Artificial Intelligence Assignment #2

Professor: YoungWuk Kim

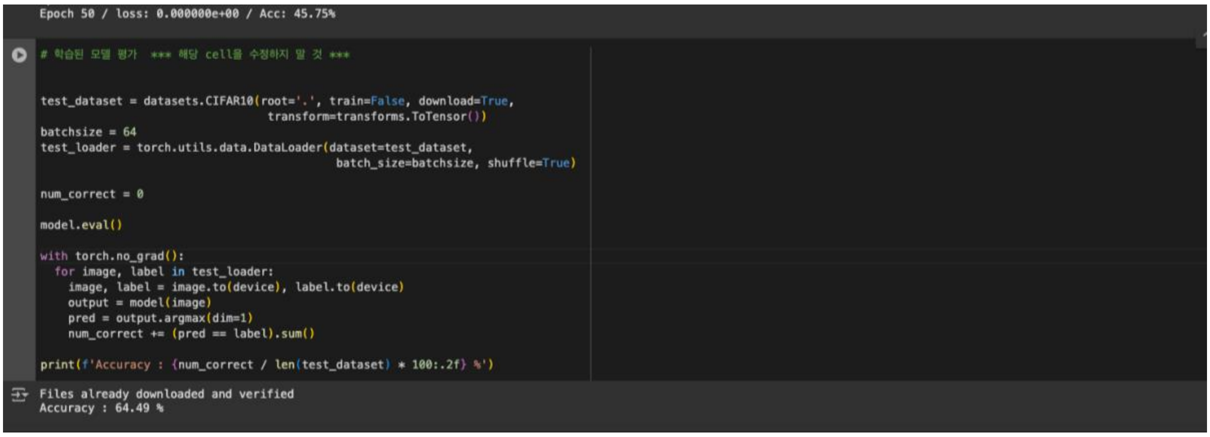
Class: Artificial Intelligence #2

No: 20203153

Name: JinU Choi

1-A)

lr=1e-4, epoch=50

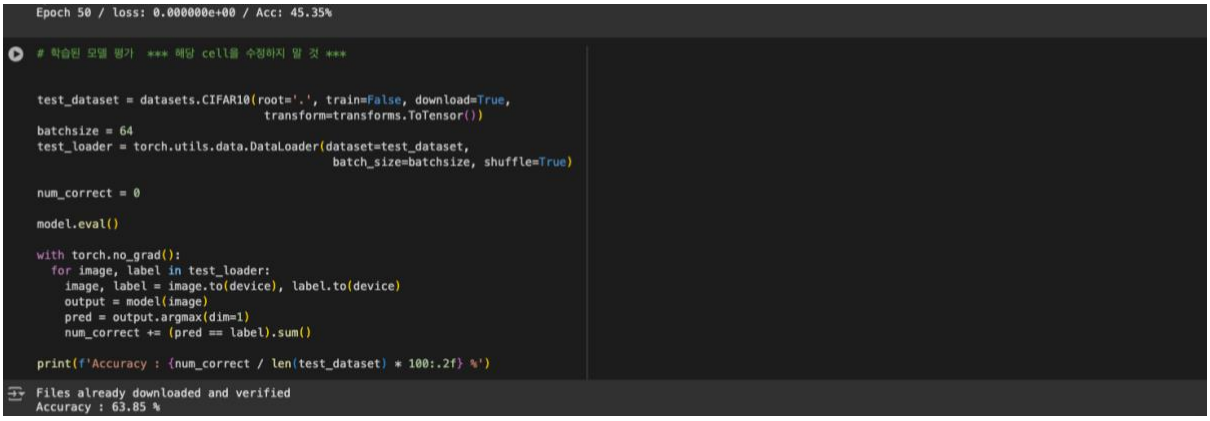


```
Epoch 50 / loss: 0.000000e+00 / Acc: 45.75%
```

```python
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 64.49 %
```

lr=1e-3, epoch=50

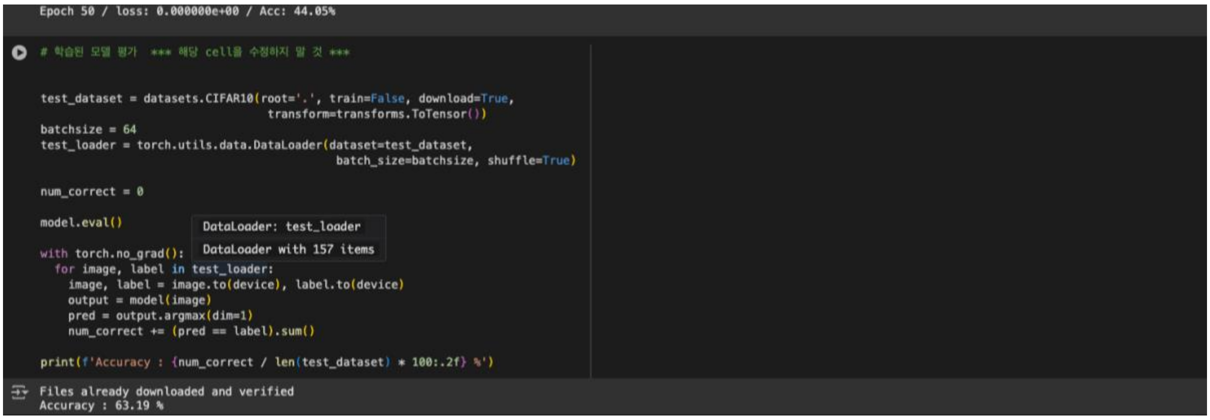


```
Epoch 50 / loss: 0.000000e+00 / Acc: 45.35%
```

```python
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 63.85 %
```

