
```
%tensorflow_version 1.x
```

```
☐➔ TensorFlow 1.x selected.
```

```
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import tensorflow as tf
```

```
np.random.seed(451)
```

```
mnist = tf.contrib.learn.datasets.load_dataset("mnist")
x_train = mnist.train.images # Returns np.array
y_train = np.asarray(mnist.train.labels, dtype=np.int32)
x_test = mnist.test.images # Returns np.array
y_test = np.asarray(mnist.test.labels, dtype=np.int32)
print(x_train.shape)
```

```
☐➔
```

WARNING:tensorflow:
The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
* <https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset>.
* <https://github.com/tensorflow/addons>
* <https://github.com/tensorflow/io> (for I/O related ops)
If you depend on functionality not listed there, please file an issue.

WARNING:tensorflow:From <ipython-input-5-9240476c1f9f>:1: load_dataset (from tensorflow
Instructions for updating:
Please use tf.data.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/contrib/learn/py
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/contrib/learn/py
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/contrib/learn/py
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/contrib/learn/py
Instructions for updating:
Please use urllib or similar directly.
Successfully downloaded train-images-idx3-ubyte.gz 9912422 bytes.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/contrib/learn/py
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST-data/train-images-idx3-ubyte.gz
Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/contrib/learn/py
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST-data/train-labels-idx1-ubyte.gz
Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.
Extracting MNIST-data/t10k-images-idx3-ubyte.gz
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.
Extracting MNIST-data/t10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/contrib/learn/py
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
(55000, 784)

```
x_train = x_train / 255.0  
x_test = x_test / 255.0
```

