深層強化学習による対話メディアのモデリング

竹原　一彰

リクルート住まいカンパニーは，不動産・住宅に関する総合情報サイト「SUUMO（スーモ）」を運営している．理想の住まい探しは，検討すべきポイントが多岐にわたり，かつ大きな金額を伴うため，意思決定が困難になりがちである．そのため，人間のアドバイザーが行うようなきめ細やかな対話機能，たとえば，ヒアリングによる要望整理と提案，不安解消に向けた Q&A などが求められている．本稿では，インターネットメディアに対話機能が求められるようになった背景から，実装における技術的課題を示しつつ，その解決策の一つである深層強化学習によるモデリング事例を紹介する．

キーワード：理想の住まい探し，対話メディア，強化学習，Q-learning，深層学習，Deep Q-networks

1. はじめに

理想の住まいに出会うためには，多くのことを検討しなければならない．将来のプラン，希望するライフスタイル，予算制約などの抽象度の高い事項から検討をはじめ，段階的に詳細化していく．そのような検討を繰り返しながら，徐々に理想の住まいがイメージできるようになり，ようやく，建物種別，価格，エリア，間取りなどの具体的なスペックにまでブレイクダウンできるのである．

人生において住み替えを経験する機会は限られており，ほとんどのカスタマーは知識と経験が十分ではないため，上述の検討事項の洗い出し自体が困難である．たとえ，うまく検討を進めることができ，具体的なスペックにまでたどり着いたとしても，家を建てるにはどの施工会社に依頼するのがよいのか，資金はどのように獲得するのかなど，実行面でも課題が存在している．

このような背景から，専門家のような対話機能，たとえば，ヒアリングによる要望整理，効率的な段取りの提案，不安解消に向けた Q&A などを備え，カスタマーの課題にタイムリーに伴走できるようなインターネットメディア（以降，対話メディア）が求められるようになってきており，弊社でも検討を進めている．

2. 対話メディア

対話メディアの主な機能を紹介し，その実装における技術的課題を取り上げる．

2.1 主な機能

本稿で取り上げる対話メディアのシステム（以降，システム）は，

1. ヒアリング
2. 提案

の二つの機能のみ備えているとする．

システムとその利用者（以降，カスタマー）は，選択式のチャットユーザインタフェースを通じてやりとりを行う（表 1）．対話は，大きくヒアリングフェーズと提案フェーズで構成されている．

ヒアリングフェーズでは，システムは住み替えのきっかけ，将来のプラン，希望するライフスタイル，予算などの制約事項，物件のスペック（間取り，広さ，立地など）について，複数の項目をヒアリングする．ヒアリング項目は，あらかじめ用意しておいた定型の文言である．カスタマーは [はい]，[いいえ] のように該当するボタンをクリックすることで回答する．

提案フェーズでは，システムはヒアリングした内容に従って，物件や施工会社の提案，検討の段取りや注意ポイント，相場や周辺施設など，さまざまな情報提供を行う．カスタマーは [内見したい]，[ほかの物件を推薦] などのボタンをクリックすることで反応をフィードバックする．

2.2 技術的課題

対話メディアを構築するにあたって，主な技術的な課題を三つ取り上げる．

一つめは，ヒアリング項目（表 2）の選別である．表 2 に示した項目以外にも，会社への要望，担当者への要望，提案内容への要望など，ヒアリング種別の拡

たけはら　かずあき
株式会社リクルート住まいカンパニー
〒 104–0028　東京都中央区八重洲 2–8–1
ktakehara@r.recruit.co.jp

表 1　対話メディアにおけるやりとりのイメージ

話者	内容
	（ヒアリングフェーズ）
システム	最適な物件をご案内するため，いくつかヒアリングさせていただいてもよろしいですか？ [はい] [いいえ]
カスタマー	[はい].
システム	ありがとうございます．それでは，10 問ほど質問させていただきます． 賃貸，物件の購入どちらを検討されていますか？ [賃貸] [購入]
カスタマー	[賃貸] を検討しています.
システム	承知しました．賃貸物件をお探しですね．今回の住み替えのきっかけはどのようなものですか？ [就学] [就職] [結婚] [家族が増えた]···
カスタマー	[就職] です.
	⋮
	（提案フェーズ）
システム	こちらの物件はいかがでしょうか．職場へのアクセスも 30 分以内，スーパー・コンビニも徒歩 5 分以内と，おすすめの環境です． [内見したい] [ほかの物件を推薦]
カスタマー	[ほかの物件を推薦] してください.
	⋮

