Deep learning 2

1871005 강예준

https://youtu.be/YeFPJ73PWmA

한성대학교 HANSUNG UNIVERSITY

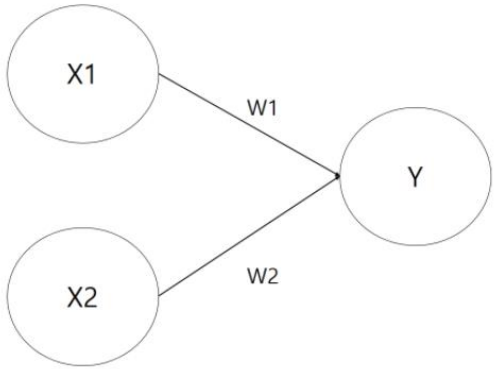
CryptoCraft LAB
https://crypto.modoo.at

Contents

CryptoCraft LAB

Perceptron

- **퍼셉트론**
  - 딥러닝의 기원이 되는 알고리즘
  - 퍼셉트론은 다수의 신호(0,1)를 입력으로 받아 하나의 신호를 출력
  - 매개변수 값을 사람이 **직접** 정해야 함
  - 활성화 함수로 **계단 함수**를 사용

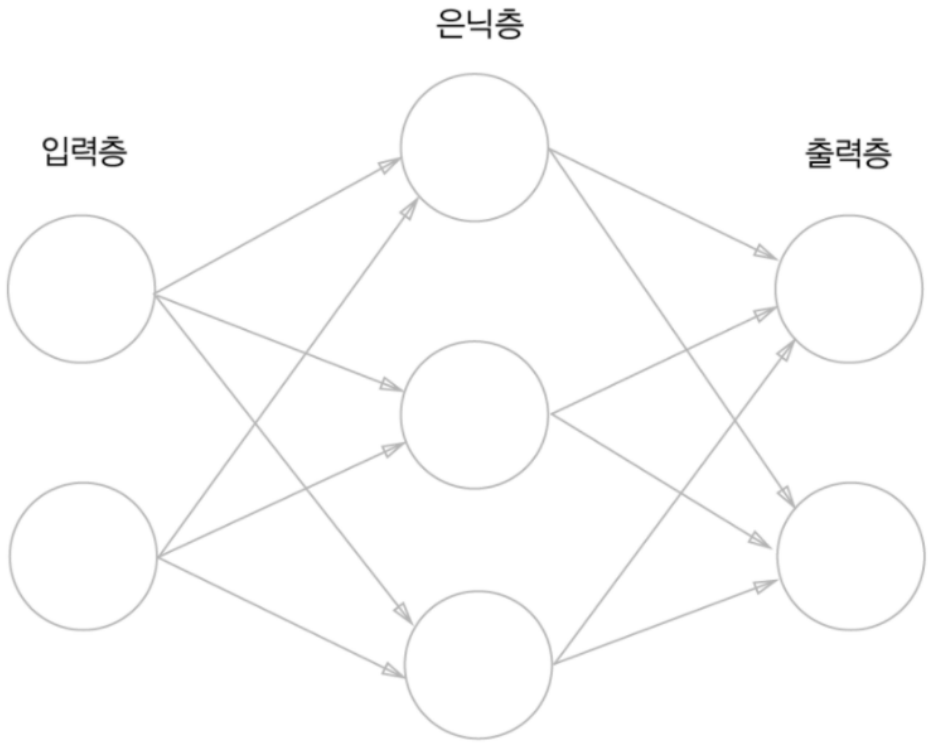


입력이 2개인 퍼셉트론

$$y = \begin{cases} 0 & (w_1 x_1 + w_2 x_2 \leq \theta) \\ 1 & (w_1 x_1 + w_2 x_2 > \theta) \end{cases}$$

Neural Network

- **신경망**
  - 활성화 함수로 계단함수 이외의 함수를 사용
  - 가중치 매개변수의 적절한 값을 데이터로부터 **자동으로** 학습



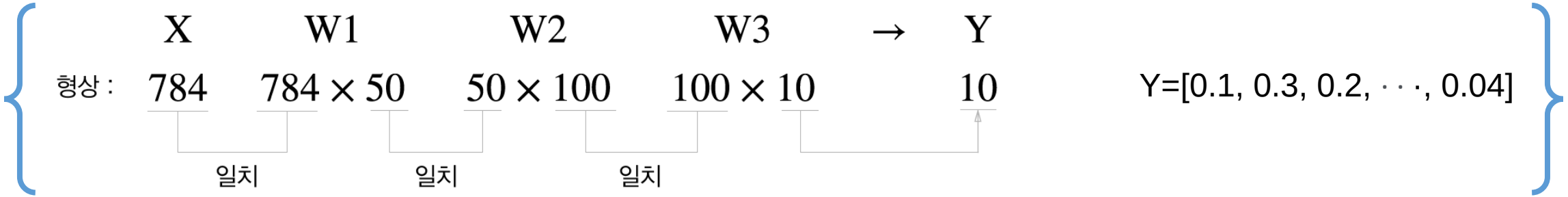
Classification & Regression

- **분류 (Classification)**
  - 미리 정의된, 가능성이 있는 여러 클래스 레이블 중 하나를 예측
  - 이중 분류 (binary classification)
  - 다중 분류 (multiclass classification)

