

NKU 深度学习（高阶课）实验报告



实验名称： _____ 循环神经网络 _____

学 院： _____ 网络空间安全学院 _____

姓 名： _____ 田晋宇 _____

专 业： _____ 物联网工程 _____

二〇二五年六月

目录

1	实验要求	2
2	RNN	2
2.1	基本原理	2
2.2	模型结构	2
2.3	训练结果可视化	3
2.4	性能分析	3
3	LSTM	4
3.1	基本原理	4
3.2	模型结构	4
3.3	训练结果可视化	4
3.4	性能分析	4
4	总结	5
4.1	性能对比	5
4.2	优势分析	6

1 实验要求

- 掌握循环神经网络（RNN）的基本原理；
- 学会使用 PyTorch 搭建 RNN 网络来进行姓名识别任务的训练；
- 学会使用 PyTorch 搭建 LSTM 网络来进行姓名识别任务的训练。

2 RNN

2.1 基本原理

循环神经网络（Recurrent Neural Network, RNN）是一类具有循环连接结构的神经网络，适用于处理序列数据，如语音、文本等。其基本思想是在处理当前输入的同时，引入上一个时间步的隐藏状态，从而实现对历史信息的记忆。

2.2 模型结构

本实验中使用的 RNN 网络结构如下（以 PyTorch 实现为基础）：

- 输入层（Input Layer）：将每个字符的独热编码作为输入，维度为 $N \times I$ ，其中 N 为批次大小， I 为字符类别数；
- RNN 层：采用 `torch.nn.RNN` 模块，隐藏层维度设为 H ，接受当前输入与前一刻的隐藏状态共同决定当前输出；
- 全连接层（Linear Layer）：将 RNN 输出映射到类别数维度的线性层，即 $H \rightarrow C$ ，其中 C 为名字类别数；
- 输出层（Softmax）：对线性层输出使用 `log_softmax` 转换为对每一类的对数概率；

网络整体结构如下所示（可由 `print(net)` 获得）：

```
RNN(  
  (rnn): RNN(input_size=57, hidden_size=128, num_layers=1)  
  (fc): Linear(in_features=128, out_features=18, bias=True)  
)
```