表 2　ヒアリング項目の例

大区分	中区分
基本情報	性別，年齢，家族構成 ···
住替え動機	不満，ライフイベント ···
検討状況	予算，スケジュール ···
物件への要望	希望エリア，価格，間取り ···
周辺環境への要望	自然，通勤，レジャー ···
	⋮

張が考えられる．また，ヒアリング文言自体に婉曲表現や直接表現などのバリエーションをもたせることを考慮すると，そのパターンは，数百〜数千のオーダーとなってしまう．一人のカスタマーに対して，すべてのパターンを試すわけにはいかないため，適切なヒアリング項目を取捨選択する必要がある．

二つめは，対話の順序と長さの制御である．ヒアリングフェーズは，抽象的な質問からはじめ，徐々に個人的かつ具体的な質問へと深めていくように Q&A を構成する必要がある．たとえば，対話の初回が「予算はいくらですか？」のような質問では，カスタマーは，その意図がわからず，回答を拒否しサイトを離脱してしまう可能性がある．また，フェーズ間の行き来も考慮する必要がある．最初に全部ヒアリングしてから提案するのがよいのか，早い段階で提案を行い，フィードバックを獲得しながら対話するのがよいのかということである．さらに，なるべく短い対話で，理想の住まいを提案できるほうがよいので，対話の長さも考慮する必要がある．

最後は，教師データの獲得である．過去のデータを用いて最適な対話行動を獲得するにあたり，システムリリース当初や，新しいヒアリング項目やほかの機能を追加した場合には，どのヒアリング項目やどの機能が，どのくらいよかったのかという教師データが存在しない．そのため，システム自身が，カスタマーとのやりとりを通じて，教師データとなる情報を集め，行動を最適化するような枠組みが必要となる．

3. 深層強化学習

上述の技術的課題を解消するための理論的枠組みとして，深層強化学習を取り上げる [1–3]．単に理論を紹介するのではなく，本稿の問題設定とのつながりを意識した解説を試みる．

3.1 強化学習

強化学習とは，ある環境におかれたエージェントが，環境との相互作用を通じて，教師データとなる情報を自律的に獲得しながら，最適な行動規則を獲得するための枠組みである [4]．

各時点 $t = 1, 2, 3, \ldots$ において，エージェントは，

・環境の状態 s_t を観測する

・方策 π に従い，行動 a_t を選択する

環境は，

・行動 a_t を受け取り，状態を s_{t+1} に遷移させる

・報酬関数 r に従い，報酬 $r_{t+1} \sim r(s_t, a_t, s_{t+1})$ を計算し，エージェントに送信する

というステップを繰り返すことで相互作用する．

本稿の問題設定では，エージェントがシステム，環境が一人ひとりのカスタマーに，相互作用がヒアリングや提案というシステムが選択した行動と，それに対するカスタマーの反応で構成される対話に相当している．

環境の状態は，カスタマーの要望を表している．しかし，エージェントはその要望を直接観測することはできず，ヒアリングや提案に対するカスタマーの反応を通じて観測する．

方策 π は，エージェントの行動選択規則を表す条件

付き確率 $\pi(a|s)$ である．この $\pi(a|s)$ により，エージェントはどの項目をヒアリングするのか，または，どういう提案をするのかという行動選択をする．

報酬は，行動の即時的なよさを表す値である．エージェントの行動に対して，カスタマーが好意的な反応を示したら +10 など，反応の良し悪しに応じて報酬の大小を設定する．

将来にわたってできるだけ多くの報酬を獲得するようにエージェントの行動規則を最適化するというのが強化学習の課題である．

3.2 Q-learning

時点 t から将来にわたって獲得できる報酬の総和を収益といい，G_t と記す．遠い将来の報酬より，すぐに獲得できる報酬のほうが価値が高いという考え方で，割引率 γ（0.9 など 1 に近い値を設定することが多い）を導入し，下式で表現する．

