Assignment A4-RNN by ShiKai@20xxxxx

December 2, 2024

任务描述

任务（30分）：构建一个用于起名字的循环神经网络

- 数据：8000多个英文名字
- 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。

附加题（10分）：

- 事实上，你也可以给定结尾的若干个字母，或者随意给出中间的若干个字母，让RNN补全其它字母，从而得到一个完整的名字。请尝试设计并实现一个这样的RNN模型。

- 从模型生成的名字中，挑选你最喜欢的一个，并采用可视化技术，绘制生成这个名字的过程。

背景知识：给定大量的文本数据，可训练一个基于循环神经网络的语言模型，该模型可用于计算一个句子的出现概率（P(w1, w2, ..., wT)），或者根据上文中的词推断某个词作为下一个词的出现概率（P(wt｜w1, w2, ..., wt-1)）。

✅ 基本任务

本任务通过循环神经网络（RNN）构建了一个名字（First Name）生成器，利用现有的英文名字数据集来训练模型。当输入名字的第一个字母或前几个字母时，模型能够自动生成后续的字母，直到生成一个完整的名字。此外，还实现了可视化功能，绘制了模型在每个时间步预测的前五个最可能的候选字母。

✅ 附加任务

在此基础上进一步扩展了生成器的功能，使得模型不仅能够根据开始字母生成名字，还可以接受部分名字作为输入，从而自动补全缺失的字符。此外，添加了可视化技术，绘制了生成名字过程中前五个最可能的字符的概率分布，帮助更直观地理解模型的预测过程。

数据集介绍

本实验使用的姓名数据集来自 name-dataset (../data/name_dataset)。挑选了 `US` 地区的 `First Name` 数据，并对数据进行了一些筛选，去除了其中的拉丁语、希腊语、希伯来语等非英文名字。最终选用的数据集包含了约 8000 个英文名字。

实验设置

1. 数据预处理

数据预处理主要包括以下几个步骤：

- 从 `name-dataset` 的 `US.csv` 文件中读取英文名字
- 对每个名字进行了简单的清洗，确保其字符全部为英文字母
- 创建了字符到索引（`char_to_index`）以及索引到字符（`index_to_char`）的映射，用于在训练过程中将字符转换为数字索引，模型输出的预测结果再通过索引映射回字符

2. 模型架构

本实验中，使用了一个简单的 RNN模型，主要包括以下三个部分：

1. **嵌入层（Embedding Layer）**：将字符转换为固定维度的向量表示，帮助模型更好地理解字符之间的关系。
2. **RNN 层**：循环神经网络用于学习名字中的字符序列，通过时间步（即每个字符）进行处理。
3. **输出层**：根据 RNN 的输出，计算下一个字符的概率分布，并选取概率最高的字符作为生成的下一个字母。

3. 训练过程

- 使用 Adam 优化器（`optim.Adam`）来优化模型参数。
- 损失函数采用交叉熵损失（`nn.CrossEntropyLoss`），用于计算预测字符和真实字符之间的差异。
- 每次训练时，我们从名字中随机选择一个名字，将该名字的前 `n-1` 个字母作为输入，后 `n-1` 个字母作为目标进行训练。

训练过程中，设置了训练迭代次数（`n_iters`）和每多少次打印一次训练损失（`print_every`）。

4. 生成名字

在训练完成后使用模型生成名字，模型通过输入第一个字母（或者部分字母，也扩展了这个过程，使得模型能够接受部分名字作为输入并补全剩余部分）来生成后续字符，直到生成一个完整的名字或者遇到结束符。

实验结果

具体效果见本 PDF 末尾。

1. 训练过程

在训练过程中，模型逐渐收敛，损失函数值逐渐降低，说明模型成功学习了如何根据已有字母生成新的字母。训练过程中，每100次迭代打印损失值，用于观察训练进度。

2. 生成的名字

使用模型生成多个名字，输入部分字母后，模型成功生成了合理的名字。例如，输入部分名字 `"trum"`，模型生成了完整的名字 `"trumiscondrand"`，可以根据需要调整生成的最大长度。生成的名字有时会长于设置的 `max_length`，这是由于模型没有严格限制长度。因此我们对生成函数进行了改进，确保名字生成过程不超过设定的最大长度。

3. 可视化结果

通过绘制生成过程中的每个时刻模型预测的前五个最可能的字符的概率分布，能够看到模型在每个时间步如何逐步优化预测，并逐渐生成完整的名字。

In [155…

```python
import torch
import torch.nn as nn
import torch.optim as optim
import numpy as np
import random, string, csv
import matplotlib.pyplot as plt


device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(device)
```

```
cuda
```

In [156…

```python
#################### Data Preprocessing ####################

name_path = "../data/name_dataset/data/US.csv"

dump_path = "../data/name_dataset/data/US_dump.csv"

names = []

def chk_alph(name:str) -> bool:
    if len(name) == 0:
        return False
    for c in name:
        _ord = ord(c)
        if _ord > 122:
            return False
        if _ord < 65:
            return False
        if _ord > 90 and _ord < 97:
            return False
    return True


def load_names() -> list:
    # load First names from the dataset
    _fst_names = []
    with open(name_path) as f:
        reader = csv.reader(f)
        for row in reader:
            name = row[0]

            # check if the name is in English
            if not chk_alph(name):
                continue

            _fst_names.append(name.lower())
    return _fst_names

def save_names(names) -> None:
    print(f"Saving {len(names)} names to: {dump_path}")
    with open(dump_path, 'w') as f:
        writer = csv.writer(f)
        for name in names:
            writer.writerow([name])

# 创建字符到索引和索引到字符的映射
all_characters = string.ascii_lowercase + " "  # 可以包括空格
n_characters = len(all_characters)

char_to_index = {ch: i for i, ch in enumerate(all_characters)}
index_to_char = {i: ch for i, ch in enumerate(all_characters)}

# 将名字转换为索引表示
def name_to_tensor(name):
    tensor = torch.zeros(len(name), 1, dtype=torch.long)
    for li, letter in enumerate(name):
        tensor[li] = char_to_index[letter]
    return tensor

names = load_names()
save_names(names)
```

```
Saving 30976530 names to: ../data/name_dataset/data/US_dump.csv
```

In [157…

```python
class RNNModel(nn.Module):
    def __init__(self, n_chars, hidden_size=128):
        super(RNNModel, self).__init__()
        self.hidden_size = hidden_size

        # 嵌入层：字符到嵌入空间的映射
        self.embedding = nn.Embedding(n_chars, hidden_size)

        # RNN层
        self.rnn = nn.RNN(hidden_size, hidden_size)
        # 输出层：从RNN的输出到字符的映射
        self.out = nn.Linear(hidden_size, n_chars)

    def forward(self, input, hidden):
        embedded = self.embedding(input).view(1, 1, -1)
        output, hidden = self.rnn(embedded, hidden)
        output = self.out(output[0])
        return output, hidden

    def init_hidden(self):
        return torch.zeros(1, 1, self.hidden_size, device=next(self.paramete
```

In [158…

```
# 训练
def train(model, names, n_iters=1000, print_every=100):
    optimizer = optim.Adam(model.parameters(), lr=0.005)
    criterion = nn.CrossEntropyLoss()

    for iter in range(1, n_iters + 1):
        name = random.choice(names)
        input_tensor = name_to_tensor(name[:-1]).to(device)  # 除去最后一个字
        target_tensor = name_to_tensor(name[1:]).to(device)  # 从第二个字符开

        hidden = model.init_hidden()

        model.zero_grad()

        loss = torch.tensor(0, dtype=torch.float, requires_grad=True)
        for i in range(input_tensor.size(0)):
            input_char = input_tensor[i]
            target_char = target_tensor[i]

            output, hidden = model(input_char, hidden)
            step_loss = criterion(output, target_char.view(-1))  # 单步损失
            loss = loss + step_loss

        loss.backward()
        optimizer.step()

        if iter % print_every == 0:
            print(f'Iteration {iter}/{n_iters} Loss: {loss.item() / len(name
```

