

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

環境の変化に適応できる 複製・競合動径基底関数ネットワーク

**Reproductive and Competitive Radial Basis Function Networks
Adapting to the Dynamical Environments**

June 28, 2024

1 本研究の背景

背景

競合動径基底関数ネットワーク (Competitive Radial Basis Function Network: 以下 CRBFN) は競合により冗長なニューロンを消滅させることができあり、その結果学習の高速化と過学習の回避が行われる。しかし、CRBFN には新しい動径基底関数を追加する能力がないため教師信号が変化するような環境の変化には対応しきれていなかった。

- 動径基底関数 (Radial Basis Function: 以下 RBF)
- 環境の変化
 - 入出力の写像を与える関数そのものが変化する場合
 - 既に観測され学習に用いられていた入出力の組が不要となり取り除かれる変化
 - 新たに観測された入出力の組が学習で用いられる変化

2 動径基底関数ネットワークの概要

3/38

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

動径基底関数ネットワーク (Radial Basis Function Network: 以下 RBFN)

非線形関数 $\eta(x)$ を動径基底関数 $\xi^i(x)$ の足し合わせで近似するニューラルネットワークである。 d 次元の第 i 入力ベクトル $X_i \in R^d (i = 1, 2, \dots, N)$ は全ての入力ニューロンに入力される。第 j 入力ニューロン ($j = 1, 2, \dots, M$) はパラメータ ϕ_j をもつ

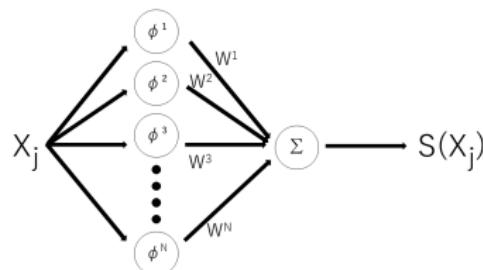


図 1

- パラメータ $\phi_j \equiv m_j, \sum_j$ 平均ベクトルと共に分散行列の集合
- $m_j \equiv [m_1^j, m_2^j, \dots, m_d^j]^T$
- \sum 正定値対称行列

2 動径基底関数ネットワークの概要

第j入力ニューロンは入力ベクトル x_i に対して(1)を出力する。以後、(1)のような出力を行う入力ニューロンの事を動径基底関数という。

$$\xi^i(x_i, \phi_j) = \exp\left\{-\frac{1}{2}(x_i - m_j)^T \sum_{j=1}^{-1} (X_i - m_j)\right\} \quad (1)$$

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

■ T ：ベクトルの転置

出力値 ($\xi^i(x_i, \phi_j)$) はシナプス荷重結合 (w_j) を通して出力ニューロンへ伝達され、出力ニューロンでこれらは足し合わされ(2)が出力される。

$$S(x_i, w, \phi) = \sum_{j=1}^M w_j \xi(x_i, \phi_j) \quad (2)$$

- $w = w_1, w_2, \dots, w_M \in R^M$
- $\phi = \text{集合 } \{\phi_1, \phi_2, \dots, \phi_M\}$

2 動径基底関数ネットワークの概要

ニューラルネットワークによる関数近似は、非線形関数 $\eta(x)$ をネットワークの出力 $s(x, w, \phi)$ で表すことである。そのため、RBFN による関数近似は累積 2 乗誤差関数 (3) の値を減少させることにより実現される。

$$E(w, \phi) = \frac{1}{2} \sum_{i=1}^N (x_i, w, \phi) \quad (3)$$

(4) は 2 乗誤差関数である。

$$E(x_i, w, \phi) = \eta(x_i) - s(x_i, w, \phi)^2 \quad (4)$$

以上の事より、RBFN が学習により獲得しなければならないのは

- 第 j 動径基底関数のシナプス結合荷重 w_j
- パラメータ m_j
- パラメータ \sum_j

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

2 動径基底関数ネットワークの概要

6/38

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

従来の RBFN と CRBFN の学習アルゴリズムの相違は、一般の RBFN の学習アルゴリズムは(3)の累積 2 乗誤差関数に Delta ルールを適用することにより、得られる。

$$\Delta w_j = -\epsilon \frac{\partial E(w, \phi)}{\partial w_j} \quad (5)$$

$$\Delta m_j^k = -\epsilon \frac{\partial E(w, \phi)}{\partial m_j^k} \quad (6)$$

$$\Delta \sigma_j^{kl} = -\epsilon \frac{\partial E(w, \phi)}{\partial \sigma_j^{kl}} \quad (7)$$

- ϵ ：適当な正の定数
- m_j^k ：パラメータ m_j の第 k 要素
- $\Delta w_j \equiv \frac{dw_j}{dt}$
- $\Delta m_j^k \equiv \frac{dm_j^k}{dt}$
- $\Delta \sigma_j^{kl} \equiv \frac{d\sigma_j^{kl}}{dt}$

