交通信号制御へのDeep Q-Networkの適用

佐藤 季久恵 [1,a)]　高屋 英知[2]　小川 亮[2]　芦原 佑太[2,3]　栗原 聡[2]

概要：近年，都市部にて発生する交通渋滞は，ドライバーの時間的損失だけでなく，輸送の遅延や燃料消費の増加に伴う経済的損失を引き起こしている．さらに排気ガスによる大気汚染や騒音，追突事故等の主要な要因としても指摘されている．交通渋滞の解消における主なアプローチとして，適切なナビゲーションにより交通流の分散を図るアプローチや信号制御によりスムーズな交通流を生みだすアプローチがある．今回は後者について交通渋滞を解消する手法を提案する．信号制御の手法として，これまでにもGAやマルチエージェントなどによる手法が提案されている．いずれの先行研究も高次元な交通情報からあらかじめ必要な情報を定め，その情報をシステムの入力値として与えている．しかし，あらかじめ定められていない情報にも信号制御に重要な情報が含まれている可能性があり，交通に関する高次元な情報量から信号制御を行うために必要不可欠な特徴をいかに抽出するかが課題となっている．本研究では，道路交通画像という高次元な情報からエージェント自身が信号制御に必要な情報を抽出し，適切な信号機のパラメータ操作を出力することを目的とする．そこで，高い特徴抽出能力を持つ深層学習法と，報酬に基づいた最適な行為を学習する強化学習法を組み合わせたDeep Q-Networkを用いた制御手法を提案する．その結果，エージェントに道路交通画像を与えると，エージェント自身が学習し，効率的に信号制御できることが確認された．

キーワード：ITS，信号制御，Deep Q-Network

1. はじめに

都市部で発生する交通渋滞は，ドライバーの時間的損失だけでなく，輸送の遅延や燃料消費の増加に伴う経済的損失を引き起こしている．さらに排気ガスによる大気汚染や騒音，追突事故などの主要な原因としても指摘されている．交通量が所与のもとで交通渋滞を防ぐ手立てとして，適切なナビゲーションにより交通量の分散を図るアプローチや信号制御によりスムーズな交通流を生み出すアプローチなどがある．本研究では，後者の信号制御に焦点を当てる．

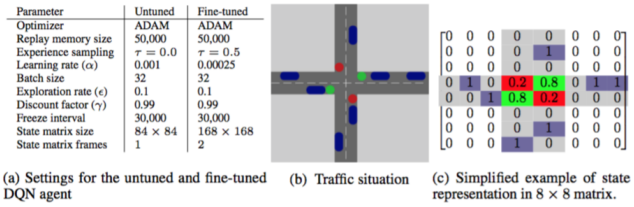
一般道路の場合，交通渋滞が起こる原因の一つが交差点である．交差点では，信号機により車両の通過を制御することによって交通流を円滑にする．一方，信号制御の仕方によっては，渋滞を発生させる原因ともなりうる．そこで，交通流を制御するには，信号機を制御するパラメータ値を適切に操作する必要がある．一般的な交通流は，朝夕のラッシュ時と昼間で交通量が大きく異なるように，時間の経過とともに変化する．しかし，多くの信号機では，平常時の交通量から算出したパラメータのパターンを複数用意し，時間帯によってそれらを使い分ける静的な信号制御方法が採用されている．そのため，交通事故やイベントの開催など，突発的な交通流の変化に対応できない．したがって，時々刻々と変化する交通流に合わせ，パラメータの操作を適切に行う必要がある．そのためには，ある時点での交通流の特徴を的確に抽出し，それに対する最適なパラメータ操作を得ることが重要である．

そこで．高い特徴抽出能力を持つDeep Learningと，報酬に基づいた最適な行動を学習する強化学習を組み合わせたDeep Q-Network(DQN)に注目する．DQNは，家庭用ゲーム機Atari2600のブロック崩しやスペースインベーダーなどの多くのゲームにおいて，映像を入力として与えるだけで人間より高いスコアをだすことに成功している[1]．本研究では，このDQNを応用し，信号機を動的に制御するシステムを提案する．

2. 関連研究

信号制御に関する研究は様々な角度から行われている．高橋ら[2]は，各交差点に存在する信号機をエージェン

[1] 電気通信大学 情報理工学部 総合情報学科 情報学専攻
The University of Electro-Communications
[2] 電気通信大学 大学院 情報理工学研究科 情報学専攻
The University of Electro-Communications
[3] 株式会社クロスコンパス・インテリジェンス
XCompass Ltd.
[a)] ksato@ni.is.uec.ac.jp



