CSCI3230 / ESTR3108 2021-22 First Term Assignment 4

I declare that the assignment here submitted is original except for source material explicitly acknowledged, and that the same or closely related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the following websites.

University Guideline on Academic Honesty:

`http://www.cuhk.edu.hk/policy/academichonesty/`

Faculty of Engineering Guidelines to Academic Honesty:

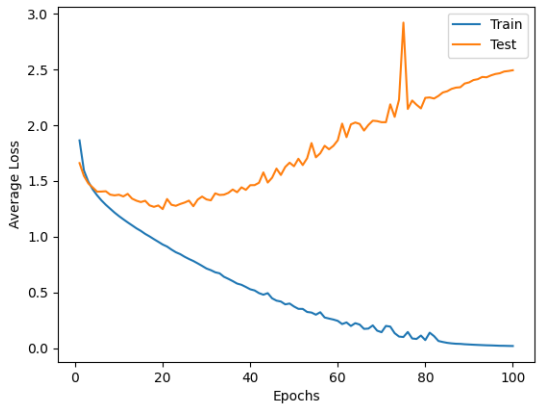
`http://www.erg.cuhk.edu.hk/erg-intra/upload/documents/ENGG_Discipline.pdf`

Student Name: ***Lai Man Hin***

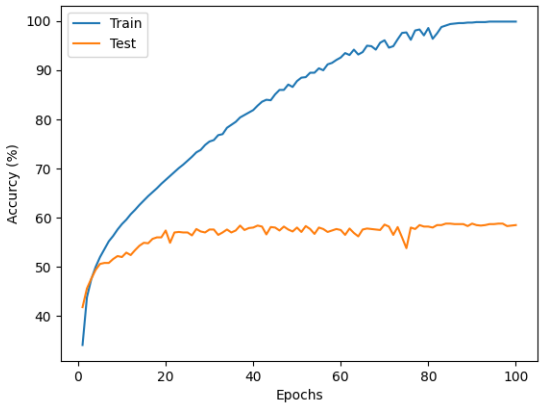
Student ID : ***1155136167***

1a)

i) The curves of training loss and testing loss in a single figure



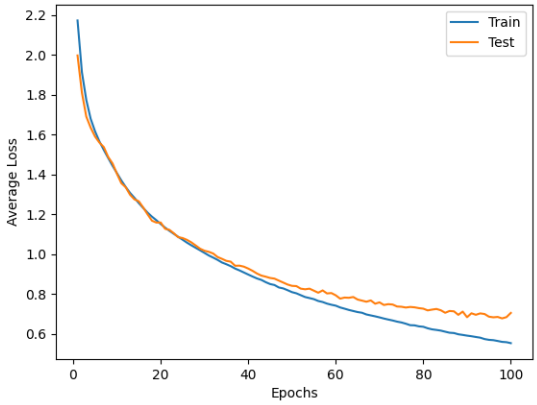
ii) The curves of training accuracy and testing accuracy in a single figure



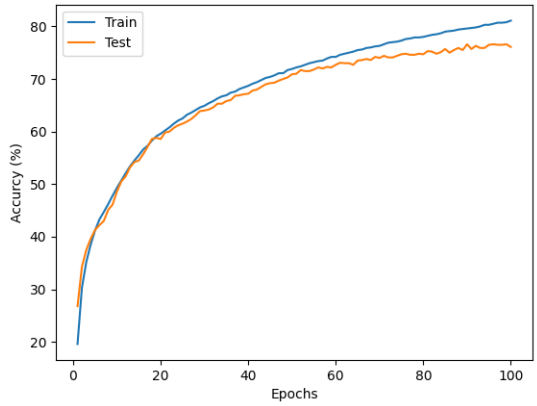
iii) The best test accuracy that achieved: 58.8%

1b)

i) The curves of training loss and testing loss in a single figure



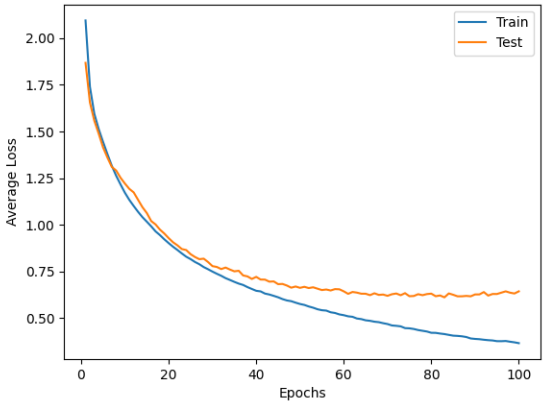
ii) The curves of training accuracy and testing accuracy in a single figure



