

ノイズのある環境下でオンライン学習が可能な  
自己増殖型ニューラルネットワークを用いた連想記憶モデル  
Associative Memory for Online Learning in Noisy Environment Using SOINN

須藤 明人†  
Akihito Sudo

佐藤 彰洋†  
Akihiro Satou

長谷川 修‡  
Osamu Hasegawa

## 1. まえがき

連想記憶の機能を智能ロボット等に持たせる研究[1]が近年行われている。ニューラルネットワークを利用した連想記憶の研究の歴史は長いですが、以下に述べる理由で従来のモデルは智能ロボット等には不十分である。

実世界で自律的に活動する智能ロボット等のシステムに適用する連想記憶モデルは、追加学習、ノイズ耐性、実数値データの取り扱い、多対多の連想を行えることが必要である。まず、非正常な環境等のシステムに必要な知識を事前に全て列挙することが不可能な環境ではシステムは知識を追加的に得ていかねばならない。その際、既存の知識が破壊されることなく新しい知識をオンラインで学習できなければならない。また、実世界からシステムが取得する情報には多かれ少なかれノイズが必ず含まれておりノイズ耐性は必須である。同じく実世界から得られる情報は2値ではなく実数値であるので実数値をそのまま記憶することが望ましい。実世界においてシステムに必要な知識が1対1の連想対だけでは限らないので多対多の連想が行えなければならない。その際システムが何対何の連想を行えるかが固定されてしまうのではなく追加される知識によって、任意の整数  $a, b$  について  $a$  対  $b$  の連想を同時に行えることが求められる。

従来の連想記憶モデルで追加学習を十分に行えるモデルは我々の知る限り存在しない。従来の連想記憶モデルは2種に大別される。すなわちホップフィールドネットワークや Bidirectional Associative Memory のように知識を複数の結合荷重に分散して記憶するタイプと、Self-organizing Map (SOM) を連想記憶が扱えるように拡張した[2]のように競合学習を利用するタイプである。知識を分散して記憶する前者のタイプは追加学習によって全ての結合荷重が変化するため既存の知識が破壊されてしまうという深刻な問題を抱える。French が指摘したように知識を分散して記憶するモデルがこの問題を避けることは難しい[3]。

一方、競合学習を利用するモデルはひとつのノードでひとつの知識を記憶するため比較的追加学習に向いている。SOM を利用した連想記憶モデルを追加学習のために拡張したモデル[4]では、結合荷重が学習データに十分近くなったノードはそれ以降の学習において変化しないため既存の知識を破壊することなく新たな知識を学習することが可能である。しかし、このモデルが学習によって獲得可能な連想対の数の上限は事前に決めたノードの数に依存して決まってしまうという問題がある。そのため、事前に学習すべき知識の量がわかっていない環境にはこのモデルは不向きである。もし事前に決めたノードの数が学習すべき連想対の数に比べて少ないと、途中から連想対を追加することが

できなくなりアルゴリズムは無限ループに陥ってしまう。逆に学習すべき知識に対して多すぎるノードを与えると、本来は不要な計算時間とメモリを浪費してしまう。このように、学習すべき知識の量を事前に決定することができない非正常な環境下で追加学習を行うことは既存の手法では困難である。

本論文では上述の追加学習の問題を克服し、かつ智能ロボット等の連想記憶に求められるノイズ耐性、実数値データの取り扱い、多対多の連想の機能を併せ持つ連想記憶モデル Associative Memory with SOINN (SOINN-AM) を提案する。提案手法は SOINN[5] を連想記憶モデルに拡張したものであり SOINN と同様に学習データの量に応じて自動的にノードが増殖する競合学習型の連想記憶モデルであるため事前にノードの数をユーザーが決めておく必要がない。そのため追加学習を行ったときに既存の知識が破壊されることが無いだけでなく、ノードの不足で知識の追加ができなくなることやノードが多すぎて計算時間やメモリを浪費することを避けることができる。

