

媒体与认知 第一次作业

(V1.0 2020.3.13)

本次作业通过多层感知机的编程实现，帮助同学们理解深度学习架构的基本要素。作业内容为多层感知机模型及优化求解方法实现，以及在手写数字样本集上进行训练和验证。

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、 调整参数运行程序
- 五、 提交程序及作业报告

本次作业原始成绩 100 分，选择题 5 分，补全程序代码 85 分，作业报告 10 分。

任务一、选择题

1. 根据如下代码，指出 b, c 的数组维度。

```
a = np.arange(60.).reshape(3, 4, 5)
```

```
b = np.sum( a , axis=0, keepdims=True)
```

```
c = np.sum( a, axis=0)
```

(A) b.shape=(3, 4, 5) c.shape=(4, 5)

(B) b.shape=(1, 4, 5) c.shape=(4, 5)

(C) b.shape=(3, 4, 5) c.shape=(1, 4, 5)

(D) b.shape=(3, 4, 5) c.shape=(3, 4, 5)

2. 关于批量归一化，请阅读 <https://tangshusen.me/Dive-into-DL-PyTorch/#/> “5.10.1.1 对全连接层做批量归一化”，指出均值 (μ_B) 和方差(σ^2)的含义。

(A) 神经网络中每层每个神经元节点输出都有均值和方差，是依据每个批次数据计算的。

(B) 神经网络中每层都有均值和方差，是依据每层所有神经元节点计算的。

(C) 神经网络中每层每个神经元节点输出都有均值和方差，是依据所有测试集数据计算的。

3. 对于 Batchnorm，是否有必要保持训练数据的均值和均值？

- (A) 是。因此我们需要记录训练过程的均值和方差的估计值，以便在测试时使用。
- (B) 否。

4. 关于 python 编程中的 zip 函数，请阅读 <https://www.geeksforgeeks.org/zip-in-python/>。zip 函数有助于在一行代码中关联数据，例如，本次作业所附自动评判程序“\autograder\hw1_autograder\tests\test_problems.py”中的第 163 行“for i, (pred, gt) in enumerate(zip(dW, soldW)):”。请问你是否已阅读推荐的网页：

- (A) 是。
- (B) 否。

5. 如下哪种说法是正确的？其中 $w.shape = (input, output)$, $x.shape = (batch\ size, input)$, $b.shape = (1, output)$ 。注：如下使用点乘 dot 的地方也可以使用矩阵乘法 matmul。

- (A) $z = activationFunction(np.dot(x, b) + w)$
- (B) $z = activationFunction(np.dot(x, w)) + b$
- (C) $z = activationFunction(np.dot(x, w) + b)$

任务二、补全程序代码

程序清单如下：

目录	说明	注意事项
\program\hw1	hw1.py MLP 程序	需要补全代码
\program\mytorch	activation.py 激活函数 batchnorm.py 批归一化 linear.py 线性层 loss.py 目标函数	需要补全代码
\program\autograder\hw1_autograde	problems.py 自动评判编程作业任务 runner.py 自动评判编程作业的主控程序 data.pkl 自动评判编程作业参照数据 test.py 训练及测试 MLP 的主控程序 setup.cfg 系统配置文件	1. 运行 runner.py 自动评判程序利用 data.pkl 文件中存储的数据及标准结果，与同学们完成的模型得到的结果进行比较，给出编程作业积分，此环节分值为 85 分。 2. 运行 test.py 在作业所附的 MNIST 手写数字数据集上进行训练和验证。
\program\autograder\hw1_autograder\data	MNIST 手写数字样本训练集、验证集和测试集，已转为 numpy 格式	-
\program\autograder\hw1_autograder\tests	自动评判程序具体实现	-

请在程序“#ToDo:”及“???”提示处补全代码；补全代码后，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：

