



python趣

python项



dnn神经网络

python环



原油直播间

小火锅加盟费



神殿栋梁 图书馆

★★★★★

11471 馆藏 34051

# TensorFlow之逻辑斯谛回归

2016-04-09 神殿栋梁 阅 6

分享： 微信 转到我的图书馆

用tensorflow来写计算过程已经有很多人在github给出了教程和代码，这里安利一个这个哥们写的：<http://s://github.com/nlintz/TensorFlow-Tutorials>，是一个系列。

这里不会说logistic regression的具体内容，内部的算法原理参考李航老师的《统计机器学习》的逻辑斯谛回归与最大熵一章和Speech and Language Processing的第六章MAXIMUM ENTROPY MODELS，两者我都觉得讲得很棒。

在TensorFlow里这个模型用5行代码就能够完成：

```
1 model = tf.add(tf.matmul(X, w), b)
2 cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(model, Y))
3 train = tf.train.GradientDescentOptimizer(0.1).minimize(cost)
```

逻辑斯谛的参数是w和b，构造计算节点完成计算 $w \cdot x + b$ ，应用到对应的logistic函数里，【这里为什么要叫和最大熵扯上关系，其实一般书上的数学解释都不从最大熵入手，而利用最大熵原理去推导也能够得到一致的形式，所以说softmax是正对形式上的叫法，而有的时候和maximum entropy有关是从数学上的角度来看这个模型的，两者不冲突。】利用梯度下降来优化损失。

如果是单纯是利用最大似然等公式写一个逻辑斯谛回归比较麻烦的一件事，用了TensorFlow之后可以看到被简化了非常多，基本不涉及到数学公式了。。。 (用BaiduML可能更夸张，连代码都不用写，给他数据选个模型就好了)

这里用的数据是MNIST，在<http://deeplearning.net/tutorial/logreg.html>可以看到这个数据集应用到逻辑斯谛回归的基本解释和代码，这里的数据和开始给出的tutorial的数据都是MNIST，但是否一样就不知道了，那个哥们的是有几个文件好像，logreg.html里使用的是Theano来完成的，也就是上一代的工具了。可以看看有助于理解。

拿完整的带注释的代码来讲：

```
1 # -*- encoding=utf-8 -*-
2 # author: Ian
3 # create on: 2015-11-17
4 # des: use TensorFlow to make log regression works
5
6 import gzip
7 import os
8 import sys
9 import numpy as np
10 import cPickle
11
12 import tensorflow as tf
13
14
15 class LogisticRegression(object):
16     def __init__(self):
17         self.X = tf.placeholder("float", [None, 784])
18         self.Y = tf.placeholder("float", [None, 10])
19
20         self.W = tf.Variable(tf.random_normal([28 * 28, 10], stddev=0.01))
21         self.b = tf.Variable(tf.zeros([10, ]))
22
23         self.model = self.create_model(self.X, self.W, self.b)
24
25         # logistic and cal error
26         self.cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(self.model, self.Y))
27
28         # gradient descent method to minimize error
29         self.train = tf.train.GradientDescentOptimizer(0.1).minimize(self.cost)
30         # calculate the max pos each row
31         self.predict = tf.argmax(self.model, 1)
32
33
34
35     def create_model(self, X, w, b):
36         # wx + b
37         return tf.add(tf.matmul(X, w), b)
38
39
40     def load_data(self):
```

TA的推荐

TA的最新馆藏

永远成功的秘密，就是每天淘汰自己

我们将永生还是灭绝？人工智能很...

我们将永生还是灭绝？人工智能很...

[转] 赞美的大能

他们还不信我要到几时呢？

基督徒的委身 【】



推荐阅读

更多

BetaCat 的前生后世

揪出bug！解析调试神经网络的技巧

深度学习计算模型中“门函数（Ga...

简易的深度学习框架Keras代码解析...

国外公司开发新型移动无线网pCell...

enum的用法

再谈：义和团史实（转）

是还没有受洗，还没有正式参加某...

