# Import Libraries and modules

In [4]:
```
from keras.datasets import mnist
import tensorflow as tf
from keras import backend
import seaborn as sns
import numpy as np
from keras.initializers import RandomNormal
```

```
Using TensorFlow backend.
```

In [5]:
```
%matplotlib notebook
import matplotlib.pyplot as plt
import numpy as np
import time
# https://gist.github.com/greydanus/f6eee59eaf1d90fcb3b534a25362cea4
# https://stackoverflow.com/a/14434334
# this function is used to update the plots for each epoch and error
def plt_dynamic(x, vy, ty, ax, colors=['b']):
    ax.plot(x, vy, 'b', label="Validation Loss")
    ax.plot(x, ty, 'r', label="Train Loss")
    plt.legend()
    plt.grid()
    fig.canvas.draw()
```

In [6]:
```
import numpy as np
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten, BatchNormalization
from keras.layers import Convolution2D, MaxPooling2D,SeparableConv2D
from keras.utils import np_utils
from keras import optimizers

from keras.datasets import mnist
```

In [7]:
```
# the data, shuffled and split between train and test sets
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

In [8]:
```
print("Number of training examples :", X_train.shape[0], "and each image is of shape (%d, %d)"%(X_train.shape[1], X_
print("Number of training examples :", X_test.shape[0], "and each image is of shape (%d, %d)"%(X_test.shape[1], X_te
```

```
Number of training examples : 60000 and each image is of shape (28, 28)
Number of training examples : 10000 and each image is of shape (28, 28)
```

In [9]:
```
X_train = X_train.reshape(X_train.shape[0], 28, 28,1)
X_test = X_test.reshape(X_test.shape[0], 28, 28,1)
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255
# here we are having a class number for each image
from keras import utils as np_utils
print("Class label of first image :", y_train[0])

# lets convert this into a 10 dimensional vector
# ex: consider an image is 5 convert it into 5 => [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
# this conversion needed for MLPs
Y_train = np_utils.to_categorical(y_train, 10)
Y_test = np_utils.to_categorical(y_test, 10)

print("After converting the output into a vector : ",Y_train[0])
X_train = X_train.reshape((-1, 28, 28, 1))

print(X_train.shape)
```

```
Class label of first image : 5
After converting the output into a vector :  [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
(60000, 28, 28, 1)
```

In [8]:

```
model = Sequential()


model.add(Convolution2D(8, (3,3),activation='relu', input_shape=(28,28,1)))
model.add(BatchNormalization(axis=-1))
model.add(Convolution2D(16, 3, activation='relu'))
model.add(BatchNormalization(axis=-1))
model.add(SeparableConv2D(32, 3, activation='relu'))
model.add(BatchNormalization(axis=-1))
model.add(Convolution2D(8,1))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(SeparableConv2D(16, 3, activation='relu'))
model.add(BatchNormalization(axis=-1))
model.add(SeparableConv2D(32, 3, activation='relu'))
model.add(BatchNormalization(axis=-1))
model.add(SeparableConv2D(64, 3, activation='relu'))
model.add(BatchNormalization(axis=-1))
model.add(SeparableConv2D(128, 2, activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(8,1))
model.add(Convolution2D(16,2))
model.add(BatchNormalization(axis=-1))
model.add(Convolution2D(10,1))
model.add(Flatten())
model.add(Activation('softmax'))
```

```
WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:74: The nam
e tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:517: The na
me tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The n
ame tf.random_uniform is deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:174: The na
me tf.get_default_session is deprecated. Please use tf.compat.v1.get_default_session instead.
```

```
WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:181: The na
me tf.ConfigProto is deprecated. Please use tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:186: The na
me tf.Session is deprecated. Please use tf.compat.v1.Session instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:190: The na
me tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:199: The na
me tf.is_variable_initialized is deprecated. Please use tf.compat.v1.is_variable_initialized instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:206: The na
me tf.variables_initializer is deprecated. Please use tf.compat.v1.variables_initializer instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:1834: The n
ame tf.nn.fused_batch_norm is deprecated. Please use tf.compat.v1.nn.fused_batch_norm instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:133: The na
me tf.placeholder_with_default is deprecated. Please use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3976: The n
ame tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.
```

In [9]:
```
1 model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 8)         80
_________________________________________________________________
batch_normalization_1 (Batch (None, 26, 26, 8)         32
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 16)        1168
_________________________________________________________________
batch_normalization_2 (Batch (None, 24, 24, 16)        64
_________________________________________________________________
separable_conv2d_1 (Separabl (None, 22, 22, 32)        688
_________________________________________________________________
batch_normalization_3 (Batch (None, 22, 22, 32)        128
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 22, 22, 8)         264
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 8)         0
_________________________________________________________________
separable_conv2d_2 (Separabl (None, 9, 9, 16)          216
_________________________________________________________________
batch_normalization_4 (Batch (None, 9, 9, 16)          64
_________________________________________________________________
separable_conv2d_3 (Separabl (None, 7, 7, 32)          688
_________________________________________________________________
batch_normalization_5 (Batch (None, 7, 7, 32)          128
_________________________________________________________________
separable_conv2d_4 (Separabl (None, 5, 5, 64)          2400
_________________________________________________________________
batch_normalization_6 (Batch (None, 5, 5, 64)          256
_________________________________________________________________
separable_conv2d_5 (Separabl (None, 4, 4, 128)         8576
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 2, 2, 128)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 2, 2, 8)           1032
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 1, 1, 16)          528
_________________________________________________________________
```

```
batch_normalization_7 (Batch (None, 1, 1, 16)          64
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 1, 1, 10)          170
_________________________________________________________________
flatten_1 (Flatten)          (None, 10)                0
_________________________________________________________________
activation_1 (Activation)    (None, 10)                0
=================================================================
Total params: 16,546
Trainable params: 16,178
Non-trainable params: 368
_________________________________________________________________
```

In [10]:

```
opt=optimizers.Adam(lr=0.0001)
model.compile(loss='categorical_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
```

```
WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3295: The name tf.log is deprecated. Please use tf.math.log instead.
```

In [11]:
```
history = model.fit(X_train, Y_train, batch_size=32, epochs=50, verbose=1, validation_data=(X_test, Y_test))
```

```
WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\tensorflow_core\python\ops\math_grad.py:1424: w
here (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:986: The na
me tf.assign_add is deprecated. Please use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:973: The na
me tf.assign is deprecated. Please use tf.compat.v1.assign instead.

