

RNN 实验报告

朱浩泽 1911530

June 17, 2022

1 老师提供的原始版本 RNN 网络结构

1.1 网络结构

```
1 RNN(  
2   (i2h): Linear(in_features=185, out_features=128, bias=True)  
3   (i2o): Linear(in_features=185, out_features=18, bias=True)  
4   (softmax): LogSoftmax(dim=1)  
5 )
```

1.2 实验结果

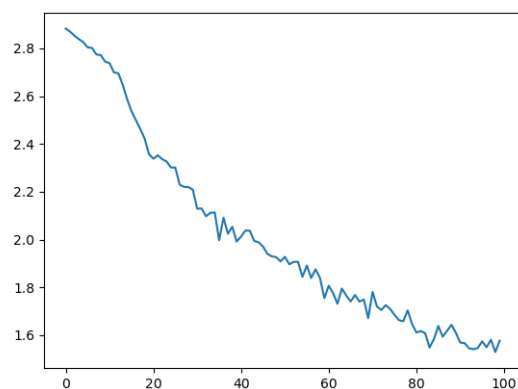


Figure 1: RNN 的损失

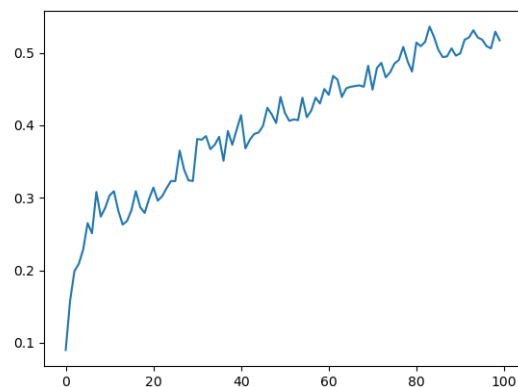


Figure 2: RNN 的准确率

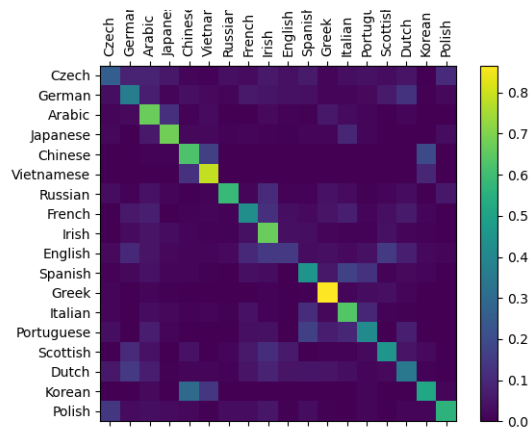


Figure 3: RNN 的预测准确图

2 利用 Pytorch 搭建的 LSTM

2.1 网络结构

```

1 LSTM(
2   (lstm): LSTM(57, 128, num_layers=2)
3   (linear): Linear(in_features=128, out_features=18, bias=True)
4   (softmax): LogSoftmax(dim=1)
5 )

