1C1-1

ラジアル基底関数ネットにおける追加型自律学習アルゴリズムの改良

Improvement of Incremental Autonomous Learning Algorithm for Radial Basis Function Networks

○学　中坂　翔　（神戸大）　正　小澤　誠一　（神戸大）

Sho NAKASAKA, Kobe University

Seiichi OZAWA, Kobe University

In this paper, we propose to implement an pruning algorithm to an autonomous incremental learning model called Automated Learning algorithm for Resource Allocating Network (AL-RAN). The proposed pruning algorithm determines whether Radial Basis Functions (RBF) are useful or not by activation level of them and remove unuseful RBFs. By implementing this algorithm to AL-RAN, we reduce the number of basis and improve the learning time. From the experimental results, we confirm that the above functions work well and the efficiency in terms of learning time is improved without sacrificing the recognition accuracy as compared with the previous version of AL-RAN.

Key Words: Pattern recognition, Incremental learning, Autonomous learning

1. はじめに

パターン認識 [1] における識別器の学習方式として，一括で訓練データが与えられるバッチ学習と，逐次的に訓練データが与えられ，学習したデータを破棄する追加学習の二つがある．近年，追加学習の重要性が認識されるようになり，ラジアル基底関数 (RBF) ネットワーク [2] を追加学習に拡張する試みも盛んに行われている [3].

RBF ネットワークにおける高性能な識別器の実現には適切なパラメータ選択が必要となる．また，データによっては正規化が有効なときとそうでないときがあることが経験的に分かっている．しかし，追加学習環境では将来得られる訓練データは未知であるため，適切なパラメータの設定や正規化の必要性を事前に判断することは困難である．よって，学習システム自身がオンラインで適切にパラメータの設定を行うことが望まれる．このように，外部の教師の介入を排除し，学習システムが自律的にパラメータの設定を行うような学習方式を「自律学習」[4] と呼ぶ.

Platt らが提案した Resource Allocating Network (RAN)[5] は，RBF ネットワークを追加学習に拡張した学習モデルであるが，追加学習で発生する知識の忘却を完全に抑制できるわけではない．そこで，過去の訓練データで代表的なものを長期記憶に保存し，それらのうち有効なものを想起して，訓練データと一緒に学習することで，安定した追加学習を可能とした Resource Allocating Network with Long-Term Memory (RAN-LTM)[6] が提案されている．しかし，RAN-LTM には RBF の基底幅をはじめとするいくつかのパラメータを適切に設定する必要があり，正規化などの前処理の必要性もあからじめ教師によって指定されることが前提となっている．そこで，RAN-LTM に対して RBF の基底幅の決定，正規化判定の必要性の決定を自動で行う自律学習アルゴリズムが望まれ，Automated Learning algorithm for Resource Allocating Network (AL-RAN)[7, 8] を提案してきた．AL-RAN では正規化の必要性が判定可能になるまで訓練データを収集し，その訓練データを用いて正規化の有無を決定し，また，オンラインで基底幅を決定する自律学習を実現している．しかし，AL-RAN では大規模な訓練データを学習する場合，必要以上に多くの RBF 基底が配置されてしまい，計算コストが高くなる問題があった.

そこで本稿では，活性化することが稀である，識別に影響を与えないような基底を活性度を基に削除することで，基底数の増大を抑制することを提案する．初期学習部で収集したデータ数を基底を不要と判定するまでに必要な学習ステップ数とし，その期間の間に活性化しない基底を今後も活性化することは無い基底としてネットワークから削除する.

本稿の構成は次の通りである．**2.** では従来手法である AL-RAN について説明し，**3.** では提案する基底削除アルゴリズムについて述べる．**4.** で実験により得られた結果を示し，**5.** で結論を述べる.

2. 追加型自律学習モデル

追加型自律学習モデルである AL-RAN は RAN-LTM を拡張し，基底幅と正規化の有無を自律的に決定するメカニズムを付与することで外部の教師の関与無しに安定した自律学習を実現したモデルである．学習は初期学習部，追加学習部に分けられる.

