INTRODUCTION

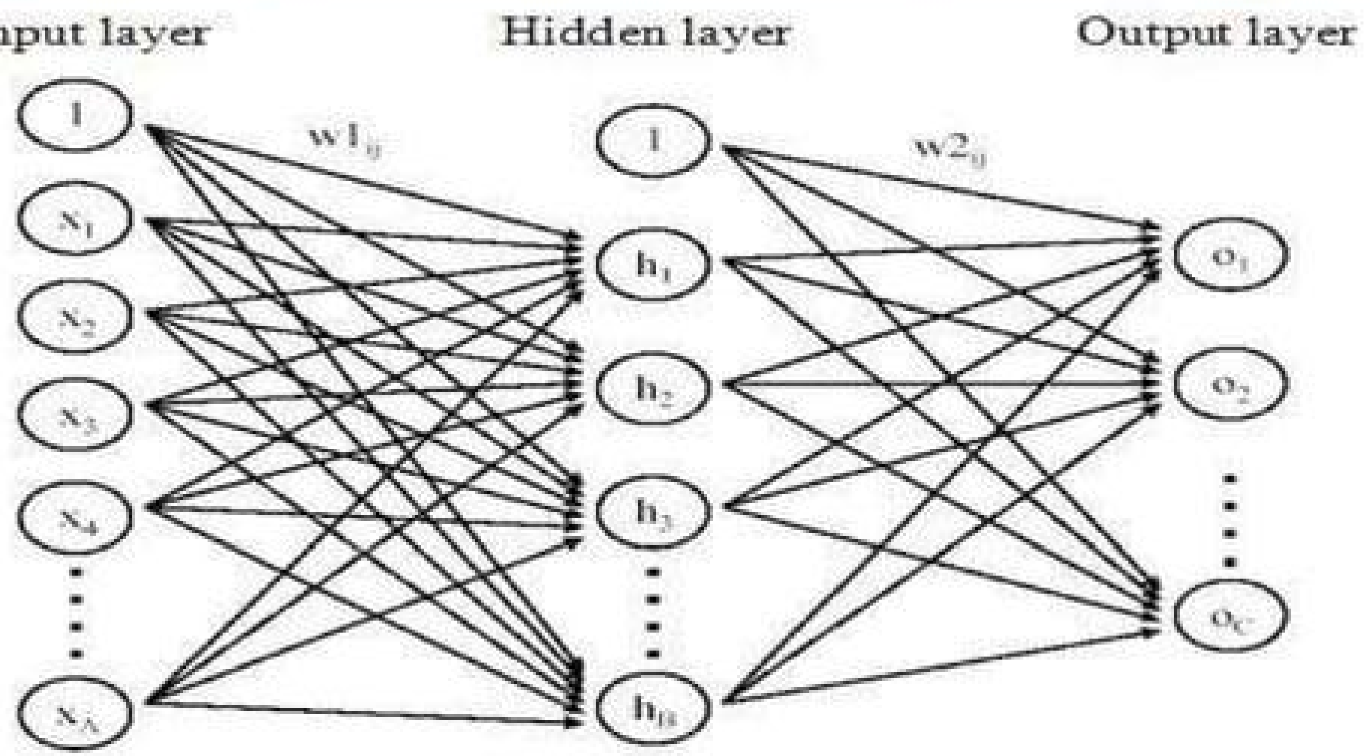
Back Propagation described by **Arthur E. Bryson** and **Yu-Chi Ho** in 1969, but it wasn't until 1986, through the work of **David E. Rumelhart, Geoffrey E. Hinton** and **Ronald J. Williams**, that it gained recognition, and it led to a “renaissance” in the field of artificial neural network research.

The term is an abbreviation for "backwards propagation of errors".

BACK PROPAGATION

- As the algorithm's name implies, the errors (and therefore the learning) propagate backwards from the output nodes to the inner nodes.
- So technically speaking, backpropagation is used to calculate the gradient of the error of the network with respect to the network's modifiable weights.
- This gradient is almost always then used in a simple stochastic gradient descent algorithm to find weights that minimize the error. "
- "B**ackpropagation**" is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient descent.
- Backpropagation usually allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited.

