



# 研究生《深度学习》课程 实验报告

实验名称:	<u>实验 4 循环神经网络实验</u>
姓 名:	<u>郑楚彬</u>
学 号:	<u>21140129</u>
上课类型:	<u>专业课</u>
日 期:	<u>2021. 09. 18</u>

# 一、实验内容

手动实现循环神经网络 RNN、torch.nn.rnn 实现循环神经网络  
不同超参数的对比分析、PyTorch 实现 LSTM 和 GRU

# 二、实验设计

# 三、实验环境及实验数据集

## 1. 实验环境

Ubuntu 18.04、CPU 4 核、32GB、Pytorch 1.6.0、Jupyter Notebook

## 2. 数据集：用历史轨迹数据预测下一个签到地点（分类任务）

- 用户签到数据
- FourSquare 是一个地点推荐网站，类似于国内的大众点评。
- 当用户到达某个地点时，可以通过手机 App 进行“签到”(check-in)。
- 将一个用户所有的签到记录按照时间顺序排序，就能得到此用户的行动轨迹
- 每个类别的后 20-30% 当作测试集
- 本实验中使用的数据包含纽约和东京两个城市的用户签到数据，分别存储在 FS\_NYC.csv 和 FS\_TKY.csv 两个文件中。

## 3. EDA：数据集探索性分析

原数据集中共有 801131 条签到数据，2293 个用户，100191 个地点 ID，时间区间为 2012-04-03 18:00:09 ~ 2013-02-16 02:35:36。

	userid	venueId	venueCategoryId	venueCategory	latitude	longitude	timezoneOffset	utcTimestamp
0	470	49bbd6c0f964a520f4531fe3	4bf58dd8d48988d127951735	Arts & Crafts Store	40.719810	-74.002581	-240	Tue Apr 03 18:00:09 +0000 2012
1	979	4a43c0aef964a520c6a61fe3	4bf58dd8d48988d1df941735	Bridge	40.606800	-74.044170	-240	Tue Apr 03 18:00:25 +0000 2012
2	69	4c5cc7b485a1e21e00d35711	4bf58dd8d48988d103941735	Home (private)	40.716162	-73.883070	-240	Tue Apr 03 18:02:24 +0000 2012
3	395	4bc7086715a7ef3bef9878da	4bf58dd8d48988d104941735	Medical Center	40.745164	-73.982519	-240	Tue Apr 03 18:02:41 +0000 2012
4	87	4cf2c5321d18a143951b5cec	4bf58dd8d48988d1cb941735	Food Truck	40.740104	-73.989658	-240	Tue Apr 03 18:03:00 +0000 2012

为方便进行模型的训练，按时间升序排列后取最后 20000 条记录，则数据集缩减为 1867 个用户，9310 个地点 ID，时间区间为 2013-02-06 00:47:34 ~ 2013-02-16 02:35:36。

	userid	venueId	venueCategoryId	venueCategory	latitude	longitude	timezoneOffset	utcTimestamp
558109	2278	4c00a5419cf52d7f422e14e7	4bf58dd8d48988d129951735	Train Station	35.562478	139.716050	540	2013-02-06 00:47:34+00:00
558110	955	4be4f932d4f7c9b61df92420	4bf58dd8d48988d129951735	Train Station	35.681284	139.766071	540	2013-02-06 00:48:51+00:00
222969	3	50f0d0bfe4b07d6a11535185	4bf58dd8d48988d1c4941735	Restaurant	40.746390	-74.000923	-300	2013-02-06 00:49:03+00:00
558111	48	4b2692a1f964a520ba7d24e3	4bf58dd8d48988d129951735	Train Station	35.669627	139.688855	540	2013-02-06 00:49:05+00:00
558112	2088	4f69c4ade4b0ffb80ab56385	4bf58dd8d48988d174941735	Office	35.661705	139.700105	540	2013-02-06 00:49:08+00:00

