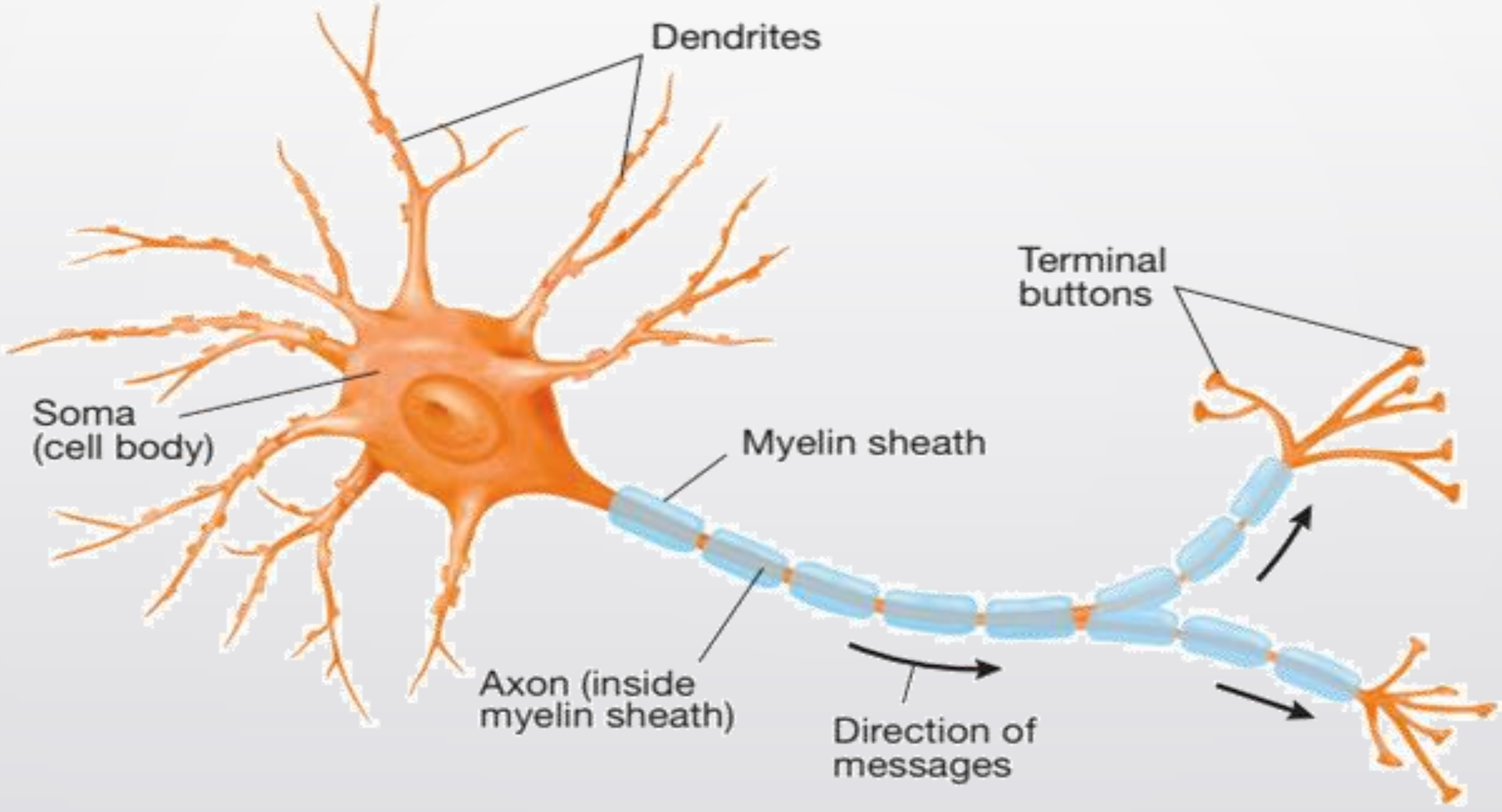
データマイニング
Data Mining

13: ニューラルネットワーク① Neural Network

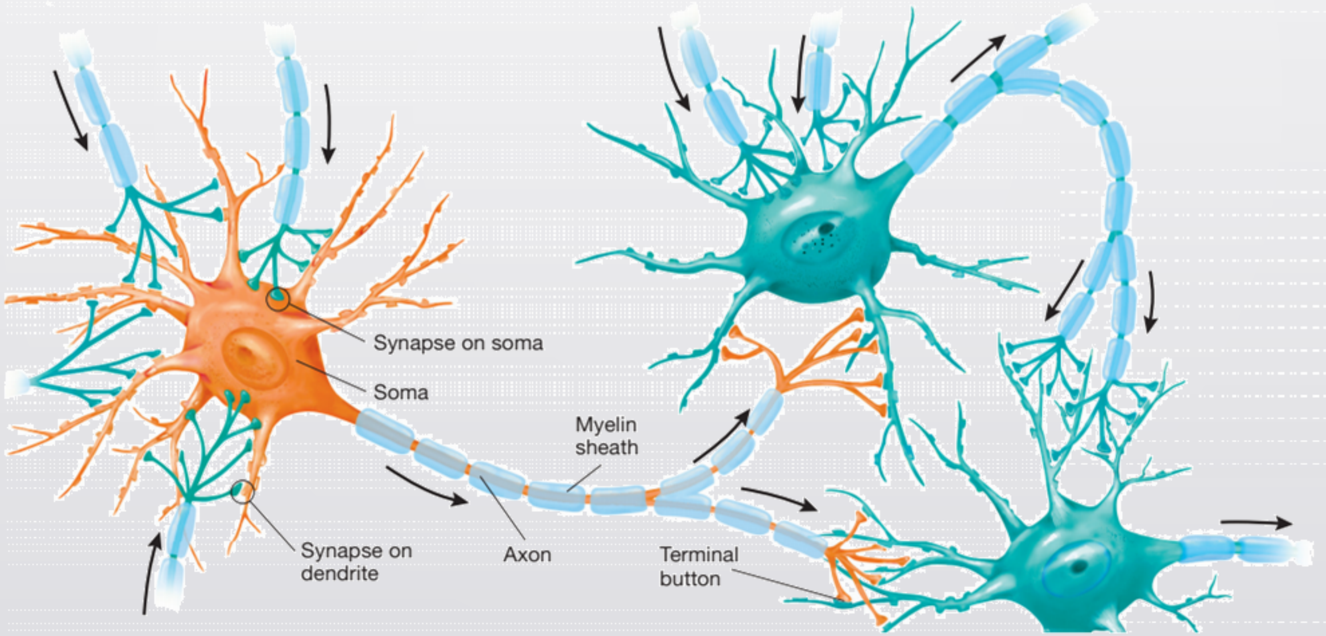
土居 裕和 Hirokazu Doi
長岡技術科学大学 Nagaoka University of Technology

神経細胞 Neuron


Dendrites
Terminal buttons
Soma (cell body)
Myelin sheath
Axon (inside myelin sheath)
Direction of messages

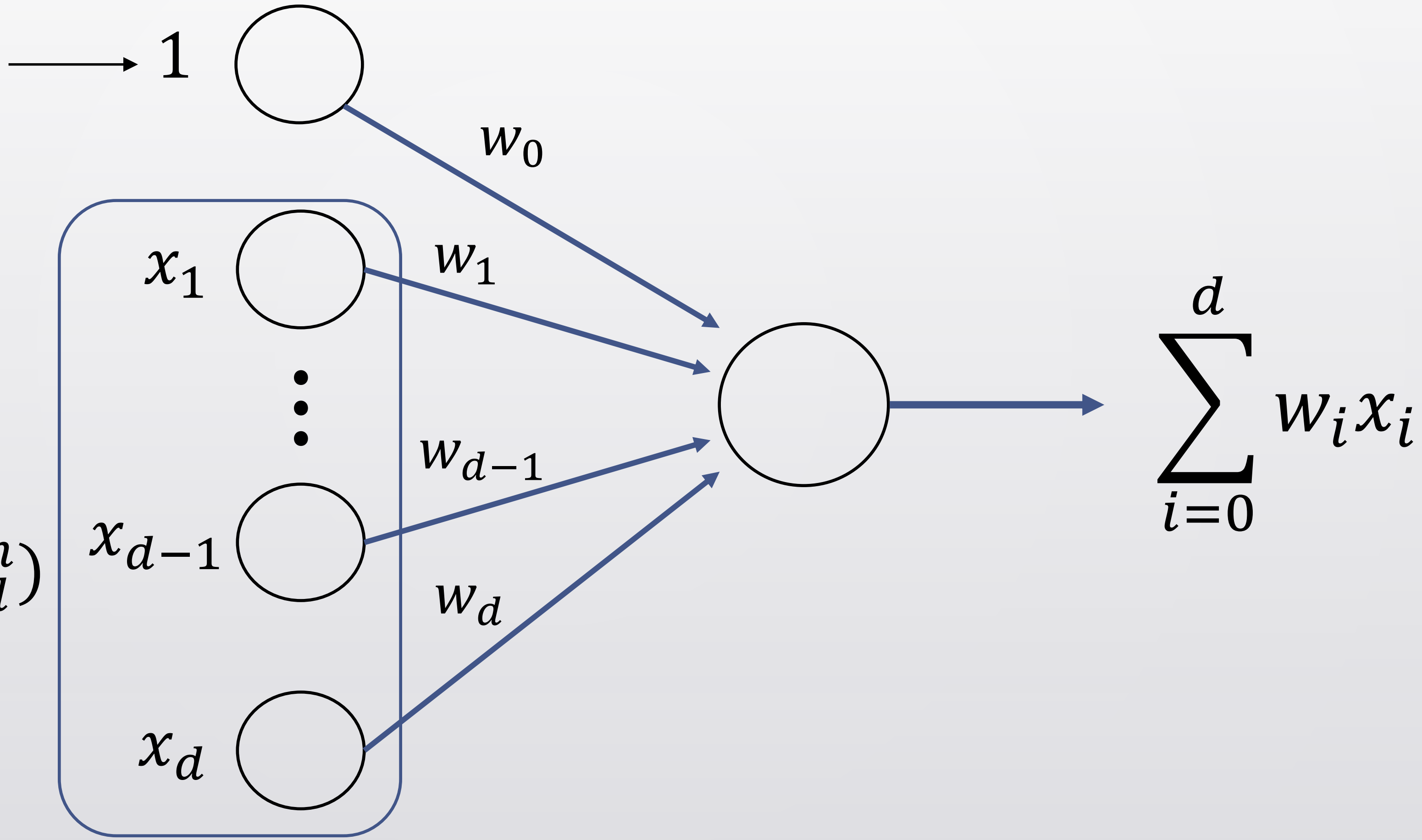
神経細胞の興奮 Neuronal Excitation

神経系の活動＝　神経細胞が電気活動を発生させ、神経細胞間で伝えていくこと
Inter-neuronal transmission of electrical activity



