修士論文

運動者に対するランニング経路推薦のための
方策勾配法に基づくランニング経路生成方法の研究

1610035　小倉裕平

主指導教員 Ho Tu Bao
審査委員主査 Ho Tu Bao
審査委員 橋本 敬
Huynh Nam Van
Dam Hieu Chi

北陸先端科学技術大学院大学

先端科学技術研究科 知識科学

平成 30 年月 2 月

目次

図目次

表目次

第 1 章　序論

1.1　研究背景

近年、健康志向の高まりから、習慣的にランニングなどの運動を行う人が増加している。スマートフォンやウェラブルデバイスの増加により、ランニングにおいて距離や時間だけでなく心拍数等の情報も記録することができるようになっている。習慣的にランニングを行う人の中には、スマートフォンやウェラブルデバイスで利用できるランナー専用のアプリを利用し日々のランニングの記録を行なっている人もいる。このようなランナー専用のアプリでは、ランニングの距離・時間・走行中の心拍数・気温・湿度等の情報の記録、走行した経路の表示、走行経路の提案等の機能が備わっている場合が多い。

日本では、旅ランというものが流行している。旅ランとは、旅行中に、旅行先の風情ある街並みや自然、観光スポット等をランニングしながら巡るというものである。旅ランを行うことで、観光を行いながら健康の増進を行うことができる。旅ランを行うにあたって、走る経路を設定せずにランニングするより、旅行先の風情ある街並みや自然、観光スポット等を走りながら楽しめる経路を設定してからランニングすることで、より旅ランを楽しむことができる。しかしながら、多くの人は旅行先についてはあまり知らないことが多く走りながら楽しめるランニング経路を設定するのは難しい。このため、旅ランを推奨している web サイトなどでは、オススメの旅ランコー スを提案している [1]。このように、旅ランそしてランニングを行う人の増加によりランニング経路の提案や推薦を行うことの需要が増加している。

ランニング経路に限らず経路の推薦を行う web サービスやアプリとして有名なものが Google Maps[2] である。しかしながら、Google Maps の経路推薦機能は、スタート地点とゴール地点の 2 点間の距離や時間のみを考慮した推薦となっておりランニングや旅ランのための経路推薦としては不十分である。また、ランナー専用アプリでは、ランニング経路の推薦の機能をもつものがあるが、それらはランナーが走りたい距離を入力することで、現在地から入力した距離を走ることができる経路を推薦する機能や、走りたい時間を入力することでその時間内で走りきれる経路を推薦する機能、と

いうものに留まっている。このようなランナー専用アプリで行われている時間や距離のみを考慮したランニング経路の推薦ではなく、ランナーの現在地や道の斜度、他のランナーがよく走行している経路など時間や距離以外の様々な情報を考慮したランニング経路の推薦が重要になっている。

1.2 関連研究と課題

ランナーに対するランニング経路の推薦に関する研究において、本研究に最も関連する 2 つの手法について述べる。1 つ目の手法は、初めに、ランニング経路の収集を行い、地形のタイプ、高度の変化等の情報を、収集したランニング経路に付加していきランニング経路のデータベースを作成する。ランナーが現在地から走行可能な経路をデータベースからフィルタリングし、フィルタリングして得られた経路を距離、時間、勾配等の情報を基にランク付けを行う。ランク付けされた経路をランナーが走った場合の心拍数をニューラルネットワークを用いて予測する。予測した心拍数とランナー が求めるランニングの負荷度を考慮しランニング経路の推薦を行う [3]。

2 つ目の手法は、初めに、ランニング経路の収集を行い、距離、標高差、水辺や公園等の周辺環境、陸上競技用トラックかどうか、オンロードかオフロードか等の経路の性質を考慮して、ランニング経路の分類を行う。分類された経路から、ランナーが過去に走ったことのある経路とランナーの現在地から走ることのできる経路を取得し、ランナーの現在地から走ることのできる経路の中で、ランナーが過去に走ったことのある経路と類似性が高い経路を推薦するという手法である [4]。

関連研究の手法では、動的にランナーにランニング経路を推薦する場合でデータベース上にランナーの現在地から走れる経路がない場合、新たに経路を取得し、取得した経路に対して情報を付加するという手順を踏まなければならず、ランニング経路の推薦を行うまでの時間が増大してしまう。また、ランニング経路を推薦するにあたって利用している情報は十分でないと考えられる。

1.3 研究の目的

本研究の目的は、距離や時間だけでなく、ランナーの過去のランニング時におけるランニング経路での速度の変化や進んだ方角等のより多くの情報を利用し、ランニング経路の推薦に関する効果的な手法を開発することである。関連研究の手法として挙げられているようなランニング経路を分類し最適な経路を推薦するという手法ではなく、強化学習の手法を用いてランニング経路を生成し推薦につなげるという手法を取る。

1.4　論文の構成

本論文では、第 2 章で提案手法の前提となる強化学習の手法について述べる。第 3 章では、強化学習の手法を用いてランニング経路を生成する手法について述べる。第 4 章では、提案手法の評価を行い、第 5 章では、本研究を総括し今後の研究の展開を述べる。

第 2 章　強化学習

本章では、強化学習の概要及び提案手法で用いる逆強化学習、方策勾配法の説明を行う。

2.1　強化学習の構成要素

強化学習 (Reinforcement learning) とは、強化学習エージェント (agent) という動作主体が、ある状態 (state) から方策 (policy) という強化学習エージェントの行動原理に従って行動し、環境 (environment) から報酬 (reward) と次の状態を受け取り、強化学習エージェントが、最終的に受け取ることのできる報酬の総和である収益 (return) を最大にするように学習を行う手法である。強化学習問題とは、置かれた環境のなかで、行動の選択を通して得られる報酬の総和を最大化する問題である [5]。方策について更に詳しく説明すると、方策は単純な関数であったり、ルックアップテーブルであったりするし、他の場合には探索などの計算をさらに行う [6]。

強化学習の構成要素は上記の、エージェント、状態、方策、環境、報酬以外にも、報酬関数、状態価値、行動価値、状態価値関数、行動価値関数が存在する。報酬関数は、エージェントがある状態である行動をとった際にどれだけ報酬を与えるかを定義する関数である。状態価値 (state value) は、エージェントがある状態から方策に従って行動を決定していったときに得られる収益の期待値である [5]。行動価値 (action value) は、エージェントがある状態である行動を取り報酬を受け取った後、方策に従って行動を決定していったときに得られる収益の期待値である。状態価値関数 (state value function) は、エージェントが現在いる状態の状態価値がどれくらい良いのかを評価する関数である。行動価値関数 (action value function) は、エージェントがある状態である行動を取った場合の行動価値がどれくらい良いのかを評価する関数である。状態価値関数の定義式を式 2.1 に、行動価値関数の定義を式 2.2 に示す。

$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right\} \tag{2.1}$$

$$Q^{\pi}(s,a) = E_{\pi}\{R_t|s_t = s, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s, a_t = a\right\} \quad (2.2)$$

s は状態、a は行動、t は単位時間、R_t は単位時間 t 以降に得られる割引された報酬の総和 (割引収益)、k は単位時間 t から進んだ時間を表す。状態価値関数と行動価値関数は、式 2.1、式 2.2 の形式だけでなく、パラメータベクトル θ でパラメータ化された関数の形で表すことができる。パラメータ化された状態価値関数と行動価値関数は、それぞれ、近似状態価値関数、近似行動価値関数という。近似状態価値関数、近似行動価値関数の一例を以下に示す。

$$V(s) = \theta_s^T \phi_s, \quad Q(s,a) = \theta_q^T \phi_{s,a} \quad (2.3)$$

θ_s と θ_q は、それぞれ、近似状態価値関数と近似行動価値関数のパラメータベクトルを表す。ϕ_s は状態 s での特徴ベクトル、$\phi_{s,a}$ は状態 s 行動 a での特徴ベクトルを表す。パラメータ化された関数で状態価値関数や行動価値関数を表す場合、パラメータベクトル θ は最急降下法などで更新する。図 2.1 に強化学習の構成要素を用いて強化学習の枠組みを示す。

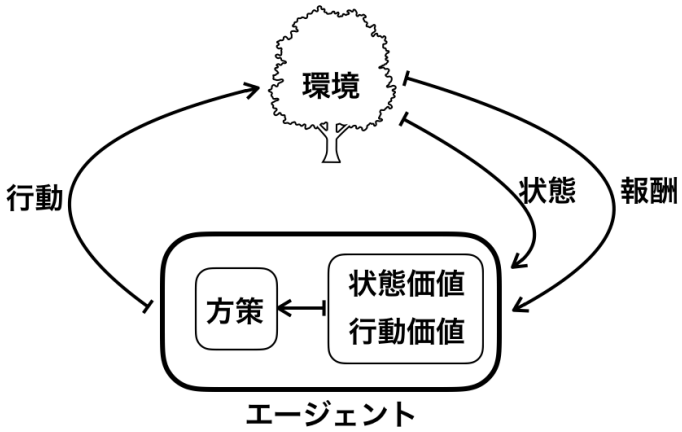

図 2.1　強化学習の枠組み

強化学習を利用する場合、基本的には、状態、行動、方策、報酬、価値関数、環境を設定する必要がある。

2.2　逆強化学習

強化学習において、適切な報酬関数を設計するのは非常に重要である。報酬関数の設計が間違っていれば、強化学習の枠組みに当てはめて学習を行っても、良い結果は得られない。しかし、適切な報酬関数を設計するのは、容易ではない。大規模で複雑な

問題に対して報酬関数を特定し、網羅的に調整することは非常に困難である [7]。このような報酬関数の設計が困難な場合に、逆強化学習 (Inverse reinforcement learning) という手法が有用である。逆強化学習は、学習したいタスクに関して模範となる行動を取ることができるエキスパートの行動から報酬関数を推定する手法である。以下に本研究で採用した Abbeel らによって提案された逆強化学習手法である projection 法 [8] について述べる。

projection 法では、ある方策 π に従った時の割引累積特徴量を特徴期待値 (feature expectation) と定義する。以下に定義式を記述する。

$$\mu(\pi) = E\left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi\right] \in \mathcal{R}^k \tag{2.4}$$

γ は割引率、$\phi(s_t)$ は単位時間 t の状態 s における特徴量、k は特徴数である。Algorithm1 に projection 法のアルゴリズを示す。特に、エキスパートの特徴期待値を以下の式で表す。

$$\hat{\mu}_E = \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{\infty} \gamma^t \phi(s_t^{(i)}) \tag{2.5}$$

m は、エキスパートのデータセット数である。

> projection 法を簡潔に説明すると、方策関数の特徴期待値から計算される正射影の値とエキスパートのデータから計算されるエキスパートの特徴期待値を用いて報酬関数のパラメータを更新する手法である。

2.3 方策勾配法

方策勾配法について述べる前に、方策勾配法を用いない強化学習アルゴリズムについて簡潔に述べる。方策勾配法を用いない強化学習アルゴリズムとして、代表的なものに Watkins により提案された Q-learning[9] や Rummery らにより提案された Sarsa[10] などが挙げられる。これらのアルゴリズムは、方策として、行動価値に基づいて行動を選択する方策を用いる。エージェントが行動価値を学習していくことで、より良い方策を見つけ出すことができる。

方策勾配法は、Q-learning や Sarsa などとは異なり、パラメータ化された関数として方策を表現し、方策のパラメータを行動価値や状態価値を用いて学習することで、より良い方策を見つけ出すことができる。一般には、連続の状態、行動空間を取り扱いたい場合は、方策勾配に基づく強化学習アルゴリズムを選択する利点が大きいと言える [5]。

方策勾配法のアルゴリズムの一種である Actor-Critic 法を Algorithm 2 に示す。

Algorithm 1 projection 法

1: 初期方策としてランダムに方策 $\pi^{(0)}$ を設定
2: 初期方策 $\pi^{(0)}$ における特徴期待値 $\mu^{(0)} = \mu(\pi^{(0)})$ を計算
3: $i = 1$ とする
4: **if** $i = 1$ **then**
$w^1 = \mu_E - \mu^0$
$\overline{\mu}^{(0)} = \mu^{(0)}$
5: **else**
以下の式で $\overline{\mu}^{(i-2)}$ と $\overline{\mu}^{(i-1)}$ を通る直線への μ_E の正射影 (orthogonal projection) を計算
$\overline{\mu}^{(i-1)} = \overline{\mu}^{(i-2)} + \frac{(\mu^{(i-1)} - \overline{\mu}^{(i-2)})^T (\mu_E - \overline{\mu}^{(i-2)})}{(\mu^{(i-1)} - \overline{\mu}^{(i-2)})^T (\mu^{(i-1)} - \overline{\mu}^{(i-2)})} (\mu^{(i-1)} - \overline{\mu}^{(i-2)})$
$w^i = \mu_E - \overline{\mu}^{(i-1)}$
$t^i = \|\mu_E - \overline{\mu}^{(i-1)}\|_2$
6: **end if**
7: **if** $t^i \leq \epsilon$ **then** 処理を終了
8: **end if**
9: 各特徴期待値に対する重み w^i と報酬関数 $R = (w^{(i)})^T \phi$ を用いて強化学習アルゴリズムで方策 π^i を計算
10: $\mu^{(i)} = \mu(\pi^{(i)})$
11: $i = i + 1$ とし、step4 に戻る

