1F3-1

エネルギベースドモデルを用いた強化学習のための多層パーセプトロン構造

An Architecture of Multilayer Perceptrons for Energy-based Reinforcement Learning

吉田尚人
Naoto Yoshida

東北大学大学院医工学研究科
Tohoku University, Graduate School of Biomedical Engineering

In this study, the energy-based reinforcement learning (RL), in which our policy is represented by a Boltzmann distribution and an energy function, is investigated. We show the relationships between energy-based actor-critic and conventional actor-critic algorithms. We reveal that the conventional actor-critic algorithms with multilayer perceptrons are the special case which use the cross-entropy as an energy function. Finally we suggest the new architecture, named "twin net", which effectively works in RL domains.

1. はじめに

強化学習は環境に関する事前知識無しに試行錯誤を通して最適行動系列を学習する枠組みであり，強化学習における主要な問題は高次元の状態・行動空間をもつ環境中での効率良い行動選択・学習であることが知られている．このような問題に対して，Actor-Critic アルゴリズムは強化学習における価値関数と行動決定を行う方策関数を明示的に分けることで，行動が連続行動で表現される場合にもコンパクトに強化学習を適用できる手法として多くの研究がなされている [14][1][4]．従来の Actor-Critic アルゴリズムでは K 種類の中から 1 つの行動を選択する 1 of K 型の離散行動や，行動が連続値のベクトルで表される連続行動が主に扱われてきたが，行動の表現にはこの他に 2 値ベクトルで行動が表される行動表現が考えられ，より一般的にはこれらの組み合わせで表現される方策が考えられる．

近年の研究ではエージェントが行動決定を行う方策をエネルギ関数で表現する手法が提案され，高次元の離散状態・行動空間においても効率的に学習が可能である事が示されている [9][8][3][5]．エネルギ関数を方策表現に用いる手法はエネルギベースドモデルと呼ばれる枠組みに基づくものであり [7]，この枠組みでは方策はエネルギ関数を用いた Boltzmann 分布で表される．エネルギ関数に基づく方策表現を用いた先行研究では特に restricted Boltzmann machines (RBMs)[6] をベースにした手法が主に用いられてきた．RBMs を関数近似器に用いる強化学習手法には価値関数をエネルギベースドモデルにおける自由エネルギで近似する価値関数ベースの手法 [9][8][3] と，方策のみエネルギベースドモデルで表現し価値関数は異なる関数で表現する Actor-Critic 手法に基づくアプローチ [5] が提案されている．RBMs を用いた方策はネットワークの隠れ変数が確率的な変数として表現されるため，高次元の行動を用いる場合はギブスサンプリング等のサンプリング手法を用いて近似的に行動選択を行う必要があった．

本論ではエネルギ関数を用いた方策に基づく新たな Actor-Critic アルゴリズムを提案する．先行研究とは異なり，エネルギ関数の隠れ変数を決定論的な関数で表現されるものとすることで，方策を改善する勾配がエネルギ関数の勾配に比例する量として表される事が示される．またエネルギ関数を特定の形で与えることで，2 値ベクトル行動を始めとする，行動空間が非常に大きい場合にも勾配の計算を効率的に行うことが可能となり，かつ行動選択を非常に効率的に行うことが可能となる．さらに本論では Twin net と呼ぶ強化学習問題に応じた特別な多層パーセプトロン構造とエネルギ関数・損失関数を新たに導入する．この損失関数に対して誤差逆伝播法を用いることで，多層パーセプトロンを用いた際により効率的に学習が行われることを示す．

