清华大学电子工程系

媒体与认知 课堂 2

2021-2022 学年春季学期

作业 5

2022 年 4 月 27 日

本次作业内容包含序列建模任务中常用的隐含马尔可夫模型和循环神经网络。具体任务分为理论部分、编程部分以及作业报告。其中理论部分包含第 1，2 题，所有同学均需完成，答案附在作业报告中；编程部分包含第 3、4 题，采用循环神经网络完成场景文字识别任务。第 5 题为撰写作业报告。已确认自选课题的同学需完成第 6 题。

1. 单选题（15 分）
2. 计算题（15 分）
3. 完成基于循环神经网络的场景文字识别相关程序代码（30 分）
4. 训练/预测/可视化（30 分）
5. 撰写作业报告（10 分）
6. 汇报自选课题进度（70 分）*

理论部分

1 单选题（15 分）

1.1 **给定 HMM 的模型参数 λ，隐含状态总数为 N。设给定观测序列 O 的条件下，在第 t 时刻处于状态 S_i 的概率为 $\gamma_t(i) = P(q_t = S_i|O, \lambda)$。若已经计算得到所有前向变量 $\alpha_t(i)$ 和后向变量 $\beta_t(i)$，则下列计算 $\gamma_t(i)$ 的方法中正确的是哪项？**

(A) $\gamma_t(i) = \alpha_t(i)\beta_t(i)$

(B) $\gamma_t(i) = \alpha_t(i)\beta_t(i) \ / \ \sum_{j=1}^{N} \alpha_t(j)\beta_t(j)$

(C) $\gamma_t(i) = \alpha_t(i) + \beta_t(i)$

(D) $\gamma_t(i) = (\alpha_t(i) + \beta_t(i)) \ / \ \sum_{j=1}^{N}(\alpha_t(j) + \beta_t(j))$

1.2 对于一个参数为 $\lambda = \{\pi, A, B\}$ 的 HMM，隐含状态总数为 5，且观测序列 O 长度为 10。若已经计算得到所有前向变量 $\alpha_t(i)$ 和后向变量 $\beta_t(i)$，则下列关于 $P(q_5 = S_2, q_6 = S_4, O|\lambda)$ 的计算方式中正确的是哪项？

(A) $\alpha_5(2)a_{24}$

(B) $b_4(O_6)\beta_6(4)$

(C) $\alpha_5(2)a_{24}b_4(O_6)$

(D) $\alpha_5(2)a_{24}b_4(O_6)\beta_6(4)$

1.3 考虑下图所示的 RNN，其运算过程为

$$\boldsymbol{h}_t = \phi\left(W\boldsymbol{x}_t + U\boldsymbol{h}_{t-1}\right)$$
$$\boldsymbol{y}_t = \mathrm{softmax}\left(V\boldsymbol{h}_t\right)$$

输入特征序列 $X = \{\boldsymbol{x}_t\}_{t=1}^{3}$ 包含三个时刻的数据。每个时刻的输入特征向量 $\boldsymbol{x}_t \in \mathbb{R}^2$，隐含状态 $\boldsymbol{h}_t \in \mathbb{R}^4$，输出特征 $\boldsymbol{y}_t \in \mathbb{R}^3$。则该 RNN 的网络参数量是多少？ 2*4+4*4+3*4=36

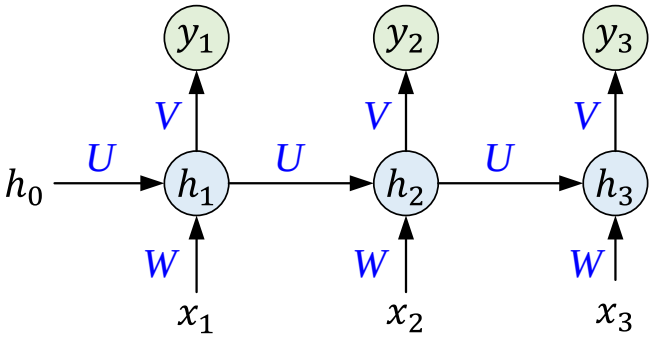

图 1: 沿时间展开后的 RNN 示意图

(A) 12

(B) 36

(C) 72

(D) 108

1.4 请查看 PyTorch 中的 LSTM 网络说明文档（参见下方注释）。现利用 nn.LSTM 定义一个 LSTM 网络，其参数为：`input_size=8, hidden_size=16, num_layers=2, bidirectional=True`，则下列说法中哪一项为正确的？

