

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
3	Weight Agnostic Neural Networks	6
3.1	変異	7
4	提案手法	9
4.1	距離関数	10
	謝辞	12

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ことを意味し、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

ニューラルネットワークとは機械学習のアルゴリズムで、人工的に構築された神経細胞を模したネットワークからなる。入力層、中間層、出力層から構成され、各層のニューロンは重み付きの結合を通じて信号を伝える [6]。データの特徴を抽出し、分類、予測、生成などのタスクに適用され、画像認識、自然言語処理、音声認識など多岐にわたる応用がある。

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1つ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]

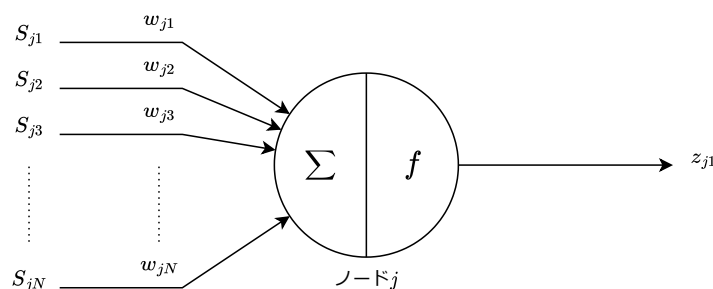


図 1: シナプスとノードの関係

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．

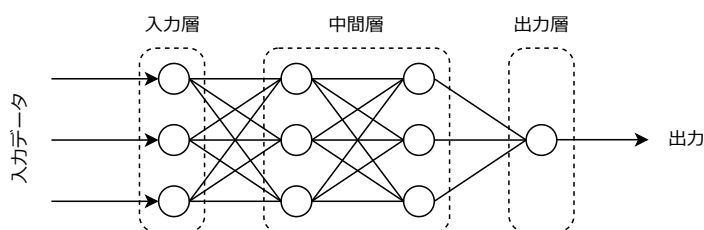


図 2: 階層型ニューラルネットワーク

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

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

今回の実験では、これら 2 種類を含む 10 種類の活性化関数を用いる。

3 Weight Agnostic Neural Networks

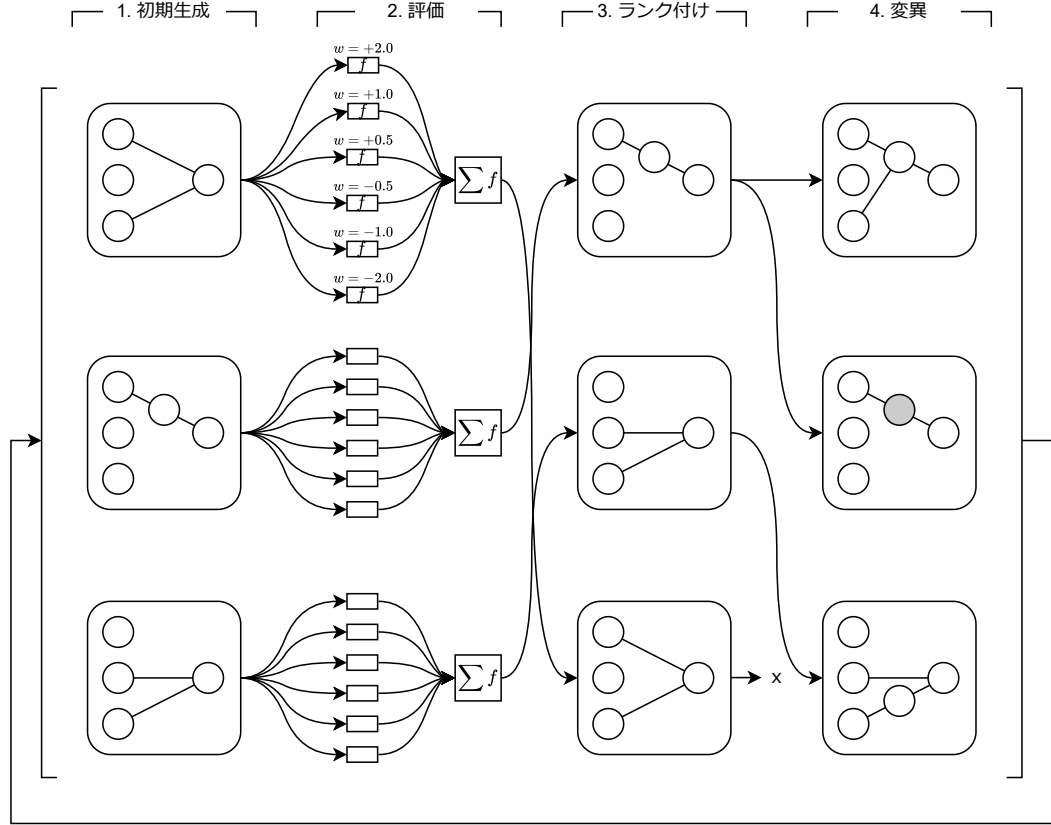


図 3: WANNs の概略図

Weight Agnostic Neural Networks(WANNs) は, 2019 年に Adam Gaier と David Ha に よって発表されたニューラルネットワークの構造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, 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. 多くのニューラルネットワークはシナプス荷重を更新することで, ネットワークの精度を向上させているが, WANNs に よって生成された個体は, ネットワーク構造自体がタスクを解く性質を持っており, どんなシナプス荷重においてもそれなりの精度でタスクを解くことができる. これは生物の先天的能力と似た性質であると言える [3].

