ラジアル基底関数ネットの高速追加学習アルゴリズム

岡本圭介*・小澤誠一*・阿部重夫*

A Fast Incremental Learning Algorithm for Radial Basis Function Networks

Keisuke OKAMOTO*, Seiichi OZAWA* and Shigeo ABE*

When training samples are given incrementally, neural networks often suffer from the catastrophic interference, which results in forgetting input-output relationships acquired in the past. To avoid the catastrophic interference, we have proposed Resource Allocating Network with Long-Term Memory (RAN-LTM). In RAN-LTM, not only a new training sample but also some memory items stored in long-term memory are used for training based on a gradient descent algorithm. In general, the gradient descent algorithm is usually slow and can be easily fallen into local minima. In this paper, to alleviate these problems, we introduce a linear regression approach into the learning of RAN-LTM, in which its centers are not trained but selected based on output errors in an incremental fashion. In this approach, the regression is carried out for not only a training sample and memory items but also pseudodata that are selected around the centers of hidden units based on the complexity of an approximated function. This selection reduces the total number of pseudodata at each learning step; as a result, fast incremental learning is realized in RAN-LTM. Since only memory items are stored in memory, the proposed RAN-LTM does not need so much memory capacity when the incremental learning is carried out. This property is useful especially for small-scale systems. To verify these characteristics of RAN-LTM, we apply it to several function approximation problems, in which the performance in approximation accuracy, learning time, and needed memory capacity are investigated by comparison with some conventional models. Moreover, when extending the learning domain with time, the increase trends in learning time and needed memory capacity are investigated. From the experimental results, it is verified that the proposed model can learn fast and accurately, and that it needs rather small memory capacity so far as the learning domain is not too large.

Key Words: incremental learning, neural networks, catastrophic interference, memory-based learning, linear regression

1. Introduction

近年、ニューラルネットがパターン認識、予測、診断、制御などの実用システムに取り入れられることは珍しくない。これらの多くでは、システムの稼動中に生じる環境変動を無視できると仮定しており、事前に用意した訓練データでオフライン学習を行ない、オンラインでの学習は行なわないのが一般的である。しかし現実的には、個人認証や株価予測など、時々刻々と変動する環境に適応し、常にベストパフォーマンスを要求される応用事例も少なくない。この場合、リアルタイムで収集された訓練データを逐次的に学習し、システムの性能改善を常に行なえる「追加学習」を実用システムに取り入れることが重要となる。

ニューラルネットでは、過去に獲得された入出力関係が結合荷重に分散して記憶されるため、新しい訓練データの学

習にともなう結合荷重の修正により、過去の記憶を蓄えた結合荷重も同時に修正され、忘却を引き起こしやすい。この現象は追加学習にともなう「干渉」と呼ばれ、以前から「可塑性と安定性のジレンマ」として知られてきた問題である¹⁾。しかし、ニューラルネットにおける記憶の分散性は少ないパラメータ数で精度の高い入出力関係を獲得できる汎化能力を実現しており、効率的な追加学習を目指す上でニューラルネットは有望な選択肢の一つである。

干渉を効率よく抑制し、安定な追加学習を行なうアプローチとしては、ネットワークの局所的な反応特性を高めて結合荷重の修正範囲を制限するもの^{1)~4)}、過去に与えられた訓練データやその代表的なものをメモリに蓄え、それらを同時に学習することで忘却を防ぐアプローチ^{5)~8)}などが提案されている。干渉を完全に抑制するためには、後者のアプローチが有効であるが、過去の訓練データを記憶し、それらを学習する必要があるため、学習に必要なメモリや時間のコストは前者に比べて一般に高くなる。前述したように、ニューラルネットに追加学習を行なわせるメリットは、少ない記憶データ量で精度の高い追加学習を行なうことにあるといえ、

* 神戸大学大学院自然科学研究科 神戸市灘区六甲台町 1-1

* Graduate School of Science and Technology, Kobe University, 1-1 Rokko-Dai, Nada, Kobe

(Received November 26, 2003)

(Revised August 12, 2004)

この意味で、後者のアプローチにおいてメモリや時間のコストを削減することは重要な課題といえる。

このような課題に対し、われわれは Resource Allocating Network (RAN)⁹⁾ に長期記憶 (Long-Term Memory, LTM) を導入した Resource Allocating Network with Long-Term Memory (RAN-LTM) を提案してきた⁸⁾。RAN-LTM では、干渉抑制に必要な最小限の入出力対を記憶アイテムとして長期的に格納し、これらの中から干渉抑制に効果的なものを学習に利用する。このメカニズムによって、従来のメモリベース学習に比べて、かなり少ない記憶データ量で正確な近似が可能となる。しかし、学習アルゴリズムが最急降下法に基づいており、局所最小解への収束や収束スピードなどの問題が以前から指摘されていた。

ラジアル基底関数ネット (RBFN)¹⁰⁾ の学習では、基底中心を固定して、隠れ層-出力層間結合の学習を線形方程式の求解に帰着する方法が古くから提案されており^{11), 12)}、実際、RBFN の学習法として一般的に用いられている。一方、ニューラルネットの追加学習では、干渉の抑制が容易になるように一度に修正する結合荷重を限定する局所情報に基づいた学習法が主流であり、このような経緯から、RAN-LTM においても最急降下法に基づいた学習法を提案してきた。しかし、線形方程式の求解に基づく学習法は非常に高速であり、これを追加学習に拡張することが望ましいことはいうまでもない。Orr は線形方程式の求解に基づく方法で追加学習を可能にしているが¹¹⁾、すべての訓練データを隠れユニットの基底中心としており、学習効率は悪い。よって、このアプローチを RAN-LTM に効果的に取り入れるには、逐次的に与えられる訓練データの中から必要なものを基底中心として選択するメカニズムが必要となる。

