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 6

2022 年 5 月 18 日

本次作业包括编程作业和调研报告两项任务。

第 1 题调研报告需要同学们调研媒体与认知相关技术在抗击新冠疫情中的相关应用，并写出自己对其的分析和感想。助教评判报告时更注重同学们的独到的想法，报告字数约 500 至 1000 字即可，课程组将根据作业提交情况邀请部分同学在课堂上做口头汇报。

编程作业通过 Transformer 网络实现场景文本识别任务，帮助同学们理解 Transformer 的基本原理并掌握其在训练和推理两阶段的实现方法。作业内容为 Transformer 的实现及模型优化求解方法，并在数据集上进行训练、测试和可视化，和对训练和推理过程的分析。具体任务分为编程部分以及作业报告，其中编程部分包含第 2、3 题，作业报告为第 4 题，已确认自选课题的同学需完成第 5 题。

1. 撰写调研报告（30 分）
2. 完成 Transformer 场景文本任务识别任务的程序代码（30 分）
3. 训练/测试/可视化（20 分）
4. 撰写作业报告（20 分）
5. 汇报自选课题进度（70 分）*

调研报告部分

1 撰写调研报告（30 分）

课程中介绍的机器学习及人类认知机制等媒体与认知相关技术在抗击新冠疫情中可以解决许多实际问题，同学们可以对以下实例进行调研分析或是搜集其他相关应用实例以完成调研报告。

- 深圳市委组织部开发的“战疫先锋”小程序，通过大数据分析和平台建设构建了“党组织引领 + 群众自组织”的战疫新模式[1]。清华大学

[1] https://baijiahao.baidu.com/s?id=1730718722527857223&wfr=spider&for=pc

深圳研究院及电子工程系多位师生对该平台志愿活动数据进行分析，并对如何利用“自组织效应”提升基层治理组织力提了多点建议，相关工作已经发表在了《Nature》期刊中[2]。

- 疫情期间卫生机构通过“互联网 +”信息平台，将同一区域内的各级医院及乡镇卫生院组成医疗联合体，加强基层医疗单位与各上级医疗单位的信息交流和共享。在这一基础上还他们还可以实现诊疗远程化，发挥大医院资源优势，借助信息技术下沉专家资源，提高基层和社区医疗卫生机构应对和处理疫情的能力[3]。
- 复旦大学博士生李小康通过“OCR 文字识别 + 正则化表达式筛选”的方案实现了对核酸检测截图的核查工作，将代码封装为小程序后，全校师生就能够不用手动收集核验核酸检测截图了[4]。

同学们在报告中需要对媒体与认知技术解决的某一个抗击新冠疫情相关的具体问题进行背景调研和现状描述，并根据自身对问题和技术的理解进行效果分析和给出未来优化现有技术的可行思路。同学们在做相关调研时需要阅读相关中英文文献，建议所使用的检索工具包括学校图书馆数据库资源中 CNKI 中国知网、Web of Science 核心合集、Engineering Village、Nature、Science 等。检索关键词例如：新冠肺炎、人工智能、机器学习等。英文检索关键词例如：covid-19、artificial intelligence、machine learning 等。同学们在此次作业中需要掌握规范引用文献的方式，报告中应至少包括两篇中文文献，作业模板中已经给出了引用中英文文献的样例。调研报告无严格格式要求，字数在 500 至 1000 字即可，报告需包含以下四个方面：问题背景调研（5 分），已有解决措施（10 分），技术效果分析（5 分），未来优化思路（10 分）。

编程部分

2 完成 Transformer 场景文本识别任务的程序代码（30 分）

本次作业将在上一次作业的基础上，将传统的“CNN-RNN-CTC”框架改进为“CNN-Transformer”框架进行场景文本识别。两次作业提取图像特征的过程均由 CNN 完成，而 Transformer 的输入和 RNN 类似，均为特

[2] https://www.nature.com/articles/s41599-022-01127-2.pdf

[3] https://sghexport.shobserver.com/html/baijiahao/2022/04/08/707661.html

[4] https://baijiahao.baidu.com/s?id=1729740855969579494&wfr=spider&for=pc

征序列，故同学们在本次作业中的特征层面相对上次作业无需作出太大改动，可以将精力放在理解 Transformer 的结构和训练推理过程中。例如 RNN 在训练和推理时都只能以“自回归”（auto-regressive）模式进行，也就是说网络必须首先生成 t 时刻的输出才能生成 t+1 时刻的输出。而 Transformer 则具有在训练时可以并行处理序列的特性，也就是网络只需要前传一次即可得到所有时刻的输出。