lr=15e-4, epoch=50



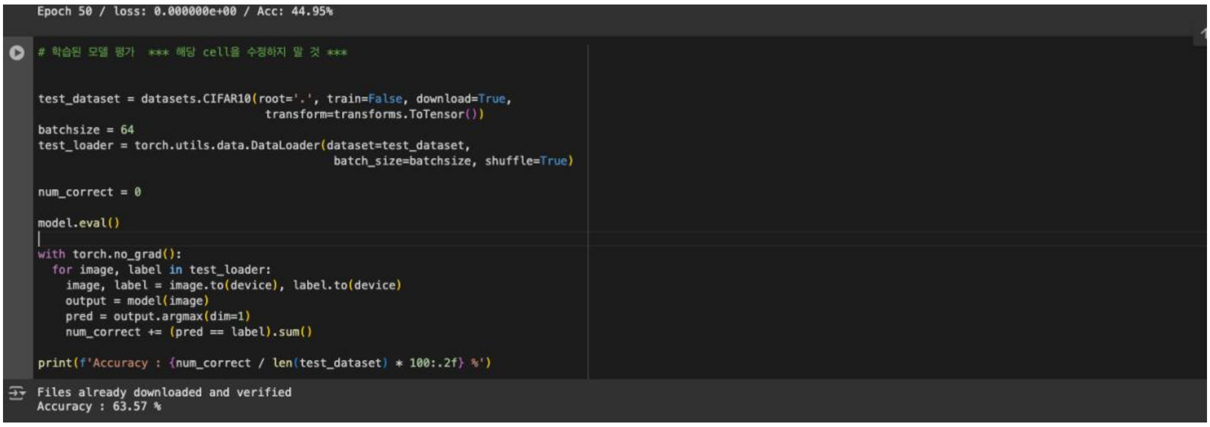
```
Epoch 50 / loss: 0.000000e+00 / Acc: 44.05%
```

```python
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
DataLoader: test_loader
DataLoader with 157 items
```

```
Files already downloaded and verified
Accuracy : 63.19 %
```

lr=15e-5, epoch=50

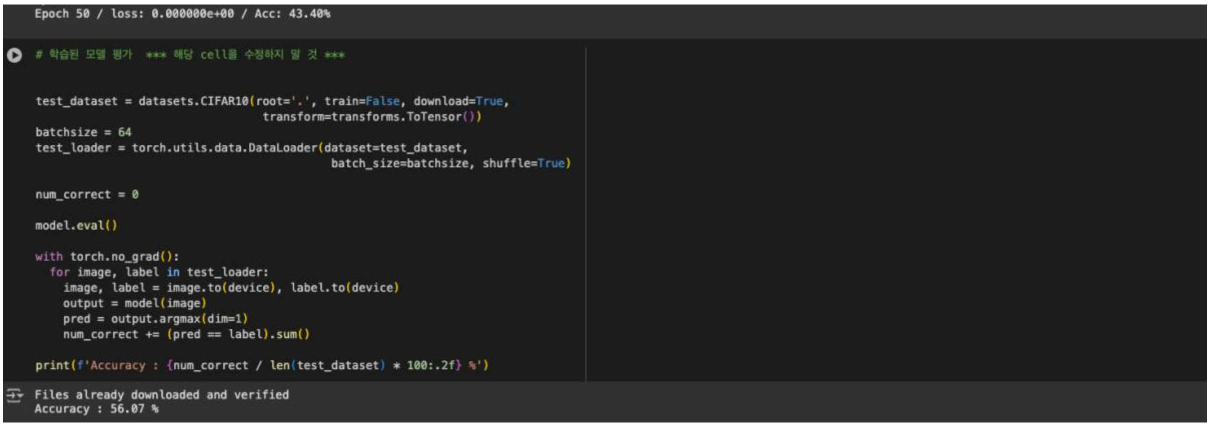


```
Epoch 50 / loss: 0.000000e+00 / Acc: 44.95%
```

```
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 63.57 %
```