In [159…

```
model = RNNModel(n_characters).to(device)
```

In [160…

```
train(model, names, n_iters=10000, print_every=10)
```

```
Iteration 10/10000 Loss: 2.1193
Iteration 20/10000 Loss: 3.4320
Iteration 30/10000 Loss: 2.8734
Iteration 40/10000 Loss: 2.2870
Iteration 50/10000 Loss: 3.2480
Iteration 60/10000 Loss: 2.7778
Iteration 70/10000 Loss: 2.0080
Iteration 80/10000 Loss: 1.7850
Iteration 90/10000 Loss: 2.1272
Iteration 100/10000 Loss: 2.5575
Iteration 110/10000 Loss: 1.8231
Iteration 120/10000 Loss: 0.8215
Iteration 130/10000 Loss: 2.2220
Iteration 140/10000 Loss: 2.4635
Iteration 150/10000 Loss: 1.9412
Iteration 160/10000 Loss: 3.4045
Iteration 170/10000 Loss: 1.9282
Iteration 180/10000 Loss: 1.8443
Iteration 190/10000 Loss: 2.0796
Iteration 200/10000 Loss: 2.6041
Iteration 210/10000 Loss: 1.4681
Iteration 220/10000 Loss: 2.6639
Iteration 230/10000 Loss: 2.5117
Iteration 240/10000 Loss: 1.9225
Iteration 250/10000 Loss: 2.7214
Iteration 260/10000 Loss: 1.9820
Iteration 270/10000 Loss: 1.9147
Iteration 280/10000 Loss: 1.8392
Iteration 290/10000 Loss: 1.9502
Iteration 300/10000 Loss: 1.7882
Iteration 310/10000 Loss: 2.6116
Iteration 320/10000 Loss: 2.1596
Iteration 330/10000 Loss: 1.9158
Iteration 340/10000 Loss: 1.8567
Iteration 350/10000 Loss: 1.9864
Iteration 360/10000 Loss: 3.2270
Iteration 370/10000 Loss: 2.5384
Iteration 380/10000 Loss: 1.8193
Iteration 390/10000 Loss: 1.9804
Iteration 400/10000 Loss: 1.8193
Iteration 410/10000 Loss: 2.1230
Iteration 420/10000 Loss: 2.3310
Iteration 430/10000 Loss: 1.7609
Iteration 440/10000 Loss: 1.0578
Iteration 450/10000 Loss: 1.8504
Iteration 460/10000 Loss: 1.4888
Iteration 470/10000 Loss: 2.2484
Iteration 480/10000 Loss: 1.7449
Iteration 490/10000 Loss: 2.0235
Iteration 500/10000 Loss: 2.1500
Iteration 510/10000 Loss: 2.2402
Iteration 520/10000 Loss: 2.0770
Iteration 530/10000 Loss: 2.1572
Iteration 540/10000 Loss: 2.1393
Iteration 550/10000 Loss: 2.3839
Iteration 560/10000 Loss: 2.8653
Iteration 570/10000 Loss: 2.4503
Iteration 580/10000 Loss: 1.4587
Iteration 590/10000 Loss: 3.0068
Iteration 600/10000 Loss: 2.1354
Iteration 610/10000 Loss: 2.0986
Iteration 620/10000 Loss: 2.5899
Iteration 630/10000 Loss: 1.9264
Iteration 640/10000 Loss: 2.4949
Iteration 650/10000 Loss: 1.7798
Iteration 660/10000 Loss: 2.3533
Iteration 670/10000 Loss: 1.4833
Iteration 680/10000 Loss: 2.2321
Iteration 690/10000 Loss: 2.3146
Iteration 700/10000 Loss: 2.6490
Iteration 710/10000 Loss: 1.9407
Iteration 720/10000 Loss: 2.9299
Iteration 730/10000 Loss: 0.9491
Iteration 740/10000 Loss: 1.1944
Iteration 750/10000 Loss: 2.7259
Iteration 760/10000 Loss: 0.6227
Iteration 770/10000 Loss: 2.1650
Iteration 780/10000 Loss: 1.8449
Iteration 790/10000 Loss: 2.0477
Iteration 800/10000 Loss: 1.5358
Iteration 810/10000 Loss: 1.5835
Iteration 820/10000 Loss: 2.1186
Iteration 830/10000 Loss: 2.5499
Iteration 840/10000 Loss: 1.1283
Iteration 850/10000 Loss: 0.7860
Iteration 860/10000 Loss: 2.5294
Iteration 870/10000 Loss: 1.2302
Iteration 880/10000 Loss: 1.5034
Iteration 890/10000 Loss: 1.7980
Iteration 900/10000 Loss: 2.2643
Iteration 910/10000 Loss: 2.5689
Iteration 920/10000 Loss: 2.2926
Iteration 930/10000 Loss: 2.2420
Iteration 940/10000 Loss: 2.1237
Iteration 950/10000 Loss: 2.3731
Iteration 960/10000 Loss: 2.2431
Iteration 970/10000 Loss: 2.3062
Iteration 980/10000 Loss: 1.0890
Iteration 990/10000 Loss: 2.2583
Iteration 1000/10000 Loss: 1.2043
Iteration 1010/10000 Loss: 1.4514
Iteration 1020/10000 Loss: 2.7405
Iteration 1030/10000 Loss: 2.9026
Iteration 1040/10000 Loss: 1.1421
Iteration 1050/10000 Loss: 1.8412
Iteration 1060/10000 Loss: 1.2709
Iteration 1070/10000 Loss: 2.3273
Iteration 1080/10000 Loss: 1.8241
Iteration 1090/10000 Loss: 1.4669
Iteration 1100/10000 Loss: 2.2823
Iteration 1110/10000 Loss: 2.1648
Iteration 1120/10000 Loss: 1.5595
Iteration 1130/10000 Loss: 0.9990
Iteration 1140/10000 Loss: 1.9654
Iteration 1150/10000 Loss: 2.5718
Iteration 1160/10000 Loss: 2.1196
Iteration 1170/10000 Loss: 1.7628
Iteration 1180/10000 Loss: 2.1185
Iteration 1190/10000 Loss: 2.0873
Iteration 1200/10000 Loss: 1.8595
Iteration 1210/10000 Loss: 1.6792
Iteration 1220/10000 Loss: 1.5408
Iteration 1230/10000 Loss: 1.0194
Iteration 1240/10000 Loss: 2.0686
Iteration 1250/10000 Loss: 2.2735
Iteration 1260/10000 Loss: 1.6037
Iteration 1270/10000 Loss: 1.5475
Iteration 1280/10000 Loss: 1.9731
Iteration 1290/10000 Loss: 2.2654
Iteration 1300/10000 Loss: 1.5181
Iteration 1310/10000 Loss: 2.8102
Iteration 1320/10000 Loss: 2.1031
Iteration 1330/10000 Loss: 2.2000
Iteration 1340/10000 Loss: 1.4515
Iteration 1350/10000 Loss: 2.6153
Iteration 1360/10000 Loss: 2.4803
Iteration 1370/10000 Loss: 1.7543
Iteration 1380/10000 Loss: 0.7942
Iteration 1390/10000 Loss: 2.8850
Iteration 1400/10000 Loss: 2.8757
Iteration 1410/10000 Loss: 1.6620
Iteration 1420/10000 Loss: 1.3351
Iteration 1430/10000 Loss: 2.0661
Iteration 1440/10000 Loss: 1.3069
Iteration 1450/10000 Loss: 2.2317
Iteration 1460/10000 Loss: 2.3027
Iteration 1470/10000 Loss: 2.2076
Iteration 1480/10000 Loss: 1.6449
Iteration 1490/10000 Loss: 1.7804
Iteration 1500/10000 Loss: 1.7337
Iteration 1510/10000 Loss: 1.2855
Iteration 1520/10000 Loss: 1.1723
Iteration 1530/10000 Loss: 1.4397
Iteration 1540/10000 Loss: 2.3252
Iteration 1550/10000 Loss: 1.9214
Iteration 1560/10000 Loss: 1.5542
Iteration 1570/10000 Loss: 1.4105
Iteration 1580/10000 Loss: 1.7258
Iteration 1590/10000 Loss: 2.8337
Iteration 1600/10000 Loss: 1.5542
Iteration 1610/10000 Loss: 1.4566
Iteration 1620/10000 Loss: 0.5264
Iteration 1630/10000 Loss: 0.5175
Iteration 1640/10000 Loss: 2.3989
Iteration 1650/10000 Loss: 2.0888
Iteration 1660/10000 Loss: 1.6694
