

즐거운  
프로그래밍  
경험



## 누구나 쉽게 시작하는 딥러닝 기초 프로그래밍

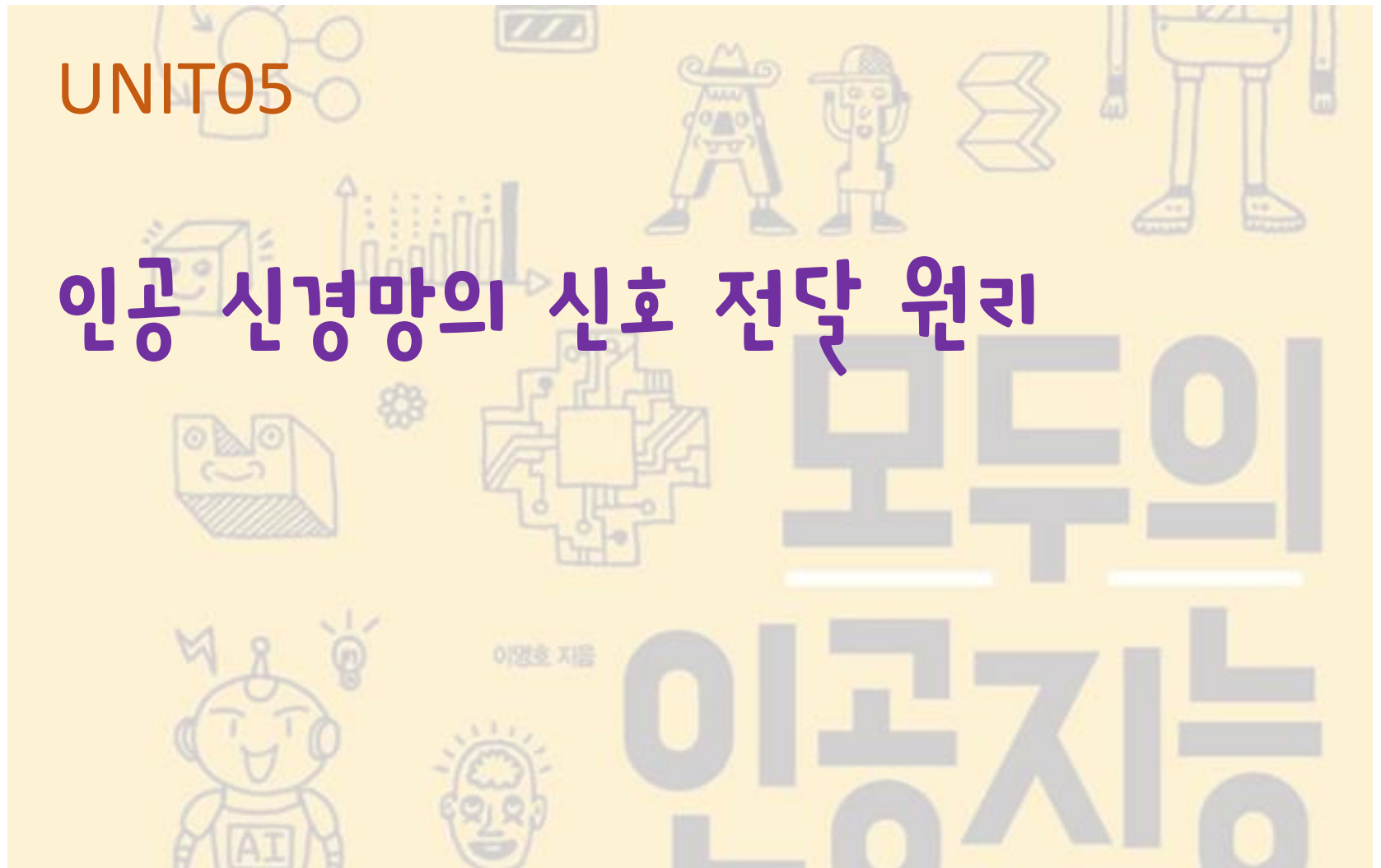
길벗

- 파이썬을 처음 시작하는 사람들의 눈높이에서 원리를 설명한다! ● 코랩, 케라스를 활용하여 간편하게 실습한다!
- 숫자 인식, 코로나 확진자 수 예측, 숫자 생성 인공지능, 모델을 직접 만들어 본다!

인공신경망의 역사는 1943년, 맥컬락과 피츠는 <A logical calculus of the ideas immanent in nervous activity> 논문에서 뇌의 복잡한 신경 구조를 구현 할 수 있다고 함. 그 후 15년 뒤 프랭크 로젠블러트는 퍼셉트론이라는 개념을 <A perceptron: A probabilistic model for information storage and organization in the brain>이라는 논문에서 발표한다. 바로 이 퍼셉트론이 인공 신경망의 시초가 되는 개념임. 오늘날 사용하고 있는 딥러닝도 퍼셉트론의 구조와 비슷하다. 그러나, 1958년에 발표한 퍼셉트론이 왜 오늘날 되어서야 주목받는 것일까? 퍼셉트론에 있던 여러 문제점을 해결할 방법을 찾았기 때문이다. 1969년 마빈 민스키와 새뮤엘 페퍼트가 <Perceptrons: an introduction to computational geometry> 논문을 통해 퍼셉트론의 치명적인 약점을 찾아냄. 하지만 연구자들은 멈추지 않고 1986년 데이비드 렘멜하트와 제임스 맥클레랜드가 <Parallel Distributed Processing>에서 퍼셉트론의 문제를 해결할수있는 방법으로 다층 퍼셉트론(MLP, Multi-Layer Perceptrons)과 오차 역전법(Backpropagation Algorithm)을 제시함. 딥러닝처럼 여러 층을 가진 신경망을 구성하여 기존 퍼셉트론의 한계 극복, 여러 층이 생기면서 늘어난 계산량을 “오차 역전파”라는 알고리즘을 통해 해결. 이후 다층 퍼셉트론에 대한 꾸준한 연구와 학습 알고리즘의 발전, 빠르게 계산할 수 있는 GPU와 같은 하드웨어의 발전에 힘입어 딥러닝 기술이라 불리는 인공 신경망이 오늘날 두각을 나타냄

## UNIT05

# 인공 신경망의 신호 전달 원리





# 인공 신경망의 신호 전달 원리

- 인공 신경망=사람의 뇌와 유사
- 인공 신경망의 학습과정
  - 우리 뇌의 어떤 특징
  - 뉴런의 신호
  - 하나의 뉴런은 다른 뉴런에게 신호 전달
- 뉴런의 특징
  - 신호 전달하는 과정
  - 신호 전달받는 과정



