

ドライバの運転行動の数理モデルの探索

ヒューマンロボティクス研究室

M1 浮田凌佑

研究背景

人間が運転しやすい自動車
違和感のない自動運転車の開発

ドライバの運転行動の理解・予測が重要

ドライバの行動理解のために、様々なドライバモデルが開発

- ・前方注視モデル 1990
 - ・Two-point steering model 2004
 - ・↑にオプティカルフローを掛け合わせたドライバモデル 2018
- etc

従来手法

変数



ゲイン

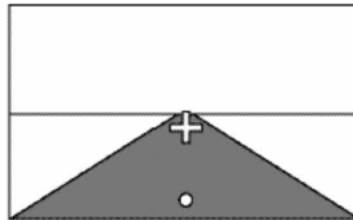


実際のドライバの操舵データのフィッティングによるゲインの学習
学習されたゲインの「強度・符号」によって、個人差や走行環境の差を考察

①Two-point model

Near point : 中央レーンの維持

Far point : 次の道路の予測



$$\dot{\phi} = k_f \dot{\theta}_f + k_n \dot{\theta}_n + k_I \theta_n$$

②Retinal-flow(視野角の変化量)を加味したモデル



Near・Far pointにおける横偏差 : ϵ

+

Retinal-flow : u

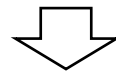
$$\dot{\delta} = \frac{1}{1 + \Delta T_s} (K_1 \epsilon_{far} + K_2 \epsilon_{near} + K_3 u_{far} + K_4 u_{near})$$

問題点①

$$\dot{\phi} = k_f \dot{\theta}_f + k_n \dot{\theta}_n + k_I \theta_n$$

$$\dot{\delta} = \frac{1}{1 + \Delta T_S} (K_1 \epsilon_{far} + K_2 \epsilon_{near} + K_3 u_{far} + K_4 u_{near})$$

入力情報(変数)が主観的に選定
構造は自明なものとして扱われている



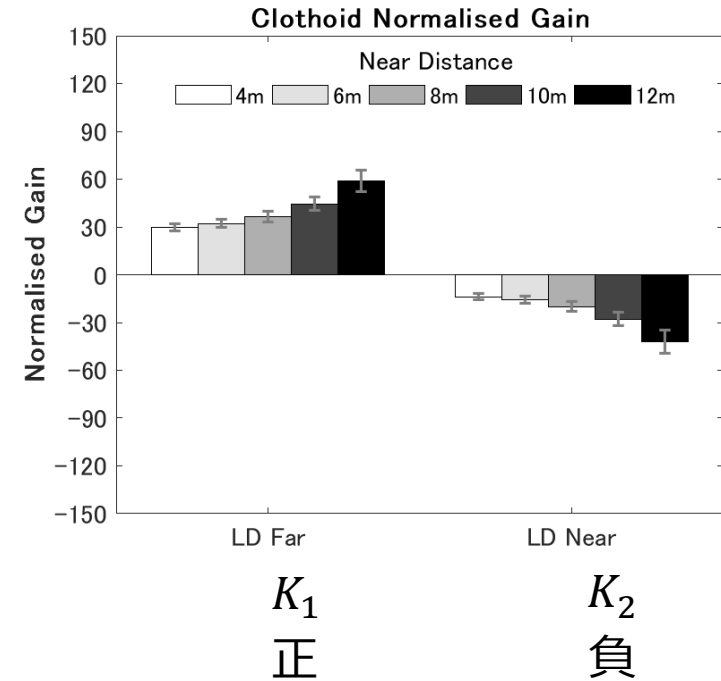
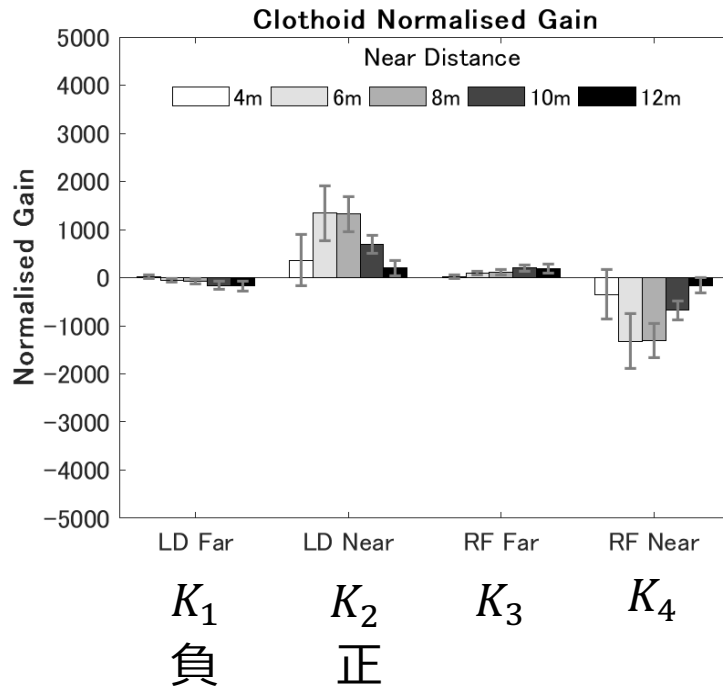
構造自体の探索に新規性

