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

郭中贺 2022 年 5 月 15 日

理论部分

1 单选题（15 分）

1.1 B

1.2 D

1.3 B

1.4 D

1.5 C

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)。

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。

2.1 (1). $\pi = \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix}$ $\pi_1 = 0.5$ $\pi_2 = 0.5$ $a_{ij} = p(q_{t+1} = S_j \mid q_t = S_i)$

$a_{11} = 0.6$ $a_{12} = 0.4$ $a_{21} = a_{22} = 0.5$ $\therefore A = \begin{bmatrix} 0.6 & 0.4 \\ 0.5 & 0.5 \end{bmatrix}$

$b_{i1} = p(O_t = H \mid q_t = S_i)$ $b_{i2} = p(O_t = M \mid q_t = S_i)$ $b_{i3} = p(O_t = L \mid q_t = S_i)$

$\therefore B = \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$

2). $O_1 = H$, $O_2 = M$, $O_3 = L$

初始化: $\delta_1(1) = \pi_1 b_1(O_1) = 0.5 \times 0.4 = 0.2$ $\varphi_1(1) = 0$

$\delta_1(2) = \pi_2 b_2(O_1) = 0.5 \times 0.2 = 0.1$ $\varphi_1(2) = 0$

$\delta_2(1) = \max[\delta_1(1) a_{11}, \delta_1(2) a_{21}] \cdot b_1(O_2) = 0.048$

$\delta_2(2) = \max[\delta_1(1) a_{12}, \delta_1(2) a_{22}]\, b_2(O_2) = 0.024$

$\varphi_2(1) = 1$ $\varphi_2(2) = 1$

$\delta_3(1) = \max[\delta_2(1) a_{11}, \delta_2(2) a_{21}] \cdot b_1(O_3) = 0.00576$

$\delta_3(2) = \max[\delta_2(1) a_{12}, \delta_2(2) a_{22}] \cdot b_2(O_3) = 0.0096$

$\varphi_3(1) = 1$ $\varphi_3(2) = 1$

$\therefore q_3^* = 2$, $q_2^* = \varphi_3(1) = 1$ $q_1^* = \varphi_2(1) = 1$

∴ 前3个月市场行情最可能为畅销、畅销、滞销.

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 x 的 RNN:

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

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：

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

(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。

(1). ∵ Λ 为特征值对角阵

设 $\Lambda = \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \lambda_i & \\ & & & & \ddots \\ & & & & & \lambda_n \end{bmatrix}$ 则 $\Lambda^t = \begin{bmatrix} \lambda_1^t & & & & \\ & \lambda_2^t & & & \\ & & \ddots & & \\ & & & \lambda_i^t & \\ & & & & \ddots \\ & & & & & \lambda_n^t \end{bmatrix}$

若 t 增大，$\lambda_i < 1$ ∴ λ_i^t 减小，

即 Λ^t 中第 i 行第 i 列值减小

(2) ∵ $\vec{h}_0 = \vec{q}_i$, $\vec{h}_t = Q\Lambda^t Q^T \vec{h}_0 = (q_1, q_2, \cdots q_n) \begin{pmatrix} \lambda_1^t & & \\ & \ddots & \\ & & \lambda_n^t \end{pmatrix} \begin{pmatrix} q_1^T \\ q_2^T \\ \vdots \\ q_n^T \end{pmatrix} q_i$

∴ $\vec{h}_t = (q_1, q_2, \cdots q_n) \begin{pmatrix} \lambda_1^t & & \\ & \ddots & \\ & & \lambda_n^t \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$ ← 第 i 行

$= (q_1, q_2, \cdots q_n) \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & \lambda_i^t & & \\ & & & \ddots & \\ & & & & 0 \end{pmatrix}$ ← 第 i 行第 i 列为 λ_i^t，其他位置为 0

$= \lambda_i^t \vec{q}_i$

∴ $\frac{\partial L}{\partial \vec{h}_0} = \frac{\partial L}{\partial \vec{q}_i}$ $\frac{\partial L}{\partial \vec{h}_t} = \frac{\partial L}{\partial (\lambda_i^t \vec{q}_i)} = \frac{1}{\lambda_i^t} \frac{\partial L}{\partial \vec{q}_i}$ 因此 $\frac{\partial L}{\partial \vec{h}_0} = \lambda_i^t \frac{\partial L}{\partial \vec{h}_t}$

