momentum ⟨□of the Delta rule and the number of iterations of this pre-training, *Ite*. We have used 8 different combinations of parameters following the recommendations in the reference. The momentum ⟨□was 0.9; the rest of pairs (*Ite*, |) were: (2.5·*N*, 2), (5·*N*, 2), (7.5·*N*, 2), (10·*N*, 2), (2.5·*N*, 4), *wi*

Ninput

<

|

(**1**)

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 119-124

(5·*N*, 4), (7.5·*N*, 4) and (10·*N*, 4). Where *N* is the number of training patterns in the problem. We will denote these combinations from “a” to “h” respectively.

2.4. Method 4.

In reference [3] it is proposed the condition of equation (2). Where w_o is the threshold and w_i the rest of weights.

The initialization procedure is not fully specified because we just have a restriction in the value of the thresholds w_o. We have finally initialized the weights w_i inside an interval of amplitude [-0.05,0.05] and after that the thresholds were the maximum w_i value.

2.5. Method 5.

Shimodaira proposed [4] another method based on geometrical considerations. The method can be summarized as the following algorithm, which determines the weights w_i connected from *n* units in the lower layer to the unit number *i*.

1) Calculate *b* by equation (3). Where *f* is the transfer function and ∑□a parameter; its appropriate value was 0.1 in the reference.

2) Using *n* and the parameter *k*, calculate with equation (4).

3) Using another parameter ©, generate a uniform random number a_i in the range of equation (5).

4) Using a_i, calculate w_i with equation (6), repeat steps 2 and 3 *n* times to calculate the *n* weights.

5) The threshold w_o is zero.

For the parameters *k* and ©, we have used several combinations recommended in the reference. The pairs (*k,* ©) were: (2, 0.3),

(5, 0.3), (8, 0.3), (2, 0.6), (5, 0.6), (8, 0.6), (2, 0.9), (5, 0.9) and (8, 0.9). We will denote these combinations from “a” to “i” respectively.

2.6. Method 6.

In the reference [5], it is proposed to assign different initial connection strengths according to the importance of the inputs. A more important input should be initialized with larger weights.

We have classified the inputs of every problem in three groups: the most important inputs were initialized in [0.5,1], the least important initialized in [0,0.5] and the rest initialized in [0,1]. The first two groups contain about one quarter of the total number inputs and the other group about one half.

2.7. Method 7.

This method is called SCAWI and it was proposed in reference [6].

The authors use the concept of “paralyzed neuron percentage” (PNP). This concept may be defined by testing how many times a neuron is in a saturated state and the

$$\frac{i}{o}$$

$$w$$

$$w <$$

(2)

$$nk$$

$$b$$

$$w$$

$$\oplus$$

$$=$$

$$\cdot 2$$

$$\hat{}$$

(4)

$$)($$

$$)$$

$$1($$

$$\frac{1}{i}$$

$$\sum$$

$$\sum$$

□

□

□□

$$=$$

$$f$$

$$f$$

$$b$$

(3)

©

©

$$\delta\delta\square_i$$

$$a$$

(5)

$$1$$

$$\hat{}$$

$$+$$

$$\oplus =$$

$$\frac{i}{i}$$

$$a$$

$$w$$

$$w$$

(6)

$$\hat{w}$$

$$\frac{ij}{input}$$

$$\frac{ij}{}$$

$$r$$

$$v$$

$$Ninput$$

$$w$$

$$\cdot$$

$$\cdot$$

$$1$$

$$3.1$$

$$\frac{}{2}$$

$$+$$

$$=$$

(7)
ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 119-124

magnitude of at least one output error is high.
They propose equations (7) and (8) for calculating the value of the weights.
In the equations $\{$ □is the mean squared value of the inputs and r_{ij} is a random number uniformly distributed in the range [-1, +1].

3. Experimental results.

The main purpose of this research was to experimentally evaluate the different initialization methods and determine their usefulness. We have applied the 7 methods to 12 different classification problems. They are from the UCI repository of machine learning databases. Their names are Balance Scale (BALANCE), Cylinders Bands (BANDS), Liver Disorders (BUPA), Credit Approval (CREDIT), Seven Leds Display (DISPLAY), Glass identification (GLASS), Heart Disease (HEART), Imagen Segmentation (IMAGEN), the Monk's Problems (MONK1, MONK2), Pima Indians Diabetes (PIMA) and Voting Records (VOTING). The complete data and a full description can be found in the UCI repository of machine learning databases.
First, we should point out that in all the above methods there is a random step in the weight initialization scheme. This means that the performance will be, in general, different for different trials. So, in our methodology we performed 30 trials with different partitions of the data and random seeds for every initialization method; the performance values are the mean.