このような基底中心の選択メカニズムを導入した RAN-LTM の高速追加学習アルゴリズムを文献 13) で提案した。この学習アルゴリズムでは、追加データに対する訓練誤差と隠れユニットの活性度に基づいて基底生成が行なわれ、干渉抑制の目的で基底中心とその近傍データに相当する擬似データを追加データとともに線形方程式の求解に基づく方法で学習する。

本論文では、近似関数の複雑さに基づいて擬似データ数を決定し、不必要に多くの擬似データが生成されないようにすることで、一回の追加学習で用いる訓練データ数を減らした改良アルゴリズムを提案する。これによって、解くべき線形方程式の数を減らすことができ、学習時間と記憶データ量に対する効率性を高めることが可能となる。また、訓練データが偏って発生することの影響や学習領域の拡大に対する影響を考察し、さらに予測問題に適用することで実用面での有効性も検証する。

第 2 章で線形方程式の求解に基づく RBFN の学習法について簡単に説明し、第 3 章で RAN-LTM のネットワーク構造と提案する追加学習アルゴリズムについて述べる。第 4 章では、関数近似問題や予測問題に適用し、提案モデルの追加

学習性能を他手法との比較を通して総合的に評価していく。最後に、第 5 章で結論と今後の課題について述べる。

2. 線形方程式の求解に基づく RBFN の学習

RBFN の入力層、隠れ層、出力層のユニット数をそれぞれ I, J, K とし、 p 番目の入力およびその隠れユニット出力とネットワーク出力をそれぞれ $\mathbf{x}_p = \{x_{p1}, \dots, x_{pI}\}'$, $\mathbf{y}(\mathbf{x}_p) = \{y_{p1}, \dots, y_{pJ}\}'$, $\mathbf{z}(\mathbf{x}_p) = \{z_{p1}, \dots, z_{pK}\}'$ で表わす。ここで、 $'$ はベクトルまたは行列の転置を表わす。 j 番目の基底中心を $\mathbf{c}_j = \{c_{j1}, \dots, c_{jI}\}'$, j 番目の隠れユニットから k 番目の出力ユニットへの結合荷重を w_{kj} とするとき、RBFN における順方向の信号伝播は以下のように表わされる。

$$y_{pj} = \exp\left(-\frac{\|\mathbf{x}_p - \mathbf{c}_j\|^2}{\sigma_j^2}\right) \quad (j = 1, \dots, J) \quad (1)$$

$$z_{pk} = \sum_{j=1}^J w_{kj} y_{pj} + \gamma_k \quad (k = 1, \dots, K) \quad (2)$$

ここで、 σ_j^2 は j 番目のラジアル基底の分散、 γ_k は k 番目の出力のバイアス値である。

線形方程式の求解に基づく方法では、つぎの 2 段階に分けて学習が行なわれる。

(1) ラジアル基底中心の決定

(2) 隠れ層-出力層間結合荷重 (以下、結合荷重) の計算
バッチ学習の場合、あらかじめ訓練データが与えられるため、(1) についてはクラスタリングなどの手法を用いて基底中心の決定を行なえる。しかし、追加学習の場合、訓練データが逐次的にしか与えられるため、バッチ学習で用いられる方法をそのまま適用できない。基底中心の決定方法は次節で述べることにし、以下では結合荷重を求める方法について説明する。

いま、 N 個の訓練データ $(\mathbf{x}_p, \mathbf{T}_p)$ ($p = 1, \dots, N$) が与えられているとする。ここで、 $\mathbf{x}_p$ と $\mathbf{T}_p$ はそれぞれ p 番目の入力とその教師である。よって、入力と出力の間には以下の関係が成り立てばよい。

$$\mathbf{A}\mathbf{W}' = \mathbf{T} \quad (3)$$

ここで行列 $\mathbf{A}$ は、各訓練データ $\mathbf{x}_p$ に対する隠れユニットの活性度を行列で表わしたもので、その (p, j) 成分は (1) 式の y_{pj} で与えられる。また、 $\mathbf{W}$ は隠れ層-出力層間の結合荷重を行列で表わしたものである。通常、 $N > J$ であり、(3) 式の最小二乗解 $\mathbf{W}^*$ は $\mathbf{A}$ の特異値分解 $\mathbf{A} = \mathbf{U}\mathbf{H}\mathbf{V}'$ を用いて、以下で与えられる。

$$\mathbf{W}^* = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{T} = \mathbf{V}\mathbf{H}^{-1}\mathbf{U}'\mathbf{T} \quad (4)$$

ここで、行列 $\mathbf{U}$ は $J \times N$ の直交行列、行列 $\mathbf{H}$ は $N \times N$ の対角行列、行列 $\mathbf{V}$ は $N \times N$ の直交行列である。

以上のように、RBFN の学習はラジアル基底中心を固定することで、線形方程式の求解に帰着される。

Fig. 1 Architecture of RAN-LTM

3. 線形方程式の求解に基づく RAN-LTM の学習

Fig. 1 に示すように、RAN-LTM は入出力関係を学習する RAN 部と代表的な入出力データ対 (以下では記憶アイテムと呼ぶ) を長期的に記憶する LTM 部で構成される⁸⁾。p 番目の訓練データ (x_p, T_p) を学習する際、LTM 部から干渉抑制に効果のある記憶アイテム $(\tilde{x}_t, \tilde{z}_t, H'(\tilde{x}_t))$ を RAN 部に送り (「想起」と呼ぶ)、これらを訓練データとともに学習する。つまり、訓練データと想起された記憶アイテムを合わせて訓練データセットとする。これにより、過去に獲得された入出力関係をできるだけ維持しながら新しい訓練データに適応できるようになり、干渉による忘却が抑制される。なお、RAN 部における順方向の信号伝播は RBFN と同じであり、(1)、(2) 式で表わされる。

