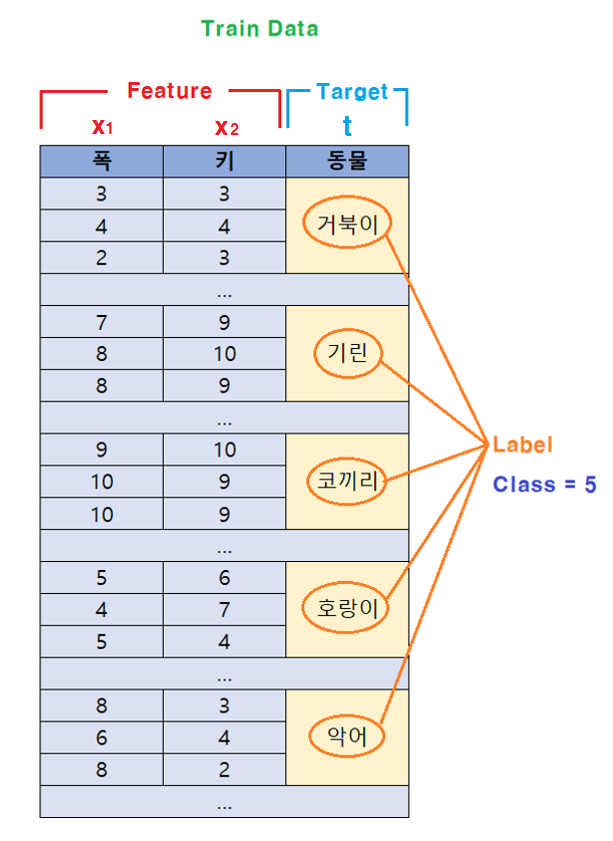
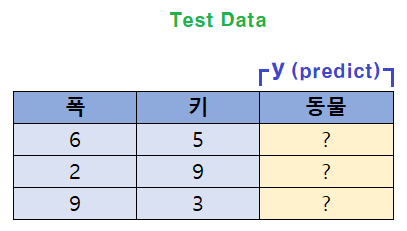
1. **문제 이해**

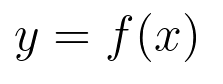
폭과 키, 그리고 그에 해당하는 동물 이름의 학습 데이터가 주어졌다. 폭, 키를 특징(Feature)라고 한다. 동물 이름이 Target이 된다, 동물 이름들을 Label 이라 한다.   
Class는 Label 개수로 아래 예제 데이터에서는 5가 된다.

****

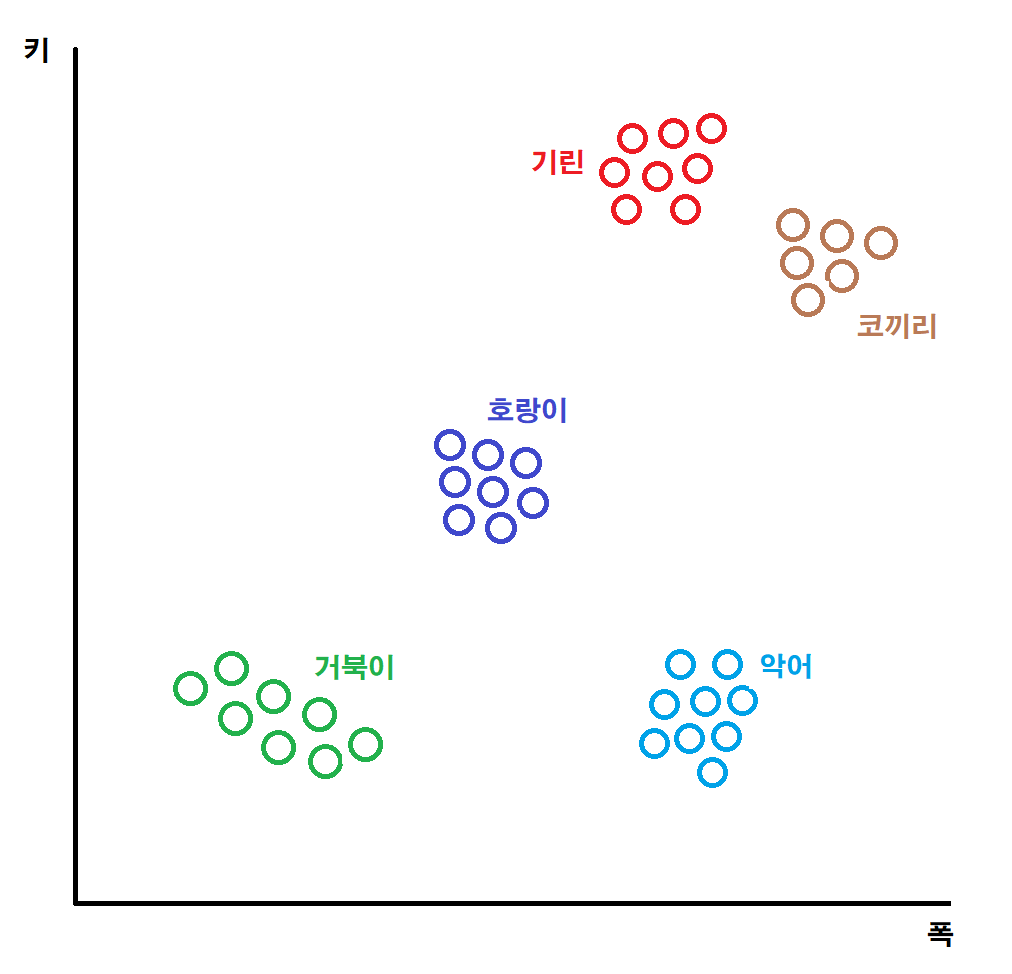
폭과 키 데이터로 동물 이름을 예측 한다. 아래 테스트 데이터에서 예측되는 동물 이름을 y 값으로 한다.