(3). $\vec{h}_0 = \sum_{i=1}^{n} k_i \vec{q}_i$ 则 $\vec{h}_t = \sum_{i=1}^{n} k_i \lambda_i^t \vec{q}_i$

$\frac{\partial L}{\partial \vec{h}_t} = \sum_{i=1}^{n} \frac{1}{k_i \lambda_i^t} \frac{\partial L}{\partial \vec{q}_i}$ $\frac{\partial L}{\partial \vec{h}_0} = \sum_{i=1}^{n} \frac{1}{k_i} \frac{\partial L}{\partial \vec{q}_i} = \sum_{i=1}^{n} \vec{\alpha}_i$ $\frac{\partial L}{\partial \vec{h}_t} = \sum_{i=1}^{n} \frac{1}{\lambda_i^t} \alpha_i$

∴ $\frac{\partial L}{\partial \vec{h}_t}$ 是 $\frac{\partial L}{\partial \vec{h}_0}$ 各个线性组合分量乘不同权重系数 $\frac{1}{\lambda_i^t}$ 后的重新线性组合

若每个特征值 λ_i 均大于 1，则随着 t 增大，$\frac{1}{\lambda_i^t} \to 0$ 则 $\frac{\partial L}{\partial \vec{h}_t} \to 0$

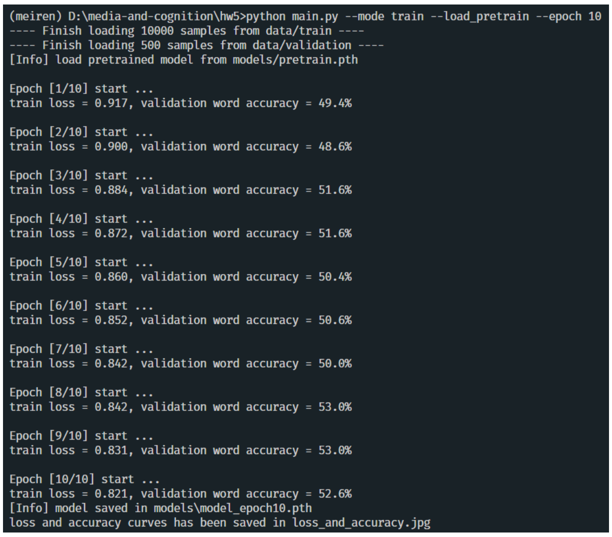
即发生梯度消失.

3 编程作业报告

3.1 训练模型

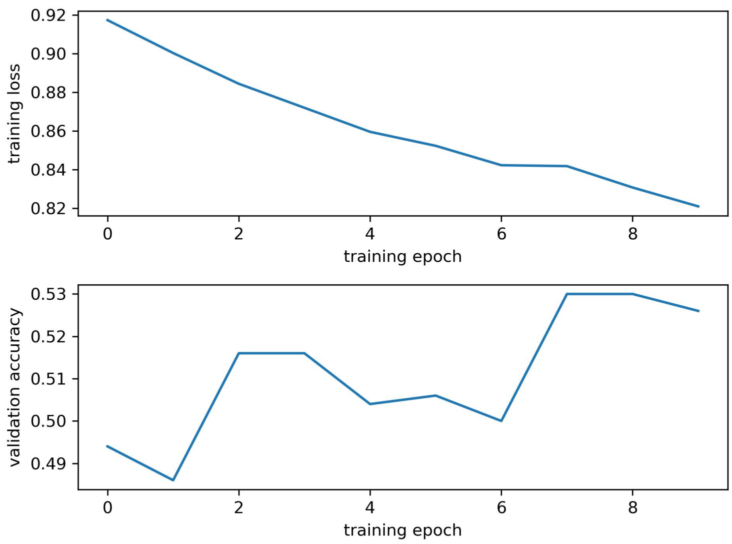
由于训练速度较慢，我选择使用已经训练完成 30 轮的预训练模型继续训练 10 轮作为训练结果，使用命令 python main.py –mode train