Lr=1e-5, epoch=50

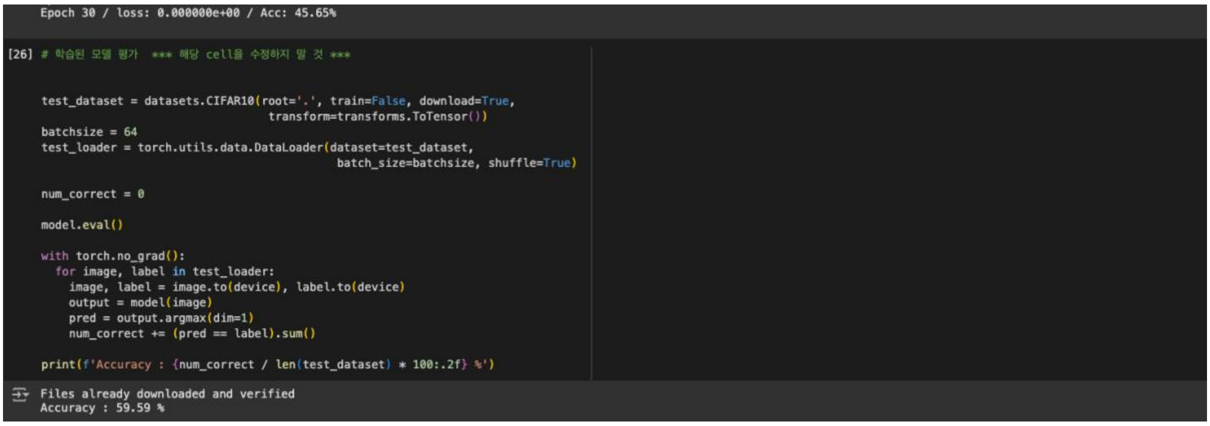


```
Epoch 50 / loss: 0.000000e+00 / Acc: 43.40%
```

```
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 56.07 %
```

Lr=1e-4, epoch=30

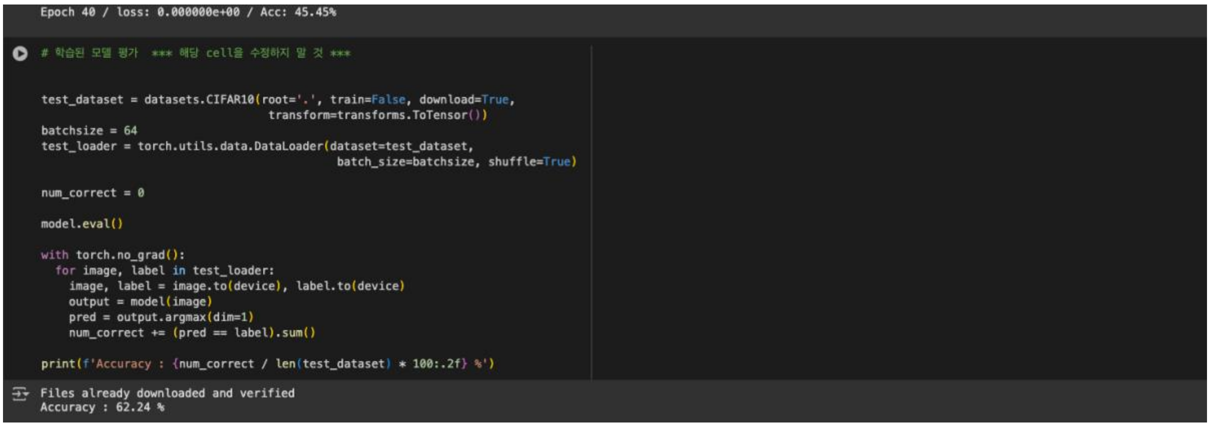


```
Epoch 30 / loss: 0.000000e+00 / Acc: 45.65%
```

```
[26] # 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 59.59 %
```

Lr=1e-4, epoch=40

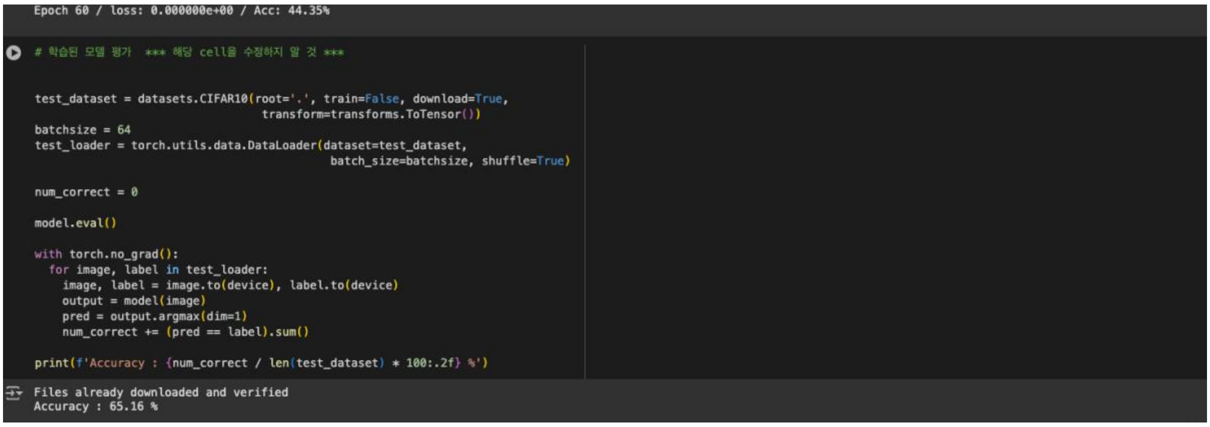


```
Epoch 40 / loss: 0.000000e+00 / Acc: 45.45%
```

```
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                            batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 62.24 %
```

