

大神为你码上作答 | 10 个学好 TensorFlow 2.2 的理由

([原文链接](#))

为了更好地帮助开发者解决应用 TensorFlow 时遇到的问题，提供更丰富的技术指引、应用灵感，我们上线了“码上作答”专栏，每个月针对开发者集中关心的问题，邀请各行各业、具有丰富 TensorFlow 经验的资深专家分享经验和心得，同时接受大家的提问，为各位开发者答疑解惑。

接下来有请“码上作答”第一期嘉宾
彭靖田

大家好，作为一路陪伴和见证 TensorFlow 从 v0.7 到现在的开发者和老用户，很开心有机会能够作为第一期嘉宾参与 TensorFlow “码上作答”栏目，与各位分享我对 AI / TensorFlow 2.2 的学习经验和心得体会。

TensorFlow 码上作答 </> **+ Vol.1**

Q

TensorFlow 2.2 的学习经验和心得体会

A

彭靖田
Google Developer Expert
谷歌开发者专家



TensorFlow contributor, Kubeflow committer, 品览 Co-founder & CTO, 畅销书《深入理解 TensorFlow》、极客时间《TensorFlow 快速入门与实战》作者

这一期，我想尽可能系统化的谈谈对最新版本 TensorFlow 2.2 的看法。希望通过我整理总结的 10 个学好 TensorFlow 2.2 的理由，AI 新玩家们可以快速入门，TensorFlow 老师傅们能够获得一个更加全面的视角来认知 TensorFlow 及其生态。

欢迎大家在阅读后[向我提问](#)，我将在“码上作答”第一期的下篇为大家解答，记得持续关注！

理由 1：快速上手只需十分钟

TensorFlow 2.2 在易用性和简化 API 方面做了大量优化。对于新手，可以使用高层封装的 API 实现面向模型编程 (Model Oriented Programming, MOP)。我们以一个典型的 MNIST CNN 模型为例，展示 TensorFlow 2.2 的实现：

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10)
])
```

(实例化一个 CNN 模型)

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 128)	100480
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1290

=====
Total params: 101,770
Trainable params: 101,770
Non-trainable params: 0

(统计模型基础信息)

```
model.compile(optimizer="adam",
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
model.fit(x_train, y_train, epochs=5)
```

