

# 危険予測のためのコンテキスト指向ソフトウェアアーキテクチャに関する研究

## —LSTM と GRU を用いて—

2019SE039 村上友太 2019SE043 並川雄貴 2019SE055 曾我康平

指導教員：野呂昌満

### 1 はじめに

コンピュータの性能向上により、自動運転システムの実現において AI 技術が盛んに取り入れられるようになった。高速道路などの特定の条件下における自動運転は実用化されており、運転の主体が人からコンピュータシステムへと変わってきている。

AI 技術が実用化される一方、ソフトウェア工学の見地からの考察がなされている研究例は数少ない。商品としてソフトウェアの開発・販売をする以上、拡張性や保守性といった非機能特性を重視し、ソフトウェアアーキテクチャを中心とした開発をすることは重要である。原因結果関係が明らかではないが、その実用性は担保できる機械学習コンポーネントを扱うソフトウェアでは、その重要性がさらに高まる。一方、機械学習の応用では、コンセプトドリフト、データドリフトという問題がある。いずれも、モデルの性能が時間経過とともに劣化する問題であり、モデルを高性能に保つためには、ソフトウェア進化に対応したソフトウェアアーキテクチャを定義することが重要である。

危険予測システムにおいて、予測精度と予測速度は対立する非機能要求である。危険予測システムに関する研究では、ニューラルネットワーク（以下、NN と呼ぶ）が用いられることが多い [1]。NN(特に RNN) の予測精度を向上させるためには、パラメータ数を増やしたり、十分な長さの時系列データを記憶したりする必要がある。これらはどちらも予測速度を低下させる要因となる。危険予測システムは、本質的に矛盾が生じる要求も満たさなければならない。上のおり、予測精度と予測速度は本質的に矛盾が生じる要求である。予測精度が悪ければ衝突する危険性が高まる。他方、死角からの飛び出しなど、突発的に危険な状況では素早い検知が求められる。すなわち、予測精度と予測速度を両立させなければ、危険予測システムの実現は困難となる。

本研究の目的は、ソフトウェア進化に対応しつつ、自動運転における危険予測システムの対立要求を調整することである。対立要求とは、予測精度と予測速度という対立する非機能要求のことを指す。

研究目的を基に、研究課題を以下のとおりに定義する。

1. 対立要求を調整する危険予測アーキテクチャの定義
2. 定義したアーキテクチャのソフトウェア進化への対応
3. 対立要求の調整とソフトウェア進化(変更容易性)の観点からの妥当性検証

本研究では、複数の NN を協調させることで対立要求を調整する。複数の NN を協調させるために、危険予測システムの構造を表すアーキテクチャを定義する。本研究の主題は対立要求の調整だが、対立要求を調整するにはソフトウェア進化に対応する必要がある。妥当性の検証では対立要求の調整に加えて変更容易性についても議論する。変更容易性の議論は、対立要求を調整するための変更限定しておこなう。

### 2 関連研究

#### 2.1 三次元物体認識技術

三次元物体認識技術は、三次元データを入力として、物体の位置やカテゴリを推定する技術である。三次元データには、Point Cloud, Voxel などの様々な形式がある。

Voxel Transformer(VoTr)[2] は、Transformer に基づく、三次元物体認識プラットフォームとしての NN であり、Voxel を Transformer に適用させている。Transformer は注意機構を用いているので、パラメータの数を抑えつつ、離れた Voxel との関連を考慮できる。その結果、計算効率を維持しつつ、検出性能を向上させた。VoTr に基づく三次元物体認識のアーキテクチャとして、VoTr-SSD が提案された。

#### 2.2 ソフトウェアアーキテクチャ

水谷ら [3] は、複数種類の偽造画像を検出するために、複数の NN を協調させるアーキテクチャを定義した。水谷らは偽造の種類分類結果をコンテキストとし、偽造検知をする NN を動的に選択する協調方式を提案した。

加地ら [4] は、予測精度と予測速度を考慮した危険予測をおこなうために、二段階の機械学習を用いたソフトウェアアーキテクチャを提案した。加地らは、危険予測を二段階でおこなうことで変更容易性を保証し、周辺状況検知で予測速度を考慮した検出をおこなった。

