選択的不感化ニューラルネットを用いた強化学習の価値関数近似

新保　智之[†a)]　山根　健[†]　田中　文英[†]　森田　昌彦[†b)]

Value Function Approximation in Reinforcement Learning Using Selective Desensitization Neural Networks

Tomoyuki SHIMBO[†a)], Ken YAMANE[†], Fumihide TANAKA[†], and Masahiko MORITA[†b)]

あらまし　連続状態空間で強化学習を行う場合，価値関数を少ないサンプルで精度良く近似することが重要であるが，従来用いられてきた局所的近似手法は，近似精度と学習効率の両立が困難である上に，状態空間の次元が高くなると学習時間や計算コストが爆発的に増大するという問題を抱えている．本研究では，選択的不感化ニューラルネットを用いて関数近似器を構成するとともに，これによって価値関数を近似することでこの問題が大きく改善できることを示す．アクロボットの振り上げ課題を用いた実験の結果，本手法は学習効率が近似精度の割に高く，冗長変数を加えてもほとんど低下しない，状態空間の次元が増えても計算コストの爆発的増加が生じない，オンライン学習が可能など，実空間における強化学習に適した性質を備えることが分かった．この結果は，膨大な情報の中から必要な情報だけを抽出する情報処理技術の開発にもつながると考えられる．

キーワード　SDNN，関数近似器，Q 学習，冗長次元，アクロボット

1. まえがき

人間を含む動物は，もともと大量かつ多様な情報が存在する現実世界において，常にほぼ適切な行動を取ることができる．これは，環境に含まれる膨大な情報の中から，その時々において必要な情報を取捨選択しているためだと考えられる．一方，ロボットの行動制御を行う場合，現状では情報あるいは環境変数の取捨選択を事前に操作者の手で行わざるを得ない．そのため，複雑な環境や未知の環境に適応可能な自律ロボットを実現することは非常に困難である．このように，大量で多様な情報から必要とする情報を効率的に取り出すことは，実空間における行動制御にとっても大きな課題である．逆に，実空間における行動制御の研究により，爆発する情報を効率的に扱うための重要な手掛りが得られる可能性がある．

動物の行動学習をモデル化し，環境に対して自律的に適応する学習の枠組みとして，強化学習 [1] がある．しかし，現在の強化学習システムは，一般に学習効率が悪く，状態空間が広いと非常に長い学習時間を要する．そのため，状態を離散化して状態数を減らすか，状態空間の次元（状態変数の数）を非常に低く抑える必要があった．つまり，動物の行動学習のモデルといいながら，実空間で生きる動物との間には非常に大きな差がある．強化学習では，各状態や行動の価値を正しく評価することが重要であるが，状態空間が連続である場合，行動主体（エージェント）は有限個の状態しか経験できないため，多くの未経験の状態に関する価値は，経験した状態の価値から推定する必要がある．これは，有限の点における関数値（サンプル）から未知の点における関数値を近似する一種の関数近似問題とみなすことができる．この関数近似の問題は，価値関数の更新則（学習アルゴリズム）とはまた別の大きな問題である．

これまで，強化学習を効率化するために，主に学習アルゴリズムに関する改良が図られてきた．しかし，このような改良は，一般に計算の複雑化や必要なメモリ量の増大を伴う．また，今のところ動物がそうした複雑な学習アルゴリズムを用いているという明確な証

† 筑波大学大学院システム情報工学研究科，つくば市
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba-shi, 305–8573 Japan
a) E-mail: shin@bcl.esys.tsukuba.ac.jp
b) E-mail: mor@bcl.esys.tsukuba.ac.jp

拠はない．我々は，上述した動物との大きな差は，学習アルゴリズムの違いによるものではなく，価値関数の近似器の違いに起因すると考えている．

既存の関数近似の手法には，大きく分けて線形和モデルなどの大域的近似手法と，テーブル参照法やタイルコーディング [1]，放射状基底関数ネットワーク（RBFN）[1], [2] のような局所的近似手法とがある．前者は効率的な近似が期待できるが，関数形がある程度分かっていないと適用できない．一方後者は，空間の分割数または局所的基底関数の数を増やせば，どんな関数でも精度良く近似できるが，必要なメモリ量が次元に対して指数的に増大する上，汎化が局所的にしか生じないため，非常に多数のサンプルが必要である．また，ニューラルネットは一概にどちらとも分類できないが，単層パーセプトロンは前者に近く，多層パーセプトロンは後者に近い．

通常，強化学習の価値関数は，事前に関数形が分からない上に，非線形性が非常に強く，部分的に不連続なこともある．そのため，一般に価値関数近似に大域的手法は適さない．ニューラルネットを適用した例はいくつかあるが [3]〜[5]，やはり一般的には適さないといわれている [6], [7]．結果的に，これまで強化学習で用いられてきた関数近似器は，ほとんどが局所的手法に基づく [1]．そしてこのことは，状態空間の次元を十分に低くしなければならないことを意味する．

しかしながら，状態空間を低次元化するのはそう容易ではない．統計的手法などにより次元を自動的に圧縮する方法は，一般に多くのサンプルや多数回の繰返し計算が必要であるため，強化学習を実行しながら行うことは困難である．そのため，これまで強化学習を実際の課題に適用する際には，あらかじめ人間ができるだけ冗長性がないよう状態空間を設計することが事実上不可欠であった．後述のアクロボットの振り上げ課題を例にとるならば，各関節の角度と角速度を状態変数として与えるが，重心の位置や角加速度は冗長な情報なので状態変数に加えない場合が多く，当然時刻や気温など無関係の情報も加えない．しかし，よく考えれば，これはシステムに非常に大きな事前知識を与えていることになる．

これに対して，実世界は極めて高次元で冗長性の高い状態空間であるにもかかわらず，動物はどの情報（状態変数）が重要でどれが冗長なのか，だれから教わるわけでもない．また，そもそも何が重要な情報であるかは状況によって変化するので，先天的知識（遺伝）によって情報を選別し次元を削減するのにも限界がある．したがって，動物が用いている関数近似器は，入力がどれだけ高次元で冗長であっても，出力値を効率的に近似でき，計算に要する時間や資源の爆発も抑えられるようなものと考えられる．