本論文では、文献 8) で提案した RAN-LTM の追加学習アルゴリズムに線形方程式の求解に基づく方法を適用するため、いくつかの点でアルゴリズムを変更する。この変更点が明確になるよう、学習アルゴリズムを構成する以下の 4 つの操作に分けて説明する。

- (1) 隠れユニットの追加と基底中心の決定
 - (2) 記憶アイテムの生成
 - (3) 記憶アイテムの想起
 - (4) 隠れ層-出力層間結合荷重 (以下、結合荷重) の計算
- また、3.2 において、提案する追加学習アルゴリズムをまとめて示す。

3.1 学習アルゴリズムの変更点

提案する追加学習アルゴリズムの理解を助けるため、以下の記述が 3.2 の提案アルゴリズムのどの部分に対応するのかわ () に付記する。

3.1.1 隠れユニットの追加と基底中心の決定

RAN の学習アルゴリズムでは、(i) 訓練データの出力誤差が閾値以上であり、(ii) 訓練データと基底中心との最短距離が閾値以上であるとき、隠れユニットが追加される⁹⁾。つまり、訓練データが未学習領域にあると判断されると隠れユニットが追加され、訓練データがその基底中心にセットされる。しかし、(i) の出力誤差の評価は訓練データを追加学習する前に行なわれるため、隠れユニットが追加されない場合

において、追加学習後の近似精度が保証されているわけではない。十分な近似精度を保証するには、(ii) の閾値を小さくする必要があるが、一般に、これによって隠れユニット数は増加し、結合荷重の計算コストが増大する。

そこで本論文では、隠れユニットの追加を結合荷重の修正前後の 2 段階に分けて行なう方法を提案する。つまり、Platt の学習アルゴリズムで行なわれていた隠れユニットの追加を第一段階とし (ステップ (2) i) に対応)、結合荷重の修正後に行なわれる隠れユニットの追加を第二段階とする (ステップ (7) に対応)。第一段階において、追加条件をいくぶん厳しく設定することで余分な隠れユニットが追加されないようにし、これでは十分な近似精度が保証されない場合、第二段階で隠れユニットが追加されるようにする。これにより、一定の近似精度を維持するのに必要かつ最小限の隠れユニットを追加でき、適切な基底中心の決定が行なわれると期待される。

3.1.2 記憶アイテムの生成

文献 8) の RAN-LTM では、特定の基底がサポートしている入力領域に対する近似精度が十分高まったとき、その基底中心と対応する出力を記憶アイテムとして抽出し、LTM 部に保存する。しかし、線形方程式の求解に基づく方法を適用する場合、隠れユニットの追加と同時に基底中心が固定されるため、そのとき与えられた訓練データをそのまま記憶アイテムとすればよい (ステップ (2) iv) に対応)。

3.1.3 記憶アイテムの想起

最急降下法に基づく学習では、隠れユニットの活性度に比例して結合荷重の修正量が大きくなるため、訓練データに対して大きな活性度をもつ隠れユニットの基底中心付近で干渉が起こりやすい。そこで文献 8) の RAN-LTM では、このような基底中心付近にある記憶アイテムに限定して LTM 部から想起し、学習を行なっている。しかし、想起した記憶アイテムの学習で生じる新たな干渉もあり、高い近似精度を保証するには、できるだけ多くの記憶アイテムを同時に学習するのが望ましい。線形方程式の求解に基づく方法によって学習スピードの改善が見込めるため、ここでは、すべての記憶アイテムを想起して訓練データとともに学習する (ステップ (5) に対応)。

3.1.4 結合荷重の計算

3.1.3 で述べたように、提案手法では、訓練データが入力されるごとにすべての記憶アイテムを想起する。3.1.2 から、記憶アイテム数は隠れユニット数に等しいため、1 回の追加学習で $J+1$ 個のデータを訓練データセットとして利用できるようになる。この訓練データセットから、(4) 式に基づいて結合荷重 W を求めるが、隠れユニット数が少ないと訓練データ数も少なくなり、十分な近似精度が得られない可能性がある。

そこで、すでに配置された基底中心近傍では近似精度が高いことを利用し、この近傍付近の入出力データ (以下では「擬似データ」と呼ぶ) を RAN で生成し (ステップ (4) に対

応), それらを学習データとして利用する. つまり, 記憶アイテム $\tilde{\mathbf{x}}_m$ ($m = 1, \dots, J$) に対して, L_m 個の擬似データを以下のように生成する.

$$(\tilde{\mathbf{x}}_m + \delta_l, \mathbf{z}(\tilde{\mathbf{x}}_m + \delta_l)) \quad (l = 1, \dots, L_m) \quad (5)$$

ここで, $\mathbf{z}(\mathbf{x}) = [z_1, \dots, z_K]'$ は $\mathbf{x}$ を RAN の入力としたときの出力, δ_l は小さな正数を要素にもつベクトルである. このように生成された擬似データは, RAN で学習した近似関数を用いて生成されるため, 真の入出力関係を表わしているとは限らない. しかし, 擬似データは追加学習が行なわれるたびに更新されるため, 干渉が抑制されていけば, 学習が進むにつれて擬似データは正確になると期待される.