連絡先: naotoyoshida@pfsl.mech.tohoku.ac.jp

2. 理論的背景

2.1 方策勾配定理と Actor-Critic 手法

強化学習における理論的枠組はマルコフ決定過程（MDP）で定式化される．MDP はタプル $< \mathcal{S}, \mathcal{A}, P, R >$ で定義される．ここで $\mathcal{S}$ は状態集合，$\mathcal{A}$ は行動集合，P は環境の遷移確率であり現状態 $s \in \mathcal{S}$ と $a \in \mathcal{A}$ で条件付けされる次の状態 $s' \in \mathcal{S}$ への状態遷移確率 $P(s'|s,a)$ で定義される．R は報酬関数 $r(s,a,s')$ であり，状態遷移の短期的な評価を与える．エージェントは環境中で確率的方策 $\pi(a|s)$ に従い行動を決定するものとする．ここで $\pi(a|s)$ は状態 s を与えられた場合の行動 a 上の確率分布を表す．本論では行動 a が離散的値を取る場合に関して，K 種類の行動から 1 つを選ぶ行動選択を 1 of K 行動選択であると呼び，K ビットの 2 値ベクトルで表現されるような離散行動を 2 値ベクトル行動と呼ぶことにする．各時刻 t でエージェントは状態 s_t を環境から受け取り，行動 a_t を返す．そして環境は次の状態 s_{t+1} に遷移するとともに報酬 $r_t = r(s_t, a_t, s_{t+1})$ をエージェントに渡す．

本論では目的関数に

$$J(\pi) \quad = \quad E\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r_t\Big] = \sum_{s\in\mathcal{S}} d^{\pi}(s) \sum_{a\in\mathcal{A}} \pi(a|s) r_t$$

を考え，強化学習の問題設定ではこの目的関数を最大化する方策を事前知識なしに学習することを考える．ここで γ は割引率と呼ばれる $0 \leq \gamma < 1$ の定数で，$E[\cdot]$ は定常分布 d^{π} と方策 π による期待値演算 $E[f(s,a)] = \sum_s d^{\pi}(s) \sum_a \pi(a|s) f(s,a)$ を表す．$d^{\pi}(s)$ は方策 $\pi(a|s)$ に沿ってエージェントが環境中を行動した場合の割引遷移確率と呼ばれ $d^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s|s_0, \pi)$ で定義される [11]．

強化学習ではさらに行動価値関数 $Q^\pi(s,a)$

$$V^\pi(s) = E_\pi\Big[\sum_{t=1}^{\infty}\gamma^{t-1}r_t\Big|s_0=s\Big]$$
$$Q^\pi(s,a) = E_\pi\Big[\sum_{t=1}^{\infty}\gamma^{t-1}r_t\Big|s_0=s,a_0=a\Big]$$

が定義される [10]．ここで $E_\pi[\cdot]$ は環境中で方策 π に従った経路に関する期待値演算を表す．なお価値関数と行動価値関数には $V^\pi(s)=\sum_a\pi(a|s)Q^\pi(s,a)$ の関係がある．

方策がパラメータ θ による滑らかな関数 $\pi_\theta(a|s)$ で表される場合，目的関数はパラメータ θ の関数 $J(\theta)$ とみなすことができ，$J(\theta)$ に対するパラメータによる勾配が

$$\nabla_\theta J(\theta)=\sum_s d^\pi(s)\sum_a Q^\pi(s,a)\nabla_\theta\pi_\theta(s,a) \tag{1}$$

で与えられる事が知られていて，この関係式は方策勾配定理と呼ばれる [11]．しかし勾配に沿ってパラメータ θ を更新しても，式 1 のままでは更新量の分散が大きすぎ，学習が非常に遅いことが知られている．そこで $\sum_a\pi_\theta(a|s)=1\Leftrightarrow\sum_a\nabla_\theta\pi_\theta(a|s)=0$ の性質から任意の状態に関する関数 $F(s)$ に対して $\sum_a\nabla_\theta\pi_\theta(a|s)F(s)=0$ が成り立つことを利用して，方策勾配定理の右辺は

$$\nabla_\theta J(\theta)=\sum_s d^\pi(s)\sum_a\big(Q^\pi(s,a)-F(s)\big)\nabla_\theta\pi_\theta(a|s)$$

と表すことができる．この $F(s)$ はベースライン関数と呼ばれる [13]．ベースライン関数を $F(s)=V^\pi(s)$ と設定することで方策勾配定理は

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_s d^\pi(s)\sum_a\big(Q^\pi(s,a)-V^\pi(s)\big)\nabla_\theta\pi_\theta(a|s)\\ &= \sum_s d^\pi(s)\sum_a A^\pi(s,a)\nabla_\theta\pi_\theta(a|s) \qquad (2)\\ &= E\Big[A^\pi(s,a)\nabla_\theta\log\pi_\theta(a|s)\Big] \qquad (3)\end{aligned}$$