パーセプトロン Perceptron

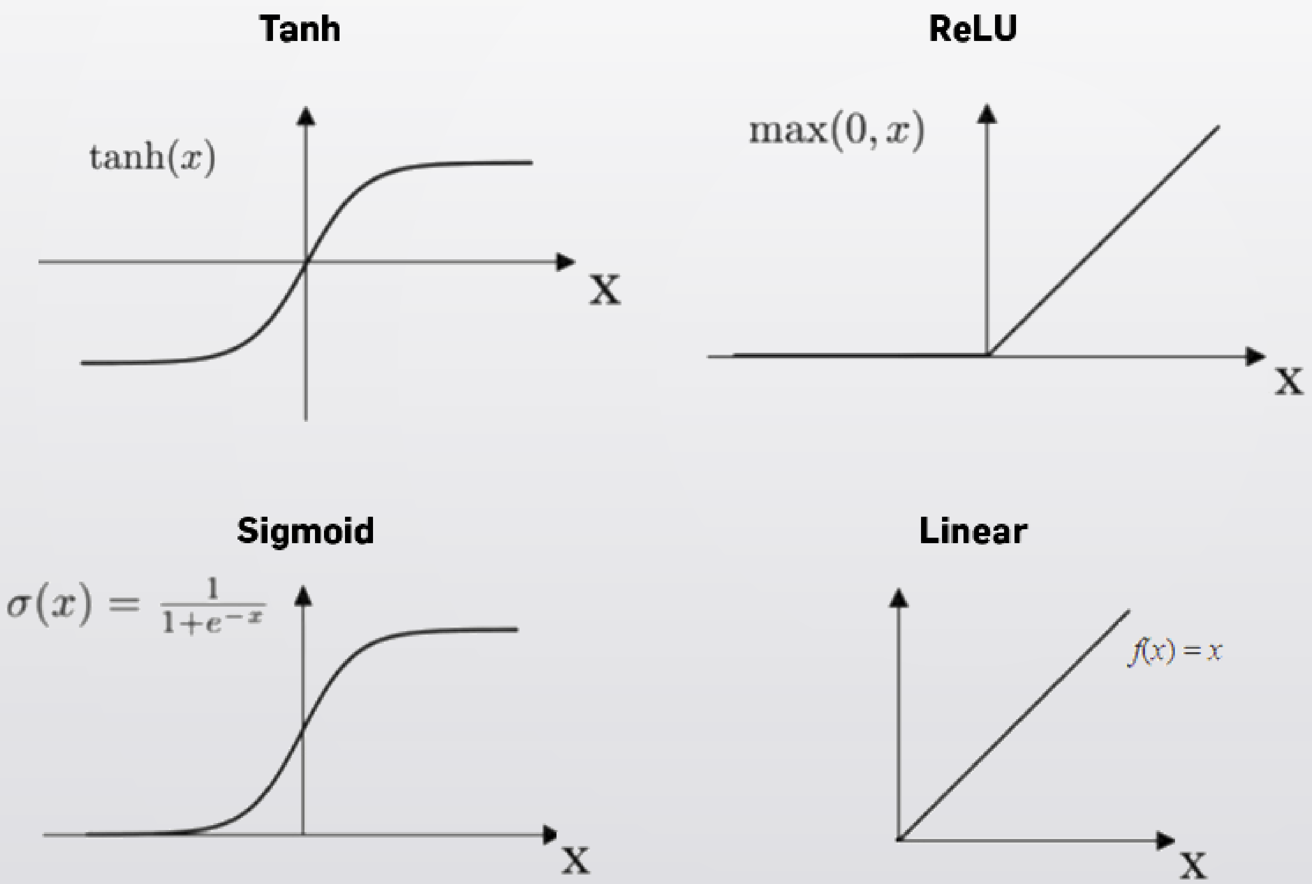
バイアス
Bias



入力 Input

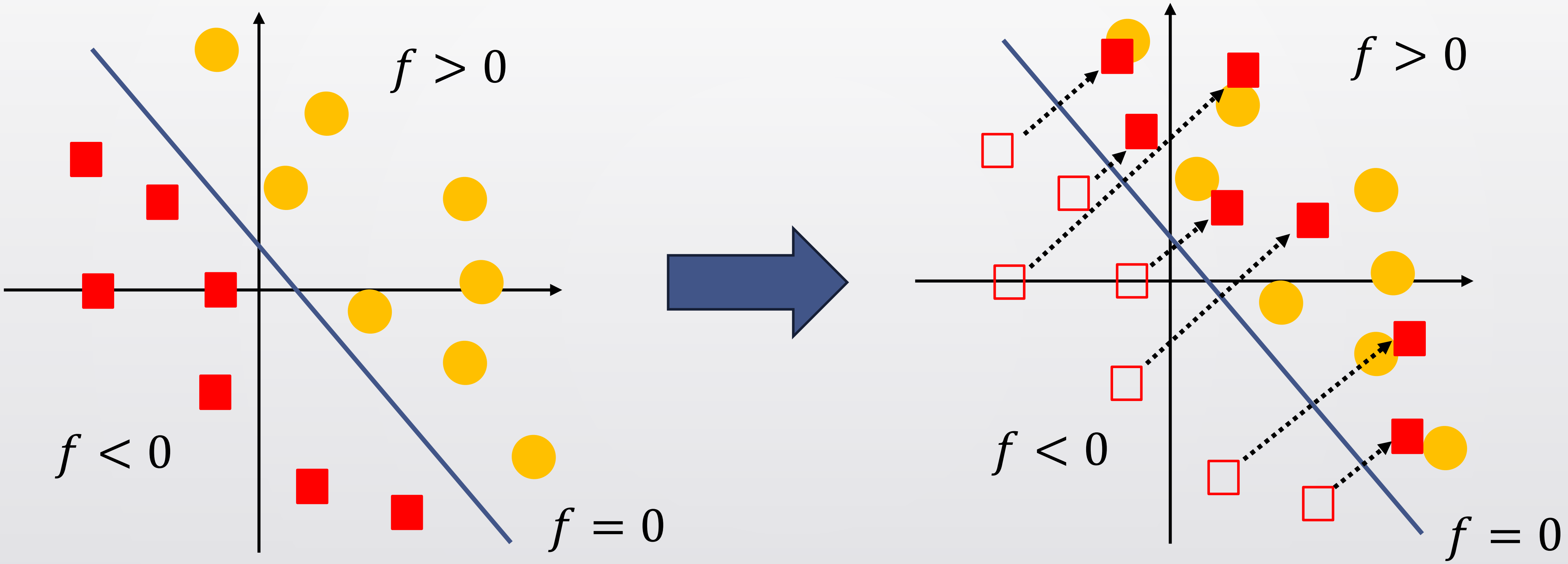
$$\boldsymbol{x^n} = (x_1^n, x_2^n, \cdots x_d^n)$$

活性化関数 Activation Function



https://machine-learning.paperspace.com/wiki/activation-function

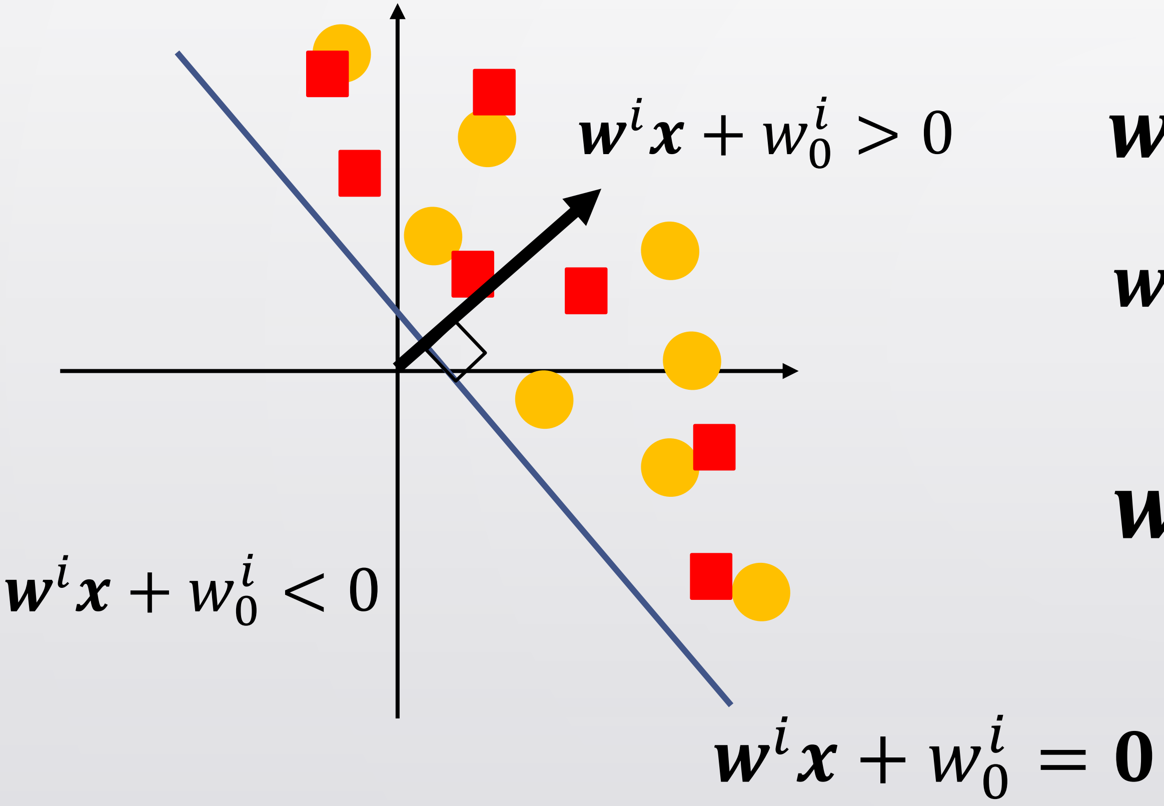
パーセプトロン学習則　Learning Rule of Perceptron



線型分離可能なデータは、符号反転により、すべて$f > 0$の領域にくる

If linearly separable, all the data can be moved to the region $f > 0$ by sign inversion

