> 姓名：巩羽飞 | 班级：2021669101 | 学号：2021522077 | NNDL Lab A5

用于起名字的 RNN 实验报告

零、快速开始

1. 克隆该项目：

```
git clone git@github.com:Conqueror712/Naming-RNN.git
```

2. 进入根目录：

```
cd Naming-RNN
```

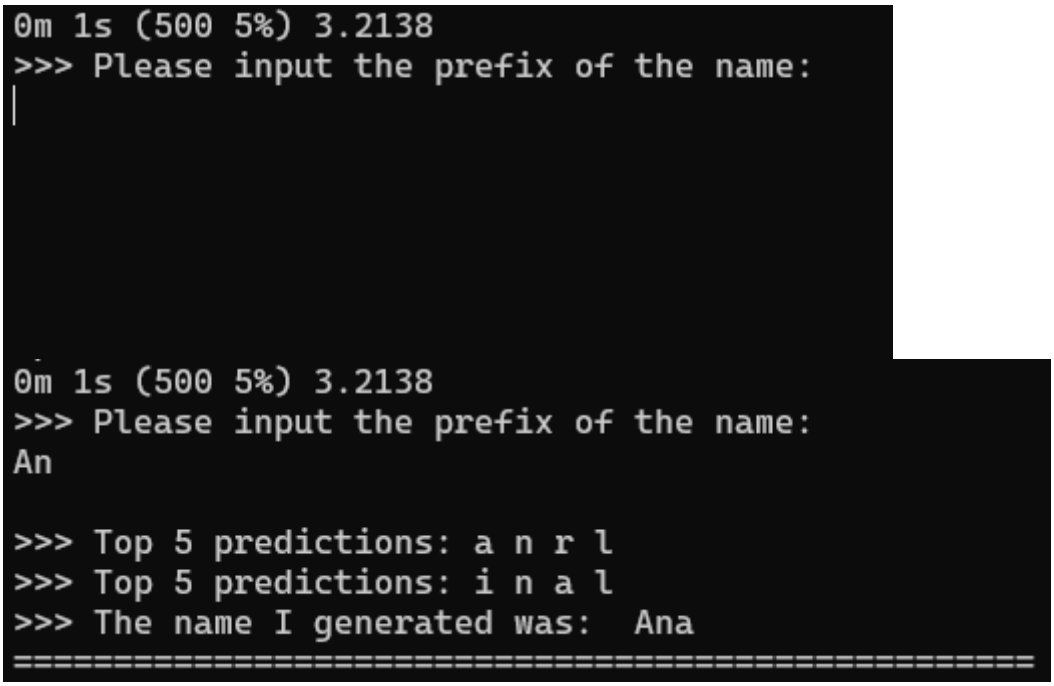
3. 下载相关依赖：

```
pip install -r requirements.txt
```

4. 运行程序：

```
python main.py
```

如果一切正常，您将会看到这样的界面，您只需要输入一个名字的前缀（甚至可以是一个字母），程序就会自动为您起名字：



您可以输入多次，直到进度到达 100%，这时模型会被保存到本地。

一、实验目的

- ✅采用已有的英文名字，训练一个 RNN，实现一个起名字的计算机程序，当输入名字的第一个或前几个字母时，程序自动生成后续的字母，直到生成一个名字的结束符。
- ✅采用可视化技术，绘制出模型为每个时刻预测的前 5 个最可能的候选字母。

二、模型原理

该模型是一个简单的 RNN，包含一个隐藏层和一个输出层。隐藏层使用线性变换将输入和上一个隐藏状态结合起来，输出层使用线性变换生成输出。输出然后通过 Softmax 函数传递，产生可能的下一个字符的概率分布。

更进一步地，模型的隐藏层使用的是线性变换，公式如下：

- $hidden = i2h(input\_combined)$

其中，$input\_combined$ 是输入和上一个隐藏状态的组合，$i2h$ 是一个线性层，用于将输入和隐藏状态映射到新的隐藏状态。

输出层的计算公式如下：

- $output = i2o(input\_combined)$

其中，$i2o$ 是另一个线性层，用于将输入和隐藏状态映射到输出。

然后，模型将隐藏状态和输出组合，并通过另一个线性层 o2o 进行处理：

- $output\_combined = torch.cat((hidden, output), 1)$
- $output = o2o(output\_combined)$

接着，模型对输出进行 $dropout$ 操作以防止过拟合，并通过 $softmax$ 函数将输出转换为概率分布：

- $output = dropout(output)$
- $output = softmax(output)$

三、数据集

包含 8000 多个英文名字，下载链接，当然你也可以直接使用本 Repo 中的数据。

需要注意的是，如果您是直接从链接下载的数据集，请删除位于开始位置的说明行，保证文件中只有名字，并且是一行一个的格式，形如：

```
Abagael
Abagail
Abbe
Abbey
Abbi
...
```

四、实验设置

4.1 训练过程

该模型使用随机梯度下降 SGD 进行训练，学习率为 0.005。使用的损失函数是负对数似然损失，适用于具有多个类别的分类问题。

训练过程中，从数据集中随机选择一个姓名，模型被训练以预测给定前一个字符的情况下下一个字符。模型的输入是姓名中字符的 One-Hot 表示，目标是姓名中实际的下一个字符。

训练过程重复进行 10000 次迭代，并且每 500 次迭代打印一次损失。模型的损失随时间的变化也会进行可视化。

4.2 生成过程

为了生成一个新的姓名，模型从一个用户在终端输入的前缀开始，根据前缀生成下一个字符。新生成的字符然后添加到前缀中，该过程重复进行，直到模型生成一个字符串结束 EOS 标记或达到姓名的最大长度。

五、实验结果分析

事实上，可以根据模型生成的姓名的多样性和质量来评估模型的性能。一个好的模型应该能够生成训练数据中不存在的各种姓名，而且这些姓名应该符合英语姓名的一般模式。

本实验可以通过多种方式来改进其性能。例如使用更复杂的模型架构，如 LSTM 或 GRU，可以提高模型对数据中长期依赖关系的捕捉能力。此外，使用更大更多样的训练数据集也可以改善模型的性能，本实验只是展示了一个最基本的例子。

在实际操作中，可以通过调整模型的参数，如隐藏层的大小、学习率、迭代次数等，来优化模型的性能。同时也可以通过增加数据的多样性，如使用不同语言的姓名，来提高模型的泛化能力。