**심층학습 중간과제 CIFAR10 리포트 – 휴먼지능정보공학과 201810787 유지우**

**최종 정확도: 75.41999816894531 (Try1, Try2 중 Try2 선택)**

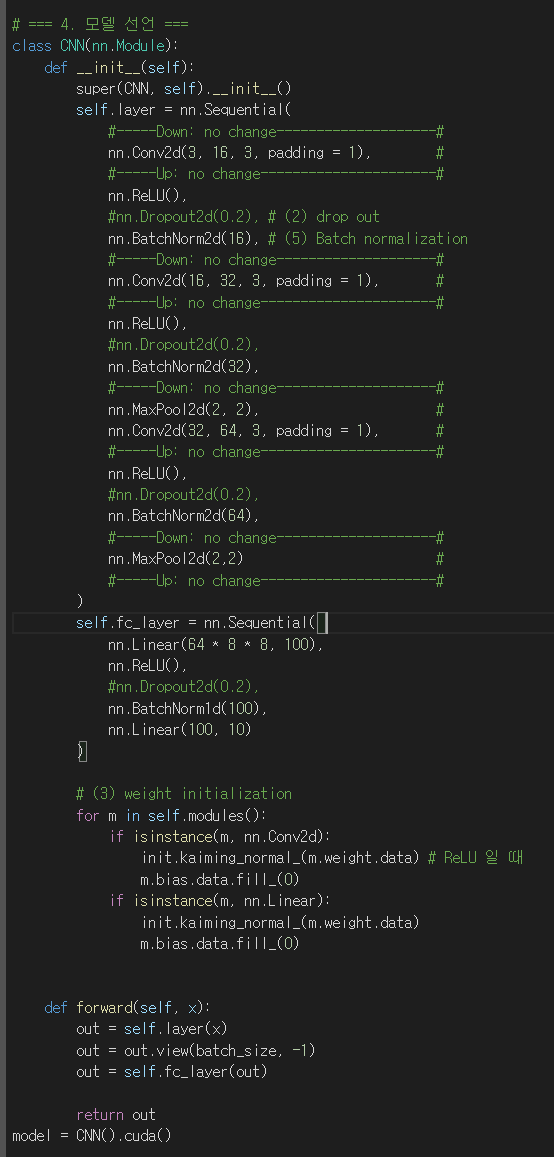
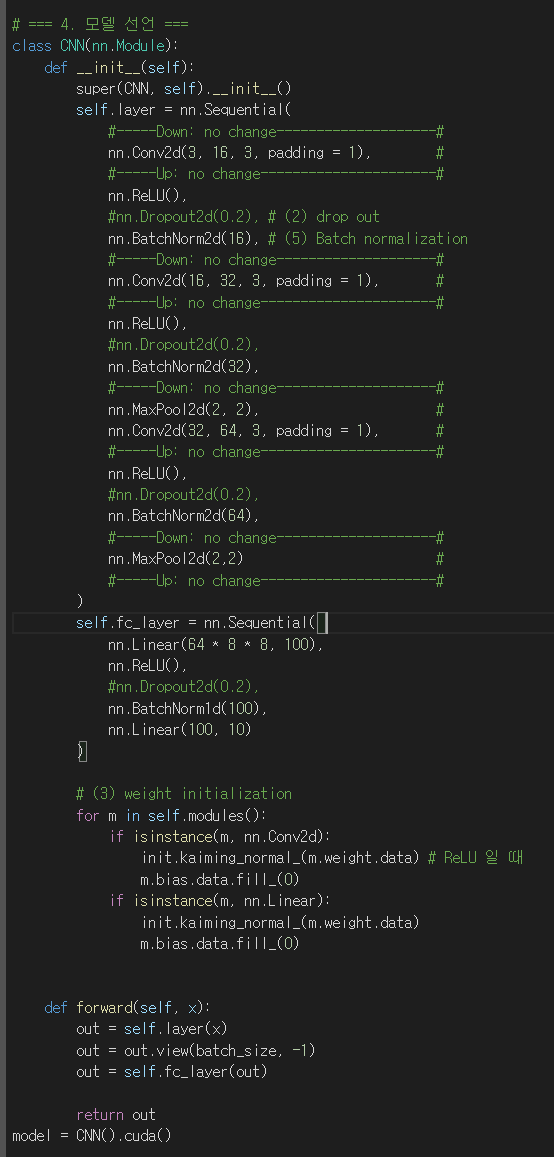
**<실습04에서 사용한 방법들의 각각 정확도 측정 tries>**

