Deep Learning HW2

曾以諾
Institute of Data Science
National Cheng Kung University
Tainan, R.O.C.
re6111032@gs.ncku.edu.tw

***Abstract*—This report is an description document for HW2. And the GitHub link is as follows**

GitHub Link:
https://github.com/butterfly2012010/DeepLearning_HW2

I. Introduction

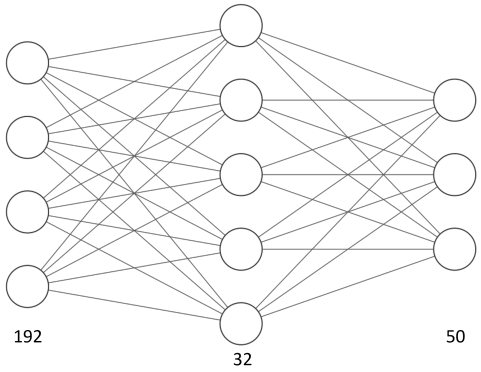
In homework 2, there are three requirements to complete,

- Two layer perceptron
- LeNet5
- Improved LeNet5

II. Implementation

A. Two Layer Perceptron

1) The architecture of my NN is as following,
   - #neurons:
     a) input layer: 192
     b) hidden layer: 32
     c) output layer: 50
   - initial weights:
     Use uniform distribution to initial the weights from $[-\frac{1}{\sqrt{\text{input\_size}}}, \frac{1}{\sqrt{\text{input\_size}}}]$.
   - EPOCH: 30
   - learning rate: 1e-3
2) In the TwoLayerPerceptron, it use *forward* method to do forward pass, and use *backward* to compute the gradient and update the weights and biases.
3) Model structure:



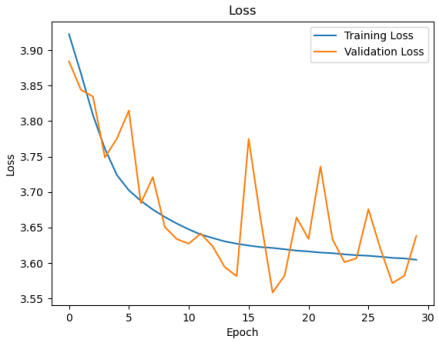
Layer 1:

$$y_1 = W_1 X + b_1$$
$$y_2 = Sigmoid(y_1)$$

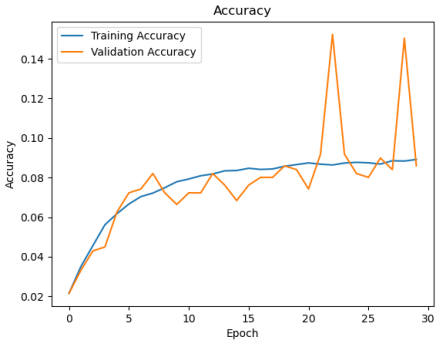
Layer 2:

$$y_3 = W_3 y_2 + b_3$$
$$y_4 = Softmax(y_3)$$

4) Results
   - Loss
     The plot of train and validation loss



   - Accuracy
     The plot of train and validation accuracy

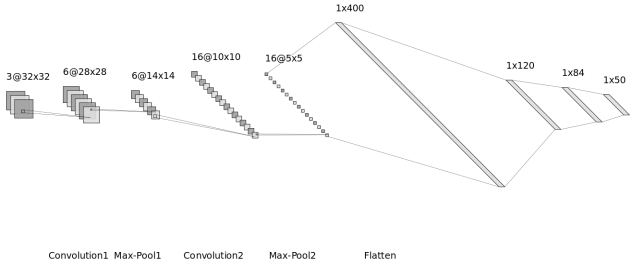


B. LeNet5

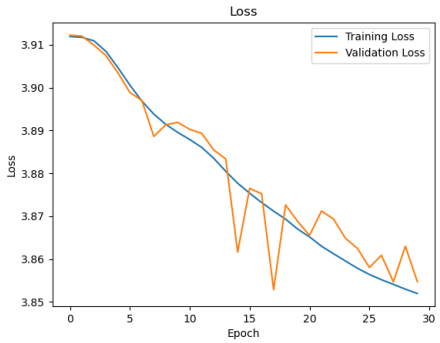
In this part, I use the reference code provided by Professor Toxtli, and modify the Sigmoid class and the activation layer of model because the local gradient of Sigmoid class is wrong and the activation function of the original code is a ReLU function.

- optimizer: SGD

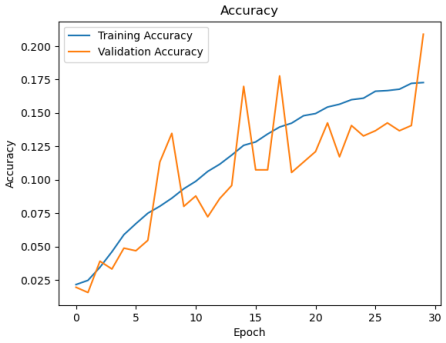
- learning_rate: 1e-3



- Loss



- Accuracy

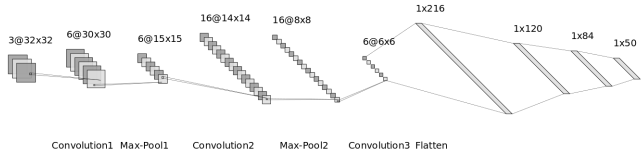


C. *ImprovedLeNet5*

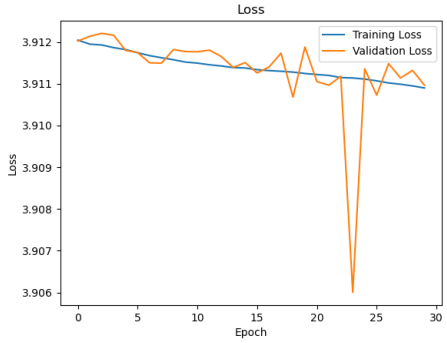
In the ImprovedLeNet5, there are three modifications.

- Activation function: x = Sigmoid(x) ⇒ x = x * Sigmoid(x)
- Kernel size: 5x5 ⇒ 3x3
- Increase one convolution layer to LeNet5 (added after maxpool2 layer in my implementation)

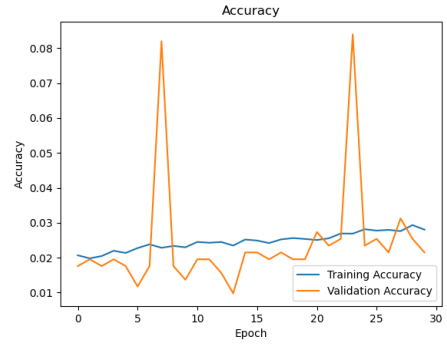
- optimizer: SGD
- learning rate: 1e-4



- Loss



- Accuracy



III. Comparison between LeNet and improved LeNet

Compared with the original LeNet5, I am of the opinion that the improved LeNet5 should perform better since the added convolution layer can subtract more information from the previous feature map. However, it looks worse than the original LeNet5 from the result of loss and accuracy. In my point of view, the problem probably arise from that the input image size is small (32x32) and hence there is not much information in the input, so the result of model is poor.

TABLE I
Performance

Methods	Top-1 Accuracy		
	train	validation	test
Two Layer Perceptron	0.0889	0.0978	0.0711
LeNet5	0.0280	0.1644	0.2000
Improved LeNet5	0.0583	0.0689	0.0667

IV. References

References

[1] ML_From_Scratch,https://github.com/eriklindernoren/ML-From-Scratch

[2] ChatGPT