2.1 初期学習部

一般に，訓練データの学習や識別を行う前に正規化を施すことで識別器の汎化能力が向上する場合が多いが，逆に汎化能力が低下してしまう場合もあることが経験的に知られている．そこで初期学習部では，正規化の必要性が判定可能となるまで訓練データの収集を行い，収集したデータを用いて正規化の必要性を判定し，必要と判定された場合は正規化を行う．また，収集したデータから，初期ネットワークを生成する.

2.1.1 正規化判定

まず，逐次的に与えられる訓練データを収集する．N 番目の入力 $\boldsymbol{x}_N$ を得たとき，次のように表される平均ベクトル $\boldsymbol{m}_N$，標準偏差ベクトル $\boldsymbol{s}_N$ を求める.

$$\boldsymbol{m}_N = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_N \tag{1}$$

$$\boldsymbol{s}_N = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{m}_N)^T(\boldsymbol{x}_n - \boldsymbol{m}_N)} \tag{2}$$

式 (1),(2) を用いて次のように正規化を行う.

$$\hat{x}_i = \frac{x_i - m_i}{s_i} \tag{3}$$

ここで x_i, m_i, s_i は，それぞれ $\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{s}$ の i 次元目の要素である．式 (3) により得られた正規化データと原データの双方において交差検定法により平均認識率を求め，比較することで正規化の必要

性を判定する．この判定を行うのは平均認識率が収束したときが適切である．しかし，平均認識率を求める交差検定法には多大な計算コストが必要となるため，平均認識率に代わりクラス分離度をデータ収集終了基準として用いる．クラス分離度はクラスの分離性を示しており，一般にクラス分離度が高いほど識別が容易なデータ分布であると言える．そのため，クラス分離度は認識率に密接に関係し，認識率に代わる指標として用いる．したがって，平均認識率の収束の判定として，正規化データと原データのクラス分離度の差分値の収束の判定を行い，その収束を確認した時点でデータ収集を終了し，正規化の必要性の判定を行う．

クラス分離度の差分値の収束判定を以下に示す．クラス分離度 P_N は，原データと式 (3) により得られた正規化データにおいて，次式に定義するクラス内分散 $\boldsymbol{S}_W$，クラス間分散 $\boldsymbol{S}_B$ を用いて次のように求める．

$$\boldsymbol{S}_W \stackrel{\text{def}}{=} \frac{1}{N}\sum_{c=1}^{C}\sum_{j=1}^{N_c}(\boldsymbol{x}_{cj}-\bar{\boldsymbol{x}}_c)(\boldsymbol{x}_{cj}-\bar{\boldsymbol{x}}_c)^T \tag{4}$$

$$\boldsymbol{S}_B \stackrel{\text{def}}{=} \frac{1}{n}\sum_{c=1}^{C}N_c(\bar{\boldsymbol{x}}_c-\bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_c-\bar{\boldsymbol{x}})^T \tag{5}$$

$$P_N = \mathrm{tr}\{\boldsymbol{S}_W^{-1}\boldsymbol{S}_B\} \tag{6}$$

このとき，$\bar{\boldsymbol{x}}_c$ はクラスラベル c をもつ訓練データの平均ベクトル，$\bar{\boldsymbol{x}}$ は全訓練データにおける平均ベクトルである．$\boldsymbol{x}_{cj}$ はクラスラベル c をもつ j 番目の訓練データであり，クラスラベル c をもつ訓練データの総数を N_c とする．ここで，原データにおけるクラス分離度を P_N^{raw}，正規化データにおけるクラス分離度を P_N^{nor} とすると，その差分値 ΔP_N を次のように定義する．

$$\Delta P_N = \frac{|P_N^{\mathrm{raw}}-P_N^{\mathrm{nor}}|}{\max\{P_N^{\mathrm{raw}},P_N^{\mathrm{nor}}\}} \tag{7}$$

式 (7) により得られるクラス分離度の差分値が収束，つまり次に示す時間変動率 $\Delta^2\bar{P}_N$ が微小となれば，十分に訓練データを収集できたと判定する．したがって，データ収集終了条件は閾値 θ_s を用いて次のように定義する．

$$\Delta^2\bar{P}_N = \frac{1}{\tau}\sum_{k=1}^{\tau}\frac{\|\Delta P_{N-k+1}-\Delta P_{N-k}\|}{\max\{\Delta P_{N-k+1},\Delta P_{N-k}\}} \le \theta_s \tag{8}$$

