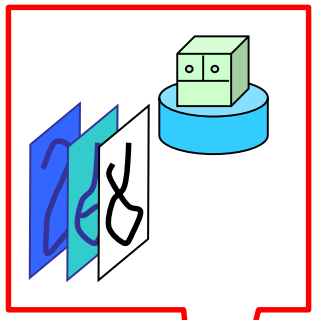
情報学習理論

渡辺澄夫
東京工業大学

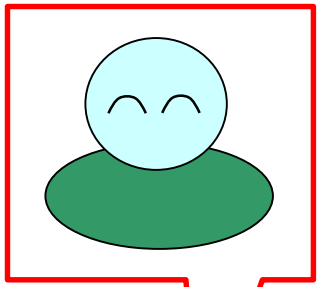
教師あり学習の枠組み



学習データ

$X_1, X_2, \ldots, X_n$

$Y_1, Y_2, \ldots, Y_n$



q(x,y)

y=f(x,w)

テストデータ

X

Y

情報源

学習モデル

復習：最急降下法

学習誤差関数　$E(w) = (1/n) \sum_{i=1}^{n} (Y_i - f(X_i, w))^2$

最急降下法：　$t=0,1,2,3,\ldots$,　$\eta>0$ は定数

$$w(t+1) - w(t) = -\eta \nabla E(w(t))$$

$\nabla E(w)$を計算するには、毎回 i=1,2…,n の足し算をしなくてはならない。サンプル数 n が大きいと大変。

∇を計算すると　$\nabla E(w) = (1/n) \sum_{i=1}^{n} \nabla \{ (Y_i - f(X_i, w))^2 \}$

確率降下法

各時刻 t において i 番目のデータをランダムに選んで

$$w(t+1) - w(t) = -\eta(t)\nabla\{ (Y_i - f(X_i, w(t)))^2 \}$$

とする。

◎1個ずつデータを取ってくるだけで学習できる。

◎ η(t) を適切に設計する必要がある。

◎　最急「バッチ学習」 ⇔　確率「オンライン学習」

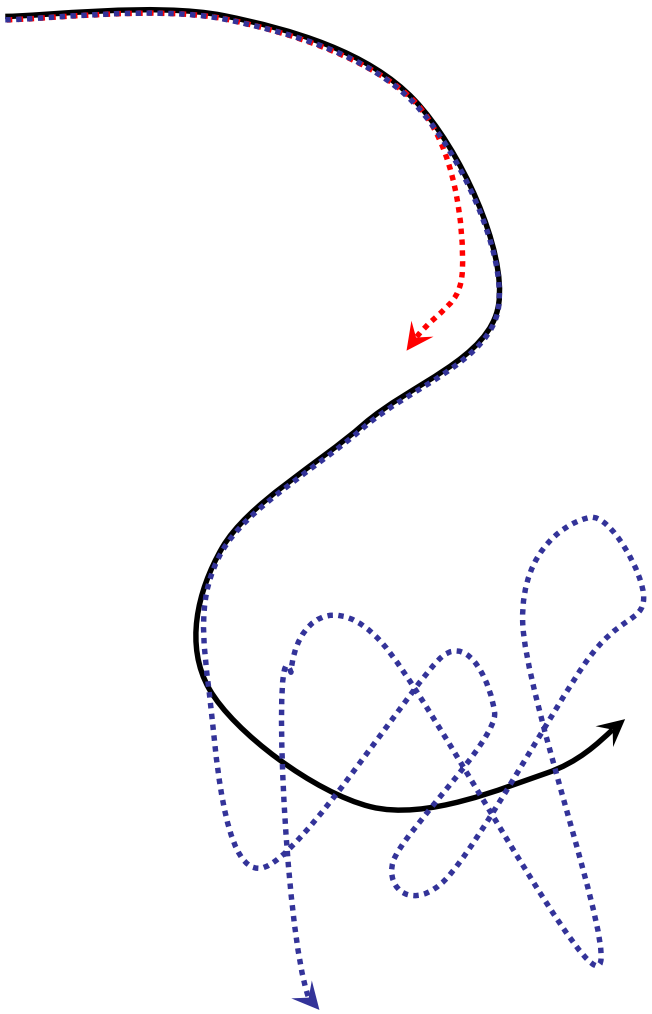
最急降下法と確率降下法

定理： $\eta(t)$ $(t=0,1,2,3,\ldots)$ が

$$\eta(t) \to 0$$

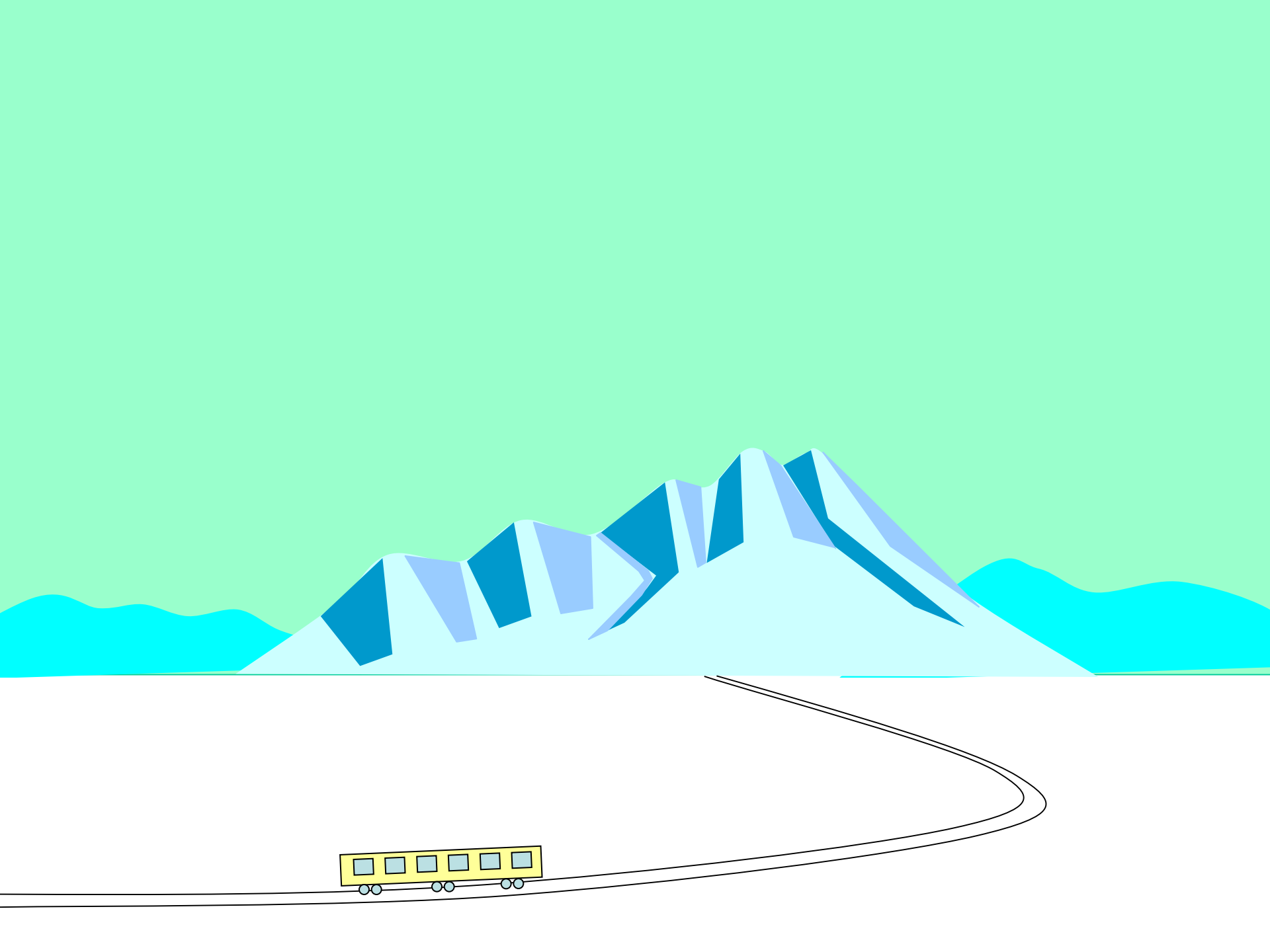
$$\eta(0)+\eta(1)+\cdots+\eta(t) \to \infty$$

