

# データの分析と知識発見

Introduction to Data Analysis

## 今回の構成

0101010101010101010101010101010101

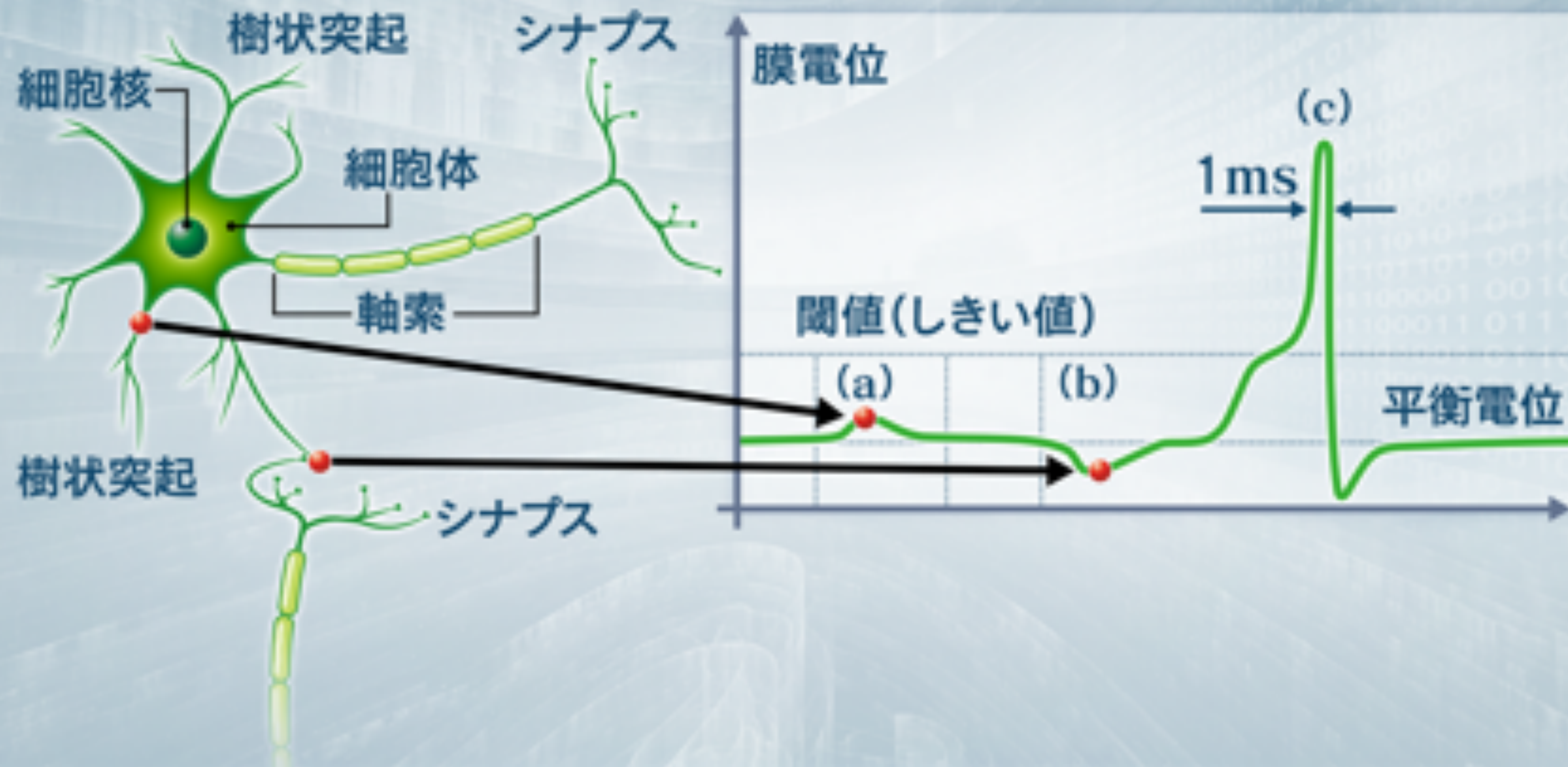
神経細胞の振る舞いと  
そのモデルについて理解する

バックプロパゲーションによる  
予測について理解する

