



南开大学  
Nankai University

# 循环神经网络实验报告

- 
- 姓名：陈睿颖
  - 学号：2013544
  - 专业：计算机科学与技术
- 

## 1. 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

## 2. 实验内容

- 老师提供的原始版本RNN网络结构（可用print(net)打印，复制文字或截图皆可）、在名字识别验证集上的训练loss曲线、准确度曲线图以及预测矩阵图
- 个人实现的LSTM网络结构在上述验证集上的训练loss曲线、准确度曲线图以及预测矩阵图
- 解释为什么LSTM网络的性能优于RNN网络（重点部分）

## 3. 实验步骤

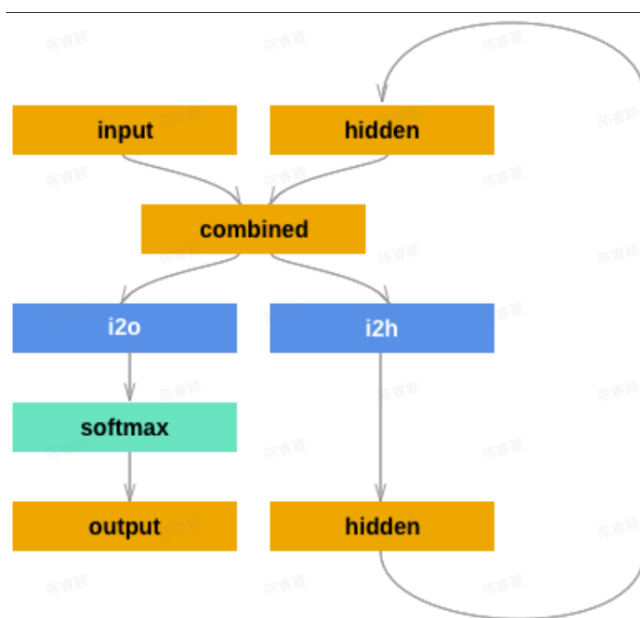
### 3.1 原始版RNN

#### 网络结构

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

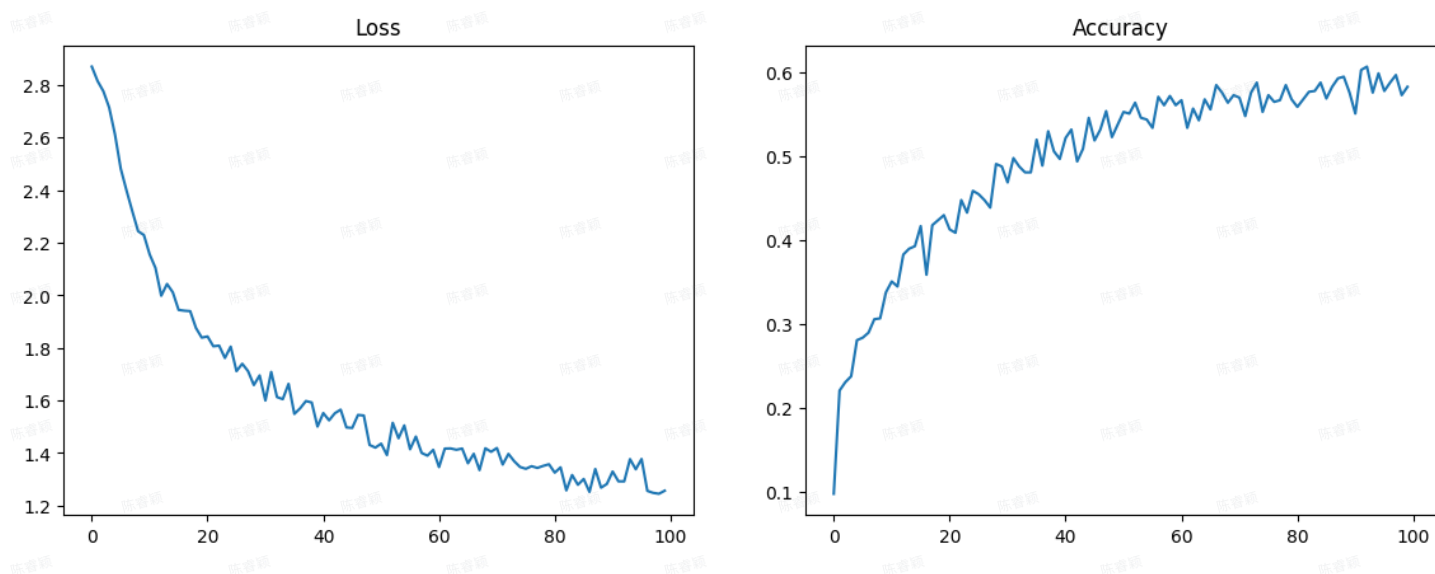
- (i2h) 是一个线性层（全连接层），输入特征数为 185，输出特征数为 128。
- (i2o) 是另一个线性层，输入特征数为 185，输出特征数为 18。
- (softmax) 是一个 LogSoftmax 层，用于将输出转换为概率分布。

