task1 report

张艺洋

January 2026

1 简介

本次任务 1 是在原先的作业 1 基础上完成的，进行了一些优化和新的实验。

2 网络结构

在任务 1 训练中我使用的网络结构如下：

表 1: LeNet 网络结构

层类型	输出尺寸	参数	备注
Conv1 + BN1 + ReLU	$6 \times 28 \times 28$	3×6×5×5 + 6	卷积核 5×5, BN
MaxPool1	$6 \times 14 \times 14$	-	2×2 池化
Conv2 + BN2 + ReLU	$16 \times 10 \times 10$	6×16×5×5 + 16	卷积核 5×5, BN
MaxPool2	$16 \times 5 \times 5$	-	2×2 池化
Flatten	400	-	展平为向量
FC1 + BN3 + ReLU	120	400×120 + 120	全连接, BN
FC2 + BN4 + ReLU	84	120×84 + 84	全连接, BN
FC3	10	84×10 + 10	输出层

3 实验结果

在 AutoDL 租用的 RTX 4090(24GB)GPU 上进行训练与测试。在 batch_size=64, num_workers=4, epochs=10, lr=0.001, momentum=0.9 的超参数下使用带动量 SGD 优化器进行训练 10 个 epoch，平均每个 epoch 的运行时间为 3.29s。训练 loss 图如图 1。

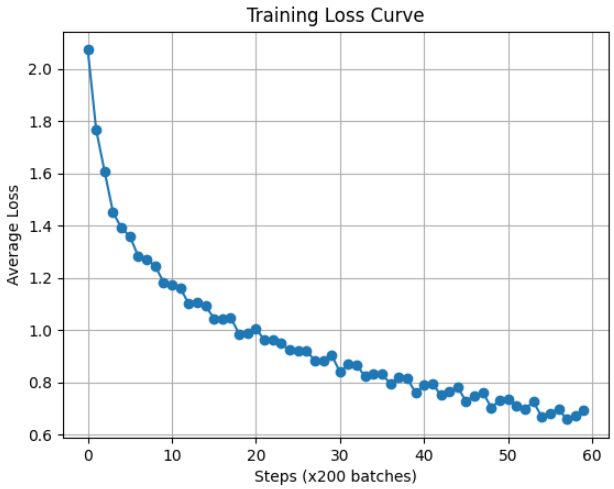

图 1: loss 曲线

使用训练好的模型在 Cifar10 的测试集上进行测试，得到的准确率为 65.39 % 。

4 讨论

看到 hw1 中的写的讨论，感觉通过几个月的实践又有了一些新的感悟，故进行一些补充讨论。

4.1 讨论 1

标准的随机梯度下降在陡峭的区域上震荡很大、在平缓的区域震荡又很小，导致要么收敛慢要么受困于局部最优解。momentum 是一种对梯度的“累计”方式，使得训练的过程中函数不仅根据当前位置梯度，也根据之前的“梯度记忆”，进行参数更新。这样就能避免在某一方向梯度过大时“走上错路”，也能在过度平缓的时候冲出“局部盆地”。一言以蔽之，momentum 就是系统对 gradient 的一种“记忆方式”，即“先验”知识，用来调整对“后验”梯度的接受程度。

4.2 讨论 2

1. 在一定范围内，momentum 越大，损失曲线的收敛速度越快。
2. 在一定范围内，增大 momentum，能够有效减小损失函数的震荡。
3. 当 momentum=0.99 时，训练出现了严重的欠拟合现象，很可能是因为动量过大，梯度下降法带来的调整被过度削弱，导致优化效果不佳。

4.3 讨论 3

在本学期的作业 6 中，我接触了 Adam 优化器。相比于 SGD+momentum 的组合，Adam 优化器通过同时记录梯度的一阶矩和二阶矩，实现了比前者更稳定的优化策略，也带来了更强的数值稳定性和更快的收敛速度。

然而在本学期一些较复杂模型的训练过程中，我发现 Adam 在处理一些大参数量的复杂模型时，往往在最终的准确率上面难以与 SGD+momentum 匹敌，主要是因为它太“保守”了，过多的约束导致其往往受困于局部最优，在“探索”方面的效果比较差。因此在实际使用的时候，如果模型相对简单或者追求较快的收敛速度，则使用 Adam，如果模型较为复杂、对准确率要求高的话，就应当使用 SGD+momentum，Adam 虽然整合了前者，但它不是万能的。