2 動径基底関数ネットワークの概要

7/38

Delta ルール

出力の誤差を最小化するようにニューラルネットワークの重みを調整することである。ニューラルネットワークが output する値と正解データとの間の誤差をもとに重みを更新する。

$$\Delta w_{ij} = \eta(t_j - y_j)x_i \quad (8)$$

- Δw_{ij} : 重み w_{ij} の更新量
- η : 学習率, 重みの更新ステップの大きさを決定する
- t_j : 教師信号 (目標出力)
- y_j : ニューラルネットワークによる現在の出力
- x_i : 入力ノード i の入力値

このルールにより、ニューラルネットワークは期待される出力と実際の出力の誤差を小さくする方向に重みを調整する。

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

2 競合動径基底関数ネットワークの概要

8/38

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

CRBFN では、パラメータ m_j とパラメータ \sum_j の学習アルゴリズムは従来の RBFN と同じであり、(6)(7) により与えられる。しかし、シナプス結合荷重 w_j に対しては Dale 則を考慮したシナプス可塑性方程式である適者生存型学習則が適用される。

$$\Delta w_j = \epsilon(a_j(\phi) - \sum_{k=1}^M \gamma_{jk}(\phi) \mu_k w_k) w_j \quad (9)$$

- w_j : 第 j ニューロンが興奮性なら正、抑制性なら負をとる。
- μ_k : 第 k ニューロンが興奮性 ($w_k > 0$) であるなら 1、抑制性 ($w_k < 0$) であるなら -1
- $a_j(\phi)$: 内的自然増加率
- $\gamma_{jk}(\phi)$: 競争係数（第 j ニューロンと第 k ニューロンとの競合の効果）

2 競合動径基底関数ネットワークの概要

9/38

内的自然増加率

$$a_j(\phi) = \sum_{i=1}^N (x_i) \xi(x_i, \phi_j) \quad (10)$$

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

競争係数

$$\gamma_{jk}(\phi) = \sum_{i=1}^N \xi(x_i, \phi_j) \xi(x_i, \phi_k) \quad (11)$$

で定義される.CRBFN の出力は

$$s(x_i, w, \phi) = \sum_{j=1}^M \mu_j w_j \xi(x_i, \phi_j) \quad (12)$$

で得られることとする。

2 競合動径基底関数ネットワークの概要

- 1はじめに
- 2競合動径基底関数ネットワークの概要
- 3パラメータが従う確率密度関数の導出
- 4動径基底関数の複製アルゴリズム
- 5シミュレーションならびに考察

学習中に $w^i \approx 0$ となった第 j シナプス結合荷重は消滅したものとして、生き残っているシナプス結合荷重とそれらにより伝達される入力が学習を続ける。累積 2 乗誤差関数 $E(w, \phi)$ は (9) に対する Lyapunov 関数であることが (13) により示される

$$\begin{aligned}
 \frac{dE(w, \phi)}{dt} &= \sum_{j=1}^M \frac{\partial E(w, \phi)}{\partial w_j} \frac{dw_j}{dt} \\
 &= - \sum_{j=1}^M \mu^j \left\{ \sum_{i=1}^N \eta(x_i) \xi(x_i, \phi_j) \right. \\
 &\quad \left. - \sum_{h=1}^M \mu_h w_h \sum_{i=1}^N \xi(x_i, \phi_j) \xi(x_i, \phi_h) \right\} \frac{dw_j}{dt} \\
 &= -\epsilon \sum_{j=1}^M \mu_j w_j (a_j(\phi) - \sum_{h=1}^M \gamma_{jh}(\phi) \mu_h w_h)^2 \\
 &\leq 0
 \end{aligned} \tag{13}$$

3 シナプス可塑性方程式の導出

11/38

Lyapunov 関数

ある平衡点の安定性を証明できる可能性のある関数

“ 定理

U を \bar{x} の近傍とする。

$V : U \rightarrow \mathbb{R}$ を次の条件を満たす C^1 級関数とする。

$$V(\bar{x}) = 0$$

$x \neq \bar{x}$ ならば、 $V(x) > 0$ (この2つの条件を、 V は正定値 positive definite であるという)

$$x \neq \bar{x} \text{ ならば、} \frac{dV}{dt}(x) \leq 0$$

このとき、 \bar{x} は (リヤプノフの意味で) 安定。

さらに、次の条件を満たすならば、 \bar{x} は漸近安定。

$$x \neq \bar{x} \text{ ならば、} \frac{dV}{dt}(x) < 0$$

図 2: 参考 : Wiggins Introduction to Applied Nonlinear Dynamical Systems and Chaos

3 確率密度関数の導出

下記のものを導出する.

