

ディープボルツマンマシンにおける動的なネットワーク構築手法に関する検討

A Study on the Dynamic Construction Method of Network for Deep Boltzmann Machine

金盛 克俊*¹
KATSUTOSHI KANAMORI

安井 清一*¹
SEIICHI YASUI

大和田 勇人*¹
HAYATO OHWADA

*¹ 東京理科大学理工学部経営工学科
Tokyo University of Science, Faculty of Science and Technology

Deep learning provide a high learning performance, but also has a problem that it takes many time to learn, because of using a complex network. Moreover, finding the best configuration network is not easy, and little attention has been given the methodology. Our purpose is to achieve the dynamic construction method of network for Deep Boltzmann Machine. We propose a method to build a network adaptively and dynamically delete nodes and arcs in Deep Boltzmann Machine. This method is achieved by applying of node delete algorithm and node fusion algorithm. We defined these algorithms, and had the experiment using random data. And, we discussed about the result and future works.

1. はじめに

近年、深層学習(ディープラーニング)というキーワードとともに多層のニューラルネットワークが注目を集めている。ディープボルツマンマシン(Deep Boltzmann Machine:DBM) [Salakhutdinov 09]はその実装の一つである。

DBM は、リストラクテッドボルツマンマシン(Restricted Boltzmann Machine: RBM)を積み重ねて構築した多層のニューラルネットワークである。すなわち、各層のノードは確率変数であり、層内の確率変数、および、層間の確率変数の交互作用のみを含む確率モデル(生成モデル)によってデータ集合の特徴を表す深層学習である。入力層は観測変数、隠れ層は潜在変数に対応するので、周辺化によりネットワークの重みを教師なしで学習できる。したがって、DBM ではネットワークの重みを教師なしで学習を行った後、ニューラルネットワークとして誤差逆伝搬法で予測(識別)器を構築する。この初めの教師なし学習は事前学習と呼ばれる。

DBM の事前学習は、DBM を構成する各 RBM の学習を順次行うことにより実現される。RBM の学習によく用いられるのが RBM の条件付き独立性を利用した CD(Contrastive Divergence)法 [Hinton 02]である。

しかし、画像認識や音声認識などの実用的なニューラルネットワークには通常大規模なネットワークが用いられ、膨大な計算時間とリソースが必要である。必要最小限かつ最適なネットワークを求めることは簡単ではなく、その方法論については十分な議論は成されていない。

DropOut [Hinton 12] はノードをランダムに削除しながら学習して行き、複数のモデルを生成し、総合することによって過学習を防ぐモデルアベリッジ手法であるが、タスクに対して柔軟に、効率的に最適なネットワークの構成を求めるものではない。

本研究の目的は、多層のニューラルネットワークにおいてネットワークを動的に構成する手法を提案し、その検証を行うことである。

本研究ではディープボルツマンマシンにおいて、事前学習時にノードの削除を動的に行うことにより適応的にネットワークを構築する手法を提案し、その影響に関する検討を行う。

このような適応的なネットワークの構成を行うことにより、事前学習によって適切なネットワークを自動的に構成することが期待される。

2. 提案手法

本論文で提案するネットワーク構成手法は、DBM において事前学習を行いながらネットワーク構造を動的に更新する手法である。

本論文では、ノードの統合と削除という2つの操作によってネットワークを更新する手法を提案する。ノードの統合は似ているノード同士を統合して1つのノードにまとめる操作であり、ノードの削除はそのノードに関する重みが非常に小さく他に与える影響が小さいと考えられるものをネットワークから取り除く操作である。

このような操作が適切に行われることにより、不要なノードが削除され、後で行われる誤差逆伝搬等のニューラルネットワークの学習の計算時間の短縮やより高い精度を得るニューラルネットワークの自動構成の実現が期待される。

DBM は多層のニューラルネットワークであるので、可視層のノードは入力変数そのものに対応し、最上層は出力に対応する。よって本手法はそれ以外の層においてノードの削減を行うことにより適切なネットワークを構成する手法である。

2.1 ノードの統合

ここでは可視層と最上層以外の層について同じ層に存在するノード間の距離を定義し、その距離が近いものを類似した役割を持つノードとみなして統合する操作を定義する。

まず、同じ層に存在するノード n_1, n_2 間の距離 $d(n_1, n_2)$ を次のように定義する。

$$d(n_1, n_2) = |b_1 - b_2| + \sum_i |w_{1i} - w_{2i}| + \sum_j |w_{j1} - w_{j2}|$$

ただし、 b_k はノード n_k のバイアスであり、 w_{kl} はノード n_k, n_l 間のアークの重みである。この距離は、バイアス項の値や同じノードへの重みの差の絶対値の総和である。バイアスの値が同じで

あるほど距離は小さく、また他のノードとの関わりが同じであるほどノード間の距離は小さくなるといえる。

次に、2つのノードに対する統合ノードを定義する。ノード n_1 , n_2 を統合した新しいノード n_{12} 間はノード n_1, n_2 間と同じ層に挿入され、そのバイアスと重みはそれぞれ n_1 と n_2 間の値の和で与えられる。

ノード n_1, n_2 に対する統合ノード n_{12} のバイアス b_{12} と重み w_{12*}, n_{*12} は以下のように定義される。

$$\begin{aligned} b_{12} &= b_1 + b_2 \\ w_{12*} &= w_{1*} + w_{2*} \\ w_{*12} &= w_{*1} + w_{*2} \end{aligned}$$

2.2 ノードの削除

ノードの削除は、事前学習後のバイアス項の値と各重みの値がある程度0に近いノードについてネットワークから取り除く操作である。バイアスと全ての重みが0であるような仮想的なノード n_0 を考えると、あるノード n の削除のされやすさは $d(n, n_0)$ で表すことができる。

2.3 アルゴリズム

本論文で提案するアルゴリズムは、事前学習を行いながらノードの統合と削除を行うアルゴリズムである。DBMはRBMを積み重ねて構成された多層のニューラルネットワークであるため、RBMの列で表現することができる。

ここではDBMを $[rbm_1, rbm_2, \dots, rbm_m]$ と表すことにする。DBMの性質から、 rbm_i の隠れ層は rbm_{i+1} の可視層となっている。

提案手法のアルゴリズムは以下の通りである。

- 1) $i = 0$ とする
- 2) rbm_i と rbm_{i+1} についてCD法を用いてRBMの学習を行う。
- 3) rbm_i の隠れ層(rbm_{i+1} の可視層でもある)の各ノード n_j について、仮想ノード n_0 との距離 $d(n_j, n_0)$ を計算し、その平均を ave_0 とする。
- 4) rbm_i の隠れ層(rbm_{i+1} の可視層でもある)の各ノード n_j について、

$$d(n_j, n_0) < ave_0/th$$

であれば、この層からノード n_j を取り除く。ただし、 th は閾値パラメータである。

- 5) rbm_i の隠れ層(rbm_{i+1} の可視層でもある)の各ノードの全ての組み合わせ n_a, n_b を順に調べ、

$$d(n_a, n_b) < ave_0/th$$

であればノード n_a, n_b を取り除き、統合ノード n_{ab} を挿入する。

- 6) $i = i + 1$ とし、 $i < m$ であれば2.へ戻る

3. 実験

提案手法の動作を確認するため、簡単なDBMを用いて実験を行う。ここではFigure 1に示すような3層のDBMを用いる。

ここで用いるのは入力層は6次元、最上層は2次元のDBMであり、中間に10個のノードを持つ層が1つ存在している。図では、アークの線の太さが重みの絶対値の大きさを表し、重みが正であれば赤線、負であれば青線で表現されている。Figure 1は初期状態のため、ノードのバイアス値が全て0となっているが、本研究で開発したこの視覚化ツールではバイアス値によってノードの大きさが変わる。この例は実験結果で確認できる。

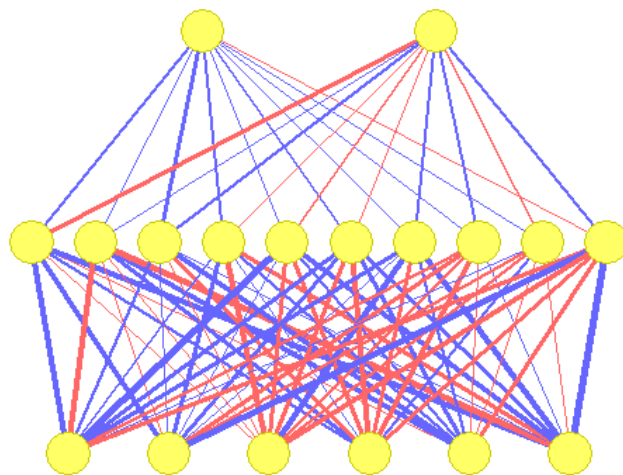


Figure 1 : 初期状態のネットワーク

このネットワークを用いて通常のCD法を用いたDBMの学習後のネットワークと、提案手法を適用した後のネットワークの状態を比較する。

実験用のトレーニングデータとして、入力次元6次元に合わせたランダムなデータを生成して与えた。

Figure 2は提案手法を用いず、ノードの数に変更がないまま学習が行われた結果である。ノードの大きさはバイアス値の大きさを表している。それぞれのアークについて、重みが強化されたところやバイアス値の変化などが確認できる。

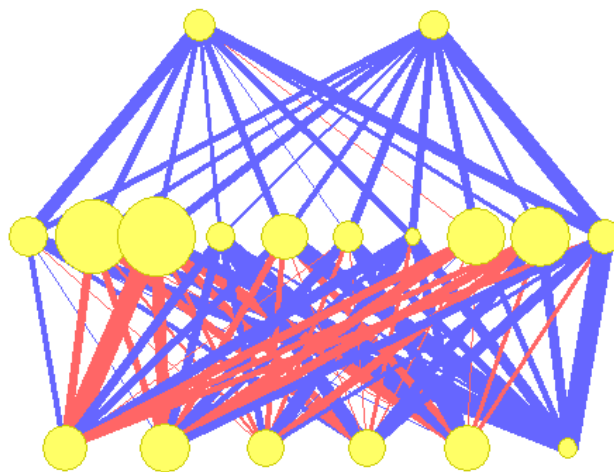


Figure 2 : 既存手法の学習結果

Figure 3は提案手法を適用した結果である。ただし、アルゴリズムにおけるパラメータ th の値は1.5とした。ノードの削除が1度、ノードの統合が2度行われ、第2層のノードが10個から7個に減っている。

提案手法によりノードを削減することができたが、これが元のネットワークの性質を損ねていないか検証するために、学習後の各ネットワークにおいて、最上層の値を与えた上での可視層の期待値を求めて比較する。

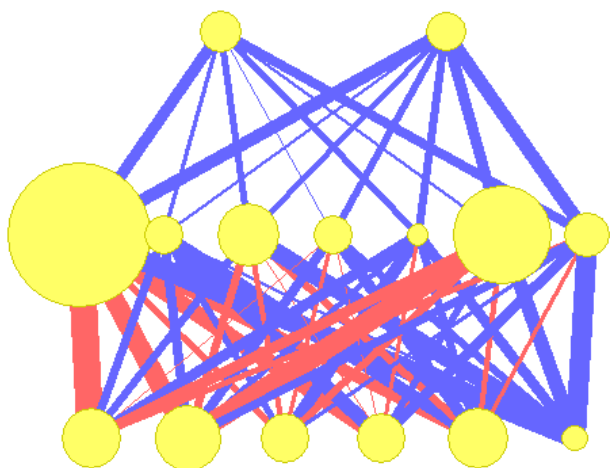


Figure 3 : 提案手法の学習結果

Figure 2 のネットワークにおける結果は表 2 の通りである.

表 1: 既存手法の結果

最上層	可視層
[0,0]	[0.60, 0.63, 0.50, 0.43, 0.54, 0.13]
[0,1]	[0.59, 0.63, 0.50, 0.43, 0.54, 0.14]
[1,0]	[0.60, 0.63, 0.50, 0.44, 0.54, 0.14]
[1,1]	[0.60, 0.63, 0.50, 0.44, 0.54, 0.15]

同様に, Figure 3 のネットワークにおける結果は表 3 の通りである.

表 2: 提案手法の結果

最上層	可視層
[0,0]	[0.64, 0.66, 0.51, 0.45, 0.56, 0.13]
[0,1]	[0.63, 0.65, 0.51, 0.46, 0.55, 0.15]
[1,0]	[0.63, 0.65, 0.51, 0.46, 0.55, 0.14]
[1,1]	[0.62, 0.65, 0.51, 0.46, 0.55, 0.16]

ネットワークの見た目の構造は大きく変わっているにも関わらず, 検証結果はほとんどその傾向を失っていない結果となった.

4. 結論

ディープボルツマンマシンにおける最適なネットワーク自動構築のため, 事前学習時に適応的にノードを統合, 削除する手法を提案した. 実験結果はこの手法によって実際にノードが削除されたにも関わらず, ネットワークのモデルの再現能力に関してほぼ変わらないことを示した.

不要なノードを削除することはニューラルネットワークの学習にかかる計算時間の削減や, 精度の向上につながると考えられる.

本論文の実験で用いたネットワークは極めて小さく単純なものであり, またデータについても実用的なものとはいえない.

今後はより大規模なネットワークへの適用や, 実用的な問題に対する検討を行う必要がある.

参考文献

- [Hinton 02] Hinton, G. E.: Training products of experts by minimizing contrastive divergence, Neural Computation, Vol. 8, No. 14, pp. 1771-1800 (2002).
- [Hinton 12] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. : Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv, 1207.0580 (2012)
- [Salakhutdinov 09] Salakhutdinov, R. R., and Hinton, G. E. : Deep Boltzmann Machines : Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, pp. 448-455 (2009)