Assignment -8

Title: program for creating a back propagation feed-forward neural network.

Aim: Write a python program for creating a back propagation feed-forward neural network

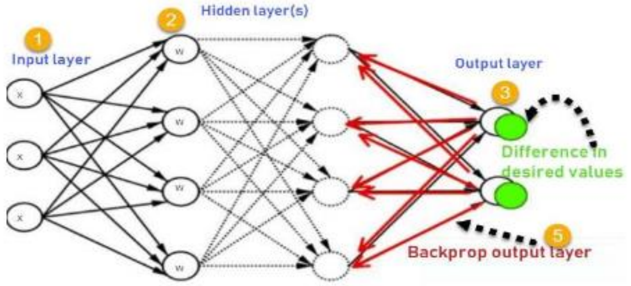
Theory:

Back propagation Neural Network:

Back propagation is a widely used algorithm for training feed forward neural networks. It computes the gradient of the loss function with respect to the network weights. It is very efficient, rather than naively directly computing the gradient concerning each weight. This efficiency makes it possible to use gradient methods to train multi-layer networks and update weights to minimize loss; variants such as gradient descent or stochastic gradient descent are often used.

The back propagation algorithm works by computing the gradient of the loss function with respect to each weight via the chain rule, computing the gradient layer by layer, and iterating backward from the last layer to avoid redundant computation of intermediate terms in the chain rule.

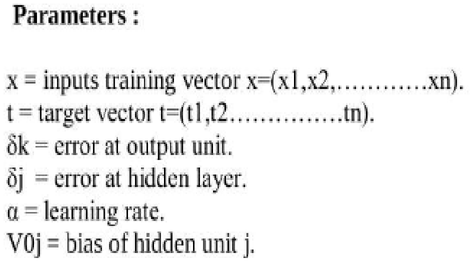
Backpropagation Algorithm:



Step 1: Inputs X, arrive through the preconnected path.
Step 2: The input is modeled using true weights W. Weights are usually chosen randomly.
Step 3: Calculate the output of each neuron from the input layer to the hidden layer to the output layer.
Step 4: Calculate the error in the outputs

Back propagation Error= Actual Output - Desired Output

Step 5: From the output layer, go back to the hidden layer to adjust the weights to reduce the error.
Step 6: Repeat the process until the desired output is achieved



Parameters :

x = inputs training vector x=(x1,x2,…………xn).
t = target vector t=(t1,t2……………tn).
δk = error at output unit.
δj = error at hidden layer.
α = learning rate.
V0j = bias of hidden unit j.

Training Algorithm :

Step 1: Initialize weight to small random values.

Step 2: While the stepsstopping condition is to be false do step 3 to 10.

Step 3: For each training pair do step 4 to 9 (Feed-Forward).

Step 4: Each input unit receives the signal unit and transmits the signal xi signal to all the units.

Step 5 : Each hidden unit Zj (z=1 to a) sums its weighted input signal to calculate its net input

$$zinj = v0j + \Sigma xivij \quad (i=1 \text{ to } n)$$

Applying activation function zj = f(zinj) and sends this signal to all units in the layer about i.e output units

For each output l=unit yk = (k=1 to m) sums its weighted input signals.

$$yink = w0k + \Sigma\ ziwjk \quad (j=1 \text{ to } a)$$

and applies its activation function to calculate the output signals.

$$yk = f(yink)$$

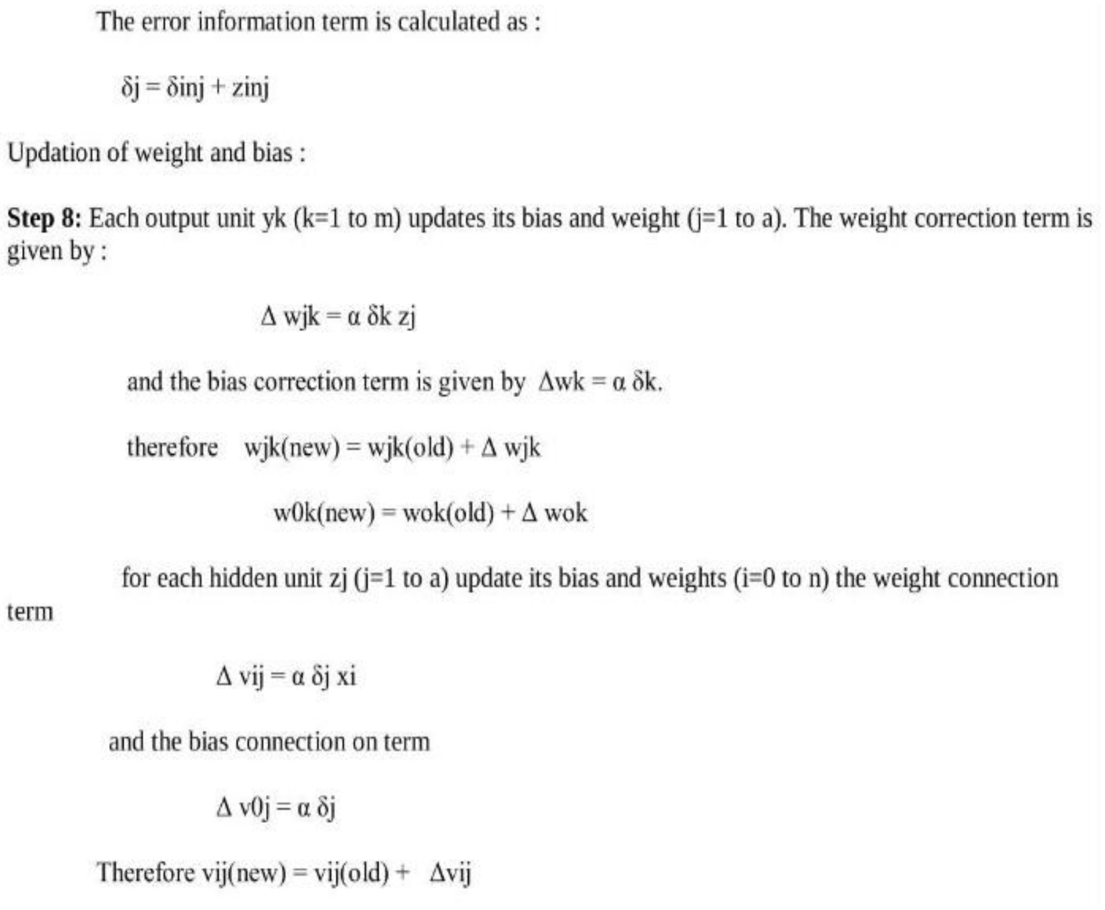
Backpropagation Error :