ここで τ は，直前 τ 回の学習ステージの平均を見る役割をしている．これにより，よりロバストな判定をすることができる．また訓練データによっては，分離度の差分値の収束を確認せずとも明らかに差がある，つまり認識率にも大きな差がある場合がある．このような場合も正規化の必要性の判定は容易であるため，正規化判定を行ってもよいと考えられる．そこで，分離度の差分値と閾値 θ_r を用いて，第二の終了条件として以下を定義する．

$$\Delta\bar{P}_N = \frac{1}{\tau}\sum_{k=1}^{\tau}\Delta P_{N-k+1} \ge \theta_r \tag{9}$$

しかし，収集したデータの総数が少ないとき，分離度は非常に大きく不安定な値をとる．そのため，式 (8) において次式のように緩い閾値を設定し，分離度に信頼性が得られるまでは差の大きさの判定は行わない．

$$\Delta^2\bar{P}_N \ge 2\theta_s \tag{10}$$

したがって，式 (10) を満たした上で，式 (8) または式 (9) どちらかを満たすことが終了条件となる．

上記基づきデータ収集を行った後，交差検定法により最近傍法における原データの平均認識率 R_{raw} と正規化データの平均認識率 R_{nor} を求め，R_{raw} の方が大きければ正規化は不要，R_{nor} の方が大きければ正規化が有効と判定する．

2.1.2 初期ネットワーク生成

ここでは，初期ネットワーク生成アルゴリズムについて述べる．生成を目指すネットワークとは，クラス境界付近には識別面を正しく表すために局所的な反応をもつ RBF 基底を密に，クラス境界から離れたところへは大域的な反応をもつ少数の RBF 基底が配置されたものである．これにより，一定の認識率を維持した上で RBF 基底の増加を抑えることができ，結果として計算コストを削減することが可能になると考えられる．以下にそのネットワークを実現する処理を説明する．

収集した訓練データにおいて，クラスごとに最近傍同クラスデータまでの距離の平均 $\bar{d}_c$ を次の式で求める．

$$\bar{d}_c = \frac{1}{J}\sum_{j=1}^{J}(\min_l\|\boldsymbol{x}_j-\boldsymbol{x}_l\|) \quad (\boldsymbol{T}_j=\boldsymbol{T}_l) \tag{11}$$

クラス c において，データ間距離が $2\bar{d}_c$ より短い場合，それらの訓練データは類似していると判定する．その手順として，まず収集した中からランダムに選び出したデータを短期記憶 $\boldsymbol{X}'$ に格納する．格納を全データにおいて試行するが，格納しようとするデータから $2\bar{d}_c$ 以下の距離にある同クラスデータが既に格納されていた場合，格納は行わない．これにより類似したデータを除くことができる．上記の処理により $\boldsymbol{X}'$ に格納されたデータからランダムに選び出し，そのデータが k 番目に生成されたものとすると，他クラス最近傍データまでの距離を以下の式で求める．

$$d_k^* = \min_m(\|\boldsymbol{x}_k-\boldsymbol{x}_m\|) \quad (\boldsymbol{T}_k\neq\boldsymbol{T}_m) \tag{12}$$

選ばれたデータから，d_k^* より短い距離にあるデータを削除する．この処理により，クラス境界から離れたデータをまとめることができる．

$\boldsymbol{X}'$ 内に残った k 番目のデータを $(\boldsymbol{x}_k,\boldsymbol{T}_k)$ とすると，以下のように RBF 基底を追加する．

$$\boldsymbol{c}_k=\boldsymbol{x}_k,\ \ \boldsymbol{w}_j=\boldsymbol{T}_k,\ \ \sigma_k=d' \tag{13}$$

ここで d' は $\boldsymbol{X}'$ 内の最近傍データまでの距離である．上記の処理により，学習において効率的なネットワークの生成ができる．

2.2 追加学習部

追加学習部では，RBF 基底を追加する際に基底幅をオンラインで適応的に決定し，学習を行う対象に依らない安定した自動追加学習を実現する．入力 $\boldsymbol{x}$ があったときの j 番目の隠れユニット（RBF 基底）の活性度を y_j とすると，以下のように表される．

$$y_j = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_j\|^2}{2\sigma_j^2}\right) \tag{14}$$

