DL Dev Course: Week 03b

What are TF Estimators ?

- A high level abstraction API for writing TensorFlow models
- What Google wants to become the standard for writing most TF models
- They consist of a number of key parts and come in both premade and custom versions
- Take inspiration from SKLearn and from Keras

Flexible: High level APIs

Low-Level Python API

```
import numpy as np
import tensorflow as tf

W = tf.Variable([.3], tf.float32)
b = tf.Variable([-.3], tf.float32)
x = tf.placeholder(tf.float32)
linear_model = W * x + b
y = tf.placeholder(tf.float32)
loss = tf.reduce_sum(tf.square(linear_model - y))
optimizer = tf.train.GradientDescentOptimizer(0.01)
train = optimizer.minimize(loss)
x_train = [1,2,3,4]
y_train = [0,-1,-2,-3]
init = tf.global_variables_initializer()
sess = tf.Session()
sess.run(init) # reset values to wrong
for i in range(1000):
  sess.run(train, {x:x_train, y:y_train})
curr_W, curr_b, curr_loss  = sess.run([W, b, loss], {x:x_train, y:y_train})
```

High-Level Python API

```
import tensorflow as tf
import numpy as np

features = [tf.contrib.layers.real_valued_column("x", dimension=1)]
estimator = tf.contrib.learn.LinearRegressor(feature_columns=features)
x = np.array([1., 2., 3., 4.])
y = np.array([0., -1., -2., -3.])
input_fn = tf.contrib.learn.io.numpy_input_fn({"x":x}, y, batch_size=4,
                                              num_epochs=1000)
estimator.fit(input_fn=input_fn, steps=1000)
estimator.evaluate(input_fn=input_fn)
```

https://www.tensorflow.org/get_started/get_started

Flexible: High level APIs

Low-Level Python API

```
import numpy as np
import tensorflow as tf

W
b =
x =
line
y =
los
op
tra
x_t
y_t
init = tf.global_variables_initializer()
sess = tf.Session()
sess.run(init) # reset values to wrong
for i in range(1000):
  sess.run(train, {x:x_train, y:y_train})
curr_W, curr_b, curr_loss  = sess.run([W, b, loss], {x:x_train, y:y_train})
```

15 lines

High-Level Python API

```
import tensorflow as tf
import numpy as np

features = [tf.contrib
estimator = tf.contrib
x = np.array([1., 2.,
y = np.array([0., -1.,
input_fn = tf.contrib.

estimator.fit(input_fr
estimator.evaluate(i
```

7 lines

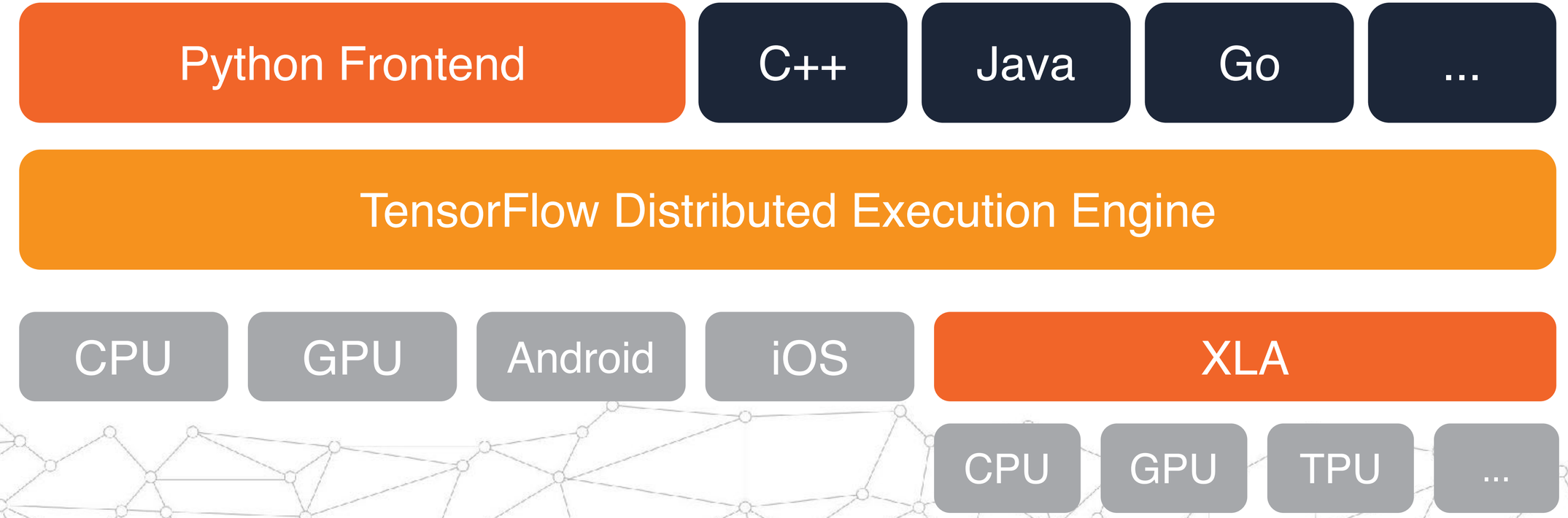
https://www.tensorflow.org/get_started/get_started

