RESEARCHERID
THOMSON REUTERS

ISSN: 2277-9655
[Gurusamy * *et al.*, 6(11): November, 2017] Impact Factor: 4.116
IC™ Value: 3.00 CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

PERFORMANCE EVALUATION OF VARIANCES IN BACKPROPAGATION NEURAL NETWORK USED FOR HANDWRITTEN CHARACTER RECOGNITION

Vairaprakash Gurusamy [*1] & K.Nandhini[2]

[*1]Department of Computer Applications, School of IT, Madurai Kamaraj University, Madurai
[2]Technical Support Engineer, Concentrix India Pvt Ltd, Chennai

DOI: 10.5281/zenodo.1054593

ABSTRACT

A Neural Network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain.Back propagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. The term back propagation refers to the manner in which the gradient is computed for nonlinear multilayer networks. There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as conjugate gradient and Newton methods. . Properly trained back propagation networks tend to give reasonable answers when presented with inputs that they have never seen. These variances of backpropagation are used for character recognition and their comparative study on the performance of different algorithm is made in the paper

KEYWORDS: Character Recognition, Neural Network, Back propagation

I. INTRODUCTION

Overview To Character Recognition

The problem of character recognition, popularly known in the literature as optical character recognition is a subset of pattern recognition field. The first three stages come from image processing and last two stages come from pattern recognition.

Recognition of handwritten characters is a problem of considerable utility. It has been widely used for practical transformation of text form its pictorial form to coded form required forelectronic data processing. The problem is much more complicated particularly for the ideographic language likeChinese andJapanese which have a large set of basic alphabets.

Some of the application areas of handwritten character recognition are Banks, Office automation, aid for thedisabled, automatic billing, public administration, industries etc.
The character recognition methodologies can be classified as under-

- Decision theoretic approach
- Syntactic approach
- Knowledge-base approach
- Neural network approach

II. NEURAL NETWORK APPROACH: AN ARTIFICIAL NEURAL NETWORK (ANN)

Is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following two ways: they acquire knowledge through learning, and the knowledge is stored within inter-neuron connection strengths known as synaptic weights.

The most common neural network model is known as a supervised network because it requires a desired output in order to learn. The goal of this network type is to create a model that maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown.
A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include: Adaptive learning, Self-Organization, Real Time Operation, Fault Tolerance via Redundant Information Coding

III. CHARACTER RECOGNITION BY NEURAL NETWORKS

Optical Character Recognition (OCR) programs are capable of reading printed text. This could be text that was scanned in from a document, or hand written text that was drawn to a hand-held device, such as a Personal Digital Assistant (PDA). OCR programs are used widely in many industries. Many of today's document scanners for the PC come with OCR software that allows you to scan in a printed document and then convert the scanned image into an electronic text format such as a Word document, enabling you to manipulate the text. In order to perform this conversion the software must analyze each group of pixels (0's and 1's) that form a letter and produce a value that corresponds to that letter. Some of the OCR software on the market uses a neural network as the classification engine

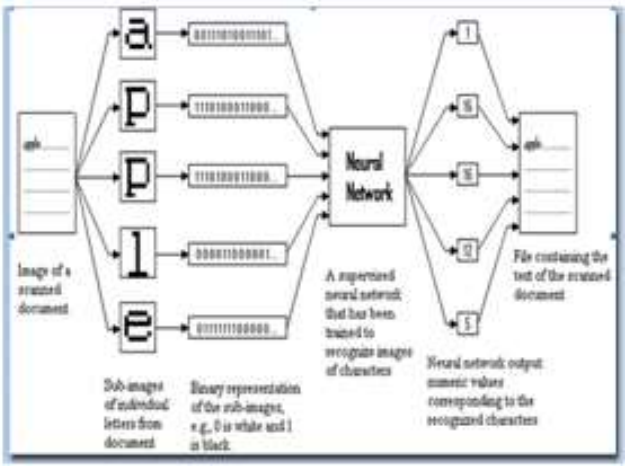

Figure 1: Example of a Neural Network for OCR

The OCR software breaks the image into sub-images, each containing a single character. The sub-images are then translated from an image format into a binary format, where each 0 and 1 represents an individual pixel of the sub image. The binary data is then fed into a neural network that has been trained to make the association between the character image data and a numeric value that corresponds to the character. The output from the neural network is then translated into ASCII text and saved as a file

IV. TEST APPLICATION

1. To generate initial receptors set. On application startup it's already generated, so it can skip this step, if are not planning to change the initial amount of receptors or the filtered amount.
2. Generate data. In this step the initial training data will be generated.
3. Filter data. In this step the initial receptors set as well as the training data will be filtered.
4. Create network - a neural network will be created.
5. Train network - neural networks training.

Clearly the term artificial neural networks encompass a great variety of different software packages with many different types of artificial neurons, network architectures, and learning rules. These different networks can, in turn, be applied to a diverse range of functions in everything from beer manufacturing to better understanding the properties of the biological brains on which they are based.

