WTF Daily Blog

斗大的熊猫

TensorFlow练习4: CNN, Convolutional Neural Networks

Convolutional Neural Networks翻译为卷积神经网络，常用在图像识别和语音分析等领域。CNN详细介绍参看：

- https://en.wikipedia.org/wiki/Convolutional_neural_network
- http://blog.csdn.net/zouxy09/article/details/8781543
- http://deeplearning.net/tutorial/lenet.html

使用TensorFlow创建CNN

```
import tensorflow as tf
import numpy as np

# 下载mnist数据集
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets('/tmp/', one_hot=True)


n_output_layer = 10

# 定义待训练的神经网络
def convolutional_neural_network(data):
    weights = {'w_conv1':tf.Variable(tf.random_normal([5,5,1,32])),
              'w_conv2':tf.Variable(tf.random_normal([5,5,32,64])),
              'w_fc':tf.Variable(tf.random_normal([7*7*64,1024])),
              'out':tf.Variable(tf.random_normal([1024,n_output_layer

    biases = {'b_conv1':tf.Variable(tf.random_normal([32])),
              'b_conv2':tf.Variable(tf.random_normal([64])),
              'b_fc':tf.Variable(tf.random_normal([1024])),
              'out':tf.Variable(tf.random_normal([n_output_layer]))}

    data = tf.reshape(data, [-1,28,28,1])

    conv1 = tf.nn.relu(tf.add(tf.nn.conv2d(data, weights['w_conv1'],
    conv1 = tf.nn.max_pool(conv1, ksize=[1,2,2,1], strides=[1,2,2,1],

    conv2 = tf.nn.relu(tf.add(tf.nn.conv2d(conv1, weights['w_conv2'],
    conv2 = tf.nn.max_pool(conv2, ksize=[1,2,2,1], strides=[1,2,2,1],

    fc = tf.reshape(conv2, [-1,7*7*64])
    fc = tf.nn.relu(tf.add(tf.matmul(fc, weights['w_fc']), biases['b_

    # dropout剔除一些"神经元"
    #fc = tf.nn.dropout(fc, 0.8)

    output = tf.add(tf.matmul(fc, weights['out']), biases['out'])
    return output


# 每次使用100条数据进行训练
batch_size = 100

X = tf.placeholder('float', [None, 28*28])
Y = tf.placeholder('float')
# 使用数据训练神经网络
def train_neural_network(X, Y):
    predict = convolutional_neural_network(X)
    cost_func = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logit
    optimizer = tf.train.AdamOptimizer().minimize(cost_func)  # learn

    epochs = 1
    with tf.Session() as session:
        session.run(tf.initialize_all_variables())
        epoch_loss = 0
        for epoch in range(epochs):
            for i in range( int(mnist.train.num_examples/batch_size)
                x, y = mnist.train.next_batch(batch_size)
                _, c = session.run([optimizer, cost_func], feed_dict=
                epoch_loss += c
            print(epoch, ' : ', epoch_loss)

        correct = tf.equal(tf.argmax(predict,1), tf.argmax(Y,1))
        accuracy = tf.reduce_mean(tf.cast(correct,'float'))
        print('准确率: ', accuracy.eval({X:mnist.test.images, Y:mnist.

train_neural_network(X,Y)
```

