

Data Science: Machine Learning: Tensorflow Applied

CPSC 501: Advanced Programming Techniques
Fall 2022

Jonathan Hudson, Ph.D
Assistant Professor (Teaching)
Department of Computer Science
University of Calgary

Thursday, October 13, 2022



import tensorflow as tf

1. TF Learn (tf.contrib.learn): simplified interface that helps users transition from the the world of one-liner such as scikit-learn
 2. TF Slim (tf.contrib.slim): lightweight library for defining, training and evaluating complex models in TensorFlow.
 3. High level API: **Keras**, TFLearn, Pretty Tensor
- Off-the-shelf models are not the main purpose of TensorFlow
 - scikit is mostly off the shelf package
 - Keras is built-in for TF 2.0
 - **We will be using TF 2.0**

Why TensorFlow 2.0

- Why TF 2.0
- It is the default
- `pip install tensorflow`
 - gives you 2.x
 - Works naturally with models and keras
- We have mostly moved out of a transition period (lots of older tutorials were 1.XX based)
- Note, sometimes tensorflow support is behind latest version of python, example. If python is at 3.10, sometimes tensorflow might only be available for 3.9

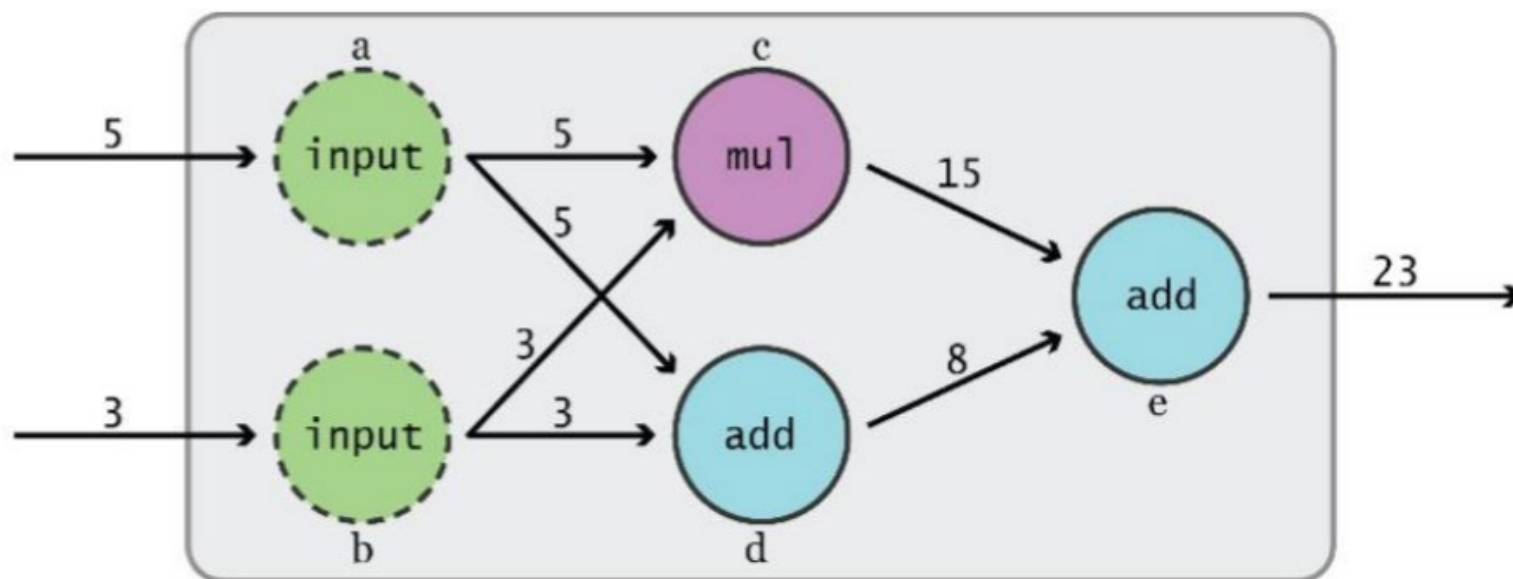
Data

What's a tensor?

