Machine Learning Yearning 翻译

作者：Andrew Ng 翻译：张雨阳

2017.1

Contents

Chapter 1

机器学习策略

机器学习是很多重要应用的基础，包括网页搜索，垃圾邮件识别，语音识别，推荐系统等等。我假设，你和你的团队，正在做一个机器学习的应用，并且你想快速进步，这本书会很好地帮助到你。

以识别猫咪图片为例

你想给广大铲屎官们提供可爱的猫咪图片，你用神经网络搭建了一个计算机视觉系统来检测图片中的猫咪。

Figure 1.1: 猫咪图片

但悲剧的是，你的算法准确率不够好，你急切地想要改进你的猫咪检测器，你打算怎么做？

你的团队可能会有如下的想法：

- 获取更多的数据：收集更多的喵咪图片。
- 收集多样化的训练数据集。比如，图片中的猫咪处于特殊的位置；不同花色的猫咪；拍照时使用了不同设置的照片。
- 继续多训练一会，进行更多的梯度下降训练
- 采用更大的神经网络，增加隐藏层或节点数量或参数
- 尝试更小的神经网络
- 尝试增加正则项，如 L2 正则化
- 改变神经网络结构，如激励函数和隐藏节点数量等
- ...

如果你在这些可能的方向上作了较好得取舍，你会在识别猫咪上取得成功。如果取舍得不好，你会白白浪费几个月。那么你要如何进步呢？

这本书将会告诉你，大部分机器学习问题都会留下有用和没用的线索，学会辨别这些线索，将会节省很多开发时间。

Chapter 2

如何使用这本书来帮助你的团队

在读完这本书后，你会对如何设置一个机器学习项目的技术方向有较深的理解。

但是你的团队成员可能会不理解你为什么要推荐这样一个特别的方向。可能你会想要你的团队来定义一个评估标准，但是它们并不令人信服。你如何劝说他们呢？

这就是我为什么要让每一章尽可能地短：这样你就可以打印出来，并且让你的同事在读完几页后理解你。

一点点小的改变会给你的团队带来巨大的产出。我希望可以通过帮助你的团队微小的改变，让你成为队伍里的英雄。

Chapter 3

定义和符号

如果你之前上过机器学习的课程，比如我在 Mooc 上的机器学习课程，或者你对监督学习有应用经验，那么你会很容易理解这章。

我假设你对**监督学习**已经很熟悉了：使用成对的已标记过的训练样本 (x,y) 学习一个从 x 到 y 的映射（函数）。常用的监督学习算法包括线性回归，逻辑回归，神经网络等等。有很多形式的机器学习，但监督学习仍是现在的主流。

我会经常提及到神经网络，也就是深度学习，你只需要对这些后文提到的概念有个基本的了解。

如果你不熟悉以上这些概念，可以去看我在 Courusera 上的课程的前三周 `http://ml-class.org`

Chapter 4

大规模数据促进机器学习发展

深度学习（神经网络）的概念已经存在了数十年，为什么现在这些概念如此流行呢？

产生近年来进步的最大两个驱动因素是：

- **大量可以使用的数据**。人们现在越来越依赖于电子设备（笔记本，手机等等）。这些电子设备为我们训练算法产生了大量的数据。
- **大规模计算的能力**。我们不久前才能够训练大规模的神经网络，以利用拥有大量数据的优势。

具体地说，即使你增加更多的数据，传统的机器学习算法的表现也会进入瓶颈，如逻辑回归。这也就是说，它的学习曲线会趋于平坦，学习效果甚至不会随着训练数据量的增加而变好。

