LA-002 非均質環境を考慮した誤差逆伝播法のノード並列アルゴリズム
Node Parallel BP Algorithm on Heterogeneous Computer Cluster

小澤 洋司 † Yoji Ozawa　**菅谷 至寛** † Yoshihiro Sugaya　**阿曽 弘具** † Hirotomo Aso

1. はじめに

階層型ニューラルネットワークの 1 つである 3 層パーセプトロン (図 1) とその学習法である誤差逆伝播法 (以下 BP) は、良い識別性能を持つ識別器を学習サンプルから比較的容易に構成できる。しかし、学習に要する計算量は多大なものとなるため計算時間の短縮が必要とされている。BP にはデータ並列とノード並列の 2 種類の並列性が内在している [1]。これらを組み合わせ、並列計算機にマッピングすることで高速化できる。

本研究では並列計算機として計算機クラスタを対象とする。各計算機の性能が均質でない場合、最も性能の低い計算機により全体の性能が制限されてしまうので、各計算機の性能に応じて割当てを変化させる必要がある。従来手法 [2] ではデータ並列しか非均質性を考慮していなかったが、本研究ではノード並列についても考慮する。

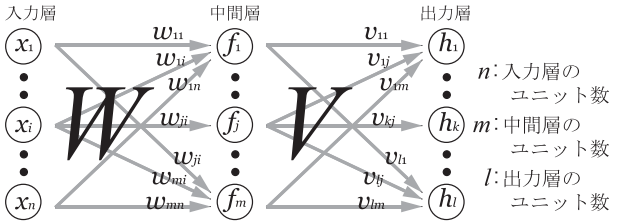

図 1: 3 層パーセプトロン

2. 誤差逆伝播法の並列化

データ並列は、学習サンプル集合 X を分割して、その部分集合を各プロセッサに割り当てる。各プロセッサは割り当てられたサンプルについて重み更新量の部分和を求め、通信により部分和の総和を計算して、配りなおす。

ノード並列は、重み行列を分割し、重み更新量の計算を並列化する [2]。入力層と中間層を結合する重み行列 $\boldsymbol{W}(m \times n$ 行列) を**列方向**に、中間層と出力層を結合する重み行列 $\boldsymbol{V}(l \times m$ 行列) を**行方向**に分割する。つまり、中間層のユニット数 m を分割している。例えば、プロセッサ数が 3 の時は次のようになる。

$$\boldsymbol{W} = \left(\begin{array}{c} \boldsymbol{W}^{(1)} \\ \hline \boldsymbol{W}^{(2)} \\ \hline \boldsymbol{W}^{(3)} \end{array} \right), \quad \boldsymbol{V} = \left(\begin{array}{c|c|c} \boldsymbol{V}^{(1)} & \boldsymbol{V}^{(2)} & \boldsymbol{V}^{(3)} \end{array} \right)$$

このようにすることで、プロセッサ間通信を出力層を求める時にのみ行なうだけで、重み更新量を計算することができる。

これら 2 種類の並列化法は組み合わせることができる。まず、データ並列を行なうデータ並列グループを作成し、そのグループの中でノード並列を行なう。全プロセッサ数 N、データ並列度 N_{DP}、ノード並列度 N_{NP} とすると、$N = N_{DP}N_{NP}$ という関係になる。

†東北大学大学院工学研究科

3. 非均質環境への対応

本研究では学習サンプルと重み行列の各プロセッサの性能に応じた割当て手法を提案する。

3.1 並列グループの作成

プロセッサを能力により昇順にソートする。これを順に N_{NP} 個ずつに分け、データ並列グループを作る。各データ並列グループで重み行列の分割の割合が一定になるようにノード並列グループを作る (図 2)。i 番目のデータ並列グループ、j 番目のノード並列グループに属するプロセッサを P_{ij} $(i = 1, \ldots, N_{DP}, j = 1, \ldots, N_{NP})$ とする。プロセッサ P_{ij} の能力を $a_j^{(i)}$ とすると、$a_1^{(1)} \leq a_2^{(1)} \leq \cdots \leq a_{N_{NP}}^{(1)} \leq a_1^{(2)} \leq \cdots \leq a_{N_{NP}}^{(N_{DP})}$ である。