- CRBFN の平均ベクトル
- 共分散行列
- シナプス結合荷重が学習終了時にとる同時確率密度 $(p(w, \phi))$

シナプス結合荷重 w_j の変数変換を行う.

$$y_j^2 = \mu_j w_j \quad (14)$$

y_j の定義域は任意の実数. この時式 (9) は、式 (15) となる.

$$\Delta w_j = \epsilon(a_j(\phi) - \sum_{k=1}^M \gamma_{jk}(\phi) \mu_k w_k) w_j \quad (9)$$

$$\frac{dy_j}{dt} = (\frac{a_j(\phi)}{2} - \sum_{k=1}^M \frac{\gamma_{jk}(\phi)}{2} \mu_k w_k^2) y_j \quad (15)$$

ϵ は省略した

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

3 確率密度関数の導出

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

$$\frac{dy_j}{dt} = \left(\frac{a_j(\phi)}{2} - \sum_{k=1}^M \frac{\gamma_{jk}(\phi)}{2} w_k^2 \right) y_j \quad (15)$$

式 (15) は積分条件

$$\frac{\partial}{\partial y_k} \frac{dy_j}{dt} = \frac{\partial}{\partial y_j} \frac{dy_k}{dt} \quad (16)$$

を満たすため、ポテンシャルを考えることができ、

$$V^*(y, \phi) = - \sum_{j=1}^M \int_a^{y_j} \left(\frac{a_j(\phi)}{2} - \sum_{k=1}^M \frac{\gamma_{jk}(\phi)}{2} y_k'^2 \right) y_j' dy_j' \quad (17)$$

変数 y_j の時間変化はポテンシャル $V^*(y, \phi)$ 式 (18) で導くことが出来る。

$$\frac{dy_j}{dt} = \frac{\partial V^*(y, \phi)}{\partial y_j} \quad (18)$$

3 確率密度関数の導出

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

$$y_j^2 = \mu_j w_j \quad (14)$$

式(14)より、ポテンシャル $V^c(y, \phi)$ は

$$V(w, \phi) = - \sum_{j=1}^M \left(\frac{a_j(\phi)}{4} \mu_j w_j - \sum_{k \neq j}^M \frac{\gamma_{jk}(\phi)}{4} \mu_j w_j \mu_k w_k - \frac{\gamma_{jj}(\phi)}{8} w_j^2 \right) \quad (19)$$

と書き直すことができる。その結果

$$E(w, \phi) = 4V(w, \phi) + \frac{1}{2} \sum_{i=1}^N \eta^2(x_i) \quad (20)$$

であることが示される。累積2乗誤差関数 $E(w, \phi)$ の最小化はポテンシャル $V(w, \phi)$ の最小化と等価であることが分かる。

3 確率密度関数の導出

今、式(15)に従う y_j はポテンシャル $V(w, \phi)$ の最急降下方向に更新される。

$$\frac{dy_j}{dt} = \left(\frac{a_j(\phi)}{2} - \sum_{k=1}^M \frac{\gamma_{jk}(\phi)}{2} w_k^2 \right) y_j \quad (15)$$

1 はじめに
2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出
4 動径基底関数の複製アルゴリズム
5 シミュレーションならびに考察

その結果、極小解に収束すると、そこから逃げられなくなる。極小解から脱出させるための手法として $y_j(21)$ の更新則をノイズを考慮した離散近似した見本過程で与えることが考えられる。

$$y_j(t + \Delta t) = u_j(t) - \frac{\partial V(y, \phi)}{\partial y_j} \Delta t + \sqrt{Q \Delta t} n_j(t) \quad (21)$$

- $n_j(t)$ ：独立な確率変数（平均 0、分散 1 の正規分布 $N(0, 1)$ ）に従う
- Q ：任意の正の定数

3 確率密度関数の導出

学習終了時に CRBFN の平均ベクトル、共分散行列とシナプス結合荷重が満たす同時確率密度 $p_\beta(w, \phi)$ は

$$p_\beta(w, \phi) = Z_\beta^{-1} \exp \beta V(w, \phi) \quad (22)$$

- $\beta : \frac{2}{Q}$
- Z_β : 分配関数

$$Z_\beta = \int_w \int_\phi \exp -\beta V(w, \phi) dw d\phi \quad (23)$$

式 (22) はポテンシャル $V(w, \phi)$ と累積 2 乗誤差関数 $E(w, \phi)$ の式 (20) より式 (24) と書き直せる。

$$p_\beta(w, \phi) = Z_\beta^{-1} \exp \beta^* E(w, \phi) \quad (24)$$

- $\beta^* : (2Q)^{-1}$
- Z_β^* : 分配関数

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

3 確率密度関数の導出

パラメータが従う確率密度関数が導出できたことにより、与えられた条件のもとで累積2乗誤差関数を最小とするパラメータの値が検出できることを示す。ここでは教師信号 $\eta(x)$ を式(25)で与えることとする。