#### 4. 准备训练集/测试集:

##### □ 预处理

用户 ID、地点 ID、地点类别 ID 均使用标签编码

##### □ 构造序列数据

训练集: 测试集=5: 1

对每个用户的签到数据按时间排序

先划分原始长序列, 后采样短序列, 滑动窗口大小为 8

##### □ 训练集/测试集

```
torch.Size([3287, 8, 6]) torch.Size([3287])
torch.Size([600, 8, 6]) torch.Size([600])
```

## 四、实验过程

### 1. 循环神经网络 RNN

#### 1.1 手动实现

##### □ 模型结构

```
class MyRNN(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        """
        :param input_size: 输入数据的维度, 若每一步的输入均为一个采样值, 则input_size=1
        :param hidden_size: 指定隐藏状态的维度
        :param output_size: 指定输出数据的维度
        """
        super().__init__()
        self.hidden_size = hidden_size

        self.w_i = nn.Parameter(torch.normal(0.0, 0.01, size=(input_size, hidden_size)))
        self.b_i = nn.Parameter(torch.normal(0.0, 0.01, size=(hidden_size,)))

        self.w_h = nn.Parameter(torch.normal(0.0, 0.01, size=(hidden_size, hidden_size)))
        self.b_h = nn.Parameter(torch.normal(0.0, 0.01, size=(hidden_size,)))

        self.w_y = nn.Parameter(torch.normal(0.0, 0.01, size=(hidden_size, output_size)))
        self.b_y = nn.Parameter(torch.normal(0.0, 0.01, size=(output_size,)))

        # 激活函数
        self.tanh = nn.Tanh()
        self.leaky_relu = nn.LeakyReLU()

        # 参数初始化
        for param in self.parameters():
            nn.init.normal_(param, std=0.01)
            # nn.init.xavier_uniform_(param)

    def forward(self, x: torch.Tensor):
        """
        :param x: 输入序列, 包含3个维度: batch, 序列长度, 每条数据的特征
        """
        batch_size = x.shape[0]
        seq_len = x.shape[1]

        # 初始化隐藏状态, 一般设为全0
        h = torch.zeros(batch_size, self.hidden_size)

        # RNN 实际上只能一步步处理序列。Many to one
        for i in range(seq_len):
            h = torch.matmul(x[:, i, :], self.w_i) + self.b_i \
                + torch.matmul(h, self.w_h) + self.b_h
            h = self.tanh(h)

            y = torch.matmul(h, self.w_y) + self.b_y
            y = self.leaky_relu(y)
            # y = self.softmax(y)

        return y
```

## □ 实验结果

分类效果都异常的差，原因在于类别较多，共有 9310 类，可能是我对题目的理解上有偏差吧。



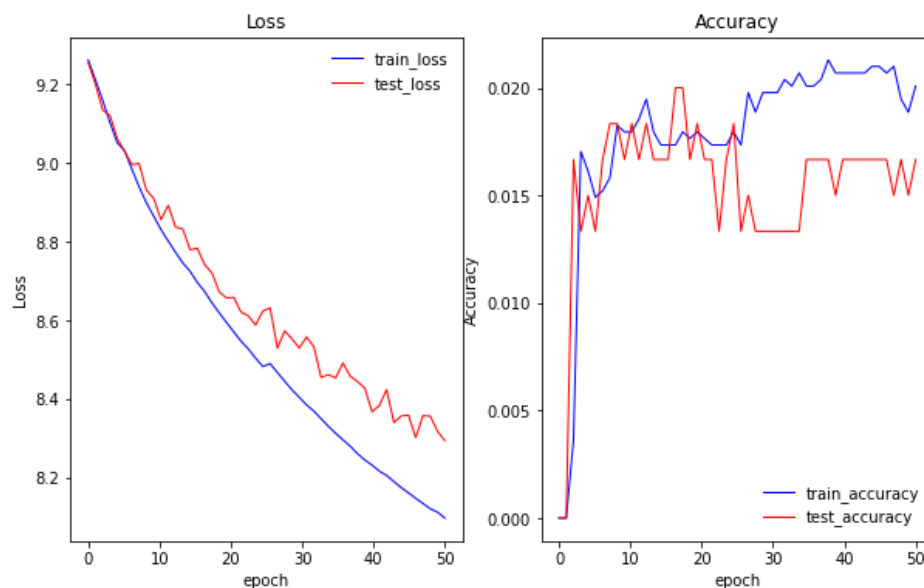
## 1.2 torch.nn 实现

### □ 模型结构

```
TorchRNN(  
    (rnn): RNN(6, 32, batch_first=True)  
    (softmax): Linear(in_features=32, out_features=9310, bias=True)  
)
```