また, 擬似データ数が多いと学習コストが増大するため, できるだけ少なくすることが望ましい. しかし, 関数が複雑であればあるほどデータ点を多く必要とすることから, 擬似データ数をすべての基底で同数にすることは効率的でない. そこで, m 番目の記憶アイテム $\tilde{\mathbf{x}}_m$ 近傍の関数の複雑度を以下のヘシアン情報 $H'(\cdot)$ で定量化する (ステップ (2) iii) に対応).

$$H'(\tilde{\mathbf{x}}_m) = \min\left\{\frac{\sum_k |H_k(\tilde{\mathbf{x}}_m)|}{H_{\max}}, 1\right\} \quad (6)$$

ここで, $H_{\max}$ は記憶アイテムがもつヘシアン情報の最大値である. また, 記憶アイテム $\tilde{\mathbf{x}}_m$ に対する隠れユニット出力とネットワーク出力をそれぞれ $\tilde{\mathbf{y}}(\tilde{\mathbf{x}}_m) = [\tilde{y}_{m1}, \dots, \tilde{y}_{mJ}]'$, $\tilde{\mathbf{z}}(\tilde{\mathbf{x}}_m) = [\tilde{z}_{m1}, \dots, \tilde{z}_{mK}]'$ としたとき, $|H_k(\cdot)|$ は (i, i') 要素が次式で与えられるヘシアン行列の行列式を表わす.

$$\frac{\partial^2 \tilde{z}_{mk}}{\partial c_{ji} \partial c_{ji'}} = \sum_{l=1}^J \frac{w_{kl}}{4\sigma_l^4} (c_{ji} - c_{lk})(c_{ji'} - c_{li'}) \tilde{y}_{ml} \quad (7)$$

このヘシアン情報に基づき, 擬似データ数 L_m を以下のように決定する (ステップ (3) に対応).

$$L_m = L_{\max} H'(\tilde{\mathbf{x}}_m) \quad (8)$$

ここで, $L_{\max}$ は擬似データ数の最大値である. これからわかるように, ヘシアン情報が大きいほど (つまり関数が複雑であるほど) 擬似データ数は多くなる. なお, 新しい訓練データを追加学習するたびに入出力関数は変更されるため, 各記憶アイテムのヘシアン情報を更新する必要があることに注意する (ステップ (8) に対応).

以上より, 訓練データが一つ与えられるたびに記憶アイテムと擬似データを同時に学習することになり, 学習データの総数は $J + \sum_{m=1}^J L_m + 1$ 個となる. これら学習データを RAN に入力し, 隠れユニットの活性度行列 $\mathbf{A}$ を求めて (ステップ (5) に対応), (4) 式に基づき結合荷重 $\mathbf{W}$ を求める (ステップ (6) に対応). なお, このように求められた結合荷重は, 過去の記憶に対する出力誤差と追加された訓練データに対する出力誤差を同等に扱ったときの最小二乗解に相当する.

3.2 提案アルゴリズム

3.1 で示した変更を加えた提案アルゴリズムを以下に示す.

(1) p 番目の入力 $\mathbf{x}_p$ に最も類似したラジアル基底中心 $\mathbf{c}^*$ を探索する. また, それに対するネットワーク出力 $\mathbf{z}_p$ と教師信号 $\mathbf{T}_p$ との誤差 E を計算する.

(2) $E < \eta_1$ または $\|\mathbf{x}_p - \mathbf{c}^*\| < \eta_2$ (η_1, η_2 : 正の定数) を満たすとき, ステップ (3) へ. それ以外るとき, 以下を実行する.

i) 隠れユニット数 J をインクリメントし, 訓練データを基底中心とする隠れユニットを追加する. (つまり, $\mathbf{c}_J = \mathbf{x}_p$)

ii) 結合荷重行列 $\mathbf{W}$ の列に教師信号との誤差ベクトル $\mathbf{T}_p - \mathbf{z}_p$ を以下のように加える.

$$\mathbf{W}^{new} = [\mathbf{W}, \mathbf{T}_p - \mathbf{z}_p] \quad (9)$$

iii) 訓練データ $\mathbf{x}_p$ を RAN に入力し, (6) 式に基づきヘシアン情報 $H'(\tilde{\mathbf{x}}_p)$ を計算する.

iv) 記憶アイテムを以下のように生成する.

$$(\tilde{\mathbf{x}}_J, \tilde{\mathbf{z}}_J, H'(\tilde{\mathbf{x}}_J)) \leftarrow (\mathbf{x}_p, \mathbf{T}_p, H'(\mathbf{x}_p))$$

v) ステップ (9) へ

(3) (8) 式に基づき, 各記憶アイテムごとに生成する擬似データ数 L_m を求める.

(4) 擬似データを (5) 式に基づき生成する.

(5) 訓練データ, 記憶アイテム, 擬似データを RAN に入力し, 隠れユニット出力を求め, 活性度行列 $\mathbf{A}$ を構成する.

(6) 活性度行列 $\mathbf{A}$ に対して特異値分解を適用し, (4) 式に基づいて結合荷重行列 $\mathbf{W}$ を求める.

(7) 訓練データ $\mathbf{x}_p$ を RAN に入力し, 出力誤差が $E < \eta_1$ を満たしていればステップ (8) へ. それ以外るとき, ステップ (2) と同様の手順に基づき, 隠れユニットの追加および記憶アイテムの生成を行なう.

(8) 各記憶アイテムに対し, $\tilde{\mathbf{x}}_m$ を RAN に入力して, ヘシアン情報を (6) 式に基づいて更新する.

(9) 新たな訓練データが提示されたら, $p \leftarrow p + 1$ としてステップ (1) へ.

4. シミュレーション実験

本節では, いくつかの関数近似の問題に適用し, 学習に要する時間と記憶データ量, 近似精度, 干渉の抑制効果に関して, 提案手法の評価を行なう. 比較するモデルは, (a) 最急降下法に基づく従来の RAN-LTM, (b) Platt が提案したオリジナルの RAN, (c) 追加された訓練データをすべて基底中心にセットして線形方程式を解く方法で学習を行なう RBFN, である. (a) との比較を通して RAN-LTM に線形方程式を解く方法を導入する効果を考察し, (b) との比較を通して干渉抑制の効果を考察する. (b) を考察するには, 干渉の程度を定量化する必要があるが, これは次節で述べる. また (c) では, 訓練データのうち代表的なもののみを記憶し, 学習を行なうことの有効性を考察する.

