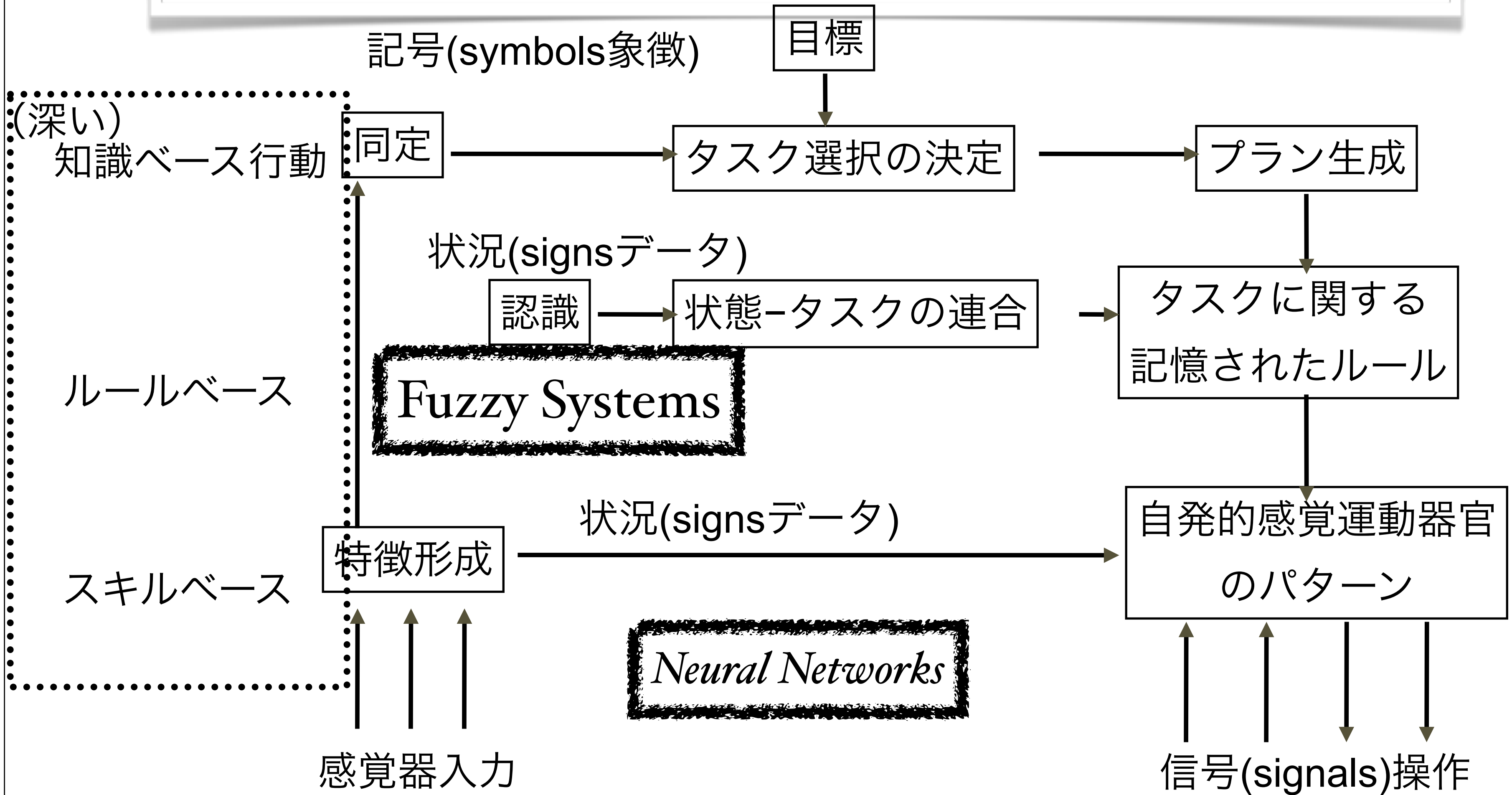
情報メディア論２
2010-8

ニューラルネットワークモデル

ー階層型ニューラルネットワークー

人間の行動様式分析
Rusmussen 1980



Fuzzy + Neural Network

Fuzzy Systems → (言語的)ルールで記述

Neural Networks → 信号処理　学習

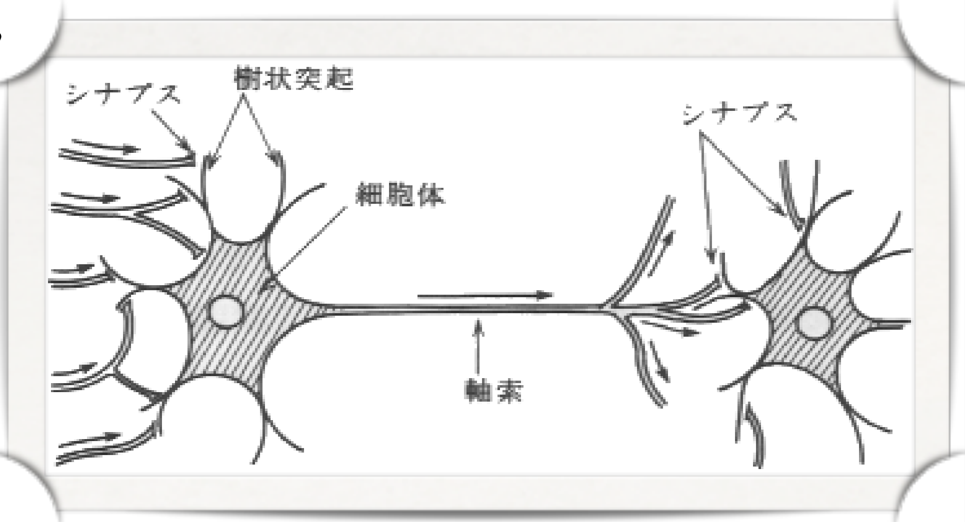
?

信号処理 → ルール化

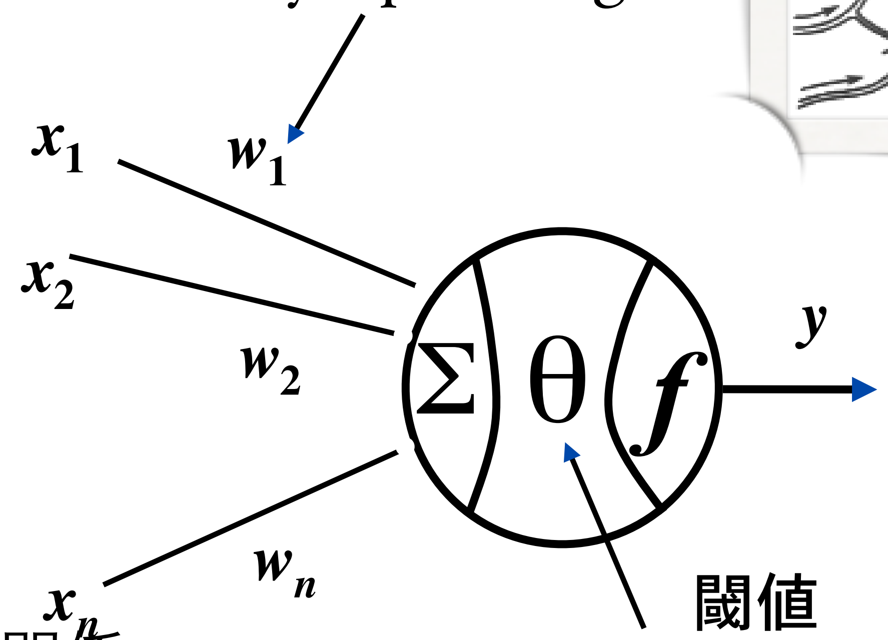
記号処理 → 信号処理

ニューロンモデル

- McCulloch, Pitts (1943)



シナプス荷重(synaptic weight)



閾値

(Threshold)

２値出力の例

$$f(u) = \begin{cases} 1(u > 0) \\ 0(u < 0) \end{cases}$$

入出力関係

$$y = f\left(\sum w_i x_i - \theta\right)$$

入力がθを超えると出力（発火）

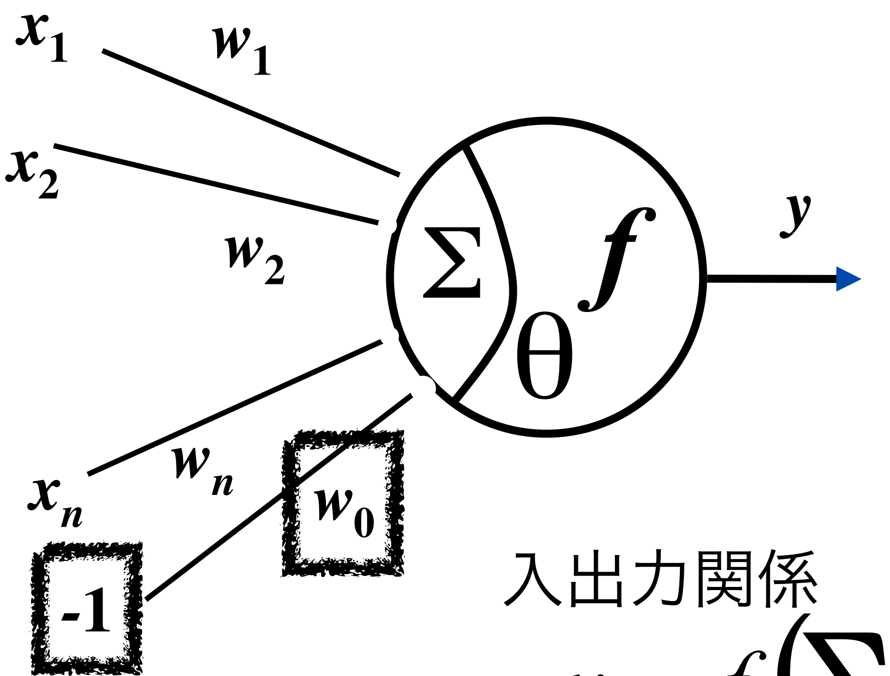
w_i を適切に変更する

||

学習

ニューロンモデル

$$\theta \longrightarrow w_0$$



簡略化

・プログラムの容易さ

・閾値の学習

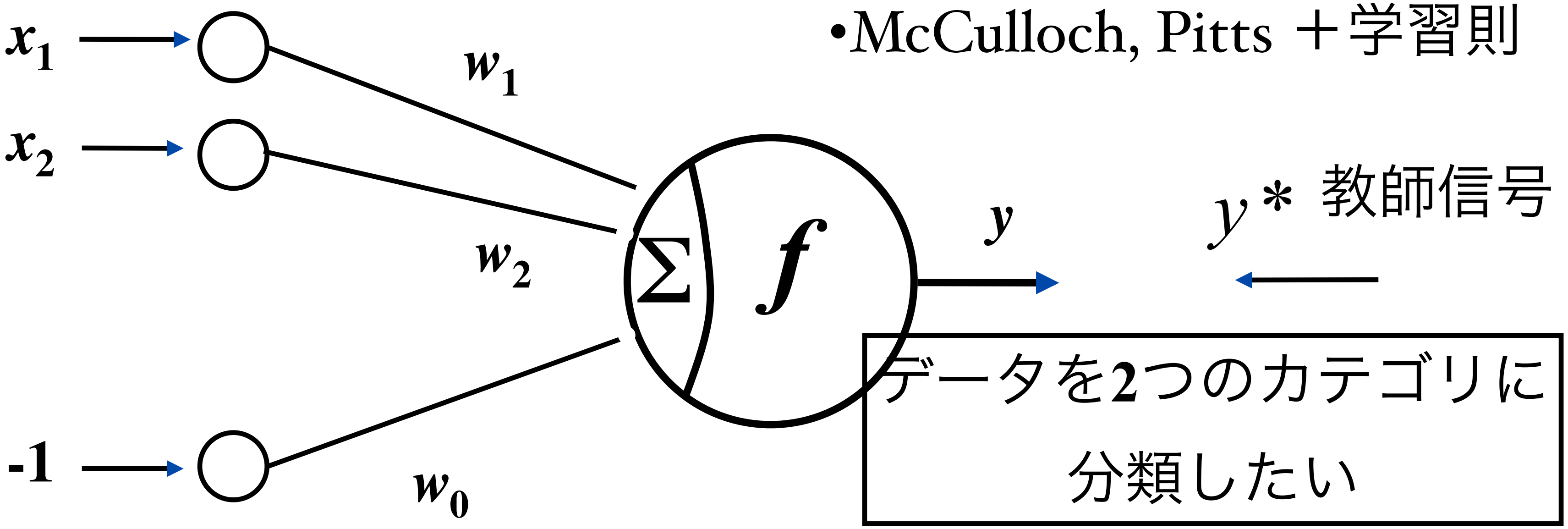
入出力関係

$$y = f\left(\sum w_i x_i - \theta\right)$$

パーセプトロン（識別機械）

- Rosenblatt (1961)

•McCulloch, Pitts ＋学習則



データを2つのカテゴリに分類したい

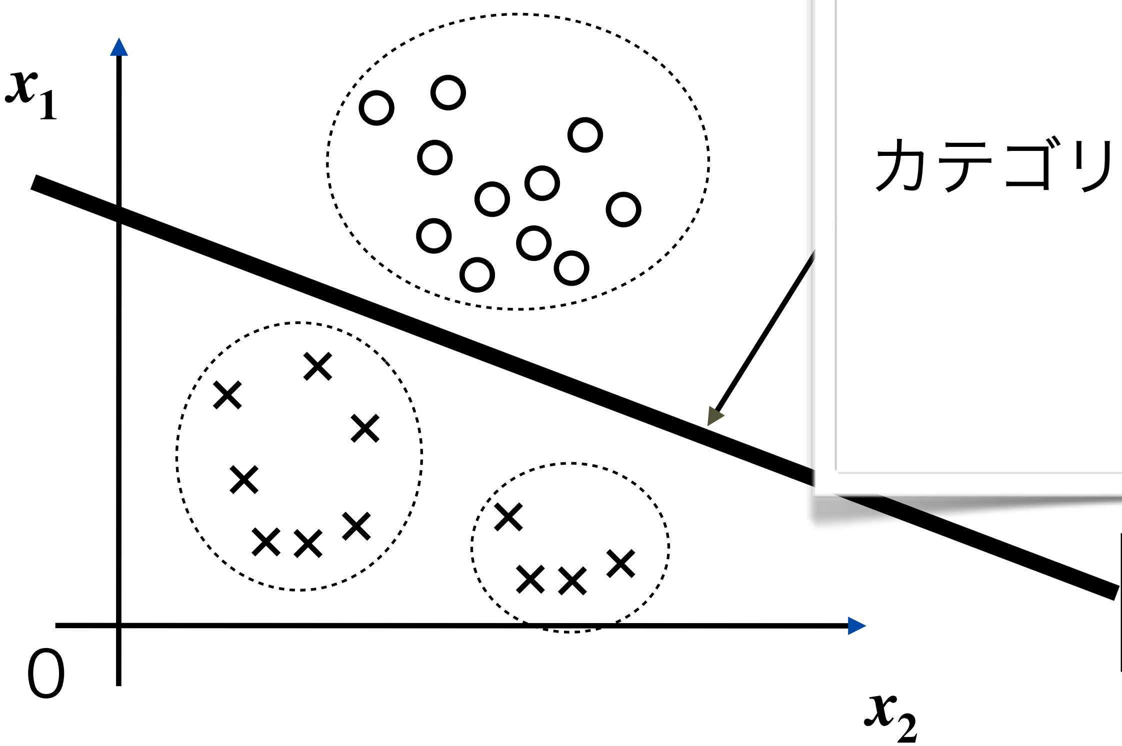
$$f(u) = \begin{cases} 1 (u > 0) \\ 0 (u < 0) \end{cases}$$

x1	x2	x3	・・・	y*
0.2	23	4.5	・・・	○
5.0	3.3	2.2	・・・	×
・・・	・・・	・・・	・・・	・・・
				○
				×
				×

パーセプトロン(2)

$$y = f(\sum w_i x_i - w_0)$$

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n - w_0$$



ニューロンが表す（超）平面

が

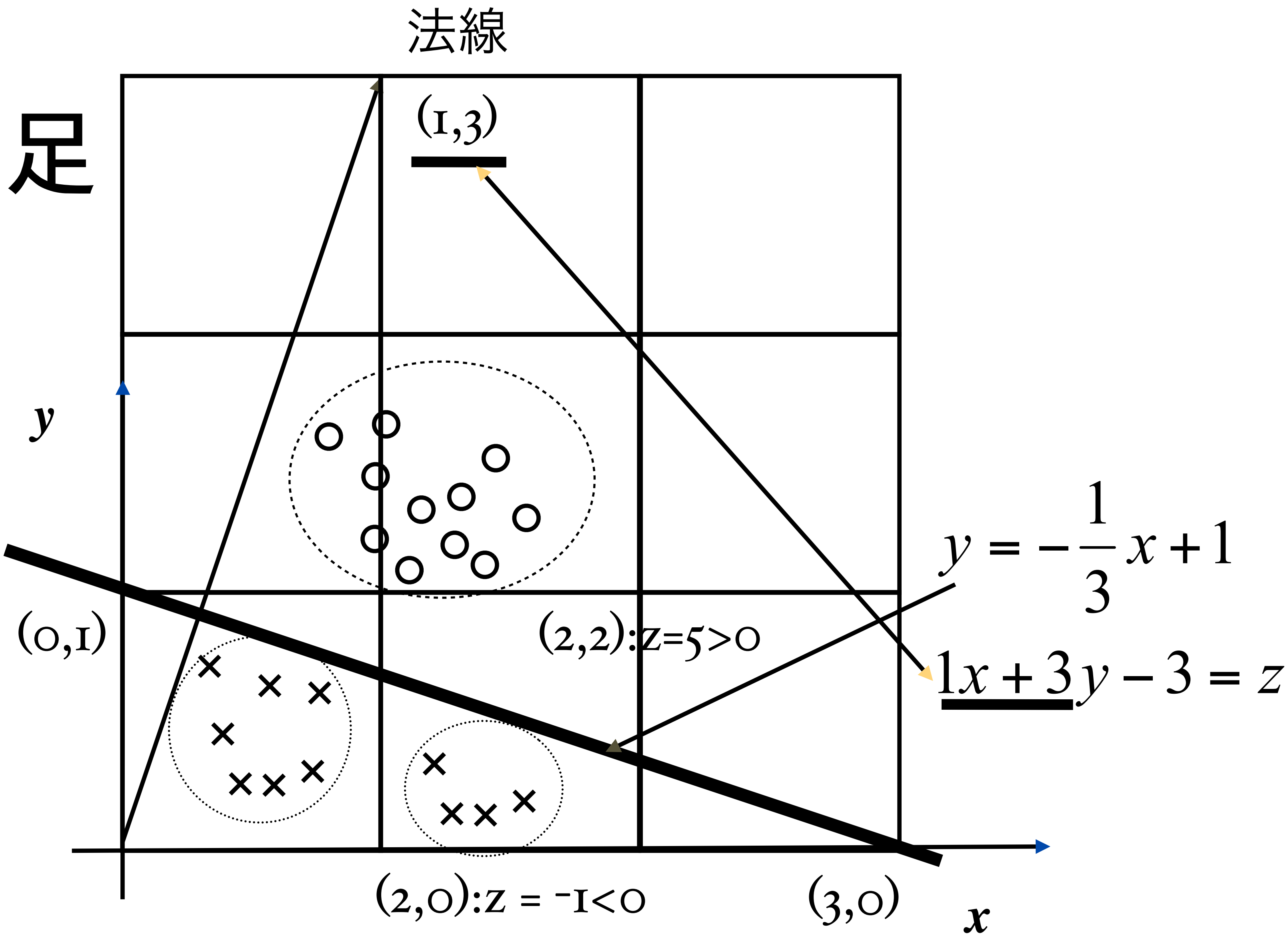
カテゴリ分離平面になるよう

に

学習

w_iの調整

補足



パーセプトロン学習則

シナプス荷重ベクトル

$$W = \{w_1, w_2, \cdots, w_n\}$$

y=X・WT

X=

x1	x2	x3	・・・	y*
x11	x12	x13	・・・	y*1
x21	x22	x23	・・・	y*2
・・・	・・・	・・・	・・・	・・・
			xnn	

$$y = \begin{cases} 1 (X \bullet W^T > 0) \\ 0 (X \bullet W^T \le 0) \end{cases}$$

荷重ベクトルの更新：学習

$$\Delta W^T = -\varepsilon \underbrace{(y - y^*)}_{\text{誤差}} X$$

0<ε<<1：（学習係数）

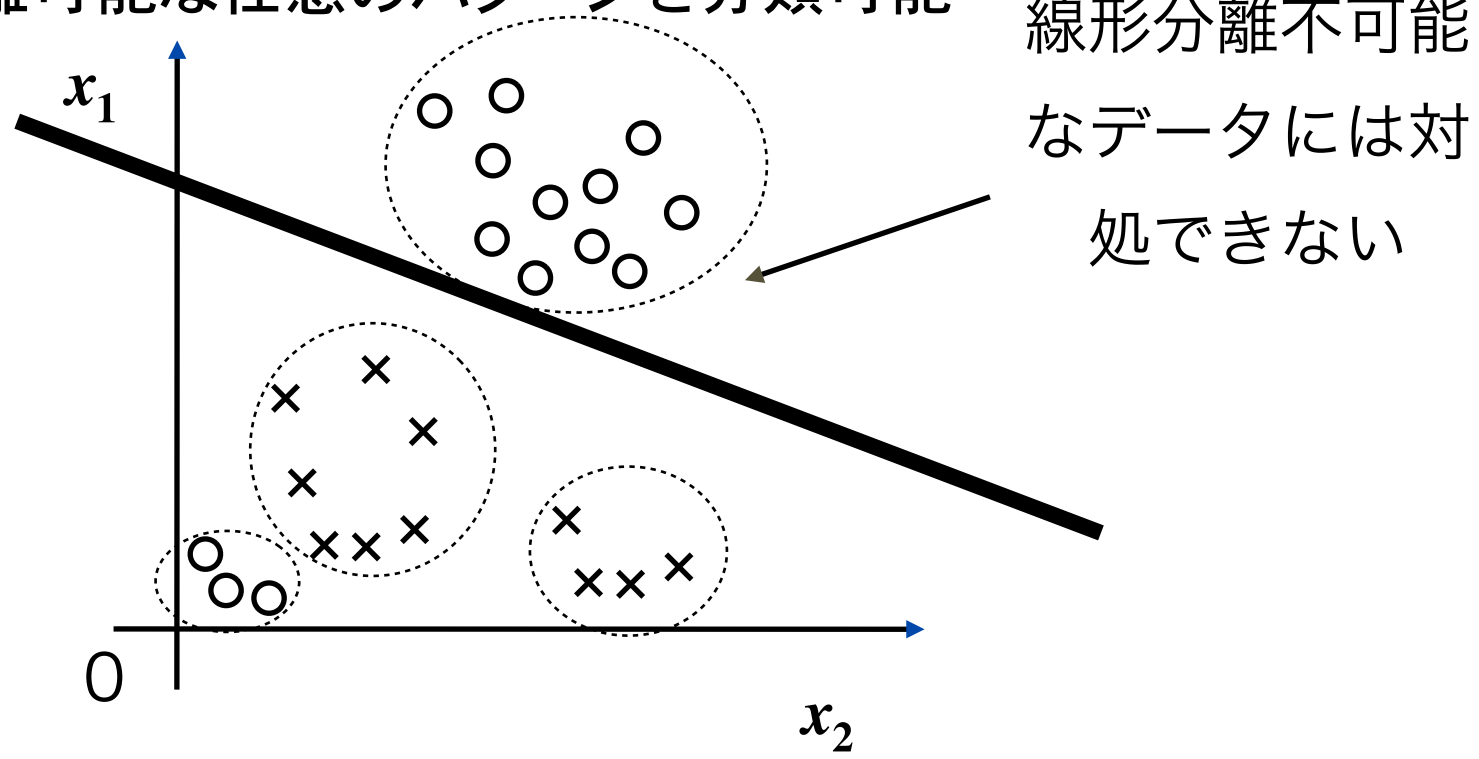
W(t+1) = W(t) + ΔW　により更新

学習則　W(t+1) = W(t) + ΔW

収束定理：有限回の学習で正しい識別をする

パーセプトロンの限界

線形分離可能な任意のパターンを分類可能



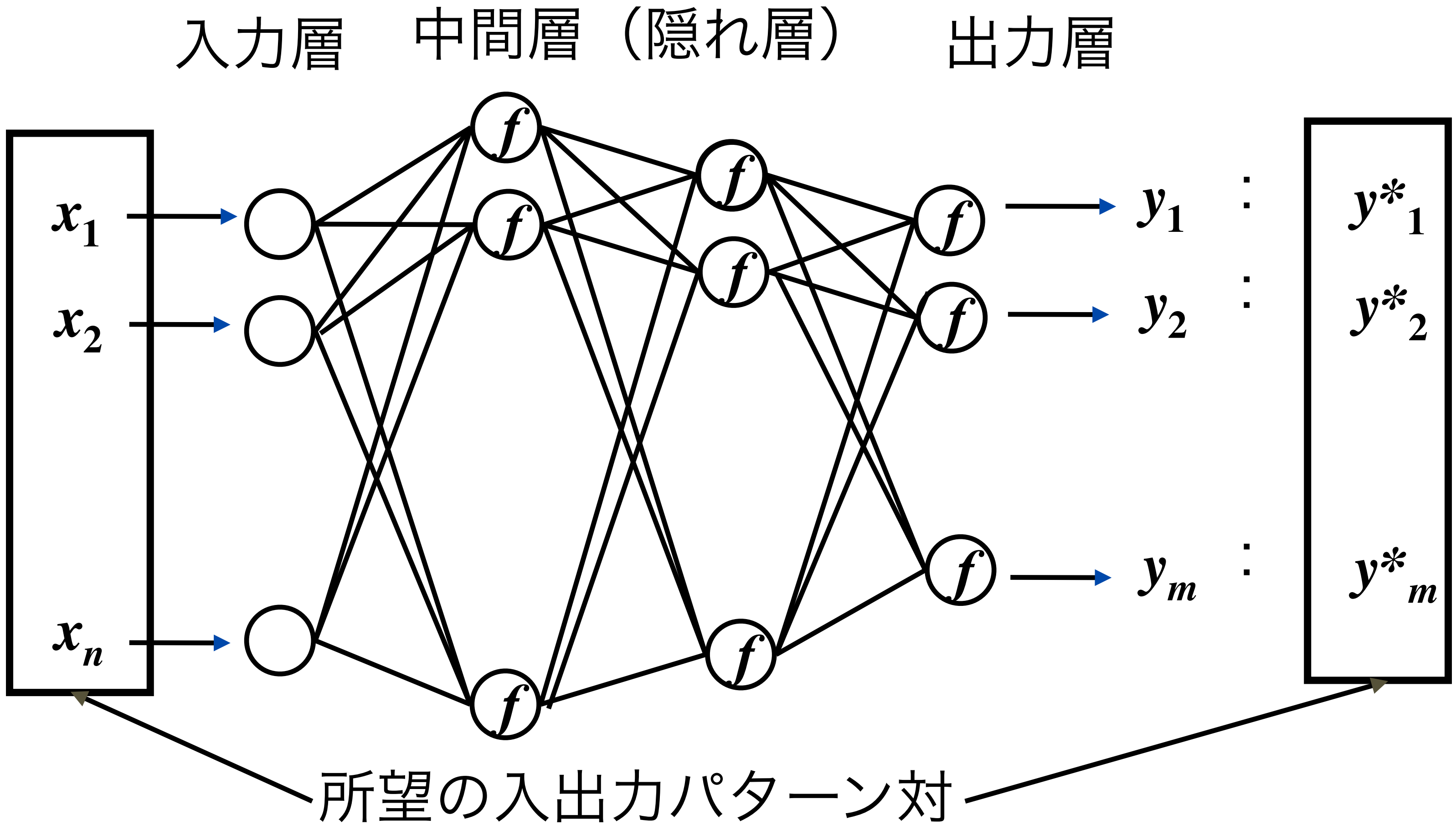
線形分離不可能な問題

○　ニューロンを多数つなげる

○　非線形関数f

f:(微分可能な)非線形関数

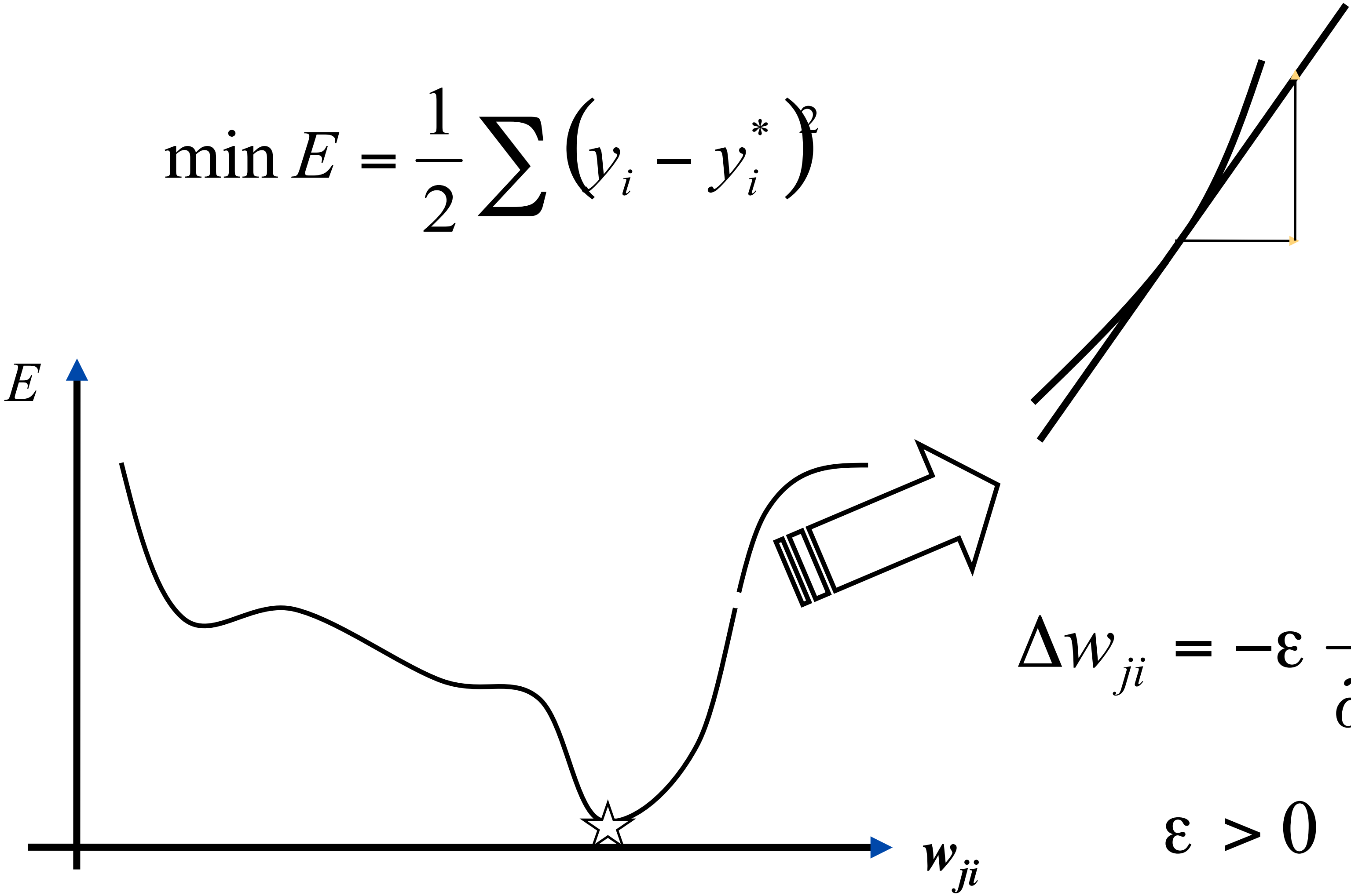
階層型ニューラルネットワーク



$$\min E = \frac{1}{2}\sum\left(y_p - y_p^*\right)^2$$

階層型NNの学習則

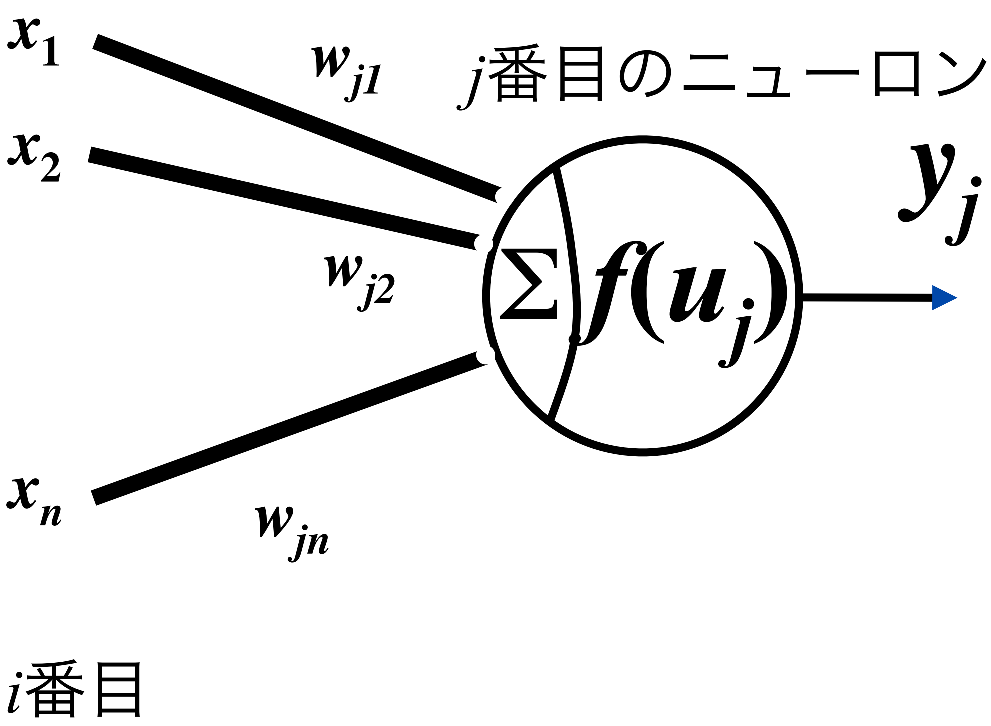
$$\min E = \frac{1}{2}\sum\left(y_i - y_i^*\right)^2$$



$$\Delta w_{ji} = -\varepsilon \frac{\partial E}{\partial w_{ji}}$$

$$\varepsilon > 0$$

学習則の導出の前に



$$u_j = \sum_i w_{ji} y_i$$

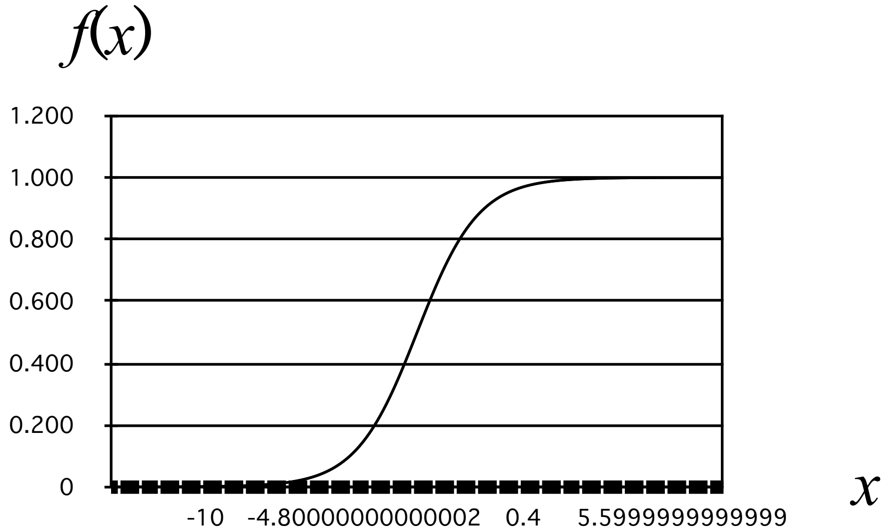
$$y_j = f\left(u_j\right)$$

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

シグモイド関数

シグモイド関数

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



○　閾値（特性）関数

○　ロジスティック方程式の一般解の一つ

$$\frac{dx}{dt} = x\,(1-x)$$

学習則 $\min E = \frac{1}{2}\sum\left(y_i - y_i^*\right)^2$

$$\boxed{u_j = \sum_i w_{ji} y_i}$$

$$\boxed{y_j = f\left(u_j\right)}$$

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

Chain rule

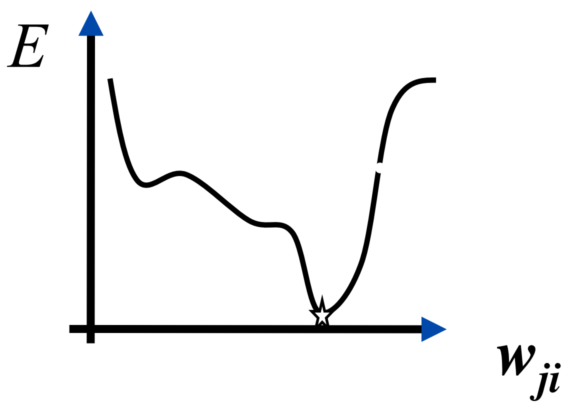
$$\frac{\partial E}{\partial w_{ji}} = \sum \frac{\partial E}{\partial y_j} \boxed{\frac{\partial y_j}{\partial u_j}} \boxed{\frac{\partial u_j}{\partial w_{ji}}}$$

$$\frac{\partial y_j}{\partial u_j} = \boxed{f'(u_j) = \underline{y_j(1 - y_j)}}$$

$$\boxed{\frac{\partial u_j}{\partial w_{ji}} = y_i}$$

シグモイド関数の場合

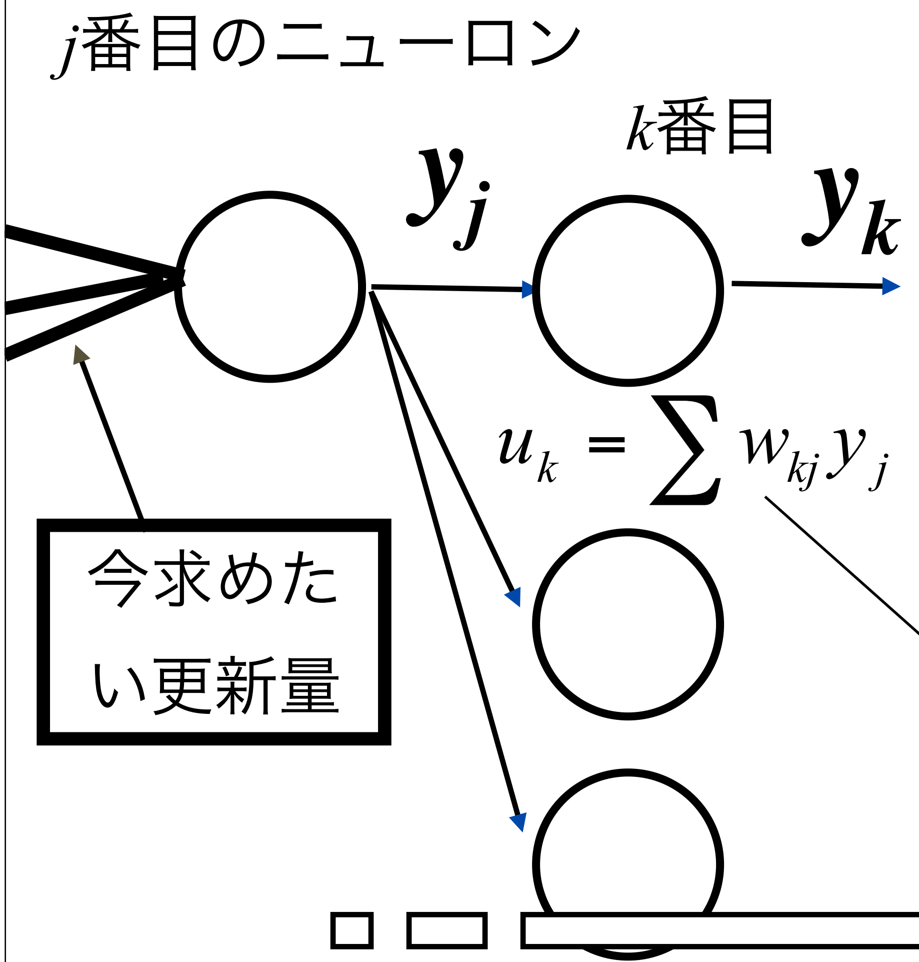
（ロジスティック写像）



逆誤差伝播法

Backpropagation

$$\frac{\partial E}{\partial w_{ji}} = \sum \boxed{\frac{\partial E}{\partial y_j}} y_j(1-y_j)y_i$$



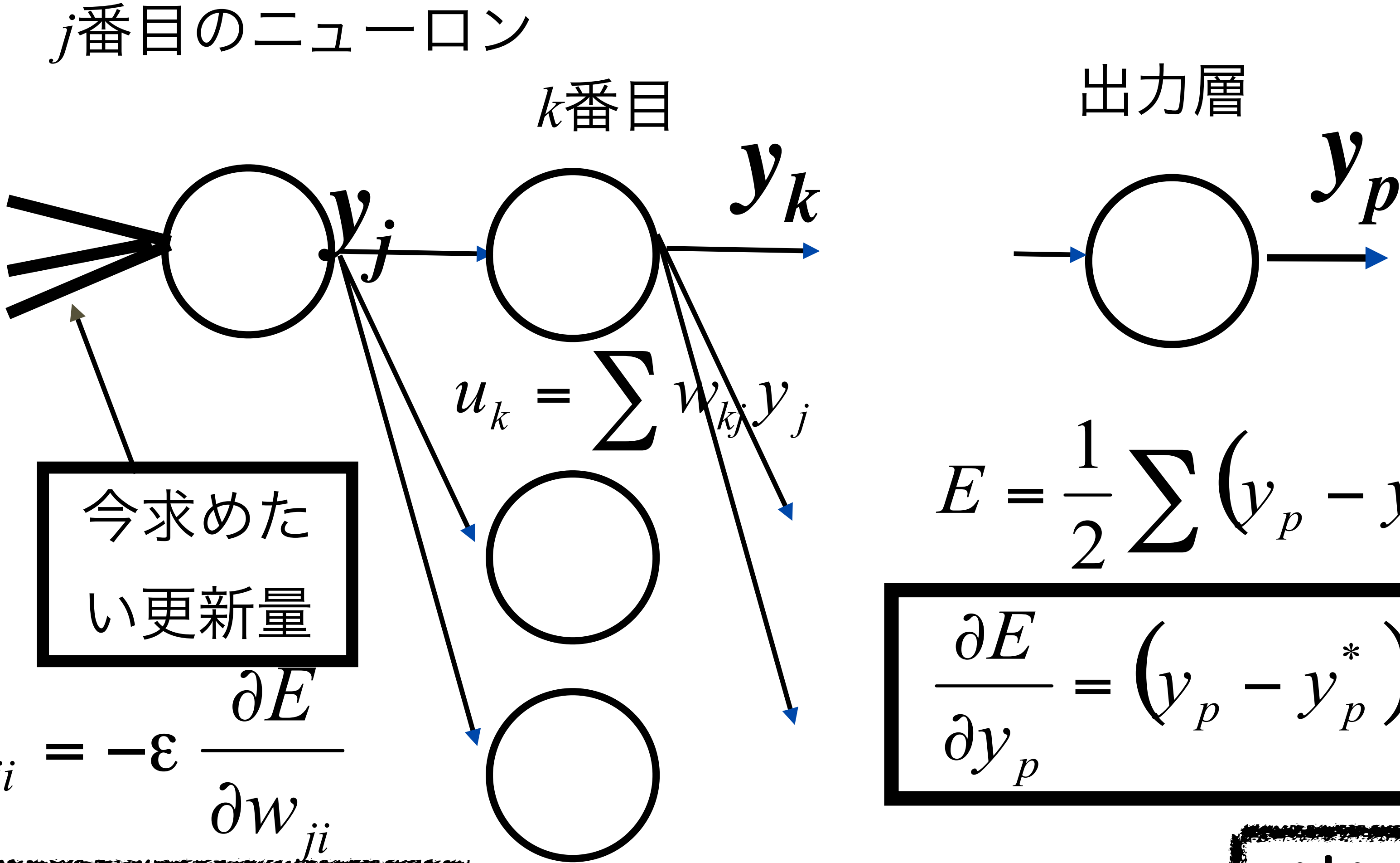
$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \boxed{\frac{\partial y_k}{\partial u_k}} \boxed{\frac{\partial u_k}{\partial y_j}}$$

$$\boxed{\frac{\partial y_k}{\partial u_k} = f'(u_k) = y_k(1-y_k)}$$

$$\boxed{\frac{\partial u_k}{\partial y_j} = w_{kj}}$$

出力y

Backpropagationまとめ



$$u_k = \sum w_{kj} y_j$$

$$E = \frac{1}{2}\sum \left(y_p - y_p^*\right)^2$$

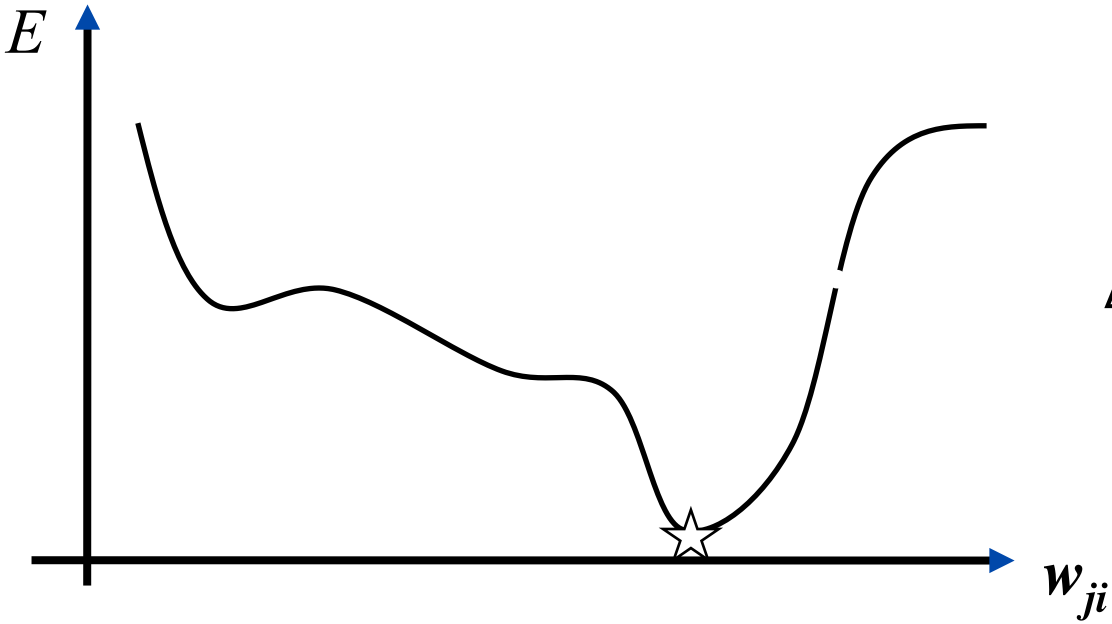
$$\boxed{\frac{\partial E}{\partial y_p} = \left(y_p - y_p^*\right)}$$

$$\Delta w_{ji} = -\varepsilon \frac{\partial E}{\partial w_{ji}}$$

具体的に数値計算可能

出力y

最急降下方向への学習



$$\Delta w_{ji} = -\varepsilon \frac{\partial E}{\partial w_{ji}}$$

Reference

1. McCulloch, W.S., Pitts, W., "A logical calculus of the ideas immanent in nervous activity," Bull. Math. Biophysics, 5, 115-133, 1943

2. Rosenblatt, F., "Principles of Neurodynamics", Spartan, 1961

3. Rumelhart, D.E., Hinton, G.E., Williams, R.J., "Learning representations by back-propagating errors", Nature, 323, 533-536, 1986

4. Rumelhart, D. E. McCleland, J.L. and the PDP Research Group, "Parallel Distributed Learning," MIT Press, 1986

5. 中野他,「ニューロコンピュータの基礎」,コロナ社,1990

6. 甘利,坂田編,「脳とニューラルネット」,朝倉書店,1994

7. 萩原,「ニューロ・ファジィ・遺伝的アルゴリズム」,産業図書,1994

8. 谷萩編,「ニューラルネットワークとファジィ情報処理」,コロナ社,1998