Neural Network From Scratch

Emmanuel Misley

March 2023

1 Neural Network From Scratch

This document shows a detailed description of the computations needed to implement a neural network from scratch.

1.1 Perceptron

1.2 Artificial Neural Network

1.3 Forward Propagation

1.3.1 What does each perceptron do? (hidden layer 1)

The computation of $h^{(1)}$ implies the following linear combination between the inputs and the weights asociated with each perceptron

$$h^{(1)} = \left[h_1^{(1)}, h_2^{(1)}, h_3^{(1)}\right] \Longrightarrow \begin{cases} h_1^{(1)} = x_1 w_{11} + x_2 w_{21} + x_3 w_{31} \\ h_2^{(1)} = x_1 w_{12} + x_2 w_{22} + x_3 w_{32} \\ h_3^{(1)} = x_1 w_{13} + x_2 w_{23} + x_3 w_{33} \end{cases}$$

as can be seen, this linear combination is focused on the inputs received by each node. In other words, it views each perceptron as an isolated system, so to speak. Therefore, these and the following operations are

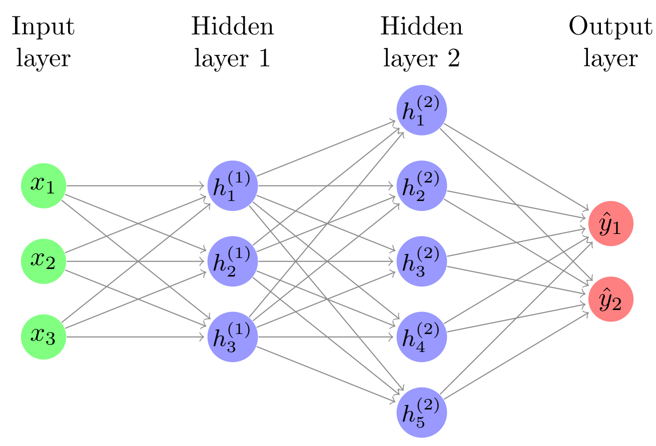

Figure 1: Artificial Neural Network

focused on the operations carried out by each perceptron in the neural network.

$$h^{(1)} = \mathbf{x}W^{(1)} = [x_1, x_2, x_3] \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = \left[h_1^{(1)}, h_2^{(1)}, h_3^{(1)}\right]$$
$$= [0.3, 0.7, 0.5] \begin{bmatrix} 0.2 & 0.1 & 0.9 \\ 0.5 & 0.1 & 0.9 \\ 0.1 & 0.5 & 0.6 \end{bmatrix} = [0.46, 0.35, 1.2]$$

So the next step is using the activation function, in this case the sigmoid activation fuction is used.

$$f(z) = \frac{1}{1 + e^{-z}}$$

the results were rounded up to 3 decimal places

$$a^{(1)} = f\left(h^{(1)}\right) = \left[\frac{1}{1 + e^{-h_1^{(1)}}}, \frac{1}{1 + e^{-h_2^{(1)}}}, \frac{1}{1 + e^{-h_3^{(1)}}}\right]$$
$$= \left[\frac{1}{1 + e^{-0.46}}, \frac{1}{1 + e^{-0.35}}, \frac{1}{1 + e^{-1.2}}\right]$$
$$= \left[0.613, 0.587, 0.769\right]$$

finally, that is the output of the first hidden layer, where each component of the vector $a^{(1)}$ is the output of each perceptron.

1.3.2 What does the rest of perceptrons do? (hidden layer 2)

As all hidden layers are connected the computation of $h^{(2)}$ uses the vector $a^{(1)}$

$$h^{(2)} = a^{(1)}W^{(2)} = \left[a_1^{(1)}, a_2^{(1)}, a_3^{(1)}\right] \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \end{bmatrix}$$
$$= \left[0.613, 0.587, 0.769\right] \begin{bmatrix} 0.5 & 0.1 & 0.8 & 0.6 & 0.6 \\ 0.9 & 0.1 & 0.9 & 0.3 & 0.8 \\ 0.4 & 0.9 & 0.4 & 0.8 & 0.2 \end{bmatrix}$$
$$= [1.142,\ 0.812,\ 1.326,\ 1.159,\ 0.991]$$

then the activation function is used

$$a^{(2)} = f\left(h^{(2)}\right) = \left[\frac{1}{1 + e^{-1.142}}, \frac{1}{1 + e^{-0.812}}, \frac{1}{1 + e^{-1.326}}, \frac{1}{1 + e^{-1.159}}, \frac{1}{1 + e^{-0.991}}\right]$$
$$= \left[0.758,\ 0.692,\ 0.790,\ 0.761,\ 0.729\right]$$

