이해동
정보과학과
202584-010115

머신러닝특론 과제4 리포트

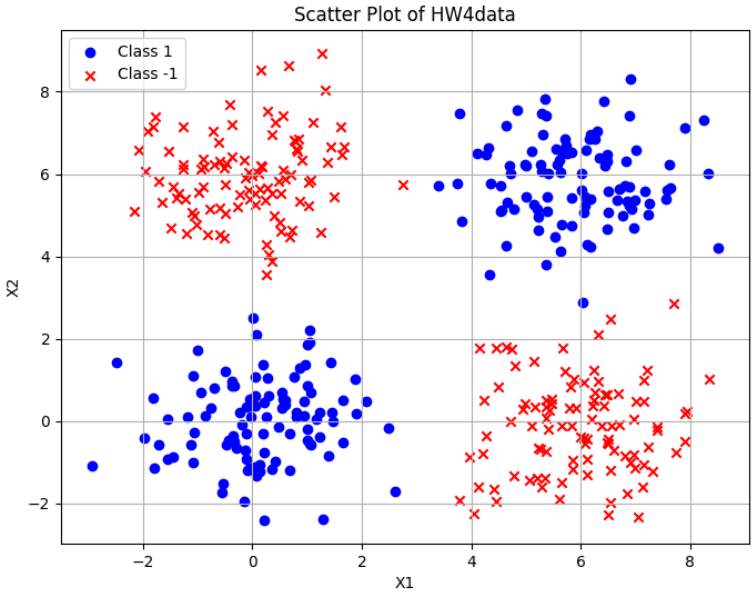
서론

본 보고서는『머신러닝특론』과목의 네 번째 과제로 주어진 다층 퍼셉트론(Multilayer Perceptron, MLP) 구현 과제를 수행한 결과를 정리한 것이다. 본 과제의 목적은 신경망의 기본 구조와 학습 과정을 직접 구현하고 실험함으로써, 이론적 이해를 실제 데이터 분석에 적용하는 능력을 향상시키는 데 있다. 주어진 데이터셋 HW4data는 2차원 평면상의 400개 샘플과 각각의 클래스 레이블로 구성되어 있으며, 이를 바탕으로 신경망의 구조 설계, 학습, 시각화 및 성능 평가 등의 다양한 실험을 수행하였다.

문제1-(1) HW4data 의 산점도 출력

주어진 데이터 HW4data 는 2차원의 샘플데이터 400 개와 1차원의 클래스 레이블 데이터로 이루어져 있다. 클래스는 1, -1 로 분류되며, 그 결과는 아래 그림과 같다.

산점도의 출력은 python 의 matplotlib 으로 수행 하였다.