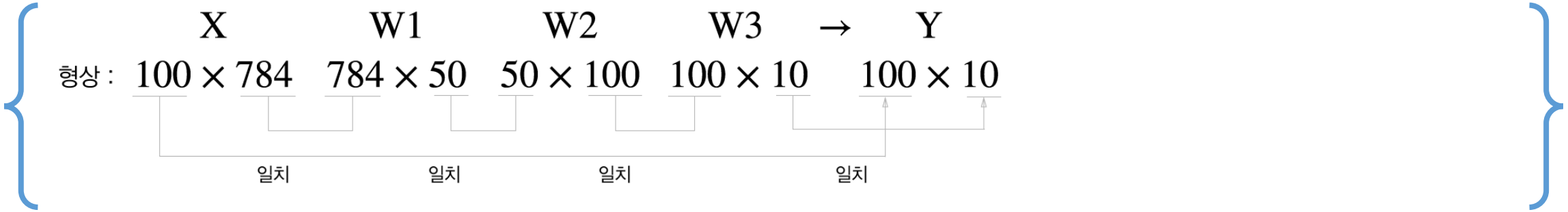
- **회귀 (Regression)**
  - 입력 데이터에서 연속적인 수치를 예측하는 문제

Batch Processing

- **배치처리**
  - **이미지 1장 처리**



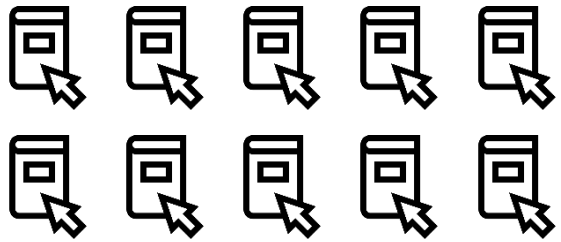
  - **이미지 100장 처리**



Batch Processing

- 이미지 1장당 처리 시간을 대폭 감소
  - 수치 계산 라이브러리
    - 큰 배열을 효율적으로 처리할 수 있도록 고도로 **최적화**
  - 버스에 부하 **감소**
    - 데이터를 읽는 횟수가 줄어 CPU나 GPU로 순수 계산을 수행하는 비율이 높아짐

Ex) 책을 가져오는 횟수가 줄어 내 머리로 순수 공부를 할 수 있는 시간이 높아짐



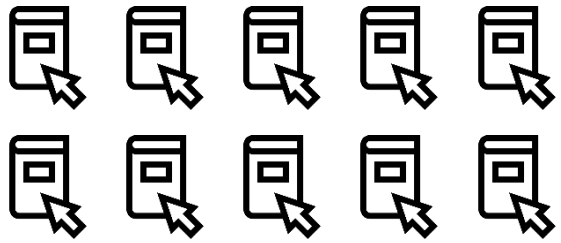
학술정보관

우촌관

Batch Processing

- 이미지 1장당 처리 시간을 대폭 감소
  - 수치 계산 라이브러리
    - 큰 배열을 효율적으로 처리할 수 있도록 고도로 **최적화**
  - 버스에 부하 **감소**
    - 데이터를 읽는 횟수가 줄어 CPU나 GPU로 순수 계산을 수행하는 비율이 높아짐

  Ex) 책을 가져오는 횟수가 줄어 내 머리로 순수 공부를 할 수 있는 시간이 높아짐



학술정보관

우촌관

Loss Function

- **학습**
  - 훈련 데이터로부터 가중치 매개변수의 최적값을 자동으로 획득하는 것
  - 손실함수의 결괏값을 작게 만드는 가중치 매개변수를 찾는 것이 학습의 목표
- **손실함수**
  - 신경망이 학습에서 신경망 성능이 얼마나 **나쁜지** 나타내는 지표
  - 오차제곱합, 교차 엔트로피 오차