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots$$

G_t を最大化する $\pi(a|s)$ を決定したいのだが，G_t は，現在の状態と行動，方策に依存しており，そのままだと評価が難しい．そのため，それらを条件にした期待値 $\mathbb{E}[G_t|s_t, a_t, \pi]$ を利用する．これを $Q^\pi(s_t, a_t)$ で表し行動価値関数と呼ぶ．整理すると下式となる．

$$\begin{aligned} Q^\pi(s_t, a_t) &= \mathbb{E}[G_t|s_t, a_t, \pi] \\ &= \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \cdots |s_t, a_t, \pi] \end{aligned}$$

$Q^\pi(s, a)$ を最大化させる最適方策 π^* を求めるのが Q-learning が対象とする課題である．最適方策に基づく行動をとる場合の行動価値関数を，最適行動価値関数と呼び，$Q^*(s, a)$ で表す．すなわち，

$$Q^*(s, a) = Q^{\pi^*}(s, a) = \max_\pi Q^\pi(s, a)$$

とする．greedy 方策では，

$$a^* = \operatorname*{argmax}_a Q^*(s, a)$$

により，エージェントは行動を選択する．

$Q^*(s, a)$ を求めるアルゴリズムの一つである Q-learning では，下記のように逐次更新することで $Q^*(s, a)$ を獲得する．

$$\begin{aligned} Q(s_t, a_t) \leftarrow (1-\alpha) &\underbrace{Q(s_t, a_t)}_{\text{第一項}} \\ &+ \underbrace{\alpha(r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}))}_{\text{第二項}} \end{aligned}$$

第一項が現時点での Q 値，第二項が 1 時点進んだ Q 値となっている．学習が収束してくると第一項と第二項は同じ値になってくるはずである．α は学習率 $(0 < \alpha < 1)$ であり，第一項に対して，第二項をどれだけ更新に用いるかを制御している．

3.3 深層学習の導入

$Q(s, a)$ はすべての状態とすべての行動の組み合わせで定義されている必要がある．そもそも状態 s が非常に多い，または，数え上げが困難な場合，すべての組み合わせを定義するのは不可能である．そこで，深層学習の技術を導入し $Q(s, a)$ をパラメータ θ のニューラルネットワークで関数近似 $(Q(s, a) \approx Q(s, a; \theta))$ することで，すべての (s, a) の組み合わせを定義することを回避する．深層学習を導入した Q-learning は，Deep Q-learning と呼ばれている [3].

損失関数 $L(\theta_t)$ は下式で定義する．

$$\begin{aligned} L(\theta_t) = \mathbb{E}[\{&\underbrace{r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_{t-1})}_{\text{第一項}} \\ &- \underbrace{Q(s_t, a_t; \theta_t)}_{\text{第二項}}\}^2] \end{aligned}$$

第一項は，第二項に対する教師データの役割を果たしているため target と呼ばれ，特に $Q(s_{t+1}, a_{t+1}; \theta_{t-1})$ の部分は target Q-network と呼ばれる．

3.4 学習効率化のテクニック

$Q(s, a; \theta)$ をうまく学習させるための三つのテクニックを紹介する．Deep Q-learning に，これらテクニックを搭載したものは，Deep Q-networks (DQN) と呼ばれる [2, 3].

3.4.1 Experience replay

エージェントの試行錯誤の結果の組 $d_t = (s_t, a_t, s_{t+1}, r_{t+1})$ をメモリー D に記録しておく．損失の計算は，D からランダムにいくつかサンプリングし，ミニバッチという単位の訓練データを構成して行う．式で表現すると次のようになる．

$$\begin{aligned} L(\theta_t) = \mathbb{E}_{\underline{d_t \sim D}}[\{&r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_{t-1}) \\ &- Q(s_t, a_t; \theta_t)\}^2] \end{aligned}$$

下線部 $d_t \sim D$ が，ランダムサンプルを表している．サンプルを増幅させる効果と，サンプル間の相関を排除する効果がある．限られたサンプルを最大限活かし，効率的に学習を進めることができる．

3.4.2 Fixed target Q-network

サンプルごとに θ_{t-1} を更新する方式だと，target が振動してしまい，学習がうまく進まない場合がある．

表 3　エージェントの行動

区分	行動
ヒアリング	1. 家族構成をヒアリング
	2. 予算をヒアリング
提案	3. 1K の物件を提案
	4. 1DK の物件を提案
	5. 1LDK の物件を提案
	6. 2LDK の物件を提案
	7. 3LDK の物件を提案
	8. 4LDK 以上の物件を提案