Parameter	Untuned	Fine-tuned
Optimizer	ADAM	ADAM
Replay memory size	50,000	50,000
Experience sampling	$\tau = 0.0$	$\tau = 0.5$
Learning rate (α)	0.001	0.00025
Batch size	32	32
Exploration rate (ϵ)	0.1	0.1
Discount factor (γ)	0.99	0.99
Freeze interval	30,000	30,000
State matrix size	84×84	168×168
State matrix frames	1	2

(a) Settings for the untuned and fine-tuned DQN agent

(b) Traffic situation

(c) Simplified example of state representation in 8×8 matrix.

図 1 車の位置を行列にした入力値※ VAN らの研究 [5] より引用.

トとみなし，エージェント同士が協調することで自律的に信号制御を行う手法を提案した．西原ら [3] は，「青，黄，赤」と表示される信号機をマルチエレメント GA を用いて適切に変化させ，効率的に信号制御を行う手法を提案した．Liu ら [4] は，隣接交差点の交通情報と Q-learning を用いて信号機のパラメータを制御し，効率的に交通量を改善することができることを示した．これらの研究では，高次元な交通情報から信号制御に必要な情報を人手で抽出したものをシステムの入力値として与えている．しかし，人手による特徴抽出の過程で，信号制御を行う際に有効な情報を配慮してしまっている可能性もある．

また，Van ら [5] は，Deep Q-Network を信号制御に適用し，入力として車の待ち行列を用いて信号制御が行えることを示した．VAN らの研究は本研究と類似しているため，ここで，本研究との違いについて詳しく述べる．VAN らは，車を 2 進数で，信号の状態を数値で表した行列を入力値とし，この行列値をもとに学習を行っている (**図 1**(b),(c) 参照)．入力画像を学習させるための報酬関数は，車の信号待ち時間や信号が切り替わる回数など，複数の要素を考慮して設計している．さらに，複数の交差点に対して提案手法を個々に適用し，互いに協調することによるマルチエージェント学習も試みている．しかし，入力画像を行列値で表すこともまた，人手による特徴抽出であるため，先述した問題を含んでいる．また，行列に変換する手間も生じる．

そこで，本研究では，これらの問題を解消するために，画像という高次元な情報からエージェント自身が信号制御に必要な特徴を抽出し，適切なパラメータ操作を出力することを目的とした，DQN を用いた交通信号制御システムを提案する．なお，パラメータや報酬の設定の仕方は，VAN らの研究を参考にした (図 1(a) 参照)．

3. Deep Q-Network(DQN)

Deep Q-Network とは，Deep Learning と Q-learning を組み合わせた技術である．本章では，DQN で用いられるアルゴリズムについて，Q-learning，NeuralNetwork による近似，Deep Learning の順に解説する．

3.1 強化学習 (Q-learning)

強化学習とは，エージェント (学習者) が環境 (制御対象) に対し試行錯誤しながら報酬を得るために行動し，学習を行う枠組みである [6]．本研究では，強化学習の中で代表的な手法である Q-learning を扱う．Q-learning では，エージェントが現状態 s_t から最終状態 s_T までの累積報酬

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

の期待値

$$Q(s_t, a_t) = \mathbb{E}\left[R_t | s_t = s, a_t = a\right] \tag{2}$$

が最大となるような行動 a_t

$$\pi^*(s) = \mathrm{argmax}_a Q^*(s, a) \tag{3}$$

を選択する．式 3 を方策 π^* と呼ぶ．ここで，T は最終時刻，$\gamma (0 \leq \gamma \leq 1)$ は割引率を表す．Q-learning は，得られた報酬 r と次の状態 s' の最大累積報酬の和であるから

$$Q(s, a) = r + \gamma \mathrm{max}_{a'} Q(s', a') \tag{4}$$

と表せる．実際の Q-learning では，式 4 から実際に得られる累積報酬和の期待値の差

$$r + \gamma \mathrm{max}_{a'} Q(s', a') - Q(s, a) \tag{5}$$

を求め，次の状態に移るごとに

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma \mathrm{max}_{a'} Q(s', a') - Q(s, a)) \tag{6}$$

により値を更新していく．ここで，$\alpha (0 \leq \alpha \leq 1)$ は学習率を表す．

このように，常に最適な行動を選択していくことは可能であるが，常に累積報酬の期待値が最大となる行動をとり続けると，状況によっては過学習を起こしてしまう可能性がある．そこで，これまでの学習結果から行動する利用 (exploitation) とランダムに行動する探索 (exploration) をバランス良く行う ϵ-greedy 法を用いる．ϵ-greedy 法は，確率 1-$\epsilon (0 \leq \epsilon \leq 1)$ で累積報酬の期待値が最大となる行動を選択し，確率 ϵ でランダムに行動を選択する方法である．学習開始時は ϵ=1 とし，学習が進むにつれ $\epsilon \to 0$ に近づける．

これまで述べてきた手法は，状態が離散値をとるという条件で十分な学習を行うことで収束するが，状態が連続値をとるような環境では，ϵ-greedy 法を用いて行動を選択しても収束するのに時間がかかってしまう．連続な状態空間を持つ事象を扱う場合は，一般に関数近似が用いられる．今回は，高次元のデータを扱うことに長けている，Neural Network による近似を用いる [6].

3.2 Neural Network による近似

$Q(s, a)$ に Neural Network による関数近似を行うことを考える．そこで．$Q(s, a)$ が近似された関数をパラメータ θ

を用いて $Q_\theta(s,a)$ と表す．この関数を用いて Q 学習に相当する関数を近似するには，勾配法によるアルゴリズム

$$\theta \leftarrow \theta - \alpha \nabla_\theta L_\theta \tag{7}$$

を用いる．ここで L_θ は 2 乗誤差を $\frac{1}{2}$ 倍した誤差関数

$$L_\theta = \frac{1}{2}\Big(target - Q_\theta(s,a)\Big)^2 \tag{8}$$

で表される．真の行動価値が分かれば，教師あり学習として $target$ =Q*(s,a) とすれば良いが，強化学習では教師信号に当たる Q*(s,a) が得られるわけではない．そこで，Q*(s,a) の代わりに

$$target = r + \gamma \max_{a'} Q_\theta(s',a') \tag{9}$$

を教師信号として用いる．ただし，s' は状態 s の次の状態，a' は状態 s' のときの行動を表す．この式 9 を微分して得られる誤差伝搬する際の勾配は

$$\nabla_\theta L_\theta = -\Big(\mathrm{target} - \mathrm{Q}_\theta(\mathrm{s},\mathrm{a})\Big)\nabla_\theta \mathrm{Q}_\theta(\mathrm{s},\mathrm{a}) \tag{10}$$

となる． 式 10 を式 7 に代入すると，近似関数を用いた Q 学習は

$$\theta \leftarrow \theta + \alpha\Big(\mathrm{target} - \mathrm{Q}_\theta(\mathrm{s},\mathrm{a})\Big)\nabla_\theta \mathrm{Q}_\theta(\mathrm{s},\mathrm{a}) \tag{11}$$

となる．

しかし，多層ニューラルネットワークのような非線形関数を使った近似は，学習が遅く，発散してしまうなどの問題がある．この現象を解決するため下記の 2 つの手法が提案されている ([7],[8]).

(1) experience replay[7]
エージェントが学習に用いるサンプルの偏りを抑制するために，エージェントが経験した (s_t, a_t, r_t, s_{t+1}) を Memory 表に保存し，保存した値からランダムに (s_t, a_t, r_t, s_{t+1}) を選び出す．そして，選び出した (s_t, a_t, r_t, s_{t+1}) を教師データとし，Q 値を更新する．また，Memory 表は，(s_t, a_t, r_t, s_{t+1}) を 10,000 組保存できる構造になっており，随時更新していくことが可能となる (**図 2** 参照).

(2) neural fitted Q[8]
experience replay で更新された Q 値 (式 4) を毎回反映してしまうと収束が安定しない．そこで，10,000step のタイミングで Q 値の更新を行う．

3.3 Deep Learning

本節では，画面の特徴抽出を行う Convolutional Neural Network(CNN) について解説する．CNN は，人の顔の認識や道路標識の認識など，様々な物体を認識することのできるネットワークである．**図 3** に，CNN の構造を示す．入力データを与えた後，畳み込み層で画像の特徴を抽出し，

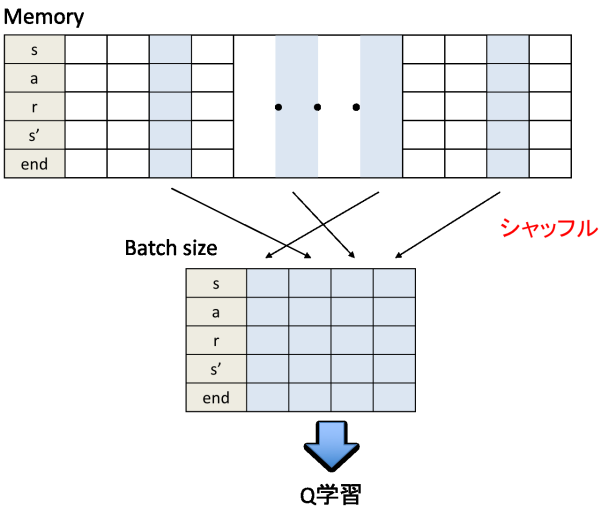


図 2 experience replay の仕組み

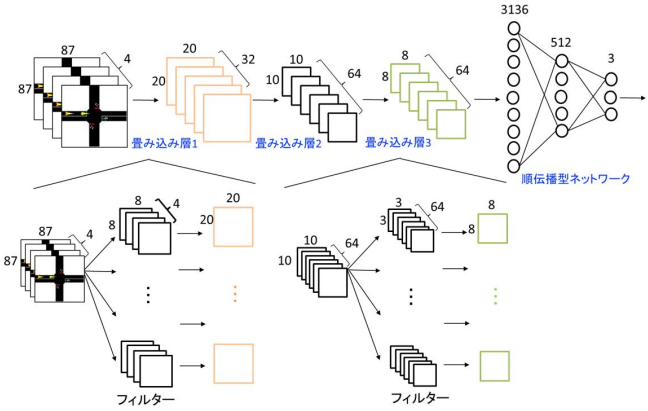


図 3 CNN の構造

最終層では順伝播型ネットワークにより画像の分類を行う．畳み込み層では，入力に対し，フィルタ (重み) を畳み込む計算を行う．即ち，入力画像をフィルタが表す特徴的な濃淡パターンにより畳み込み，画像をぼかしたり，エッジを協調するなどして画面の特徴をつかむ．最後に，3136 の出力を順伝播型ネットワークにより全結合し，画像から読み取れた最適な行動を選択する．なおフィルタ (重み) は，誤差逆伝播法により更新する．

4. 提案手法

Q-learning は，与えられた環境に対しエージェントがより多くの報酬を得るために試行錯誤しながら行動し，学習を行う仕組みであった．本研究では，環境として交通シミュレータを，エージェントとして DQN を用いる．まず，環境とエージェント，それぞれの設定について述べる．

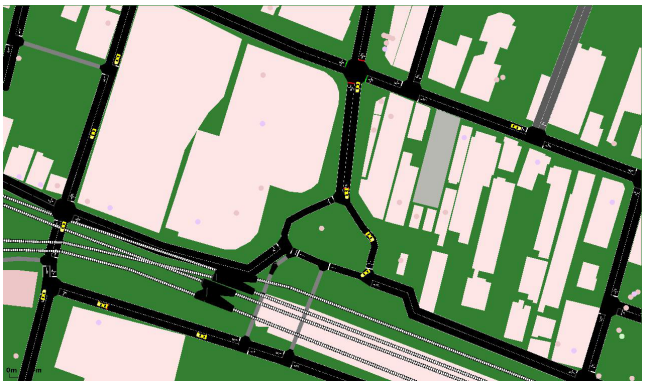


図 4　調布駅周辺を示したシミュレーション画像

表 1　実験に用いる環境設定

車の頻度	1 台 / 3 step
車の通過率	順路 1：順路 2 ＝ 1：5
順路 1 の距離	300 m
順路 2 の距離	300 m
入力切り替え頻度	1 回 / 4step