这个模型的输入特征维度为 185，即输入数据（例如名称的向量表示）具有 185 个特征。模型将这些特征输入到 (i2h) 和 (i2o) 层中进行计算，然后通过 softmax 操作将输出转换为概率分布。



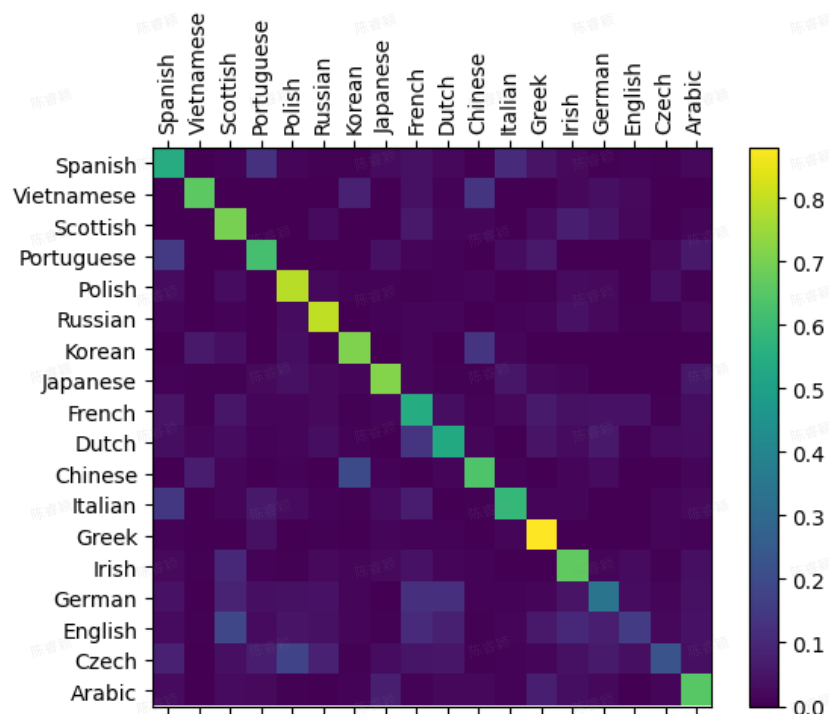
#### 运行结果

##### Loss与准确度曲线



可以看到，经过100000次迭代后，准确率在0.5-0.6之间。

预测矩阵图：



可以发现，预测Greek即希腊名字的准确率是最高的，在0.8以上，而英文和捷克文名字的准确率较低。

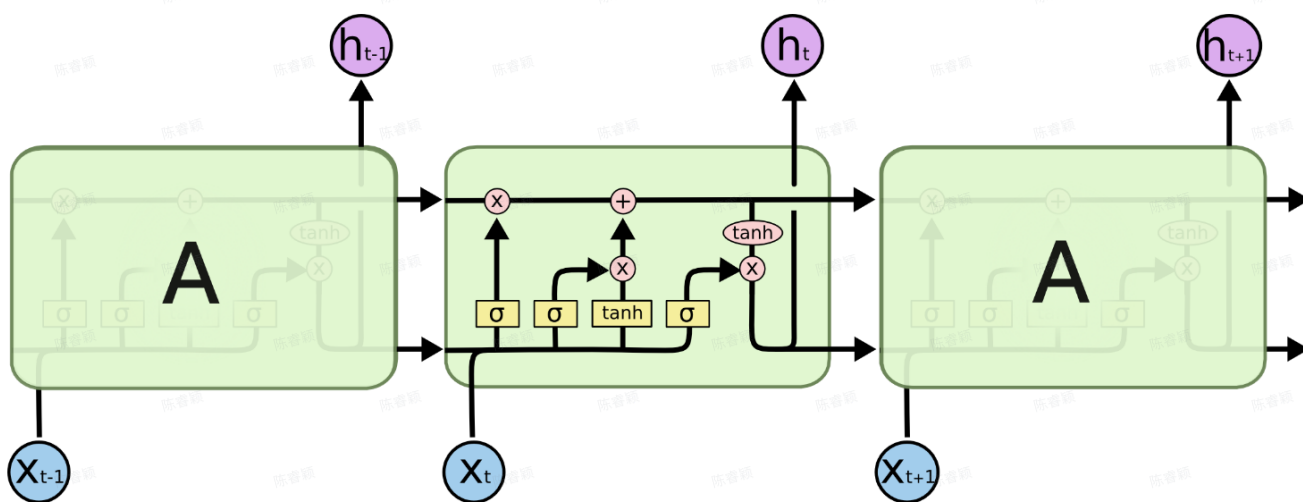
## 3.2 LSTM

LSTM相较于传统的RNN改进了以下几个方面：

1. 处理长期依赖关系：传统的RNN在处理长序列时容易出现梯度消失或梯度爆炸的问题，导致难以捕捉长期依赖关系。而LSTM引入了门控机制，能够选择性地保留或遗忘先前的信息，有效地解决了长期依赖问题，使得网络可以更好地处理长序列数据。

2. 防止梯度消失：LSTM通过遗忘门机制，能够有选择地决定在每个时间步上保留多少信息，从而减少了梯度在时间上的传播损失，防止了梯度消失问题。
3. 处理短期记忆：传统的RNN对于短期记忆相对较好，但在长序列中容易出现信息混淆的问题。LSTM引入了输入门机制，可以选择性地从当前输入中提取新信息，并将其加入到记忆单元中，增强了短期记忆的能力。
4. 建模复杂模式：LSTM的门控机制使其能够适应不同的序列模式。它可以学习何时更新和遗忘信息，何时重置记忆状态，以及何时输出信息，从而更好地建模复杂的序列模式。

LSTM的关键思想是引入了称为"记忆单元"的结构，它可以在网络中有效地传递和存储信息。记忆单元通过门控机制来控制信息的流动，包括遗忘门、输入门和输出门。



LSTM单元[1]