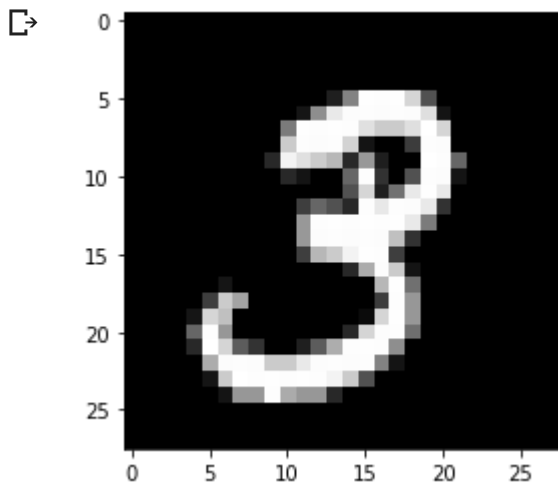
```
x_train_gray = x_train.reshape(-1,28,28,1)  
x_test_gray = x_test.reshape(-1,28,28,1)
```

```
from keras.utils.np_utils import to_categorical
```

```
y_train_cat = to_categorical(y_train)  
y_test_cat = to_categorical(y_test)
```

```
plt.imshow(x_train_gray[1,:,:,:0], cmap='gray')
```

```
plt.imshow(X_train_gray[2], cmap=gray,
plt.show()
```



```
np.random.seed(451)
```

```
import datetime
```

```
from keras.layers import Flatten, Activation, Conv2D, MaxPool2D, AvgPool2D, Dense, Dropout
from keras.optimizers import Adam, SGD
from keras.models import Sequential
import keras.backend as K
from keras.regularizers import l1,l2
from keras.callbacks import EarlyStopping, ModelCheckpoint, TensorBoard, ReduceLROnPlateau
from keras.models import model_from_json, Model
```

```
def build_tower(input_layer, features_nr, shape, tower_nr,
                dropout=False, normalization=False, regularization="l2", dropout_ratio=0.2
    #3x3 kernel tower
    tower = Conv2D(features_nr, (1,1), padding='same', activation='relu',
                    kernel_regularizer=regularization, name='tower_%d_%dx%da'%(tower_nr,
    tower = Conv2D(features_nr*2, shape, padding='same', activation='relu',
                    kernel_regularizer=regularization, name='tower_%d_%dx%db'%(tower_nr,
    #condidional dropout/normalization
    if dropout:
        tower = Dropout(dropout_ratio, name='tower_%d_%dx%ddrop'%(tower_nr, shape[0], shap
    if normalization:
        tower = BatchNormalization(name='tower_%d_%dx%dnorm'%(tower_nr, shape[0], shape[1]

    return tower
```

```
def build_simple_tower(input_layer, features_nr, shape, tower_nr,
                       dropout=False, normalization=False, regularization="l2", dropout_ratio=0.2
    #3x3 kernel tower
    tower = Conv2D(features_nr, shape, padding='same', activation='relu',
                    kernel_regularizer=regularization,
                    name='tower_simple_%d_%dx%db'%(tower_nr, shape[0], shape[1]))(input_lay
    #condidional dropout/normalization
    if dropout:
        tower = Dropout(dropout_ratio, name='tower_%d_%dx%ddrop'%(tower_nr, shape[0], shap
    if normalization:
        tower = BatchNormalization(name='tower_%d_%dx%dnorm'%(tower_nr, shape[0], shape[1]
```

```

return tower

def build_tower_subsample(input_layer, features_nr, shape, tower_nr,
                          dropout=False, normalization=False, regularization='l2', dropout_ratio=0.5):
    tower = build_tower(input_layer, features_nr, shape, tower_nr,
                        dropout, normalization, regularization, dropout_ratio)
    pool = MaxPooling2D((2,2), padding='same', name='tower_%d_2x2subsample'%(tower_nr))(tower)

    return pool

def build_simple_tower_subsample(input_layer, features_nr, shape, tower_nr,
                                 dropout=False, normalization=False, regularization='l2', dropout_ratio=0.5):
    tower = build_simple_tower(input_layer, features_nr, shape, tower_nr,
                              dropout, normalization, regularization, dropout_ratio)
    pool = MaxPooling2D((2,2), padding='same', name='tower_%d_2x2subsample'%(tower_nr))(tower)

    return pool

def build_dense(input_layer, neurons_nr, dense_nr,
                dropout=False, normalization=False, regularization='l2', dropout_ratio=0.5):
    dense = Dense(neurons_nr, kernel_regularizer=regularization,
                  name='dense_%d_%d'%(dense_nr, neurons_nr))(input_layer)

    if dropout:
        dense = Dropout(dropout_ratio, name='dense_%d_%ddrop'%(dense_nr, neurons_nr))(dense)
    if normalization:
        dense = BatchNormalization(name='dense_%d_%dnorm'%(dense_nr, neurons_nr))(dense)

    return dense

def build_inception_module(input_layer, features_nr, module_nr,
                           dropout=False, normalization=False, regularization='l2', dropout_ratio=0.5):
    #feature_nr is an array we'll use to build our layers
    #data is in the form: [1x1, 3x3 reduce, 3x3, 5x5 reduce, 5x5, pool proj]

    inception_1x1 = Conv2D(features_nr[0],1,1,border_mode='same',activation='relu',name='i%d_1x1'%(module_nr))
    inception_3x3_reduce = Conv2D(features_nr[1],1,1,border_mode='same',activation='relu',name='i%d_3x3_reduce'%(module_nr))
    inception_3x3 = Conv2D(features_nr[2],3,3,border_mode='same',activation='relu',name='i%d_3x3'%(module_nr))
    inception_5x5_reduce = Conv2D(features_nr[3],1,1,border_mode='same',activation='relu',name='i%d_5x5_reduce'%(module_nr))
    inception_5x5 = Conv2D(features_nr[4],5,5,border_mode='same',activation='relu',name='i%d_5x5'%(module_nr))
    inception_pool = MaxPooling2D(pool_size=(3,3),strides=(1,1),border_mode='same',name='i%d_pool'%(module_nr))
    inception_pool_proj = Conv2D(features_nr[5],1,1,border_mode='same',activation='relu',name='i%d_pool_proj'%(module_nr))
    inception_output = concatenate([inception_1x1,inception_3x3,inception_5x5,inception_pool_proj])

    if dropout:
        inception_output = Dropout(dropout_ratio, name='inception_%d/output_drop'%(module_nr))(inception_output)
    if normalization:
        inception_output = BatchNormalization(name='inception_%d/output_norm'%(module_nr))(inception_output)

    return inception_output