### 3 課題解決へのアプローチ

我々は、既存のアーキテクチャを統合し、統合したアーキテクチャに多相性を持たせることで目的の達成を目指す。統合する既存のアーキテクチャは、加地らが提案した「二段階の危険予測システムのアーキテクチャ」と水谷らが提案した「複数の NN を協調するためのアーキテクチャ」である。加地らの提案したアーキテクチャにより変更容易性を保障し、水谷らの提案したアーキテクチャにより対立要求を調整する。対立要求の調整は、以下のポリシーに従っ

て危険予測に用いる NN を切り替えることで実現する。本研究では、予測精度の優れた NN として LSTM を用い、予測速度の優れた NN として GRU を用いる。

- ・ 通常時は、予測精度が優れた NN で危険予測をする
- ・ 衝突する可能性が高いときは、予測速度が優れた NN で危険予測をする

加えて、統合したアーキテクチャで変更が起こりやすいと考える箇所に多相性を持たせることで、さらにコンポーネントの変更を容易にする。以上により、以下の成果が得られる。

1. 予測精度と予測速度の重要度の変化への対応
2. コンセプトドリフトやデータドリフトへの対応

提案するアーキテクチャの妥当性は対立要求の調整と変更容易性の二つの観点からおこなう。対立要求の調整の観点では、提案するアーキテクチャに基づいてプロトタイプを作成し、そのプロトタイプの出力を用いることで定量的に検証する。プロトタイプの予測精度と予測速度を用いて比較をすることで、提案するアーキテクチャが目的を達成できたか考察する。変更容易性の観点では定性的な検証をおこなう。この検証では、提案するアーキテクチャがソフトウェア進化に対応してるか考察する。

#### 4 アーキテクチャ

提案する具象アーキテクチャを図 1 に示す。図 1 のアーキテクチャは、コンテキスト指向に基づく処理と並行処理とを統合して定義したものである。加地らが提案したアーキテクチャを参考にし、危険予測システムを周辺状況検知と危険予測の二段階に分割した。周辺状況検知には VoTr-SSD を用い、危険予測には LSTM と GRU を用いた。Risk\_Assessment や Sensor はコンポーネントの変更が起きやすいと考えたので、コンポジットパターンを利用し、多相性を持つ構造とした。

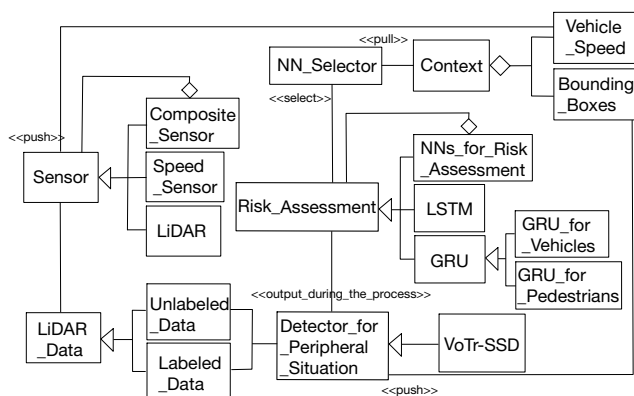


図 1 提案する具象アーキテクチャ

コンテキスト指向に基づく処理は、水谷らが提案したアーキテクチャを参考にした。コンテキストは図 1 の Context に相当する。Context は VoTr-SSD から送られる周辺状況の情報と Speed\_Sensor から送られる自車の速

度の情報を持っている。NN\_Selector が Context の情報から停止距離を計算することで、衝突する可能性が高いかどうか判断する。停止距離は、歩行者の位置、他車の位置、自車の位置で算出できるので、これらをコンテキストとする。NN\_Selector の判断結果によって、LSTM または GRU をインスタンス化する層を活性化することで、NN コンポーネントを動的に再配置する。

図 1 における並行処理コンポーネントは、Detector\_for\_Peripheral\_Situation, NN\_Selector, Risk\_Assessment であり、Context が持つ情報を共有資源としている。このとき、Context の情報に対する書き込みの処理と読み込みの処理を同時におこなわないように、相互排除が必要になる。これらのコンポーネントは、それぞれ以下の役割を果たしている。