具体来说，LSTM中的遗忘门决定了在当前时间步上需要从记忆单元中丢弃哪些信息，输入门决定了在当前时间步上需要从输入数据中获取哪些新信息，并将其加入到记忆单元中。输出门决定了在当前时间步上从记忆单元中读取哪些信息，并输出到网络的下一层或作为最终的预测结果。

通过这种门控机制，LSTM可以选择性地记住或遗忘先前的信息，并结合当前的输入信息，以有效地捕捉序列数据中的长期依赖关系。

## 代码实现

```
1 class LSTM(nn.Module):
2     def __init__(self, input_size, hidden_size, output_size):
3         super(LSTM, self).__init__()
4         self.lstm = nn.LSTM(input_size, hidden_size, num_layers=2)
5         self.linear = nn.Linear(hidden_size, output_size)
6         self.softmax = nn.LogSoftmax(dim=1)
7
8
```

```

9     def forward(self, x):
10         out, (h_n, c_n) = self.lstm(x)
11         out = (self.linear(out[-1]))
12         out = self.softmax(out)
13         return out
14

```

在模型的初始化方法中，定义了一个 `nn.LSTM` 层和一个线性层 `nn.Linear`，以及一个 `LogSoftmax` 层 `nn.LogSoftmax` 用于计算分类的概率分布。

在前向传播方法中，输入序列 `x` 经过LSTM层，产生输出 `out` 和最后一个时刻的隐状态 `h_n` 和细胞状态 `c_n`。然后，通过线性层将最后一个时刻的输出转换为模型的预测结果。最后，通过 `LogSoftmax` 层计算分类的概率分布，并将其作为模型的输出返回。

