

作业 2

2023 年 3 月 29 日

本次作业通过卷积神经网络实现非线性分类任务，帮助同学们理解卷积神经网络的基本原理并掌握卷积神经网络的实现方法。作业内容为卷积神经网络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。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，作业报告为第 5 题，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. 完成卷积神经网络分类任务的程序代码 (30 分)
4. 训练/测试/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 以下关于卷积神经网络的说法错误的是

- (A) 全卷积网络常用于图像分割任务。
- (B) Batch Normalization 和残差连接均可用于缓解模型训练过程中梯度消失的问题。
- (C) 可变形卷积中的卷积核位移矢量需要用另一个网络来学习。
- (D) 空洞卷积通过增加卷积核的参数数量来增大感受野。

- 1.2 考虑某卷积层，已知输入数据的尺寸 (s_in)，卷积核的尺寸 ($kernel_size$)，步长 ($stride$) 和边界延拓 ($padding$)。以下关于输出数据的尺寸 (s_out) 的计算正确的是：

注： $\lfloor \cdot \rfloor$ 表示向下取整， $\lceil \cdot \rceil$ 表示向上取整

- (A) $s_out = \lfloor (s_in - kernel_size + padding) / stride \rfloor + 1$
- (B) $s_out = \lfloor (s_in - kernel_size + 2 * padding) / stride \rfloor + 1$
- (C) $s_out = \lceil (s_in - kernel_size + 2 * padding) / stride \rceil + 1$
- (D) $s_out = \lceil (s_in - kernel_size + padding) / stride \rceil + 1$

- 1.3 考虑某 batch normalization 层，其通道数为 1，在训练过程中计算得到的均值和方差分别为 -0.3 和 0.16。在测试过程中输入批次包含两个样本，输入该层的特征为 0.1 和 -0.5，那么该层的输出特征为：

注：batch normalization 计算公式为 $y = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta$ ，本题中假设 $\gamma = 1, \epsilon = 0, \beta = 0$

- (A) 0.3, -0.3
- (B) 1, -1
- (C) 1, -0.5
- (D) 2.5, -1.25

- 1.4 考虑某最大池化层，其 $kernel_size$ 为 2，步长 ($stride$) 为 2，边界延拓 ($padding$) 为 1。假设输入为 X ，输出为 Y 。现已知：

$$X = \begin{bmatrix} 0.1 & -0.2 & -0.1 & -0.2 \\ 0.0 & 0.3 & 0.2 & -0.3 \end{bmatrix}$$

那么 Y 的值为：

注：最大池化层的边界延拓是填充负无穷。

- (A) $Y = [0.3, 0.2]$

(B) $Y = [0.1, 0.3, 0.2]$

(C) $Y = \begin{bmatrix} 0.1 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.0 \end{bmatrix}$

(D) $Y = \begin{bmatrix} 0.1 & -0.1 & -0.2 \\ 0.0 & 0.3 & -0.3 \end{bmatrix}$

1.5 对于上一题中的最大池化层，假设损失函数为 L ，反向传播的梯度为 $\frac{\partial L}{\partial X}, \frac{\partial L}{\partial Y}$ 。现已知：

$$\frac{\partial L}{\partial Y} = \begin{bmatrix} -0.1 & 0.2 & 0.1 \\ 0.2 & -0.3 & -0.1 \end{bmatrix}$$

那么 $\frac{\partial L}{\partial X}$ 的值为：

(A) $\frac{\partial L}{\partial X} = \begin{bmatrix} -0.1 & 0.0 & 0.2 & 0.1 \\ 0.2 & -0.3 & 0.0 & -0.1 \end{bmatrix}$

(B) $\frac{\partial L}{\partial X} = \begin{bmatrix} -0.1 & 0.0 & 0.0 & 0.0 \\ 0.2 & -0.3 & 0.0 & 0.0 \end{bmatrix}$

(C) $\frac{\partial L}{\partial X} = \begin{bmatrix} -0.1 & 0.0 & 0.1 & 0.0 \\ 0.2 & -0.3 & 0.2 & -0.1 \end{bmatrix}$

(D) $\frac{\partial L}{\partial X} = \begin{bmatrix} -0.1 & 0.2 & 0.2 & 0.1 \\ 0.2 & -0.3 & -0.3 & -0.1 \end{bmatrix}$

2 计算题 (15 分)

