

WTF Daily Blog

斗大的熊猫

TensorFlow练习3: RNN, Recurrent Neural Networks

前文《[使用Python实现神经网络](#)》和《[TensorFlow练习1: 对评论进行分类](#)》都是简单的Feed-forward Neural Networks(FNN/前向反馈神经网络)。而RNN(Recurrent Neural Networks)循环神经网络要相对复杂，它引入了循环，能够处理数据有前后关系的问题，常用在自然语言处理上。

RNN介绍：

- Wiki：[Recurrent neural network](#)
- [Understanding-LSTMs](#)
- [循环神经网络\(RNN, Recurrent Neural Networks\)介绍](#)
- 唇语识别论文：<https://arxiv.org/pdf/1611.05358v1.pdf>
- [自己动手做聊天机器人教程（入门级）](#)

RNN的目的使用来处理序列数据。在传统的神经网络模型中，是从输入层到隐含层再到输出层，层与层之间是全连接的，每层之间的节点是无连接的。但是这种普通的神经网络对于很多问题却无能为力。例如，你要预测句子的下一个单词是什么，一般需要用前面的单词，因为一个句子中前后单词并不是独立的。RNNs之所以称为循环神经网络，即一个序列当前的输出与前面的输出也有关。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。

本贴在MNIST数据集上应用RNN，看看准确率和FNN相比有没有提高。

使用TensorFlow创建RNN

```
1 import tensorflow as tf
2 import numpy as np
3
4 # 下载mnist数据集
5 from tensorflow.examples.tutorials.mnist import input_data
6 mnist = input_data.read_data_sets('/tmp/', one_hot=True)
7
8
9 # 一张图片是28*28, FNN是一次性把数据输入到网络, RNN把它分成块
10 chunk_size = 28
11 chunk_n = 28
12
13 rnn_size = 256
14
15 n_output_layer = 10 # 输出层
16
17 X = tf.placeholder('float', [None, chunk_n, chunk_size])
18 Y = tf.placeholder('float')
19 # 定义待训练的神经网络
20 def recurrent_neural_network(data):
21     layer = {'w_': tf.Variable(tf.random_normal([rnn_size, n_output_la
22
23     lstm_cell = tf.nn.rnn_cell.BasicLSTMCell(rnn_size)
24
25     data = tf.transpose(data, [1,0,2])
26     data = tf.reshape(data, [-1, chunk_size])
27     data = tf.split(0, chunk_n, data)
28     outputs, status = tf.nn.rnn(lstm_cell, data, dtype=tf.float32)
29
30     ouput = tf.add(tf.matmul(outputs[-1], layer['w_']), layer['b_'])
31
32     return ouput
33
34 # 每次使用100条数据进行训练
35 batch_size = 100
36
37 # 使用数据训练神经网络
38 def train_neural_network(X, Y):
39     predict = recurrent_neural_network(X)
40     cost_func = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits
41     optimizer = tf.train.AdamOptimizer().minimize(cost_func)
42
43     epochs = 13
44     with tf.Session() as session:
45         session.run(tf.initialize_all_variables())
```

```
46     epoch_loss = 0
47     for epoch in range(epochs):
48         for i in range(int(mnist.train.num_examples/batch_size))
49             x, y = mnist.train.next_batch(batch_size)
50             x = x.reshape([batch_size, chunk_n, chunk_size])
51             _, c = session.run([optimizer, cost_func], feed_dict={
52                 'x': x, 'y': y})
53             epoch_loss += c
54             print(epoch, ' : ', epoch_loss)
55
56     correct = tf.equal(tf.argmax(predict,1), tf.argmax(Y,1))
57     accuracy = tf.reduce_mean(tf.cast(correct,'float'))
58     print('准确率: ', accuracy.eval({X:mnist.test.images.reshape(-1,
59 train_neural_network(X,Y)
```

执行结果：

```
12 : 488.65210638
准确率: 0.9875
```

比FNN提高了3个百分点。

[f](#) Facebook [g+](#) Google+ [Twitter](#) [Weibo](#) [Email](#)

相关文章

[Ubuntu 16.04 安装 Tensorflow\(GPU支持\)](#)

[TensorFlow练习4: CNN, Convolutional Neural Networks...](#)

[TensorFlow练习8: 基于RNN生成音乐](#)

[TensorFlow练习7: 基于RNN生成古诗词](#)

[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16日 👤 wtf 📁 ML、coding 🔧 RNN、TensorFlow

Copyright © 2013-2016 WTF Daily Blog | Powered by DigitalOcean