表 4　環境（カスタマー）の状態とその存在比率

カスタマーの状態	存在比率
家族構成	シングル $= 0.6$
	ファミリー $= 0.4$
予算制約	低価格 $= 0.7$
	高価格 $= 0.3$

表 5　カスタマーの状態と希望間取りの関係

家族構成	予算制約	希望間取り
シングル	低価格	1 K
シングル	高価格	1 LDK
ファミリー	低価格	2 LDK
ファミリー	高価格	3 LDK

そこで，ミニバッチの学習中は θ_{t-1} を固定する．θ_{t-1} を θ^-, θ_t を θ で書き直すと，損失関数は，

$$L(\theta) = \mathbb{E}_{d_t\sim D}[\{r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-) - Q(s_t, a_t; \theta)\}^2]$$

と書け，勾配 $\nabla_\theta L(\theta)$ は，$L(\theta)$ を θ で微分し，

$$\nabla_\theta L(\theta) = \mathbb{E}_{d_t\sim D}[\{r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-) - Q(s_t, a_t; \theta)\}\nabla_\theta Q(s_t, a_t; \theta)]$$

となる（係数 -2 はパラメータ推定に影響がないため省略した）．target Q-network である $Q(s_{t+1}, a_{t+1}; \theta^-)$ は，一つ前までのミニバッチで求められたパラメータにより計算された教師データに相当するので，微分対象ではないことに注意が必要である．ミニバッチ完了後は $\theta^- \leftarrow \theta$ で更新する．ミニバッチ内では，target の一貫性を担保することができ，学習自体が安定することが期待できる．

3.4.3　Reward clipping

報酬の範囲を限定することにより，損失のばらつきを抑え学習を進みやすくする手法である．負の報酬は -1, 正の報酬は 1, 報酬 0 は 0 のように, 報酬を $[-1, 1]$ に限定する．ただし，ゴールの種別ごとに報酬を設定した場合に大小を区別できなくなるデメリットもあり，ケースバイケースで使い分ける必要があると思われる．本稿 4 節にて，その効果について実際に確かめてみることにする．

表 6　観測できる状態の表現

区分	行動
ヒアリング	1. 家族構成をヒアリング
	2. 予算をヒアリング
提案	3. 1K の物件を提案
	4. 1 DK の物件を提案
	5. 1 LDK の物件を提案
	6. 2 LDK の物件を提案
	7. 3 LDK の物件を提案
	8. 4 LDK 以上の物件を提案
ヒアリングへの反応	9. シングルと回答
	10. ファミリーと回答
	11. 低価格と回答
	12. 高価格と回答
提案への反応	13. 閲覧
	14. 無視
	15. 内見予約
	16. サイト離脱

4.　評価実験

以上の理論の有効性を確認するためシミュレーションによる評価実験を行った．実装は OpenAI Gym [5], Chainer RL [6] を利用した．

エージェント（=システム）は，環境（=カスタマー）とのやりとりを繰り返しながら，最適な対話行動を獲得できるかというのが論点である．

シミュレーションのシーンとしては，カスタマーが賃貸物件を探している場面を想定した．エージェントはヒアリングを行い，カスタマーの要望を満たす物件を提案する．カスタマーは，提案された物件について気に入れば，内見予約などの反応，気に入らなければ，サイト離脱などの反応をする．

評価は，訓練フェーズとテストフェーズに分けて行うことにした．

訓練フェーズでは, 学習が収束するまでにどのくらいのエピソード（システムと一人のカスタマーの一連のやりとりの開始から終了まで）が必要なのかと，Reward clipping の効果を確かめることにした．

テストフェーズでは，訓練フェーズで獲得した対話行動がランダムな行動選択と比較してどの程度よいのかを確かめることにした．

評価指標としては，カスタマーが「内見予約」するこ

表 7　評価対象のエージェントの種類

No.	方策	概要
1	ランダム	・各時点でランダムに行動を選択する.
2	完全行動	・環境（カスタマー）の状態を把握しており，毎回希望にあった提案行動を選択する. ・本来このようなエージェントを構成することはできないが，比較のために利用する.
3	DQN (Clip ON)	・$a = \mathrm{argmax}_a\, Q(s, a; \theta)$ により，行動選択を行う. ・訓練フェーズは，ε=0.3 の ε-greedy 方策を採用する. ・テストフェーズは，greedy 方策を採用する. ・Reward clipping を行う.
4	DQN (Clip OFF)	・Reward clipping を行わない以外は，No. 3 のエージェントと同様の設定とした.

