## 循环神经网络实验报告

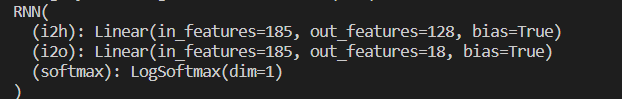
姓名：管昀玫 学号：2013750

### 实验要求

* 掌握RNN原理
* 学会使用PyTorch搭建循环神经网络来训练名字识别
* 学会使用PyTorch搭建LSTM网络来训练名字识别

### RNN

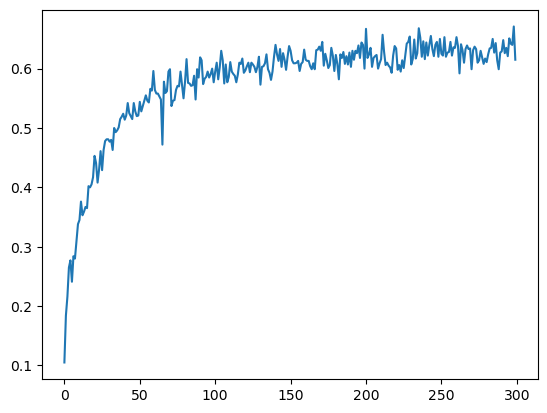
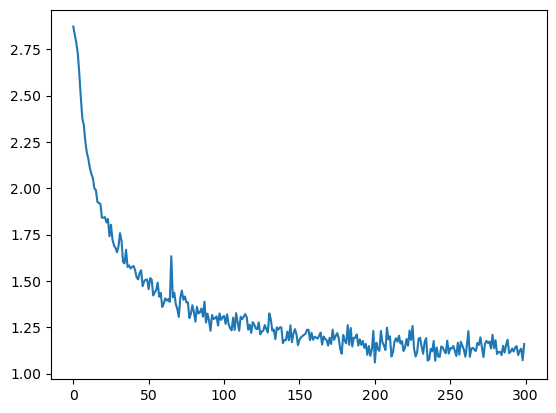
该网络结仅由一个输出隐藏状态的线性层、一个输出结果的线性层及一个softmax激活层组成，如下图所示：



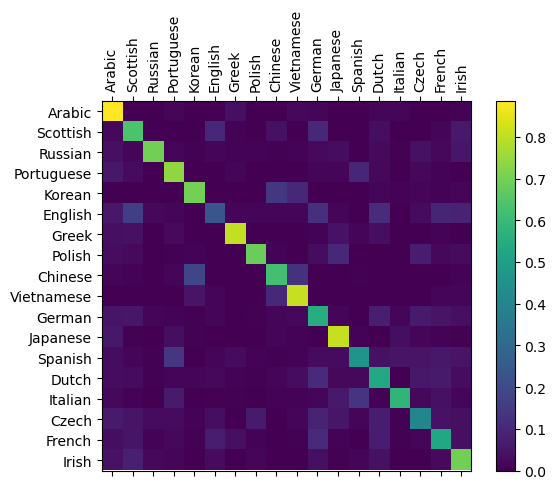
数据集为名字识别验证集。

训练该网络，得到以下结果：

左图为训练loss曲线，右图为训练accuracy曲线

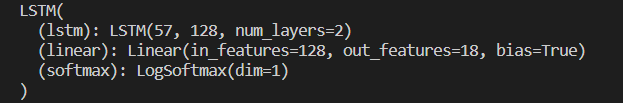


预测矩阵图如下所示：



### LSTM

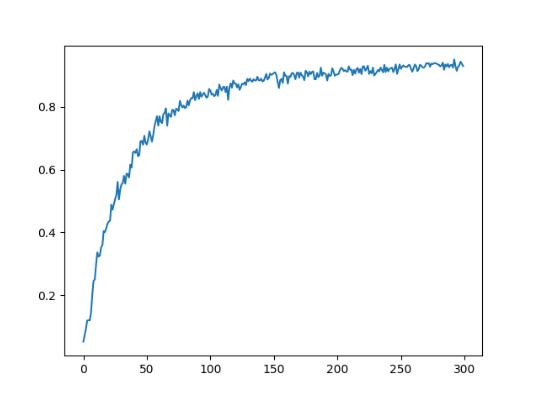
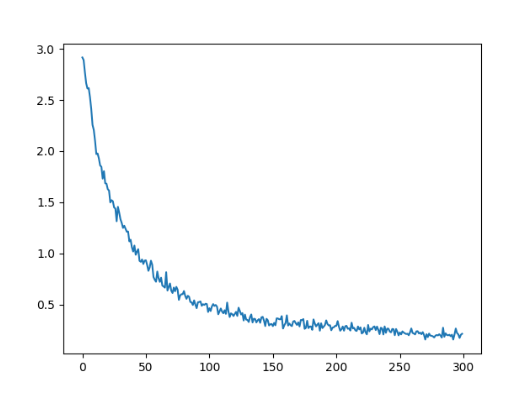
使用torch.nn.LSTM实现的网络结构如下所示：



