ボルツマン学習と平均場近似

山梨大学工学部　宗久研究室　G04MK016 鳥居 圭太　平成 18 年 2 月 14 日

1. はじめに

ボルツマンマシンは学習可能な相互結合型ネットワークの代表的なものである．ボルツマンマシンには，学習のための統計平均を取る必要があり，結果を求めるまでに長い時間がかかってしまうという欠点がある．

そこで，学習の高速化のために，統計を取る２つのステップについて，以下のことを行う．まず１つ目のステップでは，付加する隠れ素子に対し，入力素子，出力素子，隠れ素子の値が線形分離になるようにする．この線形分離よって，入力素子，出力素子固定での統計を取る必要が無くなる．２つ目のステップ，つまり入力素子固定，出力素子，隠れ素子自由での統計を，平均場理論によって近似する．また線形応答項により精度向上を図る．この解析的手法により，時間大幅に短縮し，精度も維持できる．これらの方法を，ｎビットパリティ問題と文字認識問題で検証する．

2. ボルツマンマシン

ここでは，以下のようにボルツマンマシンにでてくる量を定義する．

素子 i の状態 s_i は２値．ここで i は１から N までである．結線重みについては素子 i から j への重みは w_{ij} と表し，また重みは対称であるので，$w_{ij} = w_{ji}$ となる．各素子は自身以外の素子から入力を受け取る．入力和は，

$$h_i = \sum_j w_{ij} s_j \qquad (1)$$

で与えられる．入力和 h_i に対する素子値は以下のように決定する．

$$s_i = \begin{cases} 1 & sig(h_i) \\ 0 & 1 - sig(h_i) \end{cases} \qquad (2)$$

$$sig(h_i) = \frac{1}{1 + e^{-h_i / T}} \quad （\ sigmoid\ 関数）$$

温度はＴとする．ここでエネルギーは，

$$\mathrm{H} = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j \qquad (3)$$

と定義できる．このとき，状態 s_i の出現確率（ボルツマン分布）は

$$P(\{s_i\}) = Ae^{-H(\{s_i\})/T} \qquad (4)$$

で得られる．上式のＡ全ての状態に関する出現確率の総和を１に合わせるための規格化の定数である．ボルツマンマシンの学習は次式で得られる．

$$w_{ij}^{new} = w_{ij}^{old} + \frac{\varepsilon}{T}(< s_i s_j >^{Fix} - < s_i s_j >^{Free}) \qquad (5)$$

ここで，ε は小さい正の数である．Ｆｉｘとは入力素子値，出力素子値を共に固定した状態であり，Ｆｒｅｅとは入力素子値を固定，出力素子値は固定していない状態である．重みの更新式は以下の誤差関数（クルバックのダイバージェンス）から勾配法に基づいている．ここで，Ｑは学習目標，Ｐは実際にボルツマンマシンを動作させてえられた確率である．

$$I = \sum_X Q(X) \sum_Y Q(Y \mid X) \log \frac{Q(Y \mid X)}{P(Y \mid X)} \qquad (6)$$

3. 学習の高速化

ボルツマンマシンの学習では，ボルツマン分布に近い出現確率を得られるまで統計をとらなければならず，多くの時間がかかる．

3.1. 隠れ素子固定による線形分離

そこで，入力素子値，出力素子値を共に固定した状態での平均値 $< s_i s_j >^{fix}$ を，あらかじめ線形分離ができる理想的な隠れ素子の値を決めておき，統計をとる処理を短縮することが出来る．

例えばＸＯＲ（２ビットパリティチェック）を隠れ素子１ビットで学習するとことを考える．この場合において，学習が終わったとき，隠れ素子値の組は，各入力に対して，以下のようなパターンを多く取ることが分かった．

０００１　００１０　０１００　１０００

１１１０　１１０１　１０１１　０１１１

これは，１０００のような組を取るときは，図１に示すように線形分離している（対応した真理値表を表１に示す）．

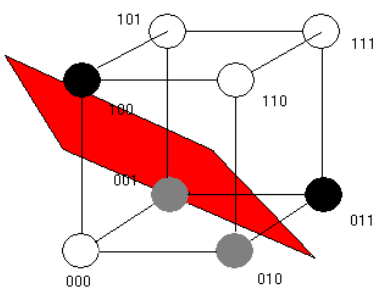

図１　ＸＯＲの線形分離

表１　ＸＯＲの線形分離真理値表

入力素子	入力素子	隠れ素子	出力素子
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0

しかし，次元が増えることによって，このような図から，隠れ素子を決めることは難しくなる.そこで，以下に示す線形分離アルゴリズムを使うことによって，様々な問題について，最小限ではないが，隠れ素子を付加し，線形分離を可能にする．このアルゴリズムの最大の利点は，隠れ素子を付加する必要があるかどうかを，判定できるということである．