注：nn.LSTM 的说明文档地址为https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html?highlight=lstm#torch.nn.LSTM

(A) 函数返回值 (outputs) 包含最后一层 LSTM 各时刻的隐含状态 (hidden state)，以及两层 LSTM 最后一个时刻的隐含状态 (hidden state)

(B) 函数返回值 (outputs) 包含最后一层 LSTM 各时刻的隐含状态 (hidden state)，以及两层 LSTM 最后一个时刻的和单元状态 (cell state)

(C) 函数返回值 (outputs) 为两层 LSTM 最后一个时刻的隐含状态 (hidden state) 和单元状态 (cell state)

(D) 函数返回值 (outputs) 包含最后一层 LSTM 各时刻的隐含状态 (hidden state)，以及两层 LSTM 最后一个时刻的隐含状态 (hidden state) 和单元状态 (cell state)

1.5 在 CTC 算法中，对于以下的对齐方式，哪项对应的解码字符序列为 “hello” ？（“-” 表示 “空白” 符号）

(A) hello

(B) hh-ee-ll-oo

(C) h-elllllllll-loo

(D) h-h-e-e-l-l-o-o

2 计算题（15 分）

2.1 隐含马尔可夫模型的解码

某手机专卖店今年元旦新开业，每月上旬进货时，由专卖店经理决策，采用三种进货方案中的一种：高档手机 (H)，中档手机 (M)，低档手机 (L)。

Parameters

- **input_size** – The number of expected features in the input *x*
- **hidden_size** – The number of features in the hidden state *h*
- **num_layers** – Number of recurrent layers. E.g., setting `num_layers=2` would mean stacking two LSTMs together to form a *stacked LSTM*, with the second LSTM taking in outputs of the first LSTM and computing the final results. Default: 1
- **bias** – If `False`, then the layer does not use bias weights *b_ih* and *b_hh*. Default: `True`
- **batch_first** – If `True`, then the input and output tensors are provided as (*batch, seq, feature*) instead of (*seq, batch, feature*). Note that this does not apply to hidden or cell states. See the Inputs/Outputs sections below for details. Default: `False`
- **dropout** – If non-zero, introduces a *Dropout* layer on the outputs of each LSTM layer except the last layer, with dropout probability equal to `dropout`. Default: 0
- **bidirectional** – If `True`, becomes a bidirectional LSTM. Default: `False`
- **proj_size** – If `> 0`, will use LSTM with projections of corresponding size. Default: 0

Inputs: input, (h_0, c_0)

- **input**: tensor of shape (L, H_{in}) for unbatched input, (L, N, H_{in}) when `batch_first=False` or (N, L, H_{in}) when `batch_first=True` containing the features of the input sequence. The input can also be a packed variable length sequence. See `torch.nn.utils.rnn.pack_padded_sequence()` or `torch.nn.utils.rnn.pack_sequence()` for details.
- **h_0**: tensor of shape $(D * \text{num\_layers}, H_{out})$ for unbatched input or $(D * \text{num\_layers}, N, H_{out})$ containing the initial hidden state for each element in the input sequence. Defaults to zeros if (h_0, c_0) is not provided.
- **c_0**: tensor of shape $(D * \text{num\_layers}, H_{cell})$ for unbatched input or $(D * \text{num\_layers}, N, H_{cell})$ containing the initial cell state for each element in the input sequence. Defaults to zeros if (h_0, c_0) is not provided.

where:

$$
\begin{aligned}
N &= \text{batch size} \\
L &= \text{sequence length} \\
D &= 2 \text{ if bidirectional=True otherwise } 1 \\
H_{in} &= \text{input\_size} \\
H_{cell} &= \text{hidden\_size} \\
H_{out} &= \text{proj\_size if proj\_size} > 0 \text{ otherwise hidden\_size}
\end{aligned}
$$

Outputs: output, (h_n, c_n)

- **output**: tensor of shape $(L, D * H_{out})$ for unbatched input, $(L, N, D * H_{out})$ when `batch_first=False` or $(N, L, D * H_{out})$ when `batch_first=True` containing the output features (*h_t*) from the last layer of the LSTM, for each *t*. If a `torch.nn.utils.rnn.PackedSequence` has been given as the input, the output will also be a packed sequence.
- **h_n**: tensor of shape $(D * \text{num\_layers}, H_{out})$ for unbatched input or $(D * \text{num\_layers}, N, H_{out})$ containing the final hidden state for each element in the sequence.
- **c_n**: tensor of shape $(D * \text{num\_layers}, H_{cell})$ for unbatched input or $(D * \text{num\_layers}, N, H_{cell})$ containing the final cell state for each element in the sequence.