4.1 干渉の定量化

訓練データを追加学習して生じる干渉には、追加学習によって汎化能力が増加する「正の干渉」と過去に学習した入出力関係を破壊する「負の干渉」がある¹⁴⁾。以下では、これら正と負の干渉を定量化する方法を提案し、学習アルゴリズムの干渉抑制能力を評価する指標として利用する。訓練データ x_p を追加学習する前後において、ニューラルネットワークが入出力関数をそれぞれ $f_p(x)$, $f'_p(x)$ 、目標とする真の関数を $f^*(x)$ とする。また、訓練データが発生する領域 $\mathcal{X}$ に対して、以下の2つの集合を定義する。

$$R_p^+ = \{x \mid x \in \mathcal{X}, |f_p(x) - f^*(x)| > |f'_p(x) - f^*(x)|\}$$

$$R_p^- = \{x \mid x \in \mathcal{X}, |f_p(x) - f^*(x)| \leq |f'_p(x) - f^*(x)|\}$$

この定義から明らかなように、 R_p^+ は x_p を追加学習することで真の関数との誤差が減少する点の集合を表わし、逆に R_p^- は誤差が増加する点の集合を表わす。これから、正と負の干渉は以下のように定量化できる。

$$I_p^+ = \int_{x \in R_p^+} |f'_p(x) - f_p(x)| dx \quad (10)$$

$$I_p^- = \int_{x \in R_p^-} |f'_p(x) - f_p(x)| dx \quad (11)$$

また、追加学習アルゴリズムによる干渉抑制の効果は、以下で定義される負の干渉量の割合で定量化できる。

$$C_p = \frac{I_p^-}{I_p^+ + I_p^-} \quad (12)$$

つまり、 C_p が1に近いとき、 x_p の追加学習によって負の干渉による忘却が深刻になると予想され、 C_p が0に近いほど安定した追加学習が行なえていることを表わす。

4.2 関数近似問題への適用

4.1 で示した干渉の定量化は、訓練データの分布が未知である現実の問題では評価が難しい。また、訓練データの発生分布が偏りをもつことの影響や問題の大きさの学習時間や記憶データ量に対する影響など、追加学習の基本性能を調べるには、現実の問題だけでなく、真の関数が既知である問題に適用するほうが考察しやすい。そこで以下の実験では、次式で与えられる2種類の1次元関数 $h_1(x)$ と $h_2(x)$ を近似関数とする問題に適用する。

$$h_1(x) = \frac{30 \sin(\pi x + 2\pi) \cos(2\pi x + 4\pi)}{x + 15} \quad (13)$$

$$h_2(x) = \begin{cases} 2(x-n)(n \leq x < n+0.5) \\ -2(x-n-1)(n+0.5 \leq x < n+1) \end{cases} \quad (14)$$

(n : 整数)

Fig. 2(a)(b) に関数 $h_1(x)$ と $h_2(x)$ に示す。なお、ニューラルネットワークに与えられる p 番目の訓練データは、(13),(14)式に基づき $(x_p, h_1(x_p))$, $(x_p, h_2(x_p))$ で与えられる。

追加学習に対する性能は、一般に訓練データの与えられ方や学習領域の大きさに対して影響を受ける。そこで、これら2つの要因に対して、学習時間、記憶データ量、近似精

Fig. 2 Target functions: (a) $h_1(x)$, (b) $h_2(x)$.

度、干渉抑制の効果を調べ、提案手法の有効性について議論する。

4.2.1 性能評価

追加学習の問題では、訓練データがどのような順序で与えられるかは学習主体にとって未知である。最も単純でかつよく用いられるのは、訓練データが学習領域において一様に発生すると仮定することである。しかし、現実的には、訓練データの発生分布が偏りを持ち、それが時間的に変動すると仮定するほうが自然である。後者の場合、ある時期に与えられた訓練データが二度と現れない可能性もあり、一般的には前者に比べて追加学習は難しく考えられる。ここでは学習領域 $\mathcal{X}$ を $[0, 12]$ とし、訓練データが領域 $\mathcal{X}$ で一様に発生するケースと偏りをもつケースの両方について性能評価を行なう。

Case 1: 学習領域 $\mathcal{X}$ から一様に 300 個の訓練データを発生させる。

Case 2: Fig. 2 に示すように、学習領域 $\mathcal{X}$ を 5 つの小領域 A~E に分け、一つの小領域で 60 個の訓練データを一様に発生して学習した後、別的小領域で同様に訓練データを発生させる。ただし、小領域 A~E の順序はランダムに決めるものとする。

上述のように、訓練データは合計 300 個与えるが、これら訓練データの一つずつネットワークに与えて追加学習を行なう。なお、追加学習が終わった訓練データは廃棄され、以後の学習には利用されないことに注意する。また、訓練データ

Table 1 Performance of incremental learning when $h_1(x)$ is approximated as a target function in (a) Case 1 and (b) Case 2

(a) Case 1				
MODEL	Time (sec.)	Training Ave. Error	Test Ave. Error	Memory (KB)
RBFN	110.6	0	0.0001	727.2
RAN	3.1	0.184	0.184	1.6
RAN-LTM(O)	229.5	0.0306	0.0304	3.4
RAN-LTM(N)	8.9	0.0110	0.0117	38.0

(b) Case 2				
MODEL	Time (sec.)	Training Ave. Error	Test Ave. Error	Memory (KB)
RBFN	101.6	0	0.0001	727.2
RAN	3.2	0.194	0.194	1.6
RAN-LTM(O)	287.9	0.0308	0.0310	3.4
RAN-LTM(N)	5.7	0.0137	0.0147	37.8

