

制限付きベイジアンネットワーク BESOM における 認識アルゴリズム OOBP

The OOBP Recognition Algorithm for the BESOM Bayesian Network

一杉裕志 *1 高橋 直人 *1
Yuuji Ichisugi Naoto Takahashi

*1産業技術総合研究所
National Institute of Advanced Industrial Science and Technology(AIST)

Recent research results in computational neuroscience imply that cerebral cortex is a Bayesian network. The functionality and performance of the Deep Learning technology may be improved if we take this hypothesis into account. We propose the OOBP algorithm, which accelerates loopy belief propagation by imposing a restriction on conditional probability tables in a Bayesian network.

1. はじめに

Deep Learning は深い層構造を持つニューラルネットワークであり、高い認識性能を発揮することで注目を集めている。この技術は、脳の視覚野の腹側経路の構造を模した神経回路モデルである、ネオコグニトロン [1] を源流とする。

最新の計算論的神経科学の知見を Deep Learning 技術に取り込めば、より本物の脳に近づき、より高機能・高性能になる可能性がある。計算論的神経科学における特に注目すべき進展は、「大脳皮質がベイジアンネットワーク [2] である」という仮説の登場である [8]*1。大脳皮質は、脳の中で知能にもっとも関係する重要な器官である。大脳皮質はベイジアンネットワークと機能と構造の面で多くの類似性を持ち、実際に様々な神経科学的現象がベイジアンネットワークを用いたモデルで再現されている (例えば [4][5][6][10][11])。

単に脳との類似性という理由だけでなく、技術的な観点からも、以下の点で、ベイジアンネットワークを用いた Deep Learning は有望だと思われる。

- ベイジアンネットワークは複数の事象の間の因果関係を確率により表現する知識表現の技術である。ベイジアンネットワークは信号源と観測データの間の因果関係を比較的少ないメモリで簡潔に表現できる場合があり、その場合は少ない計算量で様々な推論を行うことができる。
- 推論動作は、ネットワーク全体の情報を使って行われる。入力からのボトムアップの情報だけでなく、文脈からのトップダウンの予測の情報も用いたロバストな認識が行える [4]。したがって、feedforward 型ニューラルネットワークよりもはるかに高機能である。
- 階層的な生成モデルを素直に表現できるため、学習対象の事前知識が作り込みやすい。事前知識をネットワーク構造や、パラメタの事前分布の形で作り込むことで、学習の性能を上げられる可能性がある。これらの事前知識は、学習対象に対する工学的な観点からの分析だけでな

く、神経科学的知見 (領野間の接続構造や各領野が表現する情報の特徴) から得られる可能性がある。

以上の利点があるにも関わらず、ベイジアンネットワークを用いた大規模 Deep Learning があまり使われていない理由として、計算量の問題と、過適合・局所解の問題の2つが考えられる。

本論文では、計算量の問題を解決するために、制限された条件付確率表を前提に、ベイジアンネットワークによる認識時の計算量を大幅に抑えるアルゴリズム OOBP (Optimized Original Belief Propagation) を提案する。このアルゴリズムは、筆者が開発中の BESOM と呼ぶ大脳皮質モデル [7][12] に現在使われている。

なお、もう1つの過適合・局所解の問題については、大規模ベイジアンネットワークの表現力の高さがゆえに起きると考えられる。この問題は、パラメタに値を制約する事前分布を与えることで解決しつつある [12]。また、Deep Learning 技術で現在用いられている様々な正則化の技法も適用可能であると考えている。

2. BESOM の学習アルゴリズム

パラメタ θ のもとでの隠れ変数の値の組 \mathbf{h} と入力変数の値の組 \mathbf{i} との間の同時確率のモデルを $P(\mathbf{h}, \mathbf{i} | \theta)$ とする。また、時刻 t における入力変数の値の組を $\mathbf{i}(t)$ とする。各時刻の入力は i.i.d. (独立同分布) に従うと仮定すると、 θ のもとでの入力データの列 $\mathbf{i}(1), \mathbf{i}(2), \dots, \mathbf{i}(t)$ が生じる確率は以下のようになる。

$$\begin{aligned} & \prod_{i=1}^t P(\mathbf{i}(i) | \theta) \\ &= \prod_{i=1}^t \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(i) | \theta) \end{aligned} \quad (1)$$

学習の目的は、以下のようにパラメタを MAP 推定すること、すなわちパラメタ θ の事後確率を最大にすることである。

$$\theta^* = \operatorname{argmax}_{\theta} \left[\prod_{i=1}^t \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(i) | \theta) \right] P(\theta) \quad (2)$$

