# Backpropagation: In Search of Performance Parameters

ANIL KUMAR ENUMULAPALLY, LINGGUO BU,
and KHOSROW KAIKHAH, Ph.D.
Computer Science Department
Texas State University-San Marcos
San Marcos, TX-78666 USA
ae1049@TxState.edu, lb40@TxState.edu, kk02@TxState.edu

Abstract: - This work is an extensive study of the backpropagation network based on a new visual tool, Equal Opportunity for Recognition (EOR) for all inputs to be recalled, which is used to evaluate the overall network performance, in particular, its generalization capabilities. The new procedure, EOR, is used as a means to assess the effect of other system parameters.

Keywords: backpropagation, Network evaluation, generalization, EOR, Processing Elements (PEs) and parameters.

## 1 Introduction

A Backpropagation Network is a multiplayer, associative, and feed forward neural network that features supervised learning using gradient descent training procedure. Back Propagation is widely used in applications involving pattern recognition because of its powerful capability of generalization.

While its system structure and learning algorithm are well documented, there exist no mathematical criteria to assess the performance, particularly the generalization capabilities, of the network with respect to such network parameters as number of PEs on the hidden layer, the mean squared error, learning rates, initialization of weights and thresholds.

Searching for a measure of system performance, we proposed a visual method, the EOR plot, which can be used as an indicator of the overall system performance. With the aid of EOR plotting, we further studied the various parameters of the system as they relate to the overall system behavior, including MSE, hidden layer size, learning rates, weight and threshold initialization, and threshold updating.

## 2 Description of Experiments

To study the various properties of a backpropagation network, we started with 26 capital letters of the English alphabet, each of which is represented on a 24 by 24 grid as follows. Each grid was converted to a binary vector of 576 elements.

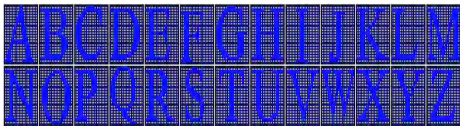


**Fig. 1: Input patterns**

Each binary vector is associated with a 8-bit ASCII code corresponding to the English letter.

Since one hidden layer is generally sufficient for most applications [4], we have designed a backpropagation network of three layers, an input layer with 576 PEs, an output layer with 8 PEs, and a hidden layer with a varying number of PEs.

In the light of the EOR (Equal Opportunity for Recognition) plot as presented below, we studied all parameters of a backpropagation network based on our implementation of the network on Mathematica 4.

## 3 EOR

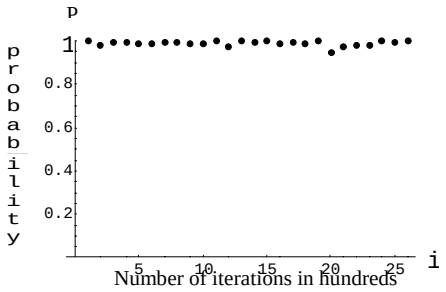
Backpropagation is a simple and powerful algorithm, yielding satisfactory results if properly implemented. Mathematical criteria, however, are still to be found that can be employed to evaluate system performance with respect to such a network parameters as the MSE, hidden layer size, initial weights, and learning rates. Many rules for choosing hidden

layer size have been proposed, however none of them seem to be superior and all are result of some empirical conjure. To guarantee the applicability of a network, however, some measures have to be taken to assess system performance. To avoid overtraining, for example, constant monitoring on system performance is necessary, including the incorporation of test data in the process of training

Given a specific application, such as the recognition of the 26 capital English letters, noise reduction and generalization capabilities in the presence of random noise are among essential requirements of the network. In other words, we need to prove the probabilistic performance of a network so that, first, all input patterns can be recovered successfully with an equal opportunity, and second, the probability that an individual input can be recovered should meet the requirements of the application. Both factors are related to all the parameters of a network.

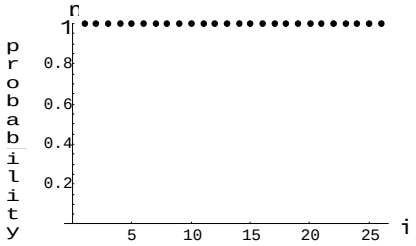
In the absence of a mathematical description, we propose the EOR plot (Equal Opportunity for Recognition) as a visual, probabilistic method to evaluate system performance. Given a set of system parameters, including MSE, initial weights, thresholds, learning rates, and hidden layer size, we train the network and estimate the probability of each individual input pattern correctly recognized at a specified rate of random noise. The latter could be done by repeating the recall process on a sufficiently large number of randomly corrupted inputs and monitoring the behavior of the network. After all individual inputs have been processed, the performance of the network can be analyzed using EOR plots.