폭과, 키를 x로 두고, 예측되는 동물 이름을 y 라고 할 때, 함수를 아래와 같이 정의 할 수 있다. 학습 데이터 통해 함수f를 먼저 만든다, 다음으로 테스트 데이터의 y를 예측 한다.

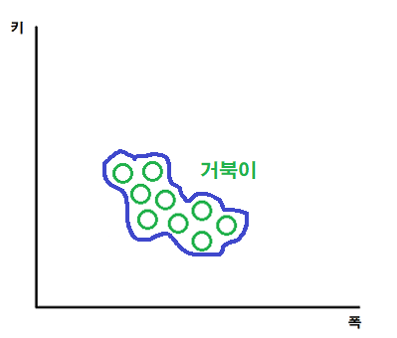


같은 동물들은 특징 값들이 비슷하거나 같을 것이다. 즉 아래와 같이 동물들의 특징 데이터를 그래프로 그린다면 아래와 같이 5개의 군집 형태로 모여 있을 것이다. 학습 데이터로 동물들을 5개 Class로 분류하는 함수 f를 만들어야 한다. 다음으로 만들어진 함수로 테스트 데이터들을 입력 값으로 하여 어떤 동물 인지 예측한다

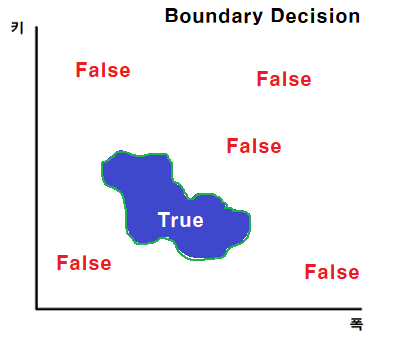


1. **XOR 문제**

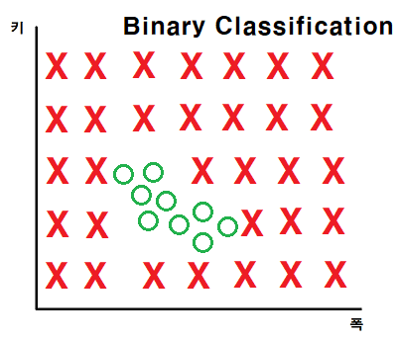
문제는 Class가 5인 Multi classification 함수를 만드는 것이다. 먼저 복잡한 문제를 간략하게 만들어 보자. 단순하게 Class가 2인 Binary classification 문제로 먼저 접근해 보자. 키와 폭 값이 “거북이” 데이터 군집 영역에 포함되면 True, 아니면 False 가 나오는 함수를 만들어 보자.



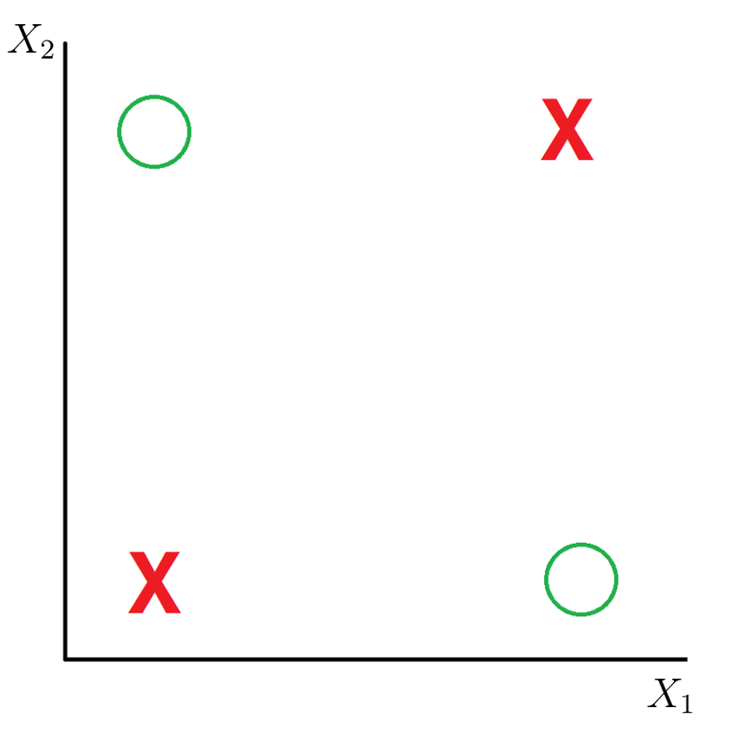
데이터의 분포 모양은 아래와 같이 복잡하다. 데이터의 분포 모양에 맞는 즉 Boundary decision 하는 함수를 구해야 한다. 아래 그림에서 녹색 테두리 안에 있으면 True, 밖이면 False 가 나오도록 해야 한다.



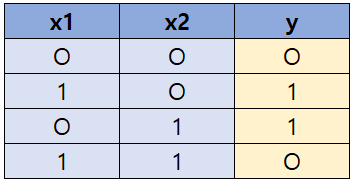
키, 폭의 데이터에 대한 True, False 결과를 그래프로 표현하면 다음과 같다

.

해당 그래프에서 데이터 분포를 한번 더 간략하게 만들어 보자. 4개(True 2개, False 2개) 데이터만 아래와 같이 분포되어 있다고 가정 하자.



