

大脳皮質モデルBESOMの クラスタ分散化とGPGPU並列化

中田秀基^{1,2}、黎明曦^{2,1}
井上辰彦^{1,3}、一杉裕志¹

1 産業技術総合研究所人工知能研究センター

2 筑波大学

3 株式会社創夢

この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務の結果得られたものです。

研究の背景

- 大脳皮質モデルBESOM
- ベイジアンネットによる大脳皮質のモデル化 → 計算量大

研究の目的

- 並列化による高速化
- 大量データの学習には大幅な高速化が必須

得られた成果

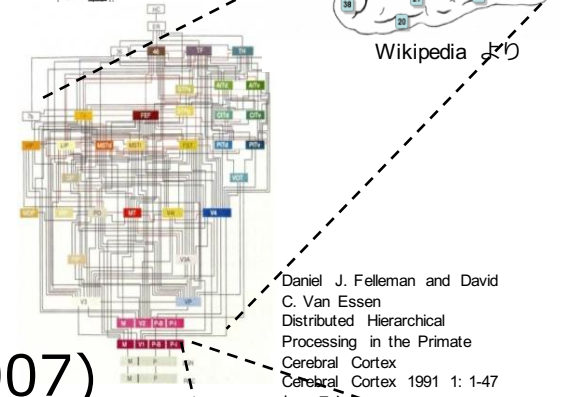
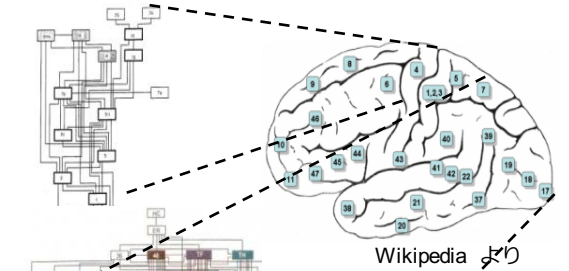
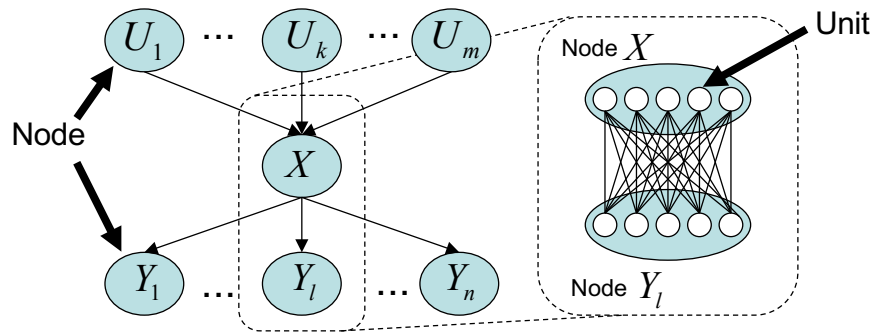
- クラスタ分散化によっておよそ40倍の高速化
- GPGPU並列化によっておよそ45倍の高速化

今後の課題

- 学習率の調整
- ミニバッチサイズの調整

大脳皮質モデルBESOM

- ベイジアンネットによるマクロカラム間ネットワークのモデル化



- 近似確率伝播アルゴリズムの改良OOBP(ichisugi 2007)

$$l_{XY}^{t+1} = z_Y^t + W_{XY} o_Y^t$$

$$o_X^{t+1} = \prod_{Y \in \text{children}(X)} l_{XY}^{t+1}$$

$$k_{UX}^{t+1} = W_{UX}^T b_U^t$$

$$p_X^{t+1} = \sum_{U \in \text{parents}(X)} k_{UX}^{t+1}$$

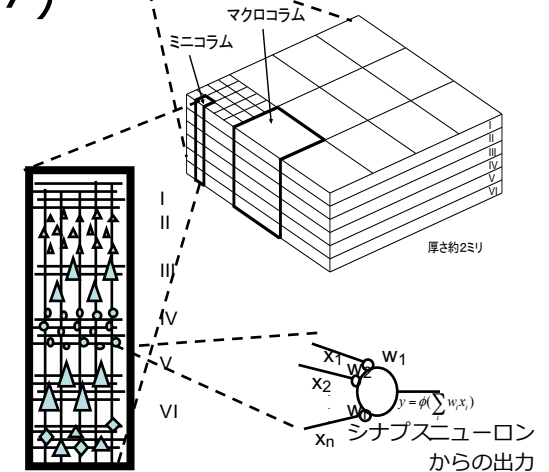
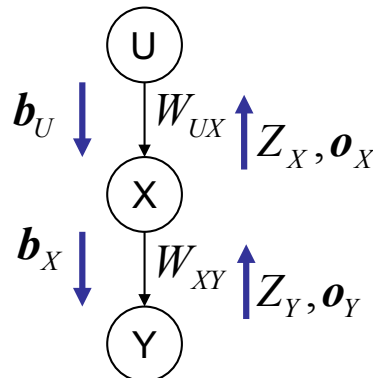
$$r_X^{t+1} = o_X^{t+1} \otimes p_X^{t+1}$$