# 1. 신호를 전달할 때 사용하는 가중치와 편향



그림 5-1 | 가중치와 편향

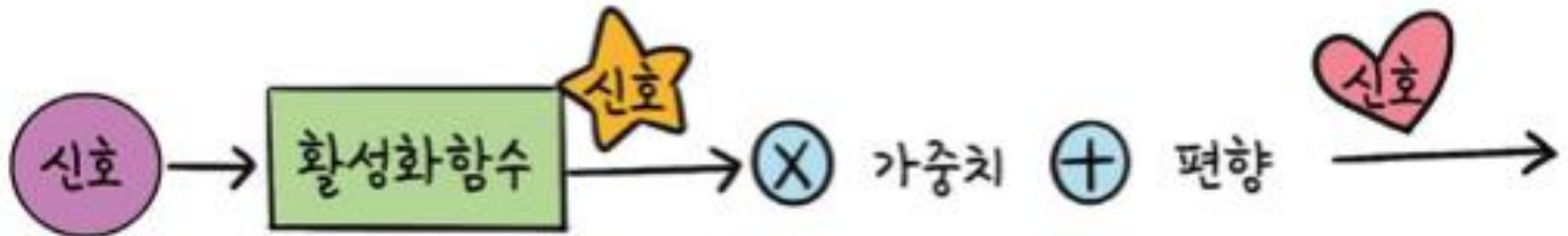


그림 5-2 | 가중치와 편향에 따라 달라지는 신호 세기



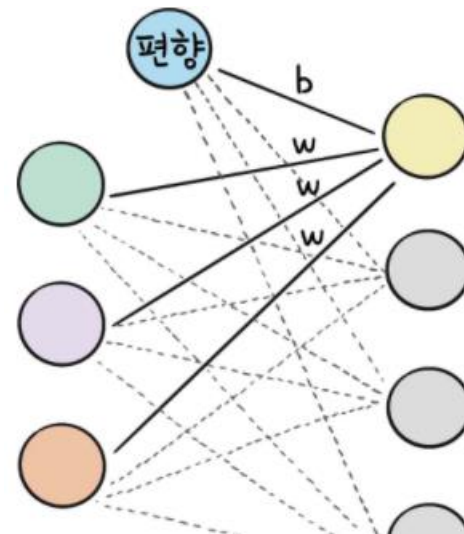
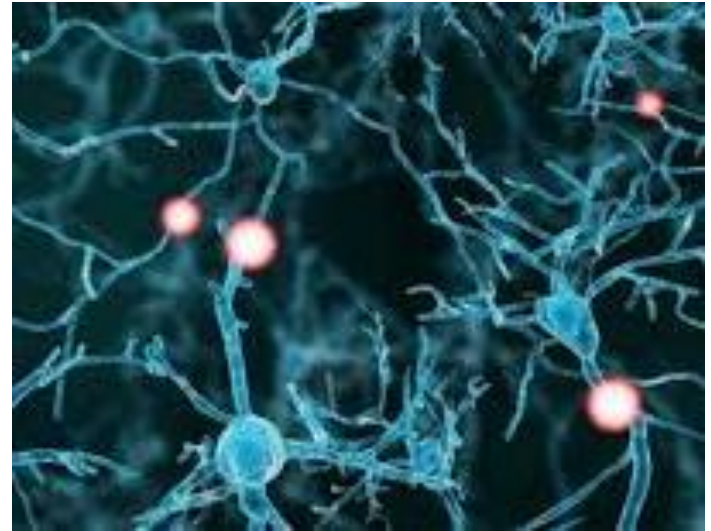
# 가중치와 편향

- 가중치(weight)

- 뉴런에서 다음 뉴런으로 신호를 전달할 때 그 값의 중요도 표현하기 위해

- 편향(bias)

- 한쪽으로 치우치는 값을 더할 때



인공신경망 조금 더 깊이 공부하면 **가중치와 편향**이  
인공신경망 기초 개념 이라는 것으로 알 수 있다.

“**가중치**” 라는 말에 뜻에서 볼 수 있듯이 가중치는 그 값이  
얼마나 중요한지 그렇지 않은지를 표현하기 위한 도구이다.

인공신경망에서도 각 뉴런에서 다음 뉴런으로 신호를  
전달할 때에 그 값의 중요도를 표현하기 위해 사용된다.

“**편향**” 이란 한쪽으로 치우치는 의미다.

예를 들어, “그 사람은 어떤 신념에 편향되어 있다” 처럼  
인공신경망 에서는 모델의 성능을 높이기 위해 가중치를  
거쳐 변환된 **신호 세기를 조절**할 필요가 있다.

이를 위해 한쪽으로 치우치는 값을 더할 때 편향이라는 값을  
사용한다.