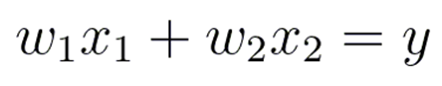
다시 표로 입력 값 x에 대한 출력 값 y를 표로 나타내면 다음과 같다. 컴퓨터는 True, False를 1, 0으로 처리하므로 결국 아래와 같다.



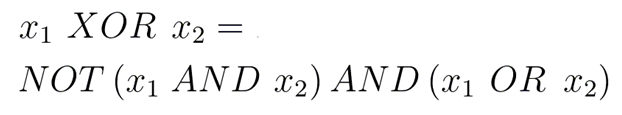
이것은 논리 게이트 XOR 동작과 같다, 그래서 이것을 XOR 문제라고 한다. 결국 Multi classification를 해결하기 위해서는 XOR 문제부터 해결 해야 한다.

1. **활성화 함수**

아래는 선형 함수와 XOR 함수이다. XOR 함수는 선형 함수로 해결 할 수 없다. 때문에 XOR 문제를 비선형 문제라고 한다.

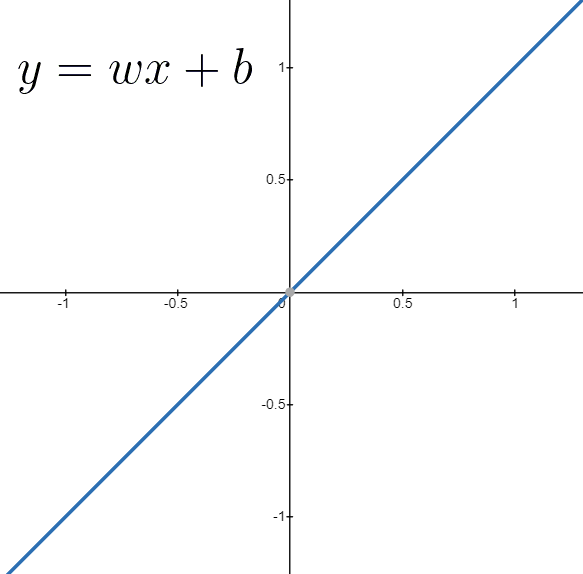


- 선형 함수 –

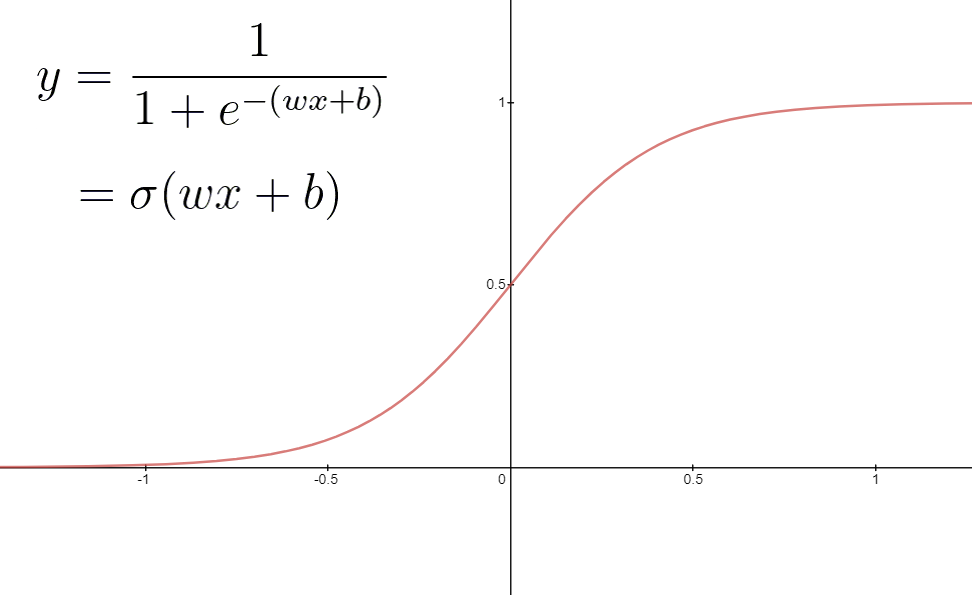


- XOR 함수 –

활성화 함수를 통해서 비선형 문제를 해결 할 수 있다. 선형 함수의 결과 값을 다시 활성화 함수의 입력 값으로 넣고 그것의 최종 결과값 y 가 1(True) 또는 0(False) 이 되도록 한다. 아래는 선형 함수이다.



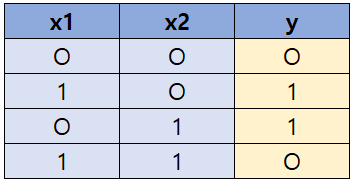
아래는 활성화 함수 중 하나인 sigmoid 이다. Tanh, elu, relu, leakyRelu 와 같은 다른 활성화 함수들도 있다.



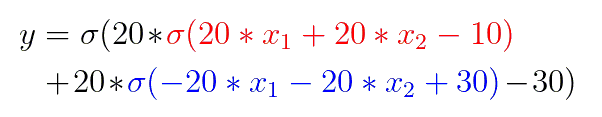
그래프를 보면 x값의 모든 범위(무한대)에서 y 값은 0 ~ 1 범위의 값들만 가진다. 이렇게 선형 함수와 활성화 함수 조합으로 최대한 결과 값이 0(False) 또는 1(True)로 나오도록 할 수 있다. 이렇게 선형 함수와 활성화 함수 조합을 로지스틱 함수 라고 한다.