其中，输入大小 57 对应于 57 个英文字母和标点的独热编码，输出类别数 18 对应于国家或地区名称的分类任务。

2.3 训练结果可视化

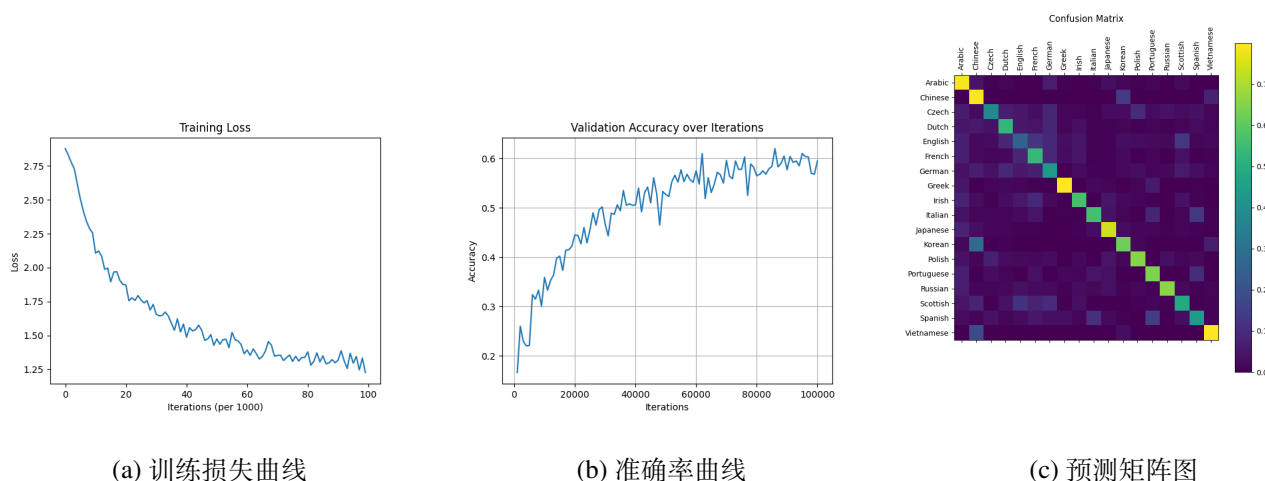


图 1: RNN 训练指标可视化结果

2.4 性能分析

- **训练损失曲线 (图 1(a)):** 可以观察到模型的训练损失随着迭代次数的增加呈明显下降趋势, 表明模型在训练集上逐渐收敛, 学习能力良好。初期损失较高, 随后迅速下降, 并在后期趋于平稳, 说明学习率和优化器设置合理。
- **准确率曲线 (图 1(b)):** 验证集上的准确率持续上升, 从初期的约 20% 提高到最终接近 60%, 体现了模型具备一定的泛化能力。准确率在 6 万次迭代之后逐渐趋于稳定, 表明模型收敛较好。
- **预测混淆矩阵 (图 1(c)):** 混淆矩阵对模型在不同类别上的预测表现进行了可视化。可以看出, 大多数对角线位置的值较高, 说明模型对所属类别的预测相对准确。但也存在部分混淆区域, 例如英语和荷兰语、韩语和日语之间存在交叉误判, 可能由于这些语言在姓名拼写上的相似性。

RNN 模型在该任务中表现出良好的训练收敛性和中等水平的识别准确率。然而, 由于其结构限制, 模型在处理较长或上下文复杂的序列时能力有限, 部分语言类别之间的区分度仍有提升空间。

3 LSTM

3.1 基本原理

长短期记忆网络 (Long Short-Term Memory, LSTM) 是 RNN 的一种改进结构, 通过引入门控机制, 有效解决了传统 RNN 在处理长序列时容易出现的梯度消失问题。LSTM 能够在较长的时间跨度内保留重要的信息, 因此更适用于捕捉字符序列中的长期依赖关系。

3.2 模型结构

本实验中使用的 LSTM 网络结构如下 (基于 PyTorch 实现):

- 输入层 (Input Layer): 将每个字符的独热编码作为输入, 维度为 $N \times I$, 其中 N 为批次大小, I 为字符类别数;
- LSTM 层: 采用 `torch.nn.LSTM` 或自定义模块实现, 包含遗忘门、输入门和输出门机制, 控制信息在时间维度上的传递与更新;
- 全连接层 (Linear Layer): 将 LSTM 最后时刻的隐藏状态输出映射到分类空间, 即 $H \rightarrow C$, 其中 H 为隐藏状态维度, C 为名字类别数;
- 输出层 (Softmax): 对线性层输出应用 `log_softmax`, 得到对每一类名字的预测概率;

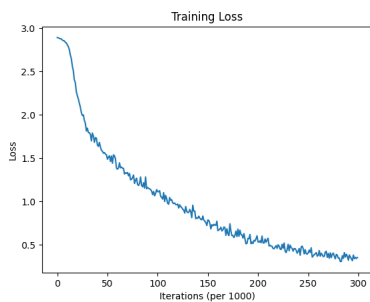
LSTM 网络具备更强的建模能力, 在保持结构信息的同时, 也能捕捉序列中较远位置之间的关联, 适合处理如姓名识别这类高度依赖上下文的序列分类任务。