パーセプトロン学習則　Learning Rule of Perceptron

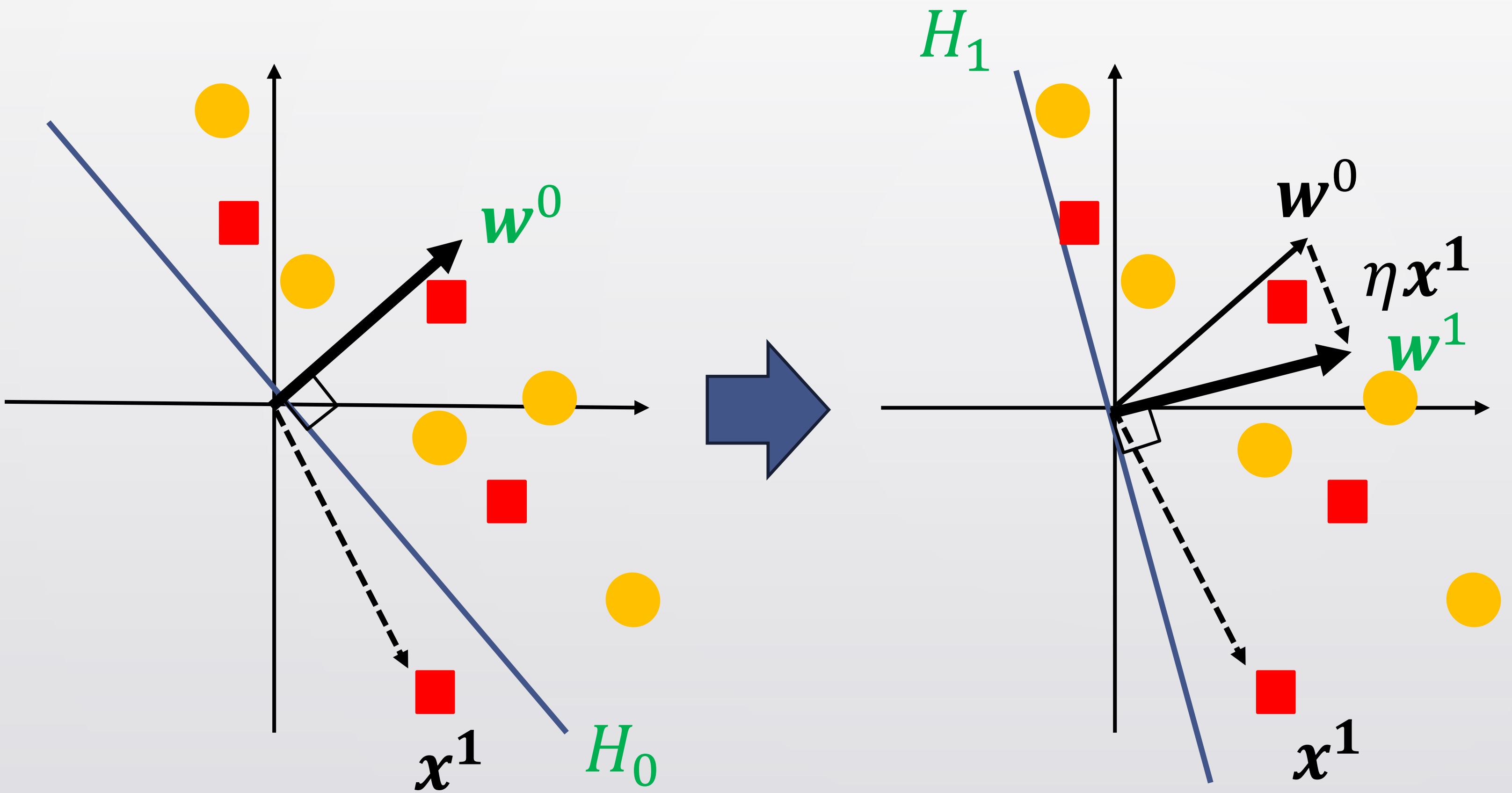


$$\boldsymbol{w^i}, \boldsymbol{x^n} \in R^d$$

$$\boldsymbol{w^i} = \left(w_1^i, w_2^i, \cdots w_d^i\right)$$

$$\boldsymbol{w}^{i+1} = \begin{cases} \boldsymbol{w}^i + \eta \boldsymbol{x^n} & (f(\boldsymbol{x^n}) < 0) \\ \boldsymbol{w}^i & (f(\boldsymbol{x^n}) > 0) \end{cases}$$

パーセプトロン学習則　Learning Rule of Perceptron



超平面を回転させることで、識別性能が向上する

Rotation of hyperplane improves classification performance

ブール論理演算子とパーセプトロン
Boolean Logic Gate and Perceptron

YES

INPUT A	OUTPUT
0	0
1	1

NOT

INPUT A	OUTPUT
0	1
1	0

AND

INPUT A	INPUT B	OUTPUT
0	0	0
1	0	0
0	1	0
1	1	1

OR

INPUT A	INPUT B	OUTPUT
0	0	0
1	0	1
0	1	1
1	1	1

XOR

INPUT A	INPUT B	OUTPUT
0	0	0
1	0	1
0	1	1
1	1	0

NAND

INPUT A	INPUT B	OUTPUT
0	0	1
1	0	1
0	1	1
1	1	0

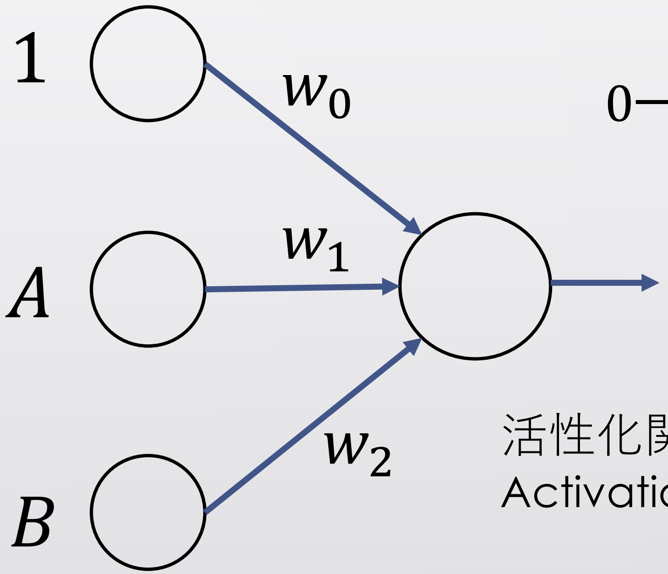
NOR

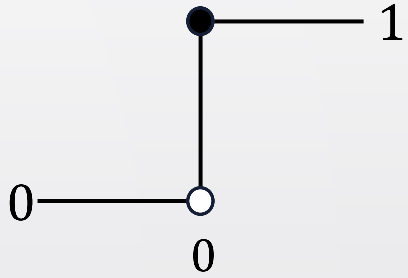
INPUT A	INPUT B	OUTPUT
0	0	1
1	0	0
0	1	0
1	1	0

XNOR

INPUT A	INPUT B	OUTPUT
0	0	1
1	0	0
0	1	0
1	1	1

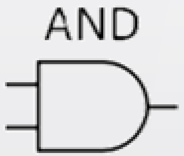
Abels et al, 2015



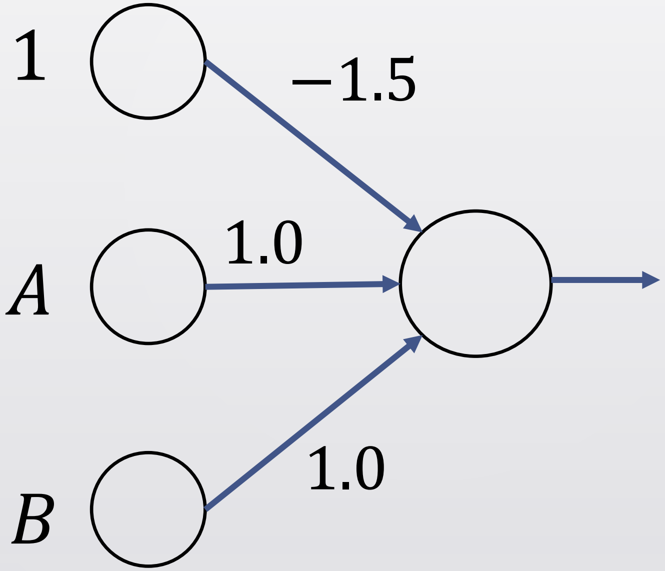


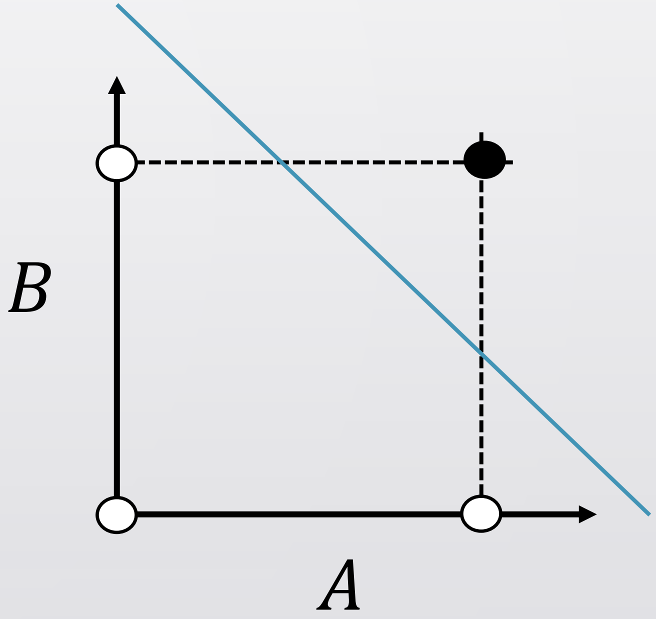
活性化関数はステップ関数とする
Activation function is step function