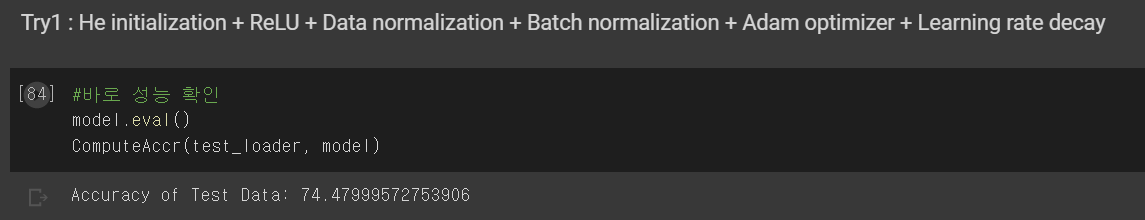
|  |  |  |
| --- | --- | --- |
| **Drop out: 약 13** | **Data augmentation: 약 13** | **Weight initialization: 약 44** |
| **Data normalization: 약 50** | **Batch normalization: 약 63** | **Adam optimizer: 약 56** |
| **Learning rate decay: 약 65** |  |  |

**<위 성능과 강의자료를 참고하여 CIFAR10 정확도 개선 방법 조합>**

**Try 1: He initialization + ReLU + Data normalization + Batch normalization + Adam optimizer + Learning rate decay (+epoch 60)**

* Epoch=60, learning rate=0.002, batch size=16

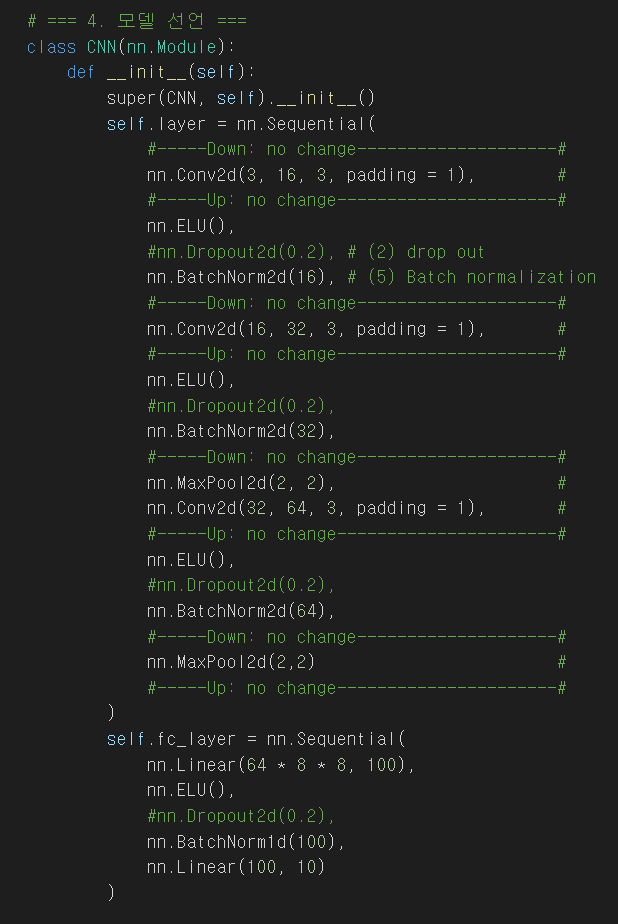




Try 1의 성능: 약 74

**Try 2: He initialization + ELU + Data normalization + Batch normalization + Adam optimizer + Learning rate decay (+epoch 100)**

