인공지능 코딩 과제 3

2021920035 신지호

1. **과제의 흐름**

이번 과제는 지난 과제에서 구현하지 못하였던 XOR을 포함하여 AND, OR, XOR, 도넛 모양의 데이터를 분류하도록 학습하는 다중 퍼셉트론을 구현하는 것이다. 단일 퍼셉트론은 하나의 직선을 표현하기 때문에 XOR과 도넛 모양은 분류할 수 없다. 그러나, 다중 퍼셉트론은 여러 직선을 합쳐 어떤 함수 모양도 구현할 수 있다는 장점이 있다. 본 과제에서는 도넛 모양의 데이터를 분류하기 위해 최소 3개의 직선이 필요하나, 다중 레이어를 활용하기 위해 2개의 은닉층과 각 층마다 2개의 퍼셉트론을 배치하여 구현하였다.

라인, 옷걸이이(가) 표시된 사진

자동 생성된 설명

그림 1 본 과제에 적용한 다중 퍼셉트론의 모습

그림 1과 같이 D1, D2 2개의 은닉층으로 각 Layer에 P1과 P2 두 개의 퍼셉트론을 배치하였다. 은닉층의 개수와 퍼셉트론의 개수를 변수로 받도록 프로그램을 구현하지 못하였지만, 다중 퍼셉트론의 개념을 최대한 활용한 2X2 모델을 구현하였다.

과제의 흐름는 다음과 같다.

1. TrainData X,Y를 입력받는다.
2. 퍼셉트론의 weight를 표준정규분포를 따르는 랜덤값으로 초기화한다.
3. D1 레이어부터 차례로 퍼셉트론의 출력값을 계산한다.
4. 마지막 Y(F\_node) 퍼셉트론부터 역전파를 이용해 weight를 업데이트 한다.
5. 한 학습 데이터 셋이 끝나면 3), 4)를 다시 반복하여 에러율이 0%가 될때까지 혹은 정해진 횟수까지 반복한다.
6. 정해진 횟수까지 에러율이 0%가 되지 못하였다면 2)으로 돌아가 weight를 다시 랜덤한 값으로 초기화하고 과정을 반복한다.
7. **논의 할 점**

* Activation function의 선택

본 과제에서 activation funcion으로 ReLU를 사용하였다. 미분값이 0 혹은 1로 미분을 계산하기 쉽기 때문에 sigmoid 함수 대신 사용하였다. 그러나, 모든 퍼셉트론을 ReLU로 사용할 경우 결과가 제대로 나오지 않았다. 마지막 퍼셉트론이 최종 y 값을 계산하는데 학습 데이터로 0 혹은 1의 y값만 제공하였으므로 만약 ReLU를 사용한다면 마지막 결과가 0 혹은 0 이상의 값 모두가 되므로 학습이 끝나지 않는 현상이 있다. 이를 해결하기 위해 마지막 퍼셉트론은 step 함수를 사용하여 0 혹은 1의 결과가 나오도록 하였고, 문제를 해결했다.

* Epoch 값 설정

처음 학습을 진행할 때 충분한 시간을 기다려도 학습이 되지 않는 상황이 발생하였다. 원인을 분석해보니, 역전파 과정에서 미분값이 너무 작아지거나 커지면 weight에 아무것도 더해지지 않거나 값이 계속해서 커지는 현상이 나타났다. 이런 Vanishing gradient problem을 해결하기 위해 Gradient Flow를 만들어주어야 하나 코드가 복잡하고, 시간이 부족한 관계로 다른 해결방법을 찾았다. Epoch 값을 설정하여 일정 횟수 학습을 반복했을 때 결과가 나타나지 않는다면, weight 값을 다시 랜덤 값으로 정하는 방식이다. 이 과정에서 현재 프로그램의 보완점으로 Gradient Flow를 만드는 것을 알았고 시간이 있다면 보완하고 싶다. 결국엔 특정 초기 weight 값에서 학습이 되지 않는 문제가 있었고 Epoch로 해결할 수 있었다.

1. **결과화면**

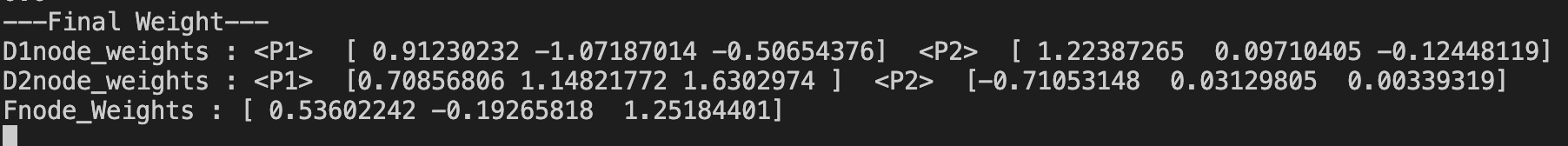
본 소스코드는 파이썬 3.11.4 환경에서 실행하였고, 에러율이 0%가 될 때까지 계속 반복하므로 시간이 조금 소요될 수 있다. AND, OR은 빠르게 결과가 출력되나 XOR, 도넛 모양 데이터는 약 30초에서 1분정도 소요될 수 있다. Coding3.py에서 학습 데이터를 입력하고 실행하면 된다. Node.py는 퍼셉트론을 클래스로 구현한 파일이다.

