GDAŃSK UNIVERSITY OF TECHNOLOGY FACULTY OF ETI ANNALS
No 8 Information Technologies 2010

**Bogdan Maciej Wilamowski**

**Auburn University, USA**
**University of Information Technology and Management, Rzeszow**

# EFFICIENT NEURAL NETWORK ARCHITECTURES AND ADVANCED TRAINING ALGORITHMS

**Abstract**

Advantages and disadvantages of various neural architectures are compared. It is shown that neural networks with connections across layers are significantly more powerful than popular MLP - Multi Layer Preceptron architectures. The most powerful are FCC Fully Connected Cascade (FCC) architectures. Unfortunately, most advanced training algorithms were developed only for popular MLP topologies and other much more powerful topologies are seldom used. The newly developed second order algorithm NBN is not only very fast and powerful, but it can train any neural network topologies. With the NBN algorithm it is possible to train close to optimal architectures which were not possible to train before.

## 1. INTRODUCTION

The error back propagation (EBP) algorithm [1, 2] can be regarded as one of the most significant breakthroughs in neural network training. The EBP algorithm is widely used today; however, it is also known as an inefficient algorithm because of its slow convergence. Many improvements [3–5] have been made to overcome the disadvantages of the EBP algorithm and some of them, such as momentum [6] and RPROP algorithm [7], work relatively well. But as long as the first order algorithms are used, improvements are not dramatic. Second order algorithms [8, 9] such as the Newton algorithm, Levenberg Marquardt (LM) algorithm [10, 11], or the Neuron by Neuron (NBN)) algorithm [12, 13] using the Hessian matrix perform better estimations on both step sizes and directions, and they can converge much faster than first order algorithms. By combining the training speed of the Newton algorithm and the stability of steepest descent method, the LM algorithm [10, 11] is regarded as one of the most efficient algorithms for neural networks training;, but it has several limitations. It cannot train other than MLP topologies and since the size of a Jacobian matrix is proportional to the number of patterns it can be used only for relatively small problems because it requires complex computation of Jacobian/Hessian matrixes and several matrix inversions at each iteration step. Also, second order algorithms use the backpropagation process to compute Jacobian matrixes [12–14]. In this case, the backpropagation is repeated for every pattern and every network output.

Besides training algorithms, neural network topology also affects training efficiency. Compared with multilayer perceptron (MLP) networks, other networks, such as fully connected cascade (FCC) networks and bridged MLP (BMLP) networks with connections across layers, are found to be more efficient in solving problems [12]. Neural networks with connections across layers are more powerful, but they require more challenging computation processes.

## 2. WHY SECOND ORDER ALGORITHMS ARE IMPORTANT

It is well-known that second order algorithms are much faster, but it is not common knowledge that networks should be as simple as possible to properly handle new patterns (not used for training). With first order algorithms, such as the EBP algorithm, it is often not possible to train those simple (close to optimal) networks. With excessive number of neurons, even first order algorithms can train the neural networks to very small errors, but this "success" is misleading because such networks will not be able to properly process new patterns, which were not used for training. This issue will be illustrated in this section with a couple of examples.

The two-spiral problem is considered as a good evaluation of training algorithms [15]. Depending on a neural network architecture, different numbers of neurons are required for successful training. For example, using standard MLP networks with one hidden layer, 34 neurons are required for the two-spiral problem [16], while with an FCC architecture it can be solved with only 8 neurons. Second order algorithms are not only much faster but they can train reduced size networks which cannot be handled by the EBP algorithm (see Tables 2.1 and 3.1). Another important fact is that it is much easier to train networks with an excessive number of neurons.

