

リカレント構造適応型 Deep Belief Network による時系列データの学習

鎌 田 真*・市 村 匠**

Adaptive Structural Learning Method of Recurrent Deep Belief Network for Time Series Analysis

Shin KAMADA* and Takumi ICHIMURA**

Deep Belief Network (DBN), which is well known to be a kind of Deep Learning methods, has a deep network architecture that can represent multiple features of input patterns hierarchically. Each layer employs a pre-trained Restricted Boltzmann Machine (RBM). In DBN model including RBMs, we may meet the difficulties in finding the optimal network structure and the best set of weights and threshold values. For the solution, we developed the adaptive structure learning method of DBN that can discover an optimal number of hidden neurons for given input data in a RBM by neuron generation / annihilation algorithm, and hidden layers in DBN by the extension of the algorithm. Moreover, the Long Short-Term Memory (LSTM) model can make an accurate prediction for a time series data set. The network architecture of LSTM has been proposed in various ways and can represent high accurate prediction for the benchmark data set, but the problems related to the optimal structure and some parameters still remains. In this paper, the adaptive structure model of RBM and DBN is applied to the LSTM model and the effectiveness was verified by 10-fold cross validation on benchmark data sets.

Key Words: adaptive structural learning, restricted Boltzmann machine, deep belief network, long short-term memory, time-series data set

1. はじめに

Deep Learning (深層学習) と呼ばれる人工知能の手法が注目されている。産業分野における工場計測データや医療分野における医療検診データなど、クラウドサーバと連携してデータの収集をリアルタイムで行ない、膨大に収集されたデータから隠れた特徴を抽出したり、つぎに出現する時系列データの予測を行なう学習システムなどが注目を集め、実用化が期待されている¹⁾。

Deep Learning とは大規模な階層構造をもったニューラルネットワークにおける学習法である^{2),3)}。従来の階層型ニューラルネットワークとの最大の違いは事前学習にあるといわれている。各層ごとに教師なし学習を行なうことで入力データに含まれる確率分布 (事前知識) を学習し、事前学習した複数の特徴を階層的に組み合わせることで、データの特徴を高い精

度で表現することができる。Convolutional Neural Network (CNN)⁴⁾は高い分類精度をもつ Deep Learning 手法として知られているが、学習には高性能なハードウェアが必要で、計算コストがかかることが問題である。また CNN のみならず一般的な Deep Learning の最適なネットワーク構造は入力データの分布に応じて異なり、パラメタ設定は試行錯誤的に求める必要がある。

われわれはこのような問題を解決するために、入力データに応じて適切なネットワーク構造を探索する構造適応型学習法を、尤度の概念を取り入れた Deep Learning 手法である Restricted Boltzmann Machine (RBM)⁵⁾および事前学習した RBM を階層化し学習する Deep Belief Network (DBN)⁶⁾において開発している⁷⁾。構造適応型学習法では、入力データ空間に応じて適切な RBM の隠れニューロン数および DBN の隠れ層数が学習中に自動で探索され、画像ベンチマークデータセット CIFAR-10 および CIFAR-100⁸⁾に対し従来手法よりも高い分類精度が得られている^{9)~11)}。

近年の IoT における工場の計測データや経年変化が記録される医療検診データなどのビックデータの分析において、上記の画像のような分類問題だけでなく、時系列データを予測する機能が求められている。時系列データの学習についてはリカレントニューラルネットワーク (RNN) が知られている。従来の RNN モデルでは、ネットワークの隠れ層にループ構造をもたせることで時間によって遷移する時系列データの特徴

* 広島市立大学大学院情報科学研究科
広島市安佐南区大塚東 3-4-1

** 県立広島大学経営情報学部 広島市南区宇品東 1-1-71

* Graduate School of Information Sciences, Hiroshima City University, 3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima

** Faculty of Management and Information Systems, Prefectural University of Hiroshima, 1-1-71 Ujinahigashi, Minami-ku, Hiroshima

(Received October 20, 2017)

(Revised February 18, 2018)

を表現しているが、データの長期に亘る系列を上手く取り扱うことができず、短期的な記憶のみしか表現できないことが問題であった¹²⁾。これは、RNNの隠れ層のループ構造を時系列を軸にして展開すると非常に深い多層ニューラルネットワークと見なすことができるが、これにより長期的に学習した記憶を再認識できる期間に限界があったためである。これに対しRNNの一種であるLong Short-Term Memory (LSTM)¹²⁾では、RNNのループ構造に入出力ゲート、忘却ゲートと呼ばれる特殊なゲートを組み込むことで、長期的な記憶を表現できるようになった。一方、RBMモデルにおける時系列データ学習モデルでは、Hintonなどが開発したTemporal RBM (TRBM)、Recurrent TRBM (RTRBM)¹³⁾、またBengioなどが開発したRecurrent Neural Network RBM (RNN-RBM)¹⁴⁾が知られている。著者らはこの中で精度が高いとされているRNN-RBMに構造適応型学習法のニューロン生成/消滅アルゴリズムを組み込み、音声データやモーションキャプチャなどの時系列ベンチマークデータセットに対し高い精度で予測可能な手法を開発した^{15), 16)}。本論文ではさらに構造適応型学習により適切な隠れニューロン数を求めた後、学習係数の調整を行ないながらネットワークを再学習(微調整)することで分類精度を向上させる手法を提案する。提案するリカレント構造適応型DBNの時系列ベンチマークテストに対する予測精度について10-foldクロスバリデーションテストを行なった結果を報告する。

本論文の構成はつぎのとおりである。2章では、従来のRBMおよびDBNモデルについて概要を説明する。また本研究において開発した構造適応型RBMにおけるニューロン生成/消滅アルゴリズム、また層の自動生成を行なう構造適応型DBNについて述べる。提案した構造適応型RBMおよび構造適応型DBNは、CIFAR-10やCIFAR-100などの画像分類問題に対し高い分類精度を示すことができたが、3章ではこれを時系列データに適用したリカレント構造適応型RBMおよびリカレント構造適応型DBNを提案し、さらに学習係数の調整方法について説明する。4章において、提案手法のベンチマークテストを用いた性能を評価を行なう。

2. 構造適応型 Deep Belief Network による学習法

2.1 Restricted Boltzmann Machine

RBM⁵⁾はFig. 1のように可視層と隠れ層の2層から構成されるネットワーク構造をもち、確率分布に基づいて入力データに含まれる特徴を隠れニューロン上で学習することができる教師なし学習モデルである。通常のBoltzmann Machineはすべてのニューロン間に接続がある相互結合型モデルであるが¹⁷⁾、RBMには各層内のニューロン間の結合はない。このため層内のニューロン間の結合に対する計算量を削減できるだけでなく、隠れ層の各ニューロンごとに独立した確率分布を学習することができる。

RBMでは与えられた入力データに対して(1)式のエネルギー関数を最小にするパラメタ $\theta = \{b, c, W\}$ が最尤

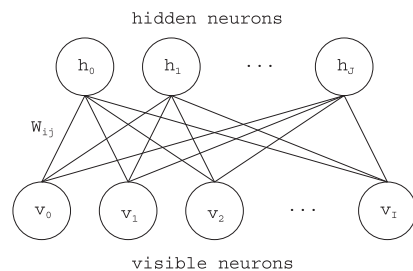


Fig. 1 Network structure of RBM

推定により求められる。ここで、 $b = \{b_1, \dots, b_i, \dots, b_I\}$, $c = \{c_1, \dots, c_j, \dots, c_J\}$, $W = \{W_{11}, \dots, W_{ij}, \dots, W_{IJ}\}$ である。

$$E(v, h) = - \sum_i b_i v_i - \sum_j c_j h_j - \sum_i \sum_j v_i W_{ij} h_j, \quad (1)$$

$$p(v, h) = \frac{1}{Z} \exp(-E(v, h)), \quad (2)$$

$$Z = \sum_v \sum_h \exp(-E(v, h)) \quad (3)$$

ここで、 v_i は*i*番目の可視ニューロンの状態変数、 h_j は*j*番目の隠れニューロンの状態変数である。 b_i は v_i に対するパラメタ、 c_j は h_j に対するパラメタ、 W_{ij} は v_i と h_j 間の重みである。 $v \in \{0, 1\}^I$ と $h \in \{0, 1\}^J$ はそれぞれ可視層と隠れ層に含まれるニューロンの状態変数のベクトルであり、 I と J は可視ニューロン数、隠れニューロン数である。(2)式は入力 v と h の確率分布である。(3)式の Z は分配関数(partition function)であり、 v と h のすべての2値のパタンの組み合わせに対するエネルギーを示す。最適なパラメタ θ は与えられた入力データの尤度 $p(v|\theta)$ を最大化することで求められるが、モデルの尤度に含まれる分配関数 Z の計算は現実的に困難である。そのため、この値を推定するためにContrastive Divergence (CD)法¹⁸⁾のサンプリングによって近似が行なわれる。このサンプリングの過程で条件付き確率である連続値が2値に変換されてしまい、機械学習が収束するための条件である連続性を満たさない状態を生じることが考えられる。この問題に対して文献19)では、RBMの各パラメタの変分がリップシツ連続で、その上限がシャッテンノルムにより与えられ、これを満たすことで学習が収束することが示されている。これらの式の導出については、文献20)を参照されたい。

2.2 Deep Belief Network

Deep Belief Network (DBN)は、事前学習を行なったRBMを複数積み重ねることで入力データの特徴を階層的に学習することができるDeep Learning手法である⁶⁾。Fig. 2は4つのRBMをもつDBNのネットワーク構造を示している。入力データを $h^0 = v$ とすると、 $l (> 0)$ 層目のRBMの*j*番目の隠れニューロンの発火確率は(4)式のように定義される。

$$p(h_j^l = 1|h^{l-1}) = \text{sigm}(c_j^l + \sum_i W_{ij}^l h_i^{l-1}) \quad (4)$$

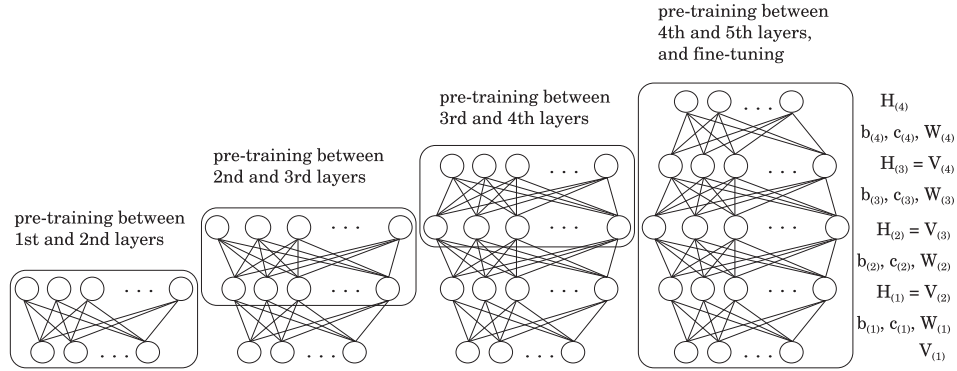


Fig. 2 Hierarchical network structure of DBN

ここで、 c_j^l と W_{ij}^l はそれぞれ l 層目の RBM の j 番目の隠れニューロンのパラメタと i, j 番目の重みである。 $\text{sigm}()$ はシグモイド関数で、文献 6) と同じ関数を使用している。 DBN 自体は RBM を積み重ねたモデルであるため教師なし学習モデルであるが、最上位層に出力層を加え、ソフトマックス法などにより計算された出力値と教師信号との誤差 (損失) を最小化することで、教師あり学習として適用できる (Fine-Tuning)。ソフトマックス法では、与えられた入力データがカテゴリ k に属する確率 y_k が (5) 式により求められる。

$$y_k = \frac{\exp(z_k)}{\sum_{j=1}^M \exp(z_j)} \quad (5)$$

ここで、 z_j は出力層における j 番目のニューロン出力活性化値である。 M は出力層のニューロン数である。

2.3 ニューロン生成/消滅アルゴリズム ⁷⁾

入力データに対して適切なニューロン数を求める構造適応型学習法については、1) ニューロンの枝刈りを行なう手法 ²¹⁾、2) ほかのクラスタリング手法の結果により構造を求める手法 ²²⁾、3) ニューロンの生成を行なう手法 ²³⁾ の 3 種類のアプローチが主に知られている。1) の手法では、枝刈りを行なうために大きなネットワークから始める必要があるため、Deep Learning のような大きなネットワークが必要な学習では、構造やパラメタなどの初期計算コストが必要となり、結果として大きな計算資源が必要となる。2) の手法では、Deep Learning 自体の精度を直接利用するのではなく、ほかの手法を用いた結果を Deep Learning の学習に利用することになるため、計算結果はクラスタリング手法の精度に依存することになる。3) の手法では、1) と 2) の手法と比べて、学習中にニューロンの増減を自己組織的に求めることが可能である点に優位性がある。

3) の手法として、従来の階層型ニューラルネットワークでは、学習中に更新される重みの変分ベクトルを用いて隠れニューロン数を入力データに対して適応的に変更する手法が提案されている ²⁴⁾。 Fig. 3 は学習が進むにつれて重みの変分ベクトルが収束する過程を示している。入力データに対して十分な隠れニューロン数があれば、Fig. 3 のように学習が進むにつれて重みの変分は小さくなり、ある一定の値に収束す

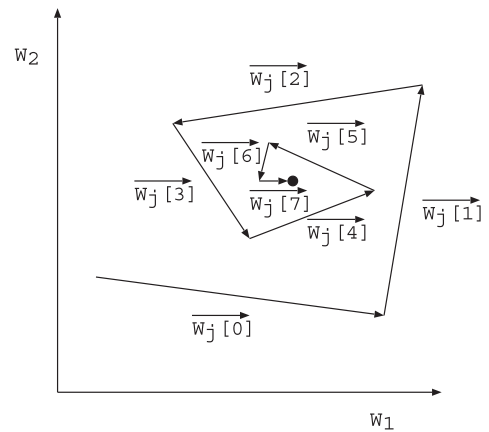


Fig. 3 Convergence situation of walking distance

ると考えられる。一方、この条件を満たさず重みが収束しない場合は、学習データに対するネットワークの表現能力が不足しており、データの 패턴を隠れニューロン上で表現できていないことが原因だと考えられる。このような場合、関連する位置に十分な数の隠れニューロンを挿入することでデータの pattern を表現できると考えられる。文献 24) では学習中における重みの変分ベクトルを Walking Distance (WD) と定義している。WD は (6) 式により、 $\tau - 1$ 回目までの学習によるパラメタの変分と τ 回目の学習のパラメタの変分の和により計算される。

$$WD_j[\tau] = \gamma_w WD_j[\tau - 1] + (1 - \gamma_w) Met(\mathbf{W}_j[\tau], \mathbf{W}_j[\tau - 1]) \quad (6)$$

ここで、 $\mathbf{W}_j[\tau]$ は τ 回目の学習における隠れニューロン j に関連する重みである。 Met はベクトル間の距離を計算する関数で、本論文ではユークリッド距離 $Met(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^N \sqrt{(a_i - b_i)^2}$ を用いる。 γ_w は $\tau - 1$ 回目までの変分と τ 回目の変分の影響度を調整するパラメタで $0 < \gamma_w < 1$ である。文献 24) ではある一定回数の訓練後、隠れニューロン j の $WD_j[\tau]$ の値があらかじめ定められた閾値よりも大きい場合、その隠れニューロンの重みを 2 分割し、新しく隠れニューロンを追加する手法が提案されている。

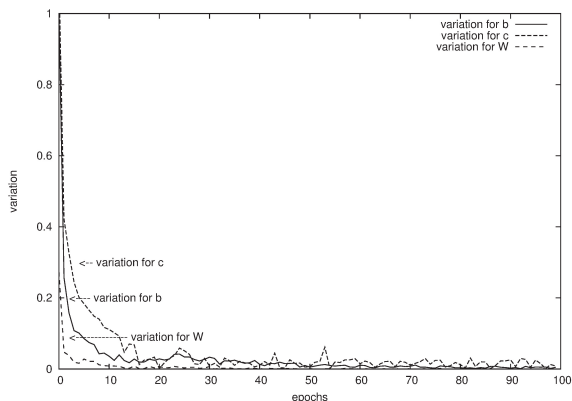


Fig. 4 Variation for parameters $\theta = \{b, c, W\}$ in RBM

上記の階層型ニューラルネットワークと異なり、RBMには3種類の学習パラメタ $\theta = \{b, c, W\}$ がある。われわれの調査では、RBM学習における3つのパラメタは、隠れニューロンに関するパラメタ c と重み W が隠れニューロンの学習状況によって変化し、入力に関するパラメタ b が入力データの分布に応じて変化することがわかった²⁵⁾。Fig. 4は、MNIST²⁶⁾と呼ばれる手書き文字データセットにおけるRBMの学習中の b, c, W の変分としてWDを示している。MNISTは数字の0から9の手書き文字画像のデータセットである。60,000枚の訓練画像を用いて学習して、10,000枚のテスト画像を用いて分類精度を評価する問題である。各画像は 28×28 ピクセルのグレースケールで表現された画像である。各パラメタのWDは(6)式の $WD_j[\tau]$ の計算方法により求めた。すなわち、 $\tau-1$ 回目までの学習による変分と、 τ 回目の学習の変分の和に基づいて、各パラメタの変分が計算される。ここでRBMの学習アルゴリズムにはStochastic Gradient Descent (SGD)⁵⁾を用い、学習率は0.1、学習のバッチサイズは100とした。予備実験では、100個の隠れニューロンがあれば、MNISTの学習が適切に行なわれたが、ここではデータに対し隠れニューロン数が不足し、パラメタの変分が振動する状態を調査するため、隠れニューロン数は10個とした。Fig. 4に示すように各パラメタの変分は学習が進むにつれて徐々に小さくなった。特に、パラメタ b や W の値は、学習の初期に急速に下がり、30回目の反復計算以降では大きな変化は見られなかった。一方、 c の値は30回目の反復計算以降においても、 b や W の値よりも大きく変化しており、42回や53回目の反復計算付近において振動している状態が見られた。このような傾向はMNISTだけでなく、CIFAR-10やCIFAR-100などのほかのデータセットに対しても見られた。そこで、本論文では、 b は入力に関するパラメタであるため考慮せず、 c と W は隠れニューロンの学習状況に依存するパラメタであるため、この2つの変分の積空間を用いた指標を考え、ニューロン生成の条件を(7)式に示すように定義した。

$$WD_{c_j} \cdot WD_{W_j} > \theta_G, \quad (7)$$

$$WD_{c_j} = \gamma_c WD_{c_j}$$

$$+ (1 - \gamma_c)(|c_j[\tau] - c_j[\tau - 1]|), \quad (8)$$

$$WD_{W_j} = \gamma_w WD_{W_j}$$

$$+ (1 - \gamma_w)Met(W_j[\tau], W_j[\tau - 1]) \quad (9)$$

ここで、 WD_{c_j} 、 WD_{W_j} はそれぞれ j 番目の隠れニューロンに関するパラメタと重みに関する学習の変分であり、0以上の値をとる。これらは、(6)式と同様の計算方法で求められ、隠れニューロン j に関連する c_j と W_j の変分が観察される。 θ_G はニューロン生成に関する閾値であり $\theta_G > 0$ の値をとる。 θ_G の値が小さくなるに従って、ニューロン生成条件は満たされやすくなる。学習中に(7)式を満たす隠れニューロンがある場合、該当する隠れニューロンの属性を継承したニューロンが(10)式のように生成され、Fig. 5(a)のように隣接する位置に挿入される。このとき、元のパラメタ値に、微小な値として平均0、標準偏差 σ の正規乱数 $N(0, \sigma^2)$ をノイズとして加えることで、ニューロン生成後の学習によるパラメタの変化が親のニューロンと全く同じ方向に進むことを抑制し、親とわずかに異なる特徴を学習させることができる。本論文では、微小な振動を与えるために、 $\sigma = 0.1$ とした。なお、加算するノイズのパラメタ σ については、 c_j と W_{ij} で同じ値を使用しているが、両者の取り得る範囲に大きな違いはなく、さらに本論文で使用している値 ($\sigma = 0.1$) であれば、元の範囲を大きく超えるような値にはならないため、個別の値を設定していない。

$$c_j^{new} = c_j + N(0, \sigma^2), W_{ij}^{new} = W_{ij} + N(0, \sigma^2) \quad (10)$$

一方、入力データに対して十分な数の隠れニューロンが生成された後、そのうち出力に寄与していない、すなわち冗長な値を出力する隠れニューロンが存在する場合がある。構造適応型RBMでは、一定の学習の後、(11)式を満たす隠れニューロンが存在すれば、該当する隠れニューロンをFig. 5(b)のように消去する。(11)式ではすべての入力データに対する隠れニューロンの出力値の平均を観察し、その値が一定の閾値以下のニューロンを冗長なニューロンとみなし、削除する。

$$\frac{1}{N} \sum_{n=1}^N p(h_j = 1 | v_n) < \theta_A \quad (11)$$

ここで、 $p(h_j = 1 | v_n)$ は n 番目の入力データ v_n を与えた際の隠れニューロン h_j の発火確率を示している。 θ_A はニューロン消滅にかかわる閾値であり、 $0 < \theta_A < 1$ の値をとる。 θ_A が大きくなるに従って、ニューロン消滅が適用されやすくなる。ニューロン生成と消滅に関するパラメタ θ_G 、 θ_A は事前に適切な値を与える。これらの値は学習データの次元数やサンプル数、および隠れニューロン数に関連すると考えている。たとえば、データ数が多いほどデータの特徴に関して各パラメタの振動は大きくなる傾向があるため、 θ_G は小さく設定し、ニューロン生成を起きやすくする。初期の隠れニューロン数が少ない場合も同様にする。一方、データに対し隠れニューロン数が少なすぎる場合は、冗長な値を出力するニュー

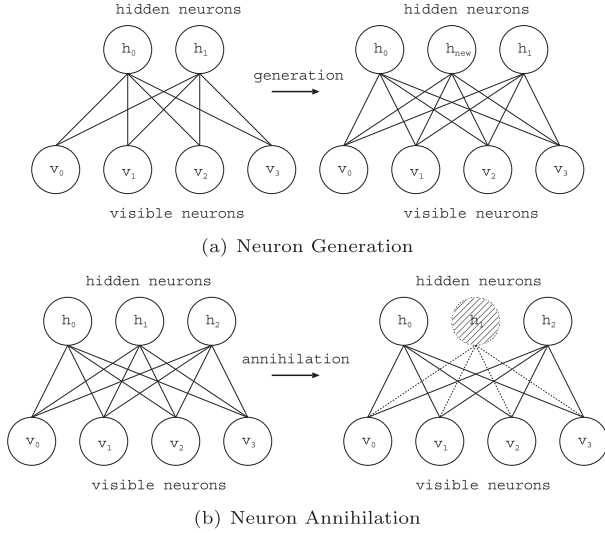


Fig. 5 Adaptive learning method of RBM

ロンが少なくなるため、 θ_A の値を大きくしてもニューロン消滅があまり起こらない。これらのパラメタの与え方の1つの方法として、予備実験の結果では、 θ_G を少し低い値 (例: 0.001) にし、 θ_A を少し高い値 (例: 0.100) にすることでうまく学習できている。ただし、 θ_A が高すぎる場合は、必要以上のニューロン消滅が生じるため、この値はニューロン消滅と関連して定める必要がある。なお、(11) 式では活性化していないニューロンを削除しているが、逆のアプローチとして、どんなサンプルに対しても常に反応するニューロンを削除する方法も考えられる。このアプローチについては、今後研究を行なっていく予定である。

2.4 層の生成条件⁹⁾

従来の階層型ニューラルネットワークでは、学習状況の訓練誤差に応じて新しい隠れ層を生成する手法がある²³⁾。一方、Deep Learningのように、従来よりも大規模なネットワーク構造をもつ手法では、層を自動で決めるよりも試行錯誤的に決定する方が一般的である。DBNでは、各層においてRBMによる事前学習を行ない、1層ずつ層を積み重ねていくことで、ネットワークの表現能力が上がる²⁷⁾。しかしながら、層の数が多くなると、層の生成により必ずしも分類精度が上がるとは限らないため、一般的には、試行錯誤的に層の生成を行なう必要がある。本論文では、事前学習において学習が適切に行なわれれば、上位層ほど与えられた入力データに対するエネルギーやパラメタの変化が小さくなることを用いて、DBNにおける層の生成条件を提案する。

構造適応型RBMにおけるニューロン生成アルゴリズムでは学習中のWD (パラメタ \mathbf{c} と \mathbf{W} の変分) を観察したが、構造適応型DBN⁹⁾では個々のRBMではなくネットワーク全体のWDとエネルギーを(12)式と(13)式を用いて観察した。

$$\sum_{l=1}^k WD^l > \theta_{L1}, \quad (12)$$

$$\sum_{l=1}^k E^l > \theta_{L2} \quad (13)$$

ここで、 $WD^l = \sum_{j=1}^J (WD_{c_j}^l \cdot WD_{W_j}^l)$ であり、 $WD_{c_j}^l$ と $WD_{W_j}^l$ はそれぞれ l 層目のRBMにおける c_j と \mathbf{W}_j に関するWDを示している。 E^l はエネルギー関数である。 θ_{L1} 、 θ_{L2} はあらかじめ与えられる閾値である。ある k 層目のRBMの学習中に(12)式および(13)式を同時に満たす場合、入力データに対するネットワーク全体の表現能力が不足していると考え、 k 層目の学習が終了した時点で、 $k+1$ 層目のRBMを生成し学習を行なう。ここでもRBMのニューロン生成アルゴリズムと同様に、 $k+1$ 層目のRBMのパラメタ \mathbf{b} 、 \mathbf{c} 、 \mathbf{W} の初期値が k 層目のRBMから継承される。

提案手法の構造適応型RBMおよびDBNは画像ベンチマークテストCIFAR-10およびCIFAR-100に対し従来のモデルよりも高い分類精度を示した^{9)~11)}。本論文では画像の分類だけでなく時系列データの予測を行なうため、3章で構造適応型学習法のアルゴリズムを時系列データに適用する方法を提案する。

3. リカレント構造適応型DBNによる時系列データの学習

3.1 RBMにおける時系列学習モデル

時系列データの学習モデルとしては通常のニューラルネットワークではリカレントニューラルネットワーク(RNN)やRNNの一種であるLSTMが知られているが、RBMモデルにおいてはTemporal RBM (TRBM)、Recurrent TRBM (RTRBM) が知られている¹³⁾。Fig. 6、Fig. 7はそれぞれTRBMとRTRBMのモデルを示している。長さ T の時系列入力データの系列を $\mathbf{V} = \{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(t)}, \dots, \mathbf{v}^{(T)}\}$ とすると、TRBM、RTRBMは時刻 t における隠れニューロンの出力 $\mathbf{h}^{(t)}$ を時刻 $t+1$ のネットワークに伝搬させることで、過去の時系列に関する文脈を表現する。RNN-RBMはRTRBMを改良したモデルである¹⁴⁾。RNN-RBMでは通常のRBMの可視層と隠れ層に加え、過去の時系列に関する文脈を表現する状態 $\mathbf{u}^{(t)}$ をもつ。ある時刻 t の入力 $\mathbf{v}^{(t)}$ に対する可視層と隠れ層のパラメタ $\mathbf{b}^{(t)}$ 、 $\mathbf{c}^{(t)}$ が、時刻 $t-1$ の状態 $\mathbf{u}^{(t-1)}$ のみから(14)式、(15)式によって求められる。時刻 t の状態 $\mathbf{u}^{(t)}$ は時刻 t の入力 $\mathbf{v}^{(t)}$ と時刻 $t-1$ の状態 $\mathbf{u}^{(t-1)}$ から(16)式のように更新される。

$$\mathbf{b}^{(t)} = \mathbf{b} + \mathbf{W}_{uv} \mathbf{u}^{(t-1)}, \quad (14)$$

$$\mathbf{c}^{(t)} = \mathbf{c} + \mathbf{W}_{uh} \mathbf{u}^{(t-1)}, \quad (15)$$

$$\mathbf{u}^{(t)} = \sigma(\mathbf{u} + \mathbf{W}_{uu} \mathbf{u}^{(t-1)} + \mathbf{W}_{vu} \mathbf{v}^{(t)}) \quad (16)$$

ここで、 $\sigma()$ は活性化関数であり、文献14)ではtanh関数を用いられており、本論文でもこれを用いている。Fig. 8はRNN-RBMにおける計算の流れを示している。 $\mathbf{u}^{(0)}$ は状態の初期値で任意の値が与えられる。RNN-RBMの学習は各時刻 t において計算された $\mathbf{b}^{(t)}$ 、 $\mathbf{c}^{(t)}$ と重み \mathbf{W} を用い

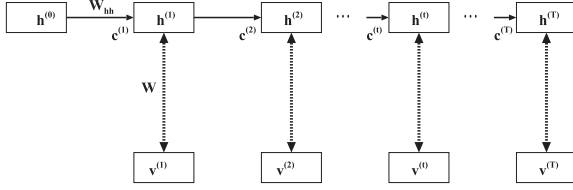


Fig. 6 Network structure of TRBM

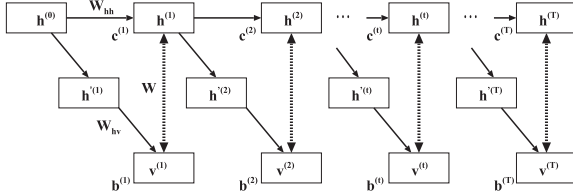


Fig. 7 Network structure of RTRBM

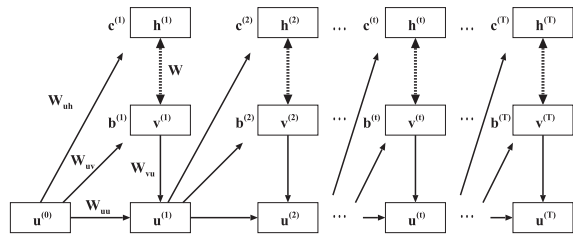


Fig. 8 Network structure of RNN-RBM

Algorithm 1 Adaptive RNN-RBM

Set initial parameter $\mathbf{u}^{(0)}$.
for all $\mathbf{v}^{(t)}$ ($1 \leq t \leq T$) **do**
 Calculate $\mathbf{b}^{(t)}$ and $\mathbf{c}^{(t)}$ from $\mathbf{u}^{(t-1)}$ by Eq. (14) and Eq. (15)).
 Update $\mathbf{u}^{(t)}$ from $\mathbf{u}^{(t-1)}$ and $\mathbf{v}^{(t)}$ by Eq. (16)).
end for
Calculate cost between $\mathbf{v}^{(t)}$ and $\mathbf{h}^{(t)}$ by CD-k.
Calculate gradients for the parameters θ by BPTT and update them.
for all hidden neuron j **do**
 if the neuron generation process is not completed **then**
 if the neuron generation condition is satisfied with Eq. (7) **then**
 A new hidden neuron $j+1$ is generated and inserted.
 end if
 else
 if the neuron annihilation condition is satisfied with Eq. (11) **then**
 The hidden neuron j is annihilated.
 end if
 end if
end for

て、 $\mathbf{v}^{(t)}$ と $\mathbf{h}^{(t)}$ 間の学習が通常の RBM と同じように行なわれ (一般的に CD 法), 誤差が計算される. 時刻 T までの誤差を計算した後, T から過去に遡るように学習パラメタ $\theta = \{\mathbf{b}, \mathbf{c}, \mathbf{W}, \mathbf{u}, \mathbf{W}_{uv}, \mathbf{W}_{uh}, \mathbf{W}_{vu}, \mathbf{W}_{uu}\}$ の勾配が BPTT (Back Proagation Through Time) ²⁸⁾ により計算され, 更新される.

Algorithm 2 Adaptive RNN-DBN

Set 1-th input $\mathbf{V}_{(1)}$ and initial value of parameters $\theta_{(1)}$.
for $1 \leq l \leq L$ **do**
 Make pre-training l -th RBM for given $\mathbf{V}_{(l)}$ and $\theta_{(l)}$.
 if the neuron layer conditions are satisfied with Eq. (12) and Eq. (13) during the learning **then**
 The layer $l+1$ is generated.
 Calculate l -th input $\mathbf{V}_{(l+1)}$ and set initial value of parameters $\theta_{(l+1)}$.
 else
 The layer generation is stopped.
 end if
end for

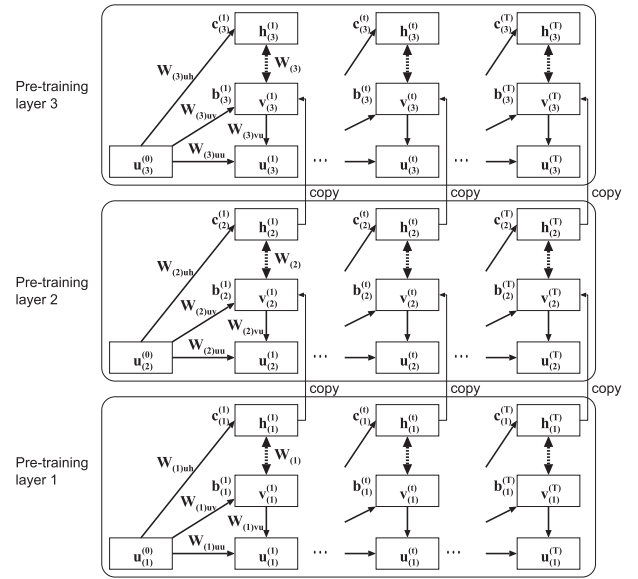


Fig. 9 Network structure of RNN-DBN

3.2 リカレント構造適応型 RBM とリカレント構造適応型 DBN

本節では, 提案するリカレント構造適応型 RBM およびリカレント構造適応型 DBN について述べる. リカレント構造適応型 RBM は, 文献 14) の RNN-RBM に構造適応型 RBM におけるニューロン生成/消滅アルゴリズムを取り入れた手法である. RNN-RBM は通常の RBM と比べて, 1) 時刻 t の入力に対するパラメタ $\mathbf{b}^{(t)}$ と $\mathbf{c}^{(t)}$ が過去の時刻 $t-1$ の状態から計算される, 2) 勾配計算に BPTT が使われる, という点において異なっているが, RBM の学習法自体に変わりはなく, 通常の RBM と同様に CD 法による学習が行なわれる. CD 法では与えられた入力データに対し目的関数を最小化するパラメタ $\mathbf{b}, \mathbf{c}, \mathbf{W}$ を学習する. そのため学習中のパラメタの変分が大きい場合, 入力データに対する表現能力を補うため, ニューロン生成を行なう手法を取り入れることができる考えた ¹⁵⁾. すなわち学習中に (6) 式を用いて隠れニューロンに関するパラメタ \mathbf{c} と重み \mathbf{W} の変分を観察し, (7) 式を満たすと該当するニューロンの生成が行なわれる. また (11) 式を満たすと該当するニューロンが削除される. Algorithm 1 は, リカレント構造適応型 RBM においてある入力データを

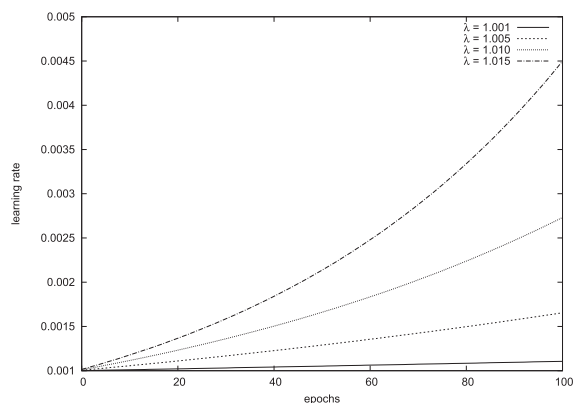


Fig. 10 Learning rate

学習する 1 ステップのアルゴリズムを示している。

また複数の事前学習されたリカレント構造適応型 RBM を階層化して学習するリカレント構造適応型 DBN を提案する。文献¹⁴⁾には RNN-RBM を階層化する手法については述べられていない。本論文では、2.2 節で述べた DBN の学習法をリカレント構造適応型 DBN に応用した¹⁶⁾。すなわち l 層の隠れニューロンの出力は $l+1$ 層の可視ニューロンの入力として見なされる。Fig. 9 は 3 つの RNN-RBM をもつリカレント構造適応型 DBN を示している。リカレント構造適応型 RBM では時刻 t の入力 $v^{(t)}$ に対する隠れニューロンの出力 $h^{(t)}$ が求まるため、この出力をつぎの層の入力とすることで、2 層以降においても適切な隠れニューロン数を求めながら階層化を行なうリカレント構造適応型 DBN を提案する。Algorithm 2 はリカレント構造適応型 DBN のアルゴリズムを示している。

3.3 学習係数調整によるネットワークの再学習

構造適応型学習における学習終了時点では入力データに含まれる特徴を表現するために適当と考えられる隠れニューロン数をもったネットワークが形成される。しかしながら、適当と考えられる隠れニューロン数が求められたとしても、学習係数などのパラメタによっては、学習は収束するものの誤差の値が小さくならないことがある。また、訓練データに出現するパターンに 100% に近い精度で予測できたとしても、テストデータに含まれるパターンを識別できず過学習の状態のような結果を出力することがある。このような過学習が生じる理由は、隠れニューロンが訓練データのみ出現するパターンの学習に留まり、テストデータに含まれる訓練データのパターン以外の特徴に対して正確に反応できないことが理由として考えられる。特に学習係数の値が小さければ、テストデータに含まれる汎用性をもった特徴を探索することができないため、学習係数の値を増加させ探索領域を広げる必要がある。そこで、構造適応型学習により適切な隠れニューロン数が求まった後、学習係数の調整を行ないながらネットワークを再学習（微調整）することで、分類精度の向上を試みた。

Deep Learning において学習係数は重要なパラメタの 1 つであり、この値を学習中に調整する手法がある。一般的には、

学習初期に学習係数を高い値に設定し、学習が進むにつれて値を減少させる調整方法が知られている^{29), 30)}。これにより、大域的かつ局所的な探索を行なうことができ、効果的に学習を行なうことができる。一方、学習途中において学習係数を適切な範囲で増加させることで、Deep Learning の分類精度が向上することが報告されている^{31), 32)}。本論文では、学習初期に学習係数を調整するのではなく、構造適応型学習により適切と考えられるネットワーク構造を求めた後に学習係数の調整を行なうため、学習係数を徐々に下げるのではなく、文献^{31), 32)} のようにわずかに増加させることで、精度を向上を図った。本論文では (17) 式のように学習係数の値を増加する方向で調整した。

$$\epsilon(\tau) = \epsilon(\tau - 1) \times \lambda, \lambda > 1 \quad (17)$$

ここで、 $\epsilon(\tau)$ は学習回数 τ 時点の学習係数を示している。 λ は学習係数の増加に関する係数である。本論文では予備実験により、学習開始時点の学習係数 $\epsilon(0) = 0.001$ で、 $\lambda = \{1.001, 1.005, 1.010, 1.015\}$ の 4 種類とし、100 回の学習を行なうことで学習係数を Fig. 10 のように変化させた。

4. 実験

4.1 データセット

提案手法の性能を評価するために、時系列ベンチマークテスト “Nottingham”³³⁾ と “CMU”³⁴⁾ を用いた。Nottingham は 88 個の階調で表現される MIDI 形式のピアノの音声データで、694 個の訓練データと 170 個のテストデータから構成される。各データからは約 210 個の長さ（約 60 秒程度）の時系列データを取得できる。CMU はカーネギーメロン大学によって収集されているモーションキャプチャのデータセットである。2017 年 9 月時点で 6 種類のカテゴリ（23 種類のサブカテゴリ）に分類された 2,605 個のデータが利用可能である。各データについては 30 個以上のマーカーを取り付けられた人の一連の動作（約 30 秒程度）を記録した時系列データである。これらのデータについて、10-fold クロスバリデーションテストを行ない、予測精度を評価した。

4.2 実験結果

4.1 節で述べたデータセットに対して従来の RNN-RBM (RT)、構造適応型 RNN-RBM (RA)、従来の RNN-DBN (DT)、構造適応型 RNN-DBN (DA) の手法を用いて、学習を行なった。従来の RNN-RBM (RT) は文献¹⁴⁾ で提案されているモデルであり、従来の RNN-DBN (DT) は従来の RNN-RBM (RT) を用いて階層化を行なったモデルである。構造適応型 RNN-RBM (RA) および構造適応型 RNN-DBN (DA) の各層の学習後、(17) 式による学習係数の調整を用いて学習した手法を、それぞれ RAL, DAL とし、性能を比較した。本論文では、勾配の学習手法を Stochastic Gradient Descent (SGD)、学習係数を 0.001、バッチサイズは 100、学習の最大反復回数を 200 回、初期隠れニューロン数を 200 とした。構造適応型 RBM のパラメタ

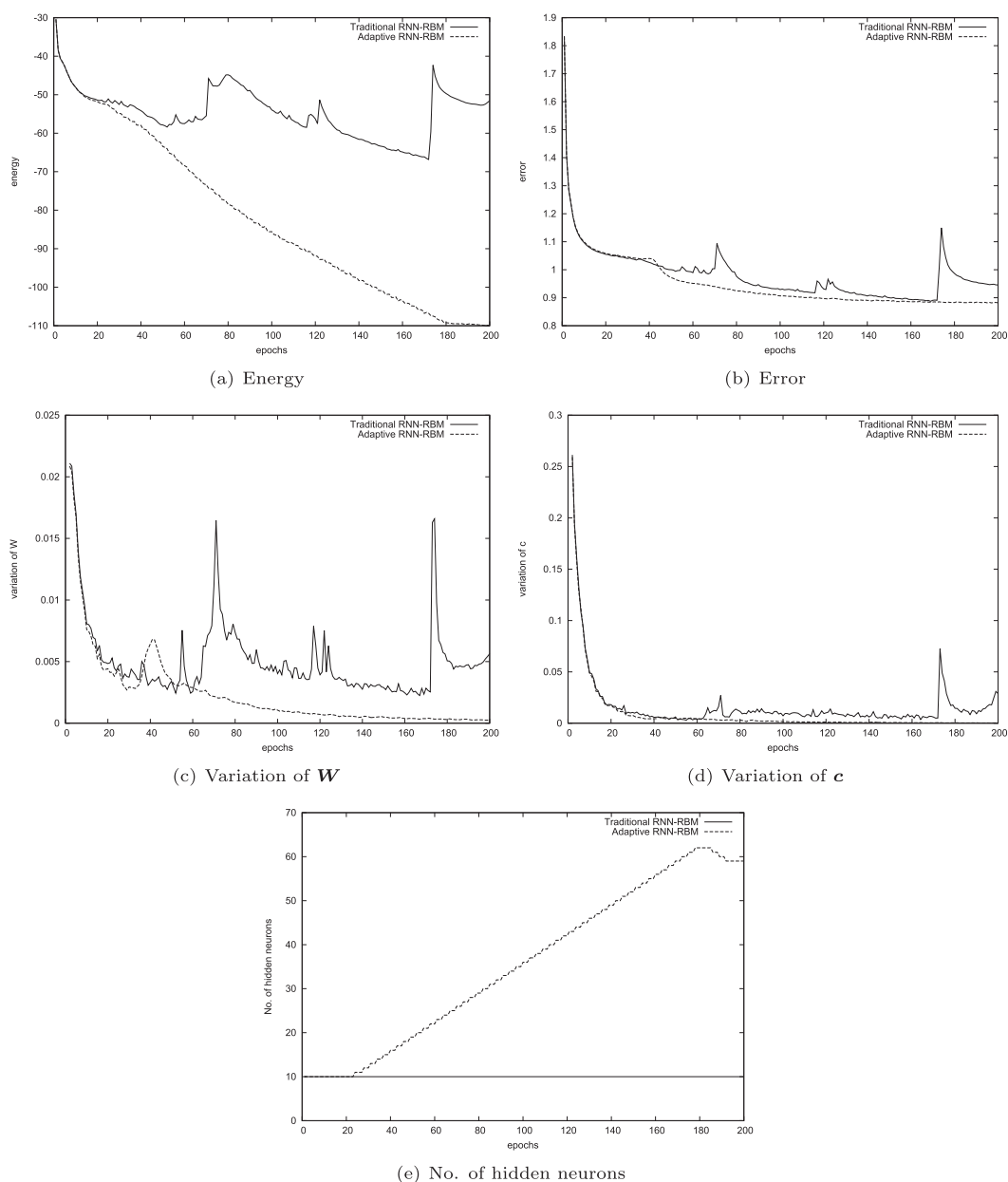


Fig. 11 Learning results on nottingham (layer 1)

として, $\theta_G = \{0.001, 0.005, 0.010, 0.050, 0.100\}$, $\theta_A = \{0.001, 0.005, 0.010, 0.050, 0.100\}$ の 25 通りの組み合わせで実験を行なった. 構造適応型 DBN のパラメタとして, $\theta_{L1} = 0.1$, $\theta_{L2} = 0.1$ とした. これらのパラメタの値は予備実験の結果, 最良の組み合わせであったものである. RAL, DAL では, 100 回の追加の反復学習を行ない, このとき, $\lambda = \{1.001, 1.005, 1.010, 1.015\}$ の 4 通りによる性能を比較した. 実験における学習にはつぎのスペックをもつ PC を用いた. CPU = Intel(R) 24Core Xeon E5-2670 v3 2.3 GHz, GPU = Tesla K80 4992 Core 24 GB \times 3, Memory = 64 GB, OS = CentOS 6.7 64 bit.

Fig. 11 はデータセット Nottingham に対する RT と RA の学習状況を示している. Fig. 11 (a) から Fig. 11 (e) まで,

Table 1 Prediction accuracy of traditional RNN-RBM

Dataset	N	Accuracy	
		Ave.	Std.
Nottingham	10	0.803	0.009
	50	0.817	0.008
	100	0.884	0.013
	150	0.891	0.013
	200	0.877	0.013
CMU	100	0.713	0.009
	150	0.623	0.009
	200	0.694	0.012
	250	0.717	0.011
	300	0.707	0.010

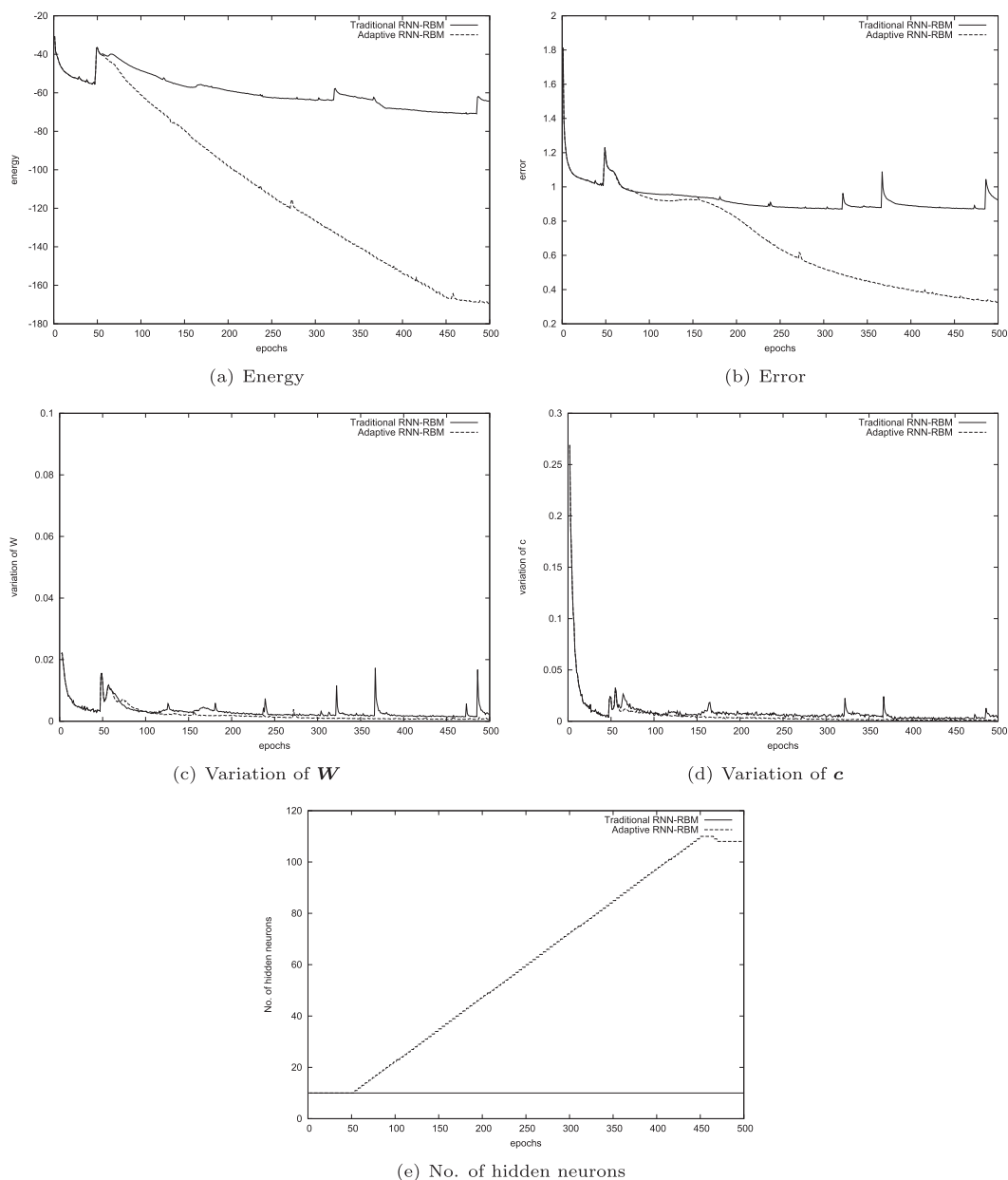


Fig. 12 Learning results on CMU (layer 1)

それぞれ学習の各反復に対する (1) 式によるエネルギー、誤差, (6) 式による W と c の変分, 隠れニューロン数を示している. RT では, Fig. 11 (c), Fig. 11 (d) に示すように 40 回目や 55 回目などの反復においてパラメタの変分が大きく振動している箇所が見られた. このことが原因となって, Fig. 11 (a) や Fig. 11 (b) の学習後半 (100 回目以降の反復) で示すように, RT ではエネルギーや誤差が小さな値にならなかったと考えられる. 一方, RA では, 40 回目の反復まで, RT と同様にパラメタの変分が振動している部分が見られたが, それ以降の学習では, Fig. 11 (e) に示すように, 適切と考えられるニューロン数の生成と消滅が行なわれたため, パラメタの変分, エネルギー, 誤差が徐々に小さな値に変化した. このような傾向は, **Fig. 12** に示すように, データセット CMU

でも見られた.

Table 1 はデータセット Nottingham と CMU のテストデータに対する RT の予測精度を示している. **Table 2** は, RA に対し同様の実験結果の一部 (パラメタの組み合わせの中, 最も良い結果のみ) を示している. ここで, Table 1 と Table 2 における “N” は隠れニューロン数, “Ave.” と “Std.” はテストデータに対する予測精度 (単位: %) で, クロスバリデーションにおける平均と標準偏差を示している. “Init. Parameter” は初期パラメタで, “Obtained Structure” は学習後の構造を示している. Table 1 の RT の分類精度は, Nottingham に対し, 初期隠れニューロン数が 150 の場合 0.891, CMU に対し, 初期隠れニューロン数が 250 の場合 0.717 となり, 最も予測精度が高くなった. 一方, Table 2 の RA では, 多

Table 2 Prediction accuracy of adaptive RNN-RBM

Dataset: Nottingham					
Init. Parameter			Obtained Structure	Accuracy	
N	θ_G	θ_A	N	Ave.	Std.
50	0.001	0.001	204±08	0.957	0.008
		0.100	201±04	0.958	0.013
	0.005	0.001	190±07	0.965	0.009
		0.100	175±03	0.975	0.014
	0.010	0.001	181±02	0.967	0.014
		0.100	173±02	0.965	0.010
	0.050	0.001	170±07	0.959	0.007
		0.100	144±16	0.944	0.010
	0.100	0.001	112±01	0.923	0.007
		0.100	110±03	0.928	0.012
100	0.001	0.001	221±08	0.957	0.012
		0.100	181±07	0.954	0.008
	0.005	0.001	192±07	0.965	0.009
		0.100	183±03	0.978	0.010
	0.010	0.001	178±02	0.971	0.014
		0.100	171±02	0.977	0.010
	0.050	0.001	150±07	0.969	0.013
		0.100	143±16	0.958	0.012
	0.100	0.001	140±01	0.953	0.007
		0.100	142±03	0.963	0.014
150	0.001	0.001	211±08	0.957	0.011
		0.100	204±07	0.958	0.013
	0.005	0.001	193±07	0.958	0.009
		0.100	195±03	0.965	0.014
	0.010	0.001	181±02	0.968	0.012
		0.100	175±02	0.965	0.013
	0.050	0.001	157±07	0.949	0.009
		0.100	152±16	0.958	0.010
	0.100	0.001	153±01	0.953	0.007
		0.100	140±03	0.938	0.012

Dataset: CMU					
Init. Parameter			Obtained Structure	Accuracy	
N	θ_G	θ_A	N	Ave.	Std.
150	0.001	0.001	262±08	0.727	0.012
		0.100	251±07	0.728	0.013
	0.005	0.001	240±07	0.718	0.009
		0.100	243±03	0.715	0.009
	0.010	0.001	240±02	0.721	0.010
		0.100	201±02	0.715	0.013
	0.050	0.001	200±07	0.722	0.013
		0.100	199±16	0.715	0.009
	0.100	0.001	195±01	0.711	0.010
		0.100	190±03	0.717	0.009
200	0.001	0.001	281±08	0.718	0.012
		0.100	280±07	0.722	0.008
		0.005	260±07	0.719	0.009
	0.005	0.100	263±03	0.722	0.009
		0.001	250±02	0.722	0.014
	0.010	0.100	241±02	0.731	0.013
		0.001	230±07	0.725	0.013
	0.050	0.100	223±16	0.722	0.013
		0.001	220±01	0.723	0.007
	0.100	0.100	219±03	0.719	0.014
0.001		0.001	299±08	0.713	0.012
	0.100	285±07	0.719	0.008	
	0.005	289±07	0.706	0.010	
250	0.005	0.100	273±03	0.711	0.009
		0.001	280±02	0.714	0.007
	0.010	0.100	261±02	0.713	0.010
		0.001	280±07	0.719	0.013
	0.050	0.100	258±16	0.724	0.011
		0.100	279±01	0.719	0.005
0.100	0.100	255±03	0.721	0.014	

くのパラメタの組み合わせにおいて RT よりも高い予測精度が得られた。特に, $\theta_G = 0.001$, $\theta_A = 0.100$ に設定した場合, データセット (Nottingham, CMU) および初期ニューロン数に依存せず, RT よりも高い予測精度が得られた。最も高い予測精度は, Nottingham に対し, 0.978 ($N = 100$, $\theta_G = 0.005$, $\theta_A = 0.100$), CMU に対し, 0.731 ($N = 200$,

$\theta_G = 0.010$, $\theta_A = 0.100$) となり, RT の最良の結果よりも高い値となった。また, このとき RA で得られたニューロン数は, Nottingham に対し 183, CMU に対し 241 となったが, 同じ構造を初期値として RT に与えて学習した場合, 予測精度は Nottingham に対し 0.890, CMU に対し 0.717 となり, RA のほうが高い予測精度が得られた。すなわち, RA は適切なニューロン数を発見するだけでなく, 学習の精度自体も向上させる効果があることがわかった。

Table 3 は, Table 1 と Table 2 で示した RT, RA の最良の結果と RAL, DA, DAL の結果を示している。パラメタの組み合わせについては, Table 1 と Table 2 で得られた最良の結果を使用した。ここで, “Time(min)” は学習にかかった実行時間 (単位: 分) である。アンダーラインは, 学習係数の調整による学習を適用する前の RA と DA の最良の結果を示している。実験の結果, どちらのデータセットについても, RT および DT よりも, RA および DA のほうが高い予測精度を示すことができ, その値はデータセット Nottingham では, RA が 0.978, DA が 0.980, データセット CMU では, RA が 0.731, DA が 0.823 となった。**Table 4** は, DA の各層における予測精度を示している。層を生成することにより, 予測精度が徐々に向上していることがわかる。また, 構造適応型学習の後, 学習係数の調整による学習を行なった結果, 学習にかかる計算時間は増加したものの, $\lambda = 1.005$ (RAL-2, DAL) の場合, 予測精度が最も向上した。データセット Nottingham では, RA が 0.015, DA が 0.014, データセット CMU では, RA が 0.034, DA が 0.058 増加し, われわれの提案手法の中で, 最も高い精度となった。一方, RAL-1 で示すように, $\lambda = 1.001$ の場合は学習係数が低すぎたため, RA と予測精度は変わらなかった。逆に, RAL-3 や RAL-4 で示すように, $\lambda = 1.010$ や $\lambda = 1.015$ の場合は予測精度が悪くなった。これは, 学習係数が増加しすぎたため, 入力データ空間の確率分布を探索する範囲を超えてしまったためだと考えられる。また, RAL の学習精度に関し, 最良の結果となったパラメタ $\lambda = 1.005$ (RAL-2) の周辺の値について, 0.001 刻みで $\lambda = \{1.002, 1.003, 1.004, 1.006, 1.007, 1.008\}$ の値を調査したところ, 10 回の試行平均で, データセット Nottingham に対し, 順に 0.980, 0.990, 0.992, 0.991, 0.991, 0.982 となり, データセット CMU に対し, 順に 0.739, 0.763, 0.763, 0.763, 0.735, 0.731 となった。この結果から, 最も高い精度は $\lambda = 1.005$ の場合に得られたが, その周辺である $\lambda = \{1.004, 1.006\}$ の場合でも, この値に近い予測精度は得られた。なお, 学習係数の値を調整しなかった場合 ($\lambda = 1.000$), 減少させた場合 ($\lambda = 0.095$) において学習を行なったところ, データセット Nottingham および CMU とともに, RA の場合と予測精度は変わらなかった。

5. おわりに

Deep Learning と呼ばれる機械学習の手法がさまざまな分野で高い能力を示しており, 現実世界の問題に応用されてい

Table 3 Prediction accuracy ($\lambda = \{1.001, 1.005, 1.010, 1.015\}$)

Dataset: Nottingham					
Model	Parameter	λ	Accuracy		Time(min)
			Ave.	Std.	
RT	Traditional RBM (Initial Neurons: 150)	-	0.891	0.013	71.5
RA	Adaptive RBM (Initial Neurons: 100, $\theta_G = 0.005, \theta_A = 0.100$)	-	<u>0.978</u>	0.010	70.3
RAL-1	Adaptive RBM (Tuning the learning rate with λ)	1.001	0.978	0.012	87.7
RAL-2	Adaptive RBM (Tuning the learning rate with λ)	1.005	0.993	0.011	88.3
RAL-3	Adaptive RBM (Tuning the learning rate with λ)	1.010	0.971	0.011	87.8
RAL-4	Adaptive RBM (Tuning the learning rate with λ)	1.015	0.970	0.010	87.9
DT	Traditional DBN (Layers: 5)	-	0.898	0.010	315.2
DA	Adaptive DBN (Layers: 5, $\theta_{L1} = 0.10, \theta_{L2} = 0.10$)	-	<u>0.980</u>	0.008	294.5
DAL	Adaptive DBN (Tuning the learning rate with λ)	1.005	0.994	0.009	357.3
Dataset: CMU					
Model	Parameter	λ	Accuracy		Time(min)
			Ave.	Std.	
RT	Traditional RBM (Initial Neurons: 250)	-	0.717	0.011	53.8
RA	Adaptive RBM (Initial Neurons: 200, $\theta_G = 0.010, \theta_A = 0.100$)	-	<u>0.731</u>	0.013	52.6
RAL-1	Adaptive RBM (Tuning the learning rate with λ)	1.001	0.731	0.009	68.4
RAL-2	Adaptive RBM (Tuning the learning rate with λ)	1.005	0.765	0.012	68.5
RAL-3	Adaptive RBM (Tuning the learning rate with λ)	1.010	0.706	0.012	68.0
RAL-4	Adaptive RBM (Tuning the learning rate with λ)	1.015	0.702	0.010	68.1
DT	Traditional DBN (Layers: 6)	-	0.708	0.011	312.6
DA	Adaptive DBN (Layers: 6, $\theta_{L1} = 0.10, \theta_{L2} = 0.10$)	-	<u>0.823</u>	0.009	294.5
DAL	Adaptive DBN (Tuning the learning rate with λ)	1.005	0.881	0.010	362.1

Table 4 Prediction accuracy for each layer in adaptive RNN-DBN

Dataset: Nottingham		
No. of layers	Accuracy	
	Ave.	Std.
1	0.978	0.010
2	0.979	0.012
3	0.979	0.012
4	0.980	0.010
5	0.980	0.008
Dataset: CMU		
No. of layers	Accuracy	
	Ave.	Std.
1	0.731	0.013
2	0.765	0.012
3	0.789	0.013
4	0.818	0.010
5	0.822	0.010
6	0.823	0.009

る。Deep Learning の手法として、畳込みニューラルネットワークがよく使われているが、計算量が多いことと、パラメタの設定が難しいと考えられ、学習モデルの構築には多くの経験が必要である。本研究では、統計的なモデルで、ネットワーク構造がCNNに比べて簡素であるRBM, DBNに着目し、学習中にニューロンの生成/消滅を行なうことで適切な構造を求める構造適応型DBNおよびこれを用いた時系列学習モデルを開発している。本論文では構造適応型学習により適切なネットワークを求めた後に、学習係数を調整しながらネットワークの再学習(微調整)を行なうことでより高い予測精度を実現するネットワークを構築した。今回は λ を用いて

学習係数を単調に増加させたが、今後はネットワークの状態や予測精度に関する影響を考慮した上で、適切な学習係数を自動で調整する手法を開発する。

謝辞 本研究開発は総務省SCOPE(受付番号:162308002)の委託およびJSPS 科研費(課題番号:JP17J11178)の助成を受けたものです。また有益なご助言をいただきました査読者に感謝の意を表します。

参 考 文 献

- 1) T. Lyons and M. Skitmore: Project risk management in the Queensland engineering construction industry: A survey, *International Journal of Project Management*, **22**-1, 51/61 (2012)
- 2) Y. Bengio: Learning Deep Architectures for AI, *Foundations and Trends in Machine Learning archive*, **2**-1, 1/127 (2009)
- 3) Q.V. Le: Building high-level features using large scale unsupervised learning, *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 8595/8598 (2013)
- 4) Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L.D. Jackel: Backpropagation applied to handwritten zip code recognition, *Neural Computation*, **1**-4, 541/551 (1989)
- 5) G.E. Hinton: A Practical Guide to Training Restricted Boltzmann Machines, *Neural Networks, Tricks of the Trade*, Lecture Notes in Computer Science (LNCS, **7700**), 599/619, Springer (2012)
- 6) G.E. Hinton, S. Osindero and Y. Teh: A fast learning algorithm for deep belief nets, *Neural Computation*, **18**-7, 1527/1554 (2006)
- 7) S. Kamada and T. Ichimura: An Adaptive Learning Method of Restricted Boltzmann Machine by Neuron Generation and Annihilation Algorithm, *Proc. of 2016 IEEE*

- International Conference on Systems, Man, and Cybernetics (IEEE SMC 2016), 1273/1278 (2016)
- 8) A. Krizhevsky: *Learning Multiple Layers of Features from Tiny Images*, master's thesis, University of Toronto (2009)
 - 9) 鎌田, 市村: 忘却機能を持った Deep Belief Network による構造適応型学習, 計測自動制御学会第9回コンピュータ・インテリジェンス研究会講演論文集, 92/97 (2016)
 - 10) S. Kamada and T. Ichimura: A Structural Learning Method of Restricted Boltzmann Machine by Neuron Generation and Annihilation Algorithm, *Neural Information Processing*, Lecture Notes in Computer Science (LNCS, **9950**), 372/380, Springer (2016)
 - 11) S. Kamada and T. Ichimura: An Adaptive Learning Method of Deep Belief Network by Layer Generation Algorithm, Proc. of 2016 IEEE Region 10 Conference (TENCON), 2971/2974 (2016)
 - 12) Y. Bengio, P. Simard and P. Frasconi: Learning long-term dependencies with gradient descent is difficult, *IEEE Transactions on Neural Networks*, **5**-2, 157/166 (1994)
 - 13) I. Sutskever, G.E. Hinton and W.T. Graham: The Recurrent Temporal Restricted Boltzmann Machine, Proc. of Advances in Neural Information Processing Systems 21 (NIPS 2008), 1601/1608 (2009)
 - 14) N. Boulanger-Lewandowski, Y. Bengio and P. Vincent: Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription, Proc. of the 29th International Conference on Machine Learning (ICML2012), 1159/1166 (2012)
 - 15) 鎌田, 市村, 丸山: 構造適応型 Deep Belief Network による時系列データの分析, 計測自動制御学会第10回コンピュータ・インテリジェンス研究会講演論文集, 73/78 (2016)
 - 16) T. Ichimura and S. Kamada: Adaptive Learning Method of Recurrent Temporal Deep Belief Network to Analyze Time Series Data, Proc. of the International Joint Conference on Neural Networks (IJCNN 2017), 2346/2353 (2017)
 - 17) D.H. Ackley, G.E. Hinton and T.J. Sejnowski: A Learning Algorithm for Boltzmann Machines, *Cognitive Science*, **9**-1, 147/169 (1985)
 - 18) G.E. Hinton: Training products of experts by minimizing contrastive divergence, *Neural Computation*, **14**-8, 1771/1800 (2002)
 - 19) D. Carlson, V. Cevher and L. Carin: Stochastic Spectral Descent for Restricted Boltzmann Machines, Proc. of the Eighteenth International Conference on Artificial Intelligence and Statistics, 111/119 (2015)
 - 20) 鎌田, 藤井, 市村: Lipschitz 連続による Restricted Boltzmann Machine のエネルギー安定性に関する一考察, 2015 IEEE SMC Hiroshima Chapter Young Researchers WorkShop, 53/56 (2015)
 - 21) 佐藤, 中野: 探索枝刈りによる複素多層パーセプトロン探索の高速化, 計測自動制御学会第5回コンピュータ・インテリジェンス研究会講演論文集, 91/96 (2014)
 - 22) L. Bruzzone and D.F. Prieto: A technique for the selection of kernel-function parameters in RBF neural networks for classification of remote-sensing images, *IEEE Trans. Geoscience and Remote Sensing*, **37**-2, 1179/1184 (1999)
 - 23) S.E. Fahlman and C. Lebiere: The cascade-correlation learning architecture, Proc. of Advances in Neural Information Processing Systems 2 (NIPS 1989), 524/532 (1990)
 - 24) T. Ichimura, S. Oeda, M. Suka and K. Yoshida: A learning method of immune multi-agent neural networks, *Neural Computing and Applications*, **14**-2, 132/148 (2005)
 - 25) 鎌田, 市村, 原: ニューロン生成/消滅アルゴリズムによる構造適応型 Restricted Boltzmann Machine, 計測自動制御学会第8回コンピュータ・インテリジェンス研究会講演論文集, 90/96 (2015)
 - 26) Y. LeCun, C. Cortes and C.J.C. Burges: THE MNIST DATABASE of handwritten digits, <http://yann.lecun.com/exdb/mnist/> (2017年9月21日閲覧)
 - 27) Y. Bengio, L. Pascal, P. Dan and H. Larochelle: Greedy Layer-Wise Training of Deep Networks, *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, 153/160 (2007)
 - 28) D.E. Rumelhart, G.E. Hinton and R.J. Williams: Learning internal representations by error propagation, *Parallel Distributed Processing: Explorations in the microstructure of cognition*, **1**, 318/362, MIT Press (1986)
 - 29) G.D. Magoulas and M.N. Vrahatis: Adaptive Algorithms For Neural Network Supervised Learning: A Deterministic Optimization Approach, *Intl. J. of Bifurcation and Chaos*, **16**-7, 1929/1950 (2006)
 - 30) M. Riedmiller and H. Braun: A direct adaptive method for faster backpropagation learning: The RPROP algorithm, Proc. of IEEE International Conference on Neural Networks, **1**, 586/591 (1993)
 - 31) L.N. Smith: Cyclical Learning Rates for Training Neural Networks, Proc. of 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 464/472 (2017)
 - 32) T. Schaul, S. Zhang and Y. LeCun: No more pesky learning rates, Proc. of the 30th International Conference on Machine Learning (PMLR), **28**-3, 343/351 (2013)
 - 33) Nottingham: <http://www-etud.iro.umontreal.ca/~boulanni/icml2012> (2017年9月21日閲覧)
 - 34) CMU Graphics Lab Motion Capture Database: <http://mocap.cs.cmu.edu/> (2017年9月21日閲覧)

[著者紹介]

鎌田 真 (学生会員)



2014年県立広島大学大学院情報科学研究科を修了。NTTアドバンステクノロジー(株)を経て、2015年に広島市立大学大学院情報科学研究科博士後期課程に進学。2014年から県立広島大学の研究補助員(臨時職員)として勤務。2017年から日本学術振興会特別研究員(DC2)。Deep Learning, 自己組織化マップ, 人工免疫システムに関する研究開発に従事。

市村 匠 (正会員)



1997年桐蔭横浜大学大学院工学研究科博士後期課程修了。博士(工学)。広島市立大学情報科学部を経て2010年県立広島大学経営情報学部准教授。2014年同大学教授。2015年同大学地域連携センター長(兼職)。2017年同大学高度人工知能プロジェクト研究センター長(兼職)。Deep Learning, ニューラルネットワーク, 進化計算, 人工免疫システム, 医療診断支援システムなどの研究開発に従事。2011年からIEEE SMC Hiroshima ChapterのChairを4年間を務めた。