ボルツマンマシンにおける並列性の抽出

5N－6

赤木琢磨　鶴田直之　谷口倫一郎　雨宮真人
九州大学

1 はじめに

ボルツマンマシンは最小値を探索する確率的なモデルである。一般的にその動作は「ランダムにひとつニューロンが選ばれ、そのニューロンは結合を持つ（おなじ制約に属する）すべてのニューロンの状態値を参照し、自分の次の状態値を確率的に決定する。状態を更新したニューロンは、直ちに自分の活性値を直接結合する他のニューロンに送る。その後、次のニューロンが選ばれ、同様に状態を更新するという動作を繰り返す。」と定式化される。

しかしボルツマンマシンをデータの依存関係に基づいて解析すると、ボルツマンマシンの制約は局所的であるため、依存関係を持たないニューロンどうしは並列に実行しても構わないことがわかる。そこで本論文では、前述の観点からボルツマンマシンの並列性の抽出方法を提案し、そのときの実行時間を確率モデルを用いて求める。

2 ボルツマンマシンの並列性

上記の定義により動作するボルツマンマシンで利用できる並列性は次の3つに分類できる。[3]

機能（状態更新、収束判定など）のパイプライン化 ボルツマンマシンでは収束状態が安定状態であるので、収束決定の情報が数フェーズ遅れて状態更新プロセスに入力されても構わない。そこで、嘘つきデータを用いることによって、収束判定を状態更新とパイプライン処理することができる。

繰り返しフェーズのオーバラップ 状態更新と収束判定をパイプライン化している場合、状態更新の並列処理にばらつきがあるときでも、繰り返し処理をオーバラップして実行できる。

ニューロン並列 直接結合を持たないニューロンどうしの処理をオーバラップさせることによって並列性を抽出するものである。最初に述べたように、ボルツマンマシンでは制約が局所的であるため、オーバラップさせることができるニューロンはかなり多い。そこで以下では、ニューロン並列の抽出法について述べる。

3 並列性の抽出

ボルツマンマシンにおけるニューロン並列の抽出法について考える。まず、すべてのニューロンにランダムにシリアル番号を付ける。そして、同じ制約の中のニューロンはシリアル番号の小さいものから順に状態更新を行なう。すべてのニューロンが状態更新を一回ずつ終えると、またランダムにシリアル番号を付け直し状態更新を行なうという動作を繰り返す。ここで、番号付けエージェントが必要になるが、これは階層化し収束判定と共に機能パイプラインの一段として挿入することにより、実行時間に影響を与えずに実行できる。

順番を付けたニューロンをデータの依存関係に従って自律的に実行させると、各ニューロンは、自分の属する制約に含まれるニューロンの内、シリアル番号が自分より小さいものすべての状態更新が終ると状態更新を行なえるので並列性が自動抽出できる。これを図に表すと図1、2のようになる。まず縦軸に沿って番号順にニューロンを並べ、横軸に沿ってそのニューロンが属する制約の列にブロックを置く（図1）。次に縦軸をニューロンの状態更新の実行順序とみなし、同じニューロンのブロックのいずれかが下のブロックに接するまで行を下げていくと、同じ制約に属さないニューロンどうしが同時に実行を行ない、図2のように、並列性が自動抽出される。このときのブロックの高さが、ボルツマンマシンの並列性を抽出したときの実行時間である。そこで、次章では確率モデルを用いて、実行時間の期待値を求める。

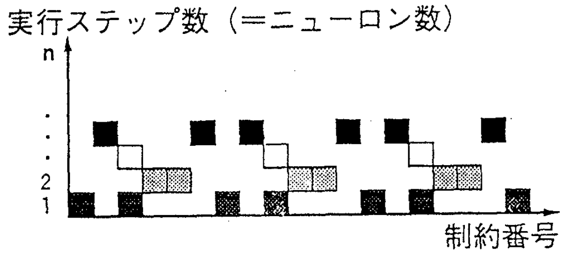

図 1: 並列性を考慮する前のブロック図例

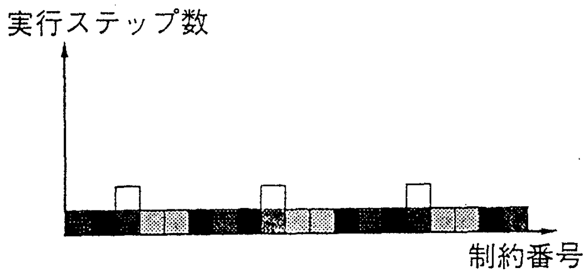

図 2: 並列性を抽出した後のブロック図例

4 実行時間の期待値

今、ひとつの制約に含まれるニューロンの数がM個であるボルツマンマシンのニューロンに、全くランダムにシリアル番号を付けてデータの依存関係に従って自律的に実行することを考える。この場合、並列性を考慮する前のブロック図は図3のようになる。ここでNはニューロンの総数、Rは制約の総数を表し、各制約にはそれぞれM個ずつのブロックがランダムに置かれている。モデル全体のニューロンの一回の状態更新を1フレーズとする。逐次的にこのモデルを実行すると1フレーズにNステップかかる。こ

Extraction of parallelism from Boltzmann Machine
Takuma Akagi, Naoyuki Tsuruta, Rin-ichirou Taniguchi, Makoto Amamiya
Kyushu University
6-1 Kasugakouen,Kasuga,Fukuoka ,Japan

の図の行を３章で述べた通り降ろしていくときの最終的な行の高さの期待値 S_N を求めたい。実際にこのモデルをデータの依存関係に従って自律的に実行したとき、期待値 S_N は１フレーズのステップ数の期待値を表す。

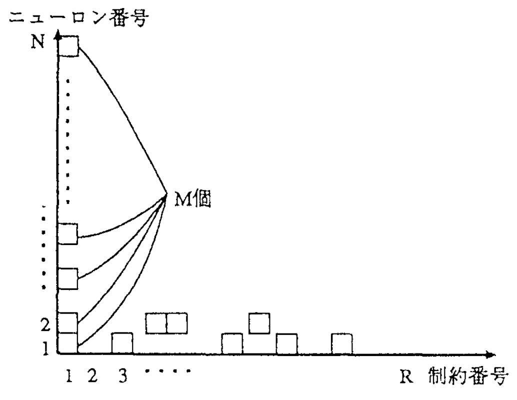

図 3: 並列性を考慮する前のブロック図（ひとつの制約にM個のニューロン）

まず、$N=2$ のとき、すなわち初期段階での行数が２行のときを考える。これらのふたつの行が重なり合って１行になれる確率は、$(1-(\frac{M}{N})^2)^R$ で表せるので、最終的な行数の期待値 S_2 の値は、

$$S_2 = 1 \times (1-(\frac{M}{N})^2)^R + 2 \times (1-(1-(\frac{M}{N})^2)^R) \qquad (1)$$

となる。次に $N=n-1$ のときの期待値が S_{n-1} だとすると、$N=n$ の期待値 S_n は、

$$S_n = S_{n-1} \times A + (S_{n-1}+1) \times (1-A) \qquad (2)$$

$$(ここで A = (1-\frac{M \times \frac{n-1}{S_{n-1}}}{N} \times \frac{M}{N})^R)$$

で表される。そこで N が十分に大きいとき

$$S_N = \sqrt{R} \times M \qquad (3)$$

であると推測する。式 (3) より $S_{n-1} = M \times \frac{n-1}{n} \times \sqrt{R}$ と表せるので $S_{n-1} = \frac{n-1}{n} \times S_n$ となる。これを式 (2) に代入して、$N \times S_n >> R \times M^2$ のもとで一次近似すると、$S_N = \sqrt{R} \times M + C(C:定数)$ となる。計算機シミュレーションにより、$C=0$ である。

実行時間の確率分布を N により正規化したときの期待値を S' すなわち $S' = \frac{S}{N}$、確率を P' すなわち $P'(n) = P(nN)$ とすると、期待値 S' は、

$$\begin{aligned} S' &= \int_0^1 n \times P'(nN)dn \\ &= \alpha \times \sqrt{\int_0^1 n^2 dn} \times \sqrt{\int_0^1 (P'(nN))^2 dn} \\ &\qquad (0 \le \alpha \le 1) \\ &= \alpha \times \frac{1}{\sqrt{3}} \times C \ \ (C : cons) \end{aligned} \qquad (4)$$