表 8　報酬の設定

行動	反応パターン	確率	報酬
ヒアリング	回答する	1.0	0
提案	（希望する間取りと一致している場合）		
	閲覧	0.6	+5
	無視	0.2	−5
	内見予約	0.2	+1000
	サイト離脱	0.0	−50
	（希望する間取りと一致していない場合）		
	閲覧	0.1	+5
	無視	0.5	−5
	内見予約	0.0	+1000
	サイト離脱	0.4	−50

とを対話のゴールとし，下記の内見予約率を採用した.

$$\text{内見予約率} = \text{総内見予約数} \div \text{総エピソード数}$$

4.1　シミュレーション設定

3 節で説明した強化学習の各要素のシミュレーション設定について述べる.

4.1.1　エージェントの行動の設定

エージェントは，表 3 に記載の 8 パターンの行動がとれるとした．本稿では，問題をシンプルにするため，ヒアリング行動は家族構成と予算の 2 パターン，提案行動は，物件の間取りのみの 6 パターン，合計 8 パターンとし，その他バリエーションをもたせないことにした.

4.1.2　環境の状態の設定

エージェントが相互作用する環境であるカスタマーは，家族構成と予算制約の状態をもつとし，表 4 の比率で存在するとした.

たとえば，家族構成が「ファミリー」かつ予算制約が「高価格」であるカスタマーは $0.4 \times 0.3 = 0.12$，すなわち 12%存在する.

カスタマーの希望の間取りは，家族構成と予算制約の状態のみから決まり，表 5 の対応関係を定義した．また，各時点で状態は変化しないことにした.

4.1.3　観測できる状態の設定

エージェントは，上述の環境の状態を直接参照することができず，環境（カスタマー）の反応を観測することで状態を把握するものとした．そこで，状態の観測値は，エージェントが選択した行動と，その行動に対するカスタマーの反応を結合し，表 6 のそれぞれの項目に $\{0, 1\}$ を割り当て，16 次元の one-hot-encoding で表現することにした.

たとえば，時点 t で，エージェントが「家族構成をヒアリング」を選択し，時点 $t+1$ で，カスタマーが「ファミリーと回答」した場合，

$$s_{t+1} = (\underbrace{1,0,0,0,0,0,0,0}_{\text{エージェントが選択した行動}}, \underbrace{0,1,0,0,0,0,0,0}_{\text{カスタマーの反応}})$$

という状態を観測する.

4.1.4　方策の設定

性能を比較するため方策の異なる 4 種類のエージェントを用意した（表 7）．DQN 方策をもつエージェントの $Q(s, a; \theta)$ は，3 層のニューラルネットワークで関数近似した．訓練フェーズでは，ε-greedy 方策 (ε=0.3) を採用した．比率 0.3 でランダムに，比率 0.7 で $\mathrm{argmax}_a\, Q(s, a; \theta)$ によって greedy に行動選択をする．テストフェーズでは，単に greedy 方策を採用した.

4.1.5　報酬関数の設定

エージェントの行動に対する，カスタマーの反応パターンとその確率，報酬を表 8 のように設定した.

さらに，カスタマーの反応が，「内見予約」か「サイト離脱」の場合，その時点でエピソードが終了する．1 エピソードは最大 20 回のやりとりで打ち切られるという制限を設けた.

4.2　評価結果（訓練フェーズの性能）

各エージェントの学習曲線を図 1 に示す．y 軸に内見予約率，x 軸にエピソード数を取っている．エピソード数は 20000 とした．なお，1. ランダム，2. 完全行動のエージェントは訓練の必要がないため，グラフには