このような性質をもつ近似器はこれまで知られていないが，脳の情報処理をモデル化したニューラルネットによって工学的に実現できるかもしれないと考えるのは自然であろう．実際，かつて多層パーセプトロンが大いに期待された時期があったが，その一つの理由は，任意の連続関数を任意の精度で近似できることが理論的に示されたことにある [8]．しかし，多層パーセプトロンによる関数近似には，学習の収束性が悪い，過剰適合によって大きな汎化誤差が生じる，サンプルの追加学習が難しい（全サンプルについて再学習が必要）といった数々の問題がある．しかも，大域的な汎化を期待しているのに，ごく少数の素子（ニューロン）を用いてうまく近似できる関数の場合を除き，実際には局所的な汎化しか生じないなど，メリットがあまりないことが経験的に知られるようになり，次第に使われることも少なくなってきた．

近年我々は，多層パーセプトロンを 2 変数以上の関数の近似に適用した場合，入力層に分散表現を用いたとしても大域的な汎化は全く生じないこと，その原因が 1 対多対応による荷重の平均化にあること，そして選択的不感化（Selective Desensitization）という手法によってこの問題が解決されることを明らかにした [9]．この選択的不感化法は，誤差逆伝搬（BP）法のような複雑な計算を必要としないだけでなく，脳内で実際に用いられている可能性が示されている [10], [11]．これらのことから，選択的不感化法を適用したニューラルネットと分散表現を組み合わせることによって，動物が用いているような関数近似器が構成できる可能性があると考えられる．

そこで本研究では，選択的不感化ニューラルネット（SDNN）によって価値関数の近似器を構成し，その効果を実験的に検証する．具体的には，アクロボットの振り上げ課題 [1] に適用することによって，冗長な状態変数を追加したときの学習効率や計算量の変化について調べる．

2. SDNN による関数近似

ある神経素子の出力を，入力または内部電位に関係なく中立値（出力の期待値）にすることを「不感化」

という．ここでは，±1 の 2 値を偏りなく出力するような素子を考え，不感化された場合には 0 を出力するものとする．また，ある素子群のうち不感化するものを別の情報（修飾パターン）に応じて決め，これによって 2 種類の情報を統合する手法を「選択的不感化法」という．

具体的には，2 種類の n 次元の 2 値パターン $S = (s_1, \ldots, s_n)$ 及び $C = (c_1, \ldots, c_n)$ があるとき，S を活動パターンとする素子群のうち約半数の素子をパターン C に応じて選んで不感化する．その結果，素子群の出力パターンは約半数が 0 で残りが ±1 の 3 値パターンとなるが，このパターンのことを「C によって修飾された S」といい，$S(C)$ で表す．同様に，S によって修飾された C を考えることもできる．

このような選択的不感化による修飾を，層状のニューラルネットに適用したものが SDNN である．SDNN によって 2 変数の関数を近似する方法は既に提案されており，その有効性も示されている [9] が，2 変数以下の関数にしか使えなければあまり有用ではない．そこで本研究では，まずこれを拡張して m 変数の関数近似器を構成した．この近似器は，図 1 に示すような 3 層構造をしている．

第 1 層は入力層であり，それぞれ n 個の素子からなる m 個の素子群で構成される．μ 番目の素子群は，入力変数 x^μ の値を ±1 がほぼ同数の 2 値パターン $S^\mu = (s_1^\mu, \ldots, s_n^\mu)$ によって分散表現する．このようなパターンのことをコードパターンと呼ぶ．例えば入力変数が -1 から 1 の連続値をとり，200 個のコードパターンがある場合，各パターンは 0.01 の刻み幅で量子化された区間を表現する．

ここで重要なのは，近い区間ほど類似した（大きな相関をもつ）コードパターンによって表現することである．すなわち，変数の値が連続的に変化すると，それに伴ってコードパターンが徐々に変化し，もとのパターンとの相関が 1 から 0 まで連続的に減少していくようにする（相関が負にはならない方がよい）．このようなコードパターンを用いることによって汎化が生じるが，コードパターンの数すなわち表現の分解能に比例した数の素子が必要となる．

具体的に，次章の実験では次のように作成したコードパターンを用いている．まず，200 次元の 2 値パターンをランダムに 9 個作成して $C^1, C^2 \ldots, C^9$ とする．ただし，変数 x^1 及び x^2（θ_1 及び θ_2 を正規化したもの）に関しては，$-\pi$ [rad] と π [rad] は同じ状態

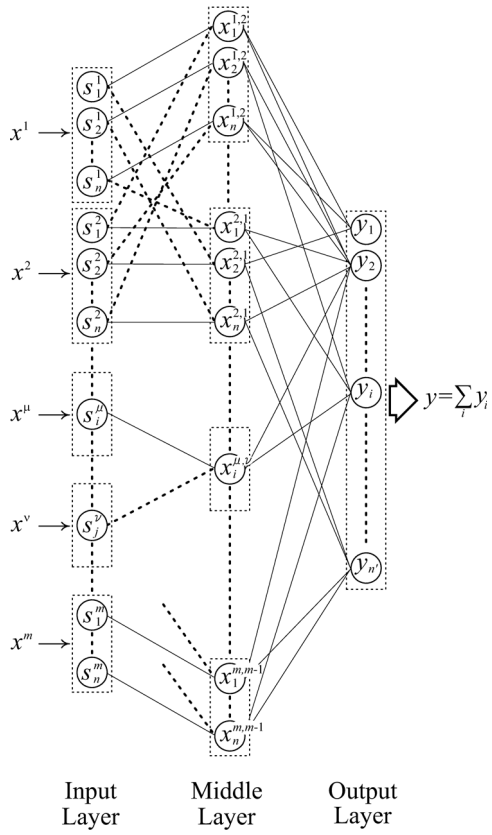


図 1　SDNN による m 変数関数近似器の構成
Fig. 1　m-values function approximator using a selective desensitization neural network.