を満たすとする。このとき、確率降下法は最急降下法と（局所的には）同じ点に収束する。



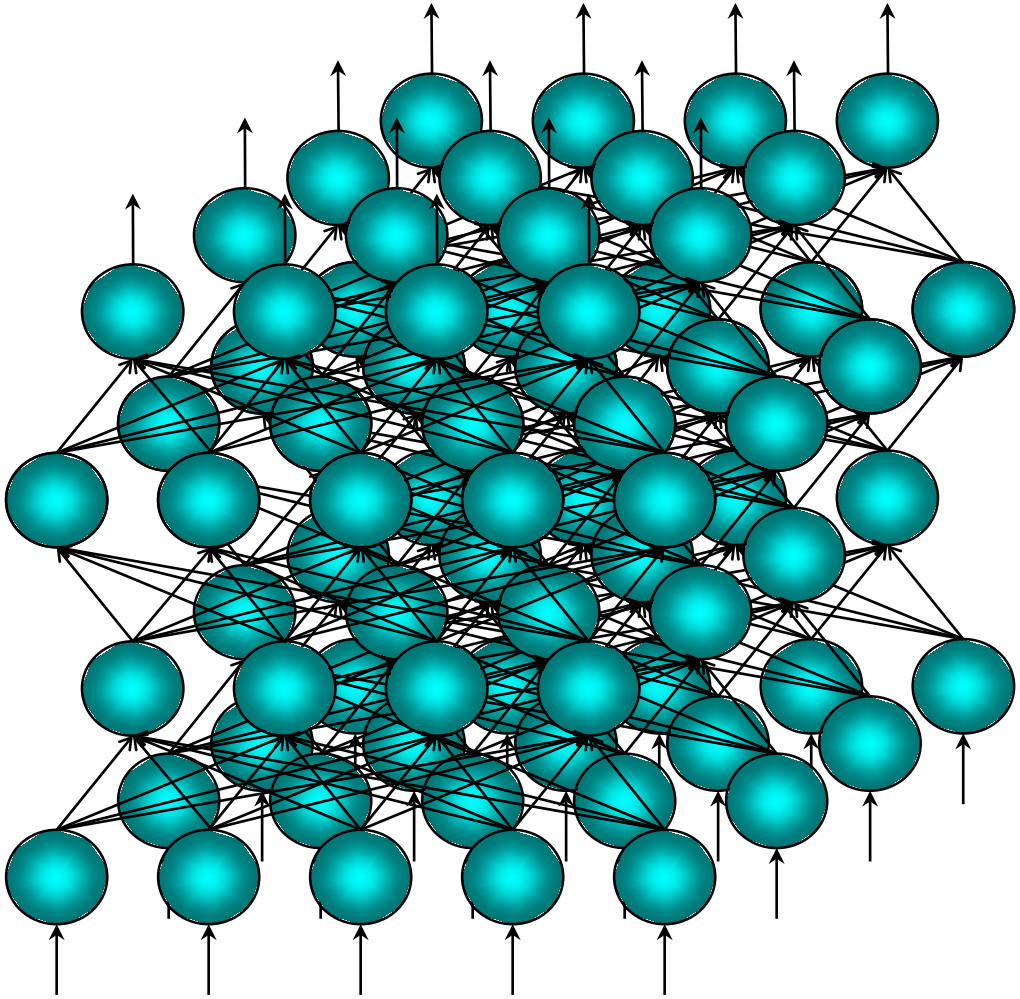
「$\eta(t) \to 0$」が成り立たないと永久に揺れたまま

「$\eta(0)+\eta(1)+\cdots+\eta(t) \to \infty$」が成り立たないと途中で止まる



神経回路網　*Neural Network*

疑問:　ものすごく複雑なモデルでも　学習することができるのだろうか



非線型　nonlinear

特定　Non
できない　identifiable

隠れた部分に
ついての対称性
symmetry

非常に多くの
パラメータ
Many parameters

歴史はめぐる

生体神経回路網を模倣して作られた大きな複雑さを持つ学習モデルが
現実の問題解決に役立つことが広く知られるようになった。