と表される．行動価値関数から価値関数を引いた関数 $A^\pi(s,a)=Q^\pi(s,a)-V^\pi(s)$ は一般に Advantage 関数と呼ばれる．Advantage 関数は価値関数と行動価値関数の関係から $\sum_a\pi(a|s)A^\pi(s,a)=\sum_a\pi(a|s)Q^\pi(s,a)-V^\pi(s)=0$ の性質がある．

実際の学習では，パラメータ v でパラメタライズされた価値関数の近似関数 $V_v(s)$ を導入し，価値関数の学習時に発生する TD 誤差 $\delta_t=r_t+\gamma V_v(s_{t+1})-V_v(s_t)$ を用いる．TD 誤差はある条件下で Advantage 関数の不偏推定量となることが知られていることから，毎ステップでのパラメータ θ の更新量 $\Delta\theta$ を

$$\Delta\theta_t=\alpha\delta_t\nabla_\theta\log\pi_\theta(a|s) \tag{4}$$

とすることで，方策を改善することができる [2][12]．ここで α は学習率を表す正のパラメータである．

3. エネルギベース Actor-Critic

従来のエネルギベースドモデルに基づく方策では行動空間が大きくなった場合，効率的な行動のサンプリングができなかった．本論ではエネルギ関数を，状態行動が与えられた場合に決定論的な値を返す形で定義することを考える．これによりエネルギ関数に基づく効率的な方策勾配法が導出できる他，既存の Actor-Critic アルゴリズムとの関係性を明らかにすることができる．

3.1 エネルギベースモデルによる Actor-Critic 学習

まず，エネルギベース強化学習では方策 $\pi_\theta(a|s)$ がパラメータ θ により Boltzmann 分布を用いて

$$\pi_\theta(a|s)=\frac{e^{-\beta E_\theta(s,a)}}{\sum_{b\in\mathcal{A}}e^{-\beta E_\theta(s,b)}} \tag{5}$$

で定義されているとする．β は逆温度と呼ばれる正の定数であり，$E_\theta(s,a)$ はエネルギ関数と呼ばれ θ でパラメタライズされた実数関数である．

このとき，次の定理が成り立つ

定理 3..1. 方策関数が式 5 で表される時，方策勾配は以下の式

$$\nabla_\theta J(\theta) = -\beta E\Big[A^\pi(s,a)\nabla_\theta E_\theta(s,a)\Big] \tag{6}$$

で与えられる．

証明 3..1. 式 5 を式 3 に代入すれば，

$$\begin{aligned}\nabla_\theta J(\theta) &= E\Big[A^\pi(s,a)\nabla_\theta\log\pi_\theta(a|s)\Big]\\ &= E\Big[A^\pi(s,a)\nabla_\theta\Big(-\beta E_\theta(s,a)-\log\sum_b e^{-\beta E_\theta(s,b)}\Big)\Big]\\ &= E\Big[A^\pi(s,a)\\ &\quad\times\beta\Big(-\nabla_\theta E_\theta(s,a)+\sum_b\pi_\theta(b|s)\nabla_\theta E_\theta(s,b)\Big)\Big]\\ &= -\beta E\Big[A^\pi(s,a)\nabla_\theta E_\theta(s,a)\Big]\\ &\quad+\beta E\Big[A^\pi(s,a)\sum_b\pi_\theta(b|s)\nabla_\theta E_\theta(s,b)\Big]\\ &= -\beta E\Big[A^\pi(s,a)\nabla_\theta E_\theta(s,a)\Big]\end{aligned}$$

□

実際の学習では通常の Actor-Critic アルゴリズムと同様に価値関数 $V_v(s)$ を導入し，その学習で発生する TD 誤差 δ_t を用いてパラメータ θ の更新量 $\Delta\theta$ を

