確率的勾配降下法

Stochastic gradient descent (SGD)

平均値の逐次計算

目的: 値sを観測値s_i $(i=1,2,...)$の平均値に近づける計算をオンライン(データ取得ごとに)行う

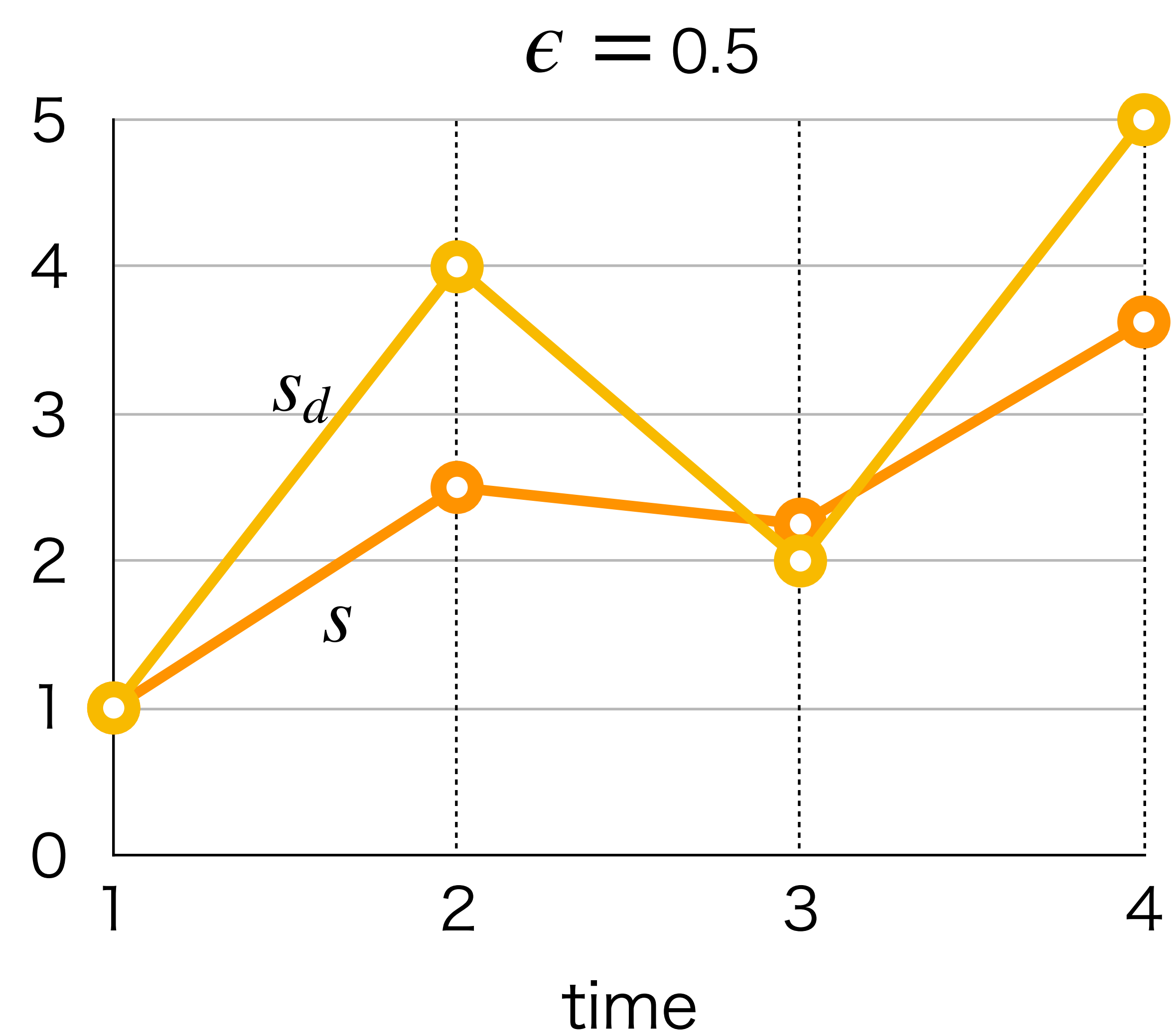
学習則: $s \leftarrow$ **s+1/k(si-s)**

$=$ **s+a(si-s)**

(k=1のとき $s \leftarrow s_i$)

単純化した表記:

$\Delta s =$ **a(si-s)**



上記の学習則をちょっと違う方法で導出してみる

確率的勾配降下法: 単純な例

目的: 値sを観測値s_i $(i=1,2,\dots)$の平均値に近づける計算をオンライン(データ取得ごとに)行う

誤差関数: $$E(s) = \frac{1}{2}(s - s_i)^2$$

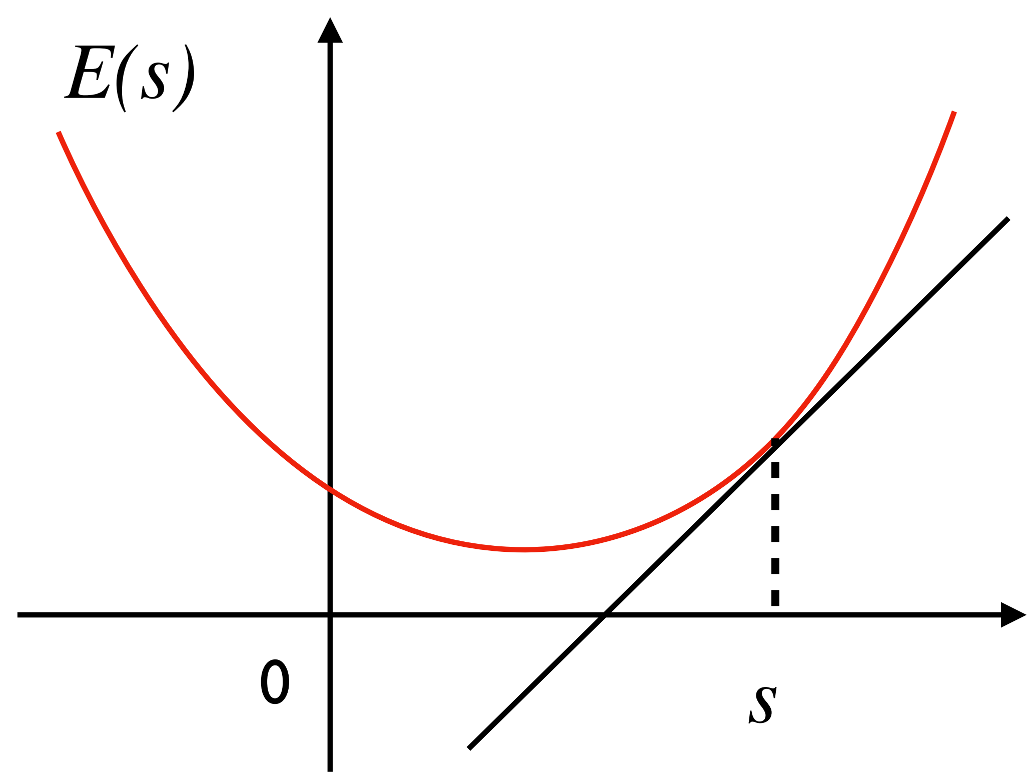
確率的勾配降下法: 誤差関数の値を小さくするように学習

$$\Delta s = -\epsilon \frac{dE}{ds}$$

$$= \boxed{}$$

$$s \leftarrow \boxed{}$$

前ページの学習則と一致



確率的勾配降下法: 単純な例

目的: 値sを観測値s_i $(i=1,2,...)$の平均値に近づける計算をオンライン(データ取得ごとに)行う

誤差関数: $E(s) = \dfrac{1}{2}(s - s_i)^2$

確率的勾配降下法: 誤差関数の値を小さくするように学習

$$\Delta s = -\epsilon \frac{dE}{ds}$$

$$= \boxed{}$$

$$s \leftarrow \boxed{}$$

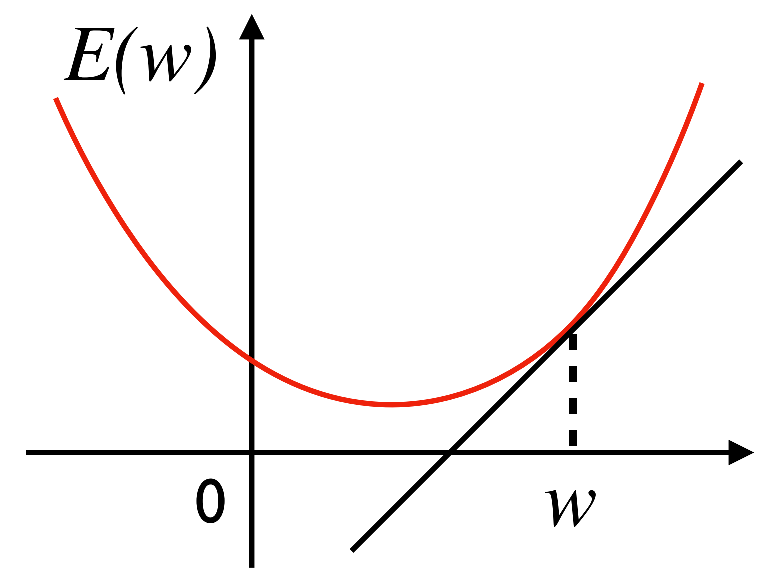
強化学習もSGDの変形版

$$Q(s,a) \leftarrow Q(s,a) + \epsilon(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