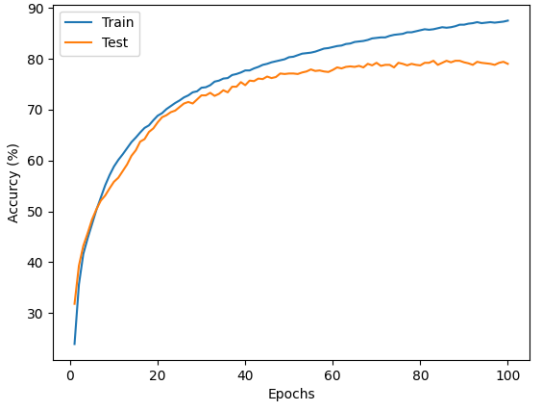
iii) The best test accuracy that achieved: 76.6%

1c)

i) The curves of training loss and testing loss in a single figure



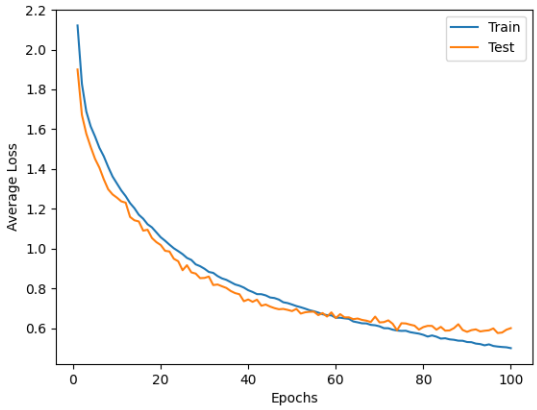
ii) The curves of training accuracy and testing accuracy in a single figure



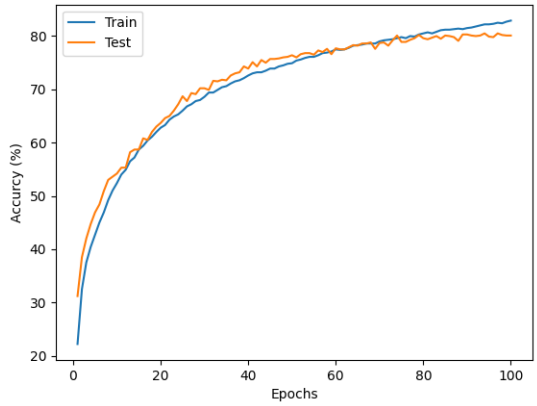
iii) The best test accuracy that achieved: 79.6%

1d)

i) The curves of training loss and testing loss in a single figure



ii) The curves of training accuracy and testing accuracy in a single figure



iii) The best test accuracy that achieved: 80.5%

1e)

i) I used EfficientNet-B7 (PreTrained) as training models and used AutoAugment function to let the data augment policy to be used in the data training (See figure below)

CNN CIFAR10 Pytorch Tutorial - 4

File Edit View Run Add-ons Help

Cancel Run Code Draft Session (3h:33m)

```
    def forward(self, x):
        x = self.conv_relu_stack(x)
        # print(x.shape)
        x = self.flatten(x)
        # print(x.shape)
        logits = self.linear_relu_stack(x)
        return logits

model = models.efficientnet_b7(pretrained=True)
model.cuda()
# for X, y in test_dataloader:
#     output = model(X)
#     break
print(model)
```

```
Using cuda device
Downloading: "https://download.pytorch.org/models/efficientnet_b7_lukemelas-dcc49843.pth" to /root/.cache/torch/hub/checkpoints/efficientnet_b7_lukemelas-dcc49843.pth
100% 255M/255M [00:09<00:00, 29.1MB/s]

EfficientNet(
  (features): Sequential(
    (0): ConvNormActivation(
      (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (1): BatchNorm2d(64, eps=0.001, momentum=0.01, affine=True, track_running_stats=True)
      (2): SiLU(inplace=True)
    )
    (1): Sequential(
      (0): MBConv(
        (block): Sequential(
          (0): ConvNormActivation(
            (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=64, bias=False)
            (1): BatchNorm2d(64, eps=0.001, momentum=0.01, affine=True, track_running_stats=True)
            (2): SiLU(inplace=True)
          )
          (1): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
```