```
Epoch 1/5
1563/1563 [=====] - 3s 2ms/step - loss: 0.3296 - accuracy: 0.9153
Epoch 2/5
1563/1563 [=====] - 3s 2ms/step - loss: 0.3318 - accuracy: 0.9192
Epoch 3/5
1563/1563 [=====] - 3s 2ms/step - loss: 0.3156 - accuracy: 0.9213
Epoch 4/5
1563/1563 [=====] - 3s 2ms/step - loss: 0.3205 - accuracy: 0.9196
Epoch 5/5
1563/1563 [=====] - 3s 2ms/step - loss: 0.3061 - accuracy: 0.9238
```

(一行代码训练模型)

通过这个简单示例，我们得以管中窥豹。tf.keras 提供了一套面向模型的编程方法：实例化 (Sequential 和 Functional API)、统计 (Summary)、定义优化器和损失函数 (Compile)、模型训练 (Fit)、评估 (Evaluate)、预测 (Predict)、保存 (Save) 和加载 (Load) 等。这使得 TensorFlow 用户可以更加专注于模型开发和测试。

- [参考](#)

理由 2：快速迁移现成 Keras 项目

TensorFlow 2.2 实现了 Keras 所有接口的完全兼容，并且解决了原生 Keras 不支持分布式训练的问题。

因此，如果你已经在使用 Keras 实现的模型，可以尝试使用 `import tensorflow.keras as keras` 来进行快速迁移。如果你想要尝试扩展 Keras 分布式训练，也可以使用 `tf.distribute` 模块赋能你之前的训练代码。

- [参考](#)

理由 3：快速部署生产环境

很多老用户都对 TF Serving 这款生产级的 AI 服务框架赞不绝口。TensorFlow 2.2 不仅发扬了前辈的优良传统，同时将 `tf.keras` 训练的 HDF5 模型格式和 SavedModel 格式打通。你可以在使用 `model.save("saved_model")` 保存模型时选择 TF Serving 可加载的 SavedModel 目录。接着，你只需要使用 TensorFlow Serving 官方镜像 `tensorflow/serving` 便可以快速将 AI 模型部署到生产环境。

- [参考](#)

理由 4：快速实现模型分布式训练

TensorFlow 早在 v0.8 就已实现基于 PS-Worker 的分布式模型训练。

经过 3 年多的迭代和更新，TensorFlow 2.2 不仅提升了分布式训练的性能，而且通过内置的 6 种分布式训练策略，帮助不熟悉分布式计算的新同学用一行代码实现分布式训练，效率提升十分明显。

Training API	MirroredStrategy	TPUStrategy	MultiWorkerMirroredStrategy	CentralStorageStrategy	ParameterServerStrategy
Keras API	Supported	Experimental support	Experimental support	Experimental support	Supported planned post 2.0
Custom training loop	Experimental support	Experimental support	Support planned post 2.0	Support planned post 2.0	No support yet
