媒体与认知 第二次作业

(V1.3　2020.4.3)

本次作业通过卷积层及卷积神经网络的编程实现，帮助同学们理解相关原理。作业内容包括卷积层的前向计算与误差反向传播过程，以及卷积神经网络模型设计实现。为便于理解，可将卷积层看作在输入数据上以局部感受野作为窗口进行扫描计算的全连接层，按步长移动窗口，并在不同窗口位置共享权值。具体任务包括：

一、　选择题　（有助于理解本次编程作业）

二、　补全程序代码

三、　自动评判程序实现效果

四、　提交程序及作业报告

本次作业原始成绩 100 分，选择题 10 分，补全程序代码 80 分，作业报告 10 分。

任务一、选择题

1. 对于如图 1 所示的一维卷积计算，请问可训练的参数总数量为：

(A) 9

(B) 18

(C) 10

(D) 20

(E) 其他结果，请写出：_____

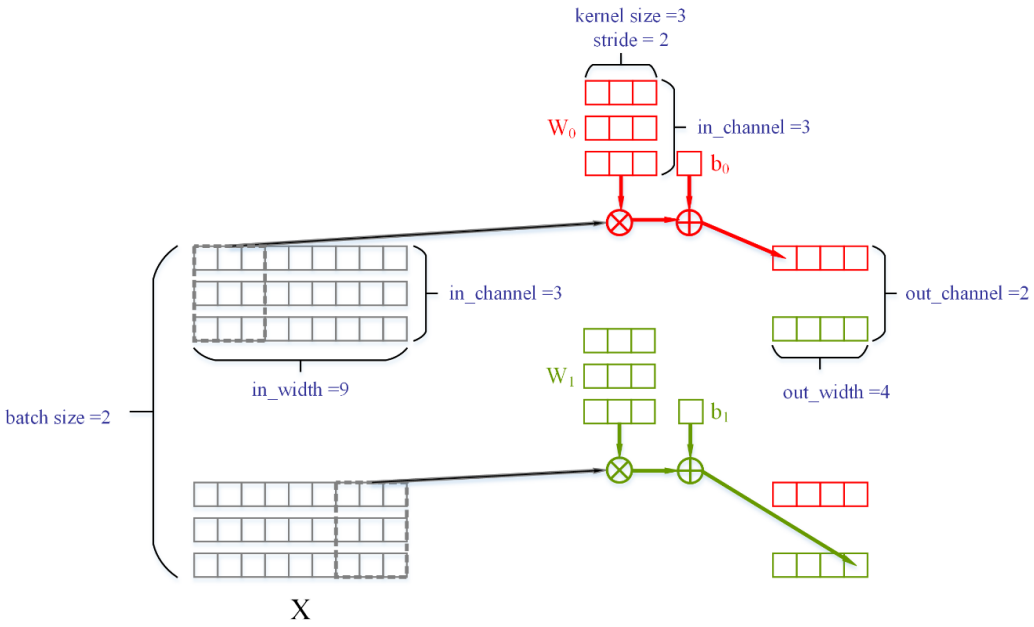


图 1. 一维卷积计算过程示例

2. 对于如图 1 所示的一维卷积，需要训练的参数为卷积权值 W 和偏置量 b。卷积计算过程对应的变量为：

batch_size = 2

input_width = 9

input_channel =3

kernel_size = 3

stride = 2

out_channel =2

out_width = 4

输入 x 和输出 result 均为 numpy 数组 (np.array)，其中：

x 的形状为: (batch_size, in_channel, input_size)；

result 的形状为(batch_size, out_channel, output_size)；

如下代码哪个是正确的？

(A) result[0, 1, 2] = b[0] + (W[0] * x[0, :, 2*stride:2*stride + kernel_size]).sum()

(B) result[0, 1, 2] = b[1] + (W[1] * x[1, :, 2*stride:2*stride + kernel_size]).sum()

(C) result[0, 1, 2] = W[0] + (b[0] * x[1, :, 2*stride:2*stride + kernel_size]).sum()

