

## 複雑な論理演算を実現するニューラルネットワークの構成について

古澤 慎太郎<sup>†</sup> 山本 修身<sup>‡</sup>  
 名城大学<sup>†</sup> 名城大学<sup>‡</sup>

## 1 はじめに

近年、画像処理やボードゲームといった多くの分野において、ニューラルネットワーク (NN) を用いた人工知能モデルが提案され、成果をあげている。NN は、予め入力とその解のペアを多数用意し、NN による推論結果と解を比較して内部の状態を変化させることで学習をすることができる [1]。本研究は、素因数分解などの従来のアルゴリズムでは多くの計算量が必要となる操作を NN 上で実現し、これらの問題に対し比較的短い時間で解の候補を与えることが目的である。曖昧さがないこれらの問題を学習することは困難であると考えられるため、足がかりとしてビット演算のような比較的単純な論理演算を取り挙げ、それがうまく機能するような NN の構造や学習の枠組みについて模索している。なお、畳み込みニューラルネットワーク (CNN) を用いて内部の信号経路を制限することにより、24bit までの加算器を構成することができている。しかし、これを超える次元の加算器や、乗算器については精度が不十分である。本稿では、これらの計算機を構成するための手法として、再帰型ニューラルネットワーク (RNN) を用いて 1 つの層を繰り返し使用する手法と、NN 内部に直接入出力を与える手法の 2 つを提案し検証を行った。

## 2 提案手法とその評価

## 2.1 RNN

通常の論理回路では、加算機は 1bit+1bit の全加算器を複数連結することで構成される。また、これは 1 つの加算器へ出力の一部を再帰的に入力することと等価なので、1 つの全加算器へ順序的に入力することでも実現可能である。このことから、信号の

再帰入力を行うことが共通している RNN を用いることで、論理回路に似た伝播を NN 上で実現できると考えた。すなわち、入力  $x, y$  と出力  $z$  を各ビットで区切られた時系列データとして与え、下位ビットから順次入力することで推論を行う。このとき、ある時刻の出力  $z_k$  は出力は RNN への再帰的入力も用いられる。今回、RNN を使った場合の精度を評価するため、以下の検証を行った。

$n = 16, 32, 64, 128$  として、 $n$ bit の乱数  $x, y$  と  $z = x + y$  となる  $n$ bit の整数  $z$  の組を教師データとする。この教師データ 10,000 個を図 1 に示す RNN に学習させ、各 epoch における精度の推移を比較した。その結果を図 2 に示す。学習後、どのビット長においても 95% 以上の精度となっており、加算器が十分に構成できたといえる。

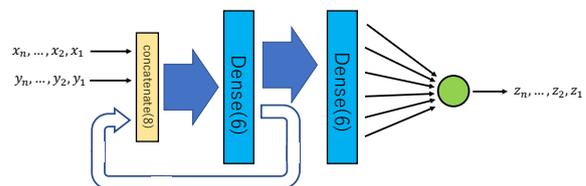


図1 検証で使用した RNN の構造図。ただし、 $(n)$  は出力の次元を表す。中央の全結合層の信号は出力層へ伝播と入力層への回帰に用いられる。

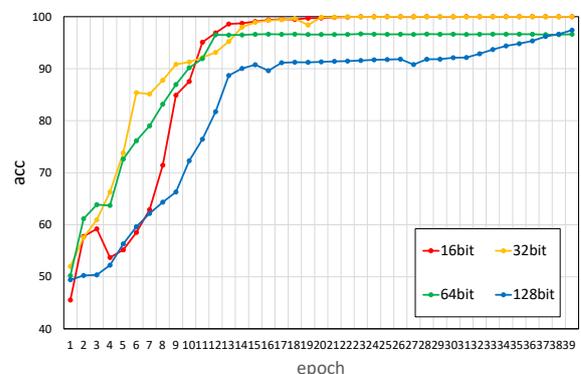


図2  $n$ bit の加法の学習中の全てのビットでの平均精度の推移。なお、精度の検証には教師データと別の独立に生成したデータ 1,000 個を用いた。

Construction of neural networks implementing complex logical operations.

