

# 機械学習を用いた運転支援システムの設計

2016SE055 宮本明征 2016SE097 吉川雅史

指導教員：野呂昌満

## 1 はじめに

自動運転に関する研究が盛んに行なわれており、第2水準までは実用化されはじめている [4]。自動運転における周囲環境の認識や運転の自動制御の実現に向けて機械学習が用いられており、これらによって今後第3水準以降が実現できると考えられる。

既存研究として、坂東ら [5] は、運行データからドライバーの正常な運転を学習し、その定義から外れた運転を漫然運転として検出、警告するシステムを提案している。勞ら [6] は、道路の認識を実現しており、地図情報を組み合わせることで自動制御を可能にしている。

機械学習の技術を適用した自動運転や運転支援機能の実現に向け、次の2つの問題を解決する必要があると考えた。

### ● 運転能力の個体差を考慮

警告時機が遅い場合、反応が遅いドライバは、判断や操作が遅れるので、危険回避が間に合わない。警告時機が早い場合、ドライバを惑わせてしまい、操作ミスに繋がる恐れがある。

すなわち、反応時間の個体差を考慮して適切な時機で警告を行なう必要がある。

### ● 実車への組み込み

経年変化するドライバの反応時間を予測できるように学習を繰り返す必要があるため、計算量の多い学習を車載 CPU 以外で行なう方式を考慮する必要がある。

本研究の目的は、反応時間の個体差を考慮して注意喚起を行なうシステムのソフトウェアアーキテクチャを設計することである。機械学習を用いて周囲環境を認識する研究が行われており [6]、これらを組み合わせた警告システムの再利用を視野に入れたアーキテクチャを設計する。

本研究では運転操作を入力、反応時間を出力として機械学習の学習モデルを定義し、アーキテクチャは、反応時間の特定と警告に関して独立して記述可能な構造にする。

オブジェクト指向を基本とし、アスペクト指向を適用することで、警告システムにおいて横断的コンサーンとなる以下のコンサーンをアスペクトとして分離する。

1. MVC コンサーン
2. リアルタイムコンサーン
3. 学習プロセスコンサーン

本研究の研究課題は以下のとおりである。

1. アーキテクチャ設計：計算量の多い学習を繰り返す行なうために、Cloud 内で学習を行なう構造を定義する。
2. 運転操作から反応時間を予測するための学習モデルおよびプロセスの定義：反応時間と運転操作の関係を学

習し、反応時間を予測可能な学習モデルを作成する。

3. ソフトウェアアーキテクチャおよび学習モデルとプロセスの妥当性の考察：シミュレータを用いてデータを計測し、学習モデルを実装し、アーキテクチャの妥当性を考察する。

## 2 背景技術

### 2.1 ニューラルネットワーク

ニューラルネットワークは単純な機能を持つユニットを連携させることで、高度な機能を実現する機械学習の技術である。学習モデルはハイパーパラメータで定義する。特に、ハイパーパラメータである損失関数、評価関数、最適化アルゴリズムを学習の目的に応じて決定する必要がある。

#### 1. 損失関数、評価関数

- 平均二乗誤差
- 平均絶対誤差

#### 2. 最適化アルゴリズム

- SGD
- Adam

ニューラルネットワークは、ユニット間の結合の仕方の違いにより、以下の3種類に分類できる。

- FCN(fully-connected-neural-network)：隣接する2つの層のすべてのユニットを網羅的に結合した構造。
- CNN(convolutional-neural-network)：ユニット間の結合を局所に限定した構造で、画像や音声などのデータに対して用いられる。
- RNN(recurrent-neural-network)：再帰構造をもつことで、過去の情報を取り入れた解析が可能となることから、時系列データなどの可変長のデータを扱うことができる。

## 3 アーキテクチャ設計

### 3.1 設計指針

本研究では以下を考慮してアーキテクチャを設計する。

#### ● 入出力の変更のための単純な構造

上記したコンサーンをアスペクトとして捉えるので、MVC コンサーンを MVC アスペクト、リアルタイムコンサーンをリアルタイムアスペクト、学習プロセスコンサーンを学習プロセスアスペクトとする。入出力の変更を可能にすることは、学習モデルをそのままにして、通知内容のみを別のものにするすることで、様々な通知システムに対応できるようになる。

#### ● 車載コンピュータの性能に応じたコンポーネントを

## 配置

近年、自動車は、制御システムの電子化が進み、システムのコントロールユニットである ECU を複数搭載している。一般に機械学習は高速処理をするために、並列処理に特化した GPU を用いる。一方で車載ソフトウェアの開発では、安全性、信頼性などに厳しい条件や、コストの制約からコントローラの処理能力やメモリ量などに制約がある。

