Optical Character Recognition Via Multilayer Perceptron

Amir M. Mousavi _ 3971242026

***Abstract*-- Artificial neural networks are appearing as useful alternatives to traditional statistical modelling techniques in many scientific disciplines.We show that neural network classifiers with multilayer perceptron training can be applied efficiently to complex real-world classification problems such as the recognition of handwritten digits. This recognition study is used MLP and back propagation algorithm that are the standard algorithm for any supervised learning pattern recognition process. This article aims to provide a comprehensive tutorial and survey about MLP and back propagation algorithm . Finally, a comparison were done between performances of the three classifiers, which presented in the previous work and this NN.**.

***Index Terms*— Multilayer Perceptron (MLP), BackPropagation, Neural Network (NN), Learning Rate (LR),Momentum, Recognition , Supervised Learning , Optical Character Recognition (OCR).**

I. Introduction

Since the criticism of single-layer perceptrons, mainly for their limitation to building linear separation surfaces [11, more powerful neural network classifiers have been developed. The most popular network is the multilayer perceptron (MLP) trained by the backpropagation algorithm [2]. It has been shown in various papers that MLP's with a single hidden layer are universal classifiers, in the sense that they can approximate decision surfaces of arbitrary complexity, provided the number of neurons in the hidden layer is large enough (see for instance [3]). However, there is no simple rule which indicates how many hidden units are required for learning a given task. Moreover, limitations on hardware requirements or computation time may influence the choice of the classifier and favor classifiers with simpler structures and faster training than MLP's.

In the present study, the given dataset were used for OCR done on Persian handwritten digits. For this recognition MLP neural network were used. Finally, we did a comparison between performances of this NN and other classification (Bayesian, KNN, Parzen) that presented in the previous work. The rest of this paper is organized as follows. Section 2 presents OCR and the NN that were used for recognition. Experimental results and comparison are depicted in section 3. Section 4 concludes the paper.

II. Multilayer Perceptron

Environmental modelling involves using a variety of approaches, possibly in combination. Choosing the most suitable approach depends on the complexity of the problem being addressed and the degree to which the problem is understood. Assuming adequate data and computing resources and if a strong theoretical understanding of the problem is available then a full numerical model is perhaps the most desirable solution. However, in general, as the complexity of a problem increases the theoretical understanding decreases (due to ill-defined interactions between systems) and statistical approaches are required. Recently, the use of neural networks, and in particular the multilayer perceptron, have been shown to be effective alternatives to more traditional statistical techniques (Schalkoff, 1992). Primarily it has been shown (Hornik et al., 1989) that the multilayer perceptron can be trained to approximate virtually any smooth, measurable function. Unlike other statistical techniques the multilayer perceptron makes no prior assumptions concerning the data distribution. It can model highly non-linear functions and can be trained to accurately generalise when presented with new, unseen data. These features of the multilayer perceptron make it an attractive alternative to developing numerical models, and also when choosing between statistical approaches. As will be seen the multilayer perceptron has many applications in the atmospheric sciences. The multilayer perceptron consists of a system of simple interconnected neurons, or nodes, as illustrated in Fig. 2, which is a model representing a nonlinear mapping between an input vector and an output vector. The nodes are connected by weights and output signals which are a function of the sum of the inputs to the node modified by a simple nonlinear transfer, or activation, function. It is the superposition of many simple nonlinear transfer functions that enables the multilayer perceptron to approximate extremely non-linear functions. If the transfer function was linear then the multilayer perceptron would only be able to model linear functions. Due to its easily computed derivative a commonly used transfer function is the logistic function, as shown in Fig. 3. The output of a node is scaled by the connecting weight and fed forward to be an input to the nodes in the next layer of the network. This implies a direction of information processing, hence the multilayer perceptron is known as a feed-forward neural network. The architecture of a multilayer perceptron is variable but in general will consist of several layers of neurons. The input layer plays no computational role but merely serves to pass the input vector to the network. The terms input and output vectors refer to the

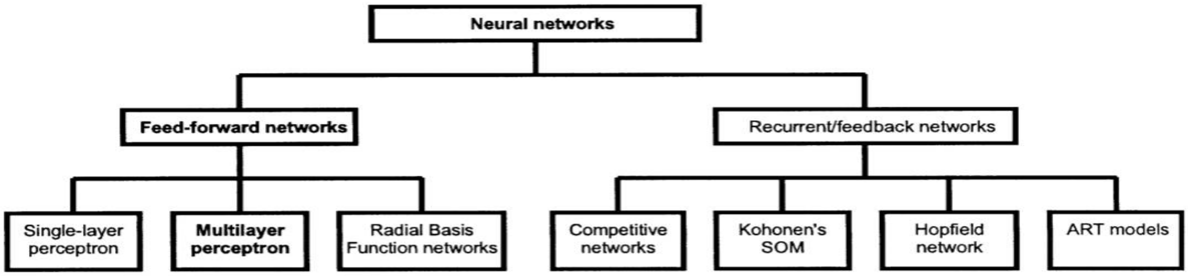

Fig. 1. A taxonomy of neural network architectures (after Jain *et al.*, 1996).