(D) result[0, 1, 2] = W[1] + (b[1] * x[0, :, 2*stride:2*stride + kernel_size]).sum()

(E) 其他结果，请写出：_____

3. PyTorch 的 Conv1d (https://pytorch.org/docs/stable/nn.html#torch.nn.Conv1d)，若 *padding=0*, *dilation=1*，一维卷积的输出数据宽度是多少？

(A) out_width = (in_width - kernel + stride) // stride

(B) out_width = ((in_width – 1) * (kernel - 1) - 1) // stride + 1

(C) 其他结果，请写出：_____

4. 对于图 2 所示的一维卷积计算，

$$y_1 = w_1 x_1 + w_2 x_2$$

$$y_2 = w_1 x_2 + w_2 x_3$$

$$y_3 = w_1 x_3 + w_2 x_4$$

损失函数为L，如下说法哪个是正确的?

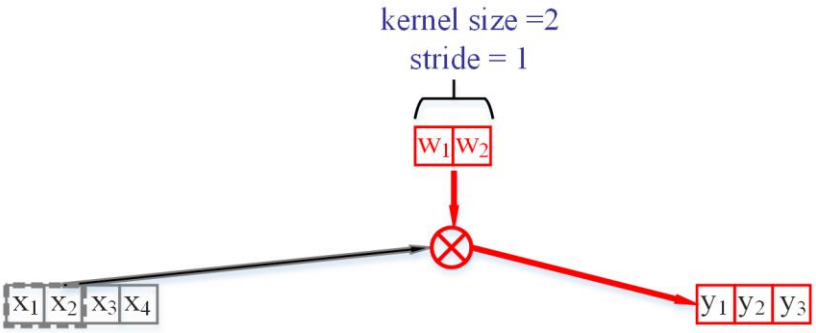


图 2 一维卷积计算简化示例

(A) $\frac{\partial L}{\partial w_2}=\frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial w_2}+\frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial w_2}=\frac{\partial L}{\partial y_1}x_1+\frac{\partial L}{\partial y_2}x_2$，$\frac{\partial L}{\partial x_2}=\frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_2}+\frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial x_2}=\frac{\partial L}{\partial y_1}w_1+\frac{\partial L}{\partial y_2}w_2$

(B) $\frac{\partial L}{\partial w_2}=\frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial w_2}+\frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial w_2}=\frac{\partial L}{\partial y_1}x_2+\frac{\partial L}{\partial y_2}x_3$，$\frac{\partial L}{\partial x_2}=\frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_2}+\frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial x_2}=\frac{\partial L}{\partial y_1}w_1+\frac{\partial L}{\partial y_2}w_2$

(C) $\frac{\partial L}{\partial w_2}=\frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial w_2}+\frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial w_2}=\frac{\partial L}{\partial y_1}x_2+\frac{\partial L}{\partial y_2}x_3$，$\frac{\partial L}{\partial x_2}=\frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_2}+\frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial x_2}=\frac{\partial L}{\partial y_1}w_2+\frac{\partial L}{\partial y_2}w_1$

