Introduction to Tensorflow

Introduction to tensorflow and it’s high level frameworks tflearn and keras

Jan Zikeš

Knowledge check

- Programming in Python, C++ or Lua?
- Experience with scikit?
- Experience with Caffe, Torch, Theano or Tensorflow?

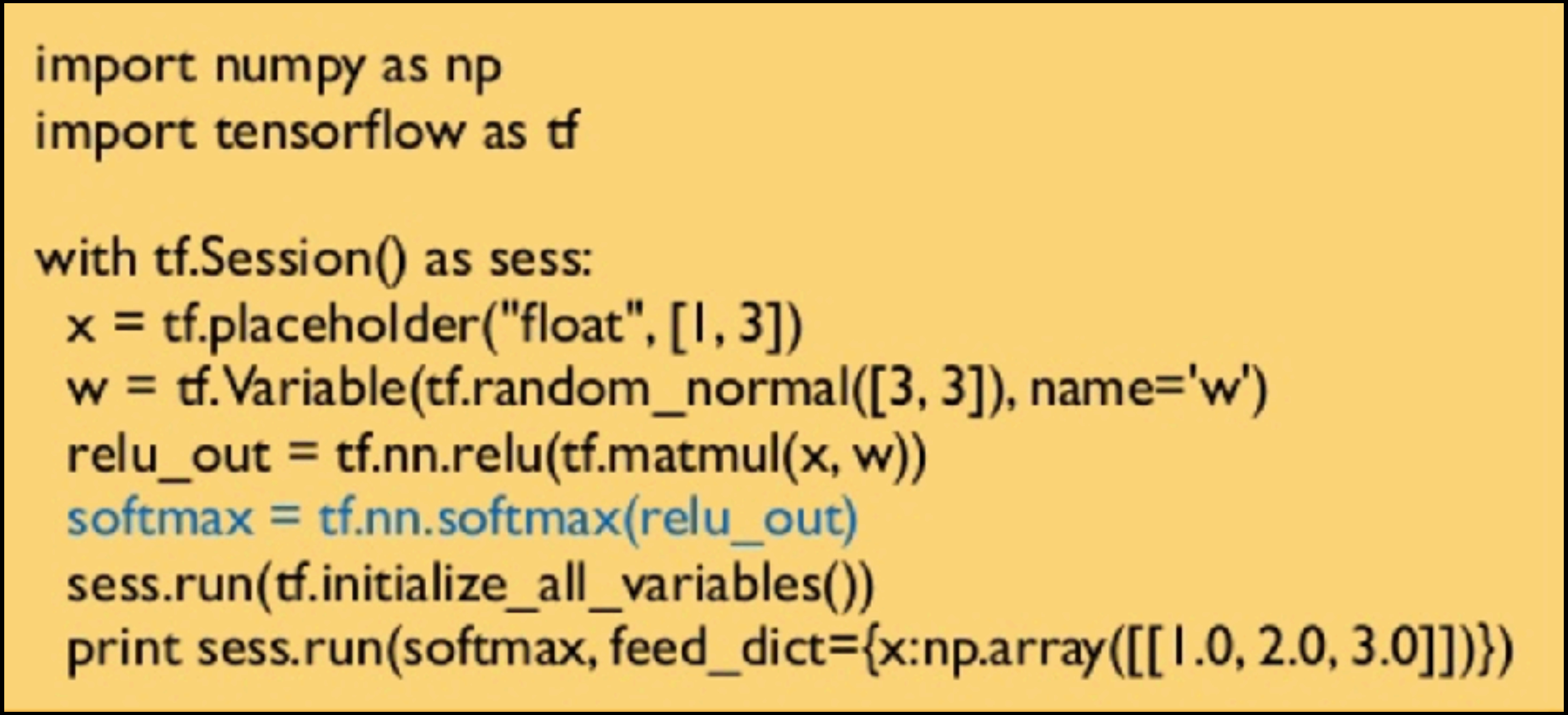
Basics

- Tensorflow is not a silver bullet it’s just another framework for Deep Learning
- The most frequently used is Python API so let’s stick with that
- The most active open-source DL framework
- Google’s pushing this as an industry standard

Installation

- Good Luck :), but it’s usually better than Caffe
- Nvidia drivers, Cuda 8 for tf>=0.11, Cuda 7.5 for older (it also depends on your GPU), CuDNN
- tf==0.11 is still a bit problematic, safe is to go with 0.10 and Cuda 7.5 (problem with Pascal architecture)

Let’s go by example



```
import numpy as np
import tensorflow as tf

with tf.Session() as sess:
  x = tf.placeholder("float", [1, 3])
  w = tf.Variable(tf.random_normal([3, 3]), name='w')
  relu_out = tf.nn.relu(tf.matmul(x, w))
  softmax = tf.nn.softmax(relu_out)
  sess.run(tf.initialize_all_variables())
  print sess.run(softmax, feed_dict={x:np.array([[1.0, 2.0, 3.0]])})
```

Practical examples

- Linear regression https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/2_BasicModels/linear_regression.py

- MLP https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/3_NeuralNetworks/multilayer_perceptron.py

- ConvNet: https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/3_NeuralNetworks/convolutional_network.py

Where to start

- https://www.tensorflow.org/versions/r0.11/api_docs/python/nn.html
- Available layers: convolution, softmax, pooling, etc.

