数独への畳み込みネットワークの適用に関する学習の比較

Comparison of Learning on Application of Convolutional Networks to Sudoku.

中央大学大学院　理工学研究科　電気電子情報通信工学専攻

Electrical Electronic and Communication Engineering Faculty of Science and Engineering, Chuo University

石井　宏一郎

ISHII Koichiro

1. まえがき

今や「AI」「人工知能」といった言葉は誰もが知るほど世間一般に認知されるほどになっている。この空前の AI ブームは，ニューラルネットワークという人間の脳神経の信号伝達を模したアルゴリズムが，画像認識の分野にて従来の手法に比べて圧倒的であったことが発表されたことに端を発している。それから今日にいたるまで，ニューラルネットワークは機械学習の技術発展と性能の向上に大きく貢献してきた。その画像処理分野では，現在畳み込みニューラルネットワーク（CNN; Convolutional Neural Network，以下 CNN と記載）という方法が主流である。この手法の特徴は畳み込み演算と呼ばれる，入力と同じ次元のフィルタをずらしながら積和計算を行う処理である。この演算により，データの形状の保持したままの学習や，層を増やすにつれより高いレベルでの特徴の判別を行うことができる。しかしその一方で，データの形状には畳み込み演算が行えるように一定の規則性を持つ必要があった。そこで畳み込みニューラルネットワークを一般化し，より構造化された任意のデータ，主にグラフ構造に対するアプローチが様々考えられており[1]，グラフ畳み込みネットワーク（GCN; Graph Convolutional Networks）はその一つの手法である

本研究は数独をグラフ構造として解釈し，上記のグラフ畳み込みネットワークを適用することで数独の解答を試みる。類似の試みとして，数独の解答に CNN を用いた例はこれまでいくつか存在する[2][3][4]。その中では，画像処理と同様な処理を行ったものや，数独を解く上で鍵となる行，列，3×3 の区画をそれぞれ畳み込みフィルタで表現したものがあるが，いずれも複数個のフィルタや可変する中間層のノード数を考えると数独の特徴や計算過程の把握は難しい。そこで筆者らは，数独をグラフ構造にすることで特徴を実際に解く時にヒントとなる数字の候補に対応させることができると考え，グラフ畳み込みネットワークを用いて学習することを試みている[5]が，出力と正解ラベルの差が問題によって大きく変動してしまっていることがわかった。

本研究では同様のネットワークモデルを引き続き用い，モデル内部のパラメータや特徴量，計算式の変更によって誤差をへらすことを目標とする。

2. 準備

2.1 数独

数独とは，9×9 のマスに数字を埋めていく論理パズルである。各行，各列，そして 3×3 の各ブロック内に 1 から 9 までの数字を 1 回ずつのみ使用するというルールがあり，最初に与えられた 20〜30 個ほどの数字から，空マスに対し未知の数字を論理的に特定することでパズルを解いていく。この始めに埋まっている数字を以降，問題マスと呼ぶこととする。数独の問題及び解答例として，以下図 2-1 に示す[6]。

	8		7			1	4	9
				8	6	7	2	3
	7	2	9					5
	3	8			1		5	
	5						9	
	1		2			4	8	
7					9	6	3	
8	6	1	3	2				
4	9				7		1	

5	8	6	7	3	2	1	4	9
1	4	9	5	8	6	7	2	3
3	7	2	9	1	4	8	6	5
6	3	8	4	9	1	2	5	7
2	5	4	6	7	8	3	9	1
9	1	7	2	5	3	4	8	6
7	2	5	1	4	9	6	3	8
8	6	1	3	2	5	9	7	4
4	9	3	8	6	7	5	1	2

