Assignment #3

AATG013/ANT5013 Deep Learning and Computer Vision

Assigned: Tuesday, 3/28, 2023

Due: Tuesday, 4/4, 2023 (by midnight)

- Colab code 1 set을 사이버캠퍼스에 제출
  - assignment3-1a.ipynb
  - assignment3-1b.ipynb

code는 colab에서 수행되어 결과를 보여야 합니다.

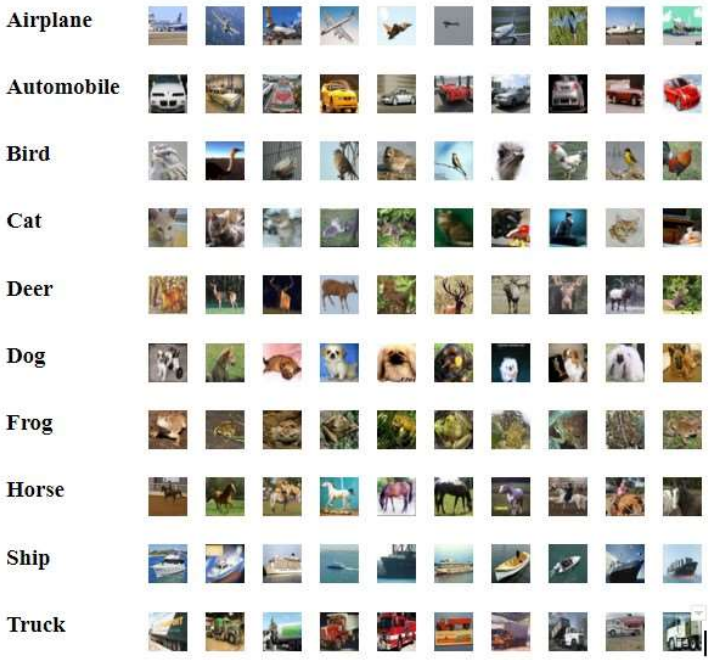
Problem [1] PyTorch code: image classification using cfar10

Problem [1] [50 points]

이 과제에서는CFAR10 dataset을 사용하여 FC MLP (fully-connected multilayer perceptron) NN을 구성하고, image classification network을 구성합니다. CFAR10 dataset은 FashionMNIST와 마찬가지로 10 classes의 물체 images로 구성되어 있으나, 그 images의 내용에 있어서 FashionMNIST images 보다 약간 더 복잡도가 높습니다. [60 points]

	image size	train images	validation images
FashionMNIST	28x28x1	60,000	10,000
CFAR10	32x32x3	50,000	10,000

https://github.com/PacktPublishing/Modern-Computer-Vision-with-PyTorch



본 과제에서는 위의 github chapter 3에 있는 code을 약간 개조하여 image classification network을 design, train, validate하는 pyTorch code를 작성하고 제출합니다.

1. 교재에서의 FashionMNIST diaply와 같이 10x10 images를 display/show하는 Colab code, assignment 3-1a.ipynb 제출 [20 points]

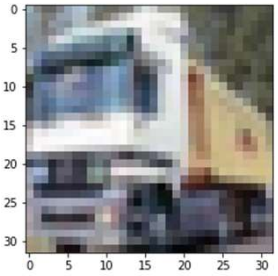
2. train dataset으로 train한 W (weights)를 보이고, train loss, validation loss, train accuracy, validation accuracy 등의 plot를 보이는 Colab code, assignment 3-1b.ipynb 를 제출합니다. [40 points] .

