深層強化学習によるロボットのバラ積みピッキング習得（第１報）

町田晃平・石黒　聡・細谷　肇*

Learning of Robot Bin Picking by Deep Reinforcement Learning
(1st Report)

Kohei MACHIDA, Satoshi ISHIGURO, Hajime HOSOYA

深層強化学習 Deep Q-Learning によるロボットのバラ積みピッキングを提案する。人間の教示を組み合わせたことにより、約 10 時間の深層強化学習でピッキングに成功した。今後は、本バラ積みピッキング技術や深層強化学習技術を展開し、企業の生産性向上を支援する。

キーワード：AI、深層強化学習、ロボット

We propose robot bin picking using deep reinforcement learning called as Deep Q-Learning. We report the successes of bin picking using deep reinforcement learning in about 10 hours because of human teaching. In the future, we expand this bin picking technology and deep reinforcement learning technology to support the improvement of productivity for enterprises.

Keywords : Artificial Intelligence, Deep Reinforcement Learning, Robot

１　はじめに

工場の生産性を高めるには、ロボットの活用が必要不可欠である。しかし、現状は多くの作業を人手に頼っている。そのような作業の一つとして、バラ積みされたワークの取り出し（バラ積みピッキング）がある。

バラ積みピッキングをロボットとマシンビジョンで実現しようとした場合、バラ積みワークが複雑に重なり合っているため、従来の画像処理手法ではピッキング箇所の取得が難しいという問題があった。

一方、近年の AI（人工知能）技術の発達により、特に画像認識分野において、様々な成果が報告されている。中でも、深層強化学習は、AI が目的を達成するために自ら試行錯誤を繰り返して学習し、最適な行動方法を獲得していく手法である。深層強化学習は、事前に学習データを用意する必要がなく、AI が自動で高いスキルを獲得することができるため、産業分野への応用が期待されている。

そこで、本研究では、この深層強化学習によるロボットのバラ積みピッキング習得を目指す。昨年度の研究では、深層強化学習による平面に置かれたワークのピッキングを実現した。本研究では、バラ積みされたワークのピッキングに取り組む。

２　研究方法

２．１　強化学習

強化学習は、エージェント（AI）が環境との相互作用により学習を行う。強化学習の枠組みを図１に示す。エージェントは、タイムステップtにおいて、観測した状態$s_t \in S$から行動$a_t \in A$を選択する。それにより、エージェントは環境から報酬$r_{t+1} \in R$と次の状態s_{t+1}を受け取る。

エージェントの目標は、受け取る報酬の累積（収益）を最大化することである。収益は以下のように定義される。

ここで、$\gamma(0 \leq \gamma \leq 1)$は割引率と呼ばれ、

電子機械係　*研究調整官

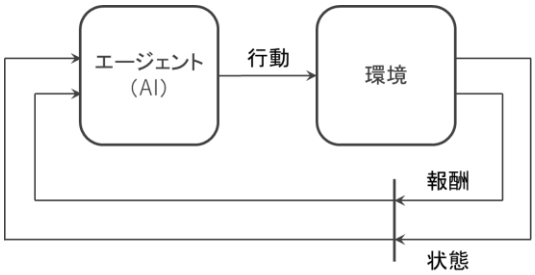

図１　強化学習の枠組み

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \quad (1)$$

将来の報酬への割引を決定するパラメータである。また、Tはタイムステップの終端を表す。

方策πのもと、状態sおいて行動aを取ることの価値を行動価値関数$Q^{\pi}(s,a)$として表す。行動価値関数は以下のように定義される。

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{T} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\} \quad (2)$$

E_{π}は方策πにおける期待値である。上式のように、行動価値関数は期待報酬として表される。エージェントに行動価値関数を学習させるアルゴリズムを Q 学習と呼ぶ。Q 学習では、以下のように行動価値関数を更新する。

$$\mathrm{Q}(s_t,a_t) \leftarrow \mathrm{Q}(s_t,a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a)\right] \quad (3)$$

学習で獲得される行動価値関数は、使われている方策とは独立に、最適行動価値関数Q^*を近似する。これにより、エージェントは現在の状態s_tから行動価値関数を用いて最適な行動$\max_a Q^*(s_t,a)$を取ることができるようになる。

２．２　深層強化学習

本研究では、深層強化学習のアルゴリズムとして、Mnih らによって提案された Deep Q-Learning[1)] を用いた。Deep Q-Learning は、Q 学習における行動価値関数をニューラルネットワークによって近似する手法である。Deep Q-Learning におけるニューラルネットワークを Deep Q-Network(DQN)と呼ぶ。DQN に状態を入力することで、各行動を行った場合の価値が出力される。

