

Neural DTS に対する型検査アルゴリズムの実装の試み

飯沼瑞稀（指導教員：戸次大介）

1 はじめに

計算意味論における推論システムには、型論理的意味論などの論理学による記号的推論と、ニューラルネットによるソフトな推論がある。記号的推論の枠組みの一つとして、依存型意味論 (Dependent Type Semantics, DTS) があり、DTS により一般量子化や照応といった複雑な言語現象を解釈できるようになりつつある。一方、ニューラルネットは類似度計算、分類、文生成、翻訳などのタスクを、実テキストのサイズや多様性に関係なく高速かつ頑健に実行できるという利点がある。

Neural DTS [1][2] は、深層ニューラルネットワークを DTS に埋め込むことで、これら 2 つの推論を組み合わせるというアイデアのもと提案された枠組みである。Neural DTS の実現により、真偽判定や意思決定支援などの場面で、人間が説明するのと同様に判断の過程を説明できるような AI システムを構築できる。

本研究では Neural DTS の実装に向け、haskell での型検査アルゴリズムと深層ニューラルネットの融合例として、Neural DTS の型検査アルゴリズムの実装について取り組んだ。このアルゴリズムを応用することで、述語と名前のみから成る単純な関係が成り立つかどうかを、ニューラル分類器により判定できる。そして、WordNet のデータベース上で、与えられた関係が成り立つかどうかを予測する実験を行った。

2 提案手法

ニューラル分類器をパラメータとする型検査アルゴリズムを定義し、本研究の型検査アルゴリズムに埋め込まれる多層パーセプトロン (MLP) を `hasktorch`[3] により実装する。これらを組み合わせることで、MLP による分類器を呼び出すことのできる、NeuralDTS の型検査アルゴリズムを定義する。

2.1 型検査アルゴリズム

ニューラル分類器をパラメータとする型検査アルゴリズムを定義するために、Löh ら [4] が Haskell によって実装した依存型理論の型検査アルゴリズムを、以下のように改訂する。ここでは、実際の実装に合わせて 2 項述語で説明しているが、任意の n 項述語を取ることもできる。

まず、DTS の語彙に含まれる述語 P_i および 2 つの名前 e_j, e_k から成る型 $P_i(e_j, e_k)$ へとアルゴリズムを拡張する。例えば、述語が `has_part_of` で名前が `cat, tail` のとき、得られる命題は `has_part_of(cat, tail)` となり、これは「猫に尻尾がある」という命題に対応することになる。命題 $P_i(e_j, e_k)$ についての型検査では、文脈 Γ および命題 $P_i(e_j, e_k)$ が与えられたとき、 Γ のもとで証拠 w_{ijk} が $P_i(e_j, e_k)$ の証明であるかどうか（つまり、 Γ のもとで w_{ijk} が型 $P_i(e_j, e_k)$ をもつかどうか）が判定される。ただし、 w_{ijk} は命題 $P_i(e_j, e_k)$ のために用意する定項である。以下では、 Γ のもとで証拠 w_{ijk} が $P_i(e_j, e_k)$ の証明であるかどうかという問いを $\Gamma \vdash w_{ijk} : P_i(e_j, e_k) ?$ というように表記する。

また、ニューラル分類器によって述語を置き換えると

いう Neural DTS の基本構想に基づき、命題 $P_i(e_j, e_k)$ についての型検査は、述語 P_i と名前 e_j, e_k をそれぞれニューラル分類器に入力した結果を用いて行う。これを可能にするため、元の型検査関数 `typeChk` にニューラル分類器をパラメータとして与える。

```
typeChk :: (Tensor -> Tensor -> Tensor
           -> Tensor) -> Int -> Context
           -> TermChk -> Type -> Result ()
```

結果となる出力が閾値を超えれば名前 e_j, e_k は述語 P_i をみたすといえるので、命題 $P_i(e_j, e_k)$ は証拠 w_{ijk} をもつとみなし、 $\Gamma \vdash w_{ijk} : P_i(e_j, e_k) ?$ という問いには YES と答える。閾値を超えなければ e_j, e_k は P_i をみたさないということであるから、命題 $P_i(e_j, e_k)$ はそもそも証明をもたない。そのためこの問いには NO と答える。

以上のように、本研究が提案する Neural DTS の型検査では、その途中でニューラル分類器が呼び出される。Löh ら [4] の型検査アルゴリズムを命題 $P_i(e_j, e_k)$ のケースまで拡張しつつ、型検査関数 `typeChk` にニューラル分類器のパラメータを加えることで、MLP を埋め込める型検査アルゴリズムを定義した。

2.2 MLP

埋め込み層、隠れ層、出力層を持つ、単純な MLP を実装した。実装には、`hasktorch` の Github リポジトリにある MLP の実装を参考にしている [3]。活性化関数にはシグモイド関数、損失関数にバイナリ交差エントロピー誤差を用いた。

2.3 型検査アルゴリズムへの MLP の埋め込み

§ 2.1 で定義した型検査関数 `typeChk` に、§ 2.2 で構成した MLP を埋め込むことで、目標となる型検査アルゴリズムを与える関数 `neuralTC` を定義する。この関数の型は以下ようになる：

```
neuralTC :: Context -> TermChk ->
           Type -> IO ()
```

関数 `neuralTC` は、文脈 Γ 、証明項 t 、型 A が与えられると、 Γ のもとで t が A を型としてもつかどうかを判定する。この点は、依存型理論に対する通常の型検査アルゴリズムと変わらないが、型検査の途中でニューラル分類器を呼び出して、その出力を用いて型検査 $\Gamma \vdash w_{ijk} : P_i(e_j, e_k) ?$ を判定する点で通常のものとは異なる。そしてこれらの入力 `neuralTC` に与えられると、データセットから学習済みのモデルを用いて型 `Tensor -> Tensor -> Tensor -> Tensor` をもつニューラル分類器を構成し、`typeChk` に入力とニューラル分類器を与えて、型検査が行われる。