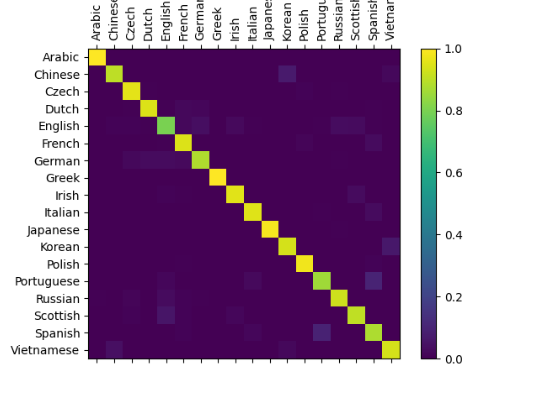
该网络由一个LSTM，一个输出结果的线性层和一个softmax激活函数组成。

训练该模型，得到以下结果：

左图为loss曲线，右图为训练accuracy曲线



混淆矩阵如下所示：



可以看到，LSTM相较于RNN有很明显的提升，在n\_iters=300000时LSTM的准确率可以达到90%以上。

### 个人实现的LSTM

LSTM有三个门：遗忘门、记忆门、输出门，可以用公式表示如下：

*it*​=*σ*(*Wii*​*xt*​+*bii*​+*Whi*​*ht*−1​+*bhi*​)

*ft*​=*σ*(*Wif*​*xt*​+*bif*​+*Whf*​*ht*−1​+*bhf*​)

*gt*​=tanh(*Wig*​*xt*​+*big*​+*Whg*​*ht*−1​+*bhg*​)

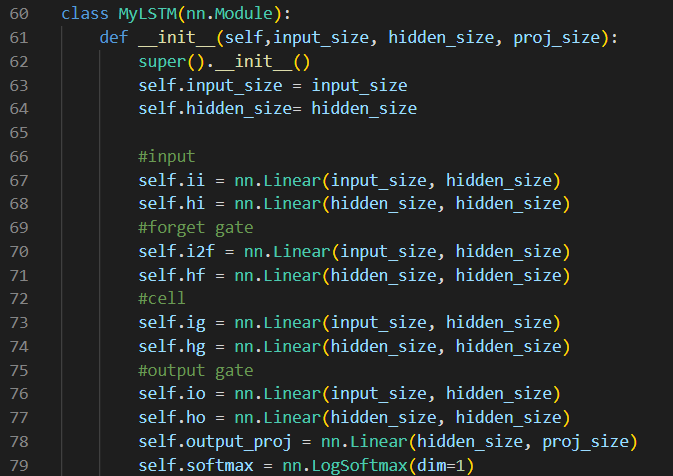
*ot*​=*σ*(*Wio*​*xt*​+*bio*​+*Who*​*ht*−1​+*bho*​)

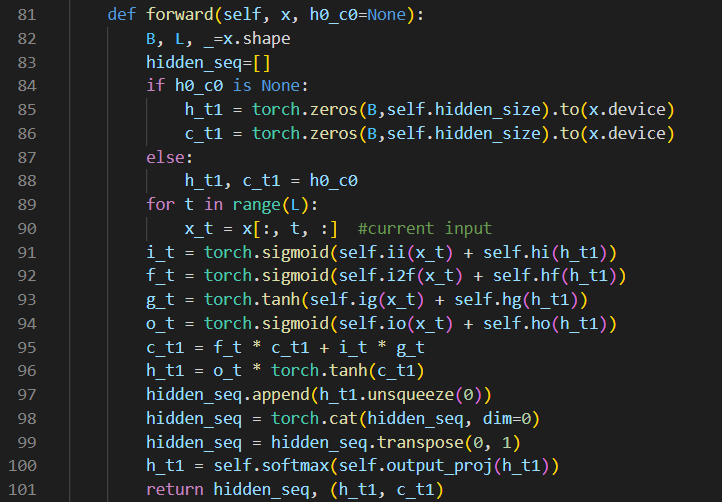
*ct*​=*ft*​⊙*ct*−1​+*it*​⊙*gt*​

*ht*​=*ot*​⊙tanh(*ct*​)​

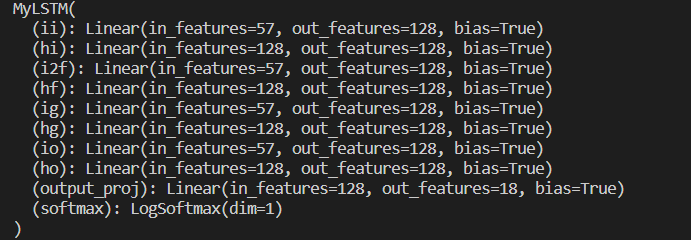
其中*it*​, *ft*​, *gt*​, *ot*​ 分别为输入、遗忘门、记忆门和输出门。