$$w_{ij} = \frac{3.1}{Nhidden_{ij}} \cdot r_{ij} \cdot 3.01 \quad (8)$$

	BALANCE	BAND	BUPA	POR N ITER	POR N ITER	POR N ITER
Mét. 1	91.8±0.4□1	120±70	66.8±1.3□11	800±300	59.4±1.0□3	1300±400
Mét. 2	91.0±0.5□0	80±50	66.9±1.6□8	720±180	60.0±0.9□1	1300±300
Mét. 3a	-----	30	-----	64.8±1.1□19	1±0	58.6±1.4□23
Mét. 3b	-----	30	-----	64.8±1.1□19	1±0	58.6±1.4□23

1±0
Mét. 3c

30

64.8±1.1□19
1±0
58.6±1.4□23
1±0
Mét. 3d

30

64.8±1.1□19
1±0
58.6±1.4□23
1±0
Mét. 3e

30

64.8±1.1□19
1±0
58.6±1.4□23
1±0
Mét. 3f

30

64.8±1.1□19
1±0
58.6±1.4□23
1±0
Mét. 3g

30

64.8±1.1□19
1±0
58.6±1.4□23
1±0
Mét. 3h

30

64.8±1.1□19
1±0
58.6±1.4□23
1±0
Mét. 4
91.1±0.4□0 120±60 68.3±1.7□8 700±200 60.8±1.3□3 2800±600
Mét. 5a 91.2±0.5□0 140±70 65.5±1.2□5 1600±400 60.8±1.5□0 1400±400
Mét. 5b 91.4±0.5□0
80±20 67.4±1.4□8 900±300 61.9±1.3□5 1500±500
Mét. 5c 91.3±0.4□0
80±40 67.1±1.2□5 600±200 60.4±1.6□3 500±160
Mét. 5d 91.1±0.5□0 150±50 68.3±1.2□4 1200±300 61.0±1.3□2 1700±500
Mét. 5e 91.0±0.4□0
80±50 68.5±1.4□7 1000±300 62.1±1.4□2 1700±400
Mét. 5f 91.7±0.4□0 200±90 66.4±1.1□6 1100±300 61.4±1.7□3 1100±300
Mét. 5g 91.5±0.5□0 100±50 67.0±1.2□6 470±160 62.4±1.1□1 1300±300
Mét. 5h 91.5±0.4□0
90±30 67.3±1.1□7 680±180 63.1±1.4□1 1500±400
Mét. 5i 91.3±0.4□0 120±70
59±5
6 1200±200 60.8±1.4□3 1400±300
Mét. 6
91.4±0.5□0 270±110 65.3±1.3□5 800±200 61.3±1.3□4 2100±500
Mét. 7
90.6±0.4□1 130±70 67.7±1.2□5 600±200 60.9±0.9□4 1600±400

Table 1. Performance of the different initialization methods

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 119-124

We evaluated the three main effects commented in the introduction: speed of convergence, generalization performance and ratio of successfully convergence. The speed of convergence was measured by the epoch of convergence. This epoch is determined by cross-validation. The second issue, generalization performance, was measured by calculating the percentage correct (PC) with the test set. Finally, the probability of successful convergence was measured by the number of networks which did not converge with respect to the total 30 trials.

Part of the results are in table 1, we do not present the full results by the lack of space. In each row of the table, we have included a different initialization method.

In the columns, we can find the databases and for every database three columns. The first and second one (headers “POR” and “N”) are the mean values of the percentage correct and the number of networks that did not converge. The third column (“ITER”) contains the epoch of convergence.

Now, we will compare the results of the different methods with respect to the reference of method 1.

The first surprising results was that method 3 did not work well. The number of networks that did not converge was unacceptable. The reason might be the selection of the parameters, but we use the range recommended in the reference. We think that even thought the reason might be in the parameter values, a method with three difficult to tune parameters is not so useful.

