様式６

申請先学部　　基礎工学部　採択番号　No.3

平成２９年度学部学生による自主研究奨励事業研究成果報告書

<table>
<tr><td>ふりがな
氏　　名</td><td colspan="2">あさか　ともき
淺香　智輝</td><td>学部
学科</td><td>基礎工学部
システム科学科</td><td>学年</td><td>１年</td></tr>
<tr><td rowspan="3">ふりがな
共　同
研究者氏名</td><td colspan="2"></td><td rowspan="3">学部
学科</td><td></td><td rowspan="3">学年</td><td>年</td></tr>
<tr><td colspan="2"></td><td></td><td>年</td></tr>
<tr><td colspan="2"></td><td></td><td>年</td></tr>
<tr><td>アドバイザー教員
氏名</td><td colspan="2">いけもと　しゅうへい
池本　周平</td><td>所属</td><td colspan="3">大阪大学大学院基礎工学研究科
システム創成専攻</td></tr>
<tr><td>研究課題名</td><td colspan="6">タスク学習時の有益性に基づきユーザーを選り好みするパーソナルロボットの開発</td></tr>
<tr><td>研究成果の概要</td><td colspan="6">研究目的、研究計画、研究方法、研究経過、研究成果等について記述すること。必要に応じて用紙を追加してもよい。(先行する研究を引用する場合は、「阪大生のためのアカデミックライティング入門」に従い、盗作剽窃にならないように引用部分を明示し文末に参考文献リストをつけること。)</td></tr>
</table>

１．初めに

本研究では、タスク学習時の有益性に基づき、ユーザーを選り好みするパーソナルロボットの開発を行った。具体的なタスクには、ロボットに目的地までの経路を自立して探索させる、リーチング学習というものを設定した。学習には、Q-Learning という教師なし学習の手法を用いた。これは、状態と行動、及びその行動による報酬で決定される評価値を得ながら、試行を繰り返すことによって、より正確な評価値のマップを作成する学習である。このリーチング学習において、良いユーザーとは、リーチングの達成を補助するユーザー（以下、補助ユーザーと呼ぶ）であり、悪いユーザーとは、妨害ユーザー（以下、妨害ユーザーと呼ぶ）である。これをもとに各ユーザーの評価値を定め、学習時に、ユーザーごとに異なる強さの反映をさせる。

本研究の意義は、悪意のあるユーザーからの学習を防げることである。かつて、Microsoft の Tay というチャットボットが、そのようなユーザーからの学習によって、暴言を吐くようになったことがある。学習対象に選好性を持たせることは、このような問題の解決策の一つになると考えられ、ひいては、より生き物らしい挙動を示すロボットの開発の一助となると考えられる。

２．目的・目標

本研究の目的は、Q-Learning に選好性を加えることによって、ユーザーを選り好みするパーソナルロボットの開発である。具体的な目標は、各ユーザーに対し、最適な行動と学習をロボットに取らせることである。（表１）