**Table 2.1**

Training results of two-spiral problem

| Neurons | Success Rate | | Average Iteration | | Average Time (s) | |
|---|---|---|---|---|---|---|
| | EBP | NBN | EBP | NBN | EBP | NBN |
| 8 | 0% | 13% | / | 287.7 | / | 0.88 |
| 9 | 0% | 24% | / | 261.4 | / | 0.98 |
| 10 | 0% | 40% | / | 243.9 | / | 1.57 |
| 11 | 0% | 69% | / | 231.8 | / | 1.62 |
| 12 | 63% | 80% | 410,254 | 175.1 | 633.91 | 1.70 |
| 13 | 85% | 89% | 335,531 | 159.7 | 620.30 | 2.09 |
| 14 | 92% | 92% | 266,237 | 137.3 | 605.32 | 2.40 |
| 15 | 96% | 96% | 216,064 | 127.7 | 601.08 | 2.89 |
| 16 | 98% | 99% | 194,041 | 112.0 | 585.74 | 3.82 |

Table 2.1 presents the training results of the two-spiral problem using different number of neurons in FCC networks. Results presented in Tables 2.1 and 3.1 were obtained with the following parameters. For the EBP algorithm, the learning constant is 0.005 (largest to avoiding oscillation) and the momentum is 0.5; the maximum iteration is 1,000,000 for the EBP algorithm and 1,000 for the NBN algorithm; the desired error = 0.01; all neurons are

in FCC networks; there are 100 trials for each case. The NBN algorithm can solve the two-spiral problem using 8 neurons (52 weights) in nearly 290 iterations (Fig. 1a). The EBP algorithm can solve the two-spiral problem only when larger networks are used. When the number of neurons is increased to 12 (102 weights), the EBP algorithm can solve it in about 400,000 iterations. The result (the best one in 100 trials), shown in Fig. 1(b), is not as good as the result (Fig. 1a) from the NBN algorithm with a much simpler architecture. One can conclude that the EBP algorithm is only successful if excessive number of neurons is used.

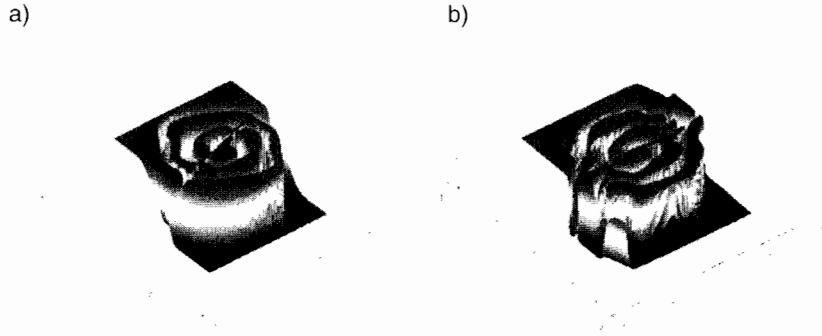


Fig. 1. Best results of two-spiral problem in 100 trails: a) 8 neurons in FCC network (52 weights), using NBN algorithm and training time = 0.82 s; b) 12 neurons in FCC network (102 weights), using EBP algorithm and training time = 694.32 s

## 3. GENERALIZATION ABILITIES

It is relatively easy to find neural network architectures so that they can be trained to very small errors. However, it is more important to find an architecture which after training will respond correctly to patterns which were not used for training. Let us illustrate this problem using another example. Let us consider a peak surface [13] shown in Fig. 2a as the required surface and let us use equally spaced 10×10 = 100 patterns (Fig. 2b) in training neural networks. The quality of trained networks is evaluated using errors computed for equally spaced 37×37 = 1,369 patterns. In order to make a valid comparison between the training and verification error, the SSE, as defined in (1), is divided by 100 and 1,369, respectively.

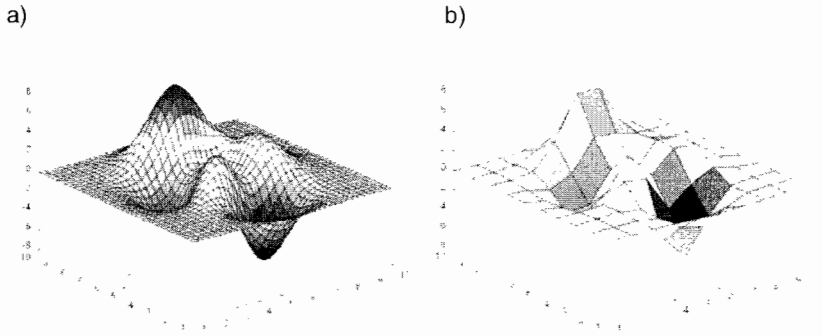


Fig. 2 Surface matching problem: a) Required 2-D surface with 37×37 = 1,369 points, used for verification; b) 10×10 = 100 training patterns extracted in equal space from (a), used for training