```

```

        inception_output = BatchNormalization(name='inception_%d/output_norm'%(module_nr)

    pooled = MaxPooling2D((2,2), padding='same', name='inception_%d_2x2subsample'%(module_

    return pooled

i='mnist-nrcrt7-'+datetime.datetime.now().strftime("%I:%M%p_%B-%d-%Y")

K.clear_session()

!mkdir -p models
!mkdir -p logs

a = EarlyStopping(monitor='val_loss', min_delta=0, patience=10, verbose=1, mode='auto')#wi
b = ModelCheckpoint(monitor='val_loss', filepath='./models/'+str(i)+'.hdf5', verbose=1, sa
c = TensorBoard(log_dir='./logs/'+str(i),
                write_grads=True,
                write_graph=True,
                write_images=True,
                batch_size=128)#saves a log file for tensorboard; remember to save differ

#we'll use this instead of decay
d = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=5, verbose=0, mode='auto',

callbacks=[a,b,c,d]

#-----model definition-----

use_norm = True
lrrate = 0.001

input_img = Input(shape = (28, 28, 1), name='input')

#conv_1 = Conv2D(1, (1,1), padding='same', activation='relu',
                # kernel_regularizer = regularization, name='conv_64x64x1_inception_in')(ir

#hopefully this will learn a good internal representation of the image channels
#conv_1 = Conv2D(1, (1,1), padding='same', activation='relu',
                #kernel_regularizer = regularization, name='conv_64x64x1_inception_in')(ir

inception_1 = build_inception_module(input_img, [64,96,128,16,32,32], 1, False, use_norm)

inception_2 = build_inception_module(inception_1, [128,128,192,32,96,64], 2, False, use_nc

inception_3 = build_inception_module(inception_2, [192,96,208,16,48,64], 3, False, use_nor

inception_4 = build_inception_module(inception_3, [160, 112, 224, 24, 64, 64], 4, False, u
#tower_3 = build_simple_tower(inception_2, 144, (3,3), 3, False, use_norm)
#tower_4 = build_simple_tower_subsample(tower_3, 144, (3,3), 4, False, use_norm)

#tower_5 = build_simple_tower(tower_4, 288, (3,3), 5, False, use_norm)
#tower_6 = build_simple_tower_subsample(tower_5, 288, (3,3), 6, False, use_norm)

#model top

def train():
    sess = tf.Session()
    saver = tf.train.Saver()
    # Create the model
    model = build_model()
    # Create the training data loader
    data_loader = DataLoader()
    # Create the validation data loader
    val_loader = ValidationLoader()
    # Create the training process
    training_process = TrainingProcess(model, data_loader, val_loader, sess, saver)
    # Start the training process
    training_process.train()

```

```

flat_pool = AveragePooling2D(pool_size=(2, 2), padding='valid')(inception_4)

flat = Flatten()(flat_pool)

dense_5 = build_dense(flat, 128, 1, True, use_norm)

dense_6 = build_dense(dense_5, 64, 2, True, use_norm)

out = Dense(10, activation='softmax')(dense_6)

model = Model(inputs = input_img, outputs = out)

#-----

model.compile(loss='binary_crossentropy',
              optimizer=Adam(lr=1e-4),
              metrics=['accuracy'])

model.summary()

model_json = model.to_json()
with open("./models/"+str(i)+".json", "w") as json_file:
    json_file.write(model_json)

print("Saved model to" + "../models/"+str(i)+".json")

```



```

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:75: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:77: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:79: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:81: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:83: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:85: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:87: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:75: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:77: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:79: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:81: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:83: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:85: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:87: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:75: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:77: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:79: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:81: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:83: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:85: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:87: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:75: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:77: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:79: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:81: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:83: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:85: UserWarning: Update
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:87: UserWarning: Update
Model: "model_1"

```

Layer (type)	Output Shape	Param #	Connected to
input (InputLayer)	(None, 28, 28, 1)	0	
inception_1_3x3_reduce (Conv2D)	(None, 28, 28, 96)	192	input[0][0]
inception_1_5x5_reduce (Conv2D)	(None, 28, 28, 16)	32	input[0][0]
inception_1_pool (MaxPooling2D)	(None, 28, 28, 1)	0	input[0][0]
inception_1_1x1 (Conv2D)	(None, 28, 28, 64)	128	input[0][0]
inception_1_3x3 (Conv2D)	(None, 28, 28, 128)	110720	inception_1_3x3_rec
inception_1_5x5 (Conv2D)	(None, 28, 28, 32)	12832	inception_1_5x5_rec
inception_1_pool_proj (Conv2D)	(None, 28, 28, 32)	64	inception_1_pool[0]
inception_1_output (Concatenat	(None, 28, 28, 256)	0	inception_1_1x1[0][inception_1_3x3[0][inception_1_5x5[0][inception_1_pool_pr
inception_1_output_norm (Batch	(None, 28, 28, 256)	1024	inception_1_output[
inception_1_2x2subsample (MaxPo	(None, 14, 14, 256)	0	inception_1_output_
inception_2_3x3_reduce (Conv2D)	(None, 14, 14, 128)	32896	inception_1_2x2subsa
inception_2_5x5_reduce (Conv2D)	(None, 14, 14, 32)	8224	inception_1_2x2subsa