単純パーセプトロン　1960頃
多層パーセプトロン　1985頃
SVM　1995頃
深層学習　2010頃

（注意） この分野は宣伝が 過大 と 過小 を繰り返している。

夏 → 冬 → 夏 → 冬 → ・・・

「生体神経回路を模倣して作られた」ということが、多すぎる期待を与え、
それはまた多くの落胆を引き起こしやすい。

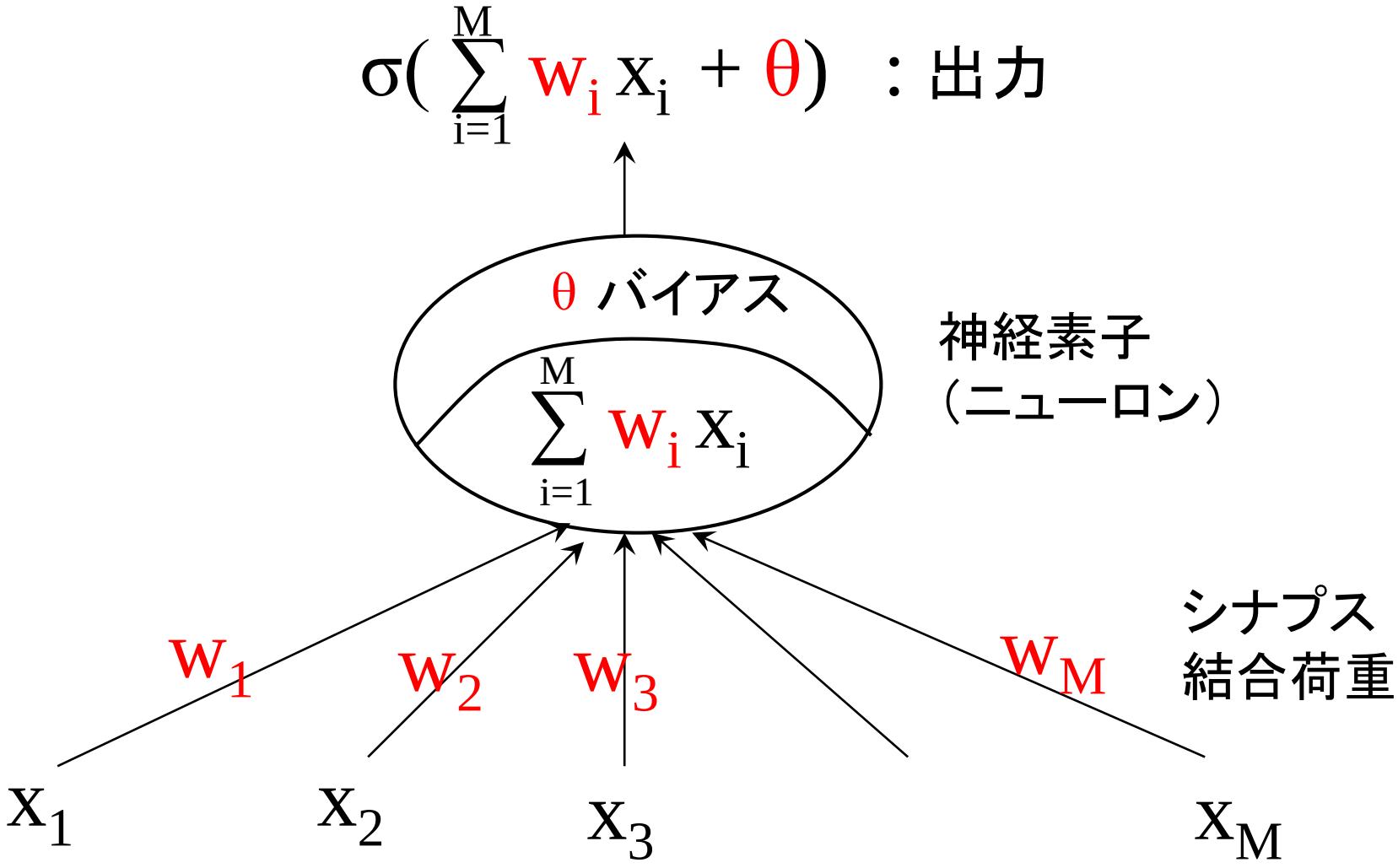
◎　実際の研究は20年の単位で一歩ずつ進展している。

神経素子

1個の Neuron のモデル

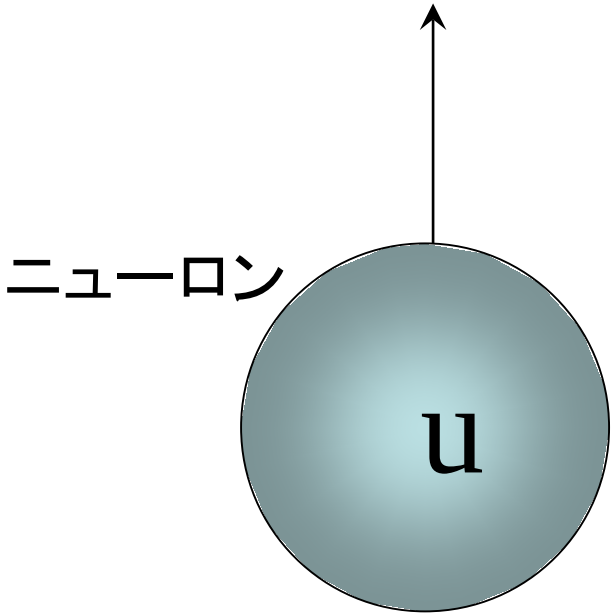
$(x_1,\ldots,x_M)$: 外界からの入力

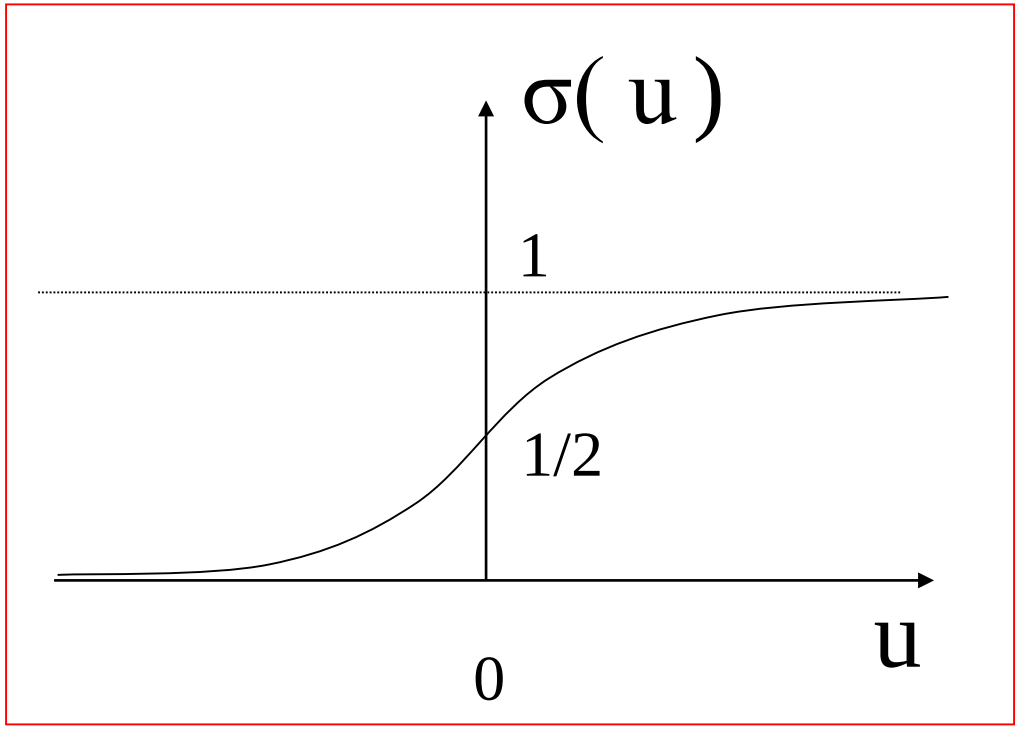
$(w_1,\ldots w_M, \theta)$: パラメータ



非線型応答:

$$\sigma(u) = 1/(1+e^{-u})$$: シグモイド関数

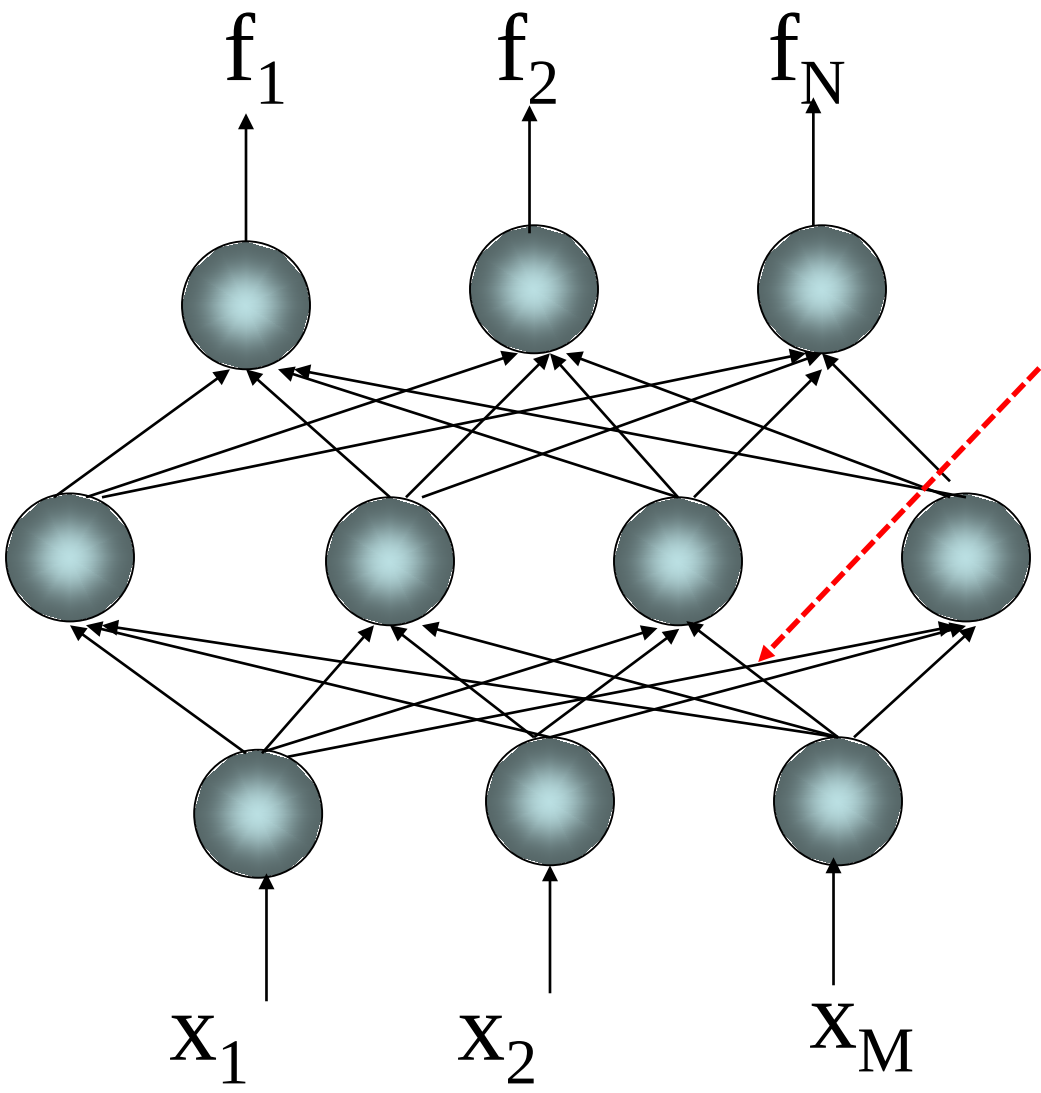




（注意）この形の関数に数学的な意味があるかどうかはわかっていない。

三層パーセプトロン

Three-Layer Perceptron

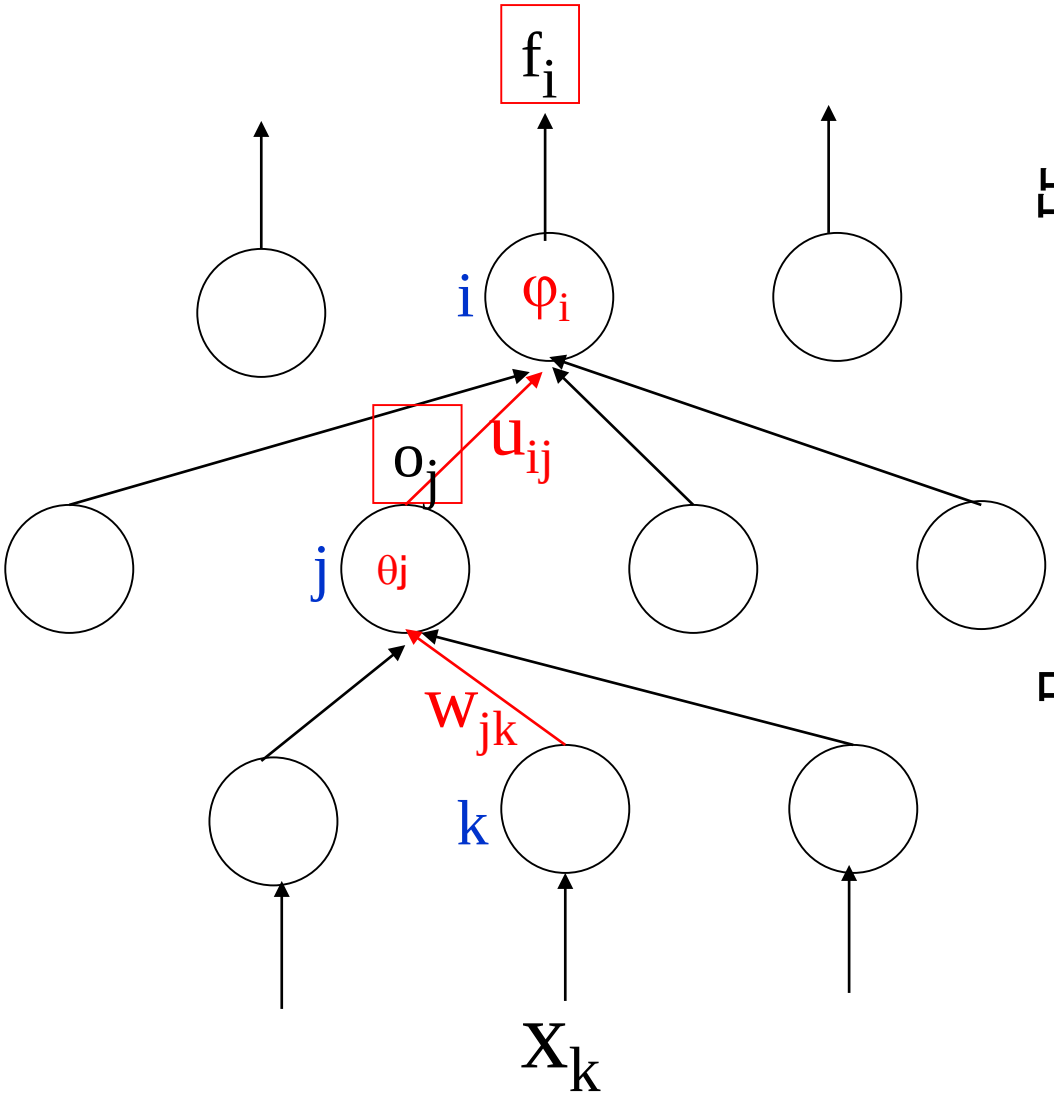


遠い遠い昔、
「入力→中間」を
学習させるのは
不可能だと
思われていた。
「出力の誤差から
ここを学習する
なんて無理・・・」

最急降下でやって
みたらできた。
(Rumelhart
＆McClelland,
1985)

モデルの説明

同じものを並べるだけなので、プログラムは簡単です。



出力

$$f_i = \sigma\left(\sum_{j=1}^{H} u_{ij} o_j + \varphi_i\right)$$

中間

$$o_j = \sigma\left(\sum_{k=1}^{M} w_{jk} x_k + \theta_j\right)$$

数式で書くと

出力ユニットN個 φ_i

u_{ij}

中間ユニットH個 θ_j

w_{jk}

入力ユニットM個

} パラメータ全部の集合を 1文字 w で表す

$$f_i(w)=\sigma\left(\sum_{j=1}^{H} u_{ij}\ \sigma\left(\sum_{k=1}^{M} w_{jk}x_k+\theta_j\right)+\varphi_i\right)$$

一個のデータ $x=(x_1,x_2,\ldots,x_M)$, $y=(y_1,y_2,\ldots,y_N)$ についての確率降下は

$$-\nabla\left\{\sum_{i=1}^{N}(y_i-f_i(x,w))^2\right\}$$