Lr=1e-4, epoch=60

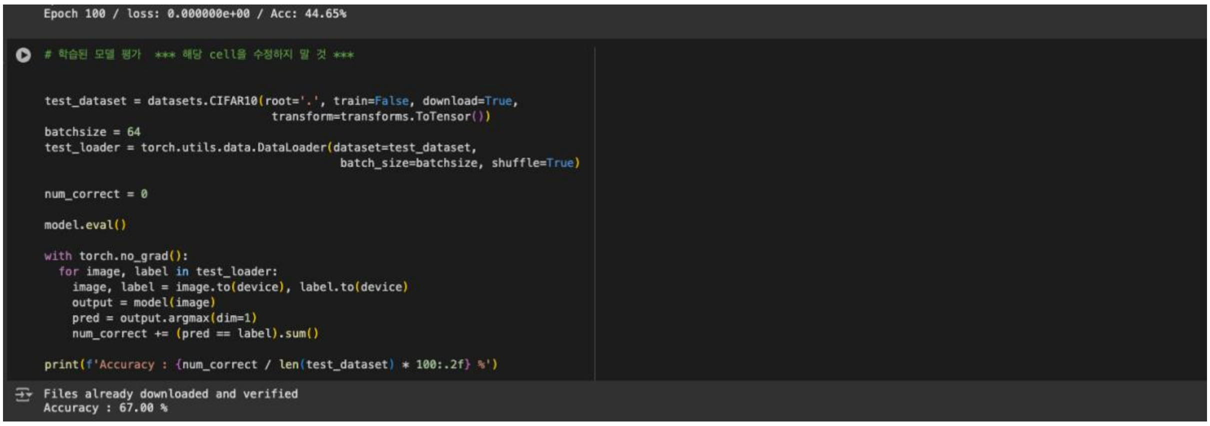


```
Epoch 60 / loss: 0.000000e+00 / Acc: 44.35%
```

```
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                            batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 65.16 %
```

Lr=1e-4, epoch=100

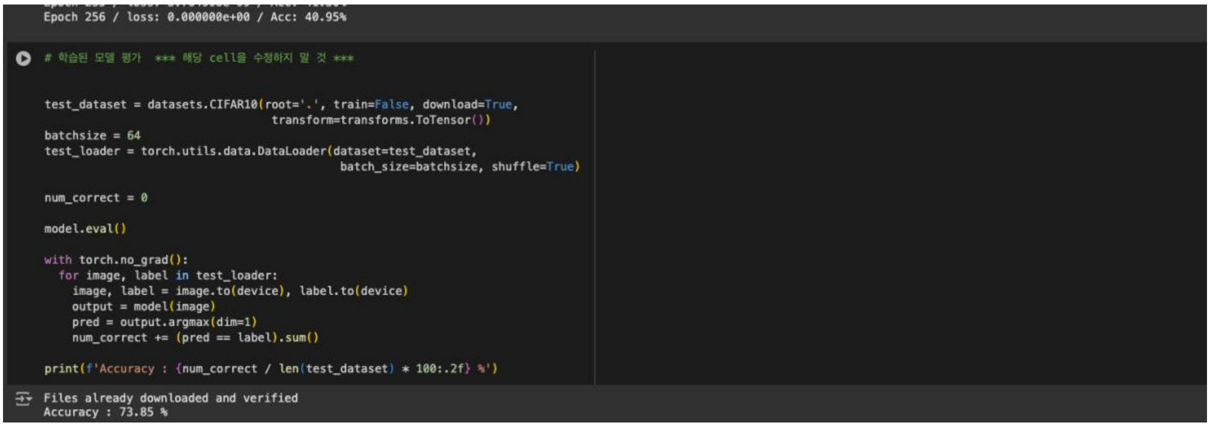


```
Epoch 100 / loss: 0.000000e+00 / Acc: 44.65%
```

```
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                            batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 67.00 %
```