inception_2_/pool (MaxPooling2D (None, 14, 14, 256)	0	inception_1_2x2subsample
inception_2_/1x1 (Conv2D) (None, 14, 14, 128)	32896	inception_1_2x2subsample
inception_2_/3x3 (Conv2D) (None, 14, 14, 192)	221376	inception_2_/3x3_reduce
inception_2_/5x5 (Conv2D) (None, 14, 14, 96)	76896	inception_2_/5x5_reduce
inception_2_/pool_proj (Conv2D) (None, 14, 14, 64)	16448	inception_2_/pool_proj
inception_2_/output (Concatenate) (None, 14, 14, 480)	0	inception_2_/1x1[0] inception_2_/3x3[0] inception_2_/5x5[0] inception_2_/pool_proj
inception_2_/output_norm (Batch Normalization) (None, 14, 14, 480)	1920	inception_2_/output_norm
inception_2_2x2subsample (MaxPooling2D) (None, 7, 7, 480)	0	inception_2_/output_norm
inception_3_/3x3_reduce (Conv2D) (None, 7, 7, 96)	46176	inception_2_2x2subsample
inception_3_/5x5_reduce (Conv2D) (None, 7, 7, 16)	7696	inception_2_2x2subsample
inception_3_/pool (MaxPooling2D) (None, 7, 7, 480)	0	inception_2_2x2subsample
inception_3_/1x1 (Conv2D) (None, 7, 7, 192)	92352	inception_2_2x2subsample
inception_3_/3x3 (Conv2D) (None, 7, 7, 208)	179920	inception_3_/3x3_reduce
inception_3_/5x5 (Conv2D) (None, 7, 7, 48)	19248	inception_3_/5x5_reduce
inception_3_/pool_proj (Conv2D) (None, 7, 7, 64)	30784	inception_3_/pool_proj
inception_3_/output (Concatenate) (None, 7, 7, 512)	0	inception_3_/1x1[0] inception_3_/3x3[0] inception_3_/5x5[0] inception_3_/pool_proj
inception_3_/output_norm (Batch Normalization) (None, 7, 7, 512)	2048	inception_3_/output_norm
inception_3_2x2subsample (MaxPooling2D) (None, 4, 4, 512)	0	inception_3_/output_norm
inception_4_/3x3_reduce (Conv2D) (None, 4, 4, 112)	57456	inception_3_2x2subsample
inception_4_/5x5_reduce (Conv2D) (None, 4, 4, 24)	12312	inception_3_2x2subsample
inception_4_/pool (MaxPooling2D) (None, 4, 4, 512)	0	inception_3_2x2subsample
inception_4_/1x1 (Conv2D) (None, 4, 4, 160)	82080	inception_3_2x2subsample
inception_4_/3x3 (Conv2D) (None, 4, 4, 224)	226016	inception_4_/3x3_reduce
inception_4_/5x5 (Conv2D) (None, 4, 4, 64)	38464	inception_4_/5x5_reduce
inception_4_/pool_proj (Conv2D) (None, 4, 4, 64)	32832	inception_4_/pool_proj
inception_4_/output (Concatenate) (None, 4, 4, 512)	0	inception_4_/1x1[0] inception_4_/3x3[0] inception_4_/5x5[0] inception_4_/pool_proj
inception_4_/output_norm (Batch Normalization) (None, 4, 4, 512)	2048	inception_4_/output_norm

