**TensorFlow Machine Learning Cookbook**

**CH1 Getting Started with TensorFlow**

1.1 How TensorFlow Works

大部分会有以下步骤：

Import or generate datasets

Transform and normalize data

Partition datasets into train, test, and validation sets

Set algorithm parameters (hyperparameters)

Initialize variables and placeholders:

tensorflow需要知道哪些可以修改哪些不能，它将修改variables和weight/bias以最小化loss。为了实现这个目标，通过placeholder来feed数据。初始化必须包括size和type。

Define the model structure

Declare the loss functions

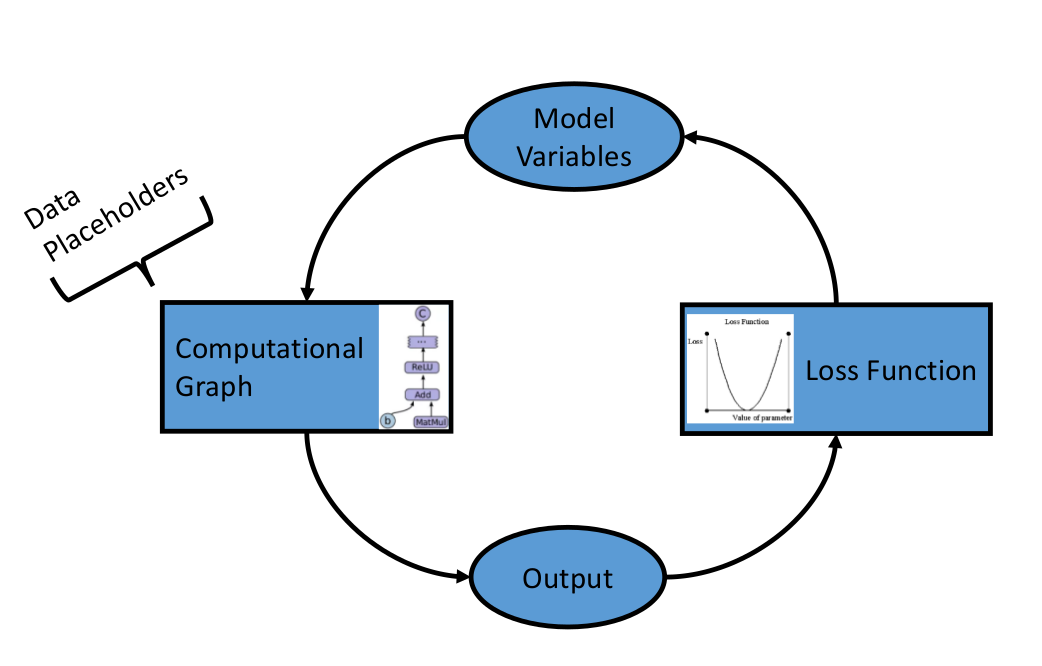
Initialize and train the model:

创建一个graph instance，feed数据给placeholder，让tensorflow修改variables。

Evaluate the model

Tune hyperparameters

Deploy/predict new outcomes



1.2 Declaring Variables and Tensors

**1.3 Using Placeholders and Variables**

placeholder是占个位置，feed数据用的。

variables是tensorfolw可变的变量，比如模型的参数w和b，所以这些变量在运算之前需要初始化。

1.4 Working with Matrices

**1.5 Declaring Operations**

除了标准的算术操作：add( ), sub( ), mul( ), div( )。也有对应的变种，比如truediv()，floordiv(), mod(), cross()。

|  |  |
| --- | --- |
| abs() | Absolute value of one input tensor |
| ceil() | Ceiling function of one input tensor |
| cos() | Cosine function of one input tensor |
| exp() | Base *e* exponential of one input tensor |
| floor() | Floor function of one input tensor |
| inv() | Multiplicative inverse (1/x) of one input tensor |
| log() | Natural logarithm of one input tensor |
| maximum() | Element-wise max of two tensors |
| minimum() | Element-wise min of two tensors |
| neg() | Negative of one input tensor |
| pow() | The first tensor raised to the second tensor element-wise |
| round() | Rounds one input tensor |
| rsqrt() | One over the square root of one tensor |
| sign() | Returns -1, 0, or 1, depending on the sign of the tensor |
| sin() | Sine function of one input tensor |
| sqrt() | Square root of one input tensor |
| square() | Square of one input tensor |

还有一些专有的计算：

|  |  |
| --- | --- |
| digamma() | Psi function, the derivative of the lgamma() function |
| erf() | Gaussian error function, element-wise, of one tensor |
| erfc() | Complimentary error function of one tensor |
| igamma() | Lower regularized incomplete gamma function |
| igammac() | Upper regularized incomplete gamma function |
| lbeta() | Natural logarithm of the absolute value of the beta function |
| lgamma() | Natural logarithm of the absolute value of the gamma function |
| squared\_difference() | Computes the square of the differences between two tensors |

除了这些内置的function，我们也可以自己定义functions。

**1.6 Implementing Activation Functions**

多层和非线性是神经网络的两个核心，其中非线性是靠activation function实现的。

而activation function也不止常见的sigmoid和ReLU，在这一节里我们看到有ReLU，ReLU6，sigmod，tanh，softsign，softplus，ELU。

选择用哪一种，如果tensors被scaled到均值为0，那么意味着选tanh货softsign；如果tensors被scaled到全是正数，那么选能在正数域保存方差的。

1.7 Working with Data Sources

1.8 Additional Resources

是不是可以写github，这样我做的大量笔记都可以查询

**CH2 The TensorFlow Way**

2.1 Operations in a Computational Graph

2.2 Layering Nested Operations

2.3 Working with Multiple Layers

2.4 Implementing Loss Functions

**2.5 Implementing Back Propagation**

1）这里面的案例都是一上来就sess = tf.Session()，打开graph的接口。

2）然后中间就会陆续调用sess.run实现graph里面的一些操作，比如sess.run(init)。

3）而在for-loop里面真正跑模型的时候只需要一句sess.run(train\_step, feed\_dict={x\_data: rand\_x, y\_target:rand\_y})。因为train\_step在上面已经定义好为求loss的最小化，所以一切交给机器去往上找到对应的操作和variable及tensor即可。

4）对机器来说，计算预测值和准确率一样都只是在sess里传送tensor和计算，只需要sess.run(tf.round(tf.sigmoid(my\_output)),feed\_dict = {})即可。

2.6 Working with Batch and Stochastic Training

随机训练每次只训练一个样本。Batch是一次训练n个样本。

x\_data = tf.placeholder(shape=[**None**, 1], dtype=tf.float32)

2.7 Combining Everything Together

2.8 Evaluating Models