International Journal of Mathematics And Its Applications
Vol.2 No.2 (2014), pp.23-28.
ISSN: 2347-1557(online)

Simulation of objective function for training of new hidden units in constructive Neural Networks

Vineeta Yadav†,1 **A.K.Verma**§ **and Rizwana Zamal**‡

‡Saifia Science College, Bhopal(MP)-462001, India.

†,§Institute of Mathematical Sciences & Computer Applications,

Bundelkhand University, Jhansi(UP)-284128, India.

vineetayadavmath@gmail.com

Abstract : The present research article represent the mathematical analysis of objective function for training new hidden units in constructive algorithms for multi-layer feed-forward networks. Neural research, now days, is highly attractive wing under research community which may lead the development of some hidden prospects by using mathematical modeling which involve the design of neurons network. The network size is highly important for neural network. Small network as well as large network size cannot be learned very well, but there is an optimum size for which the neural network can be involved for good results. Constructive algorithms started with a small network size and then grow additional hidden units until a satisfactory solution is found. A network, having n-1 hidden units, is directly connected to the output unit, which is modeled as $f_{n-1} = \sum_{j=1}^{n-1} \beta_j g_j$ and $e_{n-1} = f - f_{n-1}$ is the residual error function for current network with $n-1$ hidden units. A new hidden unit is added under a process in input as a linear combination of g_n with the current network $f_{n-1} + \beta_n g_n$ which governed the minimum residual error $\|e_n\|$ in the output process by keeping g_n fixed and adjusted value of β_n so as to minimize residual error. The function to be optimized during input training is $\frac{\langle e_{n-1}, g_n \rangle^2}{\|g_n\|^2}$ and corresponding objective function is

[1]Corresponding author E-Mail: vineetayadavmath@gmail.com (Vineeta Yadav)

AMS Subject Classification:

© JS Publication

$s_1 = \frac{(\Sigma_p E_p H_p)^2}{\|g_n\|^2}$, where H_p is the activation function of the new hidden unit and E_p is the corresponding residual error before this new hidden unit is added.

Keywords : constructive algorithms, hidden units, feed forward networks.

---

1 Introduction

Many neural network models have been proposed for pattern classification, function approximation and regression problems. Optimization of neural network topology has been one of the most important problems since neural network came in front as a method for prediction and association. Number of heuristic formula for determining the numbers of hidden units were developed and some algorithms for structure optimization were suggested, such as cascading, pruning and others [1].

The network size is very important for neural network, small network as well as large network size can not learn very well but there is an optimum size for which the neural network can be involved the good result. Constructive algorithm finds an appropriate network size automatically for a given application and optimizes the set of network weight. Constructive algorithms start with a small network size and then grow additional hidden units and weights until a satisfactory solution is found.Constructive algorithm is straight forward to specify initial network. Initial network means the smallest possible network to start with has no hidden units.

Constructive algorithms always search for small network solution first so constructive algorithm is more computationally economical because smaller network are more efficient in forward computation. The neural network with fixed architecture is only trained once, but in constructive methods, every time the architecture is changed and the training must be repeated. The computational cost of repeated training is high, so some techniques as weight freezing, re-training the whole network continuing from previous weights and re-training the whole network from scratch are applied to reduce both time and space complexity. To improve computational complexity, in constructive algorithm, it is assumed that the hidden units already exiting in the network and this useful in modeling part of the target function.

Hence the weights feeding into these hidden units fixed (input weight freezing) and allow only the weights connected to the new hidden unit and output hidden units to vary. The number of weights to be optimized, and the time and space requirements for each iteration can thus be greatly reduce. This reduction is especially significant when there are already many hidden units installed in the network. So neural networks are now days, being used to solve a whole range of problems, most common case studies include modeling hitherto intractable processes, designing complex feed-back control signal, building meta-models for posterior optimization or detecting device faults [2]. Cascade correlation learning algorithm starts with a small network. The key idea of for cascade-correlation type constructive neural network algorithms is to split the pool of candidate hidden units into two groups. The candidate, which is installed in the active network, either becomes part of the current deepest hidden layer or adds a new hidden layer to the network [7]. 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. Second order method used in Quickprop algorithm to updates the weights. The algorithm initially starts with some input and output units connected by an adjustable weight. There are no hidden units yet.