在训练过程中, 我们适当增加了训练轮次, 并提升了学习率, 以促进模型充分学习和更快收敛。这些优化策略进一步提升了模型在验证集上的表现。

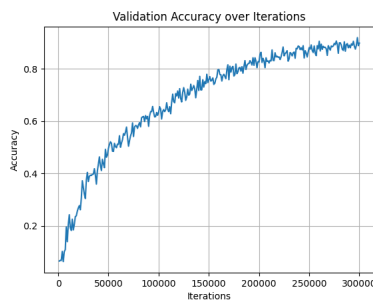
3.3 训练结果可视化

3.4 性能分析

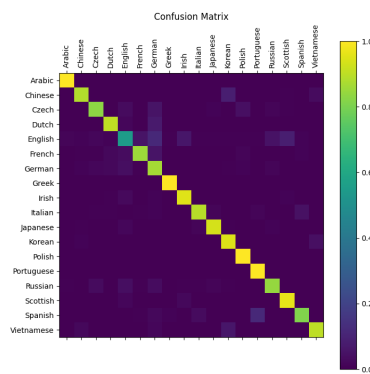
- **训练损失曲线 (图 2(a)):** LSTM 模型的训练损失整体呈平滑下降趋势, 前期下降较快, 后期逐渐趋于平稳, 表明模型训练过程稳定, 具备良好的收敛性。相较于 RNN, LSTM 能更有效地捕捉序列中的长期依赖, 因此在更多轮迭代中仍能持续优化。
- **准确率曲线 (图 2(b)):** LSTM 在验证集上的准确率迅速提升, 从初始约 20% 提高至最终超过 90%, 提升幅度显著。曲线整体平滑, 无明显震荡, 反映出模型训练稳定, 泛化能力强。这一表现明显优于 RNN, 说明 LSTM 的门控机制提升了模型对复杂序列的建模能力。



(a) 训练损失曲线



(b) 准确率曲线



(c) 预测矩阵图

图 2: LSTM 训练指标可视化结果

- **预测混淆矩阵 (图 2(c)):** 从混淆矩阵可以看出, 大多数类别在对角线上的预测概率接近于 1, 说明模型在多数语言类别的识别上达到了较高准确率。同时, 非对角线区域颜色整体较浅, 表明模型的误判概率显著降低, 类别区分能力更强。

可以看出, LSTM 模型不仅在训练过程中的损失收敛更快、准确率更高, 而且预测结果更具稳定性和精确性。相比于传统 RNN, LSTM 在本任务中具有更强的实际应用价值。

4 总结

4.1 性能对比

表 1: RNN 和 LSTM 的性能对比

模型	验证准确率
RNN	0.625
LSTM	0.928

在本实验的姓名识别任务中, LSTM 网络在训练损失下降速度、最终准确率以及混淆矩阵表现上均优于传统的 RNN 网络, 其性能优势主要体现在以下几个方面:

4.2 优势分析

- **更强的长期依赖建模能力：**RNN 在反向传播过程中容易出现梯度消失或梯度爆炸问题，导致其难以捕捉长距离依赖的信息。而 LSTM 通过引入门控机制（遗忘门、输入门和输出门）以及显式的记忆单元，可以在序列中有效保留长期状态信息，从而更好地理解姓名中后缀、前缀等长期相关特征。
- **优化的信息选择机制：**LSTM 的门控结构允许模型在每个时间步对输入信息、过去记忆以及输出结果进行选择性地保留或丢弃，这种“有选择地记忆与遗忘”机制使得 LSTM 在处理冗余或干扰信息较多的序列时更加鲁棒。
- **收敛速度与训练稳定性更佳：**实验中可以观察到，LSTM 网络的训练损失曲线下降更平稳，准确率提升更快，说明其在训练过程中更易收敛且不易陷入局部最优。这一优势也得益于其稳定的梯度流动结构。
- **泛化能力更强：**在验证集上的性能结果表明，LSTM 在处理未见过的名字样本时，仍能保持较高准确率，表明其具备更好的泛化能力。

综上所述，LSTM 网络在结构设计上对传统 RNN 进行了有效扩展，显著提升了模型对序列数据的建模能力，因而在本实验任务中取得了更优的表现。