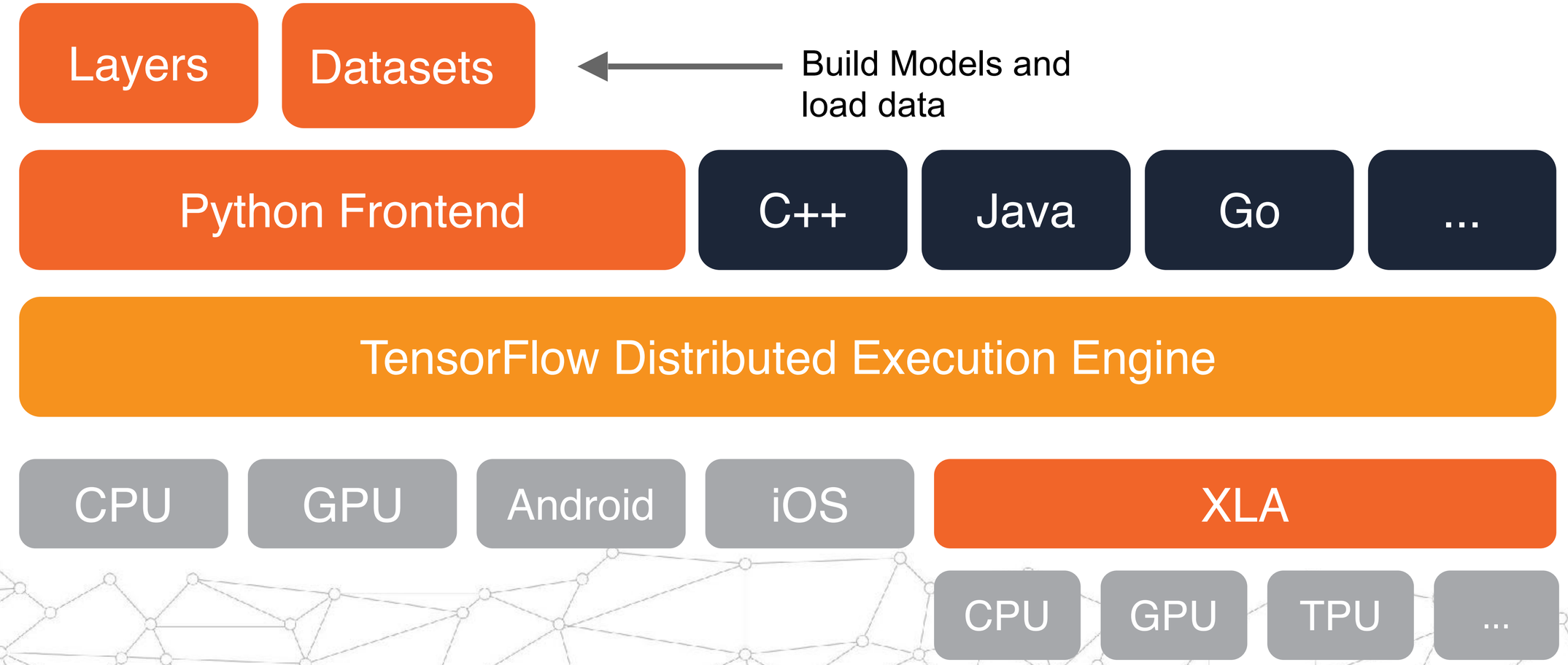
The Old Way 01

- feed_dict
- Train loop
- Low level ops
- Stuff all over the place
- Lots of lines of code

The Old Way 02

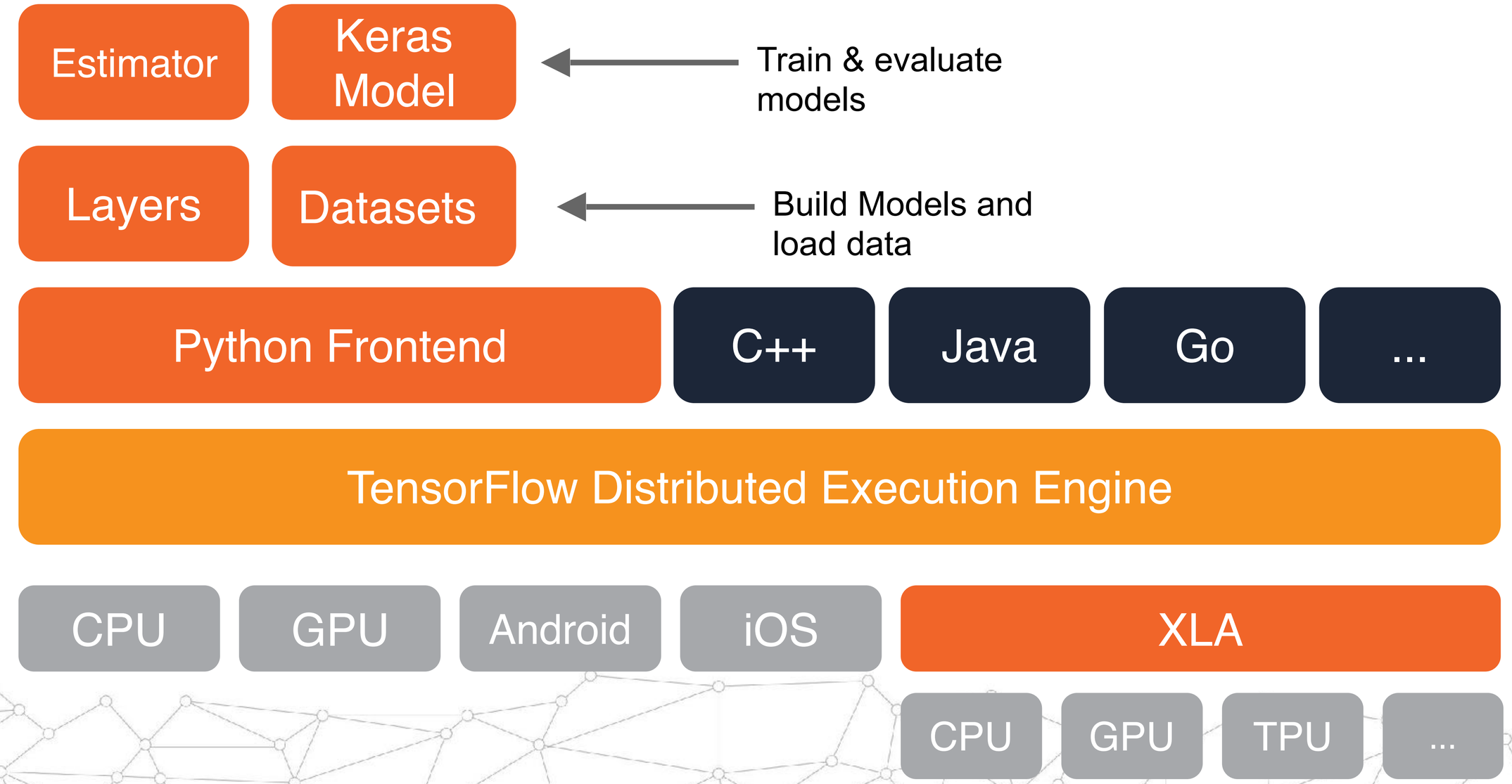
- Lots of Frameworks
- Slim
- TFLearn
- Sugar Tensor
- Etc. etc.