図２　実験環境

近似された行動価値関数を$Q(s,a;\theta)$と表す。ここで、θはニューラルネットワークのパラメータである。Deep Q-Learning では、このパラメータθを学習により更新する。

Deep Q-Learning のアルゴリズムをアルゴリズム１に示す。アルゴリズム１において、リプレイメモリDは、遷移情報$(\phi_t, a_t, r_t, \phi_{t+1})$を格納しておくためのメモリである。学習の際には、このリプレイメモリDから遷移情報のミニバッチ$(\phi_j, a_j, r_j, \phi_{j+1})$を取り出して学習を行う。これは、Experience Replay（経験再生）と呼ばれる手法である。時系列として相関のない状態・行動・報酬の組を用いて学習した方が、学習の効率が上がるため、Deep Q-Learning ではこの Experience Replay が用いられている。また、学習時は試行錯誤による探索を促すため、ε-greedy 方策に従って確率εでランダムな行動を選択する。それ以外の場合には、最適な行動$\max_a Q^*(s_t,a;\theta)$を選択する。

２．３　実験条件

実験の環境を図２に示す。ロボットは、小型のロボットアーム uArm Swift Pro（UFACTORY 社）を用いた。uArm Swift Pro は、pyuf と呼ばれる Python ライブラリを用いて、Python により制御した。カメラは、RealSense Depth Camera D435（Intel 社）を用いた。RealSense Depth Camera D435 は、ステレオビジョンの深度

アルゴリズム１　Deep Q-learning

リプレイメモリDをキャパシティNに初期化する
行動価値関数$Q(s,a;\theta)$をランダムな重みθで初期化する
各エピソード（1～M）に対して繰り返し：
　初期画像x_1を初期状態$s_1=\{x_1\}$として初期化し前処理$\phi_1=\phi(s_1)$を行う
　各タイムステップ（1～T）に対して繰り返し：
　　行動$a_t=\begin{cases}\text{random} & (\text{確率}\ \varepsilon)\\ \max_a Q^*(s_t,a;\theta) & (\text{確率}\ \varepsilon-1)\end{cases}$　を選択する
　　行動a_tを実行し報酬r_tと次の画像x_{t+1}を受け取る
　　次の状態$s_{t+1}=s_t,a_t,x_{t+1}$をセットし前処理$\phi_{t+1}=\phi(s_{t+1})$を行う
　　遷移情報$(\phi_t,a_t,r_t,\phi_{t+1})$をリプレイメモリ$D$に格納する
　　リプレイメモリDから遷移情報$(\phi_j,a_j,r_j,\phi_{j+1})$のミニバッチを取り出す
　　教師データ$y_j=\begin{cases}r_j & (\phi_{t+1}\text{が終端の場合})\\ r_j+\gamma\max_{a'}Q(\phi_j,a';\theta) & (\phi_{t+1}\text{が終端でない場合})\end{cases}$　をセットする
　　教師データと行動価値関数の誤差$\left(y_j-Q(\phi_j,a_j;\theta)\right)^2$を求め誤差逆伝搬を実行する

図３　RGB 画像

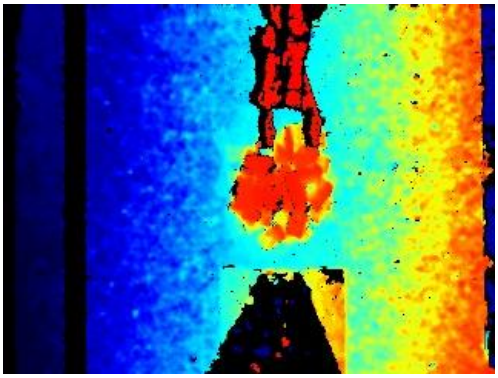

図４　深度画像

カメラである。2 つの深度センサ、RGB センサ、IR 投射器を搭載する。本実験では、RealSense Depth Camera D435 より取得した RGB 画像（図 3）と深度画像（図 4）を DQN へ入力した。画像のサイズは、どちらも 320×240 の 3 チャンネルである。バラ積み部品を想定したワークは、ドミノ 60 個を使用した。ドミノは平面の上にランダムに山積みした。

Deep Q-Learning のプログラムは、深層強化学習ライブラリ ChainerRL を用いて Python で実装した。

エージェントは各タイムステップにおいて、アーム水平方向の上・下・左・右移動と垂直方向の上・下移動およびピッキングの 7 つの行動から選択を行う。ピッキングを選択するとその場でアームを閉じ、uArm Swift Pro のピッキング判定によりピッキングの成功が確認されれば、エージェントに報酬+1 が与えられる。ピッキングに失敗した場合やアームを移動した場合の報酬は-1 である。タイムステップの終端 T は 100 とした。タイムステップの終端 T が終わると、アームは初期位置に戻り、次のエピソードが開始される。