Train on 60000 samples, validate on 10000 samples
Epoch 1/50
60000/60000 [==============================] - 64s 1ms/step - loss: 0.8888 - acc: 0.7505 - val_loss: 0.3569 - val_ac
c: 0.9215
Epoch 2/50
60000/60000 [==============================] - 50s 827us/step - loss: 0.2990 - acc: 0.9304 - val_loss: 0.1792 - val_a
cc: 0.9583
Epoch 3/50
60000/60000 [==============================] - 51s 842us/step - loss: 0.1727 - acc: 0.9577 - val_loss: 0.1165 - val_a
cc: 0.9710
Epoch 4/50
60000/60000 [==============================] - 52s 862us/step - loss: 0.1208 - acc: 0.9687 - val_loss: 0.0861 - val_a
cc: 0.9779
Epoch 5/50
60000/60000 [==============================] - 54s 895us/step - loss: 0.0936 - acc: 0.9746 - val_loss: 0.0711 - val_a
cc: 0.9812
Epoch 6/50
60000/60000 [==============================] - 50s 839us/step - loss: 0.0770 - acc: 0.9787 - val_loss: 0.0622 - val_a
cc: 0.9834
Epoch 7/50
60000/60000 [==============================] - 50s 839us/step - loss: 0.0662 - acc: 0.9814 - val_loss: 0.0559 - val_a
cc: 0.9844
Epoch 8/50
60000/60000 [==============================] - 51s 850us/step - loss: 0.0599 - acc: 0.9830 - val_loss: 0.0504 - val_a
cc: 0.9859
Epoch 9/50
60000/60000 [==============================] - 50s 839us/step - loss: 0.0529 - acc: 0.9845 - val_loss: 0.0464 - val_a
cc: 0.9869
Epoch 10/50
```