$$\eta(x) = 3N(-1.5, 1) + 2N(1, 0.5) \quad (25)$$

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

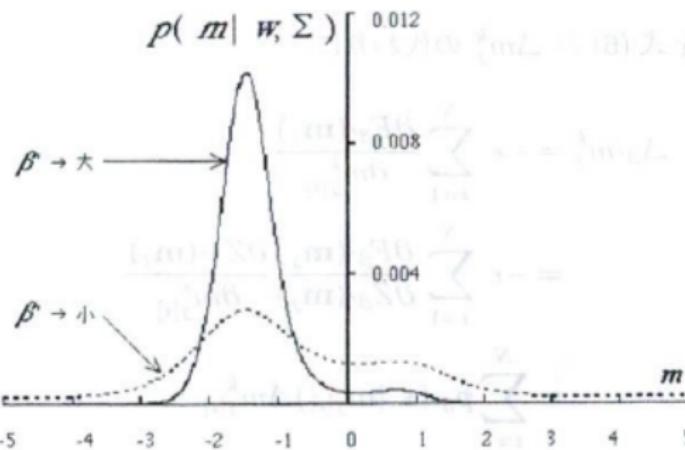
4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

- $N(m, \Sigma)$ ：平均 m , 分散 Σ の正規密度関数
- シナプス結合荷重 $w = 1$, パラメータ $\Sigma = 0.2$ だけ用いて近似することを考える

3 確率密度関数の導出

この場合、近似しようとしている非線形関数 $\eta(x)$ の複雑さに対し、必要とされている動径基底関数が十分に存在していないため、累積 2 乗誤差関数を 0 にすること自体が不可能である。しかし、個の動径基底関数のパラメータ m が従う条件付確率密度関数は導出可能であり、図 3 のようになる。



パラメータ m の条件付き確率密度関数

図 3

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

4.1 自由エネルギーの導出

1はじめに

2競合動径基底関数ネットワークの概要

3パラメータが従う確率密度関数の導出

4動径基底関数の複製アルゴリズム

5シミュレーションならびに考察

累積 2 乗誤差関数の値がある正数 $\epsilon > 0$ より大きな値に収束し学習が収束されたと判断されるときに、新たに必要な動径基底関数を追加する手法を提案する。学習が収束した時点では得られている動径基底関数の 1 部の形質（シナプス結合荷重 w_j , パラメータ \sum_j ）が新たに追加される動径基底関数が持つパラメータに引き継がれている。そのため、効率的に最も累積 2 乗誤差関数を小さくするパラメータ m に動径基底関数を追加していく事が可能である。なおかつ、最悪の場合にはすべての入力ベクトル x_i をパラメータ m_i とする動径基底関数を作成することができる。そこで、この手法を CRBFN に組み入れたニューラルネットワークを「複合・競合動径基底関数ネットワーク」と呼ぶこととする

4.1 自由エネルギーの導出

学習則の時点では得られている第 j 番目の動径基底関数に着目すると、入力ベクトル x_i の条件付確率密度関数は式 (26) で導出できる

$$\begin{aligned}
 p_{\beta^*}(x_i | m_{j[i]}, \phi^*_j, \phi^{**}_j) \\
 = Z_{\beta}^{-1}(m_j, \phi^*, \phi^{**}_j) \\
 \times \exp \beta^* E(x_i, m_{j[i]}, \phi^*_j, \phi^{**}_j)
 \end{aligned} \tag{26}$$

- パラメータ ϕ^*_j ：着目した第 j 番目の動径基底関数のシナプス結合荷重 w_j と共に分散行列 \sum_j の集合
- ϕ^{**}_j ：着目した第 j 番目の動径基底関数以外のシナプス結合荷重、共分散行列、平均ベクトルの集合

分配関数は

$$Z_{\beta^*}(m_j) = \sum_{i=1}^N \exp -\beta^* E(x_i, m_{j[i]}) \tag{27}$$

4.1 自由エネルギーの導出

条件付き確率密度関数 $p_{\beta^*}(x_i|m_{j[i]})$ は、確率の正規化と 2 乗誤差関数 $E(x_i, m_{j[i]})$ の条件付き期待値

$$\begin{aligned} \langle E(m_j) \rangle_{\beta^*} &= \sum_{i=1}^N p_{\beta^*}(x_i|m_{j[i]}) \\ &\quad \times E(x_i, m_{j[i]}) \end{aligned} \tag{28}$$

が一定となるという 2 つの制約のもとで、エントロピー

$$\begin{aligned} S_{\beta^*}(m_j) &= -\frac{1}{\beta^*} \sum_{i=1}^N p_{beta^*}(x_i|m_{j[i]}) \\ &\quad \times \log p_{\beta^*}(x_i|m_{j[i]}) \end{aligned} \tag{29}$$

を最大にする確率密度関数として導出できる

- 記号 $\langle \dots \rangle_{\beta^*} : p_{\beta^*}(x_i|m_{j[i]})$ を掛けて x_i に関する和をとる演算を表すもの