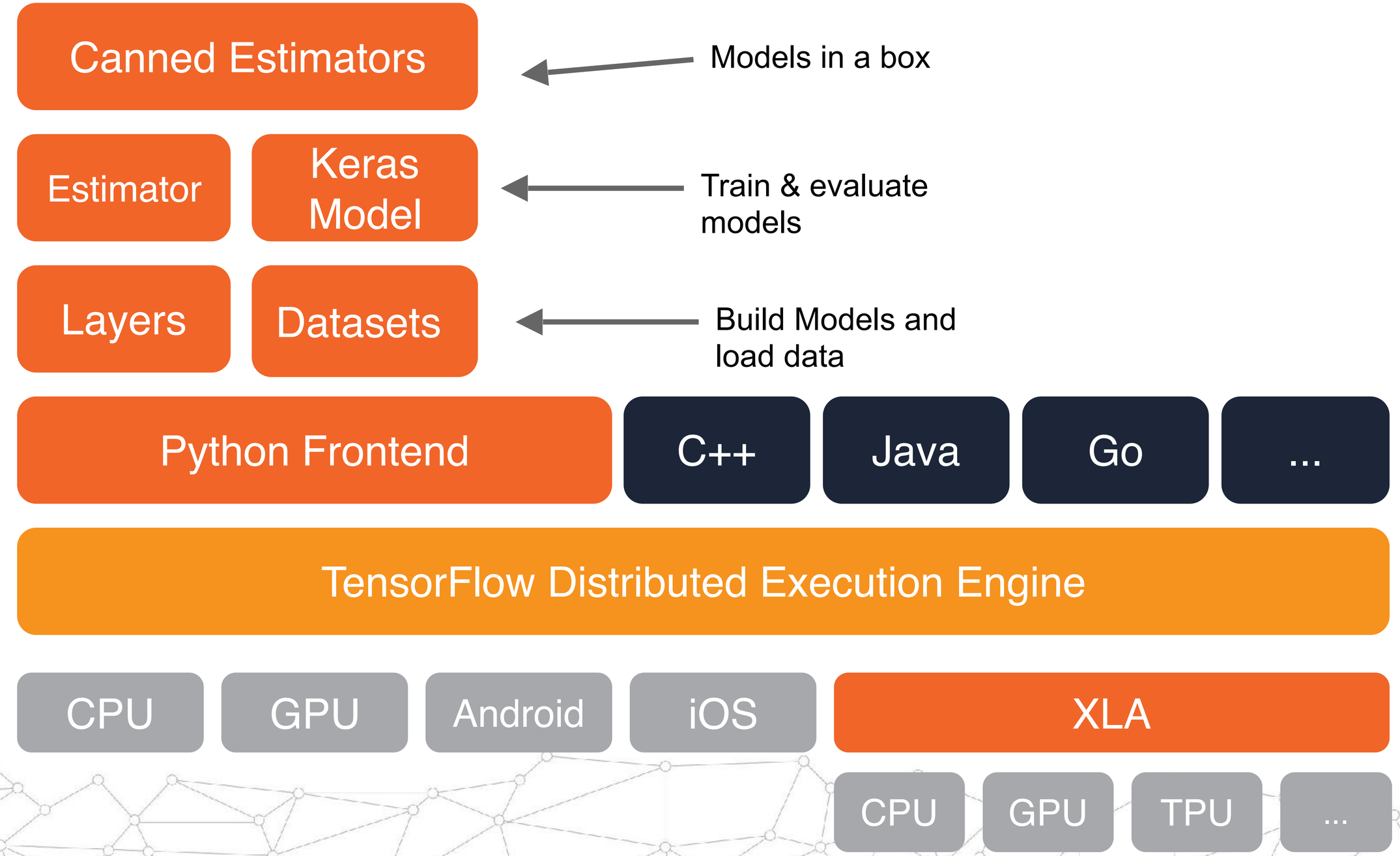

Python Frontend
C++
Java
Go
...
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
XLA
CPU
GPU
TPU
...


