Artificial Neural Networks- Growth & Learn: a Survey

Apoorvi Sood

Abstract- Incremental Learning using Constructive algorithms help us to change the structure of the neural network by adding or removing the links. These algorithms start with a small network which grows dynamically by the addition of hidden layers/units. Thus we need to overcome the problem of over-fitting and get a network with high generalization performance.

Keywords: Artificial Neural Networks, Constructive Algorithms, Optimization Algorithms, Genetic Algorithms, non-Evolutionary Algorithms.

I. INTRODUCTION

We know than an artificial neural network[12] needs the definition of the architecture before learning phase and then training the architecture and choosing the most suitable one for the application depending upon the error between the target and the actual output.

The Constructive Neural Networks differ from the traditional neural networks in a way that the network architecture is defined alongside the learning phase[31].

We can design a neural network appropriately by 2 different approaches: Evolutionary[26], where a search strategy like GA can be used to evolve the NN architecture and Non-Evolutionary, where some specific algorithms can be used to design the NN architectures, just like the constructive algorithms.

Many CoNN algorithms are available and they differ from each other in a number of ways like the number of nodes in each layer, direction of growth, stopping criteria for training, etc.,[22]:

Thus we start from a small network and keep on adding connections for decreasing the error. We therefore need some strategy which improves the network. This can be done in two ways: either local or distributed [4]:

A. Local

It is a competitive strategy where the new unit which gets the control, the old units do not get activated.

B. Distributed

In this method the network is divided into sub-networks which can be trained individually. Then these sub-networks can be added incrementally, with competition amongst each other or as a hidden layer.

II. ORGANISATION OF THE PAPER

Section 3 discusses the related work in this field, section 4 surveys the work done and section 5 is the conclusion and summary.

III. OBJECTIVE AND RELATED WORK

The Control Strategy [30] states the manner in which we try to search for a solution to a given problem in terms of architecture and weights.

Optimization algorithms can be used to find the weights. All of the constructive algorithms follow the generate-and-test approach with the following steps: We first need to find a network based on the control strategy; we then test the obtained solution before accepting it; We simply exit if the solution obtained id found to be acceptable, otherwise we repeat the steps again.

1. Therefore the main advantages of using a smaller network are:
2. The learning and computation cost increases the efficiency and
3. The functionality of hidden layer becomes easy to access.

IV. LITERATURE SURVEY OF WORK DONE

1. Cascade correlation learning architecture(CCA)[9]:

This algorithm starts with a small network, adding and training new hidden units as and when required to form a multilayer network. It uses the Quickprop algorithm to manage the step-size problem that occurs in back-propagation algorithm. Like the back-propagation algorithm, the Quickprop algorithm also calculates the error. Second order method is used in Quickprop algorithm to update the weights[9]. It also overcomes the problem of the moving target and helps in faster learning by making changes in only some of the units at a time.

The algorithm initially starts with some input and output units connected by an adjustable weight. There are no hidden units yet.The hidden units are added to the architecture one by one. It is connected to the original inputs and the other already existing hidden units. We now freeze the weights of the hidden unit and train the connections at the output. The widrow-hoff(delta) rule, Perceptron algorithm or any other learning algorithm can be used. The output weights can be trained by Quickprop algorithm as it converges faster with no hidden units. We stop the training cycles when significant error reduction has been achieved. In-case of further reducing the error we may add hidden units and repeat the above steps to achieve a small error. Now we need to adjust the input weight of the candidate unit, so that we can maximize the sum over all the output units(correlation). A collection of candidate units having different initial weights being trained simultaneously can also be used. The unit with the best correlation can be added to the network.

2. Adaptive slope sigmoidal function cascading neural network algorithm(ASCNNA)[27]:

This algorithm focuses on adapting the network architecture and its functionality. The regression problems use this algorithm. One of its variant, Basic cascading neural network algorithm(BCNNA) uses log-sigmoidal function in the hidden units.

