

## Loading the required packages for the algorithm

In [ ]:

```
import warnings
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Activation, Dense, Flatten
from tensorflow.keras import optimizers
from sklearn import preprocessing
```

In [ ]:

```
from google.colab import drive
```

## Loading the required file

In [ ]:

```
drive.mount('/content/drive')
```

Mounted at /content/drive

In [ ]:

```
import h5py
hf5=h5py.File('/content/drive/My Drive/Colab Notebooks/SVHN_single_grey1.h5','r')
hf5.keys()
```

Out[ ]:

```
<KeysViewHDF5 ['X_test', 'X_train', 'X_val', 'y_test', 'y_train', 'y_val']>
```

## Read the data from the h5py file and understand the train/test splits

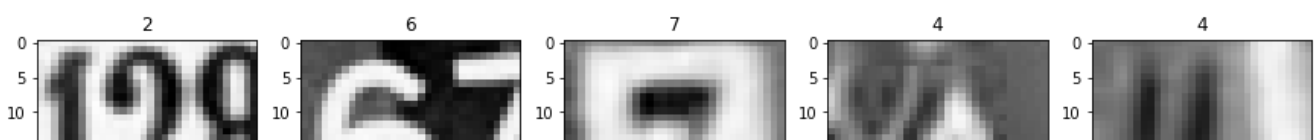
In [ ]:

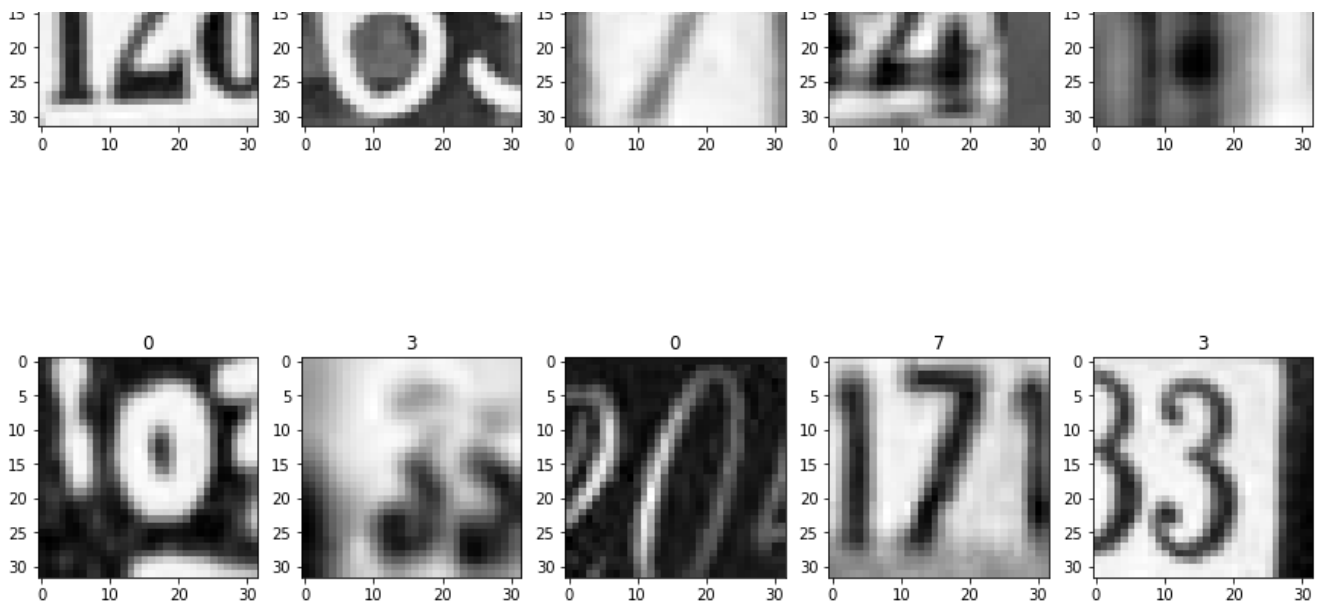
```
X_train=hf5['X_train'][:]
y_train=hf5['y_train'][:]
x_test=hf5['X_test'][:]
y_test=hf5['y_test'][:]
x_val=hf5['X_val'][:]
y_val=hf5['y_val'][:]
```

## Visualizing the dataset

In [ ]:

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.figure(figsize=(15,10))
for i in range(10):
    plt.subplot(2,5,i+1)
    plt.imshow(X_train[i], cmap="Greys")
    plt.title(y_train[i])
    #plt.tight_layout()
```





## Checking the shape of the training and test dataset

In [ ]:

```
X_train.shape
```

Out[ ]:

```
(42000, 32, 32)
```

In [ ]:

```
x_test.shape
```

Out[ ]:

```
(18000, 32, 32)
```

In [ ]:

```
x_val.shape
```

Out[ ]:

```
(60000, 32, 32)
```

In [ ]:

```
y_val.shape
```

Out[ ]:

```
(60000,)
```

In [ ]:

```
y_train.shape
```

Out[ ]:

```
(42000,)
```

In [ ]:

```
y_test.shape
```

```
Out[ ]:
```

```
(18000,)
```

```
In [ ]:
```

```
y_test.view()
```

```
Out[ ]:
```

```
array([1, 7, 2, ..., 7, 9, 2], dtype=uint8)
```

## Reshaping the training data and test data

```
In [ ]:
```

```
X_train = X_train.reshape((X_train.shape[0], -1))
x_test = x_test.reshape((x_test.shape[0], -1))
x_val=x_val.reshape((x_val.shape[0],-1))
```

## One hot encode the labels for train and test data

```
In [ ]:
```

```
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
y_val=to_categorical(y_val)
```

```
In [ ]:
```

```
print(X_train.shape, x_test.shape, y_train.shape, y_test.shape,x_val.shape,y_val.shape)
```

```
(42000, 1024) (18000, 1024) (42000, 10) (18000, 10) (60000, 1024) (60000, 10)
```

## Normalizing the test and train data

```
In [ ]:
```

```
X_train = preprocessing.normalize(X_train)
x_test = preprocessing.normalize(x_test)
x_val=preprocessing.normalize(x_val)
```

**\*Define the model architecture using TensorFlow with a flatten layer followed by dense layers with activation as ReLu and softmax**

**\*Compile the model with loss as categorical cross-entropy and adam optimizers. Use accuracy as the metric for evaluation**

**(Basic NN model)**