$$V(s) \leftarrow V(s) + \epsilon(r + \gamma V(s') - V(s))$$

確率的勾配降下法(SGD)の一般型

誤差関数 $E(\boldsymbol{w}) = \frac{1}{2}(f(x;\boldsymbol{w}) - d)^2$



学習則

$$\Delta w_i = -\epsilon \frac{\partial E(\boldsymbol{w})}{\partial w_i}$$

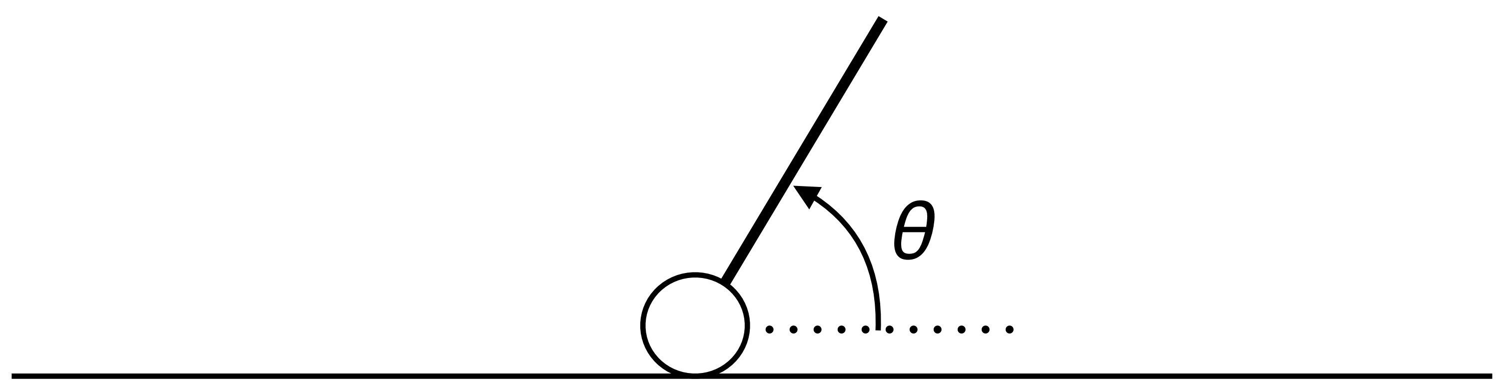
$$= -\epsilon (f(x;\boldsymbol{w}) - d) \frac{\partial f(x;\boldsymbol{w})}{\partial w_i}$$