序号	内容	程序	初始程序中 “#ToDo:” 所在行数	说明
1	激活函数及其 导数： Sigmoid Tanh ReLU	\mytorch\activation .py	68 76 92 99 114 121	Sigmoid、Tanh、ReLU 相关说明可参见： https://tangshusen.me/Dive-into-DL-PyTorch/#/ 3.8.2.1 ReLU 函数 3.8.2.2 sigmoid 函数 3.8.2.3 tanh 函数
2	线性层（Linear Layer）	\mytorch\linear.py	39 55	线性层原理参见： https://tangshusen.me/Dive-into-DL-PyTorch/#/ 3.1 线性回归 3.1.2.2 向量计算表达式
3	目标函数	\mytorch\loss.py	57 67	LogSumExp Trick: http://cs231n.github.io/linear-classify/#softmax Derivative: https://web.stanford.edu/class/cs224n/readings/gradient-notes.pdf 参见第 4 页“Cross-entropy loss with respect to logits”
4	批量归一化	\mytorch\batchnorm.py	52 63 73 98	批量归一化原理参见： https://tangshusen.me/Dive-into-DL-PyTorch/#/ 5.10.1.1 对全连接层做批量归一化 求导数的方法参见： https://kevinzakka.github.io/2016/09/14/batch_normalization/
5	多层感知机	\hw1\hw1.py	67 102 117 121 139 143 147 151 178 233 240	多层感知机原理参见： https://tangshusen.me/Dive-into-DL-PyTorch/#/ 3.8 多层感知机

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 <code>python .\autograder\hw1_autograder\runner.py</code>
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 <code>python ./autograder/hw1_autograder/runner.py</code>

若程序运行无误，将输出如图 1 所示的信息。



The screenshot shows the output of the autograder program in an Anaconda Prompt window. The output includes a run time and a table of tasks and scores.

TASK	SCORE
Sigmoid Non-Linearity (forward)	2
Sigmoid Non-Linearity (derivative)	2
Tanh Non-Linearity (forward)	2
Tanh Non-Linearity (derivative)	2
ReLU Non-Linearity (forward)	2
ReLU Non-Linearity (derivative)	2
Softmax Cross Entropy (forward)	2
Softmax Cross Entropy (derivative)	2
Batch Normalization (training time)	10
Batch Normalization (inference time)	5
Linear Layer (Forward)	2
Linear Layer (Backward)	2
Linear Classifier (forward)	2
Linear Classifier (backward)	2
Linear Classifier (step)	1
Single Hidden Layer (forward)	5
Single Hidden Layer (backward)	5
N Hidden Layer (forward) 1	5
N Hidden Layer (forward) 2	5
N Hidden Layer (forward) 3	5
N Hidden Layer (backward) 1	5
N Hidden Layer (backward) 2	5
N Hidden Layer (backward) 3	5
Momentum	5
TOTAL SCORE	85

图 1. 自动评判程序预期输出信息

任务四、调整参数运行程序

Windows 平台	新建用于保存运行结果的目录，比如“c:\output” 打开 Anaconda Prompt， 用 cd 命令进入作业所附程序解压生成的子目录 program。 运行 <code>python .\autograder\hw1_autograder\test.py --outpath c:\output</code>
Linux 平台	新建用于保存运行结果的目录，比如“~/output”。“~” 代表用户主目录，即 “/home/<用户名>”。 用 cd 命令进入作业所附程序解压生成的子目录 program。 运行 <code>python ./autograder/hw1_autograder/test.py --outpath ~/output</code>

尝试至少一种不同的 MLP 网络结构，修改 test.py 程序 visualize()函数中的如下语句，比如增加隐含层的层数或节点数，对比训练集和验证集上的错误率。

```
mlp = nn.MLP(784, 10, [32, 32, 32], [nn.Sigmoid(), nn.Sigmoid(), nn.Sigmoid(),  
nn.Identity()], np.random.randn, bias_init, nn.SoftmaxCrossEntropy(),1e-3, momentum=0.856)
```

在运行程序命令行参数 “--outpath” 所指定的输出目录中，有程序生成的 4 幅图片，如图 2 所示，分别是训练过程中随训练轮数（Epoch，一轮即是将训练集数据全部处理一遍）变化的训练集上的损失和错误率、验证集上的损失和错误率。关于网络结构调整的实验结果对比，同学们可自行设计图表展示展示训练集和验证集上的错误率的变化。