Estimator API	Limited Support	Not supported	Limited Support	Limited Support	Limited Support

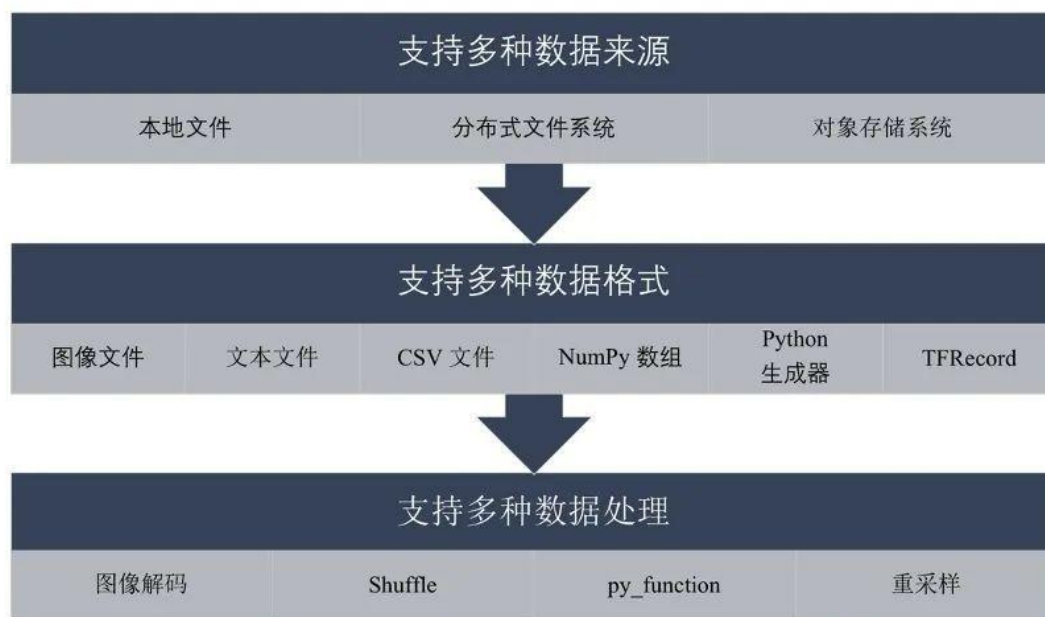
（部分内置的分布式训练策略）

- [参考](#)

理由 5：原生支持多元化的数据处理

`tf.data` 是 TensorFlow 2.2 推荐使用的数据处理 API，其强大的功能体现在对于多种数据来源、数据格式和数据处理方法的支持。对于个人开发者和小团队，可以使用本地文件和对象存储系统作为数据来源，对于拥有海量数据的大厂，则可以从分布式文件系统中分批次读取数据或搭建数据处理流水线。

从数据格式和内置数据处理方法的丰富度上，我们可以洞察到 TensorFlow 2.2 已不仅是一个处理计算机视觉或深度学习的框架，而是能够实现 NLP 和 GAN 等多种细分 AI 领域的通用 AI 框架。



(tf.data 模块介绍)

- [参考](#)

理由 6：原生支持经典分类模型

TensorFlow 2.2 原生支持了多种经典分类模型，包括重量级的 EfficientNet（最新 nightly 已支持）、DenseNet、ResNet、VGGNet 和轻量级的 MobileNet、Xception 等。你可以通过 `tf.keras.applications` 模块快速体验和使用这些经典模型。不论是在基于 ImageNet 上预训练的模型做迁移学习，亦或是从零到一的训练全新模型。

`densenet` module: DenseNet models for Keras.

`imagenet_utils` module: Utilities for ImageNet data preprocessing & prediction decoding.

`inception_resnet_v2` module: Inception-ResNet V2 model for Keras.

`inception_v3` module: Inception V3 model for Keras.

`mobilenet` module: MobileNet v1 models for Keras.

`mobilenet_v2` module: MobileNet v2 models for Keras.

`nasnet` module: NASNet-A models for Keras.

`resnet` module: ResNet models for Keras.

`resnet50` module: Public API for `tf.keras.applications.resnet50` namespace.

`resnet_v2` module: ResNet v2 models for Keras.

`vgg16` module: VGG16 model for Keras.

`vgg19` module: VGG19 model for Keras.

`xception` module: Xception V1 model for Keras.

(TensorFlow 2.2 内置分类模型)

- [参考](#)

理由 7：原生支持移动和物联网设备

在 TensorFlow 1.x 时代，SavedModel 转 tflite 格式部署到移动和物联网设备是可行的，但易用性不算很好。

TensorFlow 2.2 将 TF Lite 生态进一步整合到了原生框架中，并正式推出了 `tf.lite` 模块。现在，你可以直接调用 `tf.lite.TFLiteConverter` 实现 SavedModel、HDF5 和 ConcreteFunctions 等多种模型格式转 tflite 格式。

Converts a TensorFlow model into TensorFlow Lite model.

