PPNN: A Faster Learning and Better Generalizing Neural Net

Bo Xu[1]
Department of Physiology and Biophysics
School of Medicine, Indiana University
3656 Green Ash Ct #205
Indianapolis, IN 46222
USA

Liqing Zheng
Department of Electrical Engineering
Purdue University, Indianapolis

Abstract----It was pointed out in this paper that the planar topology of current back-propagation neural network (BPNN) sets limits to solve the slow convergence rate problem, local minima, and other problems associated with BPNN. The parallel probabilistic neural network (PPNN) using a new neural network topology, stereotopology, was proposed to overcome these problems. The learning ability and the generalization ability of BPNN and PPNN were compared for several problems. The simulation results show that PPNN was capable of learning any kinds of problems much faster than BPNN and generalized better than BPNN too. It was analyzed that the faster, universal learnability of PPNN was due to the *parallel* characteristic of PPNN's stereotopology, and the better generalization ability came from the *probabilistic* characteristic of PPNN's memory retrieval rule.

1. INTRODUCTION

In the present neural network area, the back-propagation neural network (BPNN) [1] is perhaps the most widely used neural net. However, BPNN has its own drawbacks: (1) slow rate of convergence problem, (2) local minima problem, and (3) incapability of learning problem (For some very difficult tasks, the global minimum of BPNN is so high that it in fact cannot converge to a preset convergence criterion. So these tasks actually cannot be learned by BPNN at any accuracy we designed beforehand.) These greatly hamper the practical application of BPNN.

Many attempts have been made to increase BPNN's rate of convergence[2-14] and to overcome the local minima problem [15-19]. These studies have shown encouraging improvements to the original BPNN.

The architecture of neural network consists of three parts [20]: (1) learning algorithms of weights; (2) characteristics (activation functions) of nodes; and (3) topology of the network. Most of the above attempts fall into the first two aspects: the learning algorithms and the characteristics of nodes. Although there are many improvements in increasing the rate of convergence and overcoming the local minima problems through the study of learning algorithms and/or the node characteristics, we pointed out in this paper that it is the third aspect, the topology of BPNN (a planar topology), that sets limits for the improvements of BPNN's performances.

This paper is trying to overcome the above drawbacks of BPNN by proposing a neural network called *parallel probabilistic neural network* (PPNN) with a new topology----*stereotopology*. It was shown that PPNN with the stereotopology was superior to BPNN with planar topology in both learning and generalization abilities.

2. PPNN

---

[1] We thank Drs. R. Rhoades, L. Knoebel, D. Peavy, P. G. Madhavan, D. Suzuki, H. Oner Yurtseven, and A. Sinha for their supports, thank Dr. S. Ochs for enlightenment discussion and encouragements, and thank Dr. L. Abels for providing reference materials.

2.1. PPNN Topology

In order to see the features of PPNN s new topology, let s first look at BPNN s topology (Figure 1). There is only one branch of hidden layer(s) connecting with the input and the output layers in BPNN. The input layer, the output layer, and all the hidden layers can lie in one plane. So BPNN has in fact a *planar* topology.