- An n-dimensional array
 1. 0-d tensor: scalar (number)
 2. 1-d tensor: vector
 3. 2-d tensor: matrix
 4. and so on

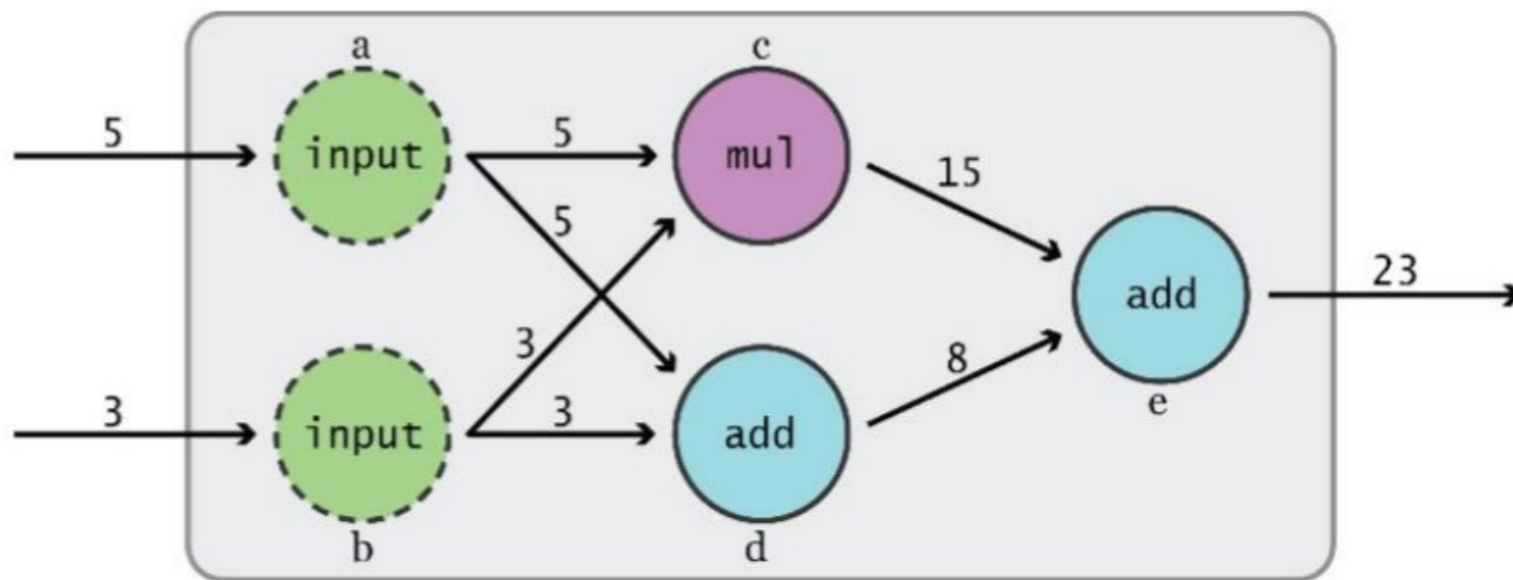
Data Flow

- Tensor -> Data
- Flow -> Move
- Visualize TensorFlow operations as graphs



Data Flow

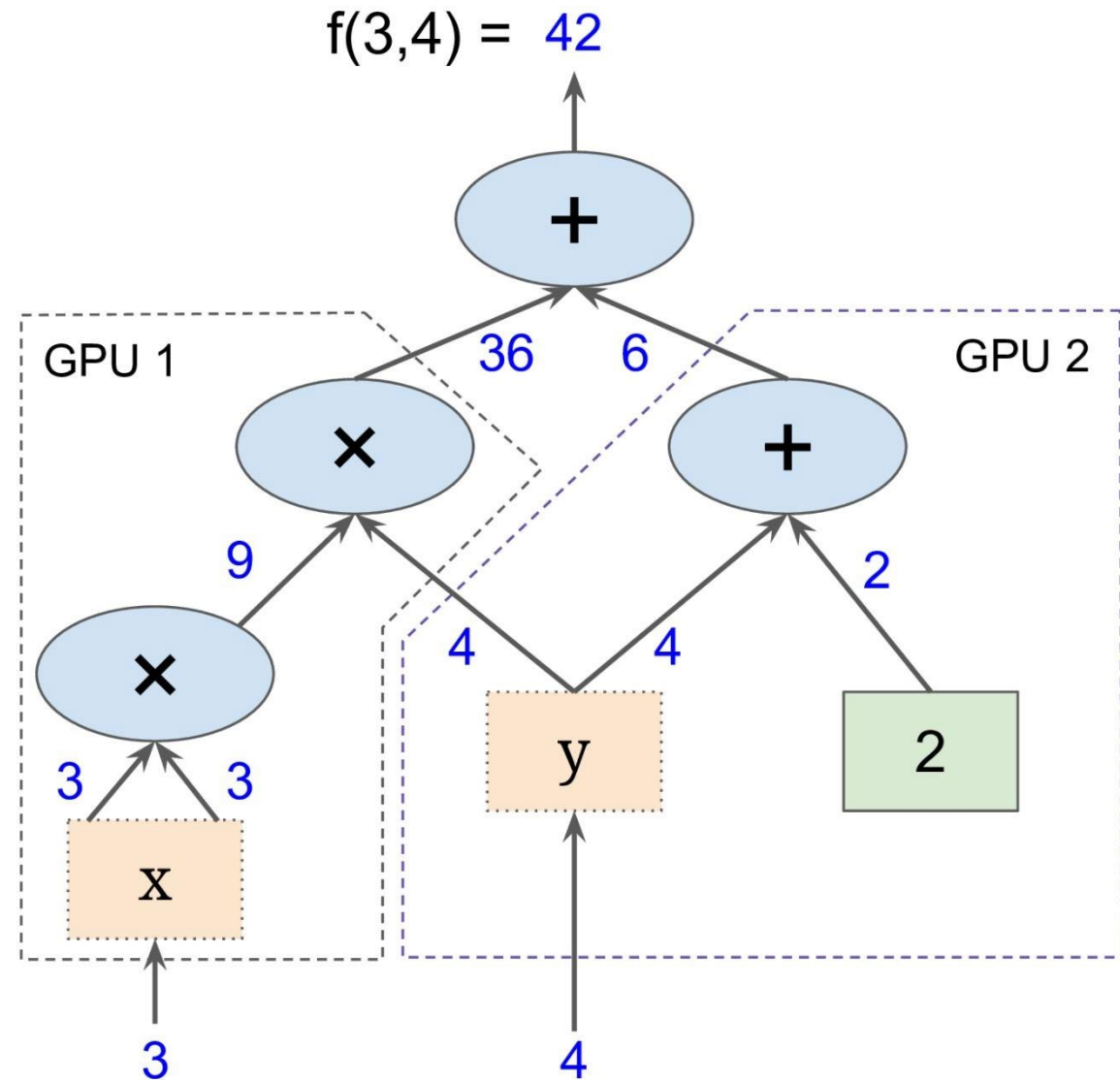
- Visualize TensorFlow operations as graphs



- **TensorFlow 2.0 (added Eager Execution)**
- Assembles and executes a graph at same time

Subgraphs

Possible to break graphs into several chunks across multiple CPUs, GPUs, TPUs, or other devices



Why graphs

1. Save computation. Only run subgraphs that lead to the values you want to fetch.
2. Break computation into small, differential pieces to facilitate auto-differentiation
3. Facilitate distributed computation, spread the work across multiple CPUs, GPUs, TPUs, or other devices
4. Many common machine learning models are taught and visualized as directed graphs

