机器学习笔记整理

2024 年 5 月 29 日

摘要

这份文档是我们在学习机器学习相关知识的同时，用来记录讨论所得的文档。这份记录将会和《动手学深度学习》这本书的章节相匹配。

1 Chapter 2.1-2.3 May 24

1. 在 PyTorch 库中使用的数据类型张量，除了可以简单地进行按元素的加减乘除等运算之外，在不同形状的张量之间还可以通过来按元素操作。广播机制的可以自动扩展张量为相等大小。如果遵守以下规则，则两个 tensor 是“可广播的”:

   (a) 每个 tensor 至少有一个维度;

   (b) 遍历 tensor 所有维度时，从末尾随开始遍历，两个 tensor 存在下列情况:
   tensor 维度相等。
   tensor 维度不等且其中一个维度为 1。
   tensor 维度不等且其中一个维度不存在。

   如果两个 tensor 是“可广播的”，则计算过程中，如果两个 tensor 的维度不同，则在维度较小的 tensor 的前面增加维度，使它们维度相等。对于每个维度，计算结果的维度值取两个 tensor 中较大的那个值。tensor 扩展维度是指将数值进行复制到扩展后的维度上。
   具体例子可以参考: https://zhuanlan.zhihu.com/p/402163854

2. 在使用 PyTorch 库中的张量数据类型时，需要注意节省内存。在机器学习中，我们可能有数百兆的参数，并且在一秒内多次更新所有参数。通常情况下，我们希望原地执行这些更新。原地更新的办法有:

   (a) 用切片表示法将操作的结果分配给先前分配的数组，如

   ```
   Z = torch.zeros_like(Y)
   Z[:] = X + Y
   ```

   这样不会为变量 Z 分配新的内存，而是会改变原有地址处的内容。

   (b) 使用 X[:] = X + Y 或 X += Y，而非 X = X + Y，最后一种办法会改变变量 X 所指向的地址，相当于重新找了一个内容空间来存放求和的结果。