



南開大學
Nankai University

计算机学院
深度学习实验报告

RNN 实验报告

姓名：徐文斌

学号：2010234

专业：计算机科学与技术

2023 年 6 月 25 日

目录

1	原始版本 RNN	2
1.1	网络结构	2
1.2	训练结果	2
2	LSTM	3
2.1	Pytorch 自带 LSTM	3
2.1.1	网络结构	3
2.1.2	训练结果	4
2.2	个人实现 LSTM	5
2.2.1	网络结构	5
2.2.2	训练结果	6
3	为什么 LSTM 网络的性能优于 RNN 网络	7

1 原始版本 RNN

1.1 网络结构

使用 print 函数打印网络结构如下所示：

```
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 训练结果

训练曲线如下，最终准确率在 60% 左右。

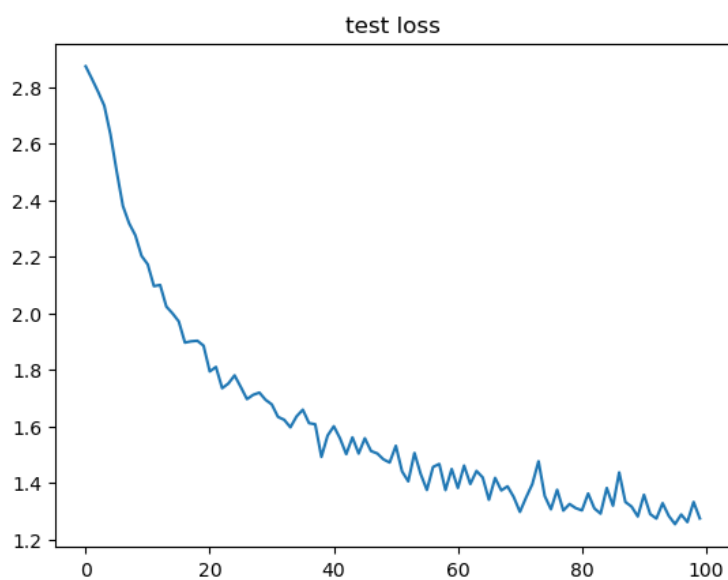


图 1.1: 损失曲线

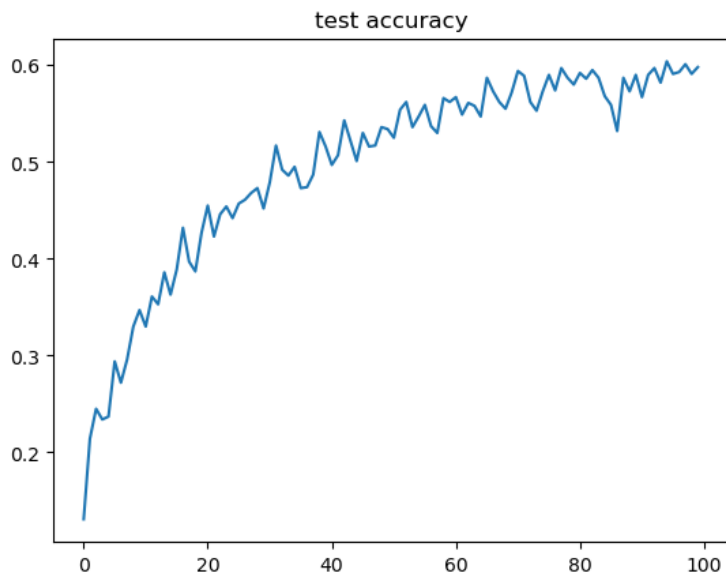


图 1.2: 准确率曲线

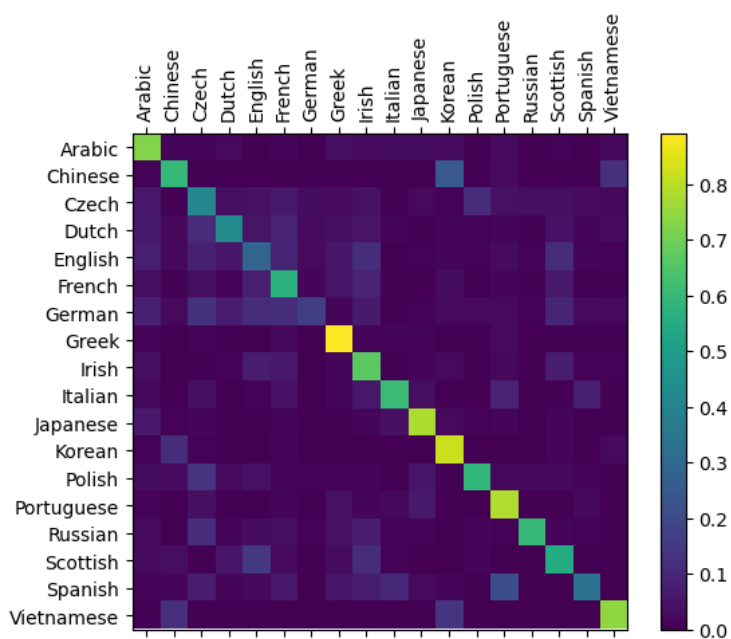


图 1.3: 预测矩阵图

2 LSTM

2.1 Pytorch 自带 LSTM

2.1.1 网络结构

首先我们使用 Pytorch 自带的 LSTM 搭建网络，使用 print 函数打印网络结构如下所示：