4.1 自由エネルギーの導出

自由エネルギーを式 (30) で定義すると

$$F_\beta(m_j) = - \sum_1^{\beta} \log Z_\beta(m_j) \quad (30)$$

$$S_\beta(m_j) = -F_\beta(m_j) + \beta < E(m_j) >_\beta \quad (31)$$

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

と表すことができる。式 (31) はエントロピー $S_\beta(m_j)$ を最大化する条件付確率密度関数 $p_\beta(x_i|m_{j[i]})$ は、自由エネルギー $F_\beta(m_j)$ を最小化するものであることを示している。

4.2 複製する位置の決定法

複合・競合動径基底関数ネットワーク（Reproductive CRBFN 以下:RC-RBFN）ではパラメータ m_j^k の更新則を

$$\begin{aligned}
 \Delta_{\beta'} m_j^k &= -\epsilon \sum_{i=1}^N \frac{\partial F_{\beta'}(m_j)}{\partial m_{j[i]}^k} \\
 &= -\epsilon \sum_{i=1}^N \frac{\partial F_{\beta'}(m_j)}{\partial m_j} \frac{\partial Z_{\beta'}(m_j)}{\partial m_{j[i]}^k} \\
 &= \sum_{i=1}^N p_{\beta'}(x_i | m_{j[i]}) \Delta m_{j[i]}^k \\
 &= \langle \Delta m_j^k \rangle_{\beta'}
 \end{aligned} \tag{32}$$

で与えるとする。ここで

$$\Delta m_{j[i]}^k = -\epsilon \frac{\partial E(x_i, m_{j[i]})}{\partial m_{j[i]}^k} \tag{33}$$

である。特に $\beta' = 0$ であり、初期の状態が $m_{j[i]} = m_j (\forall i)$ である場合は

$$\Delta_0 m_j^k = \Delta m_j^k \tag{34}$$

4.2 複製する位置の決定法

RC-RBFN のパラメータ m_j^k の更新則が従来の RBFN のパラメータ m_j^k 更新則そのものとなっていることがわかる。このとき, $\beta^i = 0$ で固定したままパラメータを $\sum_j \rightarrow 0$ にすると $\Delta_0 m_j^k = 0$ とするパラメータ $m_{j[i]}$ は

$$\sum_{i=1}^N \xi(x_i, m_{j[i]})(x_i - m_{j[i]}) \times \eta(x_i) - s(x_i, m_{j[i]}) = 0 \quad (35)$$

を満たし, $m_{j[i]} = x_j (\forall i)$ であることが分かる。つまり, 教師入力信号がパラメータ m_j の収束点として検出されることとなる。

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

4.2 複製する位置の決定法

- 1 はじめに
- 2 競合動径基底関数ネットワークの概要
- 3 パラメータが従う確率密度関数の導出
- 4 動径基底関数の複製アルゴリズム
- 5 シミュレーションならびに考察

x_i	-4.00	-3.500	-1.00	0.50	3.00
$\eta(x_i)$	0.13	1.82	2.68	1.96	0.04

図 4

得られた結果を確認するため、

$$\eta(x) = 3N(-1.5, 1) + 2N(1, 0.5) \quad (25)$$

から適当に 5 点選んだ (図 5).

4.2 複製する位置の決定法

- 1 はじめに
- 2 競合動径基底関数ネットワークの概要
- 3 パラメータが従う確率密度関数の導出
- 4 動径基底関数の複製アルゴリズム
- 5 シミュレーションならびに考察

- β' を 0
- パラメータの \sum を徐々に小さくしたとき
 - $\Delta_0 m[i] = 0$ となる点をプロットした

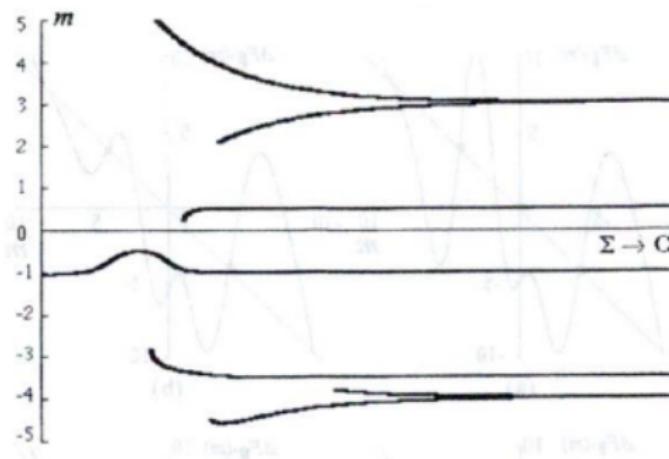


図 5

パラメータ $\sum \rightarrow 0$ において $m[i] = x_i (i = 1, 2, \dots, 5)$ に収束していることがわかる。

4.2 複製する位置の決定法

逆にパラメータ \sum_j を固定したまま $\beta^* \rightarrow \infty$ とすると, $\Delta_\infty m_j^k = 0$ とする
パラメータ $m_{j[i]}$ は

$$\begin{aligned} & \sum_{i=1}^N \exp -\beta^* E(x_i, m_{j[i]}) \xi(x_i, m_{j[i]}) \\ & \quad \times (x_i - m_{j[i]}) \eta(x_i) - s(x_i, m_{j[i]}) = 0 \end{aligned} \tag{36}$$

を満たし, $x_i (\forall i)$ を含む任意の値となることがわかる.

