

第07课：模型的获取和改进

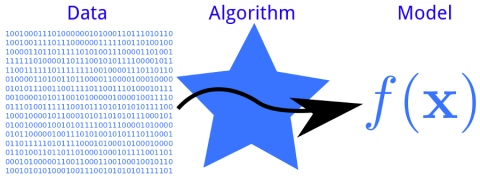
前面两篇文章，我们从直观的角度讲解了机器学习的最基本原理，并且解释了机器学习三要素：数据、算法和模型。

“应用机器学习技术”这件事情，具体到微观的行为，其实就是：使用机器学习模型来预测数据，得到预测结果。然后，预测结果可能会作为下一步业务逻辑的依据。

要使用机器学习模型，首先要获得它。在有了数据和算法的情况下，我们需要通过一个过程来获得模型，这个过程就叫做：训练。

### 获取模型的过程

在前面讲述三要素时我们已经讲过：数据 + 算法 => 模型



获得模型的过程——训练——是将算法应用到数据上进行运算的过程。

笼统而言，为了构建一个模型，我们需要经历以下步骤。

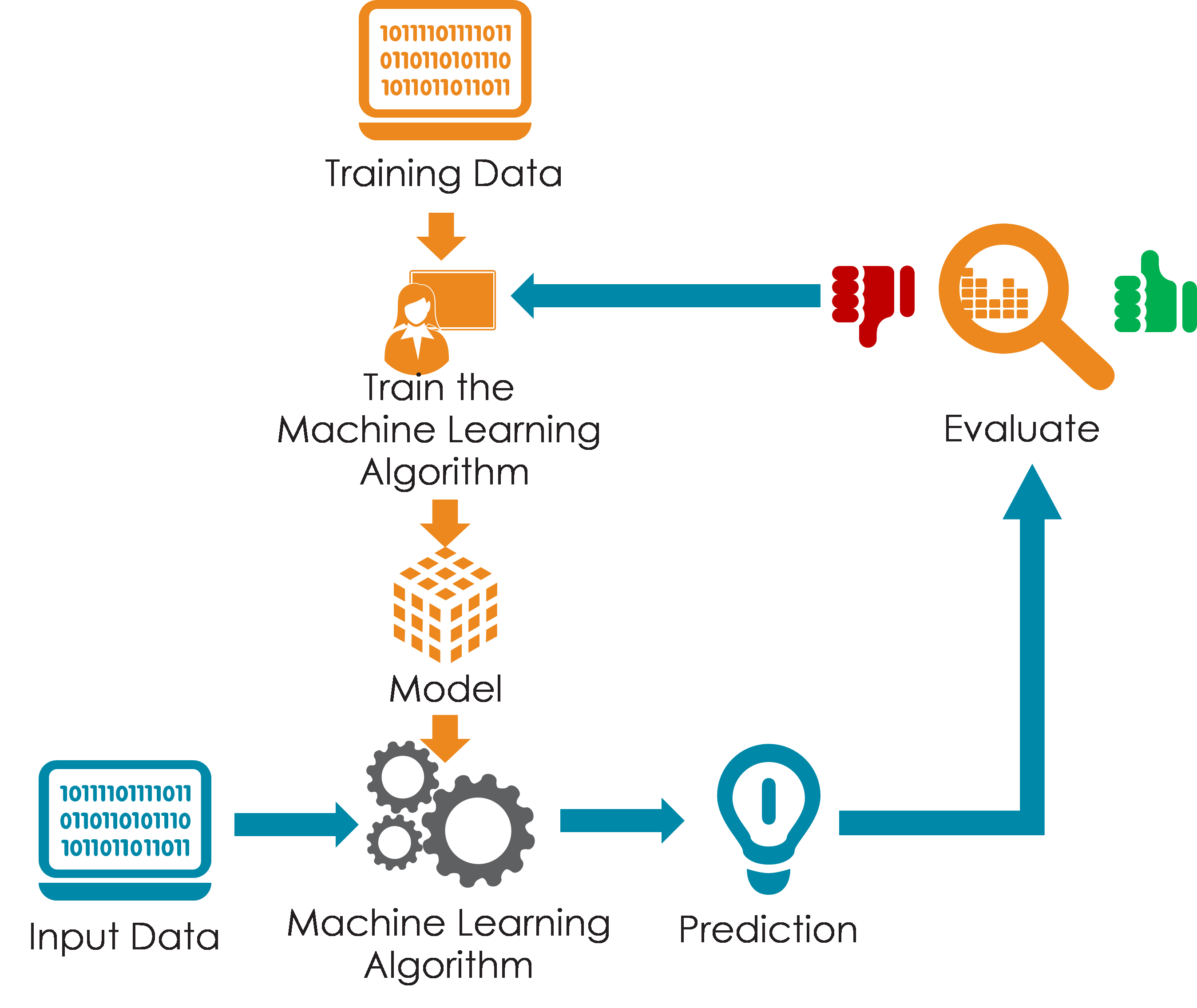
Step-1：数据准备。

* + Step-1.1 数据预处理：收集数据、清洗数据、标注数据。
  + Step-1.2 构建数据的向量空间模型（将文本、图片、音频、视频等格式的数据转换为向量）。
  + Step-1.3 将构建好向量空间模型的数据分为训练集、验证集和测试集。

Step-2：训练——将训练集输入给训练程序，进行运算。训练程序的核心是算法，所有输入的向量化数据都会按该训练程序所依据的算法进行运算。训练程序输出的结果，就是模型。

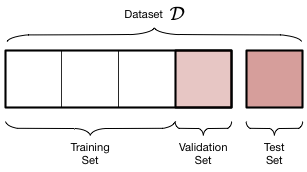
Step-3：测试——将测试集数据输入给训练获得的模型，得到预测结果；再将预测结果与这些数据原本预期的结果进行比较。

然后按一定规则计算模型质量的衡量指标，比如 Precision、Recall、F1Score 等（后面章节会介绍），根据指标的数值来衡量当前模型的质量。



### 训练集、验证集和测试集

Step-1.3 中提到我们要将全部数据分割成训练集、验证集和测试集。这些集合都是做什么的呢？