V. PROBLEMS USING NEURAL NETWORKS

Local Minimum

All the NN in this paper are described in their basic algorithm. Several suggestions for improvements and modifications have been made. One of the well-known problems in the MLP is the *local minimum:* The net does not settle in one of the learned minima but instead in a local minimum in the Energy landscape

Approaches to avoid local minimum:

- The *gain term* in the weight adaptation function can be lowered progressively as the network iterates. This would at first let the differences in weights and energy be large, and then hopefully when the network is approaching the right solution, the steps would be smaller. The tradeoff is when the gain term has decreased the network will take a longer time to converge to right solution.
- A local minimum can be caused by a bad internal representation of the patterns. This can be aided by the adding more internal nodes to the network.
- An extra term can be added to the weight adaptation: the *Momentum term*. The Momentum term should let the weight change be large if the current change in energy is large.
- The network gradient descent can be disrupted by adding random noise to ensure sure the system will take unequal steps toward the solution. This solution has the advantage that it requires no extra computation time.

VI. BACKPROPAGATION NEURAL NETWORK

Back propagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by you. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

Standard backpropagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The term backpropagation refers to the manner in which the gradient is computed for nonlinear multilayer networks. There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as conjugate gradient and Newton methods. The Neural Network Toolbox implements a number of these variations. Properly trained backpropagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs.

VII. BACKPROPAGATION ALGORITHM

The simplest implementation of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly - the negative of the gradient. There are two different ways in which this gradient descent algorithm can be implemented: **incremental mode and batch mode**. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network.

a. **Batch Training (train)**

In the batch mode all of the inputs are applied to the network before the weights are updated. In batch mode the weights and biases of the network are updated only after the entire training set has been applied to the network. The gradients calculated at each training example are added together to determine the change in the weights and biases.

b. Batch Gradient Descent (traingd)

The batch steepest descent training function is traingd. The weights and biases are updated in the direction of the negative gradient of the performance function. If you want to train a network using batch steepest descent, you

should set the network trainFcn to traingd, and then call the function train. There is only one training function associated with a given network.

c. Batch Gradient Descent with Momentum (traingdm)

In addition to traingd, there is another batch algorithm for feedforward networks that often provides faster convergence - **traingdm**, steepest descent with momentum. Momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low-pass filter, momentum allows the network to ignore small features in the error surface. Without momentum a network may get stuck in a shallow local minimum.

d. Faster Training: Thesefaster algorithms fall into two main categories.

The first category uses heuristic techniques, which were developed from an analysis of the performance of the standard steepest descent algorithm. One heuristic modification is the momentum technique, which was presented in the previous section. This section discusses two more heuristic techniques: variable learning rate backpropagation, traingda; and resilient backpropagation trainrp. The second category of fast algorithms uses standard numerical optimization techniques.

e. Variable Learning Rate (traingda, traingdx)

With standard steepest descent, the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm may oscillate and become unstable. If the learning rate is too small, the algorithm will take too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface.

The performance of the steepest descent algorithm can be improved if we allow the learning rate to change during the training process. An adaptive learning rate will attempt to keep the learning step size as large as possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface.

f. Resilient Backpropagation (trainrp)

Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called "squashing" functions, since they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slope must approach zero as the input gets large. This causes a problem when using steepest descent to train a multilayer network with sigmoid functions, since the gradient can have a very small magnitude; and therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values.

The purpose of the resilient backpropagation (Rprop) training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative is used to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value.

g. Conjugate Gradient Algorithms

The basic backpropagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient). This is the direction in which the performance function is decreasing most rapidly. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions.

h. Speed and Memory Comparison

It is very difficult to know which training algorithm will be the fastest for a given problem. It will depend on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, the error goal, and whether the network is being used for pattern recognition (discriminant analysis) or function approximation (regression)

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

VIII. PREPARATION OF DATA BASE

The database used to train and test the network consist of 260 isolated handwritten letters. These alphabets were written by 10 different subject writers which include boys and girls of different age group. They were ask to write these letters using different writing medium i.e. ball-pen or pencil etc. Each writer was requested to write a set of 26 letters (A-Z). Out of these 260 characters, 208 samples were used along with their target outputs for the training of the network and remaining 52 samples were used for the testing of the network. The important feature of this data base is that both the training set and testing set contains numerous samples that are ambiguous and unclassifiable

IX. DERIVATION OF PATTERN VECTORS

The pattern vector corresponding to a particular digit is obtained by its corresponding character matrix. The character matrix is scanned row wise and the rows are concatenated with each other as they are scanned in order.

The following parameters are used in all the experiments:

Table – 1: Parameters v/s their values used in all learning processes.