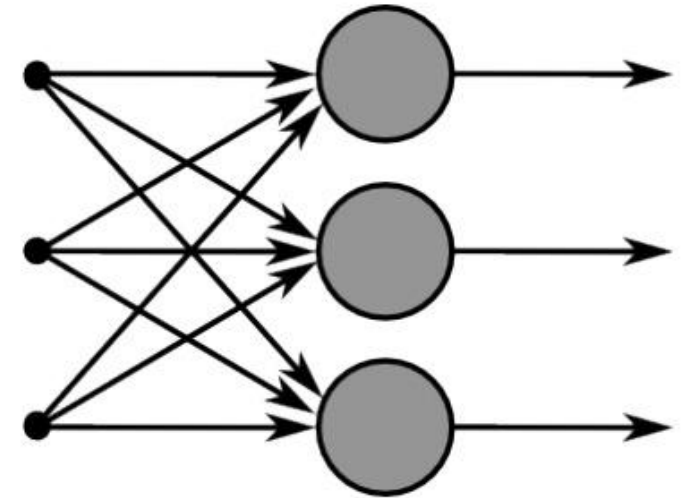
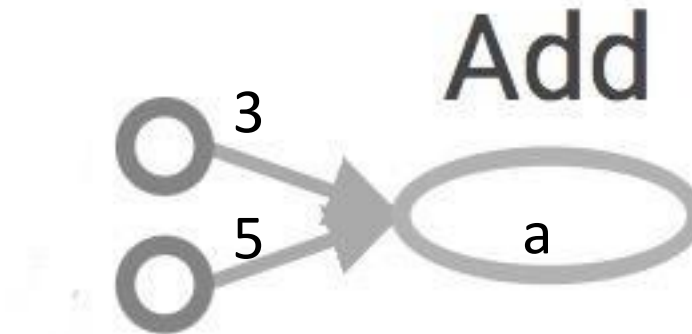


Figure 3: This image captures how multiple sigmoid units are stacked on the right, all of which receive the same input x .

Simple example

Data Flow Graphs

```
import tensorflow as tf
a = tf.add(3, 5)
print(a)
print(a.numpy())
```



```
>> tf.Tensor(8, shape=(), dtype=int32)
>> 8
```

We now have a value 8 (eager execution)

