

スパイクニューラルネットワーク解析環境のための GUI 編集機能構築
Development of a GUI-Editable Spiking Neural Network Analysis Environment

○早崎寛人¹, 佐々木芳樹²

*Hiroto Hayasaki¹, *Yoshiki Sasaki²

Abstract: In recent years, spiking neural networks (SNN) have gained attention as brain-inspired models for efficient computation. Conventional analysis methods depend on text-based netlists, which demand expertise and lack intuitive usability. To address this, we developed a GUI-based editing environment on Windows using C#, enabling users to place and connect neurons interactively, with automatic net naming and JSON-based save/load support. This approach achieved intuitive model construction and management compared with traditional netlist-based workflows.

1. まえがき

生体脳は25W程度の少ない消費電力でリアルタイムかつ高度な情報処理を、スパイクニューラルネットワークにより実現している。その優れたスパイクニューラルネットワーク (SNN) をアナログ回路により構築することで、低消費電力な動作が期待できる[1]。しかし、アナログ回路で構築した SNN が有効に動作可能であるかを解析する場合、spiceなどを用いた回路シミュレータでは、アナログ回路としての要素を対象として計算を行うことから非線形解析が必要となる、そのため、SNNの動作解析が複雑なものとしてしまうため、SNNとしての解析には回路としての要素を取り除いた状態が望ましい。

先に我々は、SNN解析ソフトウェアとしてテキストベースのネットリストによりネットワーク構造を記述し、シミュレーションエンジンに読み込ませる方法を提案[1]しているが、この手法は柔軟かつ再現性が高い一方で、専門的な記述知識を必要とし、直感的な操作性に欠けるという課題があった。

本稿では、既存の SNN シミュレーションエンジンをそのまま利用しつつ、ネットリスト編集をGUI化することで、直感的かつ視覚的にモデル構築できる解析環境の構築を検討したので報告する。

2. 本論

GUI上でSNNを編集・構築するためには、以下のモデル・機能が必要となる。

① シミュレーション対象モデル

I. ニューロンモデル

入出力ポート：入力1, 出力1

設定可能パラメータ：抵抗, 容量, しきい値, 不応期, 発火パルスの立上/立下/継続時間

II. 外部信号源モデル

入出力ポート：出力1

設定可能パラメータ：初期電流値, 更新電流値

その他：ニューロンへの外部刺激入力を担当

III. ギャップジャンクションモデル

入出力ポート：双方向ポート2

設定可能パラメータ：抵抗

IV. シナプス+遅延モデル

入出力ポート：入力1, 出力1

設定可能パラメータ：相互コンダクタンス, 結合加重, 遅延時間

② 編集・管理機能

I. 共通編集機能

配線に自動で名前 (net 名) を割り当てられ、ユーザーが後から任意に変更できる。各モデルをコピーすると別インスタンスとして自動命名される。

II. GUIエディタ機能

ニューロンモデルや配線をキャンバス上に配置し、移動・コピー・削除などの操作を行える。

III. 保存機能

ネットワーク構造を位置情報・接続情報・パラメータとともにJSON形式で保存できる。

これらの要素のうち、本研究ではまず編集・管理機能の基盤的な部分を中心に実装した。実現した機能は以下のとおりである。

I. GUIエディタ機能

ニューロンモデルをキャンバス上に配置し、移動・コピー・削除が可能である。ノードの位置はドラッグ操作で更新され、それに接続する配線も自動的に追従する。ノードをコピーすると新しいインスタンスが生成され、自動的に名称を割り当てる。

II. 自動ネット名付与とインスタンス管理

配線で接続されたポート群を「ネット」として自動的に識別し、連結成分ごとにnet1, net2のような名前を割り当てる仕組みを導入した。ユーザーはこれらの名前を任意に変更できる。また、各モデルのコピー時には

1: 日大理工・学部・電子 2: 日大理工・教員・電子

別インスタンスとして新しい名前が付与され、重複を防ぐように設計している。

III. JSON形式による保存・読み込み

GUI上で構築されたネットワーク構造は、位置情報・接続情報・パラメータを含むJSON形式で保存される。読込時にはこれを再構築し、GUI上にネットワークを復元できる。この形式により、既存のシミュレーションエンジンに渡すための前処理が容易となる。

図1に、現時点でのGUIエディタ動作画面を示す。本図では、GUI上でニューロンモデルを配置し、それらを接続した例を示している。ユーザーは、まずキャンバス上にニューロン(N1, N2)を配置し、マウス操作により両者を接続することで、自動的にネット名(net1)が付与される。次に、N2をコピー操作することで複製ノード(N2_copy)が生成される。複製後、N2とN2_copyを接続すると、新たにnet2が自動生成され、ネットワーク構造として保持される。このように、ユーザーは配置・コピー・接続といった直感的な操作により、ネットワークを構築できる。

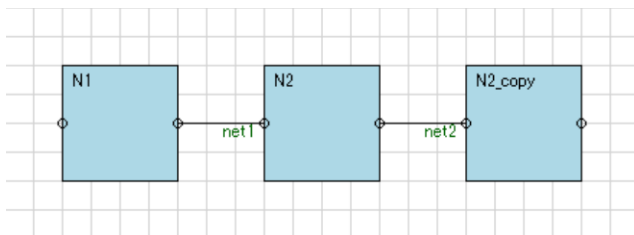


図1 現時点でのGUIエディタ動作画面

図2に、図1のネットワーク構造をJSON形式で出力した際の一部分を示す。ここでは、各ニューロンのIDや座標位置(X, Y)、さらに入力ポートおよび出力ポートに対応するネット名(NodeIn, NodeOut)が記録されていることがわかる。このようにして保存されたJSONファイルは、後に読み込むことでGUI上に同一のネットワークを再現できる。また、内部的にはこのデータを変換し、従来のシミュレーションエンジンが要求するネットリスト形式として利用可能である。

```
{
  "Neurons": [
    {
      "Id": "N1",
      "X": 100,
      "Y": 100,
      "NodeIn": "net1",
      "NodeOut": ""
    },
    {
      "Id": "N2",
      "X": 240,
      "Y": 100,
      "NodeIn": "net2",
      "NodeOut": "net1"
    }
  ],
}
```

図2 JSON形式での出力例の一部

現状、GUIエディタによりニューロンの配置・接続、自動ネット名管理、ネットワーク構造の保存・読み込みが可能となった。一方で、依然として、外部信号源モデル、シナプスモデル、ギャップジャンクションモデルといった重要要素の未対応、各モデルのパラメータをGUI上で編集する機能の不足、既存シミュレーションエンジンとの統合といった課題が残されている。

3. まとめ

本研究では、SNN解析におけるネットリスト記述の煩雑さを解消するため、GUI編集に対応した解析環境の構築を進めている。現時点で、ニューロン配置・接続、自動ネット名管理、JSON保存・読み込みを実装し、GUI上で直感的にネットワーク構造を編集できることを確認した。

今後は、シナプスモデルや外部信号源モデルなどの要素追加、パラメータ編集機能の拡充、既存シミュレーションエンジン統合を進め、より実用的なSNN設計・解析環境の実現を目指す。

謝辞

本研究は東京大学大規模集積システム設計教育研究センターを通し、シノプシス株式会社、及び日本ケイデンス株式会社の協力で行われたものである。

4. 参考文献

[1]飯島楓・佐々木芳樹「スパイクングニューラルネットワーク解析に最適化したシミュレータ作成に関する研究」, 卒業論文, 2024