## 2. 들어오는 신호 세기를 조절하는 활성화 함수

### • 뉴런의 신호 전달 방식

- 앞에서 뒤로 전달
- 단일 방향
- 여러 뉴런과 연결

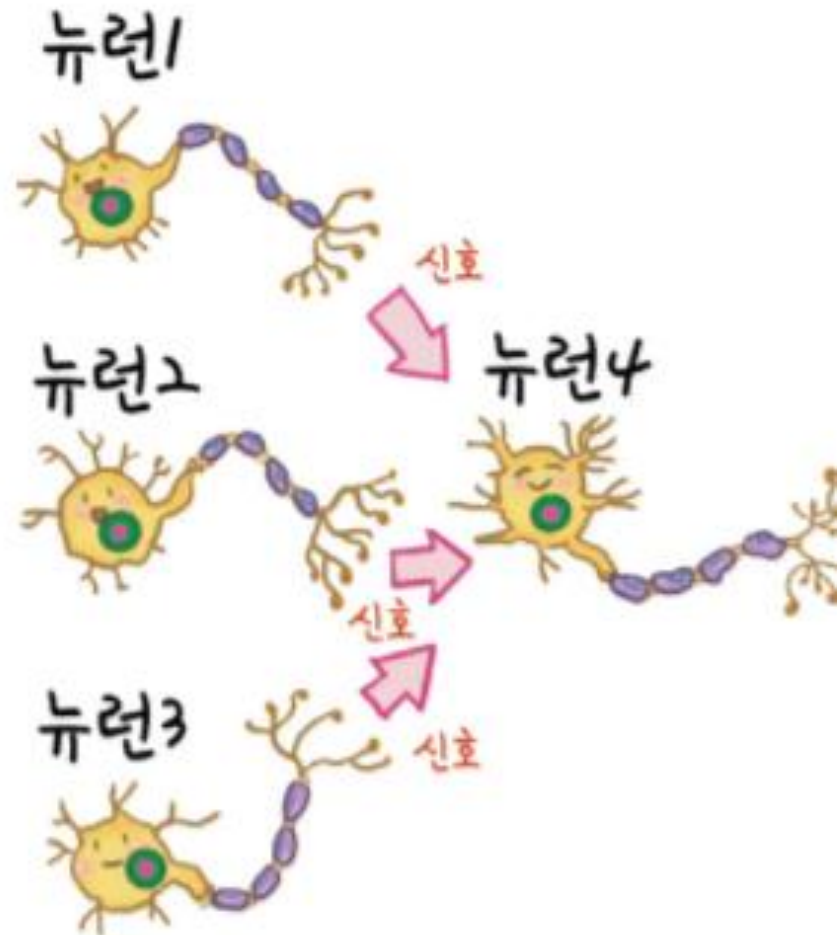


그림 5-4 | 서로 연결된 뉴런의 모습



# 뉴런의 신호 전달 방법

## • 신호 전달 결정 여부

### • 역치

- 특정한 전기적 신호가 어떤값(역치)보다 높을 경우 신호 전달
- 그렇지 않은 경우, 신호를 전달하지 않음

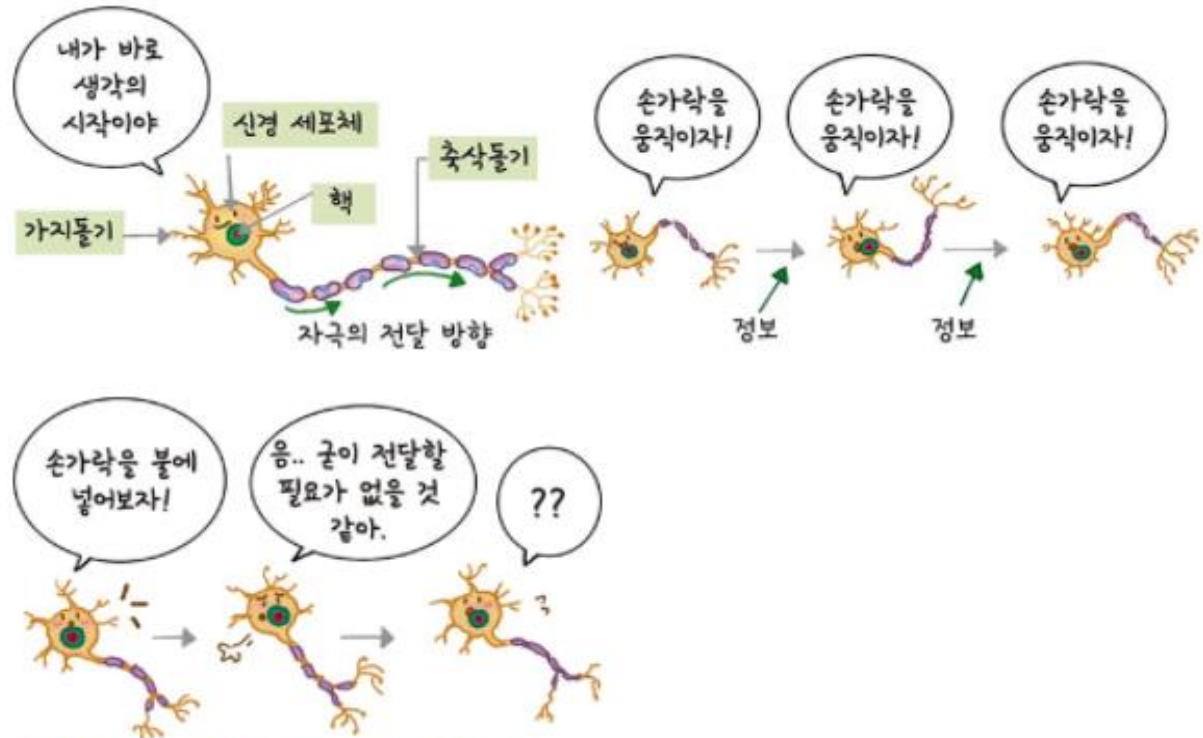


그림 5-5 | 신호를 전달하는 뉴런들의 모습



# 인공 신경망 신호 전달 방법

## • 신경망의 역치 개념 사용

- 뉴런으로 들어오는 여러 신호를 조절하기 위해
- **활성화(activation) 함수** 사용
- 여러 뉴런에서 들어온 신호 세기를 특정한 값으로 바꾸기 위해 활성화 함수 사용