Larger graph

`x = 2`

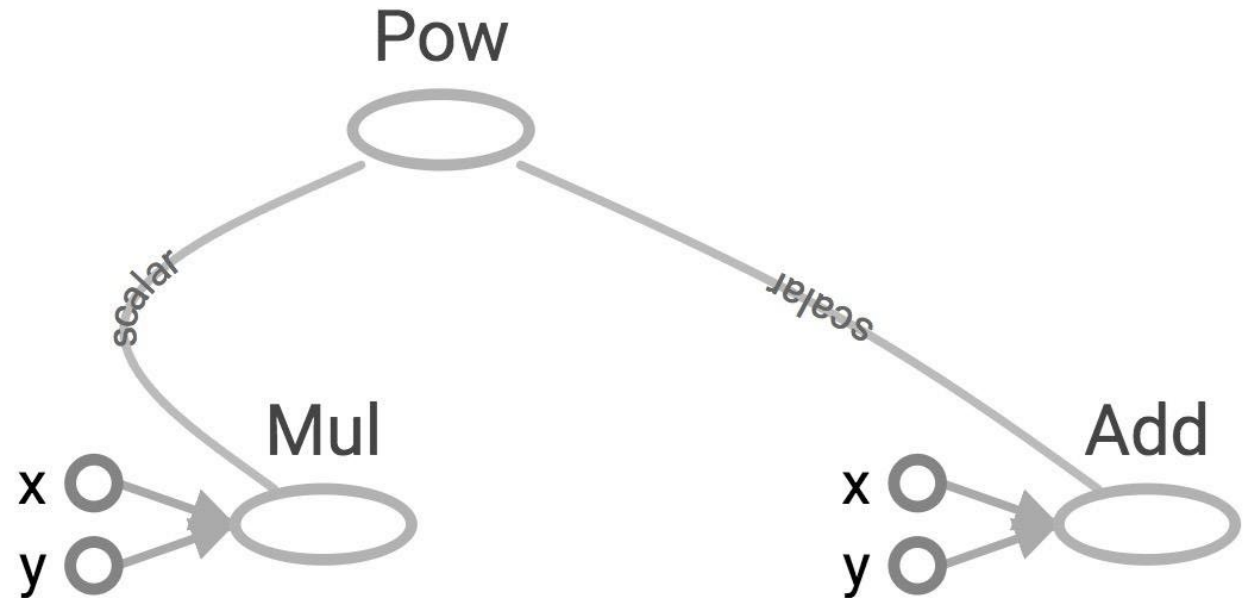
`y = 3`

`op1 = tf.add(x, y)`

`op2 = tf.multiply(x, y)`

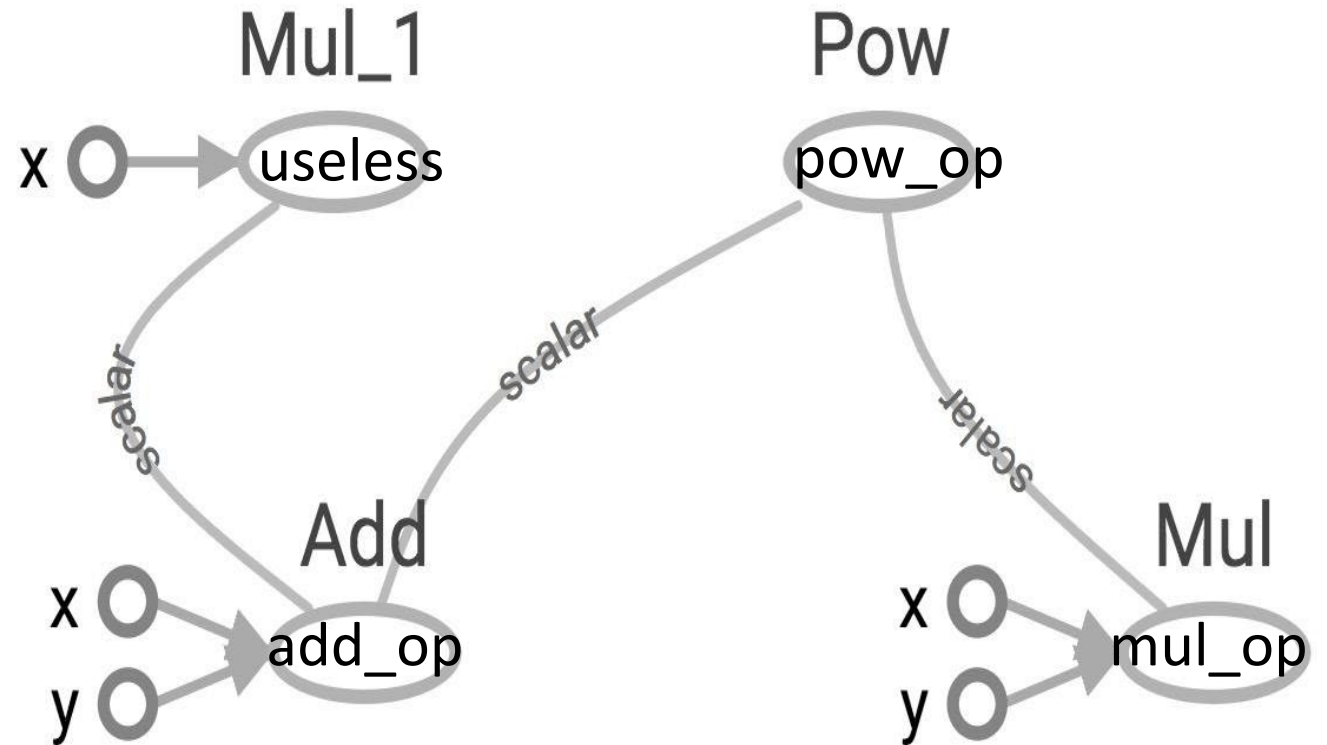
`op3 = tf.pow(op2, op1)`

`print(op3.numpy())`



Subgraphs

```
x = 2
y = 3
add_op = tf.add(x, y)
mul_op = tf.multiply(x, y)
useless = tf.multiply(x, add_op)
pow_op = tf.pow(mul_op, add_op)
print(pow_op.numpy(), useless.numpy)
```



