The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

強化学習による建築構造最適化の適用性検討

Optimization Method for Structural Engineering by Reinforcement Learning

鈴木　琢也[*1]
Takuya Suzuki

市川　享祐[*2]
Kyosuke Ichikawa

[*1] 竹中工務店
Takenaka Corporation

[*2] アーク情報システム
Ark Info. Sys.

The purpose of this paper is to confirm an applicability of the structural optimization methods by reinforcement learning. First, the outline of the developed application is explained. This application takes advantage of the features of smart devices as to anyone from adults to children can enjoy it. Secondary, the detail of optimization method by reinforcement learning is explained. And then, by proposed method, optimization analyses are conducted. Finally, the optimization result are compared with the result by the traditional method, and the applicability of proposed methods is examined.

1. はじめに

建築物の安全性を支える構造に関する技術の理解には，高度な専門的技術・知識を必要とする。そのため，一般の人々から理解されにくいものとなっている。しかし，技術者が構造解析の結果や意味をよりわかりやすく伝えることができれば，一般の人々はより正しく「安全」を理解し，今以上に「安心」を得ることができると考えられる。「いかにわかりやすく伝えるか」は，建築技術者にとって重要な課題といえる。

一方で，近年スマートフォンやタブレット端末といった高性能な携帯型情報端末（以下，「スマートデバイス」という）が広く普及している。スマートデバイスは，タッチパネルによる直感的なインターフェースや，各種センサーを内蔵することにより，これまでとは異なる計算機と人との関わり方を可能にしている。

このような背景から，著者の一人は，構造解析をよりわかりやすくするためのツールとしての可能性をスマートデバイスに見出し，これまでに，スマートデバイスの特徴を活かした建築構造解析アプリケーションに関する調査研究を行ってきた[鈴木 2017]。

本報では，このタッチで出来る構造最適化アプリケーションにおいて利用する対戦コンピューターの戦略ロジックを決定するために，新たに強化学習による構造最適化手法を提案し，その適用性検討結果を報告する。

2. 開発するアプリケーションの概要

タッチで出来る構造最適化アプリケーションであるStartOPT[鈴木 2017]を対戦型に改良したものである[鈴木 2018]。

図 1 にはゲーム画面を示す。ボールを支えるのに不要と思われる部分をタッチで消した後，残ったブロックのみで変形解析が行われる。2 人のプレイヤーがブロック（最大 10 個，パスも可）を交互に消していき，先にボールを規定ラインよりも下に落としたプレイヤーが負けとなる。その点では，子どもの遊びである砂山崩しに近い。ただし，両者がパスした場合には，その時点で，より多くのブロックを消したプレイヤーの勝ちとなる。

様々な荷重条件，境界条件下における強い形を数多く試すことを通して，効率的な構造を直感的に理解するとともに，建築

連絡先：鈴木琢也，竹中工務店，千葉県印西市大塚 1-5-1，
0476-47-1700，0476-47-6460，suzuki.takuya@takenaka.co.jp