Loss Function

- **오차제곱합 (sum of squares for errror, SSE)**
  - 가장 많이 쓰이는 손실함수

$$E = \frac{1}{2}\sum_{k}(y_k - t_k)^2$$

y_k : 신경망의 출력 (신경망이 추정한 값)
t_k : 정답레이블
k : 데이터의 차원 수

```
# 예시

# 정답이 2일 경우를 원-핫 인코딩으로 표현한 정답
t = [0,0,1,0,0,0,0,0]

# 예1:  2일 확률이 가장 높다고 추정(0.6)
y1 = [0.1, 0.05, 0.6, 0.0, 0.05, 0.1, 0.0, 0.0]
sum_square_error(np.array(y1), np.array(t))
# 결과 : 0.09250000000000003

# 예2 : 7일 확률이 가장 높다고 추정 (0.6)
y2 = [0.1, 0.05, 0.1, 0.0, 0.05, 0.6, 0.0, 0.0]
sum_square_error(np.array(y2), np.array(t))
# 결과 : 0.5925
```

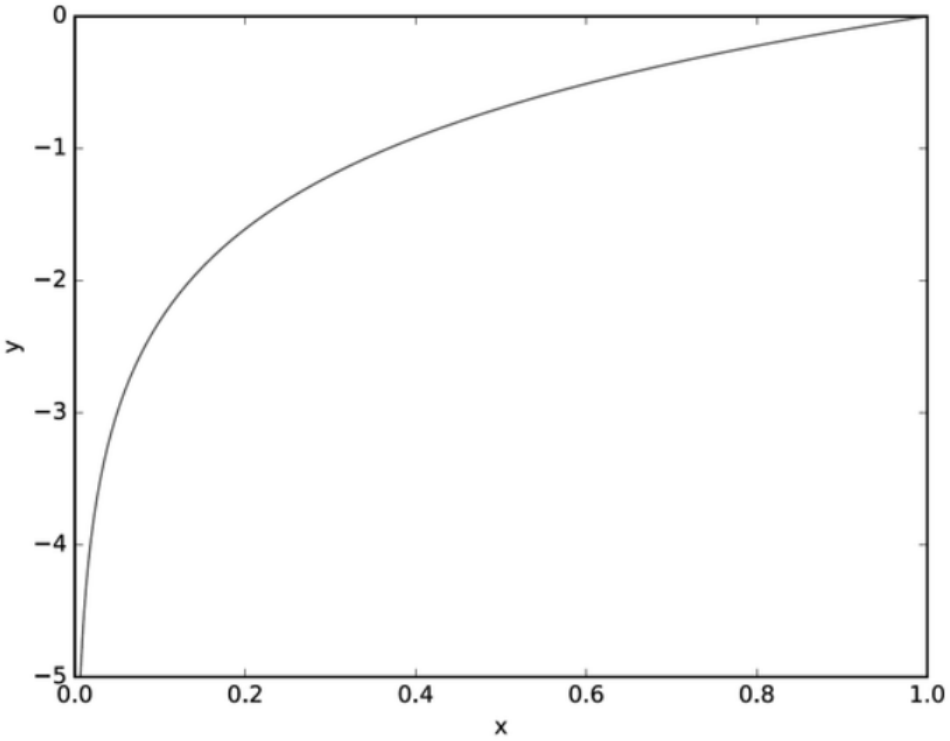
Loss Function

- **교차 엔트로피 오차 (cross entropy error, CEE)**
  - 실제로 계산 할 때는 마이너스 무한대가 되지 않도록 아주 작은 값을 더해줌

$$E = -\sum_k t_k \log y_k$$

y_k : 신경망의 출력 (신경망이 추정한 값)
t_k : 정답레이블
k : 데이터의 차원 수

```
def cross_entropy_error(y,t):
    delta = 1e-7
    return -np.sum(t*np.log(y+delta))
```



자연로그 y = logx의 그래프

Loss Function

- 정확도가 아닌 손실함수를 사용하는 이유
  - 신경망 학습 과정
    1. 손실함수의 값을 가능한 작게 하는 매개변수 값을 탐색
    2. 이때 매개변수의 미분을 계산
    3. 이 미분값을 단서로 매개변수의 값을 갱신
    4. 반복
  - 가중치 매개변수의 손실함수의 미분
    - 가중치 매개변수의 값을 아주 조금 변화 시켰을 때, 손실함수의 변화

음수 : 가중치 매개변수를 양의 방향으로 변화 ➡ 손실함수의 값 **감소**

양수 : 가중치 매개변수를 음의 방향으로 변화 ➡ 손실함수의 값 **감소**

0 : 가중치 매개변수를 어느 쪽으로 움직이든 손실함수의 값은 그대로이기 때문에 가중치 매개변수의 **갱신 중단**

Q & A