局所解に陥りやすいため，さまざまな変形版がある

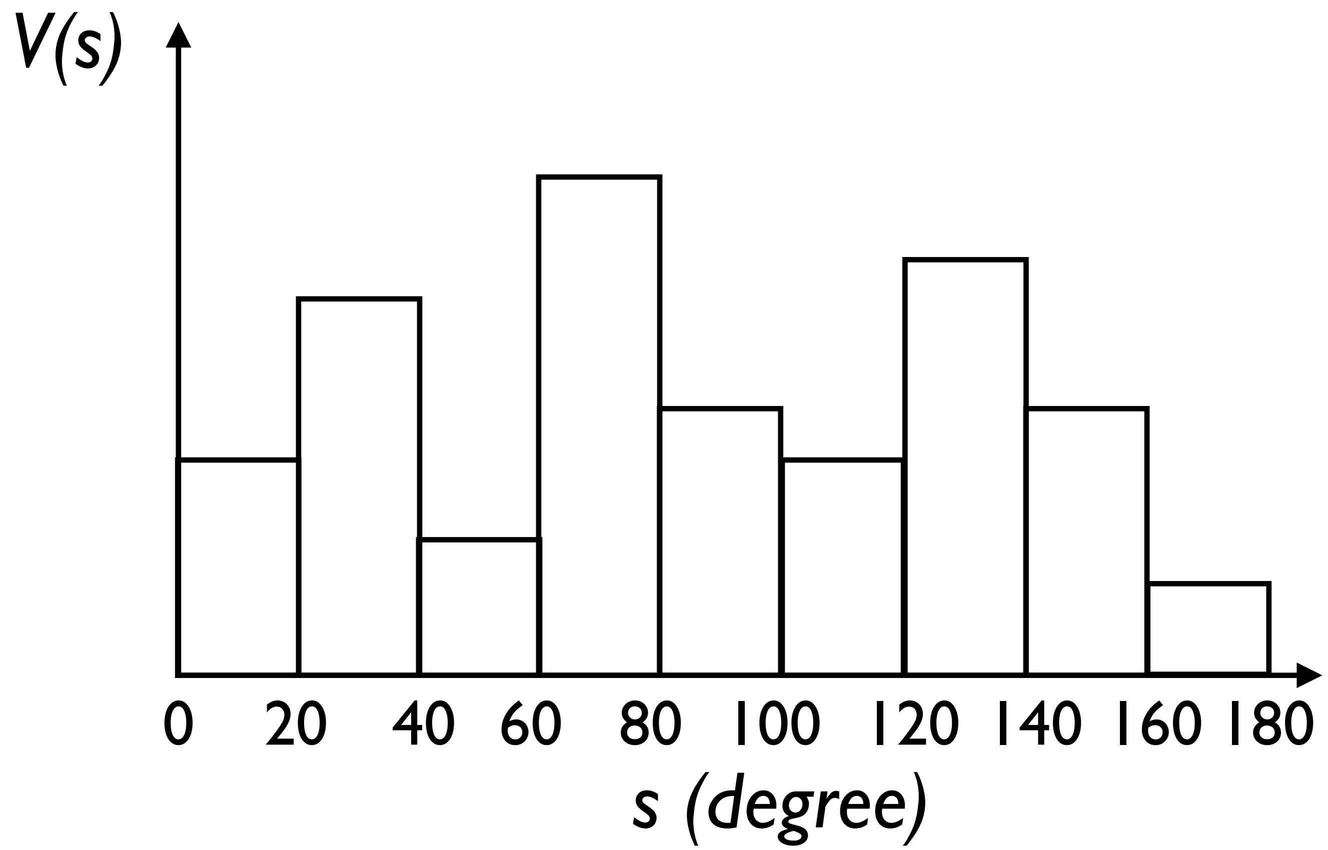
確率的勾配降下法 (SGD)の活用例

強化学習の価値関数の近似

状態数sが連続値をとる場合
価値関数の定義方法

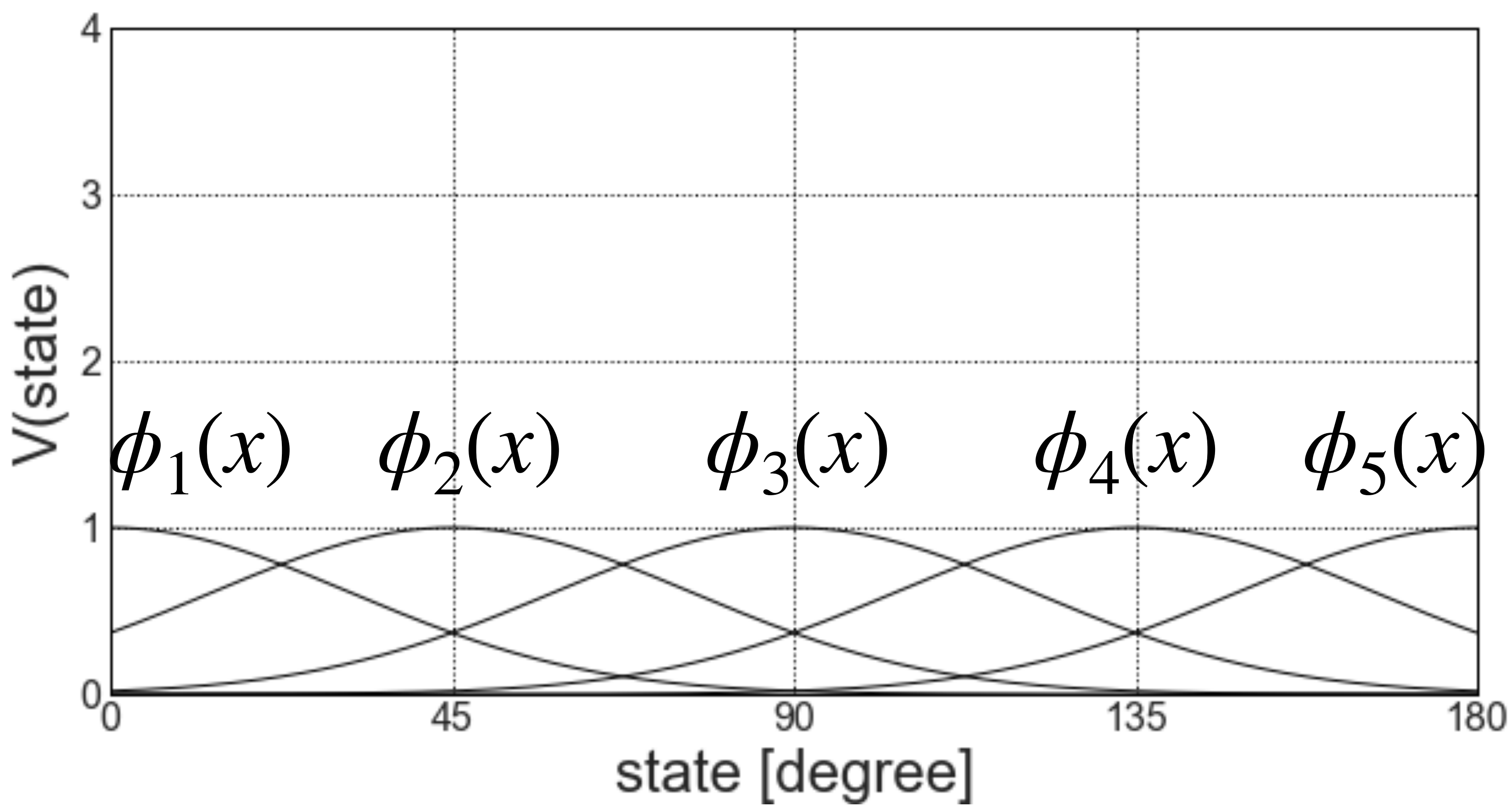


方法1: 状態を離散化



- V(s)を正確に近似するために状態数を増やすと，学習に時間がかかる
- 隣接する状態値sに対する価値関数V(s)は，多くの場合似た値になるので，各状態値に対して独立に学習を行うのは非効率

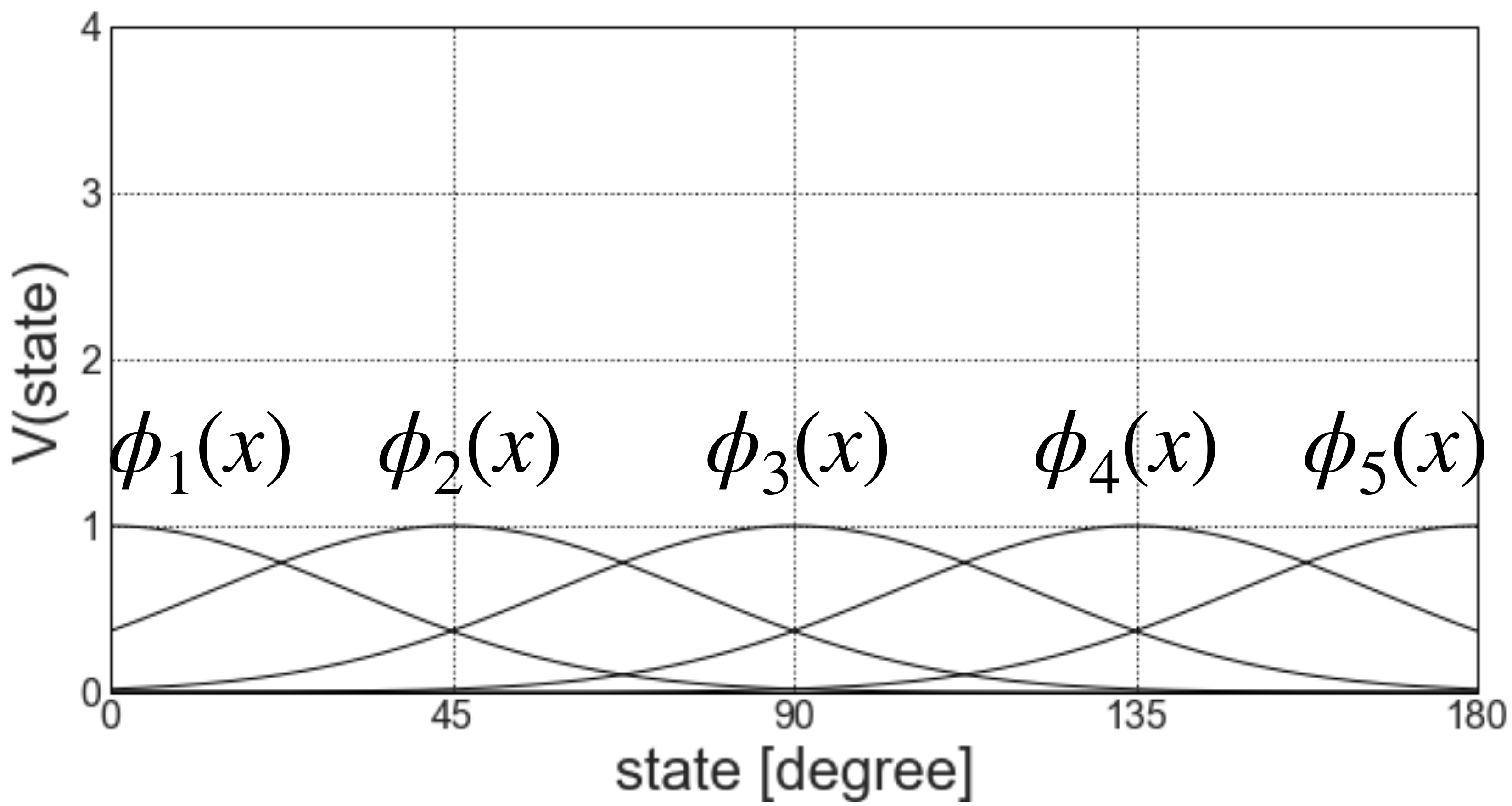
方法2: 関数近似法



基底関数の組み合わせで関数近似を行う

基底関数の例

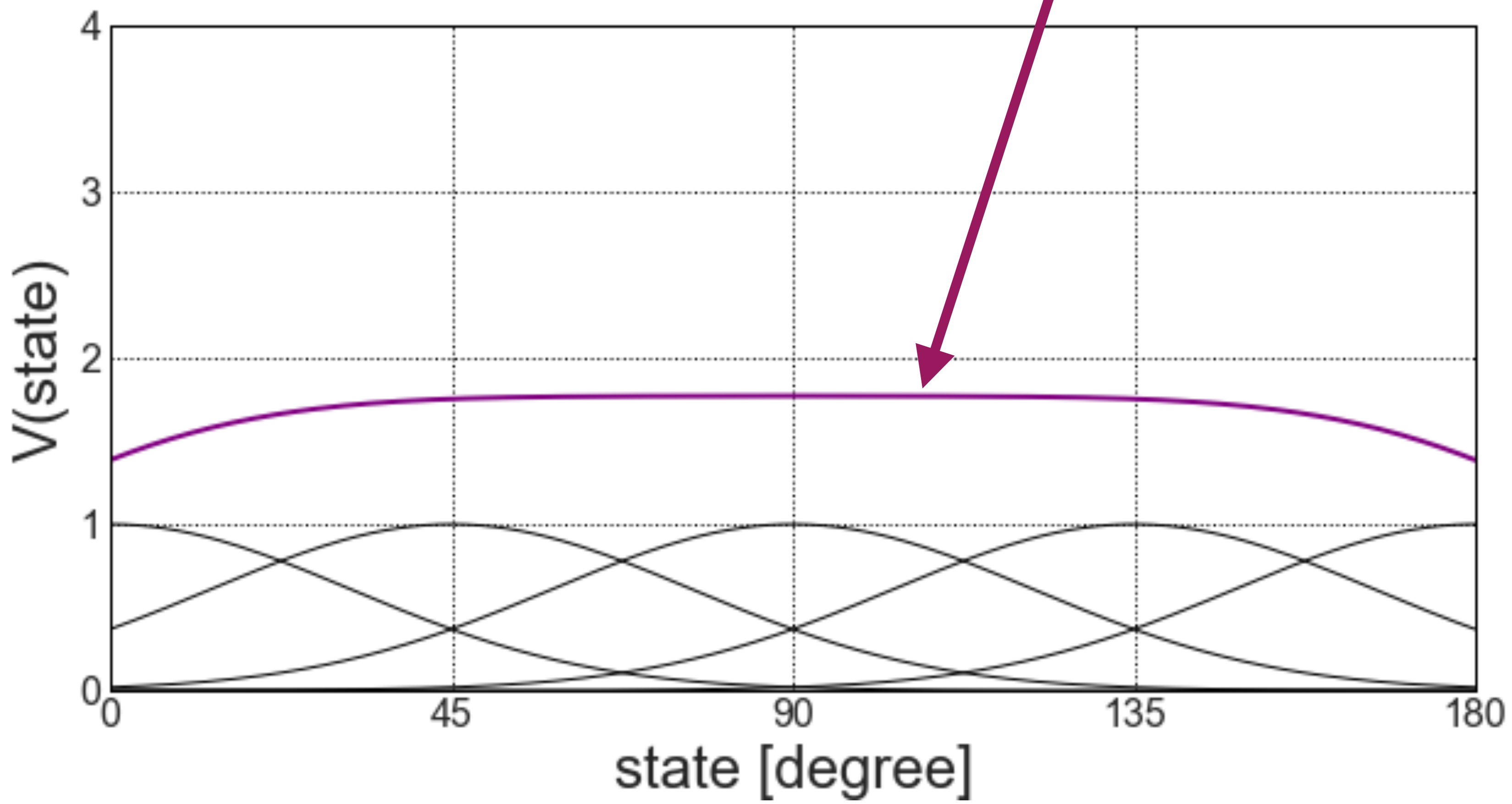
Radial Basis Function (RBF, 動径基底関数)