```
1 LSTM(
2   (lstm): LSTM(57, 128)
```

```
3 (linear): Linear(in_features=128, out_features=18, bias=True)
4 (softmax): LogSoftmax(dim=1)
5 )
```

2.1.2 训练结果

训练曲线如下，最终准确率在 85% 左右。

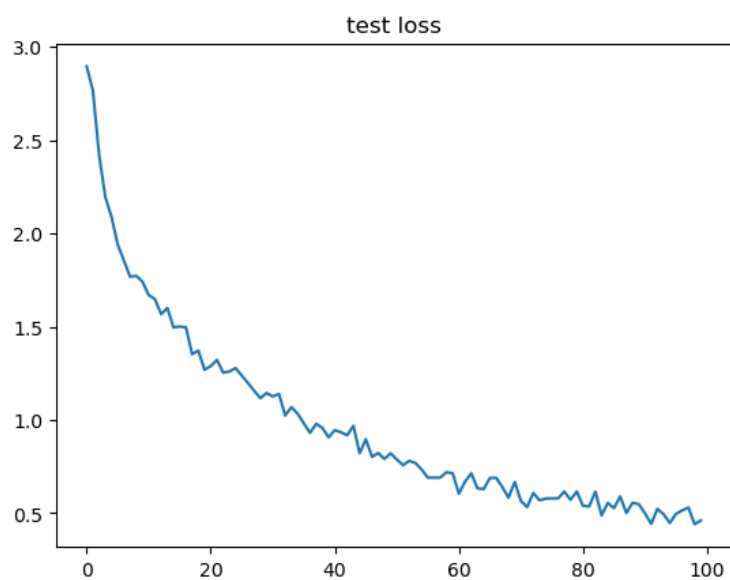


图 2.4: 损失曲线

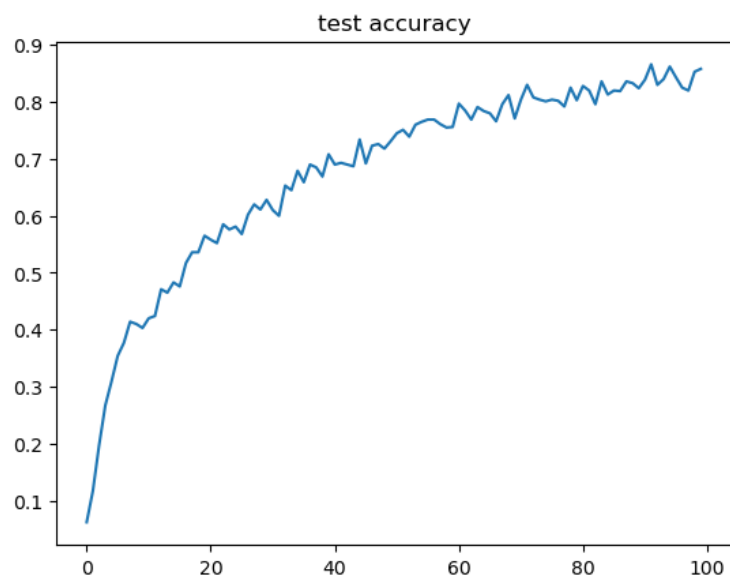


图 2.5: 准确率曲线

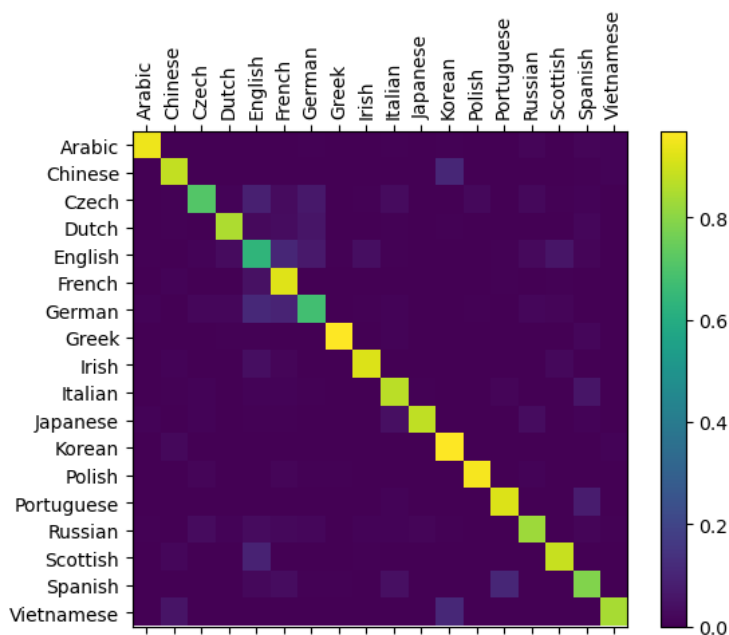


图 2.6: 预测矩阵图

2.2 个人实现 LSTM

2.2.1 网络结构

接下来自行实现 LSTM 网络，使用 print 函数打印网络结构如下所示：