・線形分離アルゴリズム

ステップ１：Ｎ個の素子（入力素子，出力素子の和）により，Ｍ個のユニットパターンを学習すると仮定する．それをＭ×Ｎ行列とする．

ステップ２：任意の４つの学習パターンを取り出し，排他的論理和の関係になっているかを判定し（同じ列，反転の関係にある列を削除，すべて１　または０を持つ列を削除して，残った４×３行列を見て判定する），線形分離不可であれば記憶する．

ステップ３：ステップ２から３を繰り返し，${}_M C_4$ 回判定を行う．判定の結果，どの４つの学習パターンも排他的論理和の関係を持たないとき，Ｍ×Ｎ行列による学習パターンは線形分離可能である．排他的論理和の関係があると判定されたとき，以下の処理を行う．

ステップ４:最も多く重複して排他的論理和の関係に絡んでいる学習パターン（行）に値１，そのほかの行に０を与えた列を（Ｍ×Ｎ）行列に加える．これにより，その学習パターンが絡んでいる全ての排他的論理和の関係が解消される．これをステップ２で記憶された排他的論理和の関係をもつ４つの学習パターンが全て無くなるまで繰り返すと，隠れ素子を付加した行列が出来上がる．

3.2. 平均場近似による高速化

(5)式の平均値を，統計平均を取る代わりに，平均場近近似で計算することを考える．これでＡＮＤやＯＲなどの問題においては，よい近似ができる．しかし，ＸＯＲなどの問題では有効でない．この違いは線形分離可能か，ということであると考えられる．

そこで，線形分離できるように隠れ素子を固定することによって，入力素子値を固定，出力素子値は固定していない状態での平均値$< s_i s_j >^{free}$を，統計をとらず，平均場近似によって求めた素子の平均値$< s_i >< s_j >$で近似して求め，統計をとる処理の短縮を図る．

平均場近似の概略は，sigmoid 関数を１次の線形と見て，以下のように近似をする．つまり，

$$< s_i >= \sum_{\{S_i\}} P(\{ s_i \})\ sig\ (\sum_j w_{ij} s_j) \quad (7)$$

を次式で近似する

$$\begin{aligned} &\Rightarrow\ sig\ \{(\sum_{\{S_i\}} P(\{ s_i \}))(\sum_j w_{ij} s_j)\} \\ &=\ sig\ (\sum_j w_{ij}\ < s_j >) \end{aligned} \quad (8)$$

に近似を行う．

以下に手順を示す.

・平均場近似アルゴリズム

ステップ１：まず、$< s_1 >, < s_2 >, \cdots, < s_N >$に初期値を与える．

ステップ２：入力和h_iの平均値$< h_i >$を

$$< h_i >= \sum_{j=1}^{N} w_{ij} < sj > \quad (9)$$

で求める．

ステップ３：次に求めた入力和の平均値$< h_i >$を使って，素子値の平均値$< s_i >$を以下の式で更新する．

$$< s_i >= sig(< h_i >) \quad (10)$$

ステップ４：$s_1, s_2, \cdots, s_N$を以上の２～３の手順で更新していく．

ステップ５：そして，更新した$< s_i >$を使って，２～４の手順を素子値の平均値$< s_i >$が収束するまで繰り返す.

4. 線形応答理論

線形応答理論を使って単純な平均場近似より，良い近似を求める．

学習における相関を求める際，

$$(A_{ij})^{-1} = \frac{\delta_{ij}}{1 - s_i^2} - w_{ij} \quad (11)$$

からA_{ij}を計算し，可視素子以外の部分で

$$< s_i s_j >^{\alpha} = s_i^{\alpha} s_j^{\alpha} + A_{ij}^{\alpha} \qquad i, j \in H \qquad (12)$$