を表すため，$C^1 = C^9$ とする．次に，C^ν と $C^{\nu+1}$ の間（$\nu = 1, \ldots, 8$）をそれぞれ間を 45 個のパターンで補間し，計 360 個のコードパターンを得る．

第 2 層（中間層）は，それぞれ n 個の素子からなる $m(m-1)$ 個の素子群で構成される．各素子群は，入力層のある素子群の出力パターン S^μ をそのまま受け取るとともに，別の素子群の出力パターン S^ν による修飾を受ける．すなわち，中間層の出力パターンは，$S^\mu(S^\nu)$ をすべての μ，ν の組合せ（ただし $\nu \neq \mu$）について並べたものである．

ここで，各素子群の i 番目の素子は，修飾パターン S^ν の成分 s_i^ν をランダムな順序で並べ替え，その i 番目の成分 $s_{\sigma(i)}^\nu$ が -1 のときに不感化されるものとする（σ は順列の置換を表す）．これは，単純に i 番目の成分同士を対応づけると，中間素子群の活動パターンが偏ったり，本来異なるべきパターン間に高い相関が生じたりして，性能が大きく低下するおそれがあるから

である．特に，異なる変数に対して同じコードパターンを使う場合，このような操作は不可欠である（置換 σ も μ 及び ν によって変えた方がよい）．式で表すならば，$S^{\mu}(S^{\nu})$ に対応する素子群の i 番目の素子の出力 $x_i^{\mu,\nu}$ は，

$$x_i^{\mu,\nu} = \frac{1 + s^{\nu}_{\sigma(i)}}{2}\ s_i^{\mu} \tag{1}$$

となる．

第 3 層は出力層であり，出力変数 y の値を表現する n' 個の素子からなる（出力変数が複数個ある場合は，複数の素子群で構成される）．各出力素子は中間層のすべての素子と結合しており，i 番目の素子の出力 y_i は

$$y_i = \phi\left(\sum_{\mu,\nu(\neq\mu)}\sum_{j=1}^{n} w_{ij}^{\mu,\nu} x_j^{\mu,\nu} - h_i\right) \tag{2}$$

で与えられる．ここで $w_{ij}^{\mu,\nu}$ は中間素子からの結合荷重，$\phi(u)$ は $u > 0$ のとき 1 それ以外では 0 をとる関数，h_i は i 番目の出力素子のしきい値である．n' 個の出力素子の出力値の合計 $y = \sum_i y_i$ がこの関数近似器の出力であり，これを適当にスケール変換することによって目的の関数の近似値が得られる．

中間層から出力層への結合荷重は，一種の誤り訂正学習によって更新する．具体的には，もし y が目標値 $\hat{y}$ より大きければ，1 を出力している出力素子のうち $y - \hat{y}$ 個（なるべく内部電位の小さい素子を選ぶとよい）について 0 を出力するよう修正し，もし y が $\hat{y}$ より小さければ，0 を出力している素子について同様に修正する．式で表すと，

$$w_{ij}^{\mu,\nu} \leftarrow w_{ij}^{\mu,\nu} + c(\hat{y}_i - y_i)x_j^{\mu,\nu} \tag{3}$$

$$h_i \leftarrow h_i - c(\hat{y}_i - y_i) \tag{4}$$

となる．次章の実験では，学習係数 c を小さめに設定した上で，$|\hat{y} - y|$ が 0 でない場合に 1 回のみ結合荷重の更新を行っている．

3. 実験方法

3.1 実験課題

強化学習の性能を評価するためのベンチマークとして，アクロボットの振り上げ課題を用いる．この課題は，連続状態空間における強化学習の性能評価にしばしば用いられている [3], [4], [6], [12].

アクロボット [13] とは，図 2 に示す 2 リンク 2 関節のロボットであり，人間の鉄棒運動をモデル化したものである．アクチュエータが第 2 関節にのみ存在する劣駆動系であり，非線形性も強いため，制御は比較的難しい．

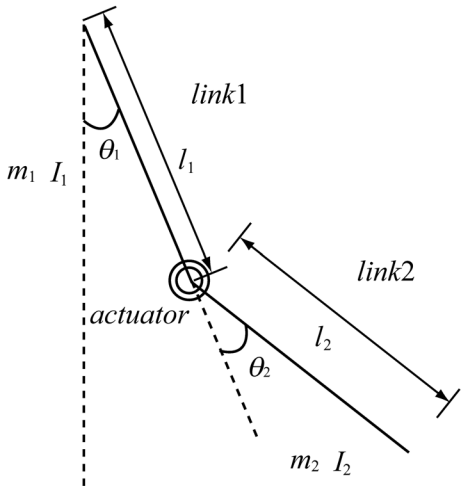


図 2 アクロボット
Fig. 2 The acrobot.

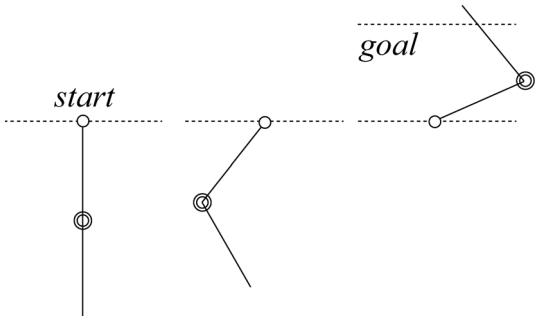


図 3 アクロボットの振り上げ課題
Fig. 3 The swing-up task of the acrobot.