$$Z_X^{t+1} = \sum_i (r_X^{t+1})_i \quad (= \|r_X^{t+1}\|_1 = o_X^{t+1} \cdot p_X^{t+1})$$

$$z_X^{t+1} = (Z_X^{t+1}, Z_X^{t+1}, \dots, Z_X^{t+1})^T$$

$$b_X^{t+1} = (1/Z_X^{t+1}) r_X^{t+1}$$


ただし、 $x \otimes y = (x_1 y_1, x_2 y_2, \dots, x_n y_n)^T$



BESOM学習の並列化

- モデル間並列化

- 複数のモデルを用いて並列に学習を実行し、結果となるモデルパラメータを更新



クラスタ
分散化

- モデル内並列化

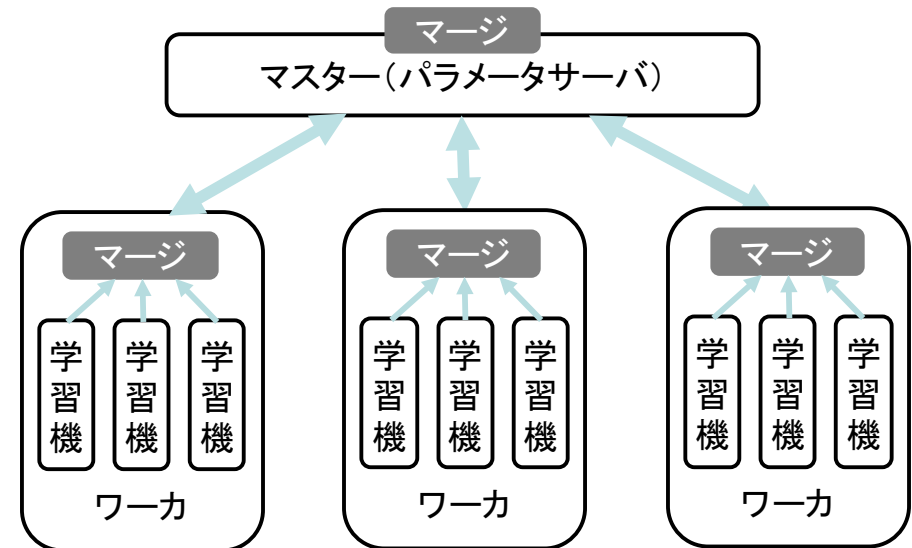
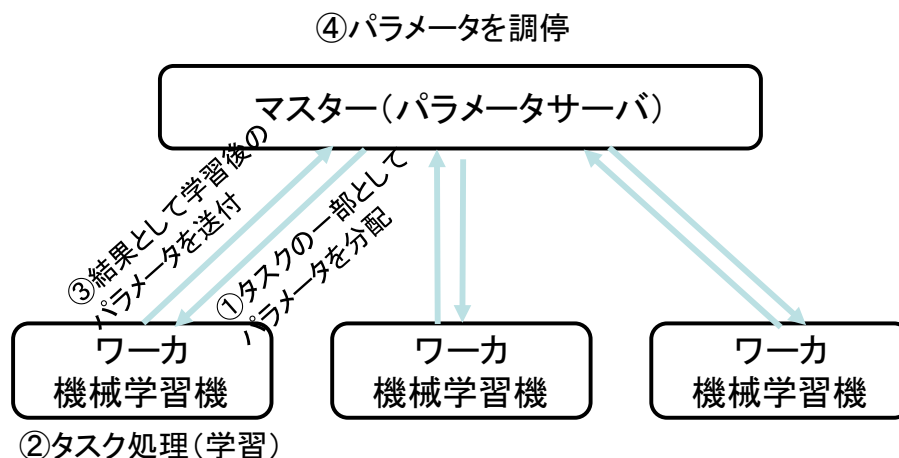
- OOBPの各ノードの更新を並列実行
 - ただし負荷は著しく不均衡なため単純な並列化には適さない
 - 単純なマルチスレッド並列化では2倍弱の高速化
- ミニバッチ化による並列化
 - オリジナルはオンライン学習
 - 同一のモデルパラメータを用いて複数の学習データに対する評価を並列に実行
 - 複数の学習データ評価結果を用いてモデルパラメータを更新