筆者のこれまでの研究では、認識時に MPE (Most Probable Explanation, 最も事後確率の高い隠れ変数の値の組み合わせ)

連絡先: 一杉裕志, 茨城県つくば市梅園 1-1-1 中央第2産業技術総合研究所, y-ichisugi@aist.go.jp

*1 参考: 「脳とベイジアンネットワーク」

<https://staff.aist.go.jp/y-ichisugi/besom/j-index.html>

の近似値を計算し、学習時には MPE を真の値とみなしてパラメタをオンライン学習する、という方法を取っていた [7]。

これに加えて、現在は、オンラインの EM アルゴリズムのような学習アルゴリズムも実装している*2。以下に概要を述べる。まず、時刻 t ごとに与えられる入力 $\mathbf{i}(t)$ とその時点でのパラメタ $\theta(t)$ を用いて、すべての親子ノードのペア X, Y のすべての値の組 x_i, y_j に対し条件付きの事後確率分布 $P(y_j|x_i, \mathbf{i}(t); \theta(t))$ を推定する。そして、それを真の観測データの分布とみなしてパラメタ $\theta(t+1)$ を計算する。事後確率分布の推定には loopy belief propagation の収束結果を用いる。現在は

$$P(y_j|x_i, \mathbf{i}(t); \theta(t)) = \alpha \lambda(y_j) w_{ij} \pi_{Y_i}(x_i) / BEL(x_i) \quad (3)$$

という式を用いている*3。ただし α は正規化係数、 w_{ij} は条件付確率 $P(y_j|x_i)$ を学習するパラメタである。

3. 条件付確率表のモデル

ベイジアンネットにおいて各ノードの条件付確率表を表現するためには、一般に親ノードの数 m に対して $O(2^m)$ 個のパラメタが必要になる。これは計算量・メモリ量の爆発や、過適合・局所解の原因になる。

そこで、条件付確率をより少ないパラメタで表現できるように、以下の形に制限する。

$$P(x|u_1, \dots, u_m) = \frac{1}{m} \sum_{k=1}^m w(x, u_k) \quad (4)$$

次節で述べる OOBP アルゴリズムは、導出においてこの形の条件付確率のモデルを仮定している。

$w(x, u_k)$ のもっとも簡単なものとしては、下記のものと考えられる*4。

$$w(x, u_k) = P(x|u_k) \quad (5)$$

4. 認識アルゴリズム OOBP

4.1 OOBP の導出

下記はオリジナルの Pearl の確率伝搬アルゴリズム [2] である。

$$\begin{aligned} BEL(x) &= \alpha \lambda(x) \pi(x) \\ \pi(x) &= \sum_{u_1, \dots, u_m} P(x|u_1, \dots, u_m) \prod_k \pi_X(u_k) \\ \lambda(x) &= \prod_l \lambda_{Y_l}(x) \\ \pi_{Y_l}(x) &= \beta_1 \pi(x) \prod_{j \neq l} \lambda_{Y_j}(x) \\ \lambda_X(u_k) &= \beta_2 \sum_x \lambda(x) \\ &\quad \sum_{u_1, \dots, u_m / u_k} P(x|u_1, \dots, u_m) \prod_{i \neq k} \pi_X(u_i) \end{aligned}$$

*2 ベイジアンネットにおけるバッチ学習の EM アルゴリズムは文献 [9] の pp.194-196 に概要が述べられており、それを参考にした。

*3 特殊なケースについてはこの式が成り立つことを確認しているが、一般の場合にも成り立つかどうかは現時点では不明である。

*4 より実際の脳の性質に近づけるためのモデルも実装している [12]。

ただし α, β_1, β_2 は正規化定数である。

このアルゴリズムを、式 (4) の形の条件付確率表を仮定したうえで、筆者の以前の研究 [5] と同様の考え方で変形する。近似は一切行わない。ここで、親ノードからのメッセージは下記のように正規化されることを前提とする。

$$\sum_{u_k} \pi_X(u_k) = 1 \quad (6)$$

なおこのとき、

$$\sum_{u_1, \dots, u_m} \prod_{i=1}^m \pi_X(u_i) = 1 \quad (7)$$

が成り立つ。

以上の前提のもとに、OOBP アルゴリズムを導出する。

まず、ノード U_k から X へのメッセージ $\kappa_{U_k}(x)$ を下記のように定義する。

$$\kappa_{U_k}(x) = \sum_{u_k} w(x, u_k) \pi_X(u_k)$$

$\pi(x)$ の計算式は下記のように変形できる。(途中から定数 $1/m$ を省略している。)