****训练集（Train Set）****：用来做训练的数据的集合。

****验证集（Validation Set）****：在训练的过程中，每个训练轮次结束后用来验证当前模型性能，为进一步优化模型提供参考的数据的集合。

****测试集（Test Set）****：用来测试的数据的集合，用于检验最终得出的模型的性能。

****每个集合都应当是独立的，和另外两个没有重叠****。

训练集是训练过程的基础，而验证和测试集则是在不同阶段用来评价训练结果的。

在个别情况下验证集和测试集可以合二为一，不过当然最好还是分开。验证可以理解为一个小规模的对于已经生成模型的测试，只不过已生成模型处理的数据不是测试集而是验证集。

这三个集合可以从同一份标注数据中随机选取。三者的比例可以是训练集：验证集：测试集=2：1：1，也可以是7：1：2。总之，**训练集占大头**。有些情况下（整体数据量不大，模型又相对简单时），验证集和测试集也可以合二为一。

### 训练的过程

Step-2 训练，又可以细化为几个步骤。

Step-2.1： 编写训练程序。

* + Step-2.1.1： 选择模型类型；
  + Step-2.1.2： 选择优化算法；
  + Step-2.1.3： 根据模型类型和算法编写程序。

Step-2.2： 训练 -> 获得临时模型。

Step-2.3： 在训练集上运行临时模型，获得训练集预测结果。

Step-2.4： 在验证集上运行临时模型，获得验证集预测结果。

Step-2.5： 综合参照 Step-2.3 和 Step-2.4 的预测结果，改进模型。

Step-2.6： Step-2.2 到 Step-2.5 反复迭代，直至获得让我们满意，或者已经无法继续优化的模型。

### 改进模型

我们来看看上面的 Step-2.5 改进模型。

对照机器学习三要素，模型的优化可以从三个方面来进行：数据、算法和模型。

#### 数据

机器学习的模型质量往往和训练数据有直接的关系。

这种关系首先表现在数量上——同一个模型，在其他条件完全不变的情况下，用 1000 条 Training Data 和 100000 条 Training Data 来训练，正常情况下，结果差异会非常明显。