**Table 3.1**

Training Results of peak surface problem

| Neurons | Success Rate | | Average Iteration | | Average Time (s) | |
|---|---|---|---|---|---|---|
| | EBP | NBN | EBP | NBN | EBP | NBN |
| 8 | 0% | 5% | / | 222.5 | / | 0.33 |
| 9 | 0% | 25% | / | 214.6 | / | 0.58 |
| 10 | 0% | 61% | / | 183.5 | / | 0.70 |
| 11 | 0% | 76% | / | 177.2 | / | 0.93 |
| 12 | 0% | 90% | / | 149.5 | / | 1.08 |
| 13 | 35% | 96% | 573,226 | 142.5 | 624.88 | 1.35 |
| 14 | 42% | 99% | 544,734 | 134.5 | 651.66 | 1.76 |
| 15 | 56% | 100% | 627,224 | 119.3 | 891.90 | 1.85 |

The training results shown in Table 3.1, point out that using the NBN algorithm, which can handle arbitrarily connected neural networks, then, it was possible to find the acceptable solution (Fig. 3a), $SSE_{Train} = 0.0044$ and $SSE_{Verify} = 0.0080$, with 8 neurons (52 weights). Unfortunately, with the EBP algorithm, it was not possible to find acceptable solutions in 100 trials within 1,000,000 iterations each. The best result out of the 100 trials with 1,000,000 iterations each was $SSE_{Train} = 0.0764$ and $SSE_{Verify} = 0.1271$. When the network size was significantly increased from 8 to 13 neurons (117 weights), the EBP algorithm was able to reach the similar training error as with the NBN algorithm, but the network lost its ability to respond correctly to new patterns (between training points). Please notice that indeed with an enlarged number of neurons, the EBP algorithm was able to train the network to the small error $SSE_{Train} = 0.0018$, but as one can see from Fig. 3b, the result is unacceptable with the verification error $SSE_{Verify} = 0.4909$.

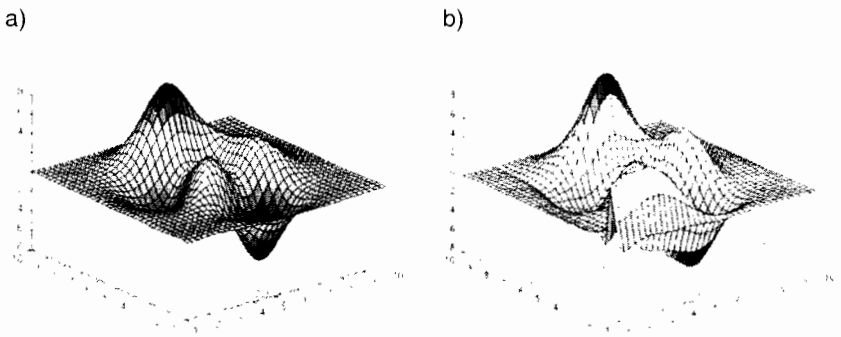


Fig. 3 Training results using 100 trials with: a) NBN algorithm, 8 neurons in FCC network (52 weights); maximum training iteration is 1,000; $SSE_{Train} = 0.0044$, $SSE_{Verify} = 0.0080$ and training time = 0.37 s; b) EBP algorithm, 13 neurons in FCC network (117 weights); maximum training iteration is 1,000,000; $SSE_{Train} = 0.0018$, $SSE_{Verify} = 0.4909$ and training time = 635.72 s

From the above analysis, one may notice that in order to sustain neural network generalization abilities the network should have as few neurons/weights as possible. This problem is very similar to function approximation by polynomials. If too high order of polyno-

mial is used then errors for training points but values between points cannot be evaluated correctly. In the example in Fig. 4 only 5-th, 6-th, and 7-th order of polynomials are giving adequate results, while higher order polynomials can be tuned to smaller errors for given points. They are useless to predict evaluated new points which were not used for training.