个人实现的LSTM代码如下所示：



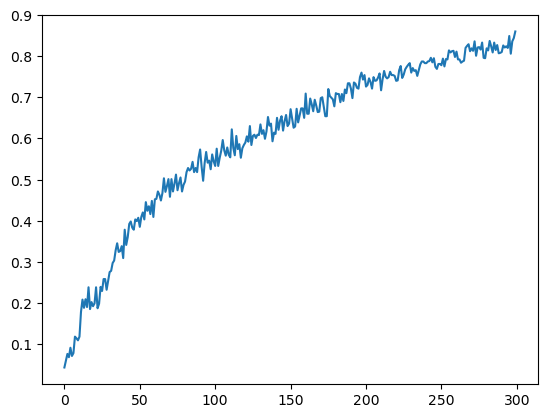
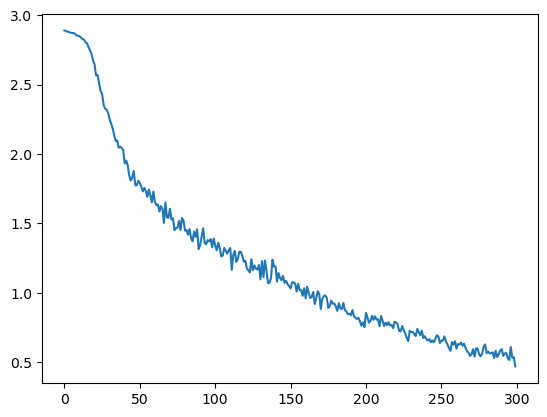


打印该网络的参数结构如下所示：

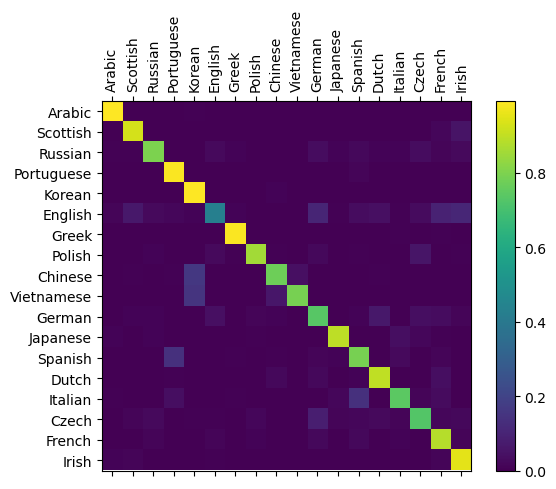


训练该模型，得到以下结果：

左图为训练loss曲线，右图为训练accuracy曲线



混淆矩阵如下所示：



自己实现的LSTM与torch.nn.LSTM相比，准确率相差不大，验证了正确性。

### 为什么LSTM网络的性能优于RNN网络

我认为LSTM网络的性能优于RNN网络的性能有以下几个原因：

1. 长期依赖建模：传统的 RNN 在处理长序列时容易面临梯度消失或梯度爆炸的问题，导致难以有效捕捉长期依赖关系。而 LSTM 通过引入门控机制，如输入门、遗忘门和输出门，能够更好地控制信息的流动，有效地缓解了长期依赖问题。LSTM 中的门控机制允许模型选择性地遗忘和更新信息，使得它能够更好地处理长期记忆。
2. 处理短期变化和长期记忆：传统的 RNN 只能对当前输入和前一时刻的隐藏状态进行操作，容易受到短期变化的影响，难以保留和利用长期记忆。相比之下，LSTM 通过隐藏状态和记忆单元的组合，可以同时维护短期和长期的信息，这使得 LSTM 能够更好地处理序列中的长期依赖和短期变化。
3. 可学习的门控机制：LSTM 的门控机制允许模型自动学习输入数据中的重要信息和关键时间步。输入门和遗忘门的权重通过训练过程中的反向传播自动学习，从而模型能够自适应地控制信息的流动和保存重要的上下文。
4. 更丰富的模型表达能力：LSTM 拥有更复杂的结构，包含多个门控单元和记忆单元，相比之下，传统的 RNN 结构相对简单。这使得 LSTM 具有更强的模型表达能力，可以更好地捕捉序列数据中的复杂关系和模式。

而RNN存在严重的遗忘问题。例如，在模型读取处理“做有中国灵魂世界眼光的现代人”这句较长的话时，模型输入到“现代人”时，隐藏单元中有关“中国”的信息很可能已经消失。这就是说，RNN不具备长期记忆，而只具备短期记忆，难以捕捉长程依赖。这一问题的本质为梯度消失或梯度爆炸，导致较前位置相关参数难以有效更新。

综上所述，LSTM能优于RNN网络的原因为引入门机制，有效解决RNN的遗忘问题。