3 実験

関数 `neuralTC` および `WordNet`[5] のデータベースを用いて、2 つの entity の間にある関係が成立するかどうかを、それ以外のデータから予測する実験を行った。データセットとして、`WordNet` から抽出した 40,943 個の entity と 18 個の relation に関する 141,442 個の

学習用関係トリプレット, 5,000 個のテスト用関係トリプレット, 5,000 個の検証用関係トリプレットを用いた. このデータセットは [6] において作成されたものである¹. 例えば, (gas jet, hypernym, burner) などのトリプレットが学習データに含まれており, これらを正解データとして使用した. 不正解データとして, 正解データと同じ数だけ, 正解データに含まれない関係トリプレットをランダムに作成した. 実験として, 作成した neuralTC がトリプレット上の関係を正しく予測できるかという 2 値分類を行った. typeChk では,

1. スコアが閾値以上で, かつ型検査が正常な場合: 2 つの entity が関係を満たしている
2. スコアが閾値未満で, かつ型検査が正常な場合: 関係を満たしていない
3. 型の不整合

の 3 種の返り値があり, このうち 1. を正解の予測データ, 2. を不正解の予測データとして扱った. 学習反復数 100, 2 値分類の閾値が 0.7 のとき, accuracy = 0.7620, precision = 0.7629, recall = 0.7602, f1 = 0.7616 となった. 学習反復数を増やしたり, 学習パラメータを最適化させたりすることで, より精度は上がると考えられる.

4 先行研究

Type theory with records (TTR) は, DTS と同じく依存型理論による自然言語意味論の枠組みである. Cooper [7] は, TTR に属する型からニューラルネット上の出来事 (events) の型への写像を与え, 外界の出来事や対象を分類するためにエージェントが学習するような型を TTR が提供できることを示した. その一方で, こうしたニューラルネット上の出来事の型とニューラル分類器の接続は示唆にとどまっており, 本研究が扱うような学習済み分類器を TTR へ埋め込むことは行われていない.

Larsson [8] は, 知覚データに関する学習可能な分類器が埋め込まれた TTR を提案した. ここでの分類は DTS における命題 $P_i(e_j)$ における分類に対応しており, このバージョンの TTR は Neural DTS と共通点を多くもつ. しかし, [8] の目的は, こうした TTR の実装というよりはその自然言語の意味論への応用であり, 型検査アルゴリズムをどう定式化するかという問題については論じられていない.

DTS に対しては, Bekki and Sato[9] が決定可能な依存型理論の断片に対する型推論・検査アルゴリズムの規則群を定式化し, Haskell によって型検査アルゴリズムを実装した.

5 おわりに

本研究ではまず, Neural DTS に対する型検査アルゴリズムを定式化した. 次に, このアルゴリズムの中で呼び出されるニューラル分類器となる MLP を実装した. 今後は, まず現在の単純な MLP ではなく, より実用的なニューラルネットワークと組み合わせることで分類精度の向上を図る. また, 現在は aRb の 2 項関係として, 2 つの entity と 1 つの関係に対して分類

しているが, aRb and ($not\ bRa$) などの複合的な論理式に対応させることも目指す. これにより, より複雑な文構造にも対応させることにつながっていくと考える. そして最終的には, parser と接続し, DTS の基本述語の集合を入力としてニューラル分類器の学習を行えるようにすることで, Neural DTS の実装を完成させ, 用意したデータセットではなく, 文書の集合から学習, 分類が行えるように改善する.

謝辞

本研究は, JST CREST JPMJCR20D2 の支援を受けたものである.

参考文献

- [1] Daisuke Bekki, Ribeka Tanaka, and Yuta Takahashi. Learning knowledge with Neural DTS. In *Proceedings of the 3rd Natural Logic Meets Machine Learning Workshop (NALOMA III)*, pp. 17–25, Galway, Ireland, August 2022. Association for Computational Linguistics.
- [2] Daisuke Bekki, Ribeka Tanaka, and Yuta Takahashi. Integrating Deep Neural Networks with Dependent Type Semantics. In R. Loukanova, P. L. Lumsdaine, and R. Muskens, editors, *Logic and Algorithms in Computational Linguistics 2021 (LACompLing2021)*. Springer Cham, to appear.
- [3] hasktorch: MLP implementation of logical XOR. Accessed Jan. 11, 2023. <https://github.com/hasktorch/hasktorch/tree/master/examples/xor-mlp>.
- [4] Andres Löh, Conor McBride, and Wouter Swierstra. A tutorial implementation of a dependently typed lambda calculus. *Fundam. Informaticae*, Vol. 102, No. 2, pp. 177–207, 2010.
- [5] Wordnet. Accessed Jan. 11, 2023. <https://wordnet.princeton.edu>.
- [6] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data - Application to word-sense disambiguation. *Mach. Learn.*, Vol. 94, No. 2, pp. 233–259, 2014.
- [7] Robin Cooper. Representing types as neural events. *J. Log. Lang. Inf.*, Vol. 28, No. 2, pp. 131–155, 2019.
- [8] Staffan Larsson. Discrete and probabilistic classifier-based semantics. In *Proceedings of the Probability and Meaning Conference (PaM 2020)*, pp. 62–68, Gothenburg, June 2020. Association for Computational Linguistics.
- [9] Daisuke Bekki and Miho Satoh. Calculating projections via type checking. In *Proceedings of TY-TLES*, 2015.

¹[6] は次の French ANR grant の支援を受けている: <https://www.hds.utc.fr/everest/doku.php?id=en:smemlj12>