Iteration 1670/10000 Loss: 1.4531
Iteration 1680/10000 Loss: 1.2951
Iteration 1690/10000 Loss: 2.4221
Iteration 1700/10000 Loss: 1.2545
Iteration 1710/10000 Loss: 2.2619
Iteration 1720/10000 Loss: 2.9482
Iteration 1730/10000 Loss: 1.6491
Iteration 1740/10000 Loss: 2.1565
Iteration 1750/10000 Loss: 1.9459
Iteration 1760/10000 Loss: 1.8602
Iteration 1770/10000 Loss: 1.9345
Iteration 1780/10000 Loss: 3.6914
Iteration 1790/10000 Loss: 2.1281
Iteration 1800/10000 Loss: 2.6177
Iteration 1810/10000 Loss: 1.7849
Iteration 1820/10000 Loss: 1.2446
Iteration 1830/10000 Loss: 1.7041
Iteration 1840/10000 Loss: 1.5753
Iteration 1850/10000 Loss: 2.0573
Iteration 1860/10000 Loss: 1.7751
Iteration 1870/10000 Loss: 1.7853
Iteration 1880/10000 Loss: 1.1787
Iteration 1890/10000 Loss: 2.5853
Iteration 1900/10000 Loss: 1.5191
Iteration 1910/10000 Loss: 1.4871
Iteration 1920/10000 Loss: 1.3514
Iteration 1930/10000 Loss: 3.0203
Iteration 1940/10000 Loss: 2.7546
Iteration 1950/10000 Loss: 0.7069
Iteration 1960/10000 Loss: 1.9355
Iteration 1970/10000 Loss: 1.7749
Iteration 1980/10000 Loss: 2.3930
Iteration 1990/10000 Loss: 1.8756
Iteration 2000/10000 Loss: 2.3621
Iteration 2010/10000 Loss: 1.7155
Iteration 2020/10000 Loss: 0.4815
Iteration 2030/10000 Loss: 2.8462
Iteration 2040/10000 Loss: 2.5023
Iteration 2050/10000 Loss: 2.3871
Iteration 2060/10000 Loss: 2.4388
Iteration 2070/10000 Loss: 0.6891
Iteration 2080/10000 Loss: 2.3948
Iteration 2090/10000 Loss: 1.7731
Iteration 2100/10000 Loss: 0.7641
Iteration 2110/10000 Loss: 1.8367
Iteration 2120/10000 Loss: 1.8145
Iteration 2130/10000 Loss: 2.8915
Iteration 2140/10000 Loss: 2.4368
Iteration 2150/10000 Loss: 1.2892
Iteration 2160/10000 Loss: 1.8526
Iteration 2170/10000 Loss: 1.9472
Iteration 2180/10000 Loss: 1.7616
Iteration 2190/10000 Loss: 2.7397
Iteration 2200/10000 Loss: 1.1205
Iteration 2210/10000 Loss: 1.4635
Iteration 2220/10000 Loss: 2.8643
Iteration 2230/10000 Loss: 3.4720
Iteration 2240/10000 Loss: 1.7176
Iteration 2250/10000 Loss: 2.1643
Iteration 2260/10000 Loss: 1.8624
Iteration 2270/10000 Loss: 1.4819
Iteration 2280/10000 Loss: 1.6182
Iteration 2290/10000 Loss: 2.2754
Iteration 2300/10000 Loss: 1.6126
Iteration 2310/10000 Loss: 1.7552
Iteration 2320/10000 Loss: 1.1558
Iteration 2330/10000 Loss: 1.7008
Iteration 2340/10000 Loss: 0.6376
Iteration 2350/10000 Loss: 2.1242
Iteration 2360/10000 Loss: 1.2777
Iteration 2370/10000 Loss: 3.2651
Iteration 2380/10000 Loss: 2.1622
Iteration 2390/10000 Loss: 1.8009
Iteration 2400/10000 Loss: 2.2585
Iteration 2410/10000 Loss: 1.5950
Iteration 2420/10000 Loss: 1.4978
Iteration 2430/10000 Loss: 1.9554
Iteration 2440/10000 Loss: 2.7743
Iteration 2450/10000 Loss: 1.5433
Iteration 2460/10000 Loss: 2.4604
Iteration 2470/10000 Loss: 2.7962
Iteration 2480/10000 Loss: 1.5149
Iteration 2490/10000 Loss: 1.7563
Iteration 2500/10000 Loss: 1.9452
Iteration 2510/10000 Loss: 2.4128
Iteration 2520/10000 Loss: 2.7342
Iteration 2530/10000 Loss: 1.7118
Iteration 2540/10000 Loss: 1.5535
Iteration 2550/10000 Loss: 0.7384
Iteration 2560/10000 Loss: 1.8482
Iteration 2570/10000 Loss: 2.1207
Iteration 2580/10000 Loss: 2.0108
Iteration 2590/10000 Loss: 2.4031
Iteration 2600/10000 Loss: 1.9832
Iteration 2610/10000 Loss: 2.0142
Iteration 2620/10000 Loss: 2.4064
Iteration 2630/10000 Loss: 2.1185
Iteration 2640/10000 Loss: 1.4869
Iteration 2650/10000 Loss: 1.8549
Iteration 2660/10000 Loss: 2.3384
Iteration 2670/10000 Loss: 2.1475
Iteration 2680/10000 Loss: 2.0782
Iteration 2690/10000 Loss: 2.6738
Iteration 2700/10000 Loss: 3.0292
Iteration 2710/10000 Loss: 1.1252
Iteration 2720/10000 Loss: 0.9970
Iteration 2730/10000 Loss: 0.7354
Iteration 2740/10000 Loss: 1.0990
Iteration 2750/10000 Loss: 3.0513
Iteration 2760/10000 Loss: 2.0681
Iteration 2770/10000 Loss: 0.8132
Iteration 2780/10000 Loss: 2.9781
Iteration 2790/10000 Loss: 1.9321
Iteration 2800/10000 Loss: 1.2846
Iteration 2810/10000 Loss: 1.2194
Iteration 2820/10000 Loss: 1.6003
Iteration 2830/10000 Loss: 1.3979
Iteration 2840/10000 Loss: 1.2495
Iteration 2850/10000 Loss: 1.0788
Iteration 2860/10000 Loss: 1.4260
Iteration 2870/10000 Loss: 2.3568
Iteration 2880/10000 Loss: 1.8638
Iteration 2890/10000 Loss: 2.0929
Iteration 2900/10000 Loss: 1.8552
Iteration 2910/10000 Loss: 1.6575
Iteration 2920/10000 Loss: 1.5315
Iteration 2930/10000 Loss: 1.1403
Iteration 2940/10000 Loss: 1.2528
Iteration 2950/10000 Loss: 1.5296
Iteration 2960/10000 Loss: 1.3923
Iteration 2970/10000 Loss: 1.6998
Iteration 2980/10000 Loss: 2.3063
Iteration 2990/10000 Loss: 1.8745
Iteration 3000/10000 Loss: 2.8770
Iteration 3010/10000 Loss: 3.0593
Iteration 3020/10000 Loss: 1.3336
Iteration 3030/10000 Loss: 2.0212
Iteration 3040/10000 Loss: 1.4829
Iteration 3050/10000 Loss: 1.4998
Iteration 3060/10000 Loss: 1.8835
Iteration 3070/10000 Loss: 1.6041
Iteration 3080/10000 Loss: 1.6496
Iteration 3090/10000 Loss: 2.2162
Iteration 3100/10000 Loss: 2.5168
Iteration 3110/10000 Loss: 1.6337
Iteration 3120/10000 Loss: 1.9016
Iteration 3130/10000 Loss: 2.2304
Iteration 3140/10000 Loss: 1.3044
Iteration 3150/10000 Loss: 1.7990
Iteration 3160/10000 Loss: 2.1127
Iteration 3170/10000 Loss: 1.5024
Iteration 3180/10000 Loss: 1.5526
Iteration 3190/10000 Loss: 2.7461
Iteration 3200/10000 Loss: 1.7689
Iteration 3210/10000 Loss: 1.8937
Iteration 3220/10000 Loss: 1.9542
Iteration 3230/10000 Loss: 1.7539
Iteration 3240/10000 Loss: 1.2983
Iteration 3250/10000 Loss: 2.3465
Iteration 3260/10000 Loss: 2.8894
Iteration 3270/10000 Loss: 2.6455
Iteration 3280/10000 Loss: 2.0157
Iteration 3290/10000 Loss: 2.4689
Iteration 3300/10000 Loss: 2.3997
Iteration 3310/10000 Loss: 3.2404
Iteration 3320/10000 Loss: 2.0080
Iteration 3330/10000 Loss: 1.4242