学習の目的は，両リンクがつり下がった静止状態から出発し，リンク 2 の先端がなるべく早く一定の高さまで振り上がるよう，アクチュエータを制御することである（図 3）．ゴールへ到達するまでのアクロボットの動き（関節の時間変化）の例を図 4 に示す．

アクロボットの物理パラメータは Sutton の実験例 [1] に合わせ，$m_1 = m_2 = 1\,[\mathrm{kg}]$，$l_1 = l_2 = 1\,[\mathrm{m}]$，$I_1 = I_2 = 1\,[\mathrm{kg \cdot m^2}]$ とした．ここで，m_1，m_2 は各リンクの質量，l_1，l_2 は長さ，I_1，I_2 は慣性モーメントである．

アクロボットの状態は，各関節の角度及び角速度 θ_1，θ_2，$\dot{\theta_1}$，$\dot{\theta_2}$ によって完全に規定される．

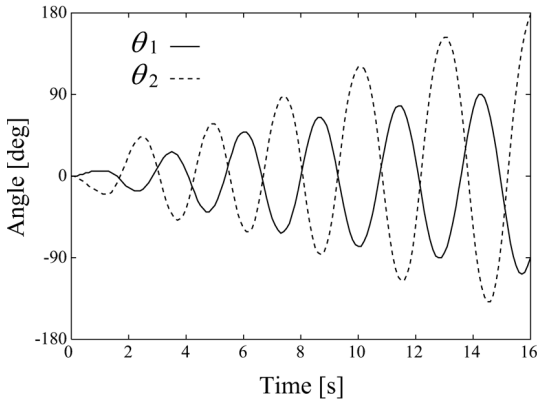


図 4　関節角度の時間変化の例
Fig. 4　Example of the transition of joint-angle values.

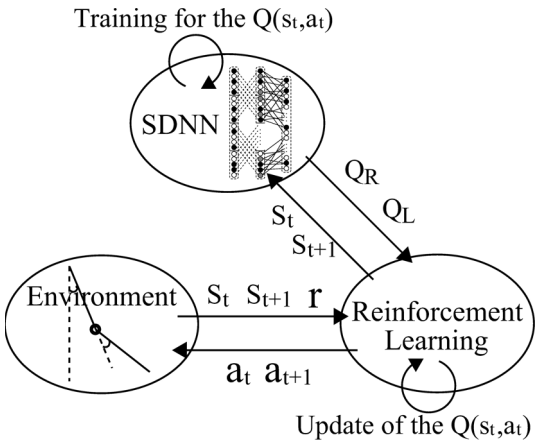


図 5　制御と学習の全体像
Fig. 5　The architecture of control and learning.

$\theta_1, \theta_2 \in [-\pi, \pi]$ [rad]，$\dot{\theta_1} \in [-3\pi, 3\pi]$ [rad/s]，$\dot{\theta_2} \in [-5\pi, 5\pi]$ [rad/s] を状態空間に対する機械的な制約条件とし，その範囲で $[-1, 1]$ に正規化したものをそれぞれ変数 x^1，x^2，x^3，x^4 とした．

選択できる行動は R または L の 2 種類（関節 2 に与えるトルク τ が -1 または 1 [Nm]），制御の時間刻みは 0.1 [s] とした．また，報酬としてリンク 2 の先端がリンク 1 の支点より 1 [m] 上まで振り上がったとき $+10$ を与えるほか，リンク 1 の振れ幅 $|\theta_1|$ が 1 度未満の場合に -5 を与え，それ以外の報酬は一切与えない．なお，ゴールに到達して正の報酬を得るか，ゴールに達しないまま 50 秒が経過した時点で 1 回の試行（エピソード）が終了するものとした．

3.2 学習方法

学習は，最も基本的かつ代表的な強化学習アルゴリズムの一つである Q-learning [14] を用いて行った．これは，ある状態 s における行動 a の評価値（Q 値）を示す行動価値関数 $Q(s, a)$ を逐次的に求めるというものである．具体的には，時刻 t において状態 s_t のエージェントが行動 a_t を行った結果，報酬 r を得て状態 s_{t+1} に遷移したとき，次式に従って関数値を更新する．

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})] \quad (5)$$

ここで，α は学習率，γ は割引率と呼ばれるパラメータであり，実験では $\alpha = 0.5$，$\gamma = 0.9$ とした．

行動選択法としては ε-greedy 法 [1] を用いた．これは，ε の確率でランダムに行動選択を行い，そうでなければより Q 値が高い行動を優先する手法である．アクロボットの振り上げ課題では，より早いタスク成功には正しい行動を連続して選択する必要があるため，環境の積極的な探索は学習初期にとどめることとする．そのため，ε は徐々に減少し，一定試行後は常に Q 値が高い行動を優先するように設定した．具体的には，$\varepsilon = 0.1$ から始め，エピソードが終了するたびに 0.01 ずつ減少させ，10 エピソード終了時に $\varepsilon = 0$ となるように設定した．

本研究では，実ロボットにおける制御を想定して，オンライン学習を行うものとする．すなわち，環境を観測し，実際に行動して $Q(s_t, a_t)$ の値を更新するという 1 ステップごとに，関数近似器の出力と更新後の Q 値との誤差が小さくなるよう近似器の修正を行う．修正後の近似器は，次のステップにおいて直ちに Q 値の推定に利用する．

制御と学習の流れを図 5 にまとめた．なお，図中の Q_R，Q_L は，右行動及び左行動に関する関数近似器の出力を表す．

3.3 実験条件

関数近似器として，前章で述べた SDNN のほか，局所的近似手法に基づく代表的な近似器である RBFN を用いる．これは，放射状の基底関数の線形結合によって関数を近似するもので，ここでは m 次元の状態空間中に等間隔に 11^m 個の格子点をとり，各点を中心とする $\sigma = 0.2$ のガウス関数を基底関数として配置した．また，SDNN のパラメータは $n = 200$，$n' = 600$，$c = 0.05$ とし，出力素子の出力の合計 $y = \sum_i y_i$ か