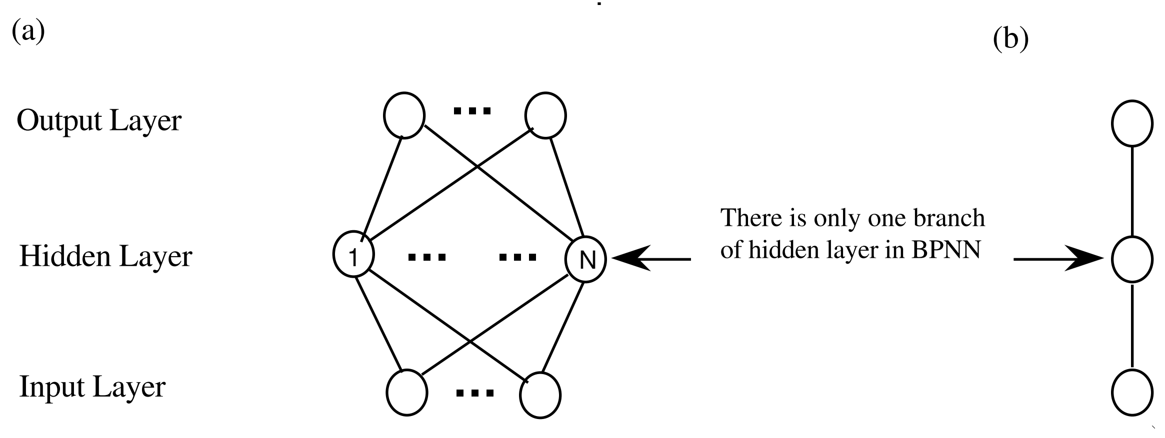

Figure 1. The planar topology of BPNN. (a). Front view of the BPNN planar topology. N is the number of hidden nodes. (b). Side view of the BPNN planar topology.

The key feature of PPNN is that there are multi-branches of parallel hidden layers in it. The topology of PPNN is shown in Figure 2. PPNN also has an input layer and an output layer. However, unlike the BPNN which has only one branch of hidden layer, the PPNN has P (the number of training patterns) parallel branches of hidden layers. There is no connection between these parallel branches of hidden layers. Each of the P parallel branches of hidden layers connects directly to the input layer and the output layer in the same way as the hidden layer in BPNN. The input layer, output layer, and all the P parallel branches of hidden layers cannot lie in one plane. Therefore, PPNN has a *stereotopology*.

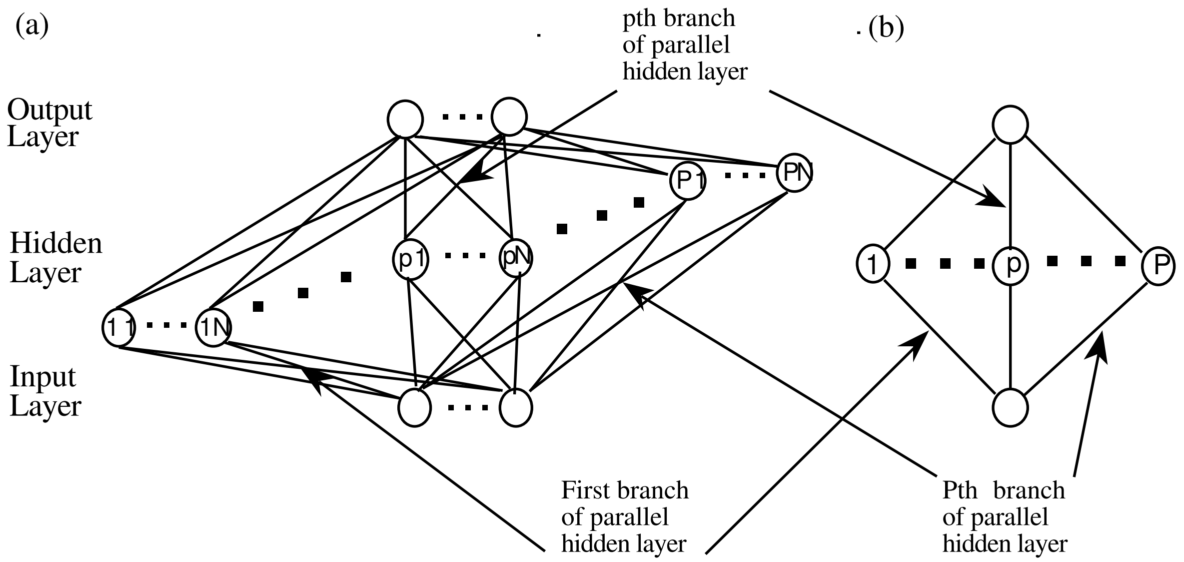

Figure 2. The stereotopology of PPNN. (a) Front view of the PPNN stereotopology. (b) Side view of the PPNN stereotopology.

