LAB-1

Exercise 1:

Exercise 1: Build a Sequential model with two layers.

Layer 1 is dense with 32 nodes and 'relu' activation function. Input dimension is 100.

Layer 2 is dense with 10 nodes and 'softmax' activation function.

Compile the model with optimizer 'rmsprop', loss='categorical_crossentropy' and metrics=['accuracy']

Generate dummy data using np.random.random 1000, 100 and dummy labels of 10 categories with size 1000,1

Convert labels to categorical one-hot encoding using below statement

from keras.utils import to_categorical one_hot_labels = to_categorical(labels, num_classes=10)

Finally, Train the model, iterating on the data in batches of 32 samples

```
[8] from keras.utils import to_categorical
    model = Sequential()
    model.add(Dense(32, activation='relu', input_dim=100))
    model.add(Dense(10, activation='softmax'))
    model.compile(optimizer='rmsprop',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

    #dummy data
    import numpy as np
    data = np.random.random((1000, 100))
    labels = np.random.randint(10, size=(1000, 1))

    one_hot_labels = to_categorical(labels, num_classes=10)

    model.fit(data, one_hot_labels, epochs=10, batch_size=32)
```

Exercise 2:

Build a Sequential model of Neural Network with three dense layers.

1. Layer 1, 2 and 3 have 512, 256 and 10 nodes respectively. Activation function of first two layers is relu and final layer is softmax.
2. Compile model with optimizer='rmsprop', loss='categorical_crossentropy' and metrics=['accuracy']
3. Train the model with two additional parameters verbose=1 and validation_data=(X_test, y_test)

```
[22] model = models.Sequential()
     model.add(Dense(512, activation='relu', input_shape=(NUM_ROWS * NUM_COLS,)))
     model.add(Dense(256, activation='relu'))
     model.add(Dense(10, activation='softmax'))

     model.compile(optimizer='rmsprop',
                   loss='categorical_crossentropy',
                   metrics=['accuracy'])

     model.fit(X_train, y_train,
               batch_size=BATCH_SIZE,
               epochs=EPOCHS,
               verbose=1,
               validation_data=(X_test, y_test))

     score = model.evaluate(X_test, y_test, verbose=0)
     print('Test loss:', score[0])
     print('Test accuracy:', score[1])
```

OUTPUT: Test loss: 0.0926784, Test accuracy: 0.9821.

Exercise 3:

Build sequential model

1. Sequential model with four layers having 1024,512,512,10 nodes respectively
2. Input shape = 3072
3. Activation of first three layers is relu and final is softmax
4. Compile model using loss='categorical_crossentropy', optimizer='adam' and metrics=['accuracy']
5. Train model

```
[27] model = Sequential()
     model.add(Dense(1024, input_shape=(3072, )))
     model.add(Activation('relu'))
     model.add(Dense(512))
     model.add(Activation('relu'))
     model.add(Dense(512))
     model.add(Activation('relu'))
     model.add(Dense(10))
     model.add(Activation('softmax'))

     model.compile(loss='categorical_crossentropy',
                   optimizer='adam',
                   metrics=['accuracy'])


     history = model.fit(X_train, Y_train,
                         batch_size=128,
                         nb_epoch=10,
                         verbose=1,
                         validation_data=(X_test, Y_test))
```