当月市场行情假设分为畅销 (S_1) 和滞销 (S_2) 两种。畅销时，三种进货方案的概率分别为 0.4, 0.4, 0.2；滞销时，三种进货方案的概率分别为 0.2, 0.3, 0.5。

某月份市场行情为畅销，下一个月份为畅销和滞销的概率分别为 0.6 和 0.4；某月份市场行情为滞销，下一个月份为畅销和滞销的概率分别为 0.5 和 0.5。

开业第一个月市场行情为畅销和滞销的可能性均为 0.5。

(1) 如果我们采用隐含马尔可夫模型 (HMM) 对该专卖店进货环节建模，请写出 HMM 对应的参数 $\lambda = \{\pi, A, B\}$。

(2) 在第一季度中，采购业务员执行的进货方案为“高档手机，中档手机，低档手机”，即观测序列为 H, M, L。请利用 Viterbi 算法推测前三个月的市场行情。

2.2 循环神经网络的长时相关性建模能力

对序列中的长距离相关信息进行建模是涉及序列的任务中十分重要的一点，例如在阅读理解任务里，题目和正文中的关键词可能相距很远，这就需要模型具备足够好的长距离相关信息建模能力。传统 RNN 在训练时存在梯度消失问题，较远的误差无法得到有效传递，因此学习长距离相关信息时面临较大挑战，在本题中我们对传统 RNN 难以学习长距离相关信息的问题进行一个简单的讨论。

对 RNN 的计算过程进行简化，考虑一个暂不采用激活函数以及输入 $\boldsymbol{x}$ 的 RNN:

$$\boldsymbol{h}_t = U\boldsymbol{h}_{t-1} = U\left(U\boldsymbol{h}_{t-2}\right) = \ldots = U^t\boldsymbol{h}_0$$

其中 U^t 为 t 个 U 矩阵连乘。若矩阵 U 存在如下特征值分解：

$$U = Q\Lambda Q^\top$$

其中 Q 为单位正交矩阵（每一列为模长为 1 的特征向量），$Q^\top$ 为 Q 的转置，Λ 为特征值对角矩阵，则上述的 RNN 计算过程可表示为：

$$\boldsymbol{h}_t = Q\Lambda^t Q^\top \boldsymbol{h}_0$$

本题目包含以下三个问题：

(1) 假设某一特征值 $\lambda_i < 1$，当时刻 t 增大时，Λ^t 中第 i 行 i 列的值会怎样变化？

(2) 假设 $\boldsymbol{h}_0 = \boldsymbol{q}_i$，其中 $\boldsymbol{q}_i$ 为 U 矩阵的第 i 个特征向量（即 Q 的第 i 列），设 $\mathcal{L}$ 为目标函数计算出的 loss。试验证：

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{h}_0} = \lambda_i^t \frac{\partial \mathcal{L}}{\partial \boldsymbol{h}_t}$$

提示：

$$\boldsymbol{h}_t = (\boldsymbol{q}_1 \ \cdots \ \boldsymbol{q}_n) \begin{pmatrix} \lambda_1^t & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^t \end{pmatrix} \begin{pmatrix} \boldsymbol{q}_1^\top \\ \vdots \\ \boldsymbol{q}_n^\top \end{pmatrix} \boldsymbol{q}_i$$

(3) 对于更一般的 $\boldsymbol{h}_0$，由于 Q 中的特征向量构成一组完备正交基，可以将 $\boldsymbol{h}_0$ 分解为 Q 中不同特征向量的线性组合，即 $\boldsymbol{h}_0 = \sum_{i=1}^n k_i \boldsymbol{q}_i$。通过上述分析，请尝试解释传统 RNN 训练中的梯度消失现象，由此理解传统 RNN 对长距离相关信息建模的困难。

提示：试讨论误差反向传播的过程中 $\frac{\partial \mathcal{L}}{\partial \boldsymbol{h}_t}$（loss 对 $\boldsymbol{h}_t$ 的导数）和 loss 对 $\boldsymbol{h}_0$ 各个线性组合分量的导数之间的关系。