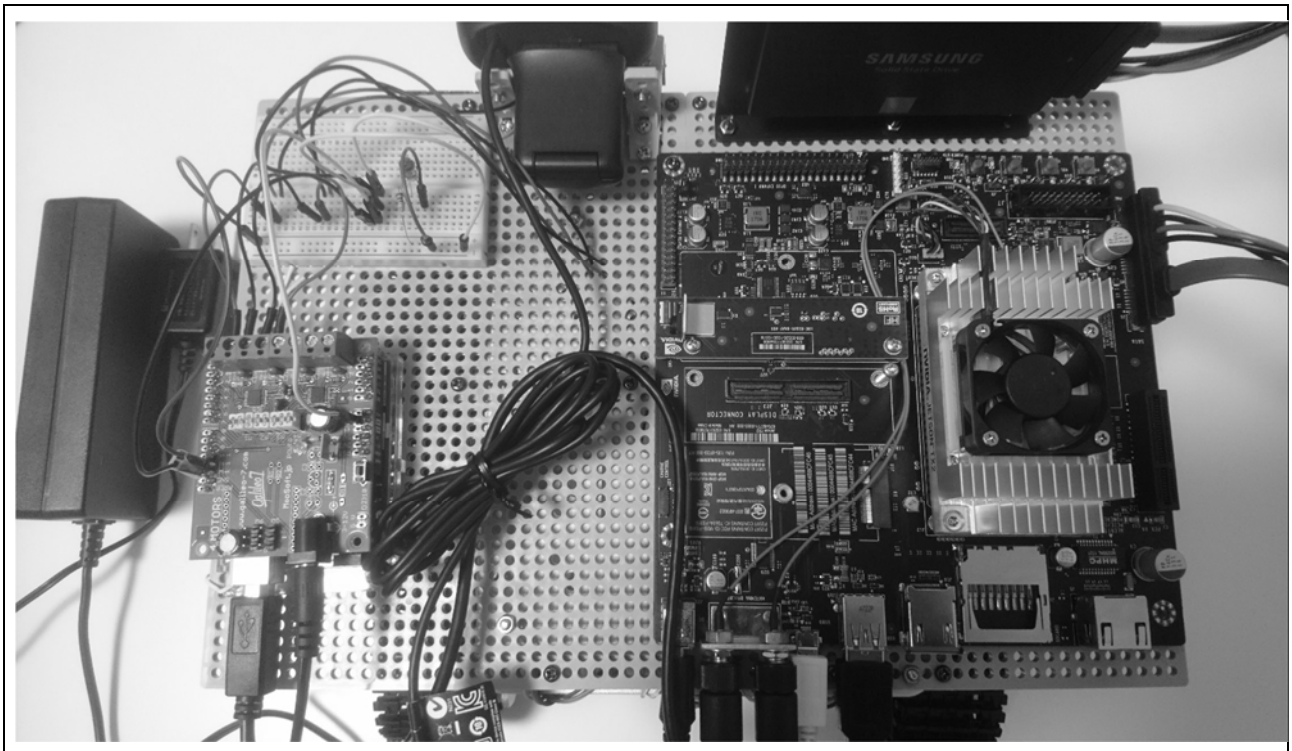
表１．目的とする Q-Learning の内容

ユーザーの種類	行動	学習
ユーザーなし	行動実行または停止	学習実行（リーチングの可不可は問わない）
補助ユーザー	行動実行	学習実行（強い反映）
妨害ユーザー	行動実行から、停止に変化	学習実行（弱い反映）

３．装置

リーチングのために、物体視と Q-Learning を行わせるコンピュータとして、NVIDIA 社の JETSON TX2(以下、JETSON と呼ぶ)を使用した。JETSON に搭載されている、Pascal という GPU（ディープラーニングに特化したプロセッサ）を使用して、物体視を行わせる。本研究では yolo_tensorflow を元コードにして、オブジェクトの学習、画像の解析を行うプログラムを作成した。また、物体視と並行して、Q-Learning を行う自作プログラムも実行させる。

また、JETSON に Arduino を接続し、シリアル通信を行う。Arduio には Galileo7 社の 4Motors Shield を載せ、ロボットのモーターを制御する。JETSON は、ホスト PC と SSH 通信することで操作する。



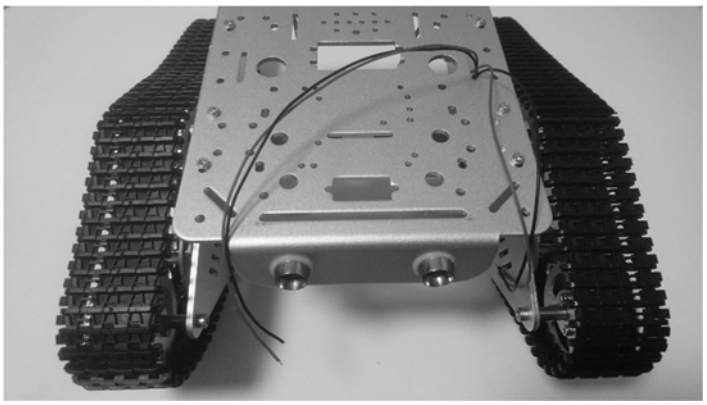
画像 1: 右下のボードが JETSON、右上が SSD、左下が Arduino+4Motors Shield、中央上が Web カメラ
左上のブレッドボードは、モーターとスイッチ制御用の回路

JETSON の内臓メモリ（eMMC）は 30GB 程度しかなく、Linux OS(Ubuntu)に、プログラム実行の依存ライブラリである TensorFlow や OpenCV などをインストールすると、容量が足りなくなるため、SSD 上（250GB、SATA 接続）で Boot するようにした。

また、モーターを動かす電力は、JETSON から得る電力では足りないため、4Motors Shield にも、電力を直接供給する。OSOYO 社のロボットカーシャーシ金属製キャタピラー（画像２）に上記の回路を搭載させた。

