



Terry.King的技术博客

目前在从事AI领域的研究, 精通iOS开发与 J2EE 开发

2017-06-29 • tensorflowone-hot

使用tensorflow api生成one-hot标签数据

在刚开始学习tensorflow的时候, 会有一个最简单的手写字符识别的程序供新手开始学习, 在tensorflow.example.tutorial.mnist中已经定义好了mnist的训练数据以及测试数据. 并且标签已经从原来的List变成了one-hot的二维矩阵的格式.看了源码的就知道mnist.input_data.read_data()这个方法中使用的是numpy中的方法来实现标签的one-hot矩阵化。那么如何使用tensorflow中自带的api来实现呢?下面我们就来一起看一下需要用到的api吧。

tf.expand_dims 方法 这个函数主要给矩阵或者数组增加一维°, 看代码可能更加清晰:

```
import tensorflow as tf
# 比如现在有一个列表
x_data = [1, 2, 3, 4]
x_data_expand = tf.expand_dims(x_data, 0) # x_data的shape是[4], 该函数
# 而对于[1, 4] 的矩阵加上x_data本身的数据, 那么可以猜想到x_data_expand = [
x_data_expand_axis1 = tf.expand_dims(x_data, axis=1) # x_data的shape
```

tf.concat(values, axis) 该函数用于将两个相同维度的数据进行合并, 如果指定



```
import tensorflow as tf
x_data = [[1, 2, 3], [4, 5, 6]]
y_data = [[7, 8, 9], [10, 11, 12]]

concat_result = tf.concat(values=[x_data, y_data], axis=0) # 这样的话
concat_result = tf.concat(values=[x_data, y_data], axis=1) # 这样的话
```

tf.sparse_to_dense() def sparse_to_dense(sparse_indices, output_shape, sparse_values, default_value=0, validate_indices=True, name=None): 该函数指定位置赋值, 并且生成一个维度为output_shape的矩阵;如果output_shape维度为1, 那么sparse_indices只能是一个列表, 如果output_shape为二维矩阵, 那么sparse_indices就可以是矩阵了。

比如如下代码:

```
import tensorflow as tf

sparse_indices = [1, 2, 6]
output_shape = tf.zeros([10]).shape
sparse_output = tf.sparse_to_dense(sparse_indices, output_shape, 2,

# 对于二维矩阵的填充也是一样的, 比如:
sparse_indices = [[0, 1], [2, 4], [4, 5], [6, 9]]
output_shape = tf.zeros([6, 10]).shape
sparse_output = tf.sparse_to_dense(sparse_indices, output_shape, 1,
```

```
[[0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 1]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]]
```



```
import tensorflow as tf

labels = [1, 3, 4, 8, 7, 5, 2, 9, 0, 8, 7]
labels_expand = tf.expand_dims(labels, axis=1) # 这样label_expand为

index_expand = tf.expand_dims(tf.range(len(labels)), axis=1) # 与1

concat_result = tf.concat(values=[index_expand, labels_expand], ax

one_hot = tf.sparse_to_dense(sparse_indices=concat_result, output_

session = tf.InteractiveSession()

print('labels_expand:{}'.format(session.run(labels_expand)))
print('index_expand:{}'.format(session.run(index_expand)))

print('concat_result:{}'.format(session.run(concat_result)))
print('one_hot_of_labels:{}'.format(session.run(one_hot)))
```

最后的结果如下打印:

```
python labels_expand:[[1] [3] [4] [8] [7]
[5] [2] [9] [0] [8] [7]] index_expand:[[ 0] [ 1] [ 2] [ 3] [
4] [ 5] [ 6] [ 7] [ 8] [ 9] [10]] concat_result:[[ 0 1] [ 1
3] [ 2 4] [ 3 8] [ 4 7] [ 5 5] [ 6 2] [ 7 9] [ 8 0] [ 9 8]
[10 7]] one_hot_of_labels:[[ 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.] [
0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.] [ 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.
0.] [ 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.] [ 0. 0. 0. 0. 0. 0. 0.
1. 0. 0. 0.] [ 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.] [ 0. 0. 1. 0. 0.
0. 0. 0. 0. 0. 0.] [ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.] [ 1. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0.] [ 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.] [ 0.
0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]]
```

这样就实现了labels的one-hot化。

使用Numpy来实现Label的one-hot化



```
labels = [1, 3, 4, 8, 7, 5, 2, 9, 0, 8, 7]
one_hot_index = np.arange(len(labels)) * 10 + labels

print ('one_hot_index:{}'.format(one_hot_index))

one_hot = np.zeros((len(labels), 10))
one_hot.flat[one_hot_index] = 1

print('one_hot:{}'.format(one_hot))
```

最近的文章

Spring中依赖注入的简单使用

依赖注入指容器负责创建和维护对象的起来关系.Spring IoC容器负责创建Bean并能够使用注入的方式供其他的类使用, 通过组合的方式实现类之间的解耦。Spring可以使用xml、注解以及java配置的方式实现依赖注入。因为xml实现依赖注入比较麻烦而且基本没有代码提示, 所以本篇主要记录较为简单的注解的方式以及java代码配置的方式。一、哪些注解能够声明类为Bean一般针对不同层的代码, 使用不同的注解来完成Bean的创建是比较明智的; 注解一方面能够实现依赖注入 另一方面增加代码的.....

2017-08-25 • Spring依赖注入

继续阅读

更早的文章

RunLoop官方解释

#前言从2014年8月到公司报到, 然后跟着导师维护Android SDK, 再到后面跟着另外一名大牛Lance(曾经在Nokia芬兰总部、三星韩国总部工作, 对于三维建模, 10年的OpenGL开发经验) 学习IOS的开发, 并且跟着项目走并且有人指导成长相对来说还是比较快的。但是对于接受新知识的能力还不足, 具体表现在英语非常的不好; 而当时学习IOS的时候Lance和其他同事也是强烈建议读官方文档的, 不过我却认为对于初学者 如果不是想紧随时代的步伐(当然这也不会是一个好的程序员), 首先需要的是入.....

2016-04-20 • RunloopThread Programming

继续阅读



0 Comments

Vno Jekyll

1 Login ▾

Recommend

Share

Sort by Best ▾



Start the discussion...

LOG IN WITH



OR SIGN UP WITH DISQUS (?)

Name

Be the first to comment.

ALSO ON VNO JEKYLL

Hello World - Vno

10 comments • 2 years ago



soap zhao — just test my comment

从零开始部署小程序服务器（腾讯云）

1 comment • 5 months ago



黄鸿亮 — 不错，看到你的详细记录，解决了我的一些疑问。

【openshift-3】添加Flask等第三方库+部署自己的应用

2 comments • a year ago



Apple King — greet

关于

1 comment • 9 months ago



郑桂森 — asdas

本站点采用知识共享 署名-非商业性使用-相同方式共享 4.0 国际 许可协议

由 Jekyll 于 2017-09-14 生成，感谢 Github 为本站提供稳定的代码托管服务

本站由 @terry.king 创建，采用 Vno - Jekyll 作为主题，您可以在 GitHub 找到本站源码 - © 2017