KU KONKUK UNIVERSITY

Artificial Neural Network

자연어처리연구실

본 챕터의 일부는 김성훈 박사님의 "모두를 위한 머신러닝/딥러닝 강의 (https://hunkim.github.io/ml/)" 자료를 참고하여 편집하였습니다.

ANN (Artificial Neural Networks)

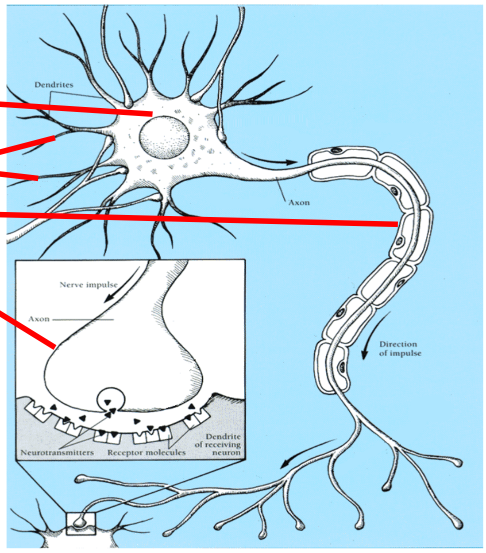
- 수학적 논리학이 아닌 <u>인간의 두뇌</u>를 모방하여 수많은 간단한 처리기들(뉴런)의 네트워크를 통해 문제를 해결하는 기계학습 모델

Cell body

Dendrites(수상돌기)

Axon(축색돌기)

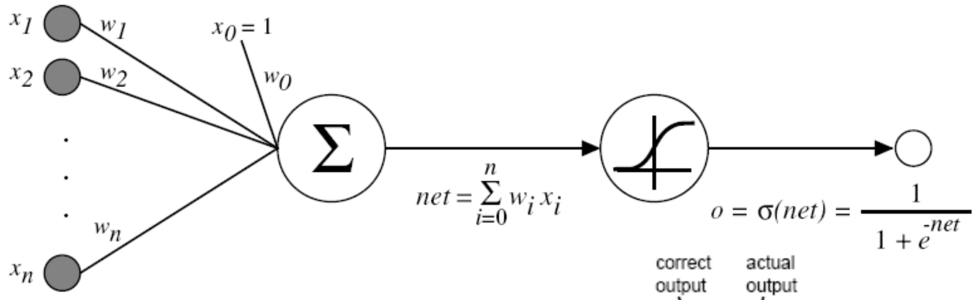
Synaptic terminals(시냅스)

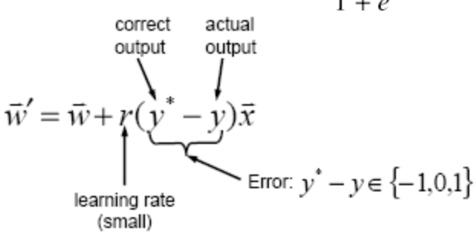


ANN (Artificial Neural Networks)

- 수학적 논리학이 아닌 인간의 두뇌를 <u>모방</u>하여 수많은 간단한 처리기들(뉴런)의 네트워크를 통해 문제를 해결하는 기계학습 모델

Dendrites(수상돌기) Axon(축색돌기) Synaptic terminals(시냅스)





