Machine Learning

8. week

- Multi Layer Perceptron (MLP) Network
- Back-propagation algorithm
- Stopping Criteria

Multi Layer Perceptron Network

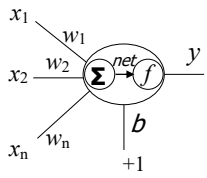
Multi layer perceptron network (MLP) is proposed after Adaline and Perceptron methods have failed producing non-linear solutions to the problem.

MLP is improved in terms of both architectural and training algorithm.

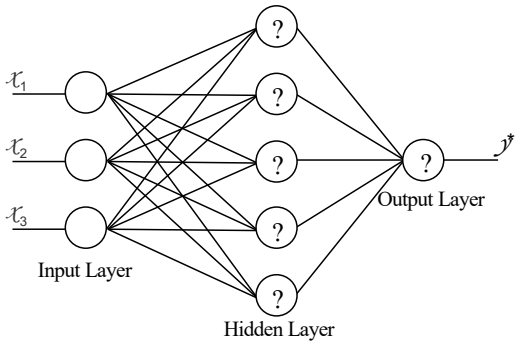
MLP is formed by many neurons each having non-linear activation function.

MLP uses back-propagation learning algorithm in addition to the advantages of Perceptron and Adaline methods.

MLP Structure



General structure of a neuron used in MLP.



General structure of an MLP network.

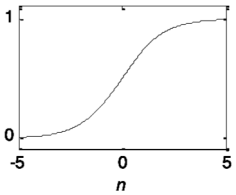
Activation Function

Activation function is basicly a linear or non-linear function that transforms a variable from one dimension into another dimension. Ease of its derivability increases the training speed. Sigmoid, hyperbolic tangent and step functions are the most frequently used activation functions.

Activation Function

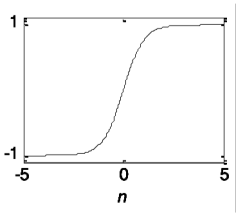
Sigmoid

$$y = \frac{1}{1+e^{-net}}$$



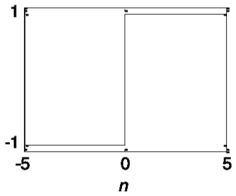
Hyperbolic tangent

$$y = \frac{1-e^{-2net}}{1+e^{2net}}$$



Step function

$$y = \begin{cases} 1 & net >= 0 \\ -1 & net < 0 \end{cases}$$



Umut ORHAN, PhD.

5

5

Backpropagation

All of the weight values are calculated by a method called gradient descent which minimizes the error function. Calculated error is adjusted by the derivation of the activation function while going back from the output end of the neuron to the input end. The applied error value to the derivation of the function is distributed to all of the weights proportional to neurons' input values.

Umut ORHAN, PhD.

6

6

Backpropagation

$$E = \frac{1}{2}\sum_j e_j^{\,2} = \frac{1}{2}\sum_j \left(d_j - y_j\right)^2$$

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \left[\frac{\partial E}{\partial y_j}\right]\left[\frac{\partial y_j}{\partial net_i}\right]\left[\frac{\partial net_i}{\partial w_{ij}}\right]$$

$$= -\eta\left[-e_j\right]\left[\frac{\partial y_j}{\partial net_i}\right]\left[x_i\right]$$

If sigmoid is used as activation function;

$$\Delta w_{ij} = -\eta\left[-e_j\right]\left[\left(1-y_j\right)y_j\right]\left[x_i\right] = \eta\, e_j\left(1-y_j\right)y_j x_i$$

Backpropagation

Prove the statement below for $y = \dfrac{1-e^{-2net}}{1+e^{2net}}$

$$-\eta\frac{\partial E}{\partial w_{ij}} = \eta\, e_j\left(1-y_j^2\right)x_i$$

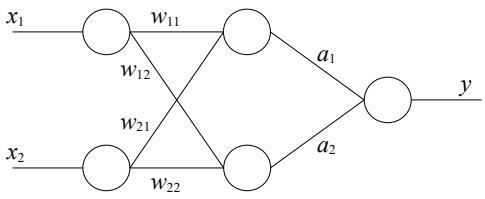
Momentum Effect

While system converges through the optimal result step by step using gradient descent, sometimes it gets stuck to one of the local minimum points and the global target cannot be reached. To prevent this, a portion (α) of value change occured in previous step is used as a momentum effect.

$$\Delta w_{ij}(t+1) = -\eta \frac{\partial E}{\partial w_{ij}} + \alpha \, \Delta w_{ij}(t)$$

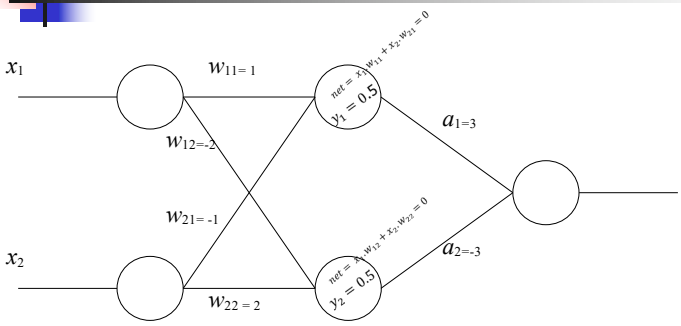
Example

Lets solve XOR problem for an MLP network which has two inputs and one output and has two neurons on its single hidden layer.