<sup>†</sup> Shintaro Furusawa, Meijo University

<sup>‡</sup> Osami Yamamoto, Meijo University

## 2.2 多入力多出力モデル

NN では一般に、学習させる計算が複雑または巨大になるにつれ、内部に必要な層の数は増加する。一方で、層の数が増えるにつれ、入力層で与えた信号は減衰または変質しやすくなり、学習は困難になる。また、逆に単純な計算に対し過剰に層を与えると、内部状態の収束が難しくなる。これらを解決するために、隠れ層の途中から入出力を行えるような NN の導入を提案する。すなわち図 3 に示すように、ユニット  $U$  を定義してこれを連結することで NN 全体を構成する。これにより、入力信号の変質や減衰を低減できるほか、単純な計算に過剰な自由度が与えられる問題を回避できる。今回、このモデルを使った場合の精度を評価するため、以下の検証を行った。

12bit の乱数  $x, y$  と  $z = x \times y$  となる 24bit の整数  $z$  の組 10,000 個を教師データとする。この教師データを以下の 3 つの NN

- (1) 5 層, 10 層の全結合型 NN,
- (2) 幅 24, 24 層の CNN,
- (3) 提案した多入力多出力の NN

に学習させ、各ビットの精度を比較する。ただし、(3) で用いる NN は図 4 で示すユニットを 24 個接続

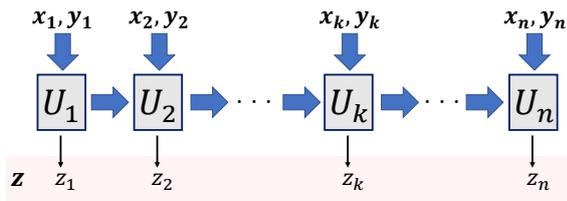


図 3 提案した多入力多出力モデルの構造図。

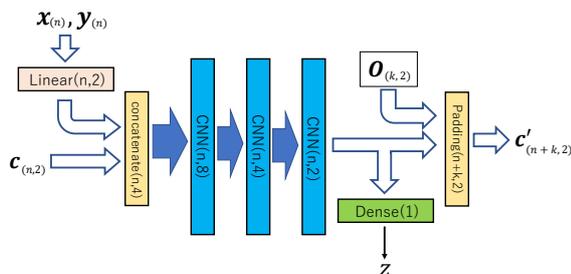


図 4 検証で使用したユニット  $U$  の構造図。ただし、 $(m, n)$  は信号または出力される行列の次元を表す。

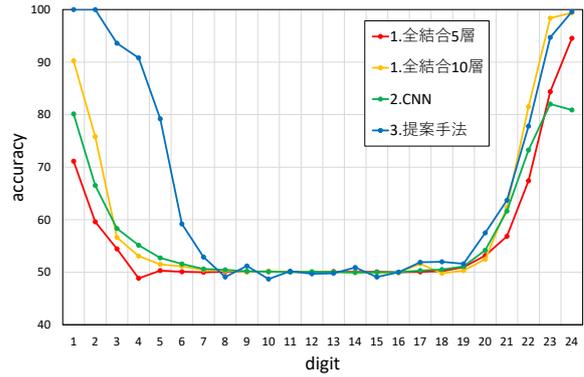


図 5 30epoch の学習後の各桁の精度。なお、精度の検証には教師データと別の独立に生成したデータ 1,000 個を用いた。

し構成した。この検証結果を図 5 に示す。従来手法と比較し、特に下位ビットにおいて精度の向上がみられ、下位 4bit では 90% 以上の精度を持つものが構成できた。

## 3 おわりに

本稿では、RNN を用いて高次元の加算器を実現する手法と、乗算器を構成するたの多入力多出力型の NN を提案し、検証した。加算器については、検証したどのビット長においても十分な精度を持つ NN が構成でき、任意長の加算器が実現できたと考えられる。一方で、入力を時系列データとして扱う性質上、計算の並列化が不可能であり、入力の次元に比例して NN の計算時間が増加してしまうため、計算を高速化できる手法については検討の余地がある。乗算器については、特に下位のビットにおいて従来手法からの精度の向上が見られたが、実用可能な精度を持つ範囲は依然として非常に狭いため、今後も新たな手法について検討する必要がある。

## 参考文献

- [1] Islam M., Chen G., and Jin S.: An overview of neural network. *American Journal of Neural Networks and Applications* 5.1: 7-11. 2019.
- [2] Nielsen, Michael A.: *Neural Networks and Deep Learning*. Vol. 25. San Francisco, CA: Determination press, 2015.
- [3] Gulli, Antonio, and Sujit Pal: *Deep Learning with Keras*. Packt Publishing Ltd, 2017.