**<프로젝트>**

편향은 b로 설정하고 뉴런이 얼마나 "쉽게 활성화" 되느냐를 제어함.

반면 w1,w2는 가중치로 각 신호에 얼마나 많은 "영향"을 주느냐

활성화 함수: 입력신호의 총합이 활성화를 일으키는지 정하는 역할

a는 입력신호의 총합 h()는 활성화 함수, y는 출력

활성화 함수는 계단 함수로도 불리는데 "시그모이드 함수" 사용

기계학습이란? 데이터에서 답을 찾고 패턴을 발견하여 데이터로 이야기를 만드는 것!!!

특징을 추출하고 그 특징의 패턴을 기계학습 기술로 학습하는 방법이 있다. 여기에서 말하는 특징이란

입력 데이터(폰트 글씨들.bmp)에서 본질적인 데이터(픽셀의 검은색과 하얀색)을 정확하게 추출할 수 있도록 설계된 변환기

훈련 데이터와 시험 데이터를 나누는 이유느?

범용능력에 있어서 인데 왜??(아직 보지 못 하였던 문제를 올바르게 찾아야함.)

그렇다면 폰트를 컴퓨터가 인식을 해야하는데 아직 한번도 보지 못하였으니깐 데이터에서 훈련 데이터 따로 테스트 데이터로 나눠야한다는 소리

# print("훈련 데이터 개수:", len(train\_data))  # 1110여야 함

# print("훈련 레이블 개수:", len(train\_labels))  # 1110여야 함

# print("테스트 데이터 개수:", len(test\_data))  # 333여야 함

# print("테스트 레이블 개수:", len(test\_labels))  # 333여야 함

# # 첫 번째 데이터와 레이블의 크기 확인

# print("첫 번째 훈련 데이터의 크기:", len(train\_data[0]))  # 4096이어야 함

# print("첫 번째 훈련 레이블의 구조:", train\_labels[0])  # 원-핫 인코딩된 배열이어야 함

# # 총 픽셀 데이터 수 계산

# total\_train\_data\_points = len(train\_data) \* len(train\_data[0])

# total\_test\_data\_points = len(test\_data) \* len(test\_data[0])

# print("총 훈련 데이터 픽셀 수:", total\_train\_data\_points)  # 예상: 4096 \* 1110

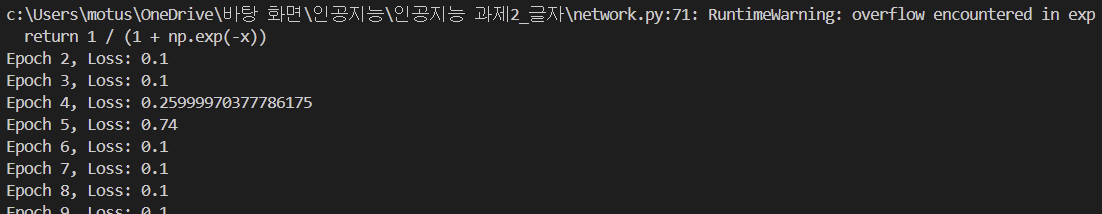
# print("총 테스트 데이터 픽셀 수:", total\_test\_data\_points)  # 예상: 4096 \* 333

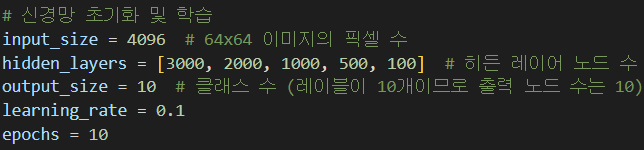
11.03

1.학습데이터 셋 만들기(굴림,새굴림,휴먼아미체,휴먼엑스포체,한양해서,HY견고딕,HY견명조,HY그래픽,HY울릉도B)

2. PIL 모듈 이용하여 한글 폰트의 경로에서 해당 문장들의 문자열에서 각각의 문자들을 추출하여 각각의 폰트로 변경 시키고 train 디렉토리에 각각의 단어들로 디렉토리 생성 함.

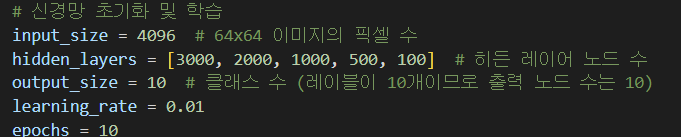
3. 테스트데이터 셋 만들기(안상수2006가는보통,휴먼옛체,휴먼편지체)

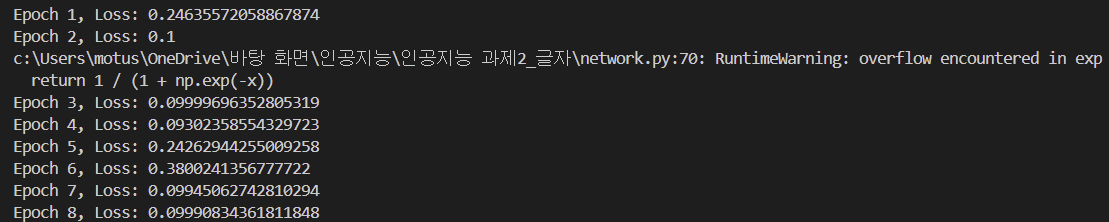




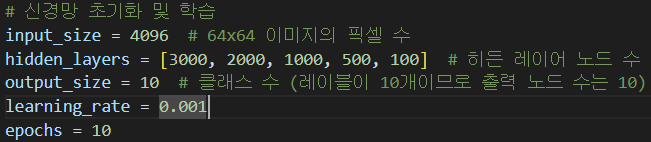
학습률 0.1로 했을때의 결과값

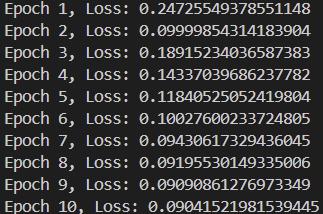
학습률 0.01로 했을때의 결과값





학습률 0.001로 했을 때의 결과값





실험 결과

학습률은 신경망의 가중치 업데이트 크기를 조절하는데 학습률이 너무 크게 되면 가중치가 큰 폭으로 변하면서 시그모이드 함수의 입력 값이 너무 커져 오버플로우 문제가 생길 수도 있다.

또한 시그모이드 함수는 입력 값이 매우 클 경우 1로 가까워지게된다. 출력이 거의 변하지 않으며 그 도함수도 0에 가까워지게 된다.

또한 입력값이 매우 큰 양수이게 되면 미분 계산중 오버 플로우가 발생 할 수도 있다. 학습률이 큰 경우, 가중치가 급격히 커지면서 시그모이드 함수의 입력값도 커져 오버플로우가 발생할 가능성이 높아진다

결론적으로 학습률이 크게되면 시그모이드 입력이 극단적으로 커져 오버플로우가 발생하고, 학습률이 작으면 가중치가 점진적으로 변화하여 시그모이드 함수의 입력이 극단적으로 커지지 않는다.

<학습률 감소 기법 채택>