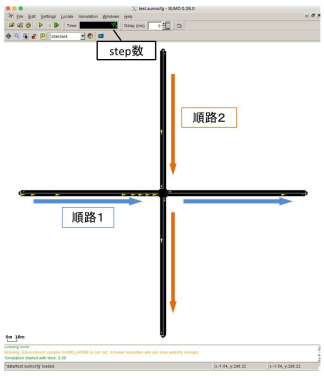


図 5　シミュレーションの全体像

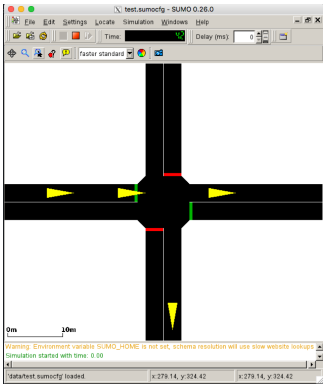


図 6　入力画像

表 2　入力操作

入力操作	機能
0	前の機能を継続
1	rgrg
2	grgr

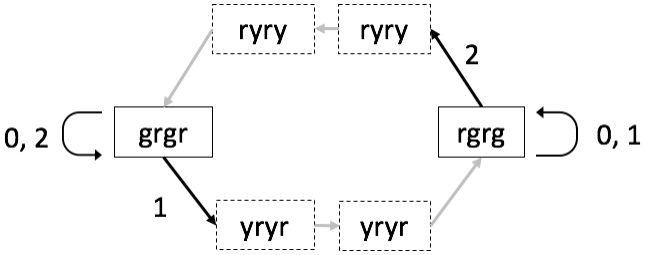


図 7　状態遷移図

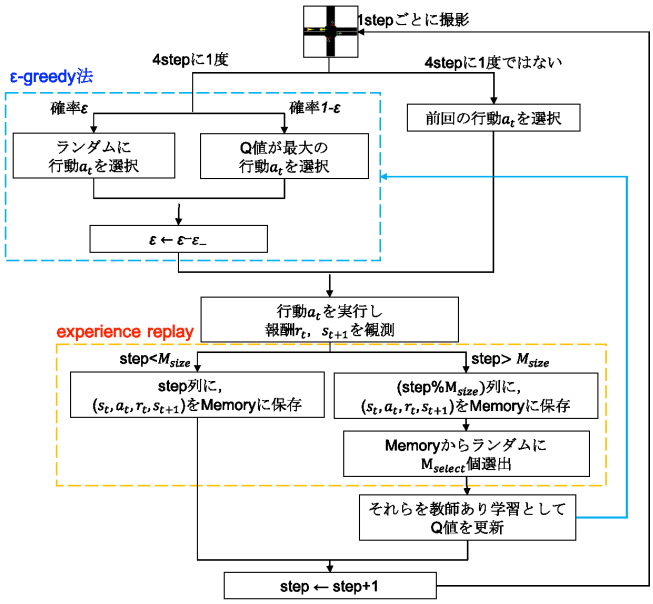


図 8　エージェント (DQN) のアルゴリズム

4.1　環境設定

4.1.1　ミクロ交通流シミュレータ SUMO(Simulation of Urban MObility)

本研究では，ミクロ交通流シミュレータ SUMO を用いて実験を行う．SUMO はドイツ航空宇宙センターを中心に開発されているオープンソースのシミュレータである．車の速度，加速度，走行ルートや信号機や道路の設置など，自由に設定することが可能である．また，**図 4** のように，実際の道路環境を表現することも可能である．

4.1.2　道路ネットワークおよび入力画像

上述した実験に用いるシミュレーションの環境設定を**表 1** にまとめる．車は 3step に 1 台の頻度で出現し，左端から右端 (順路 1) と，上端から下端 (順路 2) までの距離をそれぞれ 300m とし，車の通過率を順路 1：順路 2 = 1：5 とした環境を設定する．なお，順路 1，順路 2 はいずれも直進のみ行うものとする (**図 5** 参照)．本研究では，図 5 を拡大させた，**図 6** を入力画像とし，この画像をもとに学習を行う．

4.1.3　信号機の操作方法

信号機の入力操作およびその機能を**表 2** に示す．信号機は 4step に 1 度，入力操作 0，1，2 のいずれかの操作に切り替える．ここで，信号機の状態の表記の仕方として，例えば図 6 では，信号機の状態から上から時計回りに「red, green, red, green」となっているので，rgrg とする．なお，黄色信号 (yryr，ryry) は，信号機の状態が入力操作 1(rgrg) あるいは 2(grgr) に変更される際に自動で切り替わる仕組みになっている．例えば，入力操作 1(rgrg) を入力した 4step 後，入力操作 2(grgr) を入力した際は，2step 間 ryry の操作を経て入力操作 2(grgr) に切り替わる．