で表される。ここで $S' = S/N = \frac{\sqrt{R} \times M}{N}$ より $\lim_{N\to\infty} S' = 0$ となるように M、R を決定してやれば、式 (4) より、

$$\lim_{N\to\infty} \alpha = 0 \qquad (5)$$

よって、$N \to \infty$ において、n と $P'(nN)$ は直交している。n は単調増加関数、かつ $\int_0^1 P'(nN)dn = 1$ より、N が極限まで大きな値をとった場合 $P'(nN)$ は図 (4) のような関数になる。N が極限まで大きくなると n が期待値より大きくなる部分の確率分布の面積は０に近づく。このことからニューロンの総数が十分大きければ、最悪でも実行時間のオーダは $O(M\sqrt{R})$ といえる。このオーダは制約数とそこに含まれるニューロンの数のみによって定義されるため、２次元的なモデルでも多次元的なモデルでも実行時間のオーダは同じといえる。

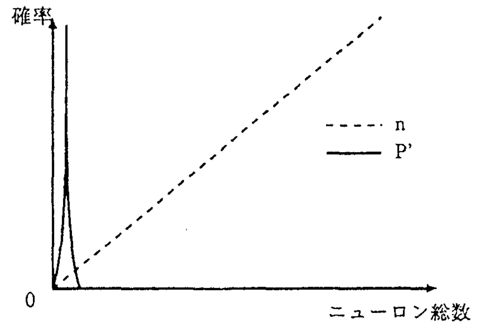

図 4: 確率分布

最終的な高さの期待値 S_N を求めると $S_N = M \times \sqrt{R}$ が得られる。M はひとつの制約に属するニューロンの数、R は制約の数である。

N 個のニューロンを持つボルツマンマシンを逐次方式で実行した場合、１フレーズに N ステップ必要である。しかし、ニューロンの総数 N が大きくなった場合、上記の方式で実行することにより、１フレーズにかかるステップ数のオーダを最悪でも $O(M \times \sqrt{R})$ ステップに抑えることができる。したがって、ボルツマンマシンの実行時間は制約の数およびひとつの制約の中に含まれるニューロンの数によって左右され、ニューロンの総数には関係ないことがわかる。また、もとのモデルの制約が局所的でなく、ひとつの制約に多くのニューロンを含む場合は、モデルを階層化するなどの方法で制約を局所的にし、制約の数が多少増えたとしても制約に含まれるニューロンの数を抑えることによって、実行速度を上げられることがわかる。

5 おわりに

ボルツマンマシンは、おなじ制約に属するニューロンどうしは同時に状態更新を行なうことができない。また、ひとつのニューロンは同時にいくつもの制約に属する。このため、ボルツマンマシンにおけるニューロン並列の研究はほとんど行なわれていなかった。

本論文では、ボルツマンマシンをデータの依存関係に基づいて解析し、データの依存関係に従って各ニューロンを自律的に動作させることにより並列性を自動抽出することを提案した。そして確率モデルを用い、ニューロン数が大きいとき、１フレーズの実行時間のオーダが、最悪でも $O(M\sqrt{R})$ であることを示した。

参考文献

[1] R.Taniguchi,M.Amamiya:“AMP:Autonomous Multi-Processor for Image Processing and Computer Vision”,Proc.10th ICPR,vol.2 pp.497-500,IEEE Computer Society Press(1990.6)

[2] M.Amamiya,R.Taniguchi:“Datarol :A Massively Parallel Archtecture for Functional Langage”,Proc.2th IEEE Symposium on Parallel and Distributed Processing,pp.726-735(1990.12)

[3] 鶴田直之, 赤木琢磨, 谷口倫一郎, 雨宮真人:“ニューロプログラミングにおける超並列実行プログラムの導出方式”, 人工知能学会全国大会第７回論文集,pp.203-206(1993.7)