パッケージnnetを用いて  
予測を行うことができる

# ニューロンの振る舞い

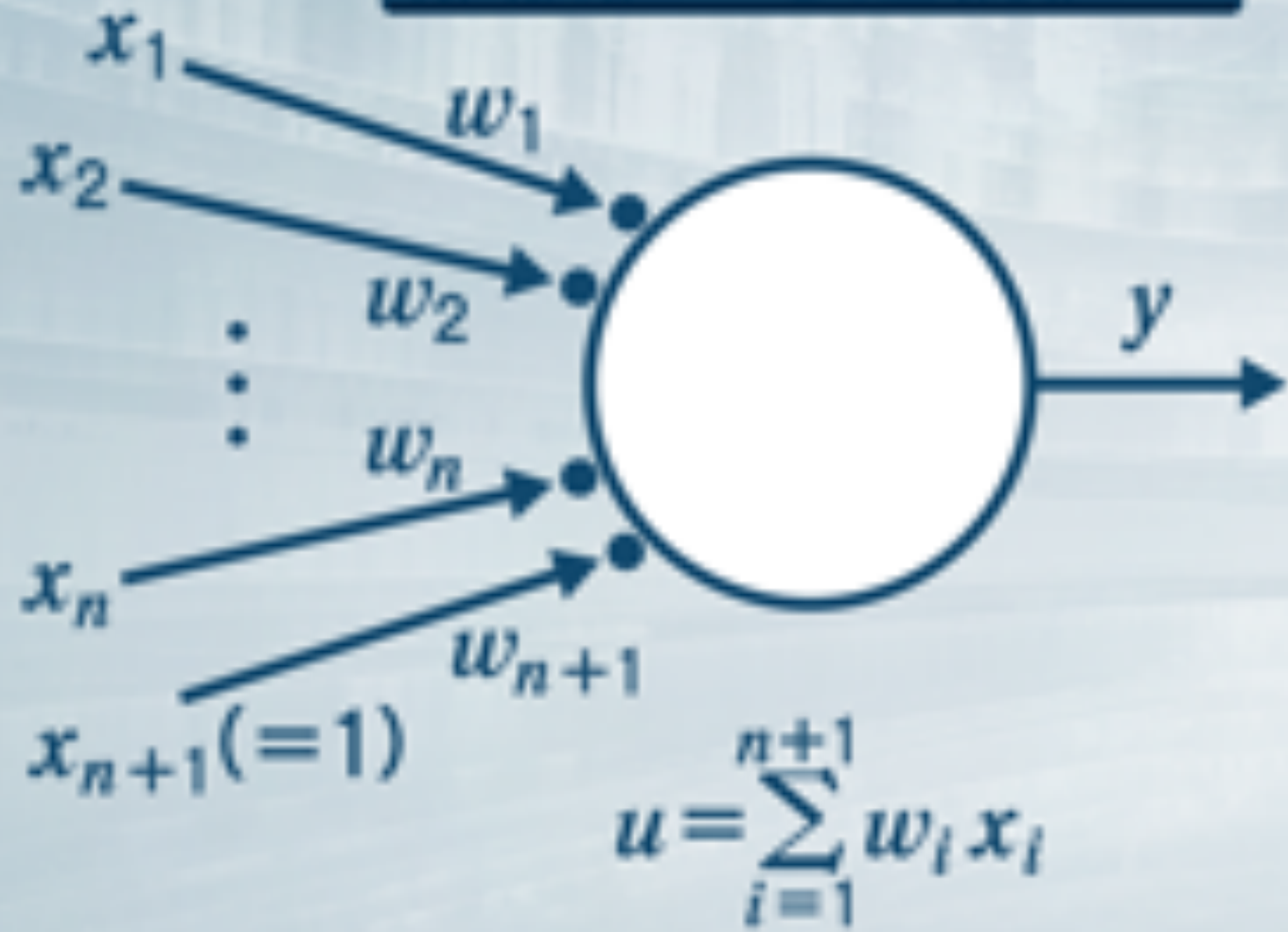
01



# 神経細胞のモデル

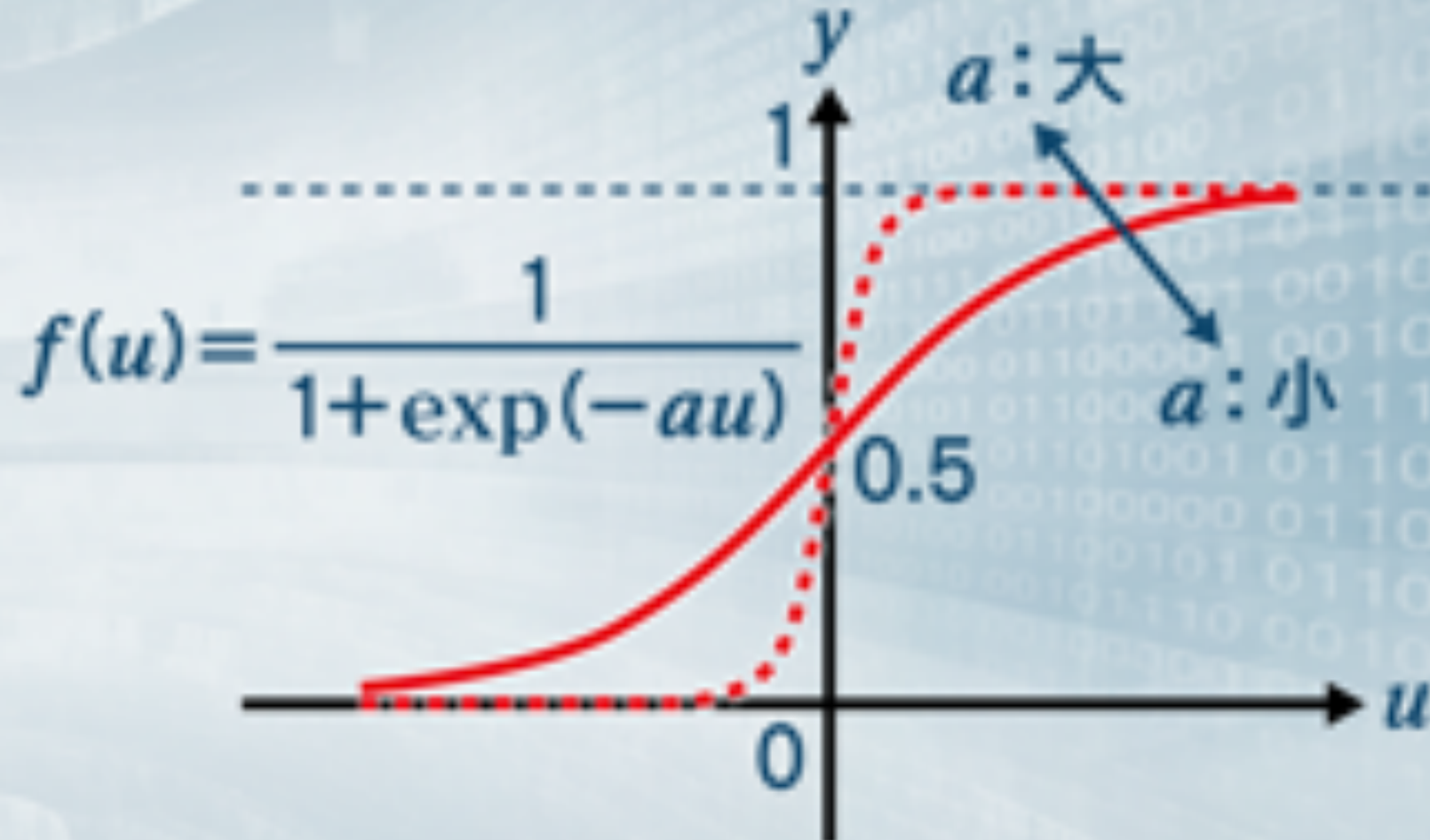
01

離散値から連続値へ



$$y = f(u)$$