```
In [ ]:
```

```
def mlp_model():
    model = Sequential()
    model.add(Flatten())
    model.add(Dense(50, input_shape = (1024, )))
    model.add(Activation('relu'))
    model.add(Dense(50))
    model.add(Activation('relu'))
    model.add(Dense(50))
```

```

model.add(Activation('relu'))
model.add(Dense(50))
model.add(Activation('relu'))
model.add(Dense(10))
model.add(Activation('softmax'))

sgd = optimizers.Adam(lr = 0.0009)
model.compile(optimizer = sgd, loss = 'categorical_crossentropy', metrics = ['accuracy'])

return model

```

In [ ]:

```
model = mlp_model()
```

## Fit and evaluate the model. Print the loss and accuracy for the test data

In [ ]:

```
history=model.fit(X_train, y_train,validation_data=(x_val,y_val),batch_size = 200, epochs = 100, verbose = 1)
```

```

Epoch 1/100
210/210 [=====] - 1s 7ms/step - loss: 2.2775 - accuracy: 0.1133 - val_loss: 2.1825 - val_accuracy: 0.1521
Epoch 2/100
210/210 [=====] - 1s 6ms/step - loss: 1.9992 - accuracy: 0.2554 - val_loss: 1.8522 - val_accuracy: 0.3261
Epoch 3/100
210/210 [=====] - 1s 7ms/step - loss: 1.7586 - accuracy: 0.3587 - val_loss: 1.6762 - val_accuracy: 0.3968
Epoch 4/100
210/210 [=====] - 2s 7ms/step - loss: 1.6427 - accuracy: 0.4139 - val_loss: 1.5959 - val_accuracy: 0.4315
Epoch 5/100
210/210 [=====] - 1s 7ms/step - loss: 1.5814 - accuracy: 0.4420 - val_loss: 1.5536 - val_accuracy: 0.4511
Epoch 6/100
210/210 [=====] - 1s 7ms/step - loss: 1.5341 - accuracy: 0.4660 - val_loss: 1.5203 - val_accuracy: 0.4719
Epoch 7/100
210/210 [=====] - 1s 6ms/step - loss: 1.4899 - accuracy: 0.4835 - val_loss: 1.4376 - val_accuracy: 0.5117
Epoch 8/100
210/210 [=====] - 2s 7ms/step - loss: 1.4316 - accuracy: 0.5101 - val_loss: 1.3927 - val_accuracy: 0.5267
Epoch 9/100
210/210 [=====] - 2s 7ms/step - loss: 1.3857 - accuracy: 0.5313 - val_loss: 1.3607 - val_accuracy: 0.5431
Epoch 10/100
210/210 [=====] - 1s 7ms/step - loss: 1.3571 - accuracy: 0.5395 - val_loss: 1.3317 - val_accuracy: 0.5522
Epoch 11/100
210/210 [=====] - 1s 6ms/step - loss: 1.3241 - accuracy: 0.5512 - val_loss: 1.3096 - val_accuracy: 0.5552
Epoch 12/100
210/210 [=====] - 1s 7ms/step - loss: 1.3005 - accuracy: 0.5599 - val_loss: 1.3031 - val_accuracy: 0.5609
Epoch 13/100
210/210 [=====] - 1s 7ms/step - loss: 1.2791 - accuracy: 0.5684 - val_loss: 1.3132 - val_accuracy: 0.5505
Epoch 14/100
210/210 [=====] - 1s 7ms/step - loss: 1.2588 - accuracy: 0.5767 - val_loss: 1.2434 - val_accuracy: 0.5861
Epoch 15/100
210/210 [=====] - 1s 7ms/step - loss: 1.2436 - accuracy: 0.5834 - val_loss: 1.2290 - val_accuracy: 0.5894
Epoch 16/100
210/210 [=====] - 1s 6ms/step - loss: 1.2246 - accuracy: 0.5915 - val_loss: 1.2086 - val_accuracy: 0.5985
Epoch 17/100
210/210 [=====] - 1s 6ms/step - loss: 1.2074 - accuracy: 0.6009 - val_loss: 1.1885 - val_accuracy: 0.6123

```