本研究では学習はつねに行ない、予測はイベント発生直前に新しい予測器によって行なう必要があると考えられるので、車外のコンピュータで学習を行なう。Cloud 内に学習器を配置することで、1 車両のデータだけでなく、様々な車両で計測したデータを使った学習が可能という利点も考えられる。

## 3.2 アーキテクチャの静的構造

ニューラルネットワークは、学習器としても予測器としても稼働することから、オブジェクト指向においては、多相型として捉えることが適切である。

図 1 はアスペクト指向技術を適用したさいのオブジェクト指向による静的構造と、アスペクトとの関係を示したものである。センサ、データベース(以下、DB)などはデータを取得し保持するものなので、データリポジトリシステムと定義した。センサは外部環境の変化を検知、データベースはデータを保持する役割をもつ。機械学習に関するシステムを知的反応時間予測システムと定義した。ニューラルネットワークは予測処理を行なう予測器と学習処理を行なう学習器として分類した。

円で囲まれた部分にコンサーンが横断し、この横断的コンサーンに関する技術をアスペクトとして分離することを示している。アーキテクチャパターンとして MVC を適用し、入力、出力それぞれをアスペクトとして取り扱う。機械学習に関するコンポーネントを Model として捉える。

予測結果を通知する View アスペクト、入力処理部分を Controller アスペクト、リアルタイム制御に関連するものをリアルタイムアスペクト、学習状況に応じた学習方法に関するものを学習プロセスアスペクトとして定義した。

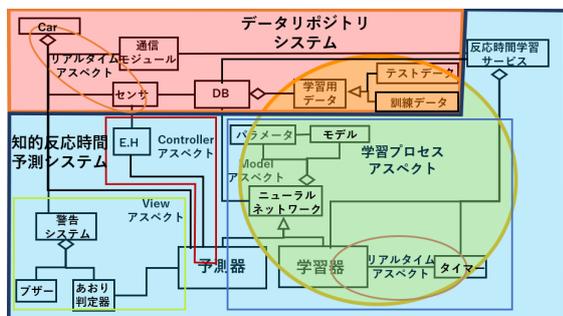


図 1 : 静的構造

## 3.3 コンポーネントの配置

図 2 は、コアコンサーンであるオブジェクト指向に基づいて設計したシステム全体でのコンポーネントを配置した図であり、車両と Cloud 内のどちらに配置するかをそれぞれ枠で囲んで示した。赤い枠で囲まれたオブジェクトは自動車に配置され、青い枠で囲まれたオブジェクトは Cloud 内に配置される。

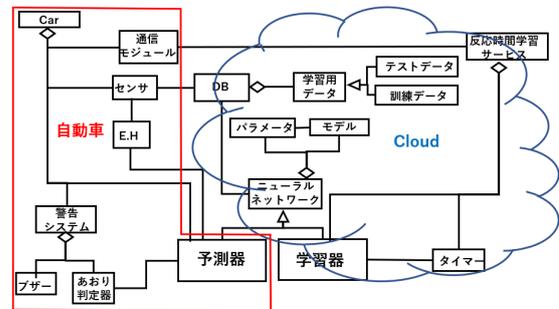


図 2 : コンポーネントの配置図

前述のように多相型として定義した学習器を Cloud 内に配置した。そして、自動車の通信モジュールから Cloud サービスにアクセスすることで実行可能とした。

Cloud 内に配置することで、高い処理能力を必要とする学習処理を車外で行なうことができる構造となった。

予測処理を行なう予測器を自動車に配置する。予測した反応時間を用いて警告時機を決定するので、予測結果をより早く取得するために予測処理を自動車内で局所的に行なうこととする。

## 4 学習モデルの設計

### 4.1 数理モデル

本研究では反応時間  $t$  を予測できる学習モデルを実装する。機械学習を適用する上で、説明変数の定義について以下に示す。

長塚 [3] の主張を参考に、反応時間に影響する要因として疲労度が大きく影響していると考え、疲労度からドライバの反応時間を予測できるようにする。しかし疲労度は数値として扱うことが難しく説明変数として用いることが困難である。上田ら [2] は運転疲労に伴い運転操作は変化すると述べていることから、運転中の誤操作の回数を説明変数とする。

以上より、この説明変数の値と反応時間の関係を機械学習の対象とする。

疲労時に発生する運転操作として以下が挙げられる。

- 急ブレーキ: 疲労による前方車両との車間距離の認知ミスや判断の遅れが原因
- 急加速: 疲労によるペダル操作能力の低下や交通状況の認識ミスによる慌てたペダル操作が原因
- ふらつき運転: 疲労やそれに伴う眠気による集中力の

