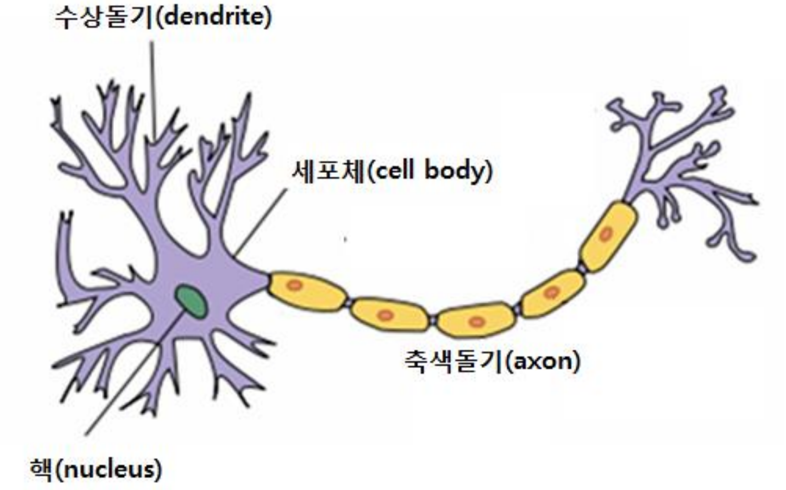
기계 학습

Part III

9. 신경망

❖ **신경망(neural network, artificial neural network)**

- 인간 두뇌에 대한 계산적 모델을 통해 인공지능을 구현하려는 분야
- **신경세포** (neuron)



신경세포 8.6×10^{10} 개

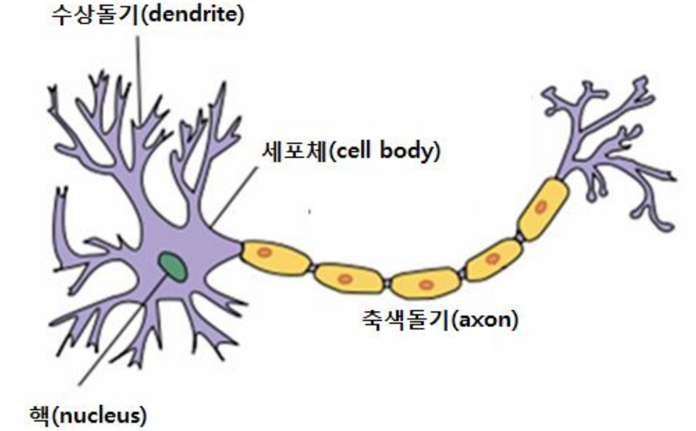
신경연접 1.5×10^{14} 개

- **수상돌기(**樹狀突起, dendrite) : 다른 신경세포의 축색돌기와 연결되어 **전기화학적 신호**를 받아들이는 부위
- **축색돌기**(軸索突起, axon) : 수신한 전기화학적 신호의 합성결과 값이 특정 임계값이 이상이면 신호를 내보는 부위.
- **신경연접**(神經連接, synapse) : 수상돌기와 축색돌기 연결 부위
  - 전달되는 신호의 증폭 또는 감쇄

