

Input Output

システム科学技術学部 機械知能システム学科

2年 太田 奨

2年 関野 晃聖

2年 小林 聖輝

2年 青山 浩也

指導教員 システム科学技術学部 知能メカトロニクス学科

教授 佐藤 和人

准教授 間所 洋和

指導補助 システム科学技術研究科 機械知能システム学専攻

修士2年 高橋 亮裕

1. はじめに

新しいプログラミング言語を勉強するために、人気な言語である Python を使用し、将棋プログラミングをしてそれに AI のプログラムを付け加えることとしたが、時間が足りず AI のプログラムを学ぶ段階には至らなかった。

将棋は藤井聡太さんの活躍があったため、新しくプログラミングを勉強し、将棋 AI を作成することとした。

2. 将棋プログラム

将棋プログラム作成にあたり、必要となってくる基本的な Python について、インターネットを利用して調査した。また将棋 AI 作成に関わるニューラルネットワークおよび、深層学習についても調べたため、まとめていく

(1) Python

Python とは現在(2018年5月調べ *1)、22.8%のシェア率で最も利用されているプログラミング言語である。これは、Python はデータ解析や Deep Learning の分野で広く活用されている。また、文法がシンプルで書きやすく、計算モジュールが充実していることが多分野で利用される理由であると考えられる。

CPU 対戦を実装するとなると、多大な計算量を要すると考えたため、将棋との相性が良いという点や多くの機能と性能性の高いモジュールがついているという点から、Python を利用することにした。

図1に Google の検索情報を元に人気をランキング付けした表を示す。(*1)

Worldwide, May 2018 compared to a year ago:				
Rank	Change	Language	Share	Trend
1	↑	Python	22.8 %	+5.5 %
2	↓	Java	22.5 %	-0.7 %
3	↑↑	Javascript	8.57 %	+0.2 %

図1 言語ランキング

(2) ニューラルネットワークと Deep Learning (*2)

ニューラルネットワークとは、神経回路網を人工ニューロンという数式的なモデルで表現したものであり、機械学習やディープラーニングなどの基本的な仕組みであり、入力層、出力層、隠れ層の3層から構成され、層と層の間には、神経細胞（ニューロン）同士のつながりの強さを示す重み「W」がある。

Deep Learning とは人工知能の要素技術の1つであり、AI そのものというわけではない。Deep Learning は大量のデータから、人間の力なしで、特徴を抽出してくれるディープニューラルネットワーク（DNN）を用いた学習のことをさす。

図2 にニューラルネットワークの構造と図3 に機械学習とディープラーニングの関係を示す。

図2 ニューラルネットワークの構造

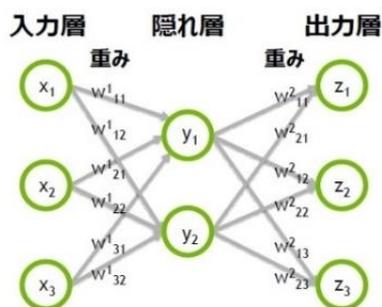
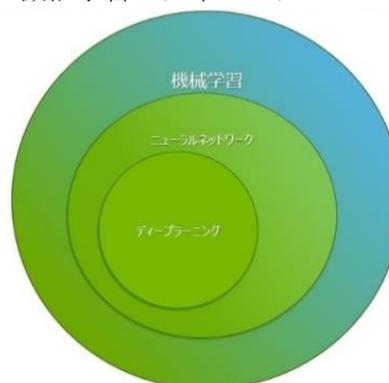


図3 機械学習とディープラーニングの関係



3. 研究内容

私たちは研究を進めるに当たり、将棋のルールについて知った上でプログラミング言語の Python について学習し、Python がどのような仕組みで機能するかを知ろうと思った。しかし、思いのほか時間が足りず、AI の仕組みを学ぶ段階には至らなかったため、将棋ゲームの作成に焦点を当てることとした。将棋プログラムの作成過程において、盤面上における駒の配置やそれぞれの駒の動きに制限をかけるなど、必要なコード（関数）を作成した。

オブジェクト機能を有する `tkinter` ツール (*4) を用いて将棋プログラムを構築した。具体的な手順を以下に示す。

- ① 将棋盤面の概形形成 (`tk.canvas`)
- ② 対局時の盤面マスを描画 (`canvas.create_rectangle`)
- ③ 将棋の駒をテキスト生成
- ④ 各駒の動きの制限 (`for` 文や `if` 文)
- ⑤ 成り駒の動作範囲の決定
- ⑥ 持ち駒の数と場所の制限

製作した将棋プログラムを実際に起動させ、対戦途中の盤面の様子を図 4 に示す。

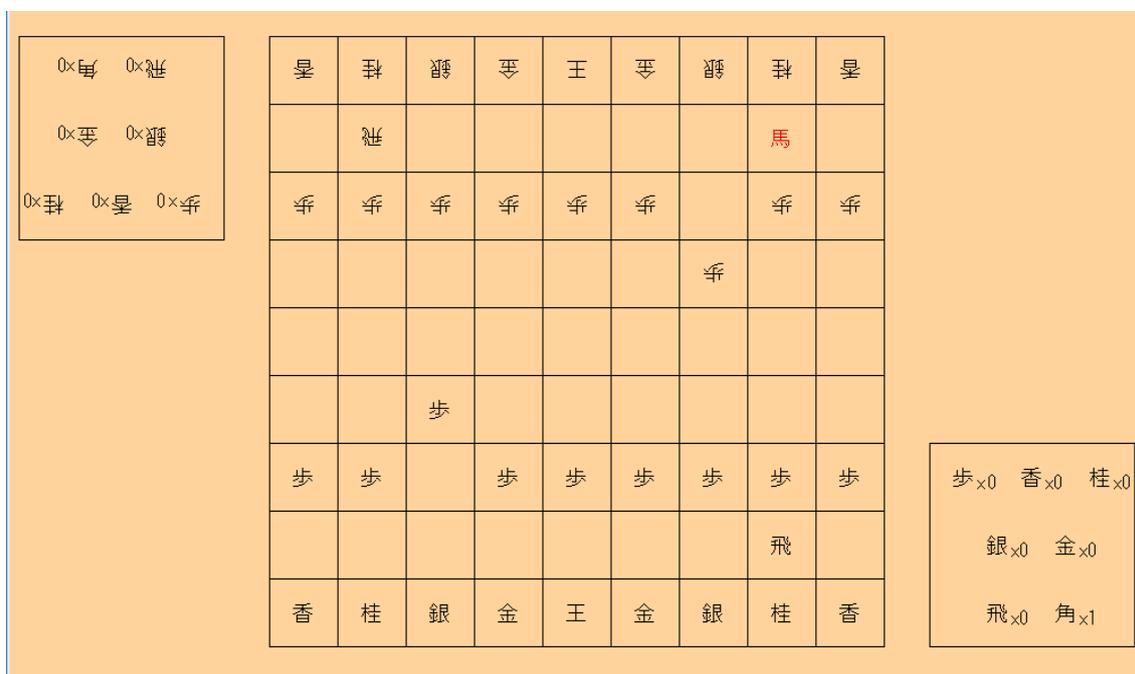


図 4 将棋の盤面

Pythonista3 (*3) は、iOS デバイス (iPhone や iPad) で Python プログラミングを可能とする統合開発環境である。その特徴は以下の通りである。

- ・ Python の基本的なライブラリと (グラフィックス・サウンド・iOS システムサービスについての) 追加のモジュールのほとんどを含む
- ・ プログラミングを始めるための例が多く含まれている
- ・ その他の人気のサードパーティーのモジュールを多く含む (request・BeautifulSoup・Flask・bottle・SymPy など)
- ・ アルファベットを 1 文字入力することで、そのアルファベットを含む関数などを予測変換し、素早い変換で入力することができる

4. 結果と考察

今回の自主研究では、26万行のコード（関数）を用いて将棋プログラムを構築した。実際には、1つの駒を動かすためにfor文やif文を用い、その動かした駒が60×60pixelの中に収まるように調整するなど、一つ一つの駒に多くのコードを使う必要があることがわかった。また、この作業を通して、Pythonがどのようなコードで動作するかが理解できた。

一方、同じコード（関数）を再利用した後の余分なコードを整理することで、プログラム全体の行数が圧縮されると共に作業時間の短縮につながったと考える。

5. まとめ

将棋プログラムの作成とデバッグを通して、最近注目されているPythonの使い方を学ぶことができた。何より一つのゲームを作成するために大量のコードが必要となることが分かった。そのため、多くの研究時間を要したことからディープラーニング学習までは至らなかった。

また、Pythonista3にtkinterのライブラリが入っていなかったため、将棋プログラムのアプリ化まで達成できなかったことは残念である。機会があれば、この研究成果にディープラーニングを取り入れたいと考える。

6. 参考文献

- *1 (PYPL PopularitY of Programming Language 2019年2月23日 15:37
<http://pypl.github.io/PYPL.html>)
- *2 将棋 AI で学ぶディープラーニング,山岡忠夫,マイナビ,2018年発行
- *3 (app store の説明欄 2019年2月18日 13:32
<https://itunes.apple.com/jp/app/pythonista-3/id1085978097>)
- *4 (お気楽 Python/Tkinter 入門 2019年2月24日 16:44
http://www.geocities.jp/m_hiroi/light/pytk01.html)