3. Generalized Constructive training algorithm using ASAF Framework(ASAFCA)[28]:

Manuscript received on July, 2013

Apoorvi Sood, Done B.E, M.Tech and now pursuing Ph.D. Working as Assistant professor in CSE & IT Department in ITM University.

This algorithm uses two parameters that are trained along with the weights simultaneously. It is a new type of adaptive sigmoidal activation function(ASAF). The computation improves by freezing the weights. This algorithm uses adaptive sigmoidal activation functions to improve the generalization and the learning capability of the network. The algorithm can be used for both fixed size neural networks as well as for constructive neural networks. The activation function used should be non-constant, bounded, monotonically increasing and differentiable everywhere and a sigmoidal function.

4. New Constructive Algorithm(NCA)[20]:

This algorithm uses different datasets for training the hidden neurons of the hidden layer to improve the learning of the network. It constructively determines the number of hidden neurons in the hidden layers. NCA incorporates the following components along with the constructive approach:
Creating new training sets: It uses AdaBoost algorithm to maintain the probability distribution D(number of training examples) over some training set T. Since D is a vector thus its component represents the probability of selection of each training example. The initial value of D is 1/Pt. This D is updated using random sampling of Pt and replacement from T. Using the new D, a new dataset T' is created by the AdaBoost algorithm. Other neurons are trained by repeating this process again. The neural network architecture is functionally adapted using a different data set for training. Only one neuron is trained at a time.
Two stage training: Training is done in two parts-Initial stage and Final stage. The algorithm first initializes the weights at the input and output of the neuron to zero. In the hidden layer, the weights are initialized with some random values in a small range. It uses back-propagation algorithm to train the neurons for modifying their weights.
In the final stage, Gaussian noise with mean zero and variance with value one is added to the connection weights of the earlier trained neuron. For optimizing the new weights, it then reapplies the back-propagation algorithm. Only one cost function is used for training a hidden neuron and adding one.

5. An Adaptive Slope Basic Dynamic Node Creation Algorithm(ASBDNCA)[29]:

Adaptive sigmoidal activation function is used at hidden nodes while the output node uses linear activation functions.When the learning cannot be improved further, we add a new hidden node to the current network and freeze the weights of the trained network. The residual error can be reduced by choosing the input and output weights, bias at the output node and the slope parameter. If we control the slope of the sigmoidal activation function, then the architecture is capable of nonlinear mapping. This algorithm learns on the basis of generalization capability and the time taken for training.

V. SUMMARY

These algorithms aim to find the optimum value for slope parameter in addition to finding the number of hidden neurons.