–load_pretrain –epoch 10 来进行训练，控制台输出结果如下图所示：



```
(meiren) D:\media-and-cognition\hw5>python main.py --mode train --load_pretrain --epoch 10
---- Finish loading 10000 samples from data/train ----
---- Finish loading 500 samples from data/validation ----
[Info] load pretrained model from models/pretrain.pth

Epoch [1/10] start ...
train loss = 0.917, validation word accuracy = 49.4%

Epoch [2/10] start ...
train loss = 0.900, validation word accuracy = 48.6%

Epoch [3/10] start ...
train loss = 0.884, validation word accuracy = 51.6%

Epoch [4/10] start ...
train loss = 0.872, validation word accuracy = 51.6%

Epoch [5/10] start ...
train loss = 0.860, validation word accuracy = 50.4%

Epoch [6/10] start ...
train loss = 0.852, validation word accuracy = 50.6%

Epoch [7/10] start ...
train loss = 0.842, validation word accuracy = 50.0%

Epoch [8/10] start ...
train loss = 0.842, validation word accuracy = 53.0%

Epoch [9/10] start ...
train loss = 0.831, validation word accuracy = 53.0%

Epoch [10/10] start ...
train loss = 0.821, validation word accuracy = 52.6%
[Info] model saved in models\model_epoch10.pth
loss and accuracy curves has been saved in loss_and_accuracy.jpg
```

可视化结果如下图所示：

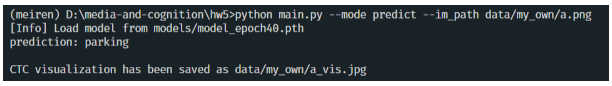


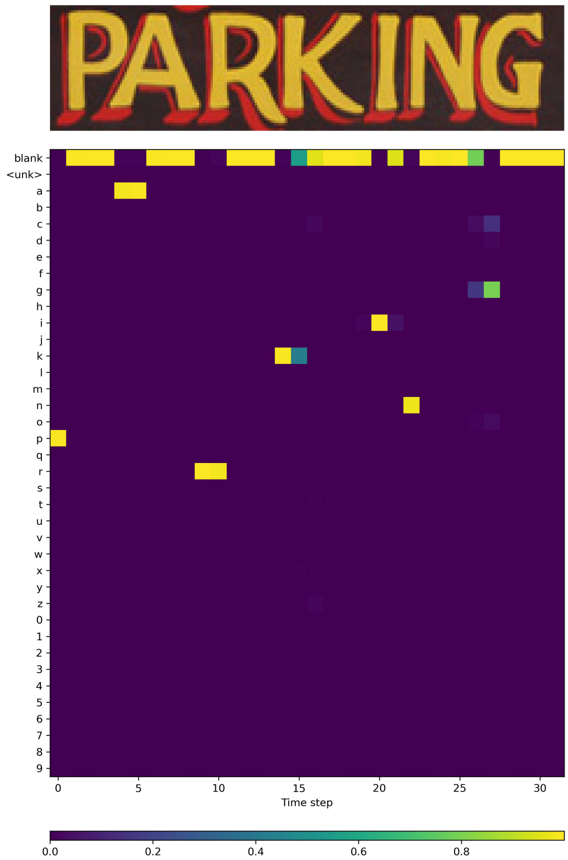
简要分析：由 cmd 输出结果可以看出，训练 40 轮之后的模型，在训练集上的 loss 可以降得比较小了，约为 0.8。同时，模型在验证集上的预测准确率只比 50% 略高，我认为这主要是因为训练数据量还是太小，同时没有进行初始参数的适当调整。由可视化结果看出，模型在训练集上的 loss 基本上还是逐步下降的，准确率是上升的一个趋势，说明训练效果较好。

3.2 使用训练好的模型预测新的文本图像

得到训练好的模型之后，使用命令 python main.py –mode predict –im_path data/my_own/a.png 来预测新的文本图像，其中输入图像为下图所示：

识别结果以及可视化结果如下：





简要分析：从上面的结果可以看出，该模型对这张图片的 parking 字符串识别效果很好，正确地识别出了每个字符。由可视化结果可以看出，按照

CTC 的输出序列为 p—aa—rr—k——i-n—-g—，去掉重复字符和空字符后为 parking，识别出的字符基本上是空字符或者正确字符，之后最后一个 G 概率稍低，识别成 C 的概率稍大，但仍然能够正确识别。能够这样进行完美识别的一个重要原因是这张图片的各个字符排列整齐，清晰且容易分辨。

3.3 本次作业遇到的问题及解决方法

无。

3.4 对本次作业的意见及建议

本次作业也较为简单，通过代码注释中的一步一步教程，我们能很快了解下一步该做什么，该如何实现。