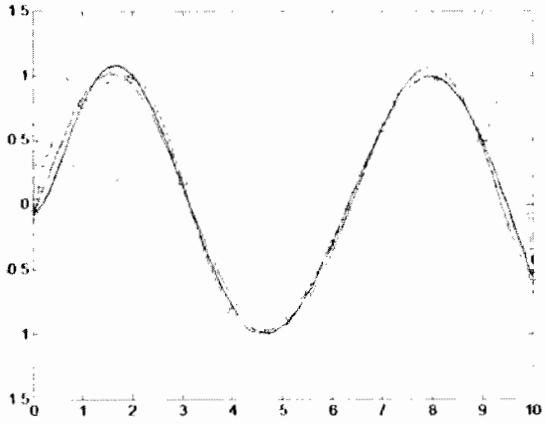


Fig. 4. Polynomial approximation to data points

When a reduced number of neurons are used the EBP algorithm cannot converge to the required training error. When the size of networks increase, the EBP algorithm can reach the required training error, but trained networks lose their generalization ability and cannot process new patterns well (Fig. 3b). On the other hand, second order algorithms, such as the NBN algorithm, work not only significantly faster but they can find good solutions with close to optimal networks (Fig. 3a).

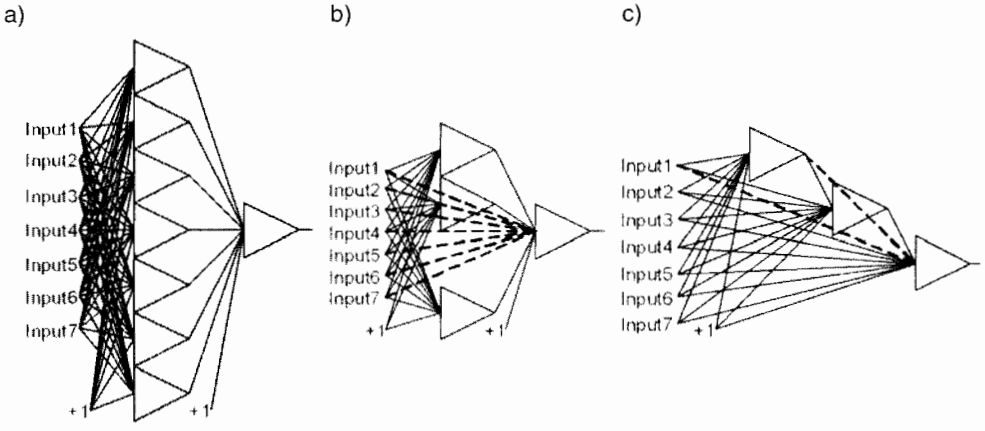


Fig. 5. Smallest neural networks required for Parity-7 problem.
a) MLP – Multi Layer Perceptron with one hidden layer: b) BMLP – Bridged Multi Layer Perceptron with one hidden layer: c) FCC – Fully Connected Cascade network

## 4. POWERFUL NEURAL NETWORK ARCHITECTURES

Parity-N problems are commonly used benchmarks for neural networks and they can be used to compare the power of different neural network architectures. Fig. 5 shows various neural network topologies, which can be used to solve the Parity-7 problem [17]. Fig. 6 shows abilities of different network topologies for solution of a Parity-N problem using a different number of neurons

For example, with 7 neurons in the MLP topology with one hidden layer the largest problem which can be solved is the Pairty-6 (**6=6=1**). With the same 7 neurons using the BMPL and also one hidden layer, it is possible to solve the Parity-13 problem using the (**13=6=1**) topology. With the BMLP with two hidden layers (**31=3=3=1**) it is possible to solve the Parity-31 problem. With the BMLP with three hidden layers (**53=2=2=2=1**) it is possible to solve the Parity-53 problem. When the FCC topology is used, then with 7 neurons it is possible to solve as large a problem as the Parity-127 one using the (**127=1=1=1=1=1=1=1**) topology. One may withdraw the conclusion that with the BMLP architectures the capabilities of neural networks rapidly increase with the depth of the network and that the FCC architecture is the most powerful. Unfortunately, most researchers are using MLP networks with one hidden layer which is the least powerful architecture. One reason why more powerful architectures are not used is that the very powerful LM learning algorithm was developed for MLP architectures.