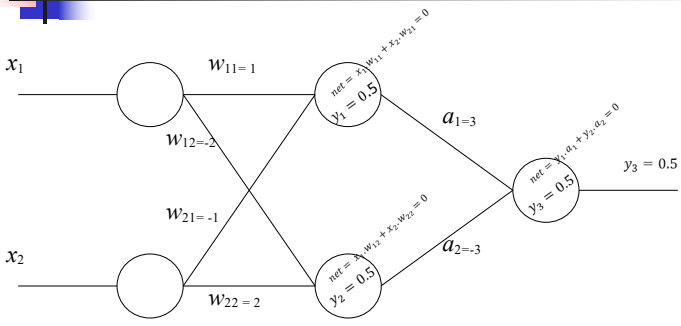
η = 0.5
α = 0.1
w_{11}= 1
w_{12}= -2
w_{21}= -1
w_{22}= 2
a_1 = 3
a_2 = -3

Example



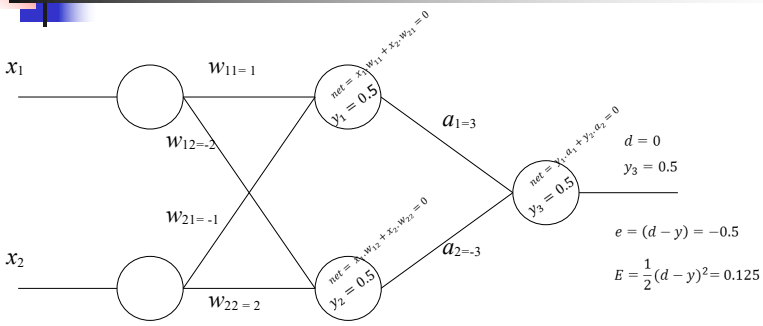
$\eta = 0.5$
$\alpha = 0.1$
$w_{11}= 1$
$w_{12}= -2$
$w_{21}= -1$
$w_{22}= 2$
$a_1 = 3$
$a_2 = -3$

Example



$\eta = 0.5$
$\alpha = 0.1$
$w_{11}= 1$
$w_{12}= -2$
$w_{21}= -1$
$w_{22}= 2$
$a_1 = 3$
$a_2 = -3$

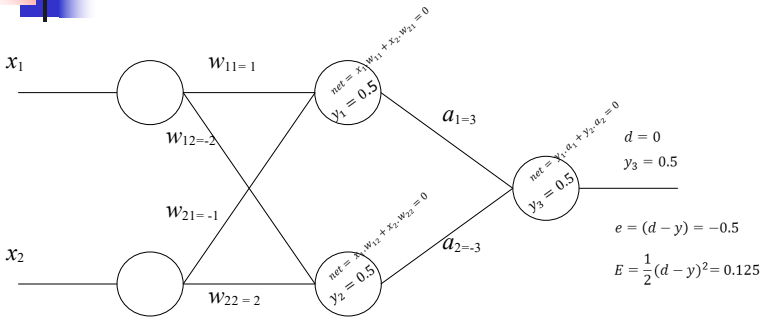
Example



η = 0.5
α = 0.1
w_{11}= 1
w_{12}= -2
w_{21}= -1
w_{22}= 2
a_1 = 3
a_2 = -3

13

Example



η = 0.5
α = 0.1
w_{11}= 1
w_{12}= -2
w_{21}= -1
w_{22}= 2
a_1 = 3
a_2 = -3

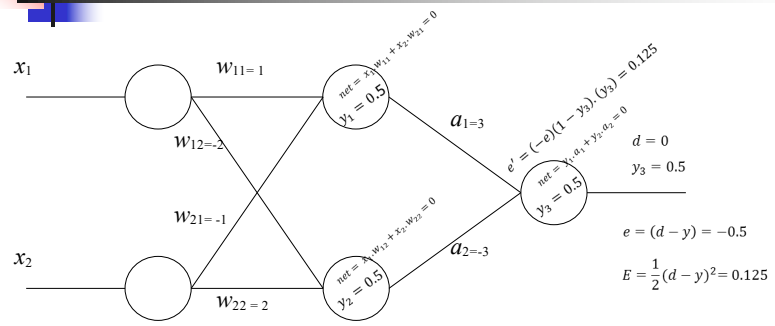
$$\Delta w_{ij} = -\eta[-e_j][(1-y_j)y_j][x_i] = \eta e_j(1-y_j)y_j x_i$$

$$\Delta a_1 = (-\eta).(-e).(1-y_3).(y_3).(y_1) = -0.03$$

$$a_{1_{new}} = 3 + (-0.03) + \alpha.\Delta a_{1_{old}} = 2.97$$

14

Example



$\eta = 0.5$
$\alpha = 0.1$
$w_{11} = 1$
$w_{12} = -2$
$w_{21} = -1$
$w_{22} = 2$
$a_1 = 3$
$a_2 = -3$

