

循环神经网络

姓名：崔江浩

学号：2011915

专业：计算机科学与技术

1 原始RNN

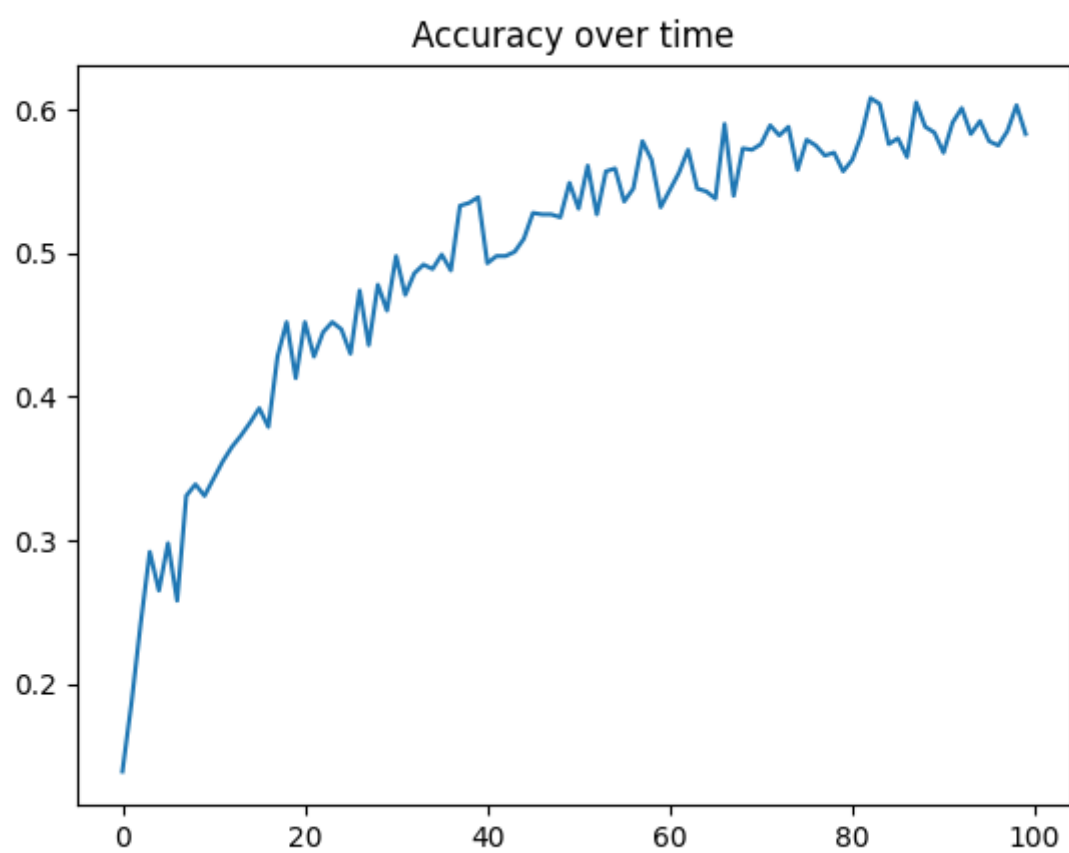
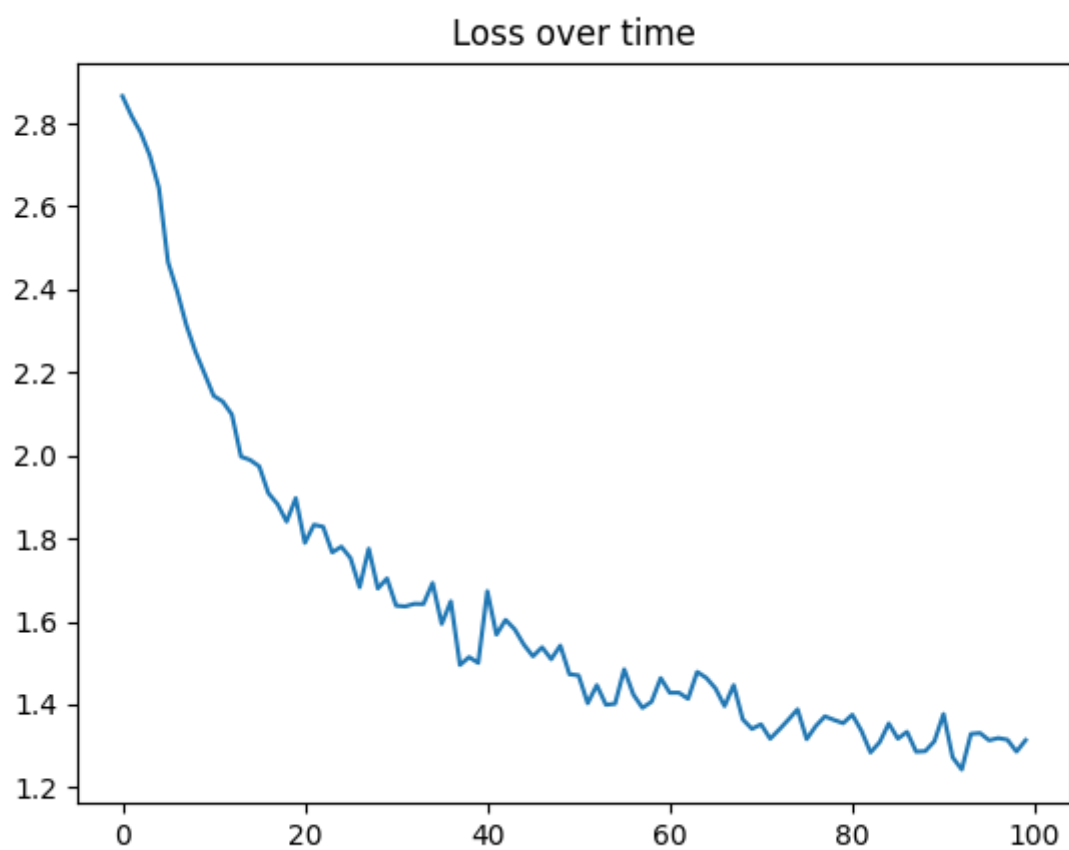
实验代码提供了一个简单的单层RNN实现。该模型由两部分组成：一部分用于根据当前输入和历史隐藏状态更新隐藏状态，另一部分用于根据当前输入和历史隐藏状态生成输出。这两部分可以通过两个简单的线性层来实现。