ら Q 値への変換には $Q = (y - 300)/30$ という式を用いた．これらは，何度か予備実験を行って決定したものである．

なお，学習開始前の初期状態において $Q(s, a) \equiv 0$，すなわち全状態について Q 値を 0 とするが，RBFN の場合，そうするには単に重み係数の初期値をすべて 0 に設定すればよい．しかし，SDNN の場合，中間層から出力層の結合荷重をすべて 0，出力素子のしきい値を微少値にすると，初期段階のわずかな学習によって多くの出力素子の出力値が同時に変化してしまうため，学習が非常に不安定になる．これを防ぐために，中間層と出力層を弱くランダムに結合した上で，初期状態ではどのような入力に対しても必ず $Q = 0$，すなわち $y = 300$ となるよう，300 個の出力素子についてはしきい値を十分大きな正の値に，残りの 300 個については十分小さな（絶対値が大きい）負の値に設定しておく．

実験は，関数近似器のみが異なる 2 種類のモデル（それぞれ SDNN モデル及び RBFN モデルと呼ぶ）について，まずアクロボットの制御に必要な四つの状態変数 x^1〜x^4 のみ与える条件で行う．次いで制御には不必要な冗長変数 x^5 及び x^6 を追加した場合について実験する．初期状態から 100 エピソードの学習を終えるまでを 1 実験試行とするが，試行ごとのばらつきの影響を抑えるため，各条件について乱数系列を変えながら 10 試行ずつ実験を行った．

4. 結果と考察

4.1 冗長変数を加えない場合

冗長な変数を含まない 4 次元状態空間において学習を行った結果を図 6 に示す．グラフの横軸はエピソード数，縦軸は各エピソードの継続時間，すなわちゴールに到達するまでの時間（到達しない場合は 50 秒）である．SDNN モデル（実線）及び RBFN モデル（破線）のそれぞれについて，10 試行の平均値がプロットされている．

学習の初期段階では，両モデルにほとんど差はないが，最終的な平均到達時間は SDNN モデルの方が短く，第 81〜100 エピソードの平均値で比較すると約 6.5 [s] の差があった．この差は統計的に有意であり（paired t-test，$p < 0.001$），SDNN モデルの方がより適切に価値関数を近似できることを示している．

実際，ある標準的な試行の終了時点（100 エピソードの学習後）における両近似器の出力には，図 7 に示すような違いがある．この図は，4 次元の状態空間における Q_R の値を色によって視覚的に表したもので，θ_1-θ_2 平面の一部（$0 \leq \theta_i \leq 180$ [deg]）を表す 9 枚のグラフで構成されている．これらのグラフは $\dot{\theta_1}$ 及び $\dot{\theta_2}$ の値が異なっており，中央はリンク 1，リンク 2 が静止している状態，右列と左列はリンク 1 が右向き及び左向きに振れている状態，上行と下行はリンク 2 が右向き及び左向きに振れている状態に対応している．

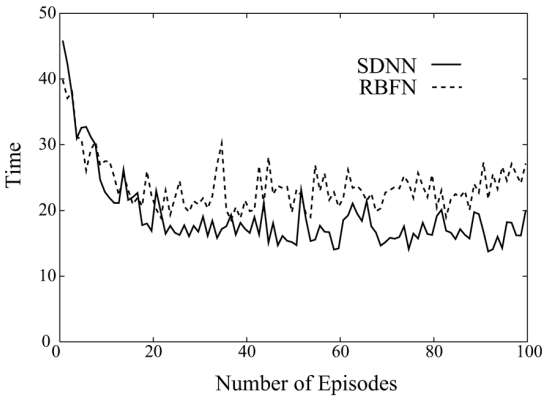
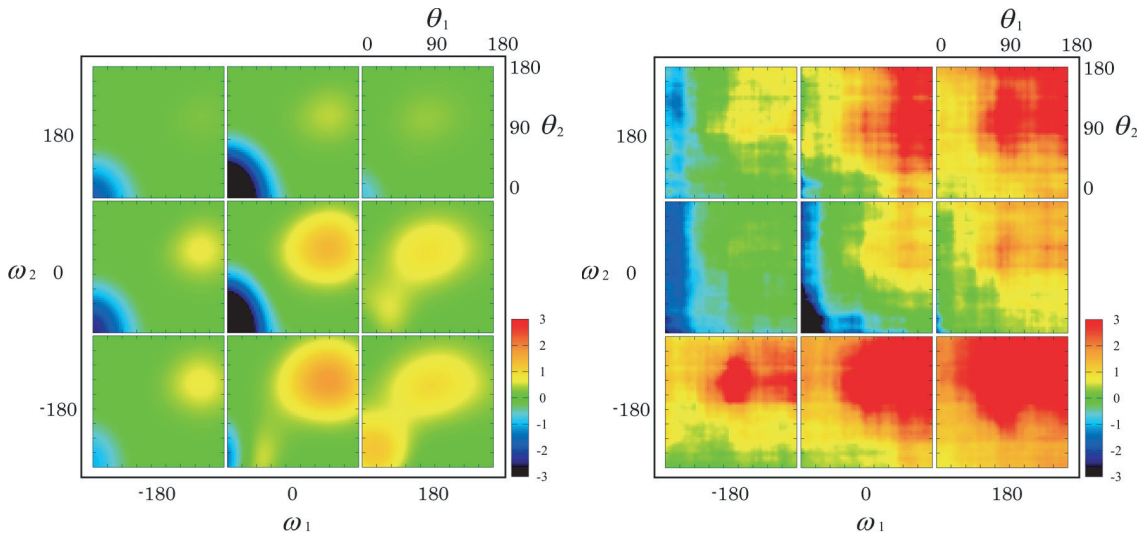


図 6　冗長変数を加えない場合の学習過程
Fig. 6　Learning curves in case of adding no redundant state variable.