```
60000/60000 [==============================] - 51s 846us/step - loss: 0.0470 - acc: 0.9869 - val_loss: 0.0468 - val_a
cc: 0.9864
Epoch 11/50
60000/60000 [==============================] - 53s 876us/step - loss: 0.0435 - acc: 0.9872 - val_loss: 0.0441 - val_a
cc: 0.9866
Epoch 12/50
60000/60000 [==============================] - 52s 866us/step - loss: 0.0404 - acc: 0.9883 - val_loss: 0.0407 - val_a
cc: 0.9876
Epoch 13/50
60000/60000 [==============================] - 52s 861us/step - loss: 0.0364 - acc: 0.9895 - val_loss: 0.0398 - val_a
cc: 0.9880
Epoch 14/50
60000/60000 [==============================] - 51s 858us/step - loss: 0.0338 - acc: 0.9902 - val_loss: 0.0400 - val_a
cc: 0.9875
Epoch 15/50
60000/60000 [==============================] - 52s 862us/step - loss: 0.0315 - acc: 0.9906 - val_loss: 0.0387 - val_a
cc: 0.9878
Epoch 16/50
60000/60000 [==============================] - 52s 860us/step - loss: 0.0284 - acc: 0.9917 - val_loss: 0.0398 - val_a
cc: 0.9870
Epoch 17/50
60000/60000 [==============================] - 52s 863us/step - loss: 0.0264 - acc: 0.9922 - val_loss: 0.0397 - val_a
cc: 0.9874
Epoch 18/50
60000/60000 [==============================] - 52s 860us/step - loss: 0.0250 - acc: 0.9928 - val_loss: 0.0390 - val_a
cc: 0.9878
Epoch 19/50
60000/60000 [==============================] - 51s 858us/step - loss: 0.0240 - acc: 0.9930 - val_loss: 0.0362 - val_a
cc: 0.9876
Epoch 20/50
60000/60000 [==============================] - 52s 873us/step - loss: 0.0221 - acc: 0.9934 - val_loss: 0.0375 - val_a
cc: 0.9876
Epoch 21/50
60000/60000 [==============================] - 52s 861us/step - loss: 0.0207 - acc: 0.9940 - val_loss: 0.0349 - val_a
cc: 0.9884
Epoch 22/50
60000/60000 [==============================] - 52s 866us/step - loss: 0.0193 - acc: 0.9944 - val_loss: 0.0371 - val_a
cc: 0.9880
Epoch 23/50
60000/60000 [==============================] - 53s 878us/step - loss: 0.0185 - acc: 0.9947 - val_loss: 0.0376 - val_a
cc: 0.9892
Epoch 24/50
```

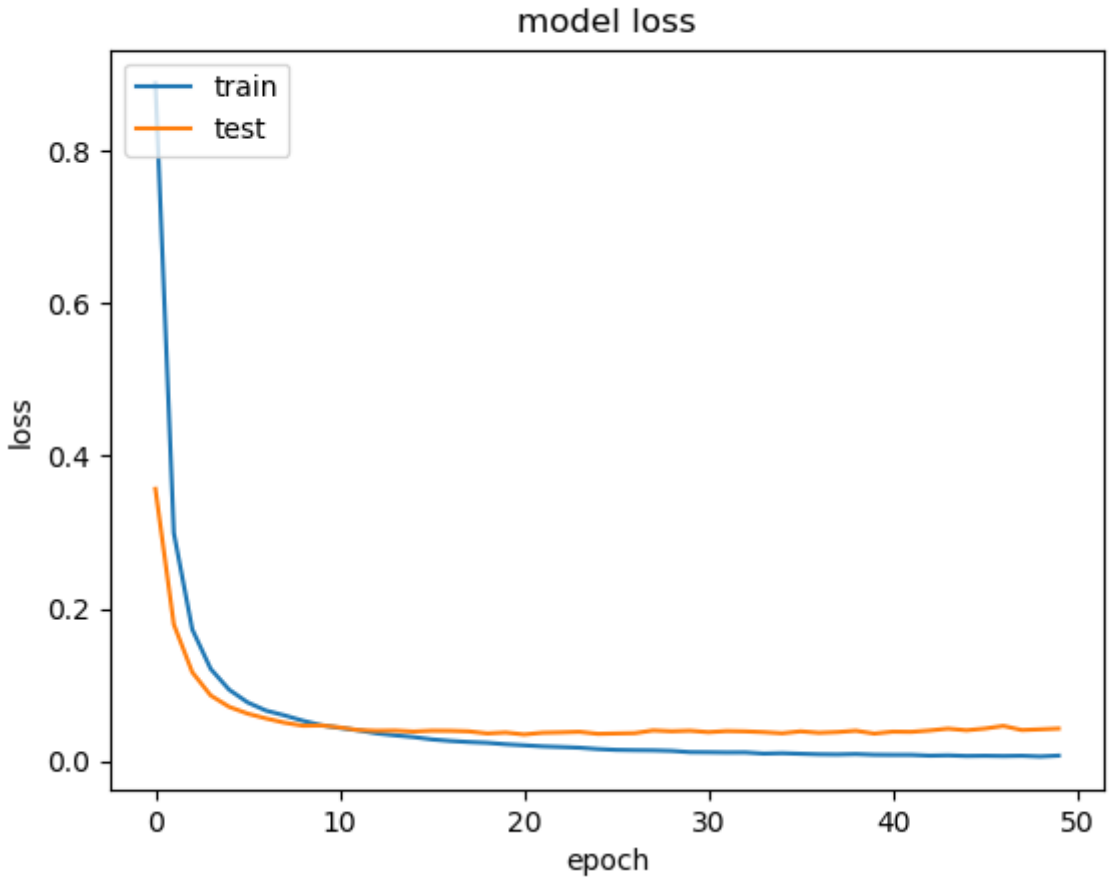
```
60000/60000 [==============================] - 52s 861us/step - loss: 0.0176 - acc: 0.9949 - val_loss: 0.0384 - val_a
cc: 0.9880
Epoch 25/50
60000/60000 [==============================] - 52s 864us/step - loss: 0.0158 - acc: 0.9954 - val_loss: 0.0357 - val_a
cc: 0.9886
Epoch 26/50
60000/60000 [==============================] - 52s 859us/step - loss: 0.0147 - acc: 0.9957 - val_loss: 0.0362 - val_a
cc: 0.9884
Epoch 27/50
60000/60000 [==============================] - 53s 878us/step - loss: 0.0143 - acc: 0.9957 - val_loss: 0.0366 - val_a
cc: 0.9888
Epoch 28/50
60000/60000 [==============================] - 52s 866us/step - loss: 0.0141 - acc: 0.9958 - val_loss: 0.0402 - val_a
cc: 0.9873
Epoch 29/50
60000/60000 [==============================] - 52s 861us/step - loss: 0.0134 - acc: 0.9961 - val_loss: 0.0391 - val_a
cc: 0.9882
Epoch 30/50
60000/60000 [==============================] - 52s 865us/step - loss: 0.0117 - acc: 0.9965 - val_loss: 0.0398 - val_a
cc: 0.9882
Epoch 31/50
60000/60000 [==============================] - 52s 863us/step - loss: 0.0116 - acc: 0.9965 - val_loss: 0.0379 - val_a
cc: 0.9885
Epoch 32/50
60000/60000 [==============================] - 52s 865us/step - loss: 0.0113 - acc: 0.9967 - val_loss: 0.0394 - val_a
cc: 0.9881
Epoch 33/50
60000/60000 [==============================] - 52s 870us/step - loss: 0.0114 - acc: 0.9967 - val_loss: 0.0389 - val_a
cc: 0.9881
Epoch 34/50
60000/60000 [==============================] - 51s 857us/step - loss: 0.0098 - acc: 0.9971 - val_loss: 0.0379 - val_a
cc: 0.9891
Epoch 35/50
60000/60000 [==============================] - 51s 855us/step - loss: 0.0103 - acc: 0.9970 - val_loss: 0.0364 - val_a
cc: 0.9903
Epoch 36/50
60000/60000 [==============================] - 52s 875us/step - loss: 0.0096 - acc: 0.9972 - val_loss: 0.0390 - val_a
cc: 0.9895
Epoch 37/50
60000/60000 [==============================] - 52s 866us/step - loss: 0.0089 - acc: 0.9975 - val_loss: 0.0370 - val_a
cc: 0.9892
Epoch 38/50
```