ここで，$\boldsymbol{c}_j$ は RBF 基底の中心ベクトル，σ_j は RBF 基底の基底幅である．このとき，k 番目の出力ユニットの出力 z_k は，j 番目の RBF 基底と k 番目の出力ユニットの結合荷重 w_{jk}，RBF 基底の総数 J を用いて以下のように計算される．

$$z_k = \sum_{j=1}^{J}w_{jk}y_j \tag{15}$$

訓練データ $(\boldsymbol{x},\boldsymbol{T})$ が入力されると，出力誤差 $E=\|\boldsymbol{T}-\boldsymbol{z}\|$，最も活性度が高くなる RBF 基底中心 $\boldsymbol{c}^*$ までの距離 $d^*=\|\boldsymbol{x}-\boldsymbol{c}^*\|$ が計算され，以下の条件を満たせば未学習領域に入力されたと判定される．

$$E>\varepsilon \ \text{ and } \ d^*>\sigma^* \tag{16}$$

ここで，ϵ は許容誤差を，σ^* は活性度が最も高い RBF 基底の基底幅を表す．式 (16) を満たすと誤差の修正のため基底を一つ追

加するし，$J \rightarrow J+1$ として以下のように新たな RBF 基底と記憶アイテム $(\hat{x}_J, \hat{T}_J)$ を生成する．

$$\boldsymbol{c}_J = \boldsymbol{x}, \quad \boldsymbol{w}_J = \boldsymbol{T} - \boldsymbol{z} \tag{17}$$

$$(\hat{\boldsymbol{x}}_J, \hat{\boldsymbol{T}}_J) = (\boldsymbol{x}, \boldsymbol{T}) \tag{18}$$

RBF 基底の基底幅 σ_J は，最近傍基底中心までの距離 $d = \|\boldsymbol{x} - \boldsymbol{c}\|$ を用いて，次のようにする．

$$\sigma_J = d \tag{19}$$

式 (16) を満たさない場合，結合荷重の修正だけで出力誤差を最小化できると判定される．結合荷重の修正は，訓練データに対する RBF 基底出力行列 $\boldsymbol{\Phi}$ とクラスラベル行列 $\boldsymbol{T}$ を用いて，以下の二乗誤差の和 E^2 を最小にするように行われる．

$$E^2 = \|\boldsymbol{T} - \boldsymbol{\Phi}\boldsymbol{W}\|^2 \tag{20}$$

式 (20) を最小化する結合荷重行列 $\boldsymbol{W}$ を次のように求める．

$$\boldsymbol{W} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{T} \tag{21}$$

また，特異値分解により $\boldsymbol{\Phi}$ を

$$\boldsymbol{\Phi} = \boldsymbol{U}\boldsymbol{H}^{-1}\boldsymbol{V}^T \tag{22}$$

と変換でき，式 (21) を次のように書き換えることができる．

$$\boldsymbol{W} = \boldsymbol{V}\boldsymbol{H}^{-1}\boldsymbol{U}^T\boldsymbol{T} \tag{23}$$

ここで，$\boldsymbol{U}$ は直交行列，$\boldsymbol{H}$ は固有値対角行列，$\boldsymbol{V}$ は直交行列である．この最小二乗法を用いると，RBF 基底数が J であるとき，計算コストのオーダーは特異値分解による $O(J^3)$ となるため，基底数が増大するにつれ，計算コストは大きく増加していくこととなる．

式 (21) に基づき結合荷重の修正を行っても，誤差を小さくできない場合がある．このとき，RBF 基底が不足していると判定され，式 (17)〜(19) に基づいて，RBF 基底と記憶アイテムが生成される．この際，最近傍の RBF 基底に対し，図 1 のように，その基底幅を新たに追加した RBF 基底中心までの距離に修正する．

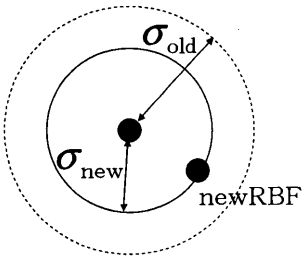

Fig. 1: Adjustment of RBF Width

上記のような基底幅のスケーリングが過度に行われると，過去の知識を忘却する，いわゆる「負の干渉」が起こりやすい．しかし，クラス境界近傍ではこの基底幅のスケーリングを避けることは難しい．このため，クラス境界近傍に大きな基底幅を持つ基底を配置しないことが重要であり，初期学習部における初期ネットワーク生成が重要となる．