低下が原因

ふらつき運転は定義が曖昧で、後述するシミュレータで計測する機能を実装することが難しいことから、本研究では計測が可能な急ブレーキと急加速の回数を説明変数とする。

以上より、本研究で扱う学習の数理モデルは以下の通りであり、これに従い学習モデルを定義し、実装を行なう。

$$t = f(x_1, x_2)$$

$t$ : ドライバの反応時間 (s)

$x_1$ : 急加速回数 (回/分)

$x_2$ : 急ブレーキ回数 (回/分)

## 4.2 学習モデルの定義

学習モデルの定義として、ニューラルネットワークの構造とパラメータ推定の方法を以下に示す。

### 4.2.1 ニューラルネットワークの構造

- 入力ユニット: 2 つ
- 出力ユニット: 1 つ
- 中間層の層数: 100 層
- 中間層の活性化関数: ReLU 関数

ReLU 関数は、正の領域において微分した値が 1 になることから勾配消失問題を防ぐことができる。

- 出力層の活性化関数: 恒等関数

中間層のユニット数については、学習モデルによって適切な値が異なるので、実験的に決定する。一般にユニット数を増やすことで複雑な入出力関係を表現することができるようになるが、増やしすぎると訓練データに適合しすぎて新しいデータに対する予測精度が下がる。

入力として用いる急加速回数、急ブレーキ回数のデータは時系列データではなく、さらに局所特徴を抽出可能であるかも不明であることから、汎用的な FCN を選択した。

### 4.2.2 パラメータ推定の方法

- 損失関数: 平均絶対誤差 (MAE)

数値予測が目的なので、正解値と予測値の誤差を表す関数を選択した。また用いるデータの中に外れ値がある可能性を考慮し、外れ値の影響が小さい平均絶対誤差を選択した。

- 学習アルゴリズム: adam

一般に用いられる確率的勾配降下法の中で学習率の調整問題が最も改善された adam を選択した。学習率はパラメータ更新の大きさを決定するもので、学習の収束に影響する。

- 評価関数: 平均二乗誤差 (MSE)

学習結果を評価する関数は損失関数とは別の関数であることが望ましいことから、平均二乗誤差を選択した。

## 4.3 実装

実装に必要なデータの計測方法と学習モデルの実装方法について以下に述べる。

### 4.3.1 データの計測方法

学習モデルの実装に必要な急加速回数、急ブレーキ回数、反応時間のデータをシミュレータを用いて計測する。実際に運転してあおり運転を計測することは困難であることから、あおり運転の状況を再現したシミュレータを作成する。実際の運転に近いシミュレーションを行なうために、ハンドルコントローラとペダルコントローラを導入した。

ドライバは運転操作を行ない、後方車両の接近時の警告を聞いたならハンドルによる回避操作 (車線変更) を実行する。運転開始から終了までの間、運転操作から急加速、急ブレーキの回数を計測する。後方車両はランダムに発生させ、その接近をきっかけに警告と反応時間の計測を開始し、ハンドル操作を行なうまでの時間を計測する。

シミュレータの実装は、GUI による直感的な操作が可能で扱いやすいことから、統合開発環境 Unity を用いた。シミュレータ内の車両の実装にあたり、Unity が作成し、提供する Standard Assets Package に実装されている StandardAssets/Vehicles/Car を利用した。

急加速と急ブレーキを計測する上でシミュレータ内の車両が実際の車両の感覚に近くなるように、被験者数人にブレいさせ、その意見をもとにアクセル、ブレーキ、ハンドルの動作の調整を行なった。

### 4.3.2 学習モデルの実装

実装にあたり、一般に用いられることから情報も多く、実装が容易になると考えられる Python を用いる。

本研究では、中間層のユニット数を実験的に決定する上でモデルの変更が必要となるので、モデルの構築、変更を容易に行えるようライブラリの Keras を用いた。Keras とは深層学習のライブラリで、ニューラルネットワークの構築、学習アルゴリズムの決定、学習、予測などのメソッドが定義されており、これを用いることで容易に実装できる。

## 4.4 テスト

本研究では、学習モデルの実装にあたり、中間層のユニット数を実験的に決定することとした。

ユニット数として 3, 4, 5, 6, 7, 8, 9, 10, 15, 30, 50, 100, 300, 500, 1000 個で実装、テストを行なった。学習の精度を確かめる指標として、テストデータに対する損失関数、評価関数の値をユニット数毎に導出した。

テストした結果、ユニット数を変更しても損失関数、評価関数の値に大きな違いは見られなかった。しかし、すべてのテストにおいて、これらの関数の値が減少し、ある値で収束したことから、学習は収束していると考えられる。また、テストデータに対する損失関数、評価関数の値が最小となったのは、ユニット数 6 であり、このときのテスト