問題点②

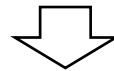
Retinal-flow(視野角の変化量)を加味したモデル

$$\dot{\delta} = \frac{1}{1 + \Delta T_S} (K_1 \epsilon_{far} + K_2 \epsilon_{near} + K_3 u_{far} + K_4 u_{near})$$

$$\dot{\delta} = \frac{1}{1 + \Delta T_S} (K_1 \epsilon_{far} + K_2 \epsilon_{near} + \cancel{K_3 u_{far}} + \cancel{K_4 u_{near}})$$



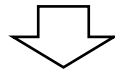
モデル構造によって結果が異なる



モデル自体を最適化すべき

研究の目的

- 入力情報が主観的に選定されている
- 構造は自明なものとして扱われている
- モデル構造によって結果は異なる



構造自体の探索に新規性，モデル自体を最適化すべき



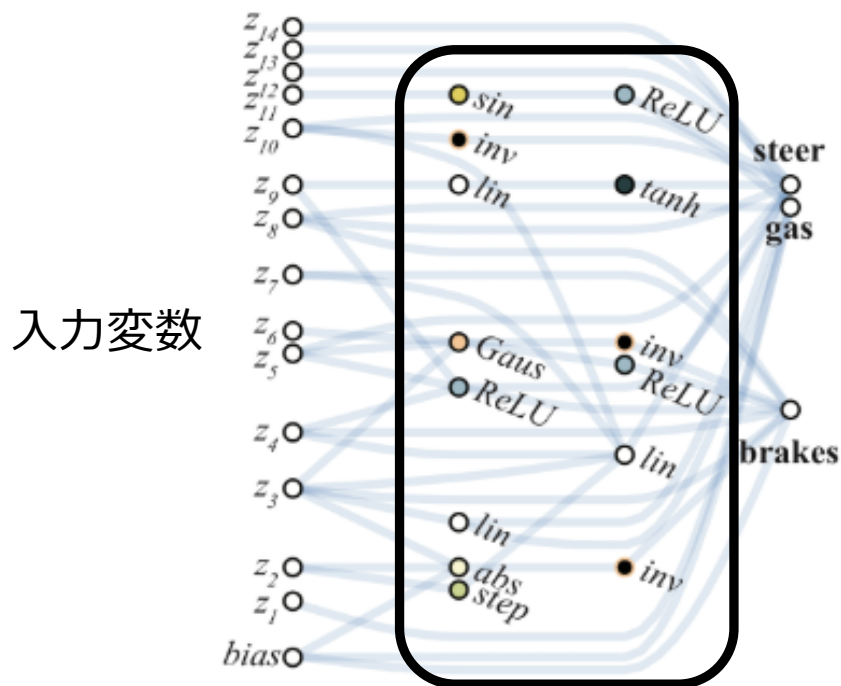
変数・構造の探索によるドライバモデルの開発，評価

モデルの探索手法

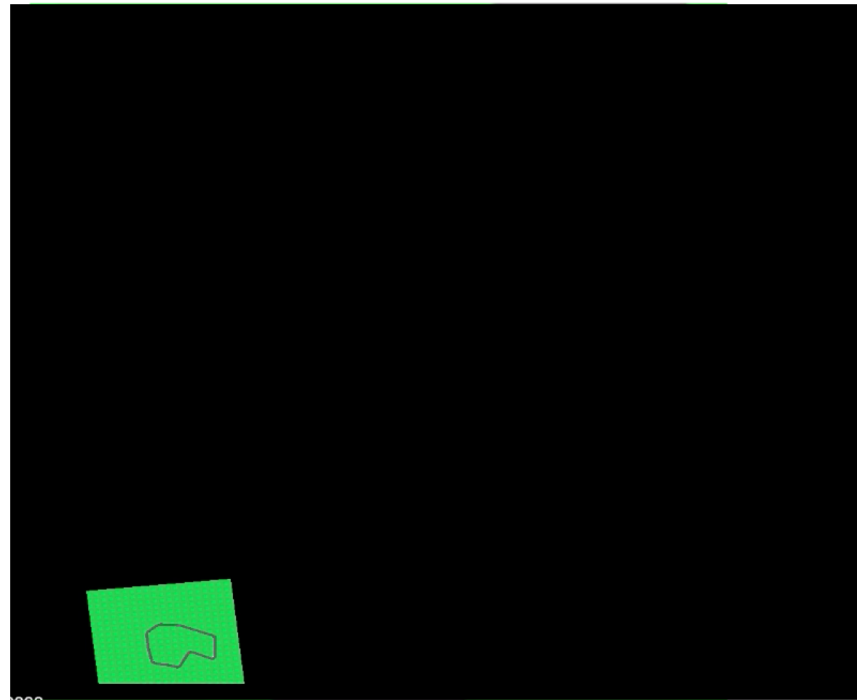
先行研究: Weight Agnostic Neural Networks (WANN)[2019]

