

# Assignment 2

2020 年 7 月

## 目录

1 Part 1	1
2 Part 2	1
3 运行代码	2

## 1 Part 1

补充完 pytorch 的基本代码后训练 `myPTRNNModel` 的结果如Figure 1，loss 在每一轮都会采集，而 accuracy 每 50 轮采集一次以保证训练性能不会受到太大影响。从图中可以看到在 100 轮的时候模型就已经有达到 1 的准确率，loss 也已经很低了，之后它们的数值变化不大。

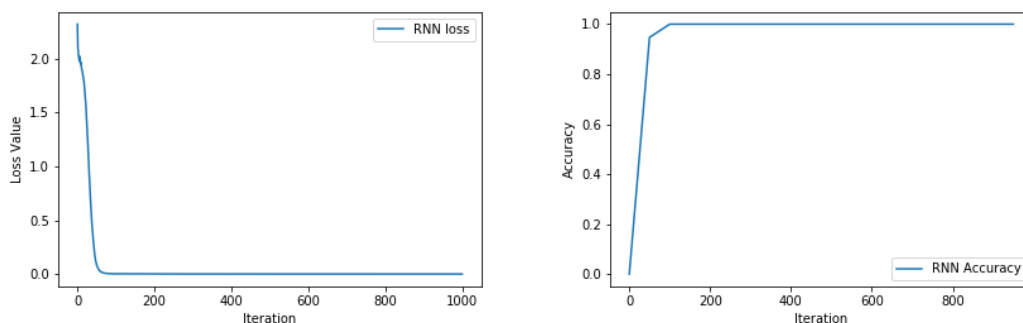


Figure 1: `myPTRNNModel` 模型的 loss 和 acc 结果

## 2 Part 2

在这一部份，我对 `pt.py` 的部份函数增加了更多的参数，以方便调整参数分析它们对模型训练和性能的影响。以下分别分析数字长度 `maxlen`、RNN 模型网络层数 `layers`、学习率 `lr` 和不同模型的影响。

Figure 2研究模型层数的影响，此时数字长度最长为 14。考虑到训练轮数太多时后面的 loss 和精度变化不大，所以只训练了 500 轮，这 500 轮中 loss 和精度变化明显，更适合用于分析。从

图中可以看到 layers 取 2 的时候精度的提升是最快的，而 loss 的收敛速度在 layers 取 2-4 的几条线都差不多，只有 1 是最差的。之后的比较测试中我还是使用默认的层数 2。

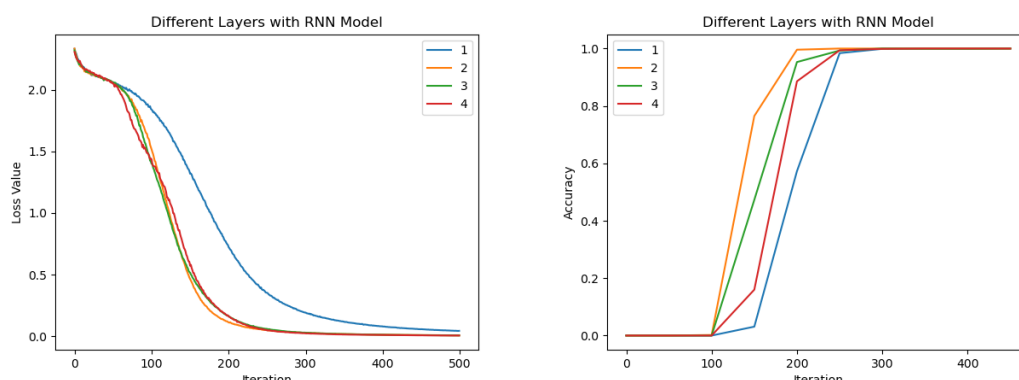


Figure 2: 模型层数对 loss 和 acc 的影响结果

如Figure 3, 因为 numpy 的 randint() 函数不支持长度大于 8 字节的整数, 所以将 data.py 中的 gen\_data\_batch() 函数中随机数生成换成了用 Python 的 random 包生成。图中可以看到, 数字长度增加, loss 收敛速度似乎更快了, 这可能是因为数字长度更长, 导致每轮训练中能获得的信息量更大了。