CNN CIFAR10 Pytorch Tutorial - 4 Draft saved

File Edit View Run Add-ons Help

Share

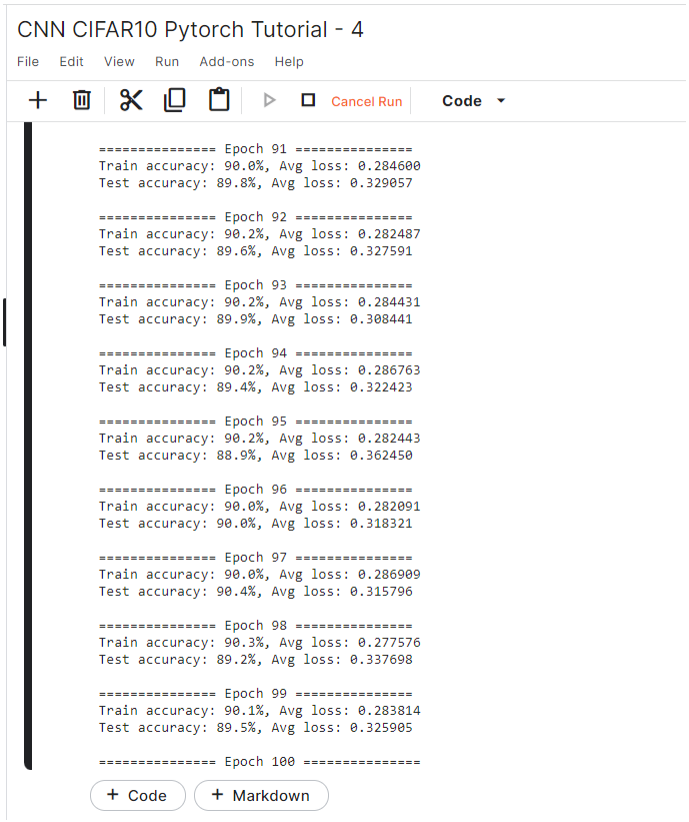
Cancel Run Code Draft Session (3h:34m)

[5]:

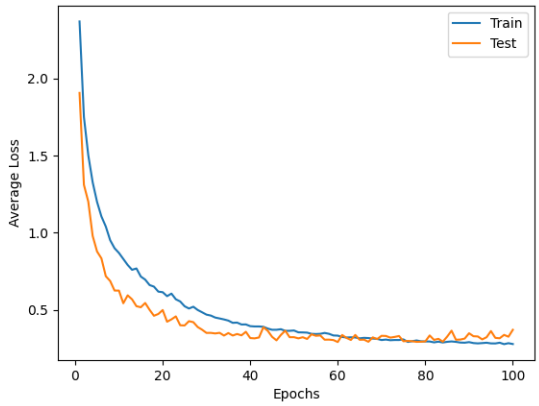
```
# Train transformation
train_transform = transforms.Compose([
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.AutoAugment(transforms.AutoAugmentPolicy.CIFAR10),
    transforms.RandomCrop(32, padding=4),
    transforms.ToTensor(),
    transforms.Normalize((0.5,), (0.5,))
])
# Test transformation
test_transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.5,), (0.5,))
])

# Download training data from open datasets.
training_data = datasets.CIFAR10(
    root="data",
    train=True,
    download=True,
    transform=train_transform,
)

# Download test data from open datasets.
test_data = datasets.CIFAR10(
    root="data",
    train=False,
    download=True,
    transform=test_transform,
)
```

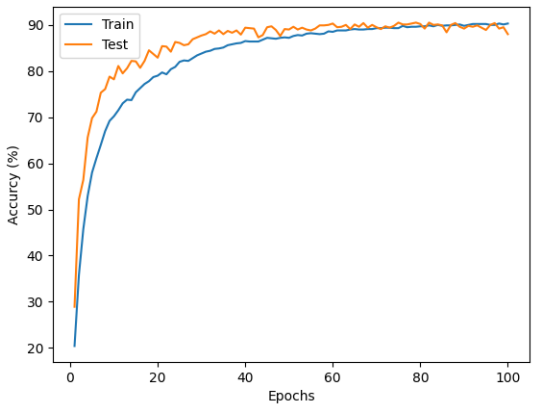


ii) The curves of training loss and testing loss in a single figure



(Continue in next page)

iii) The curves of training accuracy and testing accuracy in a single figure



iv) The best test accuracy that achieved: 90.5%