청년 AI, BigData 아카데미 27기
컴퓨터비전 실습과제 1 (HW1)

기한: 24/09/12 (목) 23:59

- Total Questions: 2 (HW1-1, 1-2)
- Datasets:
  1) FashionMNIST dataset
  2) MNIST dataset
- 제출 항목:
  1) Completed AICV_HW1.ipynb file
     - Change the file name to AICV_HW1_[**YOUR_NAME**].ipynb
     - Ex) AICV_HW1_홍길동.ipynb
     - All cells should be executable

1. Build Simple CNN

Complete HW1-1 Build Simple CNN section in AICV_HW1.ipynb file. Replace the "# Implement code here" with the relevant Python code for building a simple CNN model.

- Load FashionMNIST dataset and make DataLoader(given)
- Show first 10 samples(given)
- Define a Simple CNN Model :
    - Create a class SimpleCNN inheriting from nn.Module.
    - Implement two convolutional layers (conv1, conv2) followed by a max pooling layer (pool).
    - Add two fully connected layers (fc1, fc2).
- Set Up Loss and Optimizer :
    - Use CrossEntropyLoss for classification.
    - Choose an optimizer, such as Adam with a learning rate of 0.001.
- Train the Model :
    - Loop through training data for a set number of epochs.
    - For each batch, compute the forward pass, calculate the loss, perform the backward pass, and update weights.
- Evaluate the Model :
    - Compute the accuracy on the test set using a no-gradient context.

2. AlexNet implementation

Complete HW1-2 AlexNet implementation section in AICV_HW1.ipynb file.

- Use MNIST dataset
- Do not use pre-defined alexnet model from any library
- Layer Specification에 padding, stride 가 적히지 않은 경우는 default로 각각 0, 1입니다.

Network Architecture

Layer	Type	Specification
1	Convolution	# of channels: 96, kernel_size: 5x5, padding: 2
2	Activation	ReLU
3	Pooling	Max pooling, kernel_size: 3x3, stride: 2, padding: 1
4	Convolution	# of channels: 256, kernel_size: 5x5, padding: 2
5	Activation	ReLU
6	Pooling	Max pooling, kernel_size: 3x3, stride: 2, padding: 1
7	Convolution	# of channels: 384, kernel_size: 3x3, padding: 1
8	Activation	ReLU
9	Convolution	# of channels: 384, kernel_size: 3x3, padding: 1
10	Activation	ReLU
11	Convolution	# of channels: 256, kernel_size: 3x3, padding: 1
12	Activation	ReLU
13	Pooling	Max pooling, kernel_size: 3x3, stride: 2
14	Dropout	p = 0.5
15	Linear	# of output neurons: 4096
16	Activation	ReLU
17	Dropout	p = 0.5
18	Linear	# of output neurons: 1024
19	Activation	ReLU
20	Linear	# of output neurons: 10

Model Summary (torchsummary)

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 96, 28, 28]           2,496
              ReLU-2          [-1, 96, 28, 28]               0
         MaxPool2d-3          [-1, 96, 14, 14]               0
            Conv2d-4         [-1, 256, 14, 14]         614,656
              ReLU-5         [-1, 256, 14, 14]               0
         MaxPool2d-6           [-1, 256, 7, 7]               0
            Conv2d-7           [-1, 384, 7, 7]         885,120
              ReLU-8           [-1, 384, 7, 7]               0
            Conv2d-9           [-1, 384, 7, 7]       1,327,488
             ReLU-10           [-1, 384, 7, 7]               0
           Conv2d-11           [-1, 256, 7, 7]         884,992
             ReLU-12           [-1, 256, 7, 7]               0
        MaxPool2d-13           [-1, 256, 3, 3]               0
          Flatten-14                [-1, 2304]               0
          Dropout-15                [-1, 2304]               0
           Linear-16                [-1, 4096]       9,441,280
             ReLU-17                [-1, 4096]               0
          Dropout-18                [-1, 4096]               0
           Linear-19                [-1, 1024]       4,195,328
             ReLU-20                [-1, 1024]               0
           Linear-21                  [-1, 10]          10,250
================================================================
...
Forward/backward pass size (MB): 3.08
Params size (MB): 66.23
Estimated Total Size (MB): 69.31
----------------------------------------------------------------
```

TA's result (epoch=2):

Test metric	DataLoader 0
test_acc	0.10279999673366547
test_loss	2.302518129348755

⇨

Test metric	DataLoader 0
test_acc	0.9746000170707703
test_loss	0.09115520864725113