ブール論理演算子とパーセプトロン
Boolean Logic Gate and Perceptron



INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

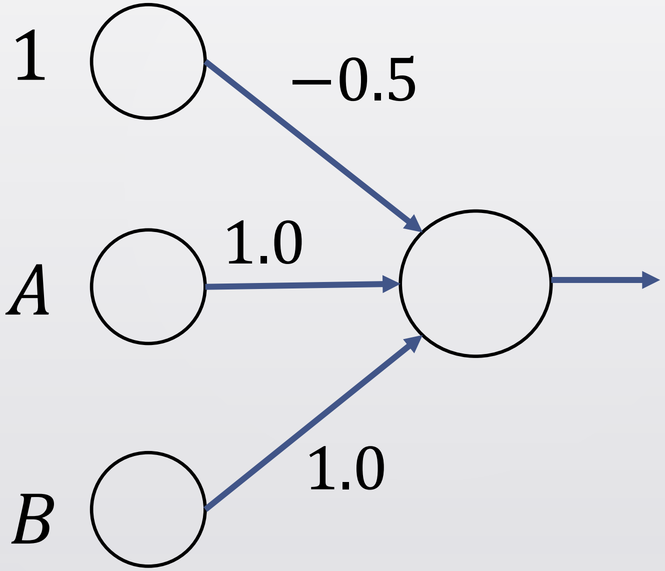


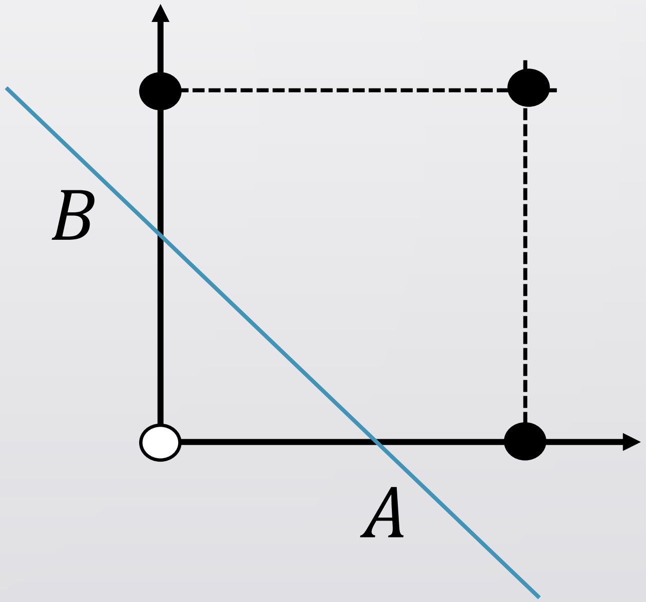


ブール論理演算子とパーセプトロン
Boolean Logic Gate and Perceptron

OR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

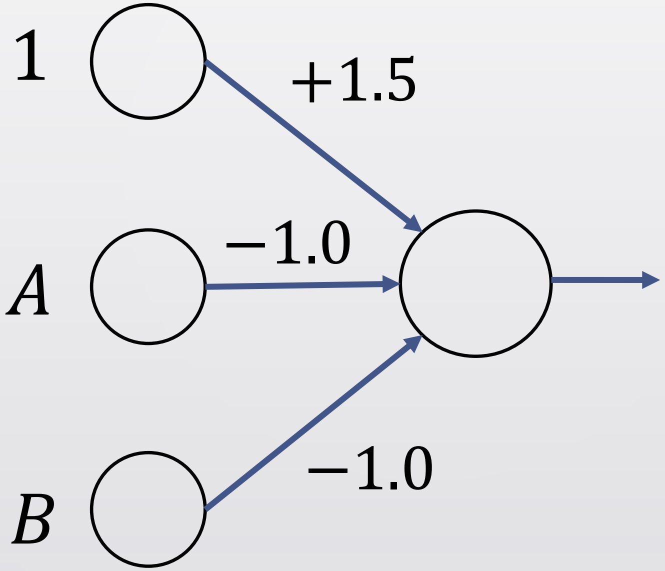


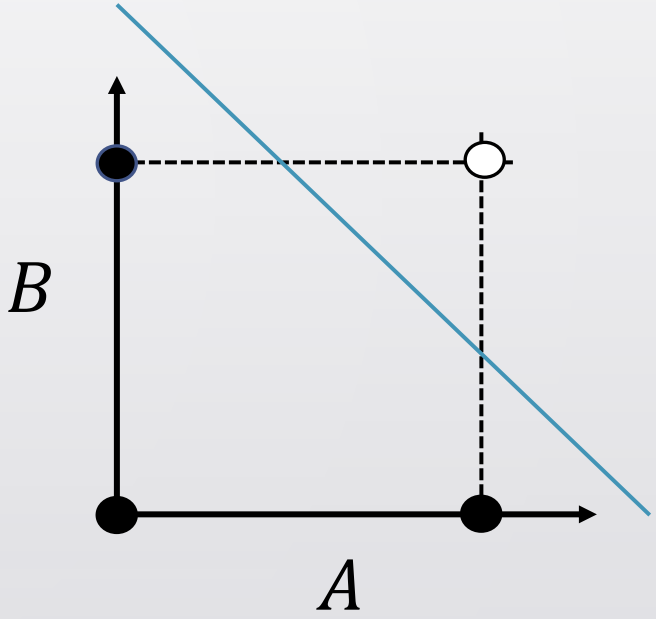


ブール論理演算子とパーセプトロン
Boolean Logic Gate and Perceptron

NAND

INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

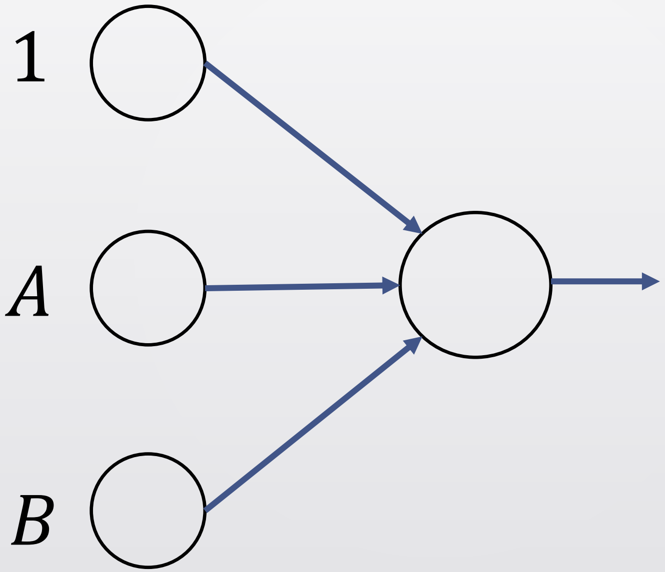


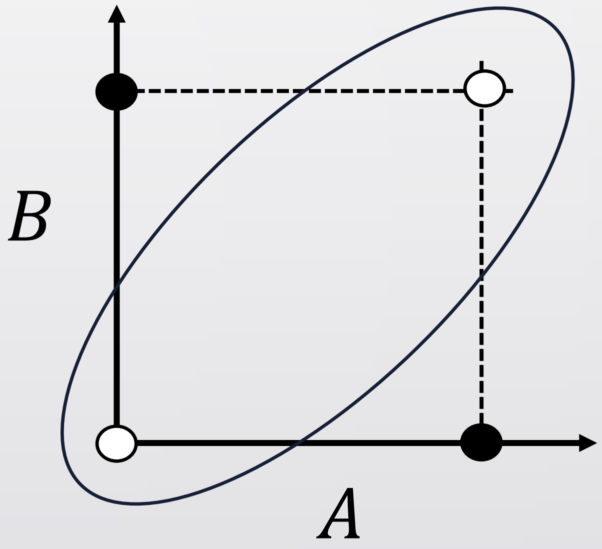


ブール論理演算子とパーセプトロン
Boolean Logic Gate and Perceptron

XOR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0





単層パーセプトロンでは排他的論理和を構成できない

Single-layered perceptron cannot compose XOR

ブール論理演算子とパーセプトロン
Boolean Logic Gate and Perceptron

OR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

$$AND$$

NAND

INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

$$=$$

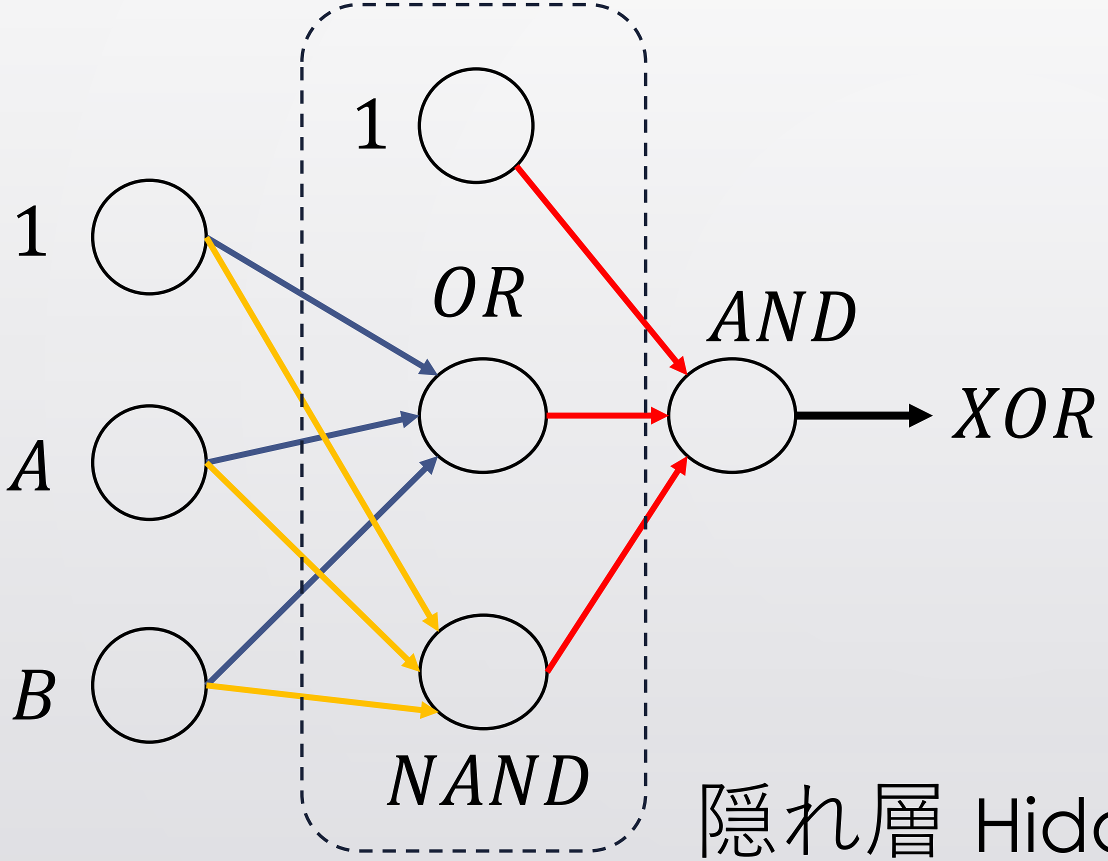
XOR

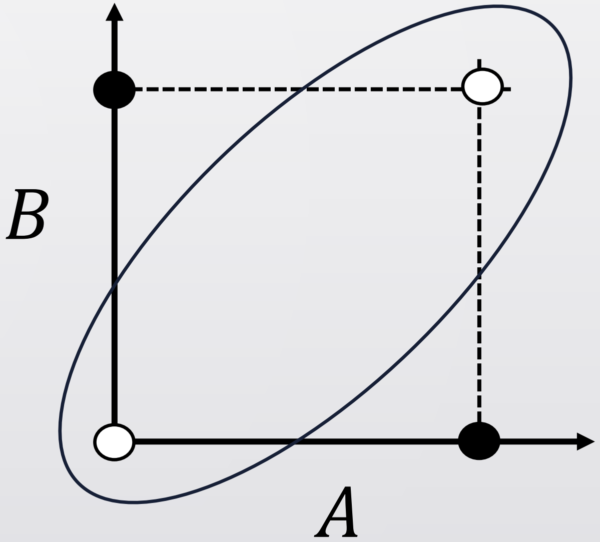
INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

