

TensorFlow 2.0 自定义操作与建模方式

导师: GAUSS

目录

1/ 张量与操作

2/ 常用层

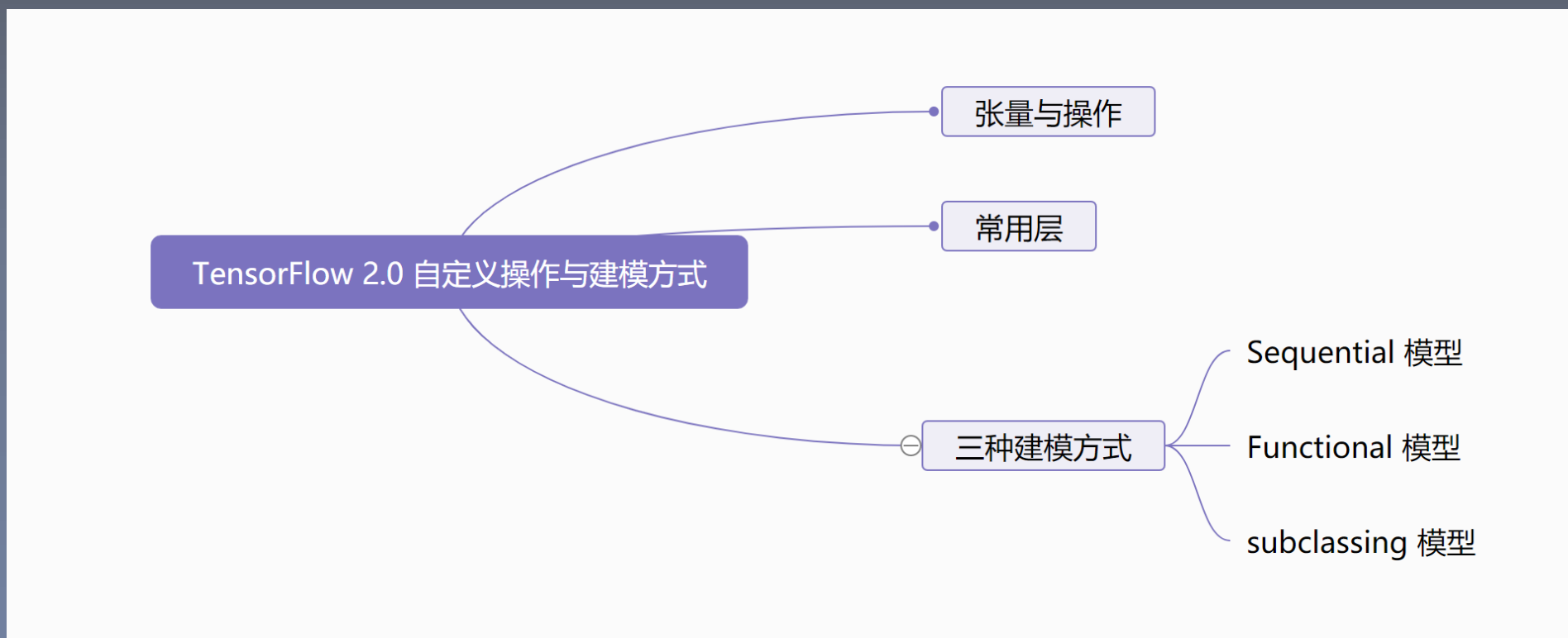
3/ 三种建模方式

知识树

Knowledge tree



深度之眼
deepshare.net



张量与操作

Tensor and Operation



张量

Tensor

TensorFlow 使用一种叫 **tensor** 的数据结构去定义所有的数据，我们可以把 **tensor** 看成是 **n 维的 array 或者 list**。在 TensorFlow 的各部分图形间流动传递的只能是**tensor**。

编写TensorFlow程序时，操纵并传递的主要对象是**tf.Tensor**:

- 一个**数据类型**（例如 float32, int32, 或string）
- 以及**shape**

张量

Tensor



深度之眼
deepshare.net

tensor 的维数

描述 tensor 的维数

Rank	Shape	Dimension number	Example
0	[]	0-D	A 0-D tensor. A scalar.
1	[D0]	1-D	A 1-D tensor with shape [5].
2	[D0, D1]	2-D	A 2-D tensor with shape [3, 4].
3	[D0, D1, D2]	3-D	A 3-D tensor with shape [1, 4, 3].
n	[D0, D1, ... Dn-1]	n-D	A tensor with shape [D0, D1, ... Dn-1].