-AND GATE

스크린샷, 소프트웨어, 멀티미디어 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

왼쪽은 에러율 그래프로 처음에는 높은 에러율에서 점차 줄어들다가 0%로 끝났다. 가운대는 0~1까지 x와 y를 0.05 간격으로 결과값 y를 구한 그래프이다. 빨간색은 0, 파란색은 1의 결과를 나타내고 실제로 AND\_GATE의 결과를 포함하도록 나왔다. 오른쪽은 D1 레이어의 노드를 직선으로 표시한 그래프이다.



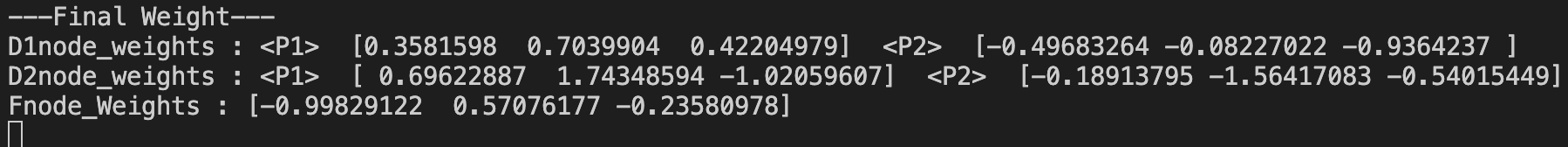
위 그래프를 출력할 때 구한 각 퍼셉트론 별 weight이다.

-OR\_GATE

소프트웨어, 멀티미디어 소프트웨어, 그래픽 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

다음은 OR\_GATE의 입력값으로 구한 그래프이다. 4번의 학습만으로 결과를 얻었으며, 가운대 그래프를 보면 OR\_GATE의 결과를 포함하는 결과이다.



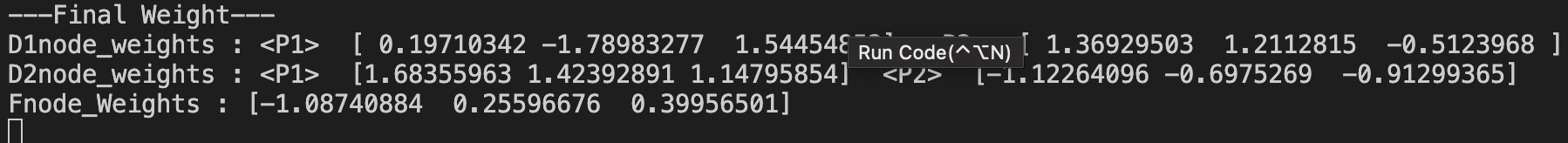
Weight 값은 위 그림과 같다.

-XOR\_GATE

소프트웨어, 멀티미디어 소프트웨어, 그래픽 소프트웨어, 스크린샷이(가) 표시된 사진

자동 생성된 설명

에러율이 적절하게 학습함에 따라 감소하였고 XOR\_GATE의 결과를 포함하는 그래프가 표시되었다.



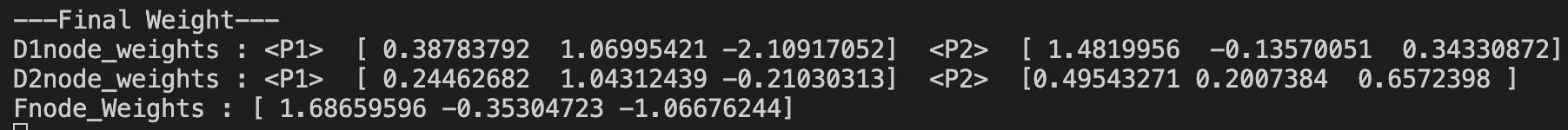
XOR\_GATE를 만족하는 학습된 weight 값이다.

-도넛 모양의 데이터

스크린샷, 소프트웨어, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

도넛 모양 데이터도 결과가 잘 나왔다. 에러율이 갈수록 작아지도록 학습이 되었고 도넛 모양 학습데이터를 포함하는 결과 그래프가 보인다.



학습된 Weight 값은 다음과 같다.