Optimizer

- tf.train.GradientDescentOptimizer
- tf.train.AdadeltaOptimizer
- tf.train.AdagradDAOptimizer
- tf.train.MomentumOptimizer
- tf.train.AdamOptimizer
- tf.train.FtrlOptimizer
- tf.train.ProximalAdagradOptimizer
- tf.train.RMSPropOptimizer

各种优化器对比

- 标准梯度下降法：先计算所有样本汇总误差，然后根据总误差来更新权值；
- 随机梯度下降法：随机抽取一个样本来计算误差，然后更新权值；
- 批量梯度下降法：一种折中的方案，从总样本中选取一个批次（比如一共有10000个样本，随机选取100个样本作为一个batch），然后计算这个batch的总误差，根据总误差来更新权值。

```
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data    #手写数字相关的数据包
```

```
# 载入数据集
mnist = input_data.read_data_sets("MNIST_data",one_hot=True)     #载入数据，{数据集包路径，把标签转化为只有0和1的形式}

#定义变量，即每个批次的大小
batch_size = 100    #一次放100章图片进去
n_batch = mnist.train.num_examples // batch_size   #计算一共有多少个批次；训练集数量（整除）一个批次大小

#定义两个placeholder
x = tf.placeholder(tf.float32,[None,784])    #[行不确定，列为784]
y = tf.placeholder(tf.float32,[None,10])    #数字为0-9，则为10

#创建简单的神经网络
W = tf.Variable(tf.zeros([784,10]))   #权重
b = tf.Variable(tf.zeros([10]))    #偏置
prediction = tf.nn.softmax(tf.matmul(x,W)+b)    #预测

#定义二次代价函数
# loss = tf.reduce_mean(tf.square(y-prediction))
#定义交叉熵代价函数
loss = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(labels=y,logits=prediction))
#使用梯度下降法
# train_step = tf.train.GradientDescentOptimizer(0.2).minimize(loss)
#使用不同优化器
train_step = tf.train.AdamOptimizer(1e-2).minimize(loss)

```

```python
#初始化变量
init = tf.global_variables_initializer()

#准确数，结果存放在一个布尔型列表中
correct_prediction = tf.equal(tf.argmax(y,1),tf.argmax(prediction,1))    #比较两个参数大小是否相同，同则返回为true，不同则返回为false；argmax()：返回张量中最大的值所在的位置

#求准确率
accuracy = tf.reduce_mean(tf.cast(correct_prediction,tf.float32))  #cast()：将布尔型转换为32位的浮点型；（比方说9个T和1个F，则为9个1，1个0，即准确率为90%）

with tf.Session() as sess:
    sess.run(init)
    for epoch in range(21):
        for batch in range(n_batch):
            batch_xs,batch_ys = mnist.train.next_batch(batch_size)
            sess.run(train_step,feed_dict={x:batch_xs,y:batch_ys})

        acc = sess.run(accuracy,feed_dict={x:mnist.test.images,y:mnist.test.labels})
        print("Iter" + str(epoch) + ",Testing Accuracy" + str(acc))

```

```
Extracting MNIST_data\train-images-idx3-ubyte.gz
Extracting MNIST_data\train-labels-idx1-ubyte.gz
Extracting MNIST_data\t10k-images-idx3-ubyte.gz
Extracting MNIST_data\t10k-labels-idx1-ubyte.gz
Iter0,Testing Accuracy0.9208
Iter1,Testing Accuracy0.9262
Iter2,Testing Accuracy0.9214
Iter3,Testing Accuracy0.9249
Iter4,Testing Accuracy0.9283
Iter5,Testing Accuracy0.932
Iter6,Testing Accuracy0.9302
Iter7,Testing Accuracy0.9328
Iter8,Testing Accuracy0.9311
Iter9,Testing Accuracy0.9299
Iter10,Testing Accuracy0.932
Iter11,Testing Accuracy0.9299
Iter12,Testing Accuracy0.9335
Iter13,Testing Accuracy0.9281
Iter14,Testing Accuracy0.9318
Iter15,Testing Accuracy0.9313
Iter16,Testing Accuracy0.9329
Iter17,Testing Accuracy0.9324
Iter18,Testing Accuracy0.9334
Iter19,Testing Accuracy0.9314
Iter20,Testing Accuracy0.9312
```