```
60000/60000 [==============================] - 52s 858us/step - loss: 0.0087 - acc: 0.9975 - val_loss: 0.0380 - val_a
cc: 0.9888
Epoch 39/50
60000/60000 [==============================] - 59s 983us/step - loss: 0.0092 - acc: 0.9973 - val_loss: 0.0398 - val_a
cc: 0.9885
Epoch 40/50
60000/60000 [==============================] - 62s 1ms/step - loss: 0.0083 - acc: 0.9975 - val_loss: 0.0360 - val_ac
c: 0.9889
Epoch 41/50
60000/60000 [==============================] - 59s 985us/step - loss: 0.0081 - acc: 0.9978 - val_loss: 0.0386 - val_a
cc: 0.9888
Epoch 42/50
60000/60000 [==============================] - 82s 1ms/step - loss: 0.0081 - acc: 0.9974 - val_loss: 0.0383 - val_ac
c: 0.9890
Epoch 43/50
60000/60000 [==============================] - 68s 1ms/step - loss: 0.0071 - acc: 0.9980 - val_loss: 0.0401 - val_ac
c: 0.9895
Epoch 44/50
60000/60000 [==============================] - 70s 1ms/step - loss: 0.0075 - acc: 0.9979 - val_loss: 0.0426 - val_ac
c: 0.9881
Epoch 45/50
60000/60000 [==============================] - 55s 918us/step - loss: 0.0067 - acc: 0.9982 - val_loss: 0.0404 - val_a
cc: 0.9889
Epoch 46/50
60000/60000 [==============================] - 53s 880us/step - loss: 0.0070 - acc: 0.9980 - val_loss: 0.0426 - val_a
cc: 0.9881
Epoch 47/50
60000/60000 [==============================] - 58s 967us/step - loss: 0.0066 - acc: 0.9982 - val_loss: 0.0461 - val_a
cc: 0.9872
Epoch 48/50
60000/60000 [==============================] - 53s 881us/step - loss: 0.0070 - acc: 0.9979 - val_loss: 0.0407 - val_a
cc: 0.9886
Epoch 49/50
60000/60000 [==============================] - 52s 862us/step - loss: 0.0060 - acc: 0.9983 - val_loss: 0.0416 - val_a
cc: 0.9886
Epoch 50/50
60000/60000 [==============================] - 51s 857us/step - loss: 0.0072 - acc: 0.9977 - val_loss: 0.0426 - val_a
cc: 0.9874
```

In [12]:
```
score = model.evaluate(X_test, Y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])
```

```
Test score: 0.04264953793902096
Test accuracy: 0.9874
```

In [24]:
```
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()
```



In [16]:

```
model2 = Sequential()


model2.add(Convolution2D(8, 3,activation='relu', input_shape=(28,28,1)))
model2.add(BatchNormalization(axis=-1))
model2.add(MaxPooling2D(pool_size=(2,2)))
model2.add(Convolution2D(16, 3, activation='relu'))
model2.add(Dropout(0.2))
model2.add(BatchNormalization(axis=-1))
model2.add(MaxPooling2D(pool_size=(2,2)))
model2.add(Convolution2D(32, 3, activation='relu'))
model2.add(Dropout(0.3))
model2.add(MaxPooling2D(pool_size=(2,2)))
model2.add(Flatten())

model2.add(Dense(30, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))
model2.add(Dense(10, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))

model2.add(Activation('softmax'))
```

```
WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3445: calling
dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
WARNING:tensorflow:From C:\Users\Akarshan\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4115: The nam
e tf.random_normal is deprecated. Please use tf.random.normal instead.
```

In [17]:
```
1 model2.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_7 (Conv2D)            (None, 26, 26, 8)         80
_________________________________________________________________
batch_normalization_8 (Batch (None, 26, 26, 8)         32
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 13, 13, 8)         0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 11, 11, 16)        1168
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 16)        0
_________________________________________________________________
batch_normalization_9 (Batch (None, 11, 11, 16)        64
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 5, 5, 16)          0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 3, 3, 32)          4640
_________________________________________________________________
dropout_2 (Dropout)          (None, 3, 3, 32)          0
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 1, 1, 32)          0
_________________________________________________________________
flatten_2 (Flatten)          (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 30)                990
_________________________________________________________________
dense_2 (Dense)              (None, 10)                310
_________________________________________________________________
activation_2 (Activation)    (None, 10)                0
=================================================================
Total params: 7,284
Trainable params: 7,236
Non-trainable params: 48
_________________________________________________________________
```

In [18]:

```
opt=optimizers.Adam(lr=0.0001)
model2.compile(loss='categorical_crossentropy',
               optimizer=opt,
               metrics=['accuracy'])
```

In [19]:
```
history2 = model2.fit(X_train, Y_train, batch_size=32, epochs=50, verbose=1, validation_data=(X_test, Y_test))
```

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/50
60000/60000 [==============================] - 22s 363us/step - loss: 1.4108 - acc: 0.5721 - val_loss: 0.9785 - val_a
cc: 0.7414
Epoch 2/50
60000/60000 [==============================] - 23s 379us/step - loss: 0.8175 - acc: 0.7501 - val_loss: 0.7734 - val_a
cc: 0.8153
Epoch 3/50
60000/60000 [==============================] - 25s 419us/step - loss: 0.7252 - acc: 0.7954 - val_loss: 0.7070 - val_a
cc: 0.8387
Epoch 4/50
60000/60000 [==============================] - 25s 413us/step - loss: 0.6803 - acc: 0.8180 - val_loss: 0.6660 - val_a
cc: 0.8502
Epoch 5/50
60000/60000 [==============================] - 25s 419us/step - loss: 0.6528 - acc: 0.8303 - val_loss: 0.6389 - val_a
cc: 0.8577
Epoch 6/50
60000/60000 [==============================] - 26s 438us/step - loss: 0.6368 - acc: 0.8364 - val_loss: 0.6169 - val_a
cc: 0.8612
Epoch 7/50
60000/60000 [==============================] - 25s 412us/step - loss: 0.6203 - acc: 0.8432 - val_loss: 0.6074 - val_a
cc: 0.8642
Epoch 8/50
60000/60000 [==============================] - 25s 412us/step - loss: 0.6058 - acc: 0.8478 - val_loss: 0.5919 - val_a
cc: 0.8682
Epoch 9/50
60000/60000 [==============================] - 25s 411us/step - loss: 0.6001 - acc: 0.8506 - val_loss: 0.5831 - val_a
cc: 0.8706
Epoch 10/50
60000/60000 [==============================] - 24s 408us/step - loss: 0.5926 - acc: 0.8538 - val_loss: 0.5742 - val_a
cc: 0.8729
Epoch 11/50
60000/60000 [==============================] - 25s 409us/step - loss: 0.5831 - acc: 0.8572 - val_loss: 0.5724 - val_a
cc: 0.8751
Epoch 12/50
60000/60000 [==============================] - 25s 409us/step - loss: 0.5787 - acc: 0.8584 - val_loss: 0.5674 - val_a
cc: 0.8751
Epoch 13/50
60000/60000 [==============================] - 24s 408us/step - loss: 0.5735 - acc: 0.8594 - val_loss: 0.5633 - val_a
```

```
cc: 0.8756
Epoch 14/50
60000/60000 [==============================] - 25s 424us/step - loss: 0.5691 - acc: 0.8619 - val_loss: 0.5570 - val_a
cc: 0.8782
Epoch 15/50
60000/60000 [==============================] - 25s 419us/step - loss: 0.5650 - acc: 0.8639 - val_loss: 0.5544 - val_a
cc: 0.8781
Epoch 16/50
60000/60000 [==============================] - 25s 409us/step - loss: 0.5596 - acc: 0.8643 - val_loss: 0.5491 - val_a
cc: 0.8780
Epoch 17/50
60000/60000 [==============================] - 25s 412us/step - loss: 0.5577 - acc: 0.8660 - val_loss: 0.5476 - val_a
cc: 0.8797
Epoch 18/50
60000/60000 [==============================] - 25s 411us/step - loss: 0.5549 - acc: 0.8661 - val_loss: 0.5454 - val_a
cc: 0.8788
Epoch 19/50
60000/60000 [==============================] - 25s 409us/step - loss: 0.5523 - acc: 0.8677 - val_loss: 0.5447 - val_a
cc: 0.8809
Epoch 20/50
60000/60000 [==============================] - 25s 413us/step - loss: 0.5488 - acc: 0.8684 - val_loss: 0.5405 - val_a
cc: 0.8803
Epoch 21/50
60000/60000 [==============================] - 25s 411us/step - loss: 0.5463 - acc: 0.8703 - val_loss: 0.5381 - val_a
cc: 0.8801
Epoch 22/50
60000/60000 [==============================] - 25s 413us/step - loss: 0.5455 - acc: 0.8700 - val_loss: 0.5348 - val_a
cc: 0.8818
Epoch 23/50
60000/60000 [==============================] - 25s 420us/step - loss: 0.5444 - acc: 0.8713 - val_loss: 0.5347 - val_a
cc: 0.8812
Epoch 24/50
60000/60000 [==============================] - 24s 408us/step - loss: 0.5423 - acc: 0.8721 - val_loss: 0.5353 - val_a
cc: 0.8819
Epoch 25/50
60000/60000 [==============================] - 25s 409us/step - loss: 0.5401 - acc: 0.8715 - val_loss: 0.5346 - val_a
cc: 0.8819
Epoch 26/50
60000/60000 [==============================] - 25s 411us/step - loss: 0.5380 - acc: 0.8728 - val_loss: 0.5329 - val_a
cc: 0.8823
Epoch 27/50
60000/60000 [==============================] - 25s 410us/step - loss: 0.5380 - acc: 0.8725 - val_loss: 0.5293 - val_a
```

```
cc: 0.8832
Epoch 28/50
60000/60000 [==============================] - 24s 408us/step - loss: 0.5362 - acc: 0.8734 - val_loss: 0.5274 - val_a
cc: 0.8829
Epoch 29/50
60000/60000 [==============================] - 25s 414us/step - loss: 0.5330 - acc: 0.8745 - val_loss: 0.5267 - val_a
cc: 0.8838
Epoch 30/50
60000/60000 [==============================] - 25s 413us/step - loss: 0.5332 - acc: 0.8744 - val_loss: 0.5276 - val_a
cc: 0.8826
Epoch 31/50
60000/60000 [==============================] - 25s 409us/step - loss: 0.5313 - acc: 0.8744 - val_loss: 0.5243 - val_a
cc: 0.8826
Epoch 32/50
60000/60000 [==============================] - 26s 426us/step - loss: 0.5294 - acc: 0.8752 - val_loss: 0.5269 - val_a
cc: 0.8833
Epoch 33/50
60000/60000 [==============================] - 25s 423us/step - loss: 0.5292 - acc: 0.8754 - val_loss: 0.5249 - val_a
cc: 0.8840
Epoch 34/50
60000/60000 [==============================] - 25s 411us/step - loss: 0.5303 - acc: 0.8760 - val_loss: 0.5239 - val_a
cc: 0.8842
Epoch 35/50
60000/60000 [==============================] - 25s 414us/step - loss: 0.5301 - acc: 0.8747 - val_loss: 0.5221 - val_a
cc: 0.8844
Epoch 36/50
60000/60000 [==============================] - 25s 412us/step - loss: 0.5274 - acc: 0.8773 - val_loss: 0.5233 - val_a
cc: 0.8839
Epoch 37/50
60000/60000 [==============================] - 25s 410us/step - loss: 0.5274 - acc: 0.8765 - val_loss: 0.5191 - val_a
cc: 0.8853
Epoch 38/50
60000/60000 [==============================] - 25s 410us/step - loss: 0.5248 - acc: 0.8772 - val_loss: 0.5189 - val_a
cc: 0.8841
Epoch 39/50
60000/60000 [==============================] - 25s 410us/step - loss: 0.5248 - acc: 0.8775 - val_loss: 0.5201 - val_a
cc: 0.8837
Epoch 40/50
60000/60000 [==============================] - 25s 411us/step - loss: 0.5253 - acc: 0.8777 - val_loss: 0.5166 - val_a
cc: 0.8841
Epoch 41/50
60000/60000 [==============================] - 25s 420us/step - loss: 0.5227 - acc: 0.8773 - val_loss: 0.5197 - val_a
```