$$\begin{aligned} \pi(x) &= \sum_{u_1, \dots, u_m} P(x|u_1, \dots, u_m) \prod_i \pi_X(u_i) \\ &= \sum_{u_1, \dots, u_m} \left(\frac{1}{m} \sum_k w(x, u_k) \right) \prod_i \pi_X(u_i) \\ &\propto \sum_{u_1, \dots, u_m} \sum_k w(x, u_k) \prod_i \pi_X(u_i) \\ &= \sum_{u_1, \dots, u_m} \sum_k w(x, u_k) \pi_X(u_k) \prod_{i \neq k} \pi_X(u_i) \\ &= \sum_{u_1, \dots, u_m} w(x, u_1) \pi_X(u_1) \prod_{i \neq 1} \pi_X(u_i) \\ &\quad + \dots + \sum_{u_1, \dots, u_m} w(x, u_m) \pi_X(u_m) \prod_{i \neq m} \pi_X(u_i) \\ &= \sum_{u_1} w(x, u_1) \pi_X(u_1) \sum_{u_1, \dots, u_m / u_1} \prod_{i \neq 1} \pi_X(u_i) \\ &\quad + \dots + \sum_{u_m} w(x, u_m) \pi_X(u_m) \sum_{u_1, \dots, u_m / u_m} \prod_{i \neq m} \pi_X(u_i) \\ &= \sum_k \sum_{u_k} w(x, u_k) \pi_X(u_k) \sum_{u_1, \dots, u_m / u_k} \prod_{i \neq k} \pi_X(u_i) \\ &= \sum_k \sum_{u_k} w(x, u_k) \pi_X(u_k) \\ &= \sum_k \kappa_{U_k}(x) \end{aligned}$$

$\pi_{Y_l}(x)$ の計算式は下記のように変形できる。(ただし $\rho(x) = \pi(x) \lambda(x)$ と定義する。)

$$\pi_{Y_l}(x) = \beta_1 \pi(x) \prod_{j \neq l} \lambda_{Y_j}(x)$$

$$= \beta_1 (\pi(x) \prod_j \lambda_{Y_j}(x)) / \lambda_{Y_l}(x)$$

$$= \beta_1 \rho(x) / \lambda_{Y_l}(x)$$

$\lambda_X(u_k)$ の計算式の中に現れる式 $\sum_{u_1, \dots, u_m/u_k} P(x|u_1, \dots, u_m) \prod_{i \neq k} \pi_X(u_i)$ は下記のように変形できる。(途中、 $\pi(x)$ の計算式の変形と同じ考え方を使う。)

$$\begin{aligned} & \sum_{u_1, \dots, u_m/u_k} P(x|u_1, \dots, u_m) \prod_{i \neq k} \pi_X(u_i) \\ &= \sum_{u_1, \dots, u_m/u_k} \frac{1}{m} \left(\sum_{j \neq k} w(x, u_j) + w(x, u_k) \right) \prod_{i \neq k} \pi_X(u_i) \\ &= \frac{1}{m} \left[\sum_{u_1, \dots, u_m/u_k} \sum_{j \neq k} w(x, u_j) \prod_{i \neq k} \pi_X(u_i) \right. \\ & \quad \left. + w(x, u_k) \sum_{u_1, \dots, u_m/u_k} \prod_{i \neq k} \pi_X(u_i) \right] \\ &= \frac{1}{m} \left(\pi(x) - \sum_{u_k} w(x, u_k) \pi_X(u_k) + w(x, u_k) \right) \\ &= \frac{1}{m} \left(\pi(x) - \kappa_{U_k}(x) + w(x, u_k) \right) \end{aligned}$$

これを使うと $\lambda_X(u_k)$ の計算式は下記ようになる。

$$\begin{aligned} & \lambda_X(u_k) \\ &= \beta_2 \sum_x \lambda(x) \sum_{u_1, \dots, u_m/u_k} P(x|u_1, \dots, u_m) \prod_{i \neq k} \pi_X(u_i) \\ &= \frac{\beta_2}{m} \sum_x \lambda(x) \left(\pi(x) - \kappa_{U_k}(x) + w(x, u_k) \right) \end{aligned}$$

