

TensorFlow 2.0模型训练

导师: GAUSS

目录

- 1/ Keras版本模型训练
- 2/ 正向传播与反向传播
- 3/ 自动求导方法
- 4/ 自定义模型训练

知识树

Knowledge tree



深度之眼
deepshare.net

TensorFlow 2.0模型训练

Keras版本模型训练

损失函数、优化函数、指标函数

回调函数、学习率函数

多输入多输出

神经网络正向传播与反向传播简介

自动求导

tf.GradientTape的使用

自定义模型训练

使用GradientTape自定义训练模型

使用GradientTape自定义训练模型进阶（加入评估函数）

Keras版本模型训练

Keras版本模型训练

Keras版本模型训练

Keras版本模型训练

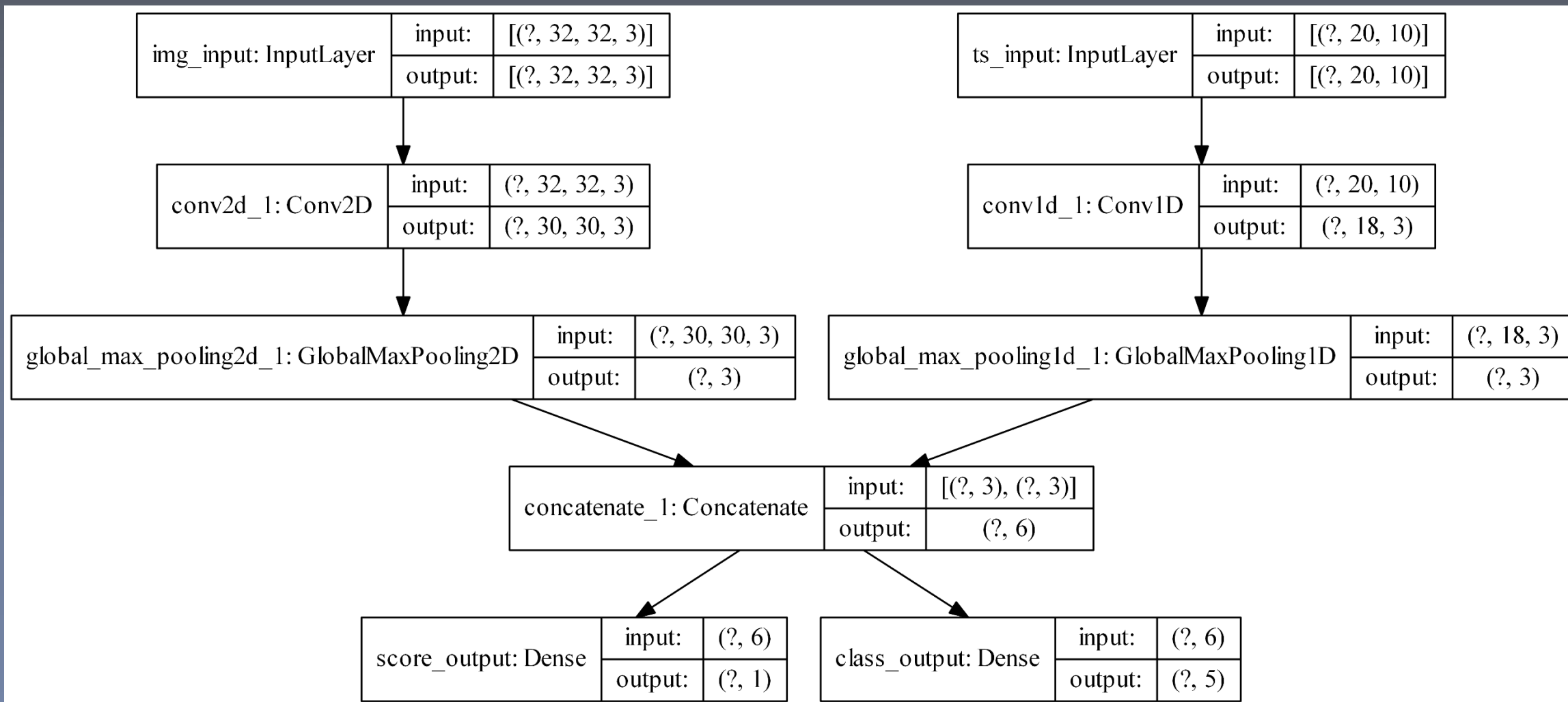
- 构建模型（顺序模型、函数式模型、子类模型）
- 模型训练： `model.fit()`
- 模型验证： `model.evaluate()`
- 模型预测： `model.predict()`
- 使用样本加权和类别加权