ニューラルネットワークの重みでなく構造を探索する手法

例: z_1 -14までの入力変数が与えられた時の自動車の運転



必要な変数・構造を探索



Weight Agnostic Neural Networks [2019]

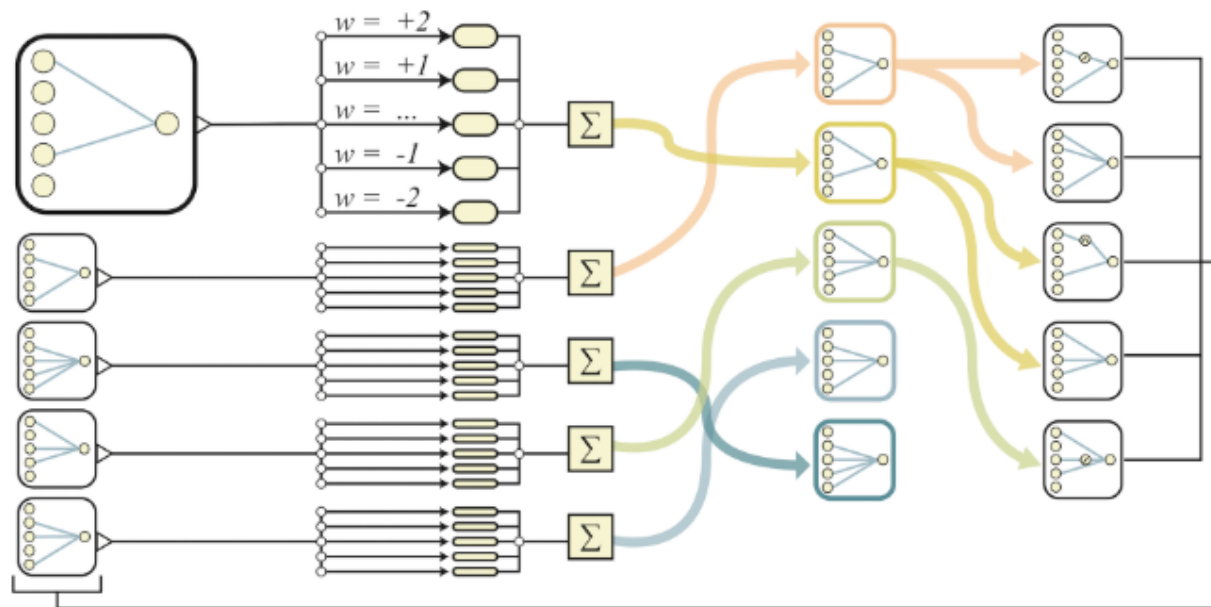
探索アルゴリズム

1.) Initialize
Create population of minimal networks.

2.) Evaluate
Test with range of shared weight values.

3.) Rank
Rank by performance and complexity

4.) Vary
Create new population by varying best networks.