Actor-Critic 法は、actor(行動器) と critic(評価器) という 2 つの学習器を用いる。actor は方策に従って行動を決定し、critic は価値関数 (状態価値関数または行動価値関数) を用いて actor が決定した行動を評価する。評価は TD 誤差 (Temporal Difference error) として出力され、TD 誤差を用いて方策のパラメータを更新する。状態価値関数の TD 誤差は、Algorithm 2 の δ として定義される。Algorithm 2 の δ で状態価値関数を用いている部分を行動価値関数に変えると、行動価値関数の TD 誤差となる。図 2.2 に、Actor-Critic 法の概要を示す。

Algorithm 2 Actor-Critic 法

方策を、パラメータ θ を用いて $\pi(a|s,\theta)$ で表す

状態価値関数を、パラメータ w を用いて $v(s,w)$ で表す

$\alpha > 0, \beta > 0$ は学習率を、γ は割引率を表す

状態 S を初期化する (エージェントを初期状態に位置させる)

$I = 1$

while 状態 $\neq$ 終端状態 (エージェントが行動を終える状態) **do**

    方策に従って行動 a を行い、環境から次の状態 s' と報酬 r を観測する

    $\delta = r + \gamma v(s', w) - v(s, w)$

    $\theta = \theta + \alpha I \delta \nabla_\theta \log \pi(s, a, \theta)$

    $w = w + \beta \delta \nabla_w v(s, w)$

    $I = \gamma I$

    $s = s'$

end while

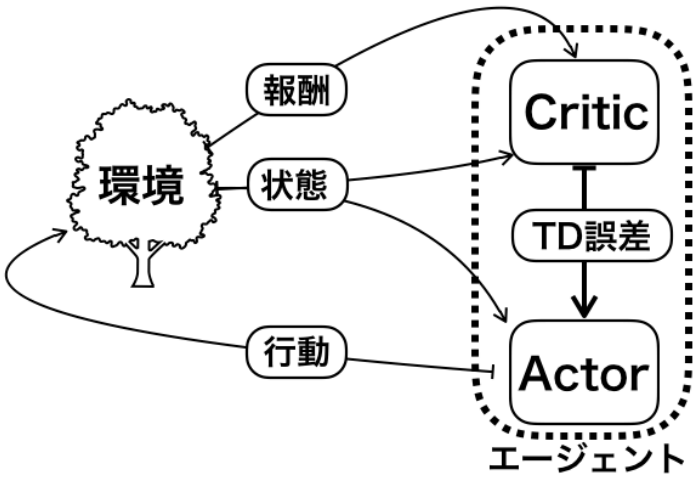

図 2.2　Actor-Critic 法の概要

第 3 章　提案手法

3.1　提案手法の概要

本章では、ランニング経路推薦のためのランニング経路生成手法の提案を行う。図 3.3 に提案手法の概要を示す。

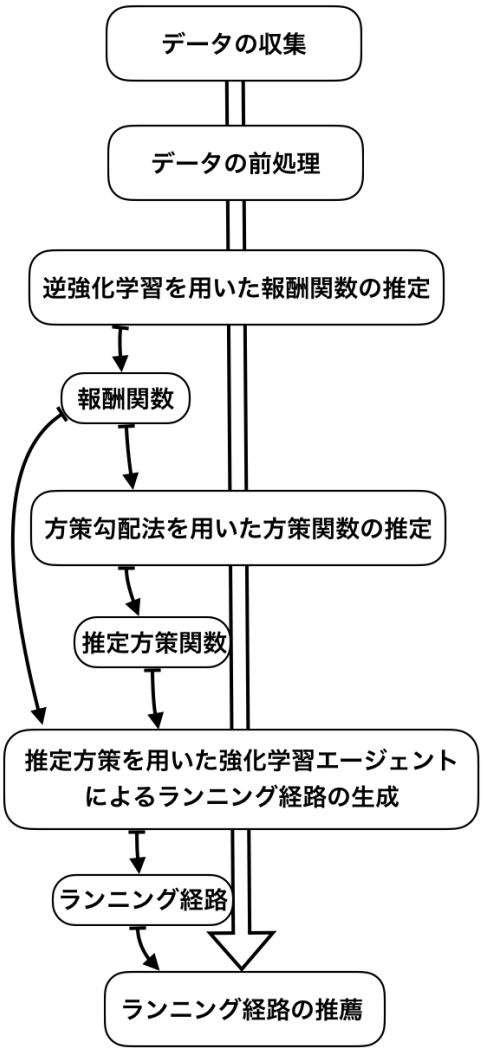

図 3.3　提案手法の概要

3.2　データの収集

本研究で用いるデータは、フィットネスアプリである endomondo[11] の公開データから収集した。endomondo は、ランニングやウォーキングなどのフィットネスの記録・管理を行うアプリである。収集したデータは、1 レコードで 1 トレーニング (ランニングやウォーキングなど) の結果を表し、1 レコードには様々なカラムが存在する。また、1 レコードには、アプリ利用者のトレーニング開始から終了までの軌跡として、数秒間隔で取得された複数の緯度経度情報が含まれている。これらのデータを逆強化学習の際に利用するエキスパートのデータとする。本研究で使用するカラムを表 3.1 に示す。表 3.1 のカラムは、逆強化学習の際のエキスパートの特徴期待値の計算とエキスパートの行動と現在の方策で出力した行動との差を計算する際に利用する。

表 3.1　データカラム

カラム名	説明
latitude	緯度
longitude	経度
distance	距離
ascent	上昇
descent	下降
calories	消費カロリー
heart_rate_avg	平均心拍数
heart_rate_max	最大心拍数
humidity	湿度
speed_avg	平均速度
speed_max	最大速度
temperature	気温
wind_speed	風速
wind_direction	風向

データの公開を行なっている 1,609 人の endomondo アプリ利用者から、859,752 件のレコードを収集した。

3.3　データの前処理

収集したデータは、欠損値が多く存在し、ランニングの結果ではなくサイクリングの結果なども存在した。表 3.1 のカラムの内、緯度経度情報に欠損値が存在するレコードは除外し、それ以外のカラムに欠損値が存在した場合は、それらのカラムの平均値で補完した。また、時速 30km を越える速度での移動が存在するレコードについても除外した。

3.4　ランニング経路の生成方法

2.1 節で述べたように、強化学習では構成要素として、状態、行動、方策関数、環境、報酬関数、価値関数 (本研究では状態価値関数、行動価値関数の両方) を定義する必要がある。本研究では、状態は緯度経度または表 3.1 に示した情報で表される。行動は速度 (単位は km/h) と方角とし、エージェントは 15 秒間隔で行動を行う。方策関数は、次節以降で詳しく述べるが、確率密度関数を用いた方策関数と深層ニューラルネットワークを用いた方策関数を用意した。方策関数は状態を受け取り、行動として速度と方角を出力する。環境は、エージェントの行動として方策関数の出力値を受け取り、エージェントの 1 単位時間前の状態と受け取った行動 (速度と方角) を基にエージェントの次の状態を返す。また、環境は報酬関数を基に報酬を返す。報酬関数は、逆強化学習の手法により求める。価値関数は式 2.3 を用いる。上記の定義を用いた本研究での強化学習の枠組みを図 3.4 に示す。

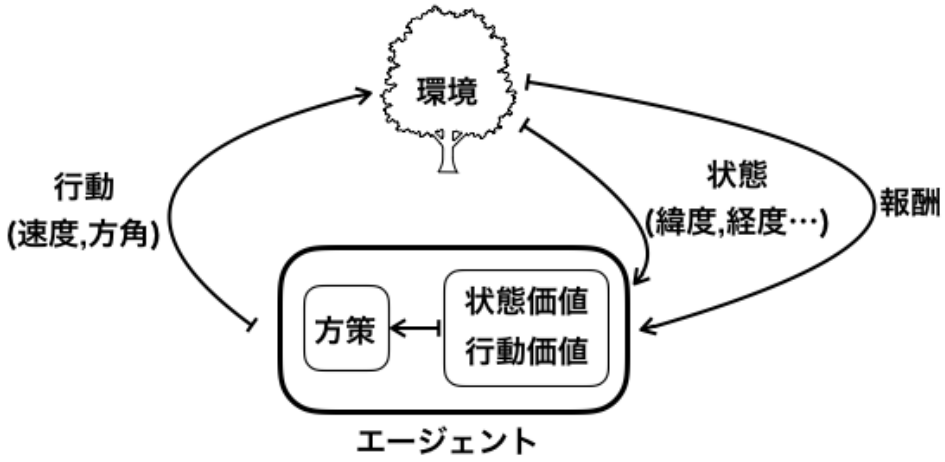

図 3.4　本研究での強化学習の枠組み

上記の強化学習の定義を用いて、強化学習の手法によりランニング経路の生成を

行う方法について述べる。エージェントは、現在の状態から行動を行い次の状態に遷移する。エージェントの行動は速度と方角で、行動は 15 秒間隔で行われるので、15 秒 × 速度 でエージェントが進んだ距離を求めることができる。エージェントの行動の結果として緯度経度が得られるので、初期状態から終端状態までの緯度経度を辿ることでランニング経路が完成する。図 3.5 に状態が緯度経度のみで表される場合のランニング経路の生成の方法を示す。また、図 3.5 にランニング経路の生成のイメージ図を示す。