<그림 1>

Source code : MPL_1.py

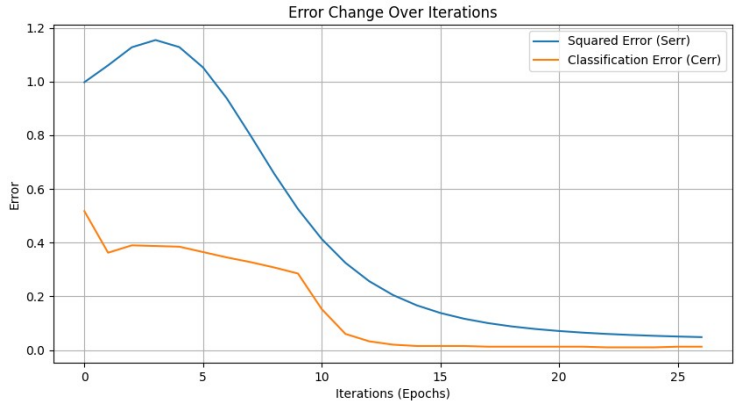
문제1-(2) 다층 퍼셉트론을 구성하여 학습 및 오차변화의 그래프 출력

다층 퍼셉트론 설정

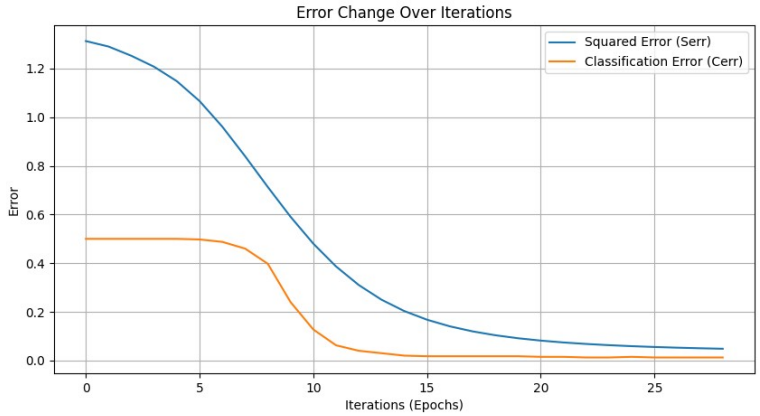
- 입력 뉴런 : 2개, 은닉 뉴런 : 5개, 출력 뉴련 : 1개

주어진 다층 퍼셉트론 설정을 바탕을 학습을 진행하면, <그림 2>, <그림 3> 과 같은 그래프를 얻을 수 있다. 매 실행 시 마다 조금씩 다른 그래프를 얻게 되는데, 그 이유는 가중치의 초기값을 무작위하게 설정하기 때문으로 판단된다.

학습 알고리즘은 참고교제 프로그램 11-1 에서 제공되는 matlab 코드를 python 으로 변환하여 사용 하였다. 전체 코드의 주석과 코드 형태를 최대한 원본 프로그램과 유사하게 유지 하였다.