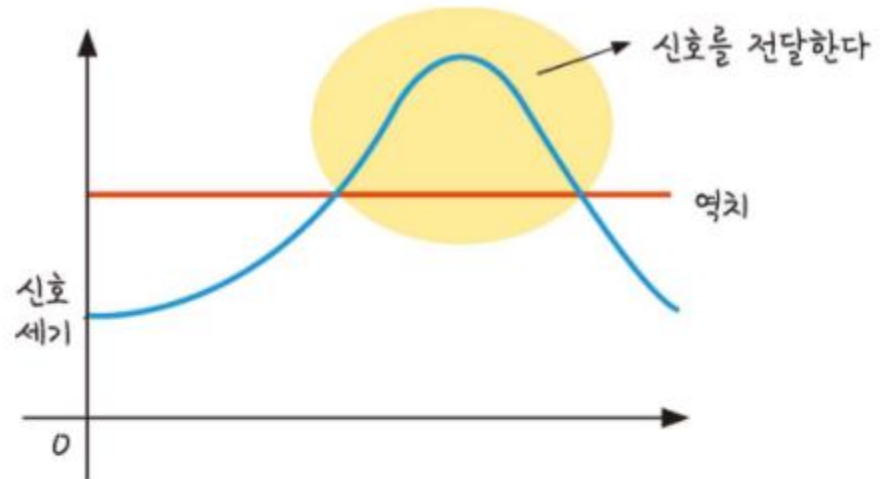


그림 5-6 | 역치 값보다 더 큰 신호가 들어왔을 때: 신호를 전달한다

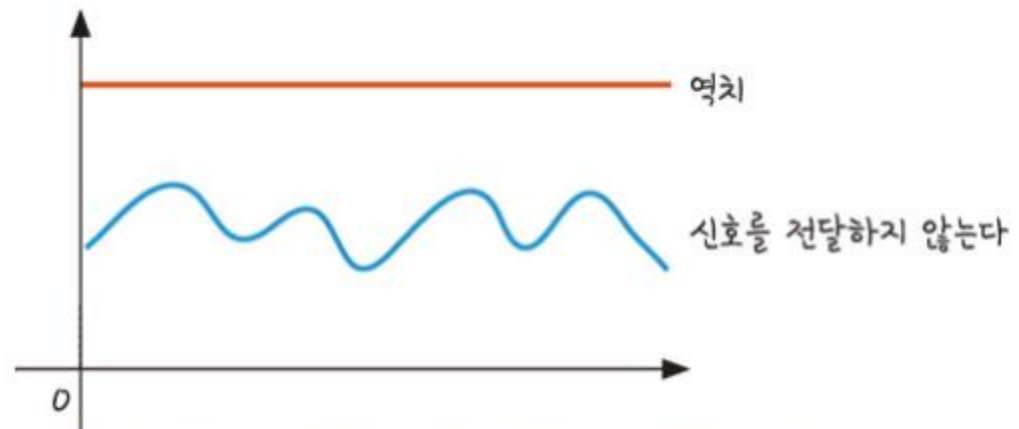


그림 5-7 | 역치 값보다 더 작은 신호가 들어왔을 때: 신호를 전달하지 않는다



# 활성화 함수

## • 활성화 함수

- 신호체계를 조절하는 데 특히 레이어와 레이어 사이에 있어서 여러 노력해서 특정한 들어가는 신호를 종합해서 하나의 값으로 바꿔주는 역할
- 인공신경망에서 활성화 함수로 신호 세기 조절