3. RBF 基底の削除による AL-RAN の改良

従来の AL-RAN の学習アルゴリズムでは，既に配置されている RBF 基底の基底幅内に新たに RBF 基底が生成される場合，既に配置されている RBF 基底の基底幅を縮小することで局所反応性を保証しようとする．しかし，これにより，それまで反応していた領域への入力に対して活性化することがなくなり，過去の知識が忘却されてしまう．特に，複数のクラスがオーバーラップしているような領域では頻繁に知識の忘却と新たな基底の生成が行われ，結果として図 2 のように必要以上に多くの基底が配置され，冗長なネットワークが構成されてしまうことで，多くの計算コストがかかることとなる．

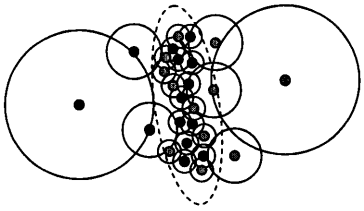

Fig. 2: Unuseful RBFs

そこで本稿では，RBF 基底の削除を行うアルゴリズムを提案する．このアルゴリズムでは，既に配置されている RBF 基底が長期に渡って活性化しない場合，今後も学習，識別に大きく影響しない基底である冗長な基底であると判定して削除を行い，基底数の抑制を図る．

基底の削除は以下のように行う．訓練データ $\boldsymbol{x}$ が与えられたとき，式 (14) より得られる活性度 y_j が，次式のように閾値 θ_a を越えない値を取ると，その基底は訓練データ $\boldsymbol{x}$ に対して行う学習，識別への関与はわずかであると判定する．

$$y_j \leq \theta_a \tag{24}$$

このとき，閾値 θ_a はユーザーパラメータであるが，微小な値として設定するため，学習する対象に影響されにくく，設定は比較的容易である．

基底が不要であると判定するまでに必要な学習ステップ数としては，データ分布をある程度表すだけのデータが与えられると期待されるのに必要なデータ数が適切である．そこで，環境を読み取るのに十分なデータ数として収集した初期訓練データ数 N を，基底が不要であると判定するまでの観測期間 T として設定する．j 番目の基底の連続不活性回数を t_j とし，学習を行うごとに次のように変化させる．

$$\textbf{If } y_j \leq \theta_a, \quad t_j = t_j + 1$$

$$\textbf{If } y_j > \theta_a, \quad t_j = 0$$

上記の処理の後，$t_j = T$ となった場合，j 番目の基底は今後も学習，識別に大きく関与することは無いと判定し，ネットワークから削除する．削除を行った後は，$j+1, \cdots, J$ 番目の基底，記憶アイテムをそれぞれ $j, \cdots, J-1$ 番目の基底，記憶アイテムとして扱うようにする．したがって，基底数や記憶アイテムの総数は $J-1$ となる．ここで削除される基底はネットワークにおいて微小な反応領域しか持たないため，削除を行った後に結合荷重の修正を行う必要は無い．

4. 評価実験

本節では，機械学習に用いられるベンチマークデータセットである UCI データセット，IDA データセットを用いて，従来の AL-RAN と基底削除を行うメカニズムを導入した AL-RAN（以

下では，AL-RAN(P) と記す）の比較を行い，性能を評価する．実験は全データを二分割し，片方を訓練データ，もう片方をテストデータとした場合と，訓練データ，テストデータを逆にした場合の 2 通りの割り当てを行う．それぞれの場合において訓練データを提示する順序をランダムに並び替えて入力する試行を 25 回ずつ行った計 50 回の試行の平均と標準偏差を示している．また，正規化を行った方が高い汎化能力が得られるデータセットには正規化を行った．学習には，クラスがオーバーラップする領域が大きく，AL-RAN において冗長な基底が配置されやすいと経験的に分かっているデータセットである Thyroid, Banana を使用し，その詳細を表 1 に示す．ここで Norm. の項では，正規化を行った方がよい性能を得られるデータセットには○を，それ以外には×を記した．生成された基底数，追加学習に要した時間，テストデータに対する認識率を表 2 に示す．ここでは，$\theta_a = 10^{-4}$ とした．基底削除メカニズムにより，認識率を低下させることなく基底数の削減ができていることが望ましい．

