**COMP 472 Artificial Intelligence Summer 2023**

**Mini project 2**

Raghad Jaafar 40157929

Therese

**Question 1**

a) How many hidden layers have you used?

There are no hidden layers in this implementation.

b) What are the weights and biases of each node?

The neural network takes two input units and has a single output node, and its weights and bias are defined as follows:

Weight 1: 0.5

Weight 2: 0.5

Bias: -0.7

**Question 2**

a) How many hidden layers have you used? And why?

There is one hidden layer made up of 16 nodes. Since the dataset is small and simple, we saw that it is sufficient to use one layer in order to learn the data pattern.

b) How many nodes in each hidden layer and why that number of nodes in particular?

c) What is the activation function that you used and why? Did you use the same activation function in all layers? Why?

We decided to use ‘relu’ (Rectified Linear Unit) as the activation function in the hidden layer, and the ‘sigmoid’ function in the output layer. The ‘relu’ allowed us to introduce nonlinearity to the model and made it possible to understand the relationships in the data. The "sigmoid" activation function, which gives an output between 0 and 1, is used in the output layer for binary classification issues so the output belongs to one class or the other (0 or 1). The hidden and output layers apply different activation functions because they serve different purposes and are appropriate for their respective places in the neural network.

d) What learning algorithm did you use to train the neural net and why?

e) Can you use one hidden layer only to solve this problem? If yes, how many nodes are you going to have in it? And why?

Yes, this issue can be solved with just one hidden layer. The complexity of the issue and the model's ability to learn the underlying patterns are taken into consideration when determining the number of nodes in the hidden layer. In our case, the hidden layer has16 nodes. The exact number of nodes is chosen through trial and fine-tuning. Complex patterns may be missed if the model is underfitted, which occurs when there aren't enough nodes. On the other hand, overfitting where the model memorises the training data without properly generalising to new data, can result from having too many nodes. Hence, we need to find a balance.

f) Can we use 5 hidden layers? Is that a good idea? Justify your answer.

g) How did the neural net do in classifying the testing set? Comment on how good or bad it learned the function from the training set.

The neural network's performance in classifying the testing set can be assessed using the test loss and test accuracy values obtained after evaluating the model on the testing data. We get a high accuracy rate and a low loss rate, indicating that the model generalized well to the unknown test set. Because we chose a batch size of one, the model was able to change the weights frequently based on a single sample and easily recognized the patterns.