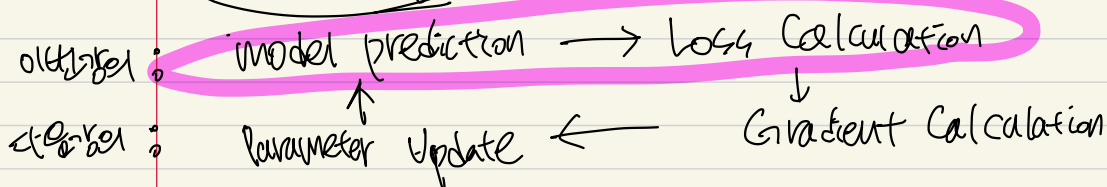


- Part 2 데이터 사이언스 필수 수학 (기초)
- Ch01 딥러닝 네트워크의 연산

Ch10-01 Introduction

- Training Neural Networks

model training



주요 특징 : 2층 perceptron을 변형한 구조, CNN의 가장 classic 한 모델.
(LeNet)

+ hidden layer를 많이 만들었는지, 어떤 연산이 이루어지는지.

forward : AN → Dense → Sigmoid / softmax → Loss → Con → Pooling → CNN

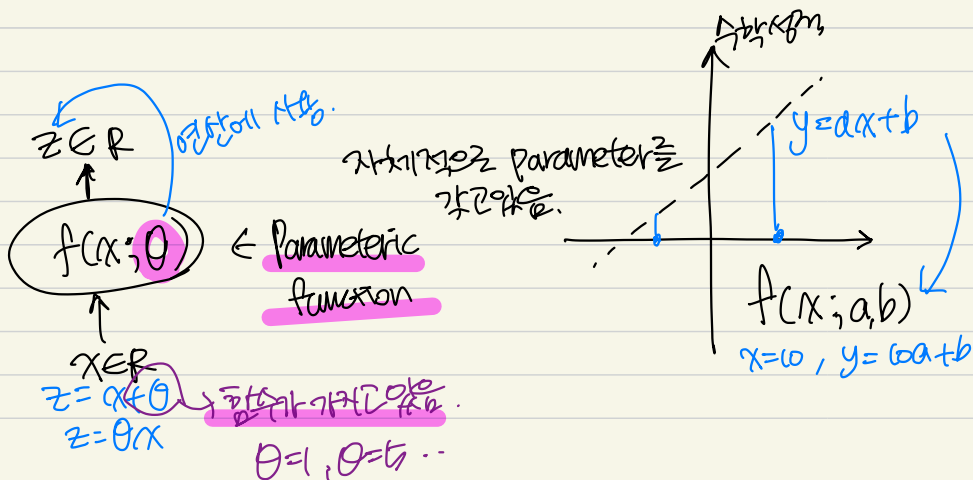
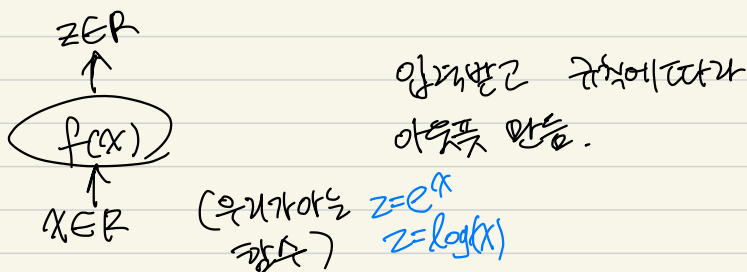
CH00-02 Colaboratory