1, シンプルなネットワーク群を作成する

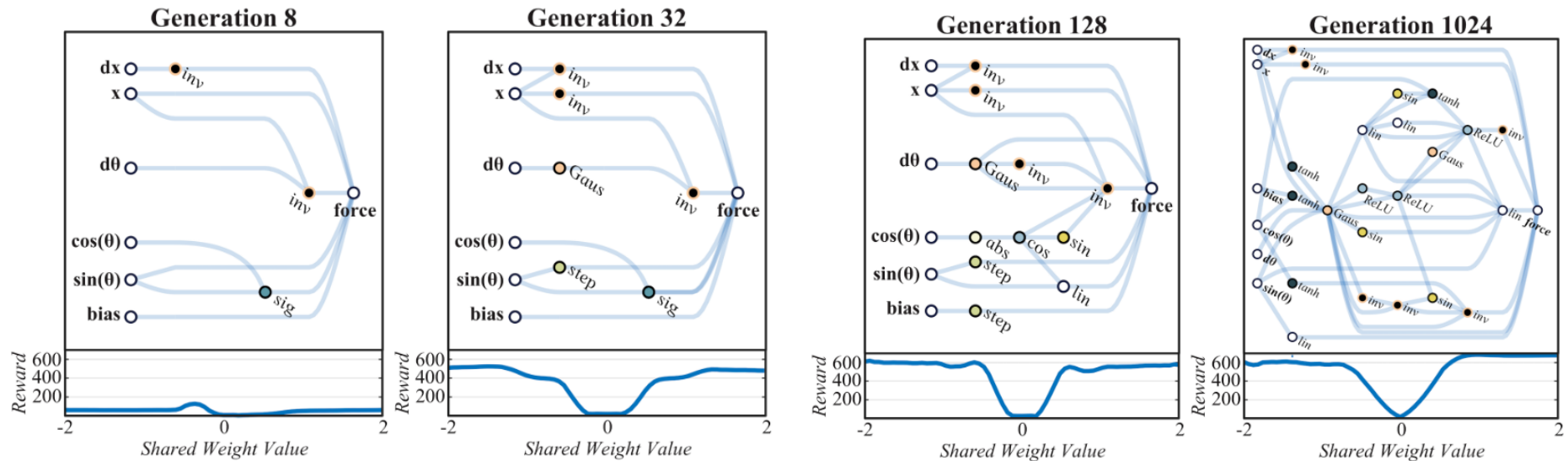
2, それぞれのネットワークに対して
共有の重みを設定, タスクを実行・評価

3, 結果の精度, ネットワークの複雑さから
各ネットワークをランク付け

4, 評価の高いネットワークを選び,
変化を加えて, 新たなNN群を作成

2.~4.を繰り返す

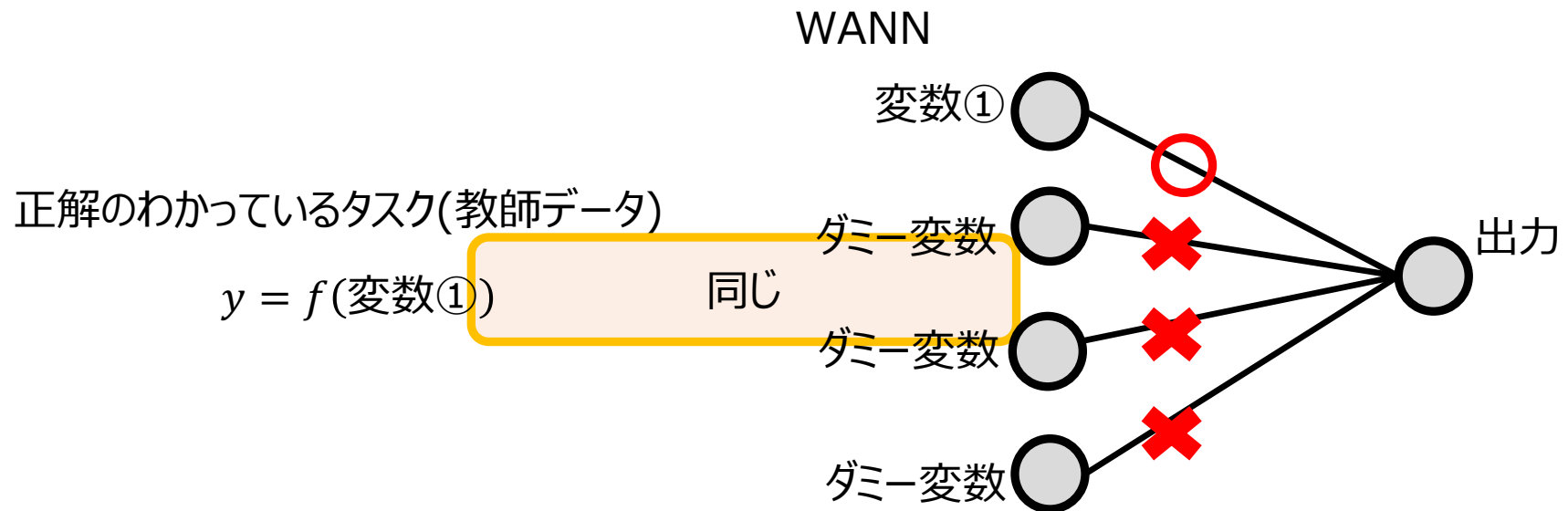
世代を経るにつれて構造が変化していく



シミュレーション

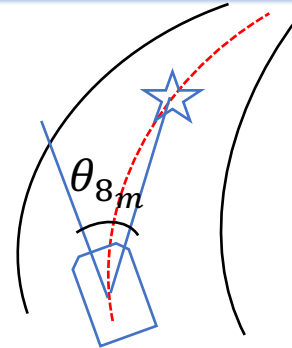
目的: WANNが本研究のモデルの最適化手法として, 有用であるかの検証

⇒ 正解のわかっている簡単なタスクにおいて,
WANN が適切な変数・構造を探索して 正解を出力することが可能か



タスクにおける教師データの取得

簡単な One-Point model を使って, シミュレーションコースを走行 $\dot{\delta} = K_{8m} \cdot \theta_{8m}$