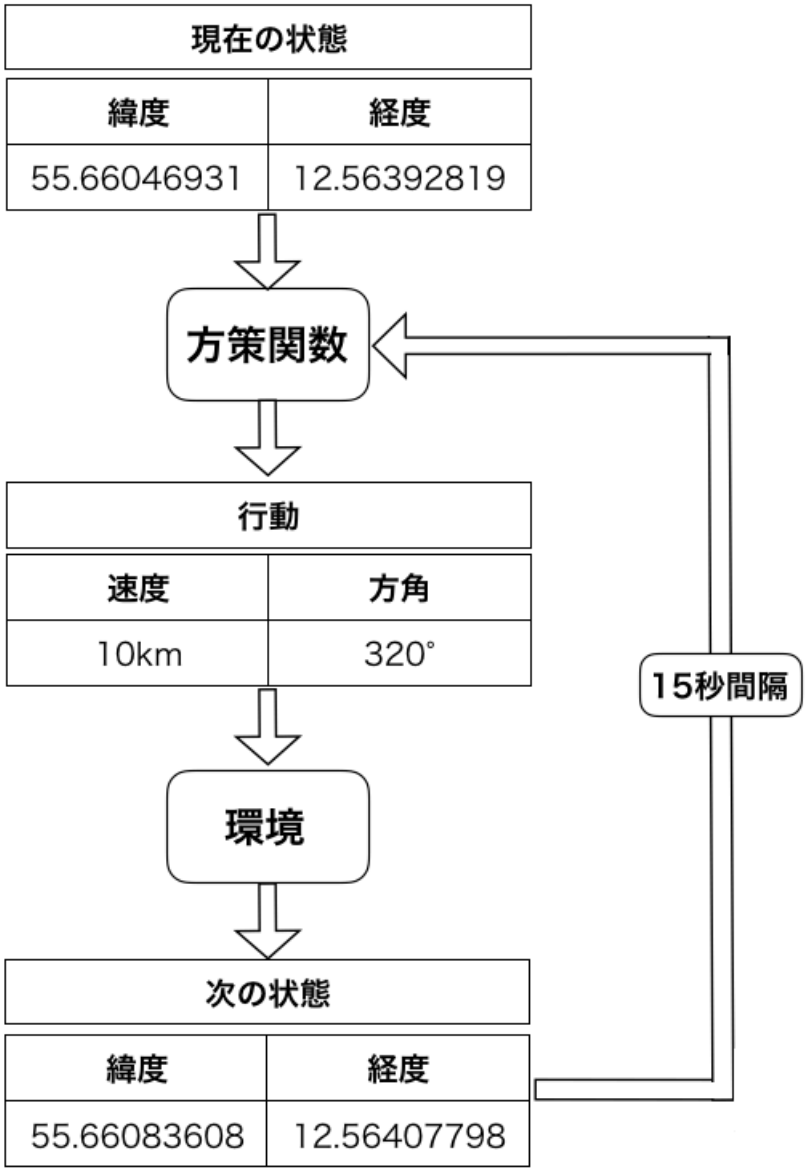

図 3.5　強化学習の手法を用いたランニング経路の生成方法

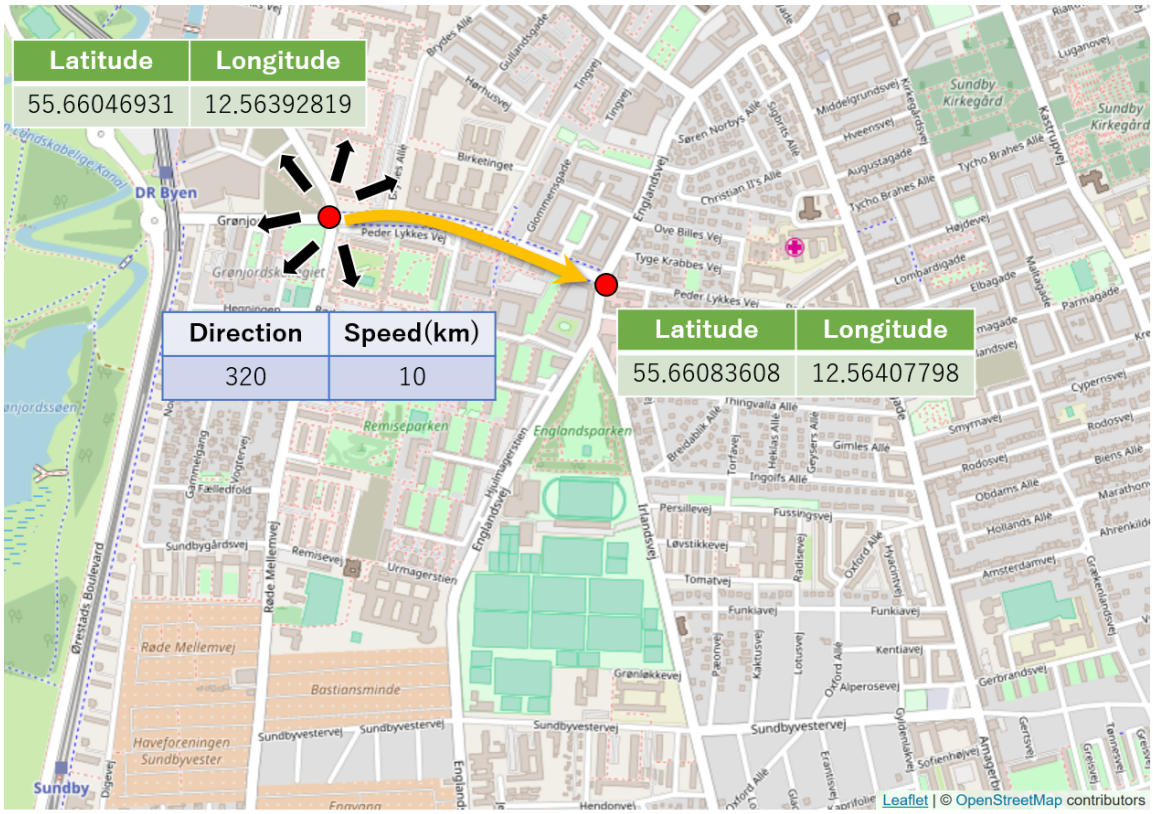

図 3.6　強化学習の手法を用いたランニング経路の生成方法のイメージ図

3.5　確率密度関数を用いた方策関数

本研究では、強化学習アルゴリズムとして Actor-Critic 法を用いる。2.3 節で述べたように Actor-Critic 法では、方策関数はパラメータ θ を用いて表される。Actor-Critic 法で連続値を扱う際に方策関数として広く利用されている確率密度関数を本研究でも利用する。確率密度関数として正規分布を用いた方策関数を 2 種類用意した。1 つは特徴数が緯度経度の 2 つのもの、もう 1 つは表 3.1 の特徴を利用するものである。2 つの方策関数は利用する特徴数は異なるが計算方法は同じである。以下に定義式を示す。

$$
\begin{aligned}
\mu &= a' \\
\sigma &= \theta^T \phi \\
\pi(a|s,a',\theta) &= \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{(a-\mu)^2}{2\sigma^2}\right]
\end{aligned}
\tag{3.6}
$$

a は方策関数が出力する値、s は状態、a' は 1 単位時間前の行動、θ は方策関数のパラメータ、ϕ は特徴を表す。

3.6　深層ニューラルネットワークを用いた方策関数

確率密度関数を用いた方策関数以外に深層ニューラルネットワークを用いた方策関数を用意した。深層ニューラルネットワーク (Deep Neural Network) とは、パーセプトロンを多層化したものである [12]。深層ニューラルネットワークなどのディープラーニングの手法を用いることで線形関数では達成することができない高い表現力を達成することができる。本研究でも深層ニューラルネットワークを用いることで、高い表現力を持った方策関数の実現を目指した。本研究で用いる深層ニューラルネットワークの隠れ層のユニット数や利用する活性化関数等のハイパーパラメータについては、4.1 節で詳しく述べる。深層ニューラルネットワークを用いた方策関数も確率密度関数を用いた方策関数と同様に 2 種類用意した。1 つは特徴数が緯度経度の 2 つのもの、もう 1 つは表 3.1 の特徴を利用するものである、また深層ニューラルネットワークの入力にはそれぞれの特徴とエージェントの 1 単位時間前の行動を利用する。2 つの方策関数は利用する特徴数は異なるが計算方法は同じである。以下に定義式を示す。

$$\pi(a|s, a', \theta) = DNN(\phi, a') + \frac{sum(\theta)}{\text{特徴数}} \tag{3.7}$$

$DNN(\phi, a')$ は入力に特徴 ϕ とエージェントの 1 単位前の行動 a' を取り、エージェントの次の行動を出力する深層ニューラルネットワークを表す。また、a は方策関数が出力する値、s は状態、a' は 1 単位時間前の行動、θ は方策関数のパラメータを表す。$sum(\theta)$ は θ の合計を返す関数である。

3.7　逆強化学習を用いた報酬関数の推定

ランニング経路の生成という課題に対して適切な報酬関数を設計するのは容易ではない。なぜなら生成した経路の良さが定義出来ない場合が多く存在するからである。エキスパートが走った経路に似た経路には高い報酬を与える、というように設計することが考えられるが、この場合、エキスパートのデータが得られていない地域に関しては報酬を与えることができなくなる。本研究では、地域を限定してデータを収集していないので上記の方法を取ることはできない。故に報酬関数の設計が困難な場合に利用される逆強化学習の手法を用いて報酬関数を設計した。図 3.7 に本研究における逆強化学習を用いた報酬関数の推定の方法を示す。

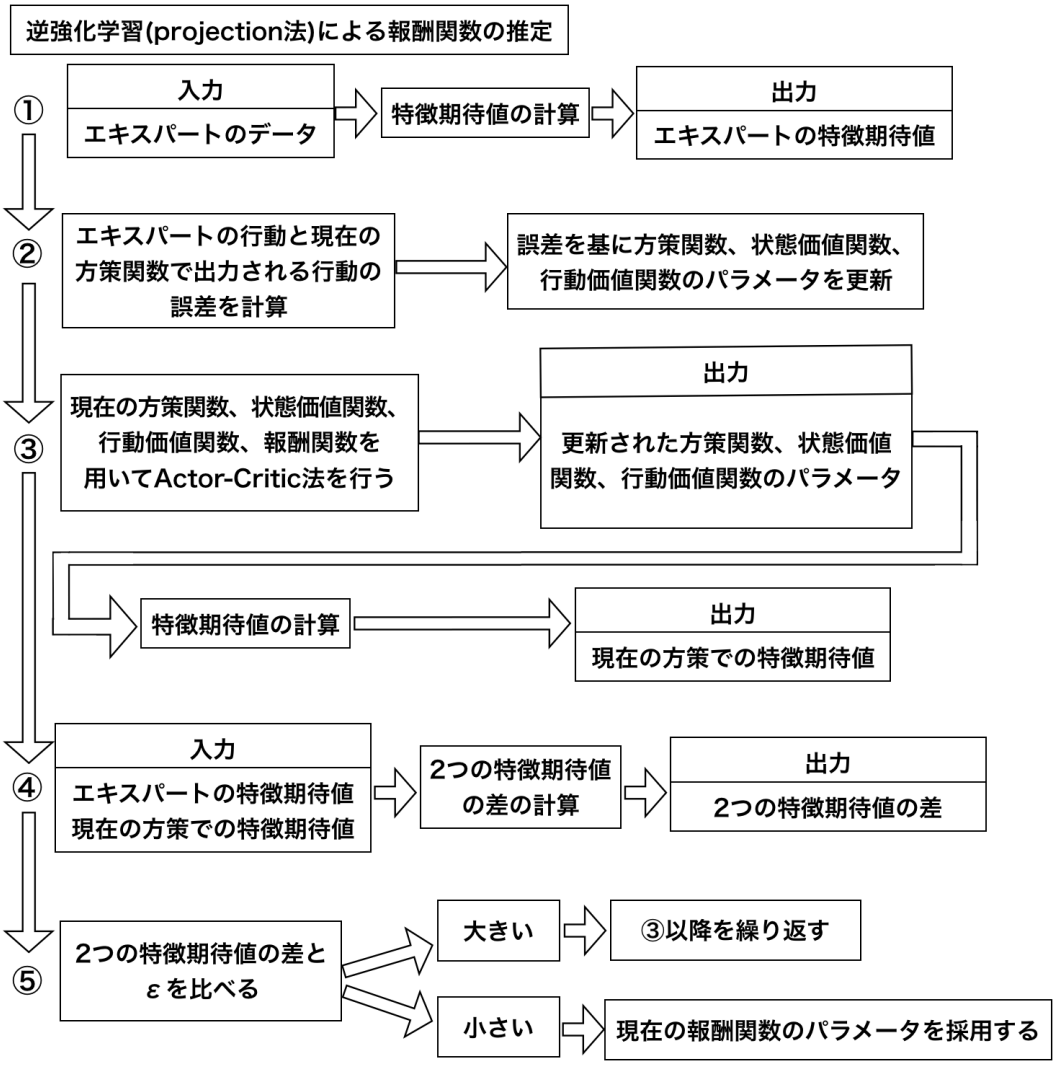

図 3.7　逆強化学習を用いた報酬関数の推定方法

図 3.7 にあるように、逆強化学習を用いた報酬関数の推定方法のステップ 1 として、収集したデータをエキスパートのデータとし、その特徴期待値を計算する。ステップ 2 として、エキスパートのデータのすべてのランニングのレコードを取り出し、そのレコードの初期状態 (ランニング開始) から終端状態 (ランニング終了) までのそれぞれの状態に関して、その状態の際に取った行動と現在の方策関数で取る行動の誤差を TD 誤差として計算し、その TD 誤差を基に方策関数、状態価値関数、行動価値関数のパラメータを更新する。ステップ 3 として、現在の方策関数、状態価値関数、行動価値関数、報酬関数を用いて Actor-Critic 法を行い、方策関数、状態価値関数、行動価値関数のパラメータを更新する。更新された方策関数を基に現在の方策関数での特徴期待値を計算する。ステップ 4 として、ステップ 1 で計算したエキスパートの特徴期待値と、ステップ 3 で計算した現在の方策関数での特徴期待値の差を計算する。ス

テップ 5 として、ステップ 4 で計算した 2 つの特徴期待値の差が十分に小さい値である ϵ より大きければ、ステップ 3 以降を繰り返す、ϵ より小さければ、その時点での報酬関数のパラメータを採用する。

3.8 方策勾配法を用いた方策関数の推定

提案手法の 4 番目のステップである方策勾配法を用いた方策関数の推定は、2.3 節の Algorithm 2 を用いて方策を推定する。方策推定の際に生成されるランニング経路は、3.4 節で述べたランニング経路生成の方法を用いて生成されるが、この方法だと、エージェントが道のない場所に進んでしまうことがある。例えば海の上を通る経路を生成してしまったりなどが挙げられる。この問題を防ぐにはエージェントが現在いる位置から進むことのできる道路情報を毎行動ごとに取得する必要があるが、これは非常に困難である。本研究では、代替策として Google Maps Roads API[13] の Snap to roads 機能を用いることにした。Google Maps Roads API は、Google Map の様々な機能を利用できる API で、Snap to roads 機能というのは、最大 100 個の GPS 座標の集合を受け取り、その GPS 座標の集合に最も合致する道路の GPS 座標の集合を返す。具体的な代替策として、エージェントが生成した経路と、エージェントが生成した経路を Snap to roads 機能を用いて修正した経路とを、軌跡の類似性を測る指標である DTW を用いて評価する、DTW については次章で詳しく述べる、評価値である DTW 値を TD 誤差とし、その TD 誤差を基に行動価値関数のパラメータを更新する。方策関数の推定の際に、エージェントができるだけ道に沿って進むように、上記の代替策を組み込んだ。