Visualize

Visualize it with TensorBoard

```
@tf.function
def my_func(x, y):
    return tf.add(x, y)
x = tf.constant(2)
y = tf.constant(3)
tf.summary.trace_on()
z = my_func(x, y)
writer = tf.summary.create_file_writer(logdir)
with writer.as_default():
    tf.summary.trace_export(name="my_func_trace",step=0,profiler_o
utdir=logdir)
```

Run it

Go to terminal, run:

```
$ python [yourprogram].py
```

```
$ tensorboard --logdir="./logs/tf1" --port 6006
```

Then open your browser and go to: <http://localhost:6006/>



Fit to screen



Download PNG

Run simple ▼

(4)

Session _____ ▼

runs (0)

Upload 

Trace inputs

Color Structure Device

▼ Close legend.

Graph (* = expandable)



Namespace* ?



OpNode ?



Unconnected series* ?



Connected series* ?



Constant ?



Summary ?



Dataflow edge ?



Control dependency edge ?



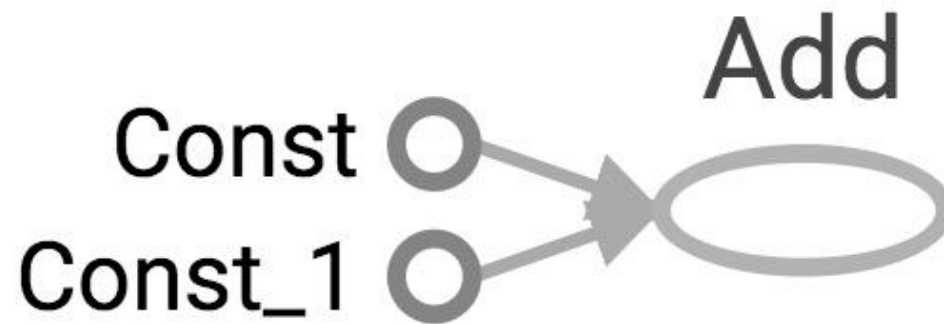
Reference edge ?

Main GraphAuxiliary Nodes



Visualize it with TensorBoard

```
import tensorflow as tf  
a = tf.constant(2)  
b = tf.constant(3)  
x = tf.add(a, b)  
writer = tf.summary.FileWriter(logdir, tf.get_default_graph())  
writer.close()
```



Question:

How to change Const, Const_1 to the names we give the variables?

Explicitly name them

```
import tensorflow as tf
```

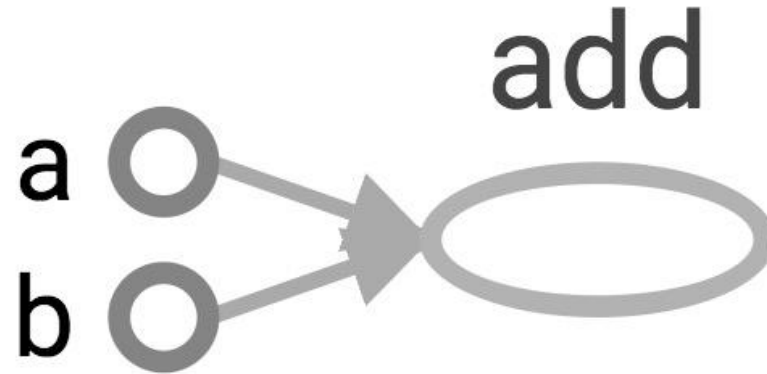
```
a = tf.constant(2, name='a')
```

```
b = tf.constant(3, name='b')
```

```
x = tf.add(a, b, name='add')
```

```
writer = tf.summary.FileWriter(logdir, tf.get_default_graph())
```

```
writer.close()
```



TensorBoard

- TensorBoard can do much more than just visualizing your graphs.
- Learning to use TensorBoard is a valuable skill for those planning to use Tensorflow for more unique creations.

Onward to ... tensorflow API.

Jonathan Hudson
jwhudson@ucalgary.ca
<https://pages.cpsc.ucalgary.ca/~jwhudson/>



UNIVERSITY OF
CALGARY