Problem 1: Application of Keras to biuld, compile, and train a neural network to perform XOR operation:

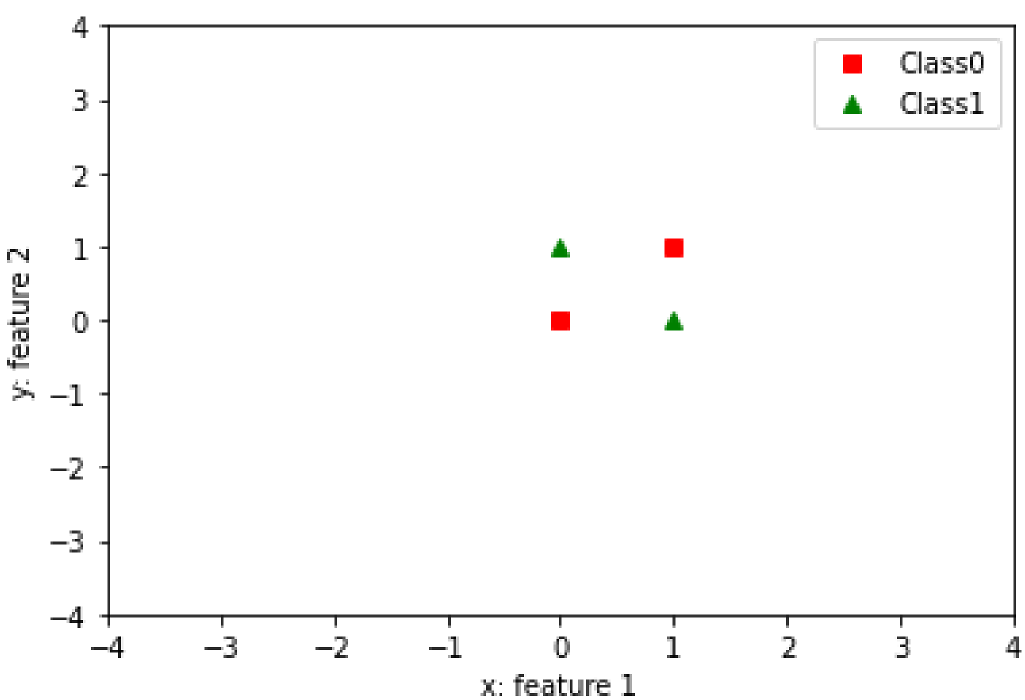
1. Create an np array of shape 4x2 for the inputs and another 4x1 array for the labels of XOR.
2. Plot the given data points with two different markers for each group.
3. Based on the plot from part (b), what is the minimum number of layers and nodes that is required to classify the training data points correctly? Explain.
4. Build the network that you proposed in part c using the Keras library.
5. Compile the network. Make sure to select a correct loss function for this classification problem. Use stochastic gradient descent learning (SGD, learning rate of 0.1). Explain your selection of the loss function.

6. Train the network for 200 epochs and a batch size of 1.
7. Use the trained weights and plot the final classifier lines in the plot of part (b).
8. Plot the training loss (i.e., the learning curve) for all the epochs.
9. Repeat steps (d) to (g) after adding 2 more nodes to the first layer and training for 400 epochs.

10. What behavior do you observe from the classifier lines after adding more nodes? Which number of nodes is more suitable in this problem? Explain.

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Dense

# Part a and b

def plot_fun(features, labels, classes):
    plt.plot(features[labels[:] == classes[0],0], features[labels[:]
== classes[0],1], 'rs', features[labels[:] == classes[1], 0],
features[labels[:] == classes[1], 1], 'g^')
    plt.axis([-4, 4, -4, 4])
    plt.xlabel('x: feature 1')
    plt.ylabel('y: feature 2')
    plt.legend(['Class'+str(classes[0]), 'Class'+str(classes[1])])
    plt.show()

def plot_fun_thr(features, labels, thre_parms, classes):
  #plotting the data points
  plt.plot(features[labels[:]==classes[0], 0],
features[labels[:]==classes[0], 1], 'rs',
features[labels[:]==classes[1],0], features[labels[:]==classes[1], 1],
'g^', markersize=15)
  x1 = np.linspace(-2, 2, 50)
```

```
  x2 = -(thre_parms[0]*x1+thre_parms[2])/thre_parms[1]
  plt.plot(x1, x2, '-r')
  plt.xlabel('x: features 1')
  plt.ylabel('y: features 2')
  plt.legend(['Class'+str(classes[0]), 'Class'+str(classes[1])])


def plot_curve(accuracy_train, loss_train):
  epochs = np.arange(loss_train.shape[0])
  plt.subplot(1,2,1)
  plt.plot(epochs, accuracy_train)
  plt.xlabel('Epoch#')
  plt.ylabel('Accuracy')
  plt.title('Training Accuracy')
  plt.subplot(1,2,2)
  plt.plot(epochs, loss_train)
  plt.xlabel('Epoch#')
  plt.ylabel('Binary crossentropy loss')
  plt.title('Training Loss')
  plt.show()

features = np.array([[0, 0],[0,1], [1,0], [1,1]])
labels = np.array([0, 1, 1, 0])
classes = [0, 1]

plot_fun(features, labels, classes)
```

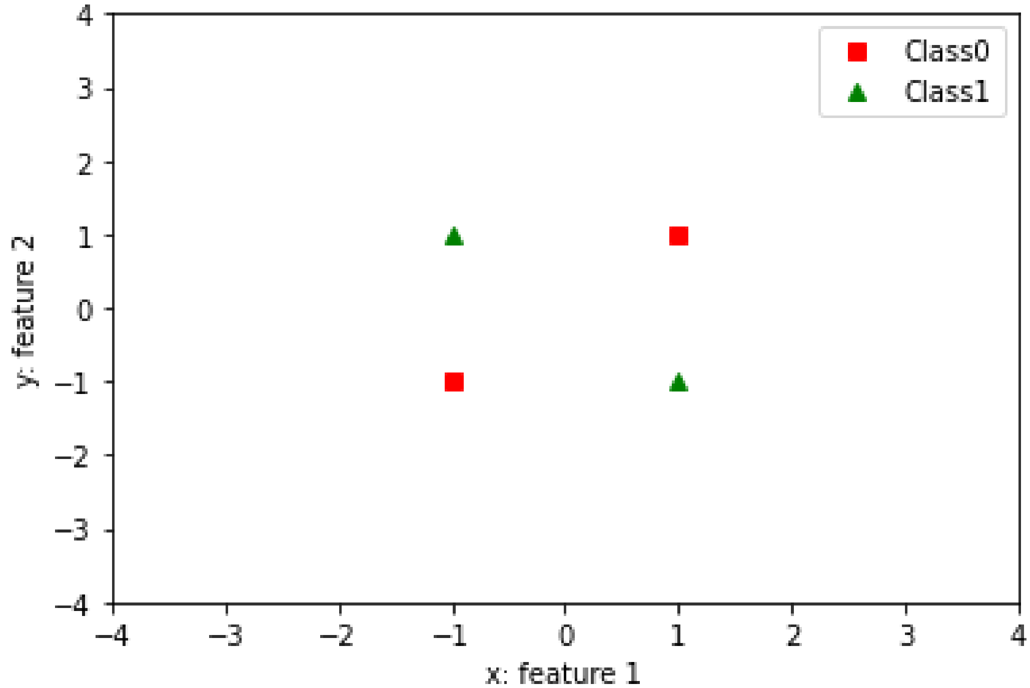


Part C: The Minimum number of layers are are 2, 1 hidden layer with two nodes and 1 output layer with one node. The hidden layer needs two nodes for each line, and the output layer needs one layer to make a decision based on the data from the two lines in the hidden layer.

```
# part d

# defining the model
model_a = Sequential()
model_a.add(Dense(input_dim = 2, units = 2, activation = 'tanh'))
model_a.add(Dense(units = 1, activation = 'sigmoid'))
model_a.summary()

# part e

# compiling the model
opt = tf.keras.optimizers.SGD(learning_rate = 0.1)
model_a.compile(loss = 'binary_crossentropy', optimizer = opt, metrics
= ['accuracy'])
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_21 (Dense)            (None, 2)                 6

 dense_22 (Dense)            (None, 1)                 3

=================================================================
Total params: 9
Trainable params: 9
Non-trainable params: 0
_________________________________________________________________
```

```
# part f

# normalization
features = (features - np.mean(features, axis = 0)) / np.std(features,
axis = 0) # normalization
plot_fun(features, labels, classes)

#train the network for 200 epochs and a batch size of 1
history = model_a.fit(features, labels, batch_size = 1, epochs = 200,
verbose = 1)
```




```
Epoch 1/200
4/4 [==============================] - 0s 4ms/step - loss: 0.7690 -
accuracy: 0.5000
Epoch 2/200
4/4 [==============================] - 0s 3ms/step - loss: 0.7562 -
accuracy: 0.5000
Epoch 3/200
4/4 [==============================] - 0s 2ms/step - loss: 0.7452 -
accuracy: 0.5000
Epoch 4/200
4/4 [==============================] - 0s 3ms/step - loss: 0.7349 -
accuracy: 0.5000
Epoch 5/200
4/4 [==============================] - 0s 3ms/step - loss: 0.7253 -
accuracy: 0.5000
Epoch 6/200
4/4 [==============================] - 0s 3ms/step - loss: 0.7169 -
accuracy: 0.5000
Epoch 7/200
4/4 [==============================] - 0s 3ms/step - loss: 0.7084 -
accuracy: 0.5000
Epoch 8/200
4/4 [==============================] - 0s 3ms/step - loss: 0.7005 -
accuracy: 0.5000
Epoch 9/200
4/4 [==============================] - 0s 2ms/step - loss: 0.6932 -
accuracy: 0.7500
Epoch 10/200
```

```
4/4 [==============================] - 0s 2ms/step - loss: 0.6866 -
accuracy: 0.7500
Epoch 11/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6806 -
accuracy: 0.7500
Epoch 12/200
4/4 [==============================] - 0s 2ms/step - loss: 0.6746 -
accuracy: 0.7500
Epoch 13/200
4/4 [==============================] - 0s 2ms/step - loss: 0.6687 -
accuracy: 0.7500
Epoch 14/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6622 -
accuracy: 0.7500
Epoch 15/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6569 -
accuracy: 0.7500
Epoch 16/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6515 -
accuracy: 0.7500
Epoch 17/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6458 -
accuracy: 0.7500
Epoch 18/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6403 -
accuracy: 0.7500
Epoch 19/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6352 -
accuracy: 0.7500
Epoch 20/200
4/4 [==============================] - 0s 4ms/step - loss: 0.6300 -
accuracy: 0.7500
Epoch 21/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6245 -
accuracy: 0.7500
Epoch 22/200
4/4 [==============================] - 0s 5ms/step - loss: 0.6193 -
accuracy: 0.7500
Epoch 23/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6142 -
accuracy: 0.7500
Epoch 24/200
4/4 [==============================] - 0s 3ms/step - loss: 0.6085 -
accuracy: 0.7500
Epoch 25/200
4/4 [==============================] - 0s 2ms/step - loss: 0.6034 -
accuracy: 0.7500
Epoch 26/200
4/4 [==============================] - 0s 4ms/step - loss: 0.5977 -
accuracy: 0.7500
```

```
Epoch 27/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5922 -
accuracy: 0.7500
Epoch 28/200
4/4 [==============================] - 0s 4ms/step - loss: 0.5866 -
accuracy: 0.7500
Epoch 29/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5813 -
accuracy: 0.7500
Epoch 30/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5748 -
accuracy: 0.7500
Epoch 31/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5688 -
accuracy: 0.7500
Epoch 32/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5622 -
accuracy: 0.7500
Epoch 33/200
4/4 [==============================] - 0s 2ms/step - loss: 0.5560 -
accuracy: 0.7500
Epoch 34/200
4/4 [==============================] - 0s 2ms/step - loss: 0.5491 -
accuracy: 0.7500
Epoch 35/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5425 -
accuracy: 0.7500
Epoch 36/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5346 -
accuracy: 0.7500
Epoch 37/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5274 -
accuracy: 0.7500
Epoch 38/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5189 -
accuracy: 0.7500
Epoch 39/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5106 -
accuracy: 0.7500
Epoch 40/200
4/4 [==============================] - 0s 3ms/step - loss: 0.5014 -
accuracy: 0.7500
Epoch 41/200
4/4 [==============================] - 0s 3ms/step - loss: 0.4927 -
accuracy: 0.7500
Epoch 42/200
4/4 [==============================] - 0s 2ms/step - loss: 0.4832 -
accuracy: 0.7500
Epoch 43/200
4/4 [==============================] - 0s 3ms/step - loss: 0.4734 -
```

```
accuracy: 0.7500
Epoch 44/200
4/4 [==============================] - 0s 4ms/step - loss: 0.4634 -
accuracy: 0.7500
Epoch 45/200
4/4 [==============================] - 0s 3ms/step - loss: 0.4532 -
accuracy: 0.7500
Epoch 46/200
4/4 [==============================] - 0s 4ms/step - loss: 0.4434 -
accuracy: 0.7500
Epoch 47/200
4/4 [==============================] - 0s 4ms/step - loss: 0.4331 -
accuracy: 0.7500
Epoch 48/200
4/4 [==============================] - 0s 4ms/step - loss: 0.4225 -
accuracy: 0.7500
Epoch 49/200
4/4 [==============================] - 0s 3ms/step - loss: 0.4126 -
accuracy: 1.0000
Epoch 50/200
4/4 [==============================] - 0s 3ms/step - loss: 0.4023 -
accuracy: 1.0000
Epoch 51/200
4/4 [==============================] - 0s 3ms/step - loss: 0.3922 -
accuracy: 1.0000
Epoch 52/200
4/4 [==============================] - 0s 3ms/step - loss: 0.3821 -
accuracy: 1.0000
Epoch 53/200
4/4 [==============================] - 0s 3ms/step - loss: 0.3723 -
accuracy: 1.0000
Epoch 54/200
4/4 [==============================] - 0s 4ms/step - loss: 0.3629 -
accuracy: 1.0000
Epoch 55/200
4/4 [==============================] - 0s 3ms/step - loss: 0.3534 -
accuracy: 1.0000
Epoch 56/200
4/4 [==============================] - 0s 3ms/step - loss: 0.3444 -
accuracy: 1.0000
Epoch 57/200
4/4 [==============================] - 0s 3ms/step - loss: 0.3354 -
accuracy: 1.0000
Epoch 58/200
4/4 [==============================] - 0s 3ms/step - loss: 0.3267 -
accuracy: 1.0000
Epoch 59/200
4/4 [==============================] - 0s 3ms/step - loss: 0.3183 -
accuracy: 1.0000
Epoch 60/200
```