inputs and outputs of the multilayer perceptron and can be represented as single vectors, as shown in Fig. 2. A multilayer perceptron may have one or more hidden layers and finally an output layer. Multilayer perceptrons are described as being fully connected, with each node connected to every node in the next and previous layer. By selecting a suitable set of connecting weights and transfer functions, it has been shown that a multilayer perceptron can approximate any smooth, measurable function between the input and output vectors (Hornik et al., 1989). Multilayer perceptrons have the ability to learn through training. Training requires a set of training data, which consists of a series of input and associated output vectors. During training the multilayer perceptron is repeatedly presented with the training data and the weights in the network are adjusted until the desired input—output mapping occurs. Multilayer perceptrons learn in a supervised manner. During training the output from the multilayer perceptron, for a given input vector, may not equal the desired output. An error signal is defined as the difference between the desired and actual output. Training uses the magnitude of this error signal to determine to what degree the weights in the network should be adjusted so that the overall error of the multilayer perceptron is reduced. There are many algorithms that can be used to train a multilayer perceptron. Once trained with suitably representative training data the multilayer perceptron can generalize to new, unseen input data.

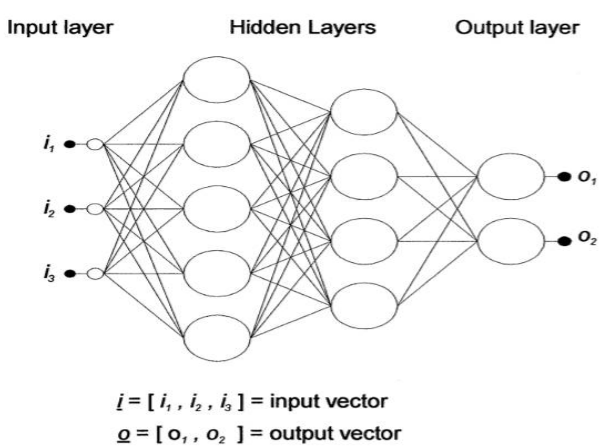

Fig. 2. A multilayer perceptron with two hidden layers.

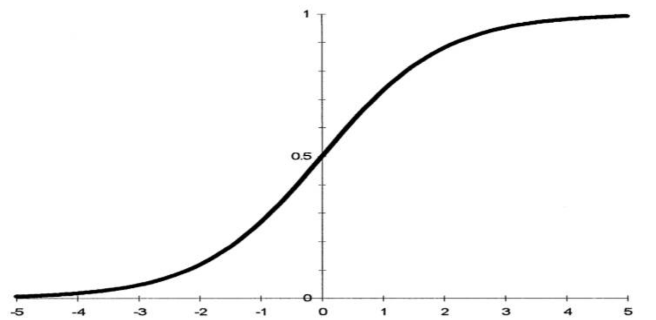

Fig. 3. The logistic function $y = 1/(1 - \exp(-x))$.

III. TRAINING A MULTILAYER PERCEPTRON—THE BACKPROPAGATION ALGORITHM

Training a multilayer perceptron is the procedure by which the values for the individual weights are determined such that the relationship the network is modelling is accurately resolved. At this point we will consider a simple multilayer perceptron that contains only two weights. For any combination of weights the network error for a given pattern can be defined. By varying the weights through all possible values, and by plotting the errors in three-dimensional space, we end up with a plot like the one shown in Fig. 4. Such a surface is known as an error surface. The objective of training is to find the combination of weights which result in the smallest error. In practice, it is not possible to plot such a surface due to the multitude of weights. What is required is a method to find the minimum point of the error surface. One possible technique is to use a procedure known as gradient descent. The backpropagation training algorithm uses this procedure to attempt to locate the absolute (or global) minimum of the error surface. The backpropagation algorithm (Rumelhart et al., 1986) is the most computationally straightforward algorithm for training the multilayer perceptron. Backpropagation has been shown to perform adequately in many applications; the majority of the applications discussed in this paper used backpropagation to train the multilayer perceptrons. A full mathematical derivation of this algorithm can be found in almost all neural network textbooks (e.g. Bishop 1995) so only the essential components of the algorithm will be discussed here. Backpropagation only refers to the training algorithm and is not another term for the multilayer perceptron or feed-forward neural networks, as is commonly reported. The weights in the network are initially set to small random values.