학습(델타룰): 정답과 출력을 비교하여 그 차이를 가중치 조정에 반영

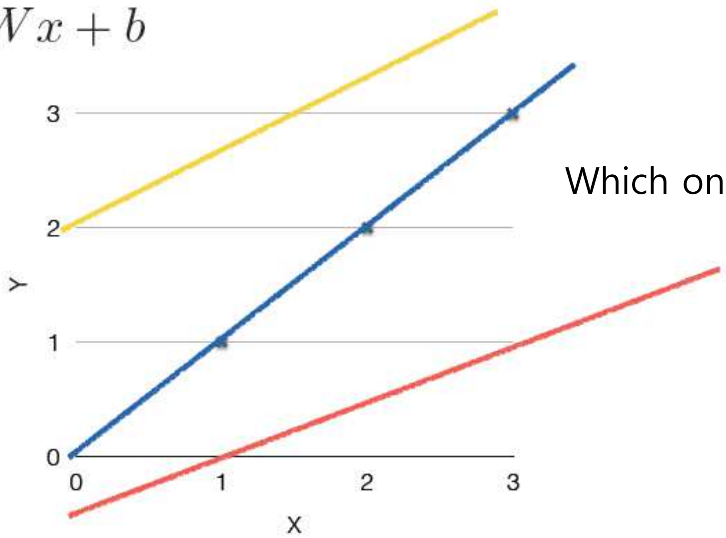
Brief ANN History

- Frank Rosenblatt, 1957
  - Single-layer perceptron
- Minsky & Papert 1969
  - ANN is a linear function (1st winter season)
- Rumelhart, Hinton & Williams, 1986
  - Back propagation algorithm for Multi-layer perceptron
  - Vanishing gradient problem! (2nd winter season)
- Geoffrey Hinton, 2009 → Yoshua Bengio, Andrew Ng, Ian Goodfellow
  - New activation function, ReLU, for deep neural networks
  - Drop-out for increasing robustness

REMIND

Linear Hypothesis

$$H(x) = Wx + b$$



Which one is better?

REMIND

Matrix Representation

$$\begin{bmatrix} w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

$$H(X) = WX + b$$

With b vector

$$\begin{bmatrix} b & w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} 1 \\ x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} b \times 1 + w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

$$H(X) = WX$$

Without b vector

$$H(X) = W^T X$$

Transpose representation

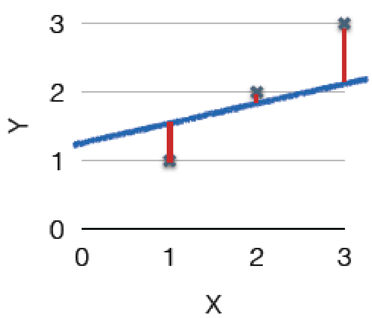
REMIND

Cost Function

$$\frac{(H(x^{(1)}) - y^{(1)})^2 + (H(x^{(2)}) - y^{(2)})^2 + (H(x^{(3)}) - y^{(3)})^2}{3}$$

$$cost = \frac{1}{m} \sum_{i=1}^{m} (H(x^{(i)}) - y^{(i)})^2$$

$$H(x) = Wx + b$$



Our goal? $\underset{W,b}{\text{minimize}}\, cost(W, b)$

Cost function을 최소로 하는hypothesis가 무엇일까?

REMIND

Formal Definition of Gradient Decent

$$cost(W) = \frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$cost(W) = \frac{1}{2m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{\partial}{\partial W} cost(W)$$

$$W := W - \alpha \frac{\partial}{\partial W} \frac{1}{2m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha \frac{1}{2m}\sum_{i=1}^{m}2(Wx^{(i)} - y^{(i)})x^{(i)}$$

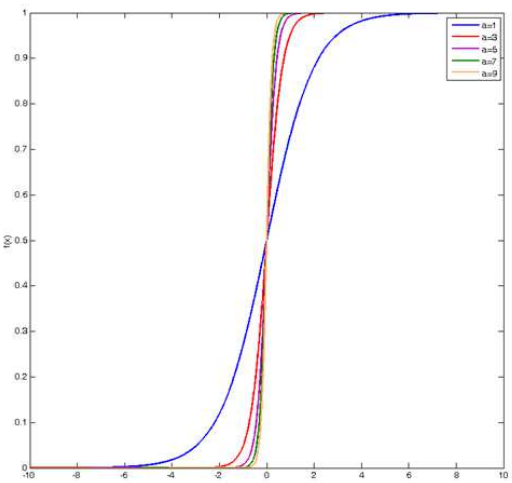
$$W := W - \alpha \frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})x^{(i)}$$

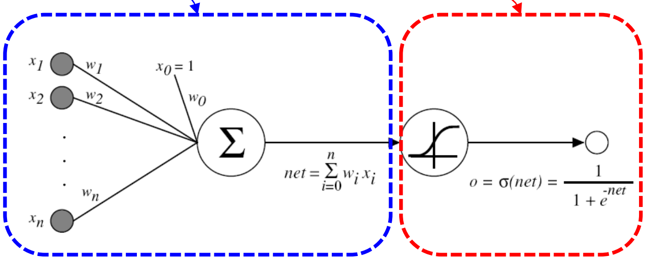
REMIND

Logistic Hypothesis

$$H(x) = Wx + b \quad \Rightarrow \quad g(z) = \frac{1}{(1 + e^{-z})}$$

WHY?
0과 1 사이 값으로 변환





Architecture of ANN

REMIND

Cost Function

$$Cost(W) = \frac{1}{m}\sum c(H(x), y)$$

$$c(H(x), y) = \begin{cases} -\,log\big(H(x)\big) & : y = 1 \\ -\,log\big(1 - H(x)\big) & : y = 0 \end{cases}$$

