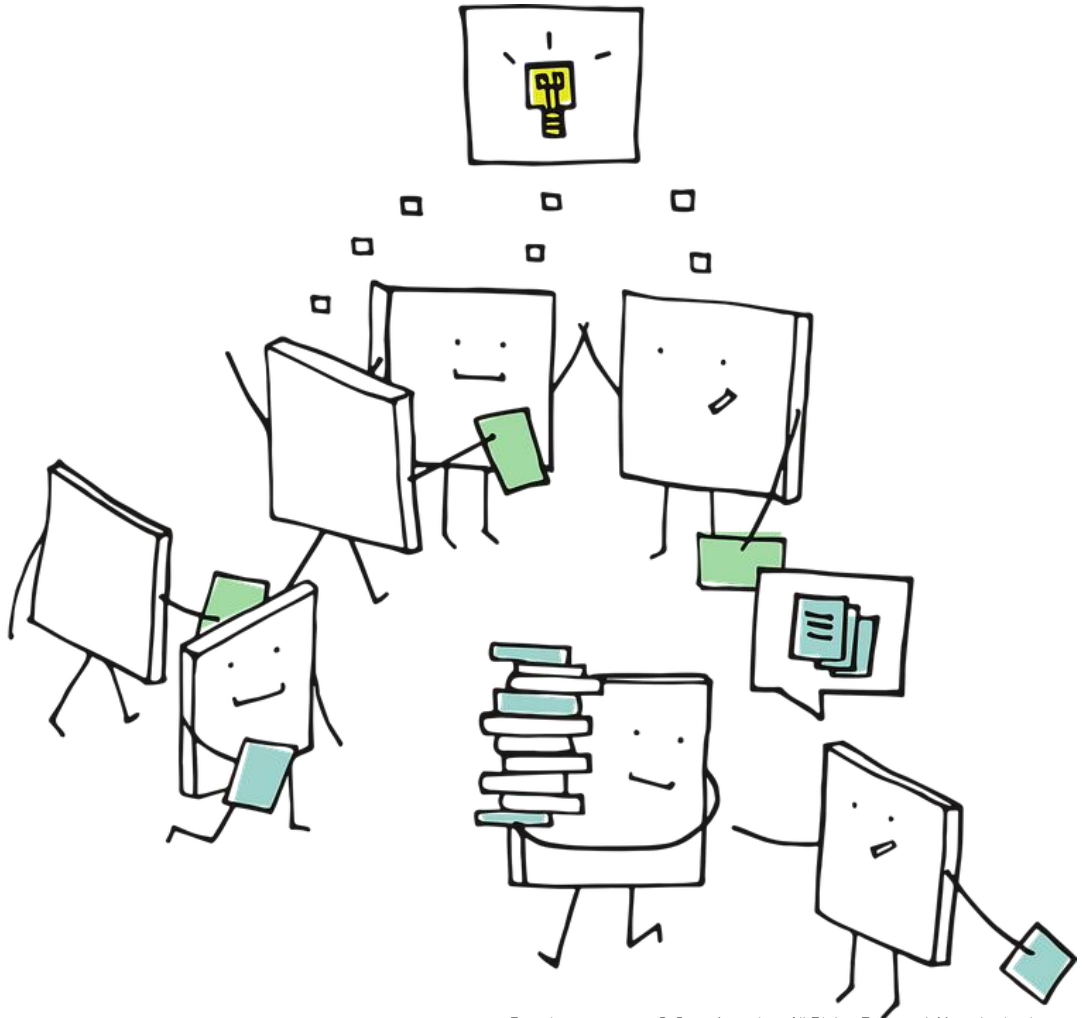


Neural Networks - I

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

TensorFlow can be tough to learn :(

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

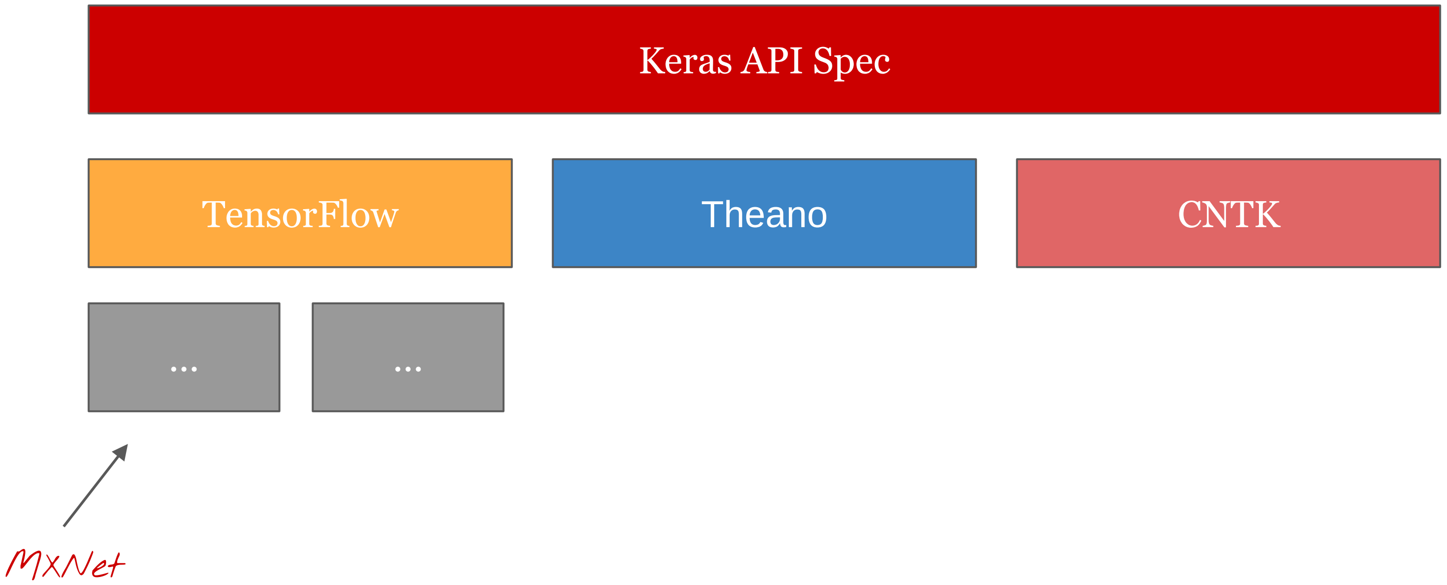
Keras

Simple

Minimal Code

Powerful

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

How do we simplify TensorFlow?

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Keras is now part of TensorFlow codebase

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

How can Keras help

Using Keras as part of TensorFlow

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Building model in TensorFlow Keras

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
#Declare Weights and Bias

w = tf.zeros(shape=(13,1))
b = tf.zeros(shape=(1))
```

```
# y = wx + b

def prediction(x, w, b):

    xw_matmul = tf.matmul(x, w)
    y = tf.add(xw_matmul, b)

return y
```

```
model = tf.keras.Sequential()

model.add(tf.keras.layers.Dense(1, input_shape=(13,))
```

TensorFlow

Keras

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
#Define Loss

def loss(y_actual, y_predicted):

    diff = y_actual - y_predicted
    sqr = tf.square(diff)
    avg = tf.reduce_mean(sqr)

    return avg
```

```
#Define Gradient Descent function

def train(x, y_actual, w, b, learning_rate=0.01):

    #Record math ops on 'tape' to calculate loss
    with tf.GradientTape() as t:

        t.watch([w,b])

        current_prediction = prediction(x, w, b)
        current_loss = loss(y_actual, current_prediction)

    #Calculate Gradients for Loss w.r.t Weights and Bias
    dw, db = t.gradient(current_loss,[w, b])

    #Update Weights and Bias
    w = w - learning_rate*dw
    b = b - learning_rate*db

    return w, b
```

```
model.compile(optimizer='sgd', loss='mse’)
```

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Training the Model

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
    #Training the model for 100 iterations

    for i in range(100):

                w, b = train(train_x, train_y, w, b)
                print('Current Loss on iteration', i, loss(train_y, prediction(train_x, w,
b)).numpy())
```

TensorFlow

Keras

```
model.fit(train_x, train_y, epochs=100)
```

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Where is my model saved?

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Saving Model

```
model.save(<file_name>)
```

Install h5py using pip

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Exercise

Build Boston Housing Predictor in TensorFlow Keras

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.