This is synonymous with selecting a random point on the error surface. The backpropagation algorithm then calculates the local gradient of the error surface and changes the weights in the direction of steepest local gradient. Given a reasonably smooth error surface, it is hoped that the weights will converge to the global minimum of the error surface. The backpropagation algorithm is summarized below. Implementation details can be found in most neural network books (e.g. Bishop, 1995).

i. initialize network weights,
ii. present first input vector, from training data, to the network,
iii. propagate the input vector through the network to obtain an output,
iv. calculate an error signal by comparing actual output to the desired (target) output,
v. propagate error signal back through the network,
vi. adjust weights to minimize overall error,
vii. repeat steps ii—vii with next input vector, until overall error is satisfactorily small.

The above implementation of the backpropagation algorithm is known as on-line training whereby the network weights are adapted after each pattern has been presented. The alternative is known as batch training, where the summed error for all patterns is used to update the weights. The benefits of each approach are discussed in Battiti (1992). In practice, many thousands of training iterations will be required before the network error reaches a satisfactory level—determined by the problem being addressed. As will be discussed later, training should be stopped when the performance of the multilayer perceptron on independent test data reaches a maximum, which is not necessarily when the network error is minimized. The error surface in Fig. 4 contains more than one minimum. It is desirable that the training algorithm does not become trapped in a local minimum. The backpropagation algorithm contains two adjustable parameters, a learning rate and a momentum term, which can assist the training process in avoiding this. The learning rate determines the step size taken during the iterative gradient descent learning process. If this is too large then the network error will change erratically due to large weight changes, with the possibility of jumping over the global minima. Conversely, if the learning rate is too small then training will take a long time. The momentum term is used to assist the gradient descent process if it becomes stuck in a local minimum. By adding a proportion of the previous weight change to the current weight change (which will be very small in a local minimum) it is possible that the weights can escape the local minimum.

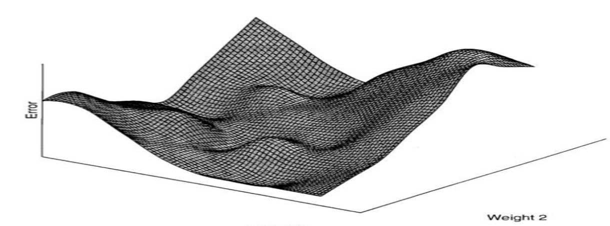

Fig. 4. An error surface for a simple multilayer perceptron containing only two weights.

IV. XOR GATE

In order to understand the back propagation we implemented XOR gate .

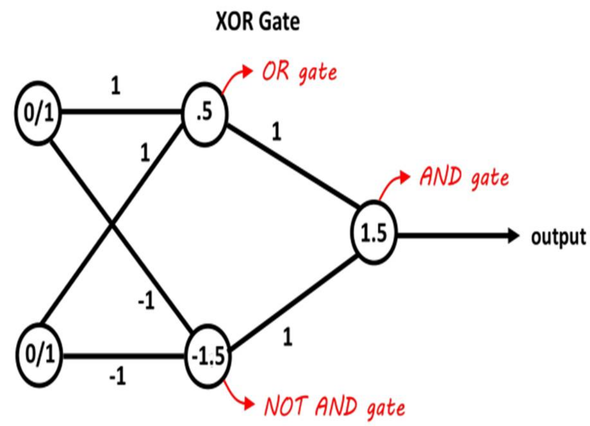

Fig.5

Here we have the result (learning rate :2 , iteration :1001):

```
output of iteration 8000
 [[0.00882965]
 [0.9913882 ]
 [0.99138366]
 [0.01012346]]
output of iteration 9000
 [[0.00828247]
 [0.99191601]
 [0.9919119 ]
 [0.00951337]]
output of iteration 10000
 [[0.00782418]
 [0.99235859]
 [0.99235482]
 [0.00900091]]
========================================
FINAL weight of hidden node:
 [[ 6.86734039  5.09196365]
 [ 6.85991811  5.09002221]
 [-3.10434523 -7.80386897]]
FINAL weight of output node:
 [[ 11.01303623]
 [-11.67691914]
 [ -5.16944871]]
```

V. OPTICAL CHARACTER RECOGNITION

Optical character recognition is a typical field of application for automatic classification methods. In addition to its practical interest (zip code recognition, automatic reading of bank checks, etc.), it exhibits all the typical problems encountered when dealing with classification: choice of the data representation, choice of a classifier of suitable type and structure, and supervised training of the classifier using a set of examples. In this paper, we focus on the recognition of

isolated handwritten digits, a task which is known to be difficult and which still lacks a technically satisfactory solution.
Technology nowadays produces a system enabling computer to read handwriting as an input, and processes it in order to provide useful information. This kind of system is called Optical Character Reader or Optical Character Recognition (OCR). An OCR system is able to read handwriting patterns and recognize them as characters corresponding with the intention of writing [4]. Handwritten digit recognition problem can be seen as a subtask of the more general OCR problem. In addition, handwritten digit recognition problem is usually used as a benchmark for comparing different classification techniques [5]. In OCR applications, handwritten character recognition, especially digit recognition, is dealt with in postal mail sorting, bank check processing, form data entry, etc. In recent decades many researchers worked on this topic (for example see (Liu, 2003; Mayraz and Hinton, 2002; Oliveira, 2002; Trier et al., 1996))