Practical Tensorflow advices

- It’s still pretty new, so APIs are changing from time to time
    - 0.10 -> 0.11 (even some examples in repo doesn’t work)
- Learn tensorflow first, then continue with Keras ;)
- Don’t expect using tf to be really easy

Tflearn

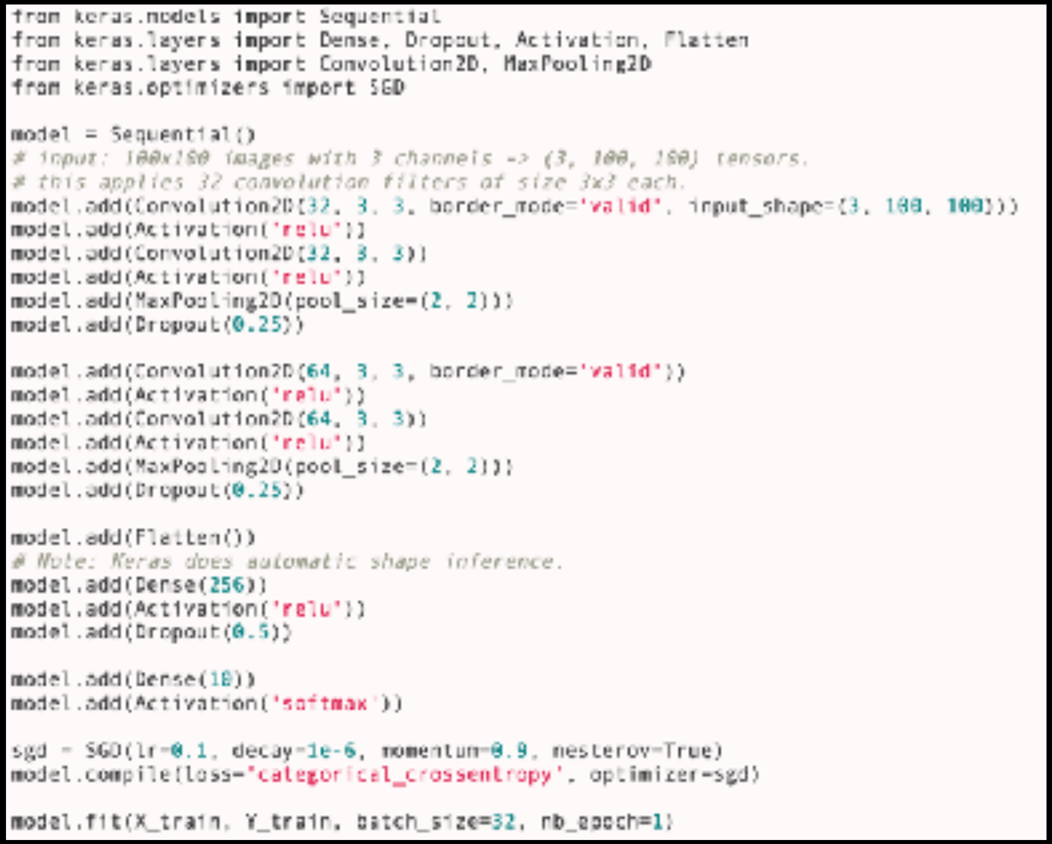
- High level framework on top of tensorflow
- When you’ll use tensorflow a lot, you’ll most likely end up with something like your own Tflearn :)

Keras

- The second most active DL open source repo
- Easy to use with lot of cool features
- Supports both, tensorflow and theano as it’s backend
- Still not stable enough for production (model saving/loading frequent problems)

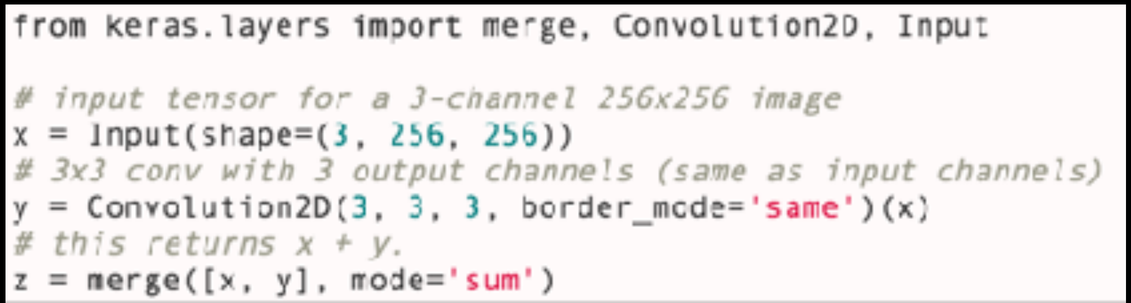
Keras example

- Sequential API



```
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D, MaxPooling2D
from keras.optimizers import SGD

model = Sequential()
# input: 100x100 images with 3 channels -> (3, 100, 100) tensors.
# this applies 32 convolution filters of size 3x3 each.
model.add(Convolution2D(32, 3, 3, border_mode='valid', input_shape=(3, 100, 100)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='valid'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
# Note: Keras does automatic shape inference.
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dropout(0.5))

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

sgd = SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)

model.fit(X_train, Y_train, batch_size=32, nb_epoch=1)
```

- Functional API



```
from keras.layers import merge, Convolution2D, Input

# input tensor for a 3-channel 256x256 image
x = Input(shape=(3, 256, 256))
# 3x3 conv with 3 output channels (same as input channels)
y = Convolution2D(3, 3, 3, border_mode='same')(x)
# this returns x + y.
z = merge([x, y], mode='sum')
```

Question?