



TensorFlow

Tutorial

Introduction

Initialize a vector of length 100 with the value 1

Multiply each element by 2

Introduction

C

```
int input[100];  
//Initialize array  
for(int i = 0; i < 100; i++){  
    input[i] = 1;  
}  
  
int output[100];  
//Multiply array by 2 element wise  
for(int i = 0; i < 100; i++){  
    output[i] = input[i] * 2;  
}
```

Introduction

C

```
int input[100];  
//Initialize array  
for(int i = 0; i < 100; i++){  
    input[i] = 1;  
}  
  
int output[100];  
//Multiply array by 2 element wise  
for(int i = 0; i < 100; i++){  
    output[i] = input[i] * 2;  
}
```

Python + Numpy

```
import numpy as np  
input = np.ones([100])  
output = input * 2
```

Introduction

C

```
int input[100];  
//Initialize array  
for(int i = 0; i < 100; i++){  
    input[i] = 1;  
}  
  
int output[100];  
//Multiply array by 2 element wise  
for(int i = 0; i < 100; i++){  
    output[i] = input[i] * 2;  
}
```

Python + Numpy

```
import numpy as np  
input = np.ones([100])  
output = input * 2
```

Matlab

```
input = ones(100, 1);  
ouptut = input .* 2;
```

Python + TensorFlow

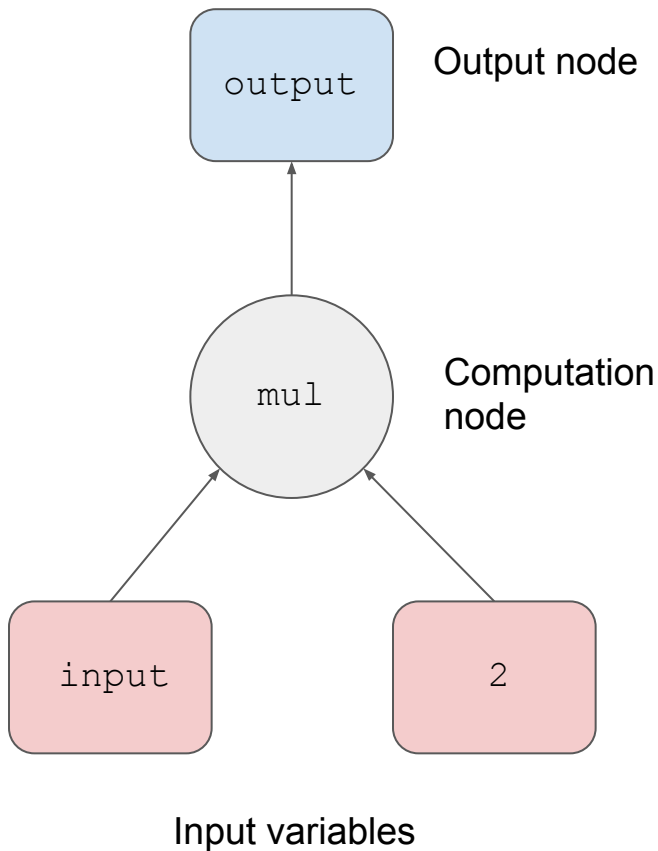
Build graph

```
import tensorflow as tf
input = tf.ones([100])
output = input * 2
>>> <tf.Tensor 'Mul:0' shape=(100,) dtype=float32>
```

Python + TensorFlow

Build graph

```
import tensorflow as tf
input = tf.ones([100])
output = input * 2
>>> <tf.Tensor 'Mul:0' shape=(100,) dtype=float32>
```



