Function Neural Network Design

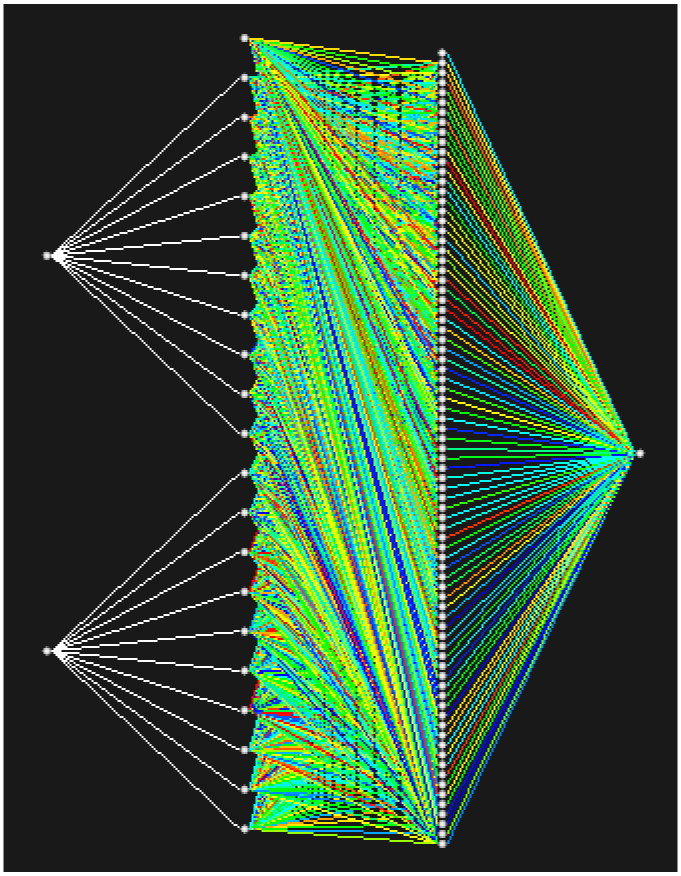
MSc. Eber Jair Flores Andonegui.

eberjair@gmail.com

Neural Network Structure

This Neural Network is designed to learn to imitate any two variables continuous function for a determined domain.

The Neural Network has been built with the input layer, two hidden layers (20 and 80 neurons) and the output layer as shown in the Neural Network Viewer in the Function Neural Network application:



Hidden layers and Output layer activation functions

First hidden layer

The first hidden layer acts as a semi input layer. Its only function is to characterize somehow the values from the input layer, given by **X1** and **X2** input values, which must be **normalized**.

The activation function for this layer is given by:

$$S_i^{(1)} = e^{-\frac{(X-\bar{x}_i)^2}{2\delta^2}} \qquad i = \overline{1,22} \tag{1}$$

where:

$X = X_1 \qquad i = \overline{1,11}$

$X = X_2 \qquad i = \overline{12,22}$

$\delta = 0.2$

$\bar{x}_1 = 0.0,\ \bar{x}_2 = 0.1,\ \bar{x}_3 = 0.2,\ \bar{x}_4 = 0.3\ldots\ \ \bar{x}_{11} = 1.0,\ \bar{x}_{12} = 0.0,\ \bar{x}_{13} = 0.1,\ \bar{x}_{14} = 0.2,$ $\bar{x}_{15} = 0.3\ldots\bar{x}_{22} = 1.0$

Second hidden layer

The second hidden layer is a fully connected layer and its activation function is given by:

$$S_j^{(2)} = \frac{1}{1+e^{-z_j}} \qquad j = \overline{1,80} \tag{2}$$

$$z_j = b_j + \sum_{i=1}^{22} w_{ij} S_i^{(1)} \qquad j = \overline{1,80} \tag{3}$$

where:

$b_j \in [-1,1]$ is the weight coming from a bias unit for this layer.

$w_{ij} \in [-1,1]$ is the weight for the connection between the neuron i of the first hidden layer and the neuron j of the second hidden layer.

Output layer

The output layer is a fully connected layer with only one neuron which gives the output value. The activation function for this layer is given by:

$$S^{(3)} = \frac{1}{1+e^{-z^{(3)}}} \qquad (4)$$

$$z^{(3)} = b^{(3)} + \sum_{j=1}^{80} w_j S_j^{(2)} \qquad (5)$$

where:

$b^{(3)} \in [-1,1]$ is the weight coming from a bias unit for this layer.

$w_j \in [-1,1]$ is the weight for the connection between the neuron j of the second hidden layer and the neuron of the output layer.

Neural network learning process

The method used for the learning process is supervised learning method with backpropagation to adjust the weights between neurons.

The used error function is given by:

$$E = \frac{1}{2}\left(Y - S^{(3)}\right)^2 \qquad (6)$$

where Y is the expected output value.