Using 9 hidden layer neurons with a range of –0.001 to +0.001 for weight and threshold random initialization, a learning rate of 0.1, an MSE of 0.005, and random noise rates of 10% and 5%, respectively, we obtain the following EOR plots as an estimation of the network performance.



Number of iterations in hundreds

Fig. 2(a)



Number of iterations in hundreds

Fig. 2(b)

**Fig. 2: EOR plots for 10% & 5% noise respectively**

According to the two EOR plots, with 10% random noise, each input pattern can be correctly recognized with a probability of over 90% in spite of the slight variations; with 5% random noise, all patterns can be recognized. Fig 3 depicts sample letters with 10% random noise.

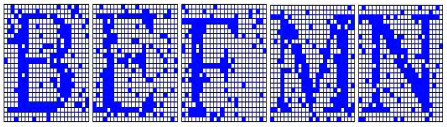


**Fig. 3: Specimen with 10% random noise**

All corrupted patterns can be correctly recovered with a probability of more than 90%. As shown by our experiments, EOR plots can be used as an objective description of system performance. EOR plots can be utilized in analyzing other network parameters.

# 4 Results and Analysis

## 4.1 Mean Squared Error (MSE)

MSE is generally used as an indicator of network convergence. However, MSE is not a sufficient factor and other network characteristics need to be considered.

First, we will show that MSE is not always a sufficient descriptor of system performance. Using 8 hidden layer PEs and an MSE of 0.005 with a different range for random initialization of weights and thresholds, we obtained the following 5% random noise EOR plots. In figure

4(a), the range of weight and threshold initialization is -1.0 to +1.0; in figure 4(b), the range of weights and threshold initialization is -0.05 to +0.05.

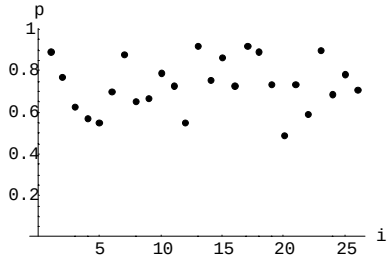



Fig. 4(a)

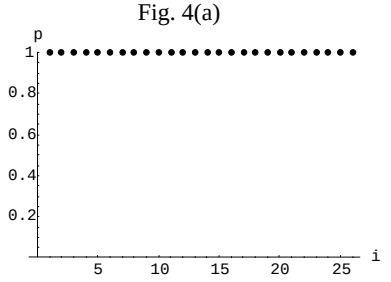



Fig. 4(b)

**Fig. 4: EOR plot for different weight and threshold initializations**

Second, given a specific topology of a network, a small MSE does not always yield better system performance. As shown by our experiment, after a certain point, the EOR plot remains virtually the same without evidence of over-fitting. The following results are obtained using 8 hidden layer nodes, a learning rate of 0.1, and a range of –0.05 to +0.05 for weight and threshold random initialization, at an MSE of 0.5, 0.05, and 0.001.

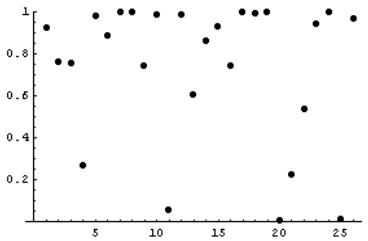



Fig. 5(a): MSE of 0.5

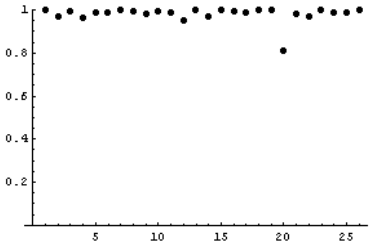



Fig. 5(b): MSE of 0.05

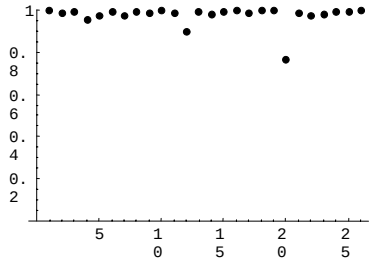



Figure 5(c): MSE of 0.001

**Fig 5: EOR plots for different MSE values**

Therefore, while MSE is an important factor of a backpropagation network, it is not sufficient for drawing conclusions about system performance. Other factors, including weight initialization and size of the hidden layer also play an important role.

