TensorFlow

Prayas Jain

Department of Computer Science and Engineering
Indian Institute of Technology (BHU)
Varanasi, India-221005
Email: prayas.jain.cse14@iitbhu.ac.in

February 16, 2017

Outline

Deep Learning Libraries

- Torch
- Caffe
- TensorFlow
- Theano
- Keras
- etc etc.

Which one to use?

- Model Specification: Configuration File (eg **Caffe**) vs Program Writing (eg **TensorFlow** , **Theano**)
- For Program Writing, Programming Language: Lua (**Torch**) vs Python (**TensorFlow**)
- Prefer Python because of rich community and library infrastructure.
- **TensorFlow** vs **Theano**: Both are very similar systems. Theano was inspiration for TensorFlow. TensorFlow has better support for distributed systems

What is TensorFlow?

- A deep learning library for Python, open-sourced by Google.
- Provides primitives for defining functions based on Tensors, automatically computing the Derivative

What is a Tensor?

- A map from vector spaces to real numbers
- Scalars, Vectors and Matrices are Tensors.
- Basically, A multi-dimensional array of numbers

Numpy vs TensorFlow

- Both are quite similar. (Both are Nd Array Libraries)
- TensorFlow allows writing of Tensor Functions with automatic derivate compution
- TensorFlow has GPU support

Numpy to TensorFlow Mapping

Numpy	TensorFlow
`a = np.zeros((2,2)); b = np.ones((2,2))`	`a = tf.zeros((2,2)), b = tf.ones((2,2))`
`np.sum(b, axis=1)`	`tf.reduce_sum(a,reduction_indices=[1])`
`a.shape`	`a.get_shape()`
`np.reshape(a, (1,4))`	`tf.reshape(a, (1,4))`
`b * 5 + 1`	`b * 5 + 1`
`np.dot(a,b)`	`tf.matmul(a, b)`
`a[0,0], a[:,0], a[0,:]`	`a[0,0], a[:,0], a[0,:]`

TensorFlow Computation Graph

- TensorFlow programs have two phases:
- **Construction Phase** : Create and assemble a computation graph
- **Evaluation Phase** : Use a session to evaluate operations and evaluate the values of nodes in the graph
- The graph will have entry points, to give input (Data) , an internal network of nodes doing various computations, and exit points to output results.

TensorFlow Sessions and Evals

- TensorFlow requires explicit evaluation (unlike Numpy).
- Session objects encapsulate the environment in which the Tensor objects are evaluated
- Evals (tensor.eval()) are used to evaluate the value of a tensor

Constants and Variables

- The value of Constant Tensors cannot be changed.
- Variable Tensors are used to store and update parameters.
- Variable Tensors need to be initialized before doing anything.
- Can be used like constants after initializing.

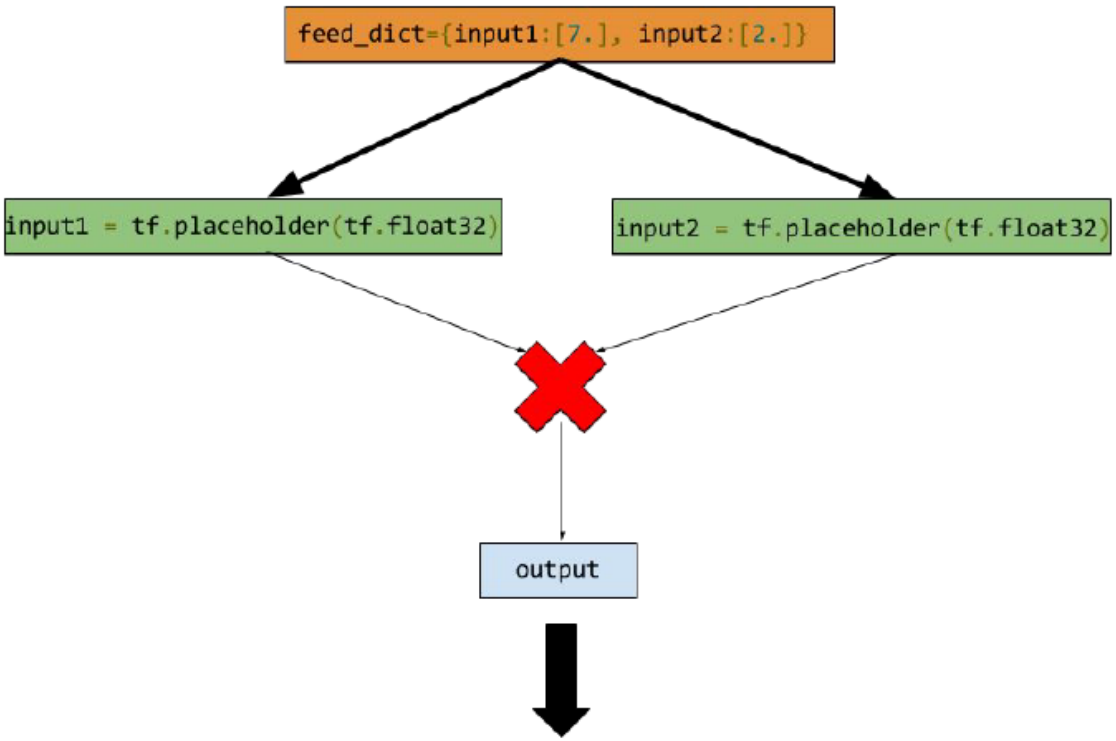
```
In [32]: W1 = tf.ones((2,2))

In [33]: W2 = tf.Variable(tf.zeros((2,2)), name="weights")

In [34]: with tf.Session() as sess:
             print(sess.run(W1))
             sess.run(tf.initialize_all_variables())
```

Sample Code - 1


```
In [63]: state = tf.Variable(0, name="counter")
In [64]: new_value = tf.add(state, tf.constant(1))
In [65]: update = tf.assign(state, new_value)
In [66]: with tf.Session() as sess:
    ....:     sess.run(tf.initialize_all_variables())
    ....:     print(sess.run(state))
    ....:     for _ in range(3):
    ....:         sess.run(update)
    ....:         print(sess.run(state))
    ....:
0
1
2
3
```

Inputting Data (Placeholders and Dictionaries)

- **Placeholders** : tf.placeholder creates dummy nodes that provide entry points in the graph.
- **Feed Dictionaries**: A dictionary mapping placeholders to data (numpy array, list, numbers etc)



Demo

We will run a simple linear regression model on synthetic data.