

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) (<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) (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.v1.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.math.log instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/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.compat.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. Please 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. Please 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=[1.557,1.2771,1.1056,0.9541,0.9302,0.9128,0.8987,0.8907,0.886,0.8804,0.8709,0.861,0.851,0.841,0.831,0.821,0.811,0.801,0.791,0.781,0.771,0.761,0.751,0.741,0.731,0.721,0.711,0.701,0.691,0.681,0.671,0.661,0.651,0.641,0.631,0.621,0.611,0.601,0.591,0.581,0.571,0.561,0.551,0.541,0.531,0.521,0.511,0.501,0.491,0.481,0.471,0.461,0.451,0.441,0.431,0.421,0.411,0.401,0.391,0.381,0.371,0.361,0.351,0.341,0.331,0.321,0.311,0.301,0.291,0.281,0.271,0.261,0.251,0.241,0.231,0.221,0.211,0.201,0.191,0.181,0.171,0.161,0.151,0.141,0.131,0.121,0.111,0.101,0.091,0.081,0.071,0.061,0.051,0.041,0.031,0.021,0.011,0.001]
vy=[0.6736,0.4818,0.394,0.2251,0.2157,0.2108,0.2288,0.2261,0.2035,0.1906,0.2279,0.2251,0.2023,0.1894,0.2267,0.224,0.2012,0.1883,0.2256,0.2229,0.2,0.1871,0.2244,0.2217,0.1843,0.1714,0.222,0.2193,0.1819,0.169,0.2206,0.2179,0.1805,0.1676,0.2192,0.2165,0.1791,0.1662,0.2178,0.2151,0.1777,0.1648,0.2164,0.2137,0.1763,0.1634,0.215,0.2123,0.1749,0.162,0.2136,0.2109,0.1735,0.1606,0.2122,0.2095,0.1721,0.1592,0.2108,0.2081,0.1707,0.1578,0.2094,0.2067,0.1693,0.1564,0.208,0.2053,0.1679,0.155,0.2066,0.2039,0.1665,0.1536,0.2052,0.2025,0.1651,0.1522,0.2038,0.2011,0.1637,0.1508,0.2024,0.1997,0.1623,0.1494,0.201,0.1983,0.1609,0.148,0.1996,0.1969,0.1595,0.1466,0.1982,0.1955,0.1581,0.1452,0.1968,0.1941,0.1567,0.1438,0.1954,0.1927,0.1553,0.1424,0.194,0.1913,0.1539,0.141,0.1926,0.1899,0.1525,0.1396,0.1912,0.1885,0.1511,0.1382,0.1898,0.1871,0.1497,0.1368,0.1884,0.1857,0.1483,0.1354,0.187,0.1843,0.1469,0.134,0.1856,0.1829,0.1455,0.1326,0.1842,0.1815,0.1441,0.1312,0.1828,0.1801,0.1427,0.1298,0.1814,0.1787,0.1413,0.1284,0.18,0.1773,0.1399,0.127,0.1786,0.1759,0.1385,0.1256,0.1772,0.1745,0.1371,0.1242,0.1758,0.1731,0.1357,0.1228,0.1744,0.1717,0.1343,0.1214,0.173,0.1703,0.1329,0.12,0.1716,0.1689,0.1315,0.1186,0.1702,0.1675,0.1301,0.1172,0.1688,0.1661,0.1287,0.1158,0.1674,0.1647,0.1273,0.1144,0.166,0.1633,0.1259,0.113,0.1646,0.1619,0.1245,0.1116,0.1632,0.1605,0.1231,0.1102,0.1618,0.1591,0.1217,0.1088,0.1604,0.1577,0.1203,0.1074,0.159,0.1563,0.1189,0.106,0.1576,0.1549,0.1175,0.1046,0.1562,0.1535,0.1161,0.1032,0.1548,0.1521,0.1147,0.1018,0.1534,0.1507,0.1133,0.1004,0.152,0.1493,0.1119,0.99,0.1506,0.1479,0.1105,0.976,0.1492,0.1465,0.1091,0.962,0.1478,0.1451,0.1077,0.948,0.1464,0.1437,0.1063,0.934,0.145,0.1423,0.1049,0.92,0.1436,0.1409,0.1035,0.906,0.1422,0.1395,0.1021,0.892,0.1408,0.1381,0.1007,0.878,0.1394,0.1367,0.994,0.138,0.1353,0.98,0.1366,0.1339,0.966,0.1352,0.1325,0.952,0.1338,0.1311,0.938,0.1324,0.1297,0.924,0.131,0.1283,0.91,0.1296,0.1269,0.896,0.1282,0.1255,0.882,0.1268,0.1241,0.868,0.1254,0.1227,0.854,0.124,0.1213,0.84,0.1226,0.1199,0.826,0.1212,0.1185,0.812,0.1198,0.1171,0.798,0.1184,0.1157,0.784,0.117,0.1143,0.77,0.1156,0.1129,0.756,0.1142,0.1115,0.742,0.1128,0.1101,0.728,0.1114,0.1087,0.714,0.11,0.1073,0.7,0.1086,0.1059,0.686,0.1072,0.1045,0.672,0.1058,0.1031,0.658,0.1044,0.1017,0.644,0.103,0.1003,0.63,0.1016,0.0989,0.616,0.1002,0.0975,0.602,0.0988,0.0964,0.588,0.0974,0.095,0.574,0.0962,0.0938,0.56,0.0948,0.0924,0.546,0.0934,0.091,0.532,0.092,0.0896,0.518,0.0906,0.0882,0.504,0.0892,0.0868,0.49,0.0878,0.0854,0.476,0.0864,0.084,0.462,0.085,0.0826,0.448,0.0836,0.0812,0.434,0.0822,0.0798,0.42,0.0808,0.0784,0.406,0.0794,0.077,0.392,0.078,0.0756,0.378,0.0766,0.0742,0.364,0.0752,0.0728,0.35,0.0738,0.0714,0.336,0.0724,0.07,0.322,0.071,0.0686,0.308,0.0696,0.0672,0.294,0.0682,0.0658,0.28,0.0668,0.0644,0.266,0.0654,0.063,0.252,0.064,0.0616,0.238,0.0626,0.0602,0.224,0.0612,0.0588,0.21,0.06,0.0576,0.196,0.0586,0.0562,0.182,0.0572,0.0548,0.168,0.0558,0.0534,0.154,0.0544,0.052,0.14,0.053,0.0506,0.126,0.0516,0.0492,0.112,0.0502,0.0478,0.098,0.0488,0.0464,0.084,0.0474,0.045,0.07,0.046,0.0436,0.056,0.0446,0.0422,0.042,0.04,0.0376,0.036,0.034,0.032,0.03,0.028,0.026,0.024,0.022,0.02,0.018,0.016,0.014,0.012,0.01,0.008,0.006,0.004,0.002,0.001]

%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)
```