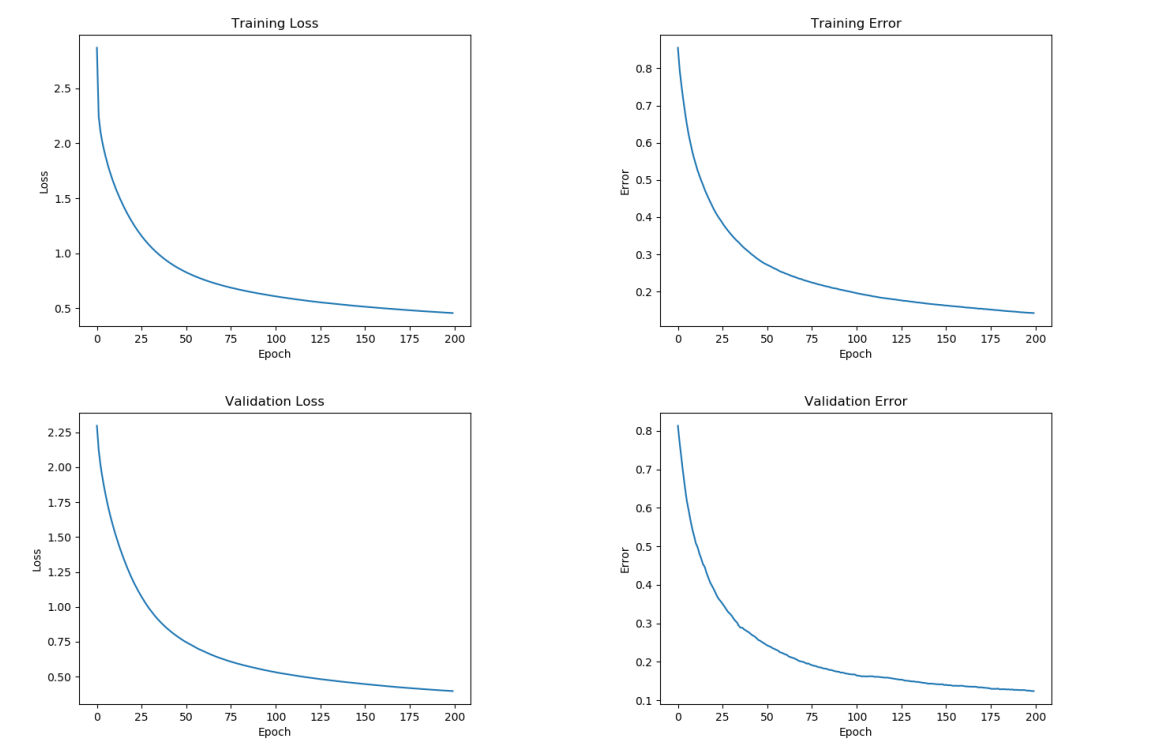


图 2 训练过程中训练集和验证集上的损失和错误率

任务五、提交作业

将 `\hw1` 和 `\mytorch` 子目录打包在一个文件中, Linux 平台可运行 `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、四运行结果, 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。

相关 python 语言 numpy 数值计算编程知识请参见:

<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 Python 编程中的缩进】

Python 使用缩进来表示代码块隶属关系, 不需要使用大括号 `{}`。同一个代码块的语句必须包含相同的缩进空格数。每一级缩进的空格数规范为 4 个, 不能出现 Tab、中文空格混用现象。可用 sublime text 工具软件(<https://www.sublimetext.com/>)打开代码, 全选所有内容就能看到哪些地方是空格, 哪些地方是 Tab, (全选所有内容状态下, 点表示空格, 横线表示 Tab), 需要将 Tab 替换成空格。

【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。

本次作业责任助教为黄翌青（Email: huang-yq17@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 字班秦达飞、彭逸凡同学协助整理，黄翌青助教测试。