- 시그모이드 함수
- 하이퍼볼릭탄젠트
- 렐루
- Leaky 렐루 함수

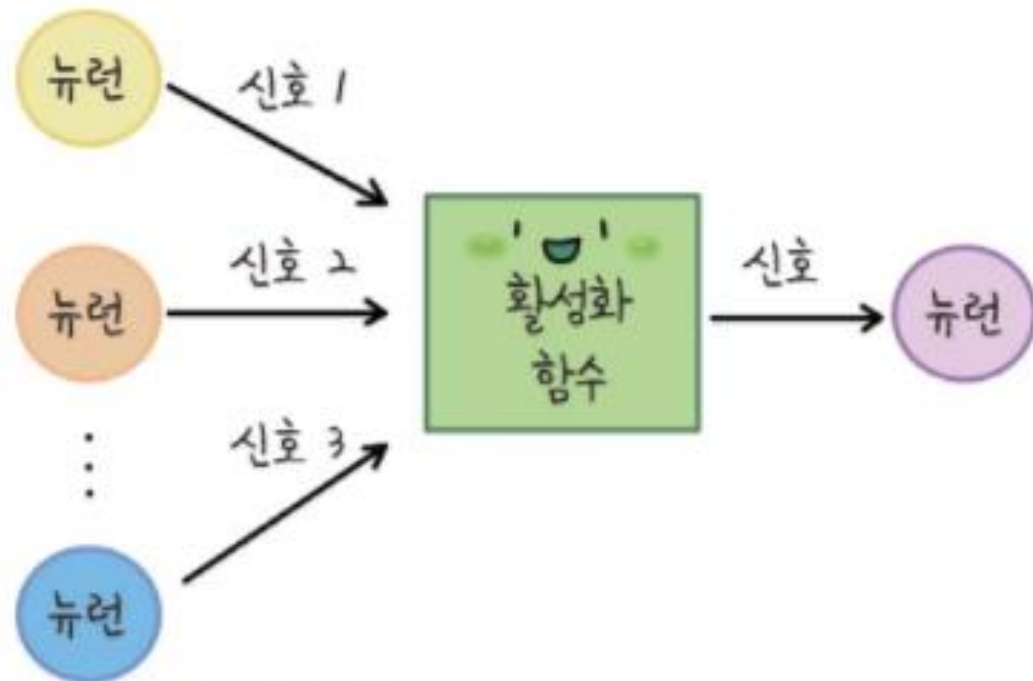


그림 5-8 | 활성화 함수



# 활성화 함수- 시그모이드 함수

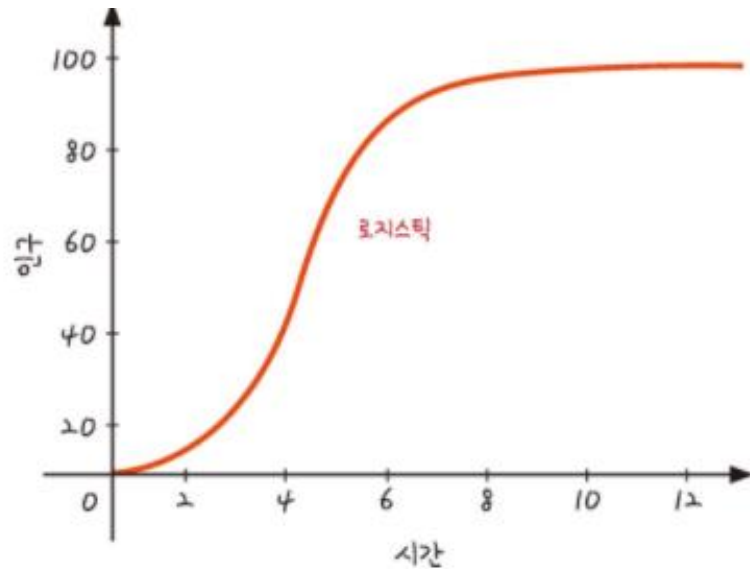


그림 5-9 | 인구 수 증가를 나타내는 로지스틱 함수

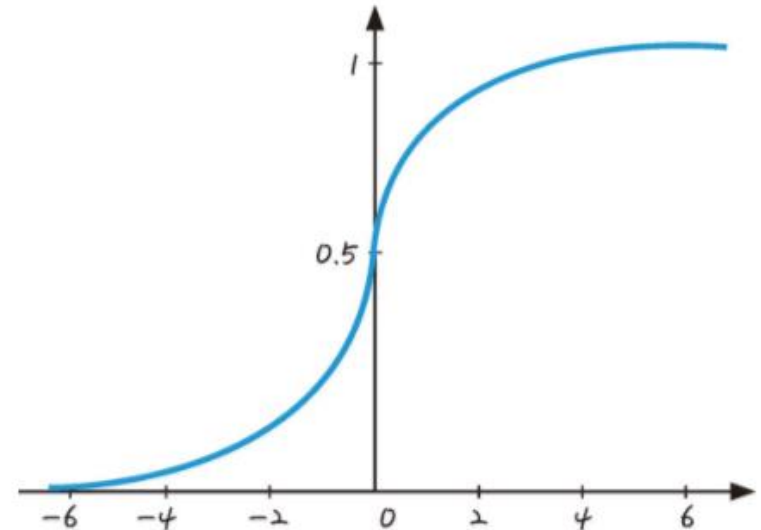


그림 5-10 | 로지스틱 곡선

# 활성화 함수- 하이퍼볼릭탄젠트 함수

- 하이퍼볼릭 탄젠트 함수는 입력 값이 음수일 때, 출력 값이  $-1$ 에 가까워지는 특징
- 즉, 하이퍼볼릭 탄젠트 함수는 값이 작은 신호를  $-1$ 에 가까운 숫자로 바꾸어서 보냄

## TIP:

하이퍼볼릭은 "쌍곡선" 이라는 의미로 쌍곡선 두 지점에서 거리가 같은 곡선을 의미.

그래프를 살펴보면 대칭이 되는 특정한 두 점에서의 거리가 항상 같은 것을 확인할 수 있음

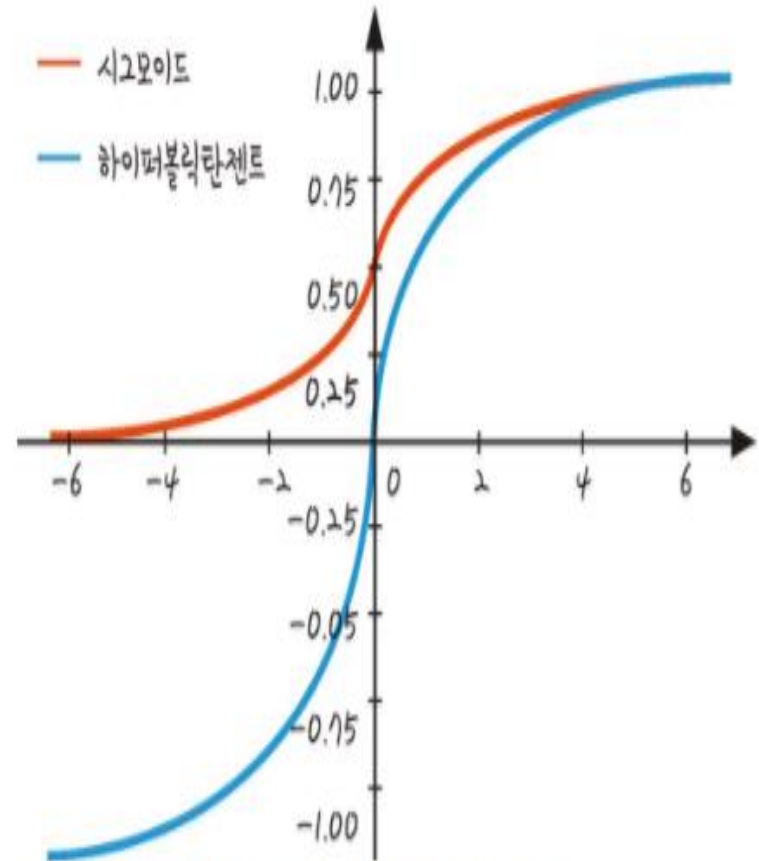


그림 5-11 | 하이퍼볼릭탄젠트 함수와 시그모이드 함수





# 활성화 함수-렐루 함수

## • ReLu 함수

- Rectified Linear Unit
- 고르게 하다/직선으로 이루어진
- 입력값이 1이상이면, 그대로 출력값 표현
- 입력값이 음수이면, 0으로 표현되는 단점
  - Leaky ReLU 함수 사용