```
s. 1.1885 - val_accuracy: 0.6125
Epoch 18/100
210/210 [=====] - 1s 7ms/step - loss: 1.1945 - accuracy: 0.6051 - val_loss: 1.1815 - val_accuracy: 0.6138
Epoch 19/100
210/210 [=====] - 1s 7ms/step - loss: 1.1765 - accuracy: 0.6125 - val_loss: 1.1480 - val_accuracy: 0.6252
Epoch 20/100
210/210 [=====] - 1s 7ms/step - loss: 1.1653 - accuracy: 0.6191 - val_loss: 1.1385 - val_accuracy: 0.6301
Epoch 21/100
210/210 [=====] - 1s 7ms/step - loss: 1.1523 - accuracy: 0.6253 - val_loss: 1.1826 - val_accuracy: 0.6088
Epoch 22/100
210/210 [=====] - 1s 7ms/step - loss: 1.1374 - accuracy: 0.6302 - val_loss: 1.1143 - val_accuracy: 0.6424
Epoch 23/100
210/210 [=====] - 2s 7ms/step - loss: 1.1125 - accuracy: 0.6402 - val_loss: 1.0977 - val_accuracy: 0.6462
Epoch 24/100
210/210 [=====] - 2s 7ms/step - loss: 1.1057 - accuracy: 0.6451 - val_loss: 1.1196 - val_accuracy: 0.6381
Epoch 25/100
210/210 [=====] - 1s 7ms/step - loss: 1.0886 - accuracy: 0.6520 - val_loss: 1.0808 - val_accuracy: 0.6532
Epoch 26/100
210/210 [=====] - 2s 7ms/step - loss: 1.0718 - accuracy: 0.6600 - val_loss: 1.0495 - val_accuracy: 0.6693
Epoch 27/100
210/210 [=====] - 1s 7ms/step - loss: 1.0580 - accuracy: 0.6642 - val_loss: 1.0560 - val_accuracy: 0.6680
Epoch 28/100
210/210 [=====] - 2s 7ms/step - loss: 1.0442 - accuracy: 0.6698 - val_loss: 1.0495 - val_accuracy: 0.6678
Epoch 29/100
210/210 [=====] - 2s 7ms/step - loss: 1.0378 - accuracy: 0.6705 - val_loss: 1.0354 - val_accuracy: 0.6710
Epoch 30/100
210/210 [=====] - 1s 7ms/step - loss: 1.0221 - accuracy: 0.6792 - val_loss: 1.0160 - val_accuracy: 0.6783
Epoch 31/100
210/210 [=====] - 2s 7ms/step - loss: 1.0140 - accuracy: 0.6803 - val_loss: 1.0398 - val_accuracy: 0.6770
Epoch 32/100
210/210 [=====] - 2s 7ms/step - loss: 1.0038 - accuracy: 0.6836 - val_loss: 0.9905 - val_accuracy: 0.6903
Epoch 33/100
210/210 [=====] - 1s 7ms/step - loss: 0.9961 - accuracy: 0.6874 - val_loss: 1.0025 - val_accuracy: 0.6854
Epoch 34/100
210/210 [=====] - 1s 7ms/step - loss: 0.9823 - accuracy: 0.6927 - val_loss: 0.9804 - val_accuracy: 0.6940
Epoch 35/100
210/210 [=====] - 1s 7ms/step - loss: 0.9756 - accuracy: 0.6937 - val_loss: 0.9739 - val_accuracy: 0.6954
Epoch 36/100
210/210 [=====] - 1s 7ms/step - loss: 0.9680 - accuracy: 0.6961 - val_loss: 0.9728 - val_accuracy: 0.6989
Epoch 37/100
210/210 [=====] - 2s 7ms/step - loss: 0.9580 - accuracy: 0.6997 - val_loss: 0.9639 - val_accuracy: 0.6994
Epoch 38/100
210/210 [=====] - 1s 7ms/step - loss: 0.9479 - accuracy: 0.7033 - val_loss: 0.9916 - val_accuracy: 0.6873
Epoch 39/100
210/210 [=====] - 1s 7ms/step - loss: 0.9418 - accuracy: 0.7052 - val_loss: 0.9613 - val_accuracy: 0.7004
Epoch 40/100
210/210 [=====] - 1s 7ms/step - loss: 0.9329 - accuracy: 0.7080 - val_loss: 0.9227 - val_accuracy: 0.7133
Epoch 41/100
210/210 [=====] - 2s 7ms/step - loss: 0.9236 - accuracy: 0.7112 - val_loss: 0.9362 - val_accuracy: 0.7061
Epoch 42/100
210/210 [=====] - 1s 7ms/step - loss: 0.9232 - accuracy: 0.7105 - val_loss: 0.9444 - val_accuracy: 0.7013
Epoch 43/100
210/210 [=====] - 1s 7ms/step - loss: 0.9142 - accuracy: 0.7122 - val_loss:
```

```
210/210 [=====] - 1s 7ms/step - loss: 0.9142 - accuracy: 0.7123 - val_loss: 0.9165 - val_accuracy: 0.7141
Epoch 44/100
210/210 [=====] - 1s 7ms/step - loss: 0.9056 - accuracy: 0.7170 - val_loss: 0.9156 - val_accuracy: 0.7124
Epoch 45/100
210/210 [=====] - 2s 7ms/step - loss: 0.8981 - accuracy: 0.7186 - val_loss: 0.9047 - val_accuracy: 0.7167
Epoch 46/100
210/210 [=====] - 2s 8ms/step - loss: 0.8867 - accuracy: 0.7239 - val_loss: 0.8777 - val_accuracy: 0.7292
Epoch 47/100
210/210 [=====] - 1s 7ms/step - loss: 0.8872 - accuracy: 0.7226 - val_loss: 0.8874 - val_accuracy: 0.7227
Epoch 48/100
210/210 [=====] - 2s 8ms/step - loss: 0.8790 - accuracy: 0.7254 - val_loss: 0.8886 - val_accuracy: 0.7222
Epoch 49/100
210/210 [=====] - 2s 8ms/step - loss: 0.8669 - accuracy: 0.7303 - val_loss: 0.8731 - val_accuracy: 0.7287
Epoch 50/100
210/210 [=====] - 2s 8ms/step - loss: 0.8641 - accuracy: 0.7305 - val_loss: 0.8645 - val_accuracy: 0.7310
Epoch 51/100
210/210 [=====] - 2s 8ms/step - loss: 0.8609 - accuracy: 0.7308 - val_loss: 0.8619 - val_accuracy: 0.7315
Epoch 52/100
210/210 [=====] - 2s 7ms/step - loss: 0.8535 - accuracy: 0.7346 - val_loss: 0.8856 - val_accuracy: 0.7244
Epoch 53/100
210/210 [=====] - 2s 8ms/step - loss: 0.8522 - accuracy: 0.7348 - val_loss: 0.8791 - val_accuracy: 0.7257
Epoch 54/100
210/210 [=====] - 1s 7ms/step - loss: 0.8401 - accuracy: 0.7381 - val_loss: 0.8666 - val_accuracy: 0.7279
Epoch 55/100
210/210 [=====] - 1s 7ms/step - loss: 0.8364 - accuracy: 0.7375 - val_loss: 0.8380 - val_accuracy: 0.7396
Epoch 56/100
210/210 [=====] - 2s 8ms/step - loss: 0.8345 - accuracy: 0.7382 - val_loss: 0.8448 - val_accuracy: 0.7373
Epoch 57/100
210/210 [=====] - 1s 7ms/step - loss: 0.8318 - accuracy: 0.7400 - val_loss: 0.8539 - val_accuracy: 0.7353
Epoch 58/100
210/210 [=====] - 1s 7ms/step - loss: 0.8286 - accuracy: 0.7408 - val_loss: 0.8489 - val_accuracy: 0.7358
Epoch 59/100
210/210 [=====] - 2s 7ms/step - loss: 0.8254 - accuracy: 0.7418 - val_loss: 0.8500 - val_accuracy: 0.7341
Epoch 60/100
210/210 [=====] - 2s 7ms/step - loss: 0.8145 - accuracy: 0.7457 - val_loss: 0.8394 - val_accuracy: 0.7372
Epoch 61/100
210/210 [=====] - 2s 8ms/step - loss: 0.8126 - accuracy: 0.7458 - val_loss: 0.8318 - val_accuracy: 0.7410
Epoch 62/100
210/210 [=====] - 2s 8ms/step - loss: 0.8067 - accuracy: 0.7505 - val_loss: 0.8169 - val_accuracy: 0.7465
Epoch 63/100
210/210 [=====] - 2s 7ms/step - loss: 0.8046 - accuracy: 0.7487 - val_loss: 0.8308 - val_accuracy: 0.7422
Epoch 64/100
210/210 [=====] - 2s 7ms/step - loss: 0.8037 - accuracy: 0.7485 - val_loss: 0.8233 - val_accuracy: 0.7433
Epoch 65/100
210/210 [=====] - 2s 8ms/step - loss: 0.7988 - accuracy: 0.7512 - val_loss: 0.7943 - val_accuracy: 0.7558
Epoch 66/100
210/210 [=====] - 1s 7ms/step - loss: 0.7892 - accuracy: 0.7538 - val_loss: 0.8055 - val_accuracy: 0.7491
Epoch 67/100
210/210 [=====] - 1s 6ms/step - loss: 0.7966 - accuracy: 0.7495 - val_loss: 0.8097 - val_accuracy: 0.7494
Epoch 68/100
210/210 [=====] - 1s 7ms/step - loss: 0.7868 - accuracy: 0.7548 - val_loss: 0.8187 - val_accuracy: 0.7446
Epoch 69/100
```

