Mini project – part 2

Mini project part-2 consists of 2 activities: 2.1 and 2.2. 2.1 related to output prediction and 2.2 related to image classification]

[Need to upload your code as well as a report in pdf format]

2.1 Consider the following Table 1, which has three input columns and one output column. The problem can be solved by measuring statistics between input and output values. The leftmost input column is correlated with the output column.

Select proper neural network architecture to solve this problem. Train the neural network using the train data set given in Table 1 and the backpropagation algorithm. Select the initial weights and bias terms using NumPy package random function. Implement the neural network in Jupyter Notebook (.ipynb). After training, your neural network should be able to give the correct output (which matches the first column) for different input combinations. Add suitable comments to your code to explain the steps.

Table 1: input-output table

	Inputs		Output
0	0	1	0
1	1	1	1
1	0	1	1
0	1	1	0

2.2 MNIST dataset contains a set of handwritten numerical digits (0-9) images. You need to develop a Neural network to identify the digit present in each image. For example, consider the following image containing the handwritten digit '9'. So, once you feed this image to your neural network, it should correctly classify it and give the output as '9'.

(i). First, try to implement a deep Neural network without any convolutional layers. Calculate the accuracy-related to your training set and test set.

(ii). Then implement a Convolutional Neural network and calculate the accuracy-related to your training set and test set.

(iii) Obtain some other images with handwritten digits (0-9) which do not belong to the MNIST dataset. Rescale them to the exact resolution as MNIST images. Feed those images to neural networks developed in above (i) and (ii). And check the predicted outputs of those neural networks.

In [1]:
```
import tensorflow as tf
```

In [2]:
```
%config Completer.use_jedi = False
```

In [12]:
```
from tensorflow.keras.datasets import mnist
```

In [14]:
```
(train_X,train_y),(test_X,test_y) = mnist.load_data()
```

In [15]:
```
print('X_train: ' + str(train_X.shape))
print('Y_train: ' + str(train_y.shape))
print('X_test:  '  + str(test_X.shape))
print('Y_test:  '  + str(test_y.shape))
```

```
X_train: (60000, 28, 28)
Y_train: (60000,)
X_test:  (10000, 28, 28)
Y_test:  (10000,)
```

In [32]:
```
import matplotlib.pyplot as plt

w=10
h=10
fig=plt.figure(figsize=(4, 4))
columns = 3
rows = 2
for i in range(1, columns*rows +1):
    fig.add_subplot(rows, columns, i)
    plt.imshow(train_X[i],cmap=plt.get_cmap('gray'))
plt.show()
```