この∇が計算できると確率降下法が実行できる。

やってみたらできた

最急降下法の準備：微分の性質(1)

任意の関数 f,g,h の合成関数の微分

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

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

例えば　$y = \sigma(o)$

$o = w_1x_1 + w_2x_2 + \cdots + w_Mx_M$

のときは　$\partial y/\partial w_i = \sigma'(o)\, x_i$

最急降下法の準備(2)

シグモイド関数のとき

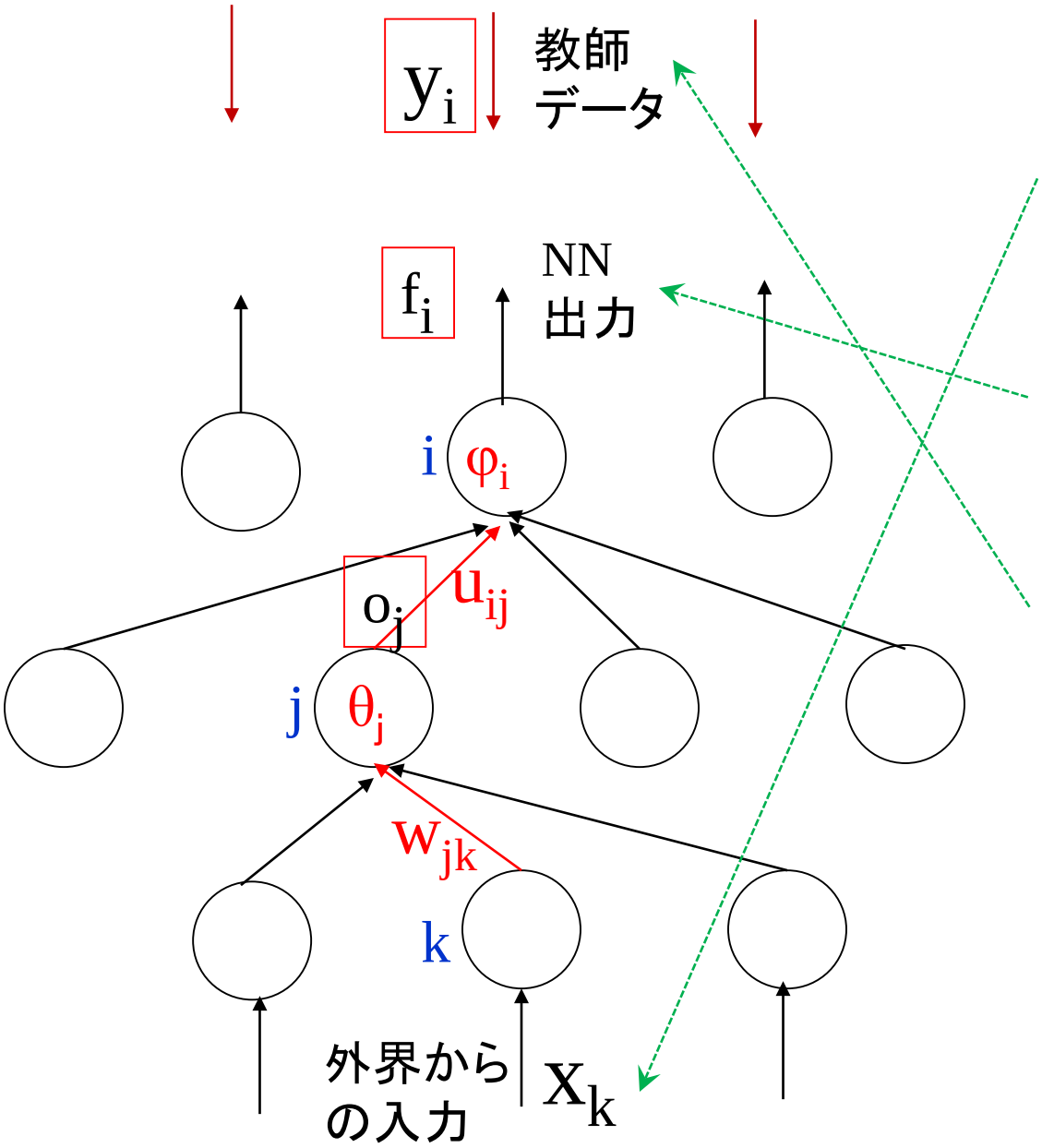
$\sigma(x) = 1 / (1+ e^{-x})$ の微分は

$\sigma'(x) = \sigma(x) (1 - \sigma(x))$

つまり関数の微分を関数の出力だけで計算できる。

以上のことを用いると確率降下の式は、
少ない演算で計算できる。

学習の様子



入力 $x = (x_1, x_2, \ldots, x_M)$

に対する

神経回路網の答え

$(f_1, f_2, \ldots, f_N)$

教師データ

$(y_1, y_2, \ldots, y_N)$

二乗誤差が小さくなるよう

に各パラメータを変更

最急降下法の計算

神経回路網の出力を $f = (y_i)$, 教師データを $y= (y_i)$ とする。

二乗誤差は $E(w) = \frac{1}{2} \sum_{i=1}^{N} (f_i - y_i)^2$ (N : 出力の次元)