```
Epoch 69/100
210/210 [=====] - 2s 7ms/step - loss: 0.7869 - accuracy: 0.7541 - val_loss: 0.8204 - val_accuracy: 0.7444
Epoch 70/100
210/210 [=====] - 1s 7ms/step - loss: 0.7807 - accuracy: 0.7557 - val_loss: 0.8154 - val_accuracy: 0.7469
Epoch 71/100
210/210 [=====] - 2s 7ms/step - loss: 0.7779 - accuracy: 0.7570 - val_loss: 0.7900 - val_accuracy: 0.7543
Epoch 72/100
210/210 [=====] - 2s 7ms/step - loss: 0.7675 - accuracy: 0.7617 - val_loss: 0.7701 - val_accuracy: 0.7624
Epoch 73/100
210/210 [=====] - 2s 7ms/step - loss: 0.7723 - accuracy: 0.7592 - val_loss: 0.7903 - val_accuracy: 0.7552
Epoch 74/100
210/210 [=====] - 2s 7ms/step - loss: 0.7739 - accuracy: 0.7578 - val_loss: 0.7847 - val_accuracy: 0.7552
Epoch 75/100
210/210 [=====] - 1s 7ms/step - loss: 0.7665 - accuracy: 0.7616 - val_loss: 0.7639 - val_accuracy: 0.7636
Epoch 76/100
210/210 [=====] - 2s 7ms/step - loss: 0.7647 - accuracy: 0.7612 - val_loss: 0.8071 - val_accuracy: 0.7478
Epoch 77/100
210/210 [=====] - 2s 7ms/step - loss: 0.7617 - accuracy: 0.7611 - val_loss: 0.8022 - val_accuracy: 0.7487
Epoch 78/100
210/210 [=====] - 2s 8ms/step - loss: 0.7549 - accuracy: 0.7648 - val_loss: 0.7615 - val_accuracy: 0.7642
Epoch 79/100
210/210 [=====] - 2s 8ms/step - loss: 0.7542 - accuracy: 0.7636 - val_loss: 0.7819 - val_accuracy: 0.7562
Epoch 80/100
210/210 [=====] - 2s 8ms/step - loss: 0.7551 - accuracy: 0.7642 - val_loss: 0.7704 - val_accuracy: 0.7586
Epoch 81/100
210/210 [=====] - 2s 8ms/step - loss: 0.7498 - accuracy: 0.7651 - val_loss: 0.7746 - val_accuracy: 0.7600
Epoch 82/100
210/210 [=====] - 2s 8ms/step - loss: 0.7518 - accuracy: 0.7640 - val_loss: 0.7574 - val_accuracy: 0.7635
Epoch 83/100
210/210 [=====] - 1s 7ms/step - loss: 0.7402 - accuracy: 0.7690 - val_loss: 0.7695 - val_accuracy: 0.7588
Epoch 84/100
210/210 [=====] - 1s 7ms/step - loss: 0.7420 - accuracy: 0.7676 - val_loss: 0.7559 - val_accuracy: 0.7663
Epoch 85/100
210/210 [=====] - 1s 7ms/step - loss: 0.7373 - accuracy: 0.7678 - val_loss: 0.7506 - val_accuracy: 0.7676
Epoch 86/100
210/210 [=====] - 1s 7ms/step - loss: 0.7346 - accuracy: 0.7696 - val_loss: 0.7428 - val_accuracy: 0.7700
Epoch 87/100
210/210 [=====] - 2s 8ms/step - loss: 0.7379 - accuracy: 0.7688 - val_loss: 0.7457 - val_accuracy: 0.7680
Epoch 88/100
210/210 [=====] - 2s 8ms/step - loss: 0.7406 - accuracy: 0.7669 - val_loss: 0.7464 - val_accuracy: 0.7670
Epoch 89/100
210/210 [=====] - 1s 7ms/step - loss: 0.7336 - accuracy: 0.7700 - val_loss: 0.7435 - val_accuracy: 0.7693
Epoch 90/100
210/210 [=====] - 2s 7ms/step - loss: 0.7286 - accuracy: 0.7714 - val_loss: 0.7464 - val_accuracy: 0.7667
Epoch 91/100
210/210 [=====] - 2s 8ms/step - loss: 0.7240 - accuracy: 0.7739 - val_loss: 0.7560 - val_accuracy: 0.7651
Epoch 92/100
210/210 [=====] - 2s 8ms/step - loss: 0.7239 - accuracy: 0.7742 - val_loss: 0.7227 - val_accuracy: 0.7767
Epoch 93/100
210/210 [=====] - 1s 7ms/step - loss: 0.7218 - accuracy: 0.7738 - val_loss: 0.7561 - val_accuracy: 0.7618
Epoch 94/100
210/210 [=====] - 2s 7ms/step - loss: 0.7178 - accuracy: 0.7745 - val_loss: 0.7560 - val_accuracy: 0.7760
```

