In [ ]:
```
import tensorflow
from keras.models import Sequential #For the model
from keras.layers import InputLayer,Dense #For hidden and output layers
from keras.optimizers import Adam #For optimizers
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
data=pd.read_csv("D:\\Workshops\\W 17 - AI Masterclass Workshop\\data\\Bank.CSV")
data.head()
```

Splitting data into traning & testing

In [ ]:
```
data.shape
```

In [ ]:
```
x=data.iloc[:,:7]
y=data.iloc[:,7]
```

In [ ]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

Defining the input neurons

In [ ]:
```
input_neurons=7
```

Defining the output neurones

In [ ]:
```
output_neurons=1
```

Defining hidden layers and neurons

In [ ]:
```
first_hid_neurons=5
second_hid_neurons=3
```

Defining the architecture of the model

In [ ]:
```
model=Sequential()
model.add(InputLayer(input_shape=(input_neurons,)))
model.add(Dense(units=first_hid_neurons, activation="relu"))
model.add(Dense(units=second_hid_neurons, activation="relu"))
model.add(Dense(units=output_neurons, activation="sigmoid"))
```

Model architecture summary

In [ ]:
```
model.summary()
```

Optimizers

In [ ]:
```
optim = Adam(lr=0.01)
```

Compiling the model

In [ ]:
```
model.compile(loss="binary_crossentropy",optimizer=optim)
```

Traning the model

In [ ]:
```
# model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=50)
model_det=model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=50)
```

Evaluating the model

In [ ]:
```
y_pred=model.predict_classes(x_test)
```

In [ ]:
```
accuracy_score(y_test,y_pred)
```

In [ ]:
```

```