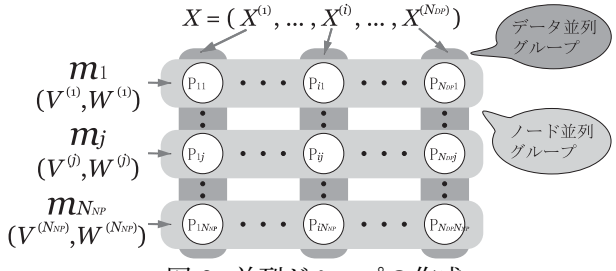

図 2: 並列グループの作成

3.2 学習サンプルの割当て (データ並列)

各データ並列グループに対し、グループ内の最低の性能のプロセッサの能力に応じて割り当てる。つまり、i 番目のデータ並列グループに $s^{(i)} = \left(a_1^{(i)} / \sum_{i=1}^{N_{DP}} a_1^{(i)} \right) s$ 個の学習サンプルを割り当てる (s は総サンプル数)。

3.3 重み行列の割当て (ノード並列)

非均質性を考慮して学習サンプルを割り当てても、データ並列グループ内のプロセッサの性能の非均質性により、各プロセッサの能力を十分に使うことが出来ない。そこで、重み行列も非均質性を考慮して割り当てる。

j 番目のノード並列グループに分割された重み行列 $\boldsymbol{W}^{(j)}(m_j \times n$ 行列)、$\boldsymbol{V}^{(j)}(l \times m_j$ 行列) を割り当てる。従って、m_j を決定することで割当てが決まる。全体の処理時間を最短とするため、m_j を全てのプロセッサの処理時間が等しくなるように決める。そこで、処理時間をモデル化し、そのモデルを用いて m_j を決定する。処理時間は計算時間と通信時間からなる。

加減算、乗除算、シグモイド演算 1 回の計算時間を各々 α, β, γ とすると、能力が $a_j^{(i)}$ のプロセッサの各演算の計算時間はそれぞれ $\alpha/a_j^{(i)}, \beta/a_j^{(i)}, \gamma/a_j^{(i)}$ となる。従って、中間層の計算、出力層の計算の一部にかかる時間は

$$M_j^{(i)} = \frac{m_j s^{(i)}}{a_j^{(i)}} \{(\alpha + \beta)(n + l) + \gamma\}$$

となる。出力層の計算の残り、修正誤差、重み更新量の計算時間は

$$N_j^{(i)} = \frac{s^{(i)}}{a_j^{(i)}} \{(\alpha + \beta)(2l + m_j(1 + 2l + n)) + \gamma l\}$$

となる。

プロセッサ間通信はオールリデュース操作となっている。従って、プロセッサ数を p、ベクトルの長さを d、ネットワークのスループットに関する項、ベクトルの長さに関わらず発生する遅延の項、通信が増加したことに関する項の係数を各々$\alpha_c, \beta_c, \gamma_c$ とし、式 (1) のようにモデル化できる。

$$T_{allreduce}(d) = kd + h$$
$$k = \begin{cases} \alpha_c(p-1) + \gamma_c(p-1) & (p \leq 4) \\ \alpha_c\lceil \log_2 p \rceil + \gamma_c(p-1) & (p > 4) \end{cases} \quad (1)$$
$$h = \begin{cases} \beta_c(p-1) & (p \leq 4) \\ \beta_c\lceil \log_2 p \rceil & (p > 4) \end{cases}$$

通信はデータ並列における通信 $t_{DP\ j} = k_{DP} m_j (n + l) + 2h_{DP}$ とノード並列における通信 $t_{NP}^{(i)} = k_{NP} l s^{(i)} + h_{NP}$ が行なわれる。ここで、k_{DP}, h_{DP} は $p = N_{NP}$ の場合の k, h であり、k_{NP}, h_{NP} は $p = N_{DP}$ の場合である。

出力層を求めるための通信 (ノード並列) を行なう時、データ並列グループで同期をとる必要がある。つまり、データ並列グループ内のそれ以前の処理時間の最大値になる。

BP アルゴリズムの流れは次のようになっている。

(i) 中間層、出力層の計算の一部

(ii) 出力層を求めるための通信 (ノード並列)

(iii) 出力層の残りの計算、修正誤差、重み更新量の計算

(iv) 重み修正量を求める通信 (データ並列)

従って処理時間 $t_j^{(i)}$ を式 (2) のようにモデル化できる。

$$t_j^{(i)} = \max_i \left(\max_j M_j^{(i)} + t_{NP}^{(i)} + N_j^{(i)} \right) + t_{DP\ j} \quad (2)$$