以上の結果を整理し、正規化定数を付け直し、さらに loop のあるネットワークに適用可能なように適宜添え字 $t, t+1$ を付けた結果の OOBP アルゴリズムは以下ようになる。

$$\begin{aligned} \lambda_{Y_l}^{t+1}(x) &= \beta_2 \sum_{y_l} \lambda^t(y_l) (\pi^t(y_l) - \kappa_X^t(y_l) + w(y_l, x)) \\ \lambda^{t+1}(x) &= \prod_{l=1}^n \lambda_{Y_l}^{t+1}(x) \\ \pi_{Y_l}^{t+1}(x) &= \beta_1 \rho^{t+1}(x) / \lambda_{Y_l}^{t+1}(x) \\ \kappa_{U_k}^{t+1}(x) &= \sum_{u_k} w(x, u_k) \pi_X^t(u_k) \\ \pi^{t+1}(x) &= \sum_{k=1}^m \kappa_{U_k}^{t+1}(x) \\ \rho^{t+1}(x) &= \lambda^{t+1}(x) \pi^{t+1}(x) \\ BEL^{t+1}(x) &= \alpha \rho^{t+1}(x) \end{aligned} \quad (8)$$

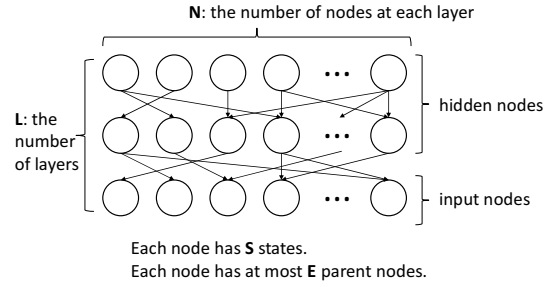


図 1: 計算量評価に用いたネットワークの構造

4.2 性能評価

以前に行った研究 [6] と同じ方法で、OOBP アルゴリズムの性能評価を行った。

OOBP は、素朴な実装での loopy belief propagation (以下、OBP, Original Belief Propagation と呼ぶ) と数学的に等価な計算をするので、事後確率の計算精度も同じはずである。計算精度は、小規模のネットワークで MPM (ノードごとの事後確率最大となる値の組) を計算し、厳密解と比較することで評価した。その結果、OOBP と OBP の計算精度は実際に同程度であり、アルゴリズム最適化と実装におそらく問題がないことが示唆された。

計算量に関しては、図 1 のネットワークにおいて、レイヤ数 $L=4$ 、各レイヤのノード数 $N=100$ 、各ノードの状態数 $S=4$ という条件で、ノードごとのおよその平均親ノード数 E の値を変えて、10 回のメッセージ伝搬の反復に必要な計算時間をデスクトップ PC 上で計測した。OBP は E の値が増えるにつれ急速に計算時間が増大した。(E=3 でも数十秒。) 一方 OOBP では E にほぼ比例する計算時間となった (図 2、数値はネットワークトポロジと条件付確率をランダムに変えて 100 回計測した平均)。

さらに、Deep Learning と同じネットワーク構造を持った制限付きベイジアンネットワークがパターン認識に利用できるかどうかを MNIST 手書き数字認識 *5 で試した。4 層のネットワーク (入力層 + 中間層 2 層 + 教師信号を与える最上位層) を用いて学習させたところ、(現在のところ pre-training などの工夫なしで) 90% 前後の認識精度を達成している。現状の認識精度は決してよくないが、その原因は、条件付確率表の制限によるものではなく、過適合・局所解にあると現時点では考えている。認識精度は、ネットワーク構造のチューニングや、微小な平行移動に対する不変性の作り込みなどによって、向上させる余地がある。

5. 関連研究

CD 法 [3] は制限付きボルツマンマシン (RBM) 上での学習を高速化する手法である。ボルツマンマシンは学習時に計算量の大きい negative phase が必要だが、CD 法はその計算量を近似計算により減らしている。それに対し、本論文では条件付確率表のモデルを線形和の形に制限することで、近似をせずに数学的な式の変形のみで、認識時の計算量を減らしている。なお、ベイジアンネットワークの学習においては、ボルツマンマシンの negative phase に相当するものはない。

*5 「MNIST handwritten digit database」
<http://yann.lecun.com/exdb/mnist/>

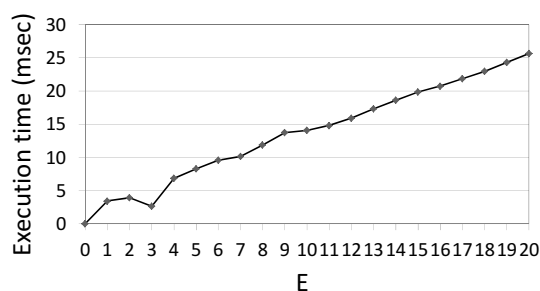


図 2: OOBP の実行時間

本論文と同様に、ベイジアンネットになんらかの制限を加えることで、計算量を減らすことはよく行われる。

Noisy-OR モデル [2] は、2 値変数の間の因果関係を簡潔に表現できる上、アルゴリズムを最適化することで計算量を少なくすることができる。Noisy-OR モデルは、「何かの事象が観測されたときは、考えられる複数の原因のどれかが成り立っているはずだ」という状況を素直に表現することができる。自然界にそのような状況は多いはずである。3. 節や [12] で述べた条件付確率表のモデルも、そのような状況の表現に適しているのではないかと考えている。

