



Improvement in the Speed of Training a Neural Network

Harikrishna B. Jethva¹, Dr. V.M.Chavda²
Ph. D. Scholar¹, Principal²,
Department of Computer Science and Engineering¹,
Bhagwant University¹, SVICS²,
Ajmer, Rajasthan-305004¹, Kadi, Gujarat-38271².
INDIA¹, INDIA².

Abstract- Today the most important use of neural network is in classifying the data. The neural network is trained for classification by giving some input-output pairs which is also known as supervised learning. As training a network is a part of the whole classification process, the time required for training should be as small as possible. In the system, optimization is carried out using multilayer neural network. The efficacy of the proposed method is observed during the training period as it converges quickly for the dataset used in test. As it converges quickly it reduces the time required for training.

Keywords—Supervised Learning, Learning rate, Classification, Optimized Training Time, Un-supervised Learning.

I. Introduction

Classification is basically a set of activities responsible for deriving a model that categorizes and describes classes of data and concepts, whose sole purpose is to determine and predict the classes of objects who have no label [1]. The model derived can be shown in various forms, such as *classification (IF-THEN) rules*, *decision trees*, *mathematical formulae*, or *neural network*.

A *neural network*, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units. Fig 1 shown below is the example of a Feed-Forward Artificial Neural Network. Every neural network has one input and one output unit along with zero or more number of hidden units. The input unit receives the input, the output unit generates the classification result and the hidden unit performs the processing [1].

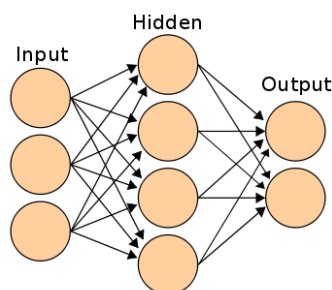


Fig 1: A Feed-Forward Neural Network

There are some very important components those make up an artificial neuron. These include weighting factor, summation function, transfer function, error function, error and back-propagated value and learning function. The most important among these is the *learning function*. Its purpose is to modify the variable connection weights when inputs are given to each processing element based on some neural based algorithm.

In neural network, training the learning algorithms plays quite important role in the process. Throughout the process, the learning algorithm is used to adjust the weight, bias and other input parameters in such a manner that the model is able to count its best fit with the environment in a minimum amount of time. Back-propagation in multilayer neural network is one of the supervised training procedure and variety of approaches are developed over the last decade for its applicability to a numerous number of application [6][7].

The slow convergence rate of steepest descent algorithm encourages many ideas to be developed for faster convergence rate in training a multilayer neural network.

The proposed algorithm is a developed method based on the Gauss-Newton numerical optimization technique. It resulting a quick convergence with stability in case of multilayer neural network learning. The proposed method converges according to the following criteria:

1. MSE (Mean Square Error): The algorithm stops when it reaches the pre-specified threshold value.
2. Correct Classification: The main idea behind the convergence criteria is that if any neural network training algorithm correctly classifies some input, then it is possible to calculate the percentage of correctly classified input, and the same is increased or decreased according to the ability of training algorithm. The proposed algorithm is designed with one pre-defined correct classification threshold value.

Further this paper is divided into following sections, section II is about training a neural network. Section III is about the proposed scheme of sampled training in a neural network, section IV is about the advantages of the proposed scheme and section V is the conclusion so derived after the study of the two schemes.

I. Training a neural network

A. Process of Classification

Every classification technique basically consists of two phases: the learning or the training phase and the other is the output or the testing phase. In the learning phase, the training data whose data objects have a known class label are given as input to the model so as to train the model. Once the model has been trained, the original data set that is to be classified is given as input to the model. The model based on information learned from the training data classifies the test data into appropriate class labels.

B. Batch Mode Training (Existing Method)

In a feed-forward neural network, training of the network is done by providing the input-output pairs to the network and then generating the model which can then be used for classification of the test data. In case of the existing system, the whole of the training set is given as input for training the network. So the total time required for performing the classification includes the time required for training and the time required for classifying the test data based on the model generated.

It has found that for better error gradient estimations, more number of input vectors for each gradient calculation should be used [2]. But the down side of using more input vectors per weight update is that the gradient has to be calculated for each input vector used [3]. In the training process it is the gradient calculations that take the longest. Using multiple input vectors quickly lengthens the training times to intolerable levels.

In batch mode of training a neural network, the complete training set is used for training. After the training has been accomplished and the model generated, the test data is given as input for classifying the unknown class labels. This is how the existing system works.