9.1 퍼셉트론

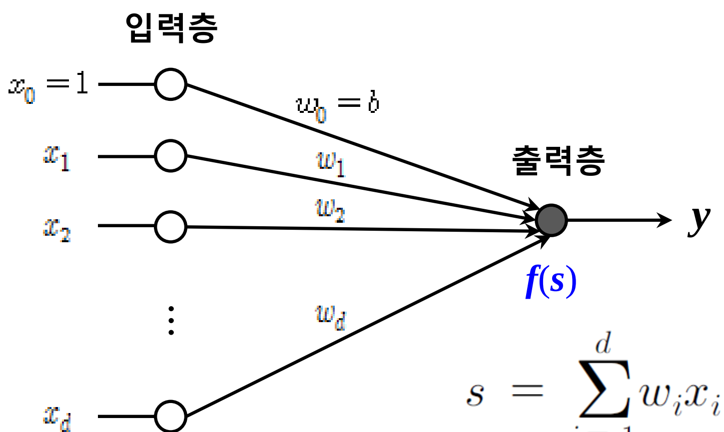
❖ 신경세포의 계산 모델

- 신경세포

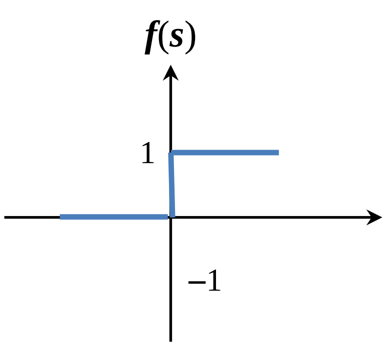


McClulloch & Pitts, 1943
신경세포의 계산 모델

- **퍼셉트론**(Perceptron)
  - 로젠블랏(Rosenblatt, 1957)이 제안한 학습가능한 신경망 모델



$$s = \sum_{i=1}^{d} w_i x_i + b = \sum_{i=0}^{d} w_i x_i$$

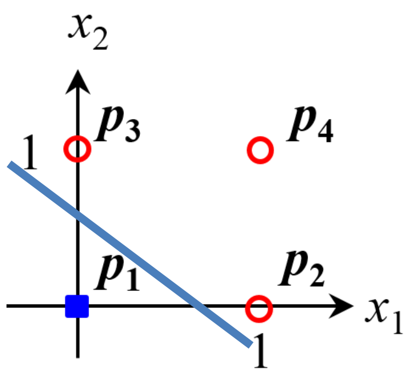


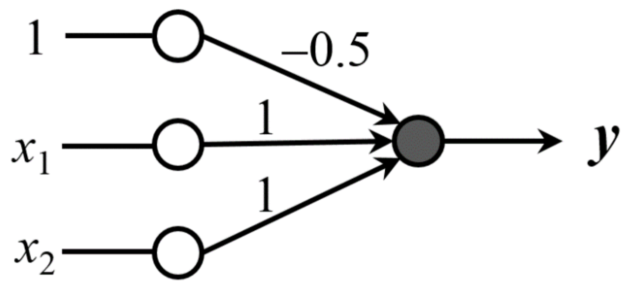
전달함수

퍼셉트론

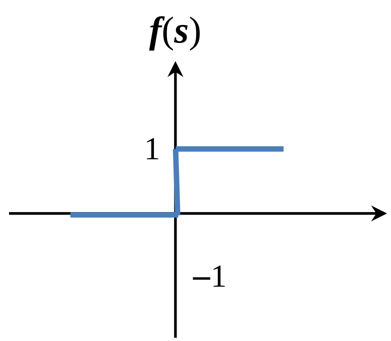
❖ 퍼셉트론(Perceptron)

- OR 연산을 수행하는 퍼셉트론





$$y = f(s) = f(\sum_{i=1}^{2} w_i x_i + b) = f(\boldsymbol{w}^\top \boldsymbol{x}) = x_1 + x_2 - 0.5$$