```
tf.lite.TFLiteConverter(  
    funcs, trackable_obj=None  
)
```

Example usage:

```
# Converting a SavedModel to a TensorFlow Lite model.  
converter = lite.TFLiteConverter.from_saved_model(saved_model_dir)  
tflite_model = converter.convert()  
  
# Converting a tf.Keras model to a TensorFlow Lite model.  
converter = lite.TFLiteConverter.from_keras_model(model)  
tflite_model = converter.convert()  
  
# Converting ConcreteFunctions to a TensorFlow Lite model.  
converter = lite.TFLiteConverter.from_concrete_functions([func])  
tflite_model = converter.convert()
```

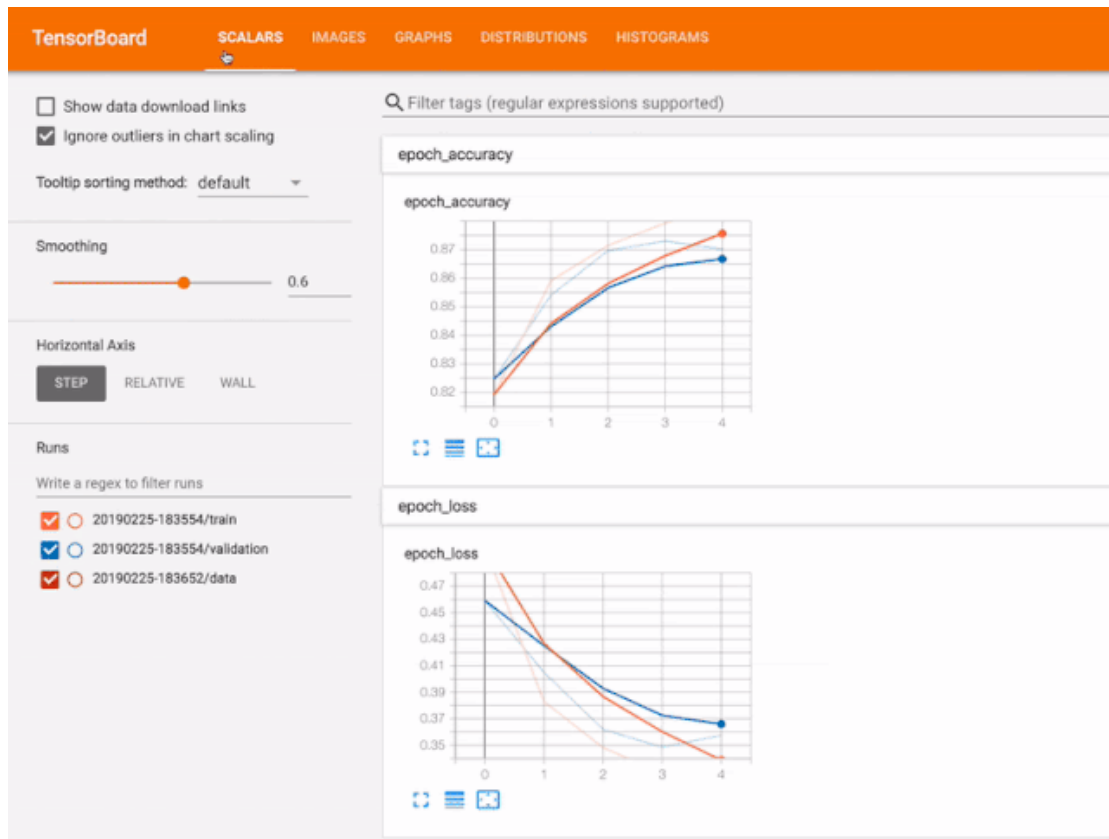
同时，你还可以通过 `tf.lite.Interpreter` 定义模型在物联网设备上运行时的预测接口。这使得 TensorFlow 2.2 真正实现框架原生支持 AIoT。

- [参考](#)

理由 8 原生支持可视化数据、模型和训练

在深度学习模型训练和评估过程中，海量数据、高维模型、多种指标的变化对于算法工程师来说是十分重要的信息。并且，如果能够有高效的手段获取其中的数据洞察，将会非常有助于提升模型训练效率和经验积累。

TensorBoard 可视化工具包自推出后一直广受好评，它的出现使得开发者能够更加直观的了解数据和模型的质量，以及关键指标分布和变化。在今年的 TensorFlow Dev Summit 上，官方团队发布了 [TensorBoard 在线版 \(https://tensorboard.dev/\)](https://tensorboard.dev/)。这使得 TensorFlow 用户不再需要独立安装 TensorBoard，而能够直接体验到 TensorFlow 生态中这一重要特性。



- [参考](#)

理由 9：前沿探索量子机器学习

量子机器学习 (QML) 是量子信息 (Quantum Information) 领域内新兴的子领域，其将量子计算的速度与机器学习所提供的学习和适应能力结合到了一起。量子机器学习可以分为两个部分：

1. 将从量子计算机和传统计算机获得的数据作为训练数据训练模型的机器学习算法。
2. 过量子力学的思想和模型来改进机器学习算法也属于量子机器学习的一部分。

量子机器学习自 1988 年首次提出细胞自动机的量子推广后，逐步发展出量子增强学习 (2008)、HHL 量子算法 (2009)、量子主成分分析 (2014)、量子 SVM 和贝叶斯网络上的量子推断 (2014)、量子推荐系统 (2016) 等经典算法的拓展。

虽然现阶段量子计算机和量子比特数量限制还无法大规模使用 QML，但作为最前沿的交叉学科，TensorFlow 2.2 也实现了对这一前沿领域的支持——[TensorFlow Quantum](#) (TFQ)。