3.9 推定方策を用いた強化学習エージェントによるランニング経路の生成及びランニング経路の推薦

提案手法の 3 番目のステップで求めた報酬関数、提案手法の 4 番目のステップで求めた方策関数を用いて、3.4 節で述べたランニング経路生成の方法の通りにランニング経路の生成を行う。ランニング経路の推薦に関しては、複数のランニング経路を生成し、生成した経路をランナーに提示するという方法が考えられる。

第 4 章　評価実験

前章では、強化学習の手法を用いてランニング経路の生成する方法を述べた。本章では、生成したランニング経路の評価を行う。

4.1　深層ニューラルネットワークの構成とハイパーパラメータの探索

3.6 節で述べたように本研究では深層ニューラルネットワークを用いた方策関数を利用している。深層ニューラルネットワークでは、ハイパーパラメータという隠れ層のユニット数、利用する活性化関数や最適化アルゴリズム等を人間の手で設定しなければならない。ハイパーパラメータの多種多様な組み合わせから最適な組み合わせを選択するのは困難であるが、精度の高いハイパーパラメータの組み合わせを選択する方法として Grid search と Randomized search[14] という方法がある。本研究では、Randomized search を利用してハイパーパラメータの選択を行なった。Randomized search は、まず、ハイパーパラメータの組み合わせを探索する回数の設定と候補となるハイパーパラメータの用意が必要となる。その後、事前に用意したハイパーパラメータの中からランダムに組み合わせを選び精度を求め、探索したハイパーパラメータの組み合わせの中で最も高い精度を出したハイパーパラメータの組み合わせを選択する、という手法である。3.6 節で述べたように、本研究において深層ニューラルネットワークを用いた方策関数は 2 種類ある。1 つは特徴数が 2 つのものである。この方策関数で用いられる、特徴を 2 つ利用する深層ニューラルネットワークの構成を図 4.8 に示す。入力層は、利用する 2 つの特徴とエージェントの 1 単位時間前の行動 (速度、方角) の 4 である。深層ニューラルネットワークを用いた方策関数の 2 つ目は、特徴数が 14 のものである。この方策関数で用いられる、14 の特徴を利用する深層ニューラルネットワークの構成を図 4.9 に示す。入力層は、利用する 14 の特徴とエージェントの 1 単位時間前の行動 (速度、方角) の 16 である。2 種類の深層ニューラルネットワークの出力層はエージェントの行動である速度と方角の 2 値を出力する。それぞれの深層ニューラルネットワーク対して、候補となるハイパーパラメータを事前に用意

した。事前に用意したハイパーパラメータをそれぞれ表 4.2、表 4.3 に示す。図 4.8 の隠れ層 3-1、3-2、図 4.9 の隠れ層の 4-1、4-2 のユニット数は 1 である。

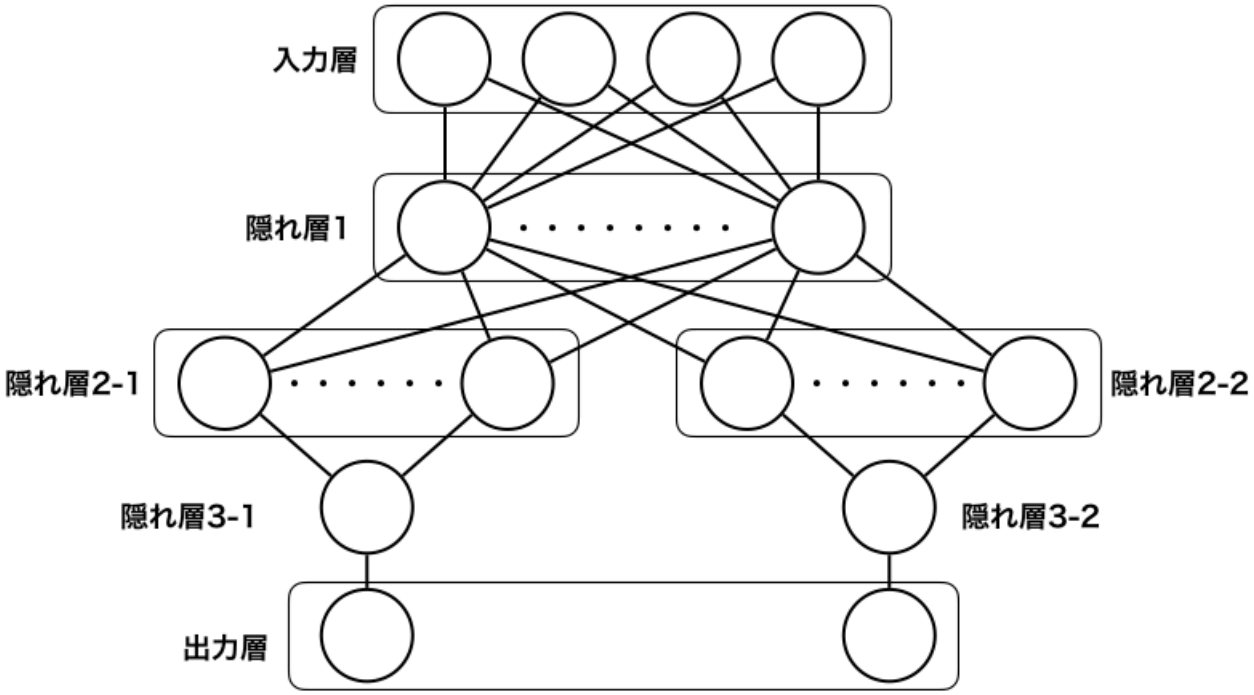

図 4.8　特徴数 2 の深層ニューラルネットワーク

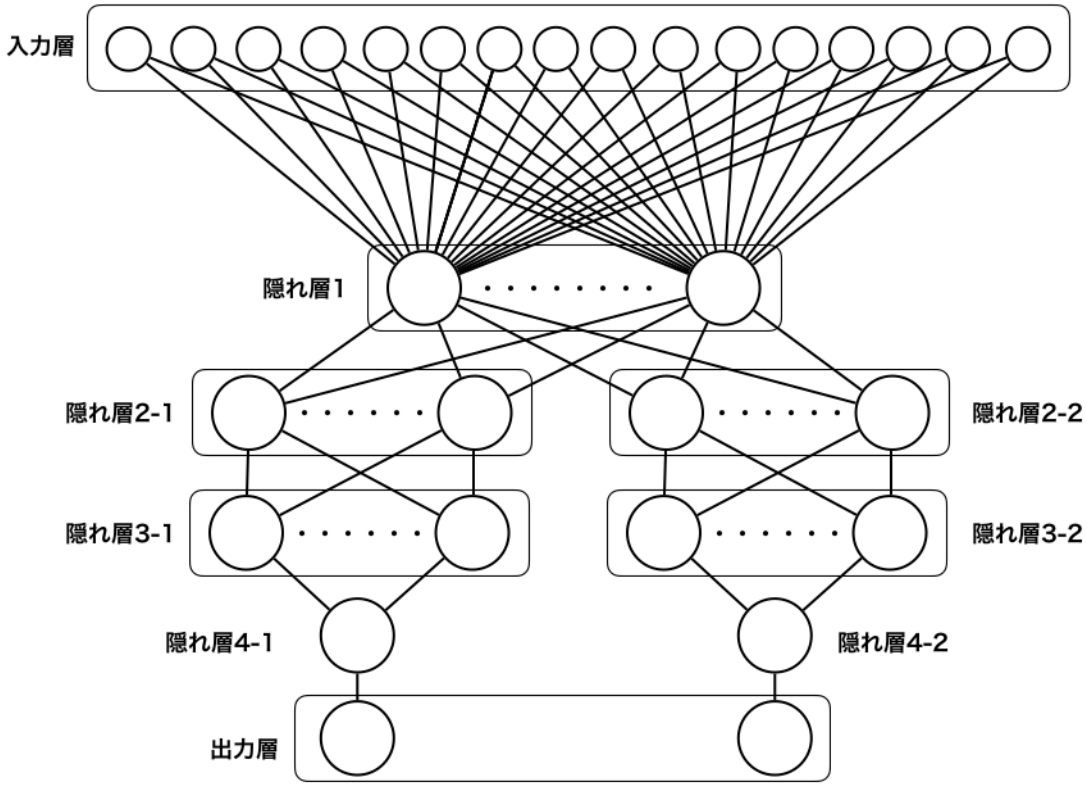

図 4.9　特徴数 14 の深層ニューラルネットワーク

表 4.2　特徴数 2 の深層ニューラルネットワークの候補となるハイパーパラメータ

パラメータ	候補
隠れ層 1 のユニット数	100, 1000
隠れ層 2-1 のユニット数	100, 1000
隠れ層 2-2 のユニット数	100, 1000
隠れ層 2-1 で使用する活性化関数	relu, sigmoid, softplus
隠れ層 2-2 で使用する活性化関数	relu, sigmoid, softplus
最適化アルゴリズム	Adam, Rmsprop, Adamax

本研究で行う Randomized search は、scikit-learn[15] の Randomized search を行える関数である RandomizedSearchCV 関数を利用した。事前に用意したハイパーパラメータの中から精度の高いの組み合わせを選択するために、ハイパーパラメータの組み合わせの探索回数を 200 回に設定し、RandomizedSearchCV 関数を利用し Randomized search を行なった。Randomized search で利用したデータは、次節の強化学習エージェントの学習で利用していないデータを用いた。データ数は、200 人の endomondo アプリ利用者から取得した 1,336 件である。Randomized search

表 4.3　特徴数 14 の深層ニューラルネットワークの候補となるハイパーパラメータ

パラメータ	候補
隠れ層 1 のユニット数	100, 500
隠れ層 2-1 のユニット数	100, 500
隠れ層 2-2 のユニット数	100, 500
隠れ層 3-1 のユニット数	100, 500
隠れ層 3-2 のユニット数	100, 500
隠れ層 2-1 で使用する活性化関数	relu, sigmoid, softplus
隠れ層 2-2 で使用する活性化関数	relu, sigmoid, softplus
隠れ層 3-1 で使用する活性化関数	relu, sigmoid, softplus
隠れ層 3-2 で使用する活性化関数	relu, sigmoid, softplus
最適化アルゴリズム	Adam, Rmsprop, Adamax

の結果の精度の高い組み合わせの上位 5 つをそれぞれ表 4.4、表 4.5 に示す。また、Randomized search のそれぞれの処理時間を表 4.6 に示す。 本研究では、表 4.4、表

表 4.4　特徴数 2 の深層ニューラルネットワークの Randomized search の結果

	精度	hp1	hp2	hp3	hp4	hp5	hp6
1	-0.67681	100	1000	1000	relu	relu	Adamax
2	-0.67731	1000	100	1000	relu	relu	Adamax
3	-0.67731	100	1000	1000	softplus	relu	Adamax
4	-0.67739	1000	100	100	relu	relu	Adam
5	-0.67751	100	100	100	relu	softplus	Adamax

※ hp1:隠れ層 1 のユニット数
hp2:隠れ層 2-1 のユニット数
hp3:隠れ層 2-2 のユニット数
hp4:隠れ層 2-1 で使用する活性化関数
hp5:隠れ層 2-2 で使用する活性化関数
hp6:最適化アルゴリズム

4.5 で最も精度の高いハイパーパラメータの組み合わせである 1 行目の結果をそれぞれ特徴数 2 の深層ニューラルネットワーク、特徴数 14 の深層ニューラルネットワークのハイパーパラメータとして採用した。

表 4.5　特徴数 14 の深層ニューラルネットワークの Randomized search の結果

	精度	hp1	hp2	hp3	hp4	hp5	hp6	hp7	hp8	hp9	hp10
1	-0.68253	100	100	100	500	500	softplus	softplus	relu	relu	Adam
2	-0.68389	100	500	500	100	500	softplus	softplus	softplus	relu	Adamax
3	-0.68492	500	500	500	500	100	relu	sigmoid	relu	softplus	Adam
4	-0.68681	100	100	100	100	100	softplus	sigmoid	softplus	sigmoid	Adam
5	-0.68713	500	100	500	100	100	relu	softplus	relu	relu	Adamax

※ hp1:隠れ層 1 のユニット数
hp2:隠れ層 2-1 のユニット数
hp3:隠れ層 2-2 のユニット数
hp4:隠れ層 3-1 のユニット数
hp5:隠れ層 3-2 のユニット数
hp6:隠れ層 2-1 で使用する活性化関数
hp7:隠れ層 2-2 で使用する活性化関数
hp8:隠れ層 3-1 で使用する活性化関数
hp9:隠れ層 3-2 で使用する活性化関数
hp10:最適化アルゴリズム

表 4.6　Randomized search の処理時間

I	II
66184.40990(秒)	77999.31557(秒)

