

6-2-cn

1 在这节课我们讨论 TensorFlow。

2 数据处理系统提供大数据计算处理能力和应用开发平台。从计算架构的角度，将数据处理系统分为数据算法层、计算模型层、计算平台层、计算引擎层等。

与大数据相关的计算算法包括机器学习算法和数据挖掘算法。计算模型是指不同类型的大数据在不同场景下的处理方式，包括批处理、流计算、结构化数据

的大规模并发处理(MPP)模型、内存计算模型和数据流图模型。

计算平台和引擎提供各种开发套件和操作环境，我们选取 Spark Mllib 和 TensorFlow 为例，本节我们讨论 TensorFlow

3 TensorFlow 是如何工作的

TensorFlow 允许开发人员创建数据流图——描述数据如何通过图或一系列处理节点移动的结构。

图中的每个节点表示一个数学运算，节点之间的每个连接或边是一个多维数据数组，或张量。

张量是 TensorFlow 的核心数据单位，在本质上是一个任意维的数组。可用的张量类型包括常数、变量、张量占位符和稀疏张量

Stochastic gradient descent (often abbreviated SGD)

TensorFlow 通过 Python 语言为程序员提供了这一切。

Python 很容易学习和使用，并且提供了方便的方法来表示如何将高级抽象耦合在一起。

TensorFlow 中的节点和张量都是 Python 对象，而 TensorFlow 应用程序本身也是 Python 应用程序。

它使用 Python 提供了一个方便的前端 API，用于使用框架构建应用程序，同时高性能 C++ 中执行这些应用程序。

然而，实际的数学操作不是在 Python 中执行的。

通过 TensorFlow 可用的转换库是用高性能的 C++ 二进制文件编写的。

Python 只是在各个部分之间引导通信，并提供高级编程抽象来将它们挂钩在一起。

TensorFlow 应用程序可以在任何方便的目标上运行：本地机器、云中的集群、iOS 和 Android 设备、CPU 或 GPU。

如果你使用谷歌自己的云，你可以在谷歌的自定义 TensorFlow 处理单元(TPU)硅上运行 TensorFlow 以进一步加速。

TensorFlow 创建的结果模型可以部署在大多数设备上，用于预测。

4 TensorFlow 已经发展成为世界上最受欢迎和广泛采用的 ML 平台之一

让我们看一段视频来了解更多关于 TensorFlow 的知识

5 TensorFlow 2.0 架构

TensorFlow 2.0 注重简单和易于使用，包括以下更新：

使用 Keras 容易建立模型，执行迅速。

在任何平台上的生产中进行健壮的模式部署。

为研究准备的强大的实验

这些 API 被打包成一个全面的平台，支持从训练到部署的机器学习工作流程。

6

Keras 是一个面向机器学习的用户友好的 API 标准，它将成为用于构建和训练模型的核心高级 API。

Keras API 使开始使用 TensorFlow 变得容易。

重要的是，Keras 提供了几种模型构建 API (Sequential、Functional 和 Subclassing)，因此您可以为您的项目选择正确的抽象级别。

TensorFlow 2.0 的实现包含了增强功能，包括

1. 急切执行，立即迭代

- 2、直观调试,
- 3 tf. 数据, 用于构建可伸缩的输入管道。

7 简单的模型构建

下面是一个工作流示例:

- 使用 `tf.data` 加载数据。
- 使用 `tf.data` 创建的输入管道读取训练数据。
- 特征特性, 例如使用 `tf.feature_column` 进行描述 bucketing 和 feature crosses 特性交叉。
- 还支持从内存数据(例如 NumPy)进行方便的输入。

8 构建、训练和验证模型

用 `tf` 构建、训练和验证模型。 `keras`, 或者使用 `Premade Estimators`。

`Keras` 与 `TensorFlow` 的其他部分紧密集成, 所以你可以随时访问 `TensorFlow` 的特性。

还可以直接使用一组标准打包模型(例如, 线性或逻辑回归、梯度增强树、随机森林)(使用 `tf.estimator` API 实现)。

如果您不打算从头开始训练一个模型, 那么您很快就可以使用迁移学习来使用 `TensorFlow Hub` 中的模块训练 `Keras` 或 `Estimator` 模型。