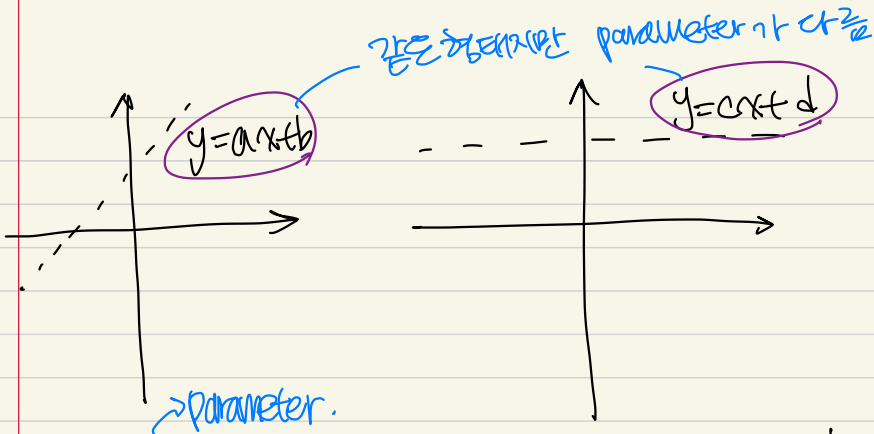
Backend를 가리지 제공.
간단한 프로그래밍 가능.

CH01-01. [이론] Parametric Functions and Datasets.

lecture.1 Artificial Neurons.

- Parametric Functions.





$f(x; \theta)$ → 이 때 θ 가 바뀌면 이에 다른 함수가 된다.
 * → θ 가 바뀌면 연산자 자체가 달라짐.

- ① Parameter를 바꿔서 얻는 함수를 다룬다.
- ② θ 가 다른 숫자가 아님. 바뀔만... 전에는 실수였다.
- ③ x (입력값)이 2차원 행렬이나, 벡터, array...

• Tensor 개념.

$\begin{pmatrix} \text{scalar} \\ \text{vector} \\ \text{matrix} \end{pmatrix} \rightarrow \text{Tensor}$
 general

• Zeroth order tensor (0차원) : scalar

$$a, b \in \mathbb{R}$$

$$a + b : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} : (5, 10) \in \mathbb{R} \times \mathbb{R}$$

$$5 + 10 \rightarrow 15$$

$$a \cdot b : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$$

Cartesian Product $\mathbb{R} \times \mathbb{R}$

$$\downarrow$$

$$15 \in \mathbb{R}$$

$$\mathbb{R}^2$$

$$= \mathbb{R} \times \mathbb{R}$$

$$B = \sum 0, 1$$

$$B \times B = \sum (0, 0), (0, 1), (1, 0), (1, 1)$$

- First-Order Tensor (Vector) (zeroth-order operation \equiv \otimes)

$$a \in \mathbb{R}$$

$$\vec{u}, \vec{v} \in \mathbb{R}^n \quad \vec{u} = \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix} \quad \vec{v} = \begin{pmatrix} 50 \\ 100 \\ 1000 \end{pmatrix}$$

$$a\vec{u} : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$2 \times \begin{pmatrix} 2 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 6 \\ 8 \\ 10 \end{pmatrix}$$

$$\mathbb{R} \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

$$\vec{u} + \vec{v} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$\vec{u} \odot \vec{v} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \Rightarrow \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} \odot \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix} = \begin{pmatrix} 20 \\ 60 \\ 120 \end{pmatrix}$$

$$\vec{u}^T \cdot \vec{v} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$$

$$\begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} \cdot (1 \ 2 \ 3) = 2 + 6 + 12 = 20$$

- Second-order Tensor (Matrix)

$$M \in \mathbb{R}^{2 \times 3}$$

$$M = \begin{pmatrix} 2 & 3 & 4 \\ 10 & 20 & 30 \end{pmatrix}$$

$$a \in \mathbb{R} \quad \vec{u} \in \mathbb{R}^n$$

$$M, N \in \mathbb{R}^{m \times n}, O \in \mathbb{R}^{n \times o}$$

$$aM : \mathbb{R} \times \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$$

$$M+N : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$$

$$M \odot N : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$$

$$M \cdot \vec{u} : \mathbb{R}^{m \times n} \times \mathbb{R}^n = \mathbb{R}^m$$

$$MO : \mathbb{R}^{m \times n} \times \mathbb{R}^{n \times o} \rightarrow \mathbb{R}^{m \times o}$$

$$\begin{array}{ccc} 2 \times \begin{pmatrix} 10 & 20 \\ 20 & 30 \end{pmatrix} & = & \begin{pmatrix} 20 & 40 \\ 40 & 60 \end{pmatrix} \\ \downarrow & & \downarrow \\ \mathbb{R} & & \mathbb{R}^{2 \times 2} \end{array}$$