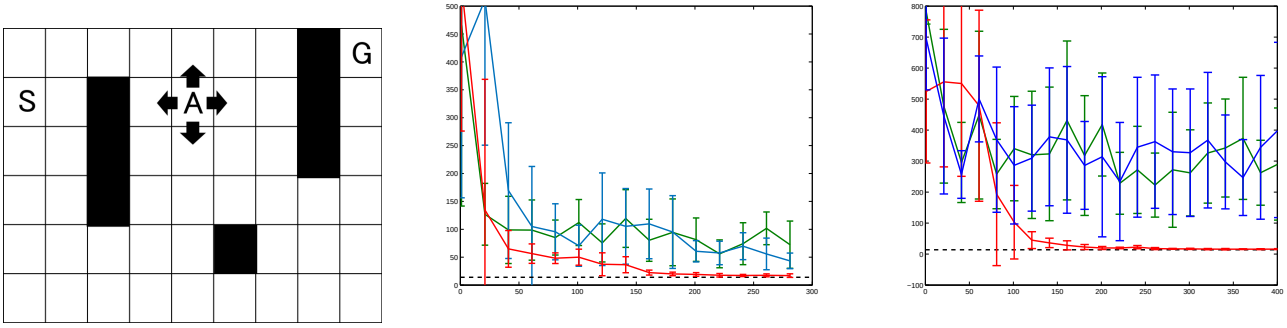
$$\Delta\theta_t=-\alpha\beta\delta_t\nabla_\theta E_\theta(s,a) \tag{7}$$

とすることで，方策を改善する．ここで α はステップサイズパラメータである．

3.2 エネルギ関数の導入

エネルギ関数を構成する一つの方法として，多層パーセプトロンなどによる，決定論的に値が決まる関数 $\mu_\theta(s)$ を導入し，エージェントの行動が離散的な表現（1 of K 行動，2 値ベクトル行動）である場合，クロスエントロピーを用いて

$$E_\theta(s,a)=-\Big(\sum_{i=1}^{K}a^i\log\mu_\theta^i(s)+(1-a^i)\log(1-\mu_\theta^i(s))\Big) \tag{8}$$



(a) (b) (c)

図 1: (a) 格子世界，(b)1 of K 行動表現の場合の結果，(c)2 値ベクトル行動表現の場合の結果.

と定義することでエネルギ関数を構成することが考えられる．ここで x^i はベクトル x の i 番目の要素を表す．このとき $\mu_\theta(s)$ はすべての要素で $0 < \mu^i_\theta(s) < 1$ が満たされるように定義する必要がある．

1-of-K 行動を行動とする場合は式 8 を明示的に計算し，式 5 の確率に基づいて K 個の行動の内の 1 つを選択すれば良い．さらに $\beta = 1$ かつ $\mu_\theta(s)$ の出力が softmax 関数として正規化されていれば，アルゴリズムは $\mu_\theta(s)$ を出力とした従来の方策勾配による更新に一致する．2 値ベクトル行動の場合，$\beta = 1$ かつ $\mu_\theta(s)$ の出力を各行動要素で sigmoid 関数にすれば，各行動要素で $P(a_i = 1|s) = \mu^i_\theta(s)$ となる．

3.3 提案する構造：Twin net

本論ではエネルギを 2 つの関数に分割し，学習は Actor 全体のパラメータを $\theta = \{\theta_p, \theta_n\}$ とすることで

$$E_\theta(s,a) = E_{\theta_p}(s,a) - E_{\theta_n}(s,a) \tag{9}$$

と定義する事を考える．$E_{\theta_p}(s,a)$ と $E_{\theta_n}(s,a)$ はそれぞれ θ_p と θ_n をパラメータとするエネルギ関数である．方策は同様に式 5 で表される．このエネルギに対して損失関数

$$\mathcal{L}_\theta(s,a) = h(\delta)E_{\theta_p}(s,a) + (1 - h(\delta))E_{\theta_n}(s,a) \tag{10}$$