Step 6: Each output unit yk (k=1 to n) receives a target pattern corresponding to an input pattern then error is calculated as:

$$\delta k = (tk - yk) + yink$$

Step 7: Each hidden unit Zj (j=1 to a) sums its input from all units in the layer above

$$\delta inj = \Sigma\ \delta j\ wjk$$



The error information term is calculated as :

$$\delta j = \delta inj + zinj$$

Updation of weight and bias :

Step 8: Each output unit yk (k=1 to m) updates its bias and weight (j=1 to a). The weight correction term is given by :

$$\Delta\ wjk = \alpha\ \delta k\ zj$$

and the bias correction term is given by $\Delta wk = \alpha\ \delta k$.

therefore $wjk(new) = wjk(old) + \Delta\ wjk$

$$w0k(new) = wok(old) + \Delta\ wok$$

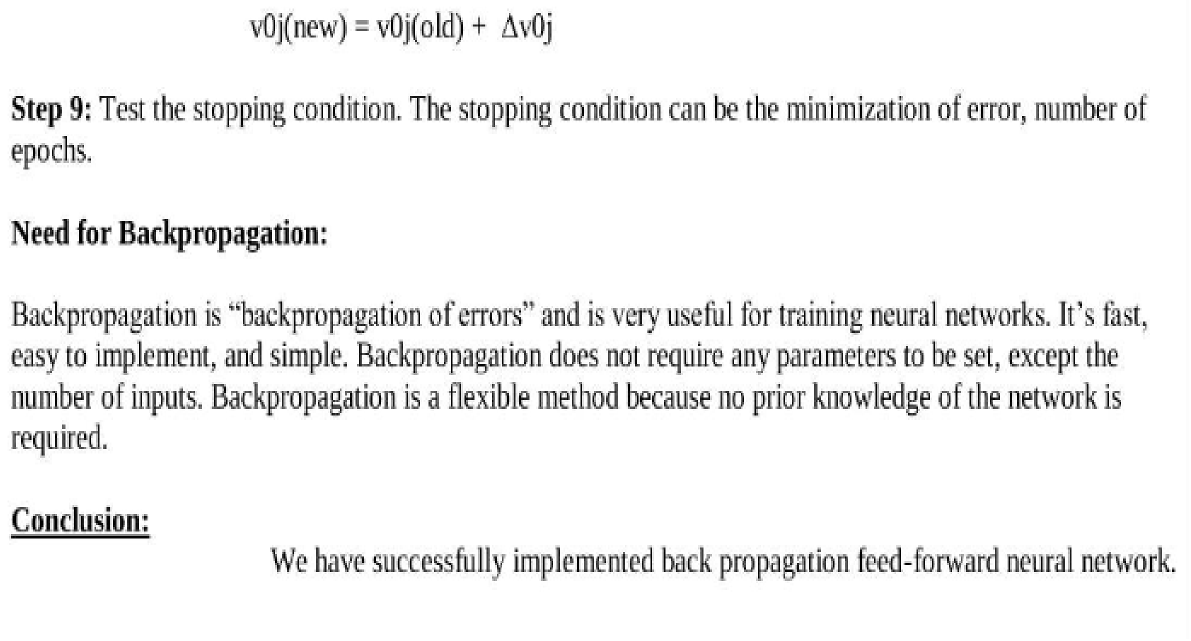
for each hidden unit zj (j=1 to a) update its bias and weights (i=0 to n) the weight connection term

$$\Delta\ vij = \alpha\ \delta j\ xi$$

and the bias connection on term

$$\Delta\ v0j = \alpha\ \delta j$$

Therefore $vij(new) = vij(old) + \Delta vij$



$$v0j(new) = v0j(old) + \Delta v0j$$

Step 9: Test the stopping condition. The stopping condition can be the minimization of error, number of epochs.

Need for Backpropagation:

Backpropagation is “backpropagation of errors” and is very useful for training neural networks. It’s fast, easy to implement, and simple. Backpropagation does not require any parameters to be set, except the number of inputs. Backpropagation is a flexible method because no prior knowledge of the network is required.

Conclusion:

We have successfully implemented back propagation feed-forward neural network.

Questions:

1. Explain Translation System for Face-To-Face Dialog & Intelligent Help Systems Using ANN.
2. Explain Building Blocks of CNNs.
3. Explain Architecture of CNN.
4. Explain Convolution/ Pooling layers of CNN.