程序清单如下：

文件或目录	说明	注意事项
hw6.zip	作业 6 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
main.py	训练、验证及预测主程序	**需要完成代码**
network.py	CNN+Transformer 模型定义	**需要完成代码**
transformer_utils.py	Transformer 模型各模块定义	**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 44	完成整体模型 的初始化	调用 transformer_utils.py 中的 TransformerModel
TODO 2	network.py line 61	完成模型的 前向计算过程	需返回网络输出 的 logits 特征
TODO 3	network.py line 74	完成模型的 推理过程	需返回网络输出 的预测标签和 logits 特征
TODO 4	transformer_utils.py line 49	完成位置编码的 前向计算过程	需返回采用了位置编码 和 dropout 的输入
TODO 5	transformer_utils.py line 95	完成 Transformer 各模块的初始化	参照 nn.Transformer 源码 调用各 Transformer 子模 块
TODO 6	transformer_utils.py line 138	完成 Transformer 的推理过程	参考网络前向计算过程 完成逐步的解码过程
TODO 7	main.py line 141	完成整体网络的 的训练过程	注意 loss 的计算方式 优化目标需要移位

为便于同学们进一步理解 Transformer 原理，本次作业的 Transofmer 各模块的初始化（TODO 5）要求同学们参照 nn.Transformer 的源码中调用 nn.TransformerEncoder 等子模块方式实现，即同学们需要把助教习题课/辅导课中调用 nn.Transformer 的环节替换为更具体地实现方式。

注：torch.nn 中 transformer 相关的说明文档和使用教程地址为
https://pytorch.org/tutorials/beginner/transformer_tutorial.html
https://pytorch.org/docs/master/generated/torch.nn.Transformer.html#torch.nn.Transformer
https://pytorch.org/docs/master/nn.html#transformer-layers

Transfomer 包括一层编码器和一层解码器，模型的尺寸 d_models 设置为 32。在完成 network.py 和 transformer_utils.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。

3 训练/预测/可视化（20 分）

本次作业的训练集和第 5 次作业相同，均为从 ICDAR 2019 MLT 场景文本识别数据集 [5] 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 [6] 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，即大写字母“A”和小写字母“a”都对应于一个类别，输出时转为小写字母“a”显示。模型的字符集 C 总共包含 40 个字符，包括用于表示单词起始的 $\langle sos \rangle$，用于表示单词结束的 $\langle eos \rangle$，占位符的 $\langle pad \rangle$，用于表示其余所有未知字符的 $\langle unk \rangle$ 符号，以及 26 个英文字母和 10 个数字、即

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

（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：

```
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 229 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，并且 Transformer 相较于 RNN 更难训练，在使用默认网络参数的条件下，训练 100 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利用 CPU 从头训练模型 100 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。

[5] https://rrc.cvc.uab.es/?ch=15
[6] https://rrc.cvc.uab.es/?ch=2

- 如果训练过慢，作业中提供了已经预训练 100 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。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，以更好地理解观察 Transformer 的输出概率。可视化结果图片文件保存在“`data/my_own`”目录中。同学们可以观测到 Transformer 的解码和第 5 次作业中 CTC 的解码过程和可视化结果有明显区别。具体区别可以比较两次作业中 utils.py 中的 LabelConverter 的 decode 函数。

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。

4 撰写作业报告（20 分）

将 hw6 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）不必一并打包。本次报告中需要同学们分析 Transformer 在训练时具有并行训练特性的原因，并指出 Transformer 在推理解码时和 CTC 解码的区别及其原因，大家可以从解码的终止条件和具体原理进行分析。作业报告中包括任务 2、3 运行结果及分析，Tansformer 训练和推理分析，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等。推荐同学们使用随作业发布的 LaTex 模板 HW6-template.zip 完成作业报告。

5 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。对于未能按时说明原因的迟交作业，将酌情扣分。

本次作业责任助教为黄翌青 (Email: huang-yq17@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 100
lr	学习率，默认为 1e-3（本次作业优化器默认采用 RMSprop）
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