Lr=1e-4, epoch=256



```
Epoch 256 / loss: 0.000000e+00 / Acc: 40.95%
```

```
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR10(root='.', train=False, download=True,
                                transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

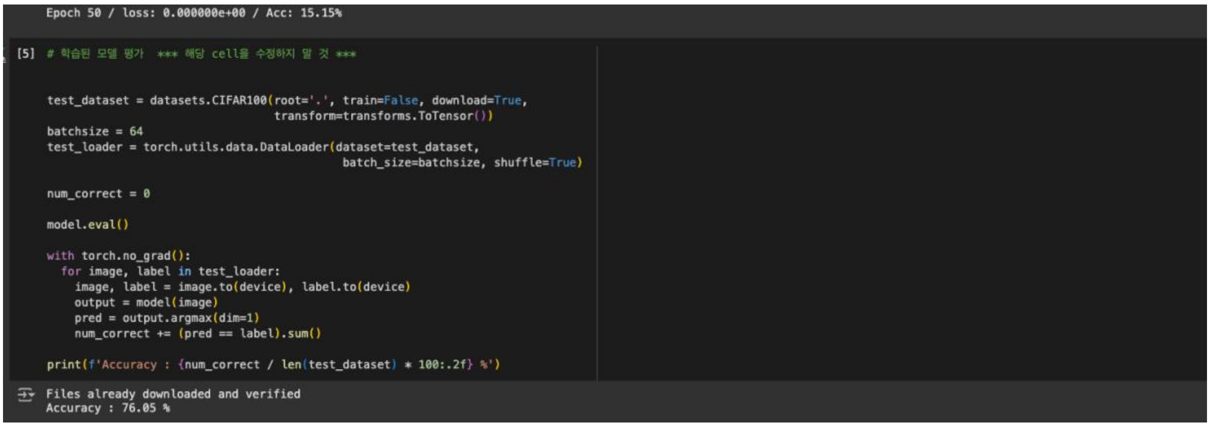
```
Files already downloaded and verified
Accuracy : 73.85 %
```

1-B) 모델의 큰 변경 없이 Learning-rate와 Epoch를 변경한 것 만으로도 이러한 변화가 나타나는 것이 흥미롭다.

1-C) 현재 모델은 2의 승수 단위로 Linear 모델을 축소해 나가는 모델임이 특징이다. 이 모델은 LR-Scheduler를 사용함으로 학습하며 LR이 천천히 줄어드는 것이 특징이다.
1-D) 별도 첨부.

2-A)

lr=1e-4, epoch=50, ResBlock=3L

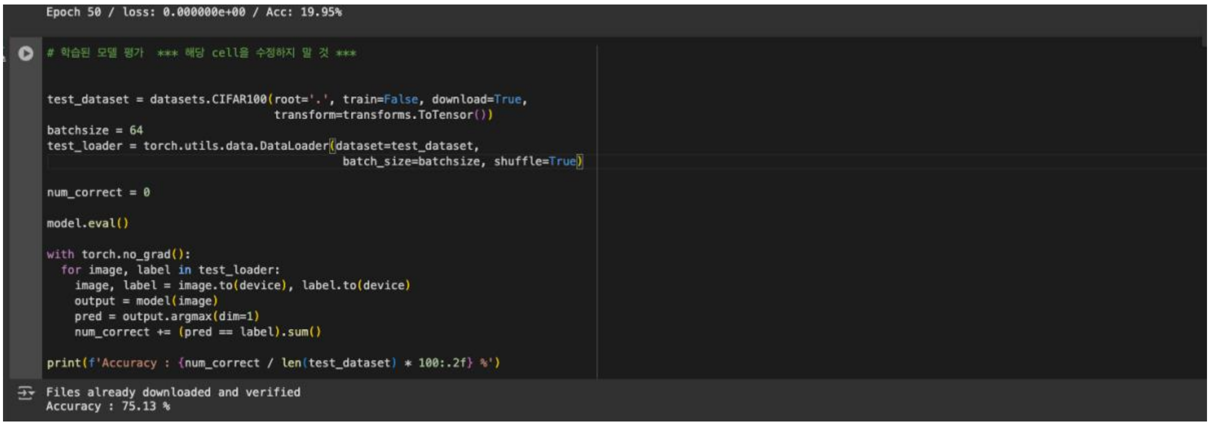


```
Epoch 50 / loss: 0.000000e+00 / Acc: 15.15%
```

```
[5] # 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


    test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                     transform=transforms.ToTensor())
    batchsize = 64
    test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                              batch_size=batchsize, shuffle=True)

    num_correct = 0

    model.eval()

    with torch.no_grad():
      for image, label in test_loader:
        image, label = image.to(device), label.to(device)
        output = model(image)
        pred = output.argmax(dim=1)
        num_correct += (pred == label).sum()

    print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 76.05 %
```

Lr=1e-3, epoch=50, ResBlock=3L

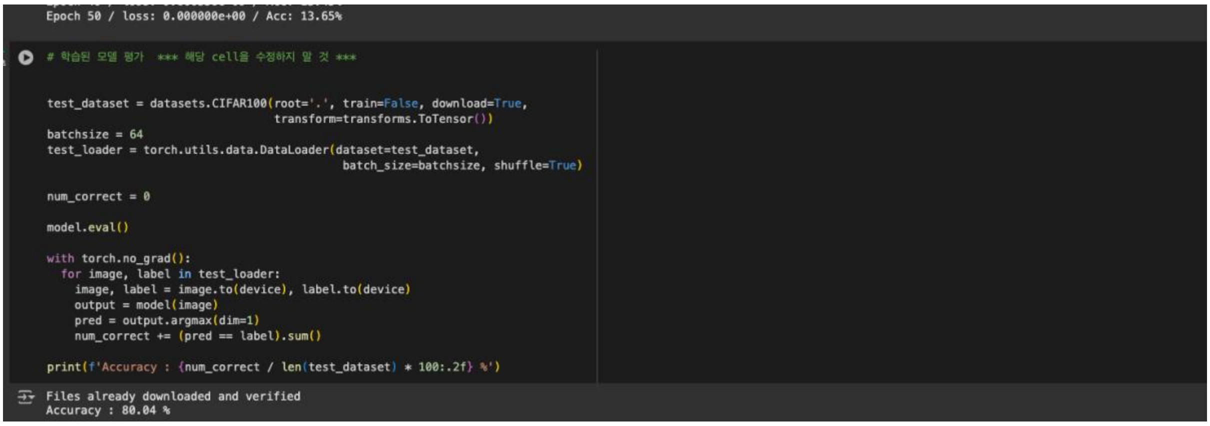