• Third-Order Tensor Operations.

$$\begin{array}{c} \mathbb{R}^{28 \times 28 \times 1} \\ \text{(MNIST)} \\ \boxed{\mathbb{R}} \boxed{\mathbb{R}} \boxed{\mathbb{R}} \end{array}$$

also MINER $\mathbb{R}^{m \times n \times o}$

$$AM: \mathbb{R} \times \mathbb{R}^{m \times n \times o} \rightarrow \mathbb{R}^{m \times n \times o}$$

$$MTN: \mathbb{R}^{m \times n \times o} \times \mathbb{R}^{m \times n \times o} \rightarrow \mathbb{R}^{m \times n \times o}$$

$$MON: \mathbb{R}^{m \times n \times o} \times \mathbb{R}^{m \times n \times o} \rightarrow \mathbb{R}^{m \times n \times o}$$

- Dataset(X Data)

$$\vec{x}^T = (x_1, x_2, x_3, \dots, x_{l_i}) : \text{각각의 row vector를 표현}$$

$$\text{Dataset} \leftarrow \left(\begin{array}{c} \text{공통} \\ \text{데이터} \end{array} \right) \begin{array}{c} \text{각각의 row} \\ \text{데이터} \end{array} \begin{array}{c} \text{공통} \\ \text{데이터} \end{array}$$

$$\text{행 1} \quad \left(\vec{x}^{(1)} \right)^T = (x_1^{(1)}, x_2^{(1)}, \dots, x_{l_i}^{(1)})$$

$$\text{행 2} \quad \left(\vec{x}^{(2)} \right)^T = (x_1^{(2)}, x_2^{(2)}, \dots, x_{l_i}^{(2)})$$

$$\vdots$$

$$\text{행 } N \quad \left(\vec{x}^{(N)} \right)^T = (x_1^{(N)}, x_2^{(N)}, \dots, x_{l_i}^{(N)})$$

length of input
→ l_i

$$X^T = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(N)} \end{pmatrix} \in \mathbb{R}^{N \times 1}$$

$$\vec{x} = (x_1, x_2, \dots, x_{n_x})$$

$$X^T = \begin{pmatrix} \leftarrow \begin{pmatrix} x^{(1)} \end{pmatrix}^T \rightarrow \\ \leftarrow \begin{pmatrix} x^{(2)} \end{pmatrix}^T \rightarrow \\ \vdots \\ \leftarrow \begin{pmatrix} x^{(N)} \end{pmatrix}^T \rightarrow \end{pmatrix} \in \mathbb{R}^{N \times n_x}$$

$$= \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_{n_x}^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_{n_x}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \dots & x_{n_x}^{(N)} \end{pmatrix} \in \mathbb{R}^{N \times n_x}$$

now vector $\vec{x} \rightarrow$ Matrix X

$$(\vec{x})^T = (x_1, x_2, x_3, x_4)$$

$$X^T = \begin{pmatrix} \leftarrow \begin{pmatrix} x^{(1)} \end{pmatrix}^T \rightarrow \\ \vdots \\ \leftarrow \begin{pmatrix} x^{(N)} \end{pmatrix}^T \rightarrow \end{pmatrix} = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_{n_x}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \dots & x_{n_x}^{(N)} \end{pmatrix} \in \mathbb{R}^{N \times n_x}$$

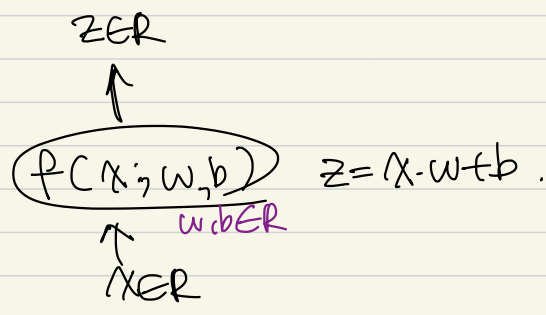
$$X^T = (x_1, x_2, x_3, \dots, x_{n_x})$$

$$X^T = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_{n_x}^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_{n_x}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \dots & x_{n_x}^{(N)} \end{pmatrix} \in \mathbb{R}^{N \times n_x}$$

CH01-02 Artificial Neuron

- Affine Function with one Feature.