【図 2-1 数独の問題例と解答】

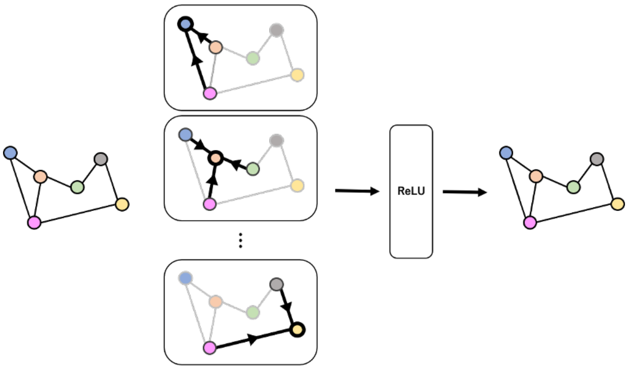
加えて，問題に対して解答が一意に決まらなければならないという条件がある。論理的に決定され，かつ一意の解答を持つという条件は，数独の問題作成や難易度決定に課題を与えている。一般に数独の難易度は，問題マスの個数によって決まることが多いが，上記の条件により必ずしも一致するわけではなく問題作成者によっては，実際に作成者自身が解いてみた体感によって難易度が決められたものも存在する。

2.2 グラフ畳み込み

画像処理分野で主流となっている CNN での畳み込み演算を，グラフ表現において再現したものがグラフ畳み込

みネットワークである。グラフ構造に対する畳み込み計算はこれまでに複数考案されているが，ここではその中でも最も主流となっているもの[7]を紹介する。

一回の畳み込みで各ノードは，それぞれ接続されているノードと過去の自身の情報を取得して重みを乗算した後，足し合わせる。これを最後に活性化関数に通し，次の層に受け渡す。この一連の処理の概略を図 2-2 に示す。



【図 2-2 グラフ畳み込み処理の流れ】

この処理における第i層の出力$h^{(i)}$は以下の計算式で表される。

$$h^{(i)} = \sigma(\widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{-\frac{1}{2}}h^{(i-1)}W^{(i)})$$

$\hat{A}$は自己ループを含めた各点の繋がりを表す隣接行列であり，次数行列$\widehat{D}$を用いた$\widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{-\frac{1}{2}}$での部分にて正規化している。$W^{(i)}$はネットワークの$i$層における学習のための重み行列，そして$\sigma(\cdot)$は活性化関数を表し，ここではランプ関数 ReLU を用いている。

グラフ畳み込みネットワークは上記の演算を繰り返し行うと，すべてのノードの表現が同じ値に収束するために，従来の深層学習とは異なりネットワーク層の数を増やすと精度が大きく落ちることがわかっている。そのため本文では 3 層のネットワークを用いることとした。これは 3 回の畳み込み演算では 3 ホップ隣接したノードの情報までを取り込むことができ，数独の各マスは 3 回ほどの演算で，81 マスすべての情報を取り込めるためである。

3. 研究

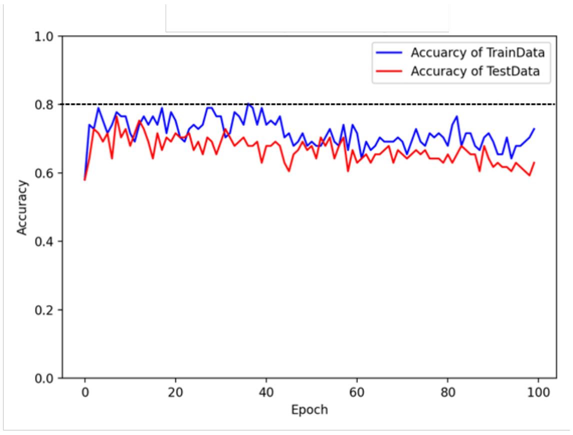
人間が実際に数独パズルを解く際，空マスに入る数字の候補を小さく書いていき周辺の数字を頼りに絞っていく。この解き方と同様に，初期配置から各マスの候補をベクトルに対応させ，これを各点の特徴量とした 81 ノードから成るグラフを構成する。また，そのまま数独をグラフ畳み込みネットワークに適用すると，すでに埋まっている問題マスも情報の更新の対象となってしまい，結果として全マスが全ての数字を等しい確率で出力してしまう。そのため，畳み込み演算時に隣接行列の問題マスにあたる部分を一時的に削ることでこれを回避している[5]。