Layers
Datasets
Build Models and
load data
Python Frontend
C++
Java
Go
...
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
XLA
CPU
GPU
TPU
...

TF Layers

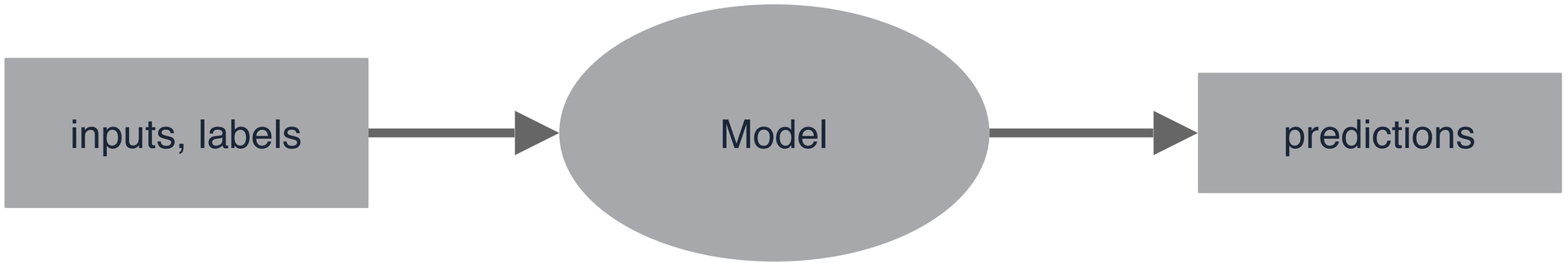
- Pre made layer that represent some of the more common types of NN operations
- tf.layers.conv2d(inputs=image_batch, filters=32, kernel_size=[3, 3], padding='same', activation=tf.nn.relu)
- tf.layers.dense(inputs=tensor_shape, units=128, activation=tf.nn.relu)
- They take inspiration from Keras