4.2　エージェント (DQN) 設定

次に，エージェント設定について述べる．1step ごとに SUMO の画像を撮り，その画像を 87 × 87(pixel) にリサイズし，グレースケールに変換したものを入力画像として与える．その画像を 4 枚集めて 1step ごとにエージェントに

表 3 パラメータ設定

パラメータ	値
α	0.00025
γ	0.99
ϵ_o	1
ϵ_-	0.0001
ϵ_e	0.1
M_{size}	10000
M_{select}	32
$Delay$	0 or 1 or 2
$Q_{frequency}$	10000

与え，それらの画像を元に ϵ-greedy 法 [*1] により行動 a_t を選択し，実行する．その後，r_t,s_{t+1} の値を得，Memory 表 (図 2 参照) に保存する．Memory 表に保存する際の注意点として，(s_t,a_t,r_t,s_{t+1}) が Memory 表に埋まっていない場合は，Memory 表の step 列目に (s_t,a_t,r_t,s_{t+1}) を保存する．Memory 表にデータが埋まっている場合は，step 数をパラメータ M_{size} で割った余りの値の列に，(s_t,a_t,r_t,s_{t+1}) を更新する．その後，Memory 表の中からランダムにパラメータ M_{select} 個選び，それらのデータをもとに Q 値を更新する．その後，次の step に移る (**図 8** 参照).

4.3 CNN の構造

CNN の内部構造は，図 3 に示す通りである．入力画像として SUMO の画像を与え，畳み込み層により画面の特徴を抽出し，Q-learning を行う．畳み込み層では，入力画像をフィルターで畳み込み，画像をぼかしたり，エッジ (色が変化する境目) を強調するなど，画像の特徴を捉える．これを 3 回繰り返した後，順伝播型ネットワークにより信号操作の 0，1，2 それぞれの Q 値を出力する．Q-learning では，CNN で得られた Q 値を元に，画面の状態に対して最も報酬が得られると期待される行動を予測し，選択する．そして，Memory 表から選出されたデータを用いて教師あり学習を行う．

4.4 パラメータ設定

本実験で扱うパラメータを**表 3** にまとめる．α, γ は Q-learning(3.1 章)，M_{size}, M_{select} は experience replay(4.2 章) で用いるパラメータである．ϵ_o, ϵ_-, ϵ_e は ϵ-greedy 法 (3.1 章) で用い，ϵ_o は ϵ の初期値を，ϵ_- は *epsilon* の値を 1step ごとに減らす値を，ϵ_e は *epsilon* の最小値を表す．$Q_{frequency}$ は neural fitted Q(3.2 章) で重みを更新するタイミングのことを表し，*Delay* は，ある状態での行動に対し数 step 先の報酬を与えるタイミングを表す．これらのパラメータは値を変更することが可能である．

[*1] ϵ の確率で無作為に選択し 1-ϵ の確率で式 3 に基づき行動を選択

5. 実験結果とその評価

車の待ち台数 (waiting count) や待ち時間 (waiting time) を負の報酬とし，Delay の値を変えて実験を行った (**図 9**, **図 10** 参照)．waitng count を報酬とした実験では 2100 エピソード (1 エピソードは 200step に等しい)，waitng time を報酬とした実験では 1500 エピソードのシミュレーションを行った．グラフは，報酬を与える時間をそれぞれ 0,1,2step でチューニングした結果である．

5.1 実験結果 (waiting count を報酬)

図 9 をみると，400 エピソードを境に waiting count が減少しているので，学習能力が発揮されていることが分かる．また，*Delay* に着目すると，収束の早さ，収束したときの最大値の観点から，*Delay*=1 のとき最も精度がよいといえる．

5.2 実験結果 (waiting time を報酬)

図 10 をみると，400 エピソードを境に waiting time が減少しているので，学習能力が発揮されていることが分かる．また，*Delay* に着目すると，どの実験も収束できているが，収束の早さから *Delay*=1 のとき最も精度がよいといえる．

5.3 静的な信号制御との比較

次に，DQN を用いた信号制御の精度を評価するために，車の交通流に合わせた静的な環境を用意し，DQN による手法で最も精度のよかった下記の 2 つの提案手法と比較手法 (静的な信号制御) をそれぞれ比較する．