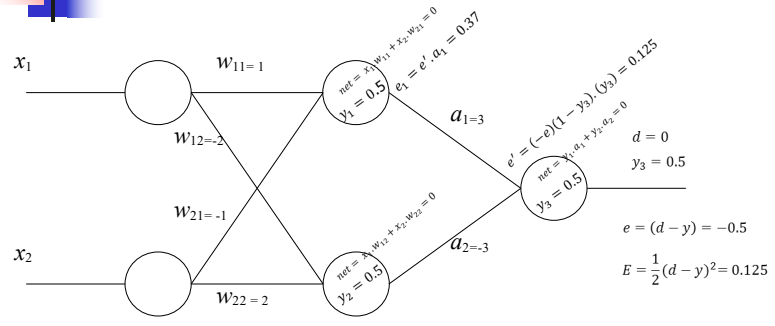
$$\Delta w_{ij} = -\eta[-e_j][(1-y_j)y_j][x_i] = \eta e_j(1-y_j)y_j x_i$$

$$\Delta a_1 = (-\eta).(-e).(1-y_3).(y_3).(y_1) = -0.03$$

$$a_{1_{new}} = 3 + (-0.03) + \alpha.\Delta a_{1_{old}} = 2.97$$

15

Example



$\eta = 0.5$
$\alpha = 0.1$
$w_{11} = 1$
$w_{12} = -2$
$w_{21} = -1$
$w_{22} = 2$
$a_1 = 3$
$a_2 = -3$

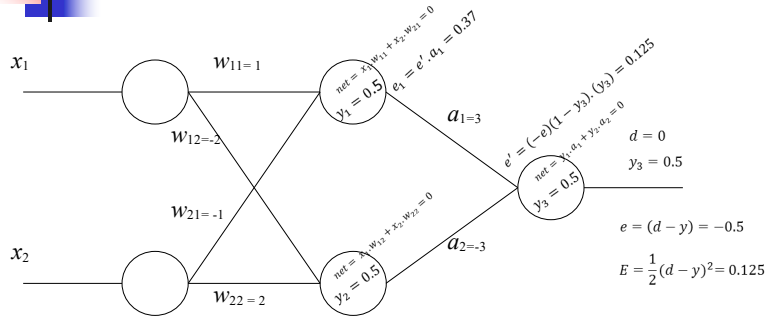
$$\Delta w_{ij} = -\eta[-e_j][(1-y_j)y_j][x_i] = \eta e_j(1-y_j)y_j x_i$$

$$\Delta a_1 = (-\eta).(-e).(1-y_3).(y_3).(y_1) = -0.03$$

$$a_{1_{new}} = 3 + (-0.03) + \alpha.\Delta a_{1_{old}} = 2.97$$

16

Example



$\eta = 0.5$
$\alpha = 0.1$
$w_{11} = 1$
$w_{12} = -2$
$w_{21} = -1$
$w_{22} = 2$
$a_1 = 3$
$a_2 = -3$

$\Delta w_{ij} = -\eta[-e_j][(1-y_j)y_j][x_i] = \eta e_j(1-y_j)y_j x_i$

$\Delta a_1 = (-\eta).(-e).(1-y_3).(y_3).(y_1) = -0.03$

$a_{1_{new}} = 3 + (-0.03) + \alpha.\Delta a_{1_{old}} = 2.97$

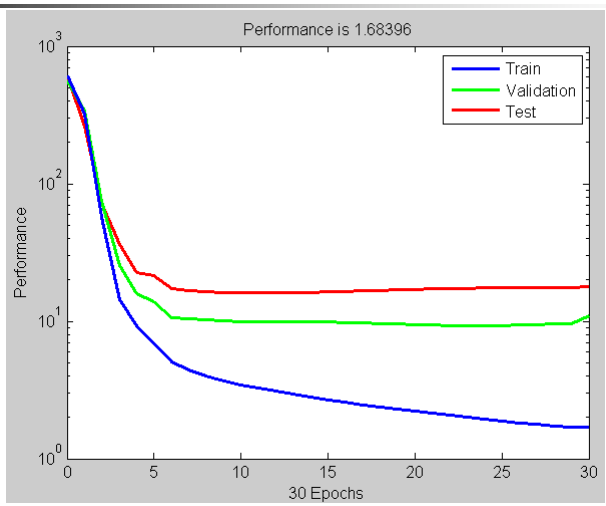
$\Delta w_{11} = (-\eta).(-e_1).(1-y_1).(y_1).(x_1) = 0.04$

$w_{11_{new}} = 1 + (0.04) + \alpha.\Delta w_{11_{old}} = 1.04$

Stopping Criteria

Without a stopping criteria to determine when to stop the training, overfitting occurs. The training set, which is a subset of the entire set, is divided into two and while one part of it is used for updating the weight values, the other part is used for validation to calculate the success of the training. If success drops rapidly, training is stopped.

Stopping Criteria



Umut ORHAN, PhD.

19

19