RC-RBFN のパラメータ m_j^k の更新則 $\Delta_\beta m_j^k$

$\Delta_0 m_j^k$ で従来の RBFN のパラメータ m_j^k の更新則を実現し, $\sum_j \rightarrow 0$ とすれば全ての入力ベクトル x_i の第 k 要素 $x_i^k (\forall i)$ をパラメータ m_j^k の安定な収束点として検出できることがわかる. あるいは, $\Delta_\infty m_j^k$ とすることで, 全ての入力ベクトル x_i の第 k 要素 $x_i^k (\forall i)$ を含む任意の値を安定な収束点とすことができる.

4.2 複製する位置の決定法

ここで、提案手法と「ゆう度解析」との関係について示す。まず、下記のような自由エネルギーを考える

$$F_{\beta}^{\circ} = -\frac{1}{\beta^{\circ}} \log Z_{\beta}^{\circ} \quad (37)$$

分配関数は

$$Z_{\beta}^{\circ} = \sum_{i=1}^N \sum_{j=1}^M \exp -\beta^{\circ} E(x_i, m_{j[i]}) \quad (38)$$

で与えられる定数である。このとき式(31)は下記のように変形することができる。

$$\begin{aligned} F_{\beta}^{\circ}(m_j) - F_{\beta}^{\circ} &= -\frac{1}{\beta^{\circ}} \log \frac{Z_{\beta}^{\circ}(m_j)}{Z_{\beta}^{\circ}} \\ &= -\frac{1}{\beta^{\circ}} \log \sum_{i=1}^N p_{beta}^{\circ}(x_i, m_{j[i]}) \end{aligned} \quad (39)$$

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

4.2 複製する位置の決定法

ここで、確率密度関数 $p_{beta}(x_i, m_{j[i]})$ は

$$p_{beta}(x_i, m_{j[i]}) = Z_\beta^{-1} \exp -\beta E(x_i, m_{j[i]}) \quad (40)$$

つまり、自由エネルギー $F_\beta(m_j)$ のパラメータ m_j^k に関する最小化は、対数ゆう度関数に関する最大化に等価であることを示すことができる。

- 1 はじめに
- 2 競合動径基底関数ネットワークの概要
- 3 パラメータが従う確率密度関数の導出
- 4 動径基底関数の複製アルゴリズム
- 5 シミュレーションならびに考察

4.2 RC-RBFN の学習則の提案

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

- STEP1 シナプラス結合荷重 w_i を式(9)のシナプラス可塑性方程式により更新、パラメータ m_j^k を式(33)の $\Delta_0 m_j^k$ により更新、パラメータ \sum_j は式(7)により更新

$$\Delta w_j = \epsilon(a_j(\phi) - \sum_{k=1}^M \gamma_{jk}(\phi) \mu_k w_k) w_j \quad (9)$$

$$\begin{aligned} \Delta_\beta m_j^k &= -\epsilon \sum_{i=1}^N \frac{\partial F_\beta'(m_j)}{\partial m_{j[i]}^k} \\ &= -\epsilon \sum_{i=1}^N \frac{\partial F_\beta'(m_j)}{\partial m_j} \frac{\partial Z_\beta'(m_j)}{\partial m_{j[i]}^k} \\ &= \sum_{i=1}^N p_\beta'(x_i | m_{j[i]}) \Delta m_{j[i]}^k \\ &= \langle \Delta m_j^k \rangle_\beta' \end{aligned}$$

$$\Delta \sigma_j^{kl} = -\epsilon \frac{\partial E(w, \phi)}{\partial \sigma_j^{kl}} \quad (7)$$

4.2 RC-RBFN の学習則の提案

1 はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

- STEP2 累積 2 乗誤差関数が $E(w, \phi) \approx 0$ となったら学習終了. ある正数 $\epsilon > 0$ より大きな値に収束したなら STEP3 に進む
- STEP3 学習収束時に得られているすべての動径基底関数について, β^* を 0 から徐々に大きくしていきながら式 (33) に従いパラメータ m_j^k を更新する

$$\begin{aligned}
 \Delta_{\beta^*} m_j^k &= -\epsilon \sum_{i=1}^N \frac{\partial F_{\beta^*}(m_j)}{\partial m_{j[i]}^k} \\
 &= -\epsilon \sum_{i=1}^N \frac{\partial F_{\beta^*}(m_j)}{\partial m_j} \frac{\partial Z_{\beta^*}(m_j)}{\partial m_{j[i]}^k} \\
 &= \sum_{i=1}^N p_{\beta^*}(x_i | m_{j[i]}) \Delta m_{j[i]}^k \\
 &= \langle \Delta m_j^k \rangle_{\beta^*}
 \end{aligned}$$