帧缓存



```
41 f = gzip.open('mnist.pkl.gz', 'rb')
42 train_set, valid_set, test_set = cPickle.load(f)
43 f.close()
44 return train_set, valid_set, test_set
45
46
47 def dense_to_one_hot(self, labels_dense, num_classes=10):
48     # ont hot copy from https://github.com/nlintz/TensorFlow-Tutorials
49     # also can use sklearn preprocessing OneHotEncoder()
50     num_labels = labels_dense.shape[0]
51     index_offset = np.arange(num_labels) * num_classes
52     labels_one_hot = np.zeros((num_labels, num_classes))
53     labels_one_hot.flat[index_offset + labels_dense.ravel()] = 1
54     return labels_one_hot
55
56
57
58 def run(self):
59     train_set, valid_set, test_set = self.load_data()
60     train_X, train_Y = train_set
61     test_X, test_Y = test_set
62     train_Y = self.dense_to_one_hot(train_Y)
63     test_Y = self.dense_to_one_hot(test_Y)
64
65     sess = tf.Session()
66     init = tf.initialize_all_variables()
67     sess.run(init)
68
69     for i in range(100):
70         for start, end in zip(range(0, len(train_X), 128), range(128,
71             sess.run(self.train, feed_dict={self.X: train_X[start:end],
72                 print i, np.mean(np.argmax(test_Y, axis=1) == sess.run(self.
73
74     sess.close()
75
76
77 if __name__ == '__main__':
    lr_model = LogisticRegression()
    lr_model.run()
```

1 美亚保险官网	7 企业邮箱申请
2 美亚保险	8 北京口腔医院
3 公司邮箱	9 企业邮箱
4 中老年妈妈装	10 企业邮箱注册
5 英语学习	11 led亮化照明
6 钱爸爸理财	12 用英语介绍美国

onehot可能需要解释一下：  
onehot也是一位有效编码，有N个状态，我们使用N个位来表示对应的状态，例如，可能取值为[0,1,2]，使用100,010,001来表示，例如

```
1 >>> from sklearn import preprocessing
2 >>> enc = preprocessing.OneHotEncoder()
3 >>> enc.fit([[0, 0], [1, 1], [1, 2]])
4 >>> enc.transform([[0, 2]]).toarray()
5 array([[ 1.,  0.,  0.,  0.,  1.]])
```

第一个元素的取值为01，第二个为012，所以是2+3=5位。  
计算节点的流向也在刚刚开始给出。

上一篇：利用UOS指标来炒股UOS(终极波动指标)指标  
下一篇：TensorFlow之尝试

猜你喜欢



- 特种部队“手势语”图。哈哈
- 减轻工作量 Word长文档的制作技巧(多图详...]
- 查询:各省市最值得去的地方 (2011-09-1...
- 上半句要多浪漫有多浪漫，下半句要多贱有...
- 值得一生坚持的养生经
- 奇景盆景(没有做不到，只有想不到！)
- 微博每日精选10.28
- 《有趣的对联》
- WiFi万能钥匙使用教程
- 创业很美好吗？来看看这8个真相
- 一张图透露网上疯传的50个养生秘方，有这些...
- 优秀简历与普通简历的区别
- 西施死因之谜
- 防碰瓷完整攻略,有车一族必看
- 宋徽宗赵佶书法
- 大便不通引起便秘 按摩五穴位治疗便秘助你通...



python趣味编程



六房间直播



人有三个错误不能



《算命字典》举例



调查问卷与量表的



lol竞猜的首页



霍兰德职业兴趣测



苏果lol的首页



中国居民膳食营养



lol职业联赛2f的首

- 1 北京特价二手房急售,不限购..

2 看老股民如何从15万变300万

3 18岁少女开店 一年买车买房
- 1 美亚保险官网

2 美亚保险

3 公司邮箱

4 企业邮箱注册

5 英语学习

6 北京口腔医院

发表评论：

请 [登录](#) 或者 [注册](#) 后再进行评论

社交帐号登录：