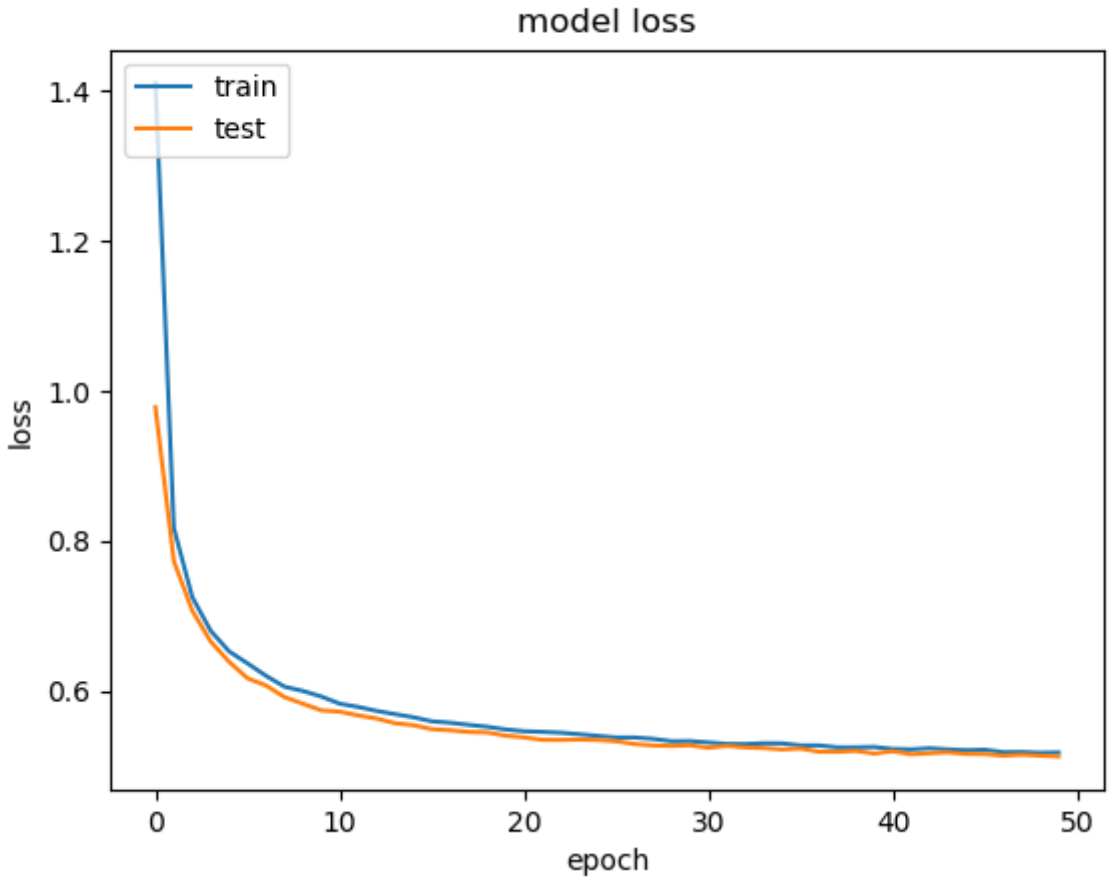
```
cc: 0.8847
Epoch 42/50
60000/60000 [==============================] - 25s 409us/step - loss: 0.5221 - acc: 0.8791 - val_loss: 0.5156 - val_a
cc: 0.8847
Epoch 43/50
60000/60000 [==============================] - 25s 410us/step - loss: 0.5236 - acc: 0.8780 - val_loss: 0.5170 - val_a
cc: 0.8842
Epoch 44/50
60000/60000 [==============================] - 25s 411us/step - loss: 0.5221 - acc: 0.8782 - val_loss: 0.5181 - val_a
cc: 0.8846
Epoch 45/50
60000/60000 [==============================] - 25s 409us/step - loss: 0.5210 - acc: 0.8791 - val_loss: 0.5161 - val_a
cc: 0.8846
Epoch 46/50
60000/60000 [==============================] - 25s 416us/step - loss: 0.5217 - acc: 0.8787 - val_loss: 0.5159 - val_a
cc: 0.8845
Epoch 47/50
60000/60000 [==============================] - 25s 420us/step - loss: 0.5184 - acc: 0.8798 - val_loss: 0.5138 - val_a
cc: 0.8851
Epoch 48/50
60000/60000 [==============================] - 26s 437us/step - loss: 0.5186 - acc: 0.8799 - val_loss: 0.5151 - val_a
cc: 0.8860
Epoch 49/50
60000/60000 [==============================] - 25s 409us/step - loss: 0.5176 - acc: 0.8798 - val_loss: 0.5138 - val_a
cc: 0.8840
Epoch 50/50
60000/60000 [==============================] - 25s 410us/step - loss: 0.5179 - acc: 0.8798 - val_loss: 0.5126 - val_a
cc: 0.8859
```