I 特徴数 2 の深層ニューラルネットワークでの Randomized search の処理時間
II 特徴数 14 の深層ニューラルネットワークでの Randomized search の処理時間

4.2　強化学習エージェントの学習

本節では、前章で述べた提案手法を用いて強化学習エージェントの学習を行った結果について述べる。3.2 節で述べた収集したデータを用いて前処理を行い、学習に利用するデータを endomondo アプリ利用者の 800 人のから、40,926 件のレコードに絞り込んだ。このデータを使用し、提案手法の 3 番目のステップである逆強化学習を用いた報酬関数の推定及び 4 番目のステップである方策勾配法を用いた方策関数の推定を行った。確率密度関数を用いた特徴数 2 の方策関数、確率密度関数を用いた特徴数 14 の方策関数、深層ニューラルネットワークを用いた特徴数 2 の方策関数、深層ニューラルネットワークを用いた特徴数 14 の方策関数のそれぞれの方策関数を用いて学習

した後の報酬関数、方策関数、状態価値関数、行動価値関数のパラメータをそれぞれ表 4.7、表 4.8、表 4.9、表 4.10 に示す。また、それぞれの方策関数での学習にかかった時間を表 4.11 に示す。

表 4.7　確率密度関数を用いた特徴数 2 の方策関数を使用し学習した後のパラメータ

	報酬関数	状態価値関数	行動価値関数	方策関数
パラメータ 1	-5.67389062e-06	1.11113042	1.12218984	2.68897416
パラメータ 2	7.17252357e-06	0.25160957	-0.17890386	0.25160957

表 4.8　確率密度関数を用いた特徴数 14 の方策関数を使用し学習した後のパラメータ

	報酬関数	状態価値関数	行動価値関数	方策関数
パラメータ 1	-2.97982123	2.8868106	2.91554897	2.8868106
パラメータ 2	-0.96339672	2.8868106	2.91554897	2.8868106
パラメータ 3	-1.12072819	2.8868106	2.91554897	2.8868106
パラメータ 4	-1.13462085	2.8868106	2.91554897	2.8868106
パラメータ 5	-1.13691117	2.8868106	2.91554897	2.8868106
パラメータ 6	-0.99563854	2.8868106	2.91554897	2.8868106
パラメータ 7	2.84503579	2.8868106	2.91554897	2.8868106
パラメータ 8	3.17804529	2.8868106	2.91554897	2.8868106
パラメータ 9	0.0	2.8868106	2.91554897	2.8868106
パラメータ 10	-1.17882287	2.8868106	2.91554897	2.8868106
パラメータ 11	-1.1834131	2.8868106	2.91554897	2.8868106
パラメータ 12	0.0	2.8868106	2.91554897	2.8868106
パラメータ 13	0.0	2.8868106	2.91554897	2.8868106
パラメータ 14	0.0	2.8868106	2.91554897	2.8868106

表 4.9　深層ニューラルネットワークを用いた特徴数 2 の方策関数を使用し学習した後のパラメータ

	報酬関数	状態価値関数	行動価値関数	方策関数
パラメータ 1	-0.48182735	0.73613521	7.43401576e-01	2.11855356
パラメータ 2	-0.28865424	-0.15484293	2.87667598e-05	-0.15484293

表 4.10　深層ニューラルネットワークを用いた特徴数 14 の方策関数を使用し学習した後のパラメータ

	報酬関数	状態価値関数	行動価値関数	方策関数
パラメータ 1	-1.7835499	8.56688449	8.65248169	8.56688449
パラメータ 2	2.57368134	8.56688449	8.65248169	8.56688449
パラメータ 3	-0.81305351	8.56688449	8.65248169	8.56688449
パラメータ 4	-0.80111073	8.56688449	8.65248169	8.56688449
パラメータ 5	-0.69895097	8.56688449	8.65248169	8.56688449
パラメータ 6	-0.68017512	8.56688449	8.65248169	8.56688449
パラメータ 7	2.89367574	8.56688449	8.65248169	8.56688449
パラメータ 8	3.90598376	8.56688449	8.65248169	8.56688449
パラメータ 9	0.0	8.56688449	8.65248169	8.56688449
パラメータ 10	–1.04752466	8.56688449	8.65248169	8.56688449
パラメータ 11	-0.67530669	8.56688449	8.65248169	8.56688449
パラメータ 12	0.0	8.56688449	8.65248169	8.56688449
パラメータ 13	0.0	8.56688449	8.65248169	8.56688449
パラメータ 14	0.0	8.56688449	8.65248169	8.56688449

表 4.11　各方策関数での処理時間

I	II	III	IV
10821.21613(秒)	12769.56206(秒)	13690.39979(秒)	18581.95481(秒)

I 確率密度関数を用いた特徴数 2 の方策関数
II 確率密度関数を用いた特徴数 14 の方策関数
III 深層ニューラルネットワークを用いた特徴数 2 の方策関数
IV 深層ニューラルネットワークを用いた特徴数 14 の方策関数

4.3　評価方法

本節では、強化学習の手法を用いて生成したランニング経路の評価方法について述べる。学習に使用していないデータから任意のレコードを取り出し、それをエキスパートのランニング経路とし、エキスパートのランニング経路のスタート地点をエージェントの初期状態としてランニング経路の生成を行なった際に、エキスパートのランニング経路とエージェントが出力したランニング経路の類似度を測ることで生成

したランニング経路を評価した。2 つの軌跡間の類似度を計算するための手段はいくつか存在し [17]、それら手段は Trajectory Similarity Measures と呼ばれている。エキスパートのランニング経路とエージェントが出力したランニング経路の類似度を測る手段として Trajectory Similarity Measures を用いた。本研究では、Trajectory Similarity Measures のうち DTW(Dynamic Time Warping)[18]、Fréchet 距離 [19]、Edit 距離 [20] の 3 つの指標で類似度を評価した。DTW、Fréchet 距離、Edit 距離の値は、低いほど類似度が高いことを示す。

類似度で評価を行う際に基準値が必要になるので、基準として標準正規分布に従うランダム方策関数を用意した。ランダム方策関数は、前章で紹介した方策関数とは異なりパラメータの学習は行わなず、エージェントの 1 単位時間前の行動を受け取り、エージェントの次の行動を以下の式に従い出力する。

$$\begin{aligned} \mu &= a' \\ \sigma &= 30 \text{ or } 180 \\ \pi(a|a') &= \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{(a-\mu)^2}{2\sigma^2}\right] \end{aligned} \tag{4.8}$$

a は方策関数が出力する値、a' は 1 単位時間前の行動、σ は、速度を出力する際は 30、方角を出力する際は 180 となる。ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路の類似度を基準値とし、前章で紹介した方策関数を利用して生成したランニング経路とエキスパートのランニング経路の類似度と基準値との比較によって生成したランニング経路を評価する。

4.4 実験結果

評価実験として、確率密度関数を用いた特徴数 2 の方策関数、確率密度関数を用いた特徴数 14 の方策関数、深層ニューラルネットワークを用いた特徴数 2 の方策関数、深層ニューラルネットワークを用いた特徴数 14 の方策関数、ランダム方策関数のそれぞれを利用してランニング経路を 1000 回ずつ生成し、生成した経路を DTW、Fréchet 距離、Edit 距離の 3 つの指標で評価実験を行った。実験結果を図 4.10〜4.21 に示す。以下の図では、確率密度関数を用いた特徴数 2 の方策関数、確率密度関数を用いた特徴数 14 の方策関数、深層ニューラルネットワークを用いた特徴数 2 の方策関数、深層ニューラルネットワークを用いた特徴数 14 の方策関数のそれぞれの方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW、Fréchet 距離、Edit 距離の値と、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW、Fréchet 距離、Edit 距離の値を示している。

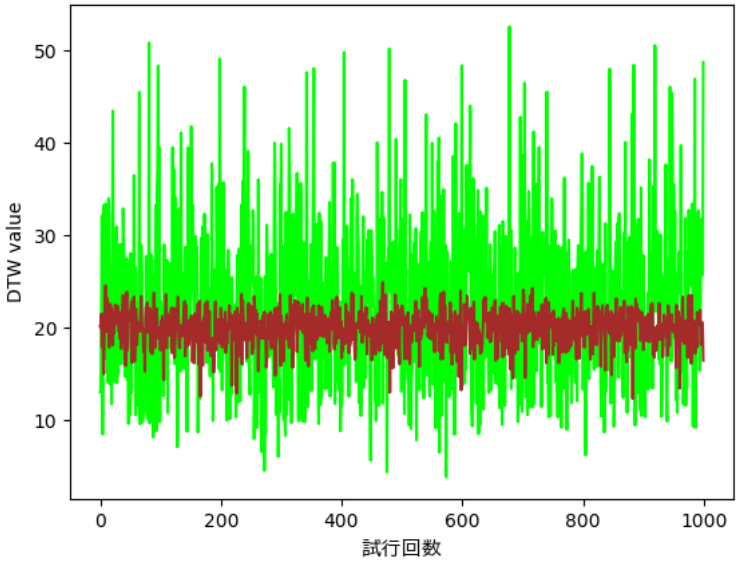

図 4.10　黄緑色の線が、確率密度関数を用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値である。

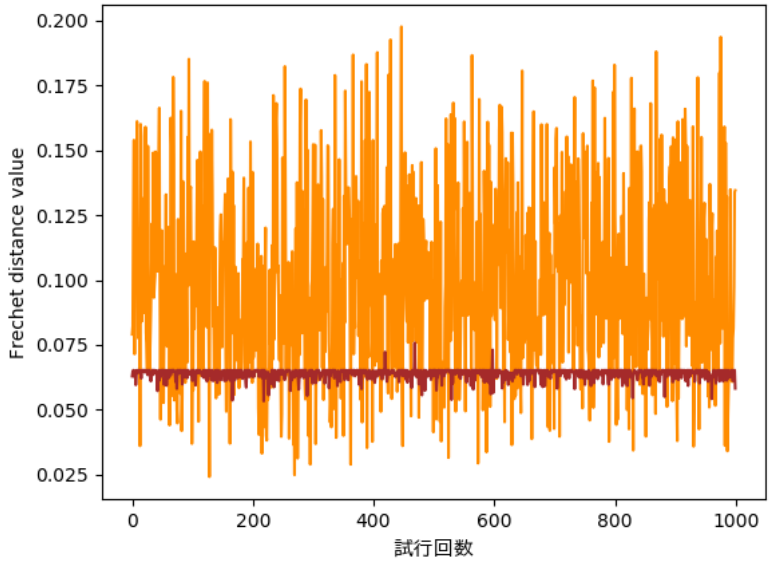

図 4.11　オレンジ色の線が、確率密度関数を用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値である。