1. **XOR 문제 해결**

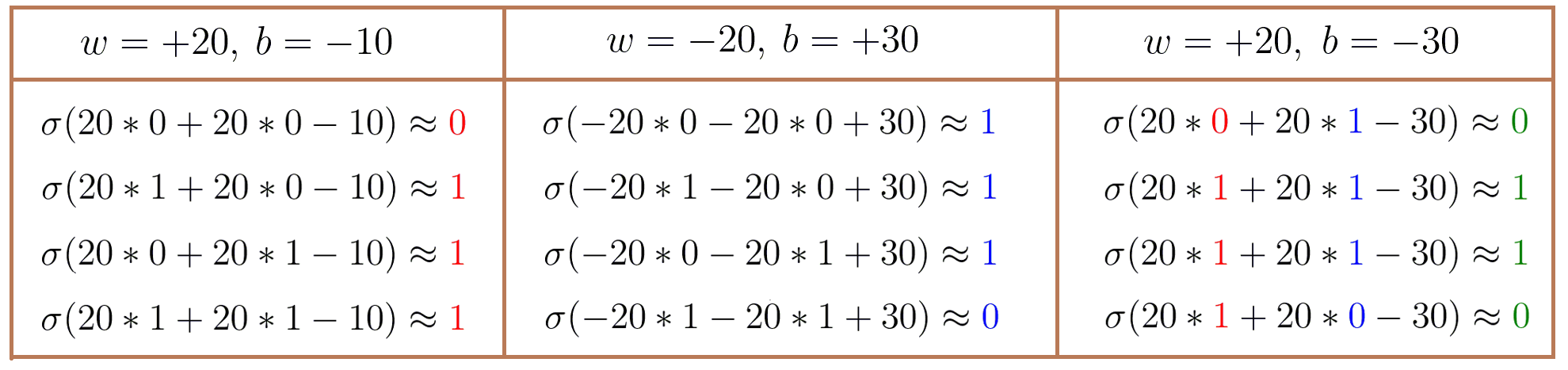
XOR 문제를 로지스틱 함수로 어떻게 해결 하는지, 확인 해보자.  
다음은 XOR 함수의 입력 값 x에 대한 출력 값 y이다



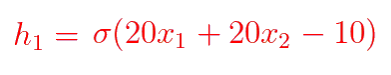
아래와 같이 로지스틱 함수들에 적절한 w, b 값을 가진다면, XOR 함수와 유사하게 만들 수 있다.



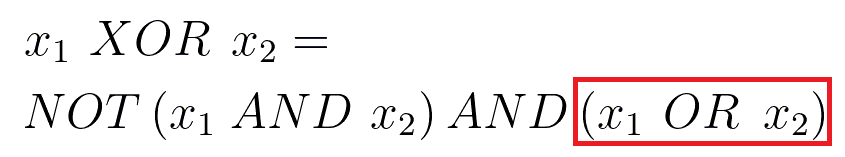
다음은 위의 함수를 입력 값 x에 따라, 계산된 출력 값 y 이다. 최종 결과 값이 XOR 함수와 유사한 것을 확인 할 수 있다.



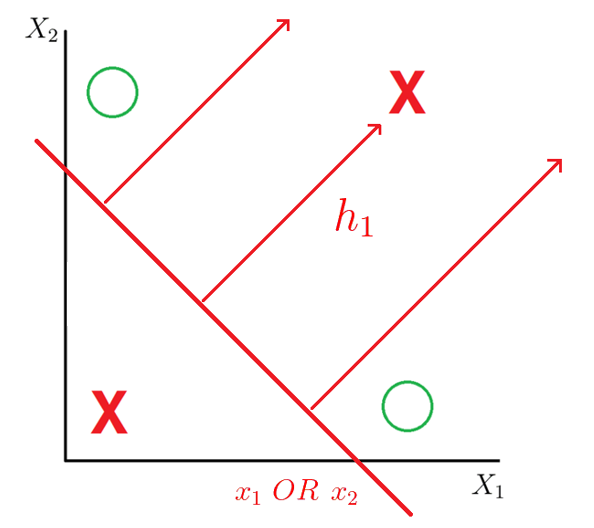
위의 함수를 XOR 함수와 비교 해보자. 위의 함수에서 빨간색 영역을 h1로 하자



h1은 아래 XOR 공식에서 빨간색 테두리 영역에 해당한다.



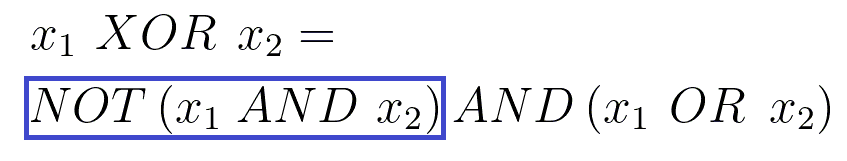
XOR 그래프에서 x1 OR x2 영역에 해당 한다.