```

1 LSTM(
2   (forget_gate): Linear(in_features=185, out_features=128, bias=True)
3   (input_gate): Linear(in_features=185, out_features=128, bias=True)
4   (cell_update): Linear(in_features=185, out_features=128, bias=True)
5   (output_gate): Linear(in_features=185, out_features=128, bias=True)
6   (classify): Linear(in_features=128, out_features=18, bias=True)
7   (softmax): LogSoftmax(dim=1)
8 )

```

```

1 def forward(self, input):
2     input_length = input.size()[0]
3     hidden = torch.zeros(1, self.hidden_dim).to(input.device)
4     cell = torch.zeros(1, self.hidden_dim).to(input.device)
5     output = None
6     for i in range(input_length):
7         x = input[i]
8         state = torch.concat((x, hidden), dim=-1)
9         f = torch.sigmoid(self.forget_gate(state))
10        i = torch.sigmoid(self.input_gate(state))
11        c = torch.tanh(self.cell_update(state))
12        cell = f * cell + i * c
13        output = torch.sigmoid(self.output_gate(state))
14        hidden = output * torch.tanh(cell)

```

```
15     output = self.softmax(self.classify(output))  
16     return output
```

2.2.2 训练结果

训练曲线如下，最终准确率在 70% 左右。

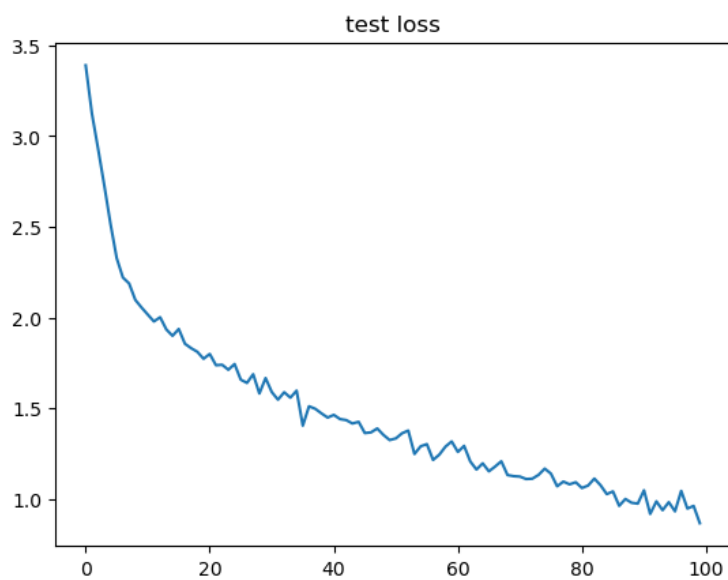


图 2.7: 损失曲线

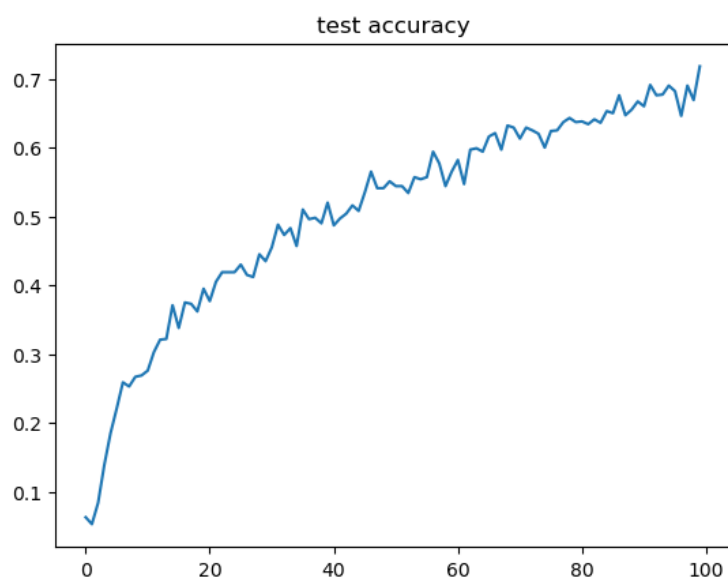


图 2.8: 准确率曲线

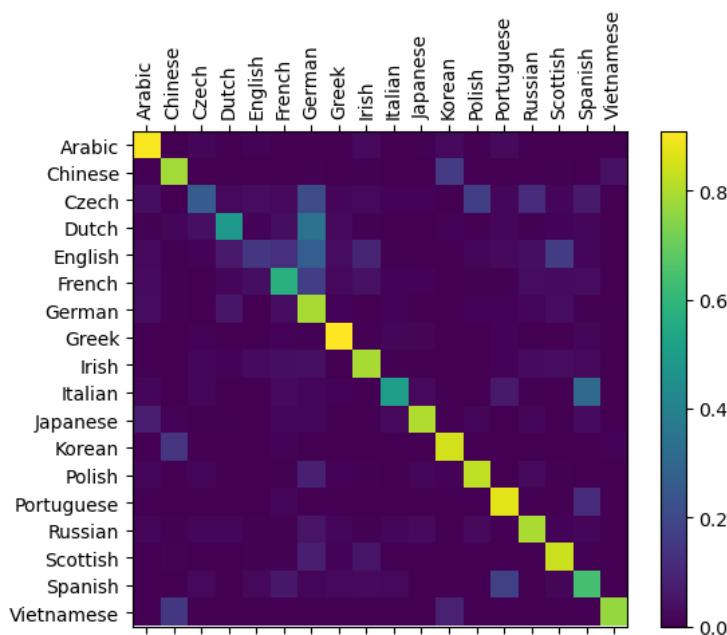


图 2.9: 预测矩阵图

3 为什么 LSTM 网络的性能优于 RNN 网络

LSTM（长短期记忆）网络相对于基本的循环神经网络（RNN）网络具有更好的性能，这主要归因于以下几个原因：

1. 处理长期依赖关系：RNN 网络在处理长序列时，会面临梯度消失或梯度爆炸的问题，导致无法有效捕捉到长期依赖关系。而 LSTM 网络通过引入门控机制，特别是遗忘门和输入门，能够更好地管理和控制信息的流动，从而更好地处理长期依赖关系。通过门控机制，LSTM 可以选择性地保留或遗忘过去的信息，有效地解决了梯度消失和梯度爆炸的问题。
2. 学习多个时间尺度的特征：LSTM 网络能够学习和利用多个时间尺度的特征，这对于处理时间序列数据非常重要。通过 LSTM 的记忆单元和各种门控机制，网络可以灵活地选择性地保留和利用不同时间尺度上的信息。这种能力使得 LSTM 网络能够更好地捕捉序列数据中的长期和短期模式。
3. 避免梯度消失和梯度爆炸：RNN 网络在反向传播过程中容易出现梯度消失或梯度爆炸的问题，导致模型难以收敛。LSTM 网络通过引入门控机制，有效地控制梯度的流动，避免了梯度的不稳定性问题。这使得 LSTM 网络更容易训练和优化。

LSTM 网络通过引入门控机制和记忆单元，能够更好地处理长期依赖关系，学习多个时间尺度的特征，并且避免了梯度消失和梯度爆炸的问题。这些特性使得 LSTM 网络在许多序列建模任务中表现出更好的性能，相对于基本的 RNN 网络而言更具优势。