とは異なる 3000 点をランダムに選択してテストデータとし、訓練データの追加学習がすべて終了した後、蓄積された干渉の影響を調べるために近似誤差を評価する。最急降下法を用いる RAN と RAN-LTM に関しては、訓練データの平均誤差が 0.01 以下となるか、100,000 回更新が行なわれた時点で追加学習を終了する。訓練データの提示系列を 5 種類変えて性能評価を行ない、学習時間 (Time)、訓練誤差 (Training Ave. Error)、テスト誤差 (Test Ave. Error)、記憶データ量 (Memory) の平均値を **Table 1** と **Table 2** に示す。表中における RAN-LTM(O) および RAN-LTM(N) は、それぞれ最急降下法と線形方程式の求解に基づく RAN-LTM モデルを表す。なお、記憶データ量とは、実行時にメモリに記憶される訓練データ、記憶アイテム、結合荷重、基底中心をあわせたデータ量 (KB) の最大値である。

Table 1 と Table 2 から、Case 1 よりも Case 2 のほうが訓練データやテストデータに対する誤差が大きく、訓練データが偏って発生することで学習が難しくなることがわかる。しかし、近似能力の高さは訓練データの与え方に関係なく、メモリベース RBFN が最も高く、RAN-LTM(N)、RAN-LTM(O)、RAN の順であった。メモリベース RBFN はかなり近似能力が高いものの、必要な記憶データ量が多く、学習時間も提案手法である RAN-LTM(N) に比べてかなり遅い。これは、訓練データが逐次的に与えられるたびに、これらを基底中心としてすべて記憶して結合荷重を再学習するため、追加学習というよりは毎回バッチ学習を行っているためである。RAN-LTM(O) は干渉を抑制する機能をもたない RAN と比べて、近似能力は高いが、学習に要する時間はほかのモデルに比べてかなり長い。一方、RAN-LTM(N) は学習精度と速度の両方において、従来手法である RAN-LTM(O) よりも優れている。これは、最急降下法に代えて線形方程式の求解に基づく方法を導入したことで記憶アイテムや擬似データを学習に利用した結果と考えられる。以上より、提案手法はメモリベース RBFN と比較

Table 2 Performance of incremental learning when $h_2(x)$ is approximated as a target function in (a) Case 1 and (b) Case 2

(a) Case 1				
MODEL	Time (sec.)	Training Ave. Error	Test Ave. Error	Memory (KB)
RBFN	116.7	0	0.0371	727.2
RAN	3.6	0.267	0.272	1.6
RAN-LTM(O)	659.7	0.103	0.124	3.1
RAN-LTM(N)	12.9	0.0549	0.0810	66.3

(b) Case 2				
MODEL	Time (sec.)	Training Ave. Error	Test Ave. Error	Memory (KB)
RBFN	110.9	0	0.0371	727.2
RAN	3.5	0.266	0.270	1.5
RAN-LTM(O)	695.6	0.122	0.140	3.2
RAN-LTM(N)	8.4	0.0702	0.0984	65.7

すると近似精度はいくぶん低いですが、学習速度がかなり速く、必要となる記憶データ量もかなり少ない。このことから、提案手法は限られた計算機資源のもとで、その有効性が発揮される手法であるといえる。

つぎに、追加学習に伴う干渉の影響について考察するため、誤差および 4.1 で定義した負の干渉量の割合の移り変わりについて調べた。Case 2 で $h_1(x)$ を近似関数としたときの結果を **Fig. 3** に示す。ただし、メモリベース RBFN は実質的にバッチ学習をしていることと等価であるため、ここでは RAN、RAN-LTM(O)、RAN-LTM(N) の結果のみを示している。横軸の p は逐次的に与えられる訓練データのインデックスであり、Fig. 3(a) の縦軸は未提示のものも含めた全訓練データに対する平均誤差を表わしている。よって、干渉による忘却を抑制しながら追加学習が正確に行なわれていれば、徐々に訓練誤差は減少することになる。Fig. 3(a)(b) からわかるように、RAN の誤差はほかのモデルに比べて減少量が小さく、負の干渉量 C_p も 0.5 付近を上下している。 C_p が 0.5 より大きくなると、負の干渉が正の干渉に対して支配的になっていることを表わしており、実際、0.5 以上となった時点で誤差が増加または横ばいになっている。一方、RAN-LTM(O) および RAN-LTM(N) では、 C_p が 0.5 より大きくなることはほとんどなく、 C_p が小さくなったときに誤差も劇的に減少している。このことから、RAN-LTM(O) や RAN-LTM(N) の近似能力の高さは、負の干渉を抑制できていることに起因すると結論できる。

4.2.2 学習領域の拡大に対する影響

追加学習は、本来、訓練データの提示に終わりがあるかどうかかわからない状況のもとで必要となる学習方式である。よって、追加学習アルゴリズムを評価する際、高い近似精度が維持されるだけでなく、学習領域が拡大しても学習が可能かどうかについても調べる必要がある。そこで、近似関数の定義域を徐々に拡大させながら追加学習を行ない、学習時間や記憶データ量の変化のようすを調べる。

(a) Average Error

(b) Negative Interference Rate C_p

Fig. 3 Time courses of (a) average errors and (b) negative interference rates C_p for RAN, RAN-LTM(O) and RAN-LTM(N). If C_p is larger than 0.5, the negative interference dominates the positive one.

近似関数には (13) 式の関数 $h_1(x)$ を用いる。ただし、小領域を $X_n: \{x | 4(n-1) < x < 4n\}$ ($n = 1, 2, \dots$) で定義し、学習領域を $X_1 \rightarrow X_2 \rightarrow \dots$ のように徐々に拡大していく。なお、訓練データは各小領域 X_n においてランダムに 100 個生成し、一つずつ追加学習される。

Fig. 4(a)(b) に、学習領域の拡大に伴う学習時間と記憶データ量の最大値の移り変わりを示す。これより、RBFN と RAN-LTM(N) では学習時間と記憶データ量が、 n に対して指数関数的に増加することがわかる。これは、隠れユニット数を J としたとき、線形方程式の求解に基づく方法における特異値分解の計算量が $O(J^3)$ 、活性値行列のサイズが $O(J^2)$ となることから説明できる。つまり、RBFN の隠れユニット数は訓練データ数に等しく、RAN-LTM(N) の隠れユニット数も領域の大きさ、つまり訓練データ数に比例する。よって、両者とも計算量と記憶データ量がそれぞれ $O(n^3)$ 、 $O(n^2)$ となるからである。RAN-LTM(N) では、記憶アイテム数と擬似訓練データの和が訓練データ数よりも少ない分、RBFN よりも計算量や記憶データ量は少ないが、学習領域が拡大されるにつれ、RAN-LTM(N) も深刻な問題を引き起こすといえる。