But the problem with this training is that the time required to train a neural network is more as the whole training set is used [4]. So, in order to reduce the training time, we propose a new scheme where the training set is sampled into smaller chunks.

II. Proposed method

The proposed algorithm includes one adjustment to the weight and bias after each iteration. The adjustment is computed for each weight and bias of each layer. The proposed algorithm introduces a new step where the weight is adjusted before each iteration. In some cases it is observed that the proposed algorithm results fast convergence. The requirement of the memory for the whole process of optimization is also minimum. The algorithm is found to behave like normal optimization, if weight adjustment is not performed. It is possible to convert the same algorithm into Levenberg-Marquardt algorithm, if one damping factor is introduced in the calculation of ΔX , and is adjusted according to the requirement of algorithmic optimization criteria.

The performance of proposed algorithm is evaluated in terms of percentage of correct classification, mean of square error and the number of iteration required to converge [5].

The system control flow is depicted in Fig.3. The non-linear input parameters are processed through the classifier and data processing unit is required to finalize the data before training and testing. The data processing unit prepared the various parameter is needed to train the neural network. The unit is responsible to design the network according to the setting defined in the file. The type of algorithm is used for analysis defined in the file. The next part is the “ANN BOX” which is the main unit of the system. The proposed method is implemented in this unit.

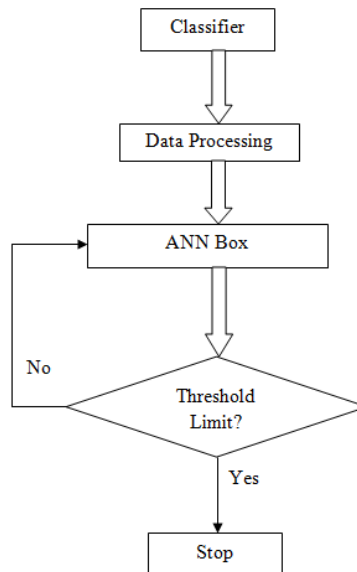


Fig 3: The system control-flow diagram

The system stops when any of the previously defined convergence criteria crosses the threshold limit.

III. Advantages of Proposed Method

Lesser Weight Updates

As the number of iteration for converging to constant values of weights and bias are reduced, the overall number of weight updates per iteration is also reduced.

Reduction in Training Time

As the model converges to a constant value in less time, the time required for training a neural network also reduces.

Low Memory Requirements

The requirement of the memory is also low in comparison with the proposed method if the convergence criteria are based on the correct percentage of classification or MSE rate.

IV. Conclusion

In Feed-Forward neural networks used with Gradient Descent Technique, training time is the most important parameter to be considered. As the gradient calculations require or consume more time than actual training. So the technique of using the training set should be checked to reduce the training time.

In batch mode training, the numbers of weight updates are more with increase in the training data which in turn results in the increase of gradient calculations. Due to more gradient calculations, the training time also increases.

An improved method based on the Gauss-Newton numerical optimization technique is. The advantages so discussed in the above section are based on the proposed method, it may be concluded that the proposed technique is superior in terms of time taken to converge and the memory space, both being reduced to a great extent. Thus the benefit of the present method is established.

V. References

- [1] Jiaweu Han and Micheline Kamber, "Data Mining Concepts and Techniques", second edition, Elsevier press.
- [2] William Daniel Nortje, "Comparison of Bayesian Learning and Conjugate Gradient Descent Training Of Neural Networks", University of Pretoria, October 2001.
- [3] Er. Parveen Shegal, Dr. Sangeeta Gupta and Prof. Dharminder Kumar, "Minimization of Error in Training A Neural Network Using Gradient Descent Technique", International Journal of Technical Research (IJTR), Vol-1, Issue-1, Mar-Apr 2012.
- [4] Liang Gong, Chengliang Liu and Fuqing Yuan, "Training Feed-forward Neural Network Using the Gradient Descent Method With The Optimal Step Size", Journal of Computational Information System 8:4 (2012).
- [5] Sudarshan Nandy, Prartha Prathim Sarkar and Achintya Das, "An Improved Gauss-Newtons Method Based Back-Propagation Algorithm For Fast Convergence", International Journal for Computer Applications, Vol 39-No. 8, February 2012.
- [6] Saman R., Bryan A. T., 2011, " A new formulation for feed forward Neural Networks", IEEE Trans. Neural Netw., vol.22, 10, pp. 1588- 1598.
- [7] Bogdan M. Wilamowski, Serdar Iplikci, Okyay Kaynak, M. Onder Efe, 2001, "An algorithm for fast convergence in training a Neural Network", IEEE Proceedings, pp. 1778-1782.