## 4.2 Weight Initialization

In a three-layer network, there are two weight sets. As a general rule, the weights should be randomly initialized to small values to avoid system oscillation and as justified by the derivative of the activation function. We started with a range of –1.0 to +1.0 and gradually reduced the range. We observed that smaller random initialization yields a better performance. For the following graphs, a network with 8 hidden layer nodes used, together with an MSE of 0.05, learning rate of 0.1 and different ranges for weight and threshold initialization.

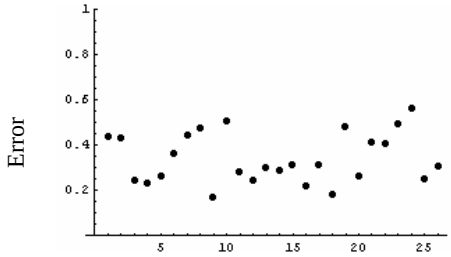



Fig. 6(a) weight and threshold initialization –1 to +1

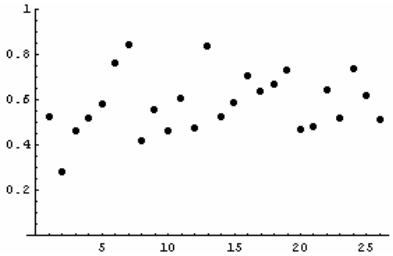



Fig. 6(b) weight and threshold initialization -0.5 to +0.5

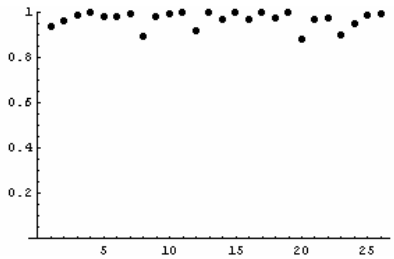



Fig. 6(c) weight and threshold initialization –0.1 to +0.1

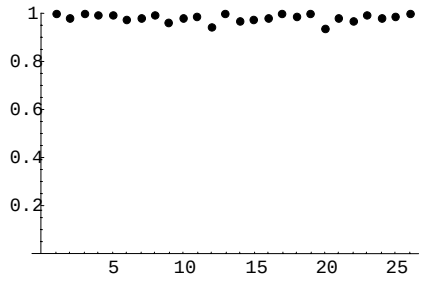



Fig. 6(d) weight and threshold initialization –0.001 to +0.001

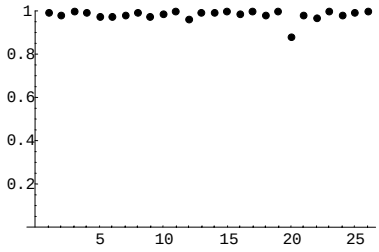



Fig. 6(e) weight and threshold initialization -0.00001 to +0.00001

**Fig. 6: EOR plots using various ranges for weight and threshold random initialization –1 to +1, -0.5 to +0.5, -0.1 to +0.1, -0.001 to +0.001, and -0.00001 to +0.00001, respectively.**

Although the weights could all be initialized to zero, this would result in a highly symmetrical network and is thus created therefore; it is not a good choice for network design. This emphasizes the statement made by Ramelhart et al.[6] “Initial weights of exactly 0 cannot be used, since symmetries in the environment are not sufficient to break symmetries in initial weights”.

## 4.3 Number of PEs on the Hidden Layer

To study the effect of the number of PEs in the hidden layer on system performance, we performed a series of experiments where all weights and thresholds were randomly initialized between –0.05 and +0.05, with a fixed MSE of 0.005 and a learning rate of 0.1. When the weights are initialized to very small values, the same MSE yields similar system performance regardless of the range of initialization, and thus can be used to compare the effect of number of PEs on the hidden layer.