2.1 已知某卷积层的输入为 X (该批量中样本数目为 1，输入样本通道数为 1)，采用一个卷积核 W ，即卷积输出通道数为 1，卷积核尺寸为 2×2 ，卷积的步长为 1，无边界延拓，偏置量为 b ：

$$X = \begin{bmatrix} -0.5 & 0.2 & 0.3 \\ 0.6 & -0.4 & 0.1 \\ 0.4 & 0.5 & -0.2 \end{bmatrix}, W = \begin{bmatrix} -0.2 & 0.1 \\ 0.4 & -0.3 \end{bmatrix}, b = 0.05$$

2.1.1 请计算卷积层的输出 Y 。

2.1.2 若训练过程中的目标函数为 L ，且已知 $\frac{\partial L}{\partial Y} = \begin{bmatrix} 0.1 & -0.2 \\ 0.2 & 0.3 \end{bmatrix}$ ，请计算 $\frac{\partial L}{\partial X}$ 。

注：本题的计算方式不限，但需要提供计算过程以及各步骤的结果。

编程部分

编程部分包括第 3，4 题。在本次任务中，我们将使用卷积神经网络识别交通标志图像，完成 26 类交通标志的分类。我们还将探索 batch normalization 和 dropout 在神经网络模型中的作用，尝试数据增强这一重要技巧，并自己动手搭建空间变换网络。最后，我们将采用可视化方法，进行数据、特征的展示。详细说明请参阅习题课课件。

本次作业将用到 scikit-learn 库。若尚未安装，请执行

```
conda install -c conda-forge scikit-learn
```

或

```
pip install -U scikit-learn
```

进行安装。

3 完成卷积神经网络分类任务的程序代码（30 分）

本次编程作业的程序清单如下：

文件或目录	说明	注意事项
hw2.zip	作业 2 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
\visualized	存放本次作业的可视化结果	请勿修改
datasets.py	数据集定义	需要完成代码
networks.py	神经网络结构定义	需要完成代码
train.py	训练程序	已完成代码
test.py	测试程序	已完成代码
visualize.py	可视化程序	已完成代码

需要完成的代码清单如下：每处需要完成的地方都有代码提示和步骤提示。

datasets.py 文件中待完成内容：

序号	行号	内容	说明
TODO 1.1	28	定义数据预处理变换	使用 <code>torchvision.transforms</code> 为训练/验证/测试集定义预处理变换，并添加对数据增广的支持
TODO 1.2	52	使用 <code>ImageFolder</code> 构建数据集	使用 <code>torchvision.datasets.ImageFolder</code>
TODO 1.3	61	使用数据加载器	使用 <code>torch.utils.data.DataLoader</code>

`networks.py` 文件中待完成内容：

序号	行号	内容	说明
TODO 2.1	42	完成卷积神经网络中卷积层和池化层的定义	请按照注释中提供的网络结构和参数完成定义
TODO 2.2	79	完成卷积神经网络中线性层的定义	请按照注释中提供的网络结构和参数完成定义
TODO 2.3	103	完成卷积神经网络的前向计算过程	请按照 step 2-4 完成卷积卷积神经网络的前向计算过程
TODO 3.1	142	完成空间变换网络 (STN) 的定义	请按照注释的提示，自行设计 STN 的网络结构。关于 STN 的详细介绍可见习题课课件
TODO 3.2	161	初始化 STN 的最后一层线性层的权重	使得 STN 在未训练前默认输出恒等变换

4 训练/测试/可视化 (30 分)

4.1 探究 batch normalization 和 dropout 的作用

本节要求同学们探究 batch normalization 和 dropout 对模型的影响。在默认参数下，没有启用 batch normalization，dropout 的概率为 0.0。请同学

们分别采用**默认参数**、**启用 batch normalization** 和 **启用 dropout**（**选择一种合适的概率即可**）训练模型，然后选择每种配置下的最优模型进行测试，并在作业报告中记录训练、测试数据和分析原因。

训练模型的命令示例如下。通过 `--ckpt_path` 参数可以指定模型的保存目录，便于后续测试和可视化；通过 `--epoch` 可以指定模型最大训练轮数。

1) 使用默认配置（不启用 BN 和 dropout），训练 baseline 模型：

```
python train.py --ckpt_path checkpoints/default
```

2) 启用 batch normalization：

```
python train.py --ckpt_path checkpoints/bn --bn
```

3) 启用 dropout 并设置概率为 0.3：

```
python train.py --ckpt_path checkpoints/dropout --dropout 0.3
```

训练过程的 loss 和验证集上 accuracy 的变化会在训练完成后以图片形式显示，并自动保存在 `ckpt_path` 文件夹中。关掉图片显示窗口后，程序方能退出。

测试模型的命令示例如下。通过 `--ckpt_path` 参数指定待加载模型的保存目录；通过 `--epoch` 指定待加载模型的轮数。

1) 测试默认配置模型，轮数为 10：

```
python test.py --ckpt_path checkpoints/default --epoch 10
```