回调函数

- ModelCheckpoint: 定期保存模型。
- EarlyStopping: 当训练不再改善验证指标时, 停止培训。
- TensorBoard: 定期编写可在TensorBoard中可视化的模型日志 (更多详细信息, 请参见“可视化”部分)。
- CSVLogger: 将损失和指标数据流式传输到CSV文件。

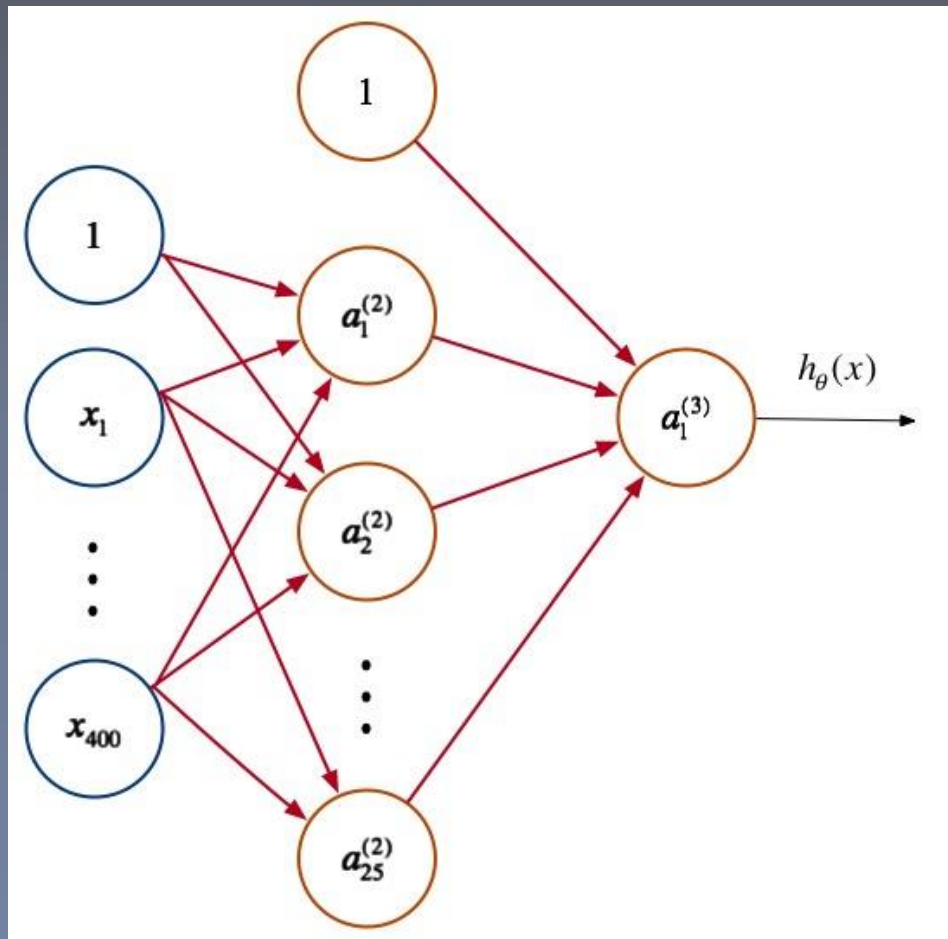


多输入、多输出模型



神经网络正向传播与 反向传播

正向传播



数据输入



隐藏层

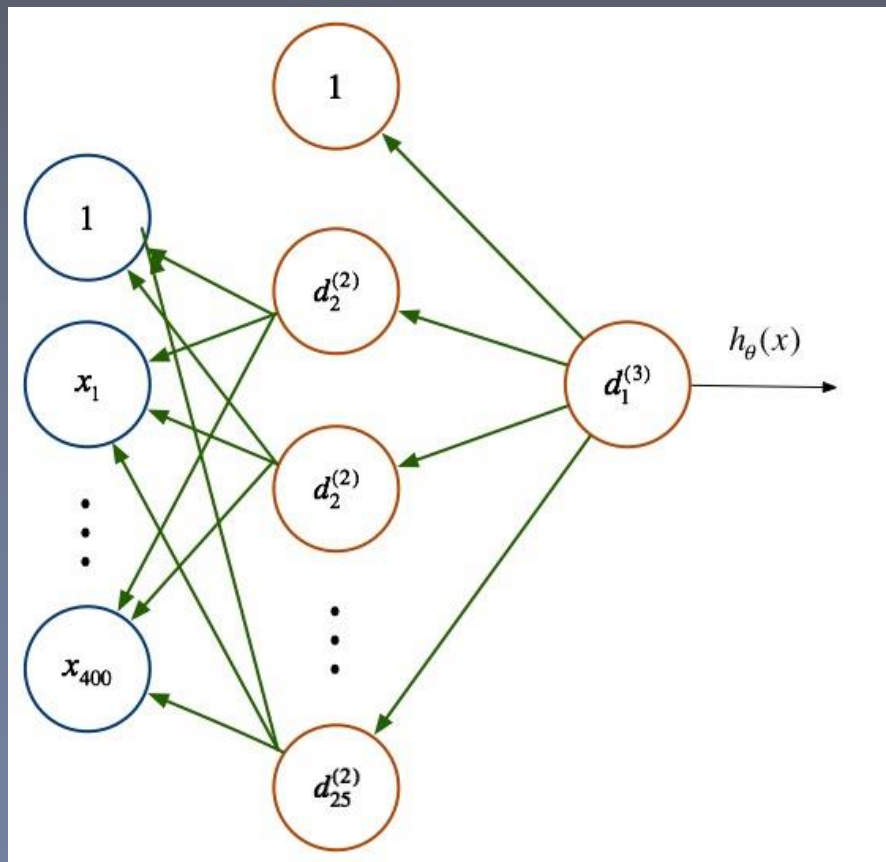


输出层



损失函数

反向传播



损失函数



参数求导



参数更新

自动求导机制

自动求导机制

梯度求解利器：**tf.GradientTape**

GradientTape是**eager模式**下计算梯度用的，而eager模式是TensorFlow 2.0的默认模式。

自动求导机制

举个例子：计算
 $y=x^2$ 在 $x = 3$ 时
的导数：

```
x = tf.constant(3.0)
with tf.GradientTape() as g:
    g.watch(x) # watch作用：确保某个tensor被tape追踪
    y = x * x
dy_dx = g.gradient(y, x)
# gradient作用：根据tape上面的上下文来计算某个或者某些tensor的梯度
#  $y' = 2*x = 2*3 = 6$ 
```




tf.GradientTape详解

```
tf.GradientTape(  
persistent=False,  
watch_accessed_variables=True  
)
```

- **persistent** : 用来指定新创建的gradient tape是否是可持续性的。默认是False, 意味着只能够调用一次gradient () 函数。
- **watch_accessed_variables**: 表明这个GradientTape是不是会自动追踪任何能被训练 (trainable) 的变量。

默认是True。要是为False的话, 意味着你需要手动去指定你想追踪的那些变量。

tf.GradientTape详解

gradient(target, sources)