在训练和推理过程中，将文本中的每个单词依次作为输入输入到RNN中，然后更新隐藏状态并生成输出。除了最后一个输入的输出之外，其他输出可以忽略。最后一个单词的输出用于进行分类预测。

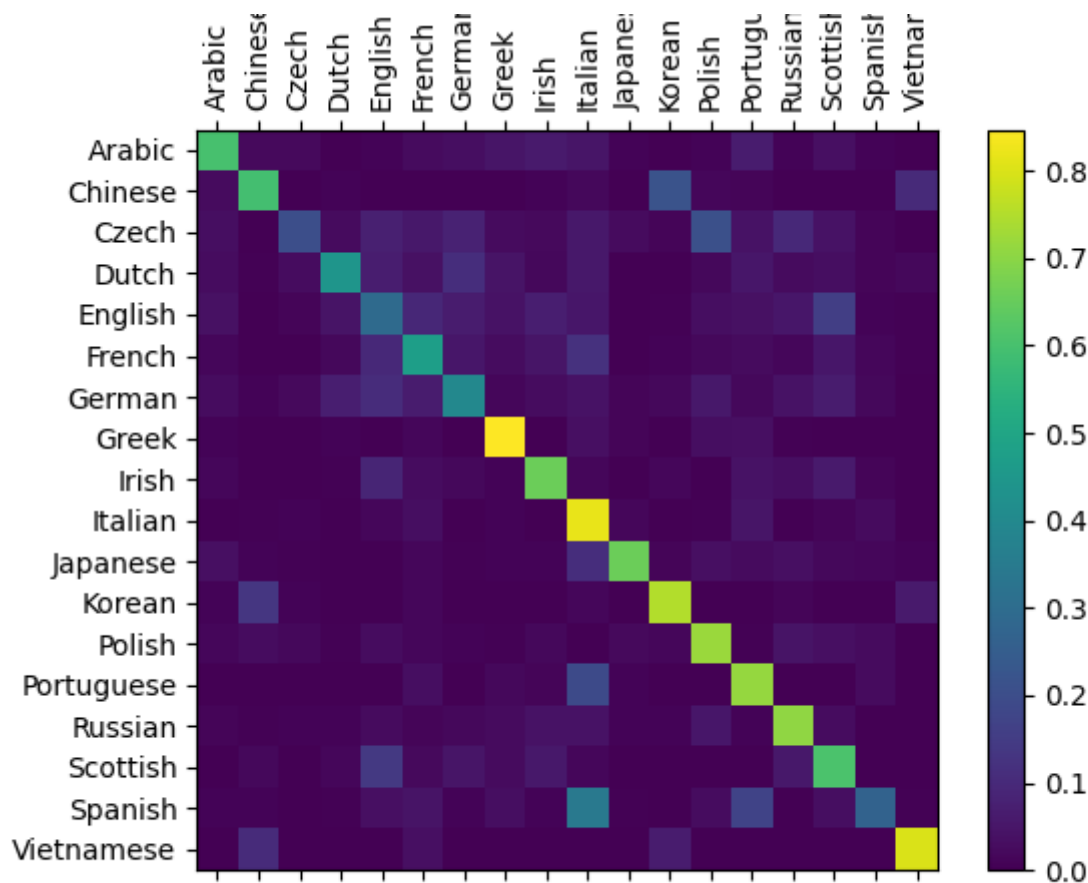
为了确保训练的公平性，采用了Adam优化器，并将学习率设置为0.01。进行了100,000次迭代的训练，在每1,000轮迭代后在验证集上进行验证。下图显示了验证集上的损失和准确率。