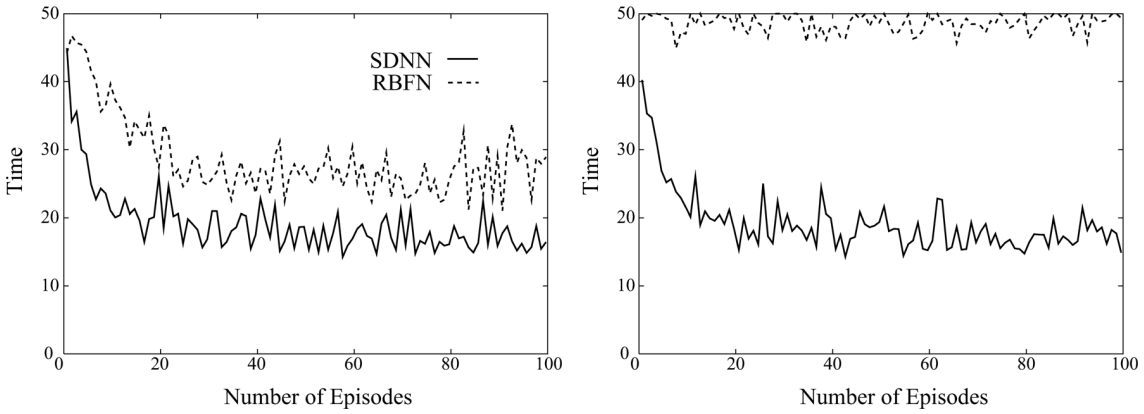
この図から分かるように，(a) の RBFN モデルでは Q 値がほぼ初期値 0 のままの領域が多く，値が大きな（グラフで赤い）領域はごく一部である．また，その広がり方も単純な同心円（4 次元状態空間では超球面）状であり，関数値は常になだらかにしか変化しない．一方，(b) の SDNN モデルでは，0 以外の値をとる領域が広く広がっており，広がり方も同心円的ではない．また，全体的には値がなだらかに変化するが，急激に変化する部分も存在する．

ただし，RBFN の関数近似能力は，基底関数の広がりや数に大きく依存することに注意が必要である．例えば，σ をより小さく（ガウス関数の幅を狭く）した上で基底関数の数を増やしたならば，より複雑な形状の関数を高い精度で近似することが可能である．しかし，そうすると計算コストが増加する上に，汎化が生じにくくなるため学習効率が低下してしまう．一方，SDNN の場合，学習効率を低下させずに十分な近似精度を確保することが可能である．本実験では，学習初期（第 20 エピソードまで）の性能が高くなるように RBFN のパラメータを設定したため，両モデルの差が学習効率よりも近似精度に現れたと考えられる．



(a) RBFN　(b) SDNN

図 7　100 エピソード学習後の価値関数（(a) RBFN 近似と (b) SDNN 近似）比較
Fig. 7　The comparison of Q-value functions (approximated by (a) RBFN and (b) SDNN) after learning 100-episodes.



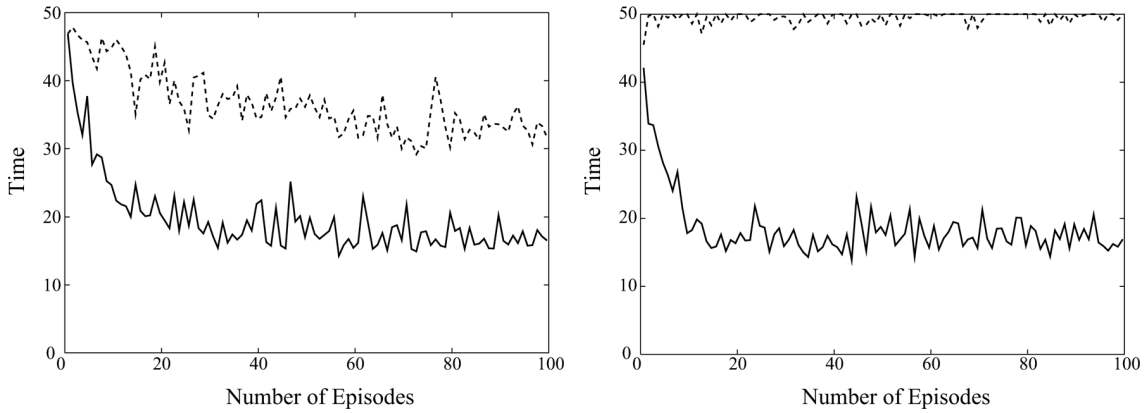
(a) $m = 5$　(b) $m = 6$

図 8　正弦波を冗長変数として加えた場合の学習過程
Fig. 8　Learning curves in case of adding redundant state variable(s): (a) $x^5 = \sin \omega t$, (b) $x^5 = \sin \omega t$ and $x^6 = \cos \omega t$ ($\omega = 0.1$).

4.2　冗長変数を加えた場合

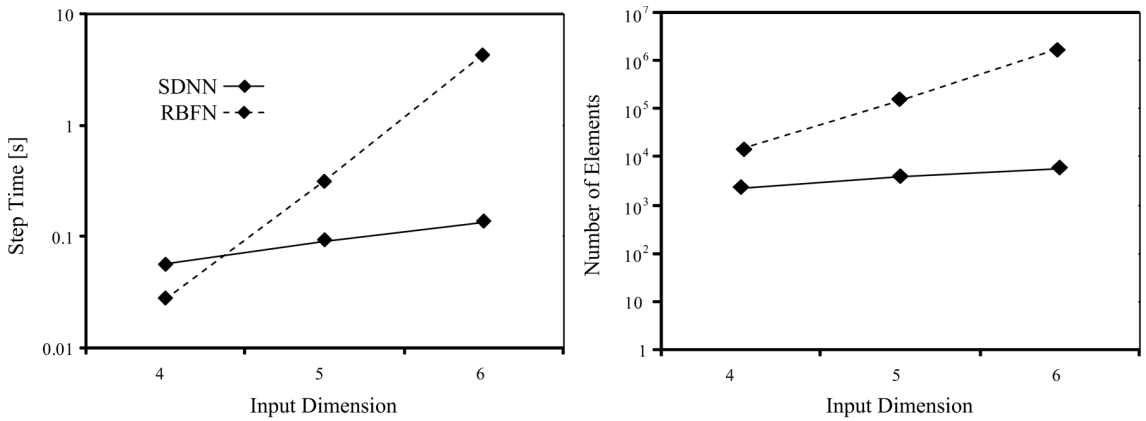
図 8 (a) は，冗長変数として $x^5 = \sin \omega t$（$\omega = 0.1$）を与えた場合である．図 6 と同様に，10 回の実験試行における各エピソードの平均継続時間をプロットした．RBFN モデル（破線）では学習があまり進まず，第 81〜100 エピソードの平均到達時間は冗長変数がない場合に比べて 4.0 [s] ほど長かった．これに対して SDNN モデル（実線）では，冗長変数がない場合との差は特に見られず，平均到達時間も同じであった．図 8 (b) は更に $x^6 = \cos \omega t$ を追加した場合であるが，RBFN ではエピソードを重ねても学習がほとんど進まないことが分かる．一方，SDNN モデルはほとんど影響を受けず，冗長変数がない場合との平均到達時間の差も統計的に有意ではなかった（paired t-test, $p = 0.58$）．

このような性質は，ω の値や関数を変えてもほとんど変わらなかった．また，極端な場合として，追加変数の値をステップごとにランダムに選び，エピソード



(a) $m = 5$ (b) $m = 6$

図 9 乱数を冗長変数として加えた場合の学習過程
Fig. 9 Learning curves in case of adding redundant state variable(s): (a) single random input, (b) double random inputs.



(a) CPU time (b) Number of elements

図 10 計算コストの比較
Fig. 10 The comparison of computational costs.

によっても全く変えるようにしたところ，より顕著に現れた（図 9）．このことから，本実験の結果は，追加した変数の種類によって生じる特異的現象ではなく，両モデルの一般的な性質を示していると考えられる．また，冗長変数に弱いという RBFN モデルの性質は，局所的関数近似手法を用いた場合一般に成り立つと考えられる．

4.3 考　　察

RBFN や多層パーセプトロンと比較したとき，SDNN にはいくつかの非常に優れた性質がある．本節では，これらの特性について考察する．

（1） 計算コスト

状態空間の次元の増加に伴う計算コストの増加を図 10 に示す．実線が SDNN モデル，破線は RBFN モデルを表している．(a) は 1 ステップに要する計算時間，(b) は必要な素子数（RBFN の場合は基底関数の数）を対数表示したものである．なお，コンピュータで計算する際に必要なメモリの量は，どちらのモデルも素子数にほぼ比例する．マシン環境は，Dell PrecisionT7400 クアッドコアインテル (R) Xeon(R) プロセッサ X5482（2×6 MB L2 キャッシュ，3.20 GHz, 1600 MHzFSB），8 GByte クアッドチャネル DDR2-SDRAM メモリである．

グラフから分かるように，RBFN は計算時間も必要なメモリも指数関数的に増大するため，6 次元程度が現実的な限界である．一方 SDNN の計算コストの増

加は m^2 のオーダでしか増えない．また，関数近似器の出力の計算や結合荷重の修正は完全に並列実行が可能であり，高性能の並列型計算機を用いれば，あるいは専用のハードウェアを開発すれば，かなりの高次元であっても実時間で計算できると考えられる．

（2） 部分的に不連続な関数の近似

ガウス関数は連続であるから，それを基底関数とする RBFN の出力も連続である．したがって，RBFN は不連続な関数を正確に近似することはできない．もちろん，効率的な近似ができるのは関数に連続性があるからであり，例えば至るところ不連続な関数を有限個のサンプルから正しく近似することはそもそも不可能である．しかし，現実の問題では，「ほとんどの領域で連続だが，部分的に不連続がある関数」がしばしば現れる．強化学習の価値関数はそのよい例であろう．

理想的な価値関数というのは，初期状態からゴールまでの経路に沿って価値が連続的に増加するような関数である．しかし，例えば一歩踏み外せば崖から転落する道を登る場合を考えれば分かるように，その経路から外れたときに連続的に価値が低下するとは限らない．むしろ，ゴールに到達可能か可能でないかの境界状態があり，そこを境に価値が不連続に変化する場合が多いと思われる．アクロボットの振り上げ課題の場合も，例えば $\dot{\theta_1}$ の角速度がある値以上ならばそのままゴールに到達できるが，そうでなければいったん逆側に振れなければゴールに到達できないという状態があり，そこでは価値関数が不連続的になっていると考えられる．

図 7 (b) からも読み取れるように，SDNN は，多数のしきい素子（2 値ニューロン）の出力パターンによって関数値を表現するため，部分的に不連続な関数を近似可能である．したがって，この点においても SDNN は価値関数の近似に適しているといえる．

（3） 冗長変数に対するロバスト性

SDNN モデルが冗長変数に対してロバストであることは前述のとおりであるが，その理由として以下の二つが考えられる．一つは，出力値に無関係な素子からの結合荷重は，そうでないものに比べて絶対値が小さくなるという，パーセプトロンと共通する性質である．もう一つは，同一のパターンが異なる複数の修飾パターンによって選択的不感化を受けたとき，もし出力パターンが同じならば非常に強い汎化が生じ，どんな修飾パターンに対してもほぼ同じ出力を出すようになるという，SDNN 特有の性質である．

後者の性質を説明するために，一つの変数が一つの冗長変数によって修飾される場合を考え，それぞれのコードパターンを S^μ 及び C^ν としよう．このとき，中間層のパターン $S^\mu(C^\nu)$ は，C^ν に関係なく，ある出力パターン T^μ に対応づけられることになる．今仮に，パターン C^1 と C^2 が完全に反対の関係（$C^1 = -C^2$）にあったとすると，一方で不感化される素子は他方で不感化されないから，$S^\mu(C^1)$ 及び $S^\mu(C^2)$ から T^μ への対応を学習した後の状態は，何の修飾も受けていないパターン S^μ から T^μ への対応を学習した状態と等価である．したがって，統計的ノイズを無視すれば，S^μ をどんなパターン C^ν で修飾しても T^μ が出力されるはずである．同様の議論は，無相関の修飾パターン $C^1, C^2, C^3, \ldots$ について学習した場合にも成り立つ．