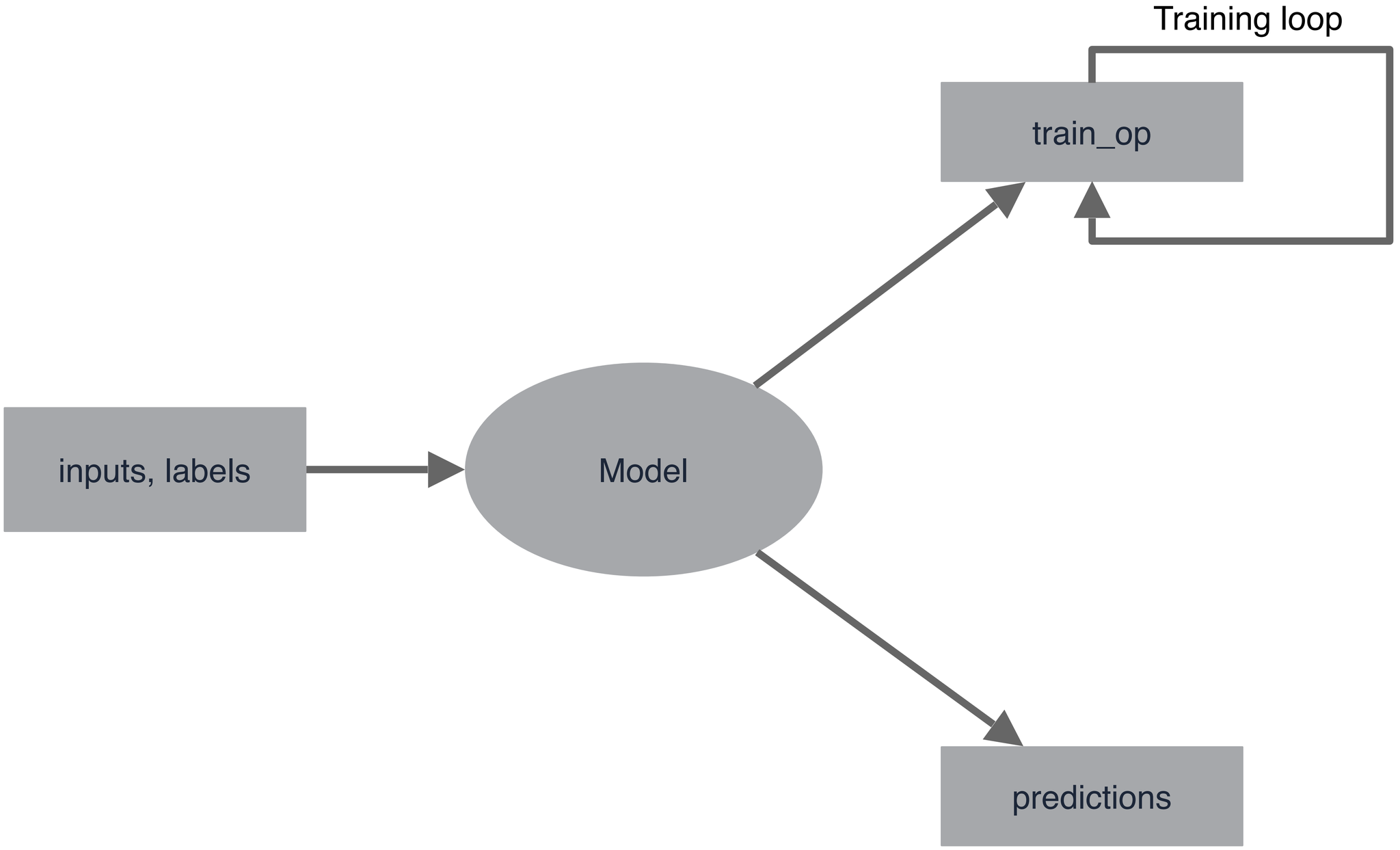

Estimator
Keras Model
Train & evaluate models
Layers
Datasets
Build Models and load data
Python Frontend
C++
Java
Go
...
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
XLA
CPU
GPU
TPU
...

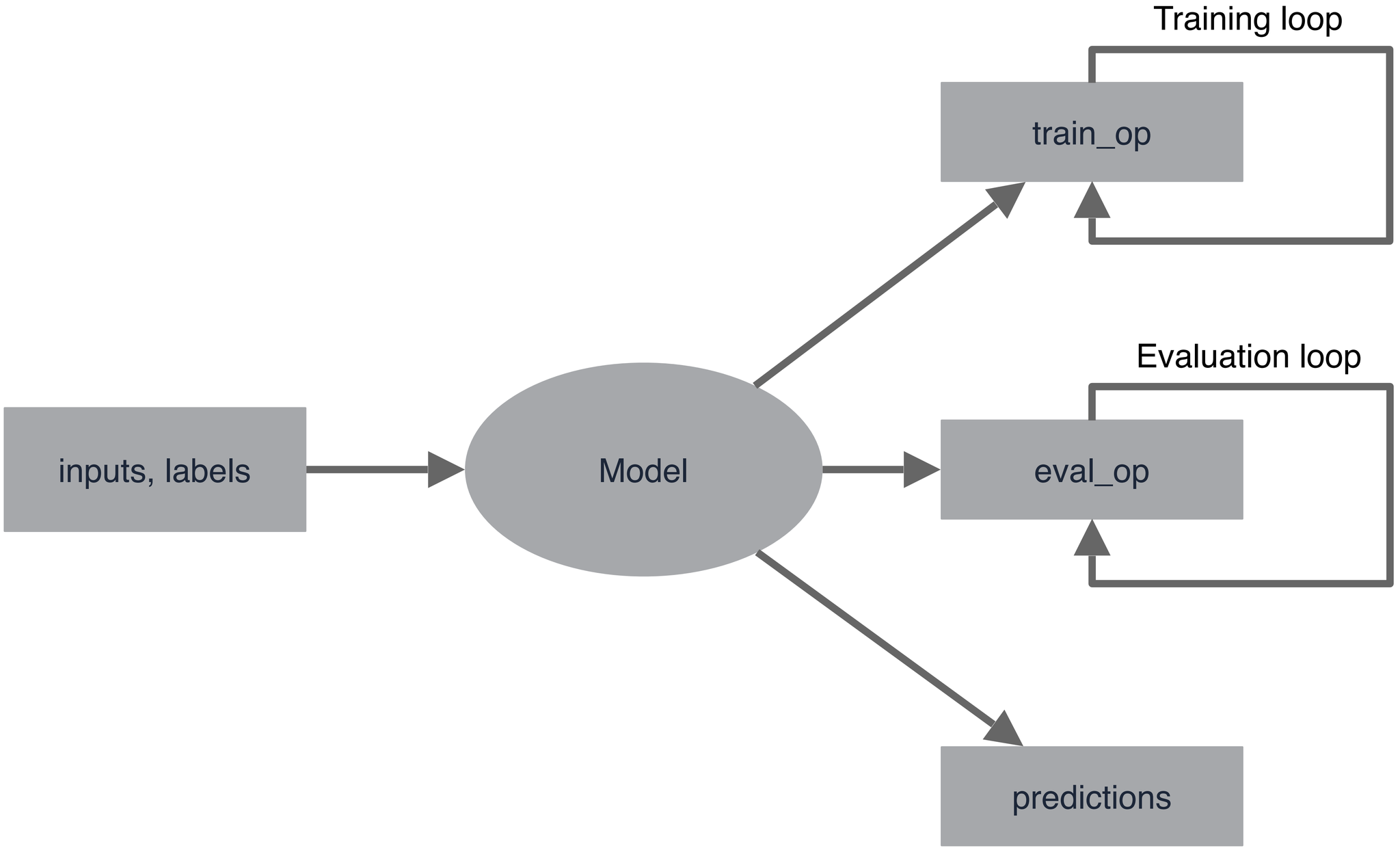

Canned Estimators
Models in a box
Estimator
Keras Model
Train & evaluate models
Layers
Datasets
Build Models and load data
Python Frontend
C++
Java
Go
...
TensorFlow Distributed Execution Engine
CPU
GPU
Android
iOS
XLA
CPU
GPU
TPU
...