这个LSTM模型具有输入大小 `input_size`、隐藏层大小 `hidden_size` 和输出大小 `output_size` 的可调节参数。你可以根据你的具体需求进行调整。

## 网络结构

```

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

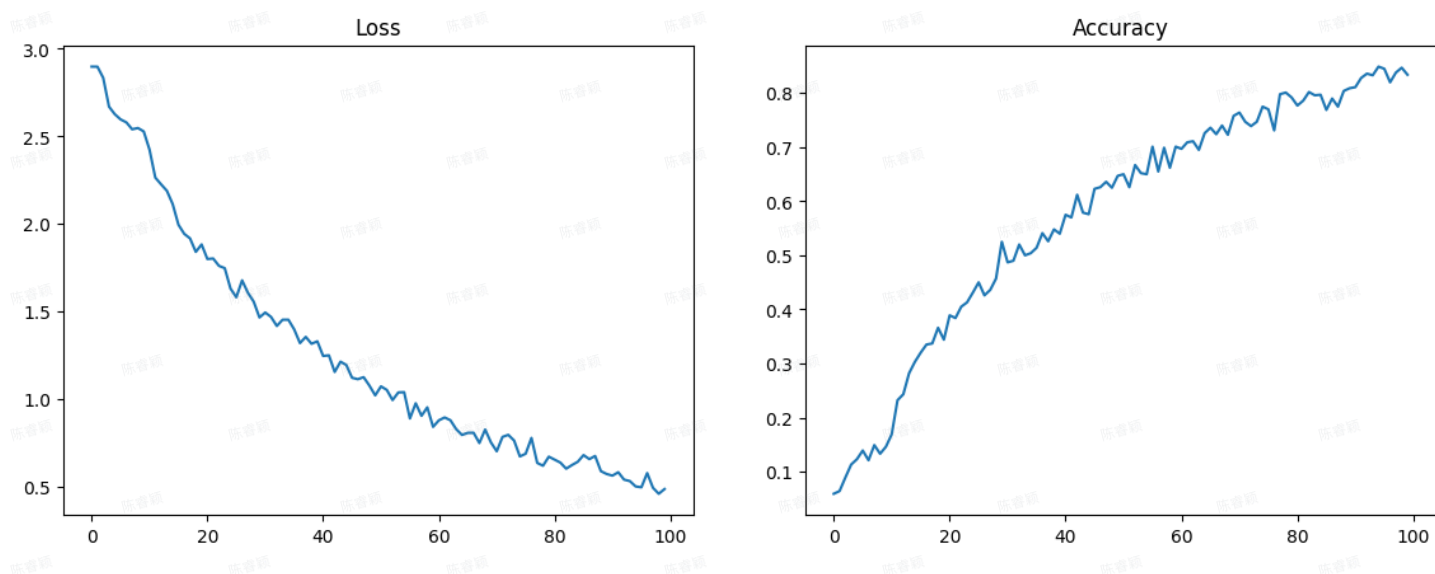
```

该模型包含以下几个部分：

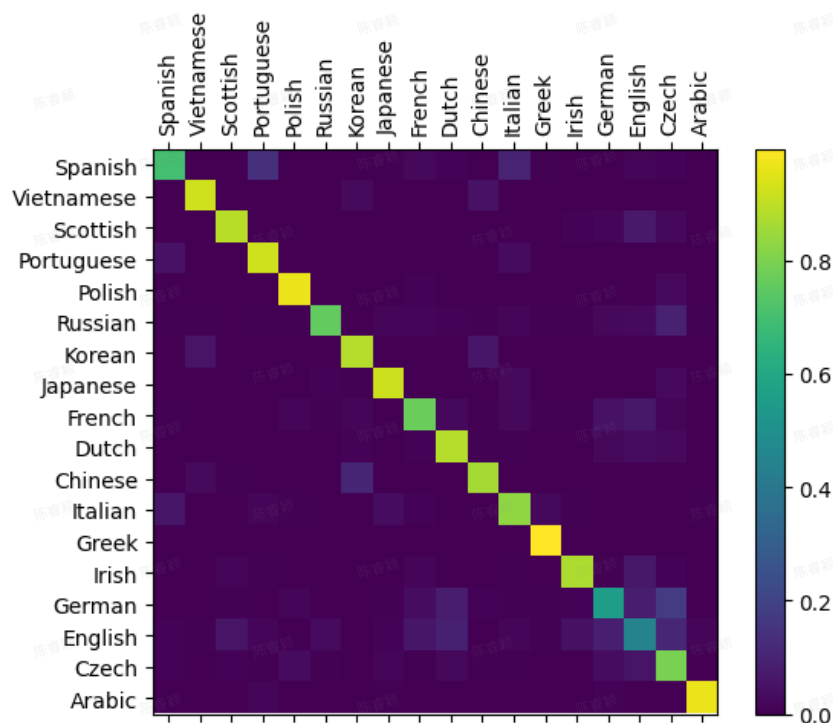
1. `lstm`：一个包含两个LSTM层的LSTM模型，输入维度为57，隐藏层维度为128。这里的 `num_layers=2` 表示有两个LSTM层堆叠在一起。
2. `linear`：一个线性层，将LSTM模型的输出维度（128）转换为输出类别的维度（18）。这个线性层用于将LSTM的最后一个时间步的输出映射到预测结果。
3. `softmax`：一个`LogSoftmax`层，用于计算分类的概率分布。它对线性层的输出进行归一化，使得每个类别的概率都落在0到1之间，并且概率总和为1。

这个LSTM模型的输入维度为57，输出维度为18。它使用LSTM层来处理输入序列，并通过线性层和`LogSoftmax`层产生最终的分类结果。

## 运行结果



可以看到，LSTM的准确率较原始版本的RNN有显著提高，准确率达到0.8以上。



预测矩阵图中也可以看到，各种语言的名字预测准确度均有提升。

### 3.3 思考题：解释为什么LSTM网络的性能优于RNN网络

当序列长度增加时，RNN网络容易出现梯度消失或梯度爆炸的问题。梯度消失指的是梯度在反向传播过程中逐渐变小，导致较早时间步的信息难以传递给后续时间步。梯度爆炸指的是梯度在反向传播过程中逐渐增大，导致数值溢出的问题。

为了解决这些问题，LSTM网络引入了门控机制和细胞状态来控制 and 传递信息。LSTM的更新公式如下：

输入门:

$$i_t = \sigma(W_{hi} \cdot h_{t-1} + W_{xi} \cdot x_t)$$

遗忘门:

$$f_t = \sigma(W_{hf} \cdot h_{t-1} + W_{xf} \cdot x_t)$$

