知能科学：ニューラルネットワーク

平井 慎一

立命館大学 ロボティクス学科

講義の流れ

1. ニューロンモデル
2. 近似定理
3. 学習
4. まとめ

ニューラルネットワーク

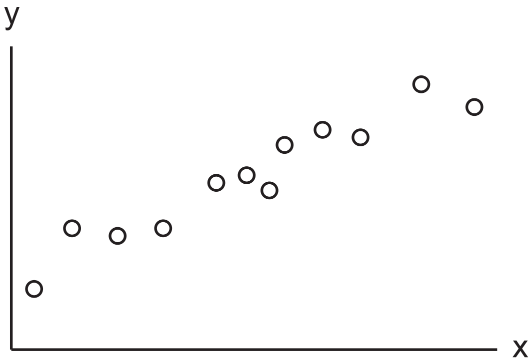
ニューラルネットワーク (Neural Network)

信号を扱う基本技術の一つ
深層学習 (Deep Learning) の基礎

深層学習 (Deep Learning)

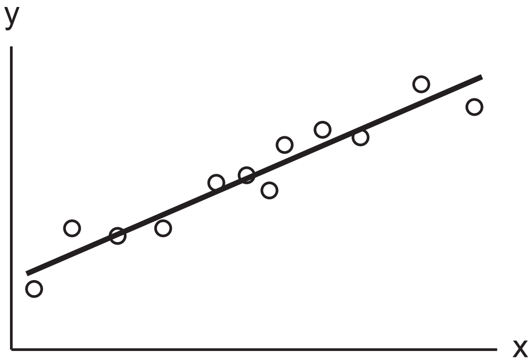
アルファ碁で使用
データを与えて，規則を自動獲得
(犬と猫の画像を与えて，犬と猫を区別する)

線形



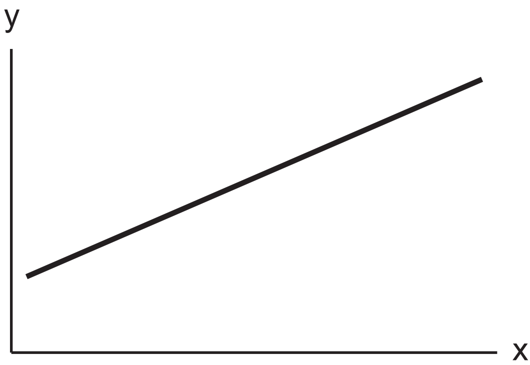
データ

線形



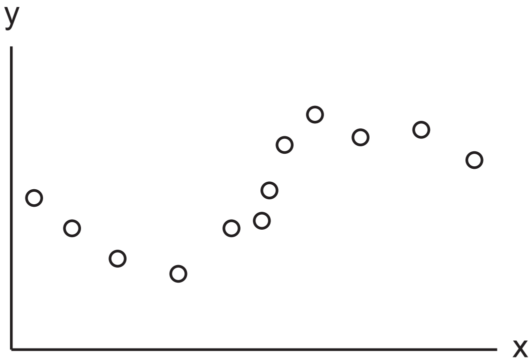
直線で近似

線形



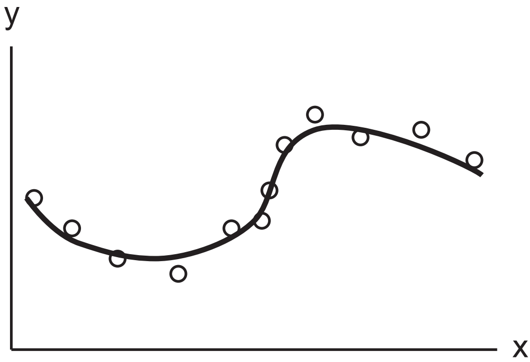
線形関数 $y = ax + b$

非線形



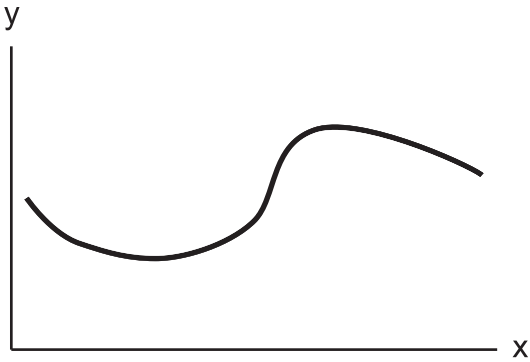
データ

非線形



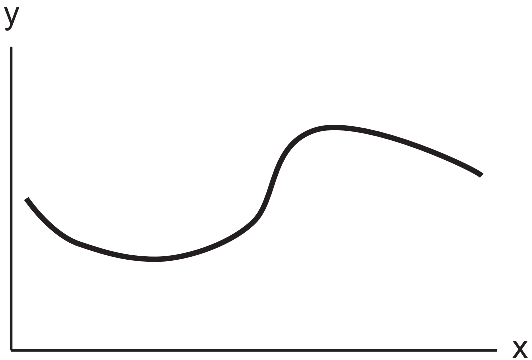
曲線で近似

非線形



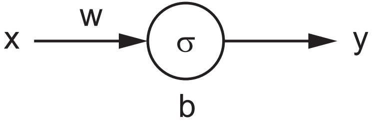
非線形関数？

非線形



非線形関数　ニューラルネットワーク

ニューロン（神経細胞）モデル



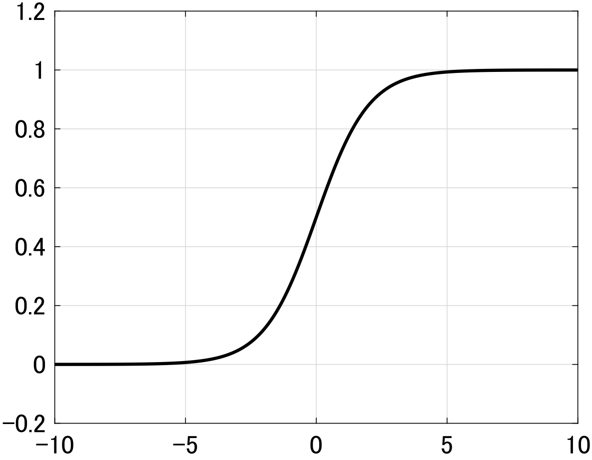
$$y = \sigma(wx + b)$$

x	入力	y	出力
w	重み	b	バイアス

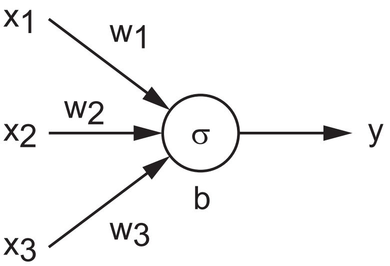
w，b は定数

シグモイド関数

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



ニューロン（神経細胞）モデル

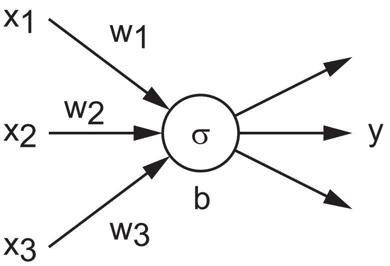


$$y = \sigma(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)$$

x_1, x_2, x_3	入力	y	出力
w_1, w_2, w_3	重み	b	バイアス

w_1, w_2, w_3，b は定数

ニューロン（神経細胞）モデル

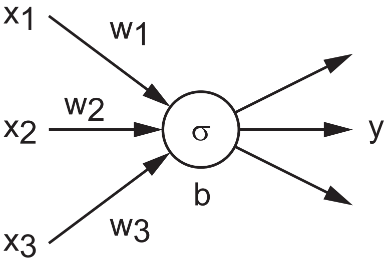


$$y = \sigma(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)$$

x_1, x_2, x_3	入力	y	出力
w_1, w_2, w_3	重み	b	バイアス

w_1, w_2, w_3，b は定数

ニューロン（神経細胞）モデル

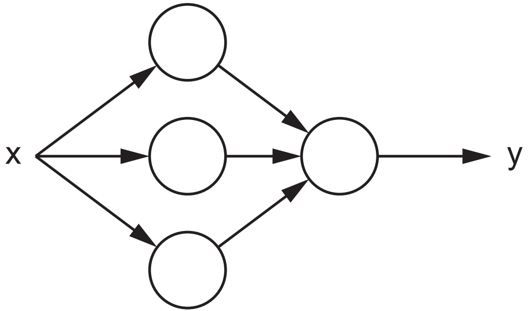


$$y = \sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b)$$

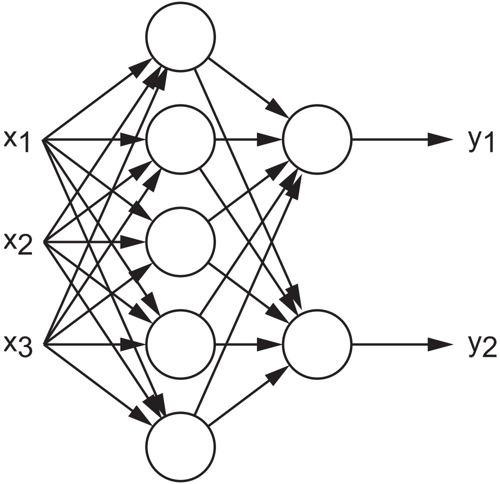
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

入力ベクトル　　重みベクトル

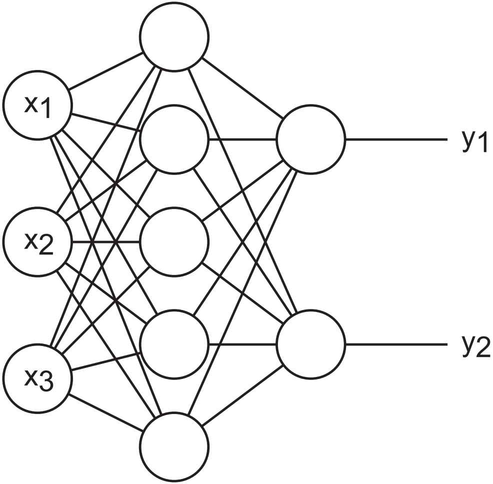
ニューラルネットワーク (1入力1出力)



ニューラルネットワーク (3 入力 2 出力)

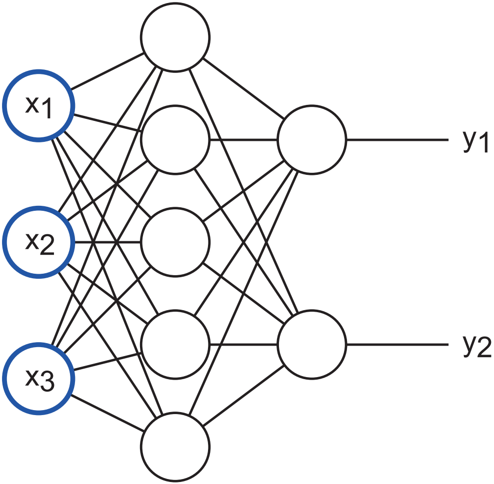


ニューラルネットワーク (3 入力 2 出力)



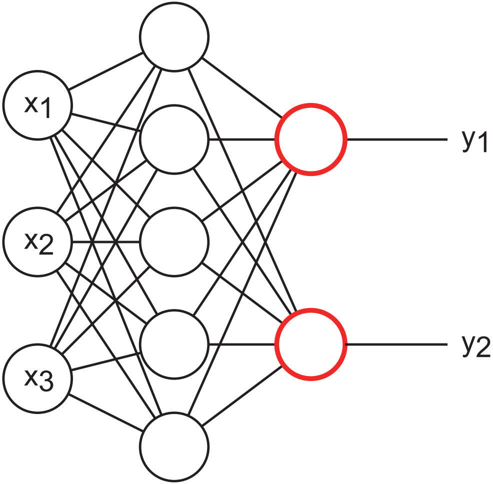
入力信号を出力するニューロンを導入

ニューラルネットワーク (3入力2出力)



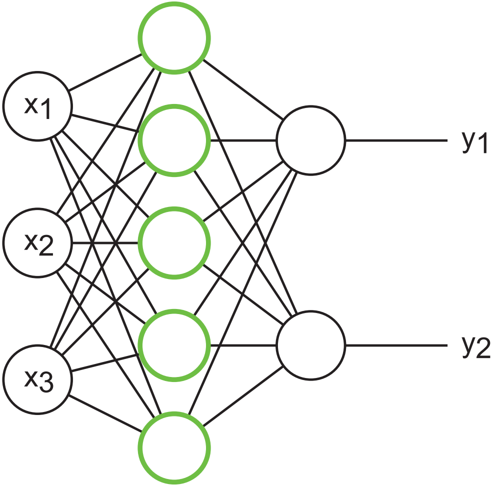
入力ニューロン

ニューラルネットワーク (3 入力 2 出力)



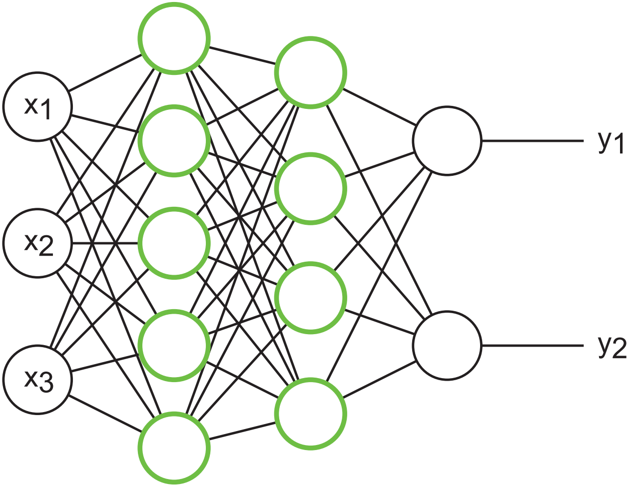
出力ニューロン

ニューラルネットワーク (3 入力 2 出力)



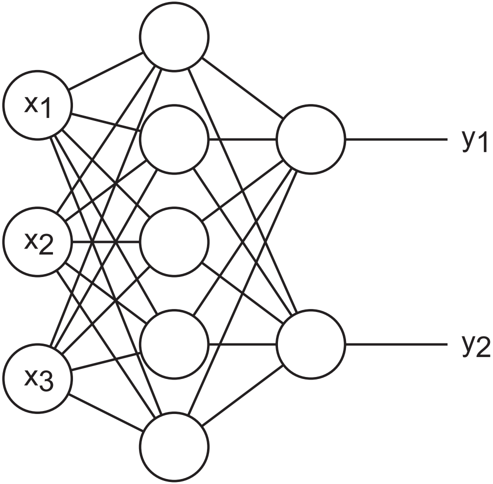
隠れニューロン

ニューラルネットワーク (3 入力 2 出力)



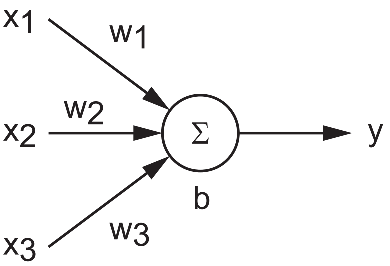
複数の隠れ層

ニューラルネットワークの出力



出力信号 $y_1, y_2 \in [0, 1]$

総和ニューロン

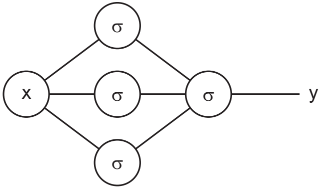


$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

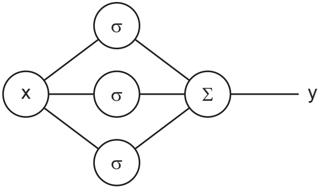
x_1, x_2, x_3	入力	y	出力
w_1, w_2, w_3	重み	b	バイアス

w_1, w_2, w_3, b は定数

ニューラルネットワークの出力

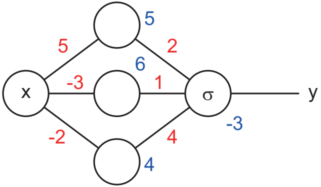


出力ニューロン：シグモイド
出力信号：$y \in [0, 1]$

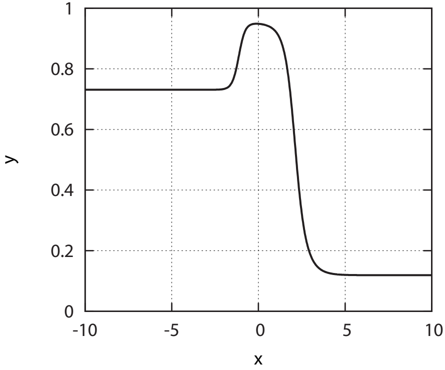


総和
$y \in (-\infty, \infty)$

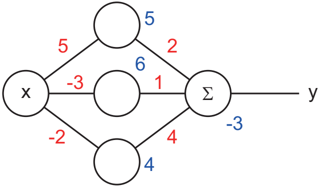
例



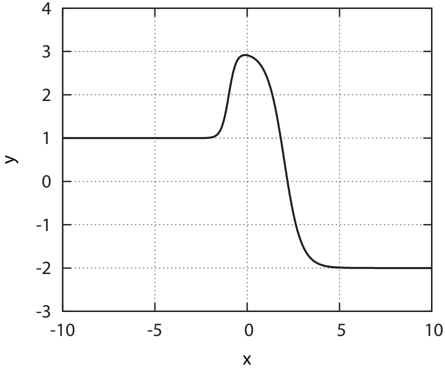
$$
\begin{aligned}
u_1 &= \sigma(5x + 5) \\
u_2 &= \sigma(-3x + 6) \\
u_3 &= \sigma(-2x + 4) \\
y &= \sigma(2u_1 + u_2 + 4u_3 - 3)
\end{aligned}
$$



例



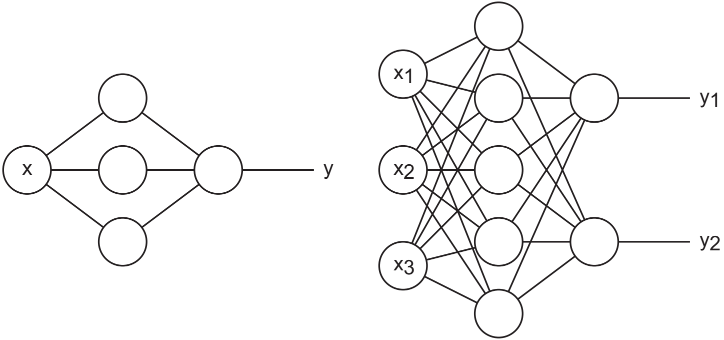
$$
\begin{aligned}
u_1 &= \sigma(5x+5) \\
u_2 &= \sigma(-3x+6) \\
u_3 &= \sigma(-2x+4) \\
y &= 2u_1 + u_2 + 4u_3 - 3
\end{aligned}
$$



基本定理

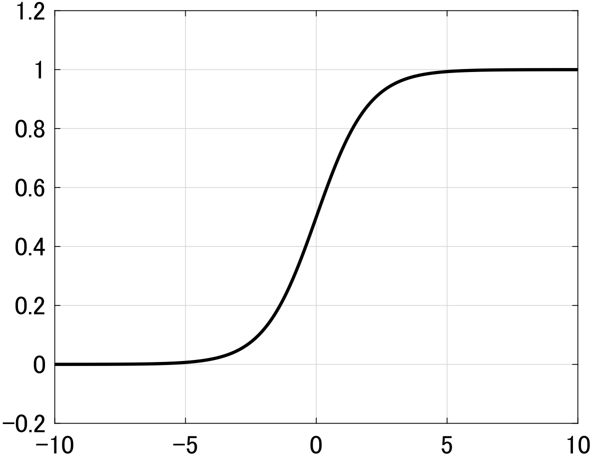
ニューラルネットワークの近似定理

隠れ層が 1 層のニューラルネットワークは,
任意の関数を任意の精度で近似することができる



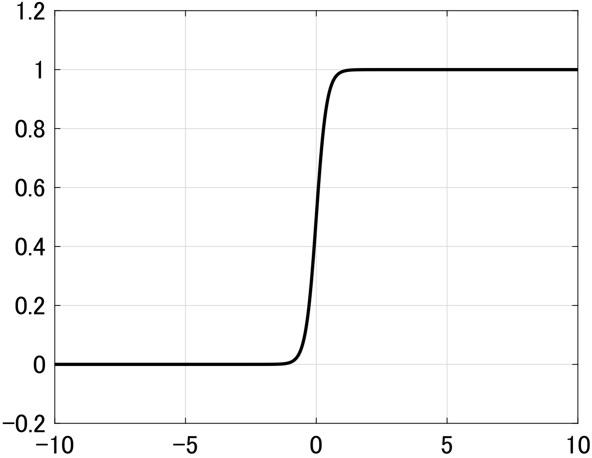
ニューロンモデルの挙動

$y = \sigma(wx + b)$ $w = 1.00$ $b = 0.00$



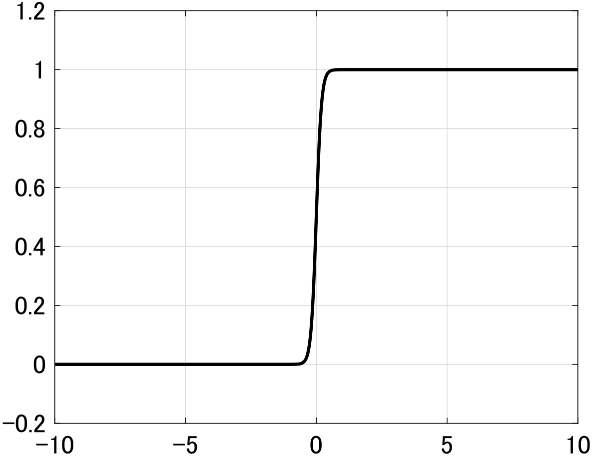
ニューロンモデルの挙動

$y = \sigma(wx + b)$ $w = 5.00$ $b = 0.00$



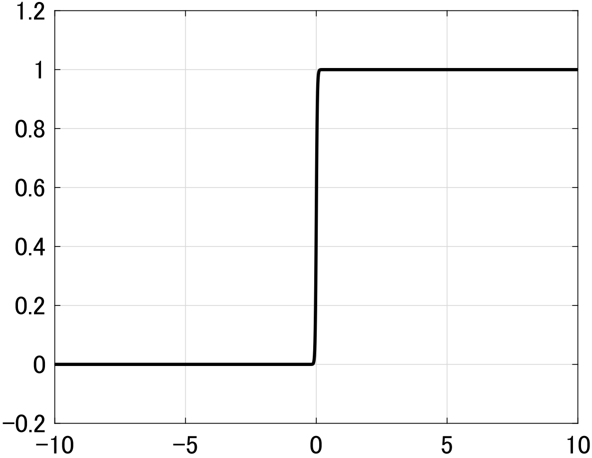
ニューロンモデルの挙動

$y = \sigma(wx + b)$　　$w = 10.00$　　$b = 0.00$



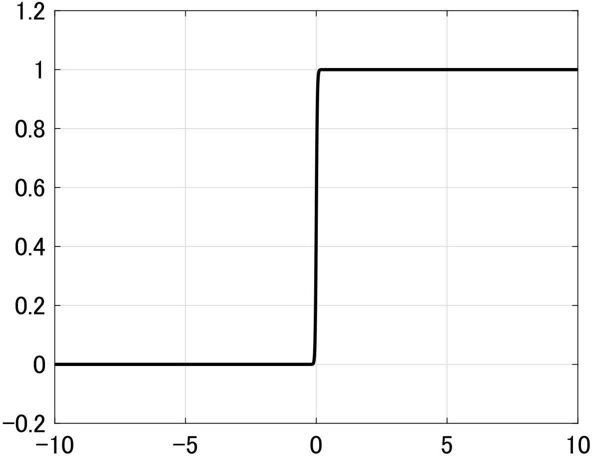
ニューロンモデルの挙動

$y = \sigma(wx + b)$ $w = 50.00$ $b = 0.00$



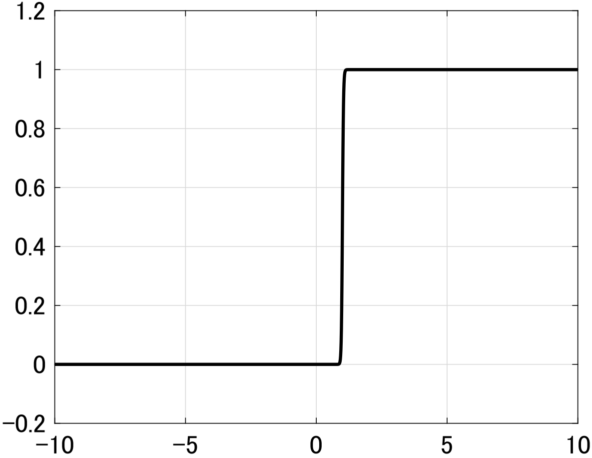
ニューロンモデルの挙動

$y = \sigma(wx + b)$ $w = 50.00$ $b = 0.00$



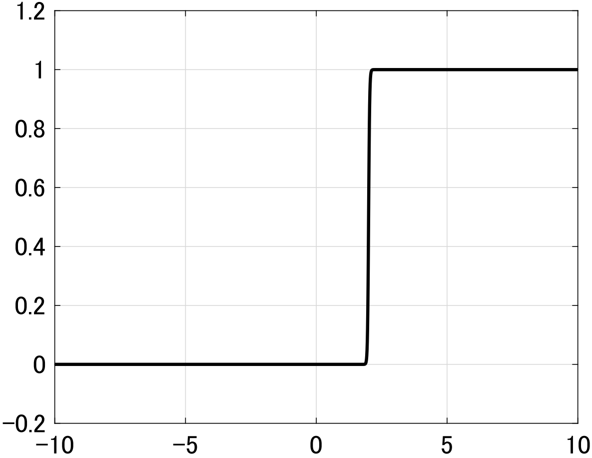
ニューロンモデルの挙動

$y = \sigma(wx + b) \qquad w = 50.00 \quad b = -50.00$



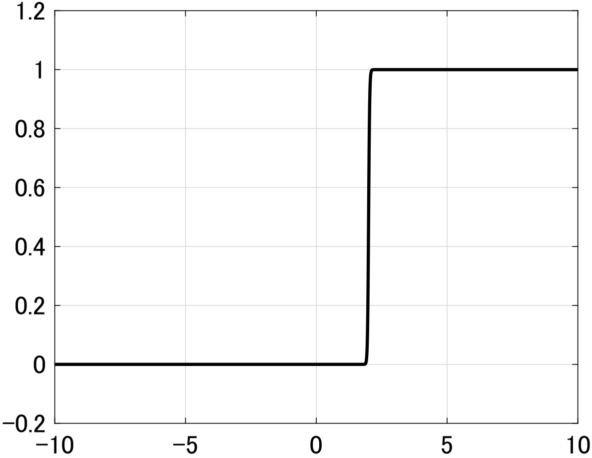
ニューロンモデルの挙動

$y = \sigma(wx + b)$ $\quad w = 50.00 \quad b = -100.00$

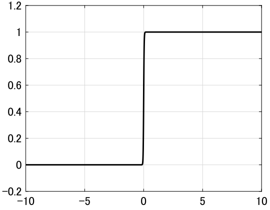


ニューロンモデルの挙動

$y = \sigma(wx + b)$ $\quad wx + b = 0 \quad -b/w = 2$

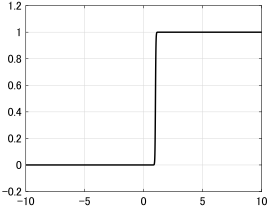


パルス関数の生成



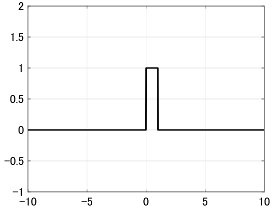
$\times 1$

$+$

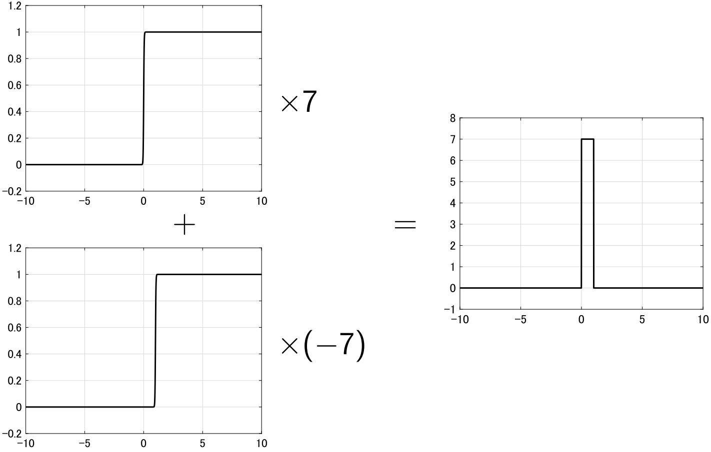


$\times(-1)$

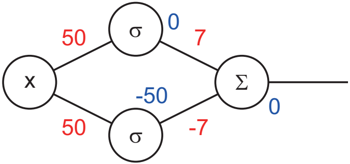
$=$



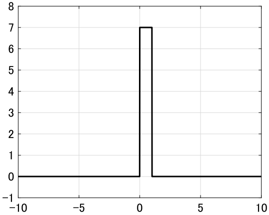
パルス関数の生成



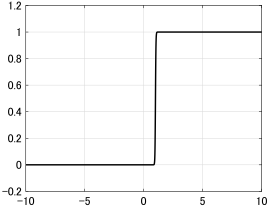
パルス関数の生成



=

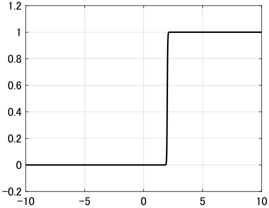


パルス関数の生成



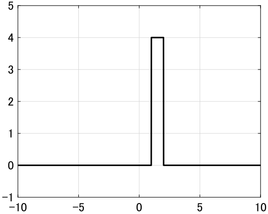
$\times 4$

$+$

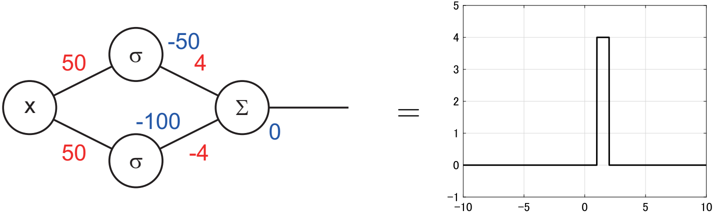


$\times(-4)$

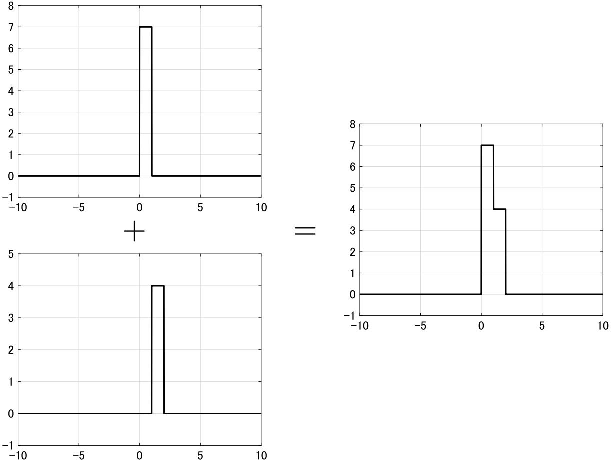
$=$



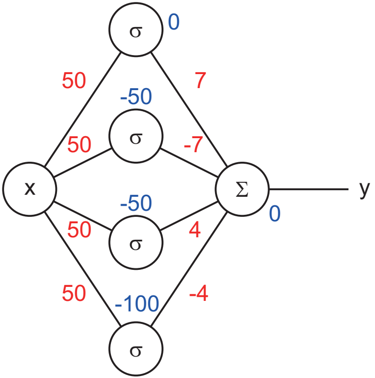
パルス関数の生成



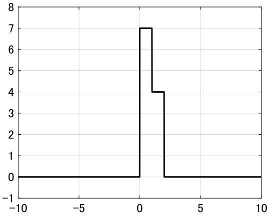
階段関数の生成



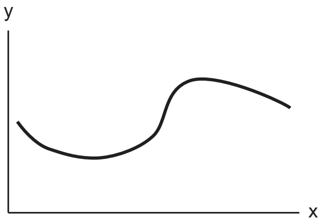
階段関数の生成



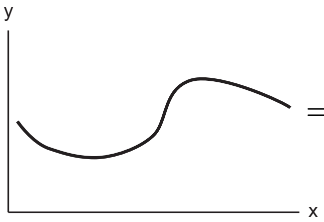
=



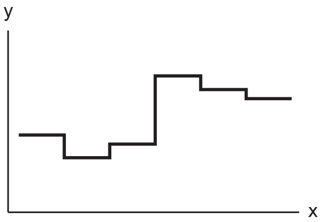
関数の近似



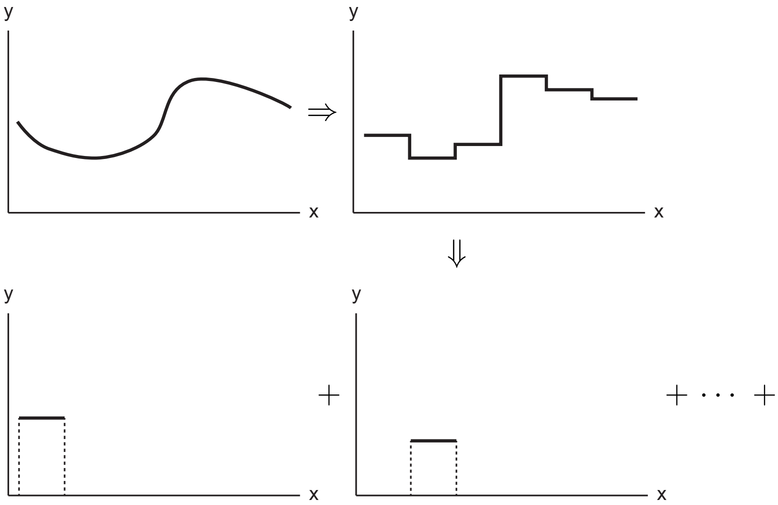
関数の近似



$\Rightarrow$



関数の近似



二変数関数の近似

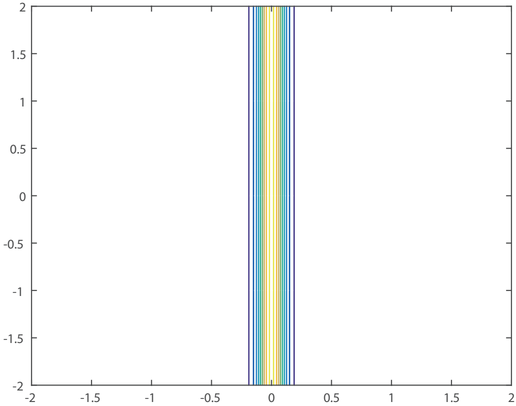
二次元ステップ関数を生成するネットワーク

$$\Downarrow$$

任意の二次元関数を近似できる.

二変数関数の近似

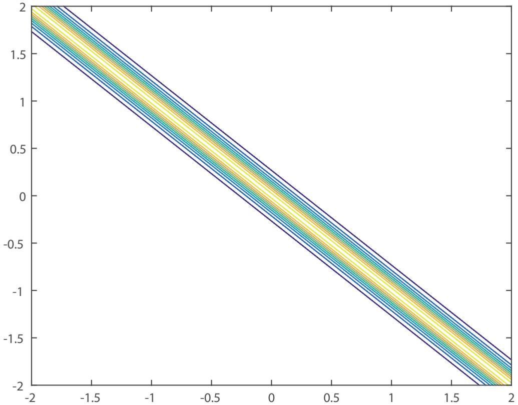
$$y = \sigma(a_1x_1 + a_2x_2 + c) - \sigma(a_1x_1 + a_2x_2 - c)$$



$$a_1 = r\cos\theta,\quad a_2 = r\sin\theta,\quad r = 20,\quad \theta = 0,\quad c = 1$$

二変数関数の近似

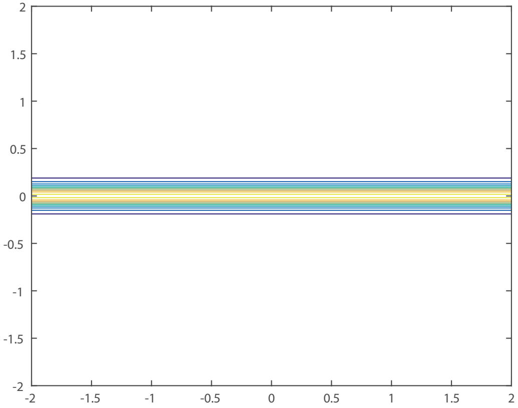
$$y = \sigma(a_1 x_1 + a_2 x_2 + c) - \sigma(a_1 x_1 + a_2 x_2 - c)$$



$$a_1 = r\cos\theta,\quad a_2 = r\sin\theta,\quad r = 20,\quad \theta = \pi/4,\quad c = 1$$

二変数関数の近似

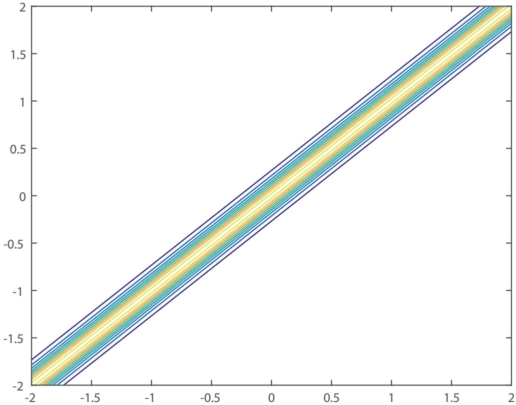
$$y = \sigma(a_1 x_1 + a_2 x_2 + c) - \sigma(a_1 x_1 + a_2 x_2 - c)$$



$$a_1 = r\cos\theta,\ \ a_2 = r\sin\theta,\ \ r = 20,\ \ \theta = 2\pi/4,\ \ c = 1$$

二変数関数の近似

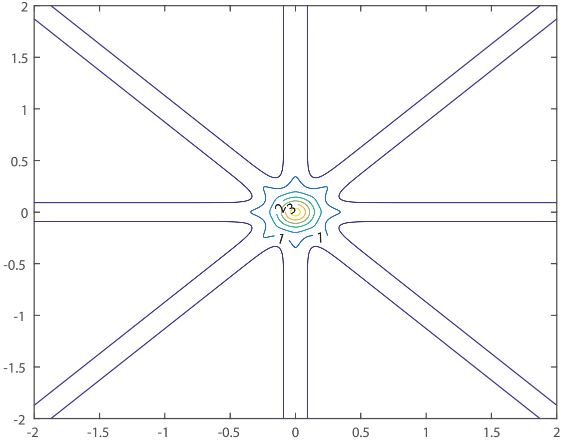
$$y = \sigma(a_1x_1 + a_2x_2 + c) - \sigma(a_1x_1 + a_2x_2 - c)$$



$$a_1 = r\cos\theta, \quad a_2 = r\sin\theta, \quad r = 20, \quad \theta = 3\pi/4, \quad c = 1$$

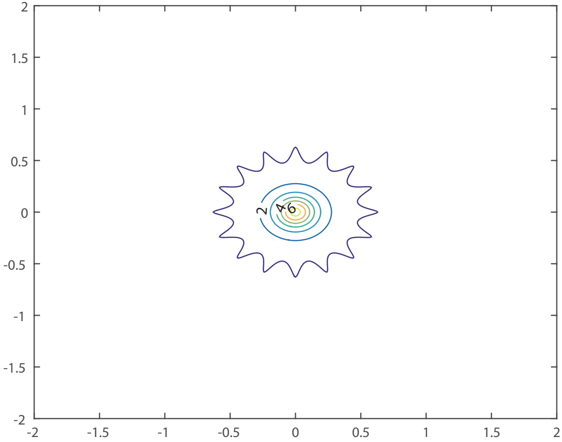
二変数関数の近似

帯状関数の重ね合わせ N = 8



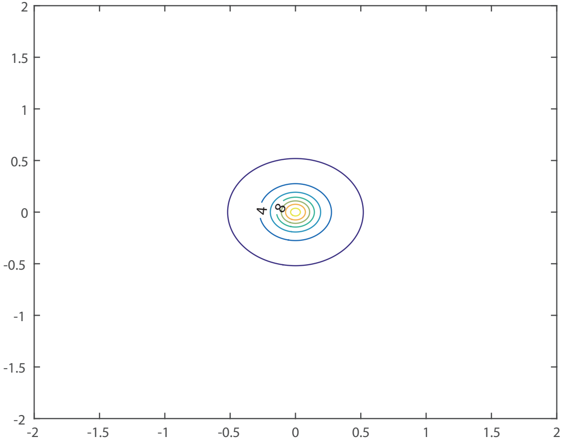
二変数関数の近似

帯状関数の重ね合わせ $N = 16$



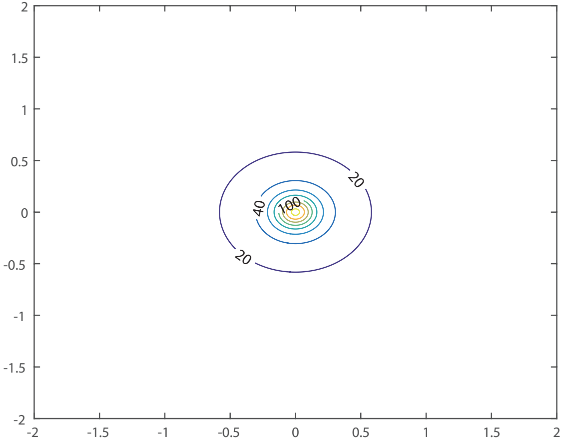
二変数関数の近似

帯状関数の重ね合わせ N = 32

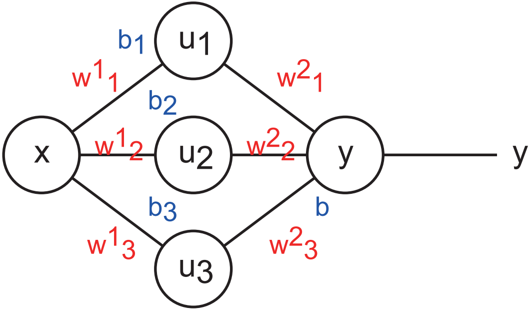


二変数関数の近似

帯状関数の重ね合わせ $N = 360$

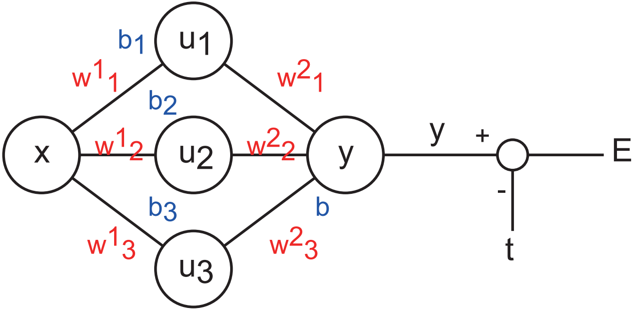


教師信号と誤差



$$u_1 = \sigma(w_1^1 x + b_1)$$
$$u_2 = \sigma(w_2^1 x + b_2)$$
$$u_3 = \sigma(w_3^1 x + b_3)$$
$$y = \sigma(w_1^2 u_1 + w_2^2 u_2 + w_3^2 u_3 + b)$$

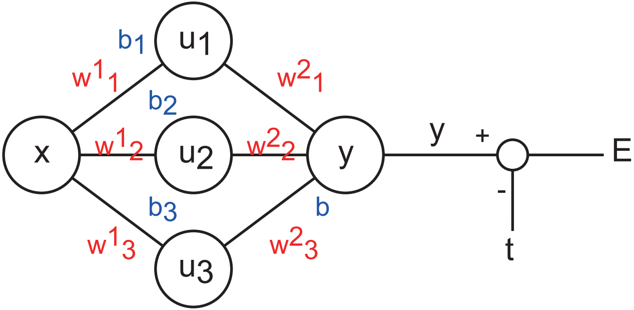
教師信号と誤差



t 教師信号

$$E = \frac{1}{2}(y - t)^2$$ 誤差

教師信号と誤差



学習

誤差 E が最小になるように，重みとバイアスを修正

最急降下法

関数 $f(x, y)$ が最小になる (x, y) を求める．

$$\min \quad f(x, y)$$

漸化式：
現在の値 (x_n, y_n) から次の値 (x_{n+1}, y_{n+1}) を計算

$$x_{n+1} = x_n - \alpha \frac{\partial f}{\partial x}(x_n, y_n)$$
$$y_{n+1} = y_n - \alpha \frac{\partial f}{\partial y}(x_n, y_n)$$

α：小さな正の定数

最急降下法

勾配ベクトル： 関数値 $f(x, y)$ が最も増加する方向

$$\begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

$$\Downarrow$$

関数値 $f(x, y)$ が最も減少する方向

$$-\begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

漸化式：

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \alpha \left(-\begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix} \right)$$

最急降下法

誤差 E が最小になるように，重みとバイアスを修正

$$\min \quad E(w_1^1, w_2^1, w_3^1, b_1, b_2, b_3, w_1^2, w_2^2, w_3^2, b)$$

偏微分

$$\frac{\partial E}{\partial w_1^1}, \frac{\partial E}{\partial w_2^1}, \frac{\partial E}{\partial w_3^1}, \frac{\partial E}{\partial b_1}, \frac{\partial E}{\partial b_2}, \frac{\partial E}{\partial b_3}, \frac{\partial E}{\partial w_1^2}, \frac{\partial E}{\partial w_2^2}, \frac{\partial E}{\partial w_3^2}, \frac{\partial E}{\partial b}$$

を計算し，再急降下法を適用

$$w_1^1 := w_1^1 - \alpha \frac{\partial E}{\partial w_1^1},\ w_2^1 := w_2^1 - \alpha \frac{\partial E}{\partial w_2^1},\ \cdots,\ b := b - \alpha \frac{\partial E}{\partial b}$$

偏微分の計算

シグモイド関数

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

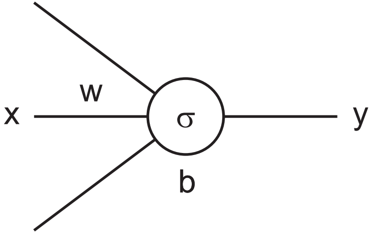
このとき

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

シグモイド関数の微分

$$\begin{aligned}\sigma'(x) &= \frac{-(-e^{-x})}{(1 + e^{-x})^2} \\ &= \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} \\ &= \sigma(x)(1 - \sigma(x))\end{aligned}$$

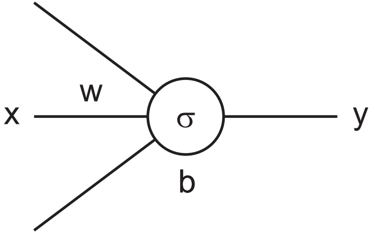
偏微分の計算



$$y = \sigma(\cdots + wx + \cdots + b)$$

$$\begin{aligned} \frac{\partial y}{\partial x} &= \sigma'(\cdots) \cdot w \\ &= \sigma(\cdots)\{1 - \sigma(\cdots)\}w \\ &= y(1-y)w \end{aligned}$$

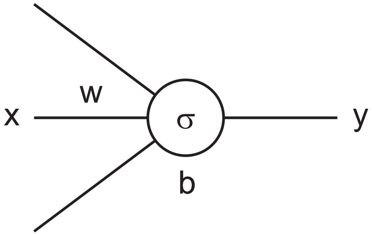
偏微分の計算



$$y = \sigma(\cdots + wx + \cdots + b)$$

$$\frac{\partial y}{\partial w} = \sigma'(\cdots) \cdot x = y(1-y)x$$
$$\frac{\partial y}{\partial b} = \sigma'(\cdots) \cdot 1 = y(1-y)$$

偏微分の計算



$$y = \sigma(\cdots + wx + \cdots + b)$$

$$\frac{\partial y}{\partial x} = y(1-y)w$$

$$\frac{\partial y}{\partial w} = y(1-y)x \qquad \frac{\partial y}{\partial b} = y(1-y)$$

偏微分の計算

$$\frac{\partial E}{\partial w_1^2} = (y - t)\,\frac{\partial y}{\partial w_1^2}$$

$$y = \sigma(\cdots + w_1^2 u_1 + \cdots)$$

$$\frac{\partial y}{\partial w_1^2} = y(1 - y)u_1$$

$$\frac{\partial E}{\partial w_1^2} = (y - t)y(1 - y)u_1$$

偏微分の計算

$$\frac{\partial E}{\partial b} = (y - t)\,\frac{\partial y}{\partial b}$$

$$y = \sigma(\cdots + b)$$

$$\frac{\partial y}{\partial b} = y(1 - y)$$

$$\frac{\partial E}{\partial b} = (y - t)y(1 - y)$$

偏微分の計算

$$\frac{\partial E}{\partial w_1^1} = (y - t)\,\frac{\partial y}{\partial w_1^1}$$

重み w_1^1 の値を変えると u_1 の値が変わる．
u_1 の値が変わると，y の値が変わる．

$$\frac{\partial y}{\partial w_1^1} = \frac{\partial y}{\partial u_1}\,\frac{\partial u_1}{\partial w_1^1}$$

$$y = \sigma(\cdots + w_1^2 u_1 + \cdots)$$

$$\frac{\partial y}{\partial u_1} = y(1 - y)w_1^2$$

偏微分の計算

$$u_1 = \sigma(\cdots + w_1^1 x + \cdots)$$

$$\frac{\partial u_1}{\partial w_1^1} = u_1(1 - u_1)x$$

$$\begin{aligned} \frac{\partial E}{\partial w_1^1} &= (y - t)y(1 - y)w_1^2 u_1(1 - u_1)x \\ &= \frac{\partial E}{\partial w_1^2}\, w_1^2(1 - u_1)x \end{aligned}$$

偏微分の計算

$$\frac{\partial E}{\partial b_1} = (y - t)\,\frac{\partial y}{\partial b_1}$$

バイアス b_1 の値を変えると u_1 の値が変わる．
u_1 の値が変わると，y の値が変わる．

$$\frac{\partial y}{\partial b_1} = \frac{\partial y}{\partial u_1}\,\frac{\partial u_1}{\partial b_1}$$

$$y = \sigma(\cdots + w_1^2 u_1 + \cdots)$$

$$\frac{\partial y}{\partial u_1} = y(1 - y)w_1^2$$

偏微分の計算

$$u_1 = \sigma(\cdots + b_1)$$

$$\frac{\partial u_1}{\partial b_1} = u_1(1 - u_1)$$

$$\begin{aligned}\frac{\partial E}{\partial b_1} &= (y - t)y(1 - y)w_1^2 u_1(1 - u_1) \\ &= \frac{\partial E}{\partial b}\, w_1^2 u_1(1 - u_1)\end{aligned}$$

誤差逆伝搬法

出力ニューロンの重みとバイアスに関する偏微分

$$\frac{\partial E}{\partial w_k^2} = (y - t)y(1 - y)u_k$$
$$\frac{\partial E}{\partial b} = (y - t)y(1 - y)$$

隠れニューロンの重みとバイアスに関する偏微分

$$\frac{\partial E}{\partial w_k^1} = \frac{\partial E}{\partial w_k^2} w_k^2(1 - u_k)x$$
$$\frac{\partial E}{\partial b_k} = \frac{\partial E}{\partial b} w_k^2 u_k(1 - u_k)$$

誤差逆伝搬法 (Back Propagation)

誤差逆伝搬法

出力ニューロンの重みとバイアスの更新

$$w_k^2 := w_k^2 - \alpha \frac{\partial E}{\partial w_k^2}$$

$$b := b - \alpha \frac{\partial E}{\partial b}$$

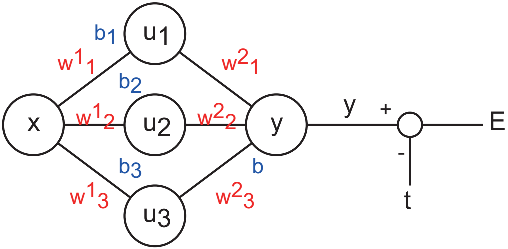
隠れニューロンの重みとバイアスの更新

$$w_k^1 := w_k^1 - \alpha \frac{\partial E}{\partial w_k^1}$$

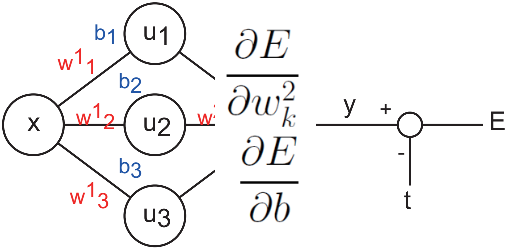
$$b_k := b_k - \alpha \frac{\partial E}{\partial b_k}$$

誤差逆伝搬法 (Back Propagation)

誤差逆伝搬法



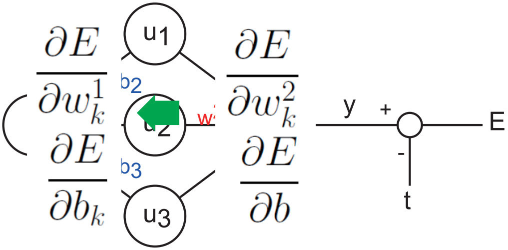
誤差逆伝搬法



$$\frac{\partial E}{\partial w_k^2}$$

$$\frac{\partial E}{\partial b}$$

誤差逆伝搬法



$\frac{\partial E}{\partial w_k^1}$
$\frac{\partial E}{\partial b_k}$
u1
u2
u3
b2
b3
$\frac{\partial E}{\partial w_k^2}$
$\frac{\partial E}{\partial b}$
y
+
-
E
t

例

入力と教師データ

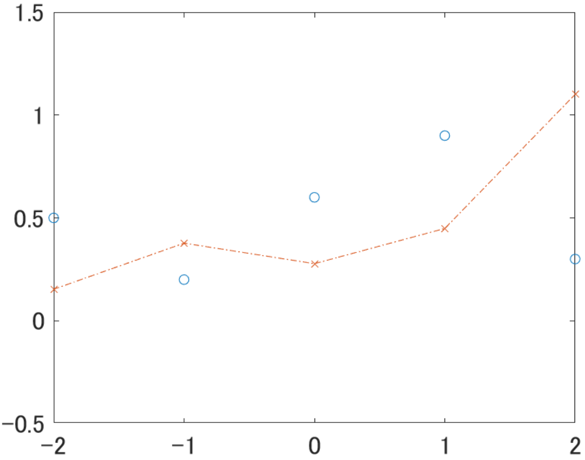
x	-2	-1	0	1	2
t	0.5	0.2	0.6	0.9	0.3

サンプルプログラム ANN_example.m
Deep Learning Toolbox が必要

隠れニューロン 10 個

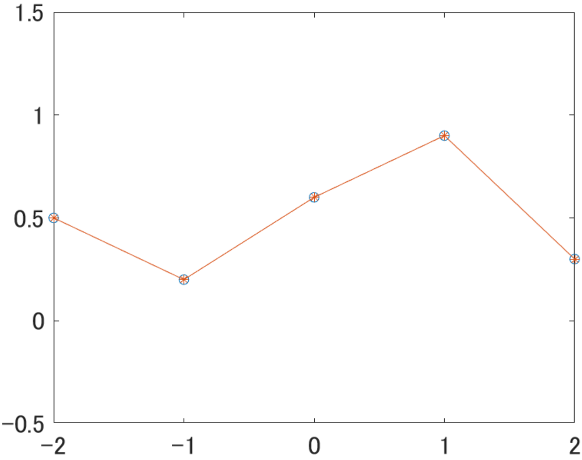
例

学習前



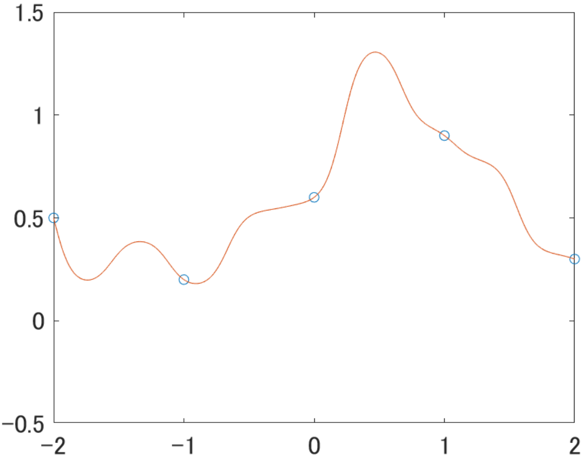
例

学習後

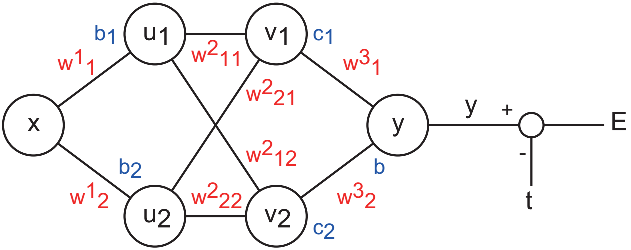


例

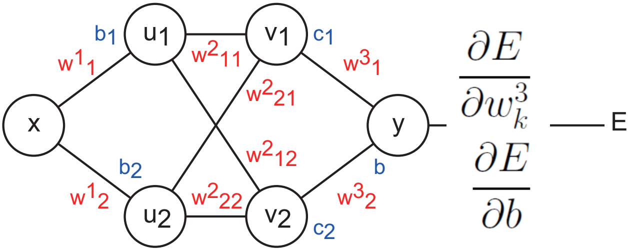
関数



誤差逆伝搬法

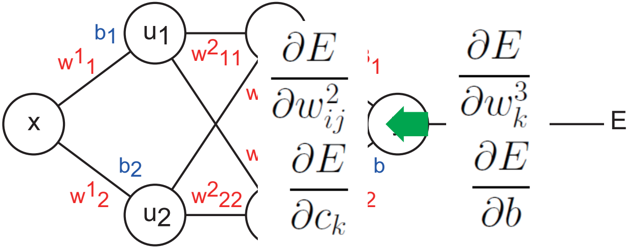

x
u1
u2
v1
v2
y
b1
b2
c1
c2
b
w11
w12
w211
w221
w212
w222
w31
w32
y
+
-
t
E

誤差逆伝搬法



x
u1
u2
v1
v2
y
b1
b2
c1
c2
b
w11
w12
w211
w221
w212
w222
w31
w32
$\frac{\partial E}{\partial w_k^3}$
$\frac{\partial E}{\partial b}$
E

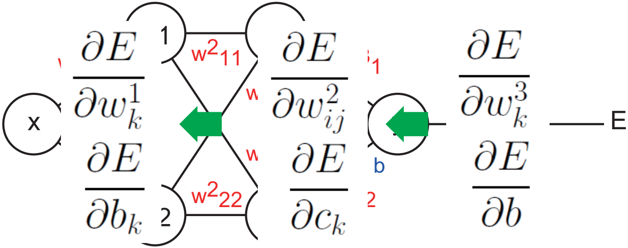
誤差逆伝搬法



$$\frac{\partial E}{\partial w_{ij}^2} \qquad \frac{\partial E}{\partial w_k^3}$$

$$\frac{\partial E}{\partial c_k} \qquad \frac{\partial E}{\partial b}$$

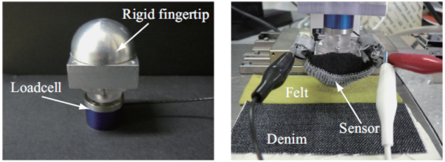
誤差逆伝搬法



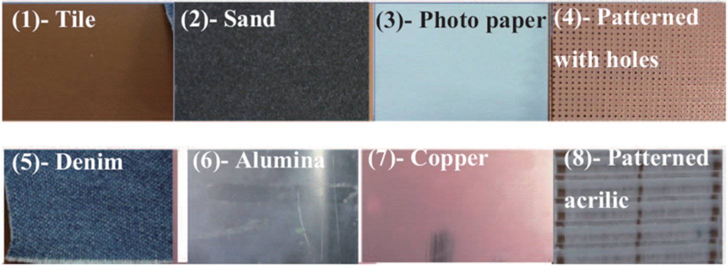
$$\frac{\partial E}{\partial w_k^1} \quad \frac{\partial E}{\partial w_{ij}^2} \quad \frac{\partial E}{\partial w_k^3}$$

$$\frac{\partial E}{\partial b_k} \quad \frac{\partial E}{\partial c_k} \quad \frac{\partial E}{\partial b}$$

テクスチャーの識別

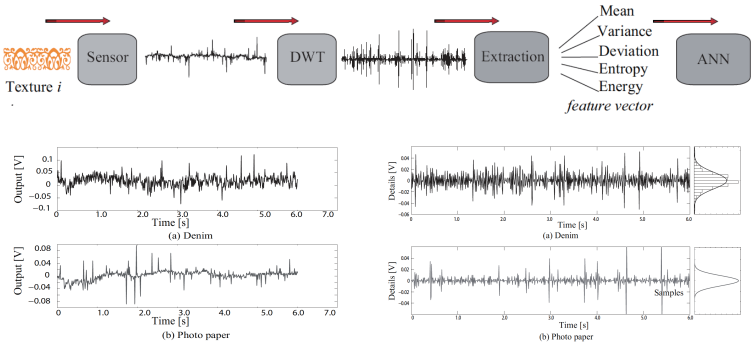


ロボット指と触覚センサ



8 種類のテクスチャー

テクスチャーの識別



センサ信号　　　ウェーブレット変換

上：デニム地　下：写真用紙

テクスチャーの識別

	1	2	3	4	5	6	7	8
1	90							
2		77	5					
3	10	11	85	8			6	
4				73				
5		12	5	19	100			
6						80	5	
7		9	5			20	89	
8								100

正解率　86.5 %

まとめ

ニューロンモデル

シグモイド関数　シグモイドニューロン

ニューラルネットワーク

入力層　隠れ層　出力層

近似定理

隠れ層が 1 層のニューラルネットワーク

学習

誤差逆伝搬法 (Back Propagation)

レポート

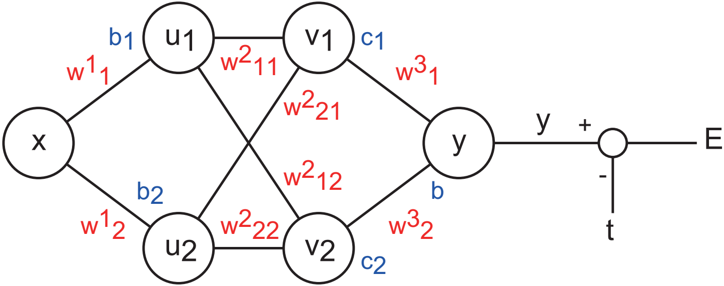
manaba+R に pdf ファイルで提出.
締切：12 月 11 日（月曜）午前 1 時

図に示すニューラルネットワークにおいて，逆誤差伝搬学習を行う．入力 x，出力 y，教師信号 t，誤差 E の関係は

$$
\begin{aligned}
u_1 &= \sigma(w_1^1 x + b_1), \quad u_2 = \sigma(w_2^1 x + b_2), \\
v_1 &= \sigma(w_{11}^2 u_1 + w_{21}^2 u_2 + c_1), \quad v_2 = \sigma(w_{12}^2 u_1 + w_{22}^2 u_2 + c_2), \\
y &= \sigma(w_1^3 v_1 + w_2^3 v_2 + b), \quad E = \frac{1}{2}(y - t)^2
\end{aligned}
$$

と表される．ここで $\sigma(x) = 1/(1 + e^{-x})$ である.
(1) 偏微分 $\partial E/\partial w_1^3$，$\partial E/\partial w_2^3$ を計算せよ.
(2) 偏微分 $\partial E/\partial w_{11}^2$ を計算し，$\partial E/\partial w_1^3$ を用いて表せ.
(3) 偏微分 $\partial E/\partial w_{12}^2$ を計算し，$\partial E/\partial w_2^3$ を用いて表せ.
(4) 偏微分 $\partial E/\partial w_1^1$ を計算し，$\partial E/\partial w_{11}^2$ と $\partial E/\partial w_{12}^2$ を用いて表せ.

レポート



x
u1
u2
v1
v2
y
b1
b2
c1
c2
b
w11
w12
w211
w221
w212
w222
w31
w32
y
+
-
E
t