(D) 其他结果，请写出：_____

5. 阅读如下文字，给出表 1 中变量的可能取值：

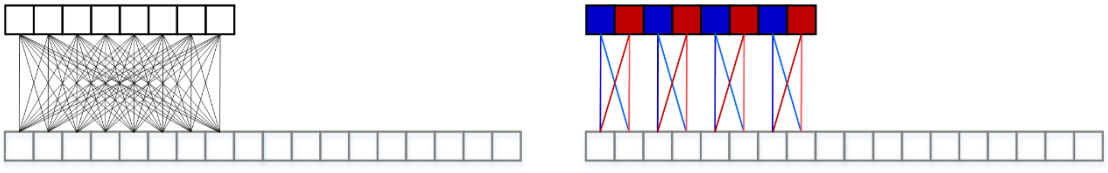
(A) L1=L2=L3=30，M1=64，M2=32，M3=30

(B) L1=L2=L3=31，M1=64，M2=32，M3=31

(C) L1=L2=L3=32，M1=64，M2=32，M3=32

(D) 其他结果，请写出：_____

关于把卷积看成用线性层扫描输入数据并在不同扫描窗口位置共享权值，作业给出的初始程序对比了不同卷积核大小的两种做法。卷积核作用于局部感受野时相当于全连接计算，因此，作业给出的初始程序装载对应的多层感知机的权值，对卷积层进行权值初始化。两种不同卷积核如图 3(a)和(b)所示。可以看出，图 3(b)所示的卷积核进一步缩小了局部感受野，并进行了权值共享（图中相同颜色的连接表示权值共享）。图 3 只显示了输入数据通道数为 1 的情形。



(a) (b)

图 3 用线性层扫描输入数据实现卷积

作业所附程序 mlp_scan.py 给出基于 CNN_SimpleScanningMLP 和 CNN_DistributedScanningMLP 的卷积网络两种实现方法。卷积网络共有 3 层，具体参数如表 1 示。用于初始化参数的权重参见“媒体与认知第 2 次作业附录.pdf”文件。两种实现方法的第一层分别对应于图 3(a)和(b)所示的卷积核。请注意输入数据通道数为 24，即每一个数据的维数为 24。输入数据宽度也可以理解为输入序列的长度。

表 1 用线性层扫描输入数据实现卷积的模型参数

方法	CNN_SimpleScanningMLP						CNN_DistributedScanningMLP					
卷积层序号	输入数据宽度 in_ width	输入数据通道数 in_channel	输出通道数(卷积核个数) out_channel	卷积核大小 kernel_size	步长 stride	输出数据宽度 out_width	输入数据宽度 in_ width	输入数据通道数 in_channel	输出通道数(卷积核个数) out_channel	卷积核大小 kernel_size	步长 stride	输出数据宽度 out_width
1	128	24	8	8	4	L1	128	24	2	2	2	M1
2	L1	8	16	1	1	L2	M1	2	8	2	2	M2
3	L2	16	4	1	1	L3	M2	8	4	2	1	M3

任务二、补全程序代码

程序清单如下：

目录	说明	注意事项
\program\hw2	hw2.py CNN mlp_scan.py 线性层扫描输入数据 mlp.py 多层感知机	hw2.py 和 mlp_scan.py 需要补全代码
\program\mytorch	conv.py 卷积层	需要补全代码
	linear.py 线性层	线性层的 b 和 db 的形状与第一次作业不同，上次作业为(1, out feature)，本次作业为(out feature)。请使用随本次作业给出的 linear.py。
	activation.py 激活函数 batchnorm.py 批量归一化 loss.py 目标函数	请将第一次作业已完成的 activation.py, batchnorm.py, loss.py 源文件拷贝过来。需要保证这些程序已通过第一次作业的自动评判测试。 activation.py 中的 sigmoid 激活函数需要考虑数值计算的稳定性，防止指数函数计算溢出。请参见：https://timvieira.github.io/blog/post/2014/02/11/exp-normalize-trick/ batchnorm.py 在本次作业中实际上并未用到。
\program\autograder\hw1_autograde	runner.py 自动评判编程作业的主控程序	自动评判程序 runner.py 利用存储的标准结果以及 PyTorch 对应程序，与同学们完成的模型得到的结果进行比较，给出各步骤通过与否的判断，此环节分值为 80 分。 请注意需要配置好 PyTorch 运行环境。

另外， \program\autograder\hw1_autograder 中的 \data, \ref_result, \weights 子目录分别为输入数据、存储的 CNN_SimpleScanningMLP 标准结果，以及 CNN_SimpleScanningMLP 和 CNN_DistributedScanningMLP 对应的多层感知机模型参数（用于卷积层初始化）。

