Submission by:

Tikhon Riazantsev

Agastya Heryudhanto

Exercise sheet 10

TASK 1: Neural Networks

1. What are input, hidden and output layers and how many of each of these does a neural network have?

In our example that we have discussed in class we only have 3 total layers, one of each: input, hidden and output. In reality, in a given neural network model there can be any number of hidden layers (>= 1), which largely depends on the required complexity of the task that this model needs to solve. Neural networks with more than one hidden layer are called deep neural networks [Source: Google]

Input layer of the neural network is responsible for receiving input data and passing it onto the hidden layers. The number of input layer nodes is determined by the number of features of input data (for example the number of pixels in a camera output for a CV model).

Hidden layers are all of the layers between the input and output layers. Each one is connected to the two adjacent layers. A node of the hidden layer represents a weighted sum of inputs (from nodes of previous layer) passed through an activation function.

The output layer is the layer of a neural network which forms the output of the model and thus it is a layer which always processes the input data last. Here the number of nodes is determined by the type of output we want to receive.

2. Explain what the activation function is used for and give an example for an activation function

An activation function is a mathematical operation applied to the output of each node in a hidden or output layer. It is used as one of the stages that the data has to go through when being processed. It could either be linear or non-linear, each with their own pluses and minuses. The activation function that have been described in class are sigmoid and threshold functions (Fig. 1).

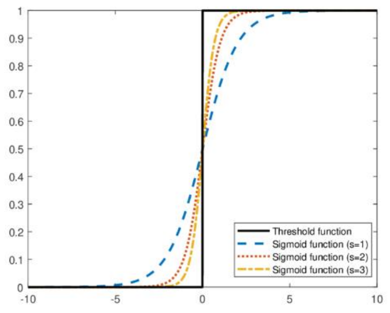

Figure 1: Sigmoid and Threshold function plots

Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$, Threshold: $f(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$

3. **Explain what the loss function is used for and give an example for a loss function.**

Loss function is a function which quantifies the difference between the predictive output of the neural network and the actual target values, providing a measure of how well the model is performing a particular task. An example of such loss function that was discussed previously is the mean square error function, which basically outputs the squared difference between the output and target values. In case there are more than one of output and target values, it calculates a sum of squared differences divided by the number of values.

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_{i\_output} - y_{i\_target}\right)^2$$

4. **Explain how a neural network is trained. What do Forward- and Backpropagation have to do with it?**

The procedure for the calculation of an output for a given input is called **forward propagation** (Fig. 2):

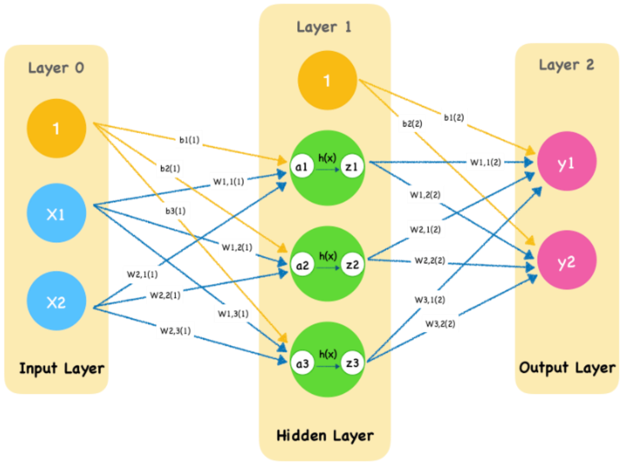

Figure 2: Schematic of forward propagation on a neural network model.

1. The weights of the input layer are first multiplied with the input parameters
2. In the hidden layer (or layers) the product is then processed by the activation function.
3. The result is then propagated into the output layer and is multiplied with the weights of the hidden layer. If there is more than one hidden layer the values go through the aforementioned steps until they reach the final output layer.
4. The final activation function calculation is performed in the output layer and the result of the neural network data processing is then outputted.

The algorithm of how a neural network is trained is called **error backpropogation**.

The loss function that we have discussed previously, quantifies the difference between the value calculated through forward propagation and the target value. It can be written not only as the function of the input variables but also as a function of all of the weights in a neural network F(h1,…,hn). To reduce the value of F we must change the values of all weights h.

1. If we use the differentiable sigmoid function as the activation function that means that F is also differentiable. We can take partial derivatives of this function by each weight:

$$F_0 = \left(\frac{\partial F}{\partial h_1}, \dots, \frac{\partial F}{\partial h_n}\right) = grad(F)$$

2. We can approach closer to the optimal solution if we change the weight vector "in the direction" of the negative gradient. To do that we can just subtract F_0 from the weight vector.

$$H = (h_1, \dots, h_n)$$
$$H_{new} = H - F_0$$

Thus, with each iteration the model approaches (local) convergence.

TASK 2: Classification using Neural Networks

1. Compare the classification results with the given data. Does the neural network classify all the data correctly?

You can see the result of the classification made by the neural network on figure 3.

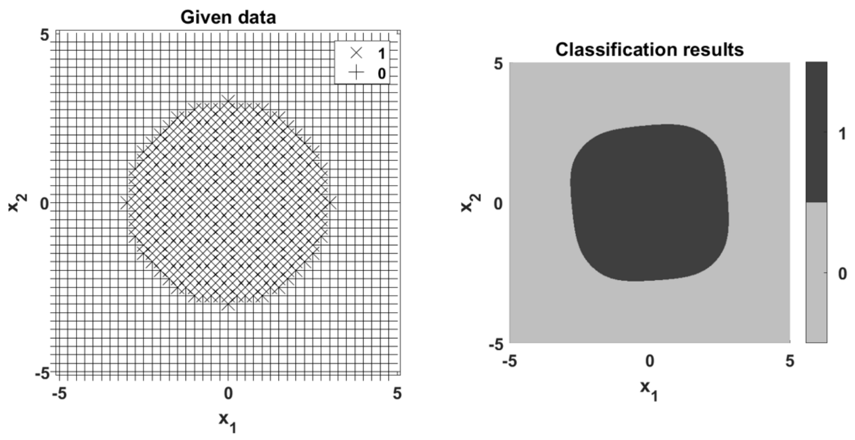

Figure 3: Neural network classification results

The model does indeed classify most of the datapoints correctly but it is relatively easy to see that this neural network makes many mistakes, outputting a square-like shape with rounded edges instead of a circle.

2. **Remember the classification results for such a task when using a support vector machine (SVM). Discuss whether using a SVM would lead to better classification results and give a reason.**

Compared to the SVM, which correctly classified 100% of data, the neural network seems to be a much worse option. This is caused mainly by the fact that SVM by definition is able to divide separable data by building a separation hyperplane with a high level of certainty. A neural network lacks this kind of certainty in its calculation.

Even with a very high amount of data points and a perfectly configured model, a neural network is never going to reach the single possible global convergent state. But it might get very close! Neural networks are better at solving similar problems when, for example, the datasets are much larger or when data points are not as closely clumped together and are more difficult to separate using an SVM.

On the other hand, we believe that it is possible to achieve a better classification result with this dataset if the neural network was configured better.