4.2 RC-RBFN の学習則の提案

- 1 はじめに
- 2 競合動径基底関数ネットワークの概要
- 3 パラメータが従う確率密度関数の導出
- 4 動径基底関数の複製アルゴリズム
- 5 シミュレーションならびに考察

- STEP4 分岐により $\Delta_\beta m_j^k = 0$ となる点が増えたとき、第 j 動径基底関数を第 p 動径基底関数として複製する。そのとき、シナプス結合荷重 w_p 、パラメータ \sum_p ならびにパラメータ $m_p^n (n \neq k)$ は形質としてもとの第 j 動径基底関数のものを引き継ぎ、パラメータ m_p^k は新たに増えた点とする。STEP1 へ戻る。

5 シミュレーション並びに考察

まず、本研究で新たに提案した動径基底関数の複製についてその有用性を確認する。そこで、教師信号を発生させる関数 $\eta(x)$ として

$$\eta(x) = \sum_{j=1}^N w_j \xi(x, \phi_j) \quad (41)$$

を考えた。

- $N = 3$
- $[w_1, w_2, w_3]^T = [2, 1, 1, 5]^T$
- $[m_1, m_2, m_3]^T = [-4, -1, 4]^T$
- $[\Sigma_1, \Sigma_2, \Sigma_3]^T = [2^2, 0.5^2, 1^2]^T$
- 初期に与えられる動径基底関数は 1 つであると仮定
 - シナプス結合荷重 $w = 1.0$
 - パラメータ $m = 0$
 - パラメータ $\Sigma = 1.0$

1 はじめに

2 繼合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

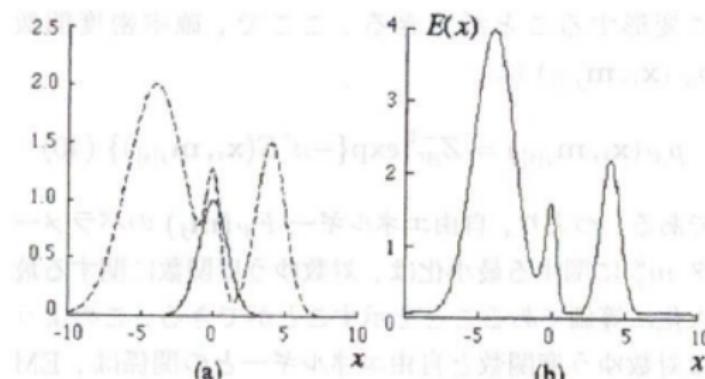
4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察

4.2 複製する位置の決定法

- 1 はじめに
- 2 競合動径基底関数ネットワークの概要
- 3 パラメータが従う確率密度関数の導出
- 4 動径基底関数の複製アルゴリズム
- 5 シミュレーションならびに考察

(a) は教師信号と書記の動径基底関数を合わせて示している。
 (b) は初期の動径基底関数の出力を 0 と仮定した時の 2 乗誤差関数 $E(x)$ を示す。
 教師信号は $x \in [-10, 10]$ の間から等間隔に 201 個とり, $(x_i, \eta(x_i)) i = 1, 2, \dots, 201$ で与えた。