张量

Tensor



深度之眼
deepshare.net



举个栗子！！

Numpy和tensorflow中的张量对比

Numpy	TensorFlow
<code>a = np.zeros((2,2)); b = np.ones((2,2))</code>	<code>a = tf.zeros((2,2)); b = tf.ones((2,2))</code>
<code>np.sum(b,axis=1)</code>	<code>tf.reduce_sum(a,axis=1)</code>
<code>a.shape</code>	<code>a.get_shape()</code>
<code>np.reshape(a,(1,4))</code>	<code>tf.reshape(a,(1,4))</code>
<code>b*5+1</code>	<code>b*5+1</code>
<code>np.dot(a,b)</code>	<code>tf.matmul(a,b)</code>
<code>a[0,0]; a[:,0]; a[0,:]</code>	<code>a[0,0]; a[:,0]; a[0,:]</code>

操作

operation

- 回顾第一节讲过的张量的操作
- tf.strings (常用于推荐算法场景、NLP场景)
- tf.debugging
- tf.dtypes
- tf.math
- tf.random
- tf.feature_column 参考: https://www.tensorflow.org/tutorials/structured_data/feature_columns

操作

operation

推荐看这些网站学习tensorflow的一些基础知识:

- https://lyhue1991.github.io/eat_tensorflow2_in_30_days/
- <https://zh.d2l.ai/>
- <https://www.tensorflow.org/api/r2.0>
- https://tf.wiki/zh_hans/



操作

operating



举个栗子！！

常用层

Commonly used layers

常用层

Commonly used layers

推荐:

tf.keras.layers

tf.nn

```
tf.keras.layers.LSTM(100)
tf.keras.layers.Dense(10)

tf.keras.layers.Dense(64, kernel_regularizer
=tf.keras.regularizers.l1(0.01))

tf.nn.softmax(b)
tf.keras.layers.Dense(64, activation=tf.nn.relu)
```

常用层

Commonly used layers

tf.nn: 底层的函数库，其他各种库可以说都是基于这个底层库来进行扩展的。

tf.keras.layers: 如果说tf.nn是轮子，那么tf.keras.layers可以说是汽车。tf.keras.layers是基于tf.nn的高度封装。



举个栗子！！

三种建模方式

Three modeling methods

三种建模方式

Three modeling methods

- Sequential model (顺序模型)
- Functional model (函数模型)
- Subclassing model (子类化模型)



顺序模型

Sequential model

Sequential方式构建模型

tf.keras.Sequential



举个栗子！！



函数式模型

Functional model

复杂模型怎么办？

- 多输入模型
- 多输出模型
- 具有共享图层（同一图层被调用多次）的模型，
- 具有非顺序数据流的模型（例如，残余连接）。
- 训练模型和**tf.keras.Sequential** 一样



举个栗子！！

https://www.tensorflow.org/guide/keras/functional#functional_api_weakness



子类模型

Subclassing model

- 通过子类化**tf.keras.Model**和定义自己的前向传播模型来构建完全可定制的模型。
- 和**eager execution**模式相辅相成。



举个栗子！！



三种构建模型的区别

模型	使用场景
Sequential model	对于顺序结构的模型，优先使用Sequential方法构建。
Functional model	如果模型有多输入或者多输出，或者模型需要共享权重，或者模型具有残差连接等非顺序结构，推荐使用函数式API进行创建。
Subclassing model	需要自定义层之间的传输、复杂模型。



本节小结

Summary

TensorFlow 2.0简介

	张量与操作	张量
		操作
	常用层	常用层
		Sequential model
	三种建模方式	Functional model
		Subclassing model

结语

——我 说——



**GAUSS老师个人公众号，主要分享NLP、
推荐、比赛实战相关知识！**





深度之眼
deepshare.net

联系我们:

电话: 18001992849

邮箱: service@deepshare.net

QQ: 2677693114



公众号



客服微信



GAUSS老师个人公众号, 主要分享NLP、推荐、比赛
实战相关知识!