3.1 全結合層との比較

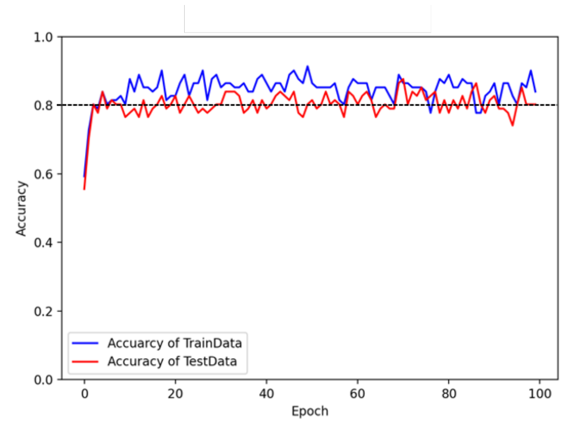
下式のように，グラフ畳み込みの式中の隣接行列（$\widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{-\frac{1}{2}}$）の部分を単位行列$I$に置き換えると，ネットワークの構成をそのままに，全結合層ニューラルネットワークとして考えることができる。

$$h^{(i)} = \sigma\left(\widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{-\frac{1}{2}}h^{(i-1)}W^{(i)}\right)$$
$$= \sigma\left(Ih^{(i-1)}W^{(i)}\right)$$

これを利用し，3 層すべてを入れ替え，グラフ畳み込み層と全結合層以上の場合を比較したのが，以下図 3-1 である。横軸は繰り返し回数，縦軸は一問の正解率を表し，図中の青線は訓練データ，赤線はテストデータでの正解率をそれぞれ示している。また，それぞれの問題は算出する際にランダムで逐一選ばれている。



(a) 全結合層



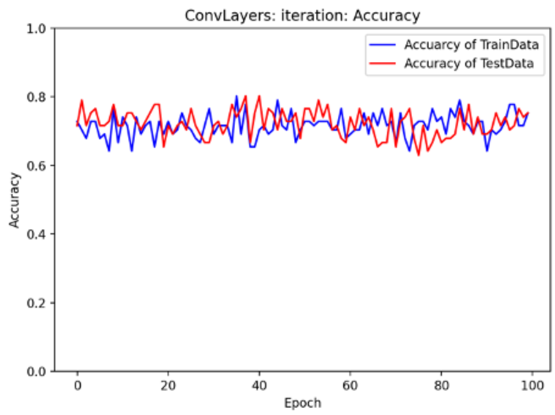
(b) グラフ畳み込み層

【図 3-1 全結合ネットワークとの比較】

ここでの全結合層は，隣接行列だけを置き換えているため一般的なものとは少し仕様が異なるが，グラフ畳み込み層の方の精度が高いことが確認できる。また，全結合層は上図より 0.8 ほどを頭打ちに緩やかに減少しているのに対し，畳み込み層では 0.8~0.9 の間まで振動しながらではあるが一定を保っている。

3.2 マスを一つずつ埋めていく

人が数独を解く際，候補を参考にしつつ確信したマスから一つずつ埋めていき，決して全てのマスを一気に埋めることはしない。そこでこの解き方を参考にし，モデルの出力のうち，割合が高いと判断したマスから一つずつ決定するように出力した時の結果が図 3-2 である。



【図 3-2 個別に出力した場合】

今までの一度に出力する方法とは違い，割合が高いものから一つごとに埋めていくため，学習の始めから正解率は高いが学習を重ねても大きな伸びは見られず，0.8 を頭打ちに，正解率 0.8〜0.6 間の振動をつづけている。正解率の上限に注目すれば，全結合層（図 3-1(a)）と変わりなく，大きな違いがないことがわかる。

この結果の理由としてモデルが行う「解き方」と「学習方法」の乖離が挙げられる。解き方は確率が高いものから個別に出力しているのに対し，学習方法としては最初の全体の配置そして各マスの候補から，全体を統計的に出すように学習している。実際，出力の経過を観察すると，一つのマスが確定し再度モデルが出力した時，出力値にさほど変化がないマスも多く見られた。