### □ 实验结果

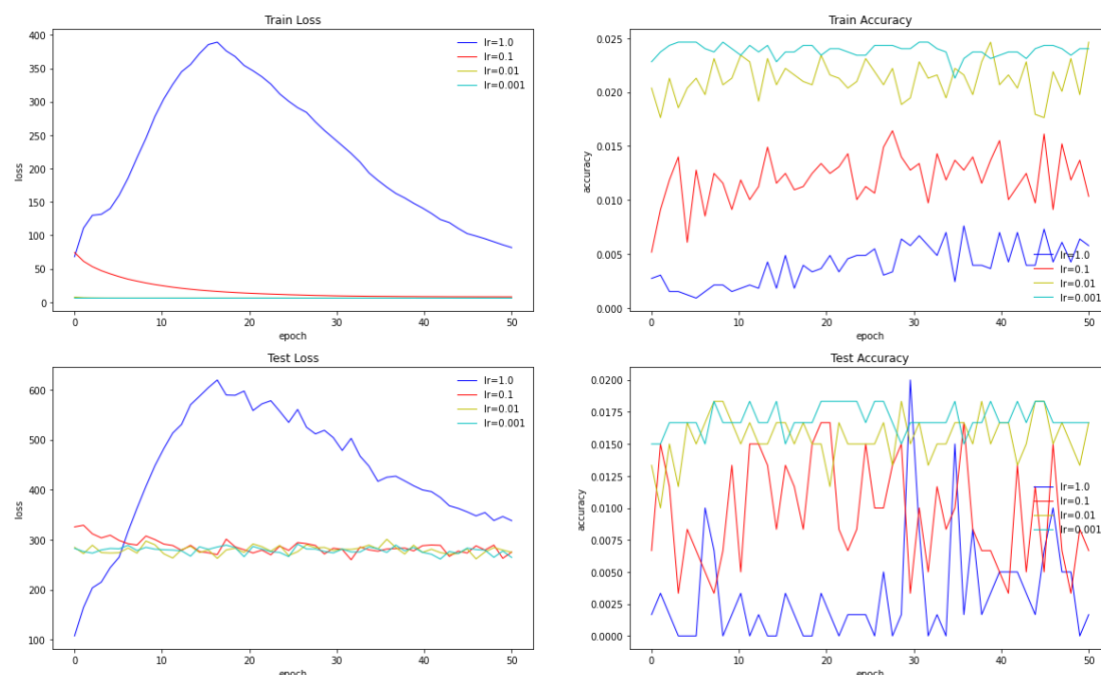
同样地，分类效果很差，但 loss 稳定下降。



## 1.3 不同超参数的对比分析

在这里，主要对学习率(LR)该参数进行了对照实验，在相同条件下，分别设置了 LR=1.0、0.1、0.01、0.001。其中，蓝色线条代表 lr=1.0、红色线条代表 lr=0.1、黄色线条代表 lr=0.01、青色线条代表 lr=0.001。

显而易见，无论是测试集还是训练集、无论是 loss 还是 accuracy，学习率越高，曲线越不稳定；而当 LR=0.01 或 0.001 时，两者的曲线都较为平稳。



## 2. PyTorch 实现 LSTM 和 GRU

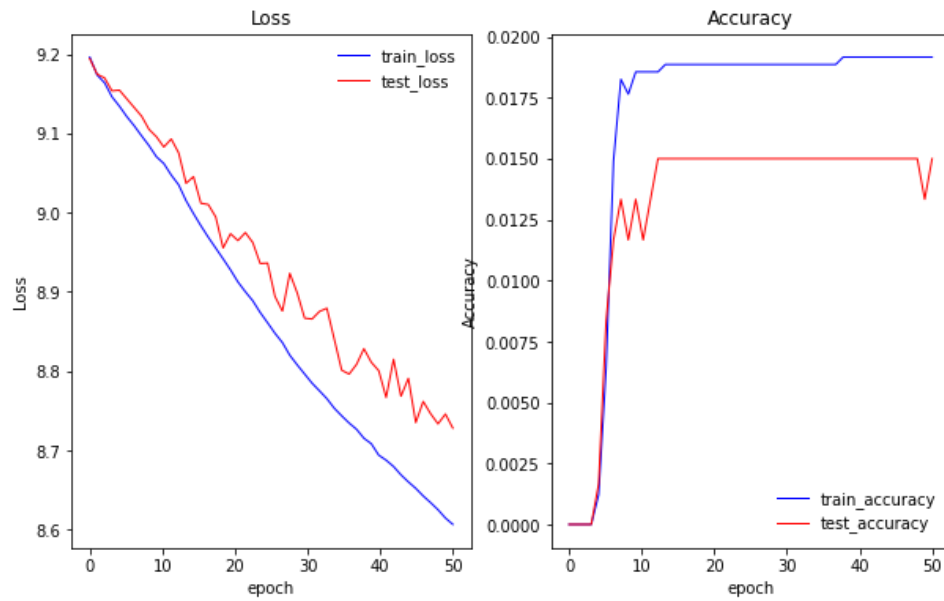
### 2.1 实现 LSTM

#### □ 模型结构

```
TorchLSTM(  
    (lstm): LSTM(6, 32, batch_first=True)  
    (softmax): Linear(in_features=32, out_features=9310, bias=True)  
)
```