作用：根据tape上面的上下文来计算某个或者某些tensor的梯度参数：

target: 被微分的Tensor，你可以理解为loss值(针对深度学习训练来说)

sources: Tensors 或者Variables列表（当然可以只有一个值）。

返回：

一个列表表示各个变量的梯度值，和source中的变量列表一一对应，表明这个变量的梯度。

自定义模型训练

自定义模型训练

tf.GradientTape

一般在网络中使用
时，**不需要显式调用watch函数**，使用默认设置，
GradientTape会
监控可训练变量。

```
loss_object = tf.keras.losses.CategoricalCrossentropy()
optimizer = tf.keras.optimizers.Adam()

with tf.GradientTape() as tape:
    predictions = model(data)
    loss = loss_object(labels, predictions)
gradients = tape.gradient(loss, model.trainable_variables)
optimizer.apply_gradients(zip(gradients, model.trainable_variables))
```


tf.GradientTape

apply_gradients(grads_and_vars, name=None)

作用：把计算出来的梯度更新到变量上面去。

参数含义：

grads_and_vars: (gradient, variable) 对的列表.

name: 操作名

自定义模型训练

- 用GradientTape自定义训练模型（案例2）
- 使用GradientTape自定义训练模型进阶（加入评估函数）（案例3）

本节小结

Summary

TensorFlow 2.0简介

Keras版本模型训练

指定损失函数、优化器、指标、回调函数等

多输入多输出

正向传播与反向传播简介

正向传播与反向传播简介

自动微分

自动微分的使用

自定义模型训练

使用GradientTape训练模型

训练进阶（加入评估函数）

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

Q Q：2677693114



公众号



客服微信