「中間→出力」の結合荷重についての微分は

$$\frac{\partial E}{\partial u_{ij}} = (f_i - y_i) \frac{\partial f_i}{\partial u_{ij}}$$

$$= (f_i - y_i) f_i (1- f_i) o_j$$

$$f_i = \sigma\left(\sum_{j=1}^{H} u_{ij} o_j + \varphi_i\right)$$

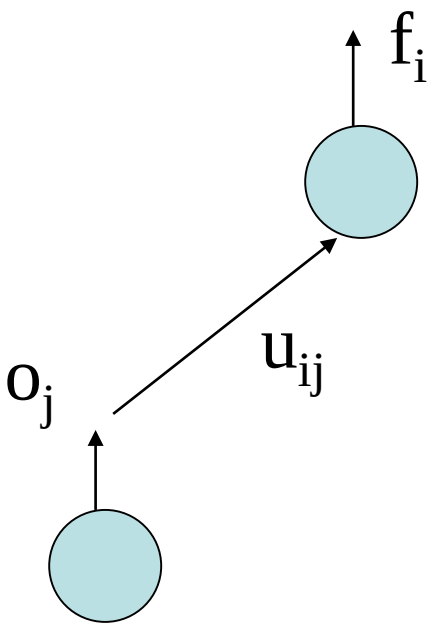
◎ つまり微分は各ニューロンの出力を組み合わせると計算できる。

（参考）生体神経回路との関係について

Biological Neural Network

◎　導出された最急降下法は、自然界にある生体神経回路でも
使われているのだろうか（？）。小脳では使われているらしい（？？）

$$u_{ij} := u_{ij} - \eta(f_i - y_i)\, f_i\,(1- f_i)\, o_j$$



システム全体の誤差の最急降下を
各ニューロンのローカルな値だけを使って
計算できる。生体神経回路と類似するか
どうかは、まだわかっていない。

（参考）　Hebb 則

生体神経回路の結合荷重は、
通った信号の強さに比例して
強くなる

問題1　$E(w) = \frac{1}{2} \sum_{i=1}^{N} (f_i - y_i)^2$

「入力→中間」の結合荷重についての微分を計算してみましょう。

---

(1) $\frac{\partial f_i}{\partial o_j}$　(2) $\frac{\partial o_j}{\partial w_{jk}}$　(3) $\frac{\partial E}{\partial w_{jk}}$　をそれぞれ求めよ。

(3) の結果を誤差逆伝播法という

ヒント：

$$\frac{\partial E}{\partial w_{jk}} = \sum_{i=1}^{N} (f_i - y_i) \frac{\partial f_i}{\partial w_{jk}}$$

$$\frac{\partial f_i}{\partial w_{jk}} = \frac{\partial f_i}{\partial o_j} \frac{\partial o_j}{\partial w_{jk}}$$