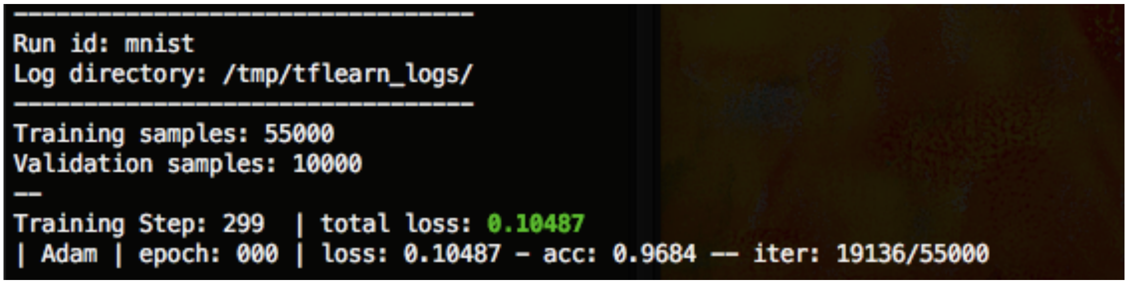
执行结果：

```
准确率:  0.9789
```

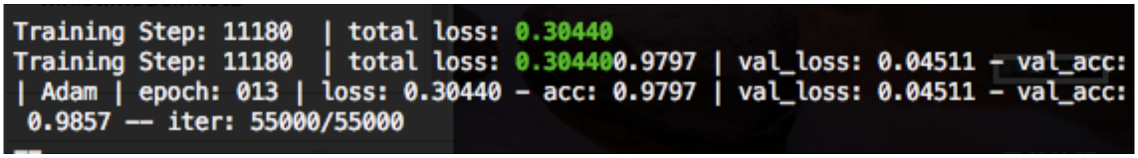
下面使用tflearn重写上面代码，tflearn是TensorFlow的高级封装，类似Keras。

tflearn提供了更简单、直观的接口。和scikit-learn差不多，代码如下：

```
import tflearn
from tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout, fully_connected
from tflearn.layers.estimator import regression

train_x, train_y, test_x, test_y = tflearn.datasets.mnist.load_data(o

train_x = train_x.reshape(-1,28,28,1)
test_x = test_x.reshape(-1,28,28,1)

# 定义神经网络模型
conv_net = input_data(shape=[None,28,28,1], name='input')
conv_net = conv_2d(conv_net, 32, 2, activation='relu')
conv_net = max_pool_2d(conv_net ,2)
conv_net = conv_2d(conv_net, 64, 2, activation='relu')
conv_net = max_pool_2d(conv_net ,2)
conv_net = fully_connected(conv_net, 1024, activation='relu')
conv_net = dropout(conv_net, 0.8)
conv_net = fully_connected(conv_net, 10, activation='softmax')
conv_net = regression(conv_net, optimizer='adam', loss='categorical_c

model = tflearn.DNN(conv_net)

# 训练
model.fit({'input':train_x}, {'output':train_y}, n_epoch=13,
          validation_set=({'input':test_x}, {'output':test_y}),
          snapshot_step=300, show_metric=True, run_id='mnist')

model.save('mnist.model')   # 保存模型

"""
model.load('mnist.model')   # 加载模型
model.predict([test_x[1]])  # 预测
"""
```



```
---------------------------------
Run id: mnist
Log directory: /tmp/tflearn_logs/
---------------------------------
Training samples: 55000
Validation samples: 10000
--
Training Step: 299  | total loss: 0.10487
| Adam | epoch: 000 | loss: 0.10487 - acc: 0.9684 -- iter: 19136/55000
```



```
Training Step: 11180  | total loss: 0.30440
Training Step: 11180  | total loss: 0.304400.9797 | val_loss: 0.04511 - val_acc:
| Adam | epoch: 013 | loss: 0.30440 - acc: 0.9797 | val_loss: 0.04511 - val_acc:
 0.9857 -- iter: 55000/55000
```

Facebook Google+ Twitter Weibo Email

相关文章

- 使用Python实现神经网络
- TensorFlow练习2: 对评论进行分类
- TensorFlow练习3: RNN, Recurrent Neural Networks
- TensorFlow练习5: 训练一个简单的游戏AI（Deep Q Network）...
- TensorFlow练习9: 生成妹子图（PixelCNN）

用微信 OR 支付宝 扫描二维码

为本文作者 打个赏

微信扫一扫转账

用支付宝扫二维码，加我好友

2016年11月17日 wtf ML、coding CNN、TensorFlow

《TensorFlow练习4: CNN, Convolutional Neural Networks》有2个想法

倾风随雨

2016年11月17日 下午4:37

No module named examples.tutorials.mnist,站长下次写代码，麻烦把数据也附加上，让我们这些初学者也学习学习~~

wtf

2016年11月17日 下午4:52

额