We now freeze the weights of the hidden units and trained the connections at the output. The output weights can be trained by Quickprop algorithm as it converges faster with no hidden units. We stop the training cycle when significant error reduction has been achieved. In case of further reducing the error we may add hidden units and repeat these steps to achieve a small error. Now we adjust the input weight of the candidate unit. So that we can minimize the sum over all the output units. 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 [5].

The puzzle of the human brain is as old as human history. By day-today experience we accumulate and associate all advances in human knowledge. The design of network based on intelligent is generally gives the network learning and the tool which is developed for learning is called neural network [3]. Artificial neural network has a lot of important capabilities such as learning from data, generalization, working with unlimited number of variables. Neural networks may be used as a direct substitute for autocorrelation, multivariable regression, linear regression, trigonometric and other statistical analytic techniques [4]. Successful application examples show that human diagnostic capabilities are significantly worse than the neural diagnostic system. A constructive neural network algorithm with backpropagation offers an approach for incremental construction of near- minimal neural network architectures for pattern classification. The algorithm starts with minimal number of hidden units in a single hidden layer, additional units are added to the hidden layer one at a time to improve the accuracy of the network and to get an optimal size of a neural network.

For building and training single hidden layer neural network a simultaneous perturbation training algorithm is used, in this method all hidden neurons are used to model the relationships between input data and modal residuals. A sigmoid hidden neuron is added to the single layer of hidden neurons after a period of training when the error has stopped decreasing below a threshold value. After the addition of the new hidden neuron, the whole network is again trained with simultaneous perturbation. In training, perturbed value to the connection weights are changed to detrap the local minima [10].The role of adaptive sigmoidal activation function has been verified in constructive neural networks for better generalization performance and lesser training time [6]. Constructive neural networks are a collection of a group of methods, which enable the network architecture to be constructed along with the training process. Constructive algorithms that construct feed-forward architecture for regression problems.

A general constructive approach for training neural networks in classification problem is used to construct a particular connectionist model, named switching neural network (SNN), based on the conversion of the original problem in a Boolean lattice domain [8]. In general a constructive algorithm has two integral components pre-specified network growing strategy and local optimization technique for updating weights during learning. In a constructive adaptive neural network, the error signals during the learning process improve the input-side training effectiveness and efficiency, and obtain better generalization performance capabilities [11].

2 Design of the Objective Function

Constructive algorithms are supervised learning algorithms for neural network. They start individual neural network with a small architecture, small number of hidden layers, nodes and connection, and then add hidden nodes and weights incrementally until a satisfactory solution is found. Constructive algorithms have been applied to benchmark problems, which are popular both in case of machine learning and neural network community [9].

A network, having n-1 hidden units directly connected to the output unit, implements the function

$$f_{n-1} = \sum_{j=1}^{n-1} \beta_j g_j$$

Where g_j represent the function implemented by the j_{th} hidden unit. These g_j may only be indirectly connected to the input units through intermediate hidden units, and the network is not restricted to having only one single hidden layer. Moreover, $e_{n-1} \equiv f - f_{n-1}$ is the residual error function for the current network with $n-1$ hidden units.

Addition of a new hidden unit proceeds in two steps:

Input training: find β_n and g_n such that the resultant linear combination of g_n with the current network, i.e. $f_{n-1} + \beta_n g_n$, gives minimum residual error $\|e_n\|$.

Output training: keeping $g_1, g_2, ...g_n$ fixed, adjust the value of $\beta_1, \beta_2, ...\beta_n$ so as to minimize the residual error.

Derivation for objective function

For a fixed $g(\|g\| \neq 0)$, the expression $\|f - (f_{n-1} + \beta g)\|$ achieves its minimum iff $\beta = \beta^* = \frac{\langle e_{n-1}, g\rangle}{\|g^2\|}$. Moreover, with β_n^* and β^* as define above, $\|f - (f_{n-1} + \beta_n^* g_n)\| \leq \|f - (f_{n-1} + \beta^* g)\| \;\forall g$, iff $\frac{\langle e_{n-1}, g_n\rangle^2}{\|g_n\|^2} \geq \frac{\langle e_{n-1}, g\rangle^2}{\|g\|^2} \;\forall g$.