提案手法は非正常な環境下で適切に追加学習を行えるだけでなく、ノイズ耐性、実数値データの取り扱い、多対多の連想を行うこともできる。従来手法でこれら全ての機能を併せ持つものは我々の知る限り存在しない。実環境におけるノイズには図1に示したような2種類がある。図1(a)の左側のデータは本来学習すべきデータや想起キーとして与えられたデータにノイズが混じったデータである。図1(b)は本来の学習データや想起キーとは全く無関係なデータを表す。図1(a)の左側のデータが入力されたときには図1(a)の右側のデータが入力された判断できることが望ましい。図1(b)が入力されたときには学習フェーズであれば知識に加えてはならず、想起フェーズであれば未知データであると判断できなければならない。提案手法はこれらを行うための機能を備えているが、この2タイプのノイズに対応できる連想記憶モデルは従来手法には無い。また、提案手法は実数値データを扱うことができる。実数値データを扱うことを目的とした連想記憶モデルは従来研究にもあるが、それらは追加学習を行えないため事前にすべてのデータを列挙できない環境での利用には適さない。多対多の連想については、先述のとおり任意の整数  $a, b$  について  $a$  対  $b$  の連想を同時に行えることが望ましいが、提案手法はこれも行える。従来の多対多の連想記憶モデルは[6]が代表的であるが、これは事前に何対何の連想を行うかを定めておかなければならず、例えば事前に3対3の連想を行うために生成したネットワークで5対5や10対10の連想を行うことはできない。

† 東京工業大学 総合理工学研究科

‡ 東京工業大学 像情報工学研究施設

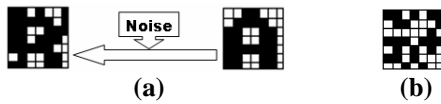


図1: 2種類のノイズ

## 2. 提案手法

図2に示したとおり、提案手法は入力層と競合層を持つニューラルネットワークモデルである。提案手法のアルゴリズムは学習フェーズと想起フェーズに分かれている。学習フェーズでは学習すべき連想対に対応したノードとエッジが学習データの数に応じて複数生成され、各連想対ごとにノードのクラスタが形成される。なお、競合層のノード間に存在するエッジによってつながっているノード同士を同じクラスタと見なす。また、想起フェーズで連想キーが入力されると、学習によって競合層に生成されたノードのうち連想キーに対応するものが発火し、そのノードが連想結果を出力する。

### 2.1 学習フェーズ

学習フェーズでは学習すべき連想対として実数値ベクトルのペアが入力される。学習データが入力されると、入力層で連想対のベクトルを結合してひとつのベクトルとし、さらにわずかな摂動として平均0、分散 $\sigma_r^2$ の正規分布から発生させたベクトルを加えて競合層への入力信号とする。例えば、Fを連想のキーとなるM次元ベクトル、Rを連想のキーから想起されるN次元ベクトルとすると、入力層は入力された2つのベクトルをM+N次元のベクトルXに統合し、摂動を加えた上で競合層に入力することになる。競合層への入力にわずかな摂動を加えるのは、競合層でひとつの連想対についてクラスタを形成するためである。以下で述べるように、提案手法は各連想対のクラスタを形成することにより、図1(a)および(b)に示したノイズへの耐性を獲得している。

競合層にデータが入力されると、本節の最後に述べる方法でそのデータが既存のノードと同じクラスタに属するかどうか判定される。もし既存のノードと同じクラスタには属しないと判定されれば、入力されたベクトルを重みベクトルとして持つノードが競合層に新たに生成しその入力データについての処理を終了する。もし既存のノードと同じクラスタに属すると判定されたなら、入力データを重みベクトルを持つノードは生成せずに、次に述べる方法で競合層のノードやエッジをアップデートする。まず、入力データに最も近い重みベクトルを持つノード(第1勝者)と2番目に近い重みベクトルを持つノード(第2勝者)を同じクラスタとするためにそれらの間に年齢1のエッジを生成する。もしも既にエッジが存在していればそのエッジの年齢を1に戻す。第2勝者以外のノードと第1勝者との間にエッジがあればそのエッジの年齢を1増やす。このときあらかじめ定めたエッジの寿命 $\Lambda_{edge}$ を越えたエッジは削除される。また第1勝者の重みベクトル $W_r$ および第1勝者とエッジでつながっているノードの重みベクトル $W_i$ にそれぞれ次式の $\Delta W_r$ と $\Delta W_i$ を加える。

$$\Delta W_r = \frac{1}{\chi_r} (I_c - W_r)$$

$$\Delta W_i = \frac{1}{100\chi_r} (I_c - W_i)$$

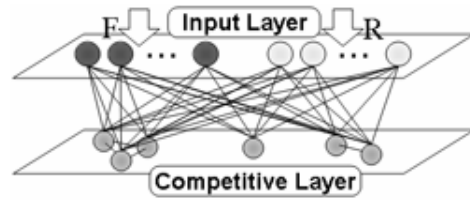


図2: SOINN - AMの構造

ここで、 $\chi_r$ は第1勝者が勝者になった回数であり、 $I_c$ は競合層に入力されたベクトルである。そして、学習回数がパラメタ $\lambda$ の整数倍であるならばエッジを持たないノードの削除を行う。エッジを持たないノードを削除することで、ノイズとして誤って入力された学習データとはまったく無関係なデータを削除することができる。最後に、各クラスタの重みベクトルの平均を重みベクトルとして持つノードを全てのクラスタについて生成する。このノードは想起フェーズでのみ使われる特殊なノードであり以下でプロトタイプノードと呼ぶ。このプロトタイプノードを推論フェーズで利用することで、学習データに混じったガウスノイズを除去することができる。

