

2023年度

卒業論文



論文題目

WANNにおける活性化関数の慎重な選択

研究者

2031133 増田 瑞樹

指導教員

山口 智 教授

2024年1月xx日

目 次

1	はじめに	1
2	ニューラルネットワーク	3
2.1	ニューラル素子	3
2.2	活性化関数	4
	謝辞	6

1 はじめに

生物学における先天的能力 (Precociality) とは、動物が生まれた瞬間からすでに持っている能力のことである。例えば、トカゲやヘビは生まれ持って捕食者から逃れる能力を有している。また、アヒルは孵化後すぐに泳いだり食事をする事ができ、七面鳥は一度も見たことがない捕食者を視認する事ができる。これは、動物の脳は高度に構造化された状態で生まれ、その構造はゲノムに記憶されていることを意味する。Zador は生物学的な学習について『動物の行動の多くは生得的なものであり学習によって生じるものではない。動物の脳は AI 研究者が思い描くようななんでも学べる汎用的な学習アルゴリズムを備えた白紙の状態ではない。』と強調している [3]。

Weight Agnostic Neural Networks(WANN) は、2019 年に Adam Gaier と David Ha によって発表されたシナプス荷重に依存しないネットワークの探索アルゴリズムである [1]。WANN は、NEAT[2] をベースに作られており、どのようなシナプス荷重においてもタスクを解ける性質をもつネットワーク、つまり構造自体にタスクを解く機能が備わっている。これは遺伝的アルゴリズム [5] を用いた Neuro-Evolution[4] の手法から実現できる。

WANN の個体変異の 1 つに、ノードの活性化関数の変更が行われる。隠れ層からランダムに選択されたノードの活性化関数は現在採用されている以外の活性化関数へランダムに変更される。これは探索後期において、それまでの良かったノードの出力を反転させてしまう懸念がある。

本論文では、活性化関数を変更する際の確率を関数同士の距離関数が小さいほど選ばれやすいようにする手法を提案する。距離関数が小さいことは、関数同士が似ているを意味し、活性化関数の変更により出力の大きな変更が起こりにくくなる。距離関数には 0 付近の活性化関数同士の出力の差を積分と、実際にその個体が体験した入力ノードから推測できる該当ノードの出力の差の合計を採用する。得られた距離に ϵ を加え逆数を取った値をルーレット選択することで活性化関数の変更前の出力と変更後の出力を小さくし、それまでの

良い出力を著しく損ねることを緩和する.

実験では **ここから追加**

2 ニューラルネットワーク

2.1 ニューラル素子

人間を含む生命の脳を構成する神経細胞はニューロンと呼ばれ、人間の脳には140億個のニューロンがありそれぞれのニューロンは平均約8,000個のシナプスを持つとされている。ニューラルネットワークのシナプスとノードはニューロンをモデルとし、計算機上でニューロンをシミュレートできるよう設計されている。[6]

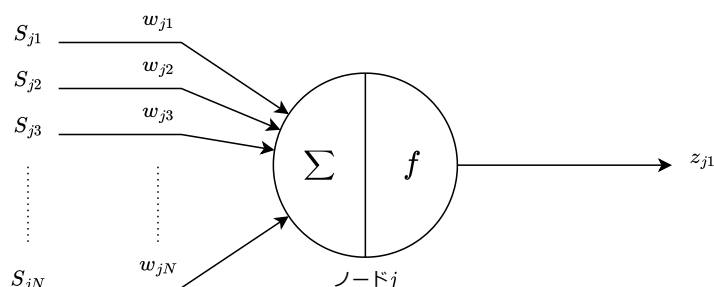


図 1: シナプスとノードの関係

ここでは、ノード j への N 個の入力 $S_{1j}, S_{2j}, S_{3j}, \dots, S_{Nj}$ に対して各々の重み $w_{j1}, w_{j2}, w_{j3}, \dots, w_{jN}$ となっている。この素子は入力とバイアス b_j を足した値 U_j を活性化関数 f_j の入力とし、活性化関数の出力をノードの出力 z_j とする。

$$U_j = \sum_{i=1}^N S_{ji} w_{ji} + b_j \quad (1)$$

$$z_j = f_j(U_j) \quad (2)$$

このノードを組み合わせ、図2のような階層型ニューラルネットワークを考える。

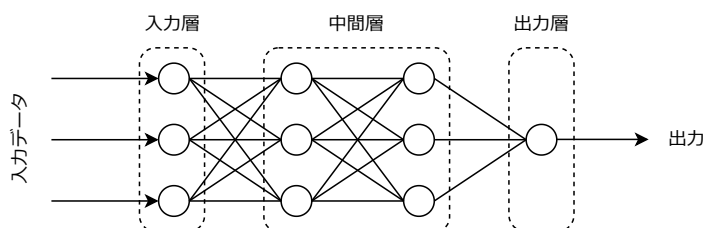


図 2: 階層型ニューラルネットワーク

階層型ニューラルネットワークは、入力層、中間層、出力層からなり、まずタスクを解くための判断材料となる入力データが入力層のノードに入力される。中間層の入力はひとつ前の層の出力からなり、出力層の結果がネットワークの解答になる。

2.2 活性化関数

活性化関数は、与えられた入力をどのように活性化するか動作を変更するものである。線形分離によって解くことができない問題に対しては、この活性化は非線形変換である必要がある。[7] 具体的には実数空間 V, W において V から W への写像 f が以下の性質を満たさない変換である必要がある。ただし $x, y \in V, c \in \mathbb{R}$ とする。

1. 加法性: $f(x + y) = f(x) + f(y)$
2. 斉一次性: $f(cx) = cf(x)$

複数のノードの出力の合計をあるひとつノードとみなすことは線形変換に過ぎず、線形変換の繰り返しはその結果もまた線形性が保たれ、活性化関数を持たないネットワークは隠れ層を持つ場合と持たない場合で表現できる幅は変わらない。複数のニューロンからの入力の合計を活性化関数を通すことはネットワークがより豊かな表現力を持つことを意味する。代表的な活性化関数 f は次のようなものがある。

1. ReLU 関数

$$f(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases} \quad (3)$$

2. tanh 関数

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

謝辞

山口先生有り難う！

参考文献

- [1] Gaier, Adam, and David Ha. "Weight agnostic neural networks." Advances in neural information processing systems 32 (2019).
- [2] Stanley, Kenneth O., and Risto Miikkulainen. "Evolving neural networks through augmenting topologies." Evolutionary computation 10.2 (2002): 99-127.
- [3] Zador, Anthony M. "A critique of pure learning and what artificial neural networks can learn from animal brains." Nature communications 10.1 (2019): 3770.
- [4] Braun, Heinrich, and Joachim Weisbrod. "Evolving neural feedforward networks." Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Innsbruck, Austria, 1993. Vienna: Springer Vienna, 1993.
- [5] 坂和正敏, 田中雅博. "遺伝的アルゴリズム" 朝倉書店 (2002).
- [6] 伊庭齊志. "進化計算と深層学習 -創発する知能-" オーム社 (2015).
- [7] Dubey, Shiv Ram, Satish Kumar Singh, and Bidyut Baran Chaudhuri. "Activation functions in deep learning: A comprehensive survey and benchmark." Neurocomputing (2022).