Case:1 When all elements are of unit norm

$$\begin{aligned}\Delta g &\equiv \|f - f_{n-1}\|^2 - \|f - (f_{n-1} + \beta g)\|^2 \\ &= 2\langle f - f_{n-1}, \beta g\rangle - \beta^2\langle g, g\rangle\end{aligned}$$

Put $f - f_{n-1} = e_{n-1}$ then we find $\Delta g = 2\beta\langle e_{n-1}, g\rangle - \beta^2\|g\|^2$

Taking $\|g\| = 1$ the expression will be $\Delta g = 2\langle e_{n-1}, g\rangle - \beta^2$. Adding and subtracting $\langle e_{n-1}, g\rangle^2$ then we find the expression $\Delta g = \langle e_{n-1}, g\rangle^2 - (\langle e_{n-1}, g\rangle - \beta)^2$. So for a fixed g the stated expression is minimized iff $\beta = \beta^* = \langle e_{n-1}, g\rangle$ with $\Delta_{max}(g) = \langle e_{n-1}, g\rangle^2$. This shows that the stated expression is minimized when $\Delta_{max}(g)$ is maximized over all g.

Case:2 When the assumption of unit norm is dropped, then the stated expression is minimized iff $\beta = \beta^* = \frac{\langle e_{n-1}, g\rangle}{\|g\|^2}$ with $\Delta_{max}(g) = \frac{\langle e_{n-1}, g\rangle}{\|g\|^2}$. This result suggests that the objective function to be optimized during input training is $\frac{\langle e_{n-1}, g\rangle^2}{\|g\|^2}$. The above function can only be calculated when the exact functional form of e_{n-1} is available, which is obviously impossible as the true f is unknown. A consistent estimate of above function using information from the training set is $\frac{(\frac{1}{N}\Sigma_p E_p H_p)^2}{\frac{1}{N}\Sigma_p H_p)^2}$, where H_p is the activation of the new hidden unit for the pattern p and E_p is the corresponding residual error before this new hidden unit is added. Dropping the factor $\frac{1}{N}$ which is common for all candidate hidden unit function, we obtain the objective function- $S = \frac{(\Sigma_p E_p H_p)^2}{\Sigma_p H_p)^2}$.

3 Conclusions

We have studied the mathematical analysis of the objective function for training new hidden units in constructive algorithm. The objective function to be optimized during input training is $\frac{\langle e_{n-1}, g_n\rangle^2}{\|g_n\|^2}$ which can be used for training of new hidden units.

References

[1] Sheng-Vei Gua, Qi Yinan, SynKiat Tan and Shanchun Li, *Output partitioning of neural network*, Neurocomputing 68(38)(2005).

[2] Roberto Lopeg Gonzalez, *Neural networks for variational problems in engineering* , Ph.D., thesis, Technical university of Catalonia (2008).

[3] Ben Krose and Patrick Vander Smagt, *An introduction to neural network*,(1996).

[4] Michael Wynne-Jones, *Model building in neural networks and hidden markov models*, Ph.D., thesis, University of Edinburgh (1993).

[5] Apporvi Sood, *Artificial neural networks-Growth and Learn: a survay*, International journal of soft computing and engineering 2(103)(2013).

[6] Sudhir Kumar Sharma and Pravin Chandra, *Constructive neural network: A review*, International journal of engineering science and technology 2(12)(2010).

[7] Jani Lahnajarvi, Mikko Lehtokangas and Jukka Saarinen, *Constructive neural network- automated allocation of hidden units into hidden layers*, Proc. WSES, (2001).

[8] Enrico Ferrari and Marco Muselli, *A constructive technique based on linear programming for training switching neural network*, Lecturer note in computer science 5164(744)(2008).

[9] Kazi Md. Rokibul Alam, Md. Shamsul Huda, Md. Mirza Hafiz Ai-asad and Md. Mostafa Zaman, *A comparison of different constructive algorithms to design neural networks*, Third international conference on electrical and computer engineering, 28-30 December, Dhaka, Bangladesh (2004).

[10] P. Tanggovel and T. Kathirvalava Kumar, *Simultaneous perturbation for single hidden layer networks-cascade learning*, Neuro computing 50(193)(2003).

[11] L. Ma and K. Kharasani, *New training strategies for constructive neural networks with application to regression problem*, Neural Networks 17(4)(2004).