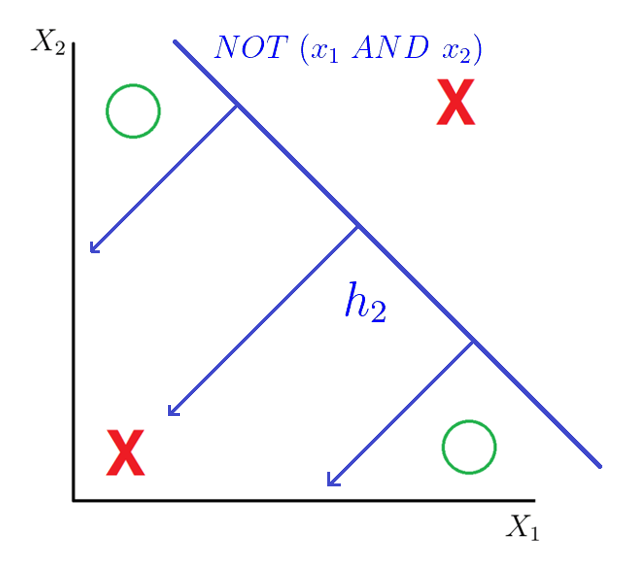
이번에는 위 함수에서 파란색 영역을 h2로 하자.



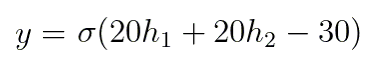
h2은 아래 XOR 함수에서 파란색 테두리 영역에 해당한다.



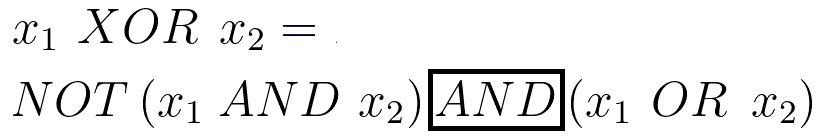
XOR 그래프에서 NOT(x1 AND x2) 영역에 해당한다



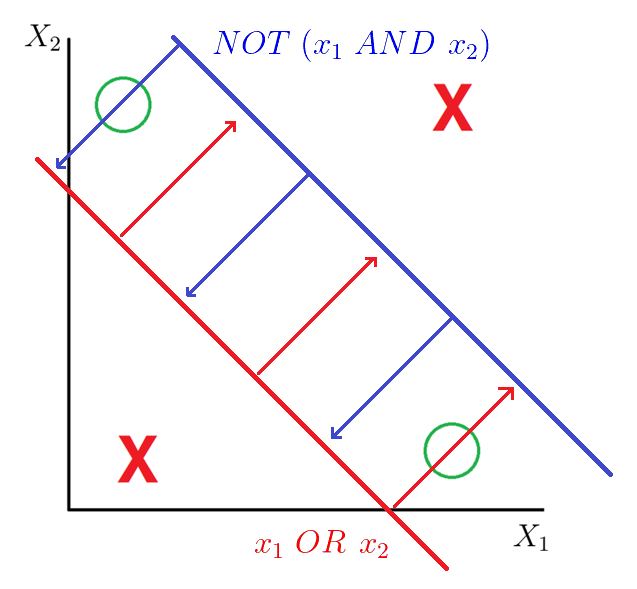
마지막으로 h1, h2 를 and 연산하여 교차 영역을 구한다.



아래 XOR 공식에서 검은색 테두리 영역에 해당한다.

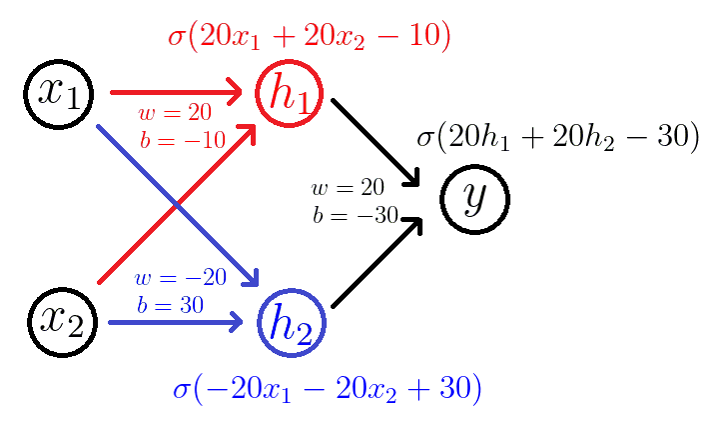


XOR 그래프에서 h1 AND h2 영역에 해당한다

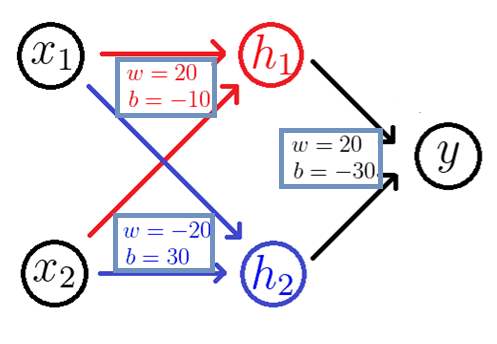


1. **DNN**

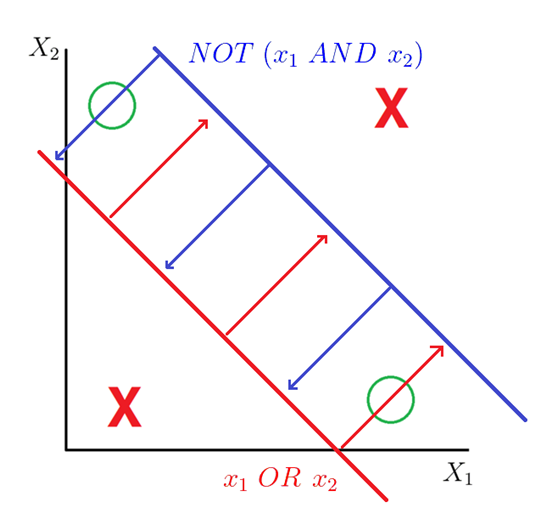
XOR 이 계산되는 과정에 대한 흐름도 이다. 아래 그림처럼 입력 층(x), 은닉 층(h), 출력 층(y) 층 순의 흐름으로 계산된다.