Iteration 3340/10000 Loss: 1.4006
Iteration 3350/10000 Loss: 2.9383
Iteration 3360/10000 Loss: 1.9694
Iteration 3370/10000 Loss: 2.5573
Iteration 3380/10000 Loss: 2.1672
Iteration 3390/10000 Loss: 1.8185
Iteration 3400/10000 Loss: 2.6010
Iteration 3410/10000 Loss: 1.0137
Iteration 3420/10000 Loss: 2.5631
Iteration 3430/10000 Loss: 1.6794
Iteration 3440/10000 Loss: 2.4083
Iteration 3450/10000 Loss: 2.3351
Iteration 3460/10000 Loss: 2.2744
Iteration 3470/10000 Loss: 1.6476
Iteration 3480/10000 Loss: 2.6865
Iteration 3490/10000 Loss: 3.2020
Iteration 3500/10000 Loss: 1.6064
Iteration 3510/10000 Loss: 1.2188
Iteration 3520/10000 Loss: 1.8167
Iteration 3530/10000 Loss: 2.3254
Iteration 3540/10000 Loss: 1.3120
Iteration 3550/10000 Loss: 2.0325
Iteration 3560/10000 Loss: 1.5090
Iteration 3570/10000 Loss: 1.7574
Iteration 3580/10000 Loss: 2.3120
Iteration 3590/10000 Loss: 2.6477
Iteration 3600/10000 Loss: 0.9460
Iteration 3610/10000 Loss: 4.1261
Iteration 3620/10000 Loss: 0.9837
Iteration 3630/10000 Loss: 1.8129
Iteration 3640/10000 Loss: 2.1636
Iteration 3650/10000 Loss: 1.9235
Iteration 3660/10000 Loss: 0.6978
Iteration 3670/10000 Loss: 1.5943
Iteration 3680/10000 Loss: 1.5656
Iteration 3690/10000 Loss: 2.4436
Iteration 3700/10000 Loss: 1.5261
Iteration 3710/10000 Loss: 1.1862
Iteration 3720/10000 Loss: 0.9501
Iteration 3730/10000 Loss: 1.5030
Iteration 3740/10000 Loss: 1.3644
Iteration 3750/10000 Loss: 1.5325
Iteration 3760/10000 Loss: 1.5036
Iteration 3770/10000 Loss: 1.6953
Iteration 3780/10000 Loss: 1.9275
Iteration 3790/10000 Loss: 2.2853
Iteration 3800/10000 Loss: 0.9203
Iteration 3810/10000 Loss: 2.3546
Iteration 3820/10000 Loss: 1.1716
Iteration 3830/10000 Loss: 2.9610
Iteration 3840/10000 Loss: 1.4991
Iteration 3850/10000 Loss: 1.9851
Iteration 3860/10000 Loss: 1.0255
Iteration 3870/10000 Loss: 1.9353
Iteration 3880/10000 Loss: 1.9498
Iteration 3890/10000 Loss: 1.3153
Iteration 3900/10000 Loss: 3.2106
Iteration 3910/10000 Loss: 2.9845
Iteration 3920/10000 Loss: 1.9524
Iteration 3930/10000 Loss: 2.3643
Iteration 3940/10000 Loss: 3.0044
Iteration 3950/10000 Loss: 2.2461
Iteration 3960/10000 Loss: 1.9296
Iteration 3970/10000 Loss: 0.8547
Iteration 3980/10000 Loss: 0.9506
Iteration 3990/10000 Loss: 2.5165
Iteration 4000/10000 Loss: 2.1758
Iteration 4010/10000 Loss: 1.1277
Iteration 4020/10000 Loss: 2.7924
Iteration 4030/10000 Loss: 2.7372
Iteration 4040/10000 Loss: 1.3772
Iteration 4050/10000 Loss: 2.1290
Iteration 4060/10000 Loss: 2.3791
Iteration 4070/10000 Loss: 1.7424
Iteration 4080/10000 Loss: 2.0902
Iteration 4090/10000 Loss: 2.2305
Iteration 4100/10000 Loss: 2.0061
Iteration 4110/10000 Loss: 1.0321
Iteration 4120/10000 Loss: 1.6918
Iteration 4130/10000 Loss: 1.4946
Iteration 4140/10000 Loss: 1.5244
Iteration 4150/10000 Loss: 2.5744
Iteration 4160/10000 Loss: 1.7276
Iteration 4170/10000 Loss: 2.8786
Iteration 4180/10000 Loss: 2.1662
Iteration 4190/10000 Loss: 2.0463
Iteration 4200/10000 Loss: 0.9311
Iteration 4210/10000 Loss: 2.2242
Iteration 4220/10000 Loss: 2.3716
Iteration 4230/10000 Loss: 2.9675
Iteration 4240/10000 Loss: 2.0357
Iteration 4250/10000 Loss: 1.5771
Iteration 4260/10000 Loss: 2.1126
Iteration 4270/10000 Loss: 1.3952
Iteration 4280/10000 Loss: 2.2127
Iteration 4290/10000 Loss: 2.2782
Iteration 4300/10000 Loss: 1.0053
Iteration 4310/10000 Loss: 2.6524
Iteration 4320/10000 Loss: 1.1509
Iteration 4330/10000 Loss: 2.4632
Iteration 4340/10000 Loss: 3.0662
Iteration 4350/10000 Loss: 2.6071
Iteration 4360/10000 Loss: 1.2291
Iteration 4370/10000 Loss: 1.2171
Iteration 4380/10000 Loss: 0.9977
Iteration 4390/10000 Loss: 1.8653
Iteration 4400/10000 Loss: 1.5348
Iteration 4410/10000 Loss: 1.8715
Iteration 4420/10000 Loss: 1.9313
Iteration 4430/10000 Loss: 1.9755
Iteration 4440/10000 Loss: 2.2953
Iteration 4450/10000 Loss: 1.0538
Iteration 4460/10000 Loss: 1.4663
Iteration 4470/10000 Loss: 2.4723
Iteration 4480/10000 Loss: 0.5502
Iteration 4490/10000 Loss: 3.1074
Iteration 4500/10000 Loss: 1.1304
Iteration 4510/10000 Loss: 1.9781
Iteration 4520/10000 Loss: 1.3898
Iteration 4530/10000 Loss: 1.9062
Iteration 4540/10000 Loss: 1.1169
Iteration 4550/10000 Loss: 1.9349
Iteration 4560/10000 Loss: 1.7307
Iteration 4570/10000 Loss: 3.4038
Iteration 4580/10000 Loss: 3.6908
Iteration 4590/10000 Loss: 2.2621
Iteration 4600/10000 Loss: 2.8988
Iteration 4610/10000 Loss: 3.1483
Iteration 4620/10000 Loss: 2.3486
Iteration 4630/10000 Loss: 2.3772
Iteration 4640/10000 Loss: 1.0707
Iteration 4650/10000 Loss: 1.1931
Iteration 4660/10000 Loss: 1.9017
Iteration 4670/10000 Loss: 1.7971
Iteration 4680/10000 Loss: 3.3358
Iteration 4690/10000 Loss: 1.5906
Iteration 4700/10000 Loss: 1.8234
Iteration 4710/10000 Loss: 0.6708
Iteration 4720/10000 Loss: 1.4954
Iteration 4730/10000 Loss: 1.5348
Iteration 4740/10000 Loss: 1.2761
Iteration 4750/10000 Loss: 2.6177
Iteration 4760/10000 Loss: 0.8358
Iteration 4770/10000 Loss: 1.3230
Iteration 4780/10000 Loss: 1.1066
Iteration 4790/10000 Loss: 3.0558
Iteration 4800/10000 Loss: 1.5667
Iteration 4810/10000 Loss: 1.8170
Iteration 4820/10000 Loss: 1.9981
Iteration 4830/10000 Loss: 2.9029
Iteration 4840/10000 Loss: 1.6648
Iteration 4850/10000 Loss: 0.8579
Iteration 4860/10000 Loss: 1.6626
Iteration 4870/10000 Loss: 1.9271
Iteration 4880/10000 Loss: 1.8844
Iteration 4890/10000 Loss: 1.3876
Iteration 4900/10000 Loss: 1.2601
Iteration 4910/10000 Loss: 2.1057
Iteration 4920/10000 Loss: 1.6432
Iteration 4930/10000 Loss: 2.2057
Iteration 4940/10000 Loss: 2.6077
Iteration 4950/10000 Loss: 2.5536
Iteration 4960/10000 Loss: 1.4925
Iteration 4970/10000 Loss: 2.8023
Iteration 4980/10000 Loss: 1.7257
Iteration 4990/10000 Loss: 1.4901
Iteration 5000/10000 Loss: 1.7102
