（P05）深層学習を用いたソースコード分類手法の比較調査

藤原裕士[1] 崔 恩瀞[2] 吉田則裕[3] 井上克郎[1]
[1]大阪大学 [2]京都工芸繊維大学 [3]名古屋大学

研究背景：ソースコード分類

- 既存ソースコードがあらかじめ分類されている機能クラスに，入力されたソースコードを自動で分類する技術
- 既存ソースコードの検索や再利用の効率化に貢献

研究背景：広く利用されているニューラルネットワーク

- 順伝播型ニューラルネットワーク（FNN）
  - ループ構造がない標準的なネットワーク
  - ソースコードのベクトルを学習させることが可能[1][2]
- 再帰型ニューラルネットワーク（LSTM）
  - 入力の値だけでなく入力の順番も出力に影響するネットワーク
  - トークン列などを学習させることが可能[3][4]
- グラフ畳み込みネットワーク（GCN）
  - グラフの特徴抽出が可能なネットワーク
  - 抽象構文木などを学習させることが可能[5]

様々なニューラルネットワークを組み合わせたり，
複数のソースコード表現を学習させる場合もある[3]~[5]

研究動機

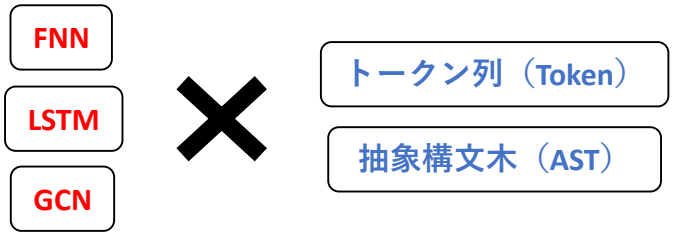
どのニューラルネットワークやソースコード表現の組み合わせが高精度なソースコード分類の実現に有効か明らかでない

- 無駄な学習に計算資源を利用するのは良くない
- 分類精度に良い影響を与えるニューラルネットワークやソースコード表現を学習に利用すべき

調査概要：深層学習を用いたソースコード分類手法の比較

RQ：高精度なソースコード分類を実現できるニューラルネットワークとソースコード表現の組み合わせは何か

ニューラルネットワークとソースコード表現を組み合わせ，6種類のソースコード分類手法を作成し，分類精度を比較



（P05）深層学習を用いたソースコード分類手法の比較調査

データセット：BigCloneBench[6]

- 各メソッドが果たす機能に基づき，43種類の機能クラスにメソッドが分類されているデータセット
- 本研究では 学習データ：評価データ＝8：2

評価尺度：Top-k

メソッドに対して機能クラス毎の分類確率を計算し，
その確率が高いクラス順にランキングにしたとき，
正解クラスがk位以内に含まれる割合

LSTMを用いた手法は
平均的に高い分類精度

調査結果

分類手法	Top-1	Top-3	Top-5	Top-10
FNN+Token	0.575	0.766	0.830	0.911
FNN+AST	0.644	0.803	0.853	0.922
LSTM+Token	**0.943**	**0.980**	**0.985**	0.991
LSTM+AST	0.939	0.977	0.981	0.991
GCN+Token	0.772	0.927	0.967	0.989
GCN+AST	0.803	0.948	0.972	**0.993**

FNN, GCNはASTと相性が良い
LSTMはトークン列と相性が良い

RQへの回答：LSTMとトークン列の組み合わせが，最も高い精度のソースコード分類を実現できる
LSTMとASTの組み合わせやGCNとASTの組み合わせも，比較的高い精度のソースコード分類を実現できる

（P05）深層学習を用いたソースコード分類手法の比較調査

今後の課題

- 他に比較対象として適当なニューラルネットワークやソースコード表現があれば比較を行いたい
  - 他にソースコード分類に適用できそうなニューラルネットワークがあるか
  - 生成にコンパイル不要なソースコード表現があるか（BigCloneBenchのメソッドのコンパイルが難しいため）
- 他のデータセットを用いて本調査結果と同様の傾向があるか調べたい
  - LSTM+TokenがBigCloneBenchに適合しているだけの可能性があるため
  - コンパイル可能なデータセットを用意できれば，コンパイルが必要なソースコード表現について調査できる

謝辞

本研究は JSPS 科研費 18H04094, JP19K20240, JP20K11745 の助成を受けたものです。

参考文献

[1] V. Saini et al., “Oreo: Detection of clones in the twilight zone”, Proc. ESEC/FSE 2018.
[2] 藤原ら，“順伝播型ニューラルネットワークを用いた類似コードブロック検索の試み”,SES2018.
[3] J. Zhang et al., “A Novel Neural Source Code Representation based on Abstrace Syntax Tree”, Proc. ICSE 2019.
[4] M. White et al., “Deep learning code fragments for code clone detection”, Proc. ASE 2016.
[5] W. Hua et al., “FCCA: Hybrid Code Representation for Functional Clone Detection Using Attention Networks”, IEEE Trans. Rel. pp.1-15, 2020.
[6] J. Svajlenko et al., “Towards a big data curated benchmark of inter-project code clones”, Proc. ICSME 2014.