

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

Object-oriented modeling of neural network

発表者：安島 卓臣

指導教員：坪井 一洋

1. はじめに

人間の脳内にはニューロンが無数に存在しており、相互結合することで情報処理・情報伝達を行っている。このような脳の仕組みを模倣した情報処理機構をニューラルネットワークという。ニューロンはそれぞれに電位を持ち、発火することによって、結合しているニューロンの電位を変化させる[1]。

ニューラルネットワークの情報処理形態は分散並列型の特徴を持っており、各ニューロンは自身の状態の更新や結合するニューロンとの入・出力処理を自律的に行っている[2]。このような分散並列処理のモデル化は、従来の機能指向方法論のようなトップダウン形式の処理形態では困難である。

一方、オブジェクト指向方法論はオブジェクトを処理単位としたボトムアップ形式の処理が可能である。さらに、オブジェクト指向プログラミング言語である Java 言語には、1台のコンピュータで並列処理を疑似的に行うマルチスレッドという並行処理機能がある。

そこで、オブジェクト指向方法論を用いることによって、ニューラルネットワークのモデルをより自然な形で設計することができると思われる。

本研究では、オブジェクト指向モデリングに基づいてニューラルネットワークを分析・設計し、モデル化することを目的とする。また、設計したモデルを用いてシミュレーションを行い、機能指向型のモデルと比較し、モデルの有用性・正確性を検証する。

2. オブジェクトモデルの分析と設計

2.1 ニューロンクラス

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、ニューロンをシステム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として、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[3]を用いる。

なお、本研究では、ルンゲ・クッタ法を用いずに、微分方程式の解析解を用いて計算を行う。

2.2 ネットワークのモデル化

ニューラルネットワークの情報処理は、各ニューロンが独立して処理を行う並列分散型の特徴を持っている。この処理形態を踏まえ、本研究では Java 言語のマルチスレッド機能による実装を前提とした設計を行った。

マルチスレッドを用いることでニューロンの独立性を高めることができるが、スレッド(ニューロン)同士を直接結合させることは難しい。そこで、各ニューロンが自由にアクセスできる結合オブジェクトを用意し、間接的にニューロンを結合させる。

各ニューロンは、発火するとその出力を結合オブジェクトに送り、結合オブジェクトはそれをデータとして格納する。そして、結合先のニューロンは必要ときに格納してあるデータを参照することで入力を得る。

このようなニューロン間の結合を考慮して設計したネットワーク全体のクラス図を図 1 に示す。

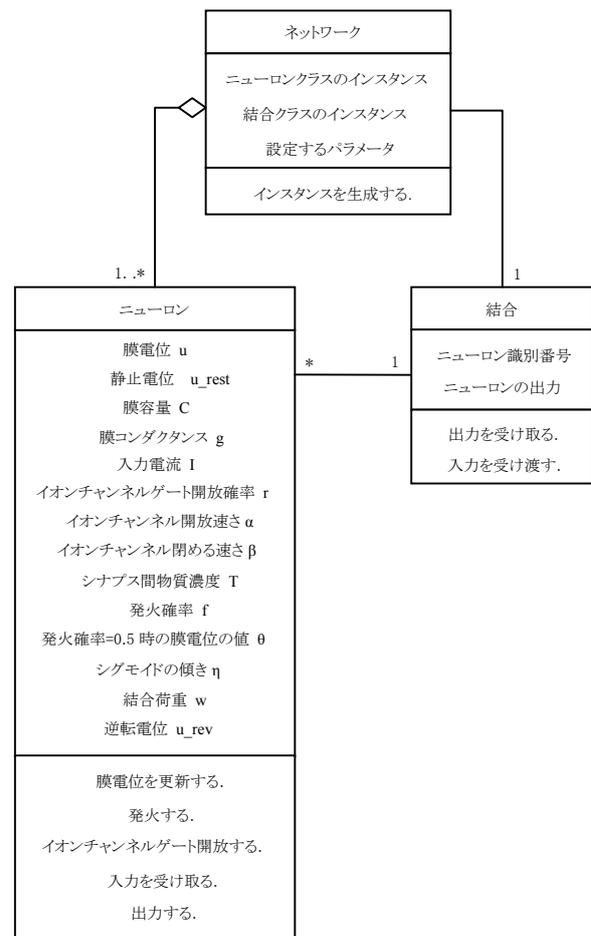


図 1 ニューラルネットワークのクラス図

3. シミュレーション

3.1 例題設定

16 個のニューロンを使ったニューラルネットワークのシミュレーションを行う。マルチスレッドを使用しない機能指向型モデルと今回設計したマルチスレッドを使用したモデルの 2 種類のモデル(図 2)を用意し、Java 言語によるプログラムの実行結果を比較する。

ネットワークは図 3 のような入力層、出力層に分かれた階層型ニューラルネットワークを想定し、入力層には外部刺激として $I = 1625\text{pA}$ の電流を流し続ける。また、各ニューロンの結合は 1 入力-1 出力とする。なお、ニューロン間の情報伝達を観

察するため、すべての結合に対して結合荷重 1, 逆転電位 0mV とする. 他に設定したニューロンのパラメータを表 1 に示す.

