



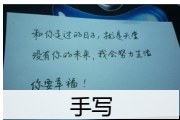
黄金白银
投资

现货白银



手写板

原油直播间



dnn神
神经网络

自我



【机器学习】Tensorflow学习笔记

2016-04-05 基陆伯 阅 21 分享： 微信 转藏到我的图书馆

构建网络模型

基本的MLP网络结构

基本的感知机模型，没有加入b

模型：

$$Y = W \cdot X$$

```
1 import tensorflow as tf
2 import numpy as np
3 import input_data
4
5 # 初始化权重 w
6 def init_weights(shape):
7     return tf.Variable(tf.random_normal(shape, stddev=0.01))
8
9 # 定义网络模型，只是基本的mlp模型，堆叠两层的逻辑回归
10 def model(X, w_h, w_o):
11     h = tf.nn.sigmoid(tf.matmul(X, w_h))
12     return tf.matmul(h, w_o) #这里没有用softmax
13
14 # 加载数据
15 mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
16 trX, trY, teX, teY = mnist.train.images, mnist.train.labels, mnist.test.images, mnist.test.labels
17
18 # 定义占位符
19 X = tf.placeholder("float", [None, 784])
20 Y = tf.placeholder("float", [None, 10])
21
22 # 初始化模型参数
23 w_h = init_weights([784, 625])
24 w_o = init_weights([625, 10])
25
26 # 定义模型
27 py_x = model(X, w_h, w_o)
28
29 # 定义损失函数
30 cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(py_x, Y))
31
32 # 定义训练操作
33 train_op = tf.train.GradientDescentOptimizer(0.05).minimize(cost) # construct an optimizer
34
35 # 定义测试操作
36 predict_op = tf.argmax(py_x, 1)
37
38 # 定义并初始化会话
39 sess = tf.Session()
40 init = tf.initialize_all_variables()
41 sess.run(init)
42
43 # 训练测试
44 for i in range(100):
45     for start, end in zip(range(0, len(trX), 128), range(128, len(trX), 128)):
46         sess.run(train_op, feed_dict={X: trX[start:end], Y: trY[start:end]})
47     print i, np.mean(np.argmax(teY, axis=1) ==
48         sess.run(predict_op, feed_dict={X: teX, Y: teY}))
```

构建多层网络

TA

永远成
我们将
我们将
[转] 贺
他们这
基督教



推荐阅

BetaC
揪出b
深度号
简易的
国外公
enum
再谈：
是还活
帧缓存



- 1 美亚保险
- 2 美亚保险
- 3 公司邮箱
- 4 企业邮箱
- 5 led亮化照
- 6 北京口腔

模型：

多层（3层模型）

```
1 import tensorflow as tf
2 import numpy as np
3 import input_data
4
5 # 初始化权重
6 def init_weights(shape):
7     return tf.Variable(tf.random_normal(shape, stddev=0.01))
8
9 # 定义模型，2层的隐藏层+ 3层的dropout
10 def model(X, w_h, w_h2, w_o, p_drop_input, p_drop_hidden):
11     X = tf.nn.dropout(X, p_drop_input) # 输入就开始用dropout
12     h = tf.nn.relu(tf.matmul(X, w_h))
13
14     h = tf.nn.dropout(h, p_drop_hidden) # dropout
15     h2 = tf.nn.relu(tf.matmul(h, w_h2))
16
17     h2 = tf.nn.dropout(h2, p_drop_hidden) # dropout
18
19     return tf.matmul(h2, w_o)
20
21 # 加载数据
22 mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
23 trX, trY, teX, teY = mnist.train.images, mnist.train.labels, mnist.test.images, mnist.test.labels
24
25 # 定义占位符+ 初始化变量
26 X = tf.placeholder("float", [None, 784])
27 Y = tf.placeholder("float", [None, 10])
28
29 w_h = init_weights([784, 625])
30 w_h2 = init_weights([625, 625])
31 w_o = init_weights([625, 10])
32
33 # dropout 的概率
34 p_keep_input = tf.placeholder("float")
35 p_keep_hidden = tf.placeholder("float")
36
37 # 模型
38 py_x = model(X, w_h, w_h2, w_o, p_keep_input, p_keep_hidden)
39
40 # 损失函数
41 cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(py_x, Y))
42 train_op = tf.train.RMSPropOptimizer(0.001, 0.9).minimize(cost)
43 predict_op = tf.argmax(py_x, 1)
44
45 sess = tf.Session()
46 init = tf.initialize_all_variables()
47 sess.run(init)
48
49 for i in range(100):
50     for start, end in zip(range(0, len(trX), 128), range(128, len(trX), 128)):
51         sess.run(train_op, feed_dict={X: trX[start:end], Y: trY[start:end],
52                                         p_keep_input: 0.8, p_keep_hidden: 0.5})
53     print i, np.mean(np.argmax(teY, axis=1) ==
54                       sess.run(predict_op, feed_dict={X: teX, Y: teY,
55                                                         p_keep_input: 1.0,
56                                                         p_keep_hidden: 1.0})))
```