inception_4_2x2subsample (MaxPo	(None, 2, 2, 512)	0	inception_4_2x2subsample
average_pooling2d_1 (AveragePoo	(None, 1, 1, 512)	0	inception_4_2x2subsample
flatten_1 (Flatten)	(None, 512)	0	average_pooling2d_1[0][0]
dense_1_128 (Dense)	(None, 128)	65664	flatten_1[0][0]
dense_1_128drop (Dropout)	(None, 128)	0	dense_1_128[0][0]
dense_1_128norm (BatchNormaliza	(None, 128)	512	dense_1_128drop[0][0]
dense_2_64 (Dense)	(None, 64)	8256	dense_1_128norm[0][0]
dense_2_64drop (Dropout)	(None, 64)	0	dense_2_64[0][0]
dense_2_64norm (BatchNormalizat	(None, 64)	256	dense_2_64drop[0][0]
dense_1 (Dense)	(None, 10)	650	dense_2_64norm[0][0]
=====			
Total params: 1,420,418			
Trainable params: 1,416,514			
Non-trainable params: 3,904			
Saved model to ../models/mnist-nrcrt7-06:48PM_May-04-2020.json			

```
import tensorflow as tf

with tf.device('/gpu:0'):
    model.fit(x_train_gray, y_train_cat, batch_size=128, epochs=100, validation_split=0.2, validation_data=(x_test_gray, y_test_cat))

result = model.evaluate(x_test_gray, y_test_cat)

print("Accuracy on test set: ",result[1]*100,"%")
```



```
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/python/ops/math_
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/tensorfl

Train on 44000 samples, validate on 11000 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks/tensor

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks/tensor

Epoch 1/100
44000/44000 [=====] - 29s 651us/step - loss: 0.6890 - accuracy: 0.0000

Epoch 00001: val_loss improved from inf to 0.54148, saving model to ./models/mnist-nr
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks/tensor

Epoch 2/100
44000/44000 [=====] - 20s 452us/step - loss: 0.1189 - accuracy: 0.0000

Epoch 00002: val_loss improved from 0.54148 to 0.37925, saving model to ./models/mnist
Epoch 3/100
44000/44000 [=====] - 20s 450us/step - loss: 0.0709 - accuracy: 0.0000

Epoch 00003: val_loss improved from 0.37925 to 0.29338, saving model to ./models/mnist
Epoch 4/100
44000/44000 [=====] - 20s 450us/step - loss: 0.0545 - accuracy: 0.0000

Epoch 00004: val_loss improved from 0.29338 to 0.13788, saving model to ./models/mnist
Epoch 5/100
44000/44000 [=====] - 20s 452us/step - loss: 0.0485 - accuracy: 0.0000

Epoch 00005: val_loss did not improve from 0.13788
Epoch 6/100
44000/44000 [=====] - 20s 445us/step - loss: 0.0443 - accuracy: 0.0000

Epoch 00006: val_loss improved from 0.13788 to 0.12291, saving model to ./models/mnist
Epoch 7/100
44000/44000 [=====] - 20s 445us/step - loss: 0.0456 - accuracy: 0.0000

Epoch 00007: val_loss did not improve from 0.12291
Epoch 8/100
44000/44000 [=====] - 20s 448us/step - loss: 0.0427 - accuracy: 0.0000

Epoch 00008: val_loss did not improve from 0.12291
Epoch 9/100
44000/44000 [=====] - 20s 444us/step - loss: 0.0381 - accuracy: 0.0000

Epoch 00009: val_loss improved from 0.12291 to 0.12233, saving model to ./models/mnist
Epoch 10/100
44000/44000 [=====] - 20s 445us/step - loss: 0.0353 - accuracy: 0.0000

Epoch 00010: val_loss did not improve from 0.12233
Epoch 11/100
44000/44000 [=====] - 19s 440us/step - loss: 0.0339 - accuracy: 0.0000

Epoch 00011: val_loss did not improve from 0.12233
Epoch 12/100
44000/44000 [=====] - 19s 439us/step - loss: 0.0340 - accuracy: 0.0000

Epoch 00012: val_loss improved from 0.12233 to 0.05556, saving model to ./models/mnist
Epoch 13/100
```

44000/44000 [=====] - 19s 437us/step - loss: 0.0322 - accuracy: 0.9511

Epoch 00013: val_loss improved from 0.05556 to 0.04186, saving model to ./models/mnist_epoch13.h5
Epoch 14/100

44000/44000 [=====] - 19s 440us/step - loss: 0.0317 - accuracy: 0.9522

Epoch 00014: val_loss did not improve from 0.04186
Epoch 15/100

44000/44000 [=====] - 19s 438us/step - loss: 0.0299 - accuracy: 0.9533

Epoch 00015: val_loss improved from 0.04186 to 0.03473, saving model to ./models/mnist_epoch15.h5
Epoch 16/100

44000/44000 [=====] - 19s 437us/step - loss: 0.0298 - accuracy: 0.9544

Epoch 00016: val_loss did not improve from 0.03473
Epoch 17/100

44000/44000 [=====] - 19s 437us/step - loss: 0.0310 - accuracy: 0.9533

Epoch 00017: val_loss did not improve from 0.03473
Epoch 18/100

44000/44000 [=====] - 19s 437us/step - loss: 0.0278 - accuracy: 0.9555

Epoch 00018: val_loss did not improve from 0.03473
Epoch 19/100

44000/44000 [=====] - 19s 435us/step - loss: 0.0286 - accuracy: 0.9544

Epoch 00019: val_loss did not improve from 0.03473
Epoch 20/100

44000/44000 [=====] - 19s 434us/step - loss: 0.0272 - accuracy: 0.9566

Epoch 00020: val_loss did not improve from 0.03473
Epoch 21/100

44000/44000 [=====] - 19s 435us/step - loss: 0.0202 - accuracy: 0.9588

Epoch 00021: val_loss improved from 0.03473 to 0.02030, saving model to ./models/mnist_epoch21.h5
Epoch 22/100