George と Hawkins [4] の大脳皮質モデルは、条件付確率表への制限ではなく、ネットワークを木構造に制限することで、メモリサイズおよび計算量の指数関数的爆発を避けている。しかし、木構造には表現力の制約が強すぎるといえる問題がある。

Hosoya [10] は Softmax を用いた条件付確率表のモデルを用いて、大脳皮質の初期視覚野の複数の現象を再現している。学習には確率伝搬アルゴリズムではなく MCMC を用いている。この学習方法と本論文で述べた学習方法のどちらが精度と計算量の点で優れているかは、現時点ではわからない。

Dura-Bernal [11] は、3. 節で述べたモデルと似た線形和の条件付確率表のモデルを用いて視覚野のモデルを構築している。メッセージ計算には本論文と同様の最適化を部分的に行っている。それに加え、メッセージに近似を入れることで計算量を減らすという工夫も行っている。この工夫は本論文のアルゴリズムと組み合わせることが可能であろう。

6. 強い人工知能の実現に向けた今後の課題

大脳皮質の機能を人工的に再現させることは、いわゆる「強い人工知能」の実現に向けた非常に重要なステップであると考えている。そのためには大脳皮質モデルを大規模かつ安定に動作させることが必要である。

現在は条件によって起きる loopy belief propagation の振動が精度を悪くする原因になっており、なんらかの対策が必要である。

脳全体の機能を再現させるためには、大脳皮質モデルだけでは当然不十分である。大脳皮質が脳内の他の器官とどう連携しているのか理解するための全脳アーキテクチャのモデル^{*6}を構築していくことが急務である。

参考文献

- [1] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4): 93-202, 1980.
- [2] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.
- [3] Hinton, G. E., Training Products of Experts by Minimizing Contrastive Divergence, *Neural Computation* 14 (8): pp.1771-1800, 2002.
- [4] George, D. Hawkins, J., A hierarchical Bayesian model of invariant pattern recognition in the visual cortex, In *proc. of IJCNN 2005*, vol. 3, pp.1812-1817, 2005.
- [5] Yuuji ICHISUGI, The cerebral cortex model that self-organizes conditional probability tables and executes belief propagation, In *Proc. of IJCNN 2007*, pp.1065-1070, Aug 2007.
<http://staff.aist.go.jp/y-ichisugi/besom/20070509ijcnn-paper.pdf>
- [6] Yuuji Ichisugi, Recognition Model of Cerebral Cortex based on Approximate Belief Revision Algorithm, In *Proc. of IJCNN 2011*, pp.386-391, 2011.
<http://staff.aist.go.jp/y-ichisugi/besom/2011ijcnn.pdf>
- [7] 一杉裕志, 「大脳皮質のアルゴリズム BESOM Ver.2.0」産業技術総合研究所テクニカルレポート AIST11-J00009, Sep 2011.
<http://staff.aist.go.jp/y-ichisugi/besom/AIST11-J00009.pdf>
- [8] 一杉裕志, 解説: 大脳皮質とベイジアンネット, *日本ロボット学会誌* Vol.29 No.5, pp.412-415, 2011.
- [9] Kevin B. Korb and Ann E. Nicholson, *Bayesian Artificial Intelligence*, Second Edition, CRC Press, 2011.
- [10] Haruo Hosoya, Multinomial Bayesian Learning for Modeling Classical and Nonclassical Receptive Field Properties, *Neural Computation*, Vol. 24, No. 8, pp. 2119-2150, 2012.
- [11] Salvador Dura-Bernal, Thomas Wennekers, Susan L. Denham, Top-Down Feedback in an HMAX-Like Cortical Model of Object Perception Based on Hierarchical Bayesian Networks and Belief Propagation *PLoS ONE*, Vol.7, No.11., 2012.
- [12] 一杉裕志, BESOM Ver.3.0 β 版のアルゴリズム, 2013.
<https://staff.aist.go.jp/y-ichisugi/besom/20130717algorithm.pdf>

*6 参考: 「全脳アーキテクチャ解明に向けて」

<https://staff.aist.go.jp/y-ichisugi/brain-archi/j-index.html>