卷积神经网络

模型：

```
import tensorflow as tf
import numpy as np
import input_data

def init_weights(shape):
    return tf.Variable(tf.random_normal(shape, stddev=0.01))

# 定义卷积神经网络模型
def model(X, w, w2, w3, w4, w_o, p_keep_conv, p_keep_hidden):
```

```
l1a = tf.nn.relu(tf.nn.conv2d(X, w, [1, 1, 1, 1], 'SAME'))
l1 = tf.nn.max_pool(l1a, ksize=[1, 2, 2, 1],
                    strides=[1, 2, 2, 1], padding='SAME')
l1 = tf.nn.dropout(l1, p_keep_conv)

l2a = tf.nn.relu(tf.nn.conv2d(l1, w2, [1, 1, 1, 1], 'SAME'))
l2 = tf.nn.max_pool(l2a, ksize=[1, 2, 2, 1],
                    strides=[1, 2, 2, 1], padding='SAME')
l2 = tf.nn.dropout(l2, p_keep_conv)

l3a = tf.nn.relu(tf.nn.conv2d(l2, w3, [1, 1, 1, 1], 'SAME'))
l3 = tf.nn.max_pool(l3a, ksize=[1, 2, 2, 1],
                    strides=[1, 2, 2, 1], padding='SAME')
l3 = tf.reshape(l3, [-1, w4.get_shape().as_list()[0]])
l3 = tf.nn.dropout(l3, p_keep_conv)

l4 = tf.nn.relu(tf.matmul(l3, w4))
l4 = tf.nn.dropout(l4, p_keep_hidden)

py_x = tf.matmul(l4, w_o)
return py_x

# 加载数据
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
trX, trY, teX, teY = mnist.train.images, mnist.train.labels, mnist.test.images, mnist.test.labels
trX = trX.reshape(-1, 28, 28, 1)
teX = teX.reshape(-1, 28, 28, 1)

X = tf.placeholder("float", [None, 28, 28, 1])
Y = tf.placeholder("float", [None, 10])

w = init_weights([3, 3, 1, 32])
w2 = init_weights([3, 3, 32, 64])
w3 = init_weights([3, 3, 64, 64])
w4 = init_weights([64, 64, 1, 10])
w_o = init_weights([10, 10, 1, 10])
```

360doc 个人图书馆

[首页](#)[阅览室](#)[馆友](#)[我的图书馆](#)[搜文章 找馆友](#)[登录](#)[注册](#)

```
p_keep_conv = tf.placeholder("float")
p_keep_hidden = tf.placeholder("float")
py_x = model(X, w, w2, w3, w4, w_o, p_keep_conv, p_keep_hidden)

# 损失函数
cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(py_x, Y))

# 训练操作
train_op = tf.train.RMSPropOptimizer(0.001, 0.9).minimize(cost)

# 测试操作
predict_op = tf.argmax(py_x, 1)

sess = tf.Session()
init = tf.initialize_all_variables()
sess.run(init)

for i in range(100):
    for start, end in zip(range(0, len(trX), 128), range(128, len(trX), 128)):
        sess.run(train_op, feed_dict={X: trX[start:end], Y: trY[start:end],
                                       p_keep_conv: 0.8, p_keep_hidden: 0.5})

    test_indices = np.arange(len(teX)) # Get A Test Batch
    np.random.shuffle(test_indices)
    test_indices = test_indices[0:256]

    print i, np.mean(np.argmax(teY[test_indices], axis=1) ==
                    sess.run(predict_op, feed_dict={X: teX[test_indices],
                                                    Y: teY[test_indices],
                                                    p_keep_conv: 1.0,
                                                    p_keep_hidden: 1.0})))
```

[转藏到我的图书馆](#)[献花 \(0\)](#)[分享：](#)[微信](#) ▼[来自：基陆伯 > 《Tensorflow》](#)[以文找文](#) | [举报](#)[上一篇：【机器学习】AlexNet 的tensorflow 实现](#)[下一篇：基于tensorflow的MNIST手写数字识别（二）](#)[猜你喜欢](#)



类似文章

更多

精选文章

- 【机器学习】AlexNet的tensorflow实现...

深入MNIST code测试

用Tensorflow基于Deep Q Learning DQN玩...

lecture1

【TensorFlow教程】TensorFlow-安装配置...

正则化方法：L1和L2 regularization、数...

Deep learning：四十一(Dropout简单理解...

【TensorFlow教程】TensorFlow 实战之 K...
- 朋友之间的11条潜规则

澳大利亚的美丽风景

解密中组部是如何运作的

梦幻般的风景油画

中医特效消炎药

青年才俊诸葛亮是如何找工作的？

65个成功的面试技巧

手掌里的奇妙知识



- 1 电梯制造厂家电梯制造报价..

2 教炒股票,让它成为你的法宝

3 夫妻俩开店年收100万!
- 1 美亚保险官网

2 美亚保险

3 公司邮箱

4 英语学习

5 北京口腔医院

6 企业邮箱注册

发表评论:

请 [登录](#) 或者 [注册](#) 后再进行评论

社交帐号登录: