深層学習を用いた自動プログラミング手法の評価

情報科学科　中村 博輝　　指導教員：粕谷 英人

1　はじめに

現在，人間が理解できる完全なソースコードを，実用レベルで自動的に生成する手法は存在しない．しかし，小規模なプログラムならば生成可能である．深層学習を利用した DeepCoder[1] は，使える関数の数に制限があり，かつ小規模なプログラムだという欠点はあるが，プログラムの自動生成ができる．

本研究では，DeepCoder が真に妥当かどうかを追試する．また，関数の数や定義など，DeepCoder の論文で述べられている条件を変更したときでも，生成されたプログラムが期待したプログラムと同じかどうかの正解率や，プログラムの自動生成にかかる実行時間の変化を調査する．

2　DeepCoder

2.1　DeepCoder とは

DeepCoder[1] とは自動でプログラムを生成する手法である．プログラムを生成する際にニューラルネットワークを用いる．DeepCoder が行うのは，統合開発環境が行うような局所的なコードの生成ではなく，完全なプログラムの生成である．使用できる関数が 34 個のみというドメイン固有言語のプログラムを生成する．また，使える型は整数型，論理型，および整数型のリストのみである．使える関数や型に制限はあるが，簡単なプログラミング問題ならば解くことができる．

2.2　DeepCoder のプログラム生成手法

次に，DeepCoder がどのようにして自動プログラミングを行うかを述べる．DeepCoder は，

1. 生成したいプログラムに与える入力と期待する出力をニューラルネットワークに与える
2. ニューラルネットワークは与えられた情報を基にして，それぞれの関数が生成したいプログラムで使用されている確率を推測する（確率については別紙）
3. 確率の高い関数から優先的に探索をして，生成したいプログラムを求める

という手順でプログラムを生成する．ニューラルネットワークを使わなくても，すべての探索空間を調査すれば生成したいプログラムを求めることはできるが，探索空間は広く時間がかかるので，ニューラルネットワークを使って探索時間の短縮をするのが目的である．

2.3　機械学習モデル

たとえば，次のプログラムを考える．

```
a <- [int]
b <- FILTER (<0) a
c <- MAP (*4) b
d <- SORT c
e <- REVERSE d
```

出力の値はすべて負で，4 で割り切れ，降順でソートされているという特徴を持つ．ニューラルネットワークはプログラムの入出力からこれらの特徴を分析し，個々の関数の有無を予測させることを学習する．

2.4　実装

DeepCoder の論文には理論のみ示されており，ソースコードなどは公開されていないので，`HiroakiMikami/deep-coder`[2] で公開されている DeepCoder の再実装を一部利用することとした．

3　DeepCoder の評価

図 1 は，訓練データのプログラム行数 T_{train} とテストデータのプログラム行数 T_{test} において，DeepCoder を使わずに探索した場合よりどれくらい速度が上昇したかの関係である．本実験では，この値が本当に妥当なのかを検証する．表 1 は，プログラムの行数が 1，および 2 のときに生成できるすべてのプログラムにおいて，テストデータから正しいプログラムが生成されたときの平均探索時間が，DeepCoder を使わずに探索した場合よりどれくらい速度が上昇したかを示したものである．これより，$T_{test} = 2$ のとき，論文に記されているほどの速度上昇は見込めず，妥当であるとは言えないことが分かる．$T_{test} = 3, 4, 5$ のときの検証も引き続き必要である．

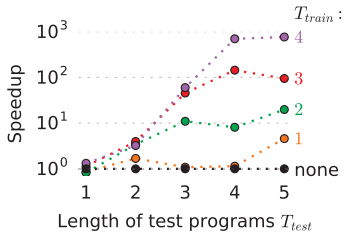

図 1　訓練データのプログラム行数，テストデータのプログラム行数，探索時間の速度上昇の関係（文献 [1]）

表 1　平均探索時間の倍率を実測した結果

T_{train} \ T_{test}	1	2
1	1.73	2.95
2	1.49	2.91

4　おわりに

本研究では，DeepCoder が真に妥当かどうかを追試した．しかし，表 1 において，$T_{test} = 3, 4, 5$ の場合の学習には計算機のメモリが足りず，測定できなかった．今後の課題として，$T_{test} = 3, 4, 5$ の場合の検証が挙げられる．

参考文献

[1] Matej Balog, Alexander L. Gaunt, Marc Brockschmidt, Sebastian Nowozin, Daniel Tarlow. “DeepCoder: Learning to Write Programs”. International Conference on Learning Representations 2017.

[2] 三上 裕明. “GitHub - HiroakiMikami/deep-coder: Re-implement DeepCoder”. `https://github.com/HiroakiMikami/deep-coder`. 2019-1-14 閲覧.