In [0]:
```
# Credits: https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py


from __future__ import print_function
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K

batch_size = 128
num_classes = 10
epochs = 12

# input image dimensions
img_rows, img_cols = 28, 28

# the data, split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()

if K.image_data_format() == 'channels_first':
    x_train = x_train.reshape(x_train.shape[0], 1, img_rows, img_cols)
    x_test = x_test.reshape(x_test.shape[0], 1, img_rows, img_cols)
    input_shape = (1, img_rows, img_cols)
else:
    x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 1)
    x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 1)
    input_shape = (img_rows, img_cols, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print('x_train shape:', x_train.shape)
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')

# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

model = Sequential()
model.add(Conv2D(256, kernel_size=(2, 2),
                 activation='relu',
                 input_shape=input_shape))
model.add(Conv2D(128, (2, 2), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2),strides=(1,1),padding='same'))
model.add(Dropout(0.25))

model.add(Conv2D(64, (2, 2), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2),strides=(1,1),padding='same'))
model.add(Dropout(0.25))
```

```
model.add(Flatten())
model.add(Dense(8, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              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])
```

```
Using TensorFlow backend.
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade (https://www.tensorflow.org/guide/migrate) now or ensure your notebook will continue to use TensorFlow 1.x via the `%tensorflow_version 1.x` magic: more info (https://colab.research.google.com/notebooks/tensorflow_version.ipynb).

```
Downloading data from https://s3.amazonaws.com/img-datasets/mnist.npz (http
s://s3.amazonaws.com/img-datasets/mnist.npz)
11493376/11490434 [==============================] - 1s 0us/step
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:66: The name tf.get_default_graph is deprecated. Please
use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:541: The name tf.placeholder is deprecated. Please use
 tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:4432: The name tf.random_uniform is deprecated. Please
 use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:4267: The name tf.nn.max_pool is deprecated. Please use
tf.nn.max_pool2d instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:148: The name tf.placeholder_with_default is deprecate
d. Please use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:3733: calling dropout (from tensorflow.python.ops.nn_op
s) 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 - k
eep_prob`.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimize
```

```
rs.py:793: The name tf.train.Optimizer is deprecated. Please use tf.compat.v
1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:3576: The name tf.log is deprecated. Please use tf.mat
h.log instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_cor
e/python/ops/math_grad.py:1424: where (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 /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Please use
 tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:1020: The name tf.assign is deprecated. Please use tf.c
ompat.v1.assign instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:3005: The name tf.Session is deprecated. Please use tf.
compat.v1.Session instead.

Train on 60000 samples, validate on 10000 samples
Epoch 1/12
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:190: The name tf.get_default_session is deprecated. Ple
ase use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:197: The name tf.ConfigProto is deprecated. Please use
 tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:207: The name tf.global_variables is deprecated. Please
use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:216: The name tf.is_variable_initialized is deprecated.
Please use tf.compat.v1.is_variable_initialized instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/
tensorflow_backend.py:223: The name tf.variables_initializer is deprecated. P
lease use tf.compat.v1.variables_initializer instead.

60000/60000 [==============================] - 988s 16ms/step - loss: 1.5570
 - acc: 0.4131 - val_loss: 0.6736 - val_acc: 0.9268
Epoch 2/12
60000/60000 [==============================] - 984s 16ms/step - loss: 1.2771
 - acc: 0.5089 - val_loss: 0.4818 - val_acc: 0.9511
Epoch 3/12
60000/60000 [==============================] - 990s 16ms/step - loss: 1.1056
 - acc: 0.5898 - val_loss: 0.3940 - val_acc: 0.9499
Epoch 4/12
60000/60000 [==============================] - 982s 16ms/step - loss: 0.9541
 - acc: 0.6658 - val_loss: 0.2251 - val_acc: 0.9683
```

```
Epoch 5/12
60000/60000 [==============================] - 1026s 17ms/step - loss: 0.9302
- acc: 0.6762 - val_loss: 0.2157 - val_acc: 0.9738
Epoch 6/12
60000/60000 [==============================] - 1039s 17ms/step - loss: 0.9128
- acc: 0.6818 - val_loss: 0.2108 - val_acc: 0.9691
Epoch 7/12
60000/60000 [==============================] - 1019s 17ms/step - loss: 0.8987
- acc: 0.6953 - val_loss: 0.2288 - val_acc: 0.9722
Epoch 8/12
60000/60000 [==============================] - 1009s 17ms/step - loss: 0.8907
- acc: 0.7032 - val_loss: 0.2261 - val_acc: 0.9707
Epoch 9/12
60000/60000 [==============================] - 1009s 17ms/step - loss: 0.8860
- acc: 0.7042 - val_loss: 0.2035 - val_acc: 0.9748
Epoch 10/12
60000/60000 [==============================] - 990s 17ms/step - loss: 0.8804
 - acc: 0.7115 - val_loss: 0.1906 - val_acc: 0.9749
Epoch 11/12
60000/60000 [==============================] - 1010s 17ms/step - loss: 0.8709
- acc: 0.7190 - val_loss: 0.2279 - val_acc: 0.9700
Epoch 12/12
60000/60000 [==============================] - 1007s 17ms/step - loss: 0.8671
- acc: 0.7197 - val_loss: 0.1930 - val_acc: 0.9745
Test loss: 0.1930493854880333
Test accuracy: 0.9745
```

In [ ]:
```
# history=model.fit(x_train, y_train,
#           batch_size=batch_size,
#           epochs=epochs,
#           verbose=1,
#           validation_data=(x_test, y_test))

# vy = history.history['val_loss']
# ty = history.history['loss']
```

Error Plots for 3 Layer CNN on MNIST

In [3]:
```
epochs=12
x = list(range(1,epochs+1))
ty=[1.557,1.2771,1.1056,0.9541,0.9302,0.9128,0.8987,0.8907,0.886,0.8804,0.8709,0
vy=[0.6736,0.4818,0.394,0.2251,0.2157,0.2108,0.2288,0.2261,0.2035,0.1906,0.2279,(

%matplotlib notebook
import matplotlib.pyplot as plt
import numpy as np
import time

def plt_dynamic(x, vy, ty, ax, colors=['b']):
    ax.plot(x, vy, 'b', label="Test Loss")
    ax.plot(x, ty, 'r', label="Train Loss")
    plt.legend()
    plt.grid()

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ;
ax.set_ylabel('Categorical Crossentropy Loss')
ax.set_title(label="3 Layer CNN on MNIST")
plt_dynamic(x, vy, ty, ax)
```