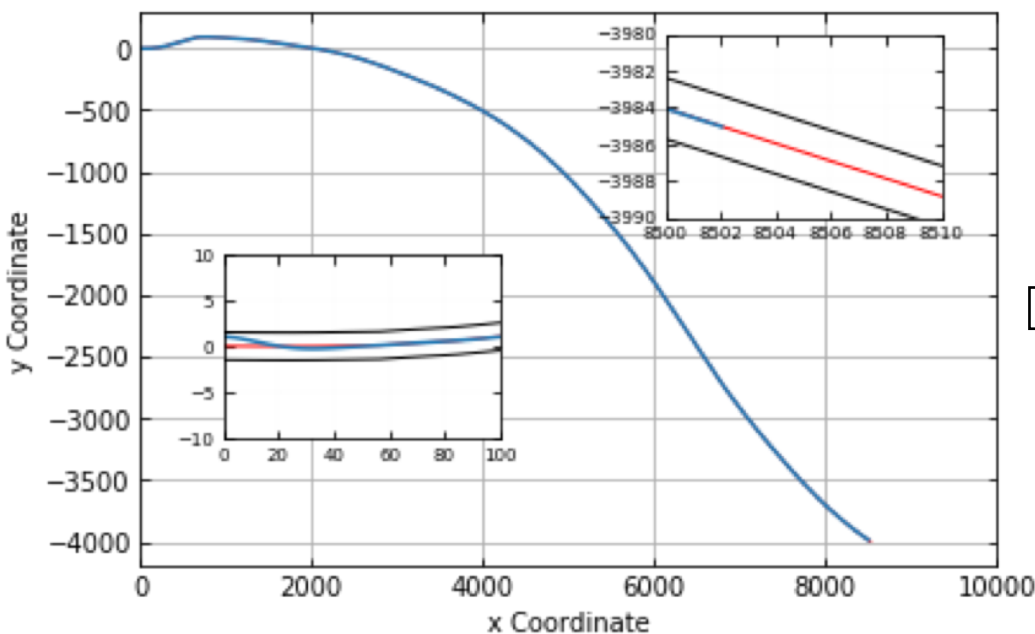


① 正解データ(教師データ)

各サンプリングタイムにおける 出力操舵角速度 $\dot{\delta}$, 使用した視野角の値 θ_{8m}

② WANN に対するダミーの変数

各サンプリングタイムにおける, 2m, 5m, 10m, 20m 先の視野角の値 $\theta_{2m}, \theta_{5m}, \theta_{10m}, \theta_{20m}$



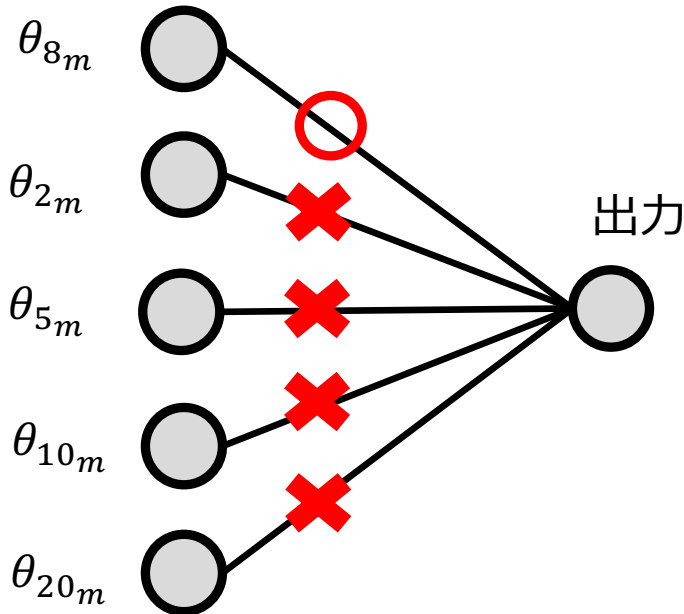
near_point1_rad_8m	near_point2_rad_2m	near_point3_rad_5m	near_point4_rad_10m	near_point5_rad_20m
0.000000	0.000000	0.000000	0.000000	0.000000
-0.126269	-0.522933	-0.201469	-0.101145	-0.052423
-0.112167	-0.506341	-0.186482	-0.087376	-0.039130
-0.097036	-0.472982	-0.170201	-0.073657	-0.026393
-0.082704	-0.436364	-0.151256	-0.060272	-0.014271
-0.068549	-0.407692	-0.133605	-0.046888	-0.003092
-0.054145	-0.365776	-0.114081	-0.034545	0.007023
-0.041025	-0.322788	-0.096346	-0.022975	0.015917
-0.028596	-0.287617	-0.078887	-0.011933	0.023754
-0.016972	-0.243543	-0.060754	-0.002061	0.030424
-0.005914	-0.200540	-0.044673	0.007225	0.036090
0.003975	-0.164716	-0.029507	0.015800	0.040737

シミュレーション結果①

条件①:正解1, ダミー4

WANN

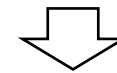
入力



WANN

$$\dot{\delta}_{WANN} = f(\theta_{2m}, \theta_{5m}, \theta_{8m}, \theta_{10m}, \theta_{20m})$$

$$\text{教師データ: } \dot{\delta} = K_{8m} \cdot \theta_{8m}$$



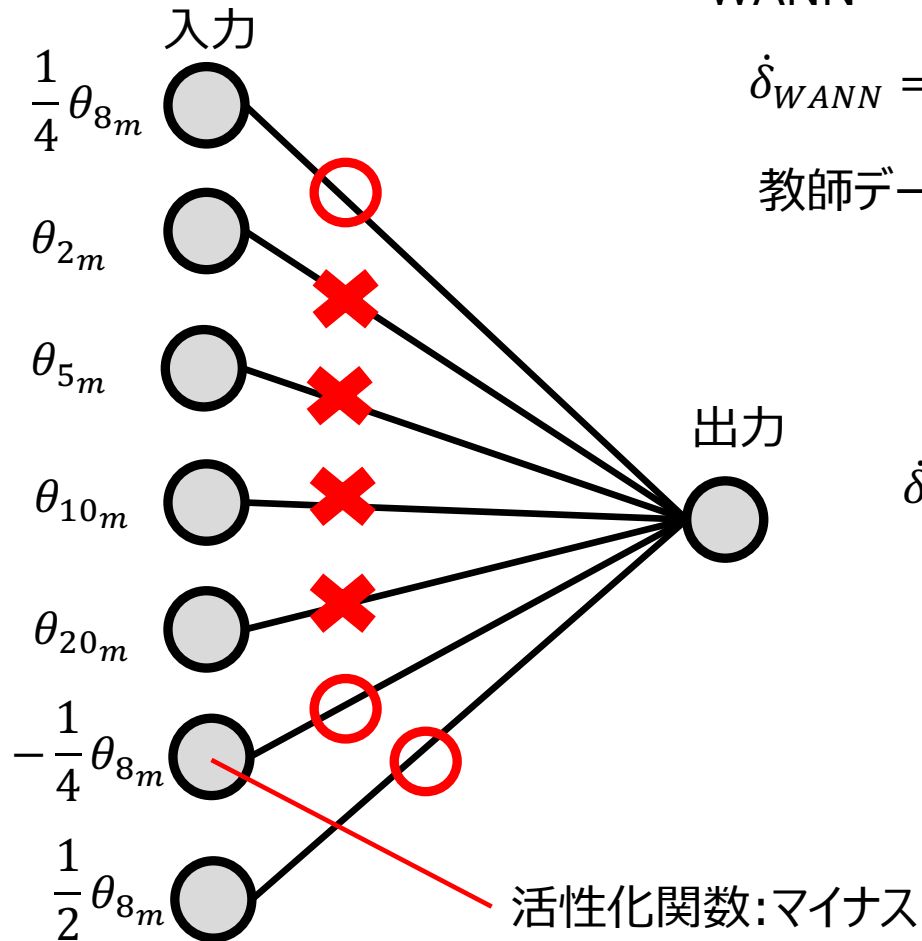
$$\dot{\delta}_{WANN} = w_i \cdot \theta_{8m} \quad w_i \in [-1, 0, 1]$$

単一の共有重みパラメータ: $w_i \in [-1, 0, 1]$,
最大世代数 : 10世代
集団サイズ : 100

WANN によって, One-Point model が出力

シミュレーション結果②

条件②:正解データの分割

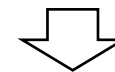


WANN

$$\dot{\delta}_{WANN} = f\left(\theta_{2m}, \theta_{5m}, \frac{1}{4}\theta_{8m}, -\frac{1}{4}\theta_{8m}, \frac{1}{2}\theta_{8m}, \theta_{10m}, \theta_{20m}\right)$$

$$\text{教師データ: } \dot{\delta} = K_{8m} \cdot \theta_{8m}$$

活性化関数「マイナス」の探索が必要



$$\dot{\delta}_{WANN} = w_i \cdot \left\{ \frac{1}{4}\theta_{8m} - \left(-\frac{1}{4}\theta_{8m} \right) + \frac{1}{2}\theta_{8m} \right\}$$