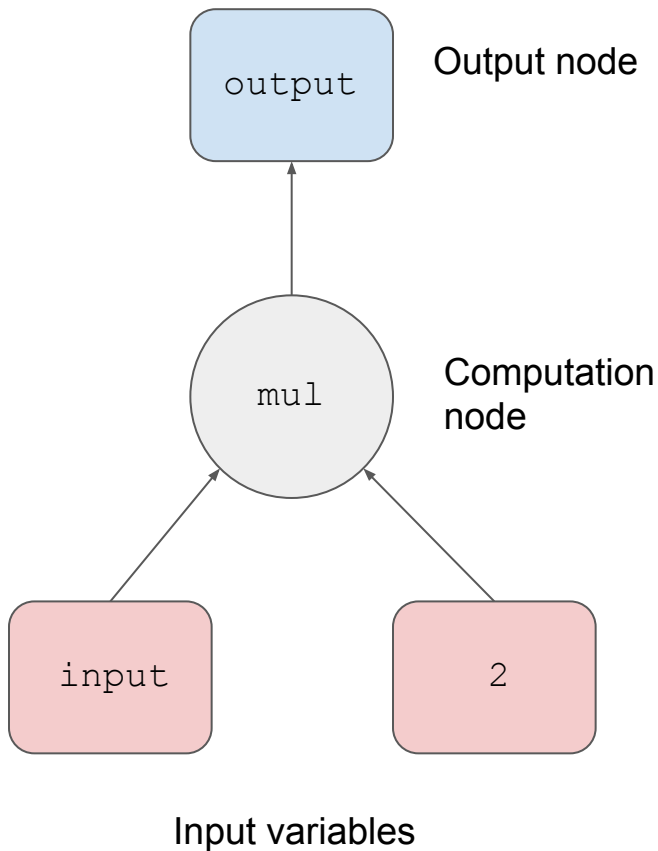
Python + TensorFlow

Build graph

```
import tensorflow as tf
input = tf.ones([100])
output = input * 2
>>> <tf.Tensor 'Mul:0' shape=(100,) dtype=float32>
```

Evaluate graph

```
sess = tf.session()
sess.run(output)
>>> array([ 2.,  2.,  2.,  2.,  2.,  2.,  2.,  2.,  2.,
           2.,  2.,  2.,  2., ...])
```



Advantages

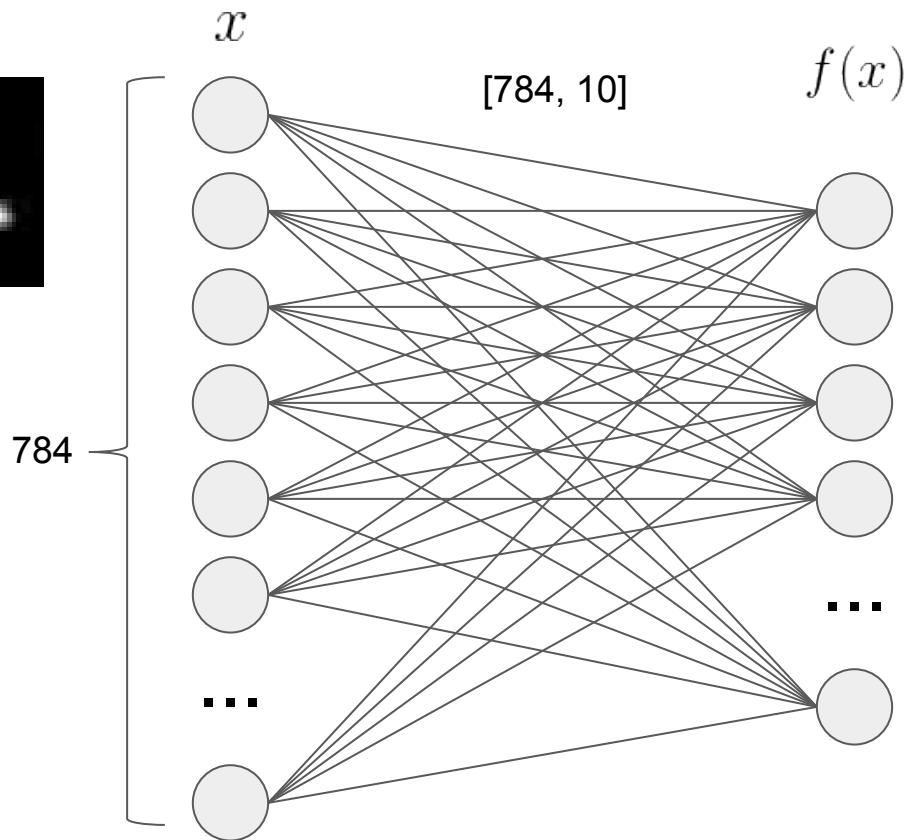
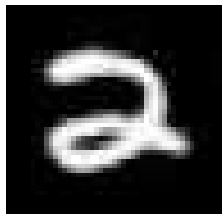
Hardware abstraction