```
s: 0.7200 - val_accuracy: 0.7783
Epoch 95/100
210/210 [=====] - 2s 7ms/step - loss: 0.7175 - accuracy: 0.7738 - val_loss: 0.7448 - val_accuracy: 0.7665
Epoch 96/100
210/210 [=====] - 2s 7ms/step - loss: 0.7153 - accuracy: 0.7774 - val_loss: 0.7277 - val_accuracy: 0.7740
Epoch 97/100
210/210 [=====] - 2s 8ms/step - loss: 0.7167 - accuracy: 0.7748 - val_loss: 0.7329 - val_accuracy: 0.7711
Epoch 98/100
210/210 [=====] - 2s 8ms/step - loss: 0.7168 - accuracy: 0.7752 - val_loss: 0.7603 - val_accuracy: 0.7604
Epoch 99/100
210/210 [=====] - 2s 7ms/step - loss: 0.7164 - accuracy: 0.7769 - val_loss: 0.7113 - val_accuracy: 0.7803
Epoch 100/100
210/210 [=====] - 2s 8ms/step - loss: 0.7040 - accuracy: 0.7792 - val_loss: 0.7259 - val_accuracy: 0.7755
```

In [ ]:

```
results = model.evaluate(x_test, y_test)
```

```
563/563 [=====] - 1s 2ms/step - loss: 0.7890 - accuracy: 0.7597
```

In [ ]:

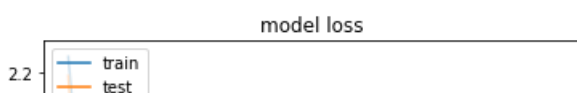
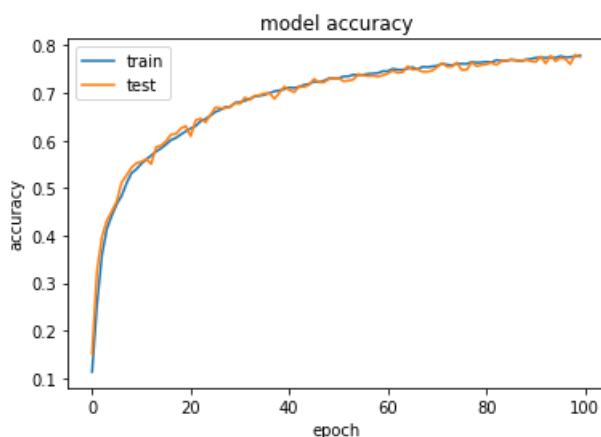
```
print('Test accuracy: ', results[1])
```

```
Test accuracy: 0.7596666812896729
```

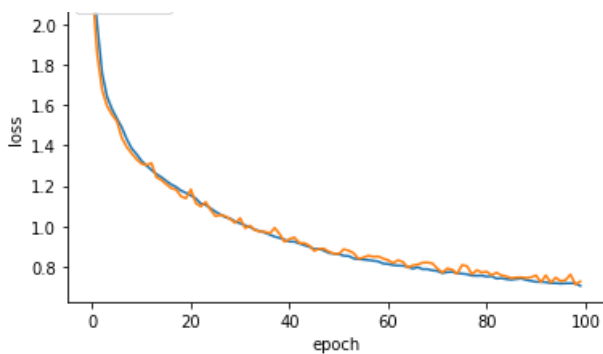
In [ ]:

```
#summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```







The accuracy was found to be 75 in basic Neural network model.\ Required graphs were also plotted.

In [ ]:

```
from tensorflow.keras.layers import BatchNormalization, Dropout
```

## Model building using BatchNormalization and Dropout method

In [ ]:

```
def mlp_model():
    model = Sequential()
    model.add(Flatten())
    model.add(Dense(50, input_shape = (1024, ), kernel_initializer='he_normal'))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(Dropout(0.2))
    model.add(Dense(50, kernel_initializer='he_normal'))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(Dropout(0.2))
    model.add(Dense(50, kernel_initializer='he_normal'))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(Dropout(0.2))
    model.add(Dense(50, kernel_initializer='he_normal'))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(Dropout(0.2))
    model.add(Dense(10, kernel_initializer='he_normal'))
    model.add(Activation('softmax'))

    adam = optimizers.Adam(lr = 0.0009)
    model.compile(optimizer = adam, loss = 'categorical_crossentropy', metrics = ['accuracy'])

    return model
```