という形で修正項を加える（ある状態α，H は隠れ素子のセットを表す）．これにより，通常の平均場近似よりよい近似が得られる．

5. 実験及び結果

以下のNビットパリティチェック問題と文字認識問題で，各ボルツマンマシンの比較を行う．

この実験では，全学習パターンを学習させてから重みを更新する一括更新法を用いる．つまり，２ビットパリティチェックの問題では，4 回の学習で 1 回重みを更新するものとする．素子の更新は１ビットずつ行う．収束条件は，10000 回の学習の間に，(6)式を使って求められた誤差が，規定以下になったところで学習を打ち切る．実験に使用したマシンの CPU は 1GHz である．今回の実験では，線形分離アルゴリズムよって判定し，図から最小に線形分離をして実験を行った．

5.1. 2ビットパリティチェック（XOR）

隠れ素子１ビットを，表１ように固定し，線形分離可能にし，動作実験を行った．それぞれの学習における誤差収束の様子を図２から図５に示す．表２に収束までの処理時間を示す．以下に示す学習の誤差収束の図は，縦軸平均誤差，横軸重み修正回数とする．平均誤差 0．003 以下で学習終了とした．

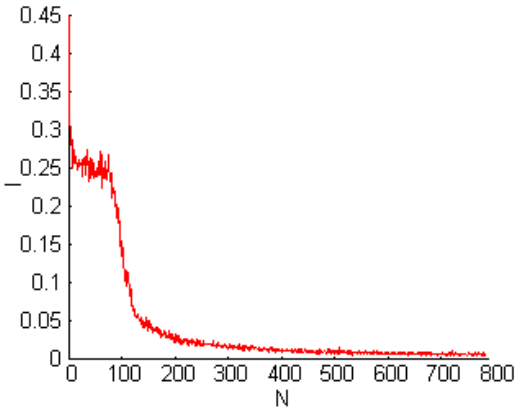

図２　従来のボルツマンマシンにおける誤差収束の様子（T＝0.5，ε＝0.2）

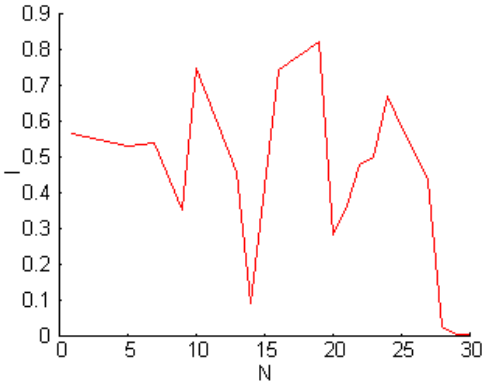

図３　理想的に線形分離を行ったボルツマンマシンにおける誤差収束の様子（T＝0.25，ε＝0.2）

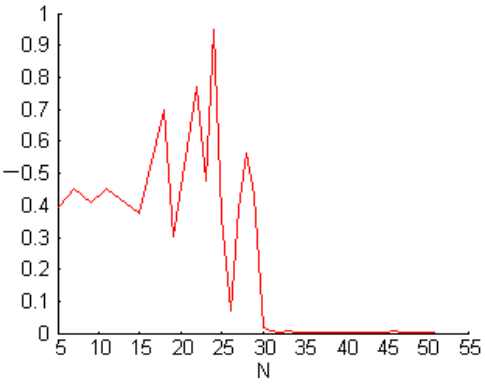

図４　理想的に線形分離し，平均場近似を使ったボルツマンマシンにおける誤差収束の様子（T＝0.25，ε＝0.2）

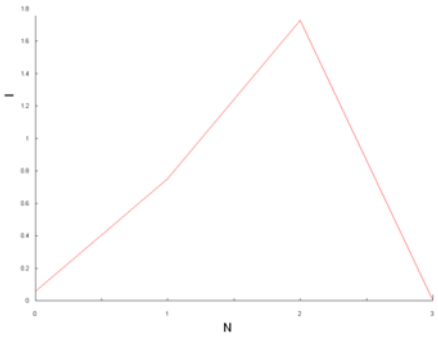

図５　理想的に線形分離し，平均場近似＋線形応答を使ったボルツマンマシンにおける誤差収束の様子（T＝0.5，ε＝1.0）

表２　実行時間

	重み更新回数	時間
従来型	783	7.505s
線形分離型	30	0.221s
線形分離＋平均場近似型	51	0.018s
線形分離＋平均場近似＋線形応答型	4	0.0016s

5.2. 3ビットパリティチェック

隠れ素子１ビットを固定し，線形分離可能にし，動作実験を行った（表略）．それぞれの学習における誤差収束の様子を図６から図９に示す．表３に収束までの処理時間を示す．平均誤差０．003 以下で学習終了とした．