p_1 (0,0): $y = 0$

p_2 (1,0): $y = 1$

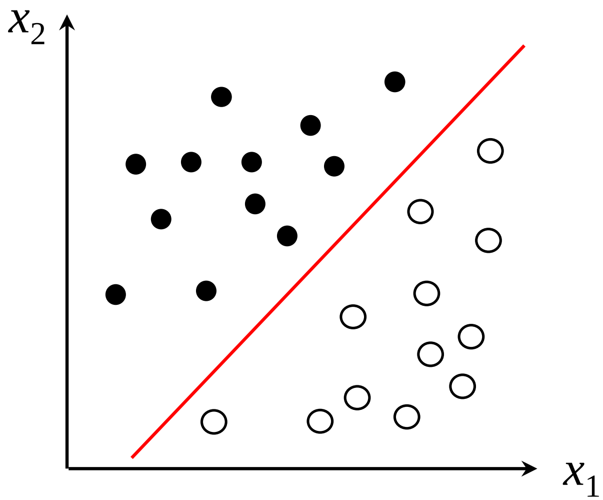
p_3 (0,1): $y = 1$

p_4 (1,1): $y = 1$

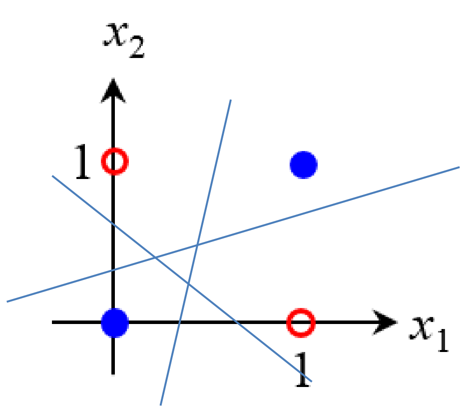
퍼셉트론

❖ **퍼셉트론**(Perceptron)

- **선형 분리가능 문제** (linearly separable problem)



- **선형 분리불가 문제**(linearly inseparable problem)
  - **XOR**(exclusive OR) **문제**

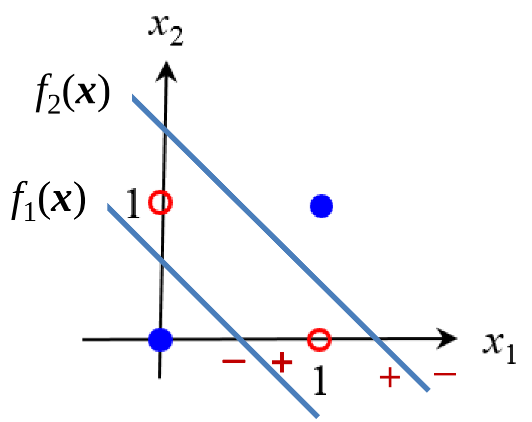


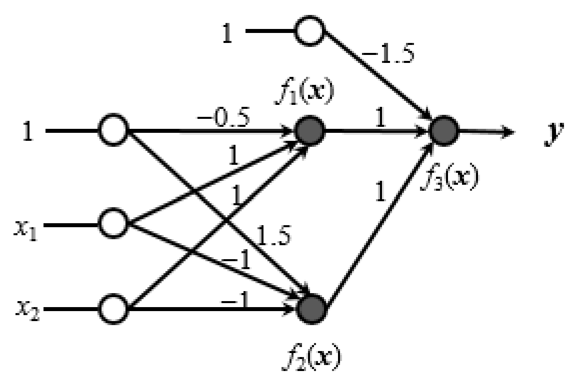
(0,0) : 0
(0,1) : 1
(1,0) : 1
(1,1) : 0

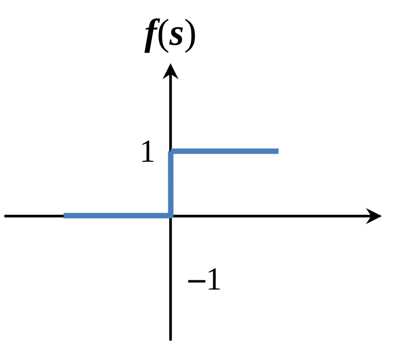
Minsky & Papert 1969

9.2 다층 퍼셉트론

- ❖ **다층 퍼셉트론**(multilayer Perceptron, **MLP**)
  - 여러 개의 퍼셉트론을 층 구조로 구성한 신경망 모델







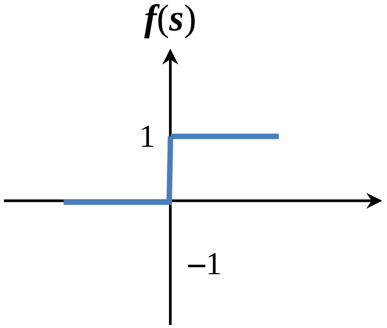
$$y = f(s) = f\left(\sum_{i=1}^{2} w_i x_i + b\right) = f(\boldsymbol{w}^\top \boldsymbol{x})$$

다층 퍼셉트론

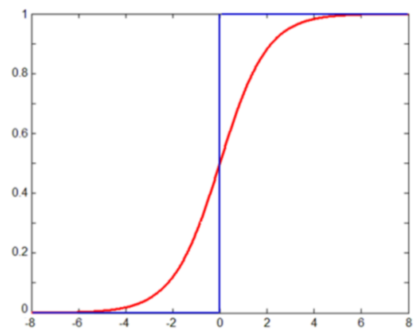
❖ **다층 퍼셉트론**(multilayer Perceptron, **MLP**) – cont.