通过这种方式，模型通过学习文本序列中的上下文信息，实现了对输入进行分类预测的能力。

1.1 损失曲线、准确率曲线和预测矩阵图



可以观察到模型已经接近收敛，并且准确率达到60%。为进一步探究每一个类别的分类准确率，输出热力图：



从实验结果可以观察到，大多数类别上RNN的分类准确率较高，但在英语和德语类别上表现较差。这是因为英语和德语属于相同的语系，相互之间存在较多的相似性和借鉴，导致人名之间的区分度较低。要实现对这两个类别的有效区分，模型需要对名字进行整体建模，而不仅仅是识别特定的模式。然而，RNN在处理长程依赖关系时的能力有限，因此对于这些相似的名字，模型很难进行准确分类。相比之下，对于希腊和韩国等具有较高区分度的类别，分类准确率接近90%。这是因为这些名字中往往包含一些独特的字符，使得模型在遇到这些特定字符或较短模式时能够做出正确的分类判断。

1.2 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)
)
```

2 LSTM网络

在本次实验那种，自己实现了LSTM网络，为了简单起见，同时也为了和RNN进行公平的比较，这里同样只实现了单层的LSTM。

实现的LSTM网络由以下几部分组成：

1. **初始化方法__init__**：这个方法定义了网络的基本结构，其中包含了输入尺寸（input_size）、隐藏层尺寸（hidden_size）和输出尺寸（output_size）。在初始化方法中，我们创建了输入门、遗忘门、细胞状态门以及输出门等各个组成部分的线性变换。

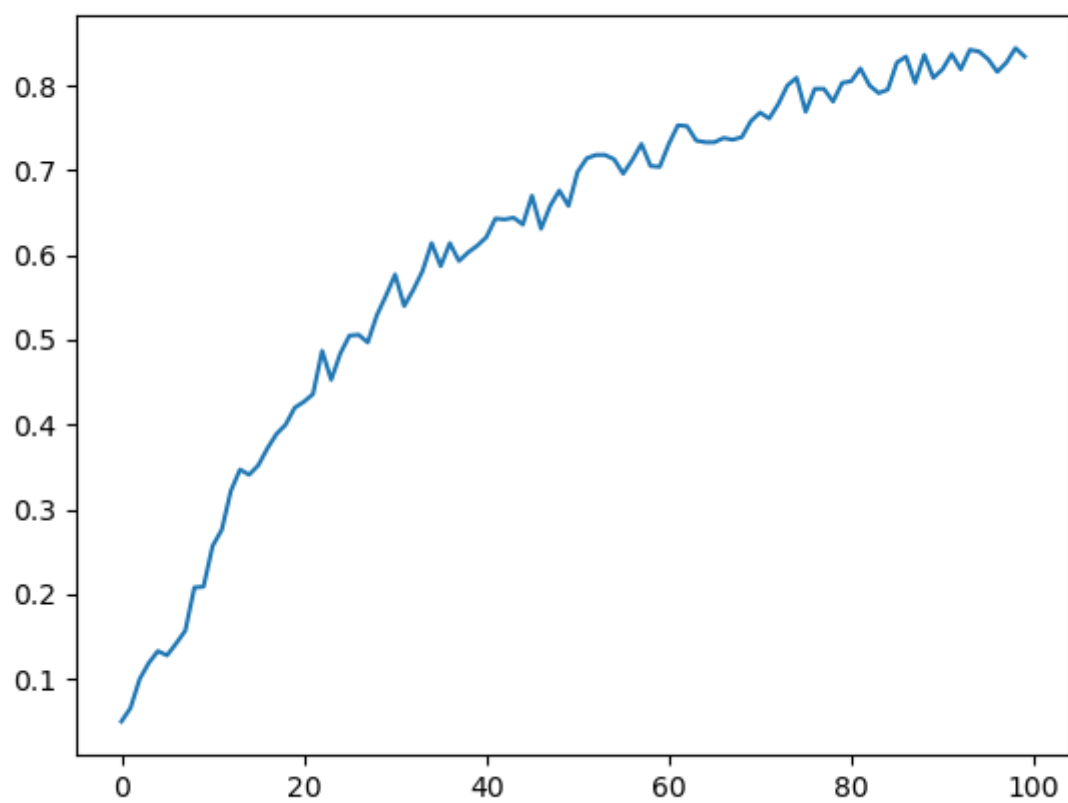
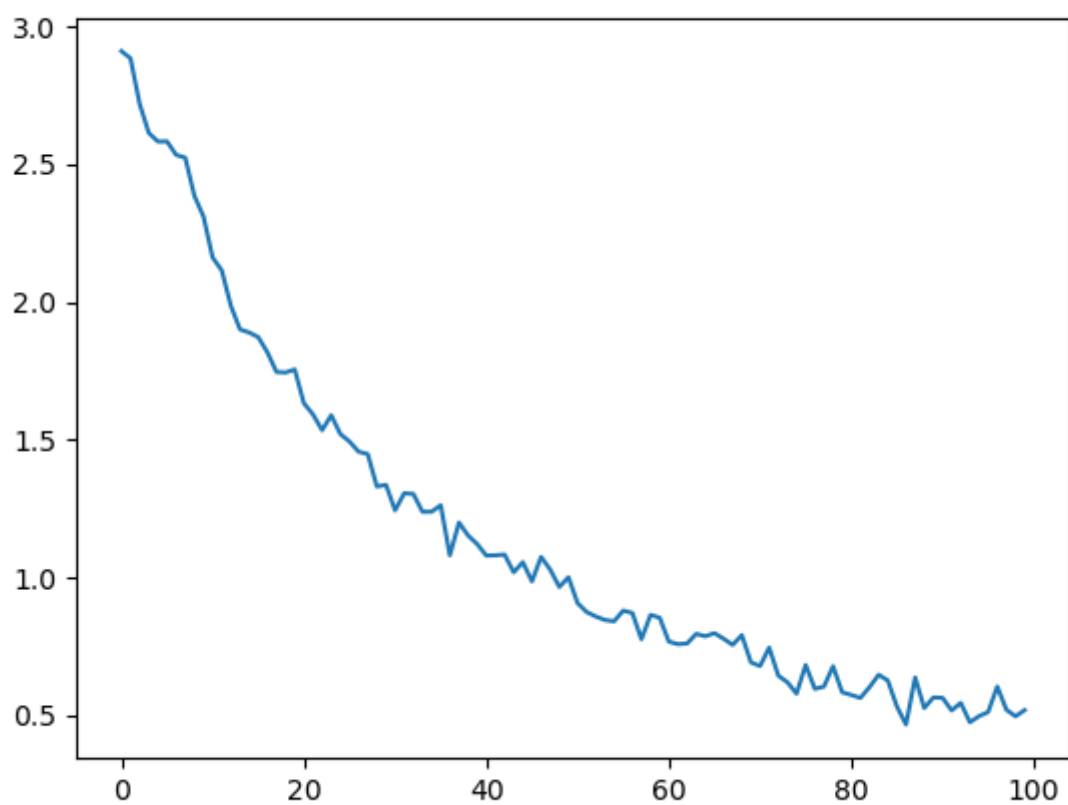
2. **前向传播方法forward**：这个方法用于定义前向传播过程，包括各个门的激活函数及其计算过程。它接受输入数据和上一个时间步的隐藏状态以及细胞状态，然后根据LSTM单元的结构进行计算，最终返回当前时间步的输出、隐藏状态和细胞状态。
3. **初始化隐藏状态方法initHidden**：这个方法用于在开始前向传播之前初始化隐藏状态和细胞状态，通常将它们初始化为零。

具体实现过程如下：

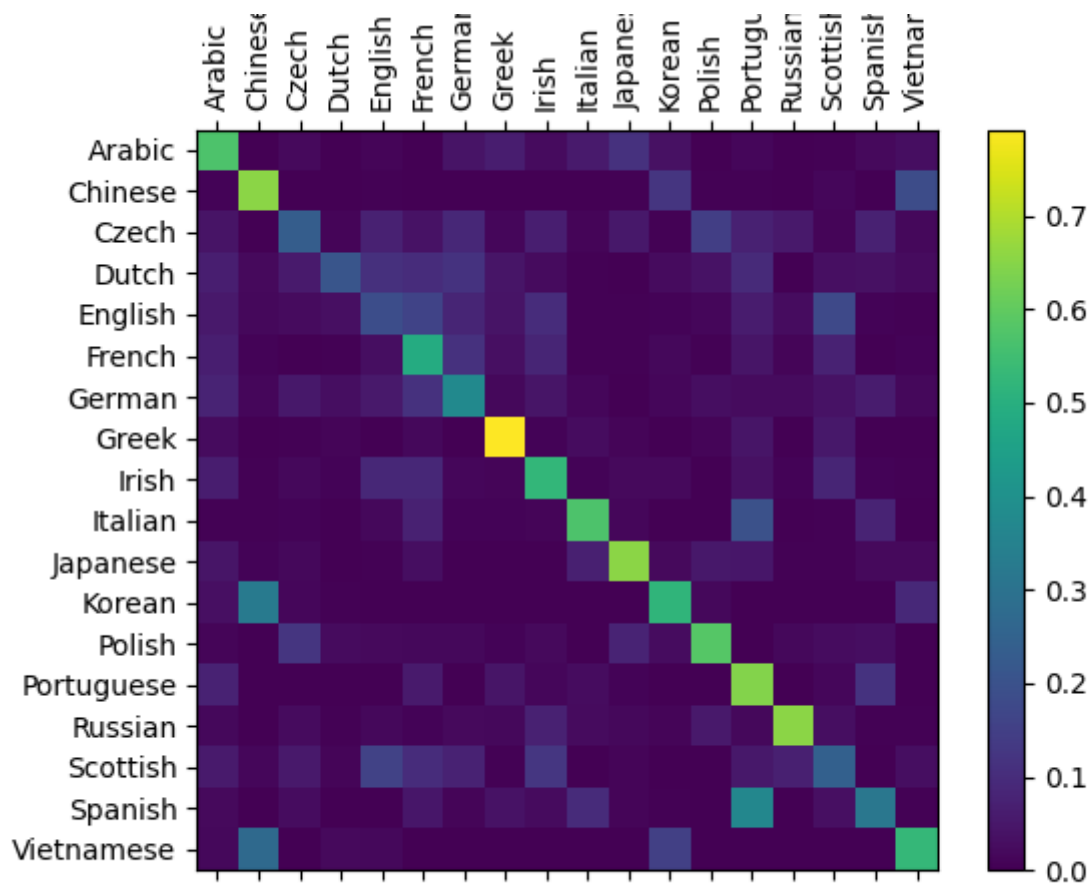
1. **定义输入、隐藏层和输出层大小**：首先，我根据任务需求定义了输入、隐藏层和输出层的大小，这些参数的选取需要根据任务的复杂度和数据的特性来设定。
2. **初始化LSTM网络**：接着，我创建了LSTM网络的实例。在LSTM网络的初始化方法中，我定义了网络的各个组成部分，包括输入门、遗忘门、细胞状态门以及输出门等。