REFERENCES

1. Alpaydm, E. (1988) "Grow and Learn" Internal Note, Laμi-EPF Lausanne, Switzerland.
2. Alpaydm, E. (1990a) Neural models of incremental supervised and unsupervised learning, PhD dissertation, Ecole Polytechnique Fédérale de Lausanne, Switzerland.
3. Alpaydm, E. (1990b) "Grow and Learn: An Incremental method for category learning" Int. Neural Network Conf., Paris, France.
4. Alpaydm, E., (1991) "GAL: Networks that grow when they learn and shrink when they forget", International Computer Science Institute.
5. Ash, T. (1989) "Dynamic node creation in backpropagation networks," Connection Science, 1, 365-375.
6. Bishop, C.M. Neural Networks for Pattern Recognition. London: Oxford University Press, 1995.
7. Chiang, K-W., Noureldin, A., El-Sheimy, N. (2008) Constructive Neural-Networks-Based MEMS/GPS Integration Scheme IEEE Trans. Vol. 44, NO. 2.
8. Diederich, J. (1988) "Connectionist recruitment learning" Proc. Of the 8th European conf. On Artificial Intelligence, London, UK.
9. Fahlman, S.E., Lebiere, C. (1990) "The cascade-correlation architecture," in Advances in neural information processing systems, D.S. Touretzky (ed.), 2, 524-532, Morgan Kaufman.
10. Feldman, J., (1982) "Dynamic connections in neural networks," Biological Cybernetics, 46,27-39.
11. Frean, M. (1990) "The Upstart Algorithm: A method for constructing and training feedforward neural networks", Neural Computation,2 198-209.
12. Haykin, S. Neural Networks: A Comprehensive Foundation (2nd ed.). Upper Saddle River, NJ: Prentice Hall, 1999.
13. Hertz, J., Krogh, A., Palmer, R.G. (1991) Introduction to the theory of neural computation, Addison Wesley.
14. Hirose, Y., Yamashita, K., Hijiya, S. (1991) "Back-propagation algorithm which varies the number of hidden units," Neural Networks, 4, 61-66.
15. Honavar, V., Uhr, L. (1988) "A network of neuron-like units that learns to perceive by generation as well as reweighting of its links," Proc. of the 1988 Connectionist Summer School, D. Touretzky, G. Hinton, T. Sejnowski (eds.), Morgan Kaufman.
16. Kitano, H., (1994) Neurogenetic learning: An integrated method of designing and training neural networks using genetic algorithms, Physica D75, pp. 225-238.
17. Knerr, S., Personnaz, L., Dreyfus, G. (1989) "Single layer learning revisited: A stepwise procedure for building and training a neural network." In Neurocomputing: Algorithms, architectures, and applications, F. Fogelman-Soulié, J. Hérault (eds.), NATO ASI Series, vol. F68, pp. 41-50. Springer, Heidelberg (1990).
18. Ma, L., Khorasani, K., (2002) "Application of Adaptive Constructive Neural Networks to Image Compression," IEEE Trans. Neural Net., Vol. 13, NO. 5.
19. Mézard, M., Nadal, J.-P. (1989) "Learning in feedforward layered networks: The tiling algorithm," Journal of Physics A, 22, 2191-2204.
20. Md. Islam M., et al (2009) "A New Constructive Algorithm for Architectural and Functional Adaptation of Artificial Neural Networks," IEEE Trans. Cybernetics, Vol. 39, No. 6.
21. Müller, B., Reinhardt, J. (1990) Neural Networks: An introduction, Springer Verlag.
22. Parekh, R., Yang, J., Honavar, V., (2000) "Constructive Neural-Network Algorithms for Pattern Classification," IEEE Trans. Neural Net., Vol. 11, NO. 2.
23. Reilly, D.L., Cooper, L.N., Elbaum, C. (1982) "A neural model for category learning," Biological Cybernetics 45,35-41.
24. Rizzi, A., Mascioli, F.M.F., Martinelli, G. (2002) "Adaptive Resolution Min-Max Classifiers," IEEE Trans. Neural Net., Vol. 13, NO. 2.
25. S,-C. Huang, Y,-F. Huang (1991) "Bounds on the number of hidden neurons in multilayer perceptrons," IEEE Trans. Neural Net,. Vol. 2, pp. 47-55.
26. Schaffer, J.D., Whitely, D., Eshelman, L.J. (1992) "Combinations of genetic algorithms and neural networks," COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks, IEEE Computer Society Press.
27. Sharma S.K., Chandra P. (2010) "An adaptive slope sigmoidal function cascading neural networks algorithm," Third International Conference on Emerging Trends in Engineering and Technology, 2010 IEEE.
28. Sharma S.K., Chandra P. (2012) "Empirical Evaluation of Adaptive Sigmoidal Activation Function on a Constructive Algorithm," CSI Journal of Computing, Vol. 1 No. 3.
29. Sharma S.K., Chandra P. (201o) "An Adaptive Slope Basic Dynamic Node Creation Algorithm," International Conference on Computational Intelligence and Communication Networks.
30. T.Y. Kwok, D.Y. Yeung (1996) "Constructive algorithms for structure learning in feedforward neural networks for regression problems," IEEE Trans. Neural Net., Vol. 7, pp. 1168-1183.
31. T.Y. Kwok, D.Y. Yeung (1997) "Objective Functions for Training new Hidden Units in Constructive Neural Networks," IEEE Trans. Neural Net., Vol. 8, NO. 5.