- **다층 퍼셉트론의 학습**
  - 입력-출력 $(\boldsymbol{x}_i, y_i)$의 학습데이터에 대해서 출력값 $f(\boldsymbol{x}_i)$의 차이, **오차**(error)가 최소가 되도록 가중치 $\boldsymbol{w}$를 결정하는 것

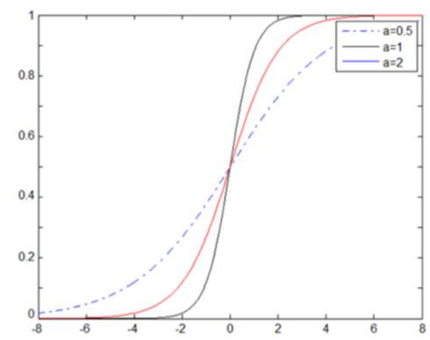
- **학습 가능한 다층 퍼셉트론**
  - 학습알고리즘 : **오차역전파** 알고리즘 발표 (Rumelhart&McClelland 1986)
  - 계단모양 전달함수를 미분가능한 시그모이드(sigmoid) 함수로 대체



계단 (step) 함수



시그모이드(sigmoid) 함수

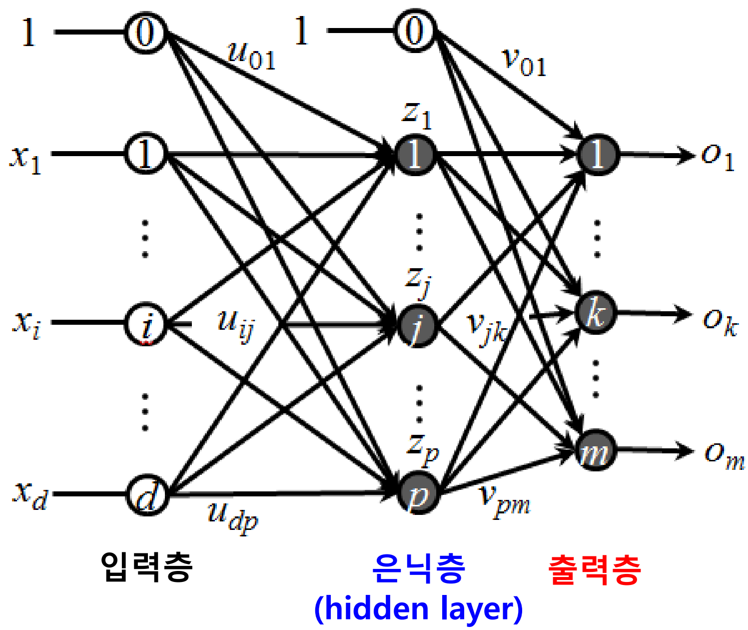


$$f(s) = \frac{1}{1+e^{-as}}$$

$$f'(s) = \frac{ae^{-as}}{(1+e^{-as})^2} = af(s)(1-f(s))$$

다층 퍼셉트론

❖ 다층 퍼셉트론 MLP의 동작



입력 $(x_1, x_2, \cdots, x_d)$

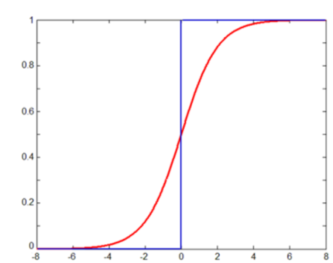
출력 $(y_1, y_2, \cdots, y_m)$

은닉층 $zsum_j = \sum_{i=1}^{d} u_{ij} x_i + u_{0j} \quad (1 \le j \le p)$

$$z_j = f(zsum_j)$$

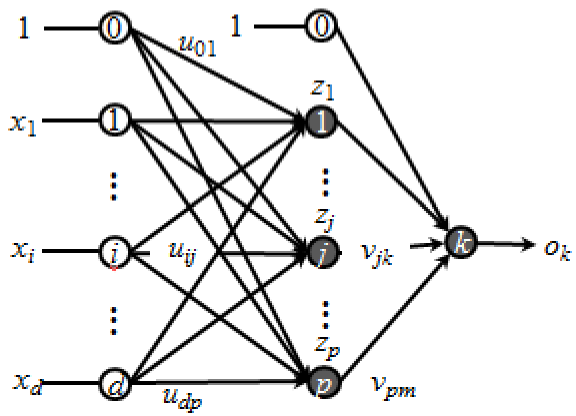
출력층 $osum_k = \sum_{j=1}^{p} v_{jk} z_k + v_{0k} \quad (1 \le k \le m)$

$$o_k = f(osum_k)$$



다층 퍼셉트론

❖ 다층 퍼셉트론 MLP의 학습



입력 : $(x_1, x_2, \ldots, x_d)$

기대 출력 : y_k

MLP 출력 : o_k

- **학습 목표**
  - **기대 출력**과 **MLP 출력**이 최대한 **비슷**해지도록 **가중치**를 **변경**하는 것

$$E = \frac{1}{2}(o_k - y_k)^2$$

  - **최대 경사법**(gradient descent method) 사용

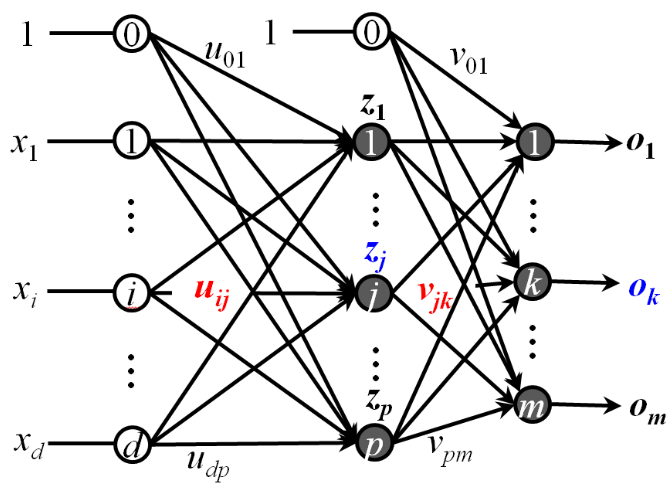