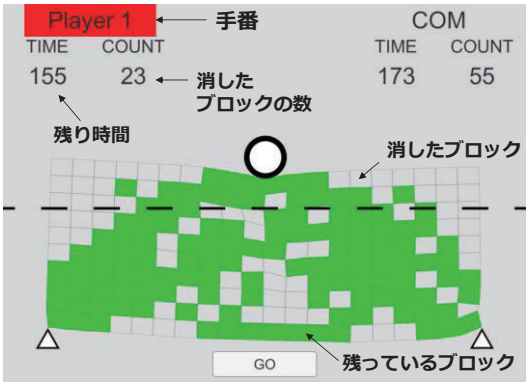


図 1　ゲーム画面

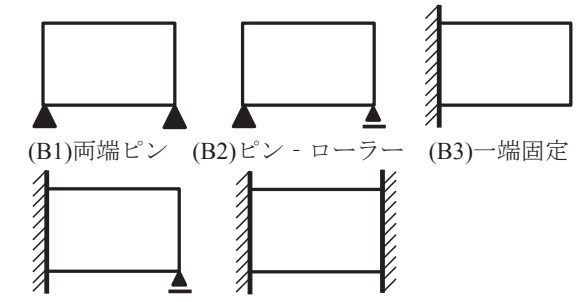


図 2　境界条件

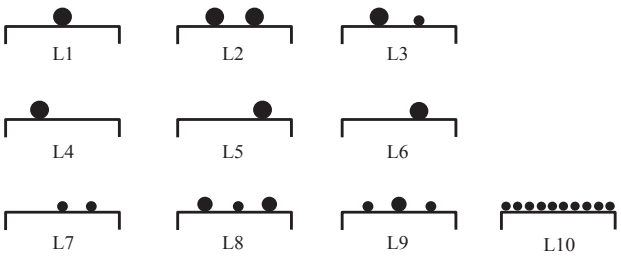


図 3　荷重条件

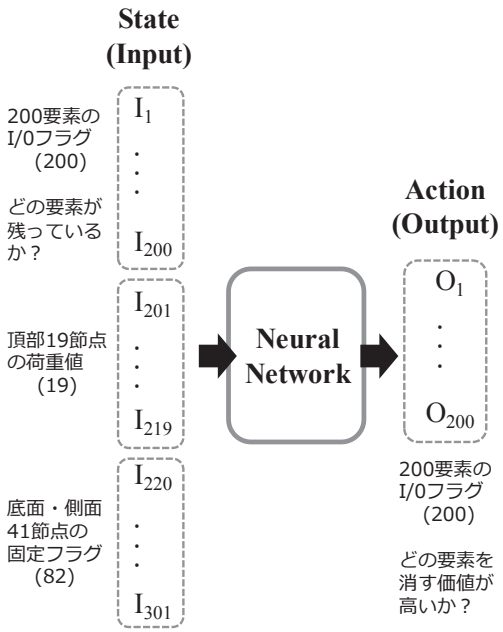


図 4　Neural Network の入出力

構造の基礎である「力の流れ」を，老若男女問わず楽しく理解することができる。

各ステージは，ブロックの境界条件（どこが支えられているか），および荷重条件（ボールの位置，個数，重さ）が変化する。

境界条件は，図 2 に示す建築構造で基本となる 5 パターン，荷重条件は図 3 に示す 10 パターン用意し，これらを組み合わせた計 50 種のステージが用意されている。これらの境界条件，荷重条件に応じて，最適な形状も様々に変化することになり，多くのステージをプレイすることで，プレイヤーの理解度も深まっていく。

3. 学習計画

与えられた状態（State）から各行動の行動価値 Q を算定し，次の行動（Action）を決定するエージェントを強化学習によって構築する。

この場合，行動価値 Q を正しく見積もることが重要である。行動価値 Q は収益の期待値であり，厳密には次のベルマン方程式で算出できる。

$$Q(s,a)=\sum_{s'}P(s'\mid s,a)\{r(s,a,s')+\gamma\sum_{a'}\pi(a'\mid s')Q(s'\mid a')\} \quad (1)$$

ここで，P(s'|s,a)は状態遷移確率であり，状態 s において行動 a を決定したとき，状態が s'に遷移する確率である。この確率は未知であるため，この値を直接使わずに行動価値を求める必要がある。その方法として，Sarsa や Q-learning が知られている[牧野 2016]。Q-learning では，収束先を(1)式ではなく，次の最適ベルマン方程式とする。

$$Q^*(s,a):=\sum_{s'}P(s'\mid s,a)\{r(s,a,s')+\gamma\max_{a'}Q^*(s'\mid a')\} \quad (2)$$

そして，次式で行動価値 Q を更新する。

$$Q(s_t,a_t)\leftarrow(1-\eta)Q(st,at)+\eta\{r(s_t,a_t,s_{t+1})+\gamma\max_{\tilde{a}\in A}Q(s_{t+1},\tilde{a})\} \quad (3)$$

ここで，η(0<η<1)は学習率と呼ばれる微小なパラメータである。行動決定と状態遷移が行われるたびに，この式によって更新する。現状値の $Q(s_t,a_t)$と，目標値の $r(s_t,a_t,s_{(t+1)})+\gamma\max Q(s_{(t+1)},a)$との内分によって，Q を更新している。状態遷移確率 P(s'|s,a)を使用しない代わりに，この計算の頻度（更新頻度）で状態遷移確率 P(s' |s,a)が表現される。