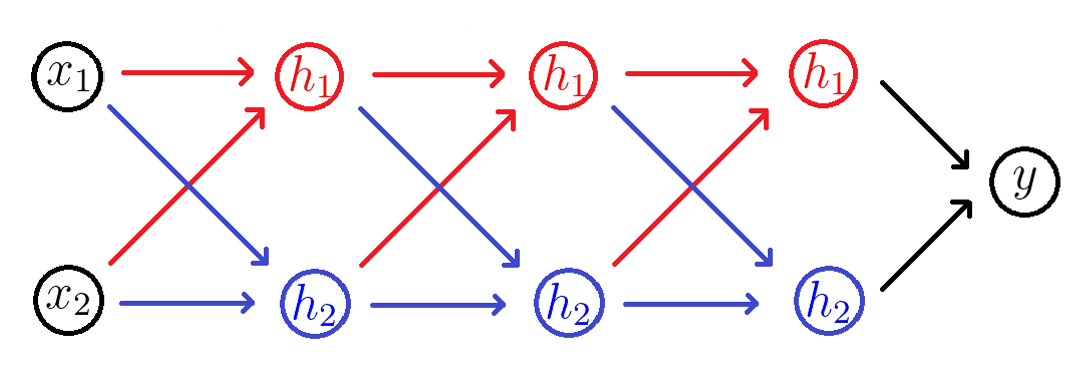
각 층 사이에는 w, b 가 있다.



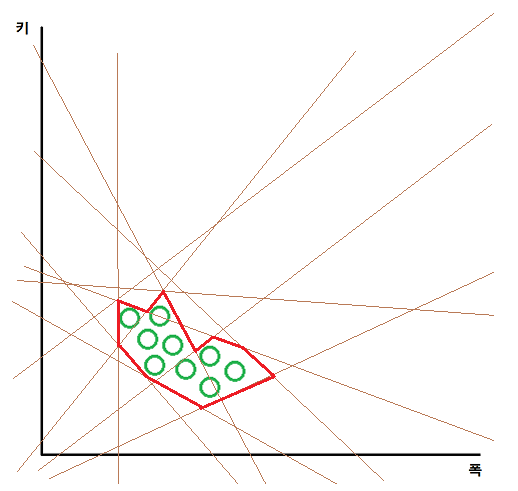
각각의 w, b는 아래 XOR 그래프에서 지나가는 선형 함수(빨간색, 파란색) 된다. 다음으로 활성화 함수가 적용되면 다음과 같이 교차 영역을 생성 할 수 있다. 즉 로지스틱 함수를 여러 개 층으로 쌓으면 XOR 데이터 구조를 Boundary decision이 가능하게 된다.



그렇다면 중간의 은닉 층이 많아진다면, 복잡한 형태의 데이터 군집도 Boundary decision 가능 하게 한다. 이렇게 2개 이상의 은닉 층의 구조를 DNN(Deep Neural Network) 이라고 한다.

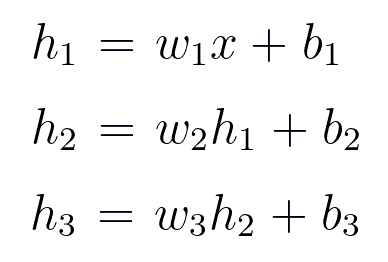


아래 그림은 DNN로 Boundary decision 하다는 것을 개념적으로 표현 하였다. 갈색 선들은 선형 함수들이고, 빨간색 테두리는 선형 함수들의 활성화 함수에 의해 결정된 교차영역이다.

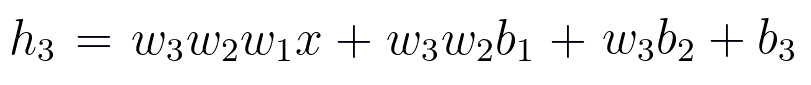


1. **문제를 해결 할 수 없는 다층 선형 함수**

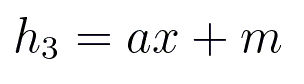
선형 함수만으로 다층을 쌓으면 Boundary decision 가능 할까?  
다음과 같이 각 층의 선형 함수는 아래와 같다.



대입하여 계산하면 마지막 은닉 층의 함수는 아래와 같다.