$$\phi_i(x) = \boxed{} \quad (i = 1, \ldots, N)$$

RBFによる関数近似

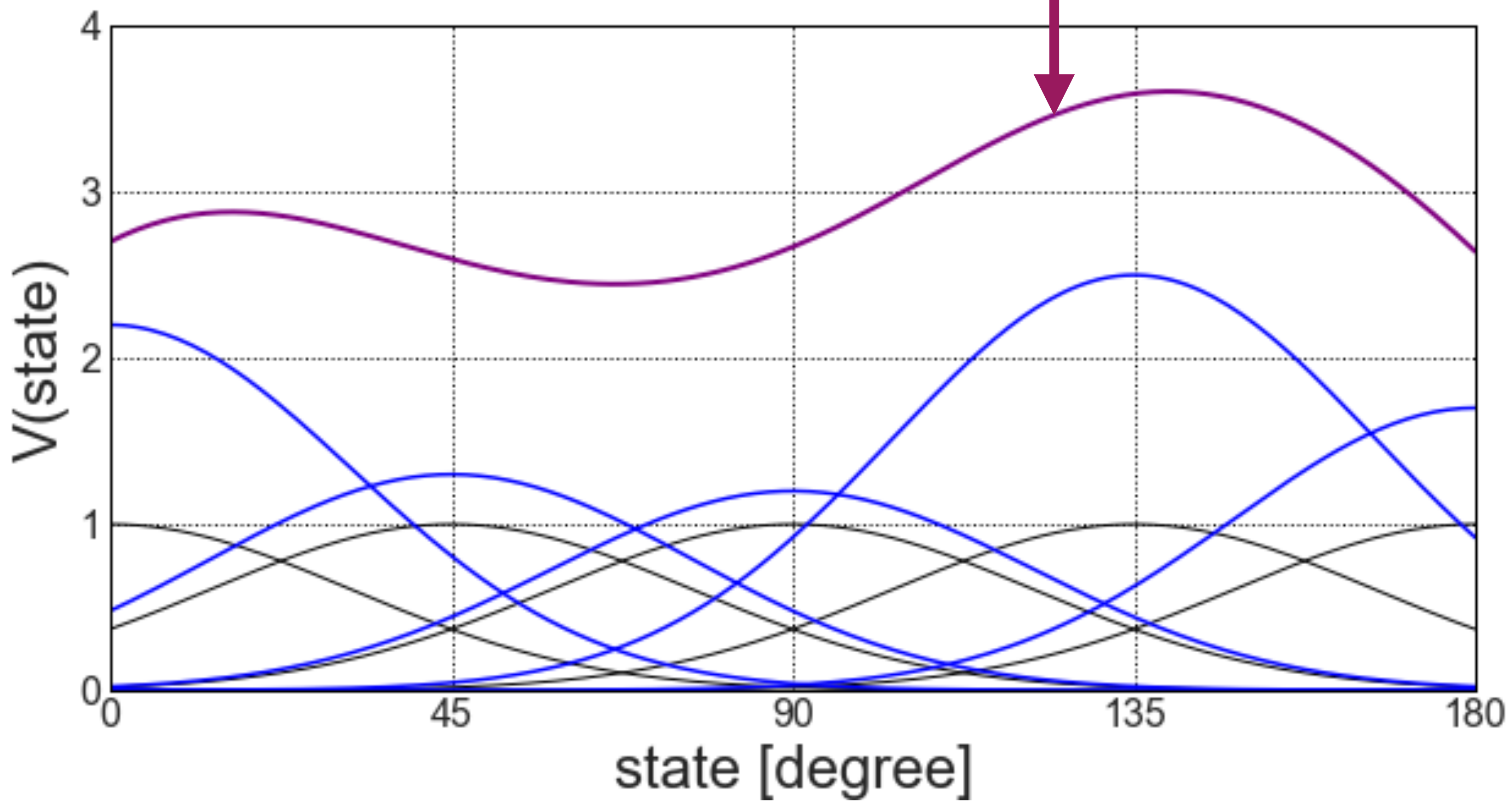
$\boldsymbol{w} = (1,1,1,1,1)$



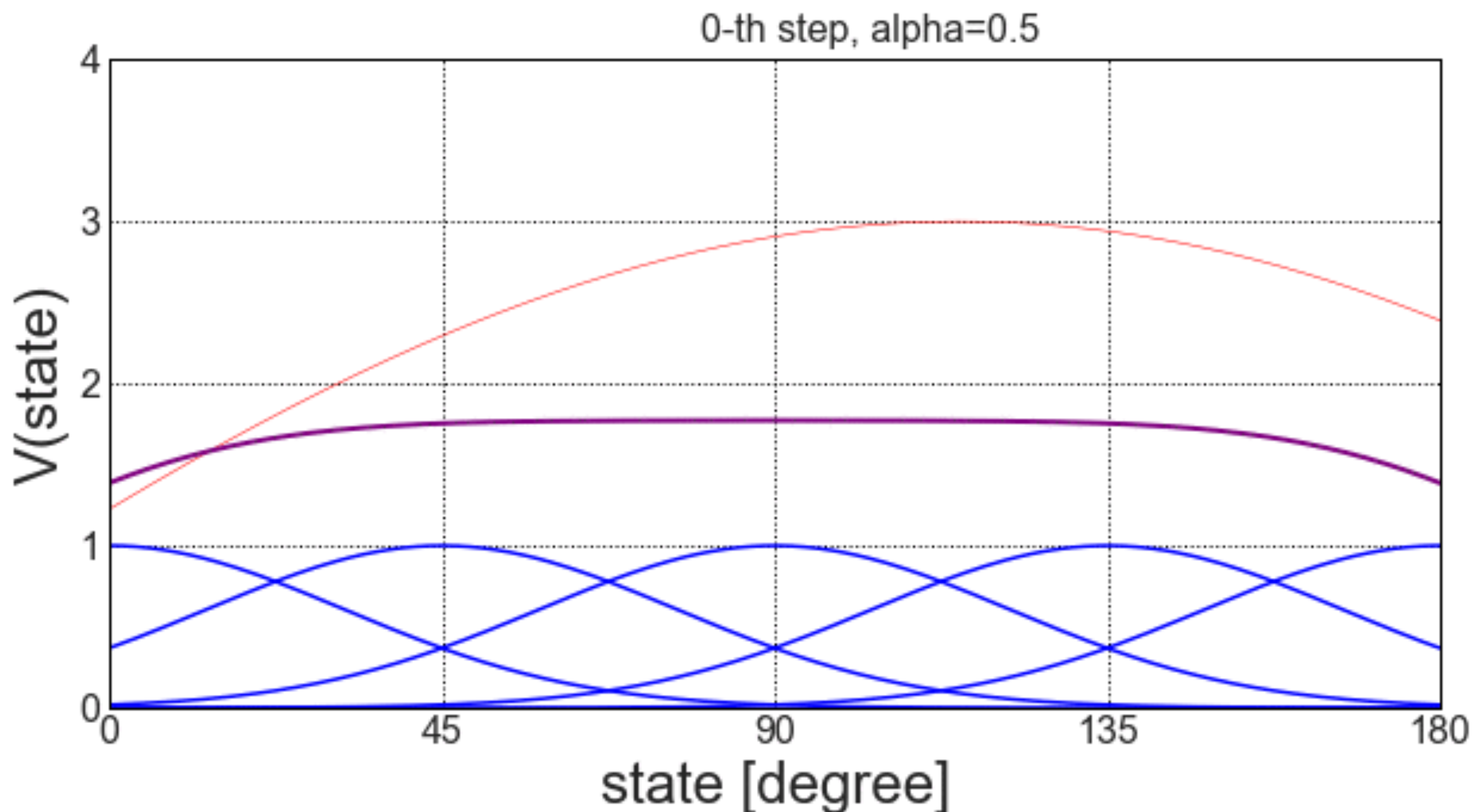
$$V(s;\boldsymbol{w}) = \sum_{i=1}^{N} w_i \phi_i(s) = \sum_{i=1}^{N} w_i \boxed{}$$