2.2. PPNN Training Process

There is a one-to-one correspondence between the P parallel branches of hidden layers and P training patterns. Each of the P parallel branches of hidden layers can be in either *on* or *off* state. The on-state is the state in which the hidden layer is connected with the input layer and the output layer. This is the working state in which this branch of hidden layer takes part in the training or testing processes of the network. The off-state is the state in which the hidden layer is disconnected with both the input layer and the output layer. This is the rest state in which this branch of hidden layer does not take part in the training or testing processes of PPNN.

During training, only one of the P parallel branches of hidden layers is in the on-state. All the other branches of hidden layers are in the off-state. If the training pattern p (p=1,2,3, ... , P) is used to train the PPNN, then the pth branch of hidden layer should be on, and all the other branches of hidden layers should be off.

2.3. PPNN Test Process

We proposed a probabilistic memory retrieval rule (PMRR) for the PPNN test process. The PMRR consists of two steps:

Step 1. Closeness Selection. First, the test pattern is compared with each of the P training patterns to find out the training pattern p which is closest to the test pattern.

Step 2. Probabilistic memory retrieval. Once the training pattern p is determined, it is needed to check whether this p is close enough to the test pattern. To do so, a closeness index ε is preset (ε can be any small positive number). The ε is used as a criterion to check the closeness between the training pattern p and the test pattern. Based on the closeness of p to the test pattern, the memory information retrieval processes are as follows: (i) If the training pattern p is not close enough to the test pattern at ε, we propose that the the net test output should be the weighted mean over the individual test outputs of each of the P parallel branches of hidden layers. The weighted mean s coefficients were proportional to the inverse square of the mean absolute difference between the test pattern and the input pattern p ; (ii) If the training pattern p is close enough to the test pattern at ε, the net test output is simply the p branch s test output.

From the probability viewpoint, above steps depend simply on the probability that the pth parallel branch of hidden layer will contribute to the net test output. This is the reason we call this kind of memory recall method probabilistic memory retrieval rule (PMRR).

3. SIMULATIONS

In order to contrast the differences between the BPNN using planar topology and the PPNN using the stereotopology, we compared PPNN with BPNN under the conditions that all the other structures and parameters are the same for both of them except the topology. This will show us the effects caused by the topological difference only.

We compared PPNN with BPNN using several problems to test their learnability and generalization ability. We choose XOR problem, parity problem [1, 11], and the binary-to-local problem [5] to study the convergence rates and learnability. We used parabolic problem to study both the learnability and generalization ability. Parabolic problem is a task of mapping parabola function $y=x^2$. This is not a toy problem, but a very difficult real problem [21] because it is non-monotonous (it decreases when $x < 0$ and increases when $x > 0$), it is symmetric and it has a multi-to-one mapping (both -x and x map the same y).

All the simulations were written in C and were run on VAX 8800. The learning ability results were shown in Table 1, and the generalization ability results in Figure 3.

Table 1. Comparison of Training Times between BPNN and PPNN

PROBLEMS		XOR	PARITY	BINARY-TO-LOCAL	PARABOLIC
BPNN	MAE [a]	0.01	0.5	0.136	10.15
	Training Times [b]	40465	100000	100000	100000
PPNN	MAE [a]	0.01	0.01	0.01	0.01
	Training Times [b]	7	9	70	13

a. The MAE listed in BPNN for parity, binary-to-local and parabolic problems were the actual MAE values when the maximum training times limit 100,000 was reached.

b. The training times for BPNN were simply the epochs. The training times shown here for PPNN were the average training times over all these different branches of hidden layers for each problem.