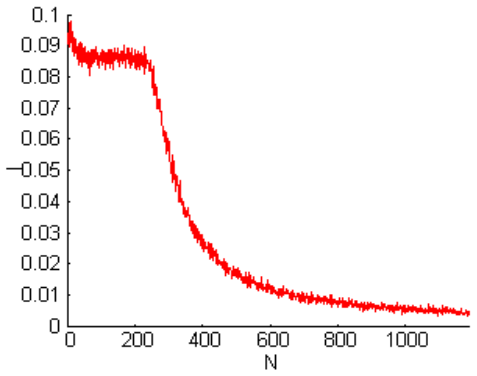

図６　従来のボルツマンマシンにおける誤差収束の様子（T＝0.5，　ε＝0.05）

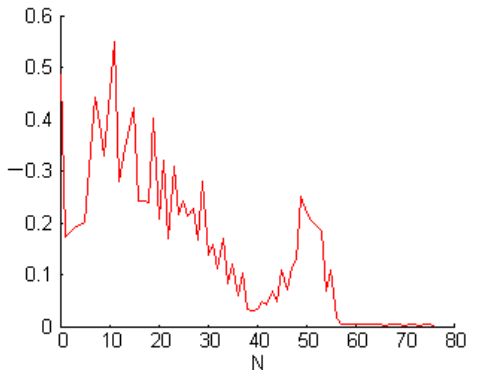

図７　理想的に線形分離を行ったボルツマンマシンにおける誤差収束の様子（T＝0.5，　ε＝0.2）

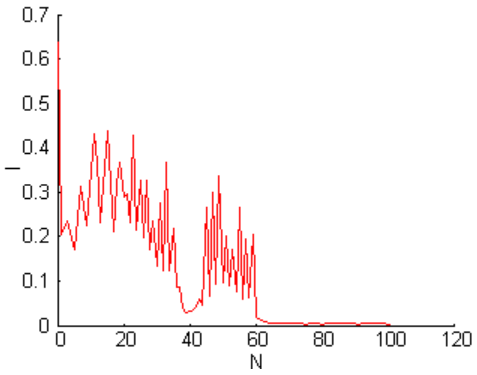

図８　理想的に線形分離し，平均場近似を使ったボルツマンマシンにおける誤差収束の様子（T＝0.5，ε＝0.2）

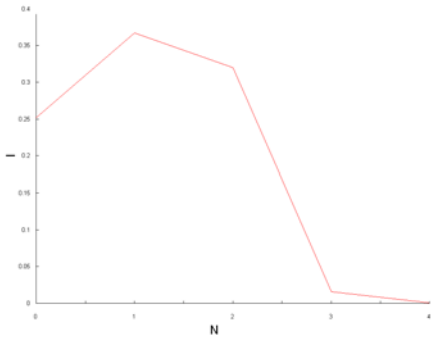

図９　理想的に線形分離し，平均場近似＋線形応答を使ったボルツマンマシンにおける誤差収束の様子（T＝0.5，　ε＝0.2）

表３　実行時間

	重み更新回数	時間
従来型	1200	27.920s
線形分離型	76	0.931s
線形分離＋平均場近似型	101	0.025s
線形分離＋平均場近似＋線形応答型	5	0.019s

5.3. 4ビットパリティチェック

隠れ素子２ビットを固定し，線形分離可能にし，動作実験を行った（表略）．それぞれの学習における誤差収束の様子を図１０から図１２に示す（従来のボルツマンマシンは 10000 回では収束しなかった）．表４に収束までの処理時間を示す．平均誤差０．003 以下で学習終了とした．