44000/44000 [=====] - 19s 436us/step - loss: 0.0155 - accuracy: 0.9610

Epoch 00022: val_loss improved from 0.02030 to 0.01851, saving model to ./models/mnist_epoch22.h5
Epoch 23/100

44000/44000 [=====] - 19s 434us/step - loss: 0.0134 - accuracy: 0.9632

Epoch 00023: val_loss improved from 0.01851 to 0.01604, saving model to ./models/mnist_epoch23.h5
Epoch 24/100

44000/44000 [=====] - 19s 434us/step - loss: 0.0122 - accuracy: 0.9654

Epoch 00024: val_loss improved from 0.01604 to 0.01561, saving model to ./models/mnist_epoch24.h5
Epoch 25/100

44000/44000 [=====] - 19s 433us/step - loss: 0.0113 - accuracy: 0.9676

Epoch 00025: val_loss improved from 0.01561 to 0.01502, saving model to ./models/mnist_epoch25.h5
Epoch 26/100

44000/44000 [=====] - 19s 434us/step - loss: 0.0106 - accuracy: 0.9698

Epoch 00026: val_loss did not improve from 0.01502
Epoch 27/100

44000/44000 [=====] - 19s 434us/step - loss: 0.0100 - accuracy: 0.9720

Epoch 00027: val_loss improved from 0.01502 to 0.01396, saving model to ./models/mnist_epoch27.h5
Epoch 28/100

44000/44000 [=====] - 19s 435us/step - loss: 0.0095 - accuracy: 0.9742

Epoch 00028: val_loss improved from 0.01396 to 0.01358, saving model to ./models/mnis
Epoch 29/100
44000/44000 [=====] - 19s 433us/step - loss: 0.0094 - accuracy: 0.9906

Epoch 00029: val_loss did not improve from 0.01358
Epoch 30/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0090 - accuracy: 0.9910

Epoch 00030: val_loss improved from 0.01358 to 0.01323, saving model to ./models/mnis
Epoch 31/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0086 - accuracy: 0.9914

Epoch 00031: val_loss did not improve from 0.01323
Epoch 32/100
44000/44000 [=====] - 19s 434us/step - loss: 0.0087 - accuracy: 0.9913

Epoch 00032: val_loss improved from 0.01323 to 0.01314, saving model to ./models/mnis
Epoch 33/100
44000/44000 [=====] - 19s 435us/step - loss: 0.0080 - accuracy: 0.9920

Epoch 00033: val_loss did not improve from 0.01314
Epoch 34/100
44000/44000 [=====] - 19s 435us/step - loss: 0.0077 - accuracy: 0.9923

Epoch 00034: val_loss improved from 0.01314 to 0.01222, saving model to ./models/mnis
Epoch 35/100
44000/44000 [=====] - 19s 438us/step - loss: 0.0076 - accuracy: 0.9924

Epoch 00035: val_loss did not improve from 0.01222
Epoch 36/100
44000/44000 [=====] - 19s 437us/step - loss: 0.0075 - accuracy: 0.9925

Epoch 00036: val_loss did not improve from 0.01222
Epoch 37/100
44000/44000 [=====] - 19s 437us/step - loss: 0.0071 - accuracy: 0.9929

Epoch 00037: val_loss did not improve from 0.01222
Epoch 38/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0068 - accuracy: 0.9932

Epoch 00038: val_loss did not improve from 0.01222
Epoch 39/100
44000/44000 [=====] - 19s 435us/step - loss: 0.0069 - accuracy: 0.9931

Epoch 00039: val_loss did not improve from 0.01222
Epoch 40/100
44000/44000 [=====] - 19s 435us/step - loss: 0.0065 - accuracy: 0.9935

Epoch 00040: val_loss improved from 0.01222 to 0.01165, saving model to ./models/mnis
Epoch 41/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0062 - accuracy: 0.9938

Epoch 00041: val_loss improved from 0.01165 to 0.01138, saving model to ./models/mnis
Epoch 42/100
44000/44000 [=====] - 19s 438us/step - loss: 0.0061 - accuracy: 0.9939

Epoch 00042: val_loss improved from 0.01138 to 0.01110, saving model to ./models/mnis
Epoch 43/100
44000/44000 [=====] - 19s 435us/step - loss: 0.0060 - accuracy: 0.9940

Epoch 00043: val_loss improved from 0.01110 to 0.01108, saving model to ./models/mnis

Epoch 44/100
44000/44000 [=====] - 19s 438us/step - loss: 0.0059 - accuracy: 0.9941