With a small number of PEs on the hidden layer, compared to the input and output layers the learning curve exhibits a great deal of fluctuations and does not converge to the specified MSE. This implies the network does not have enough learning capacity i.e. memory with 3 hidden layer PEs, we observed the following results:

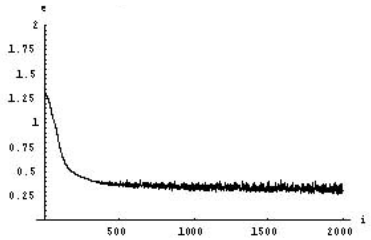



Fig. 7(a) Learning curve

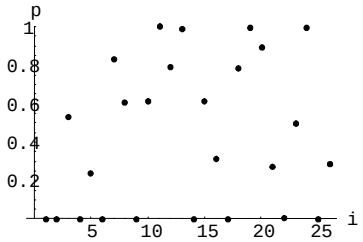



Fig. 7(b) EOR plot

**Fig. 7: Learning curve and EOR plot for a network with 3 PEs on hidden layer**

With more PEs on the hidden layer, more input patterns can be correctly recovered. Once the number of hidden layer PEs reaches an ideal range the system performance stabilizes and shows very little improvement with an addition of new PEs. The following EOR plots are based on 5, 7, 9, 12, 24, 36, 48, 100 PEs, respectively.

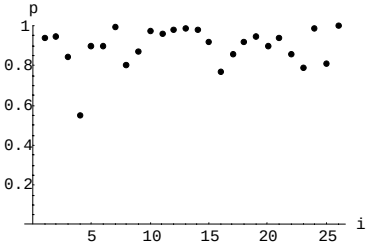



Fig. 8(a) EOR plot for 5 hidden layer PEs

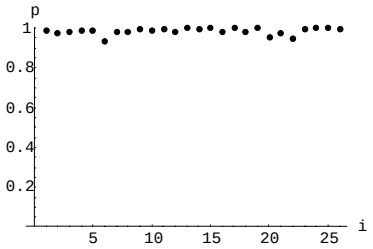



Fig. 8(b) EOR plot for 7 hidden layer PEs

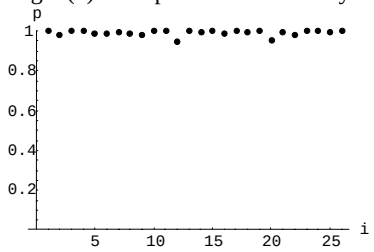



Fig. 8(c) EOR plot for 24 hidden layer PEs

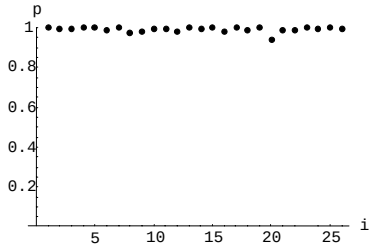



Fig. 8(d) EOR plot for 48 hidden layer PEs

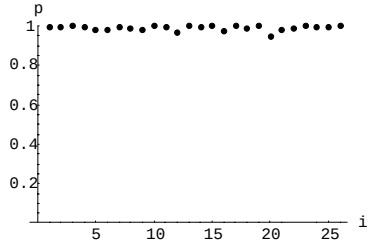



Fig. 8(e) EOR plot for 100 hidden layer PEs

**Fig. 8: EOR plot for a network with 5, 7, 24, 48, 100 PEs on hidden layer, respectively.**

We observed that using a fixed MSE, the number of iterations is related to the number of PEs on the hidden layer by the following curve.

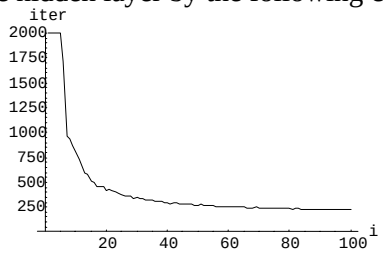



**Fig. 9: Relationship between hidden layer size and number of training iterations**

Hidden layer PE's are the feature extractors. As the hidden layer size increases, for a fixed error, the number of iterations to train the network converges to a value and will not oscillate. This tells us after certain limit the hidden layer size does not have any effect on the number of iterations.

Although the increasing the hidden layer size brings down the number of iterations there may not be much improvement in the total training time.

## 4.4 Learning Rates:

While learning rates are generally taken to be small numbers between 0 and 1, there is no criterion governing the selection of a learning rate. If it is too small, the error correction is trivial and the network does not learn well, with little chance of getting out of a local minimum; if it is too large, the learning process is one of oscillation, with little chance of convergence to the necessary MSE. The training of a network is aimed at its generalization performance, which is achieved by system convergence, the speed of which is adjusted by the learning rates. To appreciate the effects of large learning rates, consider the learning curve of a network with 9 hidden layer PEs, a weight and threshold initialization range of –0.05 to +0.05, and a learning rate of 5, as depicted in fig.10.