XORはORとNANDの論理積

XOR is logical conjunction of OR and NAND

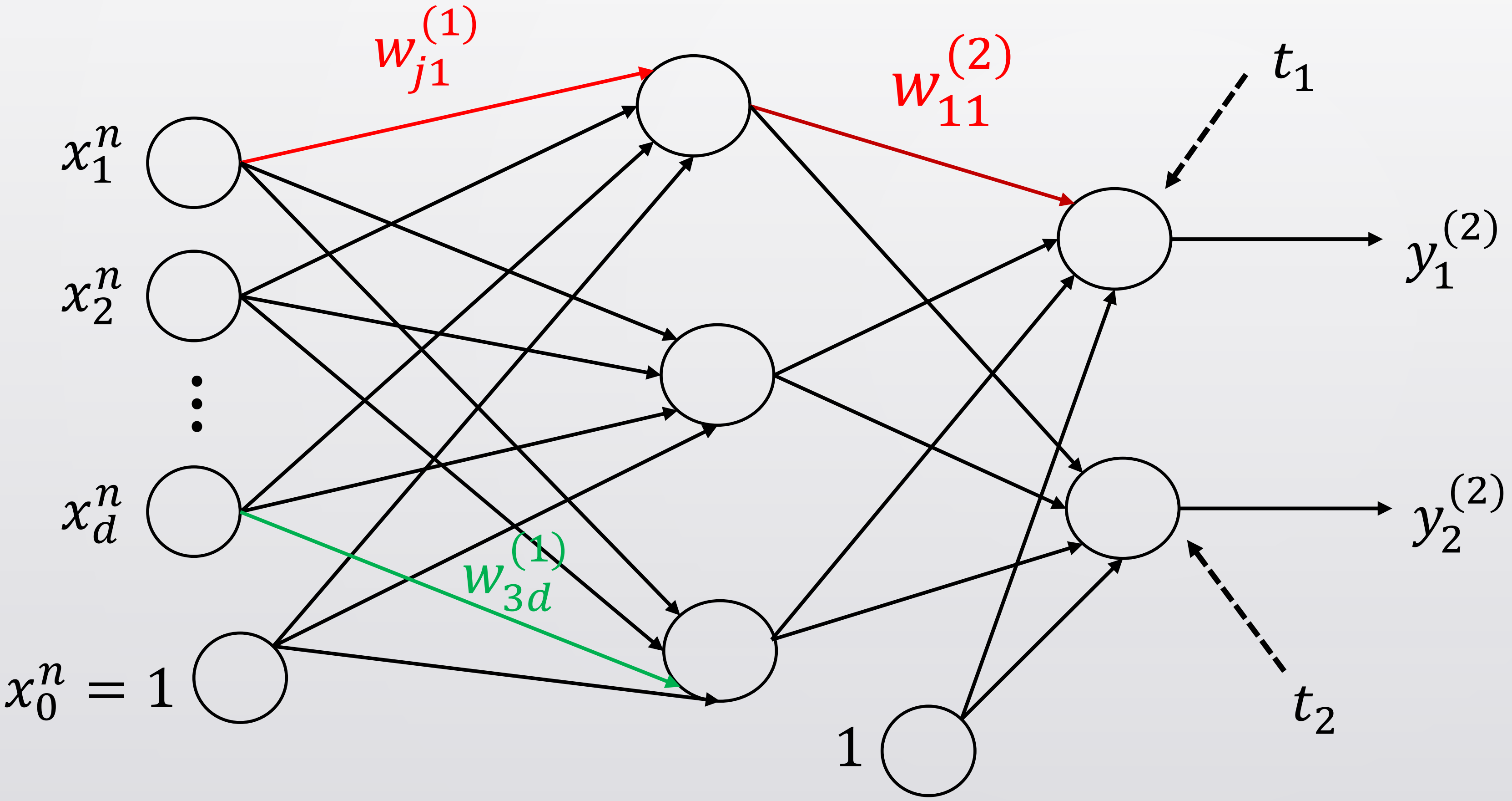
XORと多層パーセプトロン XOR and Multi-layered Perceptron