In [20]:

```
score = model2.evaluate(X_test, Y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])
```

```
Test score: 0.5126468414783478
Test accuracy: 0.8859
```

In [26]:
```
plt.plot(history2.history['loss'])
plt.plot(history2.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



In [42]:

```
model3 = Sequential()


model3.add(Convolution2D(8, 3,activation='relu', input_shape=(28,28,1)))
model3.add(BatchNormalization(axis=-1))

model3.add(Convolution2D(16, 3, activation='relu'))
model3.add(BatchNormalization(axis=-1))
model3.add(MaxPooling2D(pool_size=(2,2)))

model3.add(Convolution2D(32, 3, activation='relu'))
model3.add(BatchNormalization(axis=-1))

model3.add(Convolution2D(64, 3, activation='relu'))
model3.add(BatchNormalization(axis=-1))

model3.add(Convolution2D(128, 3, activation='relu'))
model3.add(BatchNormalization(axis=-1))
model3.add(MaxPooling2D(pool_size=(2,2)))

model3.add(Flatten())

model3.add(Dense(50, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))
model3.add(Dense(10, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))

model3.add(Activation('softmax'))
```

In [43]:
```
1 model3.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_31 (Conv2D)           (None, 26, 26, 8)         80
_________________________________________________________________
batch_normalization_16 (Batc (None, 26, 26, 8)         32
_________________________________________________________________
conv2d_32 (Conv2D)           (None, 24, 24, 16)        1168
_________________________________________________________________
batch_normalization_17 (Batc (None, 24, 24, 16)        64
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 12, 12, 16)        0
_________________________________________________________________
conv2d_33 (Conv2D)           (None, 10, 10, 32)        4640
_________________________________________________________________
batch_normalization_18 (Batc (None, 10, 10, 32)        128
_________________________________________________________________
conv2d_34 (Conv2D)           (None, 8, 8, 64)          18496
_________________________________________________________________
batch_normalization_19 (Batc (None, 8, 8, 64)          256
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 6, 6, 128)         73856
_________________________________________________________________
batch_normalization_20 (Batc (None, 6, 6, 128)         512
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 3, 3, 128)         0
_________________________________________________________________
flatten_7 (Flatten)          (None, 1152)              0
_________________________________________________________________
dense_13 (Dense)             (None, 50)                57650
_________________________________________________________________
dense_14 (Dense)             (None, 10)                510
_________________________________________________________________
activation_7 (Activation)    (None, 10)                0
=================================================================
Total params: 157,392
Trainable params: 156,896
Non-trainable params: 496
_________________________________________________________________
```

In [44]:

```
opt=optimizers.Adam(lr=0.0001)

model3.compile(loss='categorical_crossentropy',
               optimizer=opt,
               metrics=['accuracy'])
```

In [45]:
```
history3 = model3.fit(X_train, Y_train, batch_size=64, epochs=20, verbose=1, validation_data=(X_test, Y_test))
```

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/20
60000/60000 [==============================] - 48s 796us/step - loss: 0.7781 - acc: 0.7294 - val_loss: 0.5211 - val_ac
c: 0.7912
Epoch 2/20
60000/60000 [==============================] - 43s 716us/step - loss: 0.5028 - acc: 0.7954 - val_loss: 0.4964 - val_ac
c: 0.7953
Epoch 3/20
60000/60000 [==============================] - 42s 693us/step - loss: 0.4810 - acc: 0.7996 - val_loss: 0.4871 - val_ac
c: 0.7970
Epoch 4/20
60000/60000 [==============================] - 42s 705us/step - loss: 0.4696 - acc: 0.8016 - val_loss: 0.4838 - val_ac
c: 0.7963
Epoch 5/20
60000/60000 [==============================] - 45s 743us/step - loss: 0.4624 - acc: 0.8029 - val_loss: 0.4800 - val_ac
c: 0.7978
Epoch 6/20
60000/60000 [==============================] - 44s 741us/step - loss: 0.4572 - acc: 0.8039 - val_loss: 0.4813 - val_ac
c: 0.7965
Epoch 7/20
60000/60000 [==============================] - 45s 748us/step - loss: 0.4547 - acc: 0.8044 - val_loss: 0.4789 - val_ac
c: 0.7973
Epoch 8/20
60000/60000 [==============================] - 45s 752us/step - loss: 0.4527 - acc: 0.8045 - val_loss: 0.4791 - val_ac
c: 0.7971
Epoch 9/20
60000/60000 [==============================] - 43s 722us/step - loss: 0.4519 - acc: 0.8046 - val_loss: 0.4846 - val_ac
c: 0.7971
Epoch 10/20
60000/60000 [==============================] - 43s 714us/step - loss: 0.4510 - acc: 0.8046 - val_loss: 0.4839 - val_ac
c: 0.7972
Epoch 11/20
60000/60000 [==============================] - 43s 711us/step - loss: 0.4513 - acc: 0.8046 - val_loss: 0.4939 - val_ac
c: 0.7949
Epoch 12/20
60000/60000 [==============================] - 45s 751us/step - loss: 0.4514 - acc: 0.8046 - val_loss: 0.4865 - val_ac
c: 0.7981
Epoch 13/20
60000/60000 [==============================] - 44s 739us/step - loss: 0.4502 - acc: 0.8047 - val_loss: 0.4810 - val_ac
```