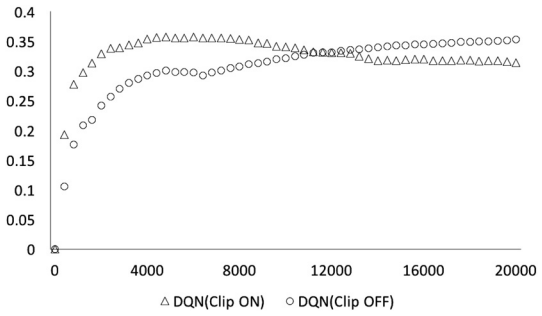

図 1　Reward clipping ON/OFF の違いによる学習曲線の比較

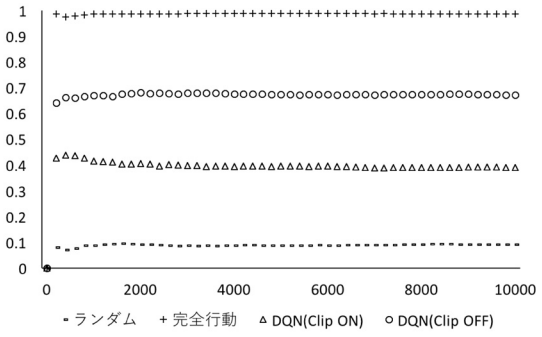

図 2　テストフェーズにおける各エージェントの内見予約率

表示していない.

DQN (Clip ON) エージェントのほうが，立ち上がりのタイミングが早く，素早く対話行動を獲得している様子がうかがえる．しかし，4000 エピソード付近より後は，内見予約率の改善がほとんど止まり，12000 エピソード付近より後は，DQN (Clip OFF) エージェントに追い抜かれている．本稿のシミュレーション設定では，Reward clipping は，学習スピードにはプラスに寄与するものの，最適な対話行動の獲得という面では，やはり Reward clipping OFF のほうが優位であった.

4.3　評価結果（テストフェーズの性能）

テストフェーズの様子を図 2 に示す．さきほどと同様に，y 軸に内見予約率，x 軸にエピソード数を取っている．ここでは，ボトムライン（1. ランダム），トップライン（2. 完全行動）との性能比較を行いたいため，表 7 記載の四つのエージェントの結果を載せた．DQN エージェントは，訓練フェーズで獲得した $Q(s, a; \theta)$ を用いた greedy 方策の結果となっている.

想定どおり，1. ランダム，2. 完全行動のエージェントが下限と上限を与え，DQN (Clip ON) / DQN (Clip OFF) エージェントがその間に収まっているグラフとなっている.

訓練フェーズと同様に，DQN (Clip ON) のエージェントより，DQN (Clip OFF) のエージェントのほうが性能がよく，内見予約率 0.7 付近まで到達していることがわかる．1. ランダムのエージェントの内見予約率が 0.1 程度にとどまることと比較すると，うまく対話行動を獲得できていることが確認できた.

5.　おわりに

本稿では，インターネットメディアに対話機能が求められるようになった背景から，実装における技術的課題を示しつつ，その解決策の一つである深層強化学習を用いたモデリング事例を紹介した．また，評価実験によりその有効性を確認することができた.

しかし，実際のサービスに適用していくためには，まだまだ検討が不足している．シミュレーション設定を現実の複雑度に合わせていくこと，Experience Replay などの学習の効率化 [7]，状態表現や関数近似の洗練 [8]，エージェントの行動パターン自体の自動獲得，行動への制約の与え方など，さまざまなテーマが考えられる.

弊社では，対話メディアの確立に向け，それらの検討・評価を進めている最中である．改めて機会があれば，そのアップデート状況について報告したいと考えている.

参考文献

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv: 1312.5602, 2013.
[2] 藤田康博，最近の DQN，https://www.slideshare.net/mooopan/dqn-58511529（2017 年 6 月 1 日閲覧）
[3] D. Silver, Google DeepMind, Deep Reinforcement Learning, http://www.iclr.cc/lib/exe/fetch.php?media=iclr2015:silver-iclr2015.pdf（2017 年 6 月 1 日閲覧）
[4] 牧野貴樹，澁谷長史，白川真一，浅田稔，麻生英樹，荒井幸代，飯間等，伊藤真，大倉和博，黒江康明，杉本徳和，坪井祐太，銅谷賢治，前田新一，松井藤五郎，南泰浩，宮崎和光，目黒豊美，森村哲郎，森本淳，保田俊行，吉本潤一郎，『これからの強化学習』，森北出版，2016.
[5] OpenAI, OpenAI Gym, https://gym.openai.com（2017 年 6 月 1 日閲覧）
[6] Preferred Networks, ChainerRL, https://github.com/chainer/chainerrl（2017 年 6 月 1 日閲覧）
[7] T. Schaul, J. Quan, I. Antonoglou and D. Silver, "Prioritized experience replay," arXiv: 1511.05952, 2015.
[8] J. D. Williams and G. Zweig, "End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning," arXiv: 1606.01269, 2016.