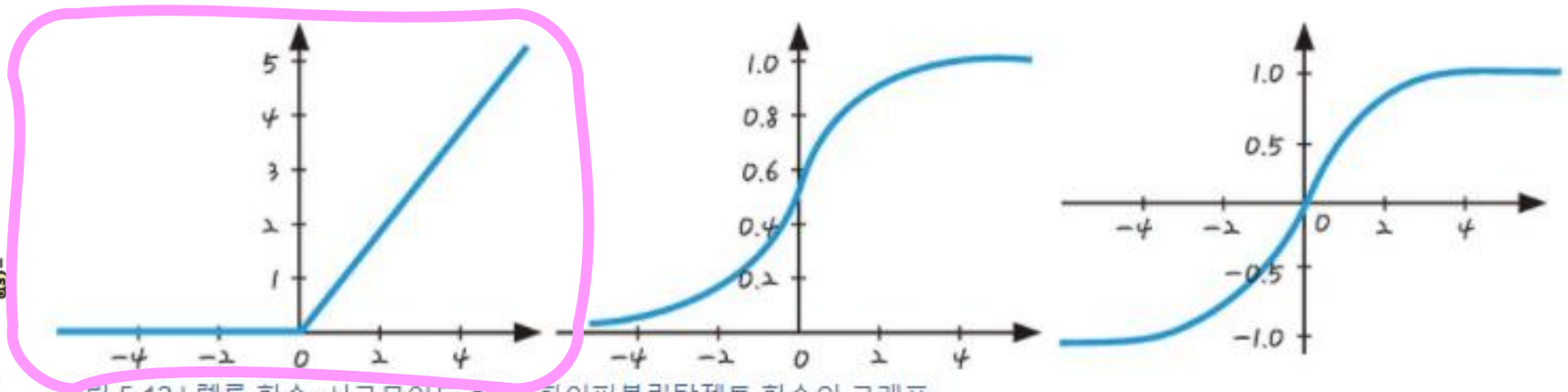


그림 5-12 | 렐루 함수, 시그모이드 함수, 하이퍼볼릭탄젠트 함수의 그래프



# 활성화 함수- Leaky ReLU함수

- 음수일 경우, 0으로 출력되지 않고 양수의 기울기와 다르게 완만한 선 표현
- 미세하게 차이나는 음수의 값 전달

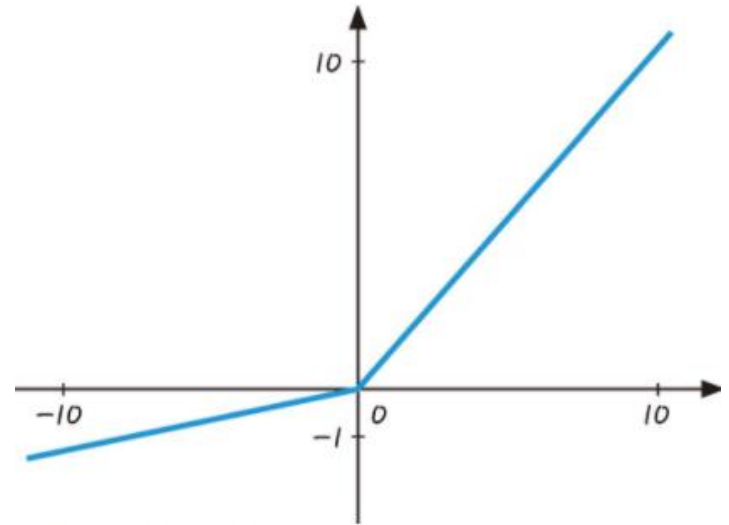


그림 5-13 | Leaky 렐루 함수



# 소프트 맥스 함수

- 최종값을 정규화하는데 사용하는 함수
- 인공지능망에서 항상 사용되는 것이 아니며 분류 문제에서 사용되는 함수

## TIP

여기에서 정규화는 특정한 범위를 지정해 주고, 데이터를 그 범위 중 하나로 바꿔 주는 것을 말합니다. 가장 작은 데이터를 0으로 가장 큰 데이터를 1로 바꾸고 그 사이의 값들을 크기에 따라 0과 1사이의 값으로 바꾸는 것이죠.

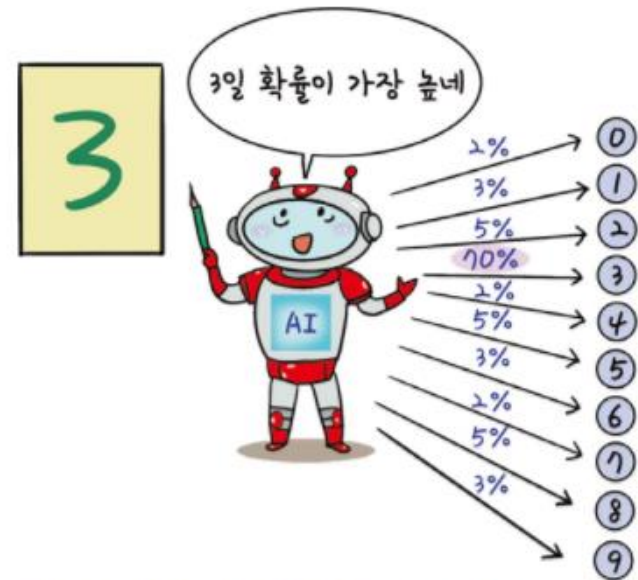


그림 5-14 | 이미지가 3일 확률은 70%로 가장 높다

## UNIT06

# 인공 신경망의 학습 원리

모두의  
인공지능

이영호 지음

# 1. 인공 신경망의 오차 구하기

- 남녀를 구분하는 인공지능 모델의 오차 구하기
- 나이대를 예측하는 모델의 오차 구하기
- 나이를 예측하는 인공지능 모델의 오차 구하기





# 남녀를 구분하는 인공지능 모델의 오차 구하기

## • 이진 분류

- 2개 중 하나로 구분하는 문제

### 잠깐만요

#### 이항 교차 엔트로피

이렇게 계산하는 방법 중 하나가 바로 이항 교차 엔트로피(바이너리 크로스엔트로피, binary crossentropy)입니다. 이항 교차 엔트로피의 원리는 다음과 같습니다. 인공지능이 잘 예측했다면 오차값을 0으로 주고, 잘 예측하지 못했다면 오차값을 상당히 크게 주는 것이죠. 그러면 인공지능이 잘 맞춘다면 오차값은 0에 가까워질 것이고, 그렇지 않다면 오차값은 상당히 커지게 됩니다.

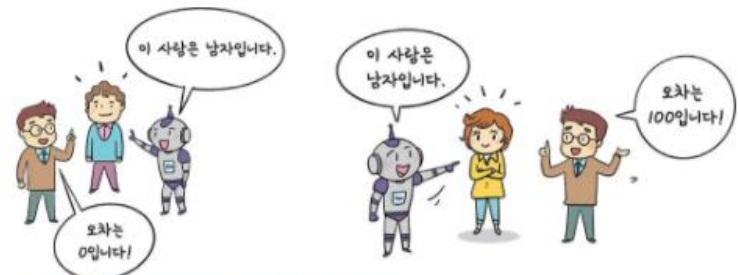


그림 6-1 | 결괏값에 대해서 오차값을 더하기

# 나이대를 예측하는 인공지능 모델의 오차 구하기

## • 다중 분류 문제

- 여럿 중 하나로 구분하는 인공지능



그림 6-2 | 나이대를 예측하는 테스트

### 잠깐만요

#### 다중 분류 손실 함수

이와 같이 오차를 계산하는 방법에는 다중 분류 손실 함수(카테고리컬 크로스엔트로피, categorical crossentropy) 방법이 있습니다. 여러 값 중 하나를 예측하는 모델일 경우에 정답을 예측할 경우에는 오차를 0으로, 정답이 아닌 값을 높은 확률로 예측하면 오차를 많게, 낮은 확률로 정답이 아닌 확률을 예측하면 오차를 적게 하는 방법입니다.

# 나이를 예측하는 인공지능 모델의 오차 구하기



그림 6-3 | 특정한 값을 예측한 결과: 10살이라는 오차 발생

## TIP

평균 제곱 오차(mean squared error)가 바로 이러한 방법 중 하나입니다. 평균 제곱 오차는 예측 값이 실제 값에서 얼마나 떨어져 있는지를 알아보는 방법입니다. 이때 생기는 오차를 제공하기 때문에 평균 제곱 오차라고 합니다. 제공하는 이유는 바로 부호를 없애기 위해서입니다. 양(+)의 방향으로 떨어져 있는지 음(-)의 방향으로 떨어져 있는지는 중요하지 않습니다. 중요한 것은 바로 얼마나 떨어져 있느냐죠. 이를 명확하게 나타내기 위해 제공하는 것입니다. 음(-)을 제공하면 양수(+)가 되기 때문이지요.