- Weighted Sum
 $z = x \cdot w$
- Affine Transformation
 $z = x \cdot w + b$



- Affine Functions with n Features

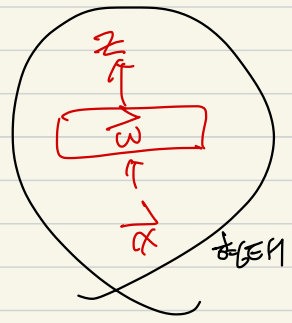
- Weighted Sum
 $z = w_1 x_1 + w_2 x_2 + \dots + w_n x_n = \begin{pmatrix} w \end{pmatrix}^T \vec{x} = \begin{pmatrix} \vec{x} \end{pmatrix} \begin{pmatrix} w \end{pmatrix}$

$w: \mathbb{R}^n$

가중치 합

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

$$\vec{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$



- Affine Transformation
 $z = \begin{pmatrix} \vec{x} \end{pmatrix}^T \vec{w} + b$

$z \in \mathbb{R}$

$$+ \begin{pmatrix} \vec{x} \end{pmatrix}^T \vec{w} + b$$

$$\begin{pmatrix} \vec{x} \end{pmatrix}^T = (x_1, x_2, \dots, x_n) \in \mathbb{R}^{1 \times n}$$

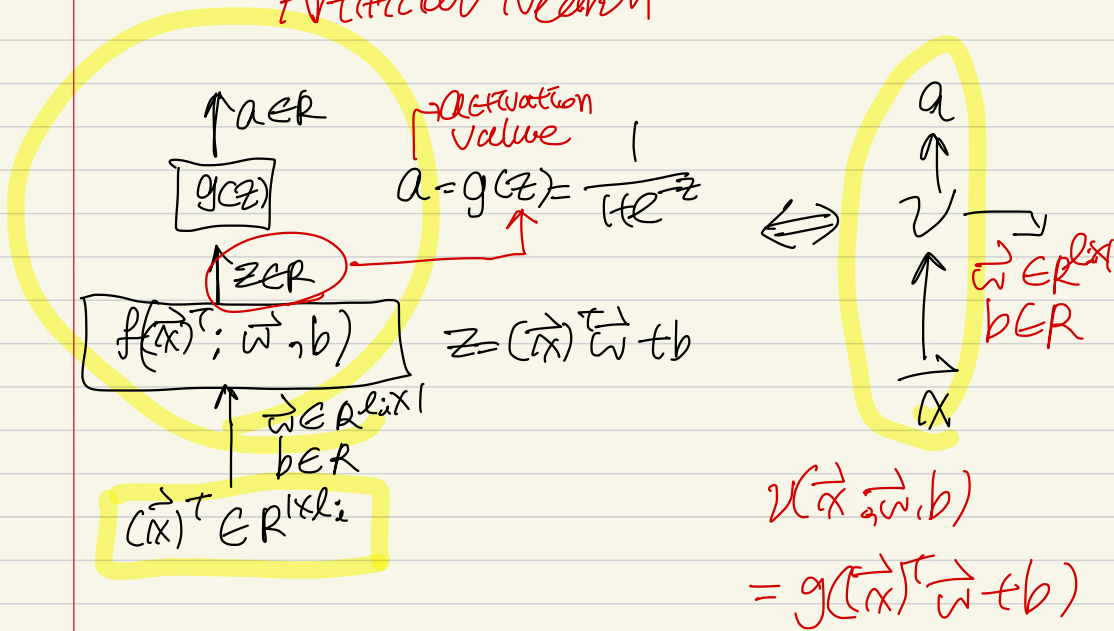
$\vec{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \in \mathbb{R}^{n \times 1}, b \in \mathbb{R}$
 $\star 1H \text{ } \mathbb{R}^n \text{ } \mathbb{R}^n$
 $\star 1H \text{ } \mathbb{R}^n$