隠れ層 Hidden Layer

多層パーセプトロン Multi-layered Perceptron



$w_{ji}^{(1)}$: 入力層から隠れ層への重み

Weights from input to hidden layer

$w_{kj}^{(2)}$: 隠れ層から出力層への重み

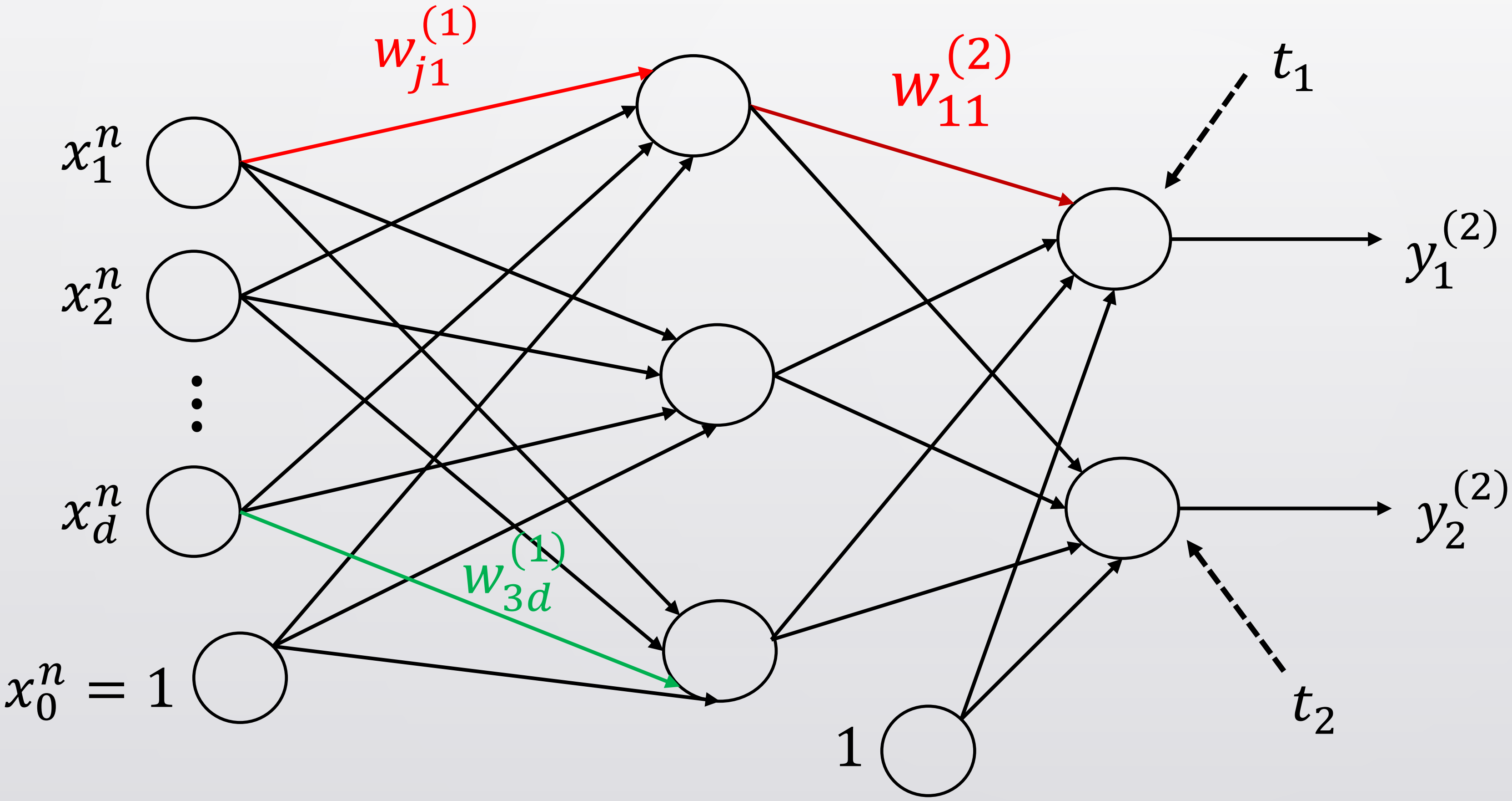
Weights from hidden to output layer

$i = 0, 1, 2 \cdots d$

$j = 0, 1, 2 \cdots M$

$k = 1, 2 \cdots C$

多層パーセプトロン Multi-layered Perceptron



$y_k^{(2)}$: 出力 Output

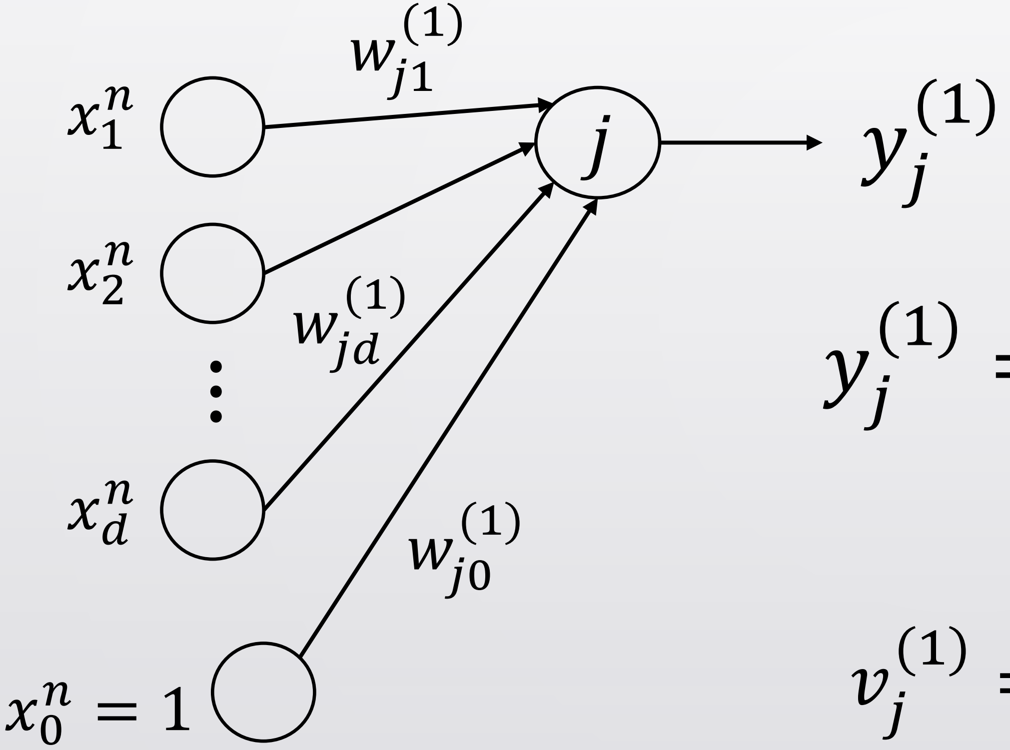
t_k : 教師信号

Teacher Signal

$$t_k \in \{0, 1\} \quad \sum_{k=1}^{k=C} t_k = 1$$

$$k = 1, 2 \cdots C$$

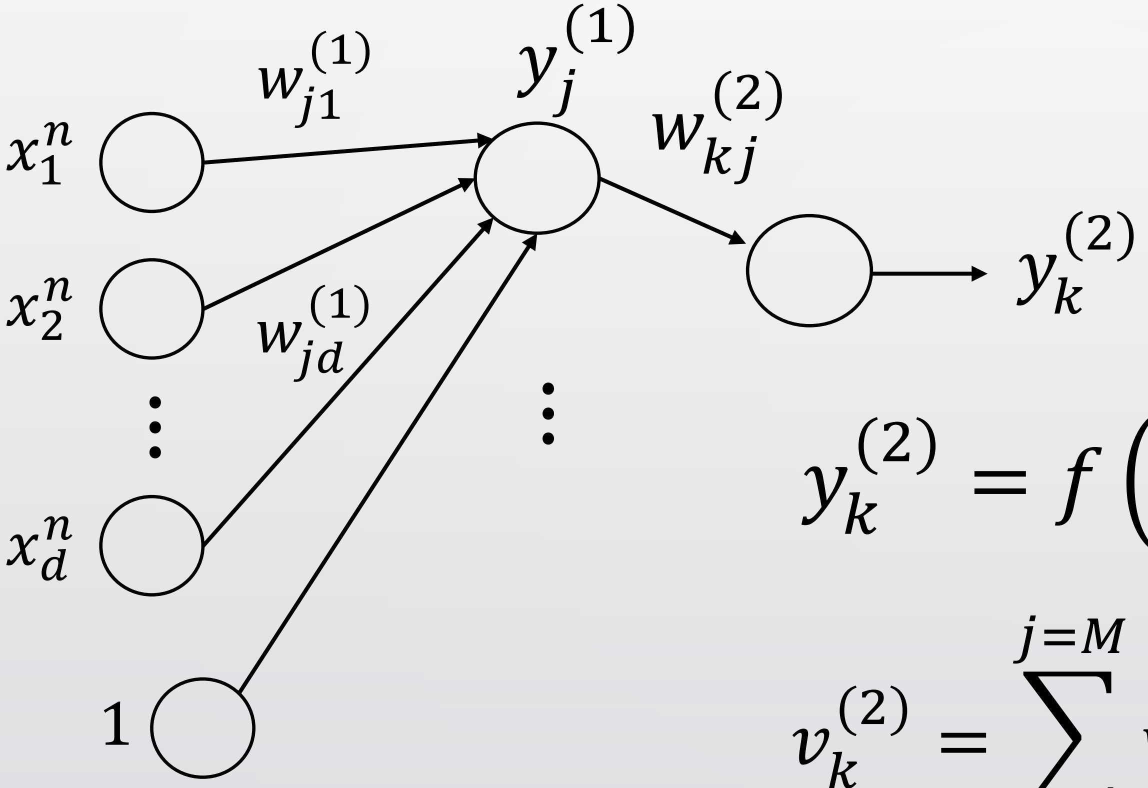
隠れ層の出力 Output of Hidden Layer



$$y_j^{(1)} = f\left(\sum_{i=0}^{i=d} w_{ji}^{(1)} x_i^n\right) = f\left(\boldsymbol{w}_j^{(1)} \boldsymbol{x}^{nT}\right)$$

$$v_j^{(1)} = \sum_{i=0}^{i=d} w_{ji}^{(1)} x_i^n$$

出力 Output



$$y_k^{(2)} = f\left(\sum_{j=0}^{j=M} w_{kj}^{(2)} y_j^{(1)}\right) = f\left(\boldsymbol{w}_k^{(2)} \boldsymbol{y}^{(1)}\right)$$

$$v_k^{(2)} = \sum_{j=0}^{j=M} w_{kj}^{(2)} y_j^{(1)}$$

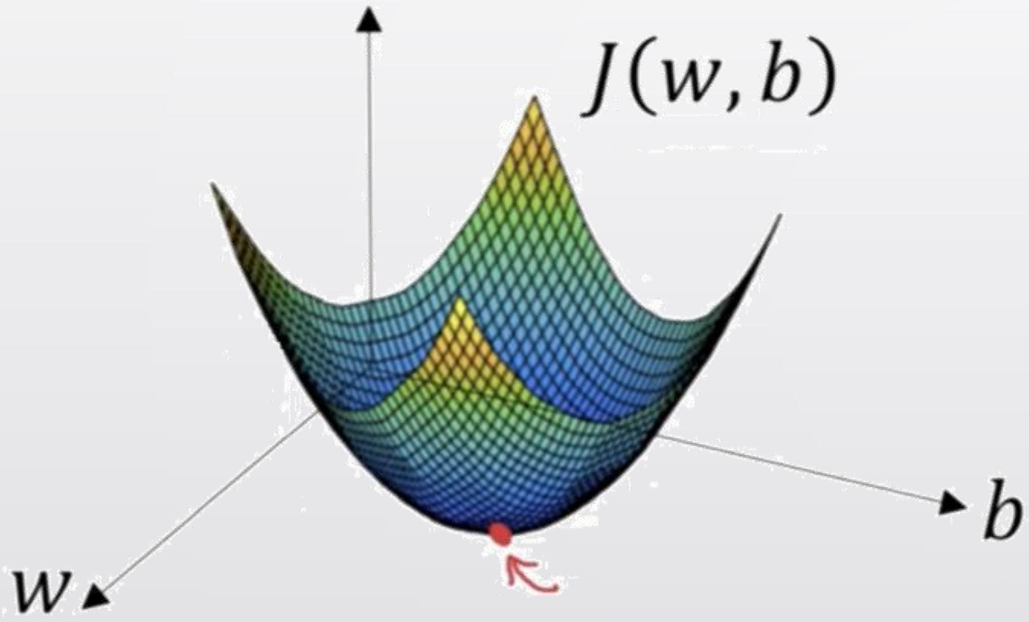
合成関数の微分 Derivative of Composite Function

$$\frac{df\big(g(x)\big)}{dx} \quad u = g(x)$$

$$\frac{df\big(g(x)\big)}{dx} = \frac{du}{dx}\frac{df(u)}{du}$$

$$= g'(x)\, f'(u) = g'(x)\, f'(g(x))$$

勾配降下 Gradient Descent



$$-\nabla J = \left(-\frac{\partial J}{\partial w}, -\frac{\partial J}{\partial b}\right)$$

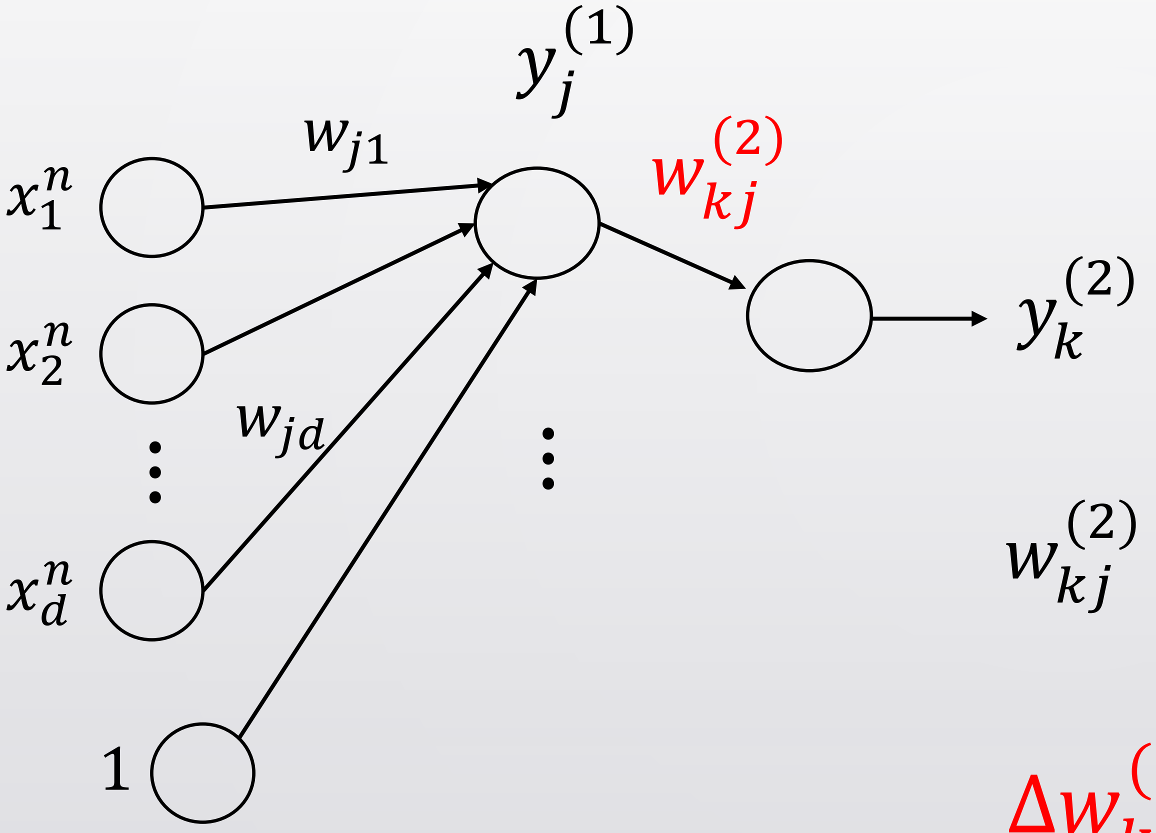
$$w \leftarrow w - \eta \frac{\partial J}{\partial w}$$

$-\nabla J$ の方向に変数を変化させることで、関数Jの値を最も素早く減少させることが出来る

The output value of function$J$decreases most rapidly along the direction of $-\nabla J$

https://www.linkedin.com/pulse/gradient-descent-its-applications-deep-learning-chirag-subramanian

出力と教師信号の誤差 Error of network output



Eを最小化するよう$w_{kj}^{(2)}$を更新する

Update $w_{kj}^{(2)}$ so as to minimize E

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

$$w_{kj}^{(2)} = w_{kj}^{(2)} - \eta \frac{\partial E}{\partial w_{kj}^{(2)}} = w_{kj}^{(2)} + \Delta w_{kj}^{(2)}$$

$$\Delta w_{kj}^{(2)} = -\eta \frac{\partial E}{\partial w_{kj}^{(2)}}$$

重みの更新 Weight Updating

$$\Delta w_{kj}^{(2)} = -\eta \frac{\partial E}{\partial w_{kj}^{(2)}} == -\eta \left(y_k^{(2)} - t_k \right) \frac{\partial y_k^{(2)}}{\partial w_{kj}^{(2)}}$$

$$= -\eta \left(y_k^{(2)} - t_k \right) \frac{\partial v_k^{(2)}}{\partial w_{kj}^{(2)}} \frac{\partial f\left(v_k^{(2)}\right)}{\partial v_k^{(2)}}$$

$$= -\eta \left(y_k^{(2)} - t_k \right) y_j^{(1)} f'\left(v_k^{(2)}\right) \qquad v_k^{(2)} = \sum_{j=0}^{j=M} w_{kj}^{(2)} y_j^{(1)}$$

重みの更新 Weight Updating

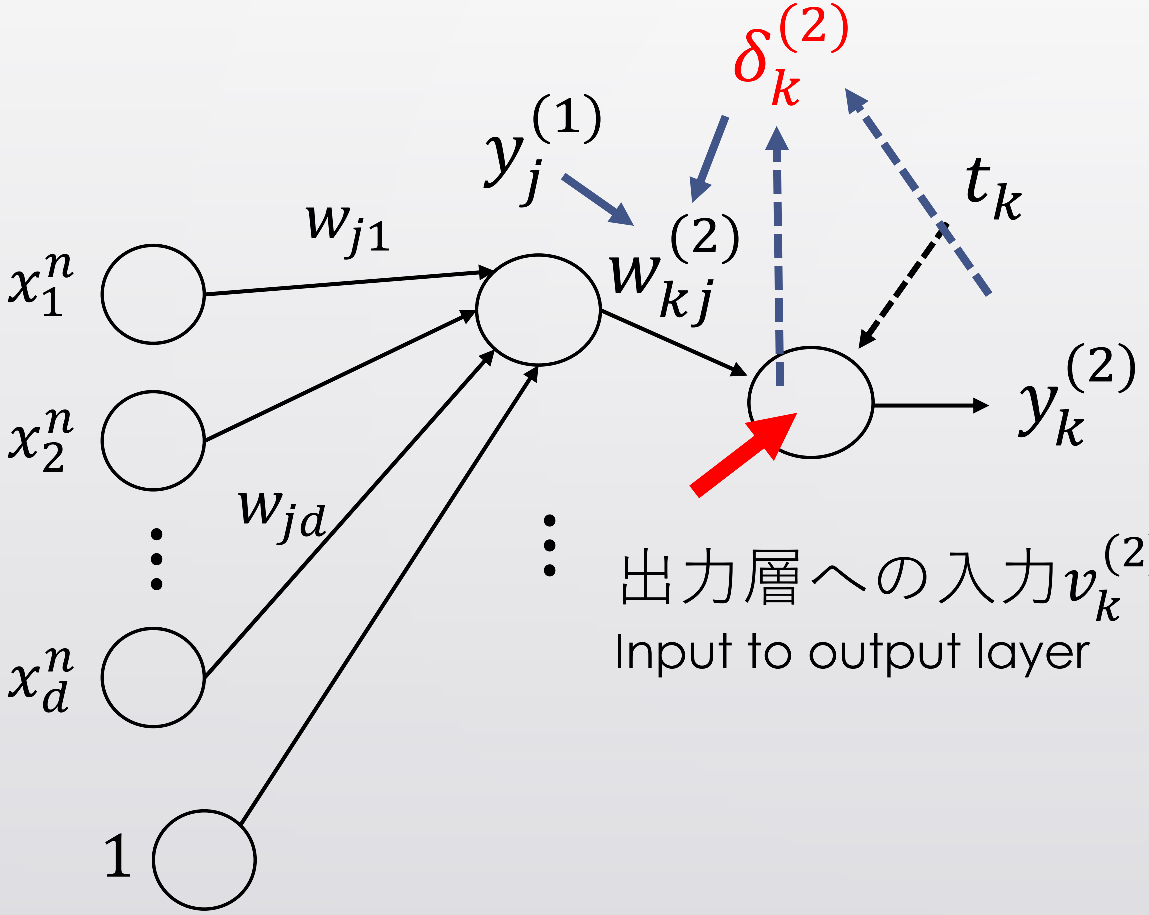
$$\Delta w_{kj}^{(2)} = -\eta \left(\underline{y_k^{(2)} - t_k} \right) y_j^{(1)} f' \left(\underline{v_k^{(2)}} \right)$$

誤差 Error

出力層への入力
Input to output layer

隠れ層の出力
Output of hidden layer

重みの更新 Weight Updating



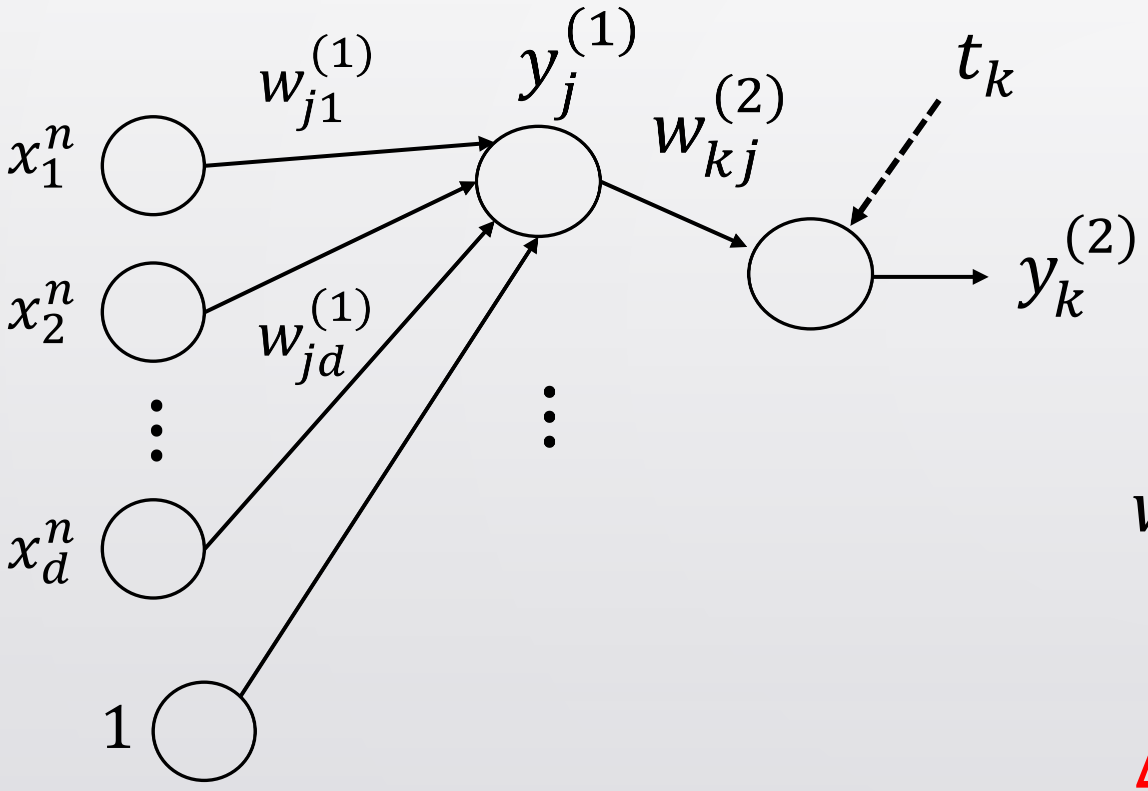
出力層への入力$v_k^{(2)}$
Input to output layer

$$w_{kj}^{(2)} = w_{kj}^{(2)} + \Delta w_{kj}^{(2)}$$

$$\Delta w_{kj}^{(2)} = -\eta\, \delta_k^{(2)} y_j^{(1)}$$

$$\delta_k^{(2)} = \underbrace{\left(y_k^{(2)} - t_k\right)}_{\text{誤差 Error}} f'\left(\underbrace{v_k^{(2)}}_{\text{出力層への入力 Input to output layer}}\right)$$

重みの更新 Weight Updating



Eを最小化するよう$w_{ji}^{(1)}$を更新する

Update $w_{ji}^{(1)}$ so as to minimize E

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

$$w_{ji}^{(1)} = w_{ji}^{(1)} - \eta\frac{\partial E}{\partial w_{ji}^{(1)}} = w_{ji}^{(1)} + \Delta w_{ji}^{(1)}$$

$$\Delta w_{ji}^{(1)} = -\eta\frac{\partial E}{\partial w_{ji}^{(1)}}$$

重みの更新 Weight Updating

$$\Delta w_{ji}^{(1)} = -\eta \frac{\partial E}{\partial w_{ji}^{(1)}} = -\eta \frac{\partial}{\partial w_{ji}^{(1)}} \frac{1}{2} \sum_{k=1}^{C} \left(y_k^{(2)} - t_k \right)^2$$

$$= -\eta \sum_{k=1}^{C} \left(y_k^{(2)} - t_k \right) \frac{\partial y_k^{(2)}}{\partial w_{ji}^{(1)}}$$

重みの更新 Weight Updating

$$v_k^{(2)} = \sum_{j=0}^{j=M} w_{kj}^{(2)} y_j^{(1)} \qquad y_k^{(2)} = f\left(v_k^{(2)}\right)$$

$$\Delta w_{ji}^{(1)} = -\eta \sum_{k=1}^{C} \left(y_k^{(2)} - t_k\right) \frac{\partial f\left(v_k^{(2)}\right)}{\partial w_{ji}^{(1)}}$$

$$= -\eta \sum_{k=1}^{C} \left(y_k^{(2)} - t_k\right) \frac{\partial v_k^{(2)}}{\partial w_{ji}^{(1)}} \frac{\partial f\left(v_k^{(2)}\right)}{\partial v_k^{(2)}}$$

重みの更新 Weight Updating

$$v_k^{(2)} = \sum_{j=0}^{j=M} w_{kj}^{(2)} y_j^{(1)} \quad v_j^{(1)} = \sum_{i=0}^{i=d} w_{ji}^{(1)} x_i^n \quad y_j^{(1)} = f\left(v_j^{(1)}\right)$$

$$\frac{\partial v_k^{(2)}}{\partial w_{ji}^{(1)}} = \frac{\partial v_j^{(1)}}{\partial w_{ji}^{(1)}} \frac{\partial v_k^{(2)}}{\partial v_j^{(1)}} = x_i^n \frac{\partial}{\partial v_j^{(1)}} \sum_{l=0}^{l=M} w_{kl}^{(2)} f\left(v_l^{(1)}\right)$$

$$= x_i^n w_{kj}^{(2)} \frac{\partial f\left(v_j^{(1)}\right)}{\partial v_j^{(1)}} = x_i^n w_{kj}^{(2)} f'\left(v_j^{(1)}\right)$$

重みの更新 Weight Updating

$$\Delta w_{ji}^{(1)} = -\eta \sum_{k=1}^{C} \left(y_k^{(2)} - t_k \right) \frac{\partial v_k^{(2)}}{\partial w_{ji}^{(1)}} \frac{\partial f\left(v_k^{(2)}\right)}{\partial v_k^{(2)}}$$

$$= -\eta \sum_{k=1}^{C} \left(y_k^{(2)} - t_k \right) x_i^n w_{kj}^{(2)} f'\left(v_j^{(1)}\right) \frac{\partial f\left(v_k^{(2)}\right)}{\partial v_k^{(2)}}$$