```
4/4 [==============================] - 0s 3ms/step - loss: 0.3102 -
accuracy: 1.0000
Epoch 61/200
4/4 [==============================] - 0s 3ms/step - loss: 0.3023 -
accuracy: 1.0000
Epoch 62/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2946 -
accuracy: 1.0000
Epoch 63/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2872 -
accuracy: 1.0000
Epoch 64/200
4/4 [==============================] - 0s 4ms/step - loss: 0.2800 -
accuracy: 1.0000
Epoch 65/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2731 -
accuracy: 1.0000
Epoch 66/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2665 -
accuracy: 1.0000
Epoch 67/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2601 -
accuracy: 1.0000
Epoch 68/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2539 -
accuracy: 1.0000
Epoch 69/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2481 -
accuracy: 1.0000
Epoch 70/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2423 -
accuracy: 1.0000
Epoch 71/200
4/4 [==============================] - 0s 4ms/step - loss: 0.2368 -
accuracy: 1.0000
Epoch 72/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2315 -
accuracy: 1.0000
Epoch 73/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2264 -
accuracy: 1.0000
Epoch 74/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2214 -
accuracy: 1.0000
Epoch 75/200
4/4 [==============================] - 0s 3ms/step - loss: 0.2167 -
accuracy: 1.0000
Epoch 76/200
4/4 [==============================] - 0s 2ms/step - loss: 0.2121 -
accuracy: 1.0000
```

```
Epoch 77/200
4/4 [==============================] - 0s 2ms/step - loss: 0.2077 -
accuracy: 1.0000
Epoch 78/200
4/4 [==============================] - 0s 2ms/step - loss: 0.2034 -
accuracy: 1.0000
Epoch 79/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1993 -
accuracy: 1.0000
Epoch 80/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1953 -
accuracy: 1.0000
Epoch 81/200
4/4 [==============================] - 0s 4ms/step - loss: 0.1915 -
accuracy: 1.0000
Epoch 82/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1877 -
accuracy: 1.0000
Epoch 83/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1841 -
accuracy: 1.0000
Epoch 84/200
4/4 [==============================] - 0s 4ms/step - loss: 0.1807 -
accuracy: 1.0000
Epoch 85/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1773 -
accuracy: 1.0000
Epoch 86/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1740 -
accuracy: 1.0000
Epoch 87/200
4/4 [==============================] - 0s 4ms/step - loss: 0.1709 -
accuracy: 1.0000
Epoch 88/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1678 -
accuracy: 1.0000
Epoch 89/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1649 -
accuracy: 1.0000
Epoch 90/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1620 -
accuracy: 1.0000
Epoch 91/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1593 -
accuracy: 1.0000
Epoch 92/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1566 -
accuracy: 1.0000
Epoch 93/200
4/4 [==============================] - 0s 2ms/step - loss: 0.1540 -
```

```
accuracy: 1.0000
Epoch 94/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1514 -
accuracy: 1.0000
Epoch 95/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1490 -
accuracy: 1.0000
Epoch 96/200
4/4 [==============================] - 0s 5ms/step - loss: 0.1466 -
accuracy: 1.0000
Epoch 97/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1443 -
accuracy: 1.0000
Epoch 98/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1420 -
accuracy: 1.0000
Epoch 99/200
4/4 [==============================] - 0s 4ms/step - loss: 0.1398 -
accuracy: 1.0000
Epoch 100/200
4/4 [==============================] - 0s 4ms/step - loss: 0.1377 -
accuracy: 1.0000
Epoch 101/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1356 -
accuracy: 1.0000
Epoch 102/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1336 -
accuracy: 1.0000
Epoch 103/200
4/4 [==============================] - 0s 4ms/step - loss: 0.1316 -
accuracy: 1.0000
Epoch 104/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1297 -
accuracy: 1.0000
Epoch 105/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1279 -
accuracy: 1.0000
Epoch 106/200
4/4 [==============================] - 0s 4ms/step - loss: 0.1260 -
accuracy: 1.0000
Epoch 107/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1243 -
accuracy: 1.0000
Epoch 108/200
4/4 [==============================] - 0s 4ms/step - loss: 0.1226 -
accuracy: 1.0000
Epoch 109/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1209 -
accuracy: 1.0000
Epoch 110/200
```

```
4/4 [==============================] - 0s 4ms/step - loss: 0.1193 -
accuracy: 1.0000
Epoch 111/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1177 -
accuracy: 1.0000
Epoch 112/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1161 -
accuracy: 1.0000
Epoch 113/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1146 -
accuracy: 1.0000
Epoch 114/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1131 -
accuracy: 1.0000
Epoch 115/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1117 -
accuracy: 1.0000
Epoch 116/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1103 -
accuracy: 1.0000
Epoch 117/200
4/4 [==============================] - 0s 4ms/step - loss: 0.1089 -
accuracy: 1.0000
Epoch 118/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1075 -
accuracy: 1.0000
Epoch 119/200
4/4 [==============================] - 0s 5ms/step - loss: 0.1062 -
accuracy: 1.0000
Epoch 120/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1049 -
accuracy: 1.0000
Epoch 121/200
4/4 [==============================] - 0s 5ms/step - loss: 0.1037 -
accuracy: 1.0000
Epoch 122/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1024 -
accuracy: 1.0000
Epoch 123/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1012 -
accuracy: 1.0000
Epoch 124/200
4/4 [==============================] - 0s 3ms/step - loss: 0.1000 -
accuracy: 1.0000
Epoch 125/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0989 -
accuracy: 1.0000
Epoch 126/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0978 -
accuracy: 1.0000
```

```
Epoch 127/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0967 -
accuracy: 1.0000
Epoch 128/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0956 -
accuracy: 1.0000
Epoch 129/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0945 -
accuracy: 1.0000
Epoch 130/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0935 -
accuracy: 1.0000
Epoch 131/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0925 -
accuracy: 1.0000
Epoch 132/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0915 -
accuracy: 1.0000
Epoch 133/200
4/4 [==============================] - 0s 2ms/step - loss: 0.0905 -
accuracy: 1.0000
Epoch 134/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0895 -
accuracy: 1.0000
Epoch 135/200
4/4 [==============================] - 0s 2ms/step - loss: 0.0886 -
accuracy: 1.0000
Epoch 136/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0877 -
accuracy: 1.0000
Epoch 137/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0868 -
accuracy: 1.0000
Epoch 138/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0859 -
accuracy: 1.0000
Epoch 139/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0851 -
accuracy: 1.0000
Epoch 140/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0842 -
accuracy: 1.0000
Epoch 141/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0834 -
accuracy: 1.0000
Epoch 142/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0825 -
accuracy: 1.0000
Epoch 143/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0817 -
```

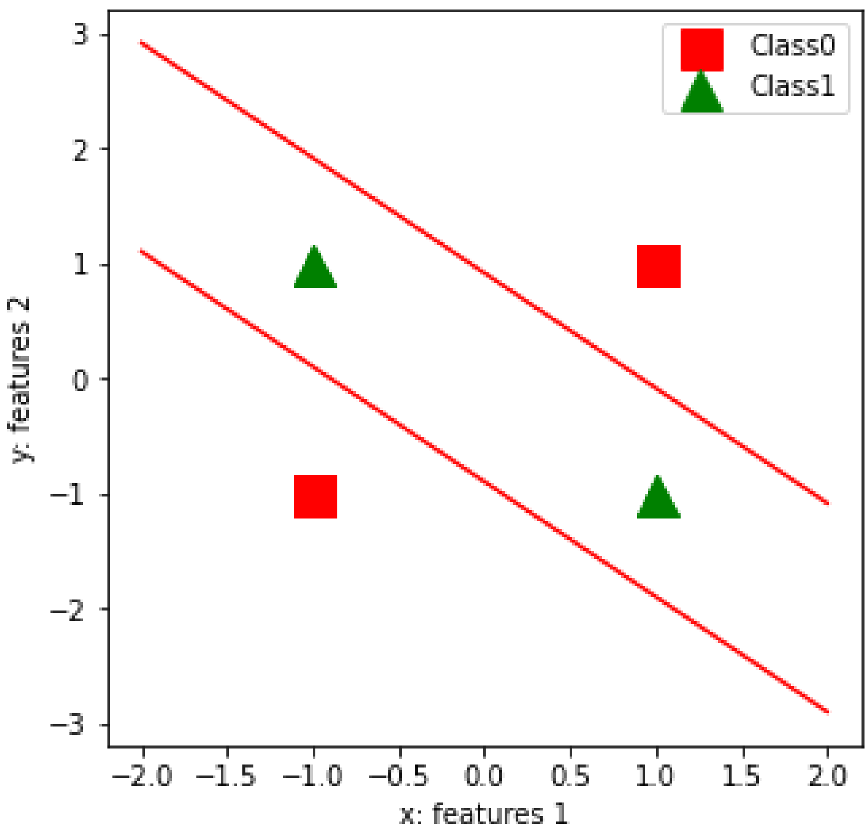
```
accuracy: 1.0000
Epoch 144/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0810 -
accuracy: 1.0000
Epoch 145/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0802 -
accuracy: 1.0000
Epoch 146/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0794 -
accuracy: 1.0000
Epoch 147/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0787 -
accuracy: 1.0000
Epoch 148/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0779 -
accuracy: 1.0000
Epoch 149/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0772 -
accuracy: 1.0000
Epoch 150/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0765 -
accuracy: 1.0000
Epoch 151/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0758 -
accuracy: 1.0000
Epoch 152/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0751 -
accuracy: 1.0000
Epoch 153/200
4/4 [==============================] - 0s 2ms/step - loss: 0.0745 -
accuracy: 1.0000
Epoch 154/200
4/4 [==============================] - 0s 2ms/step - loss: 0.0738 -
accuracy: 1.0000
Epoch 155/200
4/4 [==============================] - 0s 2ms/step - loss: 0.0731 -
accuracy: 1.0000
Epoch 156/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0725 -
accuracy: 1.0000
Epoch 157/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0719 -
accuracy: 1.0000
Epoch 158/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0713 -
accuracy: 1.0000
Epoch 159/200
4/4 [==============================] - 0s 2ms/step - loss: 0.0707 -
accuracy: 1.0000
Epoch 160/200
```

```
4/4 [==============================] - 0s 3ms/step - loss: 0.0701 -
accuracy: 1.0000
Epoch 161/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0695 -
accuracy: 1.0000
Epoch 162/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0689 -
accuracy: 1.0000
Epoch 163/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0683 -
accuracy: 1.0000
Epoch 164/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0678 -
accuracy: 1.0000
Epoch 165/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0672 -
accuracy: 1.0000
Epoch 166/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0667 -
accuracy: 1.0000
Epoch 167/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0661 -
accuracy: 1.0000
Epoch 168/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0656 -
accuracy: 1.0000
Epoch 169/200
4/4 [==============================] - 0s 6ms/step - loss: 0.0651 -
accuracy: 1.0000
Epoch 170/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0646 -
accuracy: 1.0000
Epoch 171/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0641 -
accuracy: 1.0000
Epoch 172/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0636 -
accuracy: 1.0000
Epoch 173/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0631 -
accuracy: 1.0000
Epoch 174/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0626 -
accuracy: 1.0000
Epoch 175/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0621 -
accuracy: 1.0000
Epoch 176/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0617 -
accuracy: 1.0000
```

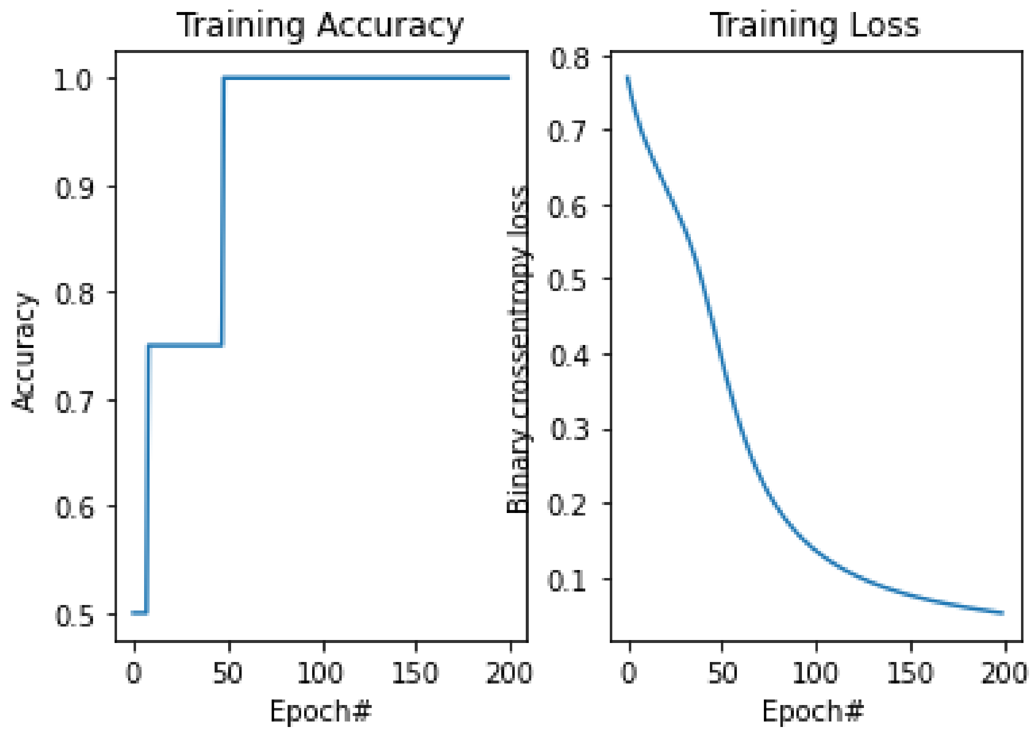
```
Epoch 177/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0612 -
accuracy: 1.0000
Epoch 178/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0607 -
accuracy: 1.0000
Epoch 179/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0603 -
accuracy: 1.0000
Epoch 180/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0598 -
accuracy: 1.0000
Epoch 181/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0594 -
accuracy: 1.0000
Epoch 182/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0590 -
accuracy: 1.0000
Epoch 183/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0586 -
accuracy: 1.0000
Epoch 184/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0581 -
accuracy: 1.0000
Epoch 185/200
4/4 [==============================] - 0s 5ms/step - loss: 0.0577 -
accuracy: 1.0000
Epoch 186/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0573 -
accuracy: 1.0000
Epoch 187/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0569 -
accuracy: 1.0000
Epoch 188/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0565 -
accuracy: 1.0000
Epoch 189/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0561 -
accuracy: 1.0000
Epoch 190/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0557 -
accuracy: 1.0000
Epoch 191/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0554 -
accuracy: 1.0000
Epoch 192/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0550 -
accuracy: 1.0000
Epoch 193/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0546 -
```