Detector\_for\_Peripheral\_Situation: 周辺状況を検知をする

NN\_Selector: 危険予測に用いる NN を選択する

Risk\_Assessment: 危険予測をする

#### 5 実装

図 1 のアーキテクチャに基づき、プロトタイプを実装する。プロトタイプの実装では、限られた研究期間で妥当性を検証するために、コンテキスト指向プログラミングをおこなわない。代わりに、条件分岐を用いることで動的な選択を実現した。並行処理については、予測速度を向上させるために必要になるので実装した。図 1 の Context が持っている自車の速度は定数として扱う。これらをアーキテクチャに反映させたプロトタイプの構成を図 2 に示す。

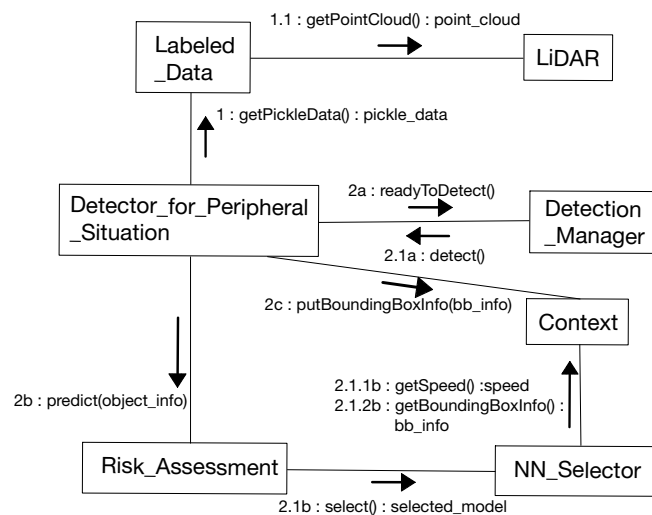


図 2 プロトタイプの構成

本研究の制約条件を以下に示す。道路の状態、センサの種類、危険予測の対象とその速度と大きさを変更しても、危険予測の精度や速度に影響は与えないと考える。すなわち、以下のとおりに条件を統一しても、提案するアーキテクチャに実用上の問題はないと考える。

1. 乾燥したアスファルトの一般道路
2. LiDAR センサを搭載（危険予測をおこなうのは自車の前方だけ）
3. 危険予測の対象とするのは歩行者と車両だけ
4. 歩行者や他車の速度、大きさは考慮しない

本研究では、Waymo Open Dataset[5][6] を使用して、プロトタイプの実装をおこなう。危険予測では、1 フレームごとの物体の位置の差異から危険を予測するので、時系列的に滑らかな動画像を必要とする。Waymo Open Dataset には、LiDAR から得られた点群データが 10Hz で収集されている。他方、アノテーションは、データセットの作成において最もコストのかかる作業である。周辺状況検知用 NN の学習時に、ラベル付けされたデータを使用することは、限られた期間でおこなう研究にとって利点となる。

## 6 考察

### 6.1 対立要求の調整

プロトタイプを実装した結果を以下に示す。危険予測に用いる NN の予測精度と予測速度を表 1 に示す。AUC(Area Under the Roc Curve) は予測精度を表す指標である。表 2 に、危険予測用 NN を選択し、その NN で危険予測をするまでの予測速度と選択結果の内訳を示す。表 1 と表 2 の予測速度は、予測を 1000 回おこなって得られた予測速度の平均値であり、入力データの整形にかかった時間を含んでいる。AUC と予測速度は、Waymo Open Dataset の検証用データセットを用いて求めた。

表 1 危険予測用 NN の予測精度と予測速度

	LSTM	GRU_for_Vehicles	GRU_for_Pedestrians
AUC	0.866	0.703	0.786
予測速度 (ms)	16.4	7.2	7.2

表 2 動的な切り替えをおこなったときの予測速度

自車の速度 (km/h)	10	20	30	40	50	60
予測速度 (ms)	17.6	10.7	8.6	10.1	10.6	10.1
LSTM の選択回数	992	334	13	0	0	0
GRU_for_Vehicles の選択回数	2	573	456	296	172	85
GRU_for_Pedestrians の選択回数	6	10	100	0	0	0
GRU_for_Vehicles と GRU_for_Pedestrians の選択回数	0	83	431	704	828	915