RBFによる関数近似

$\boldsymbol{w} = (2.2, 1.3, 1.2, 2.5, 1.7)$



$$V(s; \boldsymbol{w}) = \sum_{i=1}^{N} w_i \phi_i(s) = \sum_{i=1}^{N} w_i \boxed{}$$



近似関数 $$V(s;\mathbf{w}) = \sum_{i=1}^{N} w_i \phi_i(s) = \sum_{i=1}^{N} w_i e^{-(\frac{s-m_i}{\sigma})^2}$$

誤差関数 $$E(\mathbf{w}) = \frac{1}{2}(V(s;\mathbf{w}) - d)^2 \qquad d \equiv r + \gamma V(s';\mathbf{w})$$

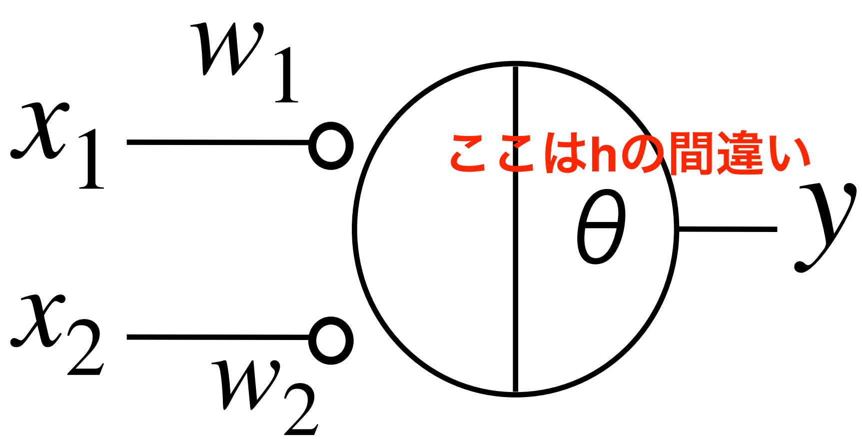
学習則

Q. 価値関数$V(s)$の近似関数$V(s;\mathbf{w})$を得るための学習則を最急降下法により導きなさい

確率的勾配降下法 (SGD)の活用例

形式ニューロンの学習

入出力関数(形式ニューロンモデル)



$$s = \sum_{i=1}^{N} w_i x_i - h$$

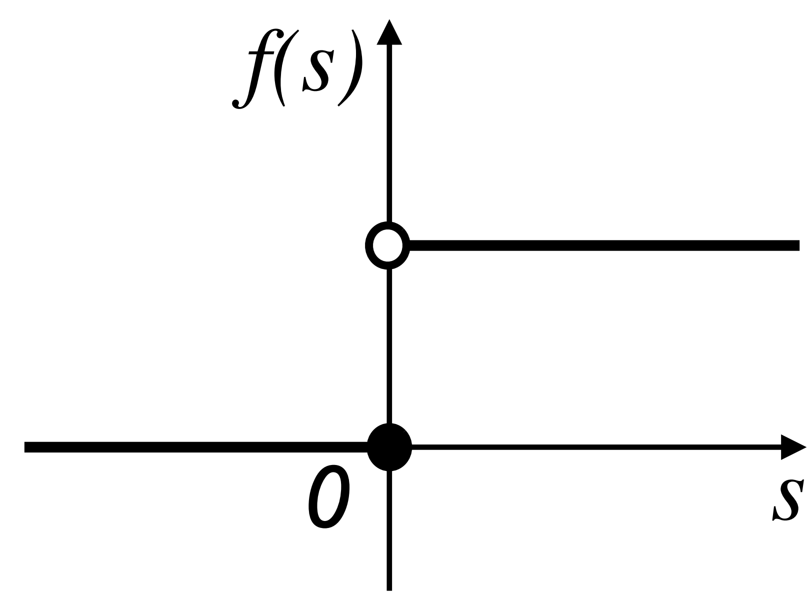
$$y = f(s) = \begin{cases} 1 & (s \geq 0) \\ 0 & (otherwise) \end{cases}$$

活性化関数はステップ関数

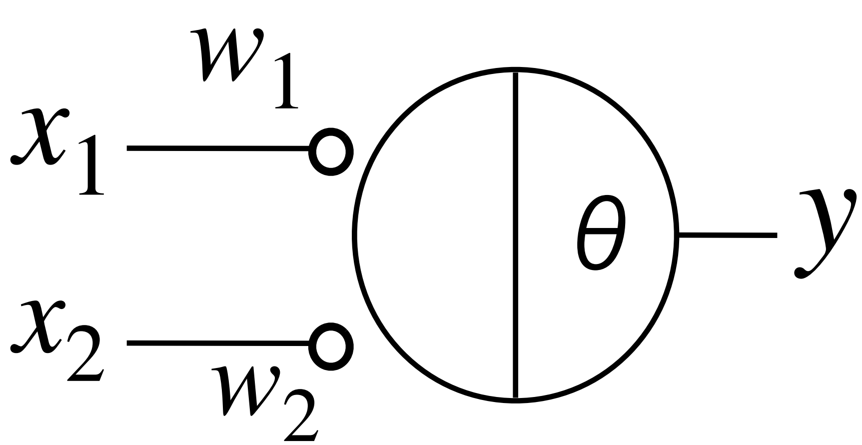
学習の目的

入力 (x_1, x_2) に対する出力 y が目標値 y_d に近づくようにパラメータ(結合重み)(w_1, w_2, h) を調整する

ydのdはdesired 正解の値だと思えばいい



入出力関数(形式ニューロンモデル)

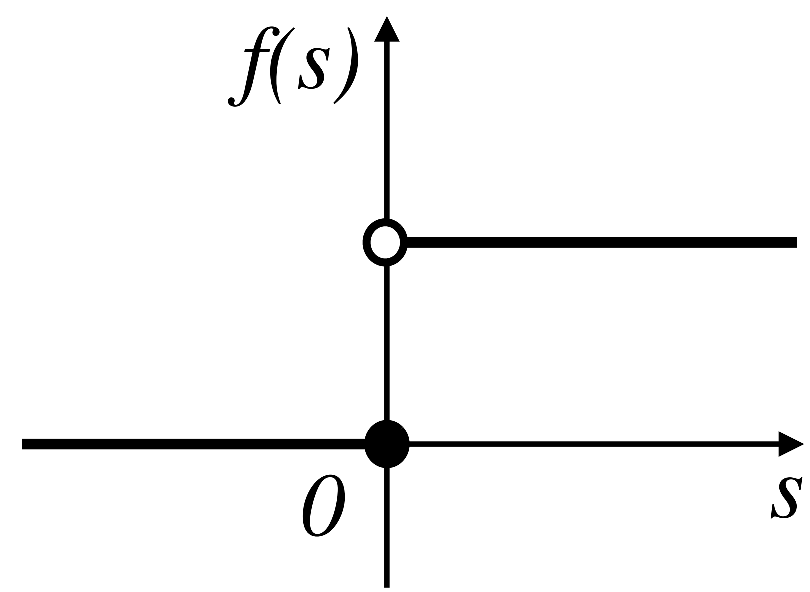


$$s = \sum_{i=1}^{N} w_i x_i - h$$

$$y = f(s) = \begin{cases} 1 & (s \geq 0) \\ 0 & (otherwise) \end{cases}$$

活性化関数はステップ関数

誤差関数 $E = \frac{1}{2}(y - y_d)^2$

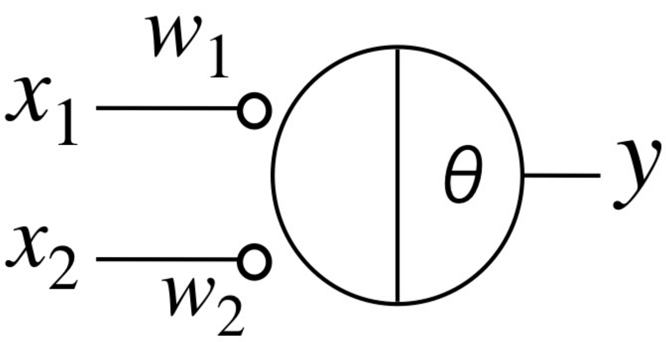


学習則 $\Delta w_i =$

$=$

$=$

入出力関数(形式ニューロンモデル)

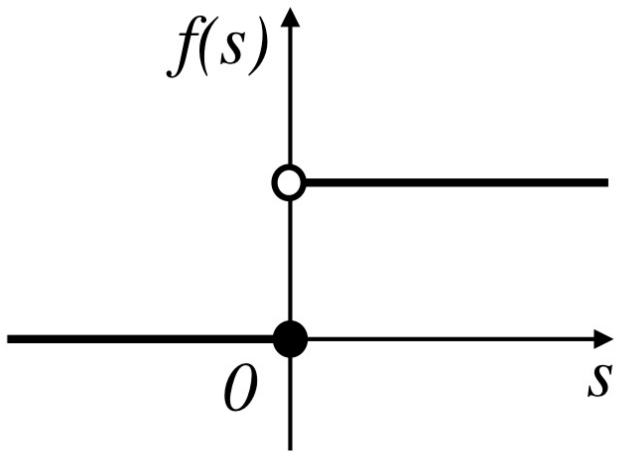


$$s = \sum_{i=1}^{N} w_i x_i - h$$

$$y = f(s) = \begin{cases} 1 & (s \geq 0) \\ 0 & (otherwise) \end{cases}$$

活性化関数はステップ関数

誤差関数 $E = \frac{1}{2}(y - y_d)^2$



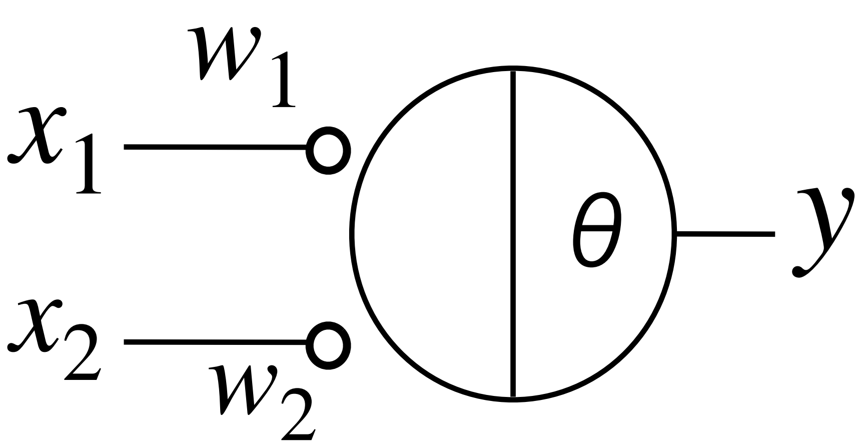
学習則 $\Delta w_i =$

$$\Delta w_i = -\,\varepsilon\,\frac{\delta E}{\partial w_i} = -\,\varepsilon\,\frac{dE}{dy}\frac{dy}{ds}\frac{\partial s}{\partial w_i}$$

$=$

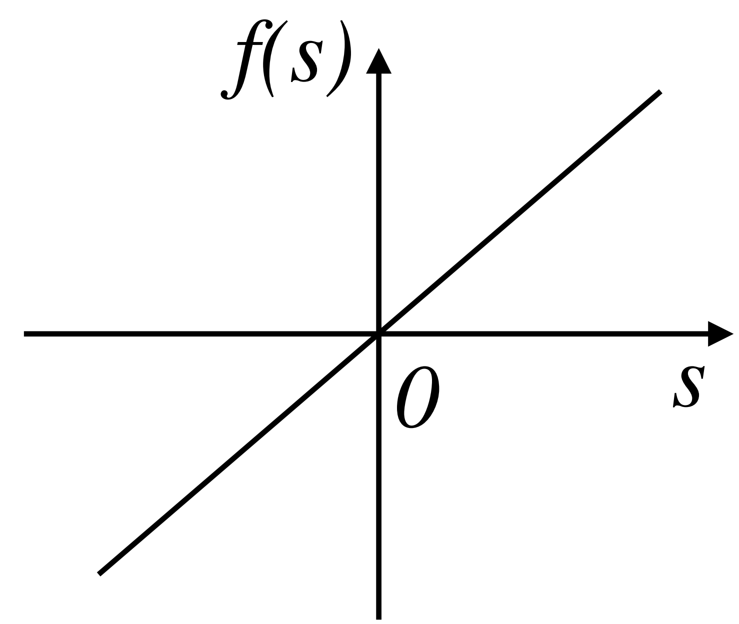
$=$

活性化関数を線形関数に



$$s = \sum_{i=1}^{N} w_i x_i - h$$

$$y = \boxed{f(s) = s}$$



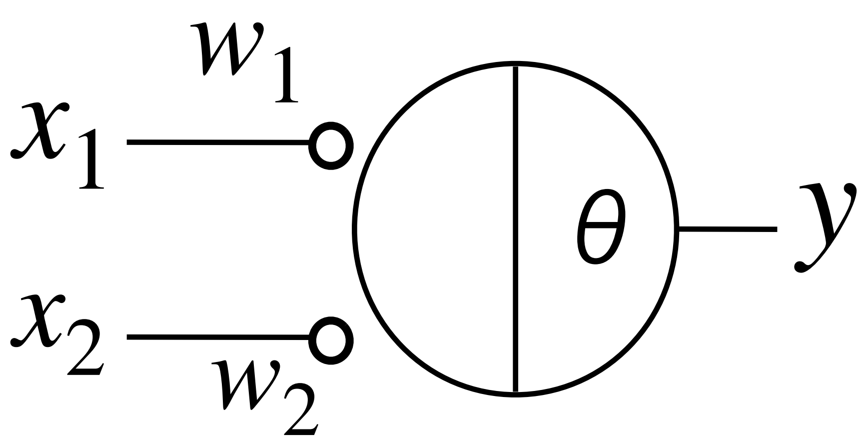
誤差関数 $E = \frac{1}{2}(y - y_d)^2 = \frac{1}{2}(f(s) - y_d)^2$

学習則 $\Delta w_i =$

$=$

$=$

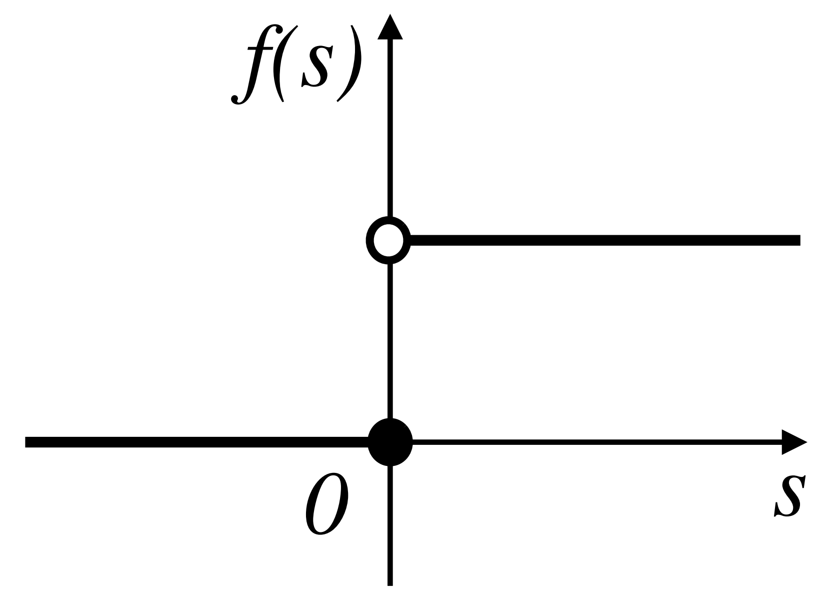
入出力関数(形式ニューロンモデル)