損失は訓練損失よりも小さかったことから、汎化性能は高いと考えられる。ユニット数6のときの損失関数のグラフが図3である。

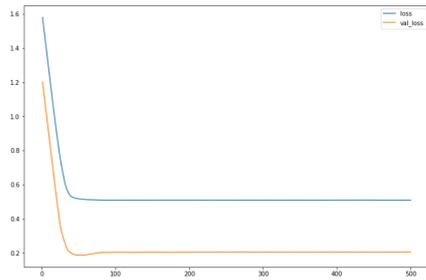


図3 損失関数 中間層100層 ユニット6個

以上より、本研究におけるニューラルネットワークの構造は表1のように決定した。

表1 ニューラルネットワークの構造

層	層の数	ユニット数	活性化関数
入力層	1	2	なし
中間層	100	6	ReLU 関数
出力層	1	1	恒等関数

## 5 考察

本研究では、説明変数として急加速回数と急ブレーキ回数の2つを用いたが、その他の運転操作を説明変数として用いることで予測精度が向上するのではないかと考えている。学習モデルのユニット数を変更して実験を行なった結果、学習結果や精度に大きな変化は見られなかった。このことからユニット数を増やしても、新たに抽出できる特徴量がなかったと考えられる。よって、説明変数を増やすことで、抽出できる特徴量が増え、ユニット数変更による精度の向上が可能になると考えられる。

反応時間を予測する学習モデルを Model として再利用することで、同様の適切な時機による警告が実現できると考える。例として、暴走車を検知するシステムが考えられる。交通状況を考慮し、適正速度が予測できれば可能だと考えた。

Automotive Software Engineering[1]では、自動運転ソフトウェアのための抽象度の高いアーキテクチャを提案している(図4)。このアーキテクチャのセットポイントジェ

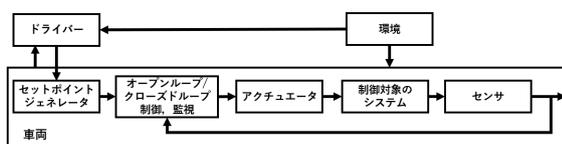


図4 Schaufeleら[1]が示したアーキテクチャ

ネレータはどのような自動制御を行ないたいかという値(セットポイント)を生成する。このセットポイントはドライバからの入力または自動で設定される。オープンループ/クローズドループ制御、監視とは自動制御するための基準変数を生成する。アクチュエータは基準変数から制御方法を決定する。

[1]ではこれらのコンポーネント毎に実現技術や開発手法、開発ツールについて説明している。

本研究では自動制御のためのソフトウェアの構造を示している。アーキテクチャに基づいて開発することで再利用、変更が容易になると考えられる。

我々のアーキテクチャは、[1]のアーキテクチャのそれぞれのコンポーネントにおけるソフトウェアコンポーネントの具体的な構造を示すものと考えられる。これらの内容を統合することで[1]で説明されている実現技術の柔軟な適用が可能になると考える。

## 6 おわりに

本研究では、運転支援として実現されている警告機能がドライバにとって適切な時機でない場合があることを課題とし、反応時間の個体差を考慮して注意喚起を行なうシステムのソフトウェアアーキテクチャを設計した。設計にあたり、警告と反応時間の特定に関して独立して記述可能な構造にするためにアスペクト指向計算を適用した。

本研究で選択した運転操作以外を用いて学習すると精度がどのように変化するかを考察する必要がある。本研究では、テスト結果からユニット数を増やしても精度が大きく変化しないことを確認したが、説明変数の種類が少ないことが原因であると考えられる。よって、説明変数を増やしたときに、学習結果や精度がどのように変化するかを検証する必要がある。

## 参考文献

- [1] J.Schauffele, T.Zurawka, Automotive Software Engineering, SAE International, 2005.
- [2] 上田誠, 近藤光男, 松本博次, 早川晴雄, 中田隆現, “追従走行実験における心理的・生理的反応に基づく運転疲労の定量化に関する研究,” 土木計画学研究, vol. 22, no. 4, pp. 805-813, 2005.
- [3] 長塚康弘, “事故傾性, 疲労および単調感と反応時間,” 人間工学, vol. 21, pp. 71-79, 1985.
- [4] 林世彬, 須田義大, 横溝英明, 込や浩資, 平山幸司, “高速バスの自動運転・陳列走行の社会実装に向けた考察,” 生産研究, vol. 71, no. 2, pp. 111-116, 2019.
- [5] 坂東誉司, 池田和司, 味間弘喜, 井上裕太, 柴田智弘, 人見謙太郎, 深谷直樹, “漫然運転検出のための適応的ブレーキ警報システムの開発,” デンソーテクニカルレビュー, vol. 15, pp. 82-88, 2010.
- [6] 勞世, 陳謙, “自動運転システムにおけるAI技術,” 計測と制御, vol. 57, no. 7, pp. 493-496, 2018.