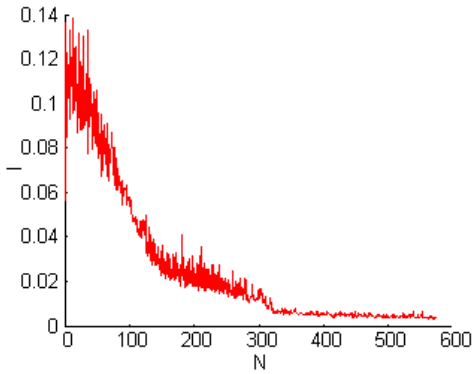

図１０　理想的に線形分離を行ったボルツマンマシンにおける誤差収束の様子（T＝0.25，　ε＝0.2）

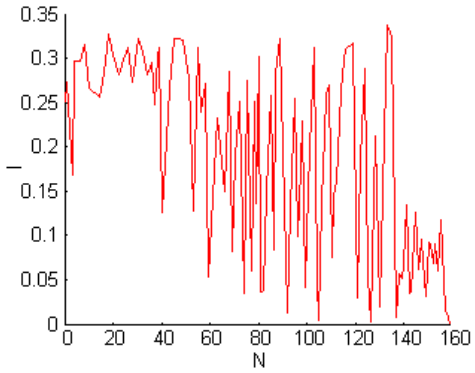

図１１　理想的に線形分離し，平均場近似を使ったボルツマンマシンにおける誤差収束の様子（T＝0.25，ε＝0.2）

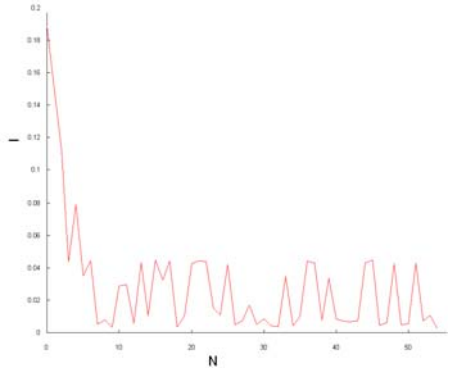

図１２　理想的に線形分離し，平均場近似＋線形応答を使ったボルツマンマシンにおける誤差収束の様子（T＝0.5，ε＝0．2）

表４　実行時間

	重み更新回数	時間
従来型	収束せず（10000 回）	1m24.826s
線形分離型	574	19.623s
線形分離＋平均場近似型	159	0.100s
線形分離＋平均場近似＋線形応答型	55	0.0018s

5.4. 6ビットパリティチェック

隠れ素子３ビットを固定し，線形分離可能にし，動作実験を行った（表略）．それぞれの学習における誤差収束の様子を図１３と図１４に示す．表５に収束までの処理時間を示す．全パターン合計誤差0.08以下で学習終了とした．

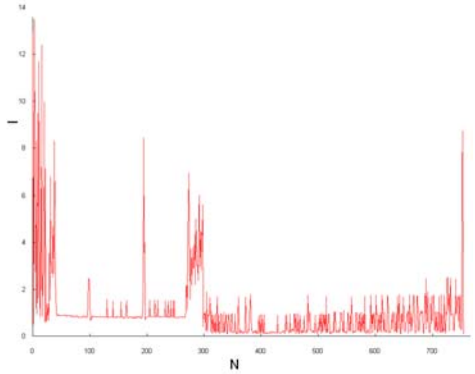

図１３　理想的に線形分離し，平均場近似を使ったボルツマンマシンにおける誤差収束の様子（T＝0.5，ε＝0.05）

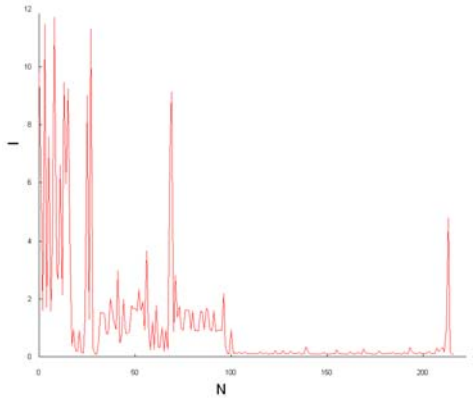

図１４　理想的に線形分離し，平均場近似＋線形応答を使ったボルツマンマシンにおける誤差収束の様子（T＝0.5，ε＝0．05）

表５　実行時間

	重み更新回数	時間
線形分離型	収束せず（10000 回）	—
線形分離＋平均場近似型	758	24.40s
線形分離＋平均場近似＋線形応答型	217	12.77s

5.5. 8ビットパリティチェック

隠れ素子４ビットを固定し，線形分離可能にし，動作実験を行った（表略）．それぞれの学習における誤差収束の様子を図１５に示す．表６に収束までの処理時間を示す．全パターン合計誤差０．１以下で学習終了とした．