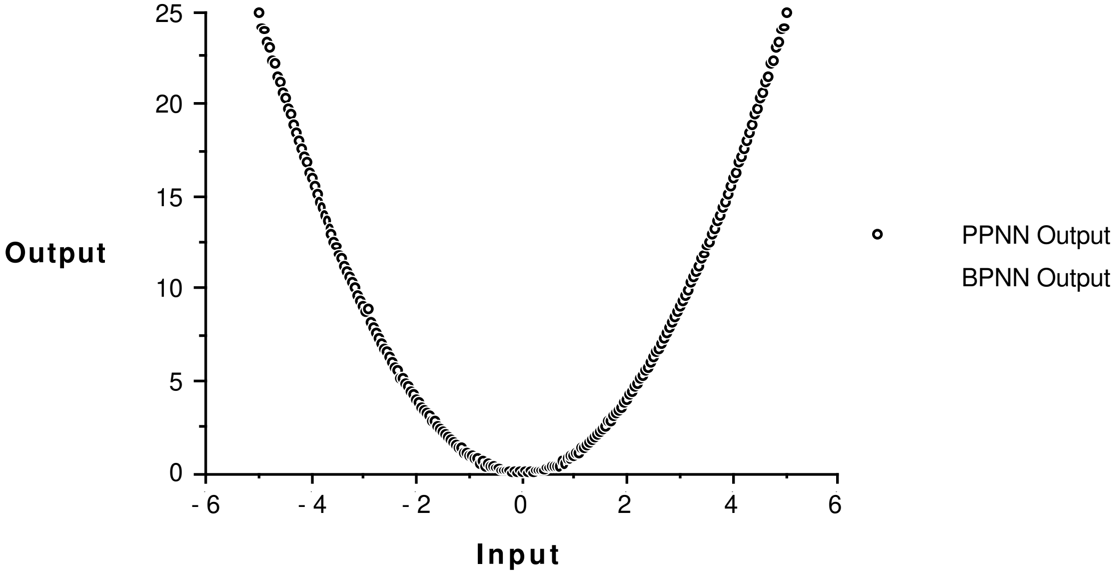

Figure 3. The net test outputs of BPNN and PPNN for parabolic problem. The test outputs generated by PPNN were very close to the desired parabola curve $y=x^2$. The test outputs generated by BPNN were lying on the straight line y=0 (on x-axis). Therefore, PPNN had a better generalization ability than BPNN.

4. DISCUSSION

4.1. Learnability

In BPNN, because there is only one branch of hidden layer, all the training patterns share the same set of weights. Therefore the weights in BPNN are influenced by all the training patterns. This leads to an oscillation and a compromise process of weights values in BPNN. The final weights of the trained BPNN are actually a compromise or trade-off among all the training patterns. It is the process of reaching a compromise that makes the training of BPNN very slow. No matter what learning algorithm is used and what node characteristics are assumed, this compromise requirement always exists as long as the network employs the BPNN planar topology. So BPNN planar topology sets limits for the convergence rate and other performances of BPNN.

In PPNN, because the weights of the pth parallel branch of hidden layer is updated by the training pattern p only, these weights are updated straightforwardly to the optimum values that best match the input/output pair of the training pattern p. The weights in the pth parallel branch of hidden layer belong to the pth training pattern only, and are not shared by other training patterns. So there is no oscillation of weights, and no compromise or trade-off process here. Due to the lack of the weights oscillation and compromise process, the local minima problem does not occur in PPNN. The global minimum can be reached very quickly. Also because there is no compromise process, the global minimum can be reduced to any small value. So PPNN can converge quickly to problems with any preset convergence criterion. Therefore, although we have not applied PPNN to *any* problems, we predict that PPNN can learn *any* problems with very fast convergence rate.

In a word, the PPNN (1) converges much faster than BPNN; (2) avoids the local minima; and (3) is able to converge to any preset convergence criterion, i. e. can learn any problems. So PPNN has overcome the three drawbacks of BPNN addressed in the Introduction part of this paper. It is the *parallel* characteristic of the stereotopology that makes PPNN a network capable of learning any kinds of problems much faster than BPNN.

4.2. Generalization Ability

In BPNN, because all the training patterns share the same branch of hidden layer, memories of all the training patterns are stored in the same set of weights. In test, this sole branch of hidden layer is on and all the memories stored there are retrieved simultaneously with the same probability. This makes the BPNN an uniformly or equal-weighted distributed interpolating and extrapolating generalizer.