$$v_{jk}^{(t+1)} = v_{jk}^{(t)} - \eta \frac{\partial E}{\partial v_{jk}}$$

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} - \eta \frac{\partial E}{\partial u_{ij}}$$

다층 퍼셉트론

❖ 다층 퍼셉트론 MLP의 학습

- **오차 역전파 알고리즘**(Error back propagation algorithm, Backprop algorithm)



입력 $(x_1, x_2, \cdots, x_d)$

출력 $(y_1, y_2, \cdots, y_m)$

$$E = \frac{1}{2}\sum_{k=1}^{n}(o_k - y_k)^2$$ **오차함수**

$$\boldsymbol{v}^{(t+1)} = \boldsymbol{v}^{(t)} - \eta\frac{\partial E}{\partial \boldsymbol{v}}$$

$$\boldsymbol{u}^{(t+1)} = \boldsymbol{u}^{(t)} - \eta\frac{\partial E}{\partial \boldsymbol{u}}$$

$$osum_k = \sum_{j=1}^{p} v_{jk} z_k + v_{0k} \qquad (1 \le k \le m)$$

$$o_k = f(osum_k)$$

$$zsum_j = \sum_{i=1}^{d} u_{ij} x_i + u_{0j} \qquad (1 \le j \le p)$$

$$z_j = f(zsum_j)$$

$$\frac{\partial E}{\partial v_{jk}} = \frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial v_{jk}} = (o_k - y_k) f'(osum_k) z_j = \delta_k z_j$$

$$\frac{\partial E}{\partial u_{ij}} = \frac{\partial E}{\partial z_j}\frac{\partial z_j}{\partial u_{ij}} = \sum_{k=1}^{m}\frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial z_j} f'(zsum_j) x_i$$

