**HW5 구현 설명서**

201820934 이준희

실행은 Anaconda Powershell에서 과제2에서 진행한 것과 같이 .\실행프로그램이름 실행파일이름 argument1을 입력하면 됩니다. (ex. .\python.exe main.py “zero.png”)  
그리고 각각의 숫자에 해당하는 그림파일의 이름을 다음과 같이 설정합니다. (ex. 0 = “zero.png”, 1 = “one.png”)  
해당파일들은 코드 맨 앞에서 save함수를 이용해 0~9까지 일괄적으로 생산합니다. 그래서 우리는 예를 들어 6의 그림을 담고있는 “six.png”라는 파일이 실행 전에 없어도 코드를 실행하면 생성이 되기때문에 계속 수행할 수 있습니다. 즉, 코드 수행시에 숫자 사진 파일이 기존에 따로 있을 필요없이 바로 생산합니다. 이후엔 이 파일을 가지고 분석에 들어갑니다. 물론 따로 가지고 있는 숫자파일을 가지고 수행해도 가능합니다.

우선 간단하게 각각의 부분의 역할에 대해서 설명하겠습니다. 구현적인 부분은 참고문헌의 코드, 구글링 그리고 교수님께서 올려주신 강의노트를 참고하였습니다.

-Transform은 그림형태로 분석을 할 수 없으니 분석이 가능한 행렬(tensor)로 바꾸고, 정규화를 하는 역할을 합니다.

-DataLoader를 이용하면 학습을 위한 방대한 데이터를 미니배치 단위로 처리할 수 있고, 데이터를 무작위로 섞음으로써 학습의 효율성을 향상시킬 수 있습니다.

-학습과정: 총 938개의 mini batches를 학습하는데 100개단위로 진행도를 표현하였습니다. 실제 학습도중의 상태를 보면 처음엔 오차가 있지만 점차 그 오차가 줄어들어 듭니다.

-학습이 완료된 이후에는 맨 위에서 생성한 숫자 사진 파일을 가지고 transform을 거칩니다. 그렇게 생산한 텐서를 차원에 맞게 변경한 뒤 위의 train\_data와 동일하게 CNN작업을 진행하면 예측값이 나오게 됩니다. 학습을 한 뒤이므로 정확도가 높은 것을 알 수 있다.

다음사진은 실행 예시입니다.