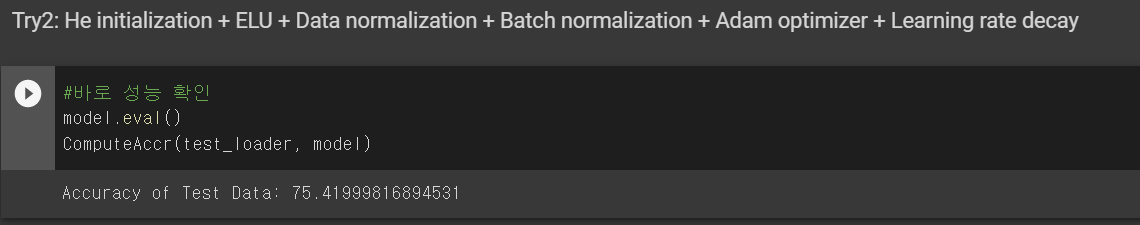
* Epoch=100, learning rate=0.002, batch size=16



Try1에서 바뀐 점

: ReLU -> ELU

Epoch 60-> Epoch 100



Try 2의 성능: 약 75

**결론: 비교적 성능이 더 높은 Try2 조합 선택**

**<warnings 유발하는 부분 수정>**

transforms.Scale -> transforms.Resize

kaimig\_normal -> kaiming\_normal\_

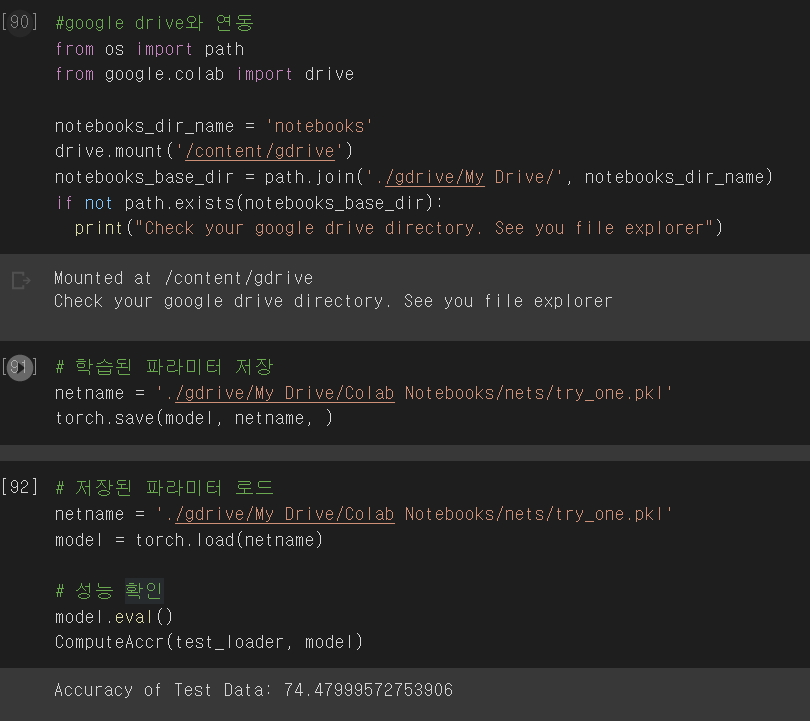
img = Variable(imgs, volatile=True) -> with torch.no\_grad(): img = Variable(imgs)

**<기타>**

**-헤매었던 부분**

**: google colab에서 pickle파일을 저장하는 방법을 찾는 데에만 시간을 많이 썼다.**

**강의자료의 code는 AWS환경에서만 사용이 가능했고, google colab은 어떠한 조치를 조금 더 취해주어야 한다는 것을 알게 되었다.**



따라서 torch.save와 torch.load 전에 google drive와 연동하는 코드를 추가했다.

실습을 무사히 마치는 것까지는 성공하였으나, 채점하실 때 저것을 어떤 방식으로 처리하실 지 조금 걱정이 된다.

**+ 화면에 웰시코기를 데려오는 기능을 사용하여 실습 내내 마음이 편할 수 있었다.**