Iteration 5010/10000 Loss: 3.3504
Iteration 5020/10000 Loss: 0.9317
Iteration 5030/10000 Loss: 1.0415
Iteration 5040/10000 Loss: 2.0652
Iteration 5050/10000 Loss: 2.9589
Iteration 5060/10000 Loss: 1.2702
Iteration 5070/10000 Loss: 1.2365
Iteration 5080/10000 Loss: 2.7194
Iteration 5090/10000 Loss: 3.6957
Iteration 5100/10000 Loss: 1.6299
Iteration 5110/10000 Loss: 2.1622
Iteration 5120/10000 Loss: 1.8017
Iteration 5130/10000 Loss: 0.6612
Iteration 5140/10000 Loss: 1.9422
Iteration 5150/10000 Loss: 1.5025
Iteration 5160/10000 Loss: 1.7541
Iteration 5170/10000 Loss: 1.8000
Iteration 5180/10000 Loss: 1.8774
Iteration 5190/10000 Loss: 3.4439
Iteration 5200/10000 Loss: 1.3562
Iteration 5210/10000 Loss: 1.7734
Iteration 5220/10000 Loss: 2.2258
Iteration 5230/10000 Loss: 2.6372
Iteration 5240/10000 Loss: 1.7550
Iteration 5250/10000 Loss: 2.4023
Iteration 5260/10000 Loss: 1.3655
Iteration 5270/10000 Loss: 2.1395
Iteration 5280/10000 Loss: 2.2216
Iteration 5290/10000 Loss: 1.4930
Iteration 5300/10000 Loss: 1.7563
Iteration 5310/10000 Loss: 1.6624
Iteration 5320/10000 Loss: 2.2005
Iteration 5330/10000 Loss: 1.3371
Iteration 5340/10000 Loss: 2.7322
Iteration 5350/10000 Loss: 1.7863
Iteration 5360/10000 Loss: 1.6321
Iteration 5370/10000 Loss: 2.1315
Iteration 5380/10000 Loss: 2.1441
Iteration 5390/10000 Loss: 1.1157
Iteration 5400/10000 Loss: 2.0030
Iteration 5410/10000 Loss: 3.1314
Iteration 5420/10000 Loss: 1.7244
Iteration 5430/10000 Loss: 1.8981
Iteration 5440/10000 Loss: 1.7459
Iteration 5450/10000 Loss: 3.3173
Iteration 5460/10000 Loss: 2.0618
Iteration 5470/10000 Loss: 1.6699
Iteration 5480/10000 Loss: 1.4782
Iteration 5490/10000 Loss: 1.9545
Iteration 5500/10000 Loss: 2.4124
Iteration 5510/10000 Loss: 2.4435
Iteration 5520/10000 Loss: 1.2226
Iteration 5530/10000 Loss: 1.6803
Iteration 5540/10000 Loss: 2.1373
Iteration 5550/10000 Loss: 1.8042
Iteration 5560/10000 Loss: 1.4520
Iteration 5570/10000 Loss: 3.0697
Iteration 5580/10000 Loss: 1.1073
Iteration 5590/10000 Loss: 2.3703
Iteration 5600/10000 Loss: 1.9086
Iteration 5610/10000 Loss: 1.0855
Iteration 5620/10000 Loss: 1.8592
Iteration 5630/10000 Loss: 2.3517
Iteration 5640/10000 Loss: 1.6711
Iteration 5650/10000 Loss: 1.4404
Iteration 5660/10000 Loss: 5.3929
Iteration 5670/10000 Loss: 1.0308
Iteration 5680/10000 Loss: 2.2039
Iteration 5690/10000 Loss: 1.7550
Iteration 5700/10000 Loss: 2.3557
Iteration 5710/10000 Loss: 1.7739
Iteration 5720/10000 Loss: 2.2575
Iteration 5730/10000 Loss: 1.0289
Iteration 5740/10000 Loss: 1.3087
Iteration 5750/10000 Loss: 2.7961
Iteration 5760/10000 Loss: 0.9061
Iteration 5770/10000 Loss: 1.9338
Iteration 5780/10000 Loss: 1.1367
Iteration 5790/10000 Loss: 1.4896
Iteration 5800/10000 Loss: 1.6877
Iteration 5810/10000 Loss: 1.7646
Iteration 5820/10000 Loss: 2.6647
Iteration 5830/10000 Loss: 1.3737
Iteration 5840/10000 Loss: 2.1735
Iteration 5850/10000 Loss: 2.0813
Iteration 5860/10000 Loss: 1.8160
Iteration 5870/10000 Loss: 1.2143
Iteration 5880/10000 Loss: 1.7389
Iteration 5890/10000 Loss: 1.2031
Iteration 5900/10000 Loss: 3.4828
Iteration 5910/10000 Loss: 1.3776
Iteration 5920/10000 Loss: 1.7346
Iteration 5930/10000 Loss: 1.6890
Iteration 5940/10000 Loss: 1.4581
Iteration 5950/10000 Loss: 1.1437
Iteration 5960/10000 Loss: 1.1572
Iteration 5970/10000 Loss: 2.0754
Iteration 5980/10000 Loss: 1.1659
Iteration 5990/10000 Loss: 1.5684
Iteration 6000/10000 Loss: 2.0022
Iteration 6010/10000 Loss: 3.2453
Iteration 6020/10000 Loss: 2.1790
Iteration 6030/10000 Loss: 2.0282
Iteration 6040/10000 Loss: 1.7519
Iteration 6050/10000 Loss: 2.3588
Iteration 6060/10000 Loss: 2.6463
Iteration 6070/10000 Loss: 1.6803
Iteration 6080/10000 Loss: 3.0874
Iteration 6090/10000 Loss: 1.2999
Iteration 6100/10000 Loss: 0.6968
Iteration 6110/10000 Loss: 1.9445
Iteration 6120/10000 Loss: 1.3843
Iteration 6130/10000 Loss: 1.5711
Iteration 6140/10000 Loss: 1.6432
Iteration 6150/10000 Loss: 1.8502
Iteration 6160/10000 Loss: 1.3725
Iteration 6170/10000 Loss: 2.6010
Iteration 6180/10000 Loss: 1.3000
Iteration 6190/10000 Loss: 1.2218
Iteration 6200/10000 Loss: 1.8177
Iteration 6210/10000 Loss: 2.5091
Iteration 6220/10000 Loss: 2.3143
Iteration 6230/10000 Loss: 1.3170
Iteration 6240/10000 Loss: 1.2168
Iteration 6250/10000 Loss: 1.2937
Iteration 6260/10000 Loss: 2.9881
Iteration 6270/10000 Loss: 1.2438
Iteration 6280/10000 Loss: 1.1572
Iteration 6290/10000 Loss: 1.3140
Iteration 6300/10000 Loss: 0.9420
Iteration 6310/10000 Loss: 2.2679
Iteration 6320/10000 Loss: 1.8467
Iteration 6330/10000 Loss: 1.8863
Iteration 6340/10000 Loss: 4.4212
Iteration 6350/10000 Loss: 2.3719
Iteration 6360/10000 Loss: 2.0181
Iteration 6370/10000 Loss: 1.9757
Iteration 6380/10000 Loss: 2.7232
Iteration 6390/10000 Loss: 1.4848
Iteration 6400/10000 Loss: 0.7380
Iteration 6410/10000 Loss: 1.7633
Iteration 6420/10000 Loss: 2.0246
Iteration 6430/10000 Loss: 2.5824
Iteration 6440/10000 Loss: 1.6893
Iteration 6450/10000 Loss: 3.3134
Iteration 6460/10000 Loss: 1.4821
Iteration 6470/10000 Loss: 1.6281
Iteration 6480/10000 Loss: 2.1779
Iteration 6490/10000 Loss: 1.2786
Iteration 6500/10000 Loss: 0.9965
Iteration 6510/10000 Loss: 2.3289
Iteration 6520/10000 Loss: 2.5397
Iteration 6530/10000 Loss: 1.8724
Iteration 6540/10000 Loss: 1.4699
Iteration 6550/10000 Loss: 2.2526
Iteration 6560/10000 Loss: 3.4133
Iteration 6570/10000 Loss: 1.2100
Iteration 6580/10000 Loss: 1.6828
Iteration 6590/10000 Loss: 2.7821
Iteration 6600/10000 Loss: 2.3853
Iteration 6610/10000 Loss: 1.7445
Iteration 6620/10000 Loss: 2.9547
Iteration 6630/10000 Loss: 1.8734
Iteration 6640/10000 Loss: 3.9109
Iteration 6650/10000 Loss: 2.1998
Iteration 6660/10000 Loss: 2.2986
Iteration 6670/10000 Loss: 0.7658