#### □ 实验结果

```
epoch 1, train_loss 9.196412, test_loss 9.194424, train_acc 0.000000, test_acc 0.000000  
epoch 2, train_loss 9.173744, test_loss 9.174764, train_acc 0.000000, test_acc 0.000000  
epoch 3, train_loss 9.163891, test_loss 9.170076, train_acc 0.000000, test_acc 0.000000  
epoch 4, train_loss 9.146211, test_loss 9.153725, train_acc 0.000000, test_acc 0.000000  
epoch 5, train_loss 9.134428, test_loss 9.154624, train_acc 0.001217, test_acc 0.001667  
epoch 6, train_loss 9.121261, test_loss 9.143652, train_acc 0.006389, test_acc 0.008333  
epoch 7, train_loss 9.109662, test_loss 9.132694, train_acc 0.014907, test_acc 0.011667  
epoch 8, train_loss 9.096899, test_loss 9.121753, train_acc 0.018254, test_acc 0.013333  
epoch 9, train_loss 9.084585, test_loss 9.105120, train_acc 0.017645, test_acc 0.011667  
epoch 10, train_loss 9.070564, test_loss 9.095808, train_acc 0.018558, test_acc 0.013333
```

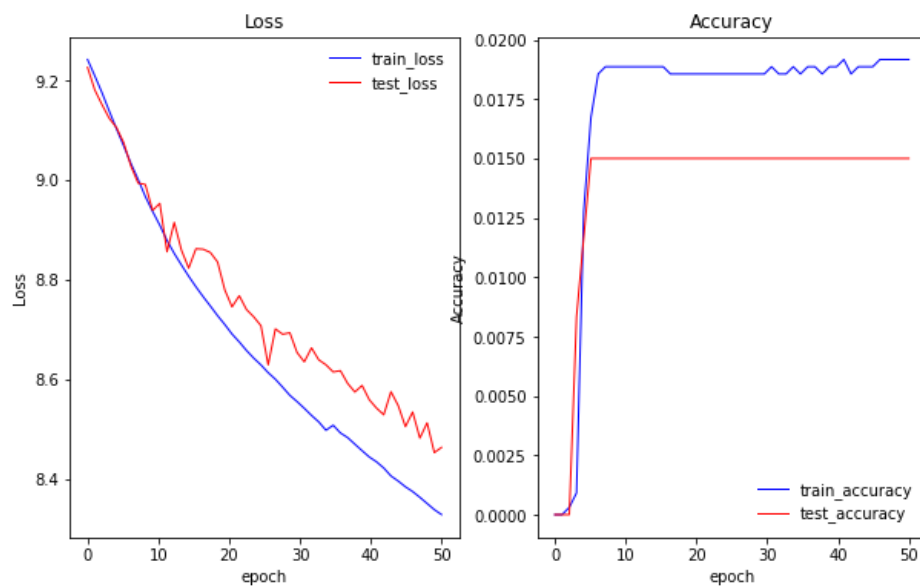


## 2.2 实现 GRU

### □ 模型结构

```
TorchGRU(
  (lstm): GRU(6, 32, batch_first=True)
  (softmax): Linear(in_features=32, out_features=9310, bias=True)
)
```

### □ 实验结果



## 五、实验结果

## 六、实验心得体会

本次实验不仅了解到了 RNN 在时间序列上的应用，还进一步探究了以此为基础优化演进的几个模型，如 LSTM、GRU。整体而言，获益匪浅。

## 七、参考文献

## 八、附录

需要补充说明的内容，如无可略。

# 实验报告编写要求

1. 正文要求小四号宋体，行间距 1.5 倍；
2. 英文要求小四号 Times New Roman；
3. 在实验内容、实验过程、实验结果三部分需要针对当次实验不同的实验内容分别填写（模版以实验一为例），实验设计中如有必要也可以分开填写；
4. 实验报告配图的每幅图应有编号和标题，编号和标题应位于图下方处，居中，中文用五号宋体；
5. 表格应为三线表，每个表格应有编号和标题，编号和标题应写在表格上方正中，距正文段前 0.5 倍行距。表格中量与单位之间用“/”分隔，编号与标题中的中文用五号宋体；
6. 图、表、公式、算式等，一律用阿拉伯数字分别依序连续编排序号。其标注形式应便于互相区别，可分别为：图 1、表 2、公式(5)等。