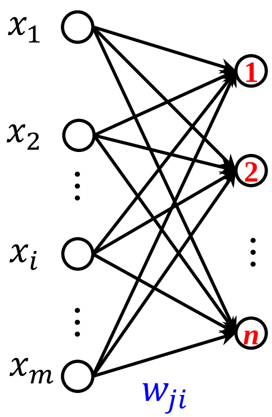
$$= \sum_{k=1}^{m}(o_k - y_k) f'(osum_k) v_{jk} f'(zsum_j) x_i$$

$$= \sum_{k=1}^{m}\delta_k v_{jk} f'(zsum_j) x_i$$

신경망

❖ **소프트맥스 층** (softmax layer)

- 최종 출력을 **분류 확률**(classification probability)로 변환하는 층
  - 출력의 합 = 1



$y_1 = p(Y = 1|X)$

$y_2 = p(Y = 2|X)$

$y_K = p(Y = K|X)$

$$y_k = \frac{e^{\sum_{j=1}^{m} w_{jk} x_j}}{\sum_{l=1}^{n} e^{\sum_{j=1}^{m} w_{jl} x_j}}$$

신경망

❖ 학습 데이터

$$D = \{(\boldsymbol{x}_1, \boldsymbol{t}_1), (\boldsymbol{x}_2, \boldsymbol{t}_2), \cdots, (\boldsymbol{x}_N, \boldsymbol{t}_N)\}$$

- i번째 데이터의 입력 : $\boldsymbol{x}_i$
- i번째 데이터의 입력 : $\boldsymbol{t}_i = (t_{i1}.t_{i2}, \dots, t_{iK})$ $t_{ij} \in \{0, 1\}$, $\sum_{j=1}^{K} t_{ij} = 1$
  one-hot 벡터 표현

❖ 학습 데이터 $(\boldsymbol{x}_i, \boldsymbol{t}_i)$의 조건부 확률

$$p(\boldsymbol{t}_i | \boldsymbol{x}_i, \boldsymbol{w}) = \prod_{k=1}^{K} y_k(\boldsymbol{x}_i, \boldsymbol{w})^{t_{ik}}$$

$\boldsymbol{w}$: 신경망의 동작을 결정하는 전체 가중치 벡터

❖ 전체 데이터 $\boldsymbol{D}$에 대한 가능도(likelihood)

$$p(D; \boldsymbol{w}) = \prod_{i=1}^{N} \prod_{k=1}^{K} y_k(\boldsymbol{x}_{\boldsymbol{i}}, \boldsymbol{w})^{t_{ik}}$$

신경망

- **최대 가능도 추정**(maximum likelihood estimation, MLE)
  - 데이터의 가능도를 최대로 하는 파라미터를 추정하는 것

$$\mathrm{argmax}_{\boldsymbol{w}} \prod_{i=1}^{N} \prod_{k=1}^{K} y_k(\boldsymbol{x_i}, \boldsymbol{w})^{t_{ik}}$$

  - 오차함수 $E(\boldsymbol{w})$: 음의 로그 가능도로 정의

$$E(\boldsymbol{w}) = -\log \prod_{i=1}^{N} \prod_{k=1}^{K} y_k(\boldsymbol{x_i}, \boldsymbol{w})^{t_{ik}} = -\sum_{i=1}^{N} \sum_{k=1}^{K} t_{ik} \log y_k(\boldsymbol{x_i}, \boldsymbol{w})$$

  - 오차함수 = $(t_{i1}, t_{i2}, \cdots, t_{iK})$와 $(y_1, y_2, \cdots, y_K)$에 대한 교차 엔트로피

