5.4 ERROR BACKPROPAGATION

Backpropagation is a training method used for a multi-layer neural network. It is also called the generalized delta rule. It is a gradient descent method which minimizes the total squared error of the output computed by the net.

The backpropagation algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent. The weights that minimize the error function is then considered to be a solution to the learning problem

Backpropagation is a systematic method for training multiple layer ANN. It is a generalization of Widrow-Hoff error correction rule. 80 % of ANN applications uses backpropagation.

Fig. shows backpropagation network.

Consider a simple neuron:

a. Neuron has a summing junction and activation function.

b. Any non-linear function which differentiable everywhere and increases everywhere with sum can be used as activation function

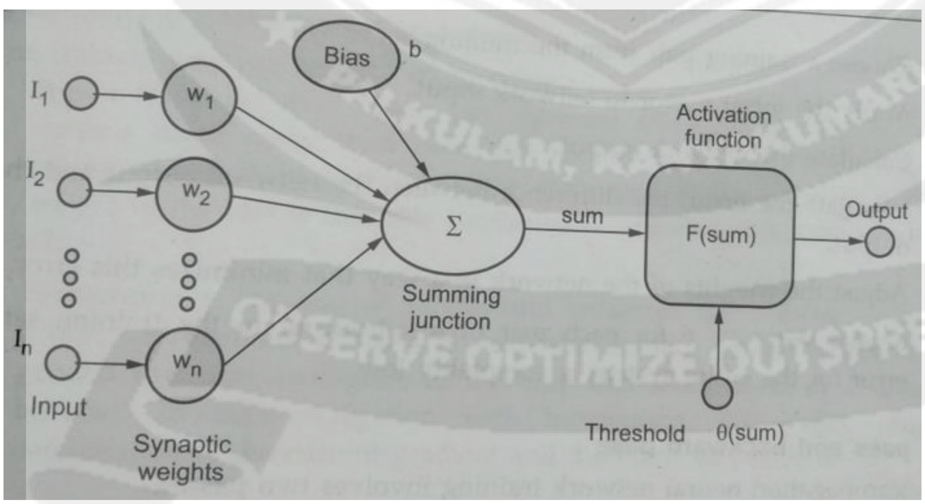

Fig : Backpropagation network

c) Examples Logistic function, Arc tangent function. Hyperbolic tangent activation function. These activation function makes the multilayer network to have greater representational power than single layer network only when non-linearity is introduced.

Need of hidden layers:

1. A network with only two layers (input and output) can only represent the input with whatever representation already exists in the input data.

2. If the data is discontinuous or non-linearly separable, the innate representation is inconsistent, and the mapping cannot be learned using two layers (Input and Output).

3. Therefore, hidden layer(s) are used between input and output layers

• **Weights** connects unit (neuron) in one layer only to those in the next higher layer. The output of the unit is scaled by the value of the connecting weight, and it is fed forward to provide a portion of the activation for the units in the next higher layer.

- Backpropagation can be applied to an artificial neural network with any number of hidden layers. The training objective is to adjust the weights so that the application of a set of inputs produces the desired outputs.
- Training procedure The network is usually trained with a large number of input-output pairs.

1. Generate weights randomly to small random values (both positive and negative) to ensure that the network is not saturated by large values of weights.

2. Choose a training pair from the training set.

3. Apply the input vector to network input

4. Calculate the network output.

5. Calculate the error, the difference between the network output and the desired output

6. Adjust the weights of the network in a way that minimizes this error.

7. Repeat steps 2-6 for each pair of input-output in the training set until the error for the entire system is acceptably low.

Forward pass and backward pass:

Backpropagation neural network training involves two passes,

1. In the forward pass, the input signals moves forward from the network input to the output

2 In the backward pass, the calculated error signals propagate backward through the network, where they are used to adjust the weights

3. In the forward pass, the calculation of the output is carried out, layer by layer, in the forward direction. The output of one layer is the input to the next layer

- In the reverse pass,

a. The weights of the output neuron layer are adjusted first since the target value of each output neuron is available to guide the adjustment of the associated weights, using the delta rule.

b. Next, we adjust the weights of the middle layers. As the middle layer neurons have no target values, it makes the problem complex.

- **Selection of number of hidden units**: The number of hidden units depends on the number of input units

1. Never choose h to be more than twice the number of input units.

2. You can load p patterns of 1 elements into log, p hidden units.

3. Ensure that we must have at least 1/e times as many training examples.

4. Feature extraction requires fewer hidden units than inputs.

5. Learning many examples of disjointed inputs requires more hidden units than inputs.

6. The number of hidden units required for a classification task increases with the number of classes in the task. Large networks require longer training times.

Factors influencing Backpropagation training

The training time can be reduced by using

1. Bias: Networks with biases can represent relationships between inputs and outputs more easily than networks without biases, Adding a bias to each neuron is usually desirable to offset the origin of the activation function. The weight of the bias is trainable similar to weight except that the input is always +1

2. Momentum: The use of momentum enhances the stability of the training process. Momentum is used to keep the training process going in the same general direction analogous to the way that momentum of a moving object behaves. In backpropagation with momentum, the weight change is a combination of the current gradient and the previous gradient.

Advantages and Disadvantages

Advantages of backpropagation:

1. It is simple, fast and easy to program

2. Only numbers of the input are tuned and not any other parameter.

3. No need to have prior knowledge about the network

4. It is flexible.

5. A standard approach and works efficiently.

6. It does not require the user to learn special functions.

Disadvantages of backpropagation:

1. Backpropagation possibly be sensitive to noisy data and irregularity.

2. The performance of this is highly reliant on the input data.

3. Needs excessive time for training.

4. The need for a matrix-based method for backpropagation instead of mini-batch.