GLASS
HEART
IMAGEN
POR
N
ITER
POR
N
ITER
POR
N
ITER
Mét. 1
90.4±1.1□3 111±12 81.2±1.1□4
220±80
72±2
5
710±130
Mét. 2
88.9±1.1□2 150±40 81.7±1.1□3
320±140
70±2
4 1010±150
Mét. 3a

30

75.1±1.5□27 700±700

30

Mét. 3b

30

56±3
28
1±0

30

Mét. 3c

30

78±4
26
9±8

30

Mét. 3d

30

72±3
25
1±0

30

Mét. 3e

30

65±5
24
1±0

30

Mét. 3f

30

64±8
28
1±0

30

Mét. 3g

30

78.4±1.9□26
1±0

30

Mét. 3h

30

70±3
26
1±0

30

Mét. 4
90.7±0.9□2 160±40 80.6±0.8□4
430±160 74.7±1.9□5
950±170
Mét. 5a 90.2±1.4□2 190±40 81.0±0.9□3
430±160
70±2
9 1350±180
Mét. 5b 90.2±1.2□3 121±14 81.3±0.8□2
700±200
76±2
10 1150±180
Mét. 5c
89.1±1.2□2 130±20 80.3±0.8□3
500±200
69±3
9 1060±170
Mét. 5d 91.9±0.8□5 220±40 81.1±0.8□2
630±180
67±2
7 1030±180
Mét. 5e

90.8±0.9□5
91±11
79.8±1.4□3
600±200
73±2
10 1420±140
Mét. 5f
87.6±0.9□2 200±60 80.7±1.1□3
390±180
75±2
13 970±190
Mét. 5g
90±0.9
2 200±60 80.0±0.9□2
600±200
75±2
7 1100±180
Mét. 5h 88.2±1.6□5 160±40 81.5±0.9□2
400±140 73.7±1.7□13 1300±200
Mét. 5i
89.7±1.0□2 139±19 81.7±1.1□5
290±130
72±2
10 1100±180
Mét. 6
85.7±1.1□7 300±100 80.9±0.8□3
500±200
59±3
11 1040±180
Mét. 7
88.2±1.1□5 230±60 80.8±1.0□4
200±100
70±2
8 1090±160

Table 1 (Continuation). Performance of the different initialization methods

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 119-124

Method 2 got a generalization performance slightly better than method 1. The number of converged network is also greater and the speed is sometimes better and sometimes worse; there is not a clear result.

Method 4 obtained a generalization performance greater than method 1 and results of probability of convergence are also better. The speed of convergence is similar or sighly lower than method 1.

The generalization performance of method 5 is clearly better than method 1. The probability of convergence can be considered similar; it is sometimes lower and sometimes better. However the speed of convergence is lower. We can get an improvement in the generalization capability on behalf of a lower speed.

Method 6 has a generalization performance lower than method 1; the probability of convergence is also slightly lower and finally, the speed is clearly worse. We can conclude that this method is worse than method 1 according to our results.

4. Conclusions.

We have presented the results of an experimental comparison among seven weight initialization methods. The experiments were performed with twelve different databases from the UCI repository. In our comparison we measured the speed, generalization and the probability of convergence for every initialization method. This is not usual in the bibliography on weight initialization methods.

The best weight initialization scheme for the BP algorithm was *method 5* but it has the disadvantage that we should determine several parameters by a trial and error procedure. However, the method was not very sensible to the parameters. Anyway, we can also use *method 4*, which also obtained better results than the usual initialization.

References.

1. Kim, Y.K., Ra, J.B.. Weight Value Initialization for Improving Training Speed in the Backpropagation Network. Proc. of Int. Joint Conf. on Neural Networks, vol. 3, pp. 2396-2401, 1991.

2. Li, G., Alnuweiri, H., Wu, Y.. Acceleration of Backpropagations through Initial Weight Pre-Training with Delta Rule. Proc. of the IEEE Int. Conference on Neural Networks, ICNN'93, vol. 1, pp. 580-585, 1993.
3. Palubinskas, G. Data-driven Weight Initialization of Back-propagation for Pattern Recognition. Pro. of the Int. Conf. on Artificial Neural Networks, vol. 2. pp. 851-854, 1994.
4. Shimodaira, H.. A Weight Value Initialization Method for Improved Learning Performance of the Back Propagation Algorithm in Neural. Networks. Proc. of the 6th International Conference on Tools with Artificial Intelligence, pp. 672-675, 1994.
5. Ho-Sub Yoon, Chang-Seok Bae, Byung-Woo Min. Neural networks using modified initial connection strengths by the importance of feature elements. Int. Joint Conf. on Systems, Man and Cybernetics, vol. 1, pp. 458-461, 1995.
6. Drago, G.P., Ridella, S. Statistically Controlled Activation Weight Initialization (SCAWI). IEEE Transactions. on Neural Networks, vol. 3, no. 4, pp. 627-631, 1992