GPGPU
並列化

クラスタによる分散並列実行

- 複数の機械学習機がそれぞれ独立して学習し、モデルを更新
- 定期的に学習結果のモデルをパラメータサーバに送信、モデルをマージ
- マージした結果を各学習機に配布、学習を続行
- 複数スレッドでの実行も併用可能
 - 個々のスレッドがそれぞれ異なる機械学習機を実行
 - モデルのマージも階層的に



クラスタ分散化の結果

- MNIST 手書き数字認識

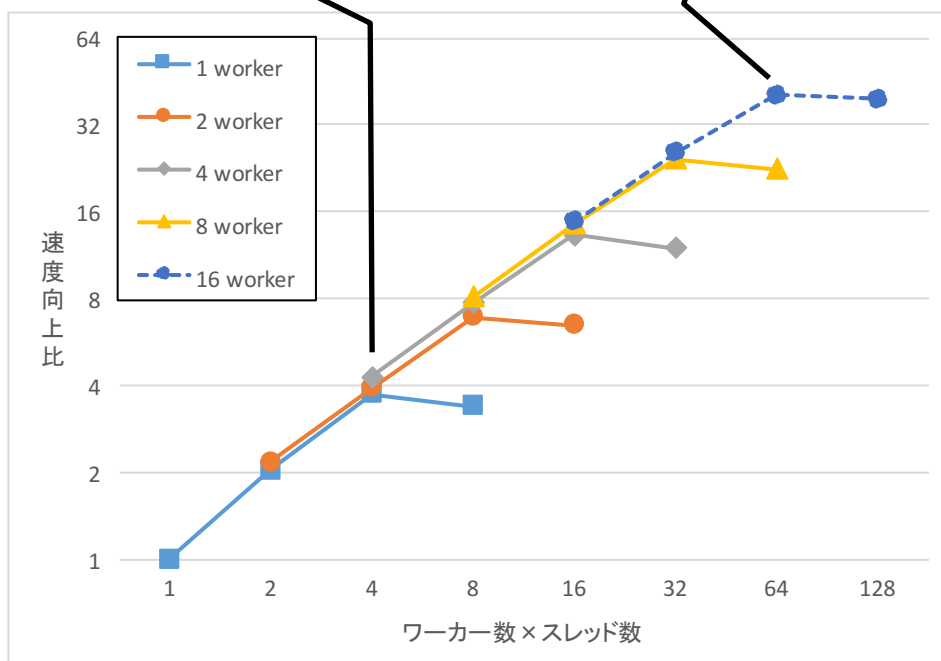
	入力層	L1	L2	出力層
ノード数	784	81	9	1
ユニット総数	1568	1620	900	11
パラメータ数	-	51840	162000	9900

