

Lab 1 实验报告

姓名: 马梓洋

学号: 21300180097

2025 年 5 月 1 日

1 最优结果展示

实验设置学习率 0.01, 在 2k 步和 4k 步 (分别对应 5 epoch 和 10 epoch) 条件下训练。

代码见 GitHub 仓库¹。表现最好的是单使用 Moment GD 优化的训练模型, 我使用了 4k 步训练, 准确率达到 97.05%。在相同步数下, SGD 优化器的模型发生了过拟合 (见8), 而在 2k 步的情况下二者的区别并不明显, 甚至 MomentGD 还稍弱 (见4)

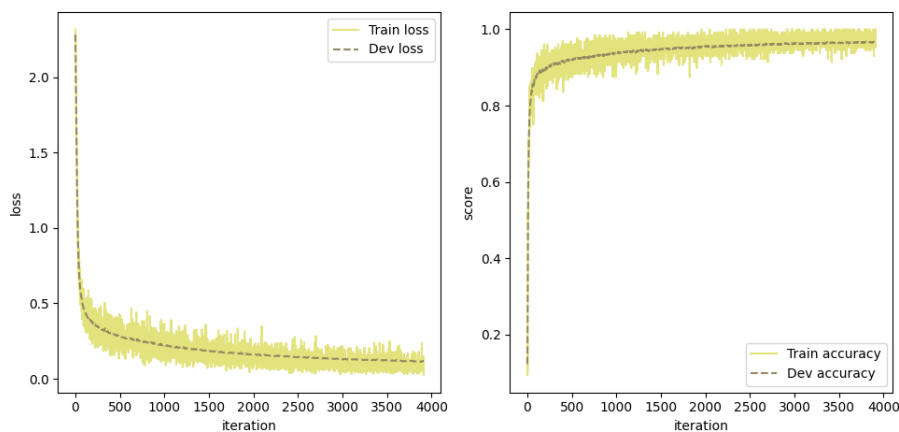


图 1: Moment GD 的训练 loss 和精度

¹代码仓库: https://github.com/Zephyr-ziyang/Deeplearning_25_S/tree/main/PJ1

2 实验过程

第一次运行代码 出现了经典的问题，参数未更新 (valid loss 没变化)，于是我写了一些 debug(现在运行模型会有参数更新检测了)，最后发现是 optimizer.SGD 出现的问题

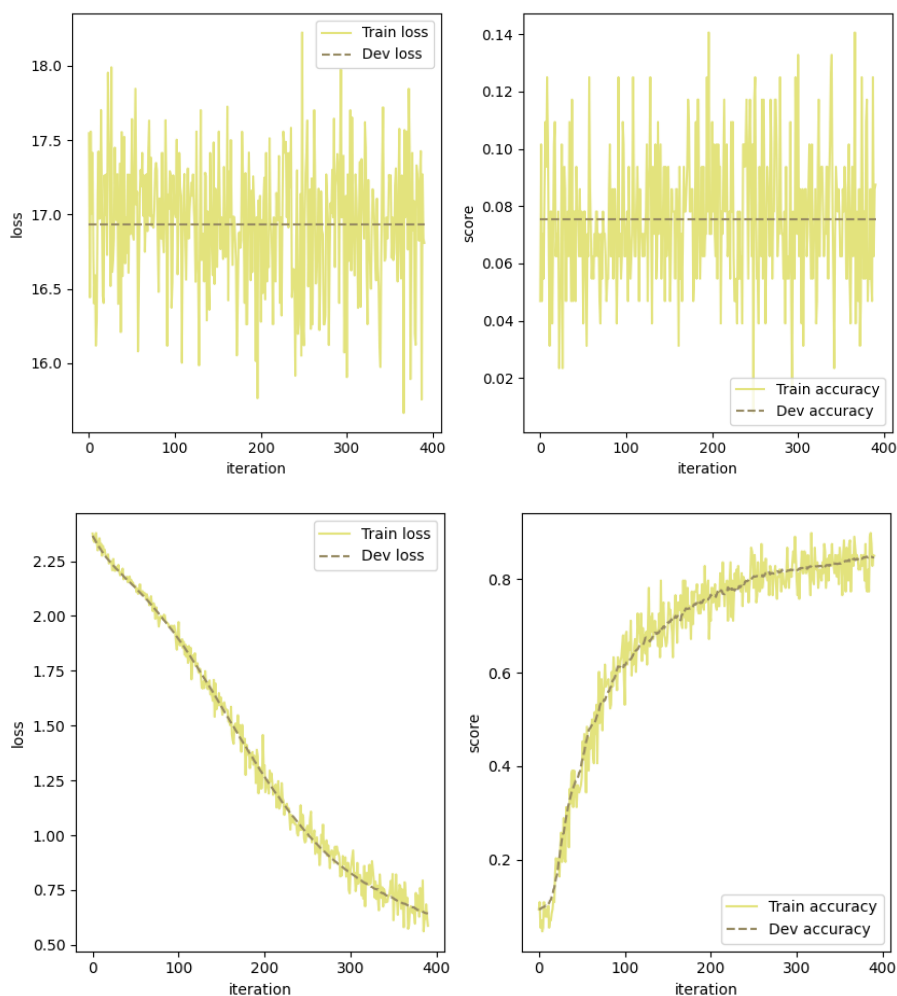


图 2: 修复后顺利的收敛了，赢！

接下来运行了 5 个 epoch, 进一步查看运行的结果：运行 testmodel.py 后输出的 accuracy 为 0.9042, 其中训练输出见图3

MomentGD 应用了 MomentGD 后, 收敛速度确实优化了, 但是精度表现没有提高, 甚至稍稍弱于最初的设置, accuracy 为 0.8896, 具体训练见图4