```
accuracy: 1.0000
Epoch 194/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0543 -
accuracy: 1.0000
Epoch 195/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0539 -
accuracy: 1.0000
Epoch 196/200
4/4 [==============================] - 0s 5ms/step - loss: 0.0535 -
accuracy: 1.0000
Epoch 197/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0532 -
accuracy: 1.0000
Epoch 198/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0528 -
accuracy: 1.0000
Epoch 199/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0525 -
accuracy: 1.0000
Epoch 200/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0522 -
accuracy: 1.0000
```

```
# part g

# Use the trained weights and plot the final classifier lines in the
plot of part (b)
weights = model_a.layers[0].get_weights()
plt.figure(figsize=[5,5])
for node_i in range(weights[0].shape[1]):
  thre_parms = np.array(weights[0][:, node_i]) #This first item is the
weights for the inputs
  thre_parms = np.append(thre_parms, weights[1][node_i]) #second item
the weights for the bias
  plot_fun_thr(features, labels, thre_parms, classes)

plt.show()
```

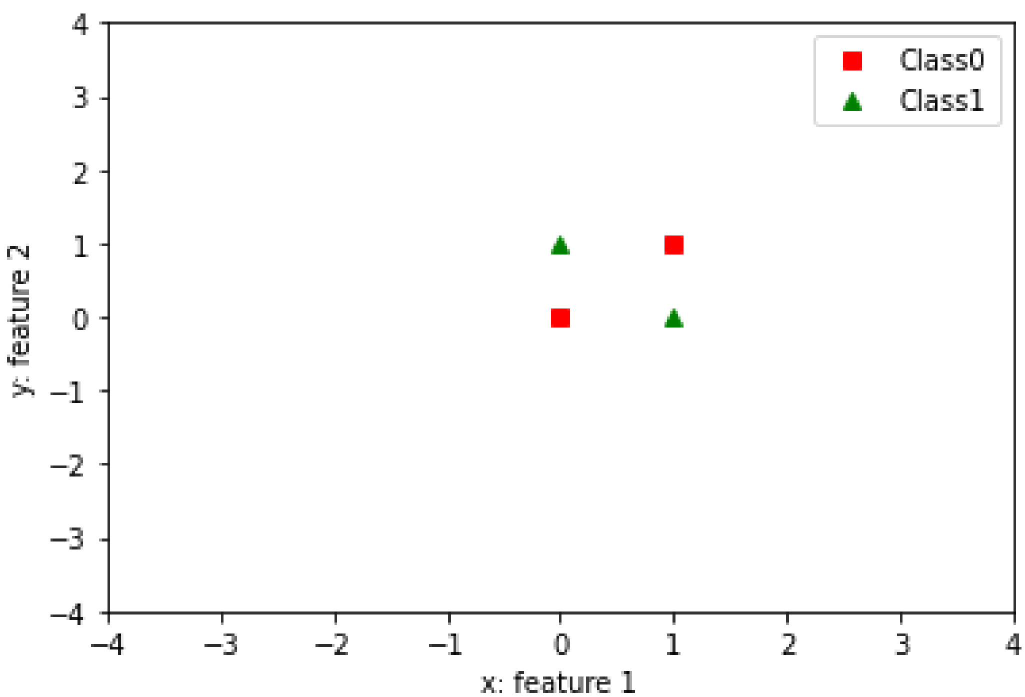



```
# part h

#plot the training loss
acc_curve = np.array(history.history['accuracy'])
loss_curve = np.array(history.history['loss'])
plot_curve(acc_curve, loss_curve)
```



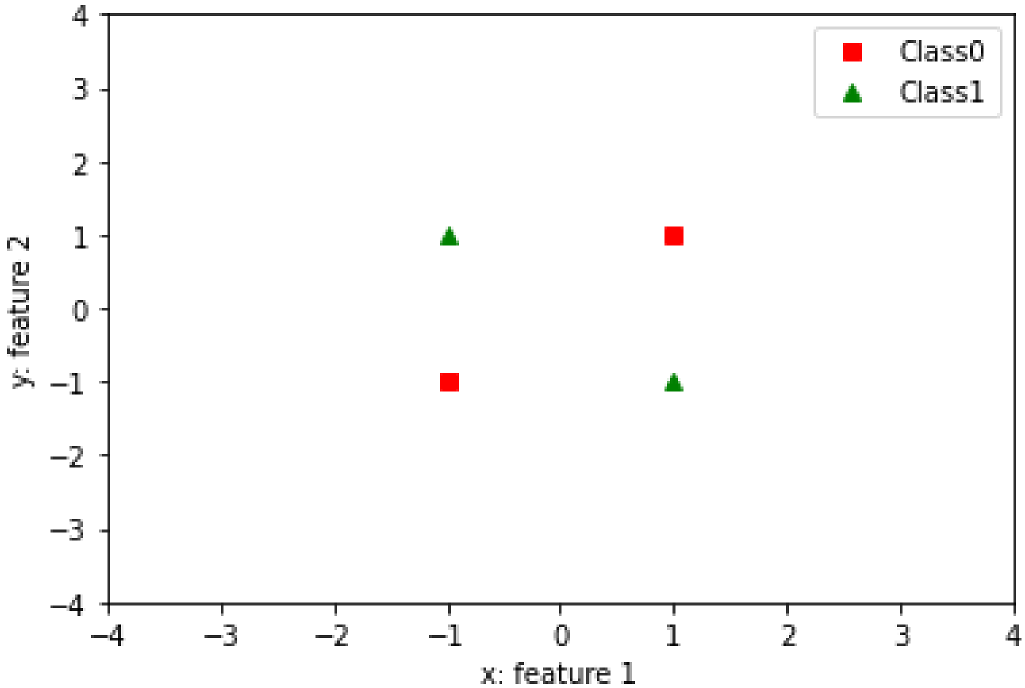

```
# part i

features = np.array([[0, 0],[0,1], [1,0], [1,1]])
labels = np.array([0, 1, 1, 0])
classes = [0, 1]

plot_fun(features, labels, classes)

# defining the model
model_b = Sequential()
model_b.add(Dense(input_dim = 2, units = 4, activation = 'tanh'))
model_b.add(Dense(units = 1, activation = 'sigmoid'))
model_b.summary()

# compiling the model
opt = tf.keras.optimizers.SGD(learning_rate = 0.1)
model_b.compile(loss = 'binary_crossentropy', optimizer = opt, metrics
= ['accuracy'])

# normalization
features = (features - np.mean(features, axis = 0)) / np.std(features,
axis = 0) # normalization
plot_fun(features, labels, classes)

#train the network for 200 epochs and a batch size of 1
history = model_b.fit(features, labels, batch_size = 1, epochs = 400,
verbose = 1)
```

```
# Use the trained weights and plot the final classifier lines in the
plot of part (b)
weights = model_b.layers[0].get_weights()
plt.figure(figsize=[5,5])
for node_i in range(weights[0].shape[1]):
  thre_parms = np.array(weights[0][:, node_i]) #This first item is the
weights for the inputs
  thre_parms = np.append(thre_parms, weights[1][node_i]) #second item
the weights for the bias
  plot_fun_thr(features, labels, thre_parms, classes)

plt.show()

#plot the training loss
acc_curve = np.array(history.history['accuracy'])
loss_curve = np.array(history.history['loss'])
plot_curve(acc_curve, loss_curve)
```




```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_23 (Dense)            (None, 4)                 12

 dense_24 (Dense)            (None, 1)                 5

=================================================================
```

```
Total params: 17
Trainable params: 17
Non-trainable params: 0
_________________________________________________________________
```




```
Epoch 1/400
4/4 [==============================] - 0s 3ms/step - loss: 0.8352 -
accuracy: 0.5000
Epoch 2/400
4/4 [==============================] - 0s 3ms/step - loss: 0.7911 -
accuracy: 0.5000
Epoch 3/400
4/4 [==============================] - 0s 4ms/step - loss: 0.7588 -
accuracy: 0.7500
Epoch 4/400
4/4 [==============================] - 0s 3ms/step - loss: 0.7365 -
accuracy: 0.5000
Epoch 5/400
4/4 [==============================] - 0s 4ms/step - loss: 0.7143 -
accuracy: 0.7500
Epoch 6/400
4/4 [==============================] - 0s 3ms/step - loss: 0.6954 -
accuracy: 0.7500
Epoch 7/400
4/4 [==============================] - 0s 3ms/step - loss: 0.6761 -
accuracy: 0.7500
Epoch 8/400
4/4 [==============================] - 0s 3ms/step - loss: 0.6561 -
accuracy: 0.7500
```

```
Epoch 9/400
4/4 [==============================] - 0s 3ms/step - loss: 0.6373 -
accuracy: 0.7500
Epoch 10/400
4/4 [==============================] - 0s 3ms/step - loss: 0.6175 -
accuracy: 0.7500
Epoch 11/400
4/4 [==============================] - 0s 3ms/step - loss: 0.5989 -
accuracy: 0.7500
Epoch 12/400
4/4 [==============================] - 0s 3ms/step - loss: 0.5777 -
accuracy: 0.7500
Epoch 13/400
4/4 [==============================] - 0s 4ms/step - loss: 0.5585 -
accuracy: 0.7500
Epoch 14/400
4/4 [==============================] - 0s 4ms/step - loss: 0.5388 -
accuracy: 0.7500
Epoch 15/400
4/4 [==============================] - 0s 3ms/step - loss: 0.5199 -
accuracy: 0.7500
Epoch 16/400
4/4 [==============================] - 0s 3ms/step - loss: 0.5008 -
accuracy: 0.7500
Epoch 17/400
4/4 [==============================] - 0s 3ms/step - loss: 0.4828 -
accuracy: 1.0000
Epoch 18/400
4/4 [==============================] - 0s 3ms/step - loss: 0.4648 -
accuracy: 1.0000
Epoch 19/400
4/4 [==============================] - 0s 3ms/step - loss: 0.4477 -
accuracy: 1.0000
Epoch 20/400
4/4 [==============================] - 0s 3ms/step - loss: 0.4312 -
accuracy: 1.0000
Epoch 21/400
4/4 [==============================] - 0s 3ms/step - loss: 0.4147 -
accuracy: 1.0000
Epoch 22/400
4/4 [==============================] - 0s 3ms/step - loss: 0.3987 -
accuracy: 1.0000
Epoch 23/400
4/4 [==============================] - 0s 3ms/step - loss: 0.3835 -
accuracy: 1.0000
Epoch 24/400
4/4 [==============================] - 0s 4ms/step - loss: 0.3691 -
accuracy: 1.0000
Epoch 25/400
4/4 [==============================] - 0s 4ms/step - loss: 0.3546 -
```

```
accuracy: 1.0000
Epoch 26/400
4/4 [==============================] - 0s 4ms/step - loss: 0.3412 -
accuracy: 1.0000
Epoch 27/400
4/4 [==============================] - 0s 3ms/step - loss: 0.3285 -
accuracy: 1.0000
Epoch 28/400
4/4 [==============================] - 0s 3ms/step - loss: 0.3161 -
accuracy: 1.0000
Epoch 29/400
4/4 [==============================] - 0s 2ms/step - loss: 0.3044 -
accuracy: 1.0000
Epoch 30/400
4/4 [==============================] - 0s 4ms/step - loss: 0.2932 -
accuracy: 1.0000
Epoch 31/400
4/4 [==============================] - 0s 3ms/step - loss: 0.2826 -
accuracy: 1.0000
Epoch 32/400
4/4 [==============================] - 0s 3ms/step - loss: 0.2725 -
accuracy: 1.0000
Epoch 33/400
4/4 [==============================] - 0s 3ms/step - loss: 0.2629 -
accuracy: 1.0000
Epoch 34/400
4/4 [==============================] - 0s 4ms/step - loss: 0.2538 -
accuracy: 1.0000
Epoch 35/400
4/4 [==============================] - 0s 4ms/step - loss: 0.2451 -
accuracy: 1.0000
Epoch 36/400
4/4 [==============================] - 0s 3ms/step - loss: 0.2368 -
accuracy: 1.0000
Epoch 37/400
4/4 [==============================] - 0s 3ms/step - loss: 0.2290 -
accuracy: 1.0000
Epoch 38/400
4/4 [==============================] - 0s 3ms/step - loss: 0.2215 -
accuracy: 1.0000
Epoch 39/400
4/4 [==============================] - 0s 5ms/step - loss: 0.2145 -
accuracy: 1.0000
Epoch 40/400
4/4 [==============================] - 0s 3ms/step - loss: 0.2077 -
accuracy: 1.0000
Epoch 41/400
4/4 [==============================] - 0s 3ms/step - loss: 0.2013 -
accuracy: 1.0000
Epoch 42/400
```