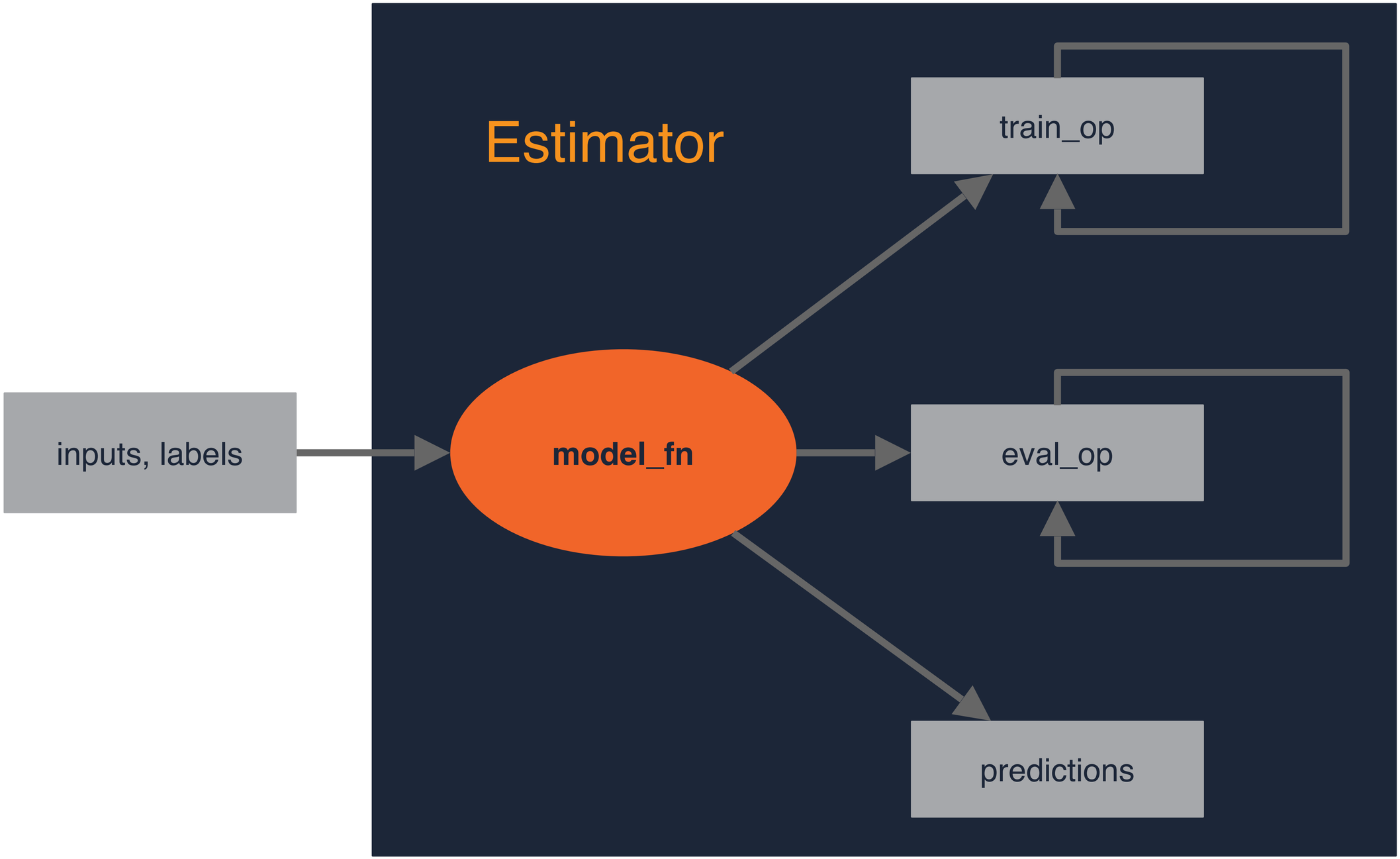
TF Estimators Parts

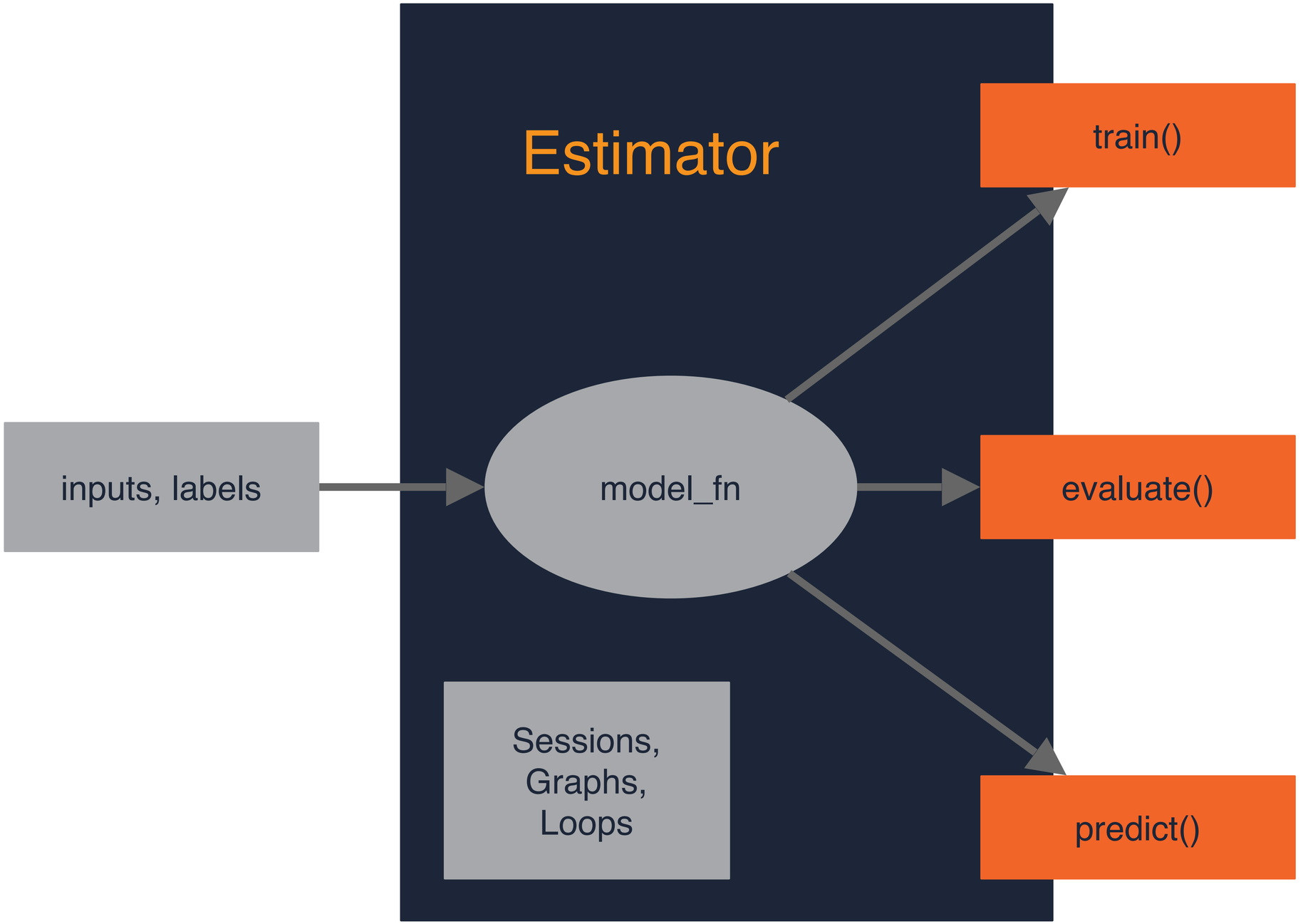
- input_fn
- network_fn
- Model_fn
- Layers
- Datasets
- The functions an Estimator runs