本節の最後に、入力された学習データが既存のノードと同じクラスタに属するかを判断する方法を述べる。学習データが入力されると、第1勝者および第2勝者のそれぞれの類似度閾値 $d$ を次式で計算する。

$$d = \begin{cases} \max_{k\text{-th node} \in N} \|W - W_k\| & (\text{if } N \neq \emptyset) \\ \min_{k\text{-th node} \in A} \|W - W_k\| & (\text{if } N = \emptyset) \end{cases}$$

ここで $N$ は第1(2)勝者とエッジで結ばれているノードの集合、 $A$ は競合層に存在する全てのノードの集合、 $W$ は第1(2)勝者の重みベクトル、 $W_k$ は $k$ 番目のノードの重みである。もしも入力ベクトルと第1勝者の距離が第1勝者の類似度閾値よりも小さく、かつ第2勝者についても同様のことが成り立てば、入力ベクトルは第1勝者および第2勝者と同じクラスタであると判断する。第1勝者または第2勝者の類似度閾値よりも入力ベクトルまでの距離が離れていれば入力ベクトルは既存のノードとは同じクラスタではないと判断する。

### 2.2 想起フェーズ

想起フェーズでは連想キーとして実数値ベクトルがひとつ入力され、連想キーに十分近い重みベクトルを持つノードが競合層に存在すればそのノードが発火し、そのノードの重みを用いて連想結果を出力する。

図2のF部またはR部のみにパターンが入力されると、競合層にある各ノードの重みベクトルのF部またはR部に対応する部分と入力データの距離が計算される。もしも閾値 $\delta_r$ を下回るノードが存在すれば、それらのノードが属するクラスタのプロトタイプノードの重みベクトルのうち、連想キーがF部に入力されたのであればR部に対応する部分、R部が入力されたのであればF部に対応する部分出力される。もし複数の異なるクラスタに属するノードの距離が閾値を下回れば、複数の連想結果を出力することになる。一方、閾値を下回るノードが存在しない場合は、入力データは未知データであると出力する。

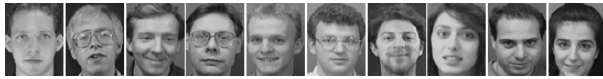


図3: 実験で用いた顔画像の例

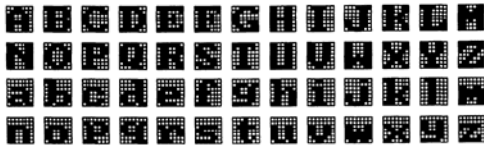


図4: 実験で用いた2値画像



図5: 追加学習実験で用いた連想対の例



図6: (a)追加学習実験の出力の例 (b)原画像

### 3. 実験

追加学習性能, ノイズ耐性, 多対多の連想機能についてそれぞれ評価実験を行った. 実験には図3のような AT&T の顔画像データベースの  $92 \times 112$  ピクセル画像を用いたが, 追加学習とノイズ耐性を従来手法と比較を行う際には図4に示した  $7 \times 7$  ピクセルの2値画像を利用した. これは代表的な従来手法が2値データしか扱えないためである. 比較実験に用いた手法として知識を分散して記憶するタイプの BAM with PRLAB[7] と, 競合学習型モデルの KFMAM[2], KFMAM を追加学習のために拡張した KFMAM-FW[4]を用いた. 各手法のパラメタを(表1)に示した. 比較手法のパラメタは[4][7]で用いられている値に従った. グレースケールの画像は各ピクセルの値を  $[-1, 1]$  に規格化して入力し, 2値データは白色を-1, 黒色を+1として入力した.