In [ ]:

```
model = mlp_model()
history1=model.fit(X_train, y_train,validation_data=(x_val,y_val), batch_size = 200, epochs = 100,
verbose = 1)
```

```
Epoch 1/100
210/210 [=====] - 2s 10ms/step - loss: 2.2202 - accuracy: 0.2045 - val_loss: 2.2366 - val_accuracy: 0.2051
Epoch 2/100
210/210 [=====] - 2s 10ms/step - loss: 1.6857 - accuracy: 0.4016 - val_loss: 1.8684 - val_accuracy: 0.3447
Epoch 3/100
210/210 [=====] - 2s 10ms/step - loss: 1.4532 - accuracy: 0.5030 - val_loss: 1.5179 - val_accuracy: 0.4836
Epoch 4/100
210/210 [=====] - 2s 9ms/step - loss: 1.3251 - accuracy: 0.5572 - val_loss: 1.1585 - val_accuracy: 0.6262
```

```
s: 1.1500 - val_accuracy: 0.6202
Epoch 5/100
210/210 [=====] - 2s 9ms/step - loss: 1.2430 - accuracy: 0.5898 - val_loss: 1.1722 - val_accuracy: 0.6249
Epoch 6/100
210/210 [=====] - 2s 9ms/step - loss: 1.1724 - accuracy: 0.6203 - val_loss: 1.2061 - val_accuracy: 0.6024
Epoch 7/100
210/210 [=====] - 2s 8ms/step - loss: 1.1253 - accuracy: 0.6374 - val_loss: 1.2435 - val_accuracy: 0.5789
Epoch 8/100
210/210 [=====] - 2s 9ms/step - loss: 1.0894 - accuracy: 0.6534 - val_loss: 0.9637 - val_accuracy: 0.6858
Epoch 9/100
210/210 [=====] - 2s 9ms/step - loss: 1.0478 - accuracy: 0.6671 - val_loss: 1.2672 - val_accuracy: 0.5871
Epoch 10/100
210/210 [=====] - 2s 9ms/step - loss: 1.0380 - accuracy: 0.6707 - val_loss: 1.0187 - val_accuracy: 0.6733
Epoch 11/100
210/210 [=====] - 2s 9ms/step - loss: 1.0132 - accuracy: 0.6809 - val_loss: 0.9724 - val_accuracy: 0.6780
Epoch 12/100
210/210 [=====] - 2s 9ms/step - loss: 0.9884 - accuracy: 0.6879 - val_loss: 1.0730 - val_accuracy: 0.6500
Epoch 13/100
210/210 [=====] - 2s 8ms/step - loss: 0.9761 - accuracy: 0.6944 - val_loss: 0.8690 - val_accuracy: 0.7224
Epoch 14/100
210/210 [=====] - 2s 9ms/step - loss: 0.9601 - accuracy: 0.6992 - val_loss: 0.8424 - val_accuracy: 0.7343
Epoch 15/100
210/210 [=====] - 2s 9ms/step - loss: 0.9468 - accuracy: 0.7054 - val_loss: 0.8267 - val_accuracy: 0.7365
Epoch 16/100
210/210 [=====] - 2s 10ms/step - loss: 0.9323 - accuracy: 0.7079 - val_loss: 0.9096 - val_accuracy: 0.7031
Epoch 17/100
210/210 [=====] - 2s 10ms/step - loss: 0.9257 - accuracy: 0.7129 - val_loss: 0.9133 - val_accuracy: 0.7122
Epoch 18/100
210/210 [=====] - 2s 9ms/step - loss: 0.9242 - accuracy: 0.7114 - val_loss: 0.8456 - val_accuracy: 0.7295
Epoch 19/100
210/210 [=====] - 2s 9ms/step - loss: 0.9099 - accuracy: 0.7160 - val_loss: 0.8256 - val_accuracy: 0.7408
Epoch 20/100
210/210 [=====] - 2s 9ms/step - loss: 0.9008 - accuracy: 0.7233 - val_loss: 0.6973 - val_accuracy: 0.7843
Epoch 21/100
210/210 [=====] - 2s 9ms/step - loss: 0.8903 - accuracy: 0.7235 - val_loss: 0.8191 - val_accuracy: 0.7390
Epoch 22/100
210/210 [=====] - 2s 8ms/step - loss: 0.8906 - accuracy: 0.7239 - val_loss: 0.8384 - val_accuracy: 0.7328
Epoch 23/100
210/210 [=====] - 2s 9ms/step - loss: 0.8806 - accuracy: 0.7271 - val_loss: 0.9991 - val_accuracy: 0.6803
Epoch 24/100
210/210 [=====] - 2s 9ms/step - loss: 0.8749 - accuracy: 0.7281 - val_loss: 0.8372 - val_accuracy: 0.7335
Epoch 25/100
210/210 [=====] - 2s 8ms/step - loss: 0.8788 - accuracy: 0.7273 - val_loss: 0.7257 - val_accuracy: 0.7736
Epoch 26/100
210/210 [=====] - 2s 9ms/step - loss: 0.8655 - accuracy: 0.7338 - val_loss: 0.9238 - val_accuracy: 0.7046
Epoch 27/100
210/210 [=====] - 2s 10ms/step - loss: 0.8617 - accuracy: 0.7341 - val_loss: 0.8021 - val_accuracy: 0.7428
Epoch 28/100
210/210 [=====] - 2s 9ms/step - loss: 0.8571 - accuracy: 0.7337 - val_loss: 0.6598 - val_accuracy: 0.7943
Epoch 29/100
210/210 [=====] - 2s 10ms/step - loss: 0.8553 - accuracy: 0.7359 - val_loss: 0.7326 - val_accuracy: 0.7670
Epoch 30/100
210/210 [=====] - 2s 9ms/step - loss: 0.8488 - accuracy: 0.7371 - val_loss: 0.8488 - val_accuracy: 0.7371
```

```
210/210 [=====] - 2s 9ms/step - loss: 0.8499 - accuracy: 0.7371 - val_loss: 0.8405 - val_accuracy: 0.7350
Epoch 31/100
210/210 [=====] - 2s 10ms/step - loss: 0.8464 - accuracy: 0.7394 - val_loss: 0.6712 - val_accuracy: 0.7910
Epoch 32/100
210/210 [=====] - 2s 9ms/step - loss: 0.8391 - accuracy: 0.7395 - val_loss: 0.6652 - val_accuracy: 0.7940
Epoch 33/100
210/210 [=====] - 2s 10ms/step - loss: 0.8394 - accuracy: 0.7415 - val_loss: 0.6473 - val_accuracy: 0.8010
Epoch 34/100
210/210 [=====] - 2s 9ms/step - loss: 0.8402 - accuracy: 0.7412 - val_loss: 0.6731 - val_accuracy: 0.7948
Epoch 35/100
210/210 [=====] - 2s 9ms/step - loss: 0.8328 - accuracy: 0.7429 - val_loss: 0.7487 - val_accuracy: 0.7612
Epoch 36/100
210/210 [=====] - 2s 9ms/step - loss: 0.8234 - accuracy: 0.7464 - val_loss: 0.7882 - val_accuracy: 0.7471
Epoch 37/100
210/210 [=====] - 2s 9ms/step - loss: 0.8325 - accuracy: 0.7434 - val_loss: 0.7478 - val_accuracy: 0.7639
Epoch 38/100
210/210 [=====] - 2s 9ms/step - loss: 0.8319 - accuracy: 0.7420 - val_loss: 0.6794 - val_accuracy: 0.7888
Epoch 39/100
210/210 [=====] - 2s 9ms/step - loss: 0.8208 - accuracy: 0.7477 - val_loss: 0.7100 - val_accuracy: 0.7789
Epoch 40/100
210/210 [=====] - 2s 9ms/step - loss: 0.8148 - accuracy: 0.7485 - val_loss: 0.6831 - val_accuracy: 0.7870
Epoch 41/100
210/210 [=====] - 2s 9ms/step - loss: 0.8176 - accuracy: 0.7488 - val_loss: 0.7229 - val_accuracy: 0.7747
Epoch 42/100
210/210 [=====] - 2s 9ms/step - loss: 0.8233 - accuracy: 0.7455 - val_loss: 0.9003 - val_accuracy: 0.7163
Epoch 43/100
210/210 [=====] - 2s 10ms/step - loss: 0.8089 - accuracy: 0.7484 - val_loss: 0.8722 - val_accuracy: 0.7138
Epoch 44/100
210/210 [=====] - 2s 10ms/step - loss: 0.8152 - accuracy: 0.7464 - val_loss: 0.7009 - val_accuracy: 0.7773
Epoch 45/100
210/210 [=====] - 2s 10ms/step - loss: 0.8036 - accuracy: 0.7513 - val_loss: 0.7389 - val_accuracy: 0.7633
Epoch 46/100
210/210 [=====] - 2s 10ms/step - loss: 0.8122 - accuracy: 0.7496 - val_loss: 0.6740 - val_accuracy: 0.7877
Epoch 47/100
210/210 [=====] - 2s 10ms/step - loss: 0.8091 - accuracy: 0.7490 - val_loss: 0.6776 - val_accuracy: 0.7891
Epoch 48/100
210/210 [=====] - 2s 9ms/step - loss: 0.7964 - accuracy: 0.7537 - val_loss: 0.6010 - val_accuracy: 0.8167
Epoch 49/100
210/210 [=====] - 2s 9ms/step - loss: 0.8010 - accuracy: 0.7525 - val_loss: 0.6819 - val_accuracy: 0.7880
Epoch 50/100