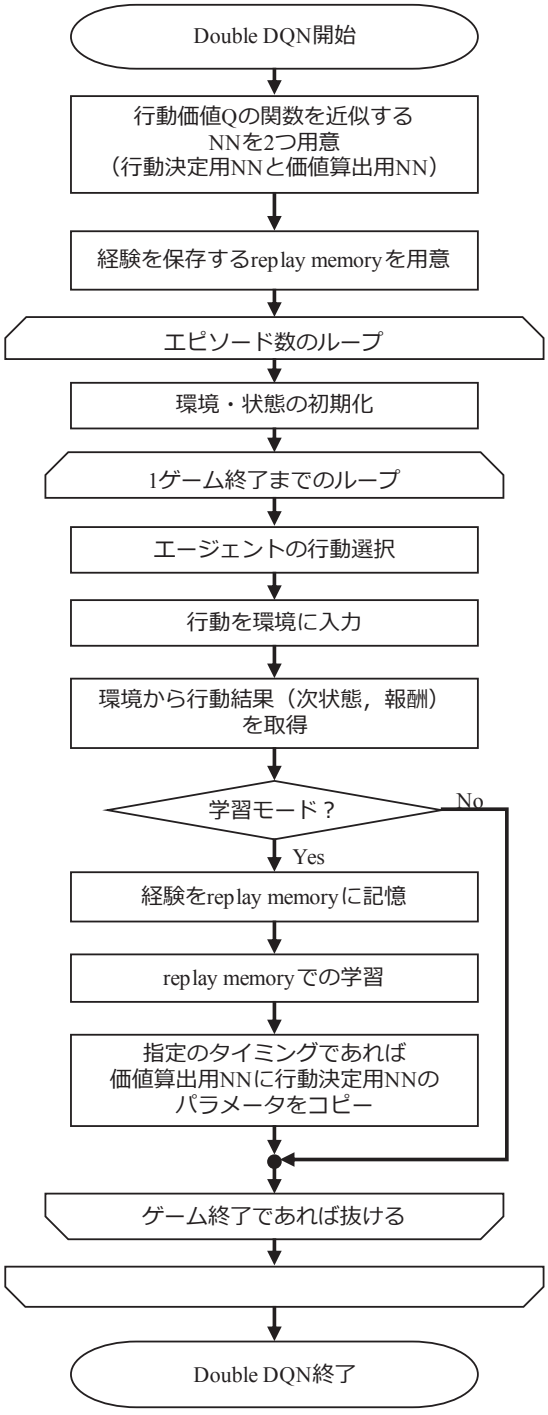


図 5　学習のフロー

さて，行動価値 Q は Q-learning によって更新可能だが，この関数をテーブルとして保存すると，状態数と行動数の組み合わせだけ行動価値が存在し，膨大なメモリが必要となる。そこで，この関数をニューラルネットワーク（以降,NN）で近似する（図 4）。この手法は Deep Q Networks[Mnih 2013]（以下，DQN）と呼ば

れる。ただし，本作業では改良型の Double DQN[Hasselt 2015] と呼ばれる手法を用いた。図 5 にそのフローを示す。

エージェントは，一般的に最適行動とは最も行動価値の高い行動を選択する。しかし，今回の問題では既に削除されたブロックを削除することはできないため，次の修正された行動価値 $\tilde{Q}$ を求め，それが最も大きい行動を最適行動とした。

$$\tilde{Q}(s_t,a_t) := Q(s_t,a_t) - \lambda(s_t,a_t)\max Q(s_{t+1},\tilde{a}) \quad (4)$$

ただし，λは行動 a_t で選択されたブロックが削除済であれば 1，未削除であれば 0 となる関数である。これにより，削除済のブロックに対する行動価値は最小となる。また，ランダムに行動する際も，削除済のブロックを削除する行動は選択されない仕様にした。

4. 検討結果

深層強化学習の検証を行った。本検証では，エージェント同士の対戦を行い学習させた。ただし，エージェントは複数用意するのではなく，同一のものを使用した。検証ケースを表 1 に示す。中間層および中間層のノード数をパラメータとし，計 3 ケースについて実施した。なお，プレイするステージは，境界条件 B1，荷重条件 L1 のステージ 1 としている。

また，全ケース共通のハイパーパラメータを表 2 に示す 。

テスト(推論)は 10 エピソードに 1 回実施し，報酬和が最大になったエピソードの学習結果を保存するようにした。また，100 エピソード毎に学習結果を保存するようにした。

図 6 には学習後のエージェントによるゲームのプレイ結果を示す。

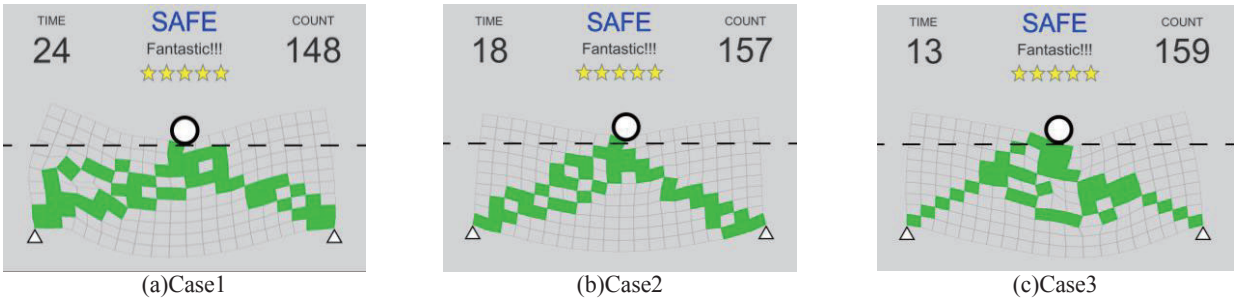
図より，ノード数，中間層数の少ない Case1 に比べて，ノード層の多い Case2 や，中間層数の多い Case3 の方が高得点を記録していることがわかる。ノード数，中間層数を増やすことでより複雑な行動判断が可能になった結果，高得点を記録できるエージェントが作成できたと考えられる。一方，Case2 と Case3 を比較すると，Case3 の方が高得点を記録できており，中間層数