## 잠깐만요

실제로 인공 신경망으로 인공지능을 만들 때 이런 방식으로 오차값을 구하나요?

원리는 동일합니다. 하지만 실제 오차값을 계산할 때에는 여러 공식을 사용합니다. 정답은 없지만, 어떤 공식을 사용하는가에 따라서 인공지능의 성능 또한 달라집니다. 그러므로 데이터에 적합한 오차 공식을 구하는 것이 필요합니다.

이 부분이 더 궁금하시다면 《케라스 창시자에게 배우는 딥러닝(길벗, 2018)》을 찾아보세요.

## 2. 인공 신경망의 핵심! 오차 줄이기

- **신호세기의 조절=가중치의 값 수정**

- 오차를 줄이기 위해 전달하는 신호의 세기를 조절 할 수 있는 방법이 바로 가중치의 값을 수정하는 것

- **경사 하강법**

- 기울기를 사용하여 가중치의 값을 변경

- **오차 역전파법**

- 경사 하강법의 개념을 사용하여 여러 가중치를 차례로 변경



# 경사 하강법

- 기울기로 가중치 값을 변경하는 경사 하강법
- 가중치에 따라 신호의 세기가 바뀌고, 그에 따라 인공지능의 결과값 결정
- 가중치가 인공지능의 성능을 결정하는 핵심

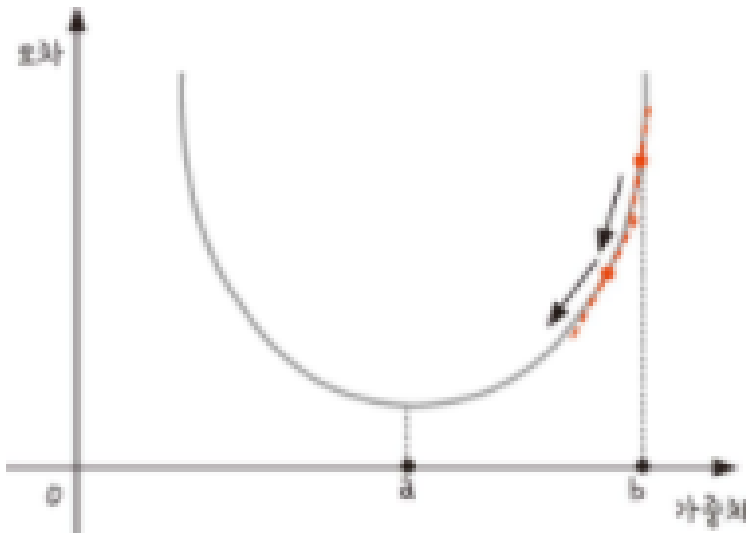


그림 6-4 | 가중치에 따라 달라지는 오차





# 경사 하강법

## • 미분

- 한 지점에서의 기울기
- 기울기로 그 다음 값에 대한 예측 가능
- 기울기를 활용하여 인공 신경망의 오차 수정

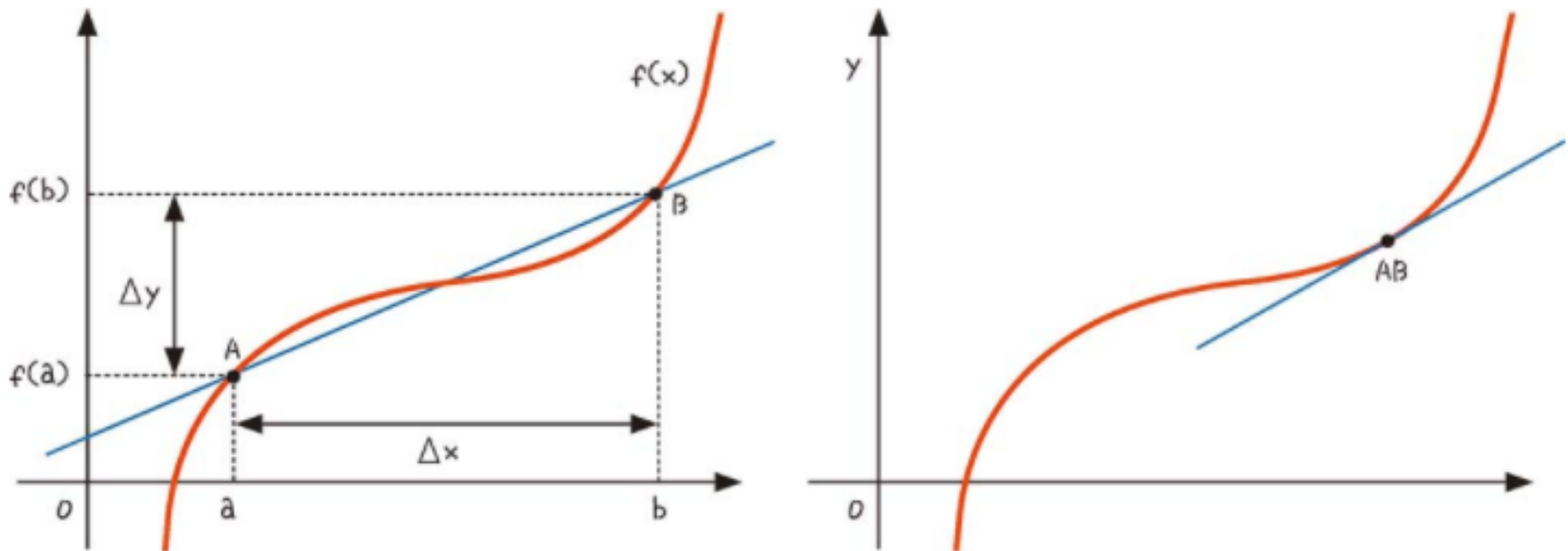


그림 6-5 | 한 지점에서의 기울기: 미분



# 오차 역전파법

- 백프로파게이션
- 체인룰
- 여러 가중치를 차례로 변경해 나가는 오차 역전법
- 은닉층이 여러 개로 존재할 때
- “오차를 끝에서부터 거꾸로 가면서 줄인다”
- 뒤에서부터 앞으로 값을 수정해 나가는 방법



# 오차 역전파법

- 백프로파게이션
- 체인룰
- 뒤로 가면서 가중치를 수정하고 다시 한번 데이터를 흘려보낸 후 결과값을 살펴보고, 그 결과값이 정답값과 어떤 차이가 있는지 살펴본 후 다시 오차 역전파법을 사용하여 가중치 수정
  - 인공 신경망은 이 과정을 반복하면서 오차를 0으로 줄여나감
  - 인공지능 학습 방법