- 16ノードクラスタで実行

- Intel Xeon W5590 (3.33GHz, 4コア) x 2

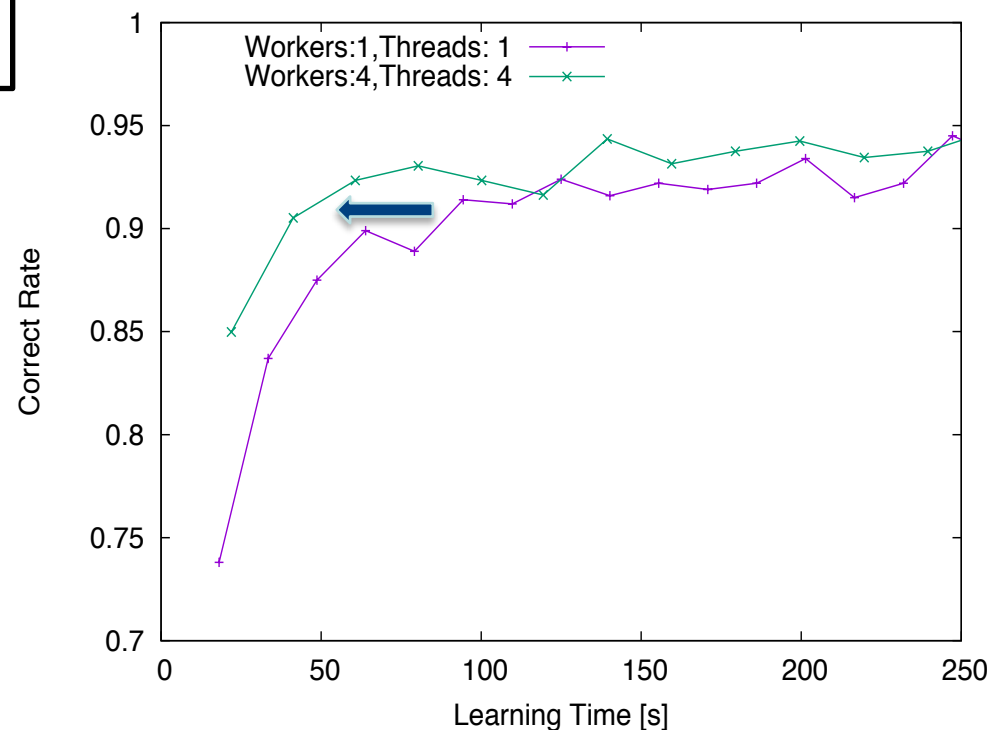
ワーカ数固定の場合、スレッド数4で最速

最大40倍の速度向上



- 正答率と学習時間をプロット

- ワーカ数・スレッド数増大の効果有りより速く正答率が向上
- 速度向上程の効果は見られず
- 学習率の調整ができていないためと思われる

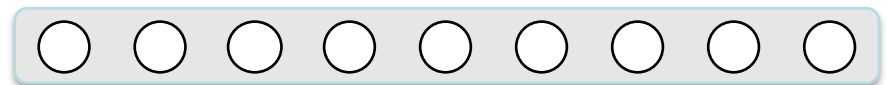
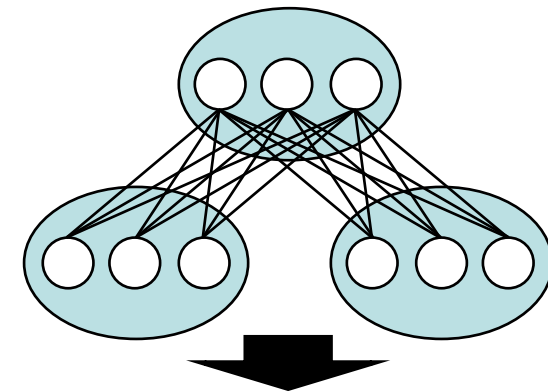


GPGPU並列化

- 近似確率伝播のノード情報更新の並列化
 - ノード単位では負荷が不均衡
 - 負荷が上下のノードの数に依存するため
 - ノードをオブジェクトとし相互に参照するデータ構造はGPGPU化に不適
 - GPGPUは別のメモリ空間 → 参照構造を引き写すにはポインタの変換が必要
 - 不均質な計算 → GPGPUコードの並列化効率の低下

➔ ユニットを基盤としたフラットなデータ構造に変更

- 負荷の均等化
- 計算の均質化



ミニバッチ化

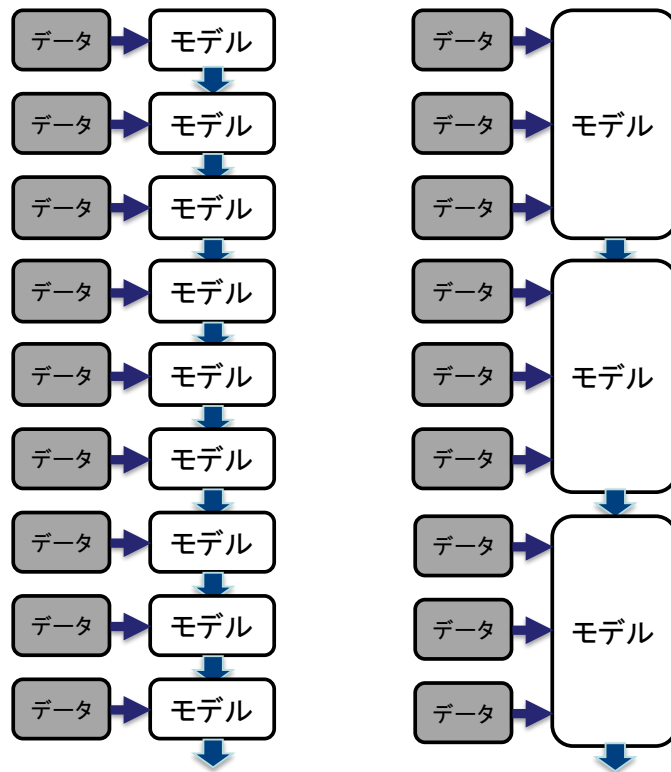
GPGPU並列化の結果

- 複数のデータを一つのモデルに適用可能
→ 並列化が可能

- MNISTの128データを認識
 - CPUは逐次
 - GPGPUは128データを同時に処理

環境

- CPU : Intel Xeon W5590 (3.3GHz)
- GPGPU : GTX980



オンライン学習

ミニバッチ学習