- 提案手法 1
  waiting count を負の報酬，*Delay* を 1 に設定.
- 提案手法 2
  waiting time を負の報酬，*Delay* を 1 に設定.
- 比較手法 (静的な信号制御)
  比較手法として一定の間隔で信号機が切り替わる静的な信号制御を用いる．そのため，車の流量，黄色信号の時間等の道路環境は，提案手法と同じとし，信号機のサイクル [*1] は 50step とする．また，信号機のスプリット [*2] は車の通過率に合わせ，順路 1 を 17%, 順路 2 を 83%とする．これらの実験設定の詳細を**表 4** に示す.

d

5.4 提案手法 1,2 の評価

提案手法 1 では学習が収束している 1300 エピソード以

[*1] 信号機が赤→黄→青へと 1 巡する時間をサイクルと呼ぶ.
[*2] 各信号で通行権を割り当てられる時間配分をスプリットと呼び，秒または%で表す.

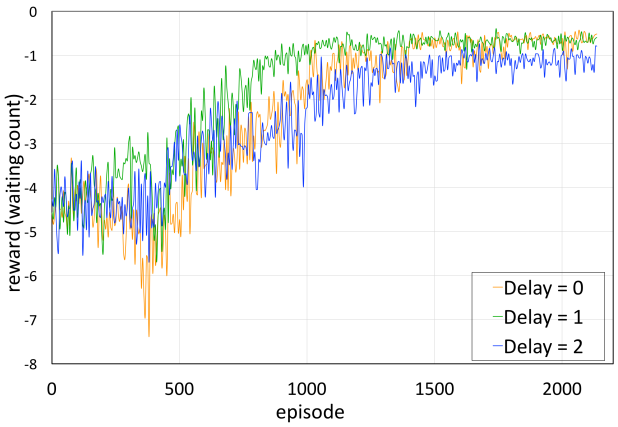


図 9 reward(waiting count)

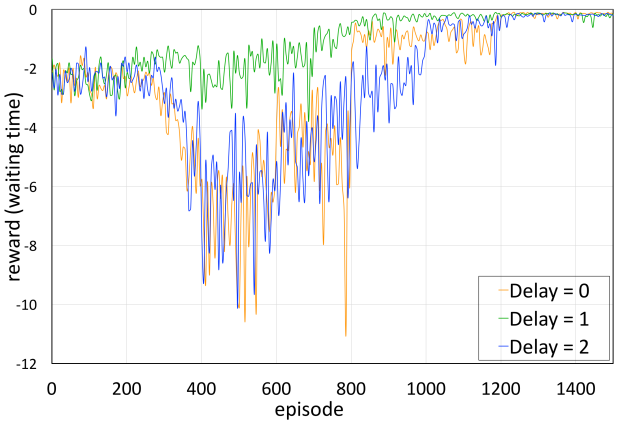


図 10 reward (waiting time)

表 4 静的な信号制御の環境設定

車の頻度	1 台 / 3 step
車の通過率	順路 1：順路 2 = 1：5
順路 1, 順路 2 の長さ	300 m
黄色信号の時間	2 step
1 エピソード あたりの step 数	200 step
スプリット	順路 1=17%, 順路 2=83%
1 エピソード間の cycle 数	10 cycle
grgr の step 数	2.65 step
yryr の step 数	2 step
rgrg の step 数	13.35 step
ryry の step 数	2 step

降のデータを抽出し，提案手法 2 で 850 エピソード以降のデータを抽出し，そのデータをもとに評価を行う．提案手法 1,2 と比較手法をそれぞれ比較した結果を**図 11**, **図 12** に示す．図 11, 図 12 は，横軸はエピソード数，縦軸は 1 エピソードごとの負の報酬 (waitng count, waiting time) を蓄積させたものである．いずれも比較手法よりも DQN による手法の方が，渋滞を解消できていると言える．

6. おわりに

本研究では，高い特徴抽出能力を持つ Deep Learning と，報酬に基づいた最適な行動を学習する強化学習を組み合わせた，Deep Q-Network による信号制御手法の提案とその