For the backpropagation we try to minimize the error function for a set of random inputs using max descend optimization.

Adjusting $b^{(3)}$

To adjust $b^{(3)}$ we use the next equation for the learning iterations:

$$b^{(3)}(t+1) = b^{(3)}(t) + \eta\left(-\frac{\partial E}{\partial b^{(3)}}\right) \qquad (7)$$

where η is the gradient factor and:

$$\frac{\partial E}{\partial b^{(3)}} = \frac{\partial E}{\partial S^{(3)}} \frac{\partial S^{(3)}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial b^{(3)}} \qquad (8)$$

Using (6), (4) and (5) we get:

$$\frac{\partial E}{\partial b^{(3)}} = \left(S^{(3)} - Y\right)\left[\left(\frac{1}{1+e^{-z^{(3)}}}\right)\left(1 - \frac{1}{1+e^{-z^{(3)}}}\right)\right] \cdot 1$$

$$\frac{\partial E}{\partial b^{(3)}} = \left(S^{(3)} - Y\right)\left[S^{(3)}\left(1 - S^{(3)}\right)\right] \qquad (9)$$

Adjusting w_j

To adjust w_j we use the next equation for the learning iterations:

$$w_j(t+1) = w_j(t) + \eta\left(-\frac{\partial E}{\partial w_j}\right) \quad (10)$$

where η is the gradient factor and:

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial S^{(3)}} \frac{\partial S^{(3)}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial w_j} \qquad (11)$$

Using (6), (4) and (5) we get:

$$\frac{\partial E}{\partial w_j} = \left(S^{(3)} - Y\right)\left[\left(\frac{1}{1+e^{-z^{(3)}}}\right)\left(1 - \frac{1}{1+e^{-z^{(3)}}}\right)\right] S_j^{(2)}$$

$$\frac{\partial E}{\partial b^{(3)}} = \left(S^{(3)} - Y\right)\left[S^{(3)}\left(1 - S^{(3)}\right)\right] S_j^{(2)} \qquad (12)$$

Adjusting b_j

To adjust b_j we use the next equation for the learning iterations:

$$b_j(t+1) = b_j(t) + \eta\left(-\frac{\partial E}{\partial b_j}\right) \quad (13)$$

where η is the gradient factor and:

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial S^{(3)}}\frac{\partial S^{(3)}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial S_j^{(2)}}\frac{\partial S_j^{(2)}}{\partial z_j}\frac{\partial z_j}{\partial b_j} \quad (14)$$

Using (6), (4), (5), (2) and (3) we get:

$$\frac{\partial E}{\partial b_j} = \left(S^{(3)} - Y\right)\left[\left(\frac{1}{1+e^{-z^{(3)}}}\right)\left(1 - \frac{1}{1+e^{-z^{(3)}}}\right)\right] w_j \left[\left(\frac{1}{1+e^{-z_j}}\right)\left(1 - \frac{1}{1+e^{-z_j}}\right)\right]\cdot 1$$

$$\frac{\partial E}{\partial b_j} = \left(S^{(3)} - Y\right)\left[S^{(3)}\left(1 - S^{(3)}\right)\right] w_j \left[S_j^{(2)}\left(1 - S_j^{(2)}\right)\right] \quad (15)$$

Adjusting w_{ij}

To adjust W_{ij} we use the next equation for the learning iterations:

$$w_{ij}(t+1) = w_{ij}(t) + \eta\left(-\frac{\partial E}{\partial w_{ij}}\right) \quad (16)$$

where η is the gradient factor and:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial S^{(3)}}\frac{\partial S^{(3)}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial S_j^{(2)}}\frac{\partial S_j^{(2)}}{\partial z_j}\frac{\partial z_j}{\partial w_{ij}} \quad (17)$$

Using (6), (4), (5), (2) and (3) we get:

$$\frac{\partial E}{\partial w_{ij}} = \left(S^{(3)} - Y\right)\left[\left(\frac{1}{1+e^{-z^{(3)}}}\right)\left(1 - \frac{1}{1+e^{-z^{(3)}}}\right)\right] w_j \left[\left(\frac{1}{1+e^{-z_j}}\right)\left(1 - \frac{1}{1+e^{-z_j}}\right)\right] S_i^{(1)}$$

$$\frac{\partial E}{\partial w_{ij}} = \left(S^{(3)} - Y\right)\left[S^{(3)}\left(1 - S^{(3)}\right)\right] w_j \left[S_j^{(2)}\left(1 - S_j^{(2)}\right)\right] S_i^{(1)} \quad (18)$$

Gradient factor η

There are three possible options for the values of the gradient factor given by:

$\eta \in [0.0001, 10]$ as a constant,

$\eta = \frac{0.1}{t}$

or $\eta = \frac{0.1}{\ln(t+1)}$

where $t = \overline{1, n}$ is the iteration of the learning process for a total of n iterations