CODE DOCUMENTATION

Question 2:

Code structure:

We structured our code in the following way:

- We start by importing all the libraries needed to build and analyze our neural network such as tensorflow, keras and numpy and one class that will represent the layer of our neural network.
- We created numpy arrays, the first containing the training data set that our neural network will use to train on along with its output in the second numpy array.
- We then created our neural network and named it as a model. We had one input layer with 5 nodes, 1 hidden layer with 8 nodes and 1 output layer with one node. We chose relu as an activation function for the first two layers and a sigmoid function for the last layer
- To configure the learning process of the model before training we compiled it using “adam” as the optimizer, to measure the the discrepancy between the predicted output and the true one we used binary cross entropy as the loss function and finally to evaluate the performance of the model we chose “accuracy” as our metrics.
- To train our model we provided the input and output data set, we set epochs, which are the number of times model will go over the training set, to 100 and batch_size, which is the number of samples that will be processed together before updating the weights and biases, to 1
- We then tried to predict the output of a test data set that we have their true output which we printed at the end.
- We calculated the accuracy using the Mean absolute error which will see the difference between the predicted output and the true output and will see to which extent we are close the true output
- At the end of our program we printed the model summary which shows the number of parameters that the model has in each layer and the total.

Code functionality:

The functionality of our code is as follows:

- We built a neural network that takes an input of bits and will try to predict a 0 or 1 based on the training that was already done using a training data set

- In our case we provided a training data set of 24 possible combination and a 5-bits sample and we tried to predict 5 samples
- We tried to keep the model as simple as possible while achieving the highest possible accuracy in our case it was 85% and at the same time avoiding overfitting at all costs
- In the first two layers we used a relu as an activation function because the inputs and outputs are not linear so we needed to capture that using a relu function but for the output layer we chose a sigmoid function because the output is either 0 or 1 so we wanted to achieve that through it.
- To optimize our model when training it we use “Adam” that will update the weights and biases in each epoch in our cases it will do it 100 times for a batch size of 1. So for each sample it will update the weights and biases.
- The loss which is the discrepancy between the true output and the predicted one will be measured through the binary cross entropy which is suitable for binary data sets.
- The performance will be measured using the accuracy as metrics. For our model it is suitable because the task is fairly simple and we don’t need something that is complex.
- We got an accuracy of about 85 % using the mean absolute error which is good for predicting data while showing that we haven’t fallen into overfitting our predictions to the training data set. We still have room for our model to learn.