$$w_i \in [-1, 0, 1]$$

単一の共有重みパラメータ: $w_i \in [-1, 0, 1]$,
 最大世代数: 10世代
 集団サイズ: 100

WANN によって, One-Point model が出力

まとめと今後の展望

< 大目的 >

Weight Agnostic Neural Networksを用いた探索による
ドライバモデルの最適化

< 小目的 >

WANNがモデル構造の最適化手法として, 有用であるかの検証

< 結果 >

WANN によって, 必要な入力情報・活性化関数を探索
簡単なタスクの達成

➡ モデル構造の最適化手法として, 有用である可能性

< 今後の展望 >

WANNを用いてドライバモデルを探索・最適化

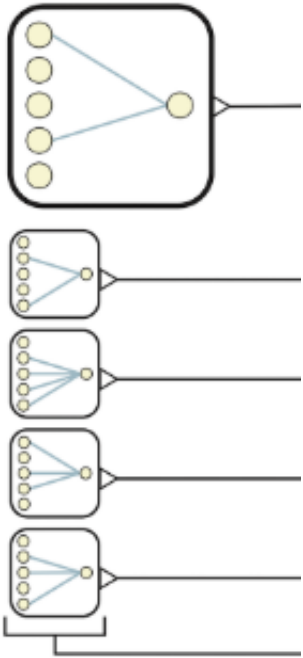
➡ ドライバモデル作成に関する先行研究から, 入力情報(変数)群を得る

付録

① Initialize

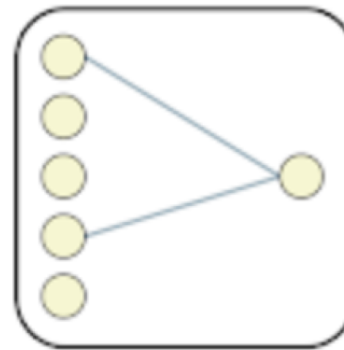
1.) Initialize

Create population of minimal networks.

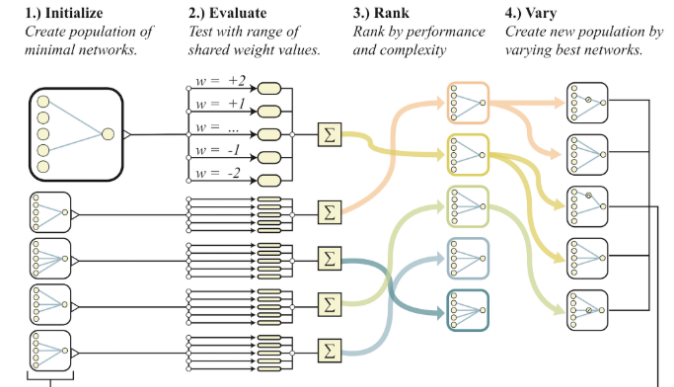


1, シンプルなネットワーク構造を複数個作成する

Minimal Network



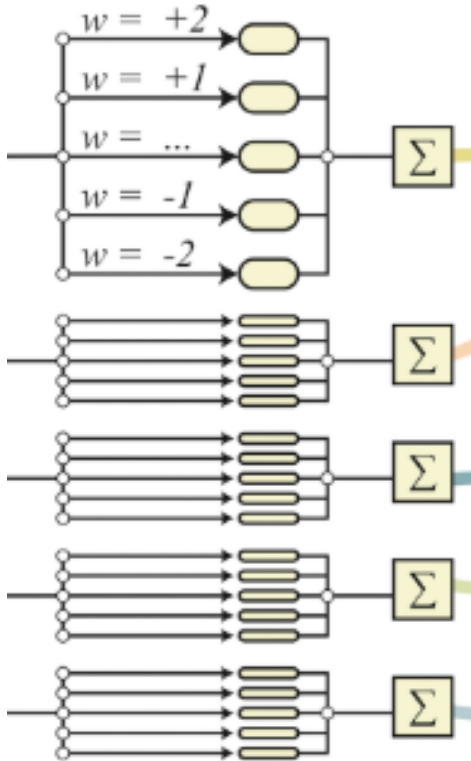
隠れ層も持たず, それぞれのノードも繋がっていない状態



② Evaluate

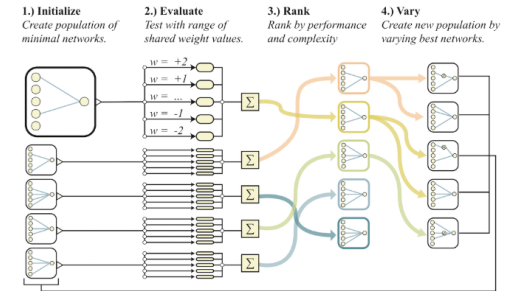
2.) Evaluate

Test with range of shared weight values.