使用即时执行运行和调试, 然后使用 `tf.` 函数的好处。

默认情况下, `TensorFlow 2.0` 运行时具有快速执行功能, 以方便使用和平稳调试。

此外, `tf. function` 函数注释透明地将你的 `Python` 程序翻译成 `TensorFlow` 图。

这个过程保留了 1 的所有优点。基于 `TensorFlow` 图的执行:性能优化,

远程执行以及轻松序列化、导出和部署的能力, 同时增加了用简单 `Python` 表示程序的灵活性和易用性。

9 使用分布式策略进行分布式训练。

对于大型 ML 训练任务, 分布策略 API 使得在不同硬件配置上分发和训练模型变得容易, 而无需更改模型定义。 由于 `TensorFlow` 提供了一系列硬件加速器的支持, 如 `cpu`、`gpu` 和 `tpu`, (张量处理器)

您可以将培训工作负载分配到单节点/多加速器, 以及多节点/多加速器配置, 包括 `TPU Pods`。

尽管此 API 支持各种集群配置, 但提供了在 on-prem 或云环境中的 `Kubernetes` 集群上部署培训的模板。

10 模型存储

导出到 `SavedModel` 组件。

`TensorFlow` 在 `SavedModel` 上进行标准化, 作为与 `TensorFlow Serving`、`TensorFlow Lite`、`TensorFlow.js`、`TensorFlow Hub` 等进行交换的格式。

11 在任何平台上的生产环境中进行健壮模型部署

在任何平台上的生产中进行健壮模型部署

`TensorFlow` 一直为生产提供直接路径。无论是在服务器, 边缘设备, 还是网络上,

`TensorFlow` 允许您轻松训练和部署模型, 无论您使用什么语言或平台。

在 `TensorFlow 2.0` 中, 通过标准化交换格式和对齐 api, 跨平台和组件的兼容性得到了提高。

TensorFlow Serving: 一个 `TensorFlow` 库, 允许模型通过 `HTTP/REST` 或 `gRPC`/协议缓冲区服务或者应用

TensorFlow Lite: `TensorFlow` 针对移动和嵌入式设备的轻量级解决方案, 提供了在 `Android`、`iOS` 和嵌入式系统(如树莓派和 `Edge TPU`)上部署模型的能力。

TensorFlow.js: 允许在 `JavaScript` 环境中部署模型, 例如通过 `Node.js` 在 `web` 浏览器或服务器端。

`TensorFlow.js` 还支持用 `JavaScript` 定义模型, 并使用类似于 `keras` 的 API 直接在 `web` 浏览器中进行训练。

12

这张图展示了通用计算、机器学习和深度学习的典型平台, `Tensorflow` 可以用于 ML 和 DL。

`TensorFlow` 是一个强大的研究实验

TensorFlow 使从概念到代码、从模型到发布的新想法变得容易。

TensorFlow 2.0 包含了许多特性，可以在不牺牲速度或性能的情况下定义和训练最先进的模型：

Keras 函数 API 和模型子类 API:允许创建复杂的拓扑，包括使用剩余层、自定义多输入/输出模型和命令式写入的前向传递。

自定义训练逻辑:用 tf 对梯度计算进行细粒度控制。GradientTape tf.custom_gradient。

为了获得更多的灵活性和控制，底层 TensorFlow API 始终可用，并与更高级别的抽象一起工作，以实现完全可定制的逻辑。

TensorFlow 2.0 带来了几个新功能，允许研究人员和高级用户使用丰富的扩展进行实验

比如粗糙张量，张量流量概率，张量张量，以及更多有待公布的内容。

除了这些功能，TensorFlow 还提供了快速执行，便于原型设计和调试，分布策略 API 和 AutoGraph 进行大规模培训，并支持 tpu，使 TensorFlow 2.0 成为一个易于使用、可定制和高度可扩展的平台，用于进行最先进的 ML 研究并将研究转化为生产管道。

13 动手实验

为了使用 TensorFlow 进行管理，设计了一个实验，

波士顿住房预测步骤实验如图所示。

波士顿住房预测包括准备、导入数据、数据可视化、建立模型、训练模型和结果可视化。

14 本节我们学习了 TensorFlow，今天内容就到这里谢谢大家。