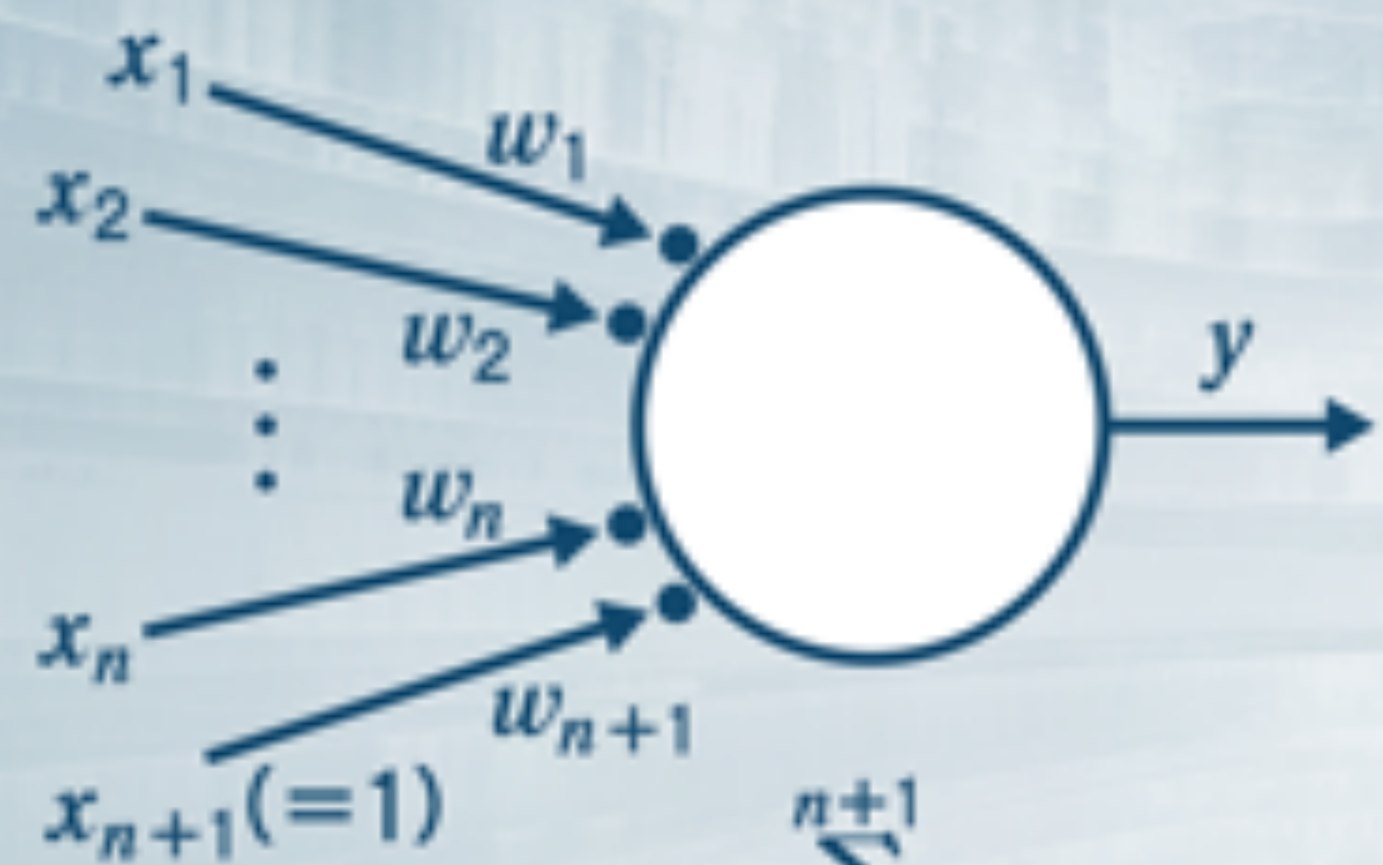
シグモイド関数



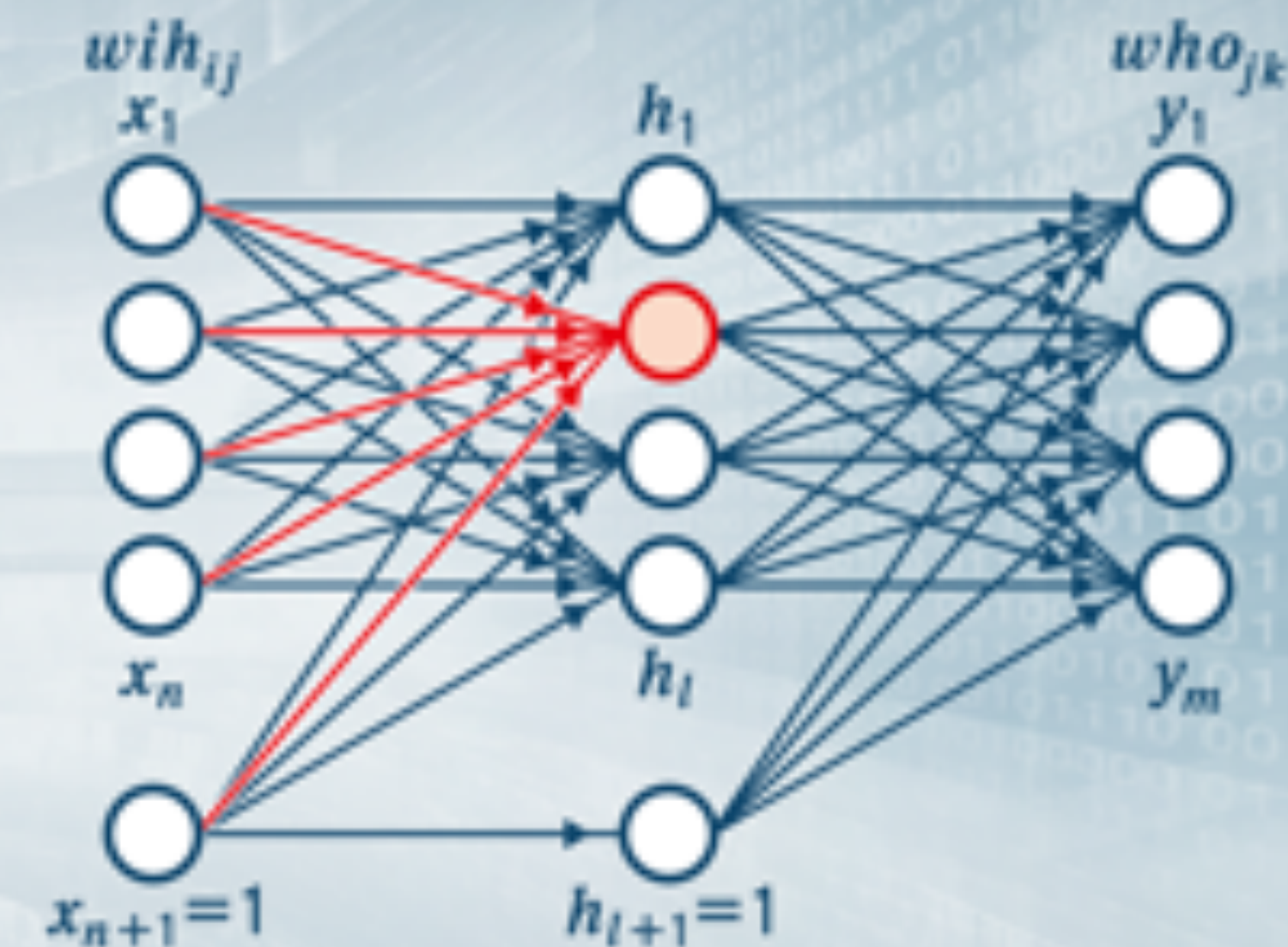
# 階層型ネットワーク

01

入力から出力へ方向のみ



$$u = \sum_{i=1}^{n+1} w_i x_i$$
$$y = f(u)$$



多入力多出力の関数



## ルールの抽出

01

