**Name: Deepthi Thalagundamatada**

**Student email ID: dt7637@uncw.edu**

**CSC515: Artificial Intelligence**

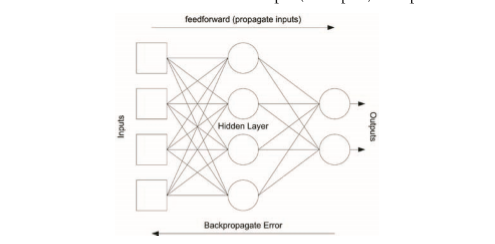
**Project Report: Back propagation**

**Programming Project 1 (Backpropagation):**Use the backpropagation algorithm to train a network to recognizecharacter representations on a 5x5 grid of pixels. Use a representation of the characters +, -, \, /, X, and | on a 5x5 grid. Test the performance of the network using “noisy” representations of the six characters.

**Back propagation algorithm:**

In Artificial intelligence in Back propagation algorithm is one of the supervised learning algorithm used on multilayer feed-forward networks and backpropagate the error to adjust the weights to provide the correct output after training the network.

Using Back propagation algorithm, the network model produces the output by adjusting the internal layer weights. As it is supervised learning algorithm we must train the network from training data set, then the error is calculated from the expected output and the actual output we have got, and weights are adjusted accordingly so the modified system recognizes the pattern of inputs trained.



**Network Structure and Inputs:** There are 4 layers: Input Layer, Hidden Layers and Output Layer .

The Input Layer consists of the 25 Neurons which takes 5x5 grid of characters as input.

**3 the Hidden Layers** so that it can identify 5 symbols.

Also, the Output Layer consists of 3 neurons which can perfectly represent 5 different symbols

* [0,0,0] as SYMBOL ‘+‘
* [0,0,1] as SYMBOL ‘-‘
* [0,1,0] as SYMBOL ‘x‘
* [0,1,1] as SYMBOL ‘\‘
* [1,0,0] as SYMBOL ‘/‘

sigmoid function: 𝑦 = 1 1+exp(−𝑧) is used for nonlinear neuron activation function.

Bias term =1 (equal to 1)

The network is trained with trainable weights from the network structure.

**Training SET:**

I am training the network to recognize symbols ‘+’, ‘-’ ,’X’,’/’,’\’. To train the network to produce error signal we will use another 1 character ‘Z’. We are interested in checking the response of the network to errors on the characters which were not involved in the training procedure. The characters to be recognized are given on 5 × 5 grid. Each set to either 0 or 1.

'SYMBOL +'

[0, 0, 1, 0, 0,

0, 0, 1, 0, 0,

1, 1, 1, 1, 1,

0, 0, 1, 0, 0,

0, 0, 1, 0, 0]

'SYMBOL -'

[0, 0, 0, 0, 0

0, 0, 0, 0, 0

1, 1, 1, 1, 1

0, 0, 0, 0, 0

0, 0, 0, 0, 0]

'SYMBOL X'

[1, 0, 0, 0, 1

0, 1, 0, 1, 0

0, 0, 1, 0, 0

0, 1, 0, 1, 0

1, 0, 0, 0, 0]

'SYMBOL /'

[0, 0, 0, 0, 1

0, 0, 0, 1, 0

0, 0, 1, 0, 0

0, 1, 0, 0, 0

1, 0, 0, 0, 0]

'SYMBOL \'

[1, 0, 0, 0, 0

0, 1, 0, 0, 0

0, 0, 1, 0, 0

0, 0, 0, 1, 0

0, 0, 0, 0, 1];

To train the network to recognize the symbols I have applied the corresponding 5×5 grids in the form of 1×25 vectors to the input of the network.

The symbol was considered recognized if both outputs of the network were no more than 0.1 off their respective desired values.

**Recognition of symbol with Noise:**

I have introduced z character as noise where it has / pattern on it

'SYMBOL Z'

[ 1, 1, 1, 1, 1

0, 0, 0, 1, 0

0, 0, 1, 0, 0

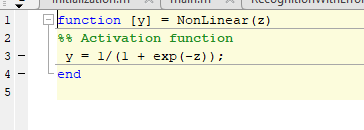
0, 1, 0, 0, 0

1, 1, 1, 1, 1]

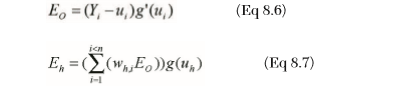
**Code snippets:**

**Activation function which is used to determine the learning rate to train the network.**





**Training network by using weights and correcting the weight:**



RateTemp = Rate;

multi = multi + Fi(i+1,k) \* w{i+1}(k,j);

end

end

Fi(i,j) = y(i,j) \* (1-y(i,j)) \* multi;

for k = 1:1:config(i)

% Counting input to each layer(number of neurons in the previous layer)

if (i == 1)

% If it is a first layer

prevOutput = Exemplar(in,k);

else

prevOutput = y(i-1,k);

end

w{i}(j,k) = w{i}(j,k) + Rate \* Fi(i,j)\* prevOutput;

end

% weight correction

w{i}(j,config(i)+1) = w{i}(j,config(i)+1) + Rate \* Fi(i,j);

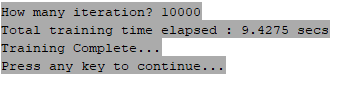
**Root mean square error function:**



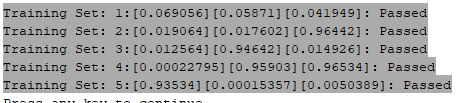
**Results:**

The initial learning rate 𝜂 is set to 1.5.

10000 iterations took place in about 9.425 secs.

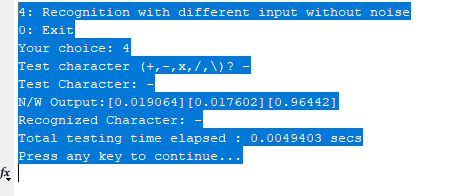


**Output of the first layer after applying weights:**

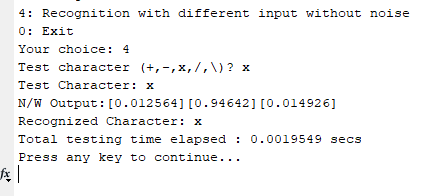


**Recognizing the characters with different inputs:**

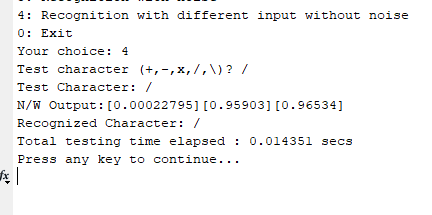
**FOR symbol ‘-‘:**



**For Symbol : X**



**For symbol ‘/’ :**



**Recognition of symbol with Noise:**

I have introduced z character as noise where it has / pattern on it I have introduced one bit noise to test it. Here is the result if one bit of noise is introduced:

