ソフトコンピューティング演習

第４回 ニューラルネットワーク II

講義用 HP：http://www.robot.t.u-tokyo.ac.jp/dcm/lec/softcomp.htm

1. 最急降下法および逆誤差伝播法

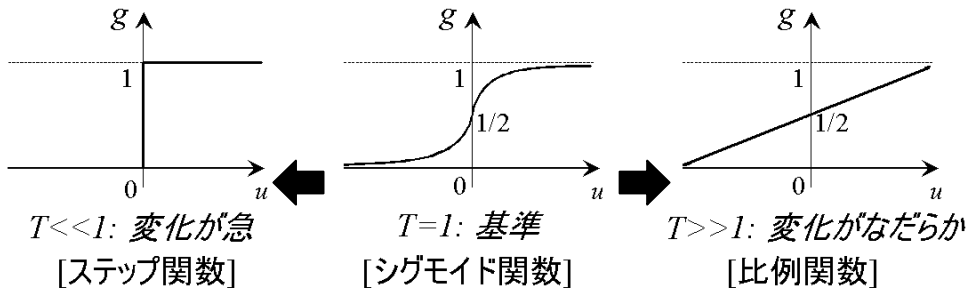
Hebb 学習則以外にも，最急降下法や逆誤差伝播法といった重み決定の学習法がある．主な違いとしては，Hebb 学習則では特性関数が｛0, 1｝の二値表現に対して，最急降下法や逆誤差伝播法では[0,1]の連続値表現が適用されている．また，最急降下法は単層パーセプトロンに適用され，逆誤差伝播法は多層ネットワークで適用されている．

ここでは，特性関数としてシグモイド関数 $g(u)=\frac{1}{1+e^{-\beta u}}$（$\beta$ は温度係数）を用い，最急降下法および逆誤差伝播法を学ぶ．なお，このような連続表現の特性関数において，出力される値は 0 もしくは 1 といった整数となることがないため，0 または 1 の近傍値を整数とみなす閾値処理が必要となる．（例：0.8 以上の出力値を 1 とする).

1.1 ニューロンの特性関数

各ニューロンは内部状態に依存した信号が出力される. このときニューロン内部状態 u から出力 g を決定する関数をニューロンの特性関数と呼ぶ．特性関数には，出力値が{0, 1}の 2 値となるステップ関数，出力値が[0, 1}の連続値となるシグモイド関数や比例関数などが挙げられる．

特性関数： $g(u)=\frac{1}{1+e^{-u/T}}$　　$\frac{dg(u)}{du}=\beta g(u)(1-g(u)),\quad \beta=\frac{1}{T}$



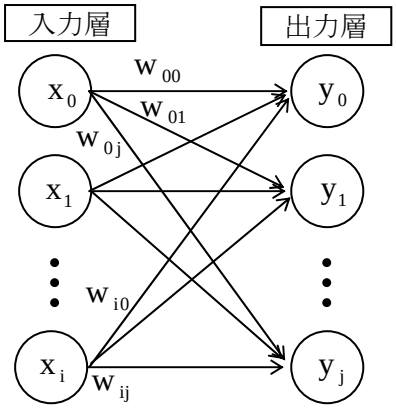
T<<1: 変化が急　*T=1: 基準*　*T>>1: 変化がなだらか*

[ステップ関数]　[シグモイド関数]　[比例関数]

ここで T は関数の勾配の緩急を決めるパラメータ（温度係数）であり，T が小さいほど勾配が急でステップ関数に近くなり, T が大きくなると変化がなだらかで比例関数に近くなる.

2008 年 12 月 19 日

1.2 最急降下法（Gradient Descent Method：GDM）



$$u_j = \sum_{i=0}^{M} w_{ij} x_i, \qquad y_j = g(u_j)$$

* バイアスニューロン：$x_0 = 1$
* 特性関数：$g(u) = \dfrac{1}{1+e^{-\beta u}}$
* 1学習セット：
  (入力) $x_1, x_2, \cdots, x_i, \cdots, x_M$
  (理想出力) $t_1, t_2, \cdots, t_j, \cdots, t_N$

最小二乗誤差：

$$E = \frac{1}{2}\sum_{j=0}^{N}[t_j - y_j]^2 = \frac{1}{2}\sum_{j=0}^{N}\left[t_j - g\left(\sum_{i=0}^{M} w_{ij}x_i\right)\right]^2 = \frac{1}{2}\sum_{j=0}^{N}[t_j - g(u_j)]^2$$

各ニューロンの最小二乗誤差の勾配：

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial u_j}\frac{\partial u_j}{\partial w_{ij}} = -(t_j - y_j)\frac{dg(u_j)}{du_j}x_i = -(t_j - y_j)\beta g(u_j)(1 - g(u_j))x_i$$

特性関数の微分：$\dfrac{dg(u)}{du} = \beta g(u)(1 - g(u))$

手順：

(1) 出力層の各ニューロン出力値 $y_1, y_2, \cdots, y_j, \cdots, y_N$ を求める.

(2) 最小二乗誤差：$E = \dfrac{1}{2}\sum_{j=0}^{N}[t_j - y_j]^2$ を求める.

(3) すべての重みに関する最小二乗誤差の勾配を求める.

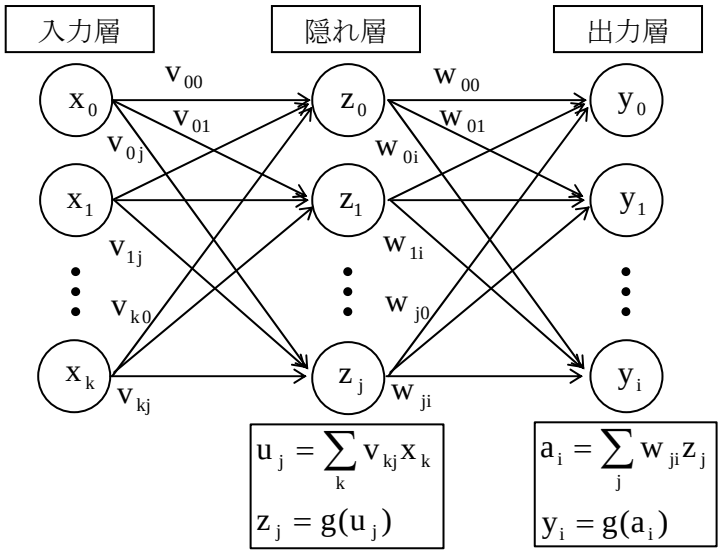
最小二乗誤差の勾配：$\dfrac{\partial E}{\partial w_{ij}} = -(t_j - y_j)\dfrac{dg(u_j)}{du_j}x_i$

(4) すべての重みの値を更新する．なお，n は学習ステップ，η は学習率.

重み更新：$w_{ij}(n+1) = w_{ij}(n) - \eta\dfrac{\partial E(n)}{\partial w_{ij}}$

(5) すべての学習セットにおいて，最小二乗誤差 E が充分に小さくなる（<<1）まで，手順(1)から(4)までを繰り返す

1.3 逆誤差伝播法(Back Propagation Method：BP)



出力値と理想出力との最小二乗誤差：$E = \frac{1}{2}\sum_i [t_i - g(a_i)]^2 = \frac{1}{2}\sum_i [t_i - g(\sum_j w_{ji} z_j)]^2$

出力層の各ニューロンの最小二乗誤差の勾配：

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial y_i} \cdot \frac{\partial y_i}{\partial a_i} \cdot \frac{\partial a_i}{\partial w_{ji}} = -(t_i - y_i)\frac{dg(a_i)}{da_i} z_j = \Delta_i z_j$$

隠れ層と出力層間の各重み更新：$w_{ji}(n+1) = w_{ji}(n) - \eta \frac{\partial E_1}{\partial w_{ji}} = w_{ji}(n) - \eta \Delta_i z_j$

出力値と理想出力との最小二乗誤差：

$$E = \frac{1}{2}\sum_i [t_i - g(\sum_j w_{ji} z_j)]^2 = \frac{1}{2}\sum_i [t_i - g(\sum_j w_{ji}(g(u_j)))]^2 = \frac{1}{2}\sum_i [t_i - g(\sum_j w_{ji}(g(\sum_k v_{kj} x_k)))]^2$$

隠れ層の各ニューロンの最小二乗誤差の勾配：

$$\frac{\partial E}{\partial v_{kj}} = \sum_i \frac{\partial E}{\partial y_i} \cdot \frac{\partial y_i}{\partial a_i} \cdot \frac{\partial a_i}{\partial z_j} \cdot \frac{\partial z_j}{\partial u_j} \frac{\partial u_j}{\partial v_{kj}} = \sum_i \Delta_i w_{ji} \frac{dg(u_j)}{du_j} x_k = \delta_j x_k$$