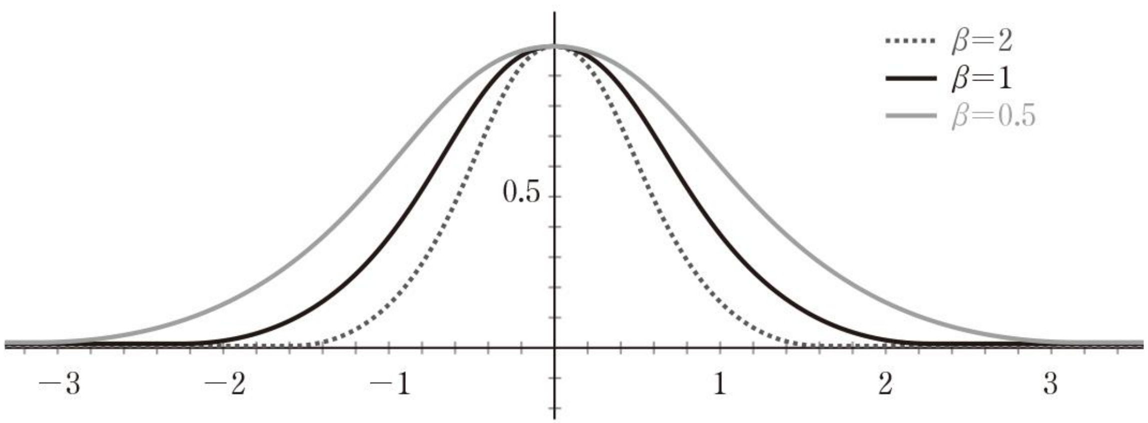
$$E(\boldsymbol{w}) = -\sum_{k=1}^{K} t_{ik} \log y_k(\boldsymbol{x_i}, \boldsymbol{w})$$

9.3 RBF 망

❖ **RBF(radial basis function) 함수**

- **기존 벡터 μ 와 입력 벡터 x 의 유사도를 측정하는 함수**
- 예.

$$\phi(\boldsymbol{x}, \boldsymbol{\mu}) = \exp(-\beta \|\boldsymbol{x} - \boldsymbol{\mu}\|^2)$$

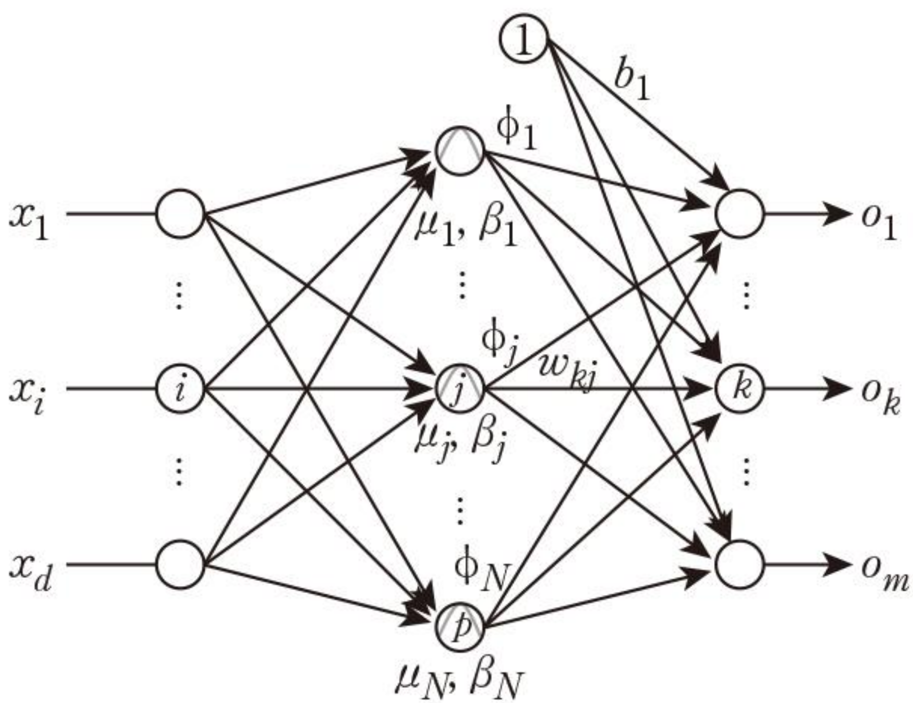


RBF 망

❖ **RBF 망 (RBF network)**

- 어떤 함수 $f_k(\boldsymbol{x})$를 다음과 같이 **RBF 함수들**의 **선형 결합 형태**로 근사시키는 모델

$$f_k(\boldsymbol{x}) \approx \sum_{i=1}^{N} w_{kj}\phi_i(\boldsymbol{x}, \boldsymbol{\mu}_i) + b_k$$