電話帳からいくつかの電話番号と名前を覚える

→ 学習していない人の電話番号はわからない

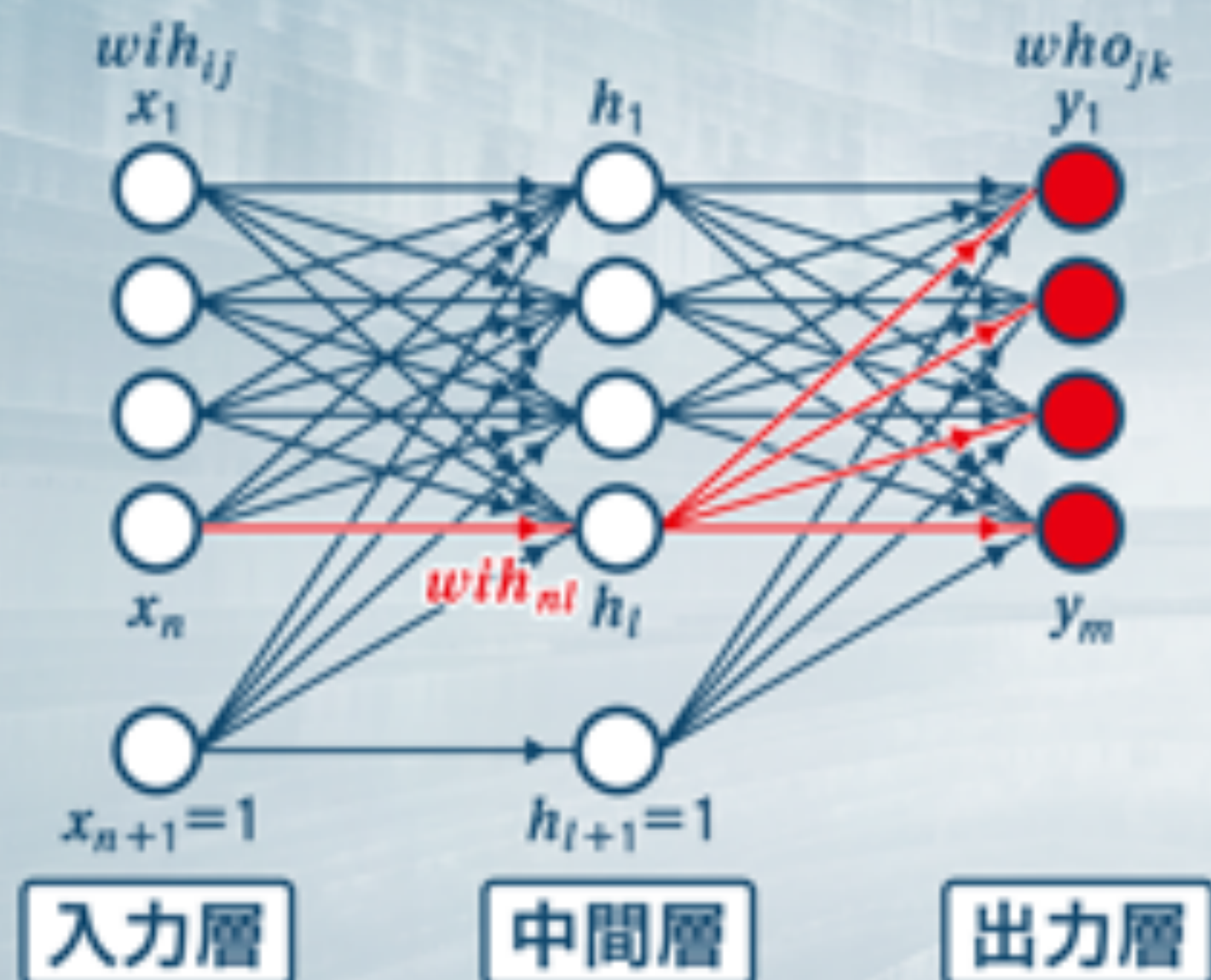
気温や湿度から海の家の上を予測する

過去数日の天気から次の日の天気を予測する

→ 何らかのルールに従って変動していると  
考えることができるものに対して用いる

# バックプロパゲーション

01



