

学习记录tf

2020年6月2日 18:38

Conv2d

数据格式 NHWC 可改为 NCHW

[filter_height, filter_width, in_channels, out_channels]

来自 <https://blog.csdn.net/g0415shenw/article/details/86081330?utm_medium=distribute.pc_relevant.none-task-blog-baidujs-3>

Conv2d filter格式

Padding="VALID" or "SAME"

表示舍弃与填充

strides为4d tensor 中间似乎表示HW 其可能也是NHWC格式 与输入一一对应

The dimension order is determined by the value of `data_format`, see below for details272679

确定

与input含义一致

filter为外部给定的Var，即卷积核可外部给定

关于 名字空间

允许重名

如果要复用在名字后面添加/

纠正：

tfds中的iter 的getnext得到的不是一个数据源，只是一个表示“下一个tensor”的符号 只能运行一次

而要读取下一个对象则需要再次调用get_next

`tf.nn.batch_normalization`

`tf.layers.batch_normalization`

`tf.contrib.layers.batch_norm`

看样子 tf也是有封装的

依次从 nn 到layers 到contrib

(1) `tf.nn`：提供神经网络相关操作的支持，包括卷积操作（conv）、池化操作（pooling）、归一化、loss、分类操作、embedding、RNN、Evaluation。

(2) `tf.layers`：主要提供的高层的神经网络，主要和卷积相关的，个人感觉是对`tf.nn`的进一步封装，`tf.nn`会更底层一些。

(3) `tf.contrib`：`tf.contrib.layers`提供够将计算图中的 网络层、正则化、摘要操作、是构建计算图的高级操作，但是`tf.contrib`包含不稳定和实验代码，有可能以后API会改变。

- `mean, variance = tf.nn.moments(x, axes, name=None, keep_dims=False)`
 - 计算统计矩，`mean` 是一阶矩即均值，`variance` 则是二阶中心矩即方差，`axes=[0]`表示按列计算；
- `tf.nn.batch_normalization(x, mean, variance, offset, scale, variance_epsilon, name=None)`
 - `tf.nn.batch_norm_with_global_normalization(x, mean, variance, beta, gamma, variance_epsilon, scale_after_normalization, name=None)`;
 - `tf.nn.moments` 计算返回的 `mean` 和 `variance` 作为 `tf.nn.batch_normalization` 参数调用；

统计矩的计算，使用 `moments`函数，可以解决很多统计和概率方面的问题

`moments` 其返回 `mean`和`variance`

其中0表示按列计算而样本是每个一行（广义

也就是计算样本的统计数据

考虑每个样本是一行，那么按1计算就是计算样本内部的特征均值和方差，得到的是一列，每个样本都有一个统计结果，如果是多维样本，那么意义就不明了（大概按某一维的意义决定

```
def moments(x, axes=[0]):
```

注意，其中`axes`表示可以提供多个维度

那么也就是说可以按多个维度统计？那是不是说，统计完了一个维度，再统计另一个，反正方差和均值可以在任何维度上进行计算