表 1 解析ケース一覧

Case No.	中間層 層数	中間層 ノード数
1	1	1000
2	1	2000
3	2	1000

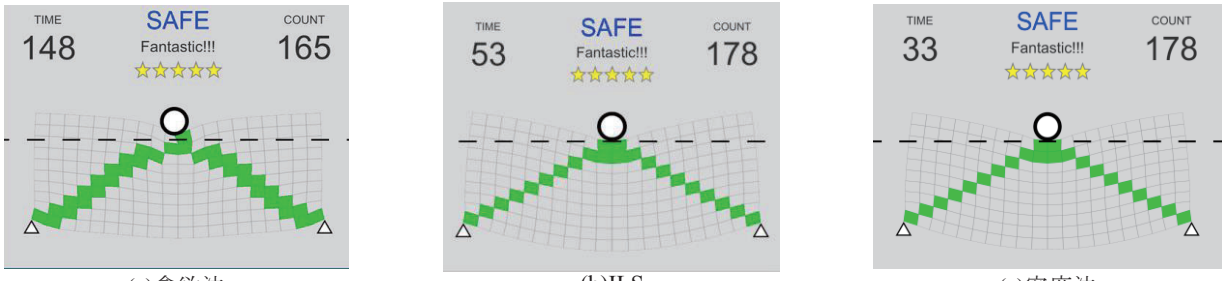
表 2 Hyper Parameters

項目	設定
エピソード数	5000
割引率 γ	0.997
ミニバッチサイズ	32
初期学習率 η	1.0E-05
重み行列初期値の 標準偏差	0.01
replay memory のサイズ	104 の経験分
学習を開始する replay memory のサイズ	ミニバッチサイズ
価値算出用 NN の 更新間隔	経験を 1000 格納する毎に 1 回
ε (学習時ランダム行動の確率)	Episode 毎に 線形に減衰 Episode 1: ε=1.00 Episode 2000: ε=0.05 以降，ε=0.05
報酬関数	両者の消した ブロック数



(a)Case1　(b)Case2　(c)Case3

図 6　強化学習による最適化結果



(a)貪欲法　(b)ILS　(c)密度法

図 7　既存手法による最適化結果 [鈴木　2018]

を増やすほうがより効果的であるといえる。

また，図 7 には比較のため，[鈴木 2018]に示された既存の最適化手法の結果を示す。図より，今回実施した，強化学習による方法に比べて，既存の最適化手法の方がよりよい結果を示していることがわかる。強化学習による方法で既存の最適化手法よりも良い結果を出すためには，今回よりも複雑な NN を用いるなどの改良が必要であるといえる。

5. まとめ

本報では，タッチで出来る構造最適化アプリケーションにおいて利用する対戦コンピューターの戦略ロジックを決定するために，新たに強化学習による構造最適化手法を提案し，その適用性について検討した。

その結果，一定の精度で最適化形状に至るエージェントを作成できたものの，既存の最適化手法に比べるとやや劣る結果となった。より精度の高い最適化を行うエージェントを作成するには，今回よりも複雑な NN を用いる等の改良が必要と考えられる。

参考文献

[鈴木 2017] 鈴木琢也ほか:スマートデバイスの特徴を活かした教育用構造解析アプリケーションの開発と展開，日本建築学会学術講演会梗概集，教育，pp. 47-48, 2017.7

[鈴木 2018] 鈴木琢也ほか:建築構造教育アプリケーションに対する各種最適化手法の適用性検討，JSAI2018，教育，3Pin1-01, 2018.6

[牧野 2016] 牧野貴樹, 澁谷長史, 白川真一, 他: これからの強化学習, 森北出版, 2016.

[Mnih 2013] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves I. Antonoglou, D. Wierstra, M. Riedmiller: Playing Atari with Deep Reinforcement Learning, arXiv:1312.5602, 2013.

[Hasselt 2015] H. van Hasselt, A. Guez, and D. Silver: Deep Reinforcement Learning with Double Q-learning, arXiv: 1509.06461, 2015.