## 誤差の計算

$$E_p = \sum_{k=1}^m (y_k^{(p)} - \hat{y}_k^{(p)})^2$$

$$E = \sum_{p=1}^N E_p$$

## 個々の結合荷重が誤差に与える影響

$$\Delta w = -\varepsilon \frac{\partial E}{\partial w}$$



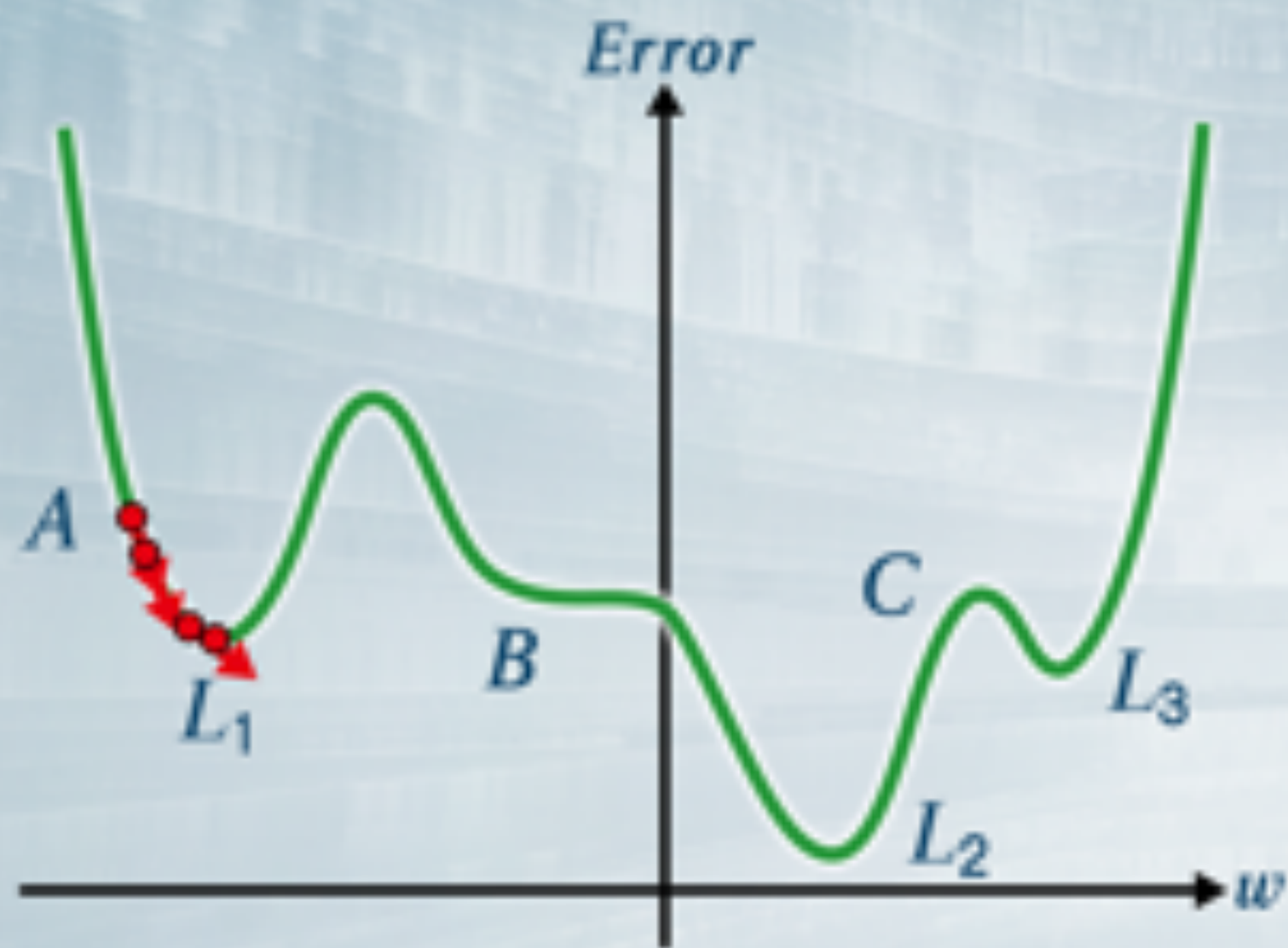
