DDA4220/MDS5122/AIR5011/AIR6011/MBI6011 Assignment 1

Due by: 23:59, Mar 16th, 2025

Instructions:

1. You must submit your assignment on Blackboard. Please upload a PDF file along with your code. The file should be renamed to something like **ZhangSan-123456789-hw1.pdf** (your name, student ID, homework ID).
2. Your submission must be clearly answered and well-presented to receive full credit. Please detail how you leverage your acquired knowledge to solve the problem and present your proposed solution step by step. Ensure that your solutions are legible and written in English.
3. You must not copy answers from other students, the Internet, or AI outputs (*e.g.*, GPT). Your answer should reflect your individuality and originality, and include a conclusion about what you have gained.
4. The programming language is Python, and your code should include necessary comments so that others can easily follow its logic.
5. If your answers are inspired by other sources (*e.g.*, discussions with classmates, the Internet, or AI), please maintain academic integrity by solving the problems as independently as possible and including a reference or acknowledgment section in your submitted document to reflect how these sources inspired you. Copying answers violates academic integrity.
6. Late submissions or instances of plagiarism will not be graded.

A. Build a Neural Network Using PyTorch (30 points)

To implement a deep learning application, the easiest approach is to write the code using a modern deep learning framework. The most popular framework in academia is PyTorch. In the first part, we will write an image classification application on toy datasets, CIFAR-10 and MNIST, and experiment with different factors to enhance performance on the testing dataset. Your code must be written using PyTorch 2.6.0 and Jupyter Notebooks. The notebooks must contain your experimental results and necessary comments.

You may try different network architectures, components, optimizers, training schemes, or any other factors you can think of to increase the accuracy. The accuracy is evaluated on the test set and should be as high as possible. You should detail the steps taken to enhance accuracy, including the factors you have considered and evaluated, and explain why you chose to incorporate each factor. Additionally, you should include the following parts in your submission to receive the corresponding credits.

`optimizer, data augmentation, network structure.`

Scoring Criteria:

1. Set up the environment and reproduce the accuracy in our code. **(5 points)**
2. What factors did you consider to improve the accuracy? For example, you might try using a deeper and wider convolutional neural network, a better optimizer, data augmentation methods, dropout layers, etc. Higher accuracy will earn higher credits. Conducted on the CIFAR-10 dataset. **(15 points)**
3. Use the same (or a similar) network and training scheme to retrain and evaluate its performance on the

MNIST dataset, with the best solutions you previously obtained on the CIFAR-10 dataset. **(5 points)**

4. Your code and submitted document are clean and well-organized. For example, the paragraphs are well organized; the results are presented in tables and effectively visualized through high-quality figures; the documents are written in LaTeX. **(3 points)**
5. What have you learned through your attempts in the assignment? **(2 points)**

B. Build a Neural Network From Scratch (70 points)

To gain a better understanding of what a neural network is, you are required to write, train, and evaluate your own deep neural networks without using any deep learning libraries. You are forbidden from using any deep learning frameworks, such as PyTorch and TensorFlow; instead, you should only use basic libraries like NumPy to implement the functionalities. We note that you may use other libraries, such as PyTorch, to facilitate data loading; however, the deep-learning related functions must be implemented without them.

Use the same network architecture and training scheme to reproduce the accuracy you previously reported in the third requirement (use the same (or a similar) network…) of Part A (the MNIST dataset). The accuracy should be close to the one you previously reported.

Notice:

1. You should implement both forward and backward passes for each network module, and they should be similar to how PyTorch implements them, *i.e.*, the module is implemented as a Python class. You may inspect the source code of PyTorch to see how they actually implement the functionalities.
2. You must not use any pre-existing convolution functions to compute convolutions. You should implement your own convolution functions using matrix multiplication and element-wise operations.
3. Some network modules shown below are marked with *config*, indicating that they should be initialized with these parameters during their initialization. Please refer to PyTorch's documentation for the meanings of these names. Your implementations should closely resemble those of PyTorch.
4. You may optionally implement them with vectorization or CuPy to boost speed; this is optional.

Scoring Criteria:

1. Implement the following network modules: **(48 points)**
   a) Sigmoid **(2 points)**
   b) LeakyReLU, with config: *negative_slope* **(2 points)**
   c) SELU **(3 points)**
   d) Linear, with config: *in_features, out_features, bias* **(2 points)**
   e) Conv2d, with config: *in_channels, out_channels, kernel_size, stride, bias* **(12 points)**
   f) Dropout, with config: *p* **(3 points)**
   g) BatchNorm2d, with config: *num_features* **(10 points)**
   h) FocalLoss, with config: *alpha, gamma* **(8 points)**
   i) SGD, with config: *lr, momentum* **(2 points)**
   j) Adam, with config: *lr* **(4 points)**
2. With the modules you wrote earlier, use the same network architecture and training scheme to reproduce the accuracy you obtained in Part A on the MNIST dataset. **(15 points)**
3. Your code and submitted document are clean and well-organized. For example, the paragraphs are well organized; the results are presented in tables and effectively visualized through high-quality figures; the documents are written in LaTeX. **(5 points)**
4. What have you learned through your attempts in the assignment? **(2 points)**

Appendix

Environment Setup Guide:

0. Assume your computer is running Win10/11. If you are using macOS or Linux, the steps below are still quite similar.
1. Firstly, install conda on your system. We suggest installing Miniconda because its size is smaller. There are many installation tutorials available on the Internet (Chinese Tutorials, English Tutorials, Mirror Download), and we assume you have already installed it before proceeding.
2. All commands below are typed into the Command Prompt. To open it, press the "`Windows`" key along with the "**R**" button simultaneously. A small window will pop up in the bottom left corner; type "`cmd`" in the text box and press the "`Enter`" key. After that, a black window will appear where you can type any commands you want and press the `` `Enter` `` key to execute them.
3. Create the environment with "`conda create python=3.10 -n DL-HW-Py310 -y`" and then activate it with "`conda activate DL-HW-Py310`"
4. Follow PyTorch's installation guide to install the latest version. For example, on my system, I have CUDA 12.7 installed, so I need to run "`pip3 install torch torchvision --index-url https://download.pytorch.org/whl/cu126`" to install PyTorch. Since CUDA is backward compatible, please install a version that is not newer than the CUDA version you have installed. If you do not have any NVIDIA graphics cards, you can only install the CPU version.
5. Install additional libraries: "`pip3 install matplotlib ipykernel`". We would expect you to plot some figures using the `matplotlib` library; for example, visualizing the loss curve is a good practice.
6. Install a coding environment. We recommend using VSCode as the development environment. To run Jupyter notebook inside VSCode, you may need to follow this guide.
7. After all the steps above are completed, the environment should be perfectly set up.

If you have any issues, please don't hesitate to contact me at `jiabaolei@link.cuhk.edu.cn`