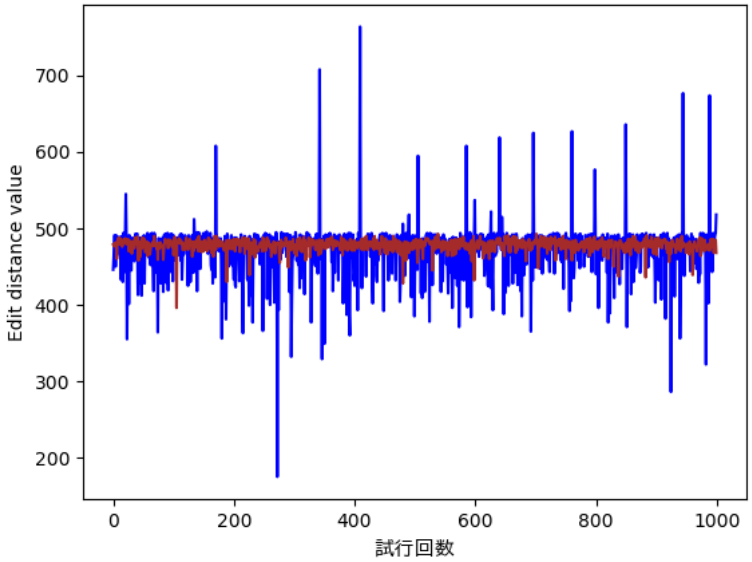

図 4.12　青色の線が、確率密度関数を用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値である。

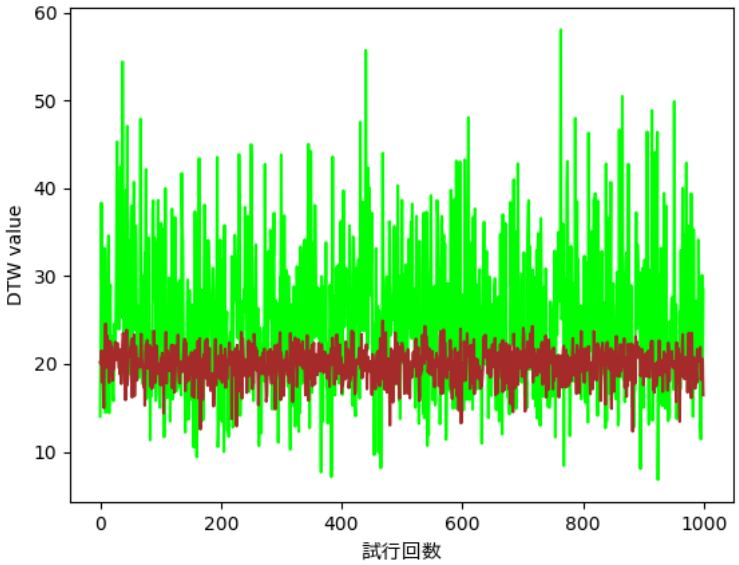

図 4.13　黄緑色の線が、確率密度関数を用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値である。

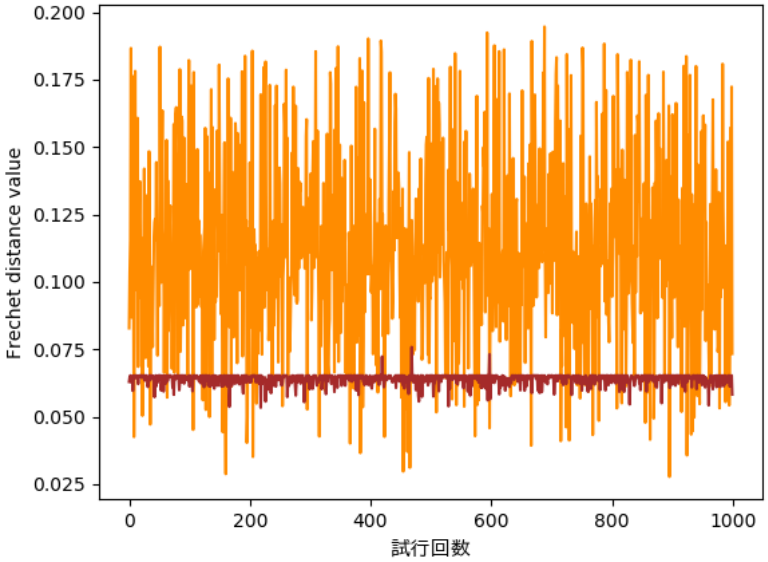

図 4.14　オレンジ色の線が、確率密度関数を用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値である。

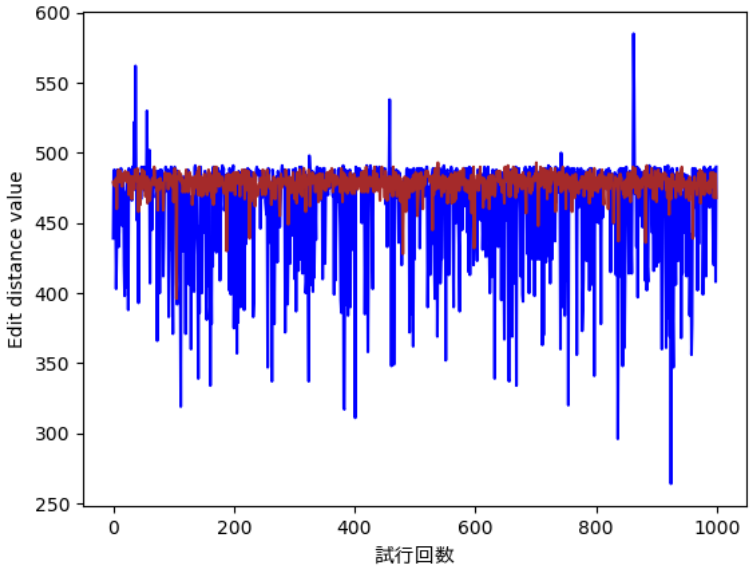

図 4.15　青色の線が、確率密度関数を用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値である。

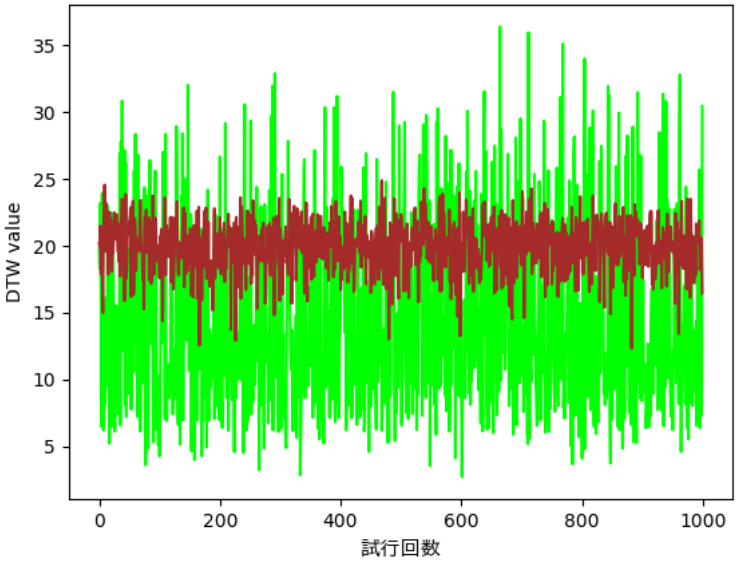

図 4.16　黄緑色の線が、深層ニューラルネットワークを用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値である。

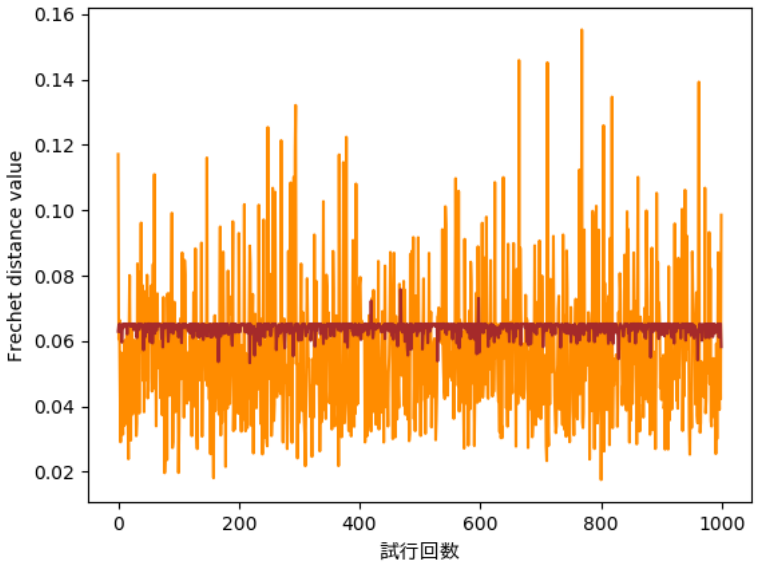

図 4.17　オレンジ色の線が、深層ニューラルネットワークを用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値である。

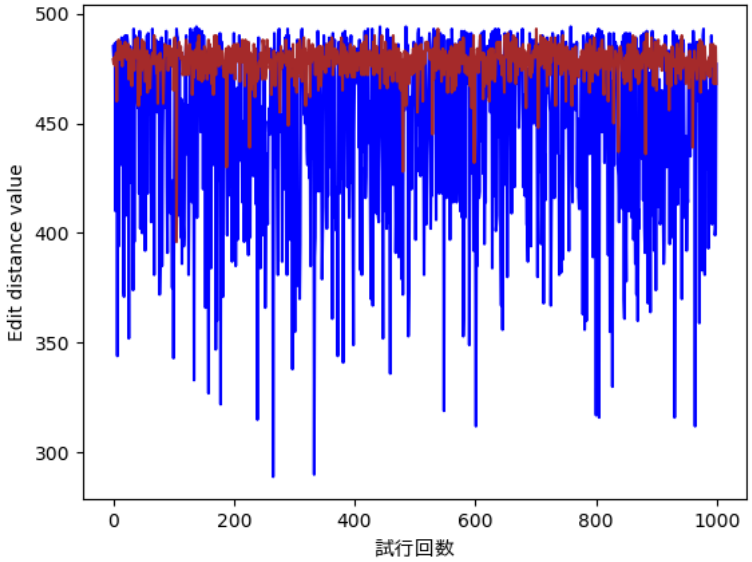

図 4.18　青色の線が、深層ニューラルネットワークを用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値である。

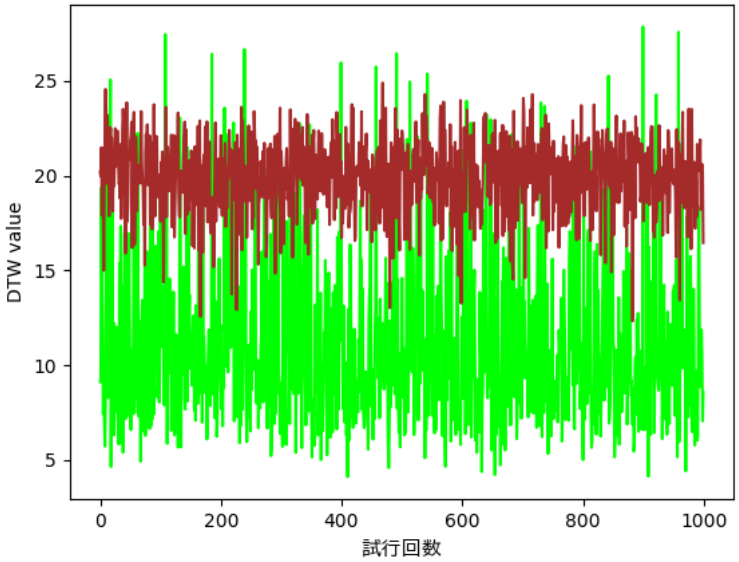

図 4.19　黄緑色の線が、深層ニューラルネットワークを用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値である。

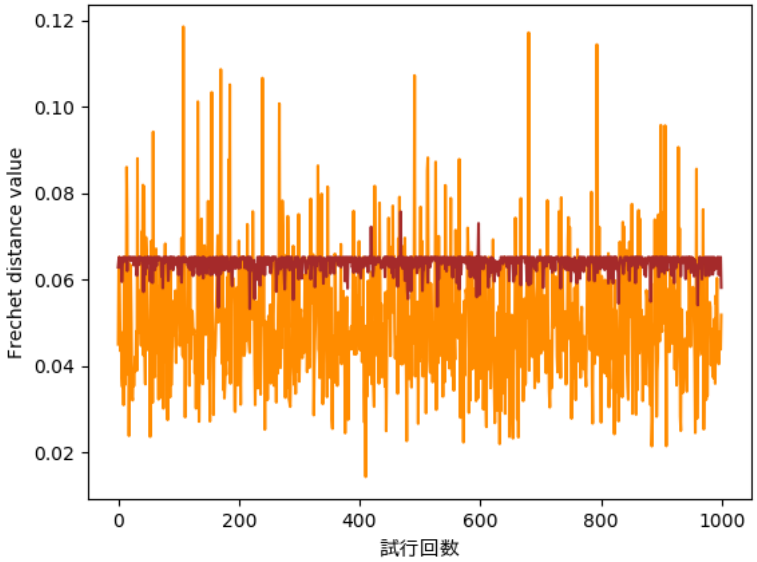

図 4.20　オレンジ色の線が、深層ニューラルネットワークを用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値である。

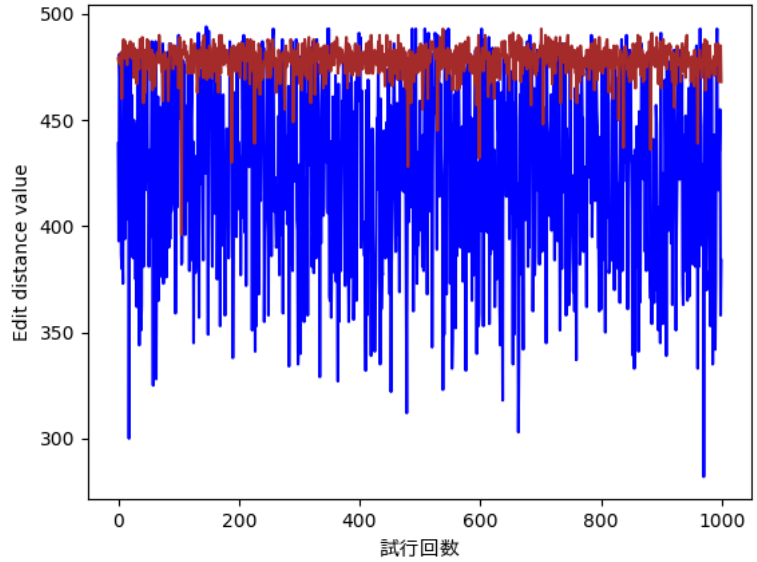

図 4.21　青色の線が、深層ニューラルネットワークを用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値である。