```
# A hybrid quantum-classical model.
model = tf.keras.Sequential([
    # Quantum circuit data comes in inside of tensors.
    tf.keras.Input(shape=(), dtype=tf.dtypes.string),

    # Parametrized Quantum Circuit (PQC) provides output
    # data from the input circuits run on a quantum computer.
    tfq.layers.PQC(my_circuit, [cirq.Z(q1), cirq.X(q0)]),

    # Output data from quantum computer passed through model.
    tf.keras.layers.Dense(50)
])
```

（基于 TFQ 实现的混合量子机器学习模型）

- [参考](#)

理由 10：企业版助力数字化转型

许多企业用户对于 TensorFlow 快速发展带来的 API 版本变化想必是心有余悸。一个好消息是此类用户将由官方团队“兜底”。TensorFlow 企业版将提供长期的技术支持。

针对某些版本，谷歌将提供长达三年的补丁程序，所有补丁和错误修复将在 TensorFlow 主线代码存储库中提供。另外，谷歌还将向正在构建 AI 模型的公司提供来自 Google Cloud 和 TensorFlow 团队的工程协助。



（TensorFlow 企业版优势）

- [参考](#)

One More Thing

我希望不论是从业者，亦或是有志于投身 AI 的新同学，都能够勇敢踏出学习 AI 的第一步：快速上手一个 AI 框架——TensorFlow 2.2。同时，在 AI 理论积累和动手实战的基础上，持之以恒地学习，逐渐成长为一个玩转 AI 技术的 TFBoys / TFGirls。

最后，打个小广告：为了系统化和全方面的分享这几年在 AI 落地实战中积累的踩坑经验和 TensorFlow 2.2 的使用技巧，我在极客时间开设了课程《[TensorFlow 2 项目进阶实战](#)》。

同时，课程中的[项目实战代码](#)也已开源，欢迎有兴趣的同学关注和订阅！

向我提问吧！

看完这 10 个学好 TensorFlow 2.2 的理由，各位是否还有哪些疑惑或者问题，尤其是在 **AI 技术落地**和 **TensorFlow 快速入门**方面的？欢迎通过 TensorFlow 官方公众号向我提问！我们将筛选问题并在 6 月 18 日周四“码上作答”第一期的下集为大家做出解答，敬请关注。

阅读以上内容后，你可以通过以下方式继续学习：

- 阅读 TensorFlow 文件夹中的其他学习资料：

如果想回顾机器学习的基础知识，可以在文件夹里查阅《**开始你的 TensorFlow 之旅：【中文教程】机器学习从零到一**》，了解基本的计算机视觉概念，卷积神经网络以及其应用和工作原理，自己上手体验机器学习的乐趣。

如果你对 TensorFlow 已经有初步了解，想要进一步深入学习，就不能错过 TensorFlow 2.2 的重大更新。对此，我们准备了《**全面了解 TensorFlow 2.2**》向你介绍最新特性。

如果你已经是 TensorFlow 1.x 的用户，建议更新至最新版本，它将为你带来更流畅的机器学习体验。具体可以查阅我们准备的代码迁移指南《**教你如何迁移：将您的代码从 TensorFlow 1 迁移到 TensorFlow 2**》，你将得到详细的手把手指引。

同时，配合官网阅读，体验更佳：<https://tensorflow.google.cn/>

- 你还可以加入 TFUG 社区，认识更多优秀开发者，在社区中进步。
[TFUG，欢迎你的加入！](#)

我们为专业的 TensorFlow 开发者提供正式认证和证书，它不仅能够证明你的学习能力，同时也助力你的职业发展点亮 LinkedIn 技能。

- 关注 TensorFlow 官方微信公众号，**回复“认证”**，即可获得《TensorFlow 开发者认证候选人手册》，助你在机器学习道路上更进一步：



最后，请持续关注 TensorFlow 官方微信公众号，关注“码上作答”栏目，收获更多大神解答！

期待你顺利迈出学习 TensorFlow 的第一步！