Q-Learning のプログラム実行前には、target（画像３）を認識できるように、訓練データを作成しておく。ロボット前部に Target（導体）が触れると通電するスイッチを取り付け、Arduino でその値を読み取り、結果を JETSON に送ることで Q-Learning の報酬を得る。

Q-Learning 用のプログラムは Python で作成した。Arduino は C 言語ベースのプログラムで動作する。



画像２：キャタピラロボット

画像３：target（導体）

４．プログラム

通常、Q-Learning のプログラムは、以下の Q 値（状態と行動及び、その連続の仕方に依存する評価値）の更新式によって行われる[1]。報酬の設定方法は今回の実験用である。

Q(s, a) = Q(s, a) + 学習率 ALPHA * {r(s, a) + 割引率 GAMMA * maxQ(s', a') - Q(s, a)}

Q(s, a): 状態 s における行動 a の Q 値

r(s, a): 状態 s での報酬

・1 つの action ごとに減少(動かないときは変動なし)

・ゴール時にのみ報酬(+30)

maxQ(s', a')　次の状態 s'における行動 a'の Q 値のセット中の最大値

本研究では、ユーザーに対する選好性を設けるために、上記の更新式に対して、

Q(s,u,a) = Q(s,u,a) + 学習率 ALPHA * {r(s,u,a) + 割引率 GAMMA * maxQ(s',u,a') - Q(s,u,a)}

Q(s,u,a): 状態 s、ユーザーu における行動 a の Q 値

r(s,u,a): 状態 s、ユーザーu での報酬(+30)

・1 つの action ごとに減少(動かないときは変動なし)

・ゴール時にのみ報酬

maxQ(s',u,a')　次の状態 s'、ユーザーu における行動 a'の Q 値のセット中の最大値

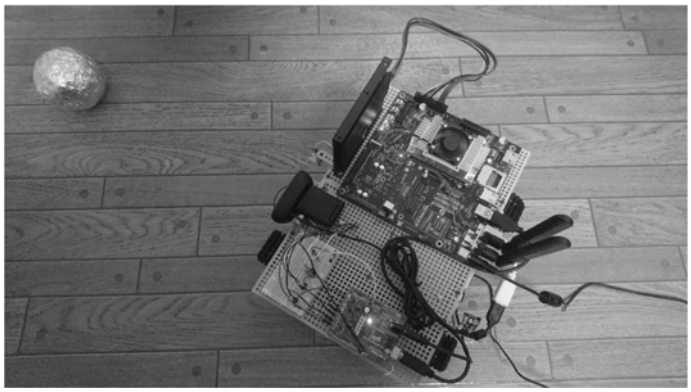
というように変更した。上記のままでは、単に状態が u の数だけ倍されることになるため、さらに u に対する評価値をそれぞれのユーザーの Q 値に依存する値として、学習に選好性を加えた。
具体的には、ユーザーに対する Q 値の和を更新回数で割り、この絶対値を重みとする。
全てのユーザーの Q マップに、それぞれの重みを掛けて和を計算することで、状態と行動の General Q マップを生成する。上記の報酬の設定では、良いユーザーの Q 値が強く反映され、悪いユーザーの評価値は弱く反映された、Q マップを得ることができる。
また、ε -greedy 法（確率 1- ε でランダムに行動する）を利用して、既知の Q 値のみで行動を選択しないようにプログラムした。

５．実験

画像４・５は user:0（ユーザーなし）の状態である。他に、user:1（妨害ユーザー）、user:2（補助ユーザー）の状態がある。次元数は、state（target の有無や位置）が 4 次元、user が 3 次元、act（離散的行動パターン）が 6 次元の計 72 次元で、 ε -greedy 法における ε の値は 0.7 に設定した。学習は 200 回（約 40 分間）行い、Q 値の正規化を目指した。

User:2（補助ユーザー）では、ユーザーに近づいた場合、target の状態に関わらず直接 target を触れさせ、報酬を与えた。state や act の詳細は**６．結果**の表３に示す。

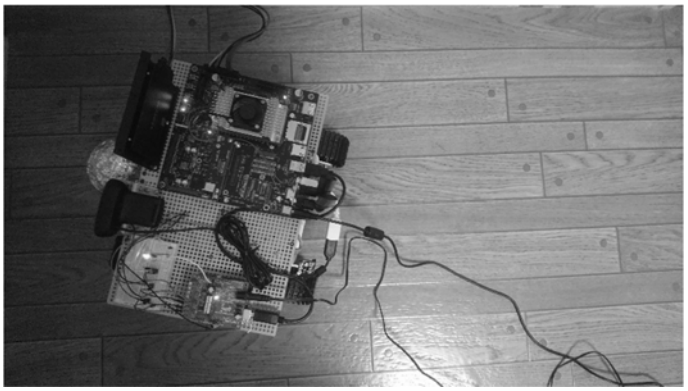
以下に、Q 値の更新中の画像およびプログラムのログを示す。



～プログラムのログ～

Count:　3
Average detecting time: 0.418s
Target is recognized
random action
state : 1　user: 0　act: 3
Q_value: 0.0　→　-0.2