図 4.10〜4.21 に示されている 1000 回の試行を 50 回ごとに分け、その平均値を示したものを図 4.22〜4.24 に示す

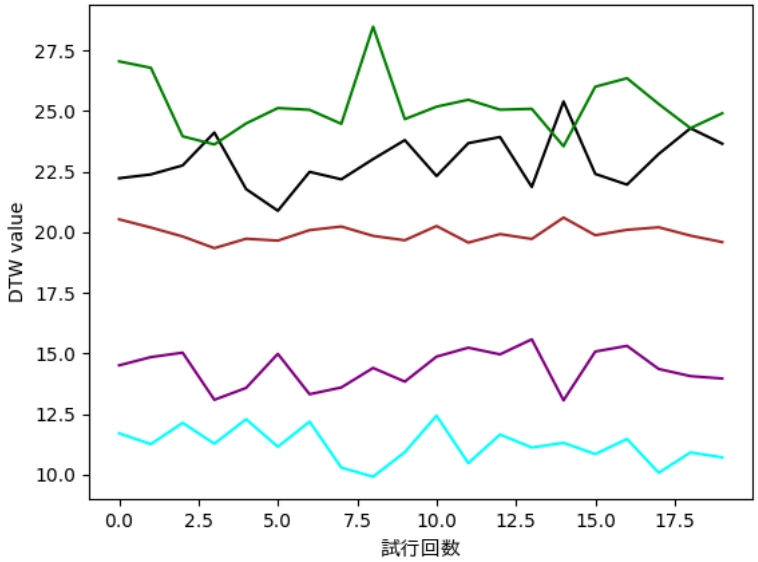

図 4.22　黒色の線が、確率密度関数を用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値の平均値である。緑色の線が、確率密度関数を用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値の平均値である。紫色の線が、深層ニューラルネットワークを用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値の平均値である。水色の線が、深層ニューラルネットワークを用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値の平均値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との DTW 値の平均値である。

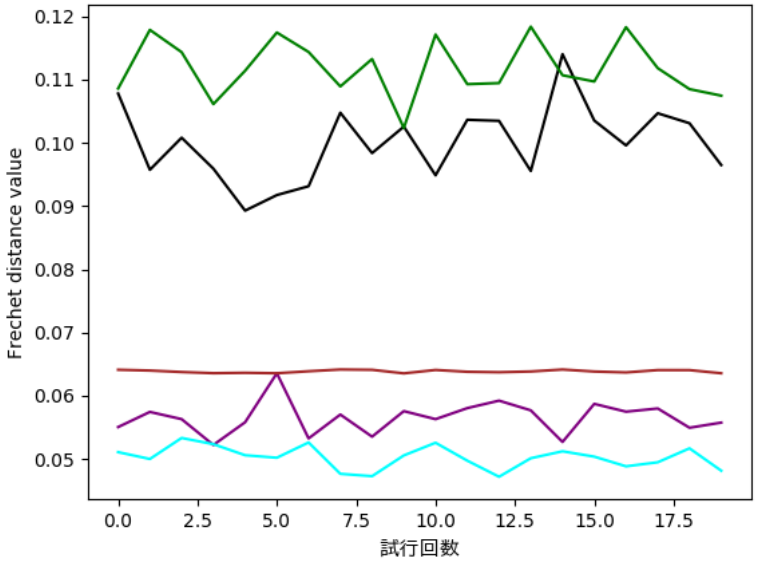

図 4.23　黒色の線が、確率密度関数を用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値の平均値である。緑色の線が、確率密度関数を用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値の平均値である。紫色の線が、深層ニューラルネットワークを用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値の平均値である。水色の線が、深層ニューラルネットワークを用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値の平均値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Fréchet 距離の値の平均値である。

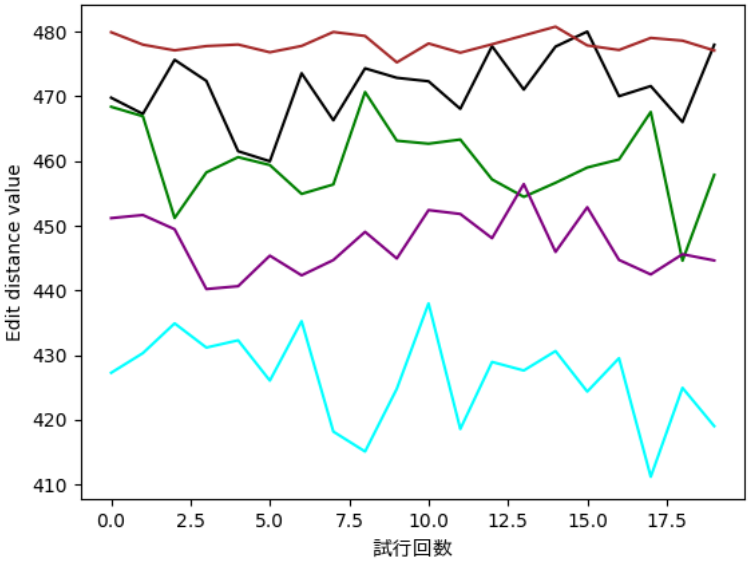

図 4.24　黒色の線が、確率密度関数を用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値の平均値である。緑色の線が、確率密度関数を用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値の平均値である。紫色の線が、深層ニューラルネットワークを用いた特徴数 2 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値の平均値である。水色の線が、深層ニューラルネットワークを用いた特徴数 14 の方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値の平均値である。茶色の線が、ランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との Edit 距離の値の平均値である。

図 4.22〜4.24 に示したように、深層ニューラルネットワークを用いた方策関数を利用して生成したランニング経路とエキスパートのランニング経路との各指標の平均値は、基準値であるランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との各指標の平均値を下回っており、ランダム方策関数を利用して生成したランニング経路と比較した場合、エキスパートのランニング経路との高い類似度を示した。逆に、確率密度関数を用いた方策関数を利用して生成したランニング経路とエキスパートのランニング経路との各指標の平均値は、基準値であるランダム方策関数を利用して生成したランニング経路とエキスパートのランニング経路との各指標の平均値を上回っており、ランダム方策関数を利用して生成したランニング経路と比較した場合、エキスパートのランニング経路との類似度としては低いものとなった。しかし、図 4.10〜4.15 で示されているように、確率密度関数を用いた方策関数での各指標の結果の最小値は、ランダム方策関数での結果の最小値よりも低く、得

られた結果の最小値で評価すると、ランダム方策関数を利用して生成したランニング経路と比較した場合、エキスパートのランニング経路との類似度は高いことを示している。基準値よりも高い類似度を示せたことから、本研究で学習を行った各方策関数を用いた強化学習エージェントは、提案手法を用いて、エキスパートのデータを上手く学習し、適切なパラメータを学習することができたと言える。

