

WANNにおけるシナプス荷重の調整と活性化関数の慎重な選択

(指導教員 山口 智 教授)

山口研究室 2031133 増田 瑞樹

1. WANN

Weight Agnostic Neural Networks(WANN) は、ネットワーク構造を更新することでどんなシナプス荷重に対してもそれなりの精度でタスクを解くことができるようになる。変更するノードは隠れ層からランダムに選択され、現在採用されている活性化関数をのぞいた活性化関数の中から、同等の確率で選択される。また、ネットワークの評価には-2.0, -1.0, -0.5, +0.5, +1.0, +2.0の共有重みを用いた6つの評価の平均を採用している。

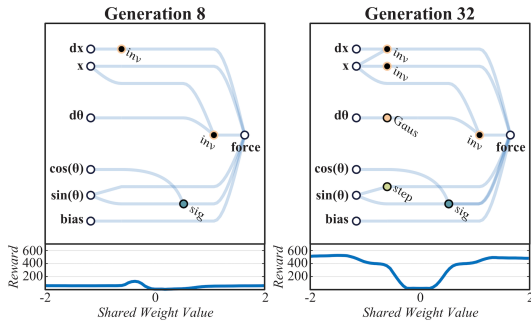


図 1: 共有重みと評価値とネットワーク構造

2. 提案手法

提案手法では、慎重な活性化関数の選択を実現するために、関数同士の距離を2つの方法で求めている。1つ目は関数同士の差の積分、2つ目は関数同士のミニバッチサイズ分の入力に対する出力の差の合計を使用している。式 (3) で使用されている ϵ は式 (4) に従い変動し、世代数の小さいうちは ϵ の値は大きく擬距離関数の影響が小さくなり、世代数の大きくなると ϵ の値は小さくなり擬距離関数を大きく考慮するようになる。変更先の関数として擬距離の近いものを選ばれやすくなるようにするのはそれまでの良い出力を反転させないようにするためだが、そもそも世代数の小さいこたいの出力は、良いものであるどころか悪いものである可能性が高い。よって、提案手法の ϵ を最初から小さくすると悪い出力がなかなか良い方に変更されないことが懸念される。

$$d(f_a, f_b) = \int_{-r}^r (f_a(x) - f_b(x))^2 dx \quad (1)$$

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

$$P_i = \begin{cases} \frac{1}{d(f_s, f_i) + \epsilon_n} & (i \neq s) \\ 0 & (i = s) \end{cases} \quad (3)$$

$$\epsilon_n = k * \epsilon_{n-1} \quad (4)$$

表 1: 説明

変数	意味
P_i	s から i へ活性化関数 ID が変更される見込み
$d(f_s, f_i)$	活性化関数が s と i の距離
f_i	ID が i の活性化関数
i	活性化関数 ID
s	現在の活性化関数 ID
m	ミニバッチサイズ x 共有重み
in	ノードに入力される値
ϵ_n	逆数の大きさを抑えるための小さい値
r	関数の考慮範囲
k	ϵ の減衰率, 0.995 など

3. 距離関数

式 (1) や式 (2) の関数は、2つの活性化関数がどれだけ似ているかを示すものである。これらの関数が距離の性質を持つかどうかを示す。距離関数として満たすべき条件は以下の4つの性質である。

性質	定義
非負性	$d(x, y) \geq 0$
同一律	$d(x, y) = 0 \Leftrightarrow x = y$
対称律	$d(x, y) = d(y, x)$
三角不等式	$d(x, y) \leq d(x, z) + d(z, y)$

4. 進捗

入力ノード情報の取得、任意の個体、ノード、共有重み、入力状況での出力の取得は実装済み。現在プログラムを走らせている。

5. 今後の目標

- 活性化関数変更方法による適応度の比較
- シナプス荷重の調整についての実装