```
4/4 [==============================] - 0s 4ms/step - loss: 0.1952 -
accuracy: 1.0000
Epoch 43/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1894 -
accuracy: 1.0000
Epoch 44/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1838 -
accuracy: 1.0000
Epoch 45/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1786 -
accuracy: 1.0000
Epoch 46/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1736 -
accuracy: 1.0000
Epoch 47/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1688 -
accuracy: 1.0000
Epoch 48/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1643 -
accuracy: 1.0000
Epoch 49/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1599 -
accuracy: 1.0000
Epoch 50/400
4/4 [==============================] - 0s 4ms/step - loss: 0.1557 -
accuracy: 1.0000
Epoch 51/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1518 -
accuracy: 1.0000
Epoch 52/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1479 -
accuracy: 1.0000
Epoch 53/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1443 -
accuracy: 1.0000
Epoch 54/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1408 -
accuracy: 1.0000
Epoch 55/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1375 -
accuracy: 1.0000
Epoch 56/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1343 -
accuracy: 1.0000
Epoch 57/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1312 -
accuracy: 1.0000
Epoch 58/400
4/4 [==============================] - 0s 4ms/step - loss: 0.1282 -
accuracy: 1.0000
```

```
Epoch 59/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1254 -
accuracy: 1.0000
Epoch 60/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1227 -
accuracy: 1.0000
Epoch 61/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1201 -
accuracy: 1.0000
Epoch 62/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1175 -
accuracy: 1.0000
Epoch 63/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1151 -
accuracy: 1.0000
Epoch 64/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1128 -
accuracy: 1.0000
Epoch 65/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1105 -
accuracy: 1.0000
Epoch 66/400
4/4 [==============================] - 0s 2ms/step - loss: 0.1083 -
accuracy: 1.0000
Epoch 67/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1063 -
accuracy: 1.0000
Epoch 68/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1042 -
accuracy: 1.0000
Epoch 69/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1023 -
accuracy: 1.0000
Epoch 70/400
4/4 [==============================] - 0s 3ms/step - loss: 0.1004 -
accuracy: 1.0000
Epoch 71/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0986 -
accuracy: 1.0000
Epoch 72/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0968 -
accuracy: 1.0000
Epoch 73/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0951 -
accuracy: 1.0000
Epoch 74/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0935 -
accuracy: 1.0000
Epoch 75/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0919 -
```

```
accuracy: 1.0000
Epoch 76/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0903 -
accuracy: 1.0000
Epoch 77/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0888 -
accuracy: 1.0000
Epoch 78/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0874 -
accuracy: 1.0000
Epoch 79/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0860 -
accuracy: 1.0000
Epoch 80/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0846 -
accuracy: 1.0000
Epoch 81/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0833 -
accuracy: 1.0000
Epoch 82/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0820 -
accuracy: 1.0000
Epoch 83/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0807 -
accuracy: 1.0000
Epoch 84/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0795 -
accuracy: 1.0000
Epoch 85/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0783 -
accuracy: 1.0000
Epoch 86/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0772 -
accuracy: 1.0000
Epoch 87/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0761 -
accuracy: 1.0000
Epoch 88/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0750 -
accuracy: 1.0000
Epoch 89/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0739 -
accuracy: 1.0000
Epoch 90/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0729 -
accuracy: 1.0000
Epoch 91/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0719 -
accuracy: 1.0000
Epoch 92/400
```

```
4/4 [==============================] - 0s 3ms/step - loss: 0.0709 -
accuracy: 1.0000
Epoch 93/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0700 -
accuracy: 1.0000
Epoch 94/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0690 -
accuracy: 1.0000
Epoch 95/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0681 -
accuracy: 1.0000
Epoch 96/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0672 -
accuracy: 1.0000
Epoch 97/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0664 -
accuracy: 1.0000
Epoch 98/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0655 -
accuracy: 1.0000
Epoch 99/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0647 -
accuracy: 1.0000
Epoch 100/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0639 -
accuracy: 1.0000
Epoch 101/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0631 -
accuracy: 1.0000
Epoch 102/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0624 -
accuracy: 1.0000
Epoch 103/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0616 -
accuracy: 1.0000
Epoch 104/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0609 -
accuracy: 1.0000
Epoch 105/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0602 -
accuracy: 1.0000
Epoch 106/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0595 -
accuracy: 1.0000
Epoch 107/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0588 -
accuracy: 1.0000
Epoch 108/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0581 -
accuracy: 1.0000
```

```
Epoch 109/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0575 -
accuracy: 1.0000
Epoch 110/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0569 -
accuracy: 1.0000
Epoch 111/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0562 -
accuracy: 1.0000
Epoch 112/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0556 -
accuracy: 1.0000
Epoch 113/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0550 -
accuracy: 1.0000
Epoch 114/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0544 -
accuracy: 1.0000
Epoch 115/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0539 -
accuracy: 1.0000
Epoch 116/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0533 -
accuracy: 1.0000
Epoch 117/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0527 -
accuracy: 1.0000
Epoch 118/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0522 -
accuracy: 1.0000
Epoch 119/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0517 -
accuracy: 1.0000
Epoch 120/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0512 -
accuracy: 1.0000
Epoch 121/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0506 -
accuracy: 1.0000
Epoch 122/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0501 -
accuracy: 1.0000
Epoch 123/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0497 -
accuracy: 1.0000
Epoch 124/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0492 -
accuracy: 1.0000
Epoch 125/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0487 -
```

```
accuracy: 1.0000
Epoch 126/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0482 -
accuracy: 1.0000
Epoch 127/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0478 -
accuracy: 1.0000
Epoch 128/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0473 -
accuracy: 1.0000
Epoch 129/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0469 -
accuracy: 1.0000
Epoch 130/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0465 -
accuracy: 1.0000
Epoch 131/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0460 -
accuracy: 1.0000
Epoch 132/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0456 -
accuracy: 1.0000
Epoch 133/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0452 -
accuracy: 1.0000
Epoch 134/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0448 -
accuracy: 1.0000
Epoch 135/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0444 -
accuracy: 1.0000
Epoch 136/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0440 -
accuracy: 1.0000
Epoch 137/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0437 -
accuracy: 1.0000
Epoch 138/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0433 -
accuracy: 1.0000
Epoch 139/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0429 -
accuracy: 1.0000
Epoch 140/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0425 -
accuracy: 1.0000
Epoch 141/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0422 -
accuracy: 1.0000
Epoch 142/400
```

```
4/4 [==============================] - 0s 3ms/step - loss: 0.0418 -
accuracy: 1.0000
Epoch 143/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0415 -
accuracy: 1.0000
Epoch 144/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0412 -
accuracy: 1.0000
Epoch 145/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0408 -
accuracy: 1.0000
Epoch 146/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0405 -
accuracy: 1.0000
Epoch 147/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0402 -
accuracy: 1.0000
Epoch 148/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0398 -
accuracy: 1.0000
Epoch 149/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0395 -
accuracy: 1.0000
Epoch 150/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0392 -
accuracy: 1.0000
Epoch 151/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0389 -
accuracy: 1.0000
Epoch 152/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0386 -
accuracy: 1.0000
Epoch 153/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0383 -
accuracy: 1.0000
Epoch 154/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0380 -
accuracy: 1.0000
Epoch 155/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0377 -
accuracy: 1.0000
Epoch 156/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0375 -
accuracy: 1.0000
Epoch 157/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0372 -
accuracy: 1.0000
Epoch 158/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0369 -
accuracy: 1.0000
```

```
Epoch 159/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0366 -
accuracy: 1.0000
Epoch 160/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0364 -
accuracy: 1.0000
Epoch 161/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0361 -
accuracy: 1.0000
Epoch 162/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0359 -
accuracy: 1.0000
Epoch 163/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0356 -
accuracy: 1.0000
Epoch 164/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0353 -
accuracy: 1.0000
Epoch 165/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0351 -
accuracy: 1.0000
Epoch 166/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0349 -
accuracy: 1.0000
Epoch 167/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0346 -
accuracy: 1.0000
Epoch 168/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0344 -
accuracy: 1.0000
Epoch 169/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0341 -
accuracy: 1.0000
Epoch 170/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0339 -
accuracy: 1.0000
Epoch 171/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0337 -
accuracy: 1.0000
Epoch 172/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0334 -
accuracy: 1.0000
Epoch 173/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0332 -
accuracy: 1.0000
Epoch 174/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0330 -
accuracy: 1.0000
Epoch 175/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0328 -
```

```
accuracy: 1.0000
Epoch 176/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0326 -
accuracy: 1.0000
Epoch 177/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0324 -
accuracy: 1.0000
Epoch 178/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0322 -
accuracy: 1.0000
Epoch 179/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0319 -
accuracy: 1.0000
Epoch 180/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0317 -
accuracy: 1.0000
Epoch 181/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0315 -
accuracy: 1.0000
Epoch 182/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0313 -
accuracy: 1.0000
Epoch 183/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0311 -
accuracy: 1.0000
Epoch 184/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0310 -
accuracy: 1.0000
Epoch 185/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0308 -
accuracy: 1.0000
Epoch 186/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0306 -
accuracy: 1.0000
Epoch 187/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0304 -
accuracy: 1.0000
Epoch 188/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0302 -
accuracy: 1.0000
Epoch 189/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0300 -
accuracy: 1.0000
Epoch 190/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0298 -
accuracy: 1.0000
Epoch 191/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0297 -
accuracy: 1.0000
Epoch 192/400
```

```
4/4 [==============================] - 0s 3ms/step - loss: 0.0295 -
accuracy: 1.0000
Epoch 193/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0293 -
accuracy: 1.0000
Epoch 194/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0291 -
accuracy: 1.0000
Epoch 195/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0290 -
accuracy: 1.0000
Epoch 196/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0288 -
accuracy: 1.0000
Epoch 197/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0286 -
accuracy: 1.0000
Epoch 198/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0285 -
accuracy: 1.0000
Epoch 199/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0283 -
accuracy: 1.0000
Epoch 200/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0281 -
accuracy: 1.0000
Epoch 201/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0280 -
accuracy: 1.0000
Epoch 202/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0278 -
accuracy: 1.0000
Epoch 203/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0277 -
accuracy: 1.0000
Epoch 204/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0275 -
accuracy: 1.0000
Epoch 205/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0274 -
accuracy: 1.0000
Epoch 206/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0272 -
accuracy: 1.0000
Epoch 207/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0271 -
accuracy: 1.0000
Epoch 208/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0269 -
accuracy: 1.0000
```

```
Epoch 209/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0268 -
accuracy: 1.0000
Epoch 210/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0266 -
accuracy: 1.0000
Epoch 211/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0265 -
accuracy: 1.0000
Epoch 212/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0264 -
accuracy: 1.0000
Epoch 213/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0262 -
accuracy: 1.0000
Epoch 214/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0261 -
accuracy: 1.0000
Epoch 215/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0259 -
accuracy: 1.0000
Epoch 216/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0258 -
accuracy: 1.0000
Epoch 217/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0257 -
accuracy: 1.0000
Epoch 218/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0255 -
accuracy: 1.0000
Epoch 219/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0254 -
accuracy: 1.0000
Epoch 220/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0253 -
accuracy: 1.0000
Epoch 221/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0251 -
accuracy: 1.0000
Epoch 222/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0250 -
accuracy: 1.0000
Epoch 223/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0249 -
accuracy: 1.0000
Epoch 224/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0248 -
accuracy: 1.0000
Epoch 225/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0246 -
```

```
accuracy: 1.0000
Epoch 226/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0245 -
accuracy: 1.0000
Epoch 227/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0244 -
accuracy: 1.0000
Epoch 228/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0243 -
accuracy: 1.0000
Epoch 229/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0242 -
accuracy: 1.0000
Epoch 230/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0240 -
accuracy: 1.0000
Epoch 231/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0239 -
accuracy: 1.0000
Epoch 232/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0238 -
accuracy: 1.0000
Epoch 233/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0237 -
accuracy: 1.0000
Epoch 234/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0236 -
accuracy: 1.0000
Epoch 235/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0235 -
accuracy: 1.0000
Epoch 236/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0234 -
accuracy: 1.0000
Epoch 237/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0233 -
accuracy: 1.0000
Epoch 238/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0231 -
accuracy: 1.0000
Epoch 239/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0230 -
accuracy: 1.0000
Epoch 240/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0229 -
accuracy: 1.0000
Epoch 241/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0228 -
accuracy: 1.0000
Epoch 242/400
```

```
4/4 [==============================] - 0s 3ms/step - loss: 0.0227 -
accuracy: 1.0000
Epoch 243/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0226 -
accuracy: 1.0000
Epoch 244/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0225 -
accuracy: 1.0000
Epoch 245/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0224 -
accuracy: 1.0000
Epoch 246/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0223 -
accuracy: 1.0000
Epoch 247/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0222 -
accuracy: 1.0000
Epoch 248/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0221 -
accuracy: 1.0000
Epoch 249/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0220 -
accuracy: 1.0000
Epoch 250/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0219 -
accuracy: 1.0000
Epoch 251/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0218 -
accuracy: 1.0000
Epoch 252/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0217 -
accuracy: 1.0000
Epoch 253/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0216 -
accuracy: 1.0000
Epoch 254/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0215 -
accuracy: 1.0000
Epoch 255/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0214 -
accuracy: 1.0000
Epoch 256/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0213 -
accuracy: 1.0000
Epoch 257/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0213 -
accuracy: 1.0000
Epoch 258/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0212 -
accuracy: 1.0000
```

