進捗報告

小澤　翔太

富山県立大学　情報システム工学科

2024 年 12 月 17 日

行ったこと

前回指摘されたこと

・2 変量 RC-RBFN が本当に競合を行えているのか確認する必要がある

行ったこと

・グリッド上に基底関数を配置し，競合が行われていることを確認した

・複製機能の実装（要修正）

RBFN

動径基底関数ネットワーク (RBFN) とは

RBFN は非線形関数を近似するための関数近似器として用いられるニューラルネットワークの１つ．基底関数 (一般的にはガウス関数) を足し合わせることによって非線形関数を近似する．

・$\xi(x)$：多変数ガウス関数, $\mathbf{m}$: 基底関数の中心位置，Σ: 分散共分散行列

$$\xi(\mathbf{x}) = exp\{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\mathbf{m})\} \tag{1}$$

基底関数と重み $\mathbf{w}$ をかけ合わせて総和 $s(\mathbf{x})$ を求めることにより，関数近似を行う．

$$s(\mathbf{x}) = \sum \mathbf{w}\xi(\mathbf{x}) \tag{2}$$

これらのパラメータ $(\mathbf{w}, \mathbf{m}, \Sigma)$ をそれぞれ学習することで，誤差を少なくしていく．

基底関数の競合

競合動径基底関数ネットワーク

RBFN は非線形関数の近似器として有用だが，基底関数の数が多くなると学習の遅延や過学習を引き起こす．

そこで，適した基底関数 (正確には重み $\mathbf{w}$) だけが自然に生き残るような適者生存型学習則を適用した RBFN として，競合動径基底関数ネットワーク (CRBFN) が提案されている．