输出门:

$$o_t = \sigma(W_{ho} \cdot h_{t-1} + W_{xo} \cdot x_t)$$

候选细胞状态:

$$g_t = \tanh(W_{hg} \cdot h_{t-1} + W_{xg} \cdot x_t)$$

细胞状态:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t$$

隐藏状态:

$$h_t = o_t \cdot \tanh(c_t)$$

输出结果:

$$y_t = W_{hy} \cdot h_t$$

其中,

$i_t$  是输入门, 控制当前时刻的候选状态  $\tilde{c}_t$  有多少信息需要保存;

$f_t$  是遗忘门, 控制上一个时刻的内部状态  $c_{t-1}$  需要遗忘多少信息;

$o_t$  是输出门, 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  $h_t$ ;

$g_t$  是候选细胞状态,  $c_t$  是细胞状态,  $h_t$  是隐藏状态,  $y_t$  是输出结果,  $\sigma$  是Sigmoid函数。

具体计算过程为:

先利用上一时刻的外部外部状态  $h_{t-1}$  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。

通过引入输入门、遗忘门和输出门, LSTM 网络可以选择性地更新和传递信息。遗忘门允许网络选择性地遗忘过去的信息, 输入门允许网络选择性地更新细胞状态, 输出门允许网络选择性地输出隐藏状

态。细胞状态可以在多个时间步之间传递，使得LSTM网络能够更好地捕捉长期依赖关系。因此，LSTM网络通过门控机制和细胞状态的引入，可以解决RNN网络中的梯度消失和梯度爆炸问题，并能够更好地捕捉长期依赖关系，从而提升了网络的性能。

## 4. 参考文献

[1] “Understanding LSTM Networks.” *Understanding LSTM Networks -- Colah's Blog*, <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>