```
Epoch 50 / loss: 0.000000e+00 / Acc: 19.95%
```

```
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

```
Files already downloaded and verified
Accuracy : 75.13 %
```

Lr=1e-5, epoch=50, ResBlock=3L

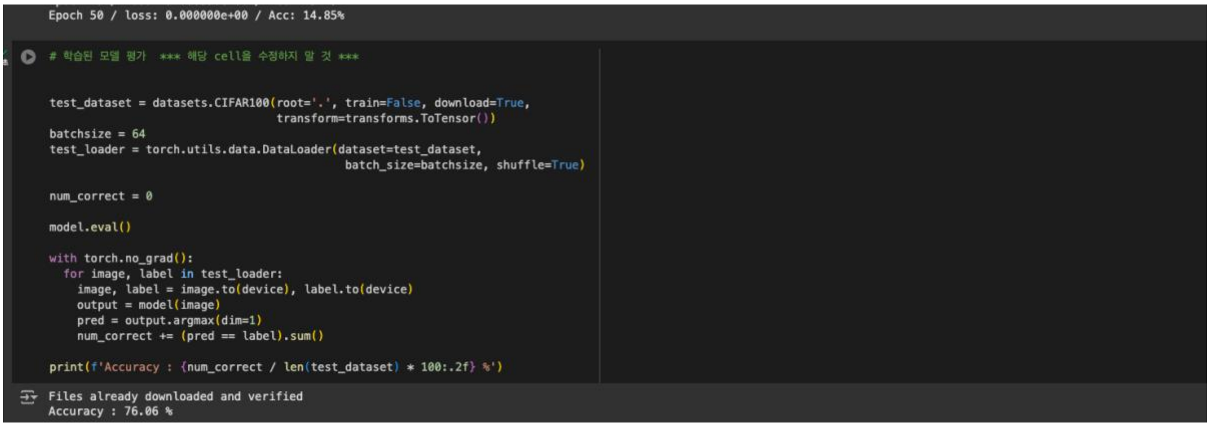


```
Epoch 50 / loss: 0.000000e+00 / Acc: 13.65%
```

```
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

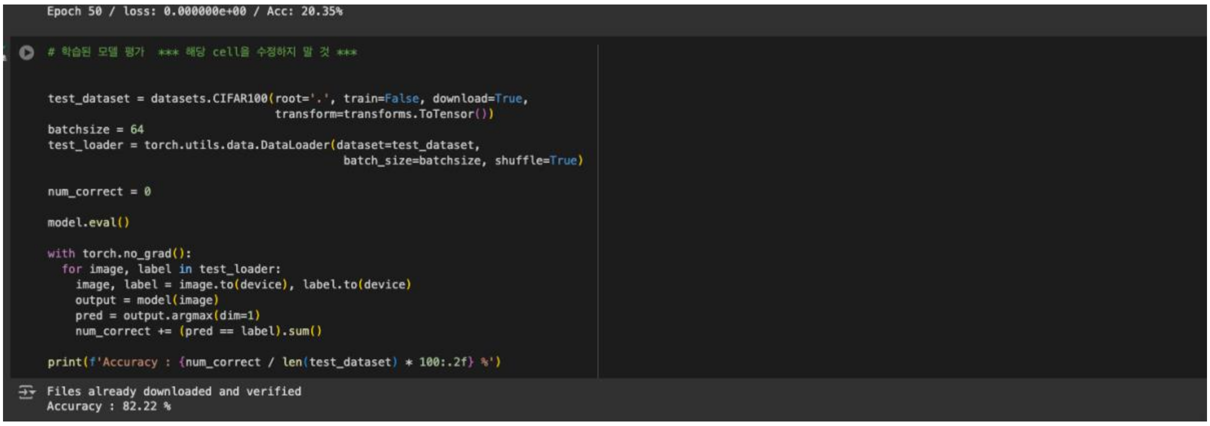
```
Files already downloaded and verified
Accuracy : 80.04 %
```

Lr=1e-4, epoch=50, ResBlock=5L



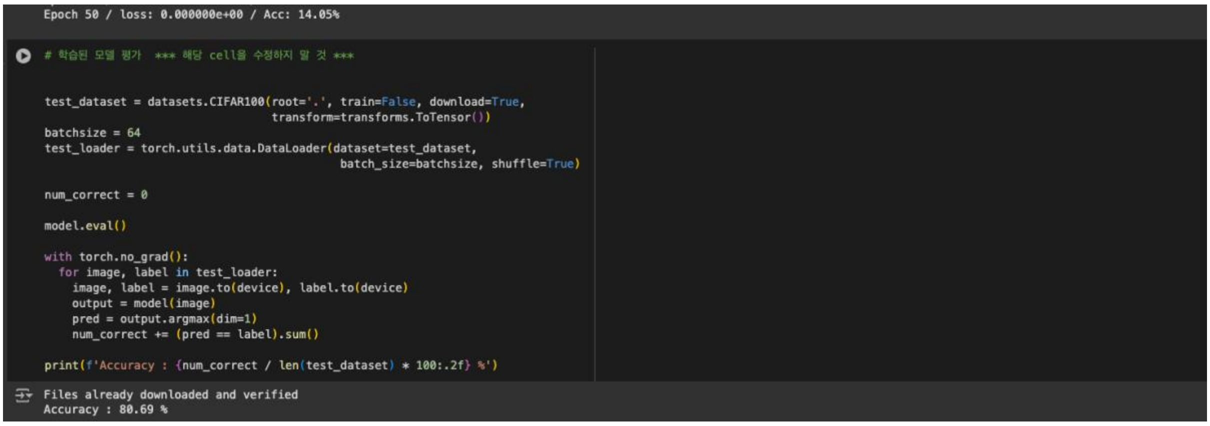
```
Epoch 50 / loss: 0.000000e+00 / Acc: 14.85%

# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')

Files already downloaded and verified
Accuracy : 76.06 %
```

lr=1e-3, epoch=50, ResBlock=5L



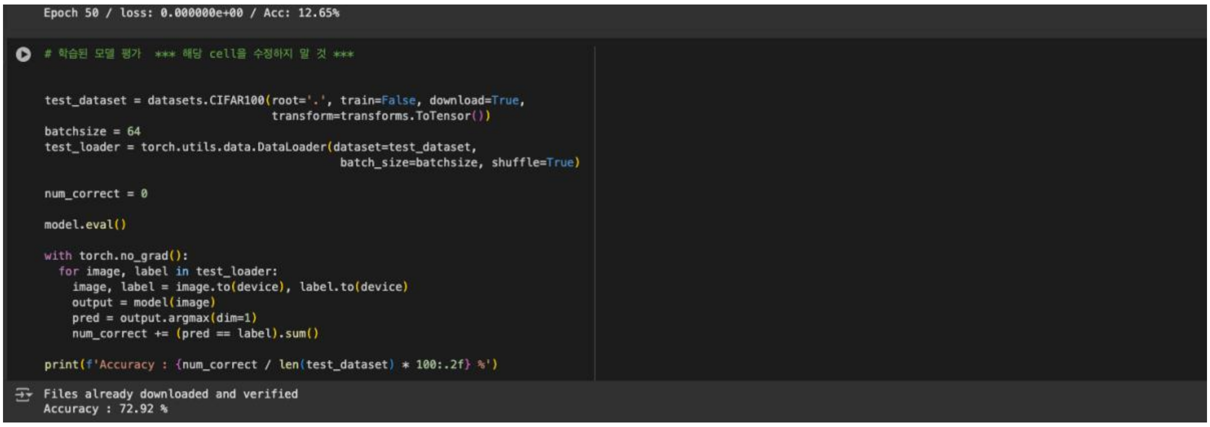
```
Epoch 50 / loss: 0.000000e+00 / Acc: 20.35%

# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')

Files already downloaded and verified
Accuracy : 82.22 %
```

lr=1e-5, epoch=50, ResBlock=5L



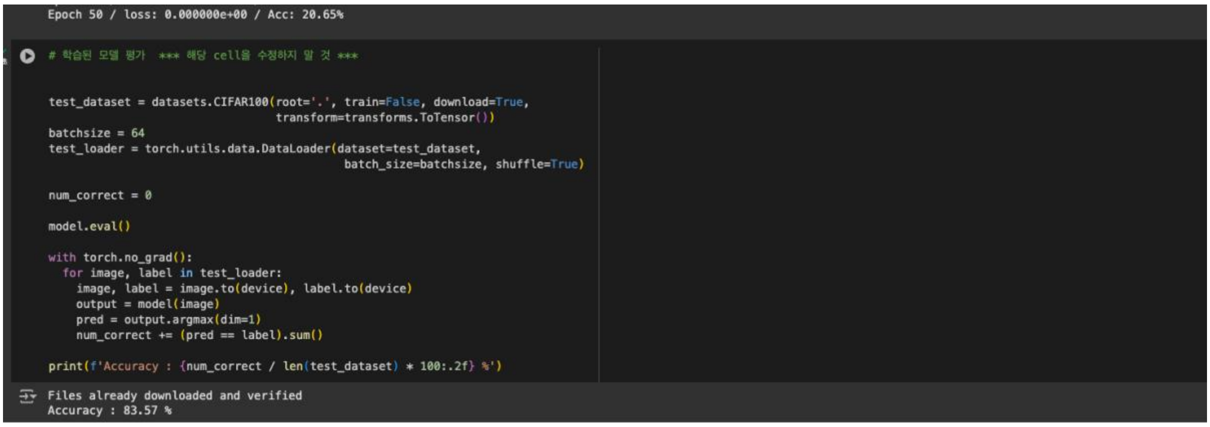
```
Epoch 50 / loss: 0.000000e+00 / Acc: 14.05%

# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')