入力層と隠れ層の各重み更新：$v_{kj}(n+1) = v_{kj}(n) - \eta \frac{\partial E_2}{\partial v_{kj}} = v_{kj}(n) - \eta \delta_j x_k$

補足：$\Delta_i = -(t_i - y_i)\frac{dg(a_i)}{da_i}$，　$\delta_j = \sum_i \Delta_i w_{ji} \frac{dg(u_j)}{du_j}$

2008 年 12 月 19 日

* バイアスニューロン：$x_0 = 1, \quad z_0 = 1$

* 特性関数：$g(u) = \dfrac{1}{1+e^{-\beta u}}$，特性関数の微分：$\dfrac{dg(u)}{du} = \beta g(u)(1-g(u))$

* 1 学習セット：（入力）$x_1, x_2, \cdots, x_k, \cdots$，（理想出力）$t_1, t_2, \cdots, t_i, \cdots$

手順：

(1) 入力 $x_1, x_2, \cdots, x_k$ に対する隠れ層の各ニューロン出力値 $z_1, z_2, \cdots, z_j$ を求める．

(2) 隠れ層出力 $z_1, z_2, \cdots, z_j$ に対する出力層の各ニューロン出力値 $y_1, y_2, \cdots, y_i$ を求める．

(3) 出力層における各ニューロンの最小二乗誤差の勾配 $\partial E / \partial w_{ji}$ を求める．

(4) 隠れ層における各ニューロンの最小二乗誤差の勾配 $\partial E / \partial v_{kj}$ を求める．

(5) 隠れ層-出力層間の各重みを更新する．なお n は学習ステップ，η は学習率．

$$w_{ji}(n+1) = w_{ji}(n) - \eta \Delta_i z_j$$

(6) 入力層-隠れ層間の各重みを更新する．なお n は学習ステップ，η は学習率．

$$v_{kj}(n+1) = v_{kj}(n) - \eta \delta_j x_k$$

(7) (1)～(6)をすべての学習セット(p=1,2,...,P)において最小二乗誤差 E_p を計算し，その総和 ΣE_p が充分に小さくなる(<<1)までこの手順を繰り返す．

2008 年 12 月 19 日

2. 倒立振り子の制御

本章では，人工ニューラルネットワーク（Artificial Neural Network：ANN）を用いた2種類の倒立振り子の制御器を構築する．

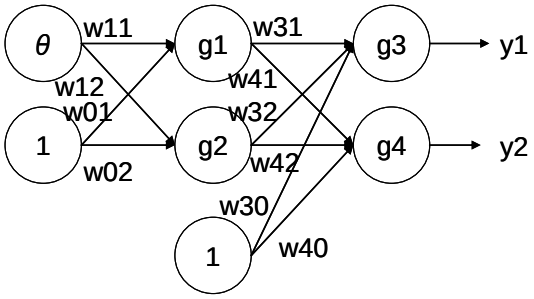
2.1 Behavior Net による制御則の決定（ANN による分類）

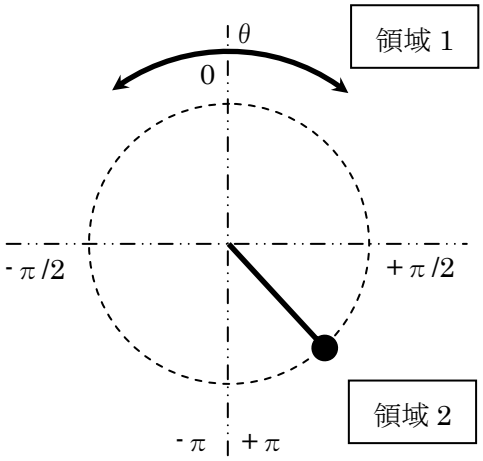
一つ目の制御器は，ANN による「分類」を適用する方法であり，演習第 1, 2 回における倒立振り子のファジー制御器で適用した if 文による条件分岐プログラムの代わりに，ANN（Behavior Net）を適用し制御則の分類を行うものである．

[課題 1] 演習第 2 回の課題 4b において作成した制御プログラムの条件分岐部を，以下に示す ANN に置き換え，下表の条件をみたすように BP を用いて ANN の重みを決定せよ．なお，BP のプログラム部は講義 HP よりダウンロードすること．