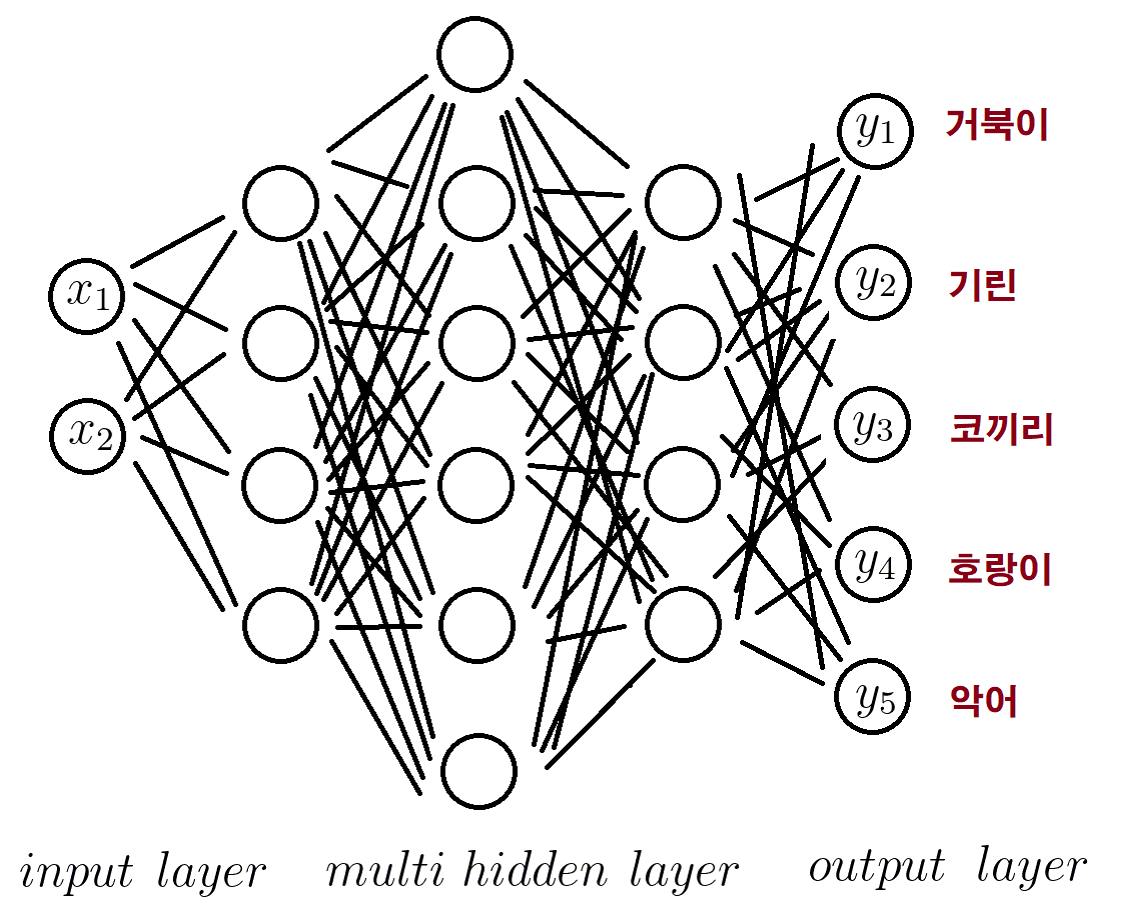


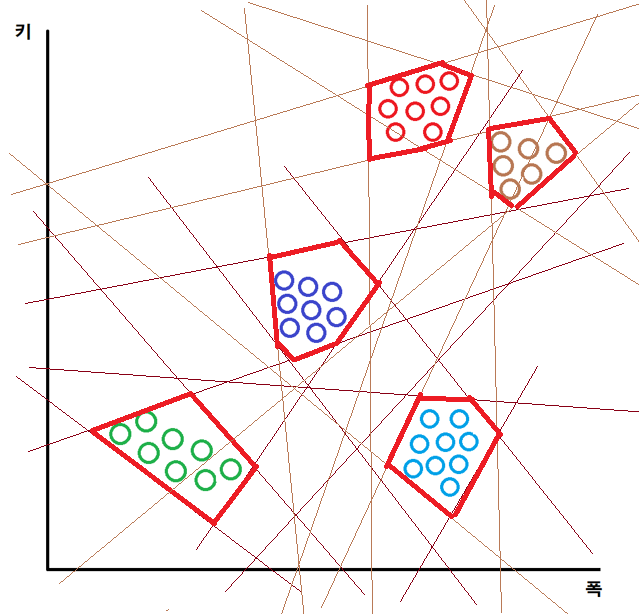
각 은닉 층의 w, b 들은 고정된 상수 값이다. 때문에 아래와 같이 치환하면, 결국 선형 함수이다.



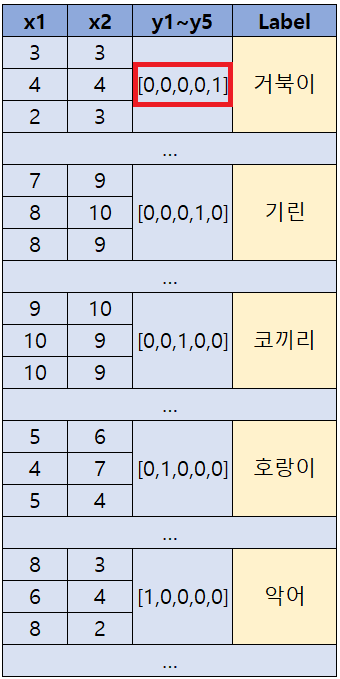
다층 선형 함수는 결국 단층 선형 함수가 된다! 그래서 Boundary decision 하는 함수를 만들 수 없다.

1. **Multi classification**  
   출력 층의 y 값을 여러 두면, Multi classification 에 대한 Boundary decision 이 가능하게 된다.