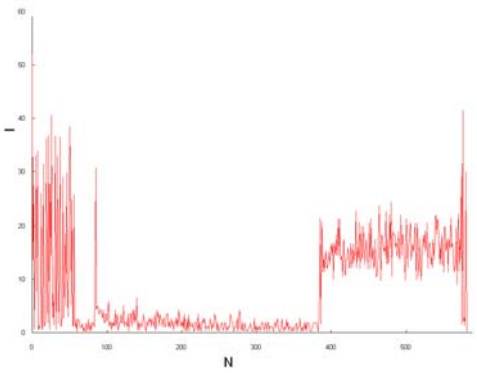

図１５　理想的に線形分離し，平均場近似＋線形応答を使ったボルツマンマシンにおける誤差収束の様子（Ｔ＝0.5, ε＝0．01）

表６　実行時間

	重み更新回数	時間
線形分離＋ 平均場近似型	収束せず (10000 回)	―
線形分離＋ 平均場近似＋ 線形応答型	583	5m58.3s

5.6. 文字認識問題

アルファベット１文字を５×５マスに描く．これをボルツマンマシンの学習パターンとし，学習を行う．平均場近似＋線形応答型の認識結果(1000 回中の成功回数)を示す．

Ａ：成功回数：678　Ｂ：成功回数：419　Ｃ：成功回数：596

Ｄ：成功回数：414　Ｅ：成功回数：339　Ｆ：成功回数：73

Ｇ：成功回数：782　Ｈ：成功回数：399　Ｉ：成功回数：621

Ｊ：成功回数：661　Ｋ：成功回数：579　Ｌ：成功回数：462

Ｍ：成功回数：44　Ｎ：成功回数：37　Ｏ：成功回数：656

Ｐ：成功回数：95　Ｑ：成功回数：841　Ｒ：成功回数：550

Ｓ：成功回数：499　Ｔ：成功回数：684　Ｕ：成功回数：477

Ｖ：成功回数：803　Ｗ：成功回数：731　Ｘ：成功回数：590

Ｙ：成功回数：777　Ｚ：成功回数：523

プログラム動作時間は２１．４２ｓ．

6. 考察

- 従来型

  問題が大きくなるにつれ，学習時間が膨大になる．原因は２つの統計処理．パラメータなどによる影響は少ない．

- 線形分離型

  線形分離を行って，２つある統計処理を１つ省略している．そのため，処理速度は従来型の半分ほどになっている．しかし，問題規模が増えることにより学習時間が膨大になる．処理時間には多少難があるが，収束安定性は高く，パラメータなどによる影響は少なく，全体として安定性はとても高い．線形分離プログラムの処理時間は，微々たる物といえる．

- 線形分離＋平均場近似型

  線形分離と平均場近似を利用することにより，統計処理が無く，処理速度はとても速い．しかし，誤差収束の安定性が低い．つまり，パラメータ（T，ε，素子値を０と１か±１か，など）の影響を大きく受け，解が得られないことがある．同様に従来型と線形分離型は，初期重みなどに余り大きな影響を受けないが，平均場近似型はとても大きな影響を受ける．誤差も単調減少せず，大きな問題ほど振動する．そのため，１回の学習は高速であるが，最適な初期重みやパラメータを見つけるために数回の試行が必要になる．だが，その試行回数を考えても，大きな問題で必要な時間は従来型，線形分離型より少ない．

- 線形分離＋平均場近似＋線形応答型

  近似補正の線形応答項を導入したことにより，精度が向上している．線形応答項は，小さな問題では効果は薄いが，大きな問題では誤差収束の安定性がより高まる．しかし，線形応答項の計算により，逆行列の計算が必要になった．このため規模の大きな問題では，規模に応じて処理時間が増える．また平均場近似を利用しているので，やはりパラメータや初期重みによって，誤差収束の様子が大きく変わる．

7. 参考文献

[1]J.Hertz, A.Krogh and R.G.Palmer：INTRODUCTION TO THE THEORY OF NEURAL COMPUTATION,
pp201-212,pp251-257 (Addison-Wesley publishing,
Massachusetts Menlo Park, 1991)

[2]H.J.Kappen and F.B.Rodriguez:Efficient Learning Using Linear Response Theory, pp1137-1156 (Massachusetts Institute of Technology, 1998)

[3]熊沢逸夫：学習とニューラルネットワーク，pp82-129（森北出版株式会社，東京，1998）

[4]伊藤大介：ボルツマンマシン学習の高速化（山梨大学修士論文，2004）

[5]伊藤大介　鳥居圭太　宗久知男：ボルツマンマシンの高速化（電子情報通信学会論文，2004）