# 最急降下法

01010101010101010101010101010101



# 最急降下法

010101010101010101010101010101



局所解(ローカルミニマム)に収束



初期値によって結果が異なる

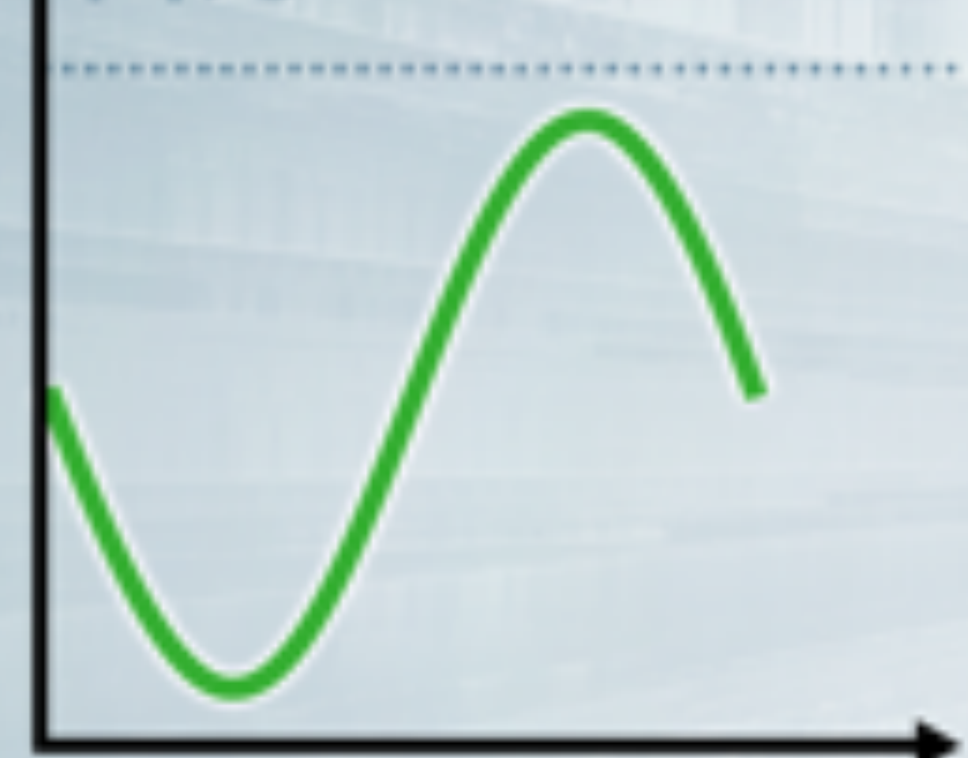


## 汎化と過学習(1)

01