図 4.25～??に各方策関数を利用して生成したランニング経路及びエキスパートのランニング経路を示す。

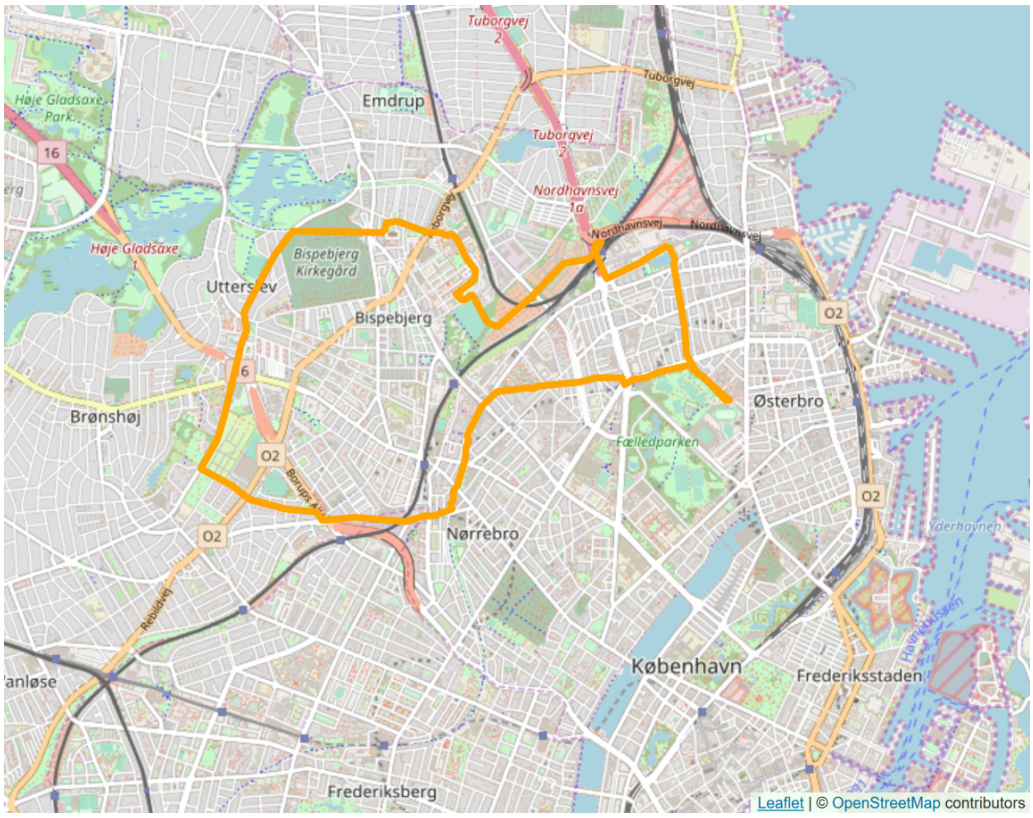

図 4.25　エキスパートのランニング経路

図 4.26　確率密度関数を用いた特徴数 2 の方策関数を利用して生成したランニング経路

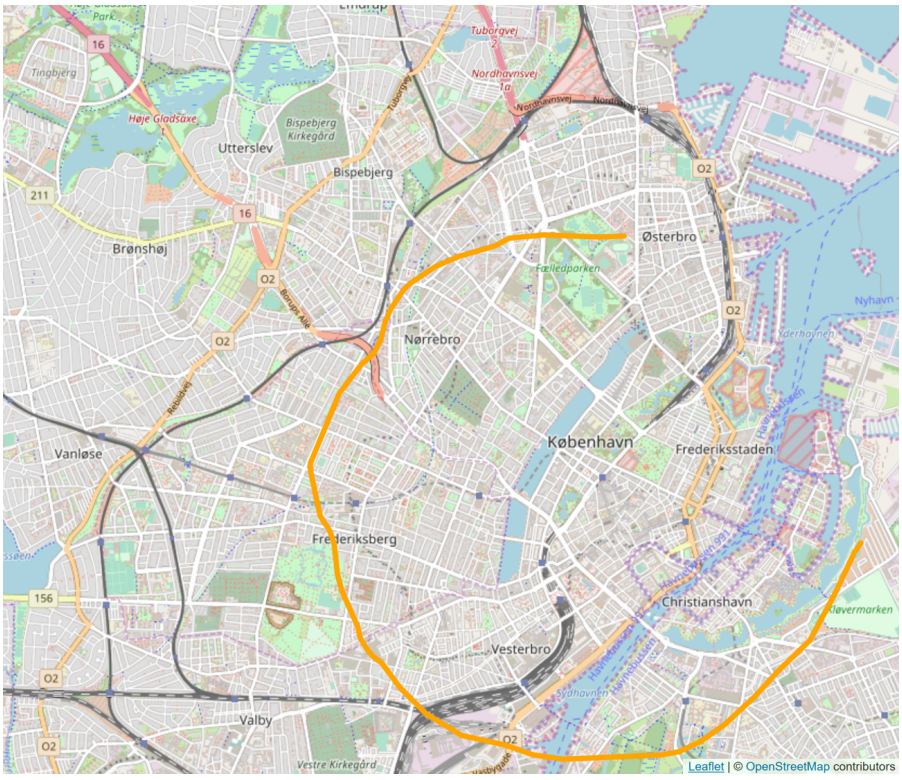

図 4.27　確率密度関数を用いた特徴数 14 の方策関数を利用して生成したランニング経路

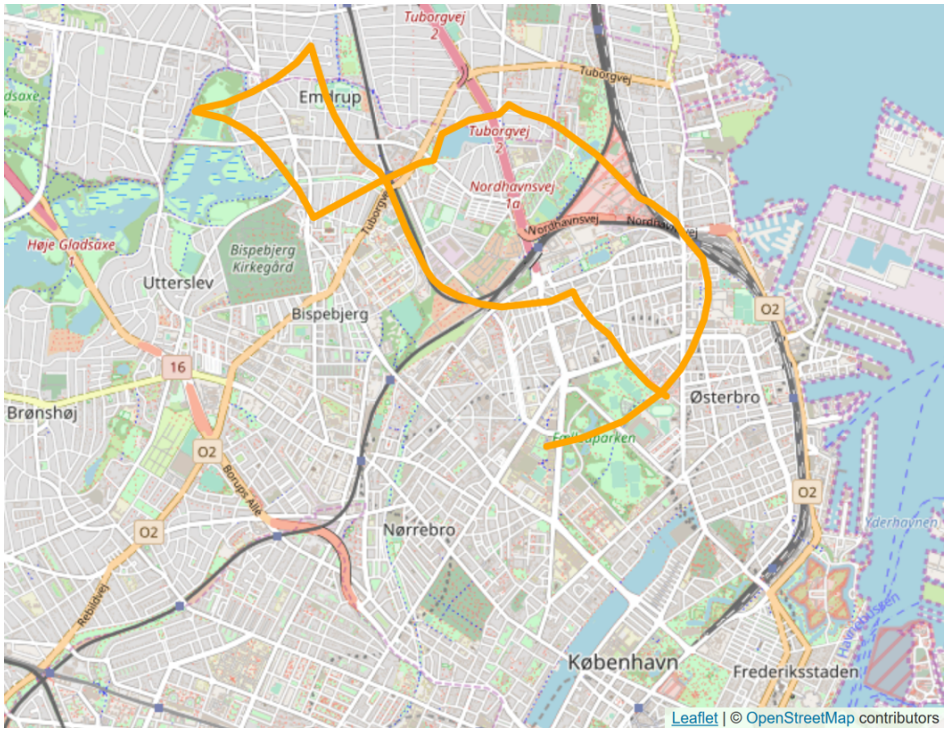

図 4.28　深層ニューラルネットワークを用いた特徴数 2 の方策関数を利用して生成したランニング経路

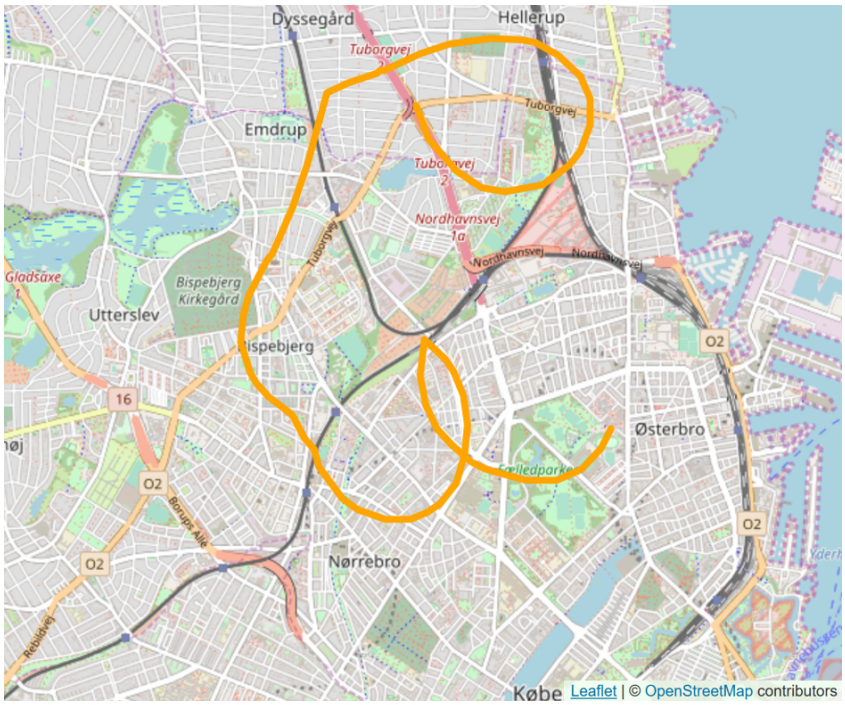

図 4.29　深層ニューラルネットワークを用いた特徴数 14 の方策関数を利用して生成したランニング経路

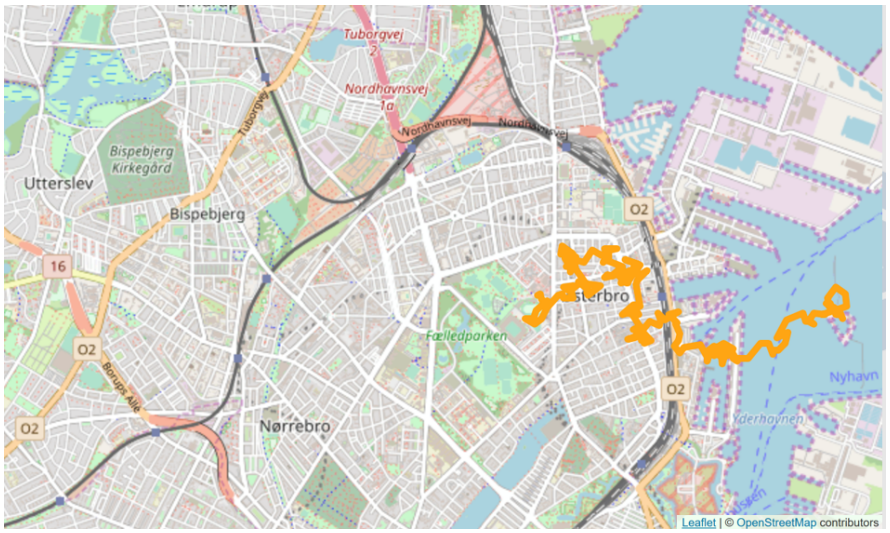

図 4.30　ランダム方策関数を利用して生成したランニング経路

第 5 章　結論

5.1　まとめ

本論文では、運動者に対してランニング経路を推薦するために、強化学習手法の一種である方策勾配法を用いてランニング経路の生成を行う手法を提案した。

強化学習エージェントが受け取る状態や方策の出力値を連続値としたため、連続の状態、行動空間を取り扱うことに長けている方策勾配法を採用した。また、報酬関数推定のため逆強化学習の手法を採用した。方策関数として、確率密度関数を用いた特徴数 2 の方策関数、確率密度関数を用いた特徴数 14 の方策関数、深層ニューラルネットワークを用いた特徴数 2 の方策関数、深層ニューラルネットワークを用いた特徴数 14 の方策関数の 4 種類の方策関数を用意した。

強化学習エージェントが生成したランニング経路の評価を、Trajectory Similarity Measures と呼ばれる軌跡類似度の評価指標を用いてエキスパートのランニング経路との類似度を測る、という形式で評価を行なった。評価結果として、特徴数に関わらず深層ニューラルネットワークを用いた方策関数を利用して生成したランニング経路は、基準値と比べエキスパートのランニング経路との類似度が高くなった。故に提案手法を利用することで、エキスパートのデータの学習及び各パラメータ化された関数のパラメータを上手く学習できたと言える。確率密度関数を用いた方策関数を利用して生成したランニング経路は、基準値と比べエキスパートのランニング経路との類似度が低い結果となった。これは、確率密度関数が深層ニューラルネットワークほど高い表現力を有していないことが原因であると考えられる。しかし、確率密度関数を用いた方策関数を利用して生成したランニング経路の類似度の最小値は、基準の最小値よりも低く、最小値で評価すると、エキスパートのランニング経路との類似度は高いことを示している。このことから、深層ニューラルネットワークを用いた方策関数には劣るが、最低限の学習はできているのではないかと考えられる。利用する特徴数が少ない方策関数よりも、特徴数が多い方策関数を利用した方が高い類似度を示していることから、本研究の目的であるより多くの情報を利用してランニング経路の推薦に関する効果的な手法を開発する、という目的が達成されたと言える。

関連研究の手法では、データベース上にランナーの現在地から走れる経路がない場合、新たに経路を取得し、取得した経路に対して情報を付加するという手順を踏まなければならなかったが、本研究の提案手法であれば、一度強化学習エージェンの学習を行えば、運動者の現在地からのランニング経路をいつでも生成できランニング経路の推薦につなげることができる。

本研究では、関連研究の手法との比較実験は行なっていない。これは、関連研究の推薦システムの実装が困難であり、異なる特徴を持った推薦システム同士の比較も困難だからである。故に提案手法が関連研究の手法と比べて優れているということを直接的に言うことはできないが、関連研究では用いられていない強化学習と深層学習という高度な機械学習技術を用いたという点では新規性があり優れていると言える。本研究は、強化学習手法によりランニング経路の生成を行うことにとどまっているが、提案手法を改良していくことにより、ランニング経路の生成以外にも運動者に対して重要な試合で最良の結果が出せるような適切な練習メニューの生成を行なったり、自動車運転者に対してドライブを楽しめるコースを推薦したり、より広範囲な目的のために利用できると考えられる。このように本研究は、長期的な研究のワンステップとして貢献していると言える。

5.2　今後の展開

今後の展開として、深層ニューラルネットワークを用いた方策関数が確率密度関数を用いた方策関数より良い結果を示したことから、深層ニューラルネットワークの層数を増やしたり、様々なハイパーパラメータの組み合わせを探索することが考えられる。また、ランニング経路の生成は行なったが、具体的なランニング経路の推薦方法は提案することができていないので、生成結果から推薦対象の運動者が好む場所 (景色が綺麗な場所、観光地、公園など) をより多く通るランニング経路を推薦する、などの手法の開発を行うことが考えられる。

第 6 章　謝辞

本研究を行うにあたり、所属研究室の主指導教員である Ho Tu Bao 教授には、ゼミ等の研究室の活動の多くの場面でお世話になりました。いつも適切で丁寧な御指導を賜りました。深く御礼申し上げます。

本論文の審査委員の橋本敬先生，Huynh Nam Van 先生，Dam Hieu Chi 先生に深く御礼申し上げます。また、副テーマ指導の長谷川忍先生をはじめ，北陸先端科学技術大学院大学先端科学技術研究科の全ての先生に御礼申し上げます。研究室の皆様には、日々御助言、御協力いただき深く御礼申し上げます。最後に、様々な面で学生生活を支援してくれた両親に感謝致します。

参考文献

[1] マイトリップ, https://travel.rakuten.co.jp/mytrip/howto/running-island/

[2] Google Maps, https://www.google.co.jp/maps/

[3] Knoch, S., Chapko, A., Emrich, A., Werth, D., and Loos, P., "A context-aware running route recommender learning from user histories using artificial neural net- works." In Database and Expert Systems Applications (DEXA), 2012 23rd International Workshop on, 106-110, 2012.

[4] Issa, H., Guirguis, A., Beshara, S., Agne, S., and Dengel, A., "Preference based Filtering and Recommendations for Running Routes" Proceedings of the 12th International Conference on Web Information Systems and Technologies, WEBIST 2016, Volume 2, 139-146, 2016.

[5] 牧野貴樹・澁谷長史・白川真一編, 『これからの強化学習』, 森北出版, 2016.

[6] Sutton, R. S., and Barto, A. G. Reinforcement Learning : An Introduction, The MIT Press, 1998.

[7] Zhifei, S., and Joo, E. M., "A Review of Inverse Reinforcement Learning Theory and Recent Advances", 2012 IEEE Congress on Evolutionary Computation, 1-8, 2012.

[8] Abbeel, P., and Ng, A, Y., "Apprenticeship Learning via Inverse Reinforcement Learning", In Proceedings of the 21st International Conference on Machine Learning, 2004

[9] Watkins, C. J. C. H., "Learning from Delayed Rewards." Ph.D. thesis, Cambridge, 1989 University.

[10] Rummery, G. A., Niranjan, M., " On-line Q-learning using connectionist systems.", Technical Report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University, 1994.

[11] endomondo, https://www.endomondo.com

[12] 市瀬龍太郎ほか, 『何ができるのか？何が必要なのか？産業利用を考える人のた

めの人工知能・機械学習・ディープラーニング関連技術とその活用』, 情報機構, 2016.

[13] Google Maps Roads API, https://developers.google.com/maps/documentation/roads/intro?hl=ja

[14] Bergstra, J., and Bengio, Y., "Random search for hyper-parameter optimization.", Journal of Machine Learning Research, 281-305, 2012.

[15] scikit-learn, http://scikit-learn.org/stable/

[16] Wang, H., Su, H., Zheng, K., Sadiq, S., and Zhou, X., "An Effectiveness Study on Trajectory Similarity Measures", ADC '13 Proceedings of the Twenty-Fourth Australasian Database Conference, Volume 137, 13-22, 2013.

[17] Toohey, K., and Duckham, M, "Trajectory similarity measures", SIGSPATIAL Special, Volume 7, 43-50, 2015.

[18] Berndt, D, J., and Clifford, J, "Using Dynamic Time Warping to Find Patterns in Time Series", AAAIWS'94 Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 359-370, 1994.

[19] H, Alt and M, Godau, "Computing the Fréchet distance between two polygonal curves", Internat. J. Comput. Geom. Appl, 75-91, 1995.

[20] Chen, L., Ozsu, M, T., and Oria, V., "Robust and fast similarity search for moving object trajectories", SIGMOD '05 Proceedings of the 2005 ACM SIGMOD international conference on Management of data, 491-502, 2005.