$$c(H(x), y) = -ylog\big(H(x)\big) - (1 - y)log(1 - H(x))$$

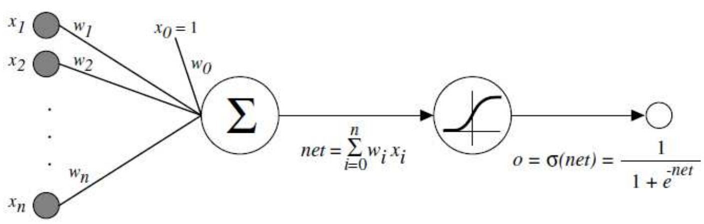
Minimize Cost → Gradient decent algorithm

$$Cost(w) = -\frac{1}{m}\sum ylog\big(H(x)\big) + (1 - y)log(1 - H(x))$$

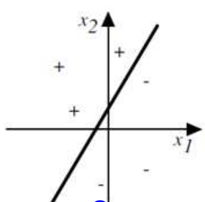
$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$

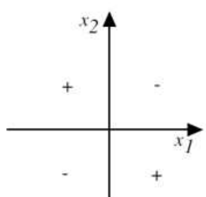
퍼셉트론 (Perceptron)

- 구조



- 결정 공간 (decision surface)



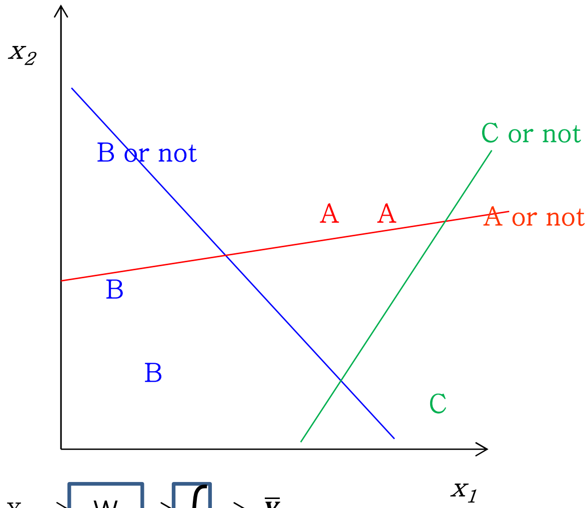


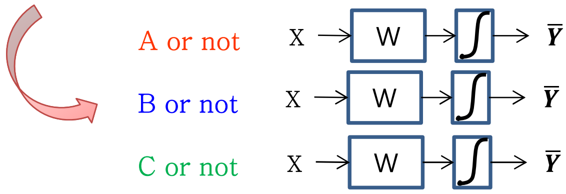
Not linearly separable!

One perceptron → Binomial Classifier

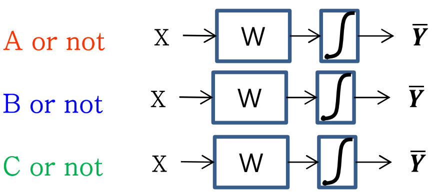
Multinomial Classification

x1 (hours)	x2 (attendance)	y (grade)
10	5	A
9	5	A
3	2	B
2	4	B
11	1	C





Multinomial Classification



$$\begin{bmatrix} w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

$$\begin{bmatrix} w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

$$\begin{bmatrix} w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \boxed{\int} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

New Cost Function for Multinomial Classification

Cross Entropy

$$S(y) = \bar{Y} \qquad\qquad L = Y$$

$$\begin{bmatrix} 0.7 \\ 0.2 \\ 0.1 \end{bmatrix} \qquad D(S, L) = -\sum_i L_i log(S_i) \qquad \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

[Example]

$$Y: \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\overset{\bar{Y}}{\begin{bmatrix} 0 \\ 1 \end{bmatrix}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot -\log\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot \begin{bmatrix} \infty \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0$$

$$\overset{\bar{Y}}{\begin{bmatrix} 1 \\ 0 \end{bmatrix}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot -\log\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot \begin{bmatrix} 0 \\ \infty \end{bmatrix} = \begin{bmatrix} 0 \\ \infty \end{bmatrix} = \infty$$

SoftMax

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \boxed{\int} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Sigmoid 함수에 의해서 각 각 0~1 사이 값이 출력됨

