NVLab Summer School 2023

Homework 2: Neural Networks

Announced: 2023 / 7 / 19 (三)
Deadline: 2023 / 8 / 9 (三) at 23:59 PM

林育丞 (st11206x@gmail.com)
莊佩蓁 (sandy.purin@gmail.com)

Introduction:

In this assignment, you need to build a shallow neural network as well as the **mini-batch SGD** training process on **Fashion-MNIST** dataset from scratch using **Python3**. The performance of your model should surpass 85% accuracy on testing data. **Please download HW2.zip from google drive first.**

Rules:

★ Built-in machine learning libraries (Sklearn, PyTorch, TensorFlow, Keras ……) are not allowed, you need to use basic mathematical operations in **NumPy** to define the behavior of each layer.
★ Do not use testing data to train the shallow neural network.
★ Please properly comment your code to let us understand your train of thought.
★ Discussions are encouraged, but plagiarism is strictly prohibited!

Data:

Fashion_MNIST dataset was packaged in ***.gz**. We will provide the *.gz to you. The first thing that you must do is to unpacked *.gz. Here, you can use **"gzip"** function to get ubyte data. Finally, we want you to be able to transfer data types from **ubytes** to **numpy arrays**.

```
# fashion mnist dataset path
url_train_image  = 'Fashion_MNIST_data/train-images-idx3-ubyte.gz'
url_train_labels = 'Fashion_MNIST_data/train-labels-idx1-ubyte.gz'
url_test_image   = 'Fashion_MNIST_data/t10k-images-idx3-ubyte.gz'
url_test_labels  = 'Fashion_MNIST_data/t10k-labels-idx1-ubyte.gz'
```

NVLab Summer School 2023

Implementation:

1. **Fashion-MNIST dataset:**
   reading & data-type transform.

2. **Design a two-layers FCNN model (1 hidden layer + 1 output layer):**
   Every output neuron has full connection to the input neurons.

3. **ReLU layer:**
   We add nonlinear activation functions after the neural layer using ReLU.

4. **Softmax output:**
   The final layer is typically a Softmax function which outputs the probability of a sample being in different classes.

5. **Cross-entropy loss calculation:**
   Use the output of a batch of data and their labels to calculate the CE loss.

6. **Backward propagation:**
   Propagate the error backward and update each parameter.

7. **Testing accuracy**

Submission:

You should compress your **code** (*.py), **model weights** (*.npy), dataset, report (HW2_ report.pdf), and **README.txt** (explain how to run your code) into a ZIP file (HW2_ name.zip, ex. **hw2_林育丞.zip**), and finally mail the **ZIP** file to st11206x@gmail.com before the deadline. We will reproduce your results based on the codes, weights, and readme file you gave.

The zip file that you give us should **at least** contain the following information. Don't upload the **Fshion-MNIST** folder again.

HW2 _name.zip

| HW2_report.pdf
| README.txt
| HW2.py (only *.py)
| best_model_weight.npy

NVLab Summer School 2023

Report:

The format is not limited, but the following matters must be discussed in your report (Plots and tables are encouraged to use in presentation):

1. Show the **model architecture, implementation detail, and testing accuracy**. Please describe the methods to achieve the final result in detail, e.g., epochs, batch size, etc.

2. During testing, the model might **overfit training data** to achieve poor performance. How do you **check overfitting** and **underfitting** during training? Please describe the methods and experiments to prove that.

3. During training, you need to **adjust the hyperparameters** (epochs, batch size, …) to achieve higher accuracy. How do you choose these hyperparameters? Please conduct experiments to show that these hyperparameters are suitable to achieve higher accuracy for the final test results. (Please use **table** to summarize your results)

4. Please do experiments to compare at least other **one** activation function in the model. Analyze the activation function you used and show if these activation functions help improve the performance during testing.

If you meet any problem, please feel free to ask us via above emails!!