4.2 探究数据增广的作用

在本节中，我们将基于 4.1 节中启用 batch normalization 的模型，加入数据增广的预处理变换进行训练。

1) 请你执行 `python datasets.py` 来测试你在 `datasets.py` 中定义的数据增广变换，结果会保存到 `visualized` 文件夹下的 `augmentation.jpg` 文件。请你在作业报告中附上该图片，列出你所使用的数据增广变换，并解释你选择它们的原因。

2) 请你训练模型并选择最优模型进行测试，在报告中附上训练和测试结果：

```
python train.py --ckpt_path checkpoints/bn_aug --bn --augment
```

```
python test.py --ckpt_path checkpoints/bn_aug
```

4.3 探究空间变换网络 (STN) 的作用

在本节中，我们将向 4.1 节中启用 batch normalization 的模型加入空间变换网络 (Spatial Transformer Network, STN)，以增强模型的空间不变性。

请你训练模型并选择最优模型进行测试，在报告中附上训练和测试结果：

```
python train.py --ckpt_path checkpoints/stn --bn --stn
```

```
python test.py --ckpt_path checkpoints/stn
```

4.4 可视化

请同学们任意选取一个模型（通过 `--ckpt_path` 和 `--epoch` 指定），使用 `visualize.py` 分别可视化 1) 卷积核, 2) 特征图, 3) t-SNE 降维后的最后一层隐藏层的输出特征, 4) 空间变换网络学习到的空间变换。请将使用的模型和可视化结果附在实验报告中，最好能简要描述观察到的结果。

注意：实验提供的卷积神经网络包含 5 个卷积层，可以任意选择若干个可视化卷积核和特征图；在特征图可视化时，可以指定任意序号的图片作为模型的输入；可视化结果会自动保存在 `visualized` 文件夹中。关掉图片显示窗口后，程序方能退出。

可视化的运行命令示例如下：

1) 可视化第 0 层卷积层的卷积核：

```
python visualize.py --type filter --ckpt_path checkpoints/bn --layer_idx 0
```

2) 可视化第 10 张图像第 1 层卷积层的输出特征图：

```
python visualize.py --type feature --ckpt_path checkpoints/bn --layer_idx 1 --image_idx 10
```

3) t-SNE 可视化最后一层隐藏层的输出特征：

```
python visualize.py --type tsne --ckpt_path checkpoints/bn
```

4) 可视化 STN 学习到的变换：

```
python visualize.py --type stn --ckpt_path checkpoints/stn
```

5 撰写作业报告（10 分）

请同学们将代码（**请勿包括数据集文件夹 `data`、可视化结果文件夹 `visualized` 和模型保存文件夹 `checkpoints`**）和作业报告打包为一个文件（例如 `*.zip`）提交到网络学堂。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 `HW2-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孙志尧 (Email: sunzy21@mails.tsinghua.edu.cn)。

7 附录

我们通过 argparse 库进行参数设置，可以查看 train.py, test.py, visualize.py 中可以调节的参数，可调节参数如下表所示。

	参数	说明
共有参数	ckpt_path	保存训练模型的路径
	device	代码运行时使用的设备, 可以为 cpu 或 cuda, 不指定则会优先使用 cuda
train.py	data_root	存放数据集的文件夹路径, 不用修改
	augment	是否进行数据增广
	epoch	训练轮数, 默认 10
	batch_size	训练时每一批数据中包含的数据个数
	lr	学习率
	momentum	SGD 优化器的动量, 只有 optim_type 设为 sgd 时使用
	weight_decay	L2 正则化系数
	optim_type	优化器类型, 可以设为 sgd, adam, adagrad, rmsprop 或 adadelta
	bn	是否使用 batch normalization
	stn	是否使用 spatial transformer network
	dropout	dropout 概率
	max_ckpt_save_num	保存训练模型数量最大值, 不用修改
	val_interval	进行验证的间隔
	resume	从已保存的模型恢复训练
test.py	path	存放数据集的文件夹路径, 不用修改
	epoch	载入模型的轮数, 默认 10
	save	是否保存测试结果
visualize.py	path	存放可视化图像的文件夹路径, 不用修改
	epoch	载入模型的轮数, 默认 10
	type	可视化类型, 可以设为 filter, feature, tsne 或 stn
	layer_idx	可视化卷积层层数, type 设为 filter 或 feature 时使用, 可以为 0、1、2、3、4
	image_idx	可视化图片索引值, type 设为 feature 时使用
	save_dir	可视化结果存放文件夹路径, 不用修改