これに対し、RAN や RAN-LTM(O) の学習は最急降下法に基づいており、繰り返し 1 回当たりの計算量と記憶デー

(a) Time courses of learning time

(b) Time courses of maximum memory size

Fig. 4 Time courses of (a) learning time and (b) maximum memory size for the expansion of learning domains

タ量はそれぞれ $O(n^2)$ と $O(n)$ となる。RAN-LTM(O) の学習時間が長いのは、訓練データと記憶アイテムを同時に学習する分、誤差の収束に必要な繰り返し回数が多くなるためである。しかし、計算量のオーダーが提案手法に比べて小さいため、学習領域が大きいときは、最急降下法に基づいた RAN-LTM(O) のほうが提案手法である RAN-LTM(N) よりも、むしろ望ましい性質をもつといえる。なお RAN では、学習領域の拡大に対する学習時間と記憶データ量への影響はかなり小さいが、前述したとおり干渉にともなう忘却が深刻になるため、追加学習には適さないことに注意する。

4.3 時系列予測問題への適用

本節では、実用的な時系列予測問題を取り上げ、多次元関数の近似性能について評価を行なう。

4.3.1 マッキークラス時系列データ予測

(15) 式の微分方程式に基づいて生成されたマッキークラスの時系列データに対し、その予測性能を評価する。

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \quad (15)$$

ここで、 t は時刻、 τ は時間遅れである。 $\tau = 17$ とし、 $x(t-18)$ 、 $x(t-12)$ 、 $x(t-6)$ 、 $x(t)$ を入力として $x(t+6)$ を予測する。つまり、入力が 4 次元、出力が 1 次元の多次元関数の近似問題に相当する。なお、この時系列データの発生分布には偏りがあり、それが変動することをあらかじめ確認している。

$t = 118$ から $t = 1117$ の時系列データのうち最初の 500

Table 3 Prediction performance for Mackey-Glass time series

MODEL	Time (sec.)	Training Ave. Error	Test Ave. Error	Memory (KB)
RBFN	1012.6	0.0395	0.809	2024
RAN	67.2	0.173	0.172	1.5
RAN-LTM(O)	261.4	0.0451	0.0450	1
RAN-LTM(N)	0.2	0.0278	0.0271	1.9

点を訓練データ、残りの 500 点をテストデータとした。訓練データは 1 つずつ追加学習し、学習アルゴリズムが最急降下法に基づく RAN と RAN-LTM(O) に対しては、平均誤差が 0.01 以下になるか、100,000 回更新が行なわれるとつぎの訓練データの学習に移ることとする。学習時間 (Time)、訓練誤差 (Training Ave. Error)、テスト誤差 (Test Ave. Error)、記憶データ量 (Memory) を **Table 3** に示す。

Table 3 から、RAN-LTM(N) はほかの 3 つのモデルに比べてテスト誤差が小さく、優れた予測性能をもっていることがわかる。RAN, RAN-LTM(O), RAN-LTM(N) において生成された隠れユニットの数は 10 以下と少なかったため、記憶データ量はメモリベース RBFN に比べてかなり少なくなっている。RAN と RAN-LTM(O) の学習時間は RAN-LTM(N) に比べて長く、これは学習が最急降下法に基づいているためと考えられる。また、メモリベース RBFN のテスト誤差が非常に大きいのは、すべての訓練データを基底中心にセットするため隠れユニットが過剰となり、過学習が起こったためと考えられる。

4.3.2 浄水プラントにおける薬品注入量予測

浄水プラントにおける薬品注入量を予測する問題¹⁵⁾に適用し、追加学習性能を評価する。水質データと凝集剤を注入したときの微粒子の塊のようすを 10 次元の入力データとし、濁度を出力としている。なお、このデータは実測値であり、多くのノイズを含んでいる。

本研究では濁度が閾値よりも小さい定常データ 478 個を用いており、訓練データが 241 個、テストデータが 237 個である。訓練データはランダムに 1 つずつ提示する。なお、RAN と RAN-LTM(O) に対しては、平均誤差が 0.1 以下になるか、50,000 回更新が行なわれるとつぎの訓練データの学習に移ることとする。学習後の訓練データ、テストデータそれぞれに対する平均誤差、学習時間、学習に要する記憶データ量を **Table 4** に示す。

Table 4 より、前項の結果と同様、メモリベース RBFN は訓練データに対して高い近似精度をもつが、テストデータに対する精度がほかの 3 つのモデルに比べて低いことがわかる。また、RAN-LTM(O) は RAN よりもテスト誤差の点で優れているものの、学習時間はかなり長い。これに対し、RAN-LTM(N) は RAN や RAN-LTM(O) に比べてデータ記憶量がやや多いものの、テスト誤差は小さく、学習も RAN-LTM(O) に比べてかなり高速化されていることがわかる。

Table 4 Prediction performance for the Water Purification data

MODEL	Time (sec.)	Training Ave. Error	Test Ave. Error	Memory (KB)
RBFN	86.1	0	13.1	487.8
RAN	1.2	1.95	2.20	2.9
RAN-LTM(O)	57.6	1.67	1.79	4.5
RAN-LTM(N)	7.8	1.12	1.39	36.4

