

ニューラルネットワークモデルにおけるマルチスレッド効果の検証

Validation of multi-thread effect in neural network model

発表者：滝澤 亮 指導教員：坪井 一洋

1. はじめに

人間の脳には大脳だけで 100 億以上の膨大な数のニューロンが存在する。各ニューロンは、多数の他のニューロンから信号を受け取り、また、他の多数のニューロンへ信号を送る。脳は、この信号の流れによって、様々な情報処理を行う。

ニューラルネットワークは、脳の情報処理をモデルにつくられた情報処理機構である[1]。現在、ニューラルネットワークは脳の一部の機能を実現しているに過ぎず、音声認識や文字認識といった問題を人間と同程度の精度で解くまでには至っていない[2]。

本研究では、脳の並列分散処理に注目する。人間の脳はニューロンが自律的に処理を行う並列分散処理である。脳の並列分散処理を再現することで、逐次処理では再現できない脳の機能を再現できる可能性がある。

実際、Java 言語のマルチスレッド機能を用いることでニューロンの並列分散性を再現する試みもある。マルチスレッドとはスレッドと呼ばれる一連の処理を並行して実行する機能である。そして、その結果から、マルチスレッドを用いたモデルの方が逐次処理モデルよりも効果的であるとの指摘もある[3]。

本研究では、ニューラルネットワークでのマルチスレッド機能による疑似並列分散処理の有用性をより詳しく検証する。ニューロン数の増加による効果として発火回数の増加や発火時間の短縮を再現すると同時にその理由についても明らかにする。

2. ニューラルネットワークモデル

2.1 ニューロンモデル

ニューラルネットワークは複数のニューロンが結合することによって構成されている。そのため、ニューロンをネットワークの構成要素とみなすことができる。

ニューロンが持つ機能は、他のニューロンからの入力電流の受け取り、膜電位の更新、発火、イオンチャンネルゲートの開放、他のニューロンへの出力などが挙げられる。これらの機能のモデルには、細胞膜の電気的性質による膜コンダクタンスをベースにしたモデルを用いる[3]。

今回用いるニューロンの設定パラメータを表 1 にまとめる。各ニューロンはこの設定パラメータで生成される。ここで u_{rest} は静止電位、 g は膜コンダクタンス、 C は膜容量、 T_{max} は発火時のシナプス間物質濃度、 α はイオンチャンネルの解放速さ、 β はイオンチャンネルの閉まる速さである。

表 1 ニューロンの設定パラメータ

u_{rest} [V]	g [A/V]	C [F]	T_{max} [mol]	α [1/s]	β [1/s]
-65×10^{-3}	25×10^{-9}	0.5×10^{-9}	0.001	1.1×10^6	190

2.2 ネットワークモデル

本研究では、階層型ニューラルネットワークを想定する。各ニューロンは 0 から番号順に設定され、各ニューロンの結合は 1 入力 1 出力とし、設定された番号順に結合する。また、先頭のニューロンが入力層、それ以外のニューロンを出力層に分け、入力層である先頭のニューロンに外部刺激として入力電流を送り続け、最後尾のニューロンからの出力をネットワークの外に出す開ループ形式である。

逐次処理モデルでは番号順に更新を行い、ネットワーク全体における情報が上流から下流に流れるように、全体で一定の方向性を持つ。それに対して、マルチスレッドモデルでは、マルチスレッドの並行処理によって、ネットワークにおける情報の流れを不規則かつ局所的に発生させる。今回用いるマルチスレッドモデルの詳細を次節で示す。

2.3 マルチスレッドモデル

本研究のマルチスレッドモデルでは、一つのスレッドに一つのニューロンの処理を割り当てる。各スレッドは一度起動すれば自動的に処理を行うため、ニューロンに自律的な処理を行わせることが可能である。このように、マルチスレッドを使うことによってニューロンの自律性を高め、疑似的な並列処理を実装することができる。

また、マルチスレッドでは各スレッドの処理を独立させて行うことができるが、その独立性ゆえに、他のスレッドと直接情報のやり取りを行うことが難しい。そこで、ニューロン間の結合をサポートするオブジェクトとして「結合オブジェクト」を定義し、間接的にニューロンを結合させている。この結合オブジェクトによって、各ニューロンは自身の更新から入出力処理までを自律的に実行することができる。

マルチスレッドモデルでは、結果の出力を 10^4 回行った時点で終了となる。ニューロンの自律性を損なわないため、結果の出力とニューロンの更新は同期していない。

3. シミュレーション

3.1 条件

ニューロンの更新処理は 0.1 ms を 1 ステップとし 1 s 間の信号伝達を想定したシミュレーションを行う。また、実行結果はマルチスレッドモデルと逐次処理モデルを各 10 回実行した結果の平均で比較を行う。ネットワーク内のニューロン数を 15, 25, 35, 45 と増加させた場合での変化の様子を比較するが、ニューロン数 n のネットワークを N_n と略記する。なお、すべてのニューロンの結果の比較は膨大なデータ数となるので、入力層から 5 個ごとのニューロンの出力を計測している。

3.2 実行結果

ニューロンの発火について、その回数と最初の発火が起き

るまでの時間に注目する。以下の実行結果はマルチスレッドモデルと逐次処理モデルの結果の差を示している。また、ニューロン数の異なるネットワークごとの全体での傾向をとらえるため、計測したニューロンの結果すべてで平均をとり、ネットワーク全体の傾向とみなした。