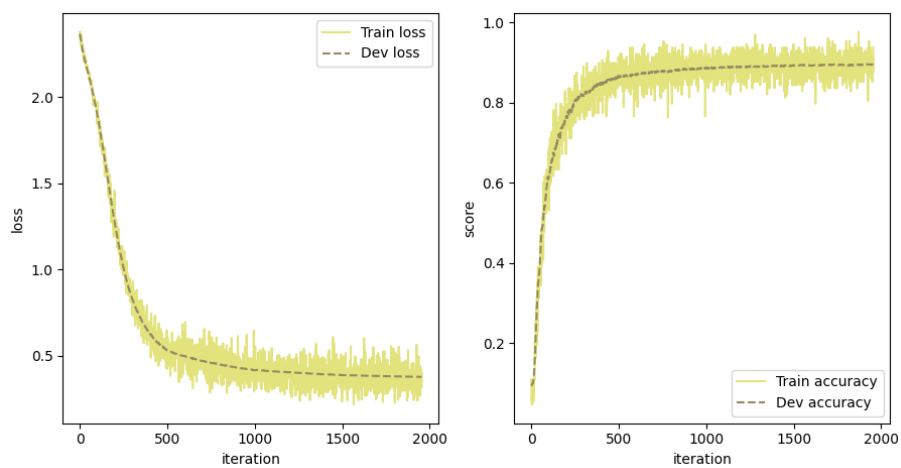


图 3: 在 2000 步训练后模型的精度明显的变高了

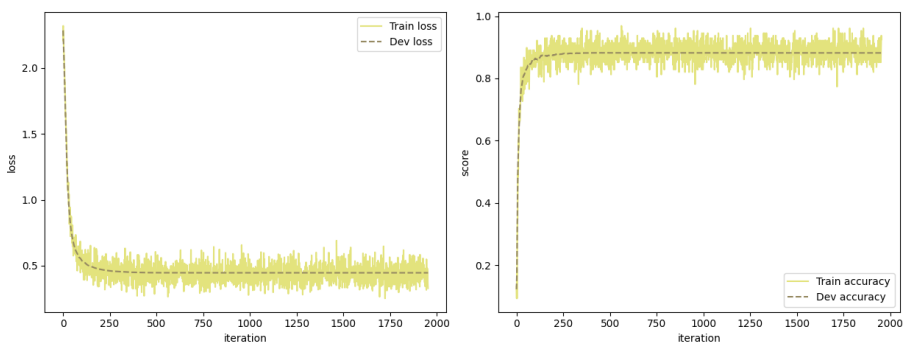


图 4: 模型收敛明显变快, 但是精度没有得到提升

正则化 使用了如 Hint 所表示的 l2 正则化, dropout 和早停 (好吧没停) 方法, 如图, 准确率 0.8896 居然和 MomentGD 一致。

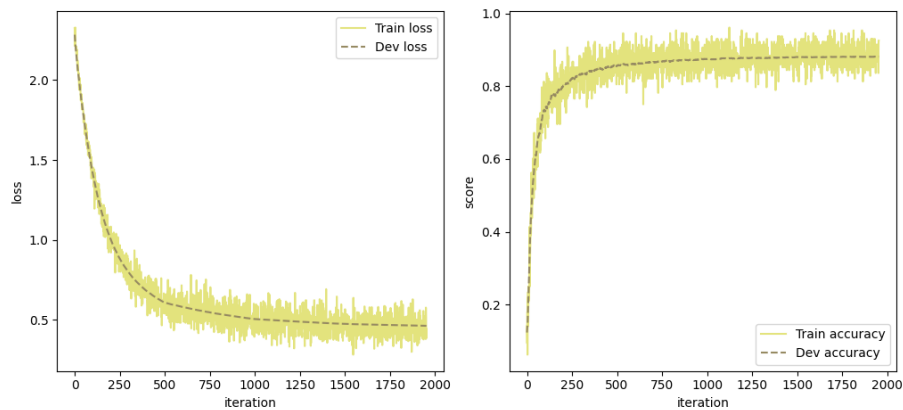


图 5: 图为正则化得到的训练误差和训练、验证集准确率

交叉熵损失函数 这个最开始就写了, 一直都用的是 cross entropy loss

训练步数优化 发现一直只训练 2k 步并不一定模型完全收敛, 于是我们扩大为 4k 步再次训练一遍, 于是有了这三张图: 图8. 他们的 accuracy 分别为: 基础模型 = 0.9246; MomentGD = 0.9705, 正则化 = 0.8916. 可以看到 MomentGD 优化下的精度收敛得明显更好. 值得一提的是基础模型在训练中的 valid accuracy 达到了 95%, 似乎是过拟合了; 而正则化项有因早停机制, 没有达到理想效果.