画像４：Q-Learning 実行中（act:3 実行後）



Count:　4
Average detecting time: 0.407s
Target is recognized
greedy action
Catched the Target
state : 1　user: 0　act: 1
Q_value: 0.0　→　6.0

画像５：Q-Learning 実行中（act:1 実行後）

６．結果、結論

５．実験によって得られた Q マップを以下に示す。（表２、３）

表 2. 学習によって得られた Q-map（行：act の番号、列：state_user）

	0_0	1_0	2_0	3_0	0_1	1_1	2_1	3_1	0_2	1_2	2_2	3_2
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	-0.3	6.3	-0.5	-0.2	-0.6		-0.2	-0.3	21.5	11.4	-0.3	-0.2
2	-0.2	-0.2	-0.2	-0.5	-0.3		-0.2	-0.2	-0.2			16.2
3	-0.5	-0.2	-0.2	-0.2	-0.8		-0.2	-0.2			17	-0.3
4	-0.3	-0.2	-0.6	-0.2	-0.3	-0.2	-0.3		-0.2	-0.2	-0.2	-0.2
5	-0.8	-0.2	-0.3	-0.2	-0.5		-0.3	-0.3				

※ε-greedy 法で学習を行ったため、一部のデータが取れておらず、空欄にしてある。

表３．学習データから生成した General Q マップ（行：act の番号、列：state）

	0(なし)	1（画面中央）	2（画面左）	3（画面右）
0（停止）	0.00	0.0	0.0	0.0
1（前進）	200.74	110.25	-3.94	-1.88
2（右前前進）	-2.00	-0.17	-0.17	147.95
3（左前前進）	-0.49	-0.17	158.41	-3.64
4（右に 45° 回転）	-2.16	-2.00	-2.43	-2.00
5（左に 45° 回転）	-0.72	-0.174	-0.34	-0.18

表２．表３．に示したように（アンダーバーが付いた値）、user:2（補助ユーザー）の Q 値が強く反映され、user:1（妨害ユーザー）の Q 値は弱く反映された Q マップを得ることができた。他の Q 値に対し、いくつかの Q 値が大きくなり過ぎてしまったが、これは報酬を下げることで対処することができる。少ない学習回数で、簡易的な実験を行った際の報酬設定のまま、200 回の Q 値の更新を行ったことが原因だと思われる。表３の state:0・前進の Q 値が大きいのは、**5．実験**に示した報酬の与え方による。

しかし、悪いユーザーの個体数が増えた場合を想定すると、プログラムの管理者を良いユーザーとして、General Q マップを用意し、Q 値の初期値とするということは有効だと考えられる。

学習が 150 回を過ぎたあたりで、ランダムに動いた場合を除き、

・User:1（妨害ユーザー）　→　act:0（停止、報酬の減少なし）

・User:2（補助ユーザー）　→　act:1～3（前進に準ずる行動）

というように、ユーザーを選り好みした挙動を、ロボットに示させるができた。これを生き物らしいと捉えるかどうかは、個人の主観によるところではあるが、状態やユーザー、行動などの次元数が上がり、より複雑な挙動を示すロボットに組み込むことで、より生き物らしくすることができると期待できる。

以上より、タスク学習の有益性からユーザーを評価し、行動・学習に選好性を与えるという点において、一定の成果を得ることができた。

７．謝辞

本研究を行うにあたり、大阪大学大学院基礎工学研究科システム創成専攻の池本周平助教から、様々なご指導をいただきました。専門知識に乏しい一回生の身で、研究室に入れて頂き、最新の機材を使用させていただき、学ばせていただいたことは、大変貴重な経験となりました。ありがとうございました。また、このような機会を提供して下さり、自主研究奨励事業を支えて下さる皆様に、この場をお借りして感謝申し上げます。

参考文献

[1] LONG-JI LIN「Self-Improving Reactive Agents Based On Reinforcement Learning, Planning and Teaching」

http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.75.7884　(2017/12/03 最終閲覧)