（注意）どんなに深い学習モデルでもこれと同じ計算法が使える。

中間→出力

$$f_i = \sigma\left(\sum_{j=1}^{H} u_{ij} o_j + \varphi_i\right)$$

入力→中間

$$o_j = \sigma\left(\sum_{k=1}^{M} w_{jk} x_k + \theta_j\right)$$

いろいろなことに応用してみよう

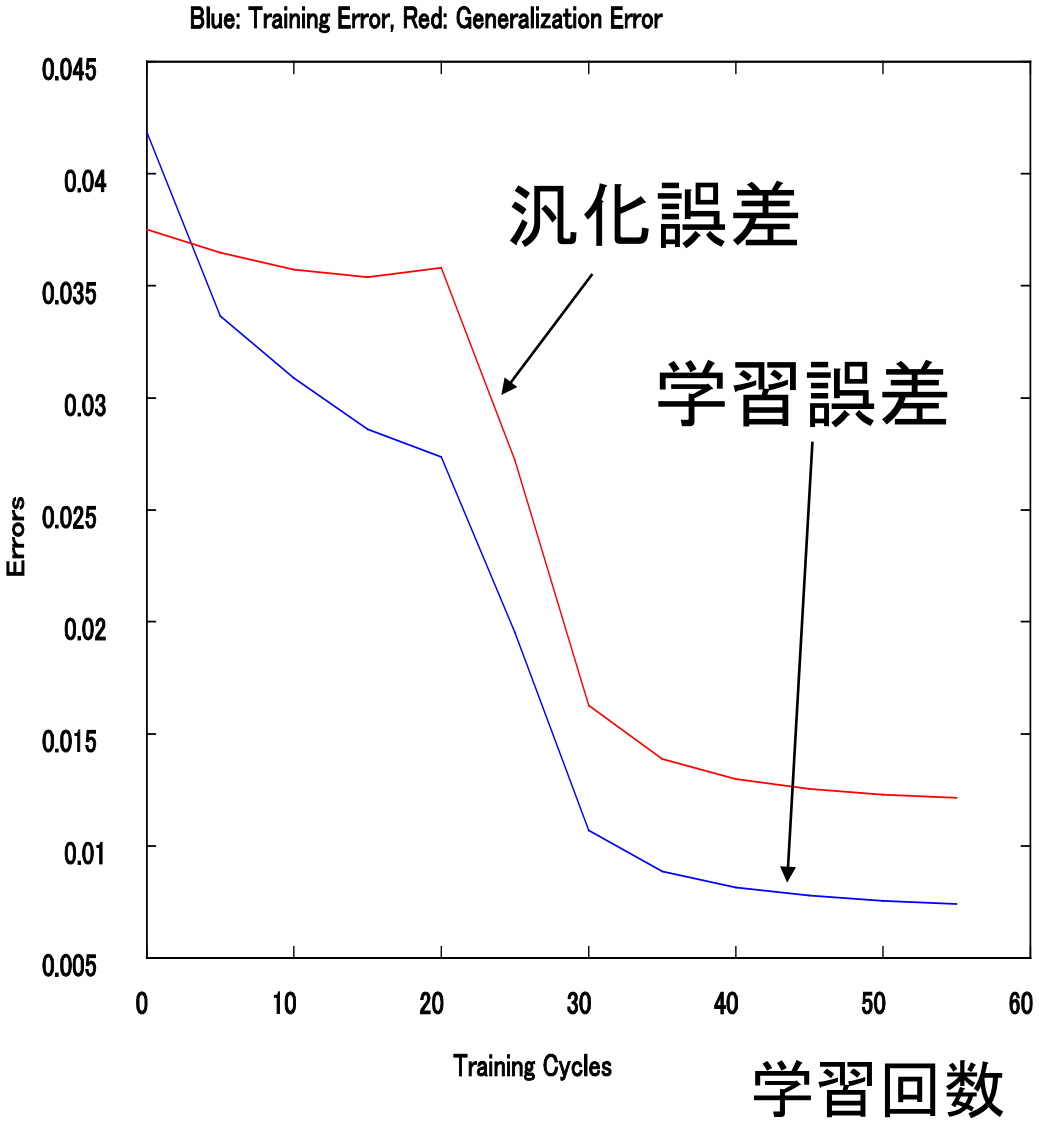
構造の発見

適切なパラメータを
見つけるには
時間がかかる。

運が悪いと
見つからないときもある。

適切なパラメータが
見つかると
急速に学習が進む。

ずっと続けると・・・。



学習誤差と汎化誤差

学習誤差 （学習データ (X_i,Y_i)を使う）

$$E(w) = (1/n) \sum_{i=1}^{n} (Y_i - f(X_i,w))^2$$

汎化誤差 （学習データとは違うテスト用データ (X_j,Y_j) を使う）

$$E(w) = (1/m) \sum_{j=1}^{m} (Y_j - f(X_j,w))^2$$

