인공지능 코딩 과제2

2021920035 신지호

실제 노드를 만들어보고, 입력값에 따라 weight를 변화하면서 학습을 시켜보았다. 매 학습 반복 단계에서 weight 값을 변화시키는 단위인 learning\_rate를 고민하였는데, 0.05가 가장 적합하게 나왔다. 너무 작은 값은 그래프를 그렸을 때, 단위가 촘촘해서 잘 보이지 않고 학습 반복 횟수가 지나치게 많아지는 단점이 있다. 너무 큰 값은 weight가 매 단계마다 크게 변해 학습이 제대로 되지 않을 수 있다. 따라서 0.001 ~ 0.5 값 중 실험적으로 0.05를 얻었다. 처음 weight는 표준정규분포를 따르는 랜덤한 값으로 하였다.

코드는 입력값을 n개로 받을 수 있게 하였다. 입력값에 맞게 x와 y를 넣어주면 학습을 진행한다. 처음에 node 클래스를 생성자로 초기화하여 초기 weight 값을 정하고 learing\_rate도 정한다. 이후 node 객체의 learning 함수를 train\_data(x,y)를 인자로 주어 실행하면 학습을 1회 진행한다. 반복을 위해 while 문으로 반복하였고 마지막 계산된 error가 0이면 break한다.

실행언어는 python이고 pyplot, numpy 패키지가 필요하다. Coding2.py는 실행파일로 and\_gate(), or\_gate(), xor\_gate() 함수를 각각 실행하면 결과를 얻는다. Node.py는 node를 class로 만들어 사용한 파일이다.

And\_gate() 실행결과이다. 왼쪽 그래프는 에러률로 약 30번의 학습을 통해 0으로 만들었다. 에러률이 0이 될 때까지 반복하도록 하여 매번 반복 횟수는 달라질 수 있다. 오른쪽 그래프는 학습 후 가중치 값으로 계산한 가로축이 x1, 세로축이 x2인 일차함수 그래프이다. And gate이므로 (1,1) 파란점과 남은 빨간점으로 잘 나누어졌다.

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

자동 생성된 설명

Or\_gate() 실행결과이다. 5회 반복만에 에러률 0을 달성하였고 오른쪽 그래프와 같이 파란점과 빨간점을 잘 구분하는 일차함수가 그려졌다. Weights 값은 아래 프롬프트에 출력되었다.

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

자동 생성된 설명

Xor\_gate()의 실행결과이다. Xor은 하나의 노드로는 학습을 시키는 것이 불가능하기 때문에 10000만 반복하도록 하였다. 에러률이 0.5에서 계속 유지되는 것으로 보아 4번의 train\_data 중 2번씩 틀렸다는 의미이다. 또한 오른쪽 그래프를 보았듯이 빨간점과 파란점을 가르는 것이 불가능하다. 아래 프롬프트의 weight와 오른쪽 그래프의 일차함수는 10000번 반복 후 결과이다.

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

자동 생성된 설명