Keras

An API, built to make it easier

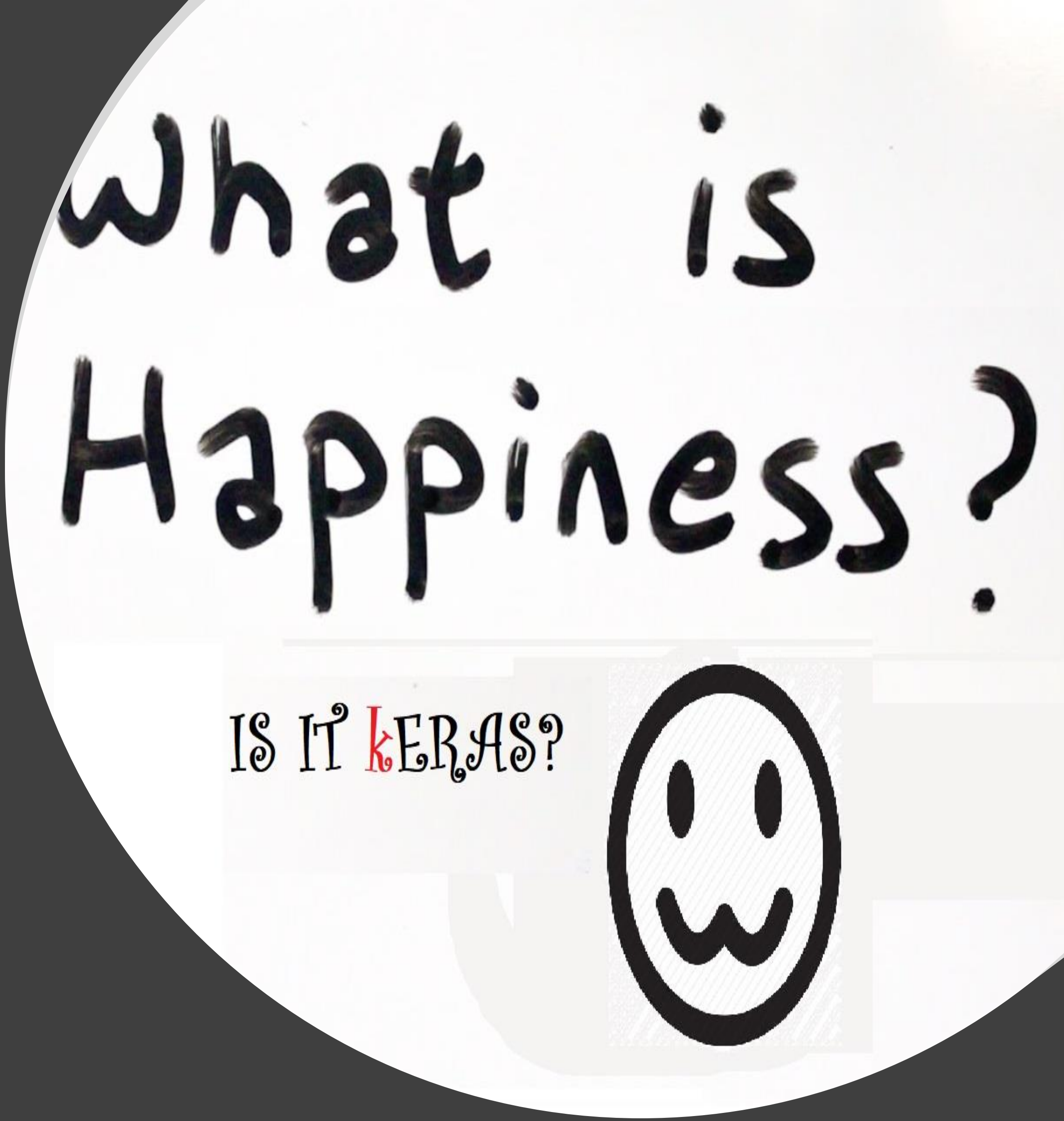
Keras

- What is Keras
- Why use Keras
- Keras Vs. TF