Epoch 00044: val_loss improved from 0.01108 to 0.01093, saving model to ./models/mnist_44.h5

Epoch 45/100
44000/44000 [=====] - 19s 435us/step - loss: 0.0058 - accuracy: 0.9942

Epoch 00045: val_loss did not improve from 0.01093

Epoch 46/100
44000/44000 [=====] - 19s 437us/step - loss: 0.0057 - accuracy: 0.9943

Epoch 00046: val_loss improved from 0.01093 to 0.01071, saving model to ./models/mnist_46.h5

Epoch 47/100
44000/44000 [=====] - 19s 438us/step - loss: 0.0057 - accuracy: 0.9943

Epoch 00047: val_loss improved from 0.01071 to 0.01069, saving model to ./models/mnist_47.h5

Epoch 48/100
44000/44000 [=====] - 19s 437us/step - loss: 0.0056 - accuracy: 0.9944

Epoch 00048: val_loss improved from 0.01069 to 0.01053, saving model to ./models/mnist_48.h5

Epoch 49/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0055 - accuracy: 0.9945

Epoch 00049: val_loss did not improve from 0.01053

Epoch 50/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0055 - accuracy: 0.9945

Epoch 00050: val_loss did not improve from 0.01053

Epoch 51/100
44000/44000 [=====] - 19s 435us/step - loss: 0.0054 - accuracy: 0.9946

Epoch 00051: val_loss improved from 0.01053 to 0.01051, saving model to ./models/mnist_51.h5

Epoch 52/100
44000/44000 [=====] - 19s 438us/step - loss: 0.0053 - accuracy: 0.9947

Epoch 00052: val_loss improved from 0.01051 to 0.01021, saving model to ./models/mnist_52.h5

Epoch 53/100
44000/44000 [=====] - 19s 435us/step - loss: 0.0053 - accuracy: 0.9947

Epoch 00053: val_loss improved from 0.01021 to 0.01021, saving model to ./models/mnist_53.h5

Epoch 54/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0052 - accuracy: 0.9948

Epoch 00054: val_loss improved from 0.01021 to 0.00999, saving model to ./models/mnist_54.h5

Epoch 55/100
44000/44000 [=====] - 19s 437us/step - loss: 0.0052 - accuracy: 0.9948

Epoch 00055: val_loss did not improve from 0.00999

Epoch 56/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0051 - accuracy: 0.9949

Epoch 00056: val_loss did not improve from 0.00999

Epoch 57/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0051 - accuracy: 0.9949

Epoch 00057: val_loss improved from 0.00999 to 0.00992, saving model to ./models/mnist_57.h5

Epoch 58/100
44000/44000 [=====] - 19s 434us/step - loss: 0.0051 - accuracy: 0.9949

Epoch 00058: val_loss did not improve from 0.00992

Epoch 59/100
44000/44000 [=====] - 19s 433us/step - loss: 0.0050 - accuracy: 0.9950

Epoch 00059: val_loss improved from 0.00992 to 0.00987, saving model to ./models/mnis
Epoch 60/100
44000/44000 [=====] - 19s 433us/step - loss: 0.0050 - accuracy: 0.9999

Epoch 00060: val_loss did not improve from 0.00987
Epoch 61/100
44000/44000 [=====] - 19s 433us/step - loss: 0.0050 - accuracy: 0.9999

Epoch 00061: val_loss did not improve from 0.00987
Epoch 62/100
44000/44000 [=====] - 19s 433us/step - loss: 0.0049 - accuracy: 0.9999

Epoch 00062: val_loss improved from 0.00987 to 0.00987, saving model to ./models/mnis
Epoch 63/100
44000/44000 [=====] - 19s 436us/step - loss: 0.0049 - accuracy: 0.9999

Epoch 00063: val_loss did not improve from 0.00987
Epoch 64/100
44000/44000 [=====] - 19s 434us/step - loss: 0.0048 - accuracy: 0.9999

Epoch 00064: val_loss did not improve from 0.00987
Epoch 65/100
44000/44000 [=====] - 19s 434us/step - loss: 0.0048 - accuracy: 0.9999

Epoch 00065: val_loss did not improve from 0.00987
Epoch 66/100
44000/44000 [=====] - 19s 433us/step - loss: 0.0048 - accuracy: 0.9999

Epoch 00066: val_loss improved from 0.00987 to 0.00981, saving model to ./models/mnis
Epoch 67/100
44000/44000 [=====] - 19s 441us/step - loss: 0.0048 - accuracy: 0.9999

Epoch 00067: val_loss improved from 0.00981 to 0.00981, saving model to ./models/mnis
Epoch 68/100
44000/44000 [=====] - 20s 443us/step - loss: 0.0048 - accuracy: 0.9999