FEED FORWARD NETWORK



Network activation flows in one direction only: from the input layer to the output layer, passing through the hidden layer. Each unit in a layer is connected in the forward direction to every unit in the next layer.

ARCHITECTURE

- Back propagation is a multilayer feed forward network with one layer of z-hidden units.
- The y output units has b(i) bias and Z-hidden unit has b(h) as bias. It is found that both the output and hidden units have bias. The bias acts like weights on connection from units whose output is always 1.
- The input layer is connected to the hidden layer and output layer is connected to the output layer by means of interconnection weights.
- The architecture of back propagation resembles a multi-layered feed forward network.
- The increasing the number of hidden layers results in the computational complexity of the network.
- As a result, the time taken for convergence and to minimize the error may be very high.
- The bias is provided for both the hidden and the output layer,to act upon the net input to be calculated.

TRAINING ALGORITHM

The training algorithm of back propagation involves four stages.

- **Initialization of weights-** some small random values are assigned.
- **Feed forward-** each input unit (X) receives an input signal and transmits this signal to each of the hidden units $Z_1, Z_2, \ldots\ldots Z_n$. Each hidden unit then calculates the activation function and sends its signal Z_i to each output unit. The output unit calculates the activation function to form the response of the given input pattern.
- **Back propagation of errors-** each output unit compares activation Y_k with its target value T_K to determine the associated error for that unit. Based on the error, the factor $\delta_K(k=1,\ldots\ldots,m)$ is computed and is used to distribute the error at output unit Y_k back to all units in the previous layer. Similarly, the factor $\delta_j(j=1,\ldots.,p)$ is compared for each hidden unit Z_j.
- **Updation of the weights and biases.**

INITIALIZATION OF WEIGHTS

STEP 1: Initialize weight to small random values.

STEP 2: While stopping condition is false, do steps
3-10.

STEP 3: For each training pair do steps 4-9.

FEED FORWARD

STEP 4: Each input unit receives the input signal x_i and transmits this signal all units in the above i.e. hidden layer.

STEP 5: Each hidden unit (z_h,h=1,….p) sums its input signals.

$$Z_{inj} = V_{oj} + \sum x_i v_{ij}$$

Applying activation function

$$Z_j = f(Z_{inj})$$

And send this signal to all units in the layer above i.e output units.

STEP 6: Each output unit ($Y_{k,}$k=1,….m) sums its weighted input signals.

$$Y_{ink} = W_{ok} + \sum Z_j W_{ij}$$

And supplies its activation function to calculate the output signals.

$$Y_k = f(Y_{ink})$$

BACK PROPAGATION OF ERRORS

STEP 7: Each output unit (Y_k,k=1,…m) receives a target pattern corresponding to an input pattern,error information term is calculated as

$$\delta_k=(t_k-Y_k)f(Y_{ink})$$

STEP 8:Each hidden unit (Z_j,j=1,….n) sums its delta inputs from units in the layer above

$$\delta_{inj}=\sum\delta_j W_{jk}$$

the error information term is calculated as

$$\delta_j= \delta_{inj}\ f(Z_{inj})$$

UPDATION OF WEIGHTS AND BIASES

STEP 9: Each output unit (Y_k,k=1,…,m) updates its bias and weights (j=0,…,n)

The weight correctiom term is given by

$$\Delta W_{jk} = \alpha\delta_k Zj$$

And the bias correction term is given by

$$\Delta W_{ok} = \alpha\delta_k$$

Therefore,

$$W_{jk(new)} = W_{jk(old)} + \Delta W_{jk}$$

$$W_{ok(new)} = W_{ok(old)} + \Delta W_{ok}$$

Each hidden unit(Z_i,j=1,….,p) updates its bias and weights (i=0,…..n)

The weight correction term

$$\Delta V_{jk} = \alpha\delta_j x_i$$

The bias correction term

$$\Delta V_{oj} = \alpha\delta_j$$

Therefore, $V_{ij(new)} = V_{ij(old)} + \Delta V_{ij}$,

$$V_{oj(new)} = V_{oj(old)} + \Delta V_{oj}$$

STEP 10: Test the stopping condition.

The stopping condition may be minimization of the errors,number epochs etc.