<그림 2>



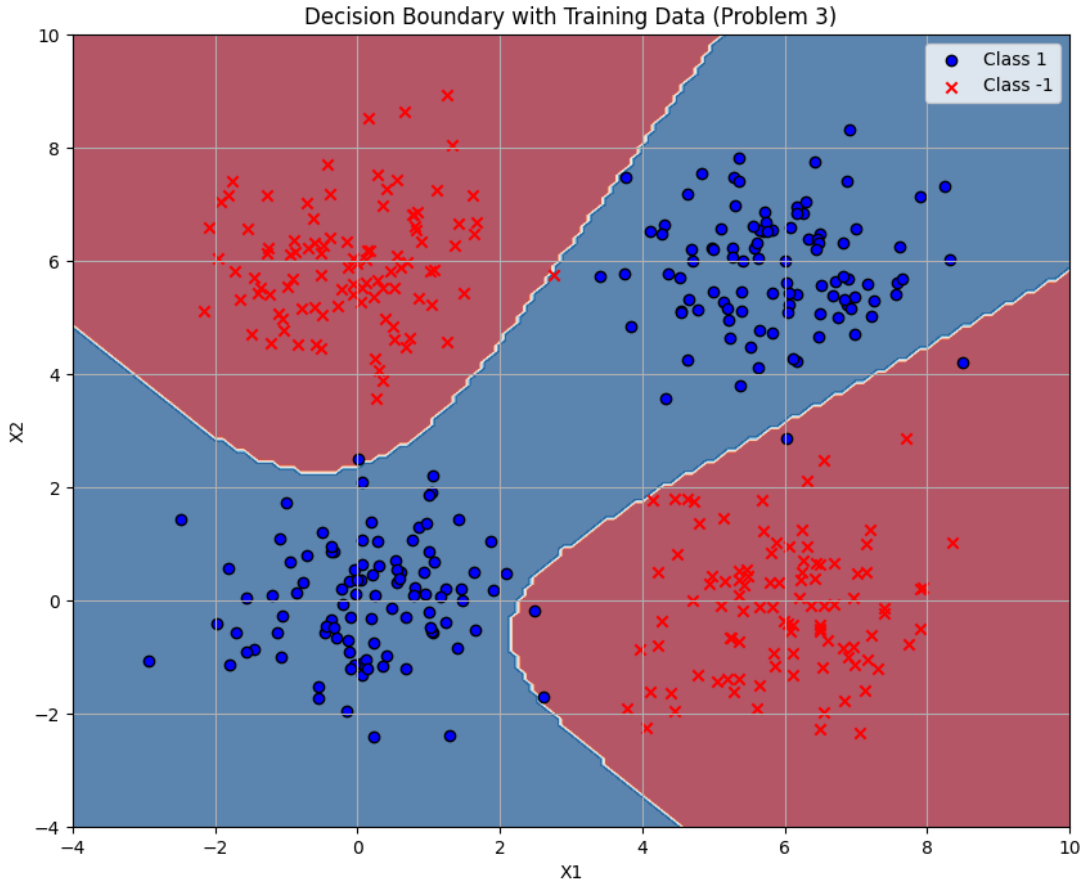
<그림 3>

Source code : MPL_2.py

문제1-(3) 앞서 문제 1-(2) 에서 학습된 신경망의 결정 경계 출력

Python 의 numpy 의 meshgrid() 함수를 통하여 결정 경계를 출력하도록 하였다.

Meshgrid 는 영역을 작은 격자로 나누어서 해당 구역이 분류기에서 어느 영역에 속하는지 결정하도록 한다. 앞서 학습한 신경망에 격자 구역의 좌표를 넣어서 계산하면 학습된 결과에 따라서 <그림 4> 와 같은 결정 경계의 그래프를 얻을 수 있다.



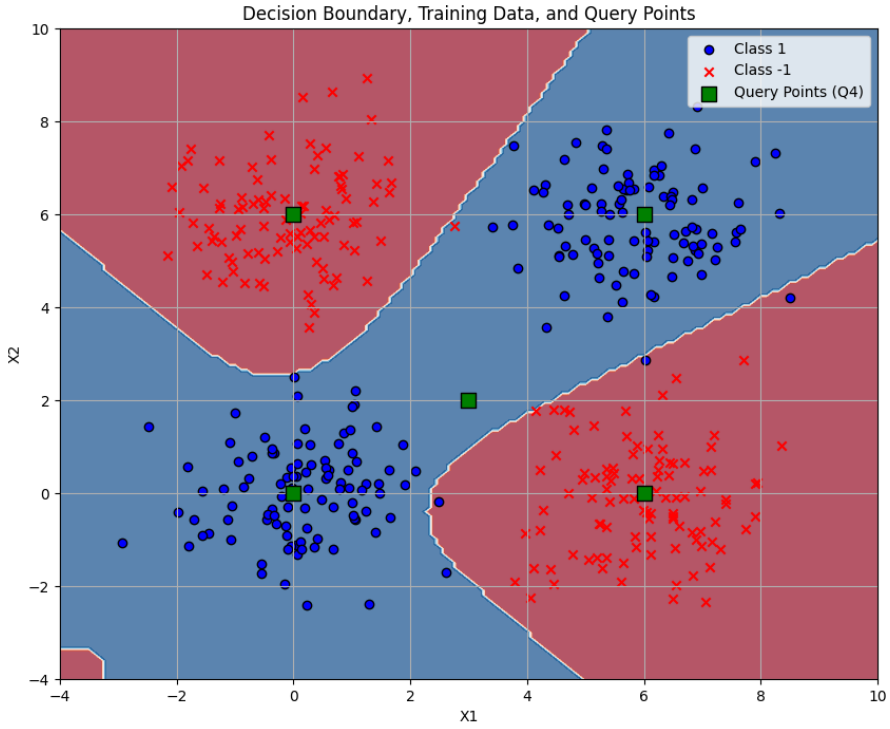
<그림 4>

Source code : MPL_3.py

문제1-(4) 입력 데이터의 예측값 출력

아래에 제시된 입력 데이터를 화면에 출력하면 <그림 5> 에서 표시된 녹색점과 같다. 해당 입력 데이터를 앞서 훈련된 뉴런에 넣어서 계산을 수행하면 각 점이 어떤 클래스에 가까운지 예측값을 얻을 수 있다. 그 결과는 <그림 6> 과 같다.