1.3.3 The Output Layer

Just as in the previous cases $h^{(3)}$ is calculated as

$$h^{(3)} = a^{(2)}W^{(3)} = \left[a_1^{(2)}, a_2^{(2)}, a_3^{(2)}, a_4^{(2)}, a_5^{(3)}\right] \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \\ w_{51} & w_{52} \end{bmatrix}$$

$$= \left[0.758,\ 0.692,\ 0.790,\ 0.761,\ 0.729\right] \begin{bmatrix} 0.9 & 0.5 \\ 0.4 & 0.1 \\ 0.1 & 0.1 \\ 0.7 & 0.5 \\ 0.8 & 0.7 \end{bmatrix}$$

$$= \left[2.154, 1.418\right]$$

and then the activation function

$$\begin{aligned} a^{(3)} = f\left(h^{(3)}\right) &= \left[\frac{1}{1+e^{-a_1^{(4)}}}, \frac{1}{1+e^{-a_2^{(4)}}}\right] \\ &= \left[\frac{1}{1+e^{-2.154}}, \frac{1}{1+e^{-1.418}}\right] \\ &= \left[0.896,\ 0.805\right] \end{aligned}$$

1.3.4 Code

the following code shows how to implement the neural network from scratch in python 3.

```python
import numpy as np
class MLP:
    def __init__(self,num_imputs,num_hidden,num_outputs,weights):
        self.num_inputs = num_imputs # num_inputs atribute is created
        self.num_hidden = num_hidden # num_hidden atribute is created
        self.num_outputs = num_outputs # num_outputs atribute is created
        self.weights = weights # weights atribute is created

    def forward_propagate(self, inputs):
        """
        Method that implements the forward propagation
        """
        activations = inputs
        for w in self.weights:
            net_inputs=np.dot(activations, w)
#applies the dot product between weights and inputs
            activations = self.sigmoid(net_inputs) #applies the activation function
        return activations

    def sigmoid(self,x):
        """
        Method that implements sigmoid function
        """
        return 1/(1+np.exp(-x))
```

Figure 2: Neural Network from scratch code part 1

```python
if __name__ == "__main__":
    #The Multi Layer Perceptron (MLP) parameters are provided
    num_imputs=3
    num_hidden=[3, 5]
    num_outputs=2
    weights = [[[0.2, 0.1, 0.9],
                [0.5, 0.1, 0.9],
                [0.1, 0.5, 0.6]],
               [[0.5, 0.1, 0.8, 0.6, 0.6],
                [0.9, 0.1, 0.9, 0.3, 0.8],
                [0.4, 0.9, 0.4, 0.8, 0.2]],
               [[0.9, 0.5],
                [0.4, 0.1],
                [0.1, 0.1],
                [0.7, 0.5],
                [0.8, 0.7]]]

    mlp = MLP(num_imputs,num_hidden,num_outputs,weights)
    inputs = [0.3,0.7,0.5]
    outputs = mlp.forward_propagate(inputs) #excute the forward propagation

    #imprimo los resultados
    print(f"The network input is: {inputs}")
    print(f"The network output is: {outputs}")
```

Figure 3: Neural Network from scratch code part 2

1.4 Backpropagation

This section will be mathematically heavy, so be prepared. The operation that allows to "go back" from the output layer to the input layer is the chain rule, figure 4 shows a diagram with the elements needed to apply this operation. Note that the superscript of h and a is equal to the number of the corresponding hidden layer.