Files already downloaded and verified
Accuracy : 80.69 %
```

Lr=1e-4, epoch=50, ResBlock=7L



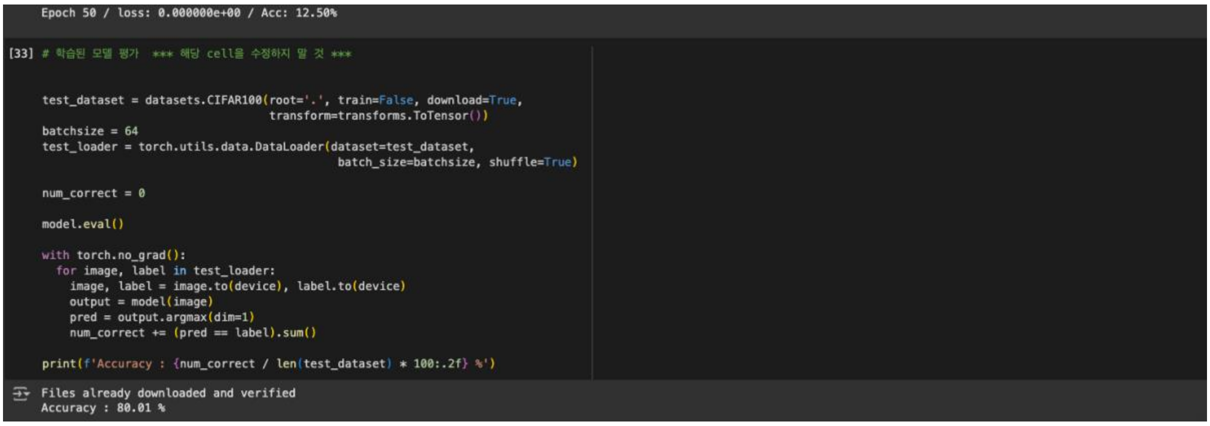
```
Epoch 50 / loss: 0.000000e+00 / Acc: 12.65%

# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')

Files already downloaded and verified
Accuracy : 72.92 %
```

lr=1e-3, epoch=50, ResBlock=7L



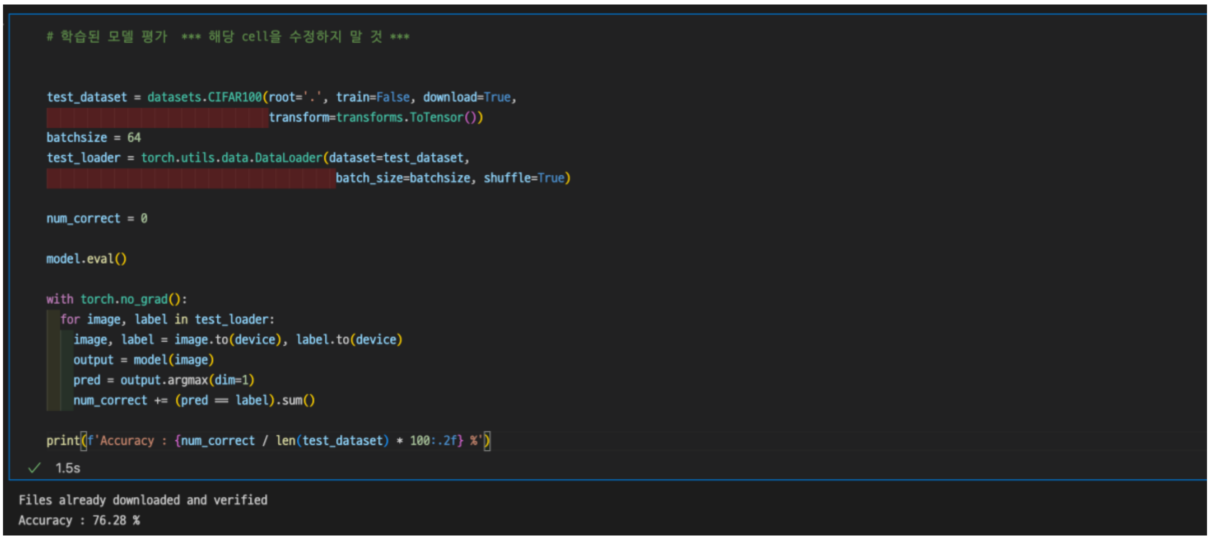
```
Epoch 50 / loss: 0.000000e+00 / Acc: 20.65%

# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')

Files already downloaded and verified
Accuracy : 83.57 %
```

lr=1e-5, epoch=50, ResBlock=7L



```
Epoch 50 / loss: 0.000000e+00 / Acc: 12.50%

[33] # 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')

Files already downloaded and verified
Accuracy : 80.01 %
```

lr=1e-3, epoch=256, ResBlock=7L



```
# 학습된 모델 평가  *** 해당 cell을 수정하지 말 것 ***


test_dataset = datasets.CIFAR100(root='.', train=False, download=True,
                                 transform=transforms.ToTensor())
batchsize = 64
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batchsize, shuffle=True)

num_correct = 0

model.eval()

with torch.no_grad():
  for image, label in test_loader:
    image, label = image.to(device), label.to(device)
    output = model(image)
    pred = output.argmax(dim=1)
    num_correct += (pred == label).sum()

print(f'Accuracy : {num_correct / len(test_dataset) * 100:.2f} %')
```

✓ 1.5s

```
Files already downloaded and verified
Accuracy : 76.28 %
```

2-B) Convolution Network가 얼마나 효과적인 방법인지 알 수 있었다.

2-C) Residual Block이 적용되어 있어 Back Propagation, Forward Pass 등에서 이득을 볼 수 있는 구조이다.

2-D) 별도 첨부.