$$s = \sum_{i=1}^{N} w_i x_i - h$$

$$y = f(s) = \begin{cases} 1 & (s \geq 0) \\ 0 & (otherwise) \end{cases}$$

活性化関数はステップ関数



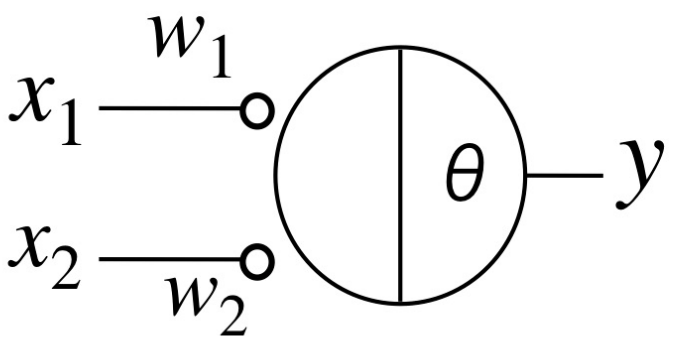
誤差関数 $E = \dfrac{1}{2}(y - y_d)^2$

学習則 $\Delta w_i =$

$=$

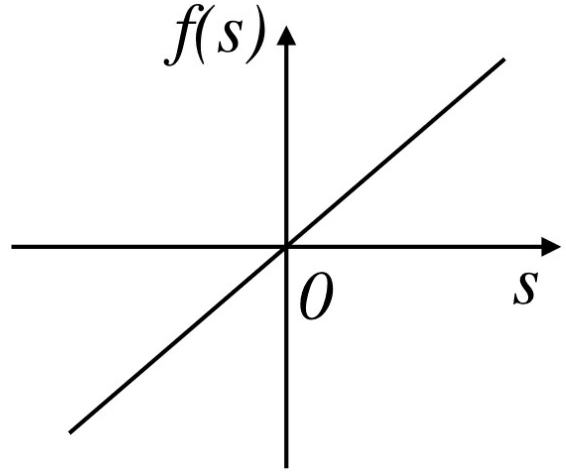
$=$

活性化関数を線形関数に



$$s = \sum_{i=1}^{N} w_i x_i - h$$

$$y = \boxed{f(s) = s}$$



誤差関数 $E = \frac{1}{2}(y - y_d)^2 = \frac{1}{2}(f(s) - y_d)^2$

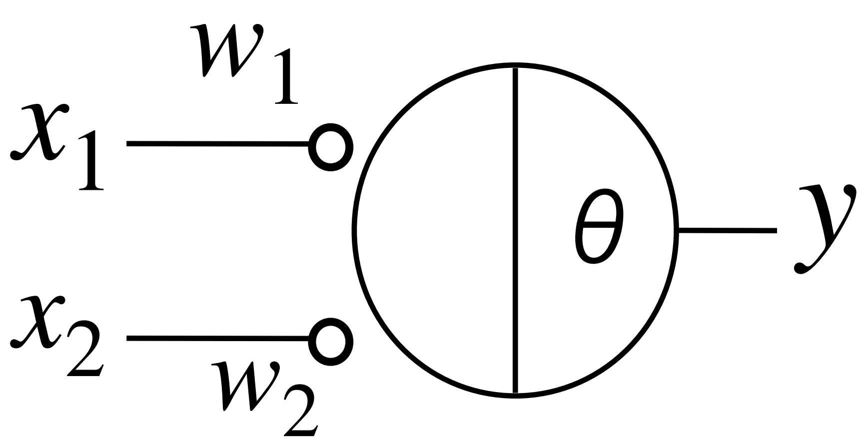
学習則 $\Delta w_i =$

$$\Delta w_i = -\epsilon \frac{\partial E}{\partial w_i} \qquad \Delta h = -\epsilon \frac{\partial E}{\partial h}$$

$$= -\epsilon \frac{dE}{dy}\frac{dy}{ds}\frac{\partial s}{\partial w_i} \qquad = -\epsilon \frac{dE}{dy}\frac{dy}{ds}\frac{\partial s}{\partial h}$$

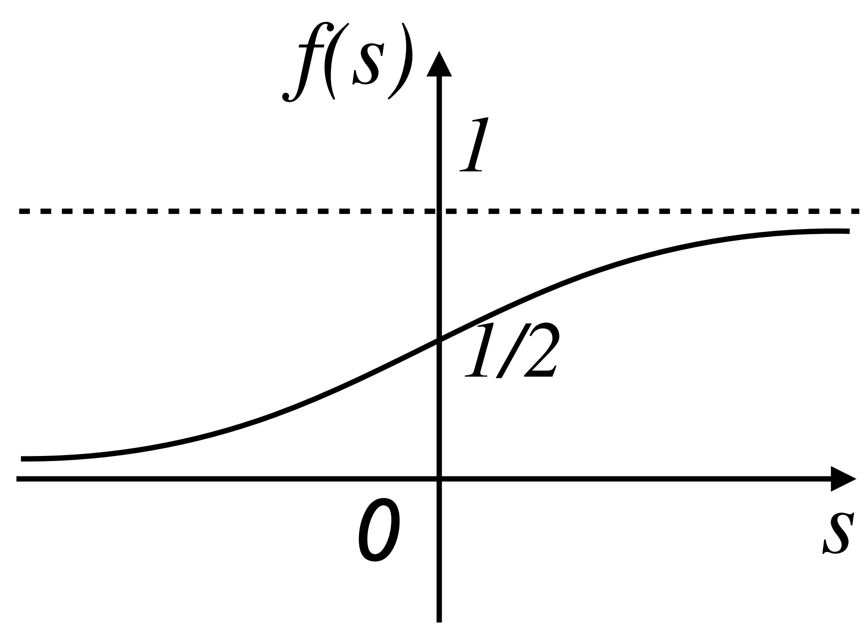
$$= -\epsilon (y - y_d) x_i \qquad = \epsilon (y - y_d)$$