210/210 [=====] - 2s 9ms/step - loss: 0.8000 - accuracy: 0.7518 - val_loss: 0.7594 - val_accuracy: 0.7581
Epoch 51/100
210/210 [=====] - 2s 9ms/step - loss: 0.7919 - accuracy: 0.7558 - val_loss: 0.7306 - val_accuracy: 0.7711
Epoch 52/100
210/210 [=====] - 2s 9ms/step - loss: 0.7922 - accuracy: 0.7562 - val_loss: 0.6618 - val_accuracy: 0.7934
Epoch 53/100
210/210 [=====] - 2s 9ms/step - loss: 0.7960 - accuracy: 0.7538 - val_loss: 0.7599 - val_accuracy: 0.7586
Epoch 54/100
210/210 [=====] - 2s 9ms/step - loss: 0.7982 - accuracy: 0.7531 - val_loss: 0.6953 - val_accuracy: 0.7808
Epoch 55/100
210/210 [=====] - 2s 9ms/step - loss: 0.7927 - accuracy: 0.7558 - val_loss: 0.6359 - val_accuracy: 0.8009
Epoch 56/100
```

```
Epoch 56/100
210/210 [=====] - 2s 9ms/step - loss: 0.7944 - accuracy: 0.7546 - val_loss: 0.6651 - val_accuracy: 0.7854
Epoch 57/100
210/210 [=====] - 2s 9ms/step - loss: 0.7857 - accuracy: 0.7594 - val_loss: 0.6960 - val_accuracy: 0.7811
Epoch 58/100
210/210 [=====] - 2s 10ms/step - loss: 0.7843 - accuracy: 0.7575 - val_loss: 0.7106 - val_accuracy: 0.7720
Epoch 59/100
210/210 [=====] - 2s 10ms/step - loss: 0.7851 - accuracy: 0.7577 - val_loss: 0.6196 - val_accuracy: 0.8036
Epoch 60/100
210/210 [=====] - 2s 9ms/step - loss: 0.7879 - accuracy: 0.7579 - val_loss: 0.6491 - val_accuracy: 0.7967
Epoch 61/100
210/210 [=====] - 2s 9ms/step - loss: 0.7858 - accuracy: 0.7579 - val_loss: 0.6705 - val_accuracy: 0.7917
Epoch 62/100
210/210 [=====] - 2s 9ms/step - loss: 0.7820 - accuracy: 0.7587 - val_loss: 0.7191 - val_accuracy: 0.7768
Epoch 63/100
210/210 [=====] - 2s 9ms/step - loss: 0.7814 - accuracy: 0.7588 - val_loss: 0.6435 - val_accuracy: 0.7998
Epoch 64/100
210/210 [=====] - 2s 8ms/step - loss: 0.7726 - accuracy: 0.7621 - val_loss: 0.6165 - val_accuracy: 0.8101
Epoch 65/100
210/210 [=====] - 2s 9ms/step - loss: 0.7840 - accuracy: 0.7585 - val_loss: 0.7272 - val_accuracy: 0.7704
Epoch 66/100
210/210 [=====] - 2s 9ms/step - loss: 0.7803 - accuracy: 0.7590 - val_loss: 0.6246 - val_accuracy: 0.8059
Epoch 67/100
210/210 [=====] - 2s 8ms/step - loss: 0.7712 - accuracy: 0.7610 - val_loss: 0.6348 - val_accuracy: 0.8023
Epoch 68/100
210/210 [=====] - 2s 8ms/step - loss: 0.7777 - accuracy: 0.7618 - val_loss: 0.6159 - val_accuracy: 0.8095
Epoch 69/100
210/210 [=====] - 2s 8ms/step - loss: 0.7775 - accuracy: 0.7618 - val_loss: 0.6907 - val_accuracy: 0.7777
Epoch 70/100
210/210 [=====] - 2s 8ms/step - loss: 0.7758 - accuracy: 0.7614 - val_loss: 0.6800 - val_accuracy: 0.7913
Epoch 71/100
210/210 [=====] - 2s 9ms/step - loss: 0.7733 - accuracy: 0.7582 - val_loss: 0.6315 - val_accuracy: 0.8044
Epoch 72/100
210/210 [=====] - 2s 10ms/step - loss: 0.7728 - accuracy: 0.7600 - val_loss: 0.7050 - val_accuracy: 0.7807
Epoch 73/100
210/210 [=====] - 2s 10ms/step - loss: 0.7656 - accuracy: 0.7633 - val_loss: 0.6037 - val_accuracy: 0.8102
Epoch 74/100
210/210 [=====] - 2s 9ms/step - loss: 0.7718 - accuracy: 0.7622 - val_loss: 0.6532 - val_accuracy: 0.7977
Epoch 75/100
210/210 [=====] - 2s 10ms/step - loss: 0.7647 - accuracy: 0.7633 - val_loss: 0.5694 - val_accuracy: 0.8286
Epoch 76/100
210/210 [=====] - 2s 9ms/step - loss: 0.7681 - accuracy: 0.7624 - val_loss: 0.6033 - val_accuracy: 0.8172
Epoch 77/100
210/210 [=====] - 2s 10ms/step - loss: 0.7653 - accuracy: 0.7631 - val_loss: 0.5909 - val_accuracy: 0.8187
Epoch 78/100
210/210 [=====] - 2s 10ms/step - loss: 0.7667 - accuracy: 0.7632 - val_loss: 0.6957 - val_accuracy: 0.7826
Epoch 79/100
210/210 [=====] - 2s 10ms/step - loss: 0.7621 - accuracy: 0.7660 - val_loss: 0.6037 - val_accuracy: 0.8127
Epoch 80/100
210/210 [=====] - 2s 9ms/step - loss: 0.7561 - accuracy: 0.7645 - val_loss: 0.5847 - val_accuracy: 0.8163
Epoch 81/100
210/210 [=====] - 2s 10ms/step - loss: 0.7614 - accuracy: 0.7646 - val_loss: 0.6600 - val_accuracy: 0.7804
```

```

ss: 0.6623 - val_accuracy: 0.7984
Epoch 82/100
210/210 [=====] - 2s 10ms/step - loss: 0.7658 - accuracy: 0.7615 - val_lo
ss: 0.6489 - val_accuracy: 0.7949
Epoch 83/100
210/210 [=====] - 2s 10ms/step - loss: 0.7687 - accuracy: 0.7630 - val_lo
ss: 0.5880 - val_accuracy: 0.8192
Epoch 84/100
210/210 [=====] - 2s 9ms/step - loss: 0.7561 - accuracy: 0.7667 - val_lo
ss: 0.6170 - val_accuracy: 0.8082
Epoch 85/100
210/210 [=====] - 2s 9ms/step - loss: 0.7642 - accuracy: 0.7641 - val_lo
ss: 0.6827 - val_accuracy: 0.7863
Epoch 86/100
210/210 [=====] - 2s 10ms/step - loss: 0.7605 - accuracy: 0.7650 - val_lo
ss: 0.6366 - val_accuracy: 0.8032
Epoch 87/100
210/210 [=====] - 2s 9ms/step - loss: 0.7573 - accuracy: 0.7656 - val_lo
ss: 0.5739 - val_accuracy: 0.8225
Epoch 88/100
210/210 [=====] - 2s 9ms/step - loss: 0.7584 - accuracy: 0.7661 - val_lo
ss: 0.6156 - val_accuracy: 0.8061
Epoch 89/100
210/210 [=====] - 2s 9ms/step - loss: 0.7571 - accuracy: 0.7656 - val_lo
ss: 0.5511 - val_accuracy: 0.8318
Epoch 90/100
210/210 [=====] - 2s 10ms/step - loss: 0.7519 - accuracy: 0.7678 - val_lo
ss: 0.5438 - val_accuracy: 0.8339
Epoch 91/100
210/210 [=====] - 2s 9ms/step - loss: 0.7571 - accuracy: 0.7652 - val_lo
ss: 0.5692 - val_accuracy: 0.8230
Epoch 92/100
210/210 [=====] - 2s 9ms/step - loss: 0.7549 - accuracy: 0.7659 - val_lo
ss: 0.5987 - val_accuracy: 0.8148
Epoch 93/100
210/210 [=====] - 2s 10ms/step - loss: 0.7470 - accuracy: 0.7701 - val_lo
ss: 0.6017 - val_accuracy: 0.8111
Epoch 94/100
210/210 [=====] - 2s 9ms/step - loss: 0.7546 - accuracy: 0.7681 - val_lo
ss: 0.6557 - val_accuracy: 0.8009
Epoch 95/100
210/210 [=====] - 2s 9ms/step - loss: 0.7488 - accuracy: 0.7681 - val_lo
ss: 0.6050 - val_accuracy: 0.8126
Epoch 96/100
210/210 [=====] - 2s 9ms/step - loss: 0.7450 - accuracy: 0.7686 - val_lo
ss: 0.7086 - val_accuracy: 0.7806
Epoch 97/100
210/210 [=====] - 2s 9ms/step - loss: 0.7534 - accuracy: 0.7682 - val_lo
ss: 0.6958 - val_accuracy: 0.7823
Epoch 98/100
210/210 [=====] - 2s 9ms/step - loss: 0.7501 - accuracy: 0.7696 - val_lo
ss: 0.5537 - val_accuracy: 0.8291
Epoch 99/100
210/210 [=====] - 2s 8ms/step - loss: 0.7436 - accuracy: 0.7707 - val_lo
ss: 0.6315 - val_accuracy: 0.8041
Epoch 100/100
210/210 [=====] - 2s 8ms/step - loss: 0.7524 - accuracy: 0.7660 - val_lo
ss: 0.5386 - val_accuracy: 0.8363