Problem [1] :

```
import torch
import torch.nn as nn
device = "cuda" if torch.cuda.is_available() else "cpu"

from torch.utils.data import Dataset, DataLoader
from torchvision import datasets, transforms

import matplotlib.pyplot as plt
import numpy as np

transform = transforms.ToTensor()

data_path = '../data_cifar/'
cifar10_train = datasets.CIFAR10(data_path, train=True, download=True, transform=transform)
cifar10_val   = datasets.CIFAR10(data_path, train=False, download=True, transform=transform)

print("Training: ",   len(cifar10_train))   #Training: 50000
print("Validating: ", len(cifar10_val))     #Testing:  10000

type(cifar10_train)           # torchvision.datasets.cifar.CIFAR10
type(cifar10_val)             # torchvision.datasets.cifar.CIFAR10

type(cifar10_train[0])        # tuple

image, label = cifar10_train[1]
type(image)                   # torch.Tensor
image.shape                   # torch.Size([3, 32, 32])
```

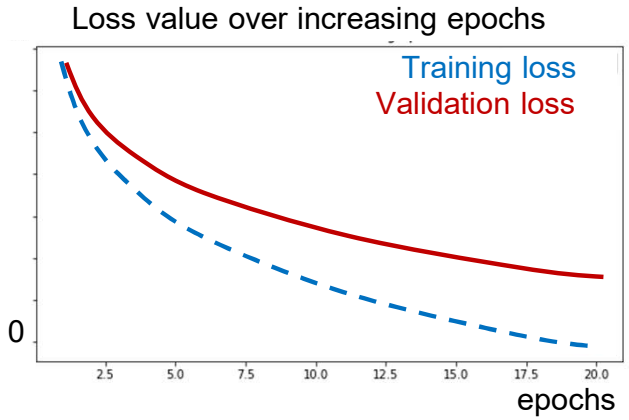
Problem [1] :

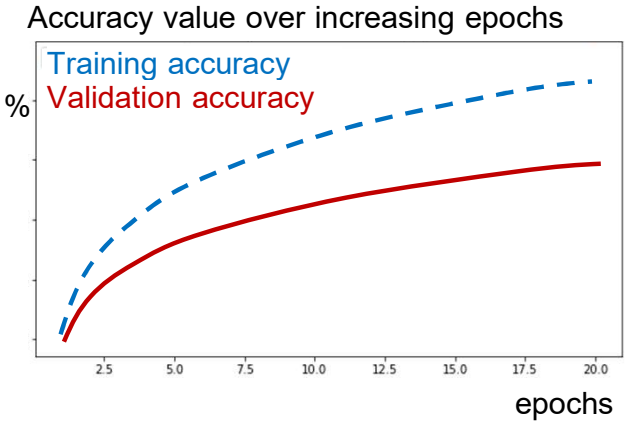
```
classes = cifar10_train.classes
print (classes)                 # ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog',
                                # 'frog', 'horse', 'ship', 'truck']
print(label)                    # 9
print(classes[label])           # truck

plt.imshow(image.permute(1, 2, 0))
```



```
torch.manual_seed(80)
train_loader = DataLoader(cifar10_train, batch_size=100, shuffle=True)
val_loader   = DataLoader(cifar10_test, batch_size=500, shuffle=False)

# . . .
def get_model():
    model = nn.Sequential(
        nn.Linear(3 * 32 * 32, 120),
        nn.ReLU(),
        nn.Linear(120, 84),
        nn.ReLU(),
        nn.Linear(84, 10)
    ).to(device)
    loss_fn = nn.CrossEntropyLoss()
    optimizer = SGD(model.parameters(), lr=1e-2)
    return model, loss_fn, optimizer
```

Problem [1] :





- **`train loss, validation loss, train accuracy, validation accuracy`** 등을 plot
- Train batch size는 100, validation은 전체 10000
- **`Tensor, array`** 등은 **`dimension`**이 맞도록 주의.
- 주어진 상황에서 최대의 성능을 이끌어내도록 다음을 조정
  - Scaling
  - Optimizer
  - Learning rate
  - Learning rate anealing
- CFAR10 image들은 FashonMNIST image들 보다 그 복잡도가 높으므로 FC MLP로는 accuracy를 높이는 데 한계가 있음.
  - 약 50%의 accuracy를 달성하면 제출 가능
  - CNN 등 특별한 image 전용 net의 필요성 부각.