また, WANNs ネットワークはそれぞれのノードが別々の活性化関数を持っている. WANN の概略図を図 3 に示す. WANN における探索は以下の動作によって実現する.

1. 初期生成個体は最初, 最もシンプルな構造を持つネットワークからなる. 中間層は存在せず, 入力層のうちの一部からシナプスが出力層に接続されている.
2. 評価各個体に対して, ネットワーク全体の共有重みを使用してタスクを実行する. 本論文では, $-2.0, -1.0, -0.5, +0.5, +1.0, +2.0$ の 6 つの共有重みを使用する. また, タスクを解く際の初期状態により優れた結果を残せずに誤った評価をしないよう, それぞれの共有重みにて 4 回のタスクを実行する. 最終的に 6 つの共有重みと 4 回の試行から得られた 24 個の評価値を合計する.
3. ランク付け個体をランク付けする. 上位の優れた個体は変更のないまま次世代へ保存され, 下位の劣った個体は淘汰される. このランク付けはネットワークの評価値とネットワークの複雑さから算出される.
4. 変異個体をランクのトーナメント選択 [5] によって選択し, 変異を起こした次世代の個体として保存する.

保存された個体は再度評価, ランク付け, 変異を行い, 任意の世代数まで繰り返される.

3.1 変異

図に示したように, 個体の変異にはノードの挿入, 接続の追加, 活性化関数の変更のいずれかを行い, 各操作の具体的な手順は以下になる.

1. ノードの挿入ネットワークの持つひとつの接続をランダムに選択し, 接続の途中にひとつのノードを追加する. この時の活性化関数はランダムに選択される.

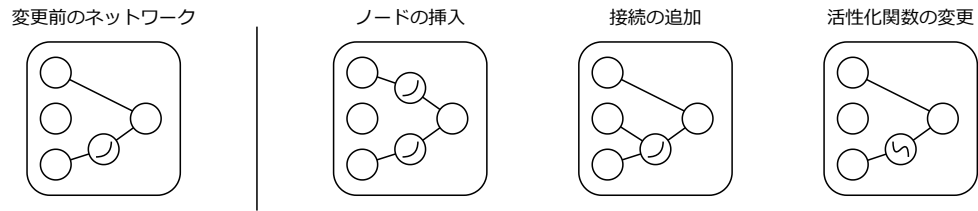


図 4: WANNs のネットワーク変異

2. 接続の追加接続を持たないふたつのノードを選択し、それらをつなぐ接続を追加する。
3. 活性化関数の変更ネットワークの持つ中間層のノードを選択し、そのノードの活性化関数を変更する。活性化関数は現在選択されている関数以外の関数がランダムに選択される。このときの活性化関数は linear, step, sin, cosine, Gaussian, tanh, sigmoid, inverse, absolute value, ReLU を利用する。