```

In [ ]:

```
results = model.evaluate(x_test, y_test)
```

```
563/563 [=====] - 1s 2ms/step - loss: 0.6292 - accuracy: 0.8063
```

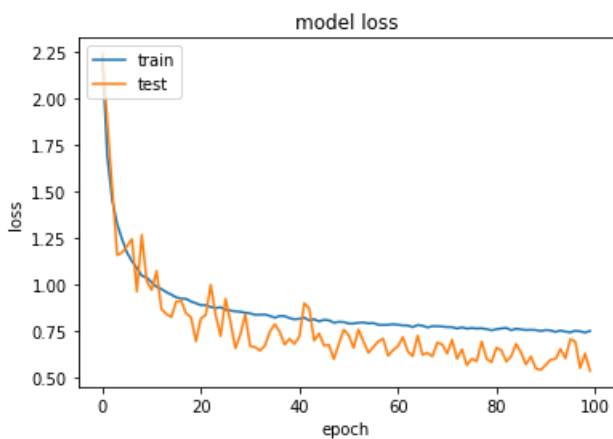
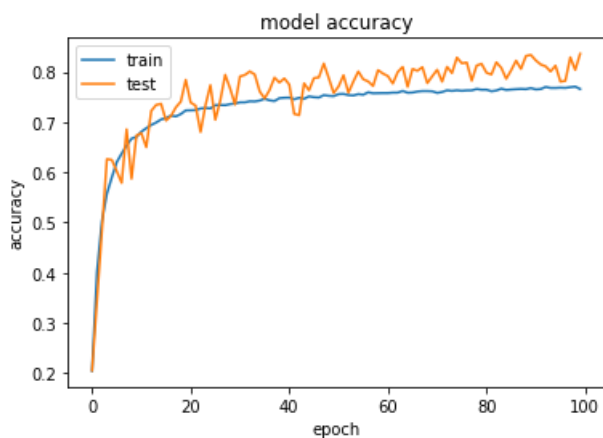
In [ ]:

```
print(model.metrics_names)
print(results)
```

```
['loss', 'accuracy']
[0.6292060613632202, 0.8062777519226074]
```

In [ ]:

```
#summarize history for accuracy
plt.plot(history1.history['accuracy'])
plt.plot(history1.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history1.history['loss'])
plt.plot(history1.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



The accuracy was found to be 80 using batch normalization and dropout methods.\ The required graphs were also plotted.

In [ ]: