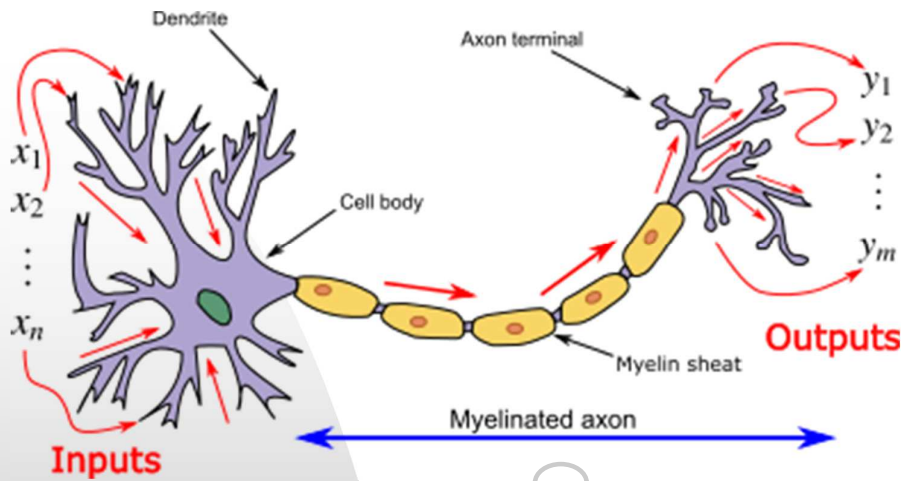


# 第13回 ニューラルネット

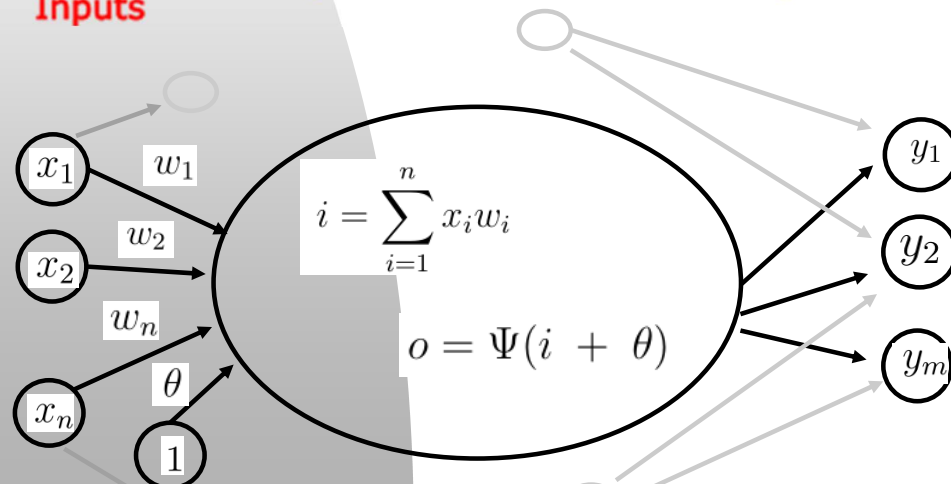
## ニューラルネットワーク



Dendrite 樹状突起  
Cell Body 細胞体  
Myelinated Axon 軸索  
Axon Terminal 軸索突起

一つのニューロンは、他の複数のニューロンからの信号を受け取り、それらを統合した形で多くの複数のニューロンに伝える。

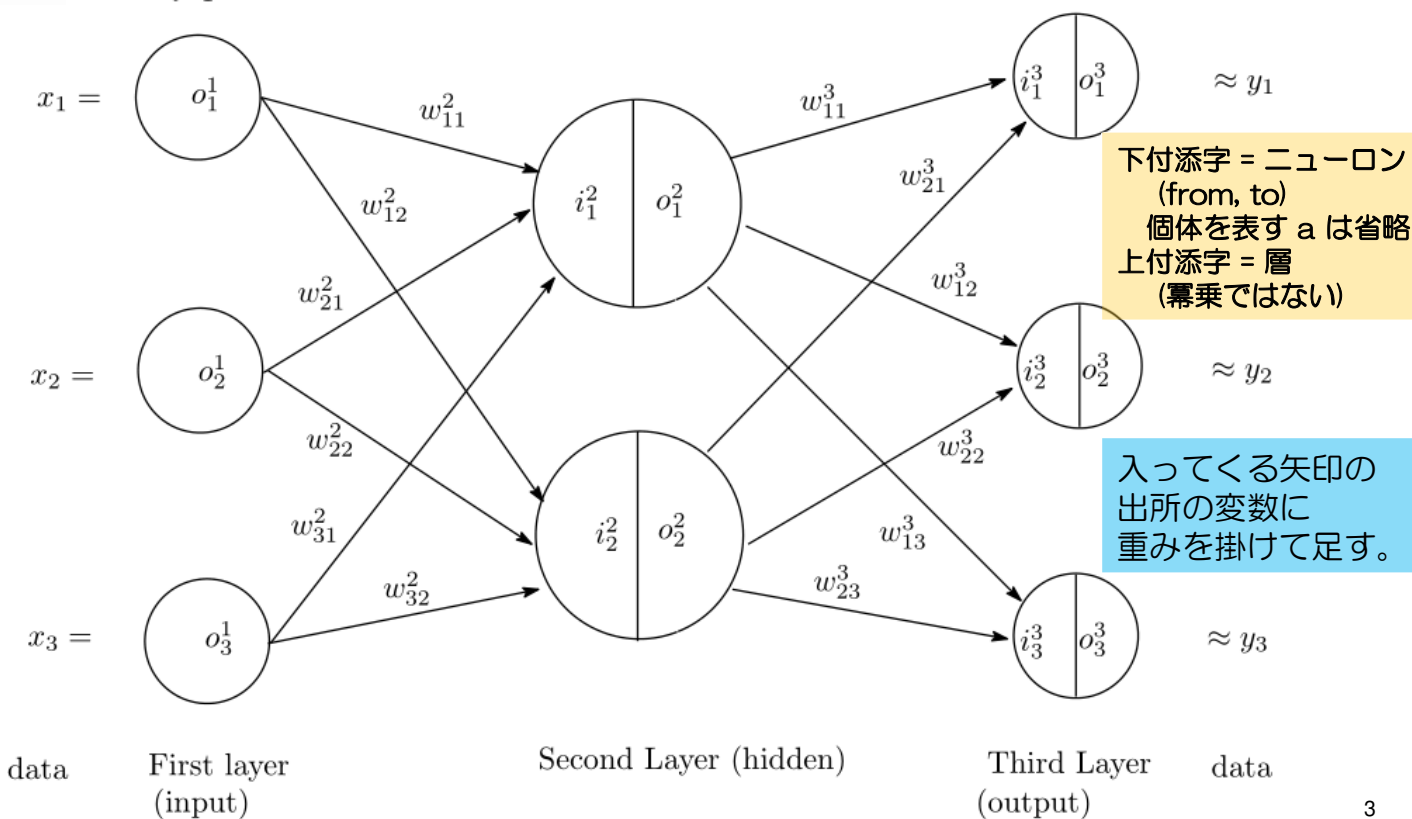
ニューロンは統合された入力の値がある閾値を超えたときに活性化し、信号を発する。



情報の統合 propagation  
活性化 activation  
活性化関数 activation function

# 3層ニューラルネットワーク

$$i_{aj}^2 = \sum_{i=1}^{n_1} o_{ai}^1 w_{ij}^2 + \theta_j^2, \quad o_{aj}^2 = \Psi(i_{aj}^2) = \frac{1}{1 + \exp(-i_{aj}^2)}, \quad i_{aj}^3 = \sum_{i=1}^{n_2} o_{ai}^2 w_{ij}^3 + \theta_j^3$$



# 3層ニューラルネットワーク

$$i_{aj}^2 = \sum_{i=1}^{n_1} o_{ai}^1 w_{ij}^2 + \theta_j^2 = \sum_{i=1}^{n_1} x_{ai} w_{ij}^2 + \theta_j^2,$$

$$o_{aj}^2 = \Psi(i_{aj}^2) = \frac{1}{1 + \exp(-i_{aj}^2)},$$

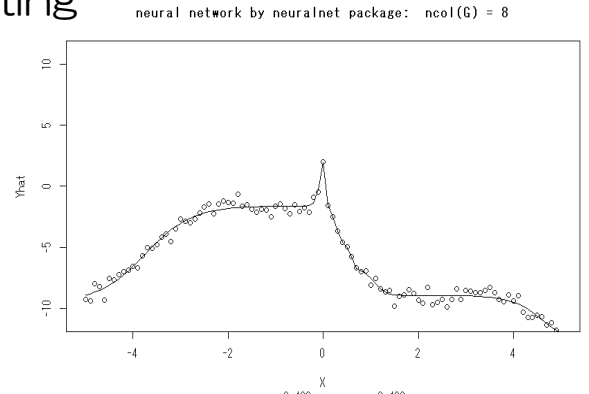
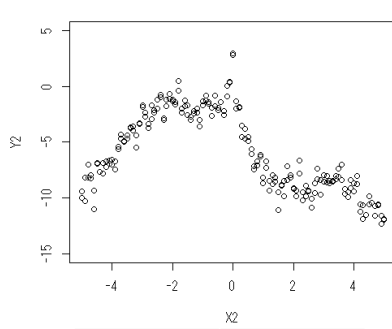
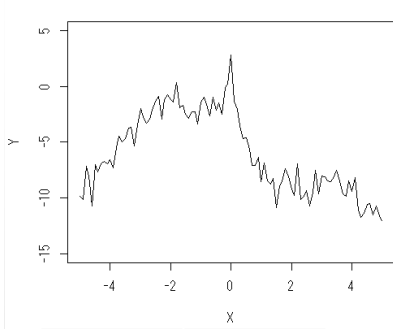
$$y_{aj} = o_{aj}^3 + e_{aj} \quad \text{or} \quad y_{aj} = \hat{y}_{aj} + e_{aj} \quad \text{where} \quad \hat{y}_{aj} = o_{aj}^3$$

$$\begin{aligned} i_{aj}^3 &= \sum_{i=1}^{n_2} o_{ai}^2 w_{ij}^3 + \theta_j^3 \\ &= \sum_{i=1}^{n_2} \Psi(i_{ai}^2) w_{ij}^3 + \theta_j^3 \\ &= \sum_{i=1}^{n_2} \frac{1}{1 + \exp(-i_{ai}^2)} w_{ij}^3 + \theta_j^3 \\ &= \sum_{i=1}^{n_2} \frac{1}{1 + \exp(-(\sum_{k=1}^{n_1} o_{ak}^1 w_{ki}^2 + \theta_i^2))} w_{ij}^3 + \theta_j^3 \\ &= \sum_{i=1}^{n_2} \frac{1}{1 + \exp(-(\sum_{k=1}^{n_1} x_{ak} w_{ki}^2 + \theta_i^2))} w_{ij}^3 + \theta_j^3 \end{aligned}$$

$$o_{aj}^3 = \begin{cases} = i_{aj}^3 & \text{最終層はそのまま} \\ = \Psi(i_{aj}^3) & \text{最終層もロジスティック変換} \end{cases}$$

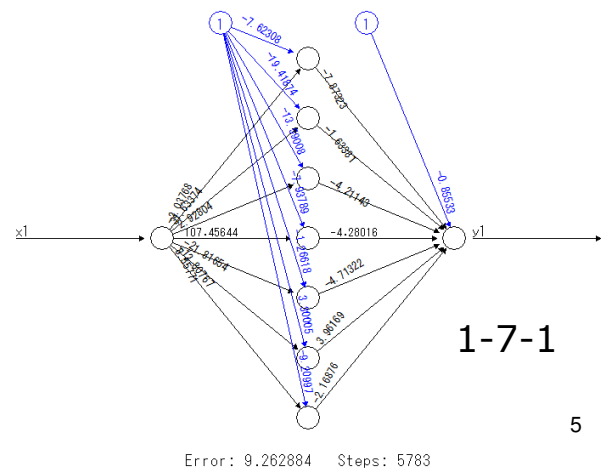
# ニューラルネットワークで出来ること

- 関数の近似やデータへの曲線の当てはめ  
function approximation or curve fitting



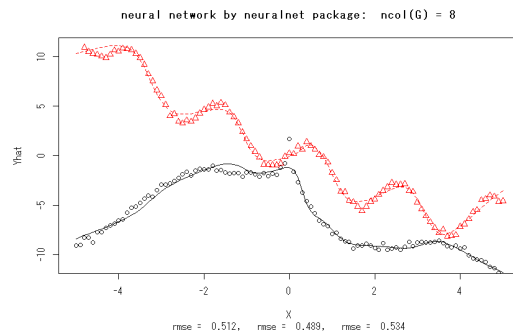
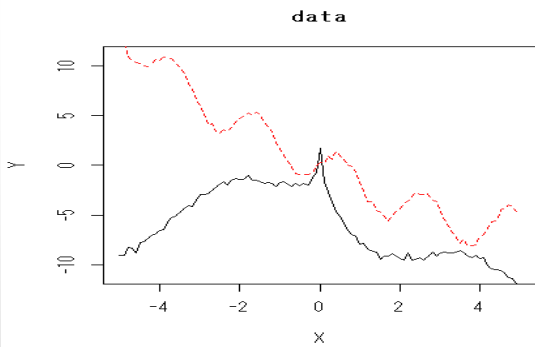
X	Y
-5.0	-10.0
-4.8	-6.5
-4.6	-11.3
-4.4	-5.0
-4.2	-4.2
-4.0	-6.0
-3.8	-4.1
-3.6	-2.0

X	Y
-5.0	-10.0
-5.0	-11.2
-4.8	-6.5
-4.8	-6.5
-4.8	-4.1
-4.6	-11.3
-4.6	-5.2
-4.4	-5.0

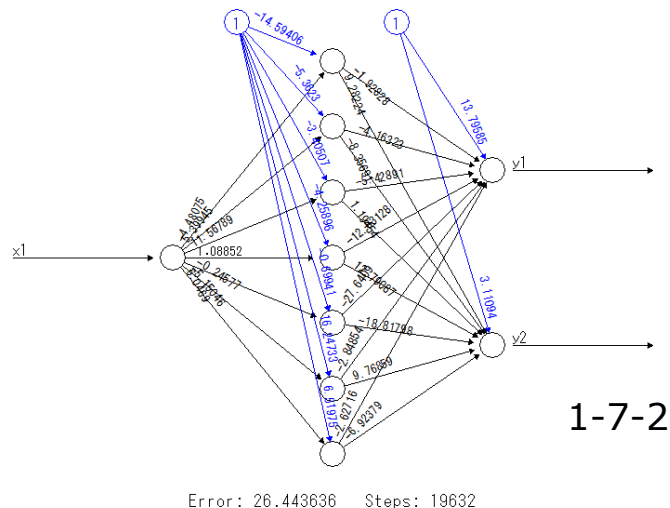


5

## 複数の出力



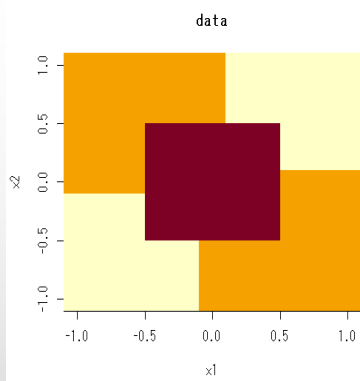
X	Y1	Y2
-5.0	-10.0	11.1
-4.8	-6.5	10.0
-4.6	-11.3	12.3
-4.4	-5.0	11.3
-4.2	-4.2	8.1
-4.0	-6.0	9.2
-3.8	-4.1	5.0
-3.6	-2.0	-2.1



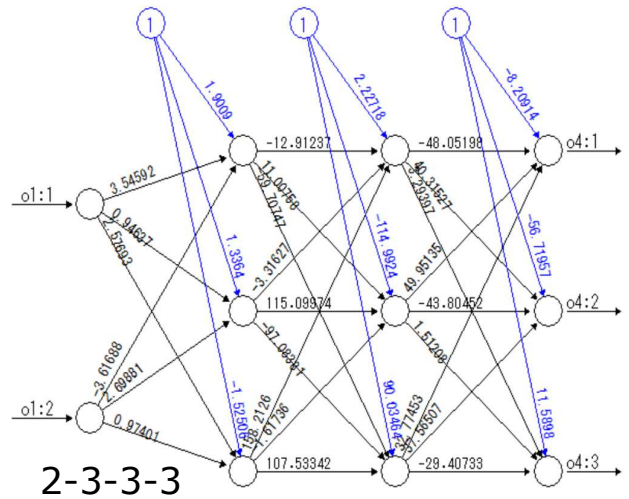
6

# ニューラルネットワークで出来ること

## ■ 判別分析、パターン認識



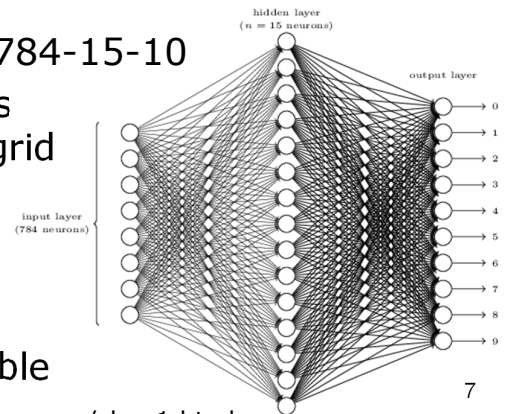
X1	X2	Y	Y1	Y2	Y3
-1.0	1.0	橙	1	0	0
-0.8	1.0	橙	1	0	0
-0.6	1.0	橙	1	0	0
...	...	...	...	...	...
0.0	1.0	黄	0	1	0
0.2	1.0	黄	0	1	0
...	...	...	...	...	...
-1.0	-0.2	黄	0	1	0
-0.8	-0.2	黄	0	1	0
-0.4	-0.2	赤	0	0	1
...	...	...	...	...	...



Data: 100 hand writings  
Greyscale at 28 x 28 grid

X  
100 x 784 data matrix  
Y  
100 x 10 dummy variable

784-15-10



<http://neuralnetworksanddeeplearning.com/chap1.html>

## 用語その他

### ■ 層 (layer) の数え方

- ◆ 入力層 = 第 1 層 or 最初の隠れ層 (hidden layer) = 第 1 層
- ◆ 入力 → 隠れ層 → 出力のネットワークは 3 層か 2 層か。
- ◆ 隠れ層の数 shallow か deep か。
- ◆ 各隠れ層におけるニューロンの数

### ■ 活性化関数の種類

- ◆ sigmoid (logistic etc) ReLU (Rectified Linear Unit, spline)
- ◆ 最終層の活性化の有無 もしくは softmax

### ■ 順伝搬 (feed forward) か否か (recurrent)

### ■ 教師信号の有無 (supervised or unsupervised)

- ◆ 回帰分析・判別分析か、クラスター分析か。

# 教師信号ありの順伝搬型ニューラルネットワーク

- supervised feed-forward neural network

- ◆ 入力層を X とし最終層を Y とすると（両方とも複数可）

- ★ Y が連続量の場合は（非線形の）回帰分析

- ★ Y がカテゴリーカル（名義尺度）の場合は多群への（非線形の）判別分析

- ★ いずれにしても、

$$Y = \hat{Y} + E$$

$$\hat{Y} = \Gamma(X|\xi)$$

の形で書ける。ただし、 $\Gamma(X|\xi)$  は非線形の関数。

- ★ したがって、最小2乗法や最尤法でパラメタを推定することが可能。

## 一般形 多層パーセプトロン

$$i_{aj}^k = \sum_{i=1}^{n_{k-1}} o_{ai}^{k-1} w_{ij}^k + \theta_j^k$$

$$o_{aj}^k = \Psi(i_{aj}^k) = \frac{1}{1 + \exp(-i_{aj}^k)}$$

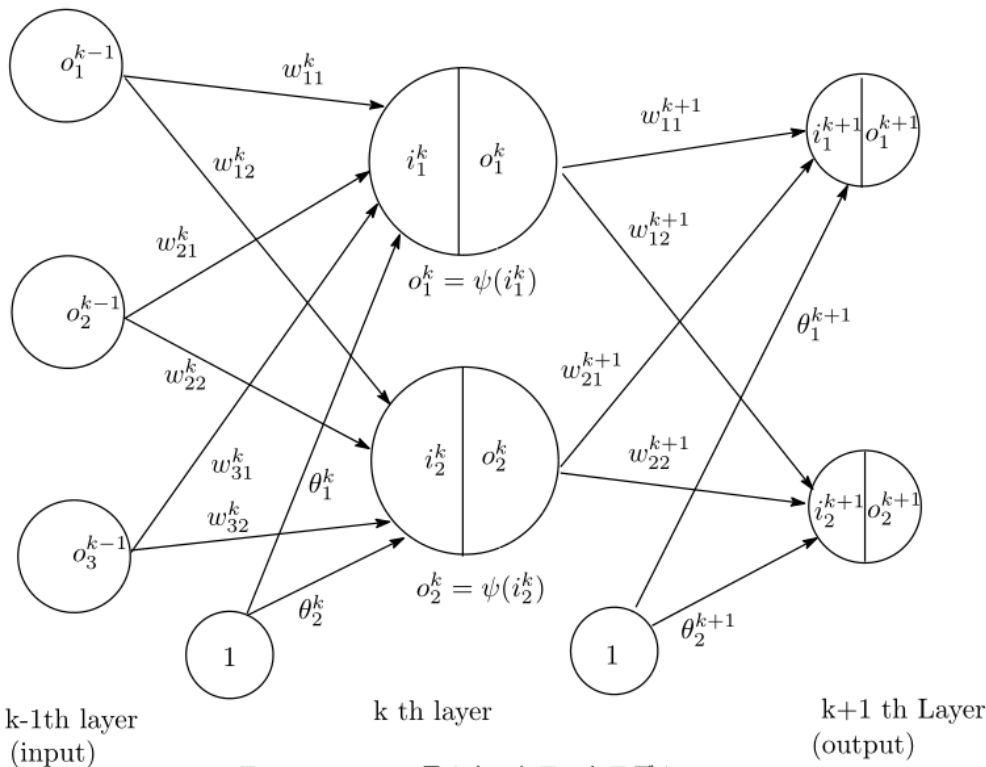


Figure1 ニューラルネットワークモデル

# 行列表記

$$\mathbf{O}^k : N \times n_k$$

$$\mathbf{W}^k : n_{k-1} \times n_k$$

$$\boldsymbol{\theta}_j : 1 \times n_k$$

$$\begin{aligned} \mathbf{O}^k &= \Psi(\mathbf{I}^k) \\ &= \Psi(\mathbf{O}^{k-1}\mathbf{W}^k + \mathbf{1}\boldsymbol{\theta}^k) \\ &= \Psi(\Psi(\mathbf{I}^{k-1})\mathbf{W}^k + \mathbf{1}\boldsymbol{\theta}^k) \\ &= \Psi(\Psi(\mathbf{O}^{k-2}\mathbf{W}^{k-1} + \mathbf{1}\boldsymbol{\theta}^{k-1})\mathbf{W}^k + \mathbf{1}\boldsymbol{\theta}^k) \\ &\quad \vdots \quad \vdots \end{aligned}$$

$$\begin{aligned} \mathbf{O}^n &= \Gamma_n(\mathbf{O}^{n-1}) \\ &= \Gamma_n(\Gamma_{n-1}(\mathbf{O}^{n-2})) \\ &\quad \vdots \quad \vdots \\ &= \Gamma_n \circ \Gamma_{n-1} \circ \cdots \circ \Gamma_2(\mathbf{O}^1) \\ &= \Gamma(\mathbf{O}^1) = \Gamma(\mathbf{X}|\boldsymbol{\xi}) \end{aligned}$$

$$\Gamma(\cdot) = \Gamma_n \circ \Gamma_{n-1} \circ \cdots \circ \Gamma_2(\cdot)$$

$$\begin{aligned} o_{aj}^k &= \Psi(i_{aj}^k) = \frac{1}{1 + \exp(-i_{aj}^k)} \\ i_{aj}^k &= \sum_{i=1}^{n_{k-1}} o_{ai}^{k-1} w_{ij}^k + \theta_j^k \end{aligned}$$

$$\mathbf{O}^k = \Psi(\mathbf{O}^{k-1}\mathbf{W}^k + \mathbf{1}\boldsymbol{\theta}^k) = \Gamma_k(\mathbf{O}^{k-1})$$

11

# 最小2乗解

$$\begin{aligned} \text{RSS} &= \sum_{s=1}^{n_n} \sum_{i=1}^N (y_{is} - \hat{y}_{is})^2 \\ &= \sum_{s=1}^{n_n} \sum_{i=1}^N (y_{is} - \gamma_{is}(\mathbf{x}_i|\boldsymbol{\xi}))^2 \\ &= \text{tr} \left[ (\mathbf{Y} - \boldsymbol{\Gamma}(\mathbf{X}|\boldsymbol{\xi}))' (\mathbf{Y} - \boldsymbol{\Gamma}(\mathbf{X}|\boldsymbol{\xi})) \right] \end{aligned}$$

$N$  は個体数、 $n$  は層の数、

$n_n$  は第  $n$  層、すなわち最終層のニューロンの数

12

# R のコード 2-2-2-1 の例

データ	重みのリスト	各層の出力のリスト
x1 x2 y1	W	0
1 0 0 0	[[1]]NULL	[[1]] x1 x2
2 0 1 1	[[2]] i2:1 i2:2	1 0 0
3 1 0 1	o1:0 4.565391 4.206300	2 0 1
4 1 1 0	o1:1 -9.225230 10.159812	3 1 0
	o1:2 12.018397 -8.564449	4 1 1
	[[3]] i3:1 i3:2	[[2]] o2:0 o2:1 o2:2
各層のニューロンの数	o2:0 1.7740988 -4.752723	1 (1) 0.98970135 0.98531739
1 2	o2:1 -7.5795366 1.726017	2 (1) 0.99999994 0.01264024
2 2	o2:2 0.7844619 6.085658	3 (1) 0.00937918 0.99999942
3 2		4 (1) 0.99936329 0.99698658
4 1	[[4]] o4:1	[[3]] o3:0 o3:1 o3:2
	o3:0 3.801048	1 (1) 0.007002681 0.95034954
	o3:1 7.987947	2 (1) 0.003031923 0.04974679
	o3:2 -8.245371	3 (1) 0.923258766 0.79398565
		4 (1) 0.006570887 0.95432764
		[[4]] o4:1
		1 0.01836091
		2 0.96817271
		3 0.99033262
		4 0.01771883

最初のカラムに  $\theta$  を加えると

$W^k$  のサイズは  $(n_{k-1} + 1) \times n_k$

$$O^k = \Psi \left( \left[ \mathbf{1}, O^{k-1} \right] W^k \right)$$

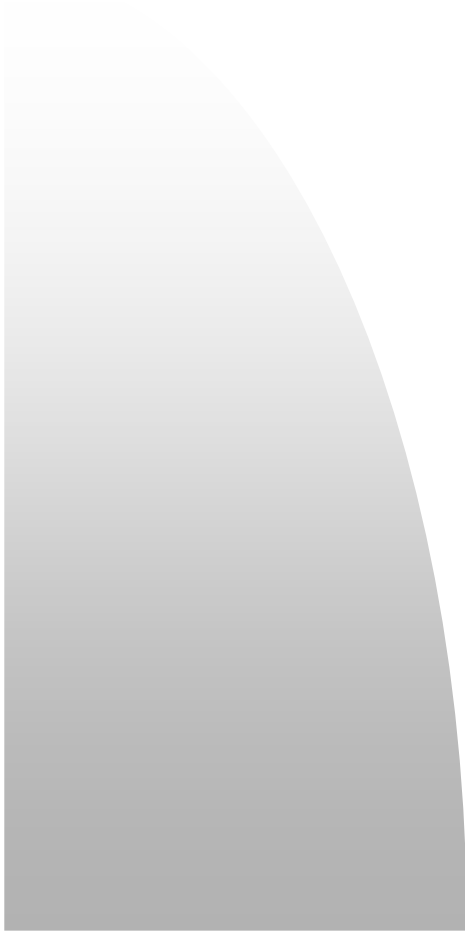
## R コード

$O^k$  の計算

```
Output <- function(W=W, O=O, k=0, linear=linear ){
  for( kk in 2:nlayer ){
    O[[kk]]=cbind(1,O[[kk-1]])%*%W[[kk]]
    if( kk < nlayer || linear == 0 ) O[[kk]]=1/(1+exp(-O[[kk]]))
  }
  return( O )
}
```

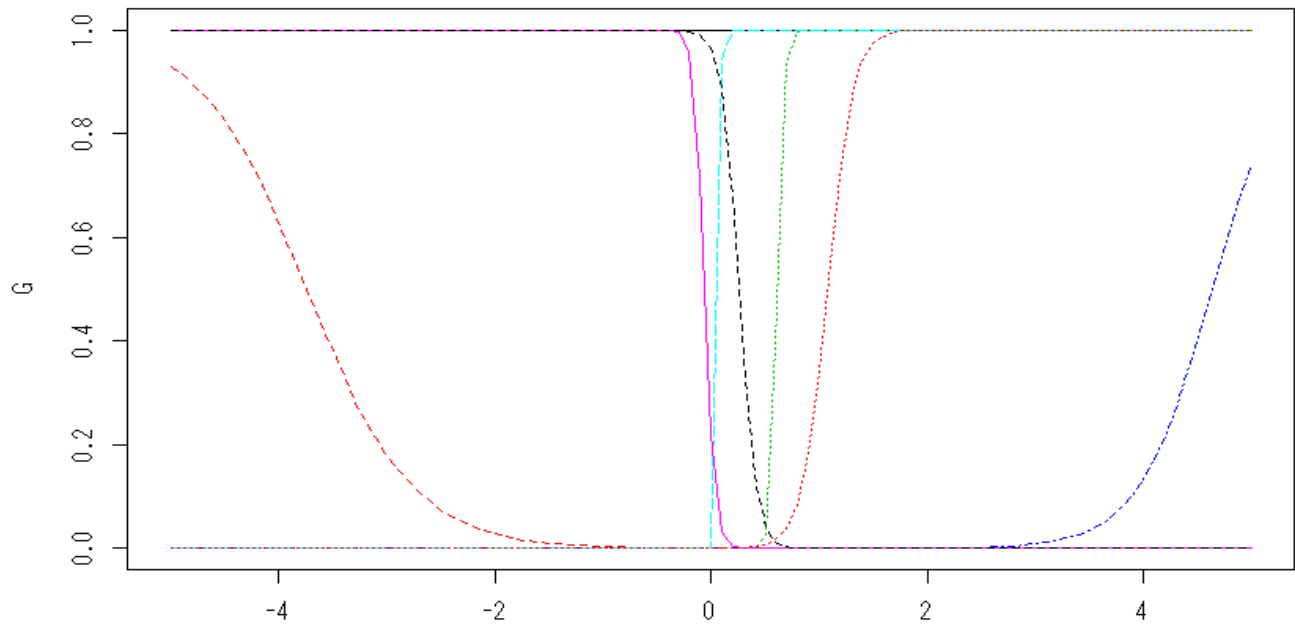
GN のための残差の計算

```
ymyhat <- function(vecw, W=W, O=O, k=0, linear=1){
  W <- vecw から W リストを復元
  c( Y - output( W=W, O=O, k=0, linear=linear )[[nlayer]] )
} # end of ymyhat
```



# 中間層の出力

neural network by neuralnet package: ncol(G) = 8

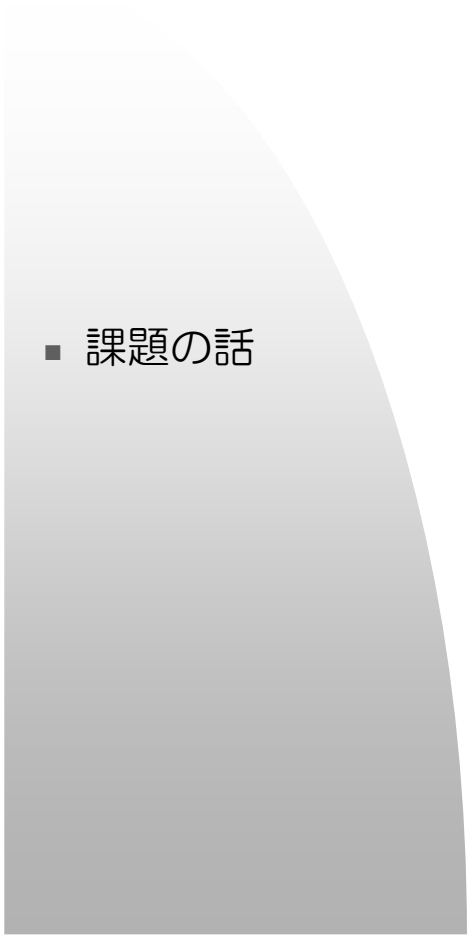


$$i_{aj}^k = \sum_{i=1}^{n_{k-1}} o_{ai}^{k-1} w_{ij}^k + \theta_j^k$$


$$o_{aj}^k = \Psi(i_{aj}^k) = \frac{1}{1 + \exp(-i_{aj}^k)}$$







- 課題の話



- お元気で！