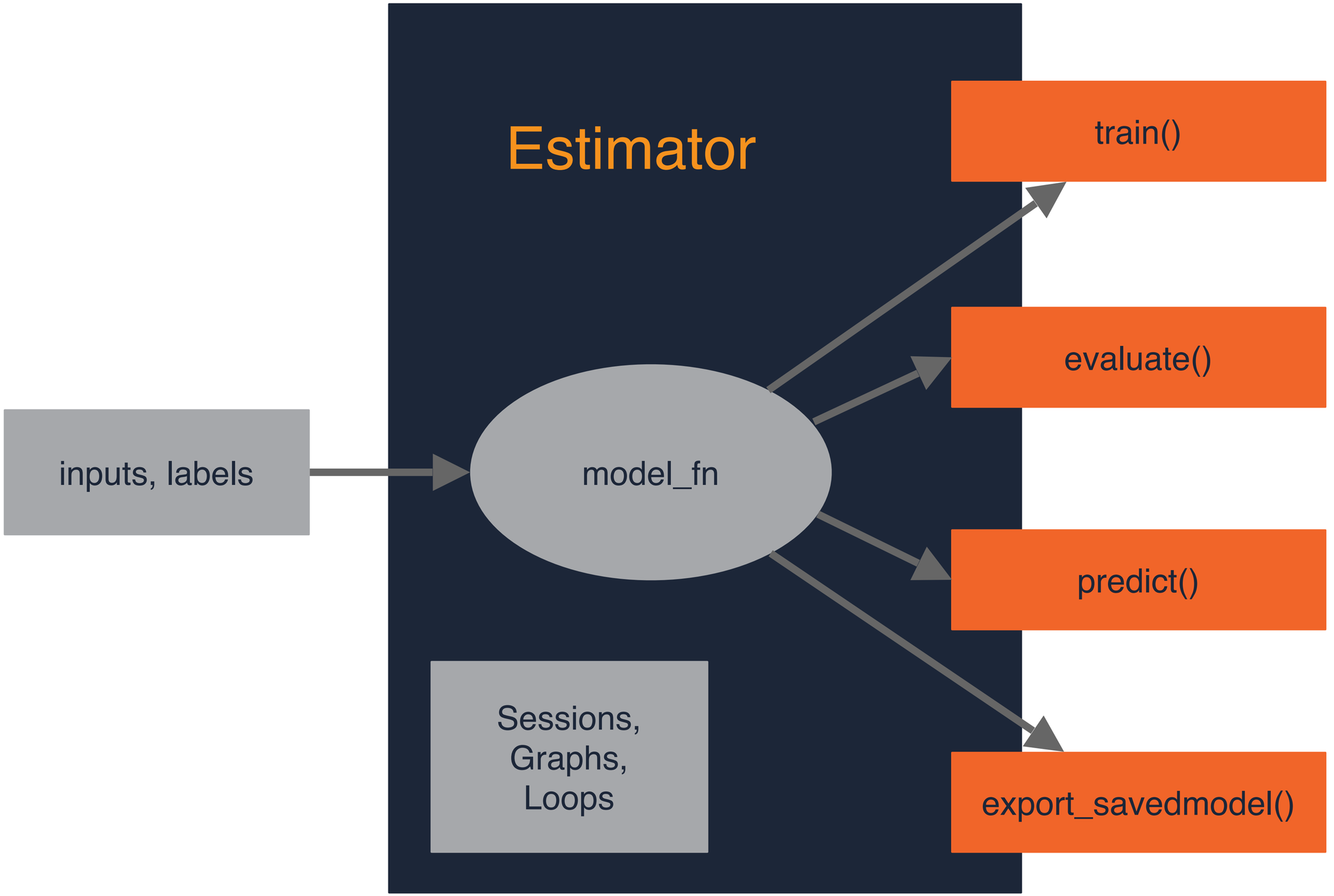

inputs, labels
Model
predictions


Training loop
train_op
inputs, labels
Model
predictions

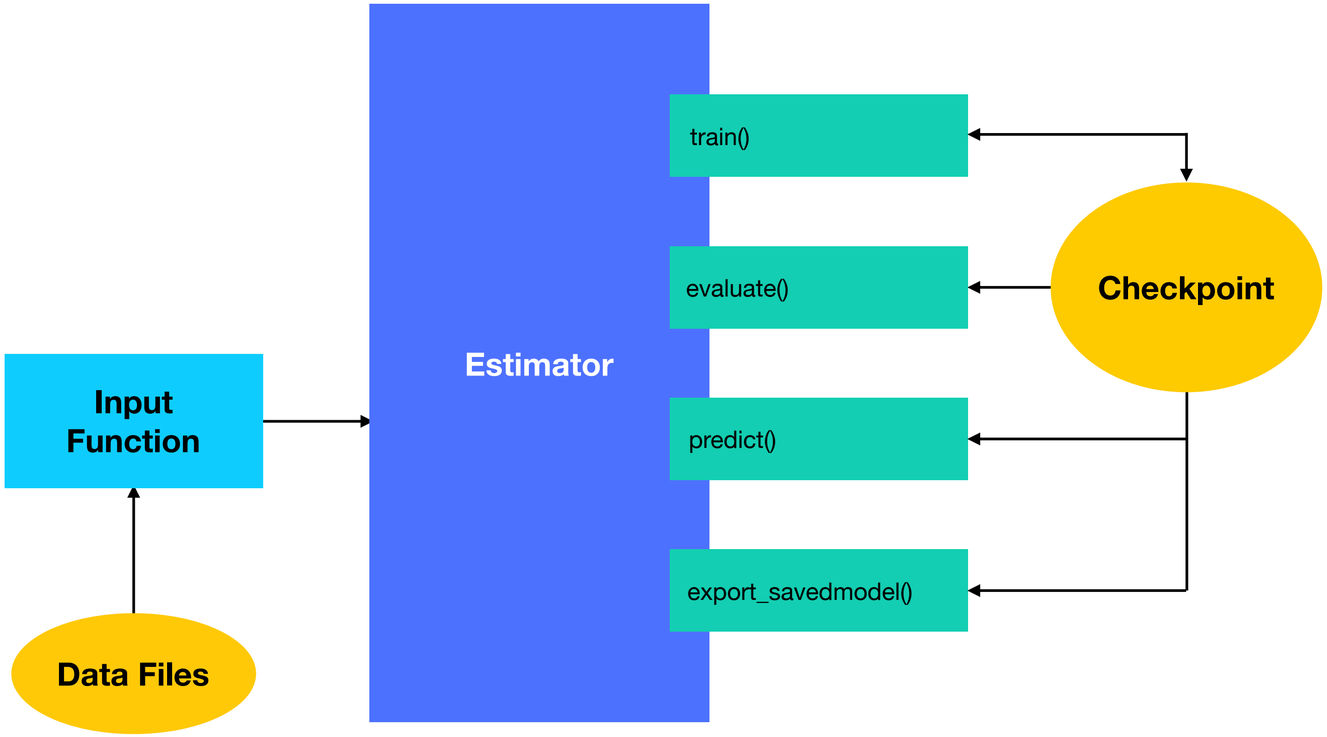

Training loop
train_op
Evaluation loop
eval_op
inputs, labels
Model
predictions


Estimator
train_op
inputs, labels
model_fn
eval_op
predictions


Estimator
train()
inputs, labels
model_fn
evaluate()
Sessions,
Graphs,
Loops
predict()


Estimator
train()
evaluate()
inputs, labels
model_fn
predict()
Sessions,
Graphs,
Loops
export_savedmodel()

Estimator Framework



Why Estimators ?

- To make it more simple and accessible.
- Allows for better parallelization when training using multi GPUs
- TensorFlow is becoming optimised for it
- It removes the python for loop and converts to C++
- TPUs will only work with Estimators and Datsets APIs

Canned Estimators

- Pre-made Types of networks
- Simple for swapping models and running experiments on the same data
- Currently the types of models are limited
- Probably a lot more in the future
- Possibly trying to take on SKLearn too

Steps

- Load you datasets
- Define column features
- Define your estimator
- .train your estimator
- .evaluate
- .predict
- .export_savedmodel

TensorFlow v1.4

tf.data

tf.keras

custom estimators

Coming soon!!