2,各ネットワークの評価

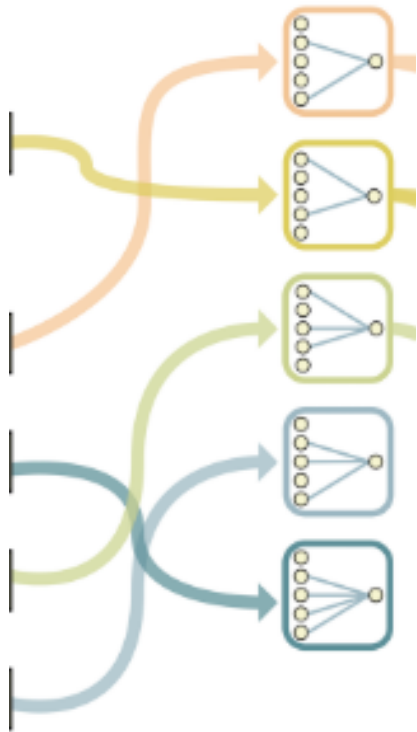
ネットワークに対して, 共有の重みを使用した際の平均値で評価
パラメータ依存をなくすため



③ Rank

3.) Rank

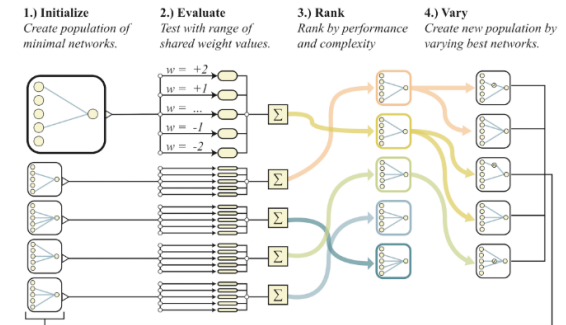
Rank by performance
and complexity



3, ネットワークのパフォーマンス, 複雑さから各ネットワークをランク付け

同様のパフォーマンスを持つ場合, よりシンプルなものを選ぶ

本研究: パフォーマンスは観測したドライバの操舵値との差
(より人間の操舵に近いネットワーク)

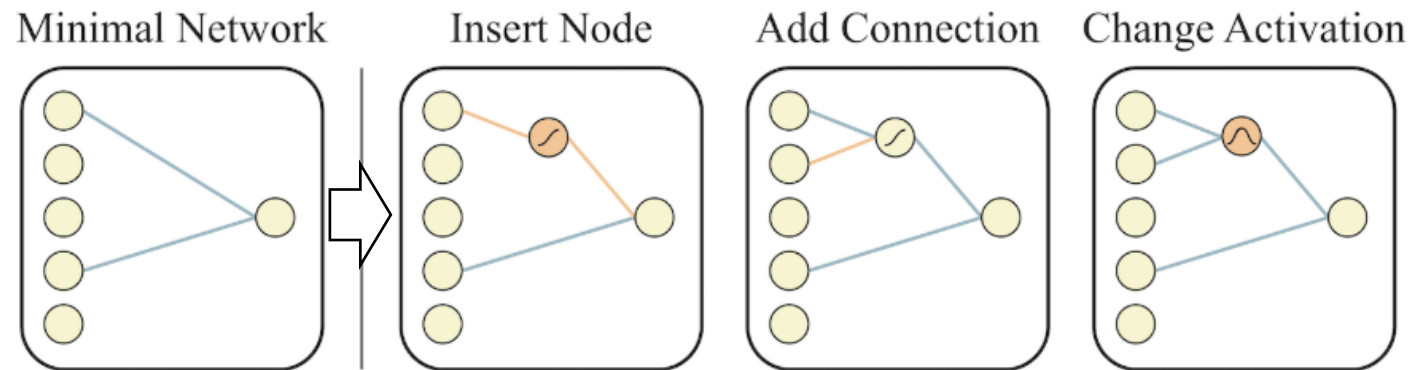


④ Vary

4.) Vary

Create new population by varying best networks.

4, 評価の高いネットワークを選び, 変化を加えて, 新たなネットワーク群を作成



変化1: ランダムに設定された活性化関数を持つノードの追加 (Insert Node)

変化2: 繋がっていなかったノードの間にコネクションを追加 (Add Connection)

変化3: 活性化関数をランダムに他のものに修正 (Change Activation)

[線形, ステップ, sin, cosine, Gaussian, tanh, sigmoid, 逆数, マイナス, ReLU]
変更可能