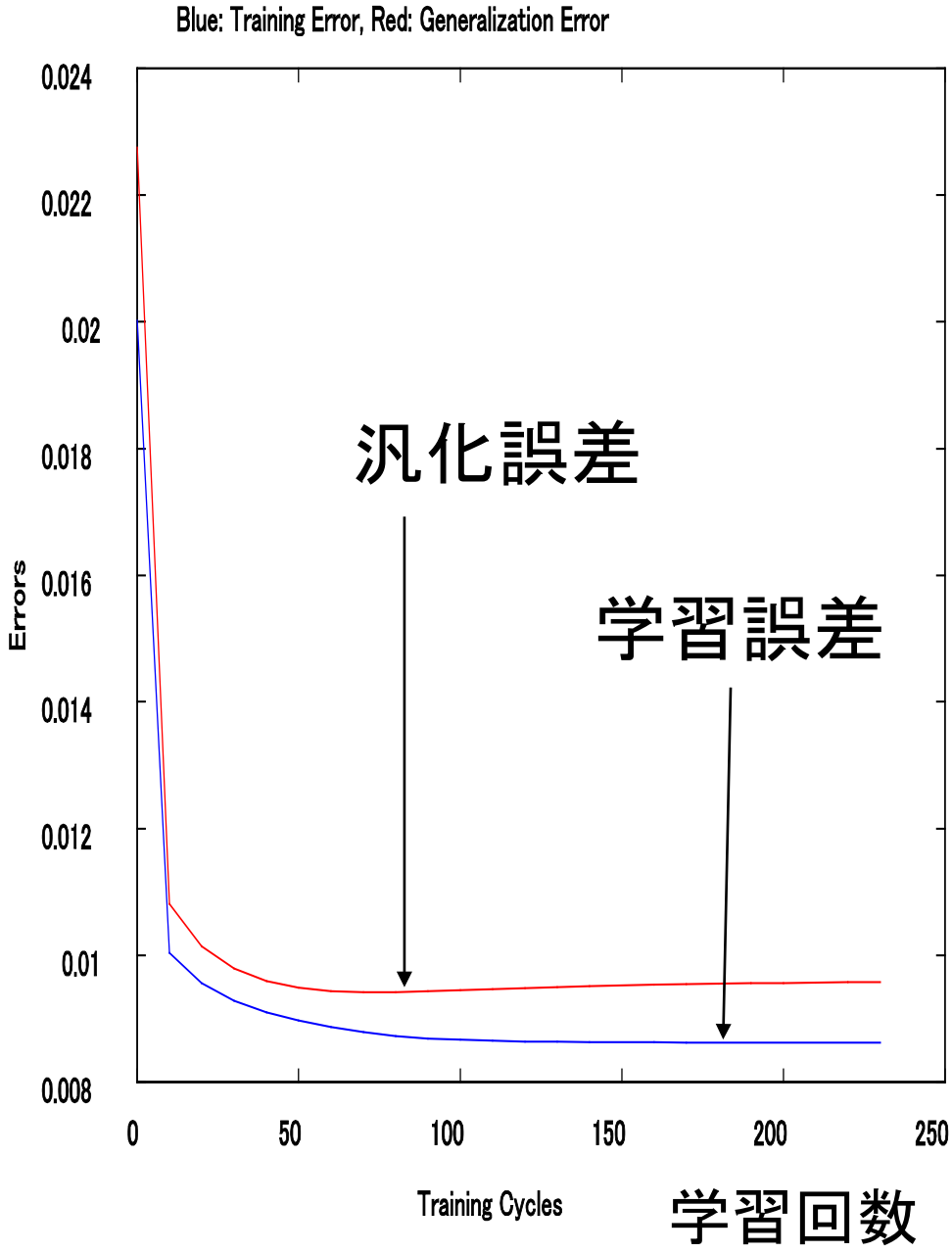
学習誤差が小さくなるように学習すると
汎化誤差も小さくなるのだろうか？

過学習現象

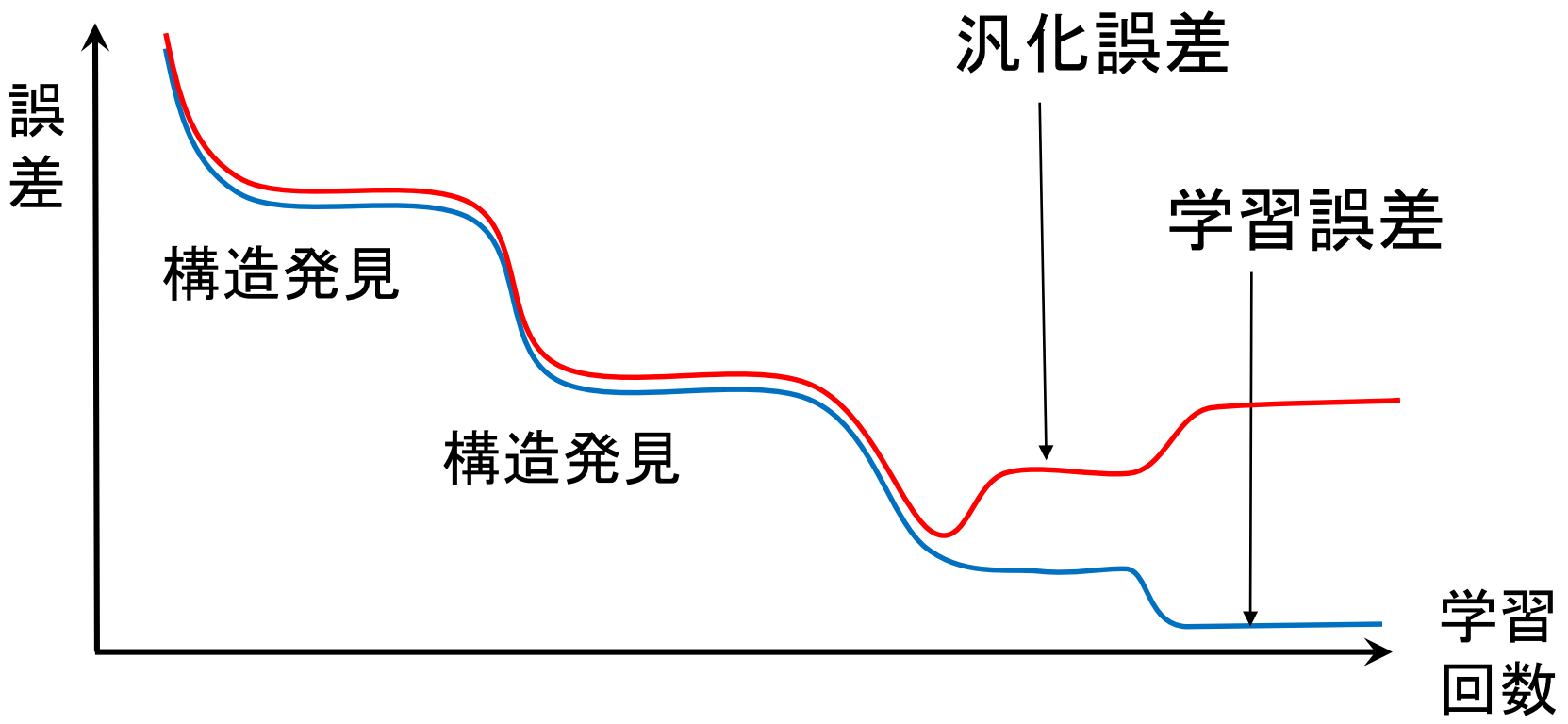
学習は「大局→局所」
の順で進んでいく。

学習の初期は
学習誤差と汎化誤差は
同様に小さくなる。

学習が進むと
学習誤差は小さくなる
にも関わらず
汎化誤差は増大する
ことがある。



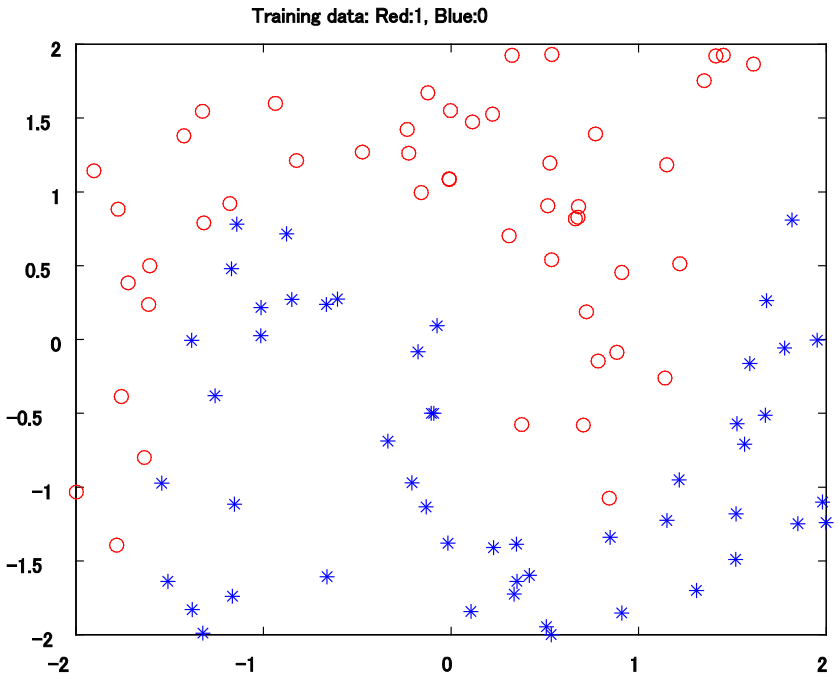
「構造の発見」と「過学習」は区別できるか



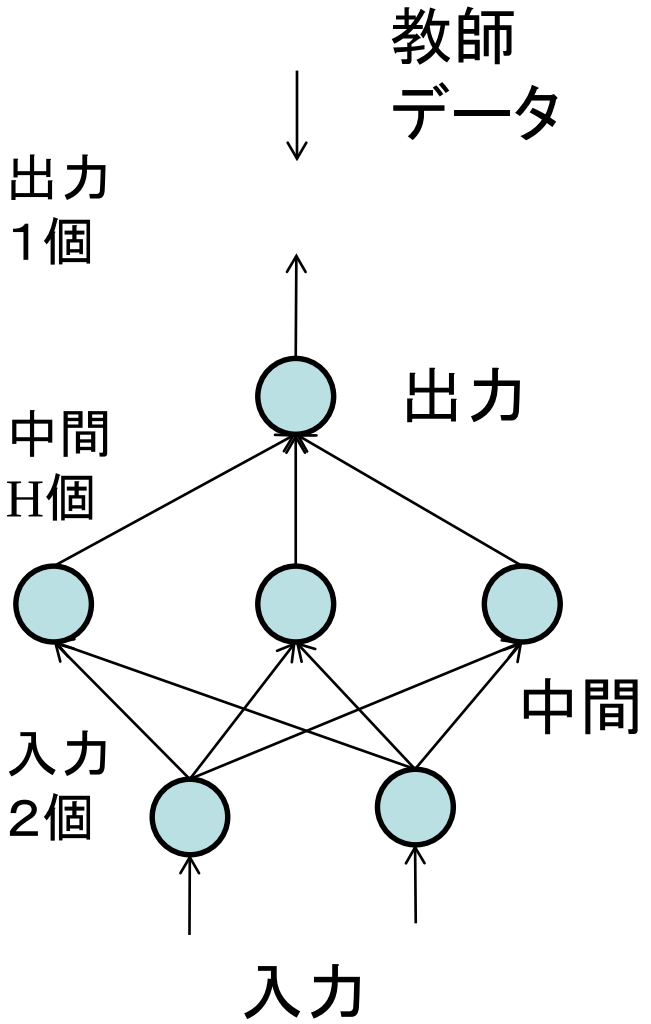
◎　複雑なモデルでも最急降下で学習できる。
◎　「構造の発見」ができる学習モデルは
　　過学習を起こしやすいモデルでもある。

問題2

ニューラルネットの学習の過程を観察してみよう。学習は、初期、中期、後期でどのように進むのだろうか。



2次元学習データの例



問題2

学習誤差と汎化誤差の挙動は似ているか？

	初期	中期	その後
学習の様子			
学習誤差と 汎化誤差			