Introduction to Tensorflow
CMPT 498/898 Deep Learning and Applications

Najeeb Khan
najeeb.khan@usask.ca

Winter 2020
TensorFlow (v1.x)

TensorFlow is an interface for expressing machine learning algorithms, and an implementation for executing such algorithms (Abadi et al., 2015).

- Represents computations as **graphs**.
  - Nodes in the graph are called ops
  - Edges are tensors

- Represents data as **tensors**.
  - A tensor is an n-dimensional array with a rank, shape and type.
  - For example `[batch, height, width, channels]`

- Executes graphs in the context of **sessions**.
  - A session places the graph ops onto devices such as CPUs/GPUs etc.

- Maintains state with **variables**.
  - Typically represent the parameters of a statistical model as a set of variables

- Uses **feeds** and **fetches** to get data into and out of arbitrary operations.
Abstraction levels

- TensorFlow Estimators: High-level, object-oriented API
- tf.layers, tf.losses, tf.metrics: Reusable libraries for common model components
- Python TensorFlow: Provides Ops, which wrap C++ Kernels
- C++ TensorFlow: Kernels work on one or more platforms

Image Source: developers.google.com
Environments

- CPU version of Tensorflow
  
  `pip install tensorflow==1.15.0`

- GPU version requires CUDA Toolkit, docs at tensorflow.org/install

- Google Colab: colab.research.google.com
TensorFlow v1.x Demos
TensorFlow (v2.0)

- Eager execution:
  - TF v1.x: Graph creation, graph execution
  - TF v2.0: Identical behavior to normal python code
    - Dynamic graphs
    - In-order execution

- No more globals:
  - TF v1.x: a variable remain in memory even if you lost track of the Python variable pointing to it
  - TF v2.0: Keep track of your variables! If you lose track of a tf.Variable, it gets garbage collected.

- Functions, not sessions
  - TF v1.x: session.run()
  - TF v2.0: Decorate a Python function using tf.function() to mark it for JIT compilation

- Inter-mix core Python and TF
  - Use python along with TF v2.0 code
  - TF code executes in contexts without a python interpreter (Mobile, C++, JS)
Digression: Python Decorators

```python
def my_decorator(func):
    def wrapper():
        print("Something before the function is called.")
        func()
        print("Something after the function is called.")
    return wrapper

def say_whee():
    print("Whee!")

say_whee = my_decorator(say_whee)
say_whee()

@my_decorator
def say_whee():
    print("Whee!")
say_whee()
```

source: https://realpython.com/primer-on-python-decorators/
TensorFlow 2.0 Demos
References