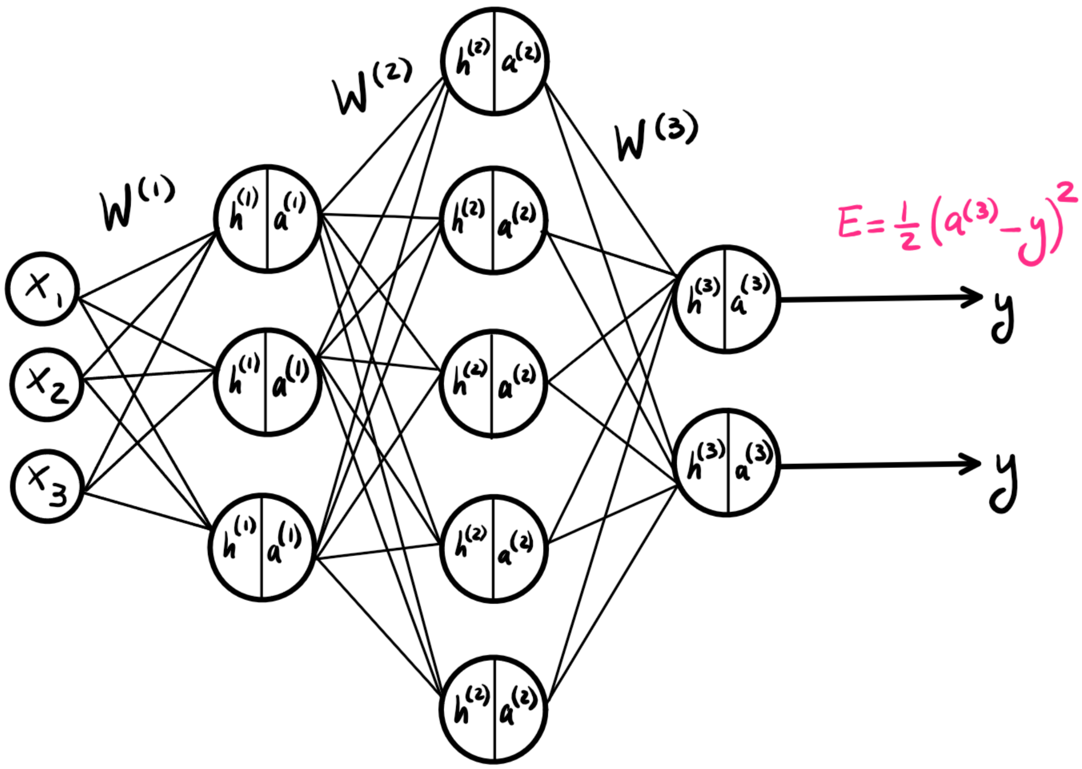

Figure 4: Artifitial neural network, with relevant annotations

the activation done with the sigmoid function, just like in the previous sections. And the error is measured using the following equation

$$E = \frac{1}{2}\left(a^{(3)} - y\right)^2 \tag{1}$$

this meant to quantify the difference between the predicted value and the actual value, therefore equation 1 measures how accurate the neural network is. the first equation is the following

$$\frac{\partial E}{\partial W^{(3)}} = \frac{\partial E}{\partial a^{(3)}} \frac{\partial a^{(3)}}{\partial h^{(3)}} \frac{\partial h^{(3)}}{\partial W^{(3)}} \tag{2}$$

the next step or element in the chain is

$$\begin{aligned} \frac{\partial E}{\partial W^{(2)}} &= \frac{\partial E}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial h^{(2)}} \frac{\partial h^{2}}{\partial W^{(2)}} \\ &= \left(\frac{\partial E}{\partial a^{(3)}} \frac{\partial a^{(3)}}{\partial h^{(3)}} \frac{\partial h^{(3)}}{\partial a^{(2)}} \right) \frac{\partial a^{(2)}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial W^{(2)}} \end{aligned}$$

now things get spicy

$$\begin{aligned} \frac{\partial E}{\partial W^{(1)}} &= \frac{\partial E}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial h^{(1)}} \frac{\partial h^{(1)}}{\partial W^{(1)}} \\ &= \left(\frac{\partial E}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial a^{(1)}} \right) \frac{\partial a^{(1)}}{\partial h^{(1)}} \frac{\partial h^{(1)}}{\partial W^{(1)}} \\ &= \left[\left(\frac{\partial E}{\partial a^{(3)}} \frac{\partial a^{(3)}}{\partial h^{(3)}} \frac{\partial h^{(3)}}{\partial a^{(2)}} \right) \frac{\partial a^{(2)}}{h^{(2)}} \frac{\partial h^{(2)}}{\partial a^{(1)}} \right] \frac{\partial a^{(1)}}{\partial h^{(1)}} \frac{\partial h^{(1)}}{\partial W^{(1)}} \end{aligned}$$

Now we are going to calculate each term and combine the result of each partial derivative to obtain the final expresion. So to summarize the equations that describes the backpropagation are

$$\begin{cases} \dfrac{\partial E}{\partial W^{(3)}} &= \dfrac{\partial E}{\partial a^{(3)}}\dfrac{\partial a^{(3)}}{\partial h^{(3)}}\dfrac{\partial h^{(3)}}{\partial W^{(3)}} \\ \\ \dfrac{\partial E}{\partial W^{(2)}} &= \left(\dfrac{\partial E}{\partial a^{(3)}}\dfrac{\partial a^{(3)}}{\partial h^{(3)}}\dfrac{\partial h^{(3)}}{\partial a^{(2)}}\right)\dfrac{\partial a^{(2)}}{\partial h^{(2)}}\dfrac{\partial h^{(2)}}{\partial W^{(2)}} \\ \\ \dfrac{\partial E}{\partial W^{(1)}} &= \left[\left(\dfrac{\partial E}{\partial a^{(3)}}\dfrac{\partial a^{(3)}}{\partial h^{(3)}}\dfrac{\partial h^{(3)}}{\partial a^{(2)}}\right)\dfrac{\partial a^{(2)}}{h^{(2)}}\dfrac{\partial h^{(2)}}{\partial a^{(1)}}\right]\dfrac{\partial a^{(1)}}{\partial h^{(1)}}\dfrac{\partial h^{(1)}}{\partial W^{(1)}} \end{cases} \tag{3}$$