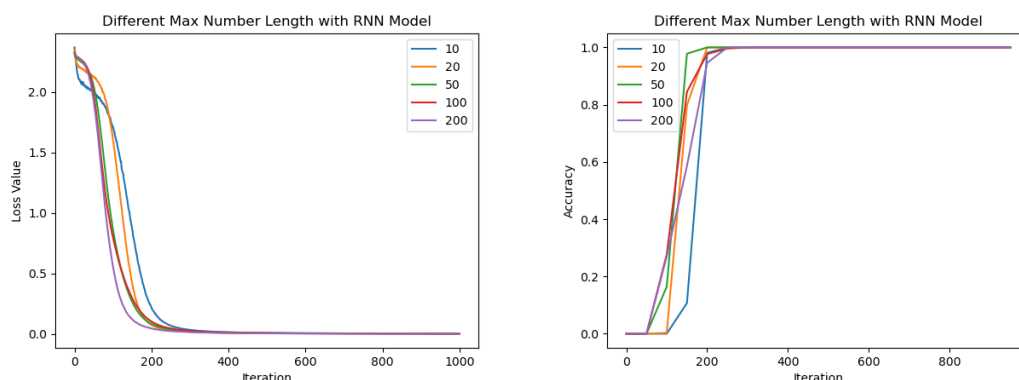


Figure 3: 数字最大长度对 loss 和 acc 的影响结果

如Figure 4, 从 loss 和精度的变化可以看出来, 学习率过低时, 需要大量的数据或者重复很多很多轮的训练才会有效果。如 0.0001 的曲线在 500 轮训练中 loss 变化很小, 且精度也一直为 0, 如果训练更多次或许会有一定效果。而学习率过高时, 会导致无法收敛到局部最优解, 在局部最优解附近跳动, 从而使得 loss 和精度受到较大影响。如图中的 0.1 的 loss 没有明显的下降, 且准确率也一直是 0。

如Figure 5, RNN 和 GRU 训练过程和性能都差不多, 而 LSTM 与它们两个相比 loss 收敛速度和精度提升速度略逊一筹, 不过最终的精度差别不大。

### 3 运行代码

在根目录执行 python source.py 可依次训练 myPTRNNModel 和 myAdvPTRNNModel 模型, 训练完成后会生成 myPTRNNModel 的 loss 和 accuracy 结果图片。在 handout 文件夹下, 执行命令

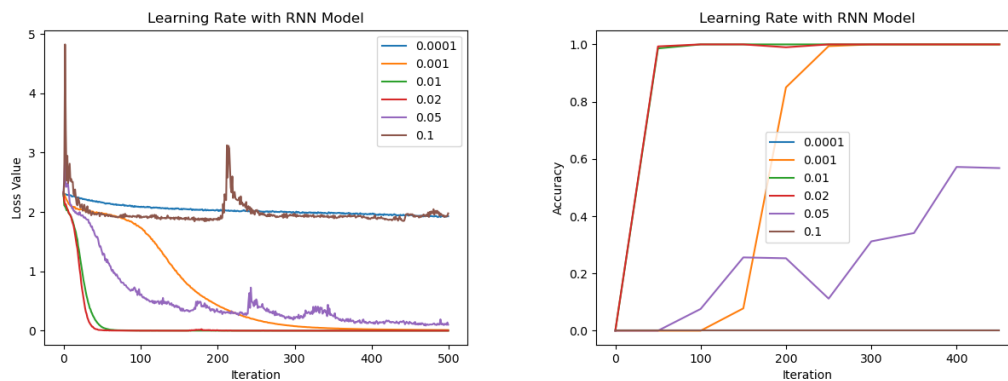


Figure 4: 学习率对 loss 和 acc 的影响结果

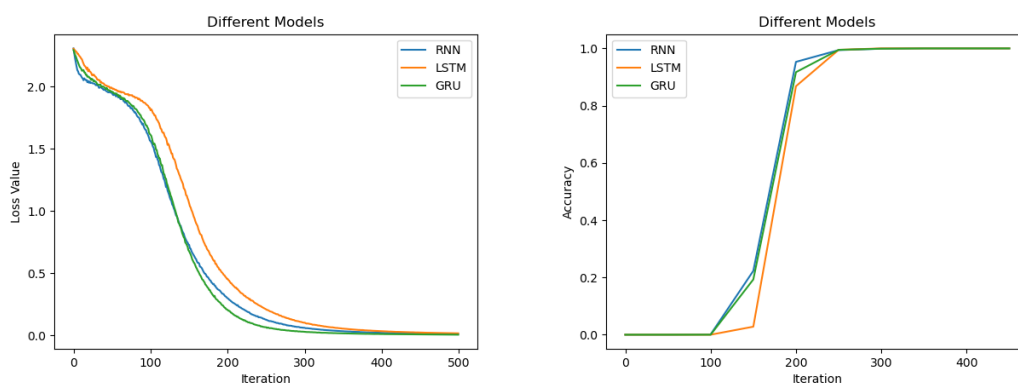


Figure 5: 模型对 loss 和 acc 的影响结果

python pt.py 可以进行训练参数的测试比较，完成后会在 handout 目录下生成几张结果图片。  
 改变 pt.py 中 train\_test() 函数的参数可以做不同的测试训练。