この処理時間モデルを用いて、重み行列の割当てを決定する。ノード並列グループ内のプロセッサの処理時間は等しく、$t_j^{(i)} = t_j$ となるようにする。全体の処理時間を最短にするため、全てのプロセッサの処理時間を等しくする。つまり、$t_1 = t_j$ が成り立てばよい。式 (2) を用いて、$t_1 = t_j$ を m_j について変形する。

$$m_j = B_{1\ j} + B_{2\ j} m_1 \quad (3)$$

$$\begin{cases} B_{1\ j} = \dfrac{sl\left(1 - \max_i (a_0^{(i)}/a_j^{(i)})\right)(2(\alpha+\beta)+\gamma)}{k_{DP}(n+l) + (1+2l+n)\max_i \left(s^{(i)}/a_j^{(i)}\right) \sum_i a_0^{(i)}} \\ B_{2\ j} = \dfrac{k_{DP}(n+l)\sum_i a_0^{(i)} + (1+2l+n)s}{k_{DP}(n+l) + (1+2l+n)\max_i \left(s^{(i)}/a_j^{(i)}\right) \sum_i a_0^{(i)}} \end{cases}$$

ここで $B_{1\ j}, B_{2\ j}$ は各プロセッサによって決まる定数である。さらに、$\sum_j m_j = m$ なので、式 (3) の両辺を j について総和をとり、m_1 について解く。

$$m_1 = \frac{\left(m - \sum_j B_{1\ j}\right)}{\sum_j B_{2\ j}} \quad (4)$$

となり、m_1 を求めることができる。この m_1 を式 (3) に代入することにより、m_j を求めることができ、重み行列の分割の割合を決定することができる。

3.4 性能評価

提案手法の非均質環境における性能を実験によって評価した。使用した計算機クラスタは、UltraSparc Ⅱ 300MHz の計算機をギガビットイーサネットスイッチで結合したものである。いくつかの計算機にジョブを投入して見かけの能力を下げ、非均質環境とした。また、ニューラルネットワークのサイズは 203-100-26 とした。表 1 の実験条件で、提案手法、従来手法 (データ並列にだけ非均質性を考慮) [2]、学習サンプルと重み行列を等分し割り当てる手法 (非均質性未考慮) について処理時間を測定した。

表 1: 実験条件

条件	学習サンプル数 s	プロセッサの能力比:台数
a	100	0.5 : 1 台 0.67 : 1 台 1.0 : 6 台
b	100	0.5 : 1 台 1.0 : 7 台
c	200	0.5 : 1 台 0.67 : 1 台 1.0 : 6 台

プロセッサ数 $N = 8$ の時の実験結果を表 2 に示す。様々な学習サンプル数やプロセッサの能力において、提案手法が最高の性能を出している。従来手法で非均質性を考慮せずにノード並列を行う場合、適切なデータ並列が必要となり、8 台並列では表 2 の値より時間がかかる。しかし、ノード並列に非均質性を考慮した時、結果的にデータ並列がない時最高性能を得ている。

表 2: 1 回のイタレーションに要する時間 [sec]

条件	提案手法	従来手法	非均質性未考慮
a	0.208 (8,1)	0.236 (2,4)	0.273 (8,1)
b	0.221 (8,1)	0.229 (4,2)	0.226 (8,1)
c	0.375 (8,1)	0.403 (2,4)	0.499 (8,1)

括弧は最適な並列度の組合せ (N_{NP}, N_{DP})

3.5 並列度の自動選択

一般的には並列度の組合せにより処理時間が異なるため、最適な組合せを自動的に選択する必要がある。処理時間モデル (式 (2)) を用いて、各並列度の組合せの処理時間を計算し、最適な組合せを選択する。

表 1 の実験条件で自動選択を行ったところ、全ての条件で表 2 に示す最適な組合せが選択された。

4. まとめ

非均質な並列計算機環境を考慮したタスク割当て手法、および並列度選択手法を提案した。従来手法 [2] では考慮されていなかったノード並列に関しても非均質性を考慮し、従来手法よりも高い性能を発揮することができた。また、最適な並列度の組合せを選択する手法を提案し、有効性を確認した。

非均質環境を考慮したタスクの割当てには通信時間など多くの値を事前に測定する必要がある。もっと少ない情報で、タスクの割当て、さらに最適な並列度の組合せを選択できる手法の開発が今後の課題である。

参考文献

[1] A.Singer : Implementation of Artificial Neural Network on the Connection Machine, *Parallel Computing*, Vol.14, pp.305-315, 1990

[2] 東大亮, 菅谷至寛, 阿曽弘具 : 非均一な並列計算機環境におけるニューラルネットワークの学習, 情処研報 HPC-89-1, pp.1-6, 2002