Scores

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

SoftMax

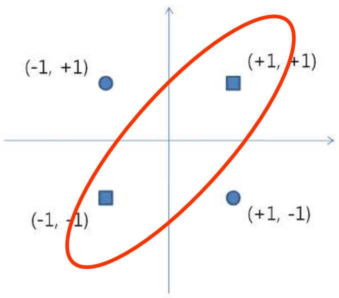
One-hot representation

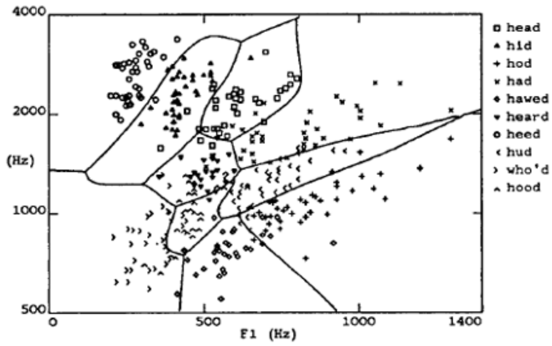
$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \xleftarrow{\text{argmax}} \begin{bmatrix} S(y_1) \\ S(y_2) \\ S(y_3) \end{bmatrix} = \begin{bmatrix} 0.7 \\ 0.2 \\ 0.1 \end{bmatrix}$$

Probabilities (Distributed representation)

Non-linear Problems

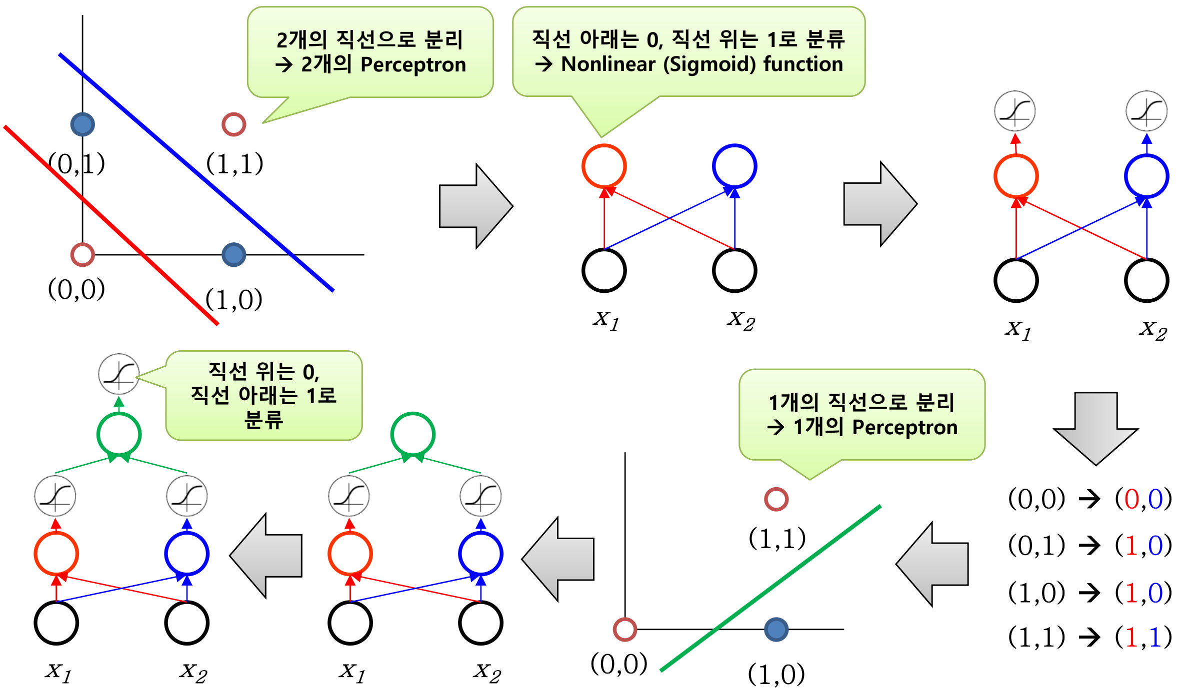
- 비선형 분리 문제





- 비선형 분리 문제 → 선형 분리 문제
  - SVM 커널 함수(kernel function)
  - Single-layer perceptron → Multi-layer perceptron

XOR in Multi-layer Perceptron



질의응답

Q&A

Homepage: http://nlp.konkuk.ac.kr
E-mail: nlpdrkim@konkuk.ac.kr