を定義し，この損失関数に基いて学習を行うものとする．$h(\cdot)$ は Heaviside 関数である．学習の更新式は

$$\Delta\theta = -\alpha\nabla_\theta\mathcal{L}_\theta(s,a)$$

で表されるものとする．本論では $E_{\theta_p}(s,a)$ と $E_{\theta_n}(s,a)$ に対応する $\mu_{\theta_p}(s,a)$，$\mu_{\theta_n}(s,a)$ をそれぞれ多層パーセプトロンで構成する．前者を Appetitive net，後者を Aversive net と呼ぶことにし，全体の構造を Twin net と呼ぶことにする．両エネルギ関数をともに式 8 で定義すれば，方策の各要素に対する確率は以下で表される．

$$\pi_\theta(a^i = 1|s) = \Big(\frac{\mu^i_{\theta_p}(s)}{\mu^i_{\theta_n}(s)}\Big)^\beta \Big/ \Big\{\Big(\frac{\mu^i_{\theta_p}(s)}{\mu^i_{\theta_n}(s)}\Big)^\beta + \Big(\frac{1-\mu^i_{\theta_p}(s)}{1-\mu^i_{\theta_n}(s)}\Big)^\beta\Big\}$$

4. 実験

提案した Twin net による Actor-Critic アルゴリズム，提案手法を用いない全結合パーセプトロンによるエネルギー学習および CACLA との比較を 2 つの異なるタスクで行った．

4.1 エージェントの設定

実験で使用する Critic を構成する価値関数は，状態 s に対応する特徴量を $\phi(s)$ とし v をパラメータベクトルとすると，線形関数 $V_v(s) = v^{\mathrm{T}}\phi(s)$ で近似されているものとした．状態遷移 $s_t \to s_{t+1}$ に対して価値関数の学習は TD 学習

$$\begin{aligned}\delta &= r_t + \gamma V_v(s_{t+1}) - V_v(s_t)\\ \Delta v_t &= \alpha_c\delta\nabla_v V_v(s_t)\end{aligned}$$

を用いた．α_c は Critic の学習率である．Actor にはすべて隠れユニットが 20 個の 3 層パーセプトロンを用い [*1]，隠れユニットの活性化関数は sigmoid 関数を用いた．出力には格子世界（GW）では softmax 関数，2 値ベクトル行動を用いた格子世界（GW-BV）では行動の各要素ごとに sigmoid 関数を用いた．

提案する Twin net にはそれぞれ 1 つずつ 3 層パーセプトロンを用い，各エネルギ関数にはクロスエントロピー関数を用いた．Actor の更新には

$$\Delta\theta_t = -\alpha_a\nabla_\theta\mathcal{L}_\theta(s,a)$$

とした．実験では v すべての要素をゼロに初期化し，θ は出力層に接続する重みは全てゼロとし，その他の重みに対しては各ユニットにおいて入力するユニットが N_{in} 個の場合に対して $[-\frac{1}{N_{in}}, \frac{1}{N_{in}}]$ の範囲の一様分布からサンプルした．

比較には，Actor が上記と同様の構成による全結合の 3 層パーセプトロンが 1 つだけの場合を用いた．具体的には Actor はエネルギ関数による更新

$$\Delta\theta_t = -\alpha_a f(\beta\delta_t)\nabla_\theta E_\theta(s,a)$$

を用いて学習するものとし，$f(x) = x$ の場合（Normal）と，CACLA に対応する Heaviside 関数 $f(x) = h(x)$ の場合（CACLA）を用いた．ただし Normal については GW-BV では実験では極めて学習が遅かったため、このドメインでは符号関数 $f(x) = \mathrm{sign}(x)$ を用いた．$\beta = 1$，$\gamma = 0.95$ とした．従って Normal は従来の方策勾配法に一致する．Critic の学習では $\alpha_c = 0.1$ を用いた．α_a は各手法ごとに $\{1.0, 0.6, 0.3\} \times 10^{-I}$，$I = 1, 2, 3, 4, 5$ の値で最も良い性能を示した値を選び，表 1 の通りに設定した．

*1 ネットワークを 1 つしか用いない手法に対し隠れユニット数が 50 と 100 の場合も予備実験として行ったが，本文の結果と大きな差は見られなかった．

表 1: Learning rates of the actor networks

Method	Normal	CACLA	Twin net
GW	0.1	0.003	0.06
GW-BV	0.1	0.03	0.1

表 2: Binary action vectors

Action	Binary Vector
North	1,1,0,0
South	0,0,1,1
East	1,0,1,0
West	0,1,0,1
Stay	otherwise