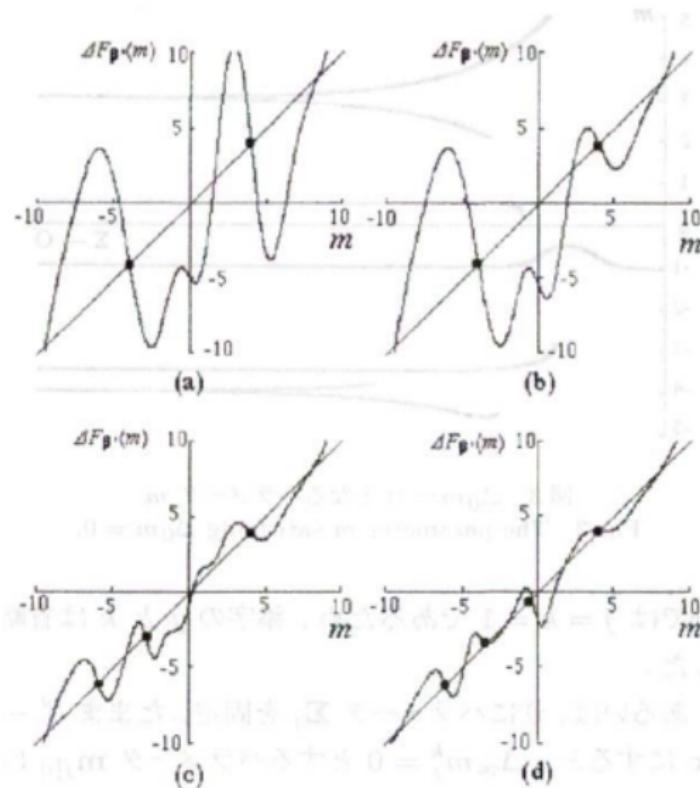


教師信号と初期の動径基底関数

図 6

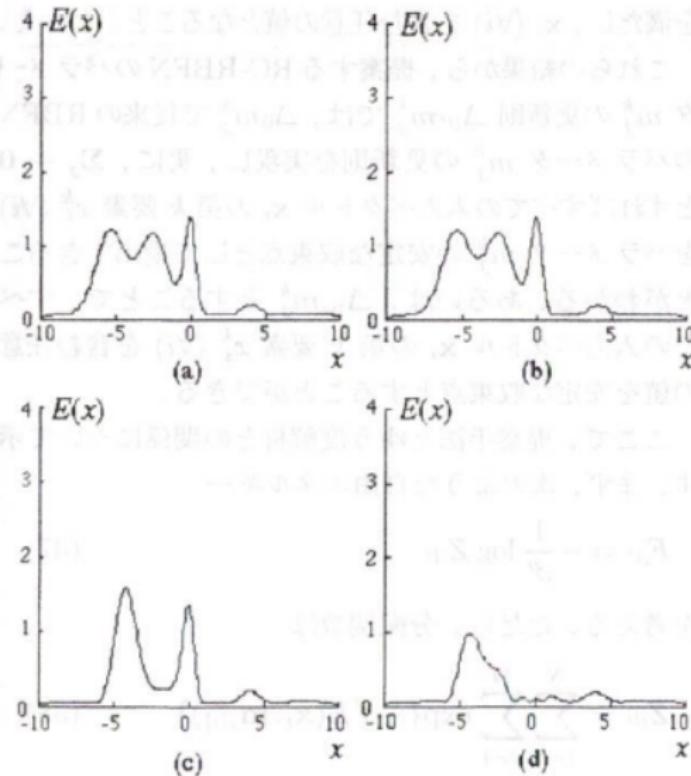
4.2 複製する位置の決定法

- 1 はじめに
- 2 競合動径基底関数ネットワークの概要
- 3 パラメータが従う確率密度関数の導出
- 4 動径基底関数の複製アルゴリズム
- 5 シミュレーションならびに考察



4.2 複製する位置の決定法

- 1 はじめに
- 2 競合動径基底関数ネットワークの概要
- 3 パラメータが従う確率密度関数の導出
- 4 動径基底関数の複製アルゴリズム
- 5 シミュレーションならびに考察



4.2 複製する位置の決定法

- 1 はじめに
- 2 競合動径基底関数ネットワークの概要
- 3 パラメータが従う確率密度関数の導出
- 4 動径基底関数の複製アルゴリズム
- 5 シミュレーションならびに考察

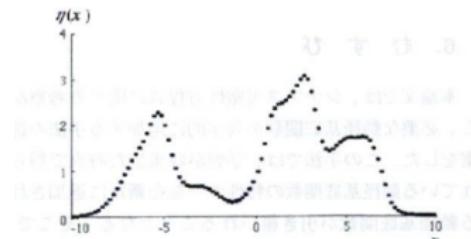


図 7 教師信号の分布

Fig. 7 The distribution of the teaching signals.

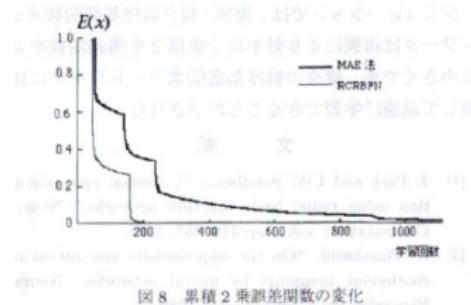


図 8 累積 2 乗誤差関数の変化

図 9

6 むすびに

シナプス可塑性方程式に関する考察から、必要な動径基底関数を効率的に追加する手法の提案をした。この手法では、学習が収束した時点で得られている動径基底関数の特性の一部を新たに追加される動径基底関数が引き継がれることとなる。シミュレーションでは、複合・競合動径基底関数ネットワークは複製により効率よく累積2乗誤差関数をより小さくでき、従来の動径基底関数ネットワークに比較して高速に学習できることが示された。

1はじめに

2 競合動径基底関数ネットワークの概要

3 パラメータが従う確率密度関数の導出

4 動径基底関数の複製アルゴリズム

5 シミュレーションならびに考察