so the equations are

$$\frac{\partial E}{\partial a^{(3)}} = \frac{\partial}{\partial a^{(3)}}\left(\frac{1}{2}(a^{(3)} - y)^2\right) = a^{(3)} - y \tag{4}$$

$$\frac{\partial a^{(3)}}{\partial h^{(3)}} = \frac{\partial}{\partial h^{(3)}}\left(\sigma\left(h^{(3)}\right)\right) = \sigma\left(h^{(3)}\right)\left[1 - \sigma\left(h^{(3)}\right)\right] \tag{5}$$

$$\frac{\partial h^{(3)}}{\partial W^{(3)}} = \frac{\partial}{\partial W^{(3)}}\left(a^{(2)}W^{(3)}\right) = a^{(2)} \tag{6}$$

so if combining equations 4,5,6 with the first equation of 3, the results is

$$\frac{\partial E}{\partial W^{(3)}} = (a^{(3)} - y)\sigma\left(h^{(3)}\right)\left[1 - \sigma\left(h^{(3)}\right)\right]a^{(2)}$$

so for the next equation there are two partial derivates that are repeated, those are equations 4 and 5. The calculation of the following derivatives are

$$\frac{\partial h^{(3)}}{\partial a^{(2)}} = \frac{\partial}{\partial a^{(2)}}\left(a^{(2)}W^{(3)}\right) = W^{(3)} \tag{7}$$

$$\frac{\partial a^{(2)}}{\partial h^{(2)}} = \frac{\partial}{\partial h^{(2)}}\left(\sigma\left(h^{(2)}\right)\right) = \sigma\left(h^{(2)}\right)\left[1 - \sigma\left(h^{(2)}\right)\right] \tag{8}$$

$$\frac{\partial h^{(2)}}{\partial W^{(2)}} = \frac{\partial}{\partial W^{(2)}}\left(a^{(1)}W^{(2)}\right) = a^{(1)} \tag{9}$$

combining this three equations the result is

$$\frac{\partial E}{\partial W^{(2)}} = \left(a^{(3)} - y\right)\sigma\left(h^{(3)}\right)\left[1 - \sigma\left(h^{(3)}\right)\right]W^{(3)}\sigma\left(h^{(2)}\right)\left[1 - \sigma\left(h^{(2)}\right)\right]a^{(1)} \tag{10}$$

$$\tag{11}$$

and the third equation 3 is

$$\frac{\partial h^{(2)}}{\partial a^{(1)}} = \frac{\partial}{\partial a^{(1)}}\left(a^{(1)}W^{(2)}\right) = W^{(2)} \tag{12}$$

$$\frac{\partial a^{(1)}}{\partial h^{(1)}} = \frac{\partial}{\partial h^{(1)}}\left(\sigma\left(h^{(1)}\right)\right) = \sigma\left(h^{(1)}\right)\left[1 - \sigma\left(h^{(1)}\right)\right] \tag{13}$$

$$\frac{\partial h^{(1)}}{\partial W^{(1)}} = \frac{\partial}{\partial W^{(1)}}\left(\mathbf{x}W^{(1)}\right) = \mathbf{x} \tag{14}$$

combining this three equations

$$\frac{\partial E}{\partial W^{(1)}} = \left(a^{(3)} - y\right)\sigma\left(h^{(3)}\right)\left[1 - \sigma\left(h^{(3)}\right)\right]W^{(3)}\sigma\left(h^{(2)}\right)\left[1 - \sigma\left(h^{(2)}\right)\right]W^{(2)}\sigma\left(h^{(1)}\right)\left[1 - \sigma\left(h^{(1)}\right)\right]\mathbf{x} \tag{15}$$

That was exhausting, lets simplify this

$$\begin{cases} \frac{\partial E}{\partial W^{(3)}} & = (a^{(3)} - y)\sigma\left(h^{(3)}\right)\left[1 - \sigma\left(h^{(3)}\right)\right]a^{(2)} \\ \\ \frac{\partial E}{\partial W^{(2)}} & = \left(a^{(3)} - y\right)\sigma\left(h^{(3)}\right)\left[1 - \sigma\left(h^{(3)}\right)\right]W^{(3)}\sigma\left(h^{(2)}\right)\left[1 - \sigma\left(h^{(2)}\right)\right]a^{(1)} \\ \\ \frac{\partial E}{\partial W^{(1)}} & = \left(a^{(3)} - y\right)\sigma\left(h^{(3)}\right)\left[1 - \sigma\left(h^{(3)}\right)\right]W^{(3)}\sigma\left(h^{(2)}\right)\left[1 - \sigma\left(h^{(2)}\right)\right]W^{(2)}\sigma\left(h^{(1)}\right)\left[1 - \sigma\left(h^{(1)}\right)\right]\mathbf{x} \end{cases} \tag{16}$$