表 1 の AUC から、予測精度は LSTM, GRU\_for\_Pedestrians, GRU\_for\_Vehicles の順に優れていたことがわかる。予測速度は GRU\_for\_Pedestrians と GRU\_for\_Vehicles, LSTM の順に優れていたことがわかる。表 1 の予測速度と表 2 の予測速度とを比較すると、自車の速度が 10km/h であるとき以外は、動的な切り替えをおこなう場合の予測速度が LSTM だけ用いる場合の予測速度よりも優れていることがわかる。表 2 の選択回数を自車の速度ごとに比較

すると、自車の速度が上がるにつれて、衝突する可能性が高いと判断して GRU を多く選択していることがわかる。これらのことから、予測精度の優れた LSTM と予測速度の優れた GRU を動的に切り替えることによって、対立要求を調整できたと考える。

### 6.2 変更容易性

提案する具象アーキテクチャの変更容易性について、以下の変更例をあげて議論する。変更例は危険予測システムの主要な機能における変更を基に考えた。主要な機能は、危険予測をする機能と周辺状況を検知する機能である。

変更例 1: 危険予測用 NN を予測精度や予測速度が優れた NN に変更する

変更例 2: コンテキストが持つ情報を変更する

変更例 3: 周辺状況検知用 NN を予測精度や予測速度が優れた NN に変更する

変更例 1 では、LSTM や GRU を予測精度や予測速度が優れた NN コンポーネントに置き換えるだけでよいと考える。LSTM や GRU が同じ役割を持つ別の NN に置き換わっても、コンポーネント間の関連は変化しない。

変更例 2 では、コンポーネントの置き換えは起きないが、新たにコンポーネントが追加されると考える。周辺物体の速度もコンテキストにする場合は、周辺物体の速度を検知するためのセンサを Sensor と is-a 関係で追加する。その際、Context に周辺物体の速度を管理するコンポーネントを has-a 関係で追加する必要がある。

変更例 3 では、変更例 1 と同様に置き換えるだけで変更できると考える。ただし、入力するセンサからの情報も変わると、コンポーネントを追加する必要がある。予測精度の向上を図るために、センサとしてカメラを追加する場合、カメラと Sensor とを is-a 関係で追加する必要がある。Sensor はコンポジットパターンを用いたコンポーネントなので、カメラを加えても、情報を得るメソッドを起動すれば、カメラの RGB データと LiDAR の点群データをまとめて得られる。

変更例 1 には多相性とコンテキスト指向で対応でき、変更例 2 と変更例 3 には多相性で対応できることを示した。すなわち、提案するアーキテクチャは多相性とコンテキスト指向に基づく動的な再構成により、変更容易性を保障していると考えられる。

危険予測システムの振舞いは、NN の学習時と運用時で異なる。運用時の振舞いは 4 章や 5 章で述べたとおり、提案するアーキテクチャで対応できる。学習時は、周辺状況検知用 NN や危険予測用 NN が図 1 の Labeled\_Data から学習用のデータセットを得ることで学習できる。危険予測用 NN は、周辺状況検知用の NN から得られる特徴量と Labeled\_Data から送られたラベルを得ることで学習できる。これらのことから、提案するアーキテクチャは学習時と運用時の振る舞いに対応できるので、変更容易性を保障

していると考える。

NN\_Selector を NN にすると、提案するアーキテクチャの Sensor, LiDAR\_Data, Context を除くすべてのコンポーネントは NN になる。これらの NN に対してオンライン学習をさせると、つねに進化し続ける危険予測システムにすることができる。例えば、事故の起こりやすい道路の幾何構造や人口密度もコンテキストとして考えると、走行するたびに、その車にあった危険予測システムに進化する。この場合、オンライン学習をさせることで、よく走る道路の幾何構造や人口密度を逐次的に学習させることができる。NN\_Selector もそれに合わせて学習をするので、危険の定義をその車が走行する地域に適応して変えていくことができると考える。

### 6.3 並行処理

以上考察をおこなったが、並行処理についても記述する。当初、我々は、並行処理記述は重要でないと考えていたが、研究が進むにつれて、その重要性を確認したので、ここにそれを記す。

4章で述べた並行処理コンポーネントを基に危険予測システムの処理を分割することで、処理の記述の煩雑さが軽減され、実行効率が向上すると考える。危険予測システムのアクティビティ図を図3に示す。逐次処理をおこなう場合、アクティビティ図は図4となる。図3の蛍光色で塗られた範囲は、それぞれ4章で述べた役割に対する処理を記述したものである。図3では、共有資源である Context に対する相互排除をするために Hoare のモニタを用いている。