#### 3.1 追加学習

追加学習を適切に行えることを確かめるためグレースケールの顔画像からなる学習データを SOINN-AM に逐次的に入力して学習させた. 学習データの連想対は図5のように同じ人物を異なった角度で撮影した5枚の写真を連想キーとし, 正面から撮影した写真を連想結果であるような5対1の連想ペアである. この5対1の連想対をそれぞれ図3に示した10人分作り, 学習データとして入力した. 学習後の SOINN-AM に50個の連想キーをそれぞれ入力したと

表1: パラメータ設定

Method	Values of Parameters
SOINN-AM	$\Lambda_{edge} = 50, \lambda = 100, \delta_r = 0.15, \sigma_i^2 = 0.02$
BAM with PRLAB	$\lambda = 1.9, \xi = 0.1$
KFMAM	$\alpha_0 = 0.1, \sigma_i = 3.0, \sigma_r = 0.5, T = 2500$
KFMAM-FW	$\alpha_0 = 0.1, \sigma_i = 3.0, \sigma_r = 0.5, T = 2500, d_f = 10^{-3}$

表2: 追加学習実験の結果

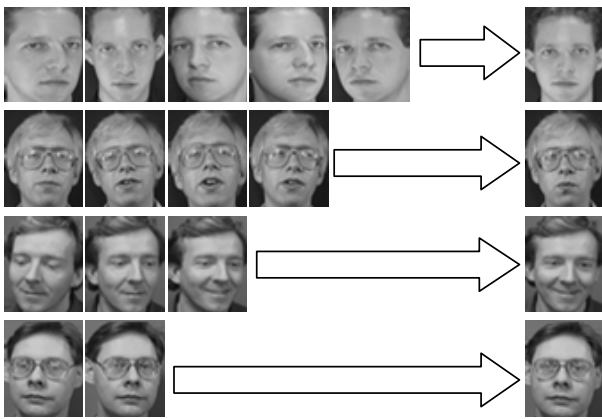
Method	Perfect Recall Rate	Perfectly Recalled Patterns
SOINN-AM	100%	a - z
BAM with PRLAB	3.8%	z
KFMAM (64 nodes)	31%	o, p, r, v - z
KFMAM (81 nodes)	38%	h, l, o, p, r, v - z
KFMAM (100 nodes)	42%	d, h, l, o, p, r, v - z
KFMAM-FW(16 nodes)	-	(無限ループ)
KFMAM-FW (25 nodes)	-	(無限ループ)
KFMAM-FW (36 nodes)	100%	a - z
KFMAM-FW (64 nodes)	100%	a - z

ころ, SOINN-AM の出力と正解の画像との誤差(ユークリッド距離)は 0.1 から 0.2 の間であった. 用いた画像の次元は 10304 次元であるから, 次元当りの誤差は  $1.0 \times 10^{-5}$  から  $2.0 \times 10^{-5}$  と極めて小さいものである. SOINN-AM の出力と正解画像の例を図6に示した. この場合の誤差は 0.126 であるが, 両者が違う画像であることは筆者らは目視では判別がつかない.

次に, 2値データを用いて従来手法と追加学習性能について比較を行った. 学習させた連想対は(A, a), (B, b), ..., (Z, z)の26組とした. これらの連想対を(A,a)から(Z,z)までアルファベットの順に追加学習させた. 学習後, 想起キーとして大文字のアルファベットの画像を入力したときに, 学習した小文字のアルファベットが出力されるかを見た. その結果を表2に示す. SOINN-AM と, ノードを事前に36個または64個与えた KFMAM-FW は全ての小文字を正しく出力した. ただし, ノードの数が16個または25個の KFMAM-FW はノードの数の不足のため学習中に無限ループに陥ってしまったため, 想起キーを入力することができなかった. この結果は, 事前に連想対の数がわかっていない場合は SOINN-AM 以外の手法は追加学習を適切に行えないことを示している.

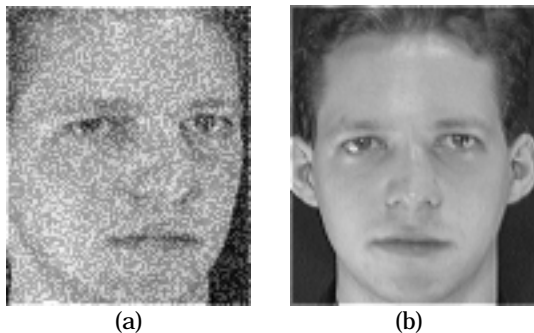
#### 3.2 多対多の連想

図7に示したような5対1, 4対1, 3対1, 2対1の学習データを用いて SOINN-AM を学習させた. これらを学習するために用いた SOINN-AM は一つだけであって4つの SOINN-AM で学習したのではないことに注意されたい. 学習後の SOINN-AM に14個の連想キーをそれぞれ入力したと