Iteration 6680/10000 Loss: 2.6862
Iteration 6690/10000 Loss: 2.7155
Iteration 6700/10000 Loss: 0.8340
Iteration 6710/10000 Loss: 1.2133
Iteration 6720/10000 Loss: 1.3470
Iteration 6730/10000 Loss: 2.8088
Iteration 6740/10000 Loss: 1.7081
Iteration 6750/10000 Loss: 1.7887
Iteration 6760/10000 Loss: 0.7854
Iteration 6770/10000 Loss: 1.0473
Iteration 6780/10000 Loss: 1.7824
Iteration 6790/10000 Loss: 0.7692
Iteration 6800/10000 Loss: 2.4996
Iteration 6810/10000 Loss: 2.4530
Iteration 6820/10000 Loss: 3.6841
Iteration 6830/10000 Loss: 1.9670
Iteration 6840/10000 Loss: 2.2658
Iteration 6850/10000 Loss: 2.8096
Iteration 6860/10000 Loss: 2.9161
Iteration 6870/10000 Loss: 1.8017
Iteration 6880/10000 Loss: 2.5674
Iteration 6890/10000 Loss: 2.1250
Iteration 6900/10000 Loss: 1.8936
Iteration 6910/10000 Loss: 2.8754
Iteration 6920/10000 Loss: 3.6757
Iteration 6930/10000 Loss: 2.1110
Iteration 6940/10000 Loss: 1.2756
Iteration 6950/10000 Loss: 2.4196
Iteration 6960/10000 Loss: 1.2216
Iteration 6970/10000 Loss: 2.0691
Iteration 6980/10000 Loss: 2.4963
Iteration 6990/10000 Loss: 3.0061
Iteration 7000/10000 Loss: 1.8285
Iteration 7010/10000 Loss: 1.7608
Iteration 7020/10000 Loss: 1.9381
Iteration 7030/10000 Loss: 2.5141
Iteration 7040/10000 Loss: 2.5771
Iteration 7050/10000 Loss: 2.1437
Iteration 7060/10000 Loss: 1.5562
Iteration 7070/10000 Loss: 1.5145
Iteration 7080/10000 Loss: 3.2564
Iteration 7090/10000 Loss: 1.9039
Iteration 7100/10000 Loss: 1.0032
Iteration 7110/10000 Loss: 1.3907
Iteration 7120/10000 Loss: 1.3174
Iteration 7130/10000 Loss: 2.1866
Iteration 7140/10000 Loss: 1.7842
Iteration 7150/10000 Loss: 1.1778
Iteration 7160/10000 Loss: 1.4157
Iteration 7170/10000 Loss: 2.8726
Iteration 7180/10000 Loss: 1.4860
Iteration 7190/10000 Loss: 1.0187
Iteration 7200/10000 Loss: 2.3824
Iteration 7210/10000 Loss: 2.7008
Iteration 7220/10000 Loss: 1.4230
Iteration 7230/10000 Loss: 2.5318
Iteration 7240/10000 Loss: 2.2796
Iteration 7250/10000 Loss: 1.3895
Iteration 7260/10000 Loss: 1.6942
Iteration 7270/10000 Loss: 0.9638
Iteration 7280/10000 Loss: 2.3730
Iteration 7290/10000 Loss: 1.0701
Iteration 7300/10000 Loss: 2.6416
Iteration 7310/10000 Loss: 1.7383
Iteration 7320/10000 Loss: 1.9229
Iteration 7330/10000 Loss: 0.9145
Iteration 7340/10000 Loss: 2.4958
Iteration 7350/10000 Loss: 1.9782
Iteration 7360/10000 Loss: 1.0516
Iteration 7370/10000 Loss: 2.1826
Iteration 7380/10000 Loss: 2.7041
Iteration 7390/10000 Loss: 1.7776
Iteration 7400/10000 Loss: 1.3955
Iteration 7410/10000 Loss: 2.2660
Iteration 7420/10000 Loss: 1.3059
Iteration 7430/10000 Loss: 1.4642
Iteration 7440/10000 Loss: 2.0743
Iteration 7450/10000 Loss: 2.2054
Iteration 7460/10000 Loss: 1.2140
Iteration 7470/10000 Loss: 2.3645
Iteration 7480/10000 Loss: 1.7734
Iteration 7490/10000 Loss: 3.1798
Iteration 7500/10000 Loss: 1.0854
Iteration 7510/10000 Loss: 1.8412
Iteration 7520/10000 Loss: 1.5286
Iteration 7530/10000 Loss: 1.4855
Iteration 7540/10000 Loss: 2.8885
Iteration 7550/10000 Loss: 1.8151
Iteration 7560/10000 Loss: 1.9999
Iteration 7570/10000 Loss: 3.4244
Iteration 7580/10000 Loss: 1.5371
Iteration 7590/10000 Loss: 2.0780
Iteration 7600/10000 Loss: 3.1094
Iteration 7610/10000 Loss: 1.4582
Iteration 7620/10000 Loss: 2.1309
Iteration 7630/10000 Loss: 2.9855
Iteration 7640/10000 Loss: 2.6915
Iteration 7650/10000 Loss: 2.6327
Iteration 7660/10000 Loss: 3.4916
Iteration 7670/10000 Loss: 1.6090
Iteration 7680/10000 Loss: 1.4995
Iteration 7690/10000 Loss: 1.5668
Iteration 7700/10000 Loss: 1.3589
Iteration 7710/10000 Loss: 2.0898
Iteration 7720/10000 Loss: 2.3478
Iteration 7730/10000 Loss: 1.3080
Iteration 7740/10000 Loss: 1.4752
Iteration 7750/10000 Loss: 2.7625
Iteration 7760/10000 Loss: 3.0076
Iteration 7770/10000 Loss: 2.5216
Iteration 7780/10000 Loss: 2.0664
Iteration 7790/10000 Loss: 1.2852
Iteration 7800/10000 Loss: 1.8559
Iteration 7810/10000 Loss: 0.8834
Iteration 7820/10000 Loss: 1.1296
Iteration 7830/10000 Loss: 3.4871
Iteration 7840/10000 Loss: 1.2025
Iteration 7850/10000 Loss: 0.8546
Iteration 7860/10000 Loss: 3.0689
Iteration 7870/10000 Loss: 1.5684
Iteration 7880/10000 Loss: 1.0690
Iteration 7890/10000 Loss: 2.1906
Iteration 7900/10000 Loss: 1.4350
Iteration 7910/10000 Loss: 0.7670
Iteration 7920/10000 Loss: 2.5209
Iteration 7930/10000 Loss: 2.1546
Iteration 7940/10000 Loss: 0.6995
Iteration 7950/10000 Loss: 2.3992
Iteration 7960/10000 Loss: 1.7787
Iteration 7970/10000 Loss: 1.6264
Iteration 7980/10000 Loss: 2.1282
Iteration 7990/10000 Loss: 1.4086
Iteration 8000/10000 Loss: 1.9470
Iteration 8010/10000 Loss: 2.2932
Iteration 8020/10000 Loss: 1.9854
Iteration 8030/10000 Loss: 1.8709
Iteration 8040/10000 Loss: 2.5247
Iteration 8050/10000 Loss: 1.0590
Iteration 8060/10000 Loss: 1.0301
Iteration 8070/10000 Loss: 1.7759
Iteration 8080/10000 Loss: 1.5404
Iteration 8090/10000 Loss: 0.8823
Iteration 8100/10000 Loss: 1.7975
Iteration 8110/10000 Loss: 1.8975
Iteration 8120/10000 Loss: 1.5739
Iteration 8130/10000 Loss: 3.1273
Iteration 8140/10000 Loss: 2.7478
Iteration 8150/10000 Loss: 0.7578
Iteration 8160/10000 Loss: 2.3599
Iteration 8170/10000 Loss: 1.4160
Iteration 8180/10000 Loss: 0.7422
Iteration 8190/10000 Loss: 2.3715
Iteration 8200/10000 Loss: 1.0694
Iteration 8210/10000 Loss: 1.2690
Iteration 8220/10000 Loss: 1.8362
Iteration 8230/10000 Loss: 1.4007
Iteration 8240/10000 Loss: 1.2809
Iteration 8250/10000 Loss: 2.2439
Iteration 8260/10000 Loss: 1.3946
Iteration 8270/10000 Loss: 3.7383
Iteration 8280/10000 Loss: 1.3882
Iteration 8290/10000 Loss: 1.2127
Iteration 8300/10000 Loss: 3.2106
Iteration 8310/10000 Loss: 1.3637
Iteration 8320/10000 Loss: 2.3379
Iteration 8330/10000 Loss: 2.0363