Con2V 写了 conv2d, 但是由于运行消耗的时间过长, 于是没有继续下去, 见9

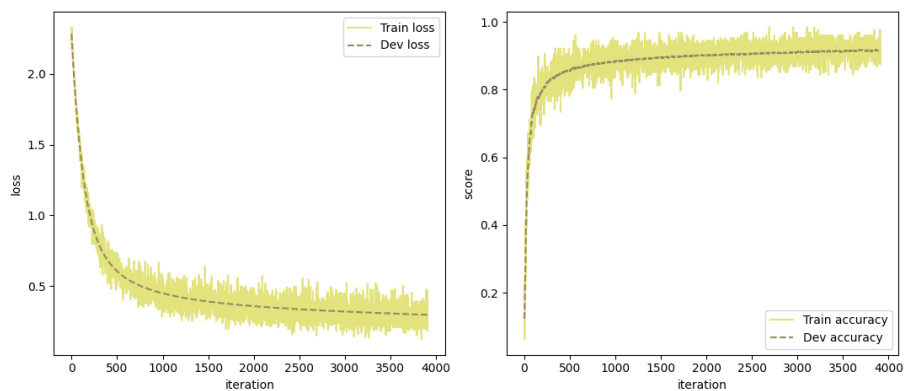


图 6: 基础模型

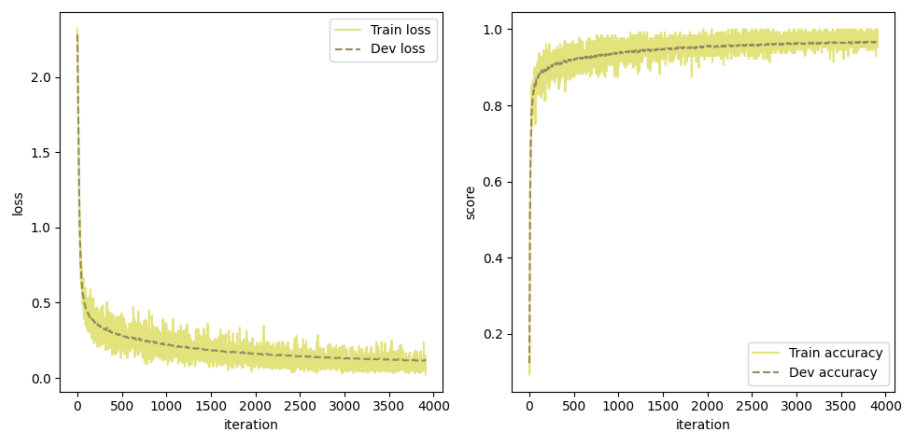


图 7: 使用 MomentumGD 优化的模型

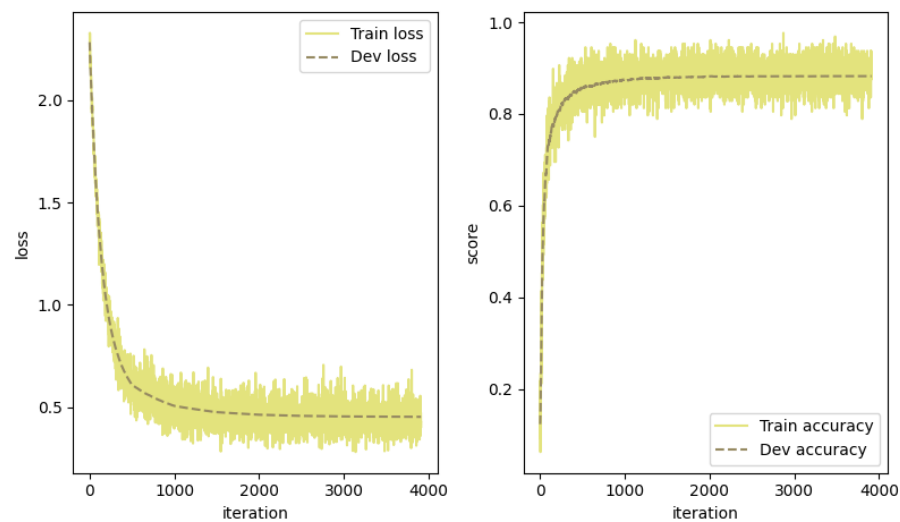


图 8: 使用多种正则化的模型

```
Training: 0% | 0/5 [00:00?, 7epoch/s]
epoch 0, iteration: 0 | 0/391 [00:00?, 7batch/s]
[Train] loss: 2.3245715155396747, score: 0.1250000
[Dev] loss: 2.293245625600633, score: 0.1159000
参数更新位置:
Layer 0 W max: 1.1154883815485423, min: -1.2902885491312425
Epoch 0: 1% | 2/391 [01:33:5:04:28, 46.96%/batch, train_loss=2.2467, train_acc=0.1484, val_acc=0.1380]
```

图 9: 运行一个 batch 需要 45s...