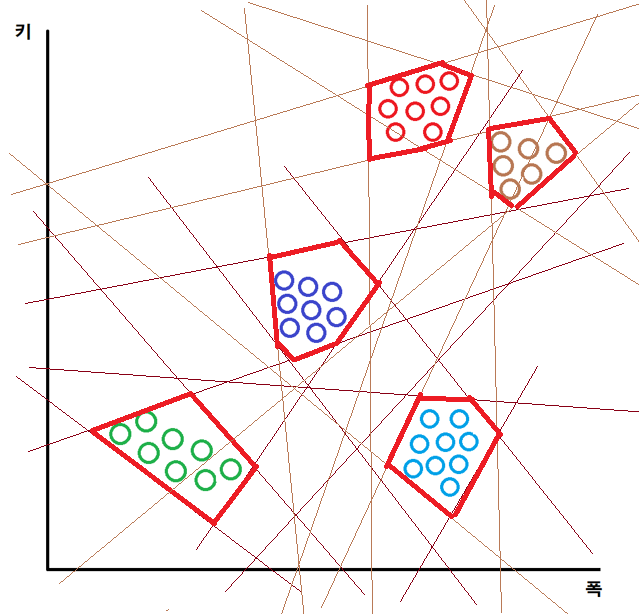


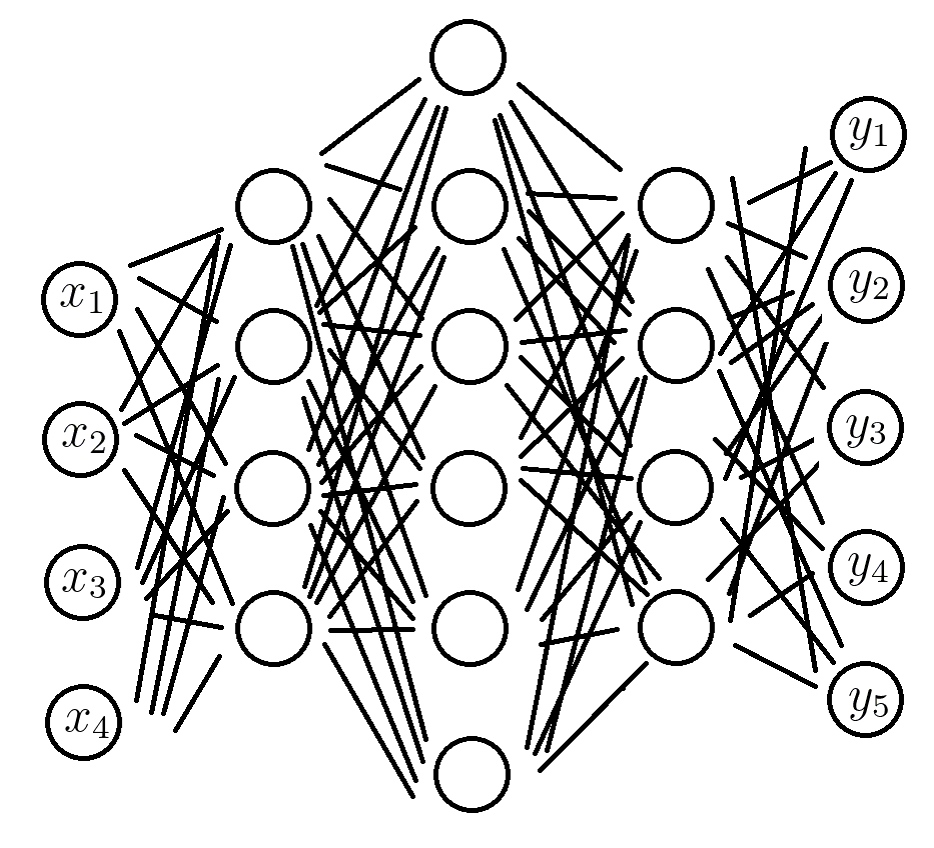


Binary Classification 은 둘 중에 무엇이냐? 의 문제, 즉 True/False 문제이다. 하지만, Multi Classification은 여러 개 중에서 무엇이냐? 의 문제 이다. Label 들은 상호 수평 관계에 있다. 이런 경우, One Hot Encode 하여 y값들을 만든다. Class(Label 개수 크기) 크기의 배열에서 각 Class 에 해당하는 특정 위치(index) 의 값을 1로 둔다. 아래는 Label들을 one hot encode변환한 y값들이다.



**8. N-Feature**

지금까지, 2차원 Feature(x1(폭), x2(키))로 설명했다. 그렇기에 Boundary decision 이 2차원 면의 형태로 표현 되었다. 그렇다면 Feature가 다 차원이면, Boundary decision 은 어떤 형태 일까? Feature가 다차원이면, Boundary decision 형태 또한 다 차원 면이 된다. 아래 그림은 4차원 Feature를 표현 하였다.

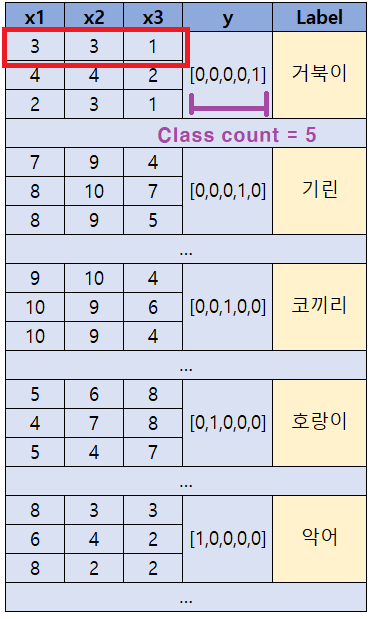


아래 구겨진 종이 사진은 3차원 Feature에 대한 Boundary decision 을 표현 하였다. 저 종이 속에 데이터들이 군집 형태로 모여 있다고 상상하라.



1. **은닉 층들의 Weight, Bias 구하기**

아래 학습 데이터는 3차원 Feature 이고 Class는 5개 이다.



아래 DNN에서 위의 입력 값 x [3, 3, 1] 들을 넣으면 출력 값 y [1, 0, 0, 0, 0] 나오도록 해야 한다. 즉 그렇게 계산 되도록 하는 각 은닉 층들의 최적의 w, b를 찾아야 한다. 다음으로 경사 하강 법, 오차 역 전파를 이해해야 한다.