a

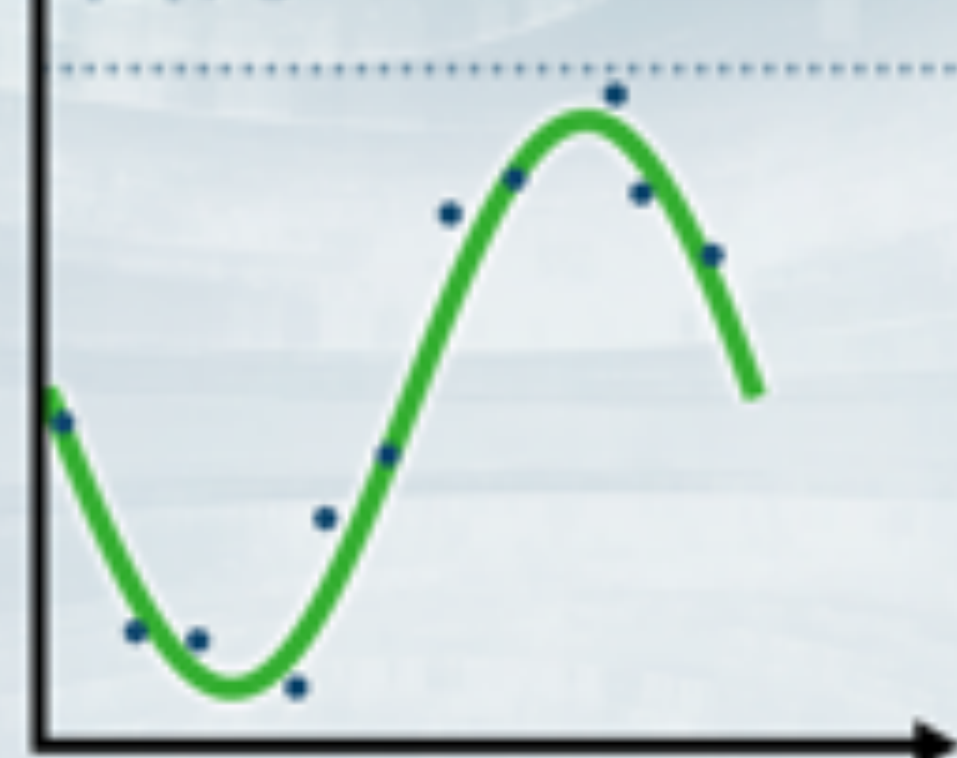
出力



入力

b

出力



入力

c

出力

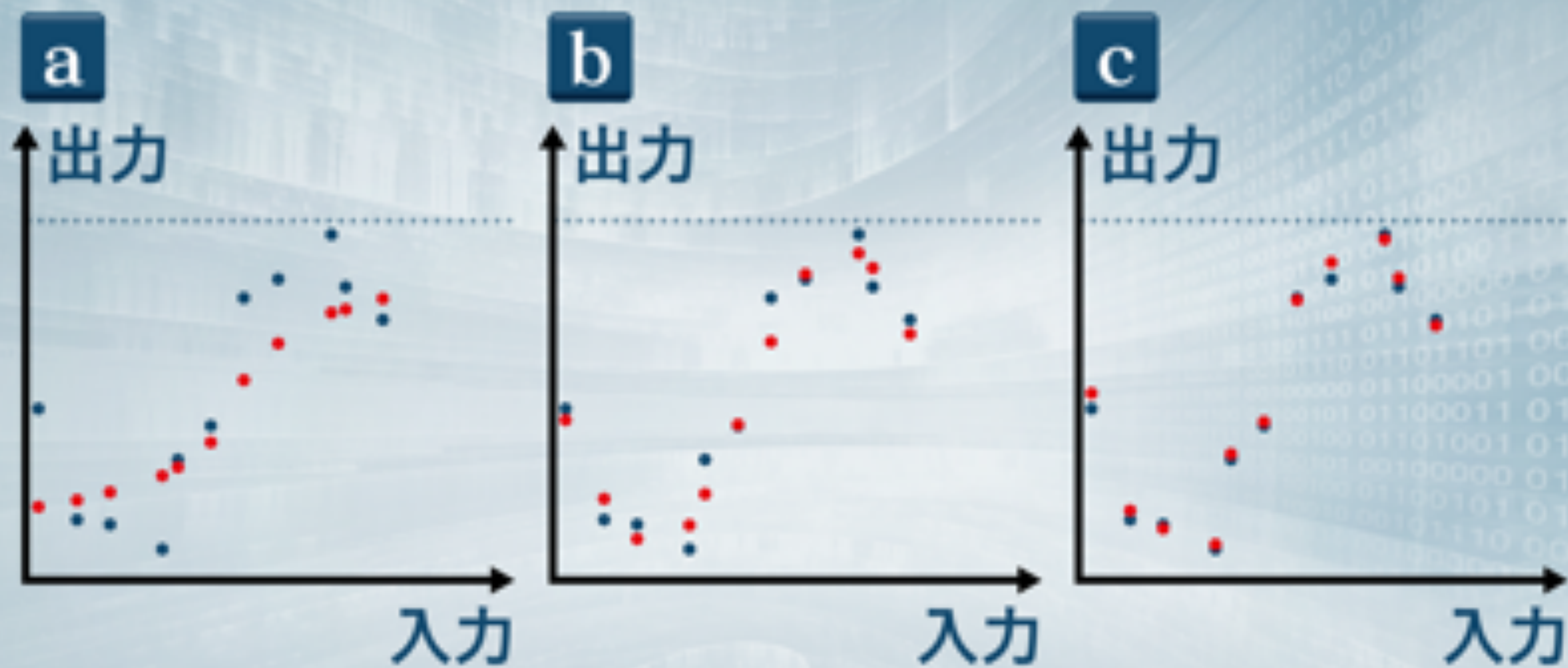


入力

教師信号が誤差を含むことがある

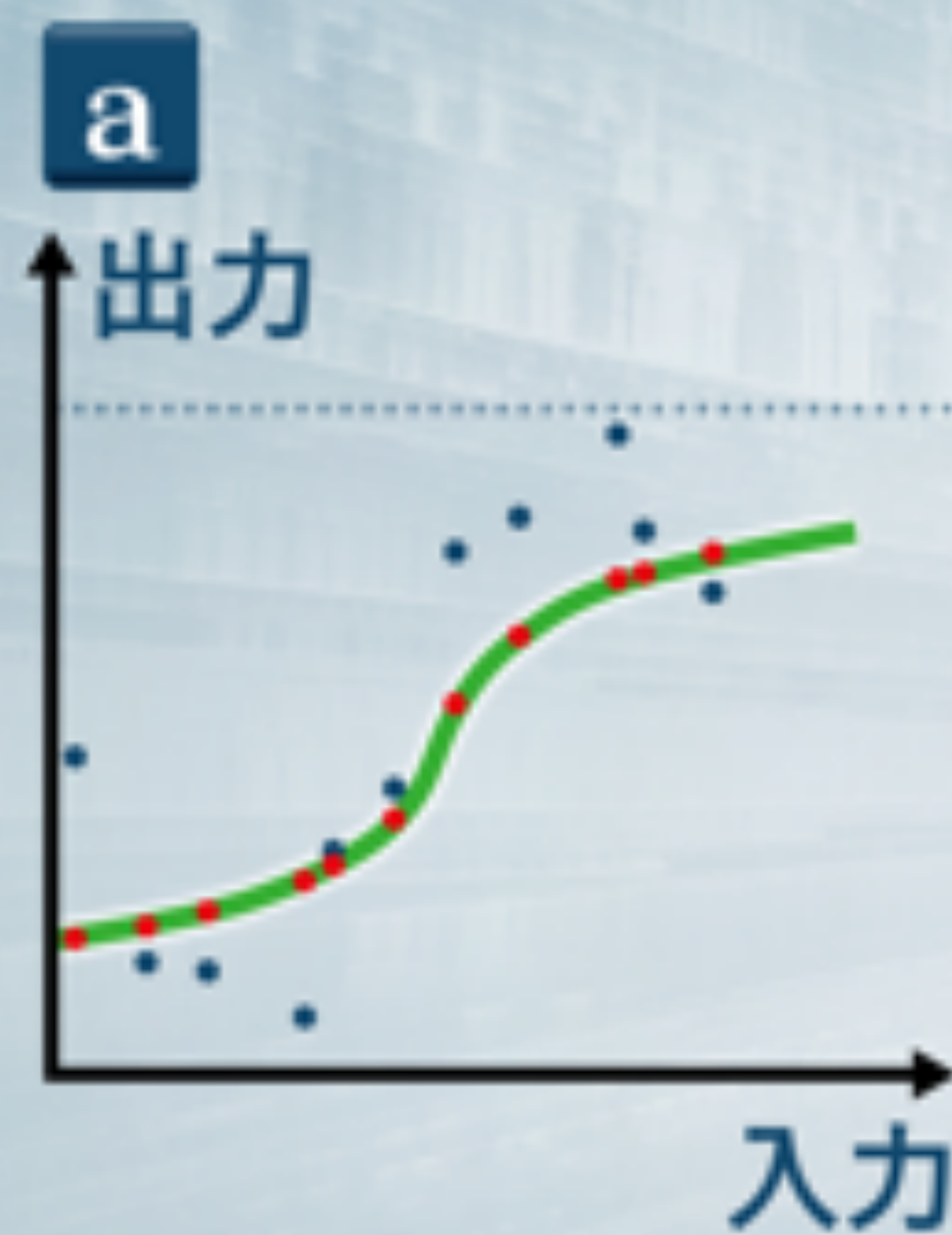
## 汎化と過学習(2)