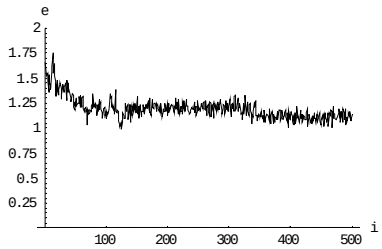



**Fig. 10: Learning curve at a high learning rate (5).**

To assess the effect of learning rate on system performance, we used a network with 16 hidden layer PEs, a range of –0.05 to +0.05 for weight and threshold initialization, an MSE of 0.005, and various learning rates. With a learning rate of 0.001 and 10% random noise, the EOR plots are as follows, corresponding to the learning rate of 0.01, 0.2, 0.8, and 2. Network did not converge with a Learning rate of .001.

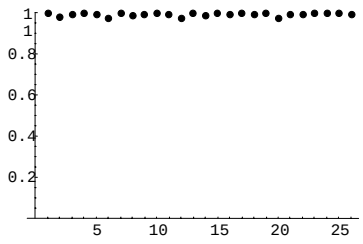



Fig. 11(a) EOR plot at learning rate of 0.01

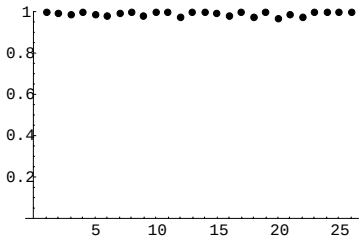



Fig. 11(b) EOR plot at learning rate of 0.8

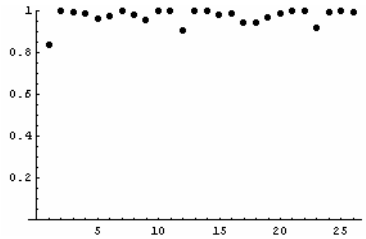



Fig. 11(d) EOR plot at learning rate of 2

**Fig. 11: EOR plots at the learning rates of 0.01, 0.8, and 2, respectively.**

### 4.5 Thresholds

Thresholds, or bias, can be used on both the hidden layer and the output layer PEs, to fine-tune the system convergence. Each PE on the hidden and output layer can a threshold value, which is updated directly based on the delta value computed for that PE. The threshold updating not only speeds up system convergence, but also it is potentially helpful in smoothing out system fluctuations that might be hard to deal with using weight updating alone.

$$o_k = f(\sum_{i=1}^{n} a_i w_{ik} - \theta_k), \qquad (1)$$

Where $O_k$ is the output of the $k^{th}$ node on the hidden or output layer and $\theta_k$ is the corresponding threshold and $f$ is the sigmoid function. If $\delta_k$ is the delta value for the node, $\theta_k$ should be updated as follows:

$$\theta_k(t) = \theta_k(t-1) - \varepsilon\delta_k \qquad (2)$$

where ε is the threshold learning rate, $\delta_k$ is the delta value, and $\theta_k$ is the threshold value.

## 5 Conclusions

As there are no formulas that can be readily used to evaluate the performance of a backpropagation network, the Equal Opportunity for Recognition (EOR) plots represent a practical tool for system assessment with respect to the application conditions. As a probabilistic method, not only can it be used to describe system performance, it can also be incorporated into the recall process for demanding pattern recognitions. The EOR, has shown a great promise in finding the optimal initial conditions for our Neural Network. The future work can be in the direction of finding general principles, to design a backpropagation network with near optimal initial conditions, using EOR.

*References:*

[1] Freeman, James A. Simulating Neural Networks. Addison-Wesley, 1994.

[2] McAuley, Devin. "The backpropagation network: learning by example", 1997.

[3] Mehrotra, Krishan, et al. Elements of artificial neural networks, Cambridge, MIT Press, 1997.

[4] Sureerattanan, Songyot, et al, "New developments on backpropagation network training", IEICE Trans., vol. E83-A, No. 6, pp. 1032-1039, June, 2000

[5] Back Propagation is Sensitive to Initial Conditions (1990) -John F. Kolen, Jordan B. Pollack

[6] Learning Representation by Back-Propagating Errors. *Nature* **323**:533-536. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986.

[7] Sarle, Warren S. ftp://ftp.sas.com/pub/neural, 2002.