

# TensorFlow 术语记录

## 1、tensor 相关

rank, type, shape, 各种 product, 需单独详解

## 2、keras

tensorflow 用 keras 充当上层接口, 且其地位还很主要, 主要特点用层组装模型, 可保存模型和训练参数, 支持js

## 3、tensorflowCore, webGl, wasm, tensorflowjs

tensorflowCore 核心库; webGl, wasm 为支持的两种 backend, tensorflowjs 使用的模型可用 tensorflowjs tensorflowjs\_converter 转换而来。

## 4、tf.function, graph, trace, eager mode, ConcreteFunction

为 tf 营造的几个概念:

**graph**: 为计算图, 其实对应也以前的概念“序列化”, 计算逻辑的序列化, 使得‘计算图’不依赖语言, 可保存和用其他语言‘反序列化’再计算。典型的 savedModel 必然是 tf.graph 模式的

**Function**: tf.function 可将一个普通的 python 函数修饰为一个 Function object, Function 包含若干 ConcreteFunctions。一位一个原来的 eager 模式和 eagerly py函数是及时执行的, 且是支持泛型的, 所以被 tf.function 修饰出来的 Function 具备 polymorphism (文档叫 polymorphism, 就是泛型概念), 每个具体参数被 type 固定之后的具体函数就是 ConcreteFunction, 一个 ConcreteFunction 对应具体的 graph

**Tracing**: 当 Function 被输入具体类型参数, 执行一遍之后, 一个 **ConcreteFunction** 和对应的 graph 才会生成, 这时候才可被 summary 和用 tensorBoard viewer 看到。

**Signature**: 函数的类型 (用输入输出类型定义), 既然 Function 是支持泛型的, 那么我也给他一个 Signature 把类型定义下来, 去除‘泛型’

总之。以上概念和传统的 序列化、泛型概念相近

```
1
2 // https://www.tensorflow.org/guide/function
3
```

- 4 A `tf.Graph` is the raw, language-agnostic, portable representation of a TensorFlow computation.
- 5 A `ConcreteFunction` wraps a `tf.Graph`.
- 6 A `Function` manages a cache of `ConcreteFunctions` and picks the right one for your inputs.
- 7 `tf.function` wraps a Python function, returning a `Function` object.
- 8 Tracing creates a `tf.Graph` and wraps it in a `ConcreteFunction`, also known as a trace.

## 5、Modules, layers, and models

用 core 直接用 Modules, keras 喜欢用 layers, and models, 这就几个概念有重复。如一个 Modules 就能 save 成 savedModel

## 6、 savedModel, pb, ptxt, layerModel, graphModel

savedModel: 文件夹下包含 pb 和其他参数;

savedModel\_keras: 包含 keras meta data 和训练参数可重新导入 keras, 继续训练 (savedModel 不行, 需要用 checkpoint)。

pb: protoBuffer

Ptxt: 文本可视化的 protoBuffer

```
1 tf.io.write_graph(  
2     infer.graph, './', 'out.ptxt', as_text=True  
3 )
```

layerModel 和 graphModel 是用转换工具转给tfjs用的。一般的 layerModel 包含训练参数可再次训练, graphModel 一般不行。

ps不要试图将 pb 转换为 ptxt 来查显示其 graph, 这样的操作的之后的 graph 看起来异常。另外如果要看一个 pb 的图, 转换为 tfjs 的 graphModel 之后, 用 netron 渲染产看 json 也是可选的工作路径。