$$\delta_k^{(2)} = \left(y_k^{(2)} - t_k \right) f'\left(v_k^{(2)}\right)$$

重みの更新 Weight Updating

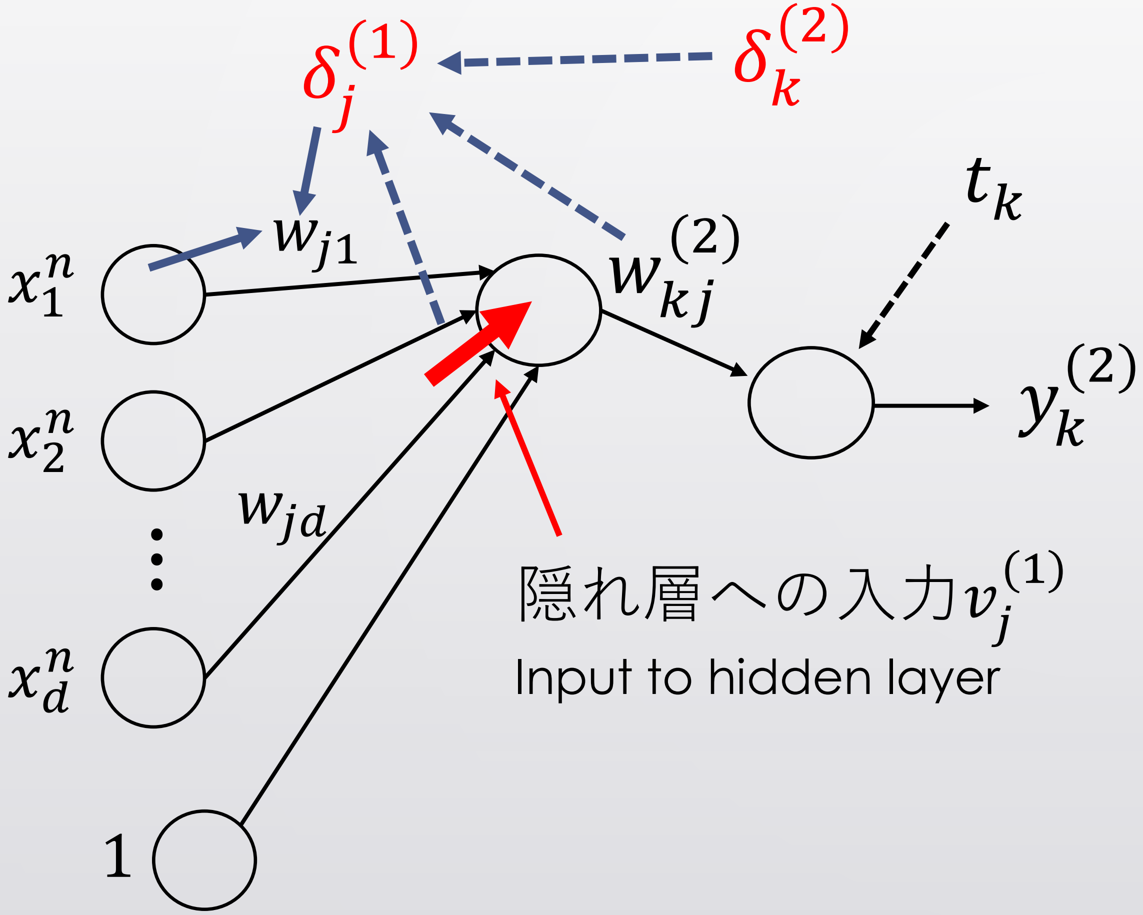
$$\Delta w_{ji}^{(1)} = -\eta \sum_{k=1}^{C} \left(y_k^{(2)} - t_k\right) x_i^n w_{kj}^{(2)} f'\left(v_j^{(1)}\right) \frac{\partial f\left(v_k^{(2)}\right)}{\partial v_k^{(2)}}$$

$$= -\eta \underline{x_i^n} f'\left(\underline{v_j^{(1)}}\right) \sum_{k=1}^{C} w_{kj}^{(2)} \delta_k^{(2)}$$

入力 Input

隠れ層への入力
Input to hidden layer

重みの更新 Weight Updating



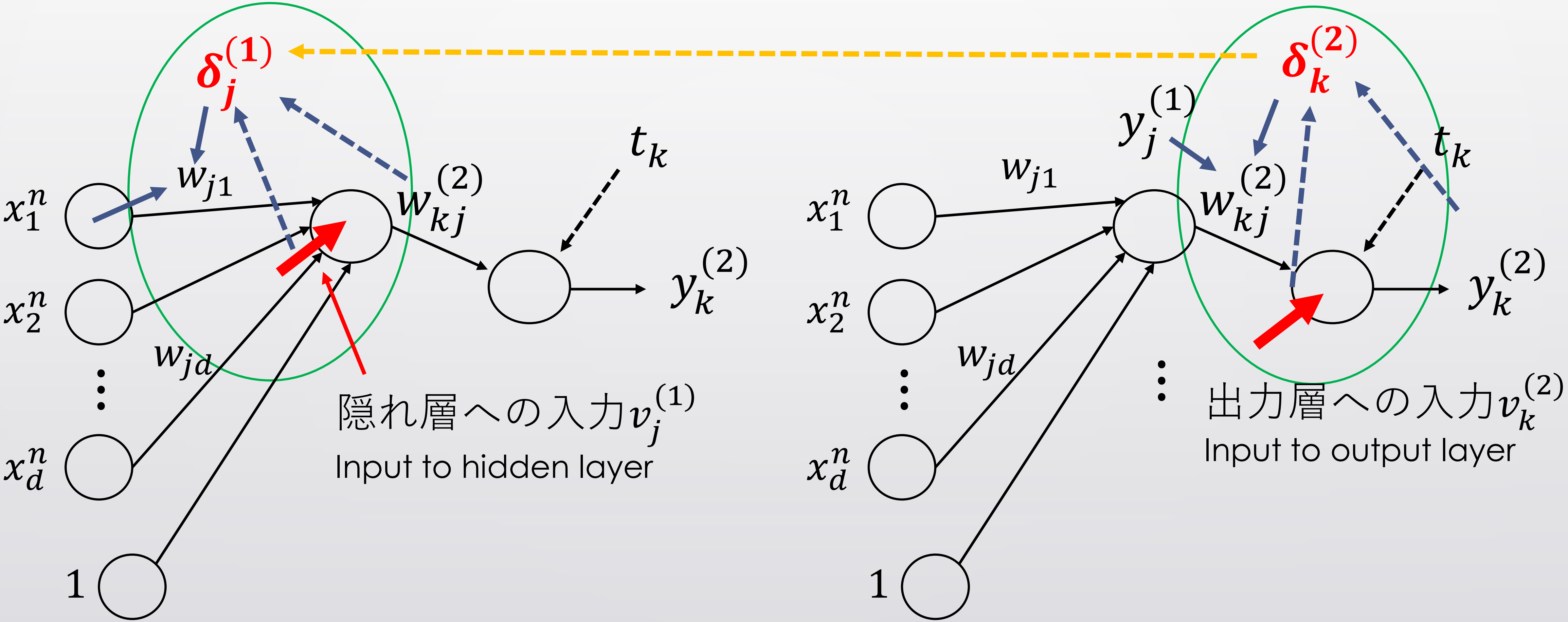
$$w_{ji}^{(1)} = w_{ji}^{(1)} + \Delta w_{ji}^{(1)}$$

$$\Delta w_{ji}^{(1)} = -\eta\, \delta_j^{(1)} x_i^n$$

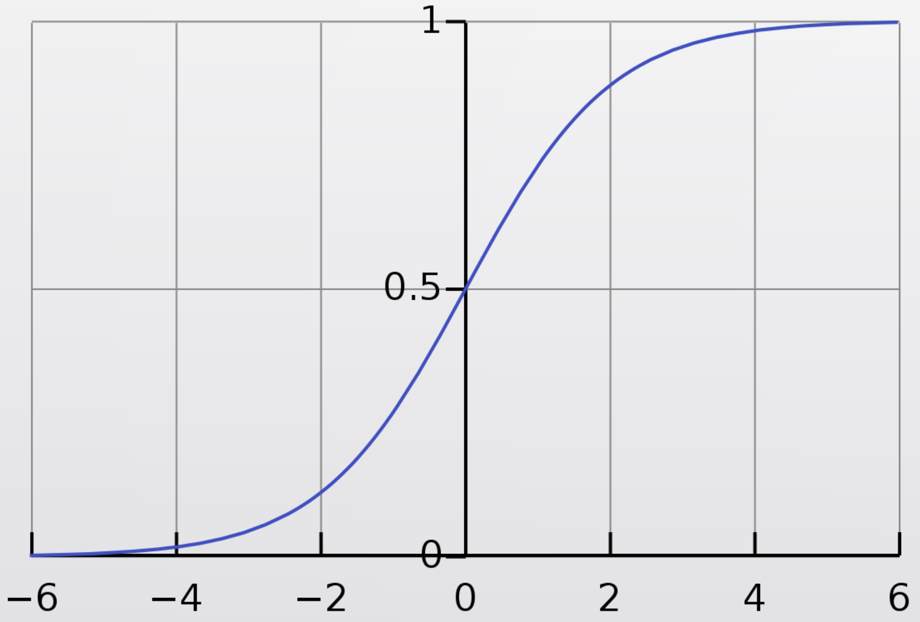
入力 Input

$$\delta_j^{(1)} = f'\left(v_j^{(1)}\right) \sum_{k=1}^{k=C} w_{kj}^{(2)} \cdot \delta_k^{(2)}$$

重みの更新 Weight Updating



シグモイド関数 Sigmoid Function



https://en.wikipedia.org/wiki/Sigmoid_function

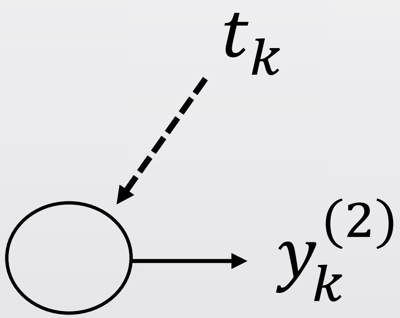
$$f(x) = \frac{1}{1 + e^{-x}}$$

$$f'(x) = \frac{1 + e^{-x}}{(1 + e^{-x})^2} = f(x)(1 - f(x))$$

ソフトマックス関数 Softmax Function

ソフトマックス関数で出力を各クラスに属する確率に変換する

Convert output of NN to the probability of belonging to each class by softmax function



$$P_k = \frac{exp\left(y_k^{(2)}\right)}{\sum_{k=1}^{k=C} exp\left(y_k^{(2)}\right)}$$

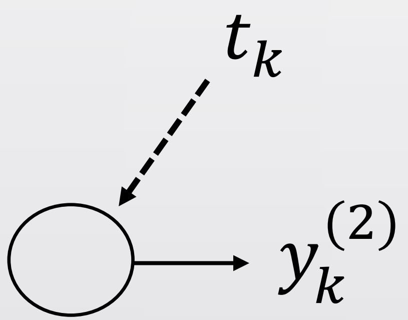
C: クラスの総数

Total number of classes

損失関数 Loss Function

出力と教師データの誤差を反映する損失関数を最小化する

Minimize loss function that reflects divergence between output of network and teacher signal



二乗誤差の和も損失関数の一つ

Sum of squared error is one type of loss function

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

交差エントロピー Cross Entropy

KLダイバージェンスを拡張した損失関数

Loss function based on the concept of KL divergence

$$CE = -\sum_{k=1}^{k=C} t_k \, log(P_k) \qquad P_k = \frac{exp\left(y_k^{(2)}\right)}{\sum_{k=1}^{k=C} exp\left(y_k^{(2)}\right)}$$

交差エントロピー Cross Entropy

$$KL(p(x)|q(x)) = \int p(x)\, log\left(\frac{p(x)}{q(x)}\right)$$

$$p = \left(t_{1,}\, t_2 \cdots t_C\right) \qquad t_k \in \{0, 1\} \qquad \sum_{k=1}^{k=C} t_k = 1$$

$$q = \left(P_{1,}\, P_2 \cdots P_C\right)$$

交差エントロピー Cross Entropy

$$KL(p|q) = \sum_{k=1}^{k=C} t_k \, log\left(\frac{t_k}{P_k}\right)$$

$$= \underbrace{\sum_{k=1}^{k=C} t_k log(t_k)}_{} - \underbrace{\sum_{k=1}^{k=C} t_k log(P_k)}_{}$$

教師信号にのみ関係
Relevant only to teacher signal

交差エントロピー
Cross Entropy