Table 1: Evaluated Dataset

Dataset	# Dim	# Class	# Data	Norm.
Thyroid	21	3	7200	○
Banana	2	2	5300	×

Table 2: Performance of AL-RAN and AL-RAN(P)

(a)Thyroid

Thyroid	AL-RAN	AL-RAN(P)
#RBFs	325.0 ± 109.6	248.2 ± 58.2
Learning Time [s]	357.64 ± 301.54	168.3 ± 104.1
Test Acc. [%]	93.9 ± 2.8	93.5 ± 3.0

(b)Banana

Banana	AL-RAN	AL-RAN(P)
#RBFs	484.4 ± 25.9	443.1 ± 82.8
Learning Time [s]	551.84 ± 127.31	501.3 ± 184.2
Test Acc. [%]	84.74 ± 0.78	84.81 ± 0.82

結果より，AL-RAN(P) は Thyroid，Banana の両方において，汎化能力を維持したまま，基底数を削減したことで計算コストの削減に成功していることが分かる．これは，オーバーラップ領域での識別は困難であり，教師と予測の誤差が大きくなるため，AL-RAN ではわずかな反応領域しか持たない多数の基底が配置されていた．そのような基底の削除により，冗長なネットワークが構成されることを回避できたと考えられる．また，Banana における AL-RAN(P) の結果の分散が大きいのは，Banana では初期学習で収集するデータ数が安定してないためであり，基底の削除がほとんど行われないケースが存在するためである．そのため，基底の削除を行うまでの期間の適切な設定方法を再検討する必要がある．

5. まとめ

追加型自律学習を行う学習モデルである AL-RAN に，適応的に基底の削除を行うメカニズムを導入することを提案した．基底が一定期間活性化しない場合，その基底を，大きく活性化することはわずかである学習，識別にほとんど関与しない冗長な基底であると判定し，それを削除することで，汎化能力を維持しつつ，基底数の削減により高速化を図った．

機械学習のベンチマークデータセットである UCI データセット，IDA データセットを用いて評価した結果，従来の AL-RAN と比べて，同等以上の汎化能力を維持したまま，学習の高速化に成功した．ただし，活性化しない基底を不要と判定するまでの学習ステップ数を初期収集データ数として設定した場合，初期収集データ数に削除性能が大きく左右されるため，今後の課題としてデータセットの種類を選ばず適切な削除までの学習ステップ数を設定できるような方法の検討が必要となる．

文献

[1] 石井健一郎，上田修功，前田英作，村瀬　洋，”わかりやすいパターン認識”，オーム社，1998.

[2] Poggio, T. and Girosi, F., “Networks for Approximation and Learning,” *IEEE Neural Networks*, vol.78, no.9, pp.1481–1497, 1990.

[3] Ozawa, S., Toh, S.-L., Abe, S., Pang, S. and Kasabov, N., “Incremental Learning of Feature Space and Classifier for Face Recognition,” *Neural Networks*, vol.18, no.5–6, pp.575–584, 2005.

[4] Roy, A., “Artificial Neural Networks - A Science in Trouble,” *SIGKDD Explorations*, vol.1, pp.33–38, 2000.

[5] Platt, J., “A Resource-Allocating Network for Function Interpolation,” *Neural Computation*, vol.3, pp.213–225, 1991.

[6] Okamoto, K., Ozawa, S. and Abe, S., “A Fast Incremental Learning Algorithm of RBF Networks with Long-Term Memory,” *Int. Joint Conf. on Neural Networks,* pp.102–107, 2003.

[7] Tabuchi, T., Ozawa, S. and Roy, A., “An Autonomous Learning Algorithm of Resource Allocating Network,” *Intelligent Data Engineering and Automated Learning*, pp.134–141, 2009.

[8] Ozawa, S., Nakasaka, S. and Roy, A., “An Autonomous Incremental Learning Algorithm of Resource Allocating Network for Online Pattern Recognition,” *Int. Joint Conf. on Neural Networks*, pp.706–713, 2010.