```
Epoch 259/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0211 -
accuracy: 1.0000
Epoch 260/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0210 -
accuracy: 1.0000
Epoch 261/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0209 -
accuracy: 1.0000
Epoch 262/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0208 -
accuracy: 1.0000
Epoch 263/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0207 -
accuracy: 1.0000
Epoch 264/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0206 -
accuracy: 1.0000
Epoch 265/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0205 -
accuracy: 1.0000
Epoch 266/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0205 -
accuracy: 1.0000
Epoch 267/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0204 -
accuracy: 1.0000
Epoch 268/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0203 -
accuracy: 1.0000
Epoch 269/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0202 -
accuracy: 1.0000
Epoch 270/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0201 -
accuracy: 1.0000
Epoch 271/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0200 -
accuracy: 1.0000
Epoch 272/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0200 -
accuracy: 1.0000
Epoch 273/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0199 -
accuracy: 1.0000
Epoch 274/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0198 -
accuracy: 1.0000
Epoch 275/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0197 -
```

```
accuracy: 1.0000
Epoch 276/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0196 -
accuracy: 1.0000
Epoch 277/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0196 -
accuracy: 1.0000
Epoch 278/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0195 -
accuracy: 1.0000
Epoch 279/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0194 -
accuracy: 1.0000
Epoch 280/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0193 -
accuracy: 1.0000
Epoch 281/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0193 -
accuracy: 1.0000
Epoch 282/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0192 -
accuracy: 1.0000
Epoch 283/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0191 -
accuracy: 1.0000
Epoch 284/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0190 -
accuracy: 1.0000
Epoch 285/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0190 -
accuracy: 1.0000
Epoch 286/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0189 -
accuracy: 1.0000
Epoch 287/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0188 -
accuracy: 1.0000
Epoch 288/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0187 -
accuracy: 1.0000
Epoch 289/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0187 -
accuracy: 1.0000
Epoch 290/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0186 -
accuracy: 1.0000
Epoch 291/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0185 -
accuracy: 1.0000
Epoch 292/400
```

```
4/4 [==============================] - 0s 4ms/step - loss: 0.0185 -
accuracy: 1.0000
Epoch 293/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0184 -
accuracy: 1.0000
Epoch 294/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0183 -
accuracy: 1.0000
Epoch 295/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0183 -
accuracy: 1.0000
Epoch 296/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0182 -
accuracy: 1.0000
Epoch 297/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0181 -
accuracy: 1.0000
Epoch 298/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0181 -
accuracy: 1.0000
Epoch 299/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0180 -
accuracy: 1.0000
Epoch 300/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0179 -
accuracy: 1.0000
Epoch 301/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0179 -
accuracy: 1.0000
Epoch 302/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0178 -
accuracy: 1.0000
Epoch 303/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0177 -
accuracy: 1.0000
Epoch 304/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0177 -
accuracy: 1.0000
Epoch 305/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0176 -
accuracy: 1.0000
Epoch 306/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0175 -
accuracy: 1.0000
Epoch 307/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0175 -
accuracy: 1.0000
Epoch 308/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0174 -
accuracy: 1.0000
```

```
Epoch 309/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0174 -
accuracy: 1.0000
Epoch 310/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0173 -
accuracy: 1.0000
Epoch 311/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0172 -
accuracy: 1.0000
Epoch 312/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0172 -
accuracy: 1.0000
Epoch 313/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0171 -
accuracy: 1.0000
Epoch 314/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0171 -
accuracy: 1.0000
Epoch 315/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0170 -
accuracy: 1.0000
Epoch 316/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0169 -
accuracy: 1.0000
Epoch 317/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0169 -
accuracy: 1.0000
Epoch 318/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0168 -
accuracy: 1.0000
Epoch 319/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0168 -
accuracy: 1.0000
Epoch 320/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0167 -
accuracy: 1.0000
Epoch 321/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0167 -
accuracy: 1.0000
Epoch 322/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0166 -
accuracy: 1.0000
Epoch 323/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0165 -
accuracy: 1.0000
Epoch 324/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0165 -
accuracy: 1.0000
Epoch 325/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0164 -
```

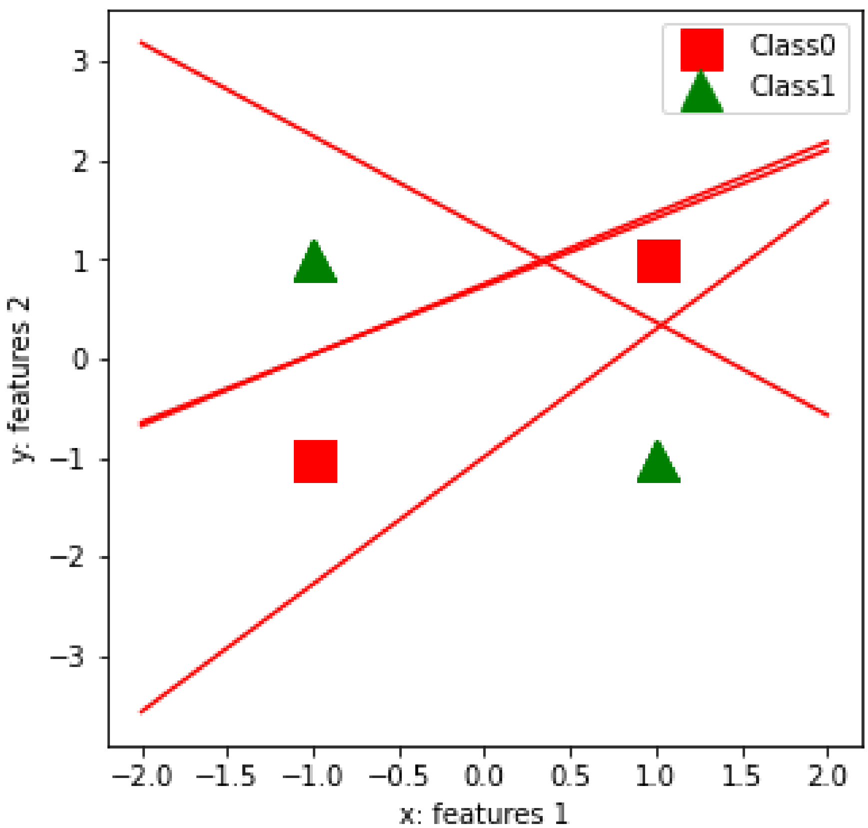
```
accuracy: 1.0000
Epoch 326/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0164 -
accuracy: 1.0000
Epoch 327/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0163 -
accuracy: 1.0000
Epoch 328/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0163 -
accuracy: 1.0000
Epoch 329/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0162 -
accuracy: 1.0000
Epoch 330/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0162 -
accuracy: 1.0000
Epoch 331/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0161 -
accuracy: 1.0000
Epoch 332/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0161 -
accuracy: 1.0000
Epoch 333/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0160 -
accuracy: 1.0000
Epoch 334/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0159 -
accuracy: 1.0000
Epoch 335/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0159 -
accuracy: 1.0000
Epoch 336/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0158 -
accuracy: 1.0000
Epoch 337/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0158 -
accuracy: 1.0000
Epoch 338/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0157 -
accuracy: 1.0000
Epoch 339/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0157 -
accuracy: 1.0000
Epoch 340/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0156 -
accuracy: 1.0000
Epoch 341/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0156 -
accuracy: 1.0000
Epoch 342/400
```

```
4/4 [==============================] - 0s 3ms/step - loss: 0.0155 -
accuracy: 1.0000
Epoch 343/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0155 -
accuracy: 1.0000
Epoch 344/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0154 -
accuracy: 1.0000
Epoch 345/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0154 -
accuracy: 1.0000
Epoch 346/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0154 -
accuracy: 1.0000
Epoch 347/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0153 -
accuracy: 1.0000
Epoch 348/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0153 -
accuracy: 1.0000
Epoch 349/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0152 -
accuracy: 1.0000
Epoch 350/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0152 -
accuracy: 1.0000
Epoch 351/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0151 -
accuracy: 1.0000
Epoch 352/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0151 -
accuracy: 1.0000
Epoch 353/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0150 -
accuracy: 1.0000
Epoch 354/400
4/4 [==============================] - 0s 5ms/step - loss: 0.0150 -
accuracy: 1.0000
Epoch 355/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0149 -
accuracy: 1.0000
Epoch 356/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0149 -
accuracy: 1.0000
Epoch 357/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0148 -
accuracy: 1.0000
Epoch 358/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0148 -
accuracy: 1.0000
```

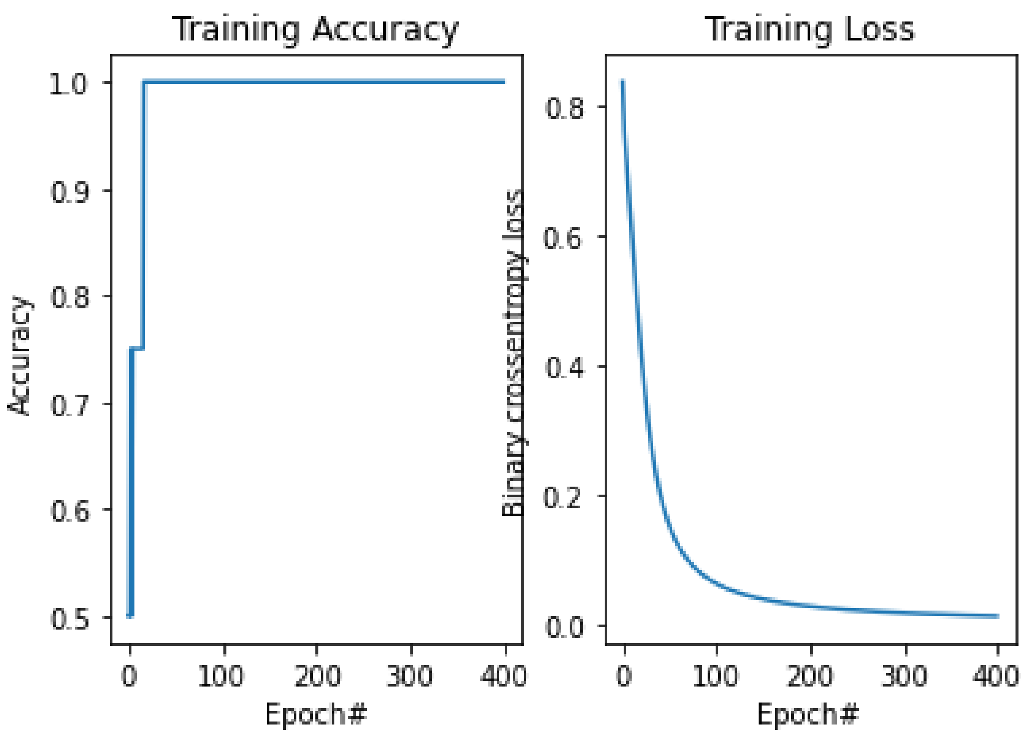
```
Epoch 359/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0148 -
accuracy: 1.0000
Epoch 360/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0147 -
accuracy: 1.0000
Epoch 361/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0147 -
accuracy: 1.0000
Epoch 362/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0146 -
accuracy: 1.0000
Epoch 363/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0146 -
accuracy: 1.0000
Epoch 364/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0145 -
accuracy: 1.0000
Epoch 365/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0145 -
accuracy: 1.0000
Epoch 366/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0144 -
accuracy: 1.0000
Epoch 367/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0144 -
accuracy: 1.0000
Epoch 368/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0144 -
accuracy: 1.0000
Epoch 369/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0143 -
accuracy: 1.0000
Epoch 370/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0143 -
accuracy: 1.0000
Epoch 371/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0142 -
accuracy: 1.0000
Epoch 372/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0142 -
accuracy: 1.0000
Epoch 373/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0142 -
accuracy: 1.0000
Epoch 374/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0141 -
accuracy: 1.0000
Epoch 375/400
4/4 [==============================] - 0s 2ms/step - loss: 0.0141 -
```

```
accuracy: 1.0000
Epoch 376/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0140 -
accuracy: 1.0000
Epoch 377/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0140 -
accuracy: 1.0000
Epoch 378/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0140 -
accuracy: 1.0000
Epoch 379/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0139 -
accuracy: 1.0000
Epoch 380/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0139 -
accuracy: 1.0000
Epoch 381/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0138 -
accuracy: 1.0000
Epoch 382/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0138 -
accuracy: 1.0000
Epoch 383/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0138 -
accuracy: 1.0000
Epoch 384/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0137 -
accuracy: 1.0000
Epoch 385/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0137 -
accuracy: 1.0000
Epoch 386/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0136 -
accuracy: 1.0000
Epoch 387/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0136 -
accuracy: 1.0000
Epoch 388/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0136 -
accuracy: 1.0000
Epoch 389/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0135 -
accuracy: 1.0000
Epoch 390/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0135 -
accuracy: 1.0000
Epoch 391/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0135 -
accuracy: 1.0000
Epoch 392/400
```