데이터 1: 입력 (0, 0) 데이터 2: 입력 (6, 6)
데이터 3: 입력 (0, 6) 데이터 4: 입력 (6, 0)
데이터 5: 입력 (3, 2)



<그림 5>

```
문제 (4): 추가 데이터 포인트에 대한 신경망 출력값 계산
데이터 1: 입력 (0.0, 0.0), 출력 뉴런 값: 0.9770
데이터 2: 입력 (6.0, 6.0), 출력 뉴런 값: 0.8628
데이터 3: 입력 (0.0, 6.0), 출력 뉴런 값: -0.8997
데이터 4: 입력 (6.0, 0.0), 출력 뉴런 값: -0.8945
데이터 5: 입력 (3.0, 2.0), 출력 뉴런 값: 0.6730
```

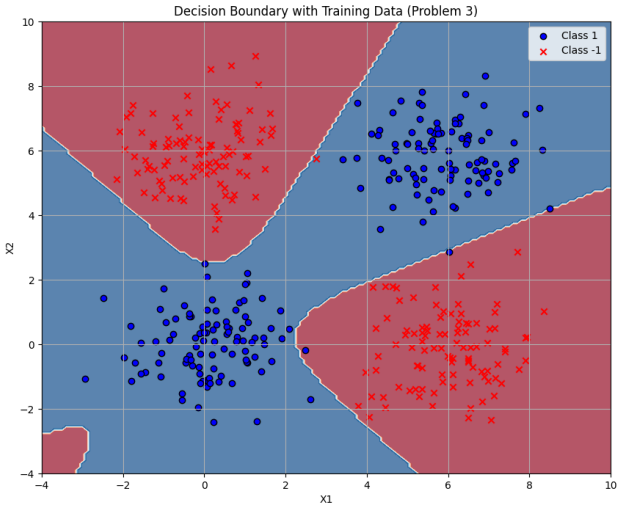
< 그림 6>

Source code : MPL_4.py

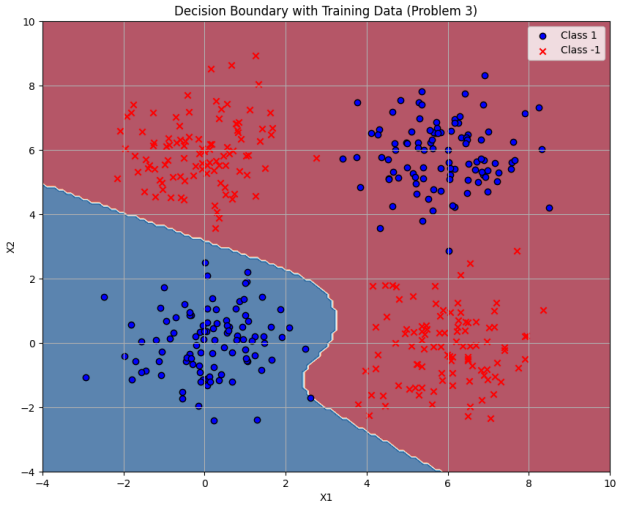
문제1-(5) 은닉 뉴런이 3개인 경우, 학습의 양상 보고

은닉 뉴런이 5개에서 3개로 줄어들 경우 뉴런이 표현할 수 있는 표현력이 줄어들기 때문에 비선형적인 결정 경계를 이루는 데이터 집합에서 학습이 실패할 가능성이 늘어나게 된다. 앞서 5개의 뉴런으로 테스트 했을 때는 10번 이상의 시도에서 학습이 실패하는 경우가 나타나지 않았지만, 은닉 뉴런이 3개로 줄어들 경우 10번의 시도 이내에 학습이 실패하는 경우를 볼 수 있었다.

학습이 성공하는 경우는 <그림 7> 과 같은 결정경계 그래프를 얻을 수 있었으며, 학습이 실패하는 경우는 <그림 8> 과 같은 결정경계 그래프를 얻을 수 있었다.



< 그림 7>



< 그림 8>

Source code : MPL_5.py