Iteration 8340/10000 Loss: 1.7653
Iteration 8350/10000 Loss: 1.4863
Iteration 8360/10000 Loss: 1.2895
Iteration 8370/10000 Loss: 2.1513
Iteration 8380/10000 Loss: 0.9121
Iteration 8390/10000 Loss: 1.5411
Iteration 8400/10000 Loss: 1.9152
Iteration 8410/10000 Loss: 2.0986
Iteration 8420/10000 Loss: 1.2089
Iteration 8430/10000 Loss: 0.8821
Iteration 8440/10000 Loss: 2.1742
Iteration 8450/10000 Loss: 0.9565
Iteration 8460/10000 Loss: 1.9328
Iteration 8470/10000 Loss: 1.5637
Iteration 8480/10000 Loss: 0.5468
Iteration 8490/10000 Loss: 1.1247
Iteration 8500/10000 Loss: 1.3439
Iteration 8510/10000 Loss: 2.8893
Iteration 8520/10000 Loss: 2.5027
Iteration 8530/10000 Loss: 2.5096
Iteration 8540/10000 Loss: 1.8289
Iteration 8550/10000 Loss: 1.0075
Iteration 8560/10000 Loss: 1.7435
Iteration 8570/10000 Loss: 2.1773
Iteration 8580/10000 Loss: 1.5170
Iteration 8590/10000 Loss: 1.2016
Iteration 8600/10000 Loss: 0.9981
Iteration 8610/10000 Loss: 1.4648
Iteration 8620/10000 Loss: 1.6050
Iteration 8630/10000 Loss: 1.5007
Iteration 8640/10000 Loss: 2.4749
Iteration 8650/10000 Loss: 2.0363
Iteration 8660/10000 Loss: 1.5226
Iteration 8670/10000 Loss: 2.1288
Iteration 8680/10000 Loss: 1.9864
Iteration 8690/10000 Loss: 1.7824
Iteration 8700/10000 Loss: 1.6674
Iteration 8710/10000 Loss: 4.2248
Iteration 8720/10000 Loss: 0.5653
Iteration 8730/10000 Loss: 1.6695
Iteration 8740/10000 Loss: 2.2511
Iteration 8750/10000 Loss: 1.8086
Iteration 8760/10000 Loss: 1.9474
Iteration 8770/10000 Loss: 1.4378
Iteration 8780/10000 Loss: 2.1604
Iteration 8790/10000 Loss: 3.1037
Iteration 8800/10000 Loss: 2.0883
Iteration 8810/10000 Loss: 2.1477
Iteration 8820/10000 Loss: 2.3813
Iteration 8830/10000 Loss: 3.0293
Iteration 8840/10000 Loss: 2.2133
Iteration 8850/10000 Loss: 1.3433
Iteration 8860/10000 Loss: 0.9618
Iteration 8870/10000 Loss: 1.4658
Iteration 8880/10000 Loss: 2.4138
Iteration 8890/10000 Loss: 1.4960
Iteration 8900/10000 Loss: 1.1445
Iteration 8910/10000 Loss: 2.9444
Iteration 8920/10000 Loss: 1.8307
Iteration 8930/10000 Loss: 1.5246
Iteration 8940/10000 Loss: 2.6904
Iteration 8950/10000 Loss: 0.7763
Iteration 8960/10000 Loss: 0.5211
Iteration 8970/10000 Loss: 1.6124
Iteration 8980/10000 Loss: 0.3199
Iteration 8990/10000 Loss: 1.6601
Iteration 9000/10000 Loss: 1.4009
Iteration 9010/10000 Loss: 2.4769
Iteration 9020/10000 Loss: 3.1088
Iteration 9030/10000 Loss: 2.9915
Iteration 9040/10000 Loss: 2.8595
Iteration 9050/10000 Loss: 2.5222
Iteration 9060/10000 Loss: 1.9786
Iteration 9070/10000 Loss: 1.1838
Iteration 9080/10000 Loss: 1.1623
Iteration 9090/10000 Loss: 1.2146
Iteration 9100/10000 Loss: 2.0676
Iteration 9110/10000 Loss: 0.9780
Iteration 9120/10000 Loss: 3.0807
Iteration 9130/10000 Loss: 1.2390
Iteration 9140/10000 Loss: 1.1101
Iteration 9150/10000 Loss: 2.5251
Iteration 9160/10000 Loss: 0.7603
Iteration 9170/10000 Loss: 2.2716
Iteration 9180/10000 Loss: 0.9097
Iteration 9190/10000 Loss: 2.6730
Iteration 9200/10000 Loss: 1.5951
Iteration 9210/10000 Loss: 1.0472
Iteration 9220/10000 Loss: 1.5653
Iteration 9230/10000 Loss: 2.1613
Iteration 9240/10000 Loss: 1.4986
Iteration 9250/10000 Loss: 1.1593
Iteration 9260/10000 Loss: 1.4691
Iteration 9270/10000 Loss: 2.1172
Iteration 9280/10000 Loss: 1.5068
Iteration 9290/10000 Loss: 1.7010
Iteration 9300/10000 Loss: 2.4169
Iteration 9310/10000 Loss: 2.3199
Iteration 9320/10000 Loss: 3.2432
Iteration 9330/10000 Loss: 1.9186
Iteration 9340/10000 Loss: 2.1831
Iteration 9350/10000 Loss: 1.2822
Iteration 9360/10000 Loss: 1.8504
Iteration 9370/10000 Loss: 2.0832
Iteration 9380/10000 Loss: 2.5334
Iteration 9390/10000 Loss: 1.7535
Iteration 9400/10000 Loss: 2.3075
Iteration 9410/10000 Loss: 1.1985
Iteration 9420/10000 Loss: 1.3658
Iteration 9430/10000 Loss: 1.7660
Iteration 9440/10000 Loss: 2.3549
Iteration 9450/10000 Loss: 2.1731
Iteration 9460/10000 Loss: 2.0121
Iteration 9470/10000 Loss: 1.0524
Iteration 9480/10000 Loss: 2.1685
Iteration 9490/10000 Loss: 2.3676
Iteration 9500/10000 Loss: 1.7220
Iteration 9510/10000 Loss: 2.0313
Iteration 9520/10000 Loss: 1.6906
Iteration 9530/10000 Loss: 2.5754
Iteration 9540/10000 Loss: 1.3359
Iteration 9550/10000 Loss: 2.1903
Iteration 9560/10000 Loss: 1.5639
Iteration 9570/10000 Loss: 3.5739
Iteration 9580/10000 Loss: 2.8693
Iteration 9590/10000 Loss: 1.8500
Iteration 9600/10000 Loss: 1.6930
Iteration 9610/10000 Loss: 2.2318
Iteration 9620/10000 Loss: 2.1411
Iteration 9630/10000 Loss: 1.8765
Iteration 9640/10000 Loss: 1.4784
Iteration 9650/10000 Loss: 0.7253
Iteration 9660/10000 Loss: 3.2831
Iteration 9670/10000 Loss: 3.6908
Iteration 9680/10000 Loss: 1.1413
Iteration 9690/10000 Loss: 1.0502
Iteration 9700/10000 Loss: 2.4617
Iteration 9710/10000 Loss: 1.7844
Iteration 9720/10000 Loss: 1.4278
Iteration 9730/10000 Loss: 2.3985
Iteration 9740/10000 Loss: 1.4672
Iteration 9750/10000 Loss: 2.0234
Iteration 9760/10000 Loss: 1.8671
Iteration 9770/10000 Loss: 1.6416
Iteration 9780/10000 Loss: 1.7679
Iteration 9790/10000 Loss: 2.3610
Iteration 9800/10000 Loss: 1.2723
Iteration 9810/10000 Loss: 2.8198
Iteration 9820/10000 Loss: 1.6535
Iteration 9830/10000 Loss: 1.4723
Iteration 9840/10000 Loss: 1.3737
Iteration 9850/10000 Loss: 0.8579
Iteration 9860/10000 Loss: 1.4617
Iteration 9870/10000 Loss: 1.9556
Iteration 9880/10000 Loss: 1.5263
Iteration 9890/10000 Loss: 1.7251
Iteration 9900/10000 Loss: 1.6782
Iteration 9910/10000 Loss: 1.1495
Iteration 9920/10000 Loss: 2.4018
Iteration 9930/10000 Loss: 2.0801
Iteration 9940/10000 Loss: 2.9250
Iteration 9950/10000 Loss: 1.9083
Iteration 9960/10000 Loss: 3.7452
Iteration 9970/10000 Loss: 1.5155
Iteration 9980/10000 Loss: 1.6649
Iteration 9990/10000 Loss: 2.3666
Iteration 10000/10000 Loss: 2.3503
```