编程部分

3 完成基于循环神经网络的场景文本识别程序代码（30 分）

我们使用 “CNN-RNN-CTC” 的网络结构对场景文本图像进行识别。模型整体结构如图 2所示。与前几次作业中对单个字符图像的分类任务不同，本次作业我们对包含多个字符的单词图像进行序列建模。我们将从头开始完整地搭建网络和实现大部分模型训练和验证过程。

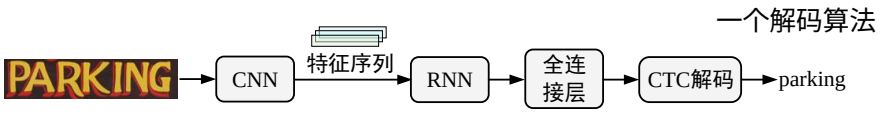

图 2: 基于 “CNN-RNN-CTC” 模型的场景文本识别过程

直观上，我们在理解文本图像时通常按照文字的阅读方向（例如从左往右）对图像进行扫描。在使用循环神经网络进行文本图像识别时也可以

遵循这个过程实现序列建模。首先使用 CNN 对图像的视觉特征进行提取，再使用 RNN 对特征中的上下文信息进行建模。由于图像中字符的数量是变化的，我们可以使用 CTC 算法寻找图像中的字符对齐方式并进行解码。需要注意的是，RNN 的输入为一个特征序列而并非 CNN 提取的原始特征图，因此我们需要在特征提取的过程中把二维的特征图转换为一维的特征序列，相当于特征图的高度为 1。

程序清单如下：

文件或目录	说明	注意事项
hw5.zip	作业 5 程序压缩包	解压可以得到下列文件
\data	存放本次作业所用数据集	请勿修改
\train	训练集（10,000 张）	请勿修改
\validation	验证集（500 张）	请勿修改
\my_own	自行搜集的文本图像	可以添加实际图像样本
\models	存放训练好的模型	请勿修改
utils.py	数据读取、转换和可视化	请阅读各函数功能和调用方式
network.py	网络结构定义	**需要完成代码**
main.py	训练、验证及预测主程序	**需要完成代码**

每处需要完成的地方都有代码提示和步骤提示，需要完成的代码清单如下：

序号	文件和行号	内容	说明
TODO 1	network.py line 48	完成 “CRNN” 模型的初始化	包含 CNN、RNN 和线性层分类器
TODO 2	network.py line 77	完成模型的前向计算过程	需返回未归一化分类概率和特征序列长度
TODO 3	main.py line 138	完成模型每轮训练过程代码	请先阅读 utils.py 各函数接口和 nn.CTCLoss 文档
TODO 4	main.py line 165	完成模型的验证过程代码	请先阅读 utils.py 各函数接口

注：nn.CTCLoss 的说明文档地址为
https://pytorch.org/docs/stable/generated/torch.nn.CTCLoss.html?highlight=ctcloss#torch.nn.CTCLoss

对于模型结构，作业中提供了一组推荐的网络参数。其中 CNN 包含 5

个卷积层，其参数设置如下:

参数	卷积层 1	卷积层 2	卷积层 3	卷积层 4	卷积层 5
输出通道数	16	32	48	64	64
卷积核尺寸	3×3	3×3	3×3	3×3	1×1
stride	2×2	1×1	1×1	1×1	1×1
padding	1×1	1×1	1×1	1×1	0
批量归一化	是	是	是	是	否
激活函数	ReLU	ReLU	ReLU	ReLU	无
最大池化	2×2	2×1	2×1	2×1	无

RNN 则使用一层双向的 LSTM 网络，每个方向的隐含层节点数设置为 32。在完成 network.py 的程序后，可以运行下列命令：

```
python network.py
```

若显示 “The output size of model is correct!”，则表明网络输出变量的尺寸是正确的。

4 训练/预测/可视化（30 分）

本次作业的训练集为从 ICDAR 2019 MLT 场景文本识别数据集 [1] 中选择出的 10,000 张场景文本图像，验证集为从 ICDAR 2013 场景文本识别数据集 [2] 中选择出的 500 张场景文本图像。任务的目标为将图像中的文本识别出来，不需要考虑英文大小写和标点符号，模型的字符集 C 包含 26 个英文字母、10 个数字、1 个用于表示其余所有未知字符的 $\langle unk \rangle$ 符号，以及 CTC 算法中引入的 “空白” 符号（用 “–” 表示），即

$$C = \{-, \langle unk \rangle, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