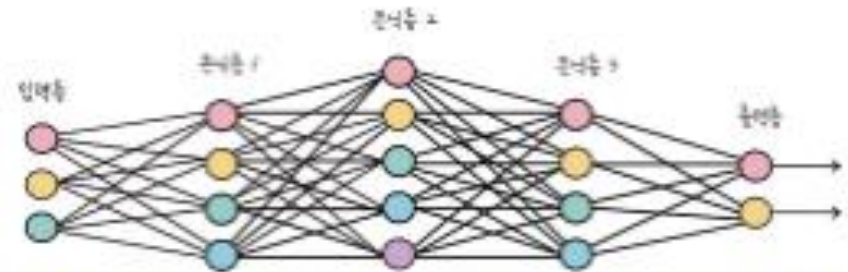


그림 6-6 | 수많은 가중치를 가진 인공 신경망

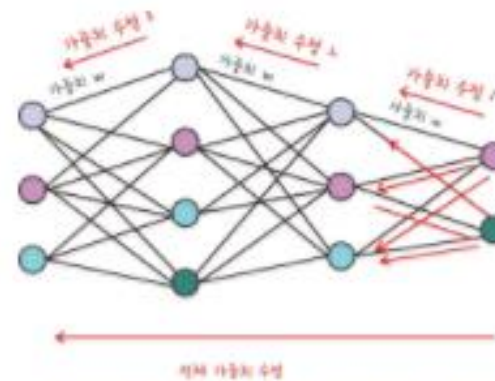


그림 6-7 | 뒤에서부터 앞으로 값을 수정해 나가는 방법 : 오차 역전파법

## TIP

이름이 '체인룰'인 이유가 궁금하지 않나요? 자전거 체인을 떠올려 봅시다. 하나하나가 서로 연결되어 있죠. 오차 역전파법을 사용하는 모습도 이와 유사합니다. 서로 맞물려서 값이 변하기 때문에 체인룰이라는 이름이 붙었습니다.



# UNIT 07

## 텐서플로 플레이그라운드로 딥 러닝 체험하기



# 텐서플로 플레이그라운드 접속하기

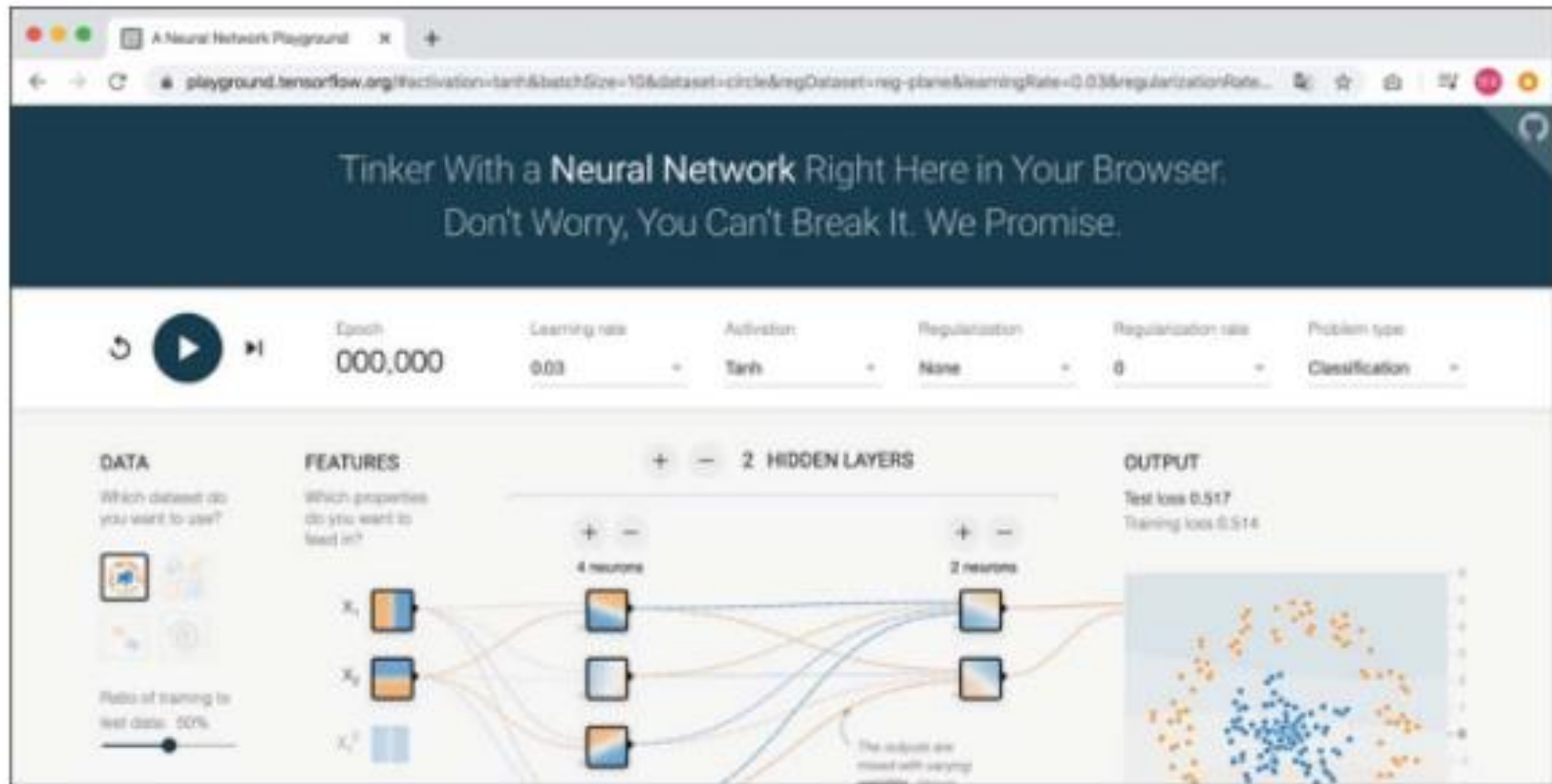


그림 7-1 | 텐서플로 플레이그라운드 접속

데이터에서 파란색은 양수인 +1, 주황색은 음수인 -1을 의미,  
이와 같이 서로 다른 두 종류의 데이터를 구분하는 인공지능 모델을  
만들어봅시다.