```
4/4 [==============================] - 0s 3ms/step - loss: 0.0134 -
accuracy: 1.0000
Epoch 393/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0134 -
accuracy: 1.0000
Epoch 394/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0133 -
accuracy: 1.0000
Epoch 395/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0133 -
accuracy: 1.0000
Epoch 396/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0133 -
accuracy: 1.0000
Epoch 397/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0132 -
accuracy: 1.0000
Epoch 398/400
4/4 [==============================] - 0s 4ms/step - loss: 0.0132 -
accuracy: 1.0000
Epoch 399/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0132 -
accuracy: 1.0000
Epoch 400/400
4/4 [==============================] - 0s 3ms/step - loss: 0.0131 -
accuracy: 1.0000
```



Class0
Class1
3
2
1
0
−1
−2
−3
y: features 2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x: features 1



Training Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
Accuracy
0
100
200
300
400
Epoch#
Training Loss
0.8
0.6
0.4
0.2
0.0
Binary crossentropy loss
0
100
200
300
400
Epoch#

With 4 nodes we can more consistently reach 100% accuracy, with 2 nodes it is very much possible but only gets to 100% accuracy half the time. So if we were trying to use least amount of nodes I would go with two nodes and change the learning rate and keep trying until it randomly learns accurately. But ultimately for efficiency and reliability it would be way more advantageous to use 4 nodes for the hidden layer.

Problem 2: Application of keras to build, compile, and train a nerual network as a three-class classifier for MNIST dataset (0 vs 1 vs 2):

1. Use mnist function in keras.datasets to load MNIST dataset and split it into training and testing sets. Then, randomly select 20% of the training images along with their corresponding labels to be the validation data.
2. Feature extraction: average the pixel values in the quadrants in each image to generate a feature vector of 4 values for each image.
3. Convert the label vectors for all the sets to binary class matrices using to_categorical() Keras function.
4. Build, compile, train, and then evaluate:

- Build a neural network with 1 layer that contains 10 nodes using the Keras library.
- Compile the network. Make sure to select a correct loss function for this classification problem. Use stochastic gradient descent learning (SGD, learning rate of 0.0001). Explain your selection of the loss function.
- Train the network for 50 epochs and a batch size of 16.
- Plot the training loss (i.e., the learning curve) for all the epochs.
- Use the evaluate() Keras function to find the training and validation loss and accuracy.

1. Repeat step (d) for each of the following networks:

Model #	Details	Training		Validation	
		loss	accuracy	loss	accuracy
1	1 layer 10 nodes				
2	1 layer 50 nodes				
3	1 layer 100 nodes				
4	2 layers 100 nodes, 10 nodes				
5	2 layers 100 nodes, 50 nodes				

1. What behavior do you observe in the training loss and the validation loss when you increase the number layers and nodes in the previous table. Which model is more suitable in this problem? Explain.

2. Evaluate the selected model in part (e) on the testing set and report the testing loss and accuracy.

```
from keras.datasets import mnist
import matplotlib.pyplot as plt
import numpy as np
from random import randint
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense

def img_plt(images, labels):
    plt.figure()
    for i in range(1, 11):
        plt.subplot(2, 5, i)
        plt.imshow(images[i - 1, :, :], cmap='gray')
        plt.title('Label: ' + str(labels[i - 1]))
    plt.show()


def feat_extract(images):
    width = images.shape[1]
```

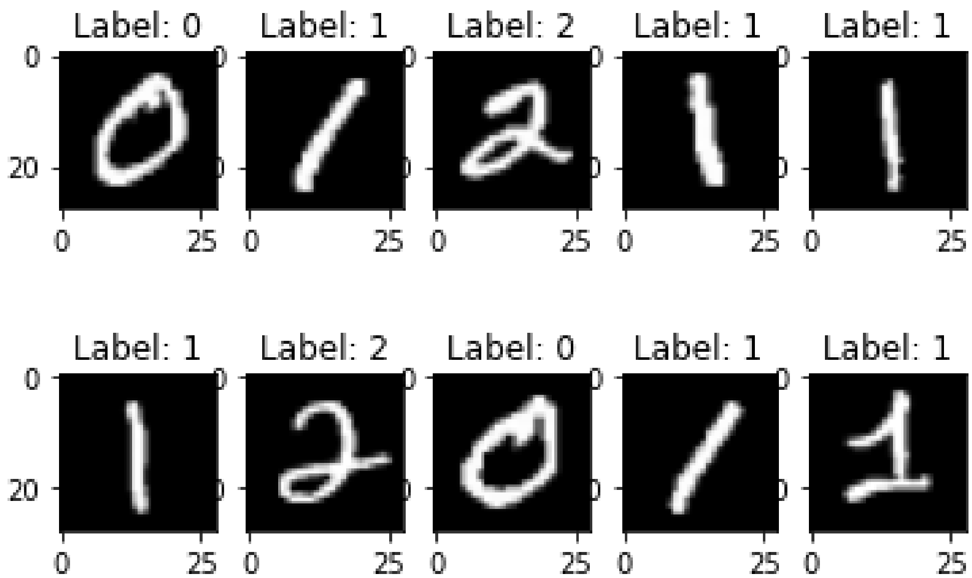
```
    height = images.shape[2]
    features = np.zeros((images.shape[0], 4))
    features_temp = np.sum(images[:, 0:int(width / 2), 0:int(height /
2)], axis=2)
    features[:, 0] = np.sum(features_temp, axis=1) / (width * height /
4)
    features_temp = np.sum(images[:, 0:int(width / 2), int(height /
2):], axis=2)
    features[:, 1] = np.sum(features_temp, axis=1) / (width * height /
4)
    features_temp = np.sum(images[:, int(width / 2):, 0:int(height /
2)], axis=2)
    features[:, 2] = np.sum(features_temp, axis=1) / (width * height /
4)
    features_temp = np.sum(images[:, int(width / 2):, int(height /
2):], axis=2)
    features[:, 3] = np.sum(features_temp, axis=1) / (width * height /
4)
    return features


def feat_plot(features, labels, classes):
    for class_i in classes:
        plt.plot(features[labels[:] == classes[class_i], 0],
features[labels[:] == classes[class_i], 1], 'o',
                 markersize=15)
    plt.xlabel('x: feature 1')
    plt.ylabel('y: feature 2')
    plt.legend(['Class' + str(classes[class_i]) for class_i in
classes])
    plt.show()


def acc_fun(labels_actual, labels_pred):
    acc = np.sum(labels_actual == labels_pred) / len(labels_actual) *
100
    return acc


def plot_curve(accuracy_train, loss_train):
    epochs = np.arange(loss_train.shape[0])
    plt.subplot(1, 2, 1)
    plt.plot(epochs, accuracy_train)
    plt.xlabel('Epoch#')
    plt.ylabel('Accuracy')
    plt.title('Training Accuracy')

    plt.subplot(1, 2, 2)
    plt.plot(epochs, loss_train)
```

```
    plt.xlabel('Epoch#')
    plt.ylabel('Binary crossentropy loss')
    plt.title('Training loss')
    plt.show()


(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Selecting only 0 and 1 digits from the training and testing sets
classes = [0, 1, 2]
x_train_012 = x_train[np.logical_or.reduce((y_train == 0, y_train ==
1, y_train == 2)), 0:28, 0:28]
y_train_012 = y_train[np.logical_or.reduce((y_train == 0, y_train ==
1, y_train == 2))]
print('Samples of the training images')
img_plt(x_train_012[0:10, :, :], y_train_012[0:10])

x_test_012 = x_test[np.logical_or.reduce((y_test == 0, y_test == 1,
y_test == 2)), 0:28, 0:28]
y_test_012 = y_test[np.logical_or.reduce((y_test == 0, y_test == 1,
y_test == 2))]
print('Samples of the testing images')
img_plt(x_test_012[0:10, :, :], y_test_012[0:10])

# shuffling training data
num_train_img = x_train_012.shape[0]
train_ind = np.arange(0, num_train_img)
train_ind_s = np.random.permutation(train_ind)

# 20% of the training set
x_val_012 = x_train_012[train_ind_s[0:int(0.2*num_train_img)], :, :]
y_val_012 = y_train_012[train_ind_s[0:int(0.2*num_train_img)]]

# The rest of the training set
x_train_012 = x_train_012[train_ind_s[int(0.2*num_train_img):], :, :]
y_train_012 = y_train_012[train_ind_s[int(0.2*num_train_img):]]

# x_train_012 = x_train_012[train_ind_s, :, :]
# y_train_012 = y_train_012[train_ind_s]
# # Selecting 500 images for validation
# x_val_012 = x_train_012[0:500, :, :]
# y_val_012 = y_train_012[0:500]
# # The rest of the training set
# x_train_012 = x_train_012[500:, :, :]
# y_train_012 = y_train_012[500:]
print('Samples of the validation images')
img_plt(x_val_012[0:10, :, :], y_val_012[0:10])

# calculating the training, validation, and testing feature (average
```

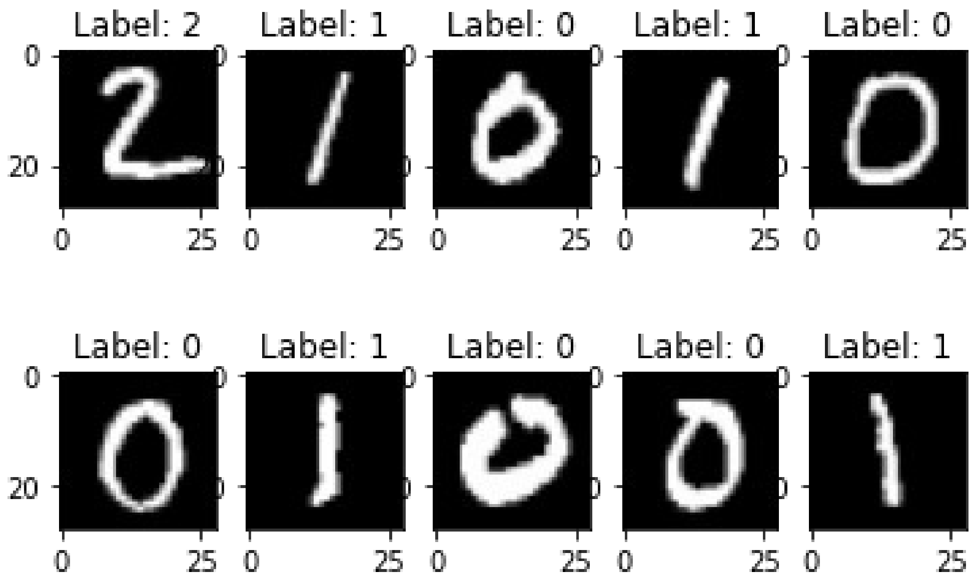
```
of the four quadrants grid)
feature_train = feat_extract(x_train_012)
feature_val = feat_extract(x_val_012)
feature_test = feat_extract(x_test_012)

print('Plotting the features of 500 training images: ')
feat_plot(feature_train[1:500, 0:2], y_train_012[1:500], classes)
feat_plot(feature_train[1:500, 2:4], y_train_012[1:500], classes)
# The combination between the features could be changed

# Defining the model
model_c = Sequential()
model_c.add(Dense(input_dim=4, units=100, activation='tanh'))
model_c.add(Dense(units=len(classes), activation='softmax'))
model_c.summary()

# Compile
opt = tf.keras.optimizers.SGD(learning_rate=0.0001)
model_c.compile(loss='categorical_crossentropy', optimizer=opt,
metrics=['accuracy'])

# Convert class vectors to binary class matricies
from keras.utils.np_utils import to_categorical

y_train_012_c = to_categorical(y_train_012, len(classes))
y_val_012_c = to_categorical(y_val_012, len(classes))
y_test_012_c = to_categorical(y_test_012, len(classes))

# Train
history = model_c.fit(feature_train, y_train_012_c, batch_size=16,
epochs=50, verbose=1)

# Evaluating the model on the training samples
score = model_c.evaluate(feature_train, y_train_012_c)
print('total loss on training set:', score[0])
print('Accuracy of training set', score[1])

# Evaluating the model on the validation samples
score = model_c.evaluate(feature_val, y_val_012_c)
print('total loss on validation set:', score[0])
print('Accuracy of validation set', score[1])

plt.figure(figsize=[9, 5])
acc_curve = np.array(history.history['accuracy'])
loss_curve = np.array(history.history['loss'])
plot_curve(acc_curve, loss_curve)

from sklearn.metrics import accuracy_score, confusion_matrix,
recall_score
```

```python
#Evaluating the model on the held-out samples
score=model_c.evaluate(feature_test,y_test_012_c)
print('Total loss on testing set: ', score[0])
print('Accuracy of testing set: ', score[1])