请在程序“#ToDo:”及“???”提示处补全代码；补全代码后，可将原有的“raise NotImplemented”语句删除或注释掉。

需要补全代码的清单如下：

步骤 (Step)	内容	程序	初始程序中“#ToDo:”所在行数	说明
I	一维卷积层： 前向计算 误差反向传播	\mytorch\conv.py	49 76	一维卷积层相关说明可参见： https://tangshusen.me/Dive-into-DL-PyTorch/#/ 10.8.1 一维卷积层
II	把卷积看成用线性层扫描输入数据，并共享权值	\hw2\mlp_scan.py	36 99	利用多层感知机进行卷积层初始化的权值矩阵请参见随本次作业给出的附录“hw2-appendix.pdf”。
III	卷积神经网络模型	\hw2\hw2.py	60 73 97 131	卷积神经网络原理可参见： https://tangshusen.me/Dive-into-DL-PyTorch/#/ 5. 卷积神经网络

任务三、自动评判程序实现效果

Windows 平台	在 Windows 搜索框中搜索并打开 Anaconda Prompt。若需安装软件，可用鼠标右键点击“Anaconda Prompt”应用图标，选“以管理员身份运行”。 用 cd 命令进入作业所附程序解压生成的子目录 program。若要换盘符，需用 “<盘符>:”命令，比如从 c 盘换至 d 盘，键入 d: 和回车即可。 运行 python .\autograder\hw2_autograder\runner.py
Linux 平台	用 cd 命令进入作业所附程序解压生成的子目录 program。 运行 python ./autograder/hw2_autograder/runner.py

若程序运行无误，将输出如图 3 所示的信息。

```
--------------------------------------------------
Step I - Convolutional Layer
Step I - Forward
Conv1D Forward: PASS
Step I - Backward
Conv1D dX: PASS
Conv1D dW: PASS
Conv1D db: PASS
--------------------------------------------------
Step II - CNN as a Simple Scanning MLP
Scanning MLP: PASS
--------------------------------------------------
Step II - CNN as a Distributed Scanning MLP
Distributed MLP: PASS
--------------------------------------------------
Step III - CNN Complete Model
Conv1D Model Forward: PASS
Conv1D Model dX: PASS
Conv1D Model dW: PASS
Conv1D Model db: PASS
```

图 3. 自动评判程序预期输出信息

任务四、提交作业

在 Windows 平台上利用 WinZip 或 WinRAR 等软件将 \hw2 和 \mytorch 子目录打包在一个文件中，在 Linux 平台上可运行 sh create_tarball.sh 命令打包修改后的程序，将程序打包文件和作业报告一同提交到网络学堂。作业报告中包括选择题答案，任务三运行结果，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。

本次作业责任助教为姚刚（Email: yg19@mails.tsinghua.edu.cn）。

【致谢】本次编程作业原始资料由 CMU 11-785 Deep Learning 课程 http://deeplearning.cs.cmu.edu/ 主讲教师 Bhiksha Raj 教授提供。程序源代码由 6 字班彭逸凡同学协助整理，姚刚助教测试。

【关于 Python 及 numpy 数组切片索引】

1. Python 数组切片索引相关内容请参见：

https://github.com/lijin-THU/notes-python/blob/master/02-python-essentials/02.05-indexing-and-slicing.ipynb

2. numpy 数值计算编程基础知识请参见：

https://github.com/lijin-THU/notes-python/blob/master/03-numpy

其中，矩阵的切片索引和不完全索引、转置和点乘等操作请参见：https://github.com/lijin-THU/notes-python/blob/master/03-numpy/03.23-from-matlab-to-numpy.ipynb

【关于程序的编辑与调试】建议使用 Anaconda 所带的 Spyder，在 Anaconda Prompt 命令行窗口键入 spyder 即可启动 Spyder IDE 环境。程序编辑也可以常用的编辑器，并在程序需要之处用 print 函数输出程序运行相关信息（比如某个变量的取值）进行手动调试。

【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，若迟交在一周以内本次作业成绩予以九折（90%）处理，一周以上本次作业成绩予以八折（80%）处理。希望同学们理解并争取按时提交作业。