• Activation Function

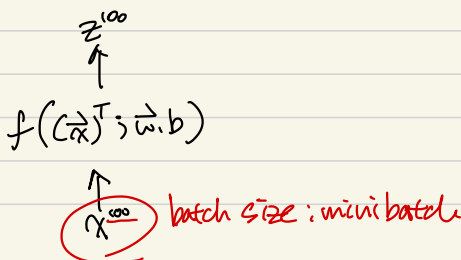
sigmoid $g(x) = \sigma(x) = \frac{1}{1+e^{-x}}$
 Tanh $g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
 ReLU $g(x) = \text{ReLU}(x) = \max(0, x)$

affine | activation step.

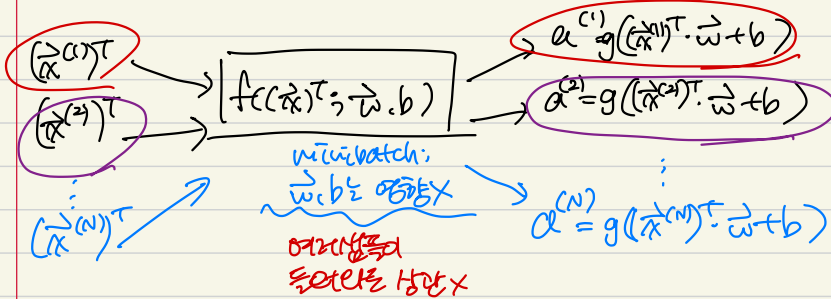
Artificial Neuron



— Mini-batch in Artificial Neurons



- Minibatch in AN



★ \vec{w} 는 가변치 않음.

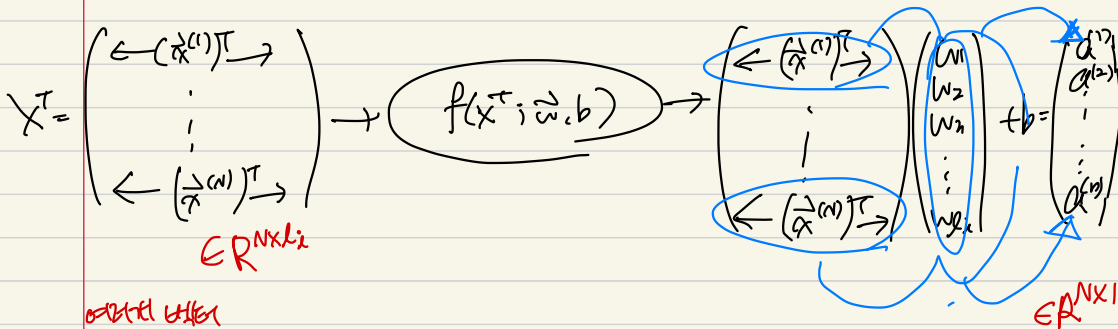
(minibatch가 들어오면 \vec{w} 는 바뀌지 않기 때문)

$$f(1; 0) = 1 + 0$$

$$f(2; 0) = 2 + 0$$

$$f(10; 0) = 10 + 0$$

$\rightarrow 0$ 는 가변치 않음 (안변함)



오랫동안

이제 Matrix

이제 \rightarrow minibatch

\therefore minibatch input \rightarrow Weight/Bias \rightarrow Affine function \rightarrow Activation Function

$$X^T \in \mathbb{R}^{N \times L_i}$$

$$Z = X^T w + b$$

$$Z \in \mathbb{R}^{N \times 1}$$

$$A \in \mathbb{R}^{N \times 1}$$

$$\begin{array}{c}
 \uparrow A \in \mathbb{R}^{N \times 1} \\
 \textcircled{g(z)} \\
 \uparrow z \in \mathbb{R}^{N \times 1} \\
 \textcircled{f(x^T; \vec{w}, b)} \\
 \uparrow \vec{w} \in \mathbb{R}^{N \times 1}, b \in \mathbb{R} \\
 x^T \in \mathbb{R}^{N \times 1}
 \end{array}
 \quad x^T \cdot \vec{w} + b$$

CHOL-03 [한글] Artificial-Neurons