実験に使用した DQN の構造を図 5 に示す。3 層の畳み込み・プーリング層と 2 層の全結合層からなるニューラルネットワークである。最初の畳み込み層へは、3 チャンネルの RGB 画像と深度画像を結合した 6 チャンネルの画像を入力した。畳み込み層と全結合層の活性化関数は、Leaky ReLU を用いた。

学習の最適化手法は、SGD（確率的勾配降下法）を用いた。SGD の学習率は 0.01 とした。学習時の GPU は、NVIDIA GeForce 1080 Ti を用いた。

報酬の割引率γは 0.99 とした。ε-greedy

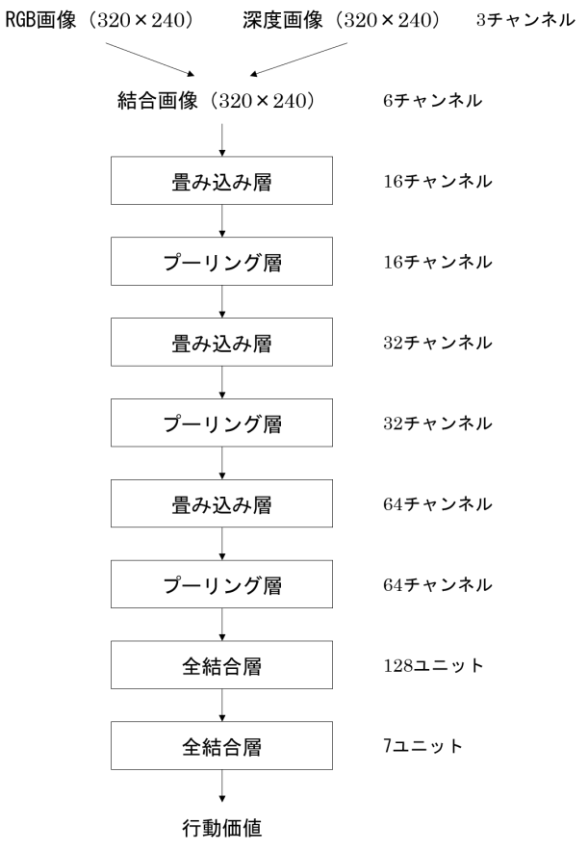

図５　Deep Q-Network の構造

方策の確率εは、初期タイムステップで 0.5、その後各タイムステップで線形に減少させ、3000 タイムステップ（30 エピソード）で 0.3 となるようにした。リプレイメモリDのキャパシティNは 1,000,000 とした。Experience Replay には、教師データと行動価値関数の誤差が大きい遷移情報を優先的に取り出して学習を行う Prioritized Experience Replay（優先順位付き経験再生）を用いた。

昨年度の研究より、事前に人間の教示を行うことで、深層強化学習の学習時間を短縮できることが確認された。そのため、本研究では、事前に約 30 分（500 タイムステップ分）の教示を行った後に、深層強化学習を行った。教示は、各タイムステップにおいて、オペレータがロボットの取るべき行動を選択した。

3　研究結果

約 30 分の人間の教示後、約 10 時間の深層強化学習により、バラ積みされたワークのピッキングに成功した（図６）。

図６　ピッキング成功時（教示 30 分、深層強化学習 10 時間後）

一方、人間の教示を行わなかった場合には、1 日以上、深層強化学習を行ってもピッキングに成功しなかった。このことから、実用的な時間内で学習を完了するには、人間の教示が重要であることが確認された。また、本バラ積みピッキングを従来の画像処理手法等で実現しようとした場合、数日以上のソフト開発工数がかかることが想定される。このことから、従来手法と比較して、深層強化学習を用いることで開発工数の削減が期待できる。

4　まとめ

深層強化学習 Deep Q-Learning によるロボットのバラ積みピッキングを実現した。本研究の成果により、人間の教示を組み合わせることで実用的な時間内に深層強化学習を完了できることや、従来手法と比較して深層強化学習がソフトの開発工数削減につながる可能性があることが確認された。

今後は、本バラ積みピッキング技術や深層強化学習技術を県内企業に展開し、企業の生産性向上を支援する。

文献

1）Volodymyr Mnih, *et al.*: Human-levelcontrol through deep reinforcement- learning, *Nature*, 518(7540):529, 02(2015)