Optimizations on graph

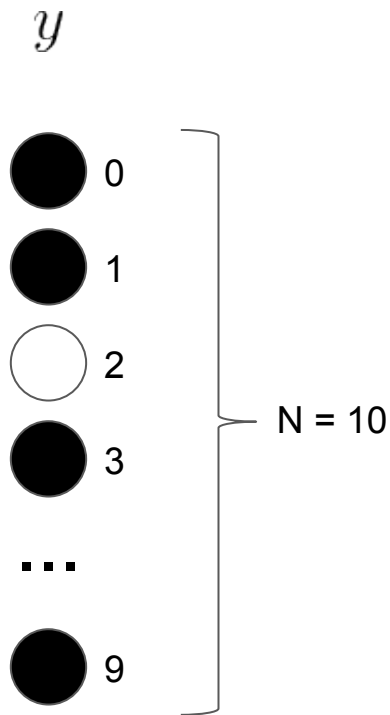
Automatic gradient calculations

Online visualization tools

Task



$$L = \frac{1}{2N} \sum_i^N (y_i - f(x)_i)^2$$



Building Graph

```
import tensorflow as tf
```



Placeholders

```
import tensorflow as tf  
inImage = tf.placeholder(tf.float32, shape=[None, 784]) #Input image is 28 x 28  
gt = tf.placeholder(tf.float32, shape=[None, 10])
```

Variables

```
import tensorflow as tf  
inImage = tf.placeholder(tf.float32, shape=[None, 784])  
gt = tf.placeholder(tf.float32, shape=[None, 10])  
W = tf.Variable(tf.zeros([784,10]))  
b = tf.Variable(tf.zeros([10]))
```

Operation

```
import tensorflow as tf  
inImage = tf.placeholder(tf.float32, shape=[None, 784])  
gt = tf.placeholder(tf.float32, shape=[None, 10])  
W = tf.Variable(tf.zeros([784,10]))  
b = tf.Variable(tf.zeros([10]))  
est = tf.matmul(inImage, W) + b
```

Operation

```
import tensorflow as tf
inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
```

Optimizer

```
import tensorflow as tf

inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.1).minimize(loss)
```


Initialization

```
import tensorflow as tf

inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.1).minimize(loss)
sess = tf.Session()
sess.run(tf.global_variables_initializer())
```

Evaluation

```
import tensorflow as tf

inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.1).minimize(loss)
sess = tf.Session()
sess.run(tf.global_variables_initializer())
for i in range(1000):
    batch_input, batch_gt = mnist.train.next_batch(256)
    sess.run(opt, feed_dict={inImage: batch_input, gt: batch_gt})
```

Evaluation

```
import tensorflow as tf

inImage = tf.placeholder(tf.float32, shape=[None, 784])
gt = tf.placeholder(tf.float32, shape=[None, 10])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
est = tf.matmul(inImage, W) + b
loss = tf.reduce_mean(tf.square(gt - est))/2
opt = tf.train.GradientDescentOptimizer(0.1).minimize(loss)
sess = tf.Session()
sess.run(tf.global_variables_initializer())
for i in range(1000):
    batch_input, batch_gt = mnist.train.next_batch(256)
    sess.run(opt, feed_dict={inImage: batch_input, gt: batch_gt})
    print("Loss on step", i, ":", sess.run(loss, feed_dict={inImage:batch_input, gt:batch_gt}))
```

Demo

Code

<https://github.com/slundqui/TFTutorial>