Epoch 00068: val_loss improved from 0.00981 to 0.00981, saving model to ./models/mnis
Epoch 69/100
44000/44000 [=====] - 20s 448us/step - loss: 0.0048 - accuracy: 0.9999

Epoch 00069: val_loss improved from 0.00981 to 0.00973, saving model to ./models/mnis
Epoch 70/100
44000/44000 [=====] - 19s 440us/step - loss: 0.0048 - accuracy: 0.9999

Epoch 00070: val_loss did not improve from 0.00973
Epoch 71/100
44000/44000 [=====] - 19s 442us/step - loss: 0.0047 - accuracy: 0.9999

Epoch 00071: val_loss did not improve from 0.00973
Epoch 72/100
44000/44000 [=====] - 19s 442us/step - loss: 0.0048 - accuracy: 0.9999

Epoch 00072: val_loss did not improve from 0.00973
Epoch 73/100
44000/44000 [=====] - 20s 444us/step - loss: 0.0047 - accuracy: 0.9999

Epoch 00073: val_loss improved from 0.00973 to 0.00971, saving model to ./models/mnis
Epoch 74/100
44000/44000 [=====] - 19s 441us/step - loss: 0.0047 - accuracy: 0.9999

```

Epoch 00074: val_loss did not improve from 0.00971
Epoch 75/100
44000/44000 [=====] - 19s 439us/step - loss: 0.0047 - accuracy: 0.9987298846244812

Epoch 00075: val_loss did not improve from 0.00971
Epoch 76/100
44000/44000 [=====] - 19s 441us/step - loss: 0.0047 - accuracy: 0.9987298846244812

Epoch 00076: val_loss did not improve from 0.00971
Epoch 77/100
44000/44000 [=====] - 19s 441us/step - loss: 0.0047 - accuracy: 0.9987298846244812

Epoch 00077: val_loss did not improve from 0.00971
Epoch 78/100
44000/44000 [=====] - 19s 439us/step - loss: 0.0047 - accuracy: 0.9987298846244812

Epoch 00078: val_loss did not improve from 0.00971
Epoch 79/100
44000/44000 [=====] - 19s 442us/step - loss: 0.0047 - accuracy: 0.9987298846244812

Epoch 00079: val_loss did not improve from 0.00971
Epoch 80/100
44000/44000 [=====] - 19s 440us/step - loss: 0.0047 - accuracy: 0.9987298846244812

Epoch 00080: val_loss did not improve from 0.00971
Epoch 81/100
44000/44000 [=====] - 19s 440us/step - loss: 0.0047 - accuracy: 0.9987298846244812

Epoch 00081: val_loss did not improve from 0.00971
Epoch 82/100
44000/44000 [=====] - 19s 438us/step - loss: 0.0047 - accuracy: 0.9987298846244812

Epoch 00082: val_loss did not improve from 0.00971
Epoch 83/100
44000/44000 [=====] - 19s 439us/step - loss: 0.0047 - accuracy: 0.9987298846244812

Epoch 00083: val_loss did not improve from 0.00971
Epoch 00083: early stopping
10000/10000 [=====] - 3s 263us/step
Accuracy on test set: 99.87298846244812 %

```

```
model.load_weights('./models/mnist-nrcrt7-06:48PM_May-04-2020.hdf5')
```

```
result = model.evaluate(x_test_gray, y_test_cat)
```

```
print(result)
```

```

[> 10000/10000 [=====] - 2s 245us/step
[0.00949306582286954, 0.9987298846244812]

```

```
with tf.Session() as sess:
```

```
    writer = tf.summary.FileWriter('./graphs',sess.graph)
```

```
%load_ext tensorboard
```

```
%tensorboard --logdir ./graphs
```

➤ The tensorboard extension is already loaded. To reload it, use:
%reload_ext tensorboard
Reusing TensorBoard on port 6006 (pid 456), started 0:01:11 ago. (Use '!kill 456' to

TensorBoard

GRAPHS

Search nodes. Regexes sup...



Fit to Screen



Download PNG

Run

(1)

.

Tag

(1)

Default

Upload

Choose File



Graph



Conceptual Graph



Profile



Trace inputs



Show health pills

Color



Structure

✓ Close legend.

Graph (* = expandable)



Namespace* [?](#)



OpNode [?](#)



Unconnected series* [?](#)



Connected series* [?](#)



Constant [?](#)



Summary [?](#)



Dataflow edge [?](#)



Control dependency edge [?](#)



Reference edge [?](#)