- Playing with Keras

What is Keras

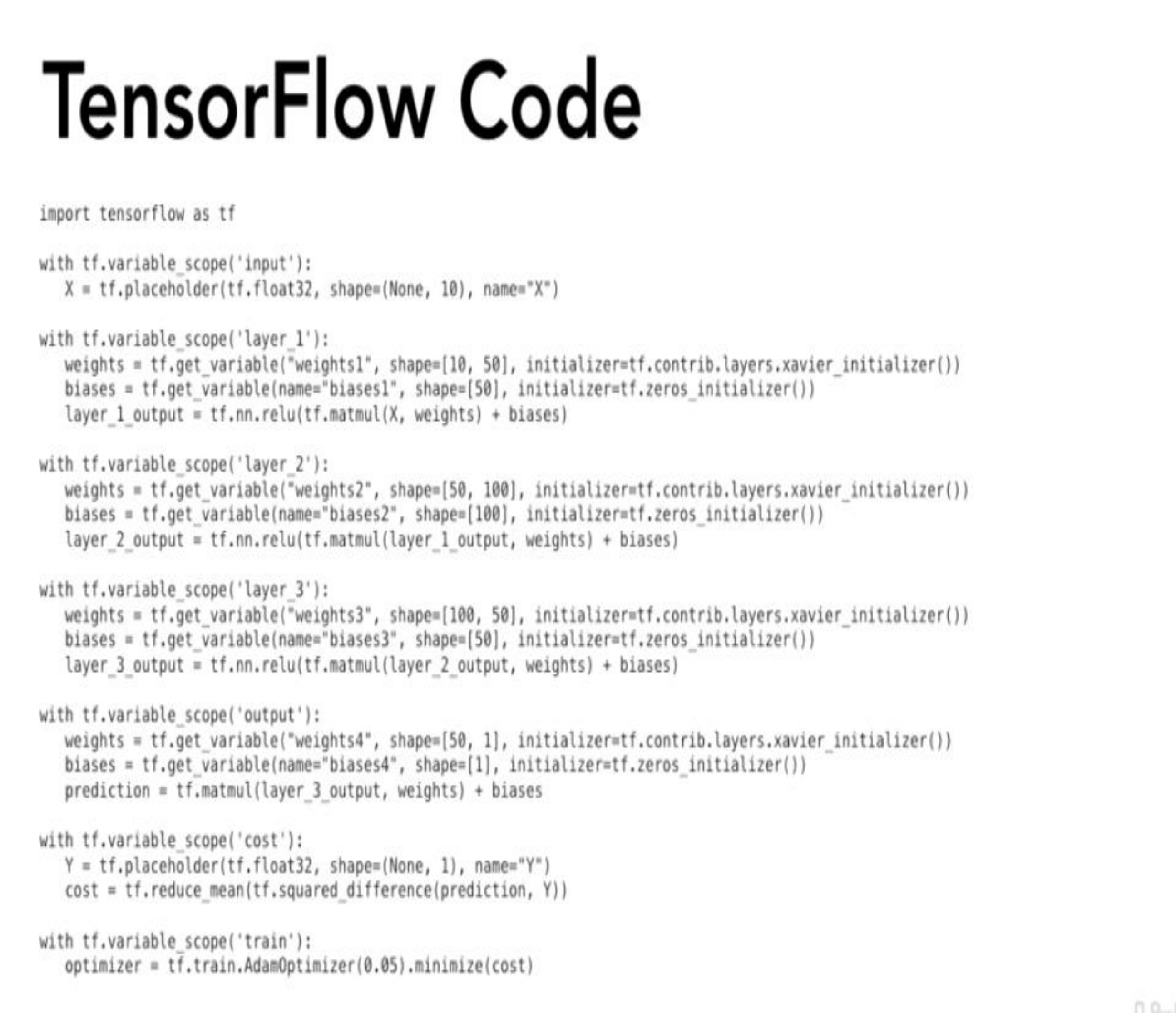
Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation.