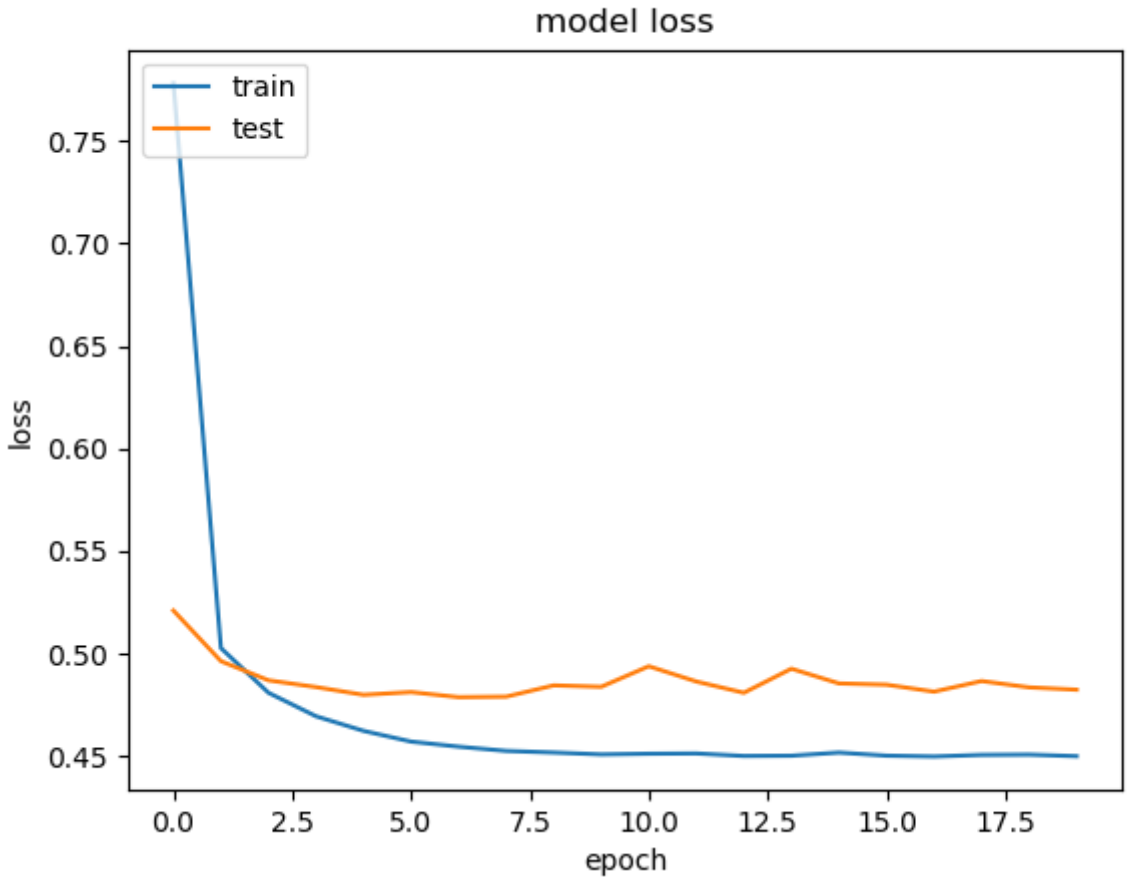
```
c: 0.7981
Epoch 14/20
60000/60000 [==============================] - 42s 707us/step - loss: 0.4503 - acc: 0.8047 - val_loss: 0.4927 - val_ac
c: 0.7952
Epoch 15/20
60000/60000 [==============================] - 48s 800us/step - loss: 0.4519 - acc: 0.8045 - val_loss: 0.4856 - val_ac
c: 0.7973
Epoch 16/20
60000/60000 [==============================] - 28s 473us/step - loss: 0.4504 - acc: 0.8047 - val_loss: 0.4850 - val_ac
c: 0.7978
Epoch 17/20
60000/60000 [==============================] - 23s 390us/step - loss: 0.4500 - acc: 0.8047 - val_loss: 0.4816 - val_ac
c: 0.7975
Epoch 18/20
60000/60000 [==============================] - 23s 389us/step - loss: 0.4507 - acc: 0.8045 - val_loss: 0.4867 - val_ac
c: 0.7976
Epoch 19/20
60000/60000 [==============================] - 23s 391us/step - loss: 0.4509 - acc: 0.8045 - val_loss: 0.4836 - val_ac
c: 0.7980
Epoch 20/20
60000/60000 [==============================] - 23s 390us/step - loss: 0.4501 - acc: 0.8047 - val_loss: 0.4825 - val_ac
c: 0.7986
```

In [46]:

```
score = model3.evaluate(X_test, Y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])
```

```
Test score: 0.48252001395225524
Test accuracy: 0.7986
```

In [47]:
```
plt.plot(history3.history['loss'])
plt.plot(history3.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



In [49]:

```
from prettytable import PrettyTable

x = PrettyTable()

x.field_names = ["S no.", "Model Description", "Accuracy", "Score"]

x.add_row(["1"," 11 layer Separable Conv with Receptive field of 28X28 "," 98.74", "0.0426"])
x.add_row(["2"," 3 layer Conv followed BN and Maxpool with dropouts", "88.59", "0.2693"])
x.add_row(["3", "5 layers COnv and Maxpool", "79.86", "0.4825"])
print(x)
```

```
+-------+------------------------------------------------------------+----------+--------+
| S no. |                   Model Description                        | Accuracy | Score  |
+-------+------------------------------------------------------------+----------+--------+
|   1   |  11 layer Separable Conv with Receptive field of 28X28     |   98.74  | 0.0426 |
|   2   |    3 layer Conv followed BN and Maxpool with dropouts      |  88.59   | 0.2693 |
|   3   |               5 layers COnv and Maxpool                    |  79.86   | 0.4825 |
+-------+------------------------------------------------------------+----------+--------+
```