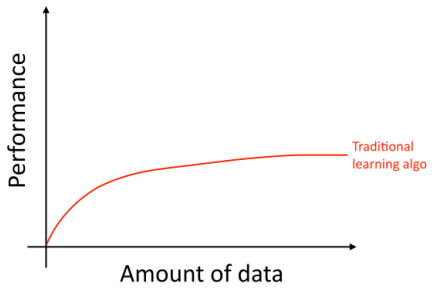

Figure 4.1: 传统机器学习模型学习表现变化

传统的机器学习算法并不会知道如何使用我们现在拥有的数据。

如果你训练一个小的神经网络在同样的监督学习任务上，你会得到轻微的改善。

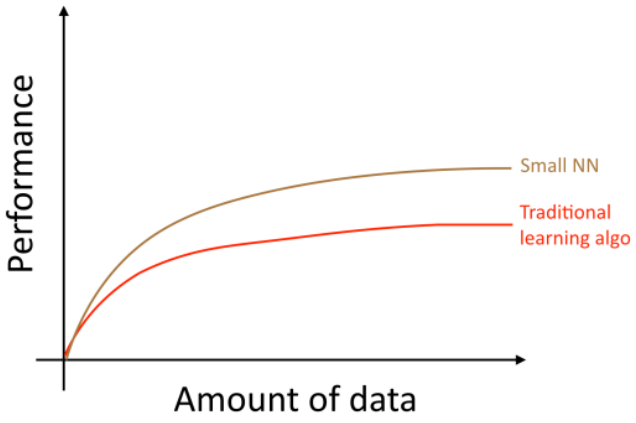

Figure 4.2: 神经网络的学习表现。这个图表展示了 NN 在小数据集下也会做的更好（横轴前半部分）。这和 NN 在大数据集中表现优秀并不具有一致性。在小数据集中，传统算法可能做得更好，也可能不会做得更好，这依赖于手工设计的特征。例如，如果你只有 20 个训练样本，那么你使用 logistic regression 或使用 neural network 可能没有多大区别；手工设计的特征将会比算法的选择产生更大的影响。但是如果你拥有一百万的数据量，我更倾向于使用神经网络。

这里的“小的神经网络”是指只有很少数量的隐藏层、节点或参数的神经网络。最终你会发现，随着你的神经网络越来越大，算法的表现会越来越好：

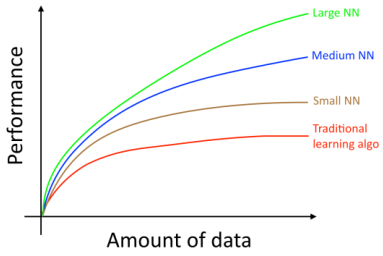

Figure 4.3: 大规模神经网络学习表现

因此，当你（1）训练一个较大的神经网络来获得绿色的学习表

现，（2）使用大量的数据时，你会得到最好的表现。

神经网络的其他细节也同样重要，这方面有很多创新。但是就目前来说，改善一个算法表现的主要途径还是：（1）训练更大的神经网络，（2）获取更多的数据。

完成上述目标的过程依然十分复杂，这本书会讨论这类细节。我们将会从既对传统机器学习算法有用又对神经网络有用的通用策略出发，实现对建立深度学习系统最先进的策略。

Chapter 5

开发集和测试集

让我们回想一下最初的例子，用户在你的 app 上面上传各种照片，你想自动的找出猫咪的图片。

你的团队通过下载猫咪的图片得到很多含猫咪的图片（正例样本），和很多不含猫咪的图片（反例样本），将数据集分成 70% 的训练集和 30% 的测试集。通过这些数据，我们建立了一个在训练集和测试集上表现还可以的猫咪探测器。

但是当你把分类器放在 app 中时，你会发现这个分类器的表现非常平庸！

发生了什么？

你发现用户上传的图片相比于之前下载收集的训练数据图片有很大的不同：用户上传的图片是手机拍摄的，与之相比分辨率更低，很模糊，同时也没有理想的灯光。因为你的训练集和测试集是由网上的图片构成的，你的算法在小的手机照片上泛化能力很差。

在大数据时代来临前，我们通常采用三七开来划分测试集与训练集。这个小技巧很有用，但它并不够好，尤其在训练数据分布（网上下载的图片）与你所关心的数据分布（手机拍照上传的图片）不同的情况下。

我们一般这样定义：

- **训练集** 用来训练算法的数据集。
- **开发集** 用来根据你的模型，调试参数，选择特征或做一些别的改善的数据集。有些时候这也称作 holdout 交叉验证集。

- **测试集** 你用来评估算法的表现，但并不对学习算法或者参数做任何改变。

一旦你定义了开发集和测试集，你的团队才能做出一些改善的尝试，比如尝试使用不同的学习算法参数，来看看哪个表现更好。开发集和测试集令你的团队能更快了解学习算法是如何工作的。

换句话说，**开发集和测试集的目的就是指导你的团队向着最重要的改变的方向来建立你的机器学习系统。**

所以，你应该这样做：

选择开发集和测试集来反映你的数据。

总的来说，你的测试集不该仅单单包括 30% 的数据，特别是你期望得到的数据与你的训练数据不同时。

如果你还没有发布你的 app，你也不会有任何的用户，因此你也不会有能够准确反应现实的数据。但是，你可以尝试来近似。比如，让你的朋友拍一些照片给你。一旦你的 app 发布了，你就可以用用户数据来更新你的开发集和测试集。

如果你真的没有任何途径来获得你未来所期待的近似数据，或许你可以先使用网络图片。但是你应该清楚，这样会降低系统的泛化能力。

我们需要决定投资多少去获取好的开发集和测试集。但是不要假设你的训练集分布和测试集有相同的分布。尝试去挑选能反映你最终想要表现很好的数据作为测试样本，而不是你遇到的任何训练数据。