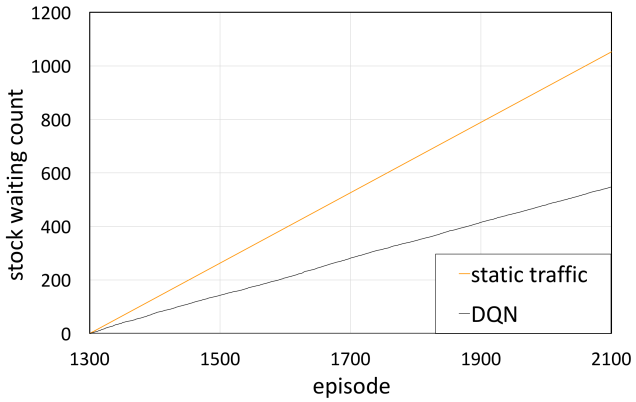


図 11 提案手法 1 と比較手法を比較

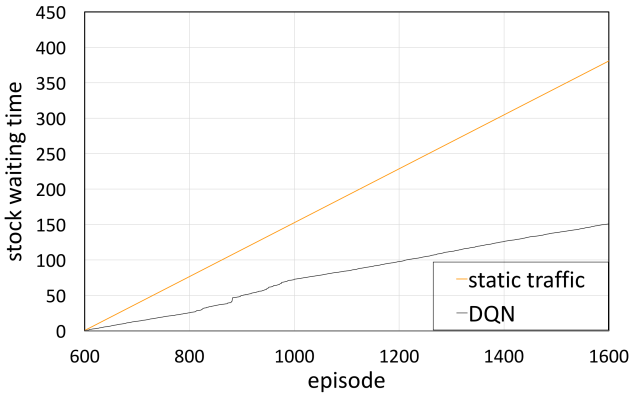


図 12 提案手法 2 と比較手法を比較

評価を行った．提案手法 1 では，報酬を待ち時間 (waiting time) とした手法を，提案手法 2 では，待ち台数 (waiting count) とした手法により実験を行った．その結果，いずれも 400 エピソード (80,000 step) 学習を繰り返した後，効率的に信号を操作することができていた．次に，提案手法 1,2 の 2 つの手法と静的な信号制御でそれぞれ比較し，実験を行った．提案手法 1,2 ともに，静的な信号制御に比べ，渋滞を発生させることなく制御できていた．このように，画像を入力するだけで，エージェント自身が信号制御に必要な特徴を抽出し，適切なパラメータ操作ができることが示された．また，静的な信号制御手法と比較し，実験を行った．その結果，従来手法よりも交通流の変化に対し適切に信号制御が行えることが確認された．

今後の課題として，本研究では，サイクル長において現実を反映させた設定になっていない．また，極小規模な交差点での学習にとどまっている．よって，現実的な設定での学習や，大規模道路ネットワークを対象とした実験を行う計画である．

参考文献

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al.: Human-level control through deep reinforcement learning, *Nature*, Vol. 518, No. 7540, pp. 529–533 (2015).

[2] 高橋光紀，篠田孝祐，諏訪博彦，　栗原聡：マルチエージェント信号制御メカニズムによるグリーンウェーブ制御法の提案と評価，人工知能学会全国大会論文集，Vol. 29, pp. 1–4 (2015).

[3] 西原稔貴，松元駿太，　上瀧剛，内村圭一，　杉谷浩，石垣信一：マルチエレメント GA による道路交通信号パラ

メータの最適化と実環境における検証 (ITS 画像処理, 映像メディア, 視覚及び一般)，電子情報通信学会技術研究報告. IE, 画像工学， Vol. 111, No. 442, pp. 263–268 (2012).
[4] Liu, W., Liu, J., Peng, J. and Zhu, Z.: Cooperative multi-agent traffic signal control system using fast gradient-descent function approximation for V2I networks, *Communications (ICC), 2014 IEEE International Conference on*, IEEE, pp. 2562–2567 (2014).
[5] van der Pol, E.: Deep reinforcement learning for coordination in traffic light control, PhD Thesis, Master' s Thesis. University of Amsterdam (2016).
[6] Sutton, R. S. and Barto, A. G.: *Reinforcement learning: An introduction*, Vol. 1, No. 1, MIT press Cambridge (1998).
[7] Kumaran, D. and Maguire, E. A.: Which computational mechanisms operate in the hippocampus during novelty detection?, *Hippocampus*, Vol. 17, No. 9, pp. 735–748 (2007).
[8] Riedmiller, M.: Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method, *European Conference on Machine Learning*, Springer, pp. 317–328 (2005).