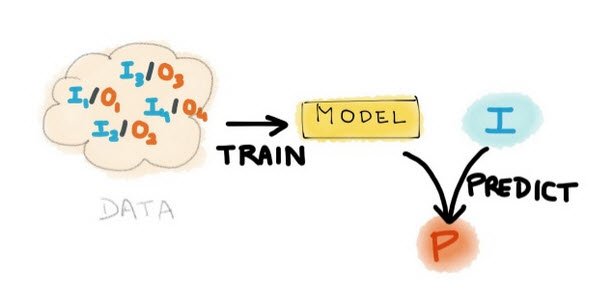


第01课：为什么要学原理和公式推导



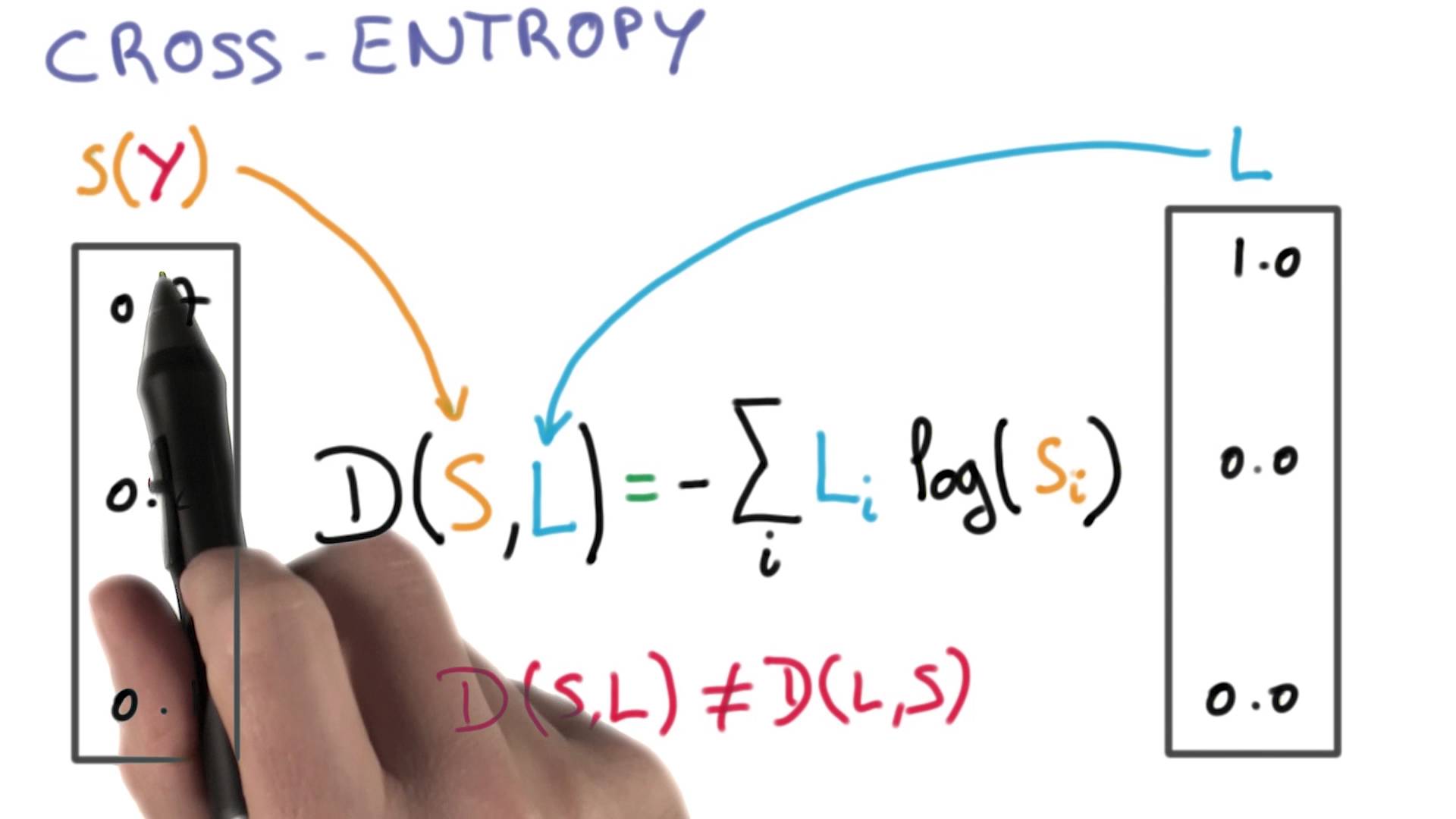
各位同学，从本节开始，我们就进入《21 天入门机器学习》的正课了。第一个学习主题是“绪论”，学时两天，共有 4 篇文章（01 课~04 课），对应一次打卡任务，大家可根据自己的时间来安排阅读和提交笔记。如有任何疑问，也可以在学习笔记中向我提问。下面我们就进入正题：

机器学习最直接的应用，就是利用模型解决实际业务中的问题。

本课所讲解的几个经典模型，均是前辈在机器学习发展的几十年间所总结出的、解决特定问题的固定模式，并且已在实践中得到很好证明。

学会这些模型，一则可以以它们为载体理解“机器学习”是一种怎样的机制；二则掌握了模型，也就掌握了当前许多实际问题的有效解决方案。

### 学模型就要学公式推导吗？



在实际工作中，我们要运用一种模型，其实有很多现成的算法库、学习框架。只要将相应的数据输入工具、框架中，用几行代码指定模型的类型和参数，就能自动计算出结果。

既然如此，何必再去学其中的原理，一步步推导让人头晕的数学公式？

对于这个问题，首先给出我的意见：

****机器学习的原理和数学推导一定要学！****

此处且举个直观的例子：

工具就像是武器，学会使用某种工具，只是学会了这种武器最基本的招式和套路。而理论学习即策略学习，决定了在未来真实对战中，遇到对手攻击时，你选取哪些招式套路，如何组合起来去迎敌。