活性化関数をシグモイド関数に

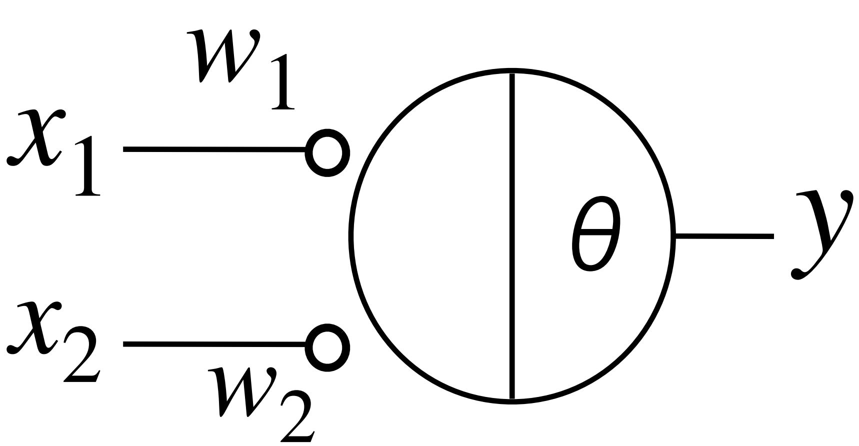


$$s = \sum_{i=1}^{N} w_i x_i - h$$

$$y = \boxed{}$$



活性化関数をシグモイド関数に



$$s = \sum_{i=1}^{N} w_i x_i - h$$

$$y =$$

誤差関数 $E =$

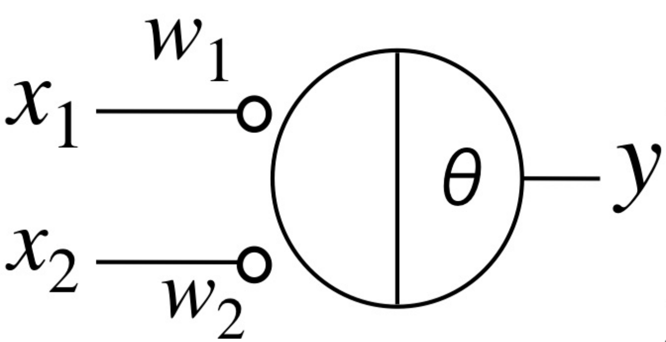
学習則 $\Delta w_i =$

$$=$$

$$=$$

$$\frac{df}{ds} =$$

活性化関数をシグモイド関数に



$$s = \sum_{i=1}^{N} w_i x_i - h$$

$$y = f(s) = \frac{1}{1+\exp(-s/T)} \quad (T: 定数)$$

誤差関数 $E = E = \frac{1}{2}(y - y_d)^2$

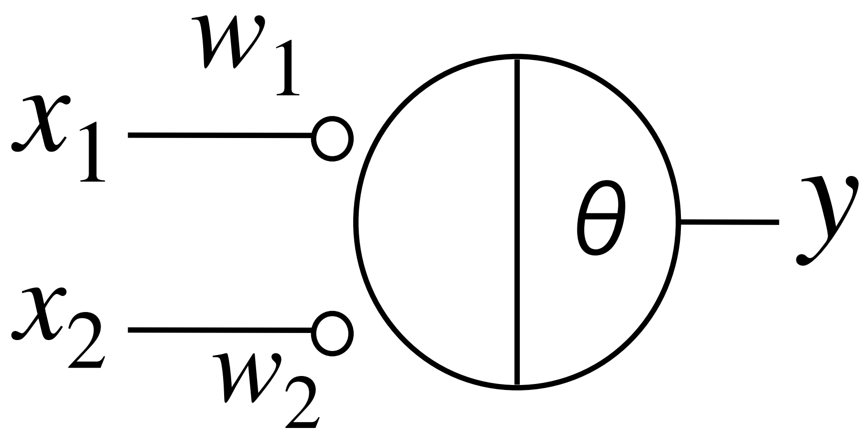
学習則

$$\Delta w_i = \Delta w_i = -\epsilon \frac{\partial E}{\partial w_i}$$

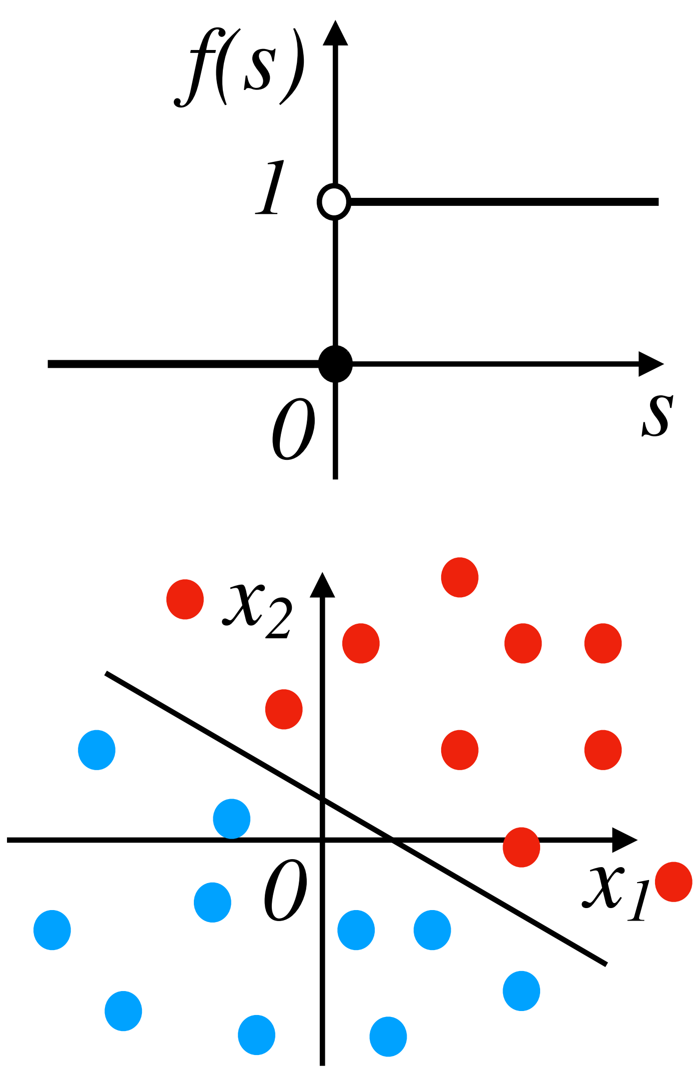
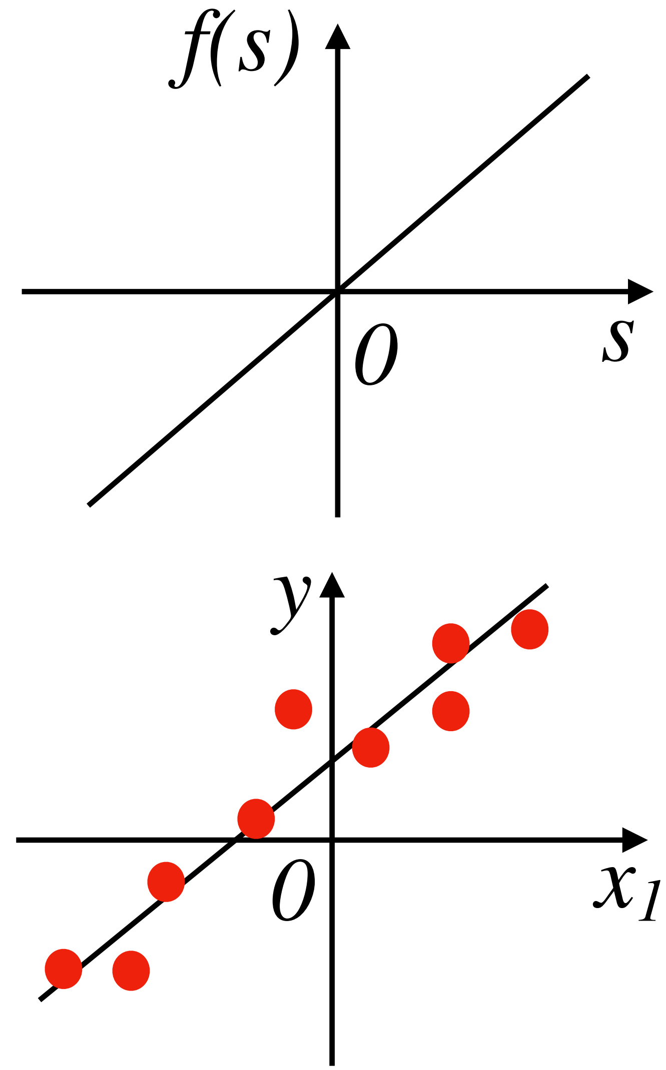
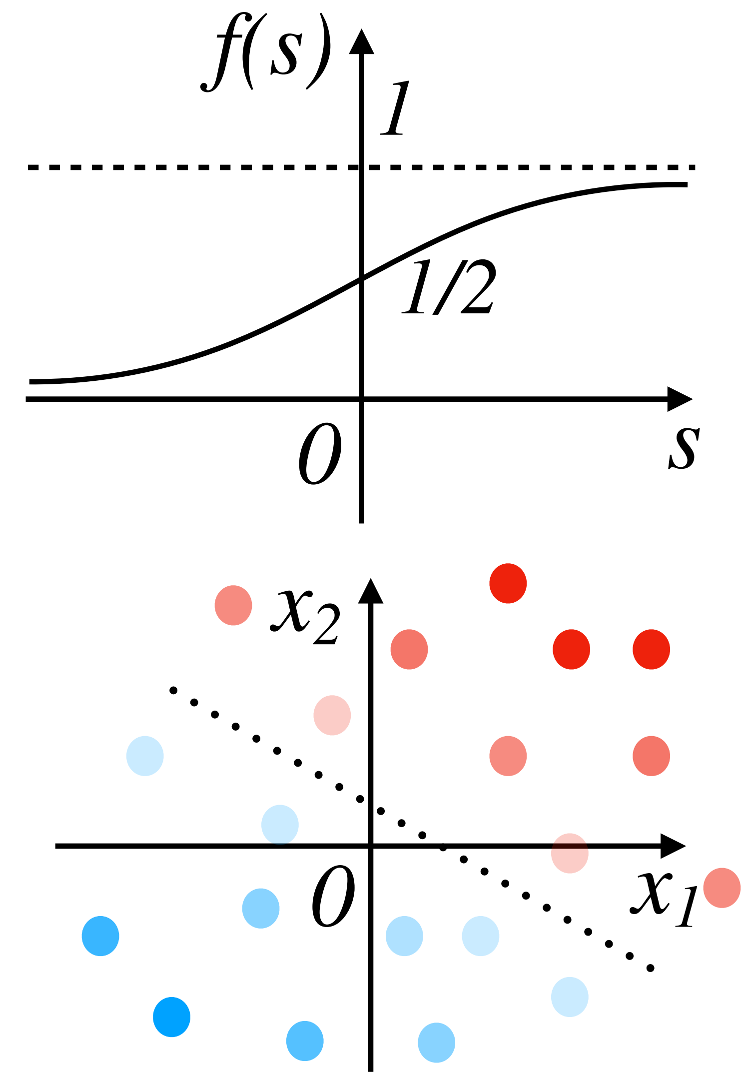
$$= -\epsilon \frac{dE}{dy}\frac{dy}{ds}\frac{\partial s}{\partial w_i}$$

$$= -\epsilon (y - y_d)\frac{dy}{ds} x_i$$

$$\frac{df}{ds} = \frac{1}{T}\frac{\exp(-\frac{s}{T})}{(1+\exp(-\frac{s}{T}))^2}$$



$$s = \sum_{i=1}^{2} w_i x_i - h \qquad y = f(s)$$

ステップ関数	恒等関数	シグモイド関数
		
二値分類	テキスト	