表 1　条件の対応

	範囲	出力（y1,y2）	ファジー制御則
領域 1	$-\pi/2 \leqq \theta < +\pi/2$	(1,0)	制御則 1
領域 2	$+\pi/2 \leqq \theta \leqq +\pi$, $-\pi \leqq \theta < -\pi/2$	(0,1)	制御則 2





隠れ層の各ニューロン出力値

$$g1 = g_1(\theta * w_{11} + 1 * w_{01})$$
$$g2 = g_2(\theta * w_{12} + 1 * w_{02})$$

出力層の各ニューロン出力値

$$y1 = g_3(g1 * w_{31} + g2 * w_{32} + 1 * w_{30})$$
$$y2 = g_4(g1 * w_{41} + g2 * w_{42} + 1 * w_{40})$$

特性関数：$g_{1,2,3,4}(u) = \dfrac{1}{1+e^{-u}}$

重み：w_{ij},　　理想出力：t_1, t_2

※学習データの例：学習データセット(θ, t_1, t_2)

(**0**,1,0) (**π/8**,1,0) (**2π/8**,1,0) (**3π/8**,1,0) (**3.14/2**,1,0) (**-π/8**,1,0) (**-π/4**,1,0)
(**-3π/8**,1,0)　(**-3.14/2**,1,0)　(**1.6**,0,1) (**5π/8**,0,1) (**6π/8**,0,1) (**7π/8**,0,1)
(**3.14**,0,1) (**-1.6**,0,1) (**-5π/8**,0,1) (**-6π/8**,0,1) (**-7π/8**,0,1) (**-3.14**, 0, 1)

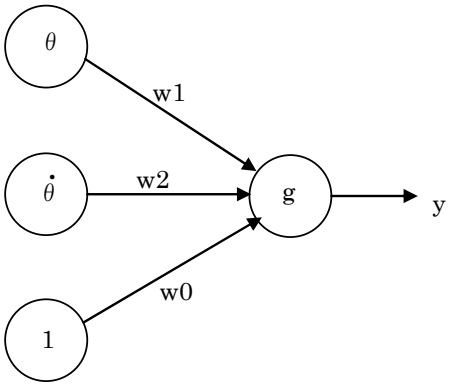
2008 年 12 月 19 日

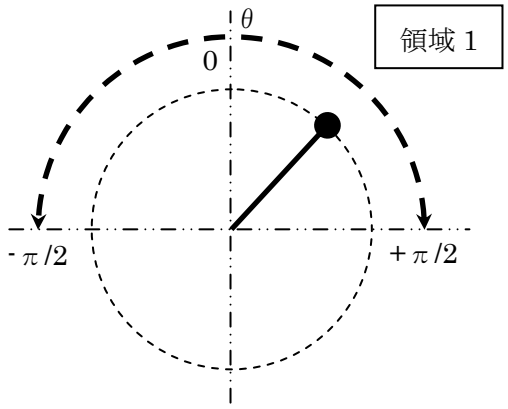
2.2 制御器としての ANN（制御パラメータのランダム探索）

前節では，分類（条件分岐）などの ANN を制御器の一部として利用する方法を紹介したが， ANN 自体を制御器とする方法もある．すなわち，倒立振り子の角度・角速度を ANN に入力することで，台車に加える力 f を決定する ANN 制御器が実現できる．

[課題 2] 下図に示される ANN の重み w0,w1,w2 をランダム探索し，θ が 0 に収束する ANN 制御器を完成させよ．なお，倒立振り子の初期角度は領域 1($-\pi/2 < \theta < +\pi/2$)にあるものとする．

(例)k=1000 に固定し，w0,w1,w2 を[-1.0,1.0]の乱数で決定し，N ステップ経過後，振れ角が最も小さくなる w0,w1,w2 を探索するなど．





ANN：$y = g(\theta * w_1 + \dot{\theta} * w_2 + w_0)$
特性関数：$g(u) = u$
制御器：$f = k \cdot y$

θ：振り子の角度
$\dot{\theta}$：振り子の角速度
k：ゲイン
f：台車に加える力

****

■レポート課題 2

出来うる限り短時間で収束すると思われる倒立振り子の ANN 制御器を BP（本演習課題 1）もしくはランダム探索（本演習課題 2）を用いて作成し考察せよ．レポートには，ANN の式および図，プログラムも記述すること．

提出方法：横井教官宛（soft_computing@robot.t.u-tokyo.ac.jp）にメールで送付
提出ファイル名：kadai2_xxxxxx.doc，または kadai2_xxxxxx.pdf ※xxxxxx は学籍番号

2008 年 12 月 19 日