3. **定义前向传播方法**：然后，我定义了前向传播方法，这个方法描述了如何根据输入数据和上一个时间步的隐藏状态以及细胞状态来计算当前时间步的输出、隐藏状态和细胞状态。
4. **定义初始化隐藏状态方法**：在开始前向传播之前，我还定义了一个方法用来初始化隐藏状态和细胞状态，将它们设置为零。
5. **进行训练**：最后，我使用梯度下降算法训练了这个LSTM网络，并保存了训练过程中的损失和准确率以便于后续进行可视化分析。

整个过程中，关键的步骤是理解并实现LSTM网络中的各个组成部分以及它们的功能，这包括输入门（决定当前输入有多少信息量应该被存储到细胞状态中）、遗忘门（决定哪些过去的信息应该被遗忘）、细胞状态门（用于更新细胞状态）和输出门（决定基于细胞状态输出什么信息）。

1.1 损失曲线、准确率曲线和预测矩阵图



可以观察到模型已经接近收敛，并且准确率超过80%。为进一步探究每一个类别的分类准确率，输出热力图：



通过深入分析LSTM在各个类别上的分类准确率，我们可以发现，除了对于"Czech"、"English"和"German"这三个类别，其余的类别的分类准确率已经相当高。即使在这三个类别中，相比于RNN，LSTM也表现出了显著的提升。这主要得益于LSTM增强的长程建模能力，它可以更有效地抽取名字的全局特征，进而能够更好地区分这些相似度较高的类别。

1.2 网络结构