from sklearn.metrics import accuracy_score, confusion_matrix,
recall_score
#predicting the class of the held-out samples
test_class1_prob=model_c.predict(feature_test)
test_lab=np.argmax(test_class1_prob,axis=1)
print('The accuracy using the testing set: ',
accuracy_score(test_lab,y_test_012))
conf_mat=confusion_matrix(test_lab,y_test_012)
print('The confusion matrix using testing set: \n', conf_mat)
```

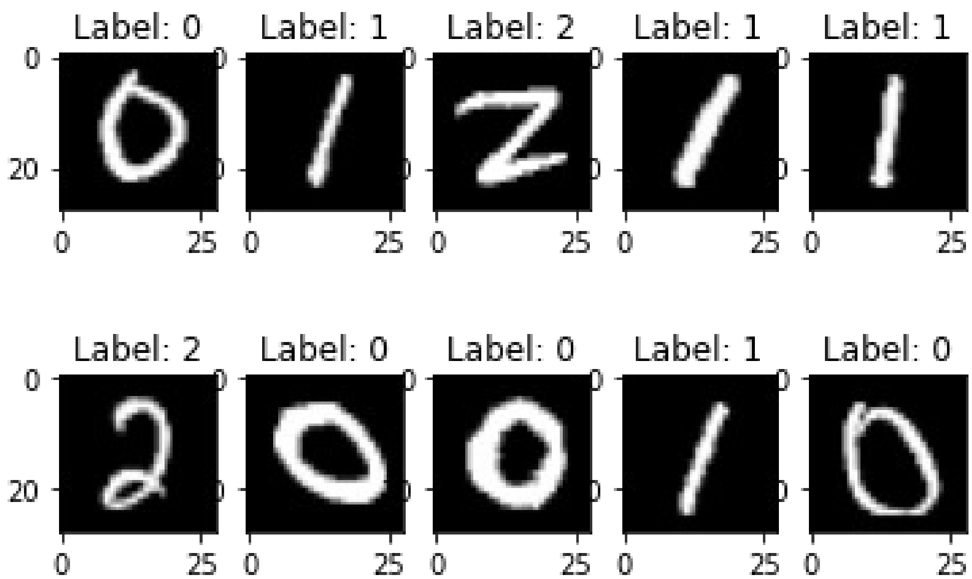
```
Downloading data from https://storage.googleapis.com/tensorflow/tf-
keras-datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
Samples of the training images
```



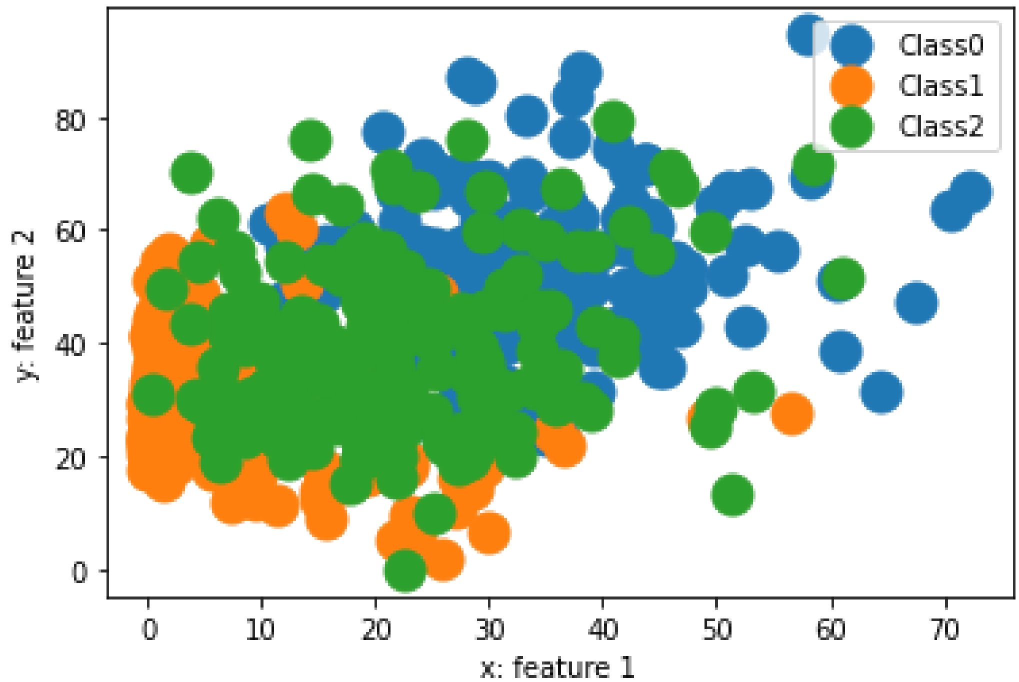

```
Samples of the testing images
```

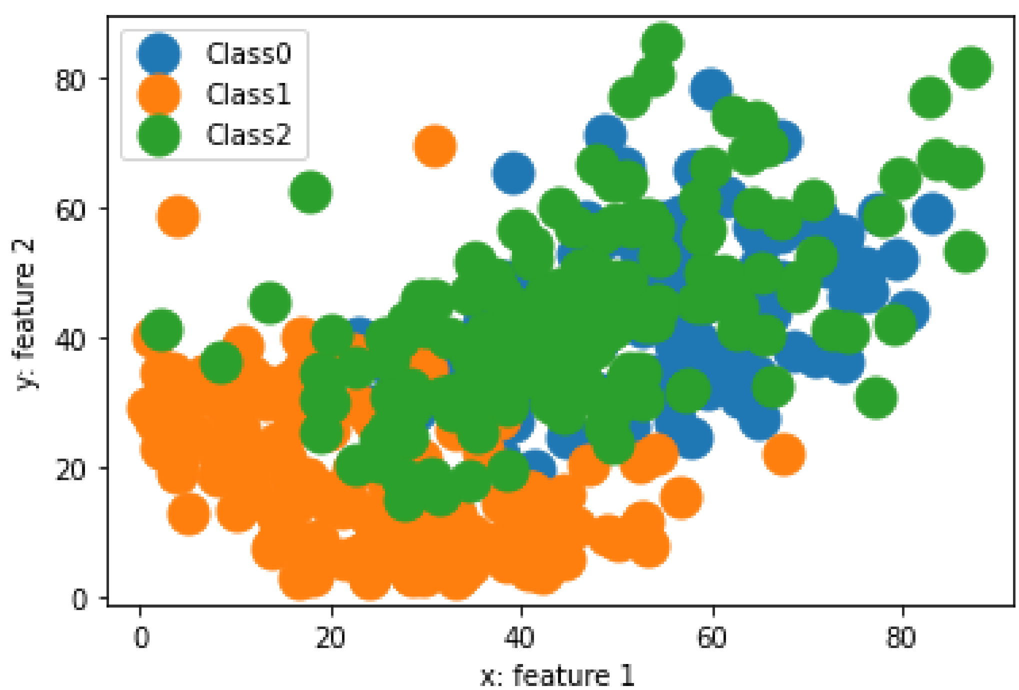


Samples of the validation images



Plotting the features of 500 training images:





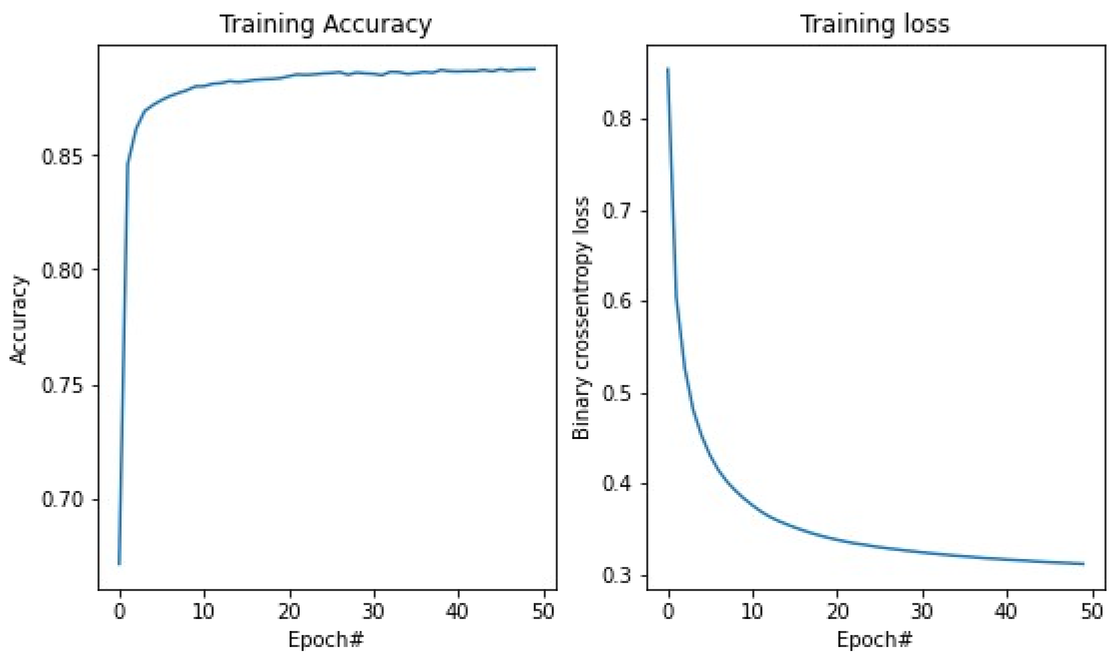

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_25 (Dense)            (None, 100)               500

 dense_26 (Dense)            (None, 3)                 303
```

```
=================================================================
Total params: 803
Trainable params: 803
Non-trainable params: 0
_________________________________________________________________
Epoch 1/50
932/932 [==============================] - 2s 1ms/step - loss: 0.8546
- accuracy: 0.6714
Epoch 2/50
932/932 [==============================] - 1s 1ms/step - loss: 0.6039
- accuracy: 0.8461
Epoch 3/50
932/932 [==============================] - 1s 1ms/step - loss: 0.5268
- accuracy: 0.8616
Epoch 4/50
932/932 [==============================] - 1s 2ms/step - loss: 0.4806
- accuracy: 0.8693
Epoch 5/50
932/932 [==============================] - 1s 1ms/step - loss: 0.4521
- accuracy: 0.8720
Epoch 6/50
932/932 [==============================] - 1s 1ms/step - loss: 0.4305
- accuracy: 0.8742
Epoch 7/50
932/932 [==============================] - 1s 1ms/step - loss: 0.4141
- accuracy: 0.8759
Epoch 8/50
932/932 [==============================] - 1s 1ms/step - loss: 0.4018
- accuracy: 0.8772
Epoch 9/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3918
- accuracy: 0.8784
Epoch 10/50
932/932 [==============================] - 1s 2ms/step - loss: 0.3833
- accuracy: 0.8801
Epoch 11/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3758
- accuracy: 0.8802
Epoch 12/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3691
- accuracy: 0.8813
Epoch 13/50
932/932 [==============================] - 1s 2ms/step - loss: 0.3634
- accuracy: 0.8815
Epoch 14/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3589
- accuracy: 0.8824
Epoch 15/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3551
```

```
- accuracy: 0.8820
Epoch 16/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3515
- accuracy: 0.8824
Epoch 17/50
932/932 [==============================] - 1s 2ms/step - loss: 0.3482
- accuracy: 0.8829
Epoch 18/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3452
- accuracy: 0.8832
Epoch 19/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3425
- accuracy: 0.8833
Epoch 20/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3401
- accuracy: 0.8837
Epoch 21/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3380
- accuracy: 0.8846
Epoch 22/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3359
- accuracy: 0.8854
Epoch 23/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3342
- accuracy: 0.8852
Epoch 24/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3328
- accuracy: 0.8854
Epoch 25/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3313
- accuracy: 0.8858
Epoch 26/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3298
- accuracy: 0.8860
Epoch 27/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3286
- accuracy: 0.8864
Epoch 28/50
932/932 [==============================] - 1s 2ms/step - loss: 0.3274
- accuracy: 0.8852
Epoch 29/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3262
- accuracy: 0.8862
Epoch 30/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3252
- accuracy: 0.8858
Epoch 31/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3241
- accuracy: 0.8856
Epoch 32/50
```

```
932/932 [==============================] - 1s 1ms/step - loss: 0.3231
- accuracy: 0.8850
Epoch 33/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3223
- accuracy: 0.8865
Epoch 34/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3213
- accuracy: 0.8864
Epoch 35/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3205
- accuracy: 0.8856
Epoch 36/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3198
- accuracy: 0.8860
Epoch 37/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3190
- accuracy: 0.8864
Epoch 38/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3181
- accuracy: 0.8861
Epoch 39/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3175
- accuracy: 0.8873
Epoch 40/50
932/932 [==============================] - 1s 2ms/step - loss: 0.3170
- accuracy: 0.8868
Epoch 41/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3162
- accuracy: 0.8866
Epoch 42/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3157
- accuracy: 0.8868
Epoch 43/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3152
- accuracy: 0.8868
Epoch 44/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3145
- accuracy: 0.8873
Epoch 45/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3139
- accuracy: 0.8867
Epoch 46/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3134
- accuracy: 0.8876
Epoch 47/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3129
- accuracy: 0.8870
Epoch 48/50
932/932 [==============================] - 1s 2ms/step - loss: 0.3126
- accuracy: 0.8875
```

```
Epoch 49/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3120
- accuracy: 0.8875
Epoch 50/50
932/932 [==============================] - 1s 1ms/step - loss: 0.3115
- accuracy: 0.8876
466/466 [==============================] - 1s 1ms/step - loss: 0.3108
- accuracy: 0.8876
total loss on training set: 0.31077972054481506
Accuracy of training set 0.8875763416290283
117/117 [==============================] - 0s 1ms/step - loss: 0.3197
- accuracy: 0.8832
total loss on validation set: 0.3197272717952728
Accuracy of validation set 0.8831900954246521
```




```
99/99 [==============================] - 0s 1ms/step - loss: 0.3301 -
accuracy: 0.8796
Total loss on testing set:  0.3300917148590088
Accuracy of testing set:  0.8795678615570068
The accuracy using the testing set:  0.8795678423895774
The confusion matrix using testing set:
 [[ 901   42  174]
 [   9 1054   45]
 [  70   39  813]]
```

Problem 3 Application of Keras to build, compile, and train a neural network to classify songs from Spotify dataset.

The Spotify dataset is a publicly available dataset with information about songs that did and didn't make it in the weekly Hot-100 list issued by Billboard. The goal is to develop a model to predict if a song will make this list. The dataset contains a total of 6,398 tracks with 15 features extracted from the audio features of these tracks. The classes are 1 and 0 which describes whether that track as made it in the Hot-100 list or not respectively.

1. Import the data file (spotify_preprocessed.csv) to your code. The data is preprocessed and ready to use.

2. Shuffle the data then split it into training (90% of the data) and test set (10% of the data). Split the training set further into training and validation sets with 80% and 20% percentages respectively.

3. Build, compile, train, and then evaluate:

- Build a neural network with 2 hidden layers that contain 32 nodes each and an output layer that has 1 unit using the Keras library.
- Compile the network. Select binary cross-entropy (binary_crossentropy) as the loss function. Use stochastic gradient descent learning (SGD, learning rate of 0.01).
- Train the network for 50 epochs and a batch size of 16.
- Plot the training loss and validation loss (i.e., the learning curve) for all the epochs.
- Use the evaluate() Keras function to find the training and validation loss and accuracy.