実行結果を図1と2に示す。結合するニューロン数の増加とともに全体的に発火回数は増加し、最初の発火までの時間は早くなる傾向がみられる。これらの結果は、ニューロン数の増加に比例して、逐次処理モデルよりもマルチスレッドモデルの方が情報伝達に有用であることを示唆している。

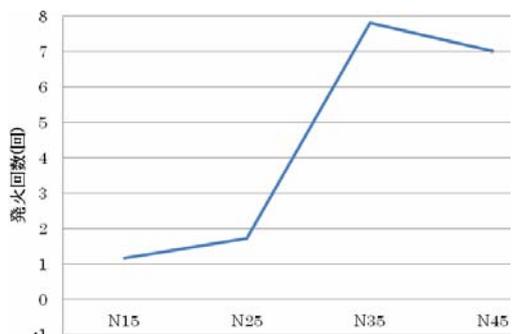


図1 ニューロン数の増加に伴う発火回数の変化

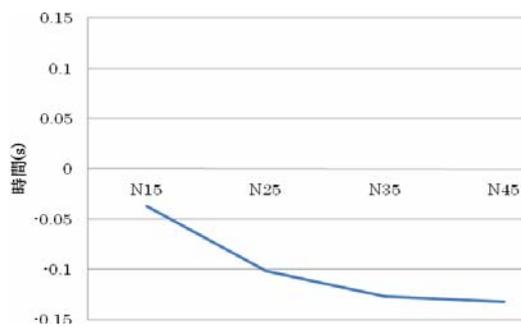


図2 ニューロン数の増加に伴う発火時間の変化

次に、マルチスレッドの効果を検証するためにマルチスレッドモデルの各ニューロンの経過時間に注目した。ニューロン数の異なるネットワークごとに10回のシミュレーションを行い、平均経過時間を求めた。平均経過時間は計測したニューロンの経過時間から平均を求め、ネットワーク全体の経過時間とする。ネットワーク全体の経過時間を、さらに平均し、各ネットワークの平均経過時間とした。その結果を表2に示す。

マルチスレッドモデルでは、ニューロン数が増加するほどにニューロンの経過時間が増加している。ニューロンの経過時間は、ニューロンの更新ごとに増加するようになっている。マルチスレッドモデルでは、ニューロンの自律性を生かすために結果の出力とニューロンの更新は完全に同期していない。そのため、ネットワーク全体としての結果の出力時間と各ニューロンとの時間にずれが生じている。

表2 マルチスレッドモデルでの平均経過時間

N15	1.259711
N25	2.624094
N35	2.67146
N45	3.12954

ニューロンの経過時間を同じ程度にしてマルチスレッドモデルと逐次処理モデルでの比較を行った。その結果を図3と4に示す。これまでの比較結果を破線、経過時間を考慮した結果を実線で表す。これより、マルチスレッドモデルの発火回数、発火時間ともに逐次処理モデルを下回る結果になった。

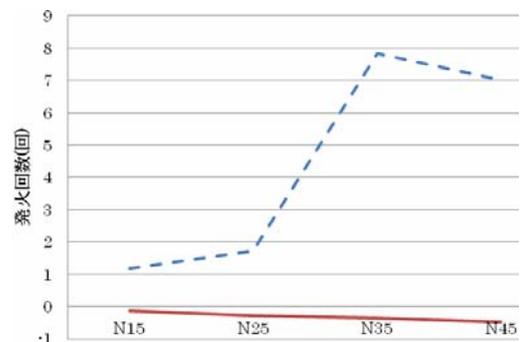


図3 経過時間を考慮した発火回数比較

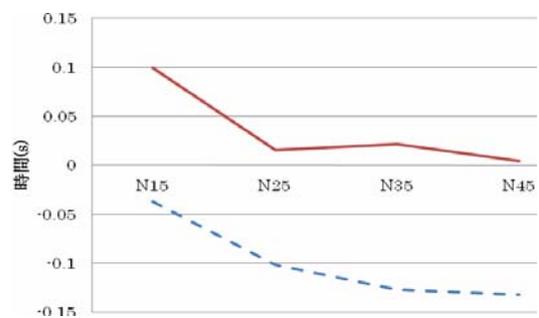


図4 経過時間を考慮した発火時間比較

4. まとめと今後の課題

脳の並列分散性を疑似的に再現したマルチスレッドモデルによるニューラルネットワークの有用性について検証した。

その結果、マルチスレッドモデルでは各ニューロンの自律性を優先して、ネットワーク全体での同期をとらないため、ニューロンごとに異なる時間経過となっていることが分かった。

この点を踏まえて、逐次処理モデルでも同じ経過時間でシミュレーションしたところマルチスレッドモデルが逐次処理モデルを上回る結果にはならなかった。

今後の課題としては、信号伝達に双方向性を持つ相互結合型ネットワークなどより複雑な構造に対してマルチスレッドモデルの適用を考えたい。

参考文献

- [1]Akira Iwata, Toshiyuki Matubara:『ニューラルネットワーク入門』,(名古屋工業大学岩田彰研究室, 1996)
<http://www-ailab.elcom.nitech.ac.jp/lecture/neuro/menu.html>
- [2]石立 喬:『ニューラルネットワークを用いたパターン認識』,(翔泳社, 2006/05/01)
<http://codezine.jp/article/detail/372>
- [3]安島 卓臣:『ニューラルネットワークのオブジェクト指向モデリング』,(茨城大学知能システム工学科平成21年度卒業研究論文, 2010)