```

2.2 实验结果

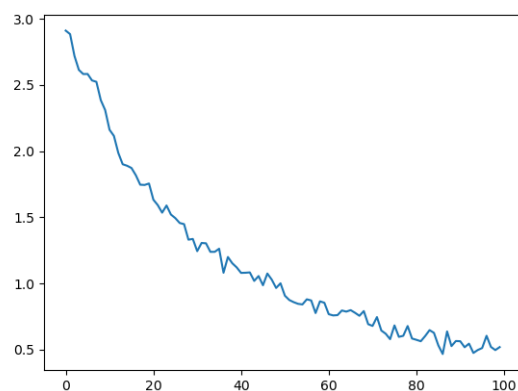


Figure 4: LSTM 的损失

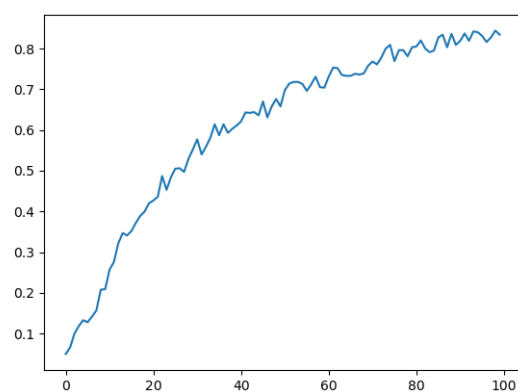


Figure 5: LSTM 的准确率

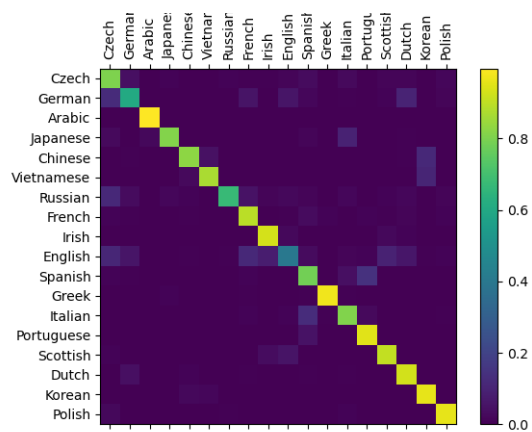


Figure 6: LSTM 的预测准确图

3 为什么 LSTM 网络的性能优于 RNN 网络

RNN 即循环神经网络，其可以处理一定的短期依赖，但是对长序列进行学习时，循环神经网络会出现梯度消失和梯度爆炸现象，无法掌握长时间跨度的非线性关系。这种现象的主要原因是，当序列比较长的时候，序列后部的梯度很难反向传播到前面的序列，这便导致了前面提到的问题。假设 h_t 指的是时间 t 时刻的隐藏层状态值， x_t 为 t 时刻的输入，我们可以通过以下公式进行隐藏层的计算

$$h_t = \tanh(W_t \cdot [h_{t-1}, x_t] + b_h)$$

可以看出，传统的 RNN 的任意时刻的隐藏层获取的信息只来源于当前输入和上一时刻的隐藏层的信息，没有任何记忆功能，所以并不能处理长程依赖。

而 1997 年 Long Short Memory 一种提出的 LSTM 结构，巧妙的应用了自循环的思想产生了梯度长时间持续流动的路径来解决这一问题。具体来说，LSTM 网络引入了三个门控机制来控制信息传递的路径，分别是输入门 i_t 、遗忘门 f_t 和输出门 o_t 。

- 遗忘门 f_t 控制上一个时刻的内部状态 c_{t-1} 需要遗忘多少信息
- 输入门 i_t 控制当前时刻的候选状态 \tilde{c}_t 有多少信息需要保存
- 输出门 o_t 控制当前时刻的内部状态 c_t 有多少信息需要输出给外部状态 h_t

其具体的计算公式为

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

具体计算过程为首先利用上一时刻的外部状态 h_{t-1} 和当前时刻的输入 x_t ，计算出三个门，以及候选状态 \tilde{c}_t ，结合遗忘门 f_t 和输入门 i_t 来更新记忆单元 c_t ，结合输出门 o_t ，将内部状态的信息传递给外部状态 h_t 。和 RNN 的隐状态 h 相同，LSTM 中的隐藏层也存储了历史信息。在基础的 RNN 中，隐状态每个时刻都会被重写，因此可以看作一种短期记忆。而长期记忆则可以看作是一种网络参数，隐含了从训练数据中学到的经验，其更新周期要远远慢于 RNN 的隐藏层的更新。而在 LSTM 网络中，记忆单元 c 可以在某个时刻捕捉到某个关键信息，并有能力将此关键信息保存一定的时间间隔。记忆单元 c 中保存信息的生命周期要长于短期记忆 h ，所以 LSTM 可以更好的学习到长程依赖。而文字中有时需要利用上下文信息才可以更好的进行理解，所以 LSTM 网络的性能优于 RNN。