3.3 正規化ラプラシアン行列

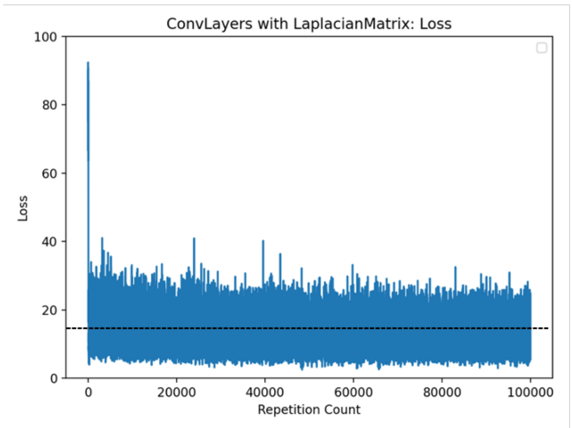
各マスは自身が所属している，行，列，3×3 の区画の他マスから情報を受け取る。グラフ畳み込みではそれに加え，自身のマスからも情報を受け取るが，それはあくまで過去の情報であり更新の対象として処理される。つまり，他マスから受け取る情報と自身のマスから受け取る情報，及びそれぞれのつながりの関係は少し意味合いが異なる。そこで，隣接行列の代わりにラプラシアン行列に置き換えることで，その接続関係の異なりを正負で表現することができると考えた。畳み込みで用いられる正規化ラプラシアン行列L^{sym}は以下の式で求められる。

$$L^{sym} = I - \widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{-\frac{1}{2}}$$

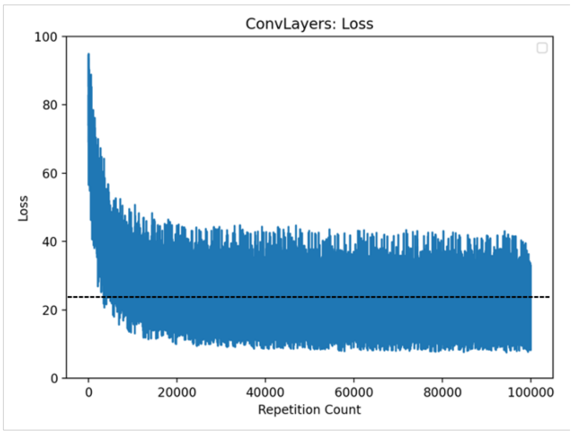
そのため，各畳み込み層の計算式は以下のように書き直す事ができる。

$$h^{(i)} = \sigma((I - \widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{-\frac{1}{2}})h^{(i-1)}W^{(i)})$$

損失値の推移が以下の図 3-3(a)である。縦軸は損失値，横軸は繰り返し回数を表す。損失値とは，出力値と正解ラベルとの差を交差エントロピー誤差として計算したものをここでは表している。また比較のため隣接行列を用いた場合の推移を図 3-3(b)に示している。



(a) ラプラシアン行列



(b) 隣接行列

【図 3-3 全結合ネットワークとの比較】

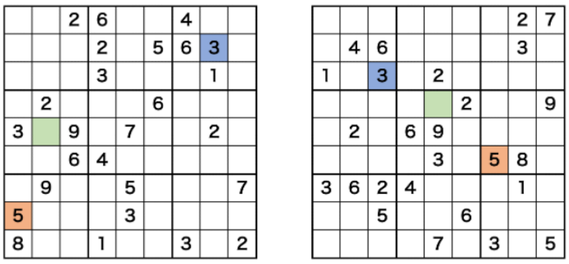
ラプラシアン行列を用いた場合，正解率に関しては特に大きな差は見られなかったが，損失値の比較（図 3-3）では振動の中心となる値，そして幅がいずれも小さくなっていることが確認できる。つまり出力は正解ラベルに高い割合の値を出力するようになったが，正解率に大き