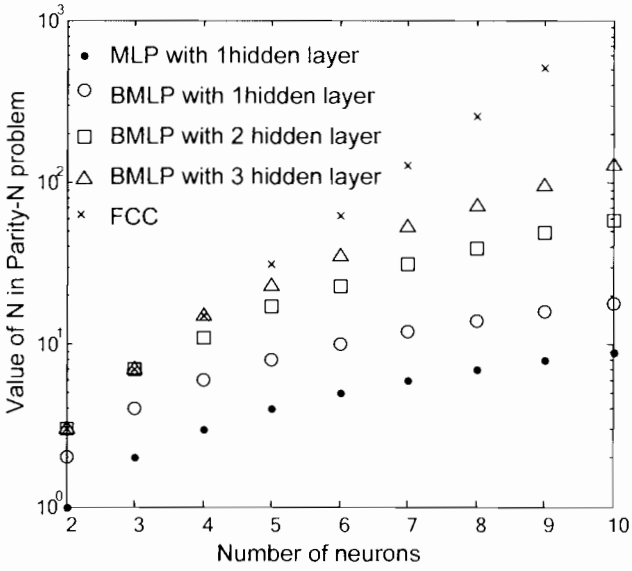


Fig. 6. Abilities of solving Parity-N problems as function of number of neurons

## 5. TRAINING ALGORITHMS

As one can see from the example shown in Section 2 the EBP algorithm cannot converge to the required training error unless a significant number of excessive neurons are used. When the size of networks increase, the EBP algorithm can reach the required training error, but trained networks lose their generalization ability and cannot process new

patterns well (Fig. 3b). The newly developed NBN algorithm works not only significantly faster than the EBP one (or even faster than the LM algorithm) but it can find good solutions with close to optimal networks which are very difficult to train (Fig. 3a).

The NBN algorithm eliminates most deficiencies of the LM algorithm and it can be used to train neural networks with arbitrarily connected neurons (not just the MLP architecture). It does not require it to compute and to store large Jacobians so it can train problems with basically an unlimited number of patterns [13]. Error derivatives are computed only in the forward pass, so the backward computation process is not needed. It is equally fast, but in the case of networks with multiple outputs faster than the LM algorithms, it can train feed forward networks which are impossible to train with other algorithms.

## 6. CONCLUSIONS

It is much easier to train neural networks where the number of neurons is larger than the required one. However, with a smaller number of neurons the neural network has much better generalization abilities. This means it will respond correctly to patterns not used for training. If too many neurons are used, then the network can be over-trained on the training patterns, but it will fail on patterns never used in training. With a smaller number of neurons, the network cannot be trained to very small errors, but it may produce much better approximations for new patterns. The most common mistake made by many researchers is that in order to speed up the training process and to reduce the training errors they use neural networks with larger number of neurons than required. Such networks would perform very poorly for new patterns not used for training [18].

## BIBLIOGRAPHY

[1] Rumelhart D. E., Hinton G. E., Williams R. J.: *Learning representations by back-propagating errors*. Nature, vol. 323, pp. 533–536, 1986.

[2] Werbos P. J.: *Back-propagation: Past and Future*. Proceeding of International Conference on Neural Networks, San Diego, CA, 1, 343-354, 1988.

[3] Ferrari S., Jensenius M.: *A Constrained Optimization Approach to Preserving Prior Knowledge During Incremental Training*, IEEE Trans. on Neural Networks, vol. 19, no. 6, pp. 996-1009, June 2008.

[4] Qing Song, J.C. Spall, Yeng Chai Soh, Jie Ni: *Robust Neural Network Tracking Controller Using Simultaneous Perturbation Stochastic Approximation*, IEEE Trans. on Neural Networks, vol. 19, no. 5, pp. 817–835, May 2008.

[5] Yinyin Liu, Starzyk J. A., Zhen Zhu: *Optimized Approximation Algorithm in Neural Networks Without Overfitting*, IEEE Trans. on Neural Networks, vol. 19, no. 6, pp. 983–995, June 2008.

[6] Phansalkar V. V., Sastry P. S.: *Analysis of the back-propagation algorithm with momentum*, IEEE Trans. on Neural Networks, vol. 5, no. 3, pp. 505–506, March 1994.

[7] Riedmiller M., Braun H.: *A direct adaptive method for faster backpropagation learning: The RPROP algorithm*. Proc. International Conference on Neural Networks, San Francisco, CA, 1993, pp. 586–591.