1. Try different design ideas with the model until you get the best training and validation performance. For example, changing the number of hidden layers and number of units in each, changing the loss function, the learning algorithm, the learning rate, number of epochs and the batch size. Repeat the scores in a table.

2. Repeat parts (c) and (d) and select the model with the best performance.

3. Evaluate the selected model on the test set and report the testing loss and accuracy.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

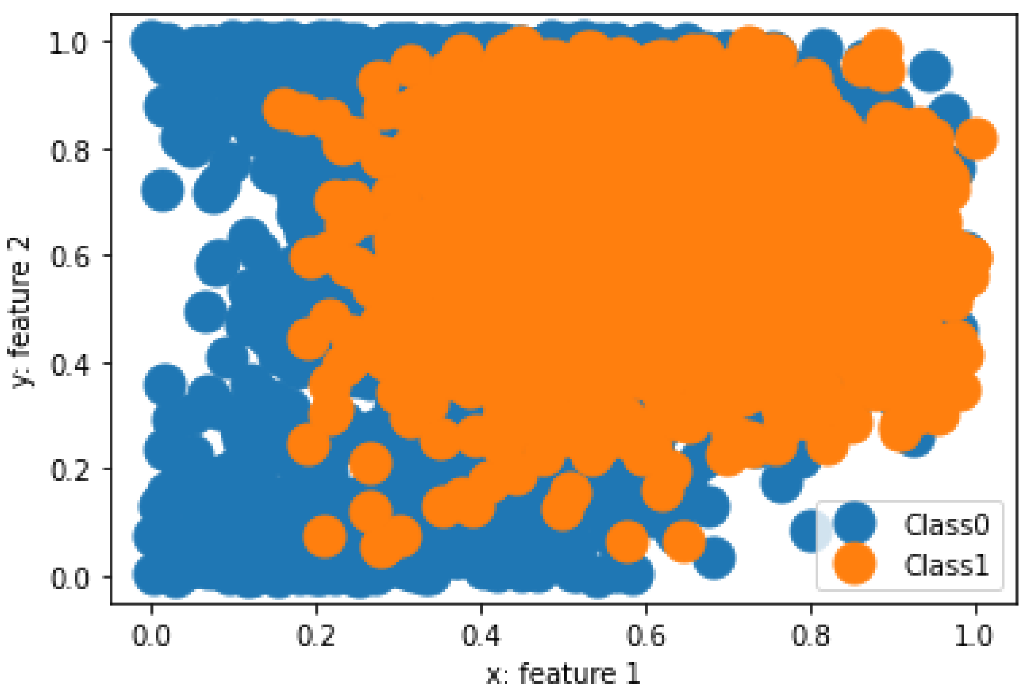
```
from keras.models import Sequential
import matplotlib.pyplot as plt
from keras.layers import Dense
from random import randint
import tensorflow as tf
import pandas as pd
import numpy as np

def feat_plot(features, labels, classes):
    for class_i in classes:
        plt.plot(features[labels[:] == classes[class_i], 0],
features[labels[:] == classes[class_i], 1], 'o',
```

```
                 markersize=15)
    plt.xlabel('x: feature 1')
    plt.ylabel('y: feature 2')
    plt.legend(['Class' + str(classes[class_i]) for class_i in
classes])
    plt.show()

def plot_curve(accuracy_train, loss_train):
    epochs = np.arange(loss_train.shape[0])
    plt.subplot(1, 2, 1)
    plt.plot(epochs, accuracy_train)
    plt.xlabel('Epoch#')
    plt.ylabel('Accuracy')
    plt.title('Training Accuracy')

    plt.subplot(1, 2, 2)
    plt.plot(epochs, loss_train)
    plt.xlabel('Epoch#')
    plt.ylabel('Binary crossentropy loss')
    plt.title('Training loss')
    plt.show()



data = pd.read_csv('/content/drive/MyDrive/spotify_preprocessed.csv')

spotify_features = np.array(data.drop(columns=['target']))
spotify_labels = np.array(data['target'])
spotify_classes = [0, 1]

# Shuffling the data
num_train_samples = spotify_features.shape[0]
spotify_train_ind = np.arange(0, num_train_samples)
spotify_train_ind_s = np.random.permutation(spotify_train_ind)

# Split 90% of the data for training
spotify_x_train =
spotify_features[spotify_train_ind_s[0:int(0.9*num_train_samples)], :]
spotify_y_train =
spotify_labels[spotify_train_ind_s[0:int(0.9*num_train_samples)]]
print(spotify_x_train.shape)

# The other half will be used for testing
spotify_x_test =
spotify_features[spotify_train_ind_s[int(0.9*num_train_samples):], :]
spotify_y_test =
spotify_labels[spotify_train_ind_s[int(0.9*num_train_samples):]]
print(spotify_x_test.shape)
```

```
# Update some variables that affect the training data
num_train_samples = spotify_x_train.shape[0]
spotify_train_ind = np.arange(0, num_train_samples)
spotify_train_ind_s = spotify_train_ind #do not randomly permute
again, you will misalign the test with the validation and training
data

# Of the training data, 80% of it will be used for the actual training
spotify_x_train_f =
spotify_x_train[spotify_train_ind_s[0:int(0.8*num_train_samples)], :]
spotify_y_train_f =
spotify_y_train[spotify_train_ind_s[0:int(0.8*num_train_samples)]]
print(spotify_x_train_f.shape)

# The other 20% will be used for validation
spotify_x_val =
spotify_x_train[spotify_train_ind_s[int(0.8*num_train_samples):]]
spotify_y_val =
spotify_y_train[spotify_train_ind_s[int(0.8*num_train_samples):]]
print(spotify_x_val.shape)

# Building the Neural Network
model_s = Sequential()
model_s.add(Dense(input_dim=15, units=100, activation='relu'))
model_s.add(Dense(units=100, activation='relu'))
model_s.add(Dense(units=100, activation='relu'))
model_s.add(Dense(units=100, activation='relu'))
model_s.add(Dense(units=100, activation='relu'))
model_s.add(Dense(units=50, activation='relu'))
model_s.add(Dense(units=1, activation='sigmoid'))

# Compile
opt = tf.keras.optimizers.SGD(learning_rate=0.01)
model_s.compile(loss='mean_squared_error', optimizer=opt,
metrics=['accuracy'])

feat_plot(spotify_x_train_f, spotify_y_train_f, spotify_classes)

# train
history = model_s.fit(spotify_x_train_f, spotify_y_train_f,
batch_size=16, epochs=50, verbose=1)

# # Evaluating the model on the training samples
score = model_s.evaluate(spotify_x_train_f, spotify_y_train_f)
print('total loss on training set:', score[0])
print('Accuracy of training set', score[1])

plt.figure(figsize=[9, 5])
acc_curve = np.array(history.history['accuracy'])
```

```
loss_curve = np.array(history.history['loss'])
plot_curve(acc_curve, loss_curve)

# Evaluating the model on the validation samples
score = model_s.evaluate(spotify_x_val, spotify_y_val)
print('total loss on validation set:', score[0])
print('Accuracy of validation set', score[1])

from sklearn.metrics import accuracy_score, confusion_matrix,
recall_score
#predicting the class of the held-out samples
test_class1_prob = model_s.predict(spotify_x_test)
test_lab = np.argmax(test_class1_prob,axis=1)
print('The accuracy using the testing set: ',
accuracy_score(test_lab,spotify_y_test))
conf_mat=confusion_matrix(test_lab,spotify_y_test)
print('The confusion matrix using testing set: \n', conf_mat)
```

```
(5758, 15)
(640, 15)
(4606, 15)
(1152, 15)
```



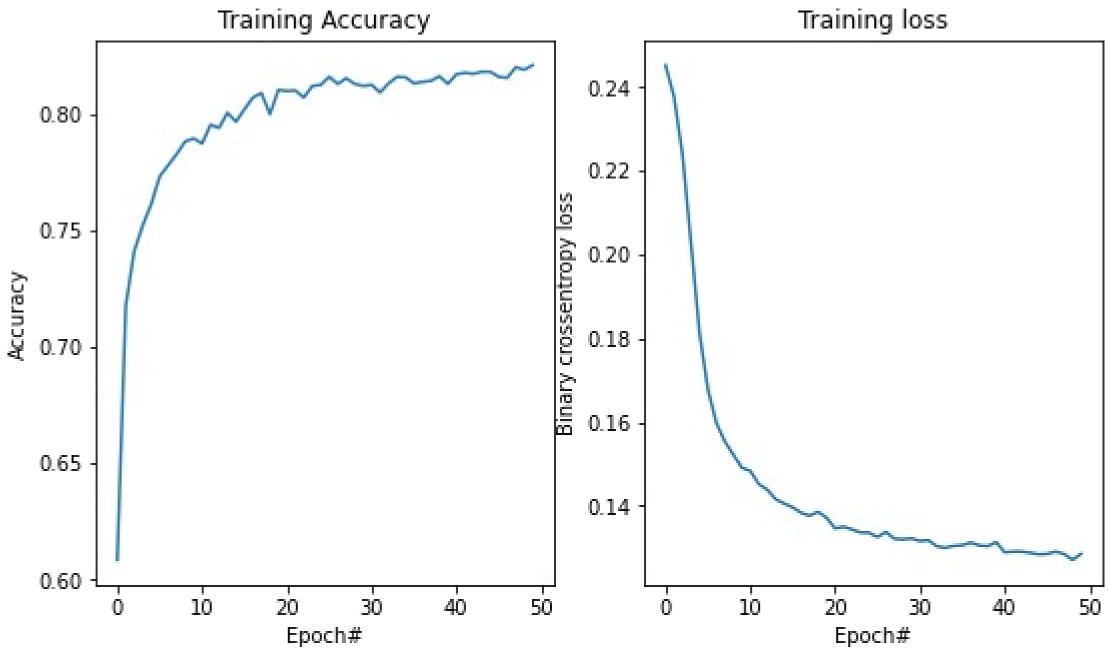

```
Epoch 1/50
288/288 [==============================] - 1s 2ms/step - loss: 0.2453
```

```
- accuracy: 0.6086
Epoch 2/50
288/288 [==============================] - 1s 2ms/step - loss: 0.2378
- accuracy: 0.7175
Epoch 3/50
288/288 [==============================] - 1s 2ms/step - loss: 0.2239
- accuracy: 0.7410
Epoch 4/50
288/288 [==============================] - 1s 2ms/step - loss: 0.2028
- accuracy: 0.7521
Epoch 5/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1814
- accuracy: 0.7612
Epoch 6/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1678
- accuracy: 0.7731
Epoch 7/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1598
- accuracy: 0.7779
Epoch 8/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1553
- accuracy: 0.7827
Epoch 9/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1522
- accuracy: 0.7881
Epoch 10/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1491
- accuracy: 0.7894
Epoch 11/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1483
- accuracy: 0.7872
Epoch 12/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1452
- accuracy: 0.7953
Epoch 13/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1438
- accuracy: 0.7940
Epoch 14/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1415
- accuracy: 0.8005
Epoch 15/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1405
- accuracy: 0.7966
Epoch 16/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1396
- accuracy: 0.8020
Epoch 17/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1383
- accuracy: 0.8070
Epoch 18/50
```

```
288/288 [==============================] - 1s 2ms/step - loss: 0.1376
- accuracy: 0.8089
Epoch 19/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1385
- accuracy: 0.7998
Epoch 20/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1371
- accuracy: 0.8102
Epoch 21/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1346
- accuracy: 0.8098
Epoch 22/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1349
- accuracy: 0.8100
Epoch 23/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1343
- accuracy: 0.8070
Epoch 24/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1336
- accuracy: 0.8120
Epoch 25/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1335
- accuracy: 0.8124
Epoch 26/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1325
- accuracy: 0.8159
Epoch 27/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1337
- accuracy: 0.8129
Epoch 28/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1321
- accuracy: 0.8152
Epoch 29/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1320
- accuracy: 0.8129
Epoch 30/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1321
- accuracy: 0.8120
Epoch 31/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1315
- accuracy: 0.8124
Epoch 32/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1317
- accuracy: 0.8092
Epoch 33/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1303
- accuracy: 0.8131
Epoch 34/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1299
- accuracy: 0.8159
```

```
Epoch 35/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1304
- accuracy: 0.8157
Epoch 36/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1305
- accuracy: 0.8131
Epoch 37/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1312
- accuracy: 0.8137
Epoch 38/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1305
- accuracy: 0.8142
Epoch 39/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1303
- accuracy: 0.8161
Epoch 40/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1313
- accuracy: 0.8129
Epoch 41/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1288
- accuracy: 0.8170
Epoch 42/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1290
- accuracy: 0.8176
Epoch 43/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1290
- accuracy: 0.8172
Epoch 44/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1287
- accuracy: 0.8181
Epoch 45/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1283
- accuracy: 0.8181
Epoch 46/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1284
- accuracy: 0.8159
Epoch 47/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1290
- accuracy: 0.8155
Epoch 48/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1284
- accuracy: 0.8200
Epoch 49/50
288/288 [==============================] - 1s 4ms/step - loss: 0.1270
- accuracy: 0.8189
Epoch 50/50
288/288 [==============================] - 1s 2ms/step - loss: 0.1284
- accuracy: 0.8209
144/144 [==============================] - 0s 1ms/step - loss: 0.1229
- accuracy: 0.8261
```

```
total loss on training set: 0.12293843924999237
Accuracy of training set 0.8260964155197144
```




```
36/36 [==============================] - 0s 2ms/step - loss: 0.1252 -
accuracy: 0.8325
total loss on validation set: 0.12518316507339478
Accuracy of validation set 0.8324652910232544
The accuracy using the testing set:  0.4953125
The confusion matrix using testing set:
 [[317 323]
 [  0   0]]
```