

Dive into Deep Learning 学习笔记

任务:

1、对深度学习技术及其应用进行总结，用自己的话，重点谈自己的理解。

学习内容:

- 1、数据操作、数据处理、线性代数、微积分、自动微分和概率的基本知识。
- 2、线性回归和 softmax 回归的从零开始的编程实现和简洁编程实现。
- 3、主要理解和掌握线性回归和 softmax 回归的理论原理。

学习笔记:

深度学习是机器学习的一个子领域，指使用多层神经网络来对数据进行建模和分析。与传统的机器学习方法相比，深度学习能够自动提取特征，而不需要为特征工程花费大量时间和精力。在我看来，深度学习的核心价值在于其自动特征学习的能力（反向传播-参数优化）和在大规模数据集上的优越性能（切分 Batch 使用 GPU 并行运算）。相较于传统方法，深度学习降低了模型设计的门槛，使得非专家也能利用先进技术（已经成熟的简洁实现工具）。

而我这周的任务最重要的主要还是学习了回归相关的知识，它是一个能为一个或多个自变量与因变量之间关系建模的一类方法。我们的目的就是数据切分 batch 和相关处理之后，通过前向传播得到预测，计算损失函数，并对损失函数求梯度，得到优化方向，再利用例如 SGD 的优化器对模型参数进行优化，这里的参数优化指权重 w 和偏置 b ，这两个一般是直接随机正态分布所获得的，我们的目的就是找到他们的最优值使损失函数最小化。由于这里的数据只考虑的我们收集的数据，所以我们需要添加噪声项来考虑观测误差等带来的影响。

对于我们在回归问题上最常用的 SGD 优化器，我们可以将梯度下降理解为小球滚下山直到最低点，其中梯度计算就是获得下山的最佳路线，学习率就是滚动的距离，如果学习率过大可能会在最低点徘徊-也就是震荡（可以通过动量法解决），直接进行计算可能方向很正确，但是滚动地很慢，如果随机选取 Batch 利用并行运算的话，滚动下降地会更快，哪怕不是整个数据集的最佳路线。

模型设置的批量大小会影响学习率的选择：较小的批量大小可能需要较小的学习率，以避免模型在更新时出现不稳定；较大的批量大小通常能够支持较大的学习率，因为它们的梯度估计更为准确，较少受到噪声的影响。

切分 Batch 就可以利用 GPU 并行运算的优势，对每个样本并行地进行模型计算，且每个样本损失函数的梯度也可以被并行计算，提升运行效率。其中小批量计算出的梯度不一定对很适合整个数据集，但是 GPU 并行计算快，学习率相对小，风险小；大批量计算出的梯度更适合整个数据集，但是计算相对较慢，且因为学习率大，风险也相对大一些（在优化参数时， $lr * grad$ 记得除以 batch_size 来稳定更新的大小，避免异常波动）。

分类问题可以用独热编码处理数据，交叉熵损失一般用于分类问题，数学理解更多的写在了中文版的电子书上，其中对数和导数设计很巧妙，方便了运算，又在独热编码非 0 即 1 的情况下简化了优化参数时的运算。

原理和代码其实不算很难，但是困难的是很多深层的知识点都隐藏在表面的知识点之下，眼睛扫过理解一下感觉学懂了，但实际的原理深究却发现有所不知，因此反反复复看了很多遍内容，通过 GPT 的问答和学习网上大牛的视频教程，对很多本来已经知道的知识点有了恍然大悟的理解，这四章的学习也算是对深度学习基础知识的搭建了，在不断的反思和重新学习中，感觉深度学习有着极大的学习魅力。

我每天的学习与理解也会同步到 Github 上留痕，老师您如果想看的话地址在这：[WDMX820/Deep-Learning-Notes: 深度学习笔记汇总（Google Colab 运行）\(github.com\)](https://github.com/WDMX820/Deep-Learning-Notes)。