给出开始位置的字母

In [161…

```python
def generate_name(model, start_letters='a', max_length=20):
    device = next(model.parameters()).device
    hidden = model.init_hidden().to(device)
    # 将多个字母的字符串转换为输入张量
    input_tensor = torch.tensor([char_to_index[ch] for ch in start_letters],

    name = start_letters

    for i in range(max_length):
        output, hidden = model(input_tensor[-1], hidden)  # 只使用最后一个字符(
        topv, topi = output.topk(1)
        next_char_index = topi[0][0].item()
        next_char = index_to_char[next_char_index]

        name += next_char
        input_tensor = torch.cat((input_tensor, torch.tensor([[next_char_inc

        if next_char == ' ':
            break

    return name.strip()
```

In [162…

```
def visualize(model, start_letters='a', max_length=20):
    device = next(model.parameters()).device
    hidden = model.init_hidden().to(device)

    # 将多个字母的字符串转换为输入张量
    input_tensor = torch.tensor([char_to_index[ch] for ch in start_letters],

    name = start_letters
    all_probs = []

    for i in range(max_length):
        output, hidden = model(input_tensor[-1], hidden)  # 只使用最后一个字符(
        topv, topi = output.topk(5)
        top_chars = [index_to_char[topi[0][j].item()] for j in range(5)]
        top_probs = torch.nn.functional.softmax(topv, dim=1).squeeze().tolis

        all_probs.append(top_probs)

        name += top_chars[0]  # 选择概率最大的字符
        input_tensor = torch.cat((input_tensor, torch.tensor([[topi[0][0].it

        if top_chars[0] == ' ':
            break

    # 可视化每个时刻的前5个字符的概率分布
    fig, ax = plt.subplots(figsize=(10, 5))
    all_probs = np.array(all_probs)
    for i in range(5):
        ax.plot(all_probs[:, i], label=f"Top {i+1}: {top_chars[i]}")
    ax.set_title("Top 5 Character Prediction Probabilities")
    ax.set_xlabel("Time Step")
    ax.set_ylabel("Probability")
    ax.legend()
    plt.show()

    return name.strip()
```

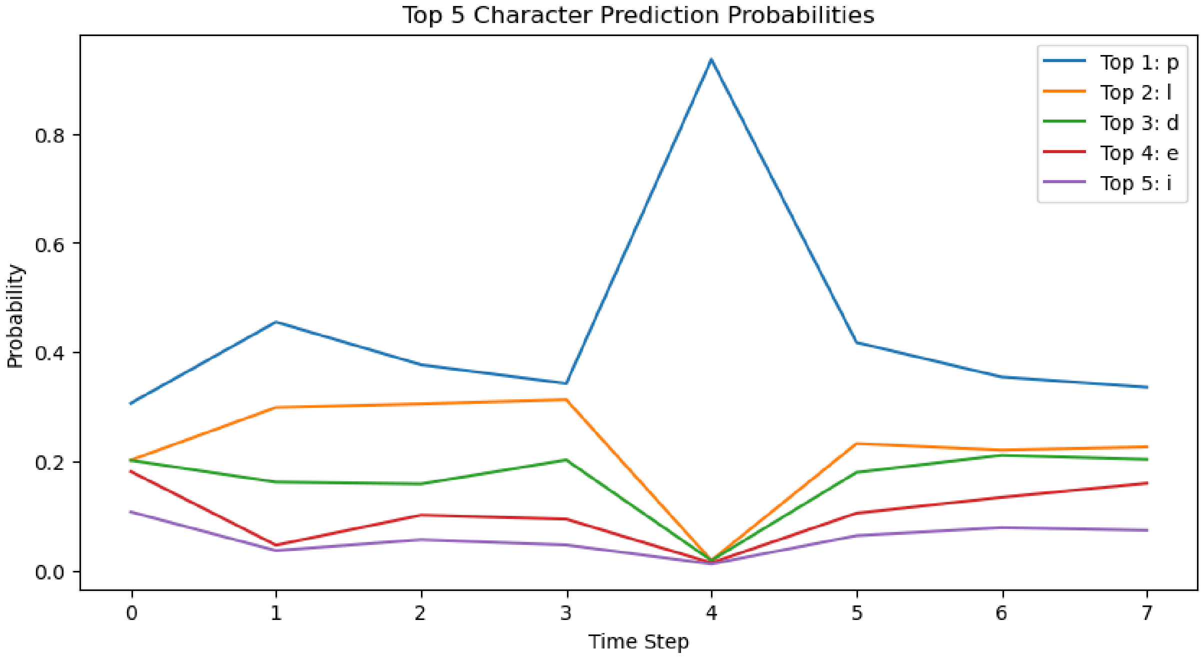
In [163…

```
generated_name = generate_name(model, start_letters ='albre', max_length=8)
print(f"Generated Name: {generated_name}")
```

```
Generated Name: albrettianalp
```

In [164…

```
visualize(model, start_letters='albre', max_length=8)
```



Out[164…

```
'albrettianalp'
```

给出任意位置的字母

In [165…

```
def generate_name(model, partial_name='', max_length=10):
    device = next(model.parameters()).device
    hidden = model.init_hidden().to(device)

    # 将部分名字转换为张量（包括开始字母、结尾字母或中间字母）
    input_tensor = torch.tensor([char_to_index[ch] for ch in partial_name],

    # 生成的名字
    name = partial_name

    # 从已知部分开始生成
    name_len = max_length - len(partial_name)
    for i in range(name_len):
        output, hidden = model(input_tensor[-1], hidden)  # 只使用最后一个字符
        topv, topi = output.topk(1)
        next_char_index = topi[0][0].item()
        next_char = index_to_char[next_char_index]

        name += next_char
        input_tensor = torch.cat((input_tensor, torch.tensor([[next_char_inc

        if next_char == ' ':
            break
    return name.strip()
```

In [166…

```
def visualize(model, partial_name='', max_length=10):
    device = next(model.parameters()).device
    hidden = model.init_hidden().to(device)

    # 将部分名字转换为张量（包括开始字母、结尾字母或中间字母）
    input_tensor = torch.tensor([char_to_index[ch] for ch in partial_name],

    name = partial_name
    all_probs = []

    # 从已知部分开始生成
    name_len = max_length - len(partial_name)
    for i in range(name_len):
        output, hidden = model(input_tensor[-1], hidden)  # 只使用最后一个字符
        topv, topi = output.topk(5)
        top_chars = [index_to_char[topi[0][j].item()] for j in range(5)]
        top_probs = torch.nn.functional.softmax(topv, dim=1).squeeze().tolis

        all_probs.append(top_probs)

        name += top_chars[0]  # 选择概率最大的字符
        input_tensor = torch.cat((input_tensor, torch.tensor([[topi[0][0].it

        if top_chars[0] == ' ':
            break

    # 可视化每个时刻的前5个字符的概率分布
    fig, ax = plt.subplots(figsize=(10, 5))
    all_probs = np.array(all_probs)
    for i in range(5):
        ax.plot(all_probs[:, i], label=f"Top {i+1}: {top_chars[i]}")
    ax.set_title("Top 5 Character Prediction Probabilities")
    ax.set_xlabel("Time Step")
    ax.set_ylabel("Probability")
    ax.legend()
    plt.show()

    return name.strip()
```

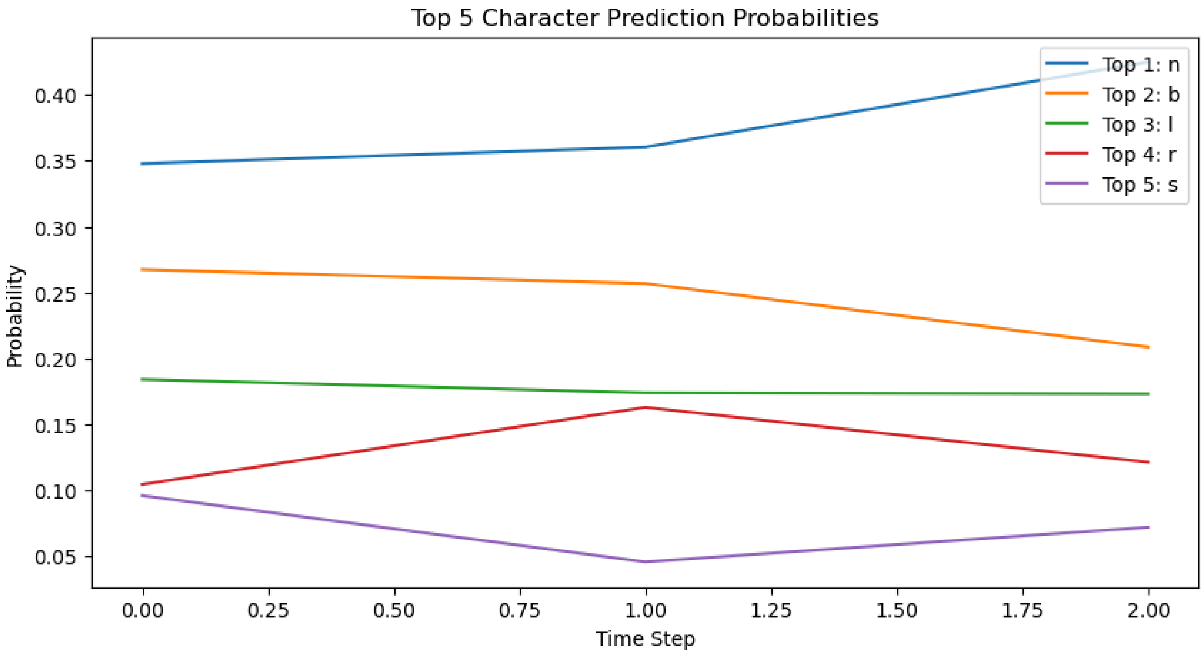
In [167…

```
generated_name = generate_name(model, partial_name='mich', max_length=7)
print(f"Generated Name: {generated_name}")
```

```
Generated Name: michnan
```

In [168…

```
visualize(model, partial_name='mich', max_length=7)
```



```
Out[168…  'michnan'
```