$$\frac{dw^j}{dt} = (\alpha^j - \sum_h \gamma^{jh} w^h) w^j \tag{3}$$

$$\alpha^j = \sum \eta(\mathbf{x}) \xi^j(\mathbf{x}) \tag{4}$$

$$\gamma^{jh} = \sum \xi^j(\mathbf{x}) \xi^h(\mathbf{x}) \tag{5}$$

w^j:j 番目の重み，α^j: 内的自然増加率，γ^{jh}: 競合の効果，$\eta(\mathbf{x})$: 教師信号

デモデータでの確認

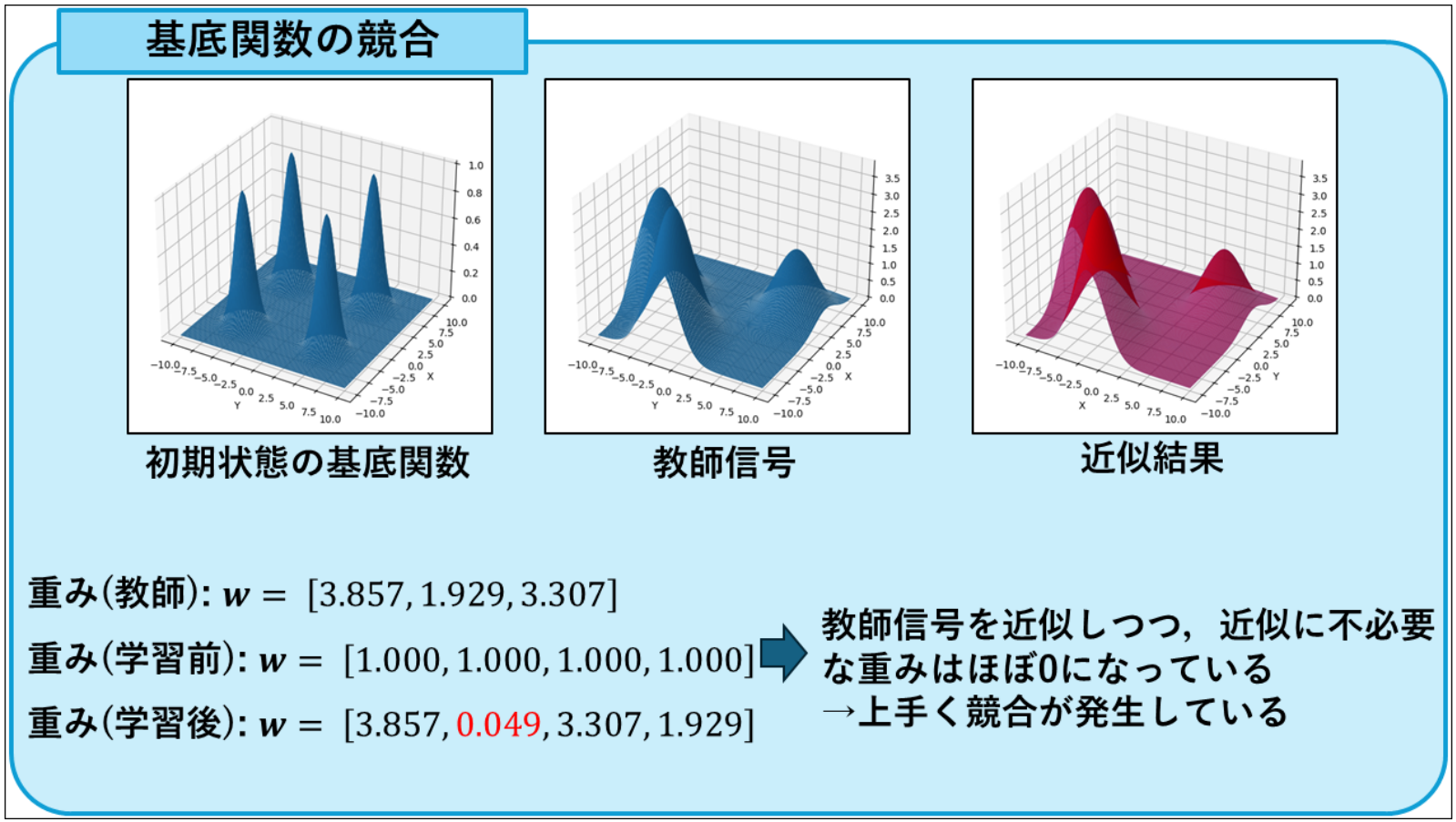

図 1: 競合の確認

基底関数の複製

複製・競合動径基底関数ネットワーク

CRBFN は冗長な基底関数を削除する機能をもつものの，足りない基底関数を追加する機能は備わっていない．目的関数が多峰性の場合，基底関数の数が足りないと近似精度が落ちる可能性がある．

そこで，効率的に基底関数を複製する機能を追加した CRBFN として，複製・競合動径基底関数ネットワーク (RC-RBFN) が提案されている．

複製する位置の決定法

1. 基底関数の中心位置の学習

$$\Delta_\beta m^j = \sum_x p_\beta\left(x|m^j_{[x]}\right)\Delta m^j_{[x]}$$

β：定数　$m^j_{[x]}$：xに依存する平均位置

$p_\beta\left(x|m^j_{[x]}\right)$：条件付き確率密度関数

2. 累積2乗誤差関数の計算

$$E(m^j) = \frac{1}{2}\sum_x E\left(x, m^j_{[x]}\right)$$

$$= \frac{1}{2}\sum_x \{\eta(x) - s(x)\}^2$$

$E(m^j) \approx 0$ → 終了

$\Delta E(m^j) < \epsilon$ → 3へ

3. 複製する位置の決定

$$\sum_x e^{-\beta E\left(x, m^j_{[x]}\right)} \xi^j\left(x, m^j_{[x]}\right)\left(x - m^j_{[x]}\right)\left\{\eta(x) - s\left(x, m^j_{[x]}\right)\right\} = 0$$

βを大きくしながら平均位置を$\Delta_\beta m^j$で更新

→上記の方程式を満たす位置に基底関数を複製→1へ

デモデータでの確認

問題点

・超越方程式を解くソルバーがないため，xy 平面上の点を 1 つずつ計算する必要がある．しかし，刻み幅を細かくしたり，変数を増やしたりすると処理時間が爆発的に増えるため，実用的ではなくなってしまう．
→ python の高速化ライブラリである numba の使用などを検討する必要がある．
・そもそも 1 変数のときと同様に効率的に基底関数を追加できてるのか確認する必要がある．

TOYAMA Prefectural University

これから行うこと

- PSO に組み込むのではなく，複製の時間短縮を卒論とするかの話し合い
- 複製部分の完成，ターミナルアトラクタをどうするか
- 本論執筆