**Difference Between Keras and TensorFlow.Keras – 2019-07**

* **TensorFlow.Keras**
  + Version of Keras API implemented specifically for use with TensorFlow
  + Part of TensorFlow repo currently on 1.14.0
  + Installation
    - pip install tensorflow-gpu
      * Latest Version – 1.10.0
    - pip install tensorflow
    - Python 3.6 CPU-only
      * <https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow-1.14.0-cp36-cp36m-win_amd64.whl>
    - Python 3.6 GPU-only
      * <https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-1.14.0-cp36-cp36m-win_amd64.whl>
  + Sample Code
    - import tensorflow as tf
    - from tensorflow import keras
  + Sample Code 2
    - tensorflow==1.4.0 "from tensorflow.python import keras"
* **Keras**
  + One of the leading high-level neural networks APIs
  + User friendly, modular, composable, easy to extend
  + Currently compatibility is up to Python 3.6
    - Latest version with 3.6 is 2.2.4
  + tf.keras can run any Keras-compatible code, but keep in mind:
    - The tf.keras version in the latest TensorFlow release might not be the same as the latest keras version from PyPI. Check tf.keras.version.
    - When saving a model's weights, tf.keras defaults to the checkpoint format. Pass save\_format='h5' to use HDF5.
  + Standalone Keras is used to maintain framework-agnostic or use with another backend
  + Installation
    - Before installing keras, install TensorFlow or Theano
  + How to find backend
    - Keras will build a folder “.keras” where the keras.json file is located
    - %USERPROFILE%/.keras/keras.json
  + How to change backend
    - KERAS\_BACKEND=tensorflow python –c
    - Check
      * keras.backend.backend()

**Which one should you use?**

Keras as a library will still operate independently and separately from TensorFlow so there is a possibility that the two will diverge in the future; however, given that Google officially supports both Keras and TensorFlow, that divergence seems extremely unlikely.

The point is this:

* If you’re comfortable writing code using pure Keras, go for it, and keep doing it.
* But if you find yourself working in TensorFlow, you should start leveraging the Keras API:
* It’s built right into TensorFlow
* It’s easier to use
* And when you need pure TensorFlow to implement a specific feature or functionality, it can be dropped right into your Keras model.

Example 1: Keras vs. TensorFlow

Keras

# import the necessary packages

from keras.models import Sequential

from keras.layers.core import Dens

# define the 3072-1024-512-3 architecture using Keras

model = Sequential()

model.add(Dense(1024, input\_shape=(3072,), activation="sigmoid"))

model.add(Dense(512, activation="sigmoid"))

model.add(Dense(10, activation="softmax"))

TensorFlow

import tensorflow as tf

# define the 3072-1024-512-3 architecture using tf.keras

model = tf.keras.models.Sequential()

model.add(tf.keras.layers.Dense(1024, input\_shape=(3072,), activation="sigmoid"))

model.add(tf.keras.layers.Dense(512, activation="sigmoid"))

model.add(tf.keras.layers.Dense(10, activation="softmax"))