4.1.1 格子世界（GW）での比較

この環境では 2 値ベクトル行動による行動選択を確認するため，Grid World での最短経路問題を考える．Grid World は図 1(a) に示される，Sutton により考案された環境を用いることにする．環境は全 47 状態で構成されており，各状態に対応する特徴量 $\phi(s)$ は 47 ビットのベクトルで構成されており，各状態で対応する 1 ビットのみが 1 でそれ以外は 0 を返す．エージェントの行動は東西南北の方向に移動する 4 つの行動とを選択可能で，移動する場合はその方向に壁がない場合，確率 1 で対応する方向へ 1 マス移動するものとした．エージェントは毎エピソードの開始時に状態 “S”からスタートし，状態 “G”に入ることでエピソードが終了するものとした．エージェントの報酬はゴール到達時に +1 を受けるものとし，その他はゼロとした．1 エピソードはエージェントがゴール状態に入るか，あるいは 800 ステップが経過した場合に終了するものとし，次のエピソードに移るものとした．

Figure 1(b) に 10 回の実験の平均値と標準偏差を示す．縦軸はスタートからゴールまでのステップ数であり，横軸はエピソード数を表す．結果から Twin net による学習手法は 3 つの手法のうち最も速く学習が行われている事がわかる．Figure 1(b) 中の破線はエージェントが最短経路をとった時の最小ステップ（14 ステップ）を表しており，提案手法でのみ最短経路が学習されている事が示されている．

4.1.2 2 値ベクトル行動を必要とする格子世界（GW-BV）での比較

この環境は格子世界と同じマップ・エピソードのルールを用いるが，行動が 4 ビットの 2 値ベクトルでコードされている点でよりチャレンジングな内容になっている．4 ビットの行動ベクトルは $2^4 = 16$ 種類ある行動のうち表 2 に示される 4 つのパターンのみが各 4 方向への行動に対応するものとし，それ以外はその場に留まるものとした．従って，この環境でエージェントが最短経路でゴール状態へ移動するためには，各状態で移動に必要な適切な行動パターンを学習する必要がある．このタスクでは報酬は毎ステップ −1 が与えられるものとした．

Figure 1(c) は 10 回の実験の平均値と標準偏差を示す．縦軸はスタートからゴールまでのステップ数であり，横軸はエピソード数を表す．結果から Twin net による学習手法は 3 つの手法のうち最も速く学習が行われている事がわかる．Figure 1(c) 中の broken line はエージェントが最短経路をとった時の最小ステップ（14 ステップ）を表しており，Twin net を用いた手法でのみ最短経路が学習されている事が示されている．

参考文献

[1] Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *Systems, Man and Cybernetics, IEEE Transactions on*, (5):834–846, 1983.

[2] Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor–critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[3] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Scaled free-energy based reinforcement learning for robust and efficient learning in high-dimensional state spaces. *Frontiers in neurorobotics*, 7, 2013.

[4] Ivo Grondman, Lucian Busoniu, Gabriel AD Lopes, and Robert Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1291–1307, 2012.

[5] Nicolas Heess, David Silver, and Yee Whye Teh. Actor-critic reinforcement learning with energy-based policies. In *EWRL*, pages 43–58. Citeseer, 2012.

[6] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[7] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 2006.

[8] Makoto Ohtsuka. Goal-oriented representations of the external world: A free-energy-based approach. 2010.

[9] Brian Sallans and Geoffrey E Hinton. Reinforcement learning with factored states and actions. *The Journal of Machine Learning Research*, 5:1063–1088, 2004.

[10] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998.

[11] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, volume 99, pages 1057–1063. Citeseer, 1999.

[12] Hado Van Hasselt. Reinforcement learning in continuous state and action spaces. In *Reinforcement Learning*, pages 207–251. Springer, 2012.

[13] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[14] Ian H Witten. An adaptive optimal controller for discrete-time markov environments. *Information and control*, 34(4):286–295, 1977.