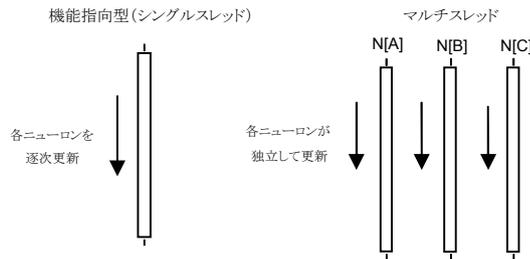


図 2 使用する 2 種類のモデル

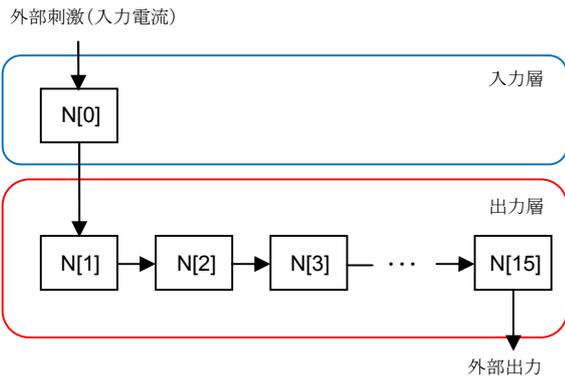


図 3 設定したニューラルネットワーク

表 1 設定したパラメータ

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

3.2 実行結果

1 ステップあたりの時間間隔を 0.1ms として実行時間 1.0s までニューロンの更新を行った. 実行結果として 6, 11, 16 番目のニューロン (N[5], N[10], N[15]) における膜電位の時間変化を図 4~6 に, N[15] における発火付近の拡大図を図 7 に示す.

実行結果から, 入力層に近い N[5] では機能指向型モデルの方が多く発火し, 情報の伝達が速いことが確認できる. N[10] では双方のモデルともほぼ同一の発火数, 変化パターンが確認できる. 最も入力層から離れている N[15] ではマルチスレッドを用いたモデルの発火数が多い. さらに, 図 7 から情報伝達の速度が僅かではあるが速いことが確認できる.

このことから, マルチスレッドの並行処理を用いることによって, 情報処理, 情報伝達の速度が入力層との距離に影響されにくくなると考えられる.

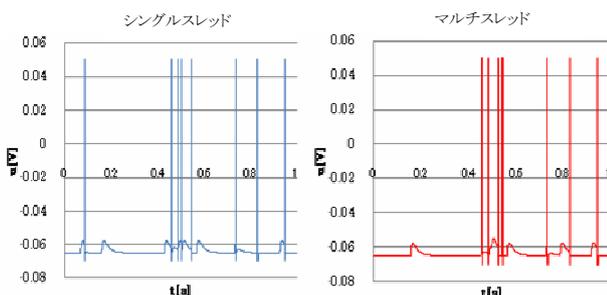


図 4 N[5]の膜電位の変化

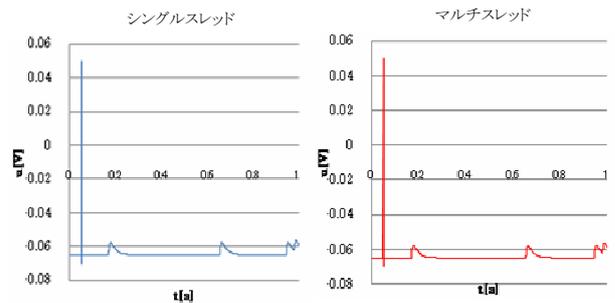


図 5 N[10]の膜電位の変化

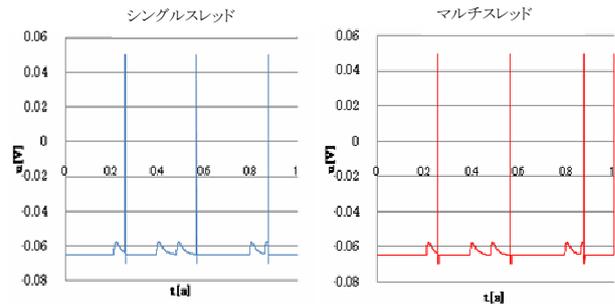


図 6 N[15]の膜電位の変化

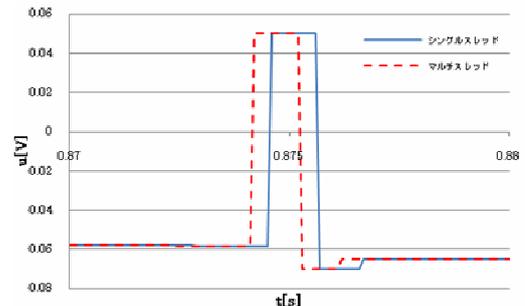


図 7 N[15]の 0.87~0.88 付近の拡大図

4. まとめと今後の課題

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のために, まずネットワーク全体の構造と処理形態について調べ, その構成要素であるニューロンの設計を行った. その後, ニューラルネットワークの情報処理形態を踏まえ, ネットワーク全体の設計を行った. そして, 設計したニューラルネットワークのモデルを基に Java 言語によるプログラムを実装し, シミュレーションによって機能指向型モデルとの比較を行った. その結果, オブジェクト指向型モデルの特徴と利点を発見することができた.

今後の課題は, ニューロンの数を増やし, より複雑なニューラルネットワークのシミュレーションを行い, 機能指向型モデルとの比較を引き続き調査することが挙げられる.

参考文献

- [1] 白井 支朗 編:『基礎と実践ニューラルネットワーク』, (コロナ社, 1995)
- [2] 麻生 英樹:『ニューラルネットワーク情報処理』, (産業図書, 1994)
- [3] 高橋 英雅:『感覚刺激情報の認識における大籠細胞と小籠細胞の役割』, (茨城大学工学部システム工学科平成 18 年度卒業論文, 2005)