$$\phi_j = \exp(-\beta_j \|\boldsymbol{x} - \boldsymbol{\mu}_i\|^2)$$

$$o_k = \sum_{j=1}^{N} w_{kj}\phi_j + b_k$$

RBF 망

❖ **RBF 망의 학습**

- **오차 함수 $\boldsymbol{E}$**

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

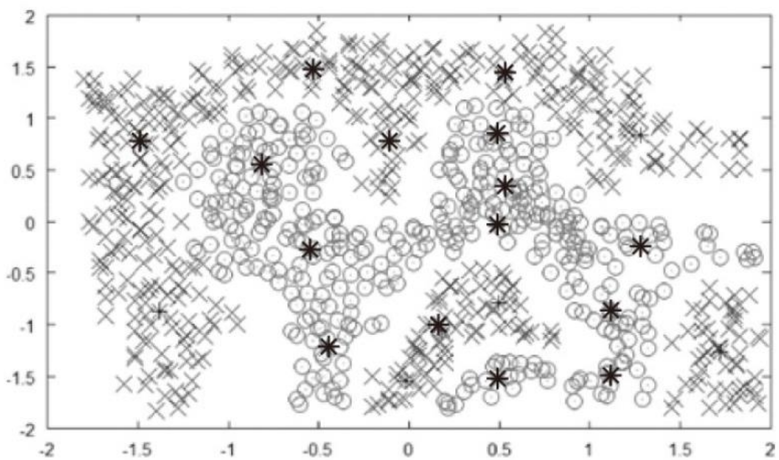
- **경사 하강법**(gradient-descent method) **사용**
  - 기준 벡터 $\boldsymbol{\mu}_j$와 파라미터 β_j, 가중치 w_{kj} 결정

- 부류 별 군집화 결과를 사용한 기준 벡터 $\boldsymbol{\mu}_j$와 파라미터 β_j 초기화
  - 군집 중심 : 기준(평균) 벡터 $\boldsymbol{\mu}_j$
  - 분산의 역수 : β_j

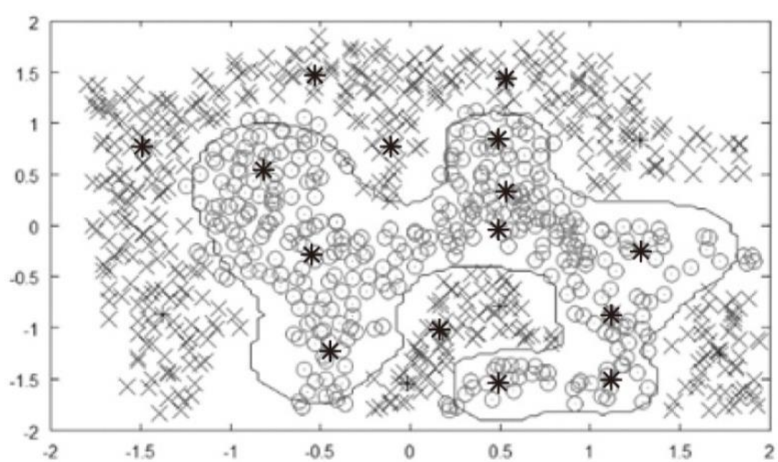
$$\sigma = \frac{1}{m}\sum_{i=1}^{m}\|x_i - \mu\| \qquad \beta = \frac{1}{2\sigma^2}$$

RBF 망

- ❖ **RBF 망을 이용한 분류의 예**
  - **o 부류와 x 부류**



★ : 군집화 결과 군집 중심



곡선 : 분류 결정 경계