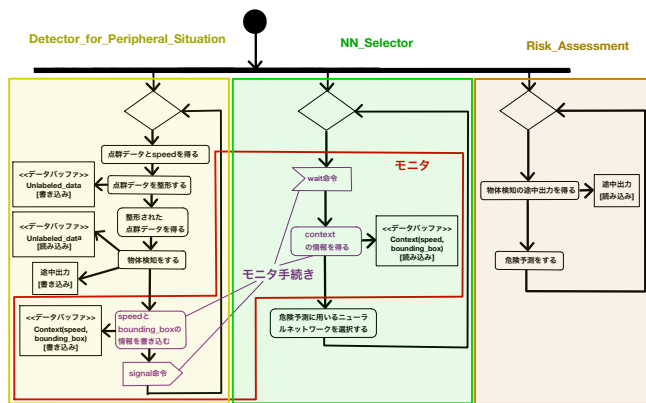


図3 危険予測システムのアクティビティ図 (並行処理)

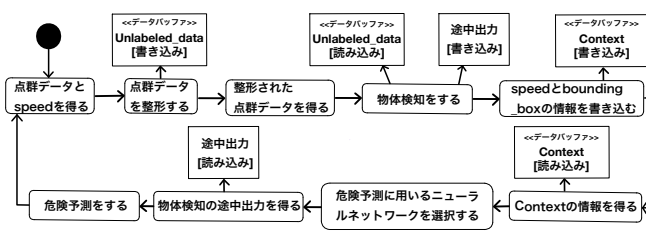


図4 危険予測システムのアクティビティ図 (逐次処理)

図3と図4を比較すると、複数の逐次プロセスに分割した記述の方が一括の逐次処理の記述よりも短く、独立した役割に沿って処理内容を記述できている。このことから、並行処理は逐次処理より処理の記述の煩雑さを減らせると考える。これを仕様と実装の問題と考えれば、順路式などの抽象度の高い記述を仕様とし、それをどのように実装するかという考察に発展すると考える。

## 7 おわりに

本研究では、危険予測システムにおける対立要求を調整するために、危険予測に用いる NN をコンテキストに応じて切り替えるシステムのアーキテクチャを提案した。提案するアーキテクチャの妥当性を検証するために、対立要求の調整について定量的に議論し、変更容易性を定性的に議論した。対立要求の調整の観点からの議論では、実験結果により、対立要求を調整できていることが示せた。変更容易性の観点からの議論では、提案したアーキテクチャがソフトウェア進化に対応できることを示した。しかし、オンライン学習をさせることで、つねに進化し続ける危険予測システムにすることができるので、提案したアーキテクチャには改良の余地が残されている。

今後の課題は以下のとおりである。

1. データセット数を増やして、NN の再学習
  2. 周辺状況検知用 NN の変更
  3. シミュレータを用いた、アーキテクチャの実用性の検証
  4. 変更容易性の観点からのアーキテクチャの定量的な議論
2. の周辺状況検知用 NN の変更では、周辺状況検知の精度を向上させることで、危険予測用 NN の予測精度を向上させる。4. の変更容易性の定量的な検証では、6章で議論した変更例における、コンポーネントの変更数、変更したコードの行数、変更に必要な時間などを示す必要がある。

## 参考文献

- [1] S.Grigorescu, et al, "A Survey of Deep Learning Techniques for Autonomous Driving," *Journal of Field Robotics*, 2020.
- [2] J.Mao, et al, "Voxel Transformer for 3d Object Detection," *Proc. IEEE/CVF ICCV*, 2021.
- [3] A.Mizutani, et al, "Design of Software Architecture for Neural Network Cooperation: Case of Forgery Detection," *Proc. 28th APSEC*, 2021.
- [4] 加地智也, et al, "2段階機械学習を用いた危険予測システムアーキテクチャの提案," 研報 SE, 2022.
- [5] Waymo Open Dataset: *An Autonomous Driving Dataset*, <https://www.waymo.com/open>, 2019.
- [6] P.Sun, et al, "Scalability in Perception for Autonomous Driving: Waymo Open Dataset," *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2020.