Associative keys

図7: 多対多連想実験で用いた学習データ



(a) (b)

図8: (a)ノイズの乗った連想キー (b)SOINN-AMの出力

ところ想起結果と原画像の誤差は追加学習実験と同様に0.1から0.2という十分小さな値であった。

### 3.3 ノイズ耐性

まず 3.1 で顔画像を用いて行った実験の学習データを学習したあとホワイトノイズを乗せた図 8 (a)を連想キーとして入力したところ図 8(b)を想起することができた。出力と原画像との誤差は0.126という小さいものであった。

次に、図 1(a)の種類ノイズに対する耐性の比較実験のためバイナリノイズの乗った連想キーを用いて想起させる実験を行った。学習データは図 4 の画像の大文字と小文字のペア (A, a)から(Z, z)であった。連想キーは原画像に 0%から 26%の割合でバイナリノイズを加えたものをそれぞれ 100 組ずつ作成した。このようにして作った連想対を学習後の各手法に入力したところ、図 9 に示すように正しく連想できる割合は SOINN - AM が最も高かった。

また図 1(b)の種類ノイズ耐性を確かめるため、同じく (A, a)から(Z, z)を学習した後に図 10 に示したランダムに作成した連想キーを入力した。その結果、SOINN-AM は未知の連想キーであると望ましい結果を出力したが他の手法は学習データとは無関係なパターンを想起結果として出力してしまった。

## 4. まとめと今後の予定

学習すべき知識量の上限が事前に不明な環境下でも、ノードを自律的に増やすことで追加学習が行える連想記憶モデルを提案した。提案手法はノイズ耐性、実数値データの

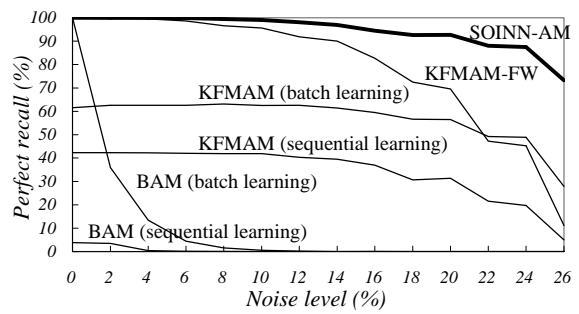


図9: 連想キーのノイズの割合に対するPerfect Recall

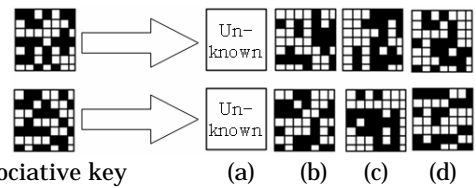


図10: ランダムパターンを想起キーとした時の出力.

(a) SOINN-AM. (b) BAM with PRLAB on batch learning (c) KFMAM with 100 nodes. (d) KFMAM-FW with 100 nodes.

取り扱い、多対多の連想も実現している。今後は提案手法の性能をさらに精査するための評価実験およびロボットへの適用を行う予定である。

**謝辞** 本研究の実施にあたり NEDO 産業技術研究助成事業から支援を頂きました。記して感謝いたします。

## 参考文献

- [1] K. Itoh et al., "New memory model for humanoid robots - introduction of co-associative memory using mutually coupled chaotic neural networks," *Proc. of the 2005 International Joint Conference on Neural Networks*, pp. 2790-2795, 2005
- [2] H. Ichiki et al., "Kohonen feature maps as a supervised learning machine," in *Proc. of the IEEE International Conference on Neural Networks*, pp. 1944-1948, 1993.
- [3] R. French, "Using Semi-Distributed Representation to Overcome Catastrophic Forgetting in Connectionist Networks," *Pm. of the 13th Annual Cognitive Science Society Conference*, pp. 173-178, 1991
- [4] T. Yamada et al., "Sequential Learning for Associative Memory using Kohonen Feature Map," in *Proc. of the 1999 International Joint Conference on Neural Networks*, pp. 1920-1923, 1999.
- [5] F. Shen and O. Hasegawa, "An incremental network for on-line unsupervised classification and topology learning," *Neural Networks*, Vol.19, No.1, 2006.
- [6] M. Hagiwara, "Multidirectional associative memory," in *Proc. of the 1990 International Joint Conference on Neural Networks*, pp. 3-6, 1990.
- [7] H. Oh and S.C. Kothari, "Adaptation of the relaxation method for learning in bidirectional associative memory," *IEEE Trans. Neural Networks*, Vol.5, No.4, pp. 576-583, 1994.