（1）模型的训练和验证

在完成代码后，运行如下命令进行模型的训练和验证：

[1] https://rrc.cvc.uab.es/?ch=15

[2] https://rrc.cvc.uab.es/?ch=2

```
python main.py --mode train
```

本次作业利用默认参数运行程序即可，各参数的说明详见附录部分，或 main.py 文件 218 行设置 “parser” 部分。

完成训练后，程序主目录下会生成 “loss_and_accuracy.jpg” 的图像文件，显示每轮模型在训练集上的 loss 和验证集上的单词识别正确率（即完全识别正确的图像样本所占总样本的比例）变化情况。

请将训练集 loss 和验证集单词识别正确率的可视化结果，以及验证集上的最终正确率写入作业报告中，并进行简要分析。

温馨提示：

- 由于作业侧重于理解原理，数据量较小，在使用默认网络参数的条件下，训练 40 轮后在验证集上的单词识别正确率约为 50%。
- 由于任务相对比较复杂，模型训练时间可能较长，在笔记本电脑上如果从头训练模型 40 轮，可能需要 1 小时左右。
- 训练初期模型的 loss 下降缓慢，同时验证集上的正确率为 0 是正常现象。在使用默认网络参数的条件下，大约训练 10 轮之后，验证集上的正确率才会开始上升。
- 如果训练过慢，作业中提供了已经预训练 30 轮的模型，位于 “models/pretrain.pth”。如需加载预训练模型，需要完全按照上述默认网络参数定义模型，且 CNN 的网络层使用 “nn.Sequential” 进行构建。加载预训练模型并继续训练 10 轮的命令为：

  ```
  python main.py --mode train --load_pretrain --epoch 10
  ```

（2）使用训练好的模型预测新的文本图像

训练好的模型将默认保存在 “models” 子目录中，使用训练好的模型预测新的文本图像的命令为：

```
python main.py --mode predict --im_path data/my_own/a.png
```

默认的 “im_path” 参数为 “data/my_own/a.png”，也可以自行选取其它的文本图像。同时，如果在训练时调整了训练轮数（epoch 参数）或模型保存频率（model_save_epoch 参数），则也可以通过设置 model_path 参数使用不同的模型文件。

在预测过程中，我们对模型在每个时刻的分类概率进行可视化，以更好地理解 CTC 算法原理。可视化结果图片文件保存在 “data/my_own” 目录中。

请将输入图像、识别结果以及可视化结果写入作业报告中，并对识别结果或可视化结果进行简要分析。

5 撰写作业报告（10 分）

将 hw5 目录和作业报告打包为一个文件（例如 *.zip）提交到网络学堂，图像数据（“data” 目录）不必打包在内。作业报告中包括选择题答案，计算题的解题步骤及答案、任务 3、4 运行结果及分析，本次作业遇到的问题及解决方法，对本次作业的意见及建议。推荐同学们使用随作业发布的 LaTex 模板 HW5-template.zip 完成作业报告。

6 自选课题进度汇报（70 分）*

请已确认自选课题的同学，完成简短的自选课题工作进度汇报，例如，文献阅读、或者研究方案设计、或者原型系统搭建及实验结果等内容。

关于作业迟交的说明：由于平时作业计入总评成绩，希望同学们能按时提交作业。若有特殊原因不能按时提交，请在提交截止时间之前给本次作业责任助教发 Email 说明情况并给出预计提交作业的时间。对于未能提前说明原因的迟交作业，将酌情扣分。

本次作业责任助教为闫睿劼 (Email: yrj17@mails.tsinghua.edu.cn)。

附录

程序利用 argparse 库进行参数设置，可以查看 main.py 中可以调节的参数。不同参数说明如下表所示。

参数	说明
mode	程序运行模式，train 或 predict，默认为 train
batchsize	训练和验证时的批处理大小，默认为 32
device	程序运行设备，cpu 或 cuda，默认为 cpu
norm_height	图像归一化高度，默认为 32
norm_width	图像归一化宽度，默认为 128
epoch	训练轮数，默认为 40
lr	学习率，默认为 1e-3（本次作业优化器默认采用 Adam）
model_save_epoch	模型保存的周期，默认 10 轮保存一次
load_pretrain	是否加载预训练模型，使用该参数表示加载
pretrain_path	预训练模型的路径
model_path	predict 模式下加载模型的路径
im_path	predict 模式下待预测图像的路径