```
LSTM(
  (i2i): Linear(in_features=185, out_features=128, bias=True)
  (i2o): Linear(in_features=185, out_features=18, bias=True)
  (i2f): Linear(in_features=185, out_features=128, bias=True)
  (i2c): Linear(in_features=185, out_features=128, bias=True)
  (i2g): Linear(in_features=185, out_features=128, bias=True)
  (softmax): LogSoftmax(dim=1)
)
```

3 解释为什么LSTM网络的性能优于RNN网络

RNNs（循环神经网络）和LSTMs（长短期记忆网络）都是重复使用神经网络模块的网络类型，它们在处理序列数据（例如时间序列或自然语言）时特别有用。然而，LSTM通常优于普通RNN，主要是由于以下几个原因：

1. **梯度消失问题的处理：** 在训练深度神经网络时，梯度消失问题常常会出现。当网络开始反向传播误差以更新权重时，随着反向传播层数的增加，梯度可能变得越来越小，导致底层（接近输入层）的权重更新困难。这个问题在RNN中尤其严重，因为它们在时间维度上进行反向传播（称为反向传播时间，

BPTT)。LSTM通过引入了一种“记忆单元”来处理这个问题，可以保持或丢弃信息，从而有效地通过时间“传播”或者说保留梯度，解决了RNN的这个问题。

2. **长期依赖问题：** 在处理序列数据时，当前的输出可能依赖于序列中的早期输入。例如，在文本生成任务中，句子的主题可能已在段落的开始处设定。在这种情况下，生成的模型需要有一种方式来“记住”那些早期的输入。RNNs在这方面表现得不好，因为它们往往只能记住最近的输入。相反，LSTM通过它的门控机制（遗忘门、输入门和输出门），可以在很长的时间跨度内记住和访问信息，这使得它们在处理有长期依赖的问题时表现得更好。
3. **更灵活的信息流动：** LSTM通过其独特的结构，提供了更为复杂和灵活的信息流动方式。在LSTM中，信息可以在记忆细胞（cell）中得以存储，并通过门控机制进行调节，允许网络学习在何时添加或者忘记信息。

总的来说，LSTM通过解决RNNs中的梯度消失问题和长期依赖问题，以及提供更为灵活的信息流动方式，从而在处理序列数据时，通常能够提供比普通的RNN更好的性能。