4 提案手法

既存手法の問題点として、ランダムな活性化関数の変更が挙げられる。探索後期において良いノードの出力の個体が生まれたとしても、例えば良いノードの活性化関数を linear 関数 $f(x) = x$ から inverse 関数 $f(x) = -x$ に変更されてしまうと、ノードの出力は反転されてしまう。ネットワークの一部の構造化された部分の出力の大小は、同符号であれば出力層に大きな変化をもたらさないが、出力が0より大きい小さいかは、ネットワークの出力は大きく変わることが知られている [1]。

提案手法ではこの問題を緩和するために、活性化関数の慎重な選択について言及する。2つの活性化関数の距離を計算し、距離が小さければ小さいほど変更先の活性化関数として選択されやすくなる。

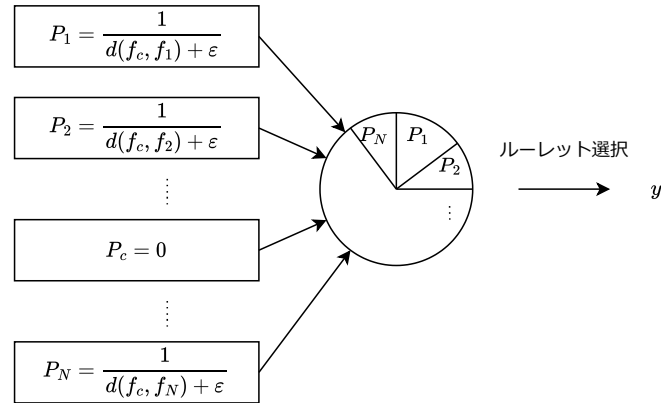


図 5: 提案手法の概略図

$$P_i = \begin{cases} \frac{1}{d(f_c, f_i) + \epsilon_n} & (i \neq c) \\ 0 & (i = c) \end{cases} \quad (5)$$

$$\epsilon_n = s * \epsilon_{n-1} \quad (6)$$

表 1: 変数の説明

変数	意味
P_i	c から i へ活性化関数 ID が変更される見込み
$d(f_c, f_i)$	活性化関数が c と i の距離
f_i	ID が i の活性化関数
i	活性化関数 ID
c	現在の活性化関数 ID

まず現在の活性化関数 f_c と、すべての活性化関数との距離を計算する。この時の距離が小さいことは両者の関数が似ていることを意味し、距離の短い関数への変更は既存手法の問題点である出力の反転を防ぐ。距離が小さければ小さいほど選択されやすくするため、距離と ϵ の和の逆数をルーレット選択の材料とする。 ϵ が大きくなると、ルーレット選択によって選ばれる確率はランダムに近くなる。 ϵ は探索初期においては、良いノードの出力を持つ個体が少ないことから、良い出力の個体を反転させてしまうデメリットより、悪い出力の個体を反転させるメリットの方が大きいと考え、探索初期の ϵ は大きく、探索後期の ϵ は小さく設定する。

距離関数 d については、範囲内の関数の差と、個体が経験した入力の際の出力の差を採用する。

4.1 距離関数

式 (7) と式 (8) は関数 f_a と f_b の範囲内の出力の差と、個体が経験した入力の際の出力の差を意味している。

$$d(f_a, f_b) = \int_{-r}^r (f_a(x) - f_b(x))^2 \quad (7)$$

$$d(f_a, f_b) = \sum_m (f_a(in_m) - f_b(in_m))^2 \quad (8)$$

表 2: 変数の説明

変数	意味
$d(f_a, f_b)$	活性化関数が a と b の距離
f_i	ID が i の活性化関数
m	ミニバッチサイズ \times 共有重み
in	経験したノードに入力される値
ϵ_n	探索範囲を考慮する
r	関数の考慮範囲
k	ϵ の減衰率, 0.995 など

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] 伊庭斉志. "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).