****大量的高质量训练数据，是提高模型质量的最有效手段。****

当然，对于有监督学习而言，****标注****是一个****痛点****，通常我们可以用来训练的数据量相当有限。

在****有限的数据****上，我们能做些什么来尽量提高其质量呢？大概有如下手段：

对数据进行归一化（Normalization，又译作正规化、标准化）等操作。

采用 Bootstrap 等采样方法处理有限的训练/测试数据，以达到更好的运算效果。

根据业务进行特征选取：从业务角度区分输入数据包含的特征，并理解这些特征对结果的贡献。

#### 调参（算法）

很多机器学习工程师被戏称为调参工程师。这是因为调参，是模型训练过程中必不可少又非常重要的一步。

大家已经知道，我们训练模型的目的就是为了得到模型对应公式中的若干参数。这些参数是训练过程的输出，并不需要我们来调。

除了这些参数外，还有一类被称为超参数的参数，例如用梯度下降方法学习 LR 模型时的步长（Alpha），用 BFGS 方法学习 Linear-chain CRF 时的权重（w）等。

****超参数是需要模型训练者自己来设置和调整的****。

调参本身有点像一个完整的 Project，需要经历:

1. 制定目标
2. 制定策略
3. 执行
4. 验证
5. 调整策略 -> 3

这样一个完整的过程，其中 3-5 往往要经由多次迭代。

调参有许多现成的方法可循（比如 Grid Search），应用这些方法，使得制定和调整“调参策略”有章可循，也可以减少许多工作量。

算法库本身对于各个算法的超参数也都有默认值，第一次上手尝试训练模型，可以从默认值开始，一般也能得出结果。

机器学习模型的训练，得出结果容易，得出优质结果（高质量的模型）难。

调参这一步很多时候没有一定之规。特别是在超参数比较多的时候，如何组合调参简直可以称为一门艺术。

遇到这样的时候，只能具体情况具体分析，很多时候其实就是尝试，能否试出一个好的结果确实有一定运气的成分存在。但是，乱调一气最后能够得到好结果的可能性几乎为零。

只有当调参工程师对于模型的理论原理和算法的执行细节有深入了解的时候，才有可能找到正确的调参方向。

方向对了，即使找不到最优解，也总能找到次优解。这也就是我们要特别认真地学习理论的原因。

#### 模型类型选择

有的时候，训练数据已经确定，而某个模型再怎么调参，都无法满足在某个特定指标上的要求。那就只好换个模型试试了。

比如，对于某个分类问题，Logistic Regression 不行，可以换 Decision Tree 或者 SVM 试试。

现在还有很多深度学习的模型逐步投入使用。不过一般情况下，DL 模型（CNN、DNN、RNN、LSTM 等等）对于训练数据的需求比我们今天讲的统计学习模型要高至少一个量级。在训练数据不足的情况下，DL 模型很可能性能更差。

而且 DL 模型的训练时间普遍较长，可解释性极弱，一旦出现问题，很难 Debug。因此再次建议大家：在应用场景下，不要迷信 Cutting-edge Technology，无论工具还是方法，选对的，别选贵的。

[邀请好友一起学，获得 25% 返现奖励](https://gitbook.cn/m/mazi/columns/5bc6ac7442d7d32f50f19a98/topics/5bf28104fd72950cafdccd7d)

[IMG_260](https://gitbook.cn/m/mazi/comp/column?columnId=5bc6ac7442d7d32f50f19a98%26tag=2#catalog)