[8] Cheol-Taek Kim, Ju-Jang Lee: *Training Two-Layered Feedforward Networks With Variable Projection Method*, IEEE Trans. on Neural Networks, vol. 19, no. 2, pp. 371–375, Feb 2008.

[9] Ampazis N., Perantonis S. J.: *Two highly efficient second-order algorithms for training feedforward networks*, IEEE Trans. on Neural Networks, vol. 13, no. 5, pp. 1064–1074, May 2002.

[10] Wu, J.-M.: *Multilayer Potts Perceptrons with Levenberg–Marquardt Learning*. IEEE Trans. on Neural Networks, vol. 19, no. 12, pp. 2032–2043, Feb 2008.

[11] Toledo A., Pinzolas M., Ibarrola J. J., Lera G.: *Improvement of the neighborhood based Levenberg-Marquardt algorithm by local adaptation of the learning coefficient*, IEEE Trans. on Neural Networks, vol. 16, no. 4, pp. 988–992, April 2005.
[12] Wilamowski B.M., Cotton N. J., Kaynak O., Dundar G.: *Computing Gradient Vector and Jacobian Matrix in Arbitrarily Connected Neural Networks*, IEEE Trans. on Industrial Electronics, vol. 55, no. 10, pp. 3784–3790, Oct. 2008.
[13] Wilamowski B. M., Yu H.: *Improved Computation in Levenberg Marquardt Training*, IEEE Trans. on Neural Networks (available as preprint).
[14] Hagan M. T., Menhaj M. B.: *Training feedforward networks with the Marquardt algorithm*, IEEE Trans. on Neural Networks, vol. 5, no. 6, pp. 989–993, Nov. 1994.
[15] Sheng Wan, Banta L. E.: *Parameter Incremental Learning Algorithm for Neural Networks*, IEEE Trans. on Neural Networks, vol. 17, no. 6, pp. 1424–1438, June 2006.
[16] Jian-Xun Peng, Kang Li, Irwin G. W.: *A New Jacobian Matrix for Optimal Learning of Single-Layer Neural Networks*, IEEE Trans. on Neural Networks, vol. 19, no. 1, pp. 119–129, Jan 2008.
[17] Wilamowski B. M., Hunter D., Malinowski A.: *Solving Parity-n Problems with Feedforward Neural Network*, Proc. of the IJCNN'03 International Joint Conference on Neural Networks, pp. 2546–2551, Portland, Oregon, July 20–23, 2003.
[18] Wilamowski B. M.: *Neural Network Architectures and Learning Algorithms: How Not to Be Frustrated with Neural Networks*, IEEE Industrial Electronics Magazine, vol. 3, no. 4, pp. 56–63, Dec. 2009.

## EFEKTYWNE ARCHITEKTURY SIECI NEURONOWYCH I ZAAWANSOWANE ALGORYTMY UCZENIA

**Streszczenie**

W pracy porównano zalety i wady rożnych topologii sieci neuronowych. Pokazano ze sieci neuronowe z połączeniami poprzez warstwy są znacznie bardziej efektywne niż popularne topologie MLP. Najbardziej efektywne są topologie FCC. Niestety większość zaawansowanych algorytmów uczenia zostało zaimplementowanych tylko dla popularnych topologii MLP i inne bardziej efektywne topologie są rzadko używane. Niedawno opracowany drugiego rzędu algorytm jest nie tylko bardzo szybki i efektywny, ale również umożliwia uczenie dowolnych topologii sieci neuronowych. NBN potrafi uczyć zbliżone do optymalnych architektury sieci neuronowych, których nie można było uczyć poprzednio.

GDANSK UNIVERSITY OF TECHNOLOGY
FACULTY OF ETI ANNALS

# INFORMATION TECHNOLOGIES

# VOLUME 18

Gdańsk 2010

# EDITORIAL COMMITTEE



Under the auspices of
DEAN OF ETI FACULTY GDANSK UNIVERSITY OF TECHNOLOGY

Each paper of this Scientific Annals
have been peer positively reviewed by three independent reviewers



ISBN 978-83-60779-02-6

Painting and binding: *EXPOL P. Rybiński, J. Dąbek, Sp. Jawna*
ul. Brzeska 4, 87-800 Włocławek, tel.: 54 232 37 23