Sr. No	Parameter	Value
1	Allowed Maximum Error For Learning	0.1, 0.01, 0.001, 0.00001
2	Performance Function	Mean Square Error (MSE)
3	Maximum Training Iterations	25,000
4	Initial Weights values	Randomly generated values between 0 and 1

X. COMPARISON TABLE OF DIFFERENT VARIANCES IN BACKPROPAGATION NEURAL NETWORK

S.No	Name of Algorithm	Error Goal	Iterations taken for training	Correct Recognition Rate (%)	Training Acronym
1	Batch Gradient Descent	1e-1	18182	91	TRAINGD
2	Variable Learning Rate	1e-2	13327	93	TRAINGDA
3	Adaptive Learning Rate with Momentum	1e-2	5829	93	TRAINGDX
4	Resilient Backpropagation	1e-3	7725	95	TRAINRP

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

S.No	Name of Algorithm	Error Goal	Iterations taken for training	Correct Recognition Rate (%)	Training Acronym
5	Fletcher-Reeves Update	1e-1	351	92	TRAINCGF
6	One-Step Secant	1e-3	2108	96	TRAINOSS
7	Polak- Ribiére Update	1e-3	1104	96	TRAINCGP
8	Powell-Beale Restarts	1e-3	371	96	TRAINCGB
9	Scaled Conjugate Gradient	1e-3	625	97	TRAINSCG
10	Levenberg-Marquardt	1e-5	2786	98	TRAINLM

XI. CONCLUSION

There are several algorithm characteristics that we can deduce from the experiments we have described. In general, on function approximation problems, for networks that contain up to a few hundred weights, the Levenberg-Marquardt algorithm will have the fastest convergence. This advantage is especially noticeable if very accurate training is required. In many cases, trainlm is able to obtain lower mean square errors than any of the other algorithms tested. However, as the number of weights in the network increases, the advantage of the trainlm decreases. The storage requirements of trainlm are larger than the other algorithms tested.

The trainrp function is the fastest algorithm on pattern recognition problems. However, it does not perform well on function approximation problems. Its performance also degrades as the error goal is reduced. The memory requirements for this algorithm are relatively small in comparison to the other algorithms considered.

The conjugate gradient algorithms, in particular trainscg, seem to perform well over a wide variety of problems, particularly for networks with a large number of weights. The SCG algorithm is almost as fast as the LM algorithm on function approximation problems (faster for large networks) and is almost as fast as trainrp on pattern recognition problems. Its performance does not degrade as quickly as trainrp performance does when the error is reduced. The conjugate gradient algorithms have relatively modest memory requirements.

The variable learning rate algorithm traingdx is usually much slower than the other methods, and has about the same storage requirements as trainrp, but it can still be useful for some problems. There are certain situations in which it is better to converge more slowly. For example, when using early stopping you may have inconsistent results if you use an algorithm that converges too quickly. We may overshoot the point at which the error on the validation set is minimized.

XII. REFERENCES

[1] V K Dhar1, A K Tickoo1, R Koul And B P Dubey. “Comparative performance of some popular artificial neural network algorithms on benchmark and function approximation problems” Astrophysical Sciences Division; Electronic Instruments & Services Division, Bhabha Atomic Research Centre, Mumbai. PRAMANA °c IndianAcademy of Sciences Vol. 74, No. 2| journal of physics pp. 307-324,February 2010

[2] V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis “Optimization Strategies And Backpropagation Neural Networks” University of Athens, Department of Informatics, University of Patras Artificial Intelligence Research Center, Athens, Greece 2009.

[3] S.SanthoshBaboo , P.Subashini, M.Krishnaveni, "Combining Self-Organizing Maps and Radial Basis Function Networks for Tamil handwritten Character Recognition" ICGST-GVIP Journal, ISSN: 1687-398X, Volume 9, Issue 4, August 2009
[4] Yash Pal Singh, V.S.Yadav, Amit Gupta, Abhilash Khare "Bi Directional Associative Memory Neural Network Method In The Character Recognition: By Yash Pal Singh, V.S.Yadav, Amit Gupta, Abhilash Khare, Journal Of Theoretical And Applied Information Technology Jatit (Page:382-386), 2005-2009
[5] V.S.Dhaka And M.P. Singh "Simulating Biological Neural Network Structure In Computers With Help Of Matlab For Handwriting Recognition Tasks", Department Of Computer Science,Icis, Dr. B. R. Ambedkar University, Agra, Asian J. Exp. Sci., Vol. 21, No.2, 2007.
[6] Kartalopoulos, S.V. Understanding Neural Networks and Fuzzy Logic- Basic Concepts and Applications, Prentice Hall, New-Delhi, (1996)
[7] .Soni M.G. "Isolated Handwritten Digit Recognition Using Artificial Neural Network", M.E. dissertation (Dr.V.S.Bansal) electrical engineering Dept., JNVUniversity, Jodhpur, 1995.

CITE AN ARTICLE

Gurusamy , V., & Nandhini, K. (2017). PERFORMANCE EVALUATION OF VARIANCES IN BACKPROPAGATION NEURAL NETWORK USED FOR HANDWRITTEN CHARACTER RECOGNITION. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY, 6*(11), 372-378.