VI. Experimental Result

In this experiment, we used the given dataset, which includes about 11000 images of Persian single digits [1].
First we extracted 1000 images of numbers of 0 to 9, 900 for training and 100 for testing , considering that the size of data were not identical we resize them to 6×6 matrix.

A. Comparing Performance with MLP and other Classifiers:

In this part were done a comparison between MLP and the other classifiers, which presented at the previous work. This experiment were done for images of numbers of 1 and 5. As shown in table1, MLP was the best, and performance of train and validation vs. number of ephoc is shown in Fig(6).

	Mlp	1-nn	k-nn	parzen	Bayesian
performance	100	91	87	80	88

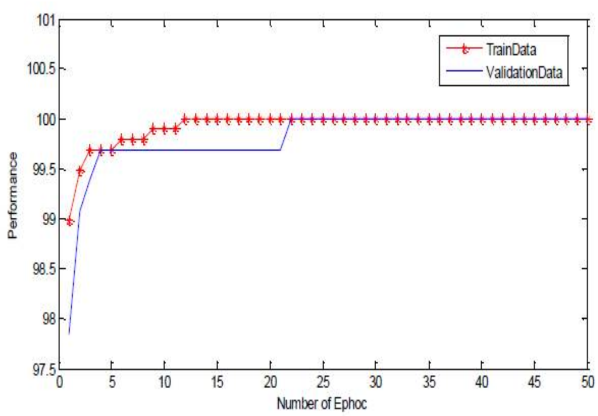

Fig. 6. Figure shows Performance of MLP for Different Ephocs with 100 Input Data and Hidden Neuron=45 and Validation Percent=20%

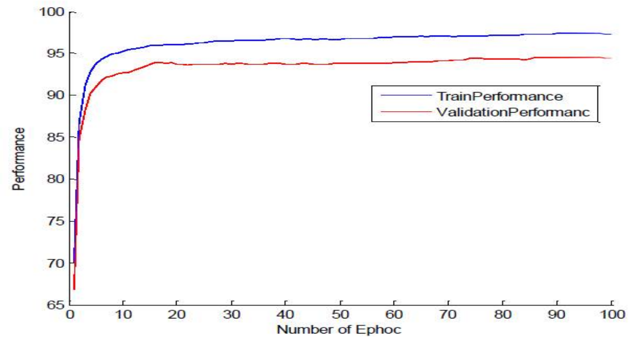

Figure 7. shows Performance of MLP with 1500 Input

VII. Conclusion

In the present study, the results obtained from previous work which done on the supervised classifiers were compared with MLP, the results of MLP was much better than other classifiers.
The results obtained from this experiment which were done on Persian digits of 0 to 9 were that by considering momentum rule results in increasing performance and also dynamic learning rate has better results than static learning rate and also the bigger the size of train data, the better the performance and the results very depend on the first weights.
The multilayer perceptron has been shown to be a useful tool for prediction, function approximation and classification. The practical benefits of a modelling system that can accurately reproduce any measurable relationship is huge. The benefits of the multilayer perceptron approach are particularly apparent in applications where a full theoretical model cannot be constructed, and especially when dealing with non-linear systems. The numerous difficulties in implementing, training and interpreting the multilayer perceptron must be balanced against the performance benefits when compared to more traditional, and often inappropriate, techniques.

VIII. References

[1] Knerr, S., Personnaz, L., & Dreyfus, G. (1992). Handwritten digit recognition by neural networks with single-layer training. IEEE Transactions on Neural Networks

[2] . Minsky and S. Papert, Perceptrons Cambridge MA: MIT Press, 1969.

[3] D. E Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” in Parallel Distributed Processmg. Cambridge, MA: MIT Press, 1986, vol 1

[4] K. Hornik, “Multilayer feedforward networks are universal approximators,” Neural Networks, vol 2, pp. 359-366, 1989.

[5] Badran, F. and Thiria, S. (1991) Wind ambiguity removal by the use of neural network techniques. Journal of Geophysical Research 96 (C11), 20521—20529

[6] Bankert, R. L. (1994) Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network. Journal of Applied Meteorology, 33, 909—918

[7] Gardner, M. ., & Dorling, S. . (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric Environment

[8] https://www.youtube.com/watch?v=c8BQqxL5wr4

[9] https://www.youtube.com/watch?v=c8BQqxL5wr4