In PPNN, the memory retrieval process is divided into two stages. First, the previously stored memories stored in each of the parallel branches of hidden layers are retrieved individually. Each of the P parallel branches of hidden layers generates its own test output independently within its own branch of hidden layer. Second, if the test pattern is not close enough to any training patterns, the net test output is produced interpolatively and extrapolatively by means of weighted mean of all the individual outputs from these parallel branches of hidden layers. If the test pattern is close enough to a training pattern p , the net test output is just the p th parallel branch of hidden layer s test output. So it is the *probabilistic* characteristic of PPNN s memory retrieval rule that makes PPNN a better generalizer than BPNN (Figure 3).

5. CONCLUSION

Current BPNN has several problems which severely restrict its practical applications. PPNN was proposed to overcome these problems by adopting a new topology, the stereotopology to replace the planar topology which most of the current neural networks are using. It was shown in this paper that (1) the *parallel* characteristic of PPNN s stereotopology makes PPNN a universal network capable of learning any kinds of problems with faster convergence rate than BPNN, and (2) the *probabilistic* characteristic of PPNN s memory retrieval rule ensures that PPNN has a better generalization ability than BPNN.

REFERENCES

1. Rumelhart, S. E., and McClelland, J. L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition.* Cambridge, MA: MIT Press.
2. Austin, J. (1988). Hybrid neural networks. *Neural Networks,* **1,** Suppl. 1, 5.
3. Hayduk, J. W., and Fong, Y. (1988). Fast learning of arbitrarily shaped decision regions in neural networks. *Neural Networks,* **1,** Suppl. 1, 25.
4. Pao, Y., and Beer, R. D. (1988). The functional link net: a unifying network architecture incorporating higher order effects. *Neural Networks,* **1,** Suppl. 1, 40.
5. Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks,* **1,** 295-307.
6. Shepanski, J. F. (1988). Multilayer perceptron training using optimal estimation. *Neural Networks,* **1,** Suppl. 1, 51.
7. Fogaca, M., Kramer, A., and Moore, B. (1988). Scalability issues in neural networks. *Neural Networks,* **1 ,** Suppl. 1, 92.
8. Winter, R., and Widrow, B. (1988). MADALINE Rule II: a training algorithm for neural networks. *Neural Networks,* **1,** Suppl. 1, 148.
9. Vogl, T. P., Mangis, J. K., Rigler, A. K., Zink, W. T., and Alkon, D. L. (1988). Accelerating the convergence of the back-propagation method. *Biological Cybernetics,* **59,** 257-263.
10. Peterson, C., and Hartman, E. (1989). Explorations of the mean field theory learning algo rithm. *Neural Networks,* **2,** 475-494.
11. Tollenaere, T. (1990). SuperSAB: Fast adaptive back propagation with good scaling properties. *Neural Networks,* **3 ,** 561-573.
12. Brent, R. P. (1991). Fast training algorithms for multilayer neural nets. *IEEE Transactions on Neural Networks,* **2 ,** 346-354.
13. Madhavan, P. G., Xu, B., and Stephens, B. E. (1991). On the convergence rate and generalization ability of tri-state neural networks. Proceedings of the Fourth Conference on Neural Networks and Parallel Distributed Processing, Ft Wayne.
14. Rigler, A. K., Irvine, J. M., and Vogl, T. P. (1991). Rescaling of variables in back propagation learning. *Neural Networks,* **4,** 225-229.
15. Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science,* **9,** 147-169.
16. Cuykendall, R., and Reese, R. (1988). Scaling techniques for neural algorithms. *Neural Networks,* **1,** Suppl. 1, 87.
17. Baba, N. (1989). A new approach for finding the global minimum of error function of neural networks. *Neural Networks,* **2,** 367-373.
18. Baldi, P., and Hornik, K. (1989). Neural networks and principal component analysis: learning from examples without local minima. *Neural Networks,* **2,** 53-58.
19. Hirose, Y., Yamashita, K., and Hijiya, S. (1991). Back-propagation algorithm which varies the number of hidden units. *Neural Networks,* **4 ,** 61-66.
20. Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE Acoustics Speech Signal Processing Magazine,* **4,** 4-22.
21. Leonard, J., and Holz, M. (1988). System applications. In DARPA Neural Network Study, AFCEA