反过来讲，如果根本不学模型原理，只是把一个个应用场景背诵下来，需要的时候直接把模型当黑盒使用——这样做我们能学到什么？

我们将学到：

1. 算法库的安装；
2. 库函数的调用；
3. 数据的 I/O 转换。

这和调用任何一个封装好的 API（无论其功能）有什么区别？和调用同事撰写的模块接口又有什么不同？

学会这几件事能让你相对他人产生什么样的壁垒？作为一个原本非 AI 领域的开发者，难道因为会安装几个支持库，会调几个接口就身价倍增，就成为机器学习工程师了？

### 学习原理的必要性

回过头来，我们从正面来看学习原理的必要性。

#### 功利层面

咱们先来看看最直接的用处。

##### ****面试会考****

最起码，找工作的时候，但凡稍微靠谱点的企业，****在面试“机器学习工程师”时，一定会问到模型原理和推导过程****！所问到的模型，随着时间推移会越来越复杂。

三四年前甚至更早，企业技术面试时，大多会问线性回归。这两年，已经基本从逻辑回归开始问了。再过一两年是否会问支持向量机，不得而知，但理论考察势必会越来越难。

真想入这行，为了面试也得学其中的原理。

##### ****老板会问****

在日常编码中，可能确实只是调 API 而已。

很多时候，在决定使用哪个工具、框架，调用哪个模型算法后，你还需要向老板、合作方甚至客户解释其中的缘由。

以前谁负责哪个模块都是自己搞定所有事，你怎么还想让别人给你标注数据呀？

花费这么多时间和人力训练出的模型，怎么连个 DSAT 都 Fix 不了啊？

既然再多投入几倍资源也达不到 95% 以上的正确率，为什么不干脆直接用 Rule-Base 来解决？

……

Manager、Tech Lead、PM 都有可能围绕机器学习/深度学习投入产出比，提出各种问题。要在工作中运用这些技术，首先要说服他们。这个时候，原理就派上用场了。

##### ****同事会质疑****

对你应用 ML/DL 的疑问，不仅来自上司，很多时候也来自于同级别的同事。

相对于老板对性价比的关注，同事可能更关心技术细节——质疑新框架/工具/模型/算法，与之前的实施方案相比，在功能、性能等方面的差异。

别人都用 TensorFlow，你为什么要用 Caffe 呢？

以前这个分类器，我们用逻辑回归挺好的，你为什么非要换成 RNN 呢？

用这个谱聚类做数据预处理，归根到底不还是利用词袋模型算词频，比直接计算 tf-idf 做排序能好多少呢？

……

到了这个层面，只简单概括说说原理已经不够了，需要深入细节做对比：

* 不同模型的特质、适用场景，对当前数据的匹配程度；
* 不同算法对算力和时间的消耗；
* 不同框架对软硬件的需求和并行化的力度；

了解了这些，才有资格讨论技术。

【文-1】

#### 实用层面

当然，在日常工作中可以完全不理会 Peer 的质疑，对于 Boss 的决定也可以照单全收绝无二话。Engineer 嘛，只要埋头干活就好了。但总得把活儿干好吧。

作为一个机器学习工程师，把活干好的****基本标准****是：针对技术需求，提供高质量模型。

****再高一个层次****则是：针对业务需求，提供高质量的解决方案。

##### ****优化模型****

机器学习工程师又被戏称为调参工程师。

其所要做的工作就是在限定的数据上、规定时间内，为具体技术需求（比如训练一个分类器）提供性能尽量高、消耗资源尽量少的模型。

选特征、调超参、换模型，称为调参工程师的三板斧，要想有章法地使用它们，理论基础还是必不可少的。

至此，上面****【文-1】****处所描述内容，不再是和他人论战的“弹药”，而成了工作步骤的指导。

* 评判模型性能的指标有哪些，如何计算？
* 正在使用的模型是怎么工作的？
* 这些超参数是什么含义，调整它们会产生哪些影响？
* 特征选取有哪些原则、方法可运用？

如果连以上问题都不了解，又怎么优化模型？

##### ****针对实际问题定向建模****

成为一名合格的调参工程师，固然可以在 AI 领域占据一席之地，但对于业务和团队而言，仍然是个可有可无的角色。

****真正创造价值的，从来都是解决实际问题的人。****

这些经典的模型、算法，是前人在解决实际问题中所研究出的、具备通行性的解决方案。它们被广泛应用，是因为所解决的目标问题总会持续出现。

然而，新问题也会随着新需求不断涌现，现有成果可能无法解决这些新问题，调参工程师这时将束手无策。

但对于理论知识扎实的机器学习工程师来说，他们完全有可能针对具体业务问题，构造出目标函数，甚至开发出符合自身软硬件资源特点的求解算法。

到了这一步，即使还使用现成工具，也不是靠 Google 一下 Best Practice，以及复制粘贴下代码就能解决的了。必须具备理论基础和数学层面的建模能力才行。

[邀请好友一起学，获得 25% 返现奖励](https://gitbook.cn/m/mazi/columns/5bc6ac7442d7d32f50f19a98/topics/5bc70dfb42d7d32f50f1a46a)

[IMG_259](https://gitbook.cn/m/mazi/comp/column?columnId=5bc6ac7442d7d32f50f19a98%26tag=2#catalog)