な影響を与えるほどではないという結果になった。

4. まとめと課題

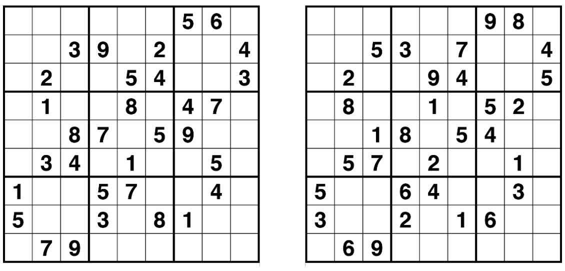
結果として，グラフ畳み込みのみの修正だけでは，全体の損失値の減少や正解率の少しの底上げにしか寄与しなかった。正解率が 10 割を常に下回っている以上，やはり実用的とは言い難い。この原因としては一つにパラメータの少なさが一つに挙げられる。このモデルが持つ動的な重み付き行列$W^{(i)}$は，各層が持つ 9×9 の行列のみである。CNN などと比較するとこの数はやはり小さい。この研究では人間が解く工程の再現が一つのテーマであり，各数字の候補がどのように移り変わるかという途中経過の形を保存するため，このパラメータ数を維持したが，パラメータを増やすため，グラフ畳み込み層や活性化関数の間に CNN のような別の手法を挟み込むことでの精度の向上は十分に考えられる。

数独をグラフ構造で置き換えることは学習の特徴量として「候補」を再現することで，人間の解き方を模することができるということを考えてのことだった。しかし，グラフにすることでの課題も同時に浮上した。例えば以下の 2 つの問題は，全てのマスの候補は一対一に対応しているため，グラフ構造にすることで全く同様の構造になる。色の同じマスが互いに同様の候補を持つことからも確認できる。



【図 4-1 同じ特徴量を持つ問題】

また数独に関しても課題は残っており，例えば以下のように，全く同じ問題マスの配置でも数字の配置の違いによって難易度が変わることがある（図 4-2）[8]。この問題では手筋の難しさから左の問題の方が難しいとされている。



【図 4-2 同じ配置だが難易度が異なる問題】

もちろんこの 2 つは解き方の手順も異なる。そのため数独における学習では，この性質や学習の手順をいかにうまく表現するかが課題である。

2.1 数独にもある通り，現在数独の問題の難易度決定は曖昧な状況である。ニューラルネットワークにより難易度を計算する手法[9]や作成支援[10]も提案されているなど，問題を解く以上に作成側にも課題は存在している。数独をグラフ構造に置き換えるという発想は[11]で既に行われており，ここでは彩色多項式を用いて最初に与えられる問題マスの数の最小を求めている。このようにグラフにすることによって数独は彩色問題として扱うことができ，同時に数独を抽象化する際の手段と考えることができる。そのためアルゴリズムへの転用なども含めると，上記のような課題に対して十分活用する価値があると考えている。

文 献

[1] Ziwei Zhang, Peng Cui, Wenwu Zhu: Deep Learning on Graphs: A Survey, IEEE Transactions on Knowledge and Data Engineering, Volume 34, pp 249–270 (2022)

[2] Shiva Verma: Solving Sudoku with Convolution Neural Network | Keras (2019) https://towardsdatascience.com/solving-sudoku-with-convolution-neural-network-keras-655ba4be3b11

[3] Artificial Intelligence Solves Sudoku: Convolutional Neural Networks (CNN). (2021) https://medium.com/analytics-vidhya/how-to-solve-sudoku-with-convolutional-neural-networks-cnn-c92be8830c52

[4] GitHub – nnethercott /Sudoku-CNN (2019) https://github.com/nnethercott/Sudoku-CNN

[5] 石井宏一郎 田村裕: Solving Sudoku with Graph Convolutional Networks, The 40th JSST Annual Conf. Student Session, pp.21-24 (2021)

[6] ナンプレ自動生成エンジン V2.0 – PUZZLE GneRator JaPan http://jp.sudoku99.com/search/E/E/EEDB431778D_43177.html

[7] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, Max Welling: Modeling Relational Data with Graph Convolutional Networks, European Semantic Web Conference ESWC 2018: The Semantic Web pp 593-607 (2018)

[8] 【第 1 話】ナンプレの難しさは何で決まる？ その 1 https://puzkan.jp/03-2

[9] 天野秀亮 篠埜功 杉本徹: ニューラルネットワークによる数独の難易度判定手法の提案 信学技報, vol. 112, no. 319, AI2012-17, pp. 13-18 (2012)

[10] 前田一貴 奥乃博: 数独の問題作成支援システムの設計と開発 全国大会講演論文集 第 70 回(コンピュータと人間社会), 799-800, (2008)

[11] Jason Rosenhouse, Laura Taalman 著．小野木明恵 訳: 数独を数学する -世界中を魅了するパズルの奥深い世界-，青木社（2014)