5. 結 論

これまで提案してきた追加学習モデルである Resource Allocating Network with Long-Term Memory (RAN-LTM)⁸⁾ に線形方程式の求解に基づく方法を適用し、追加学習の高速化を試みた。RAN-LTM は、訓練データの中から代表的なものをメモリに記憶し、それらを学習に利用することで追加学習で問題となる忘却を抑制する。この意味でメモリベース学習に属するが、従来手法に比べて、記憶データ量が少なく、高速な学習が可能である。これを確認するため、1 次元関数の近似、マッキィーグラス時系列データの予測や浄水プラントにおける薬品注入量予測に適用した。

テストデータに対する誤差、学習時間、実行に必要な記憶データ量に関して調べたシミュレーション実験において、提案した追加学習アルゴリズムの有効性を確認した。しかし、学習領域が非常に大きくなる問題では、生成される隠れユニット数が多くなるため、提案手法における特異値分解の計算量がかなり多くなる。この結果、最急降下法に基づく従来の RAN-LTM を用いたほうが、学習が速くなるケースもあるとわかった。これは、すべての隠れユニットに対して、記憶アイテムと擬似訓練データを生成して学習を行なっているためである。追加学習にともなう忘却は、活性度の高い隠れユニットの中心付近で主に起こることから、一回の追加学習で利用する記憶アイテムと擬似データがある程度限定できると考えられる。これによって、学習領域の大きさに依存しない高速な追加学習の実現が期待される。また、提案アルゴリズムでは、過去の記憶と訓練データに対する適応度を同程度にして追加学習を行なっている。しかし、このバランスはステップ (5)~(6) に重み付き最小二乗法を適用することで自由に設定でき、これにより、自然な形で過去の記憶を緩やかに忘却するメカニズムを導入できると思われる。これらに対する検討は、今後の課題としたい。

謝辞 本研究は、文部科学省科学技術研究費補助金 若手研究 (B) の助成によって行なわれた。

参 考 文 献

- 1) G.A. Carpenter and S. Grossberg: The ART of adaptive pattern recognition by a self-organizing neural network, *IEEE Computer*, **21**-3, 77/88 (1988)
- 2) R. Rosipal, M. Koska and I. Frakas: Prediction of chaotic time-series with a resource allocating RBF network, *Neural Processing Letters*, **10**-26 (1997)
- 3) S. Weaver, L. Baird and M. Polycarpou: An analytical framework for local feedforward networks, *IEEE Transac-*

- tions on Neural Networks, 9-3, 473/482 (1998)
- 4) 小谷, 宮田, 小澤, 赤澤: 変化する環境への適応を目指したモジュール構造ネットワークによる音響診断, 計測自動制御学会論文集, 36-9, 797/803 (2000)
 - 5) H. Nakayama and M. Yoshida: Additional learning and forgetting by potential method for pattern classification, *Proc. Int. Conf. on Neural Networks 97*, 1839/1844 (1997)
 - 6) C. G. Atkeson, A. W. Moore, and S. Schaal: Locally weighted learning, *Artificial Intelligence Review*, 11, 75/113 (1997)
 - 7) K. Yamauchi, N. Yamaguchi and N. Ishii: Incremental learning methods with retrieving of interfered patterns, *IEEE Trans. on Neural Networks*, 10-6, 1351/1365 (1999)
 - 8) 小林, 小澤, 阿部: 長期記憶を導入した階層型ニューラルネットの追加学習アルゴリズム, 計測自動制御学会論文集, 38-9, 792/799 (2002)
 - 9) J. Platt: A resource allocating network for function interpolation, *Neural Computation*, 3, 213/225 (1991)
 - 10) T. Poggio and F. Girosi: Networks for approximation and learning, *IEEE Trans. on Neural Networks*, 78-9, 1481/1497 (1990)
 - 11) M. J. L. Orr: Introduction to radial basis function networks, *Tech. Report of Institute for Adaptive and Neural Computation*, Division of Informatics, Edinburgh University (1996)
 - 12) S. Haykin: *Neural Networks - A Comprehensive Foundation* (2nd Ed.), Prentice Hall (1999)
 - 13) K. Okamoto, S. Ozawa and S. Abe: A fast incremental learning algorithm of RBF networks with long-term memory, *Proc. Int. Conf. on Neural Networks 2003*, 102/107 (2003)
 - 14) S. Schaal and C. G. Atkeson: Constructive incremental learning from only local information, *Neural Computation*, 10-8, 2047/2084 (1998)
 - 15) 圓沸, 馬場, 松崎, 依田: ニューラルネットを用いたプラント運転ルールの抽出に関する研究-浄水プラント薬品注入操作への適用, 電学論 D, 111-D-1, 20/28 (1991)

阿部 重夫 (正会員)

1972年京都大学大学院修士課程(電気工学専攻)修了。企業の研究所を経て97年より神戸大学工学部教授, 99年より神戸大学大学院教授となり現在に至る。78年から一年間米国 Texas 大学客員研究員, ニューラルネット, ファジィシステム, サポートベクトルマシンの研究に従事。工学博士。84年, 95年電気学会論文賞受賞。著書 *Pattern Classification: Neuro-fuzzy Methods and Their Comparison* (Springer) ほか。情報処理学会, 電気学会, システム制御情報学会, IEEE, INNS 等の会員。

[著者紹介]

岡本圭介

2002年神戸大学電気電子工学科卒業, 同年神戸大学自然科学研究科修士課程電気電子工学専攻入学, 現在に至る。ニューラルネットの追加学習に関する研究に従事。

小澤 誠一 (正会員)

1989年神戸大学大学院工学研究科修士課程(計測工学専攻)修了。同年奈良工業高等専門学校情報工学科助手, 93年大阪教育大学教育学部教養学科情報科学専攻助手, 同講師を経て, 2000年1月より神戸大学大学院自然科学研究科助教授となり, 現在に至る。ニューラルネット, 機械学習, パターン認識の研究に従事。博士(工学)。電子情報通信学会, 電気学会, システム制御情報学会, IEEE, INNS の会員。