“Being able to go from idea to result with the least possible delay is key to doing good research.”(keras.io)



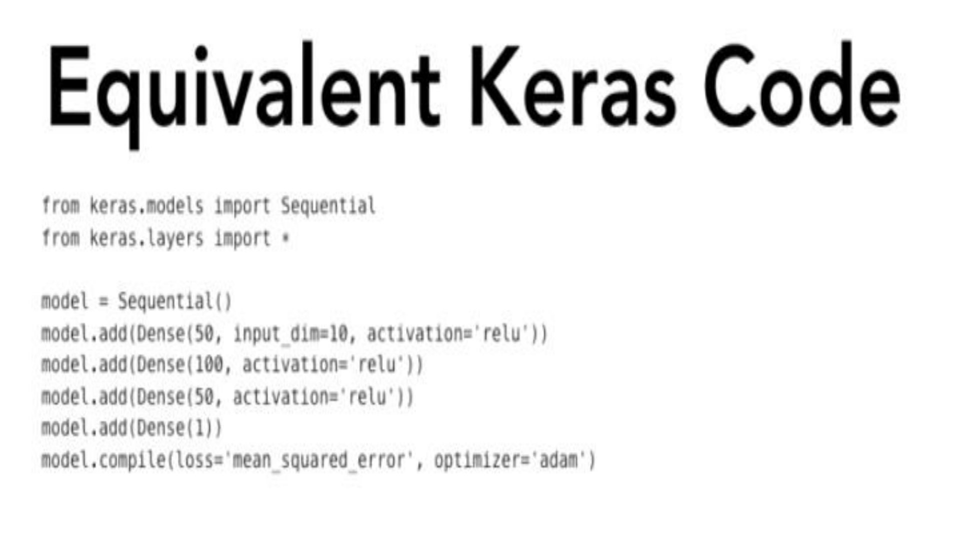
Why Keras

- Rapid prototyping
- Easily switching between APIs
- Industry best practices are already built in
- The default settings are a big help
- Prebuilt systems
- GPU acceleration



TensorFlow Code

```
import tensorflow as tf

with tf.variable_scope('input'):
   X = tf.placeholder(tf.float32, shape=(None, 10), name="X")

with tf.variable_scope('layer_1'):
   weights = tf.get_variable("weights1", shape=[10, 50], initializer=tf.contrib.layers.xavier_initializer())
   biases = tf.get_variable(name="biases1", shape=[50], initializer=tf.zeros_initializer())
   layer_1_output = tf.nn.relu(tf.matmul(X, weights) + biases)

with tf.variable_scope('layer_2'):
   weights = tf.get_variable("weights2", shape=[50, 100], initializer=tf.contrib.layers.xavier_initializer())
   biases = tf.get_variable(name="biases2", shape=[100], initializer=tf.zeros_initializer())
   layer_2_output = tf.nn.relu(tf.matmul(layer_1_output, weights) + biases)

with tf.variable_scope('layer_3'):
   weights = tf.get_variable("weights3", shape=[100, 50], initializer=tf.contrib.layers.xavier_initializer())
   biases = tf.get_variable(name="biases3", shape=[50], initializer=tf.zeros_initializer())
   layer_3_output = tf.nn.relu(tf.matmul(layer_2_output, weights) + biases)

with tf.variable_scope('output'):
   weights = tf.get_variable("weights4", shape=[50, 1], initializer=tf.contrib.layers.xavier_initializer())
   biases = tf.get_variable(name="biases4", shape=[1], initializer=tf.zeros_initializer())
   prediction = tf.matmul(layer_3_output, weights) + biases

with tf.variable_scope('cost'):
   Y = tf.placeholder(tf.float32, shape=(None, 1), name="Y")
   cost = tf.reduce_mean(tf.squared_difference(prediction, Y))

with tf.variable_scope('train'):
   optimizer = tf.train.AdamOptimizer(0.05).minimize(cost)
```



Equivalent Keras Code

```
from keras.models import Sequential
from keras.layers import *

model = Sequential()
model.add(Dense(50, input_dim=10, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(50, activation='relu'))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
```

Time for Koding

- Building a simple model
- Using ResNet50 prebuilt system
- Using Keras with tensor board

Photos Credits

- https://store.waitbutwhy.com/
- http://clipart.info/devil-emoji-png-transparent-icon-4689
- https://peru.com/epic/epic-mobile/facebook-messenger-no-solo-pacman-este-emoji-tambien-desaparecio-fotos-noticia-458421
- https://www.youtube.com/watch?v=BiD5AhFcNW8
- https://www.emojibase.com/emojilist/hooray
- https://tensorflow.org
- http://deathbattle.wikia.com/wiki/File:Vs.png
- https://keras.io

धन्यवाद

Thank you

谢谢

شُكْرَاً لَكُمْ

Terima kasih

Merci a vous

شکریہ