01

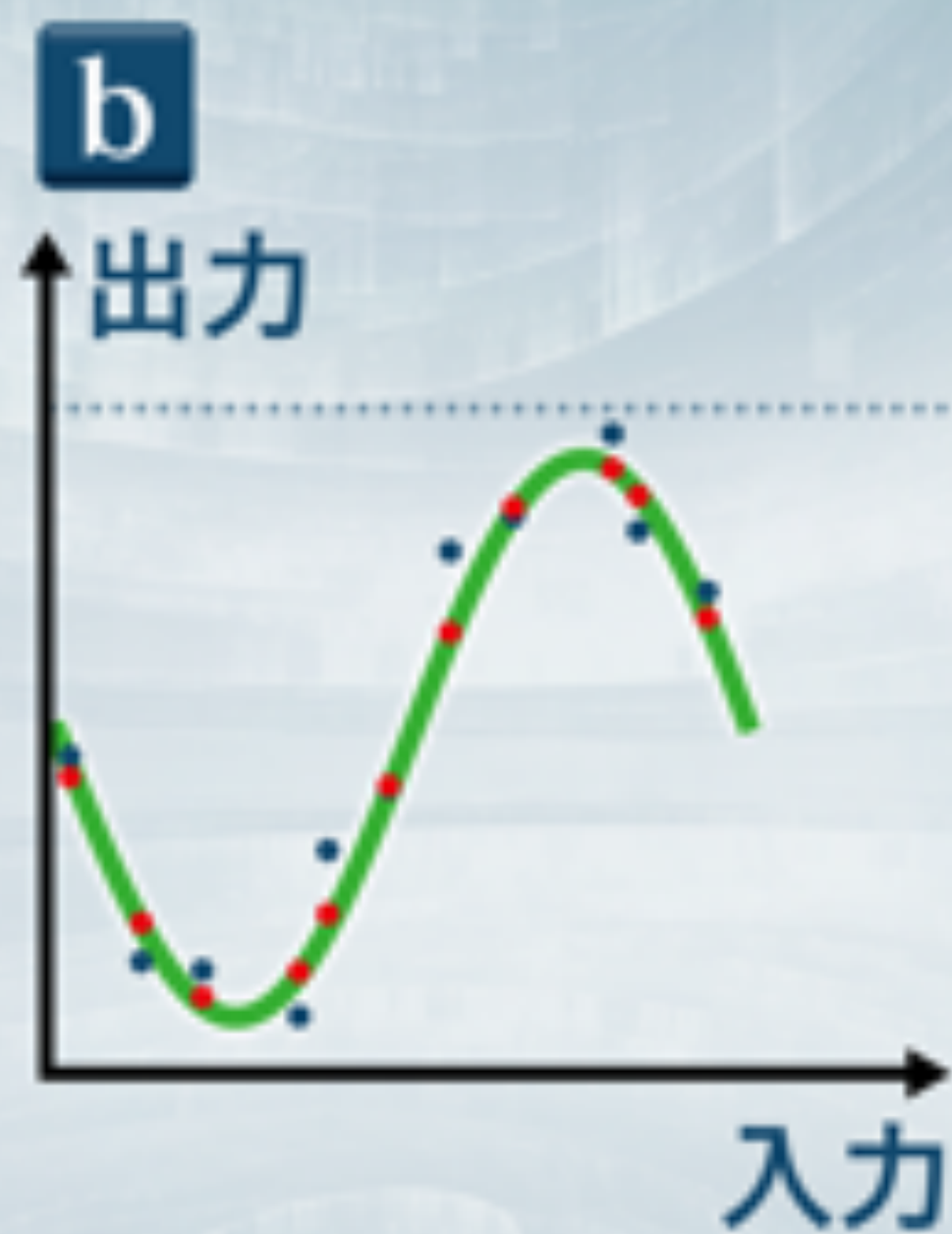


## 汎化と過学習(3)

01



学習が不十分



過学習

# データについて

01

## 訓練用データ

input,	output
0.015,	0.500912724
0.105,	0.304927043
0.195,	0.223999883
0.285,	0.185455886
0.375,	0.301246846
0.465,	0.40991515
0.555,	0.621385785
0.645,	0.732367781
0.735,	0.854840194
0.825,	0.753472695
0.915,	0.657707115

## 検証用

input,	output
0.025,	0.45306966
0.05,	0.407294902
0.075,	0.36380285
0.1,	0.323664424
0.125,	0.287867966
0.15,	0.257294902
0.175,	0.232698043
0.2,	0.214683045
0.225,	0.203693498
0.25,	0.2
	⋮



R

## Rによる演習(1)

01

```
> train <-read.csv("ch141.csv", header=T)
```

```
> valid <-read.csv("ch142.csv", header=T)
```

```
> sin1 <-nnet (output~input,data=train,size=1,maxit=1000)
```

```
> yosoku1 <-predict (sin1, valid)
```





## Rによる演習(2)

01

```
> plot(valid, type="b", pch=16, ylim=c(0,1))  
> points(train, pch=16, col="2")
```

```
o3 <- subset(o1,o1$trial=="New")
```

```
> points(valid$input, yosoku1, pch=16, col="4")  
> lines(valid$input, yosoku1)
```