現在詳しい解析を進めている最中であるが，恐らく上記の性質の両方が寄与していると思われる．また，より多くの冗長変数を追加した場合の SDNN モデルの挙動についても解析中であるが，現時点での結果と上記の性質から考えて，どれだけ追加したとしても学習効率はそれほど低下しないと思われる．

（4） 素子数と近似精度の関係

従来の代表的なニューラルネットである多層パーセプトロンの場合，中間層の素子数と学習回数を必要以上に増やすと，かえって汎化誤差が大きくなってしまうことがある．これは過剰適合と呼ばれる現象であり，これを避けるために素子数をどう決めるかが一つの問題となっている．一方，SDNN の中間素子数は，入力変数のコードパターンの次元 n によって自動的に決まる．また，n がいくら大きくても過剰適合が生じることはなく，むしろ大きいほど量子化誤差及び統計的ノイズ（無関係のコードパターン間にたまたま生じる相関など）が減って近似精度が高くなる．したがって，素子数は必要な近似精度と計算コストのみを考慮して決定すればよい．

（5） オンライン学習への適性

大域的な近似手法やニューラルネットでは，しばしば同じサンプルについて何度も繰り返し学習する必要がある．特に，多層パーセプトロンで一般に用いられる BP 法は，計算量も多く収束に非常に時間がかかる．これに比べて，SDNN の学習則は単純で計算量が少ない上に必要な繰返しも非常に少なく，本実験では 1 回のみである．しかも，多層パーセプトロンで大きな問題となるカタストロフィック干渉 [15]（新たなサンプルのみを追加学習すると，既学習の入出力関係全体が

壊れてしまう現象）が生じないため，過去の学習サンプルを別途メモリに保存しておく必要が全くない．これらは，強化学習をオンラインで行う場合の大きな利点となる．

5. むすび

選択的不感化ニューラルネット（SDNN）を用いて強化学習の価値関数近似器を構成し，アクロボットの振り上げ課題を用いてその有効性を検証した．その結果，SDNN による関数近似には，以下のような実空間における強化学習に適した特徴があることを示した．

- 近似精度と汎化能力が相反しないため，比較的少ないサンプルで精度良く近似することができる．
- 大部分の領域で連続な関数であれば，部分的に不連続であったとしてもうまく近似できる．
- 冗長変数による影響をほとんど受けない．
- 入力次元が増えても計算コストが爆発しない．
- 素子数を増やしても過剰適合が生じない．
- 計算が単純で，並列化も容易である．
- 繰返し学習が不要で，追加学習も容易であるため，オンラインでの使用に適している．

これらの特徴の多くは，無数の感覚器を通して環境から大量の情報を受け取りながら生活している動物にも備わっていると思われる．我々は，本研究を発展させることにより，実空間において動物のように適切に行動できる自律ロボットなどが実現される可能性があると考えている．また本研究の成果は，膨大な情報の中から必要な情報だけを抽出する情報処理技術の開発にもつながる可能性がある．

今後の課題として，SDNN モデルの学習性能などの解析ほか，本手法を他の様々な課題に適用することによって有効性の検証や手法の改良を進めることが挙げられる．また，状況によって冗長性（必要な変数）が変わるような場合への適用や，行動が連続値によって表される場合への拡張も重要な課題である．更に，関数近似は様々な問題で必要となる基礎的技術であり，強化学習以外に SDNN が有効な問題を探ることも今後の大きな課題である．

謝辞 本研究の一部は，科学研究費補助金特定領域研究「情報爆発 IT 基盤」（課題番号 21013007）の支援を受けて行われた．

文　献

[1] R.S. Sutton and A.G. Barto, Reinforcement Learning, MIT Press, 1998.
[2] J. Park and I.W. Sandberg, “Universal approximation using radial-basis-function networks,” Neural Comput., vol.3, no.2, pp.246–257, 1991.
[3] F. Rivest, Y. Bengio, and J. Kalaska, “Brain inspired reinforcement learning,” Neural Information Processing Systems, Vancouver, CA, 2004.
[4] R. Coulom, “High-accuracy value-function approximation with neural networks applied to the acrobot,” European Symposium on Artificial Neural Networks, pp.7–12, 2004.
[5] 柴田克成，岡部洋一，伊藤宏司，“ニューラルネットワークを用いた Direct-vision-based 強化学習—センサからモータまで,” 計測自動制御学会論文集，vol.37, no.2, pp.168–177, 2001.
[6] R.S. Sutton, “Generalization in reinforcement learning: Successful examples using sparse coarse coding,” Neural Information Processing Systems, vol.8, pp.1038–1044, 1996.
[7] R.S. Sutton, “Reinforcement learning FAQ,” http://www.cs.ualberta.ca/˜sutton/RL-FAQ.html, 2004.
[8] B. Irie and S. Miyake, “Capabilities of three-layered perceptrons,” Proc. IEEE International Conference on Neural Nerworks, vol.1, pp.641–648, 1988.
[9] 森田昌彦，村田和彦，諸上茂光，末光厚夫，“選択的不感化法を適用した層状ニューラルネットの情報統合能力,” 信学論（D-II），vol.J87-D-II, no.12, pp.2242–2252, Dec. 2004.
[10] 末光厚夫，諸上茂光，森田昌彦，“下側頭葉における文脈依存的連想の計算論的モデル,” 信学論（D-II），vol.J87-D-II, no.8, pp.1665–1677, Aug. 2004.
[11] 宮澤泰弘，末光厚夫，森田昌彦，“選択的不感化理論に基づく海馬ニューロン活動のモデル化,” 日本神経回路学会誌，vol.14, no.1, pp.3–12, 2007.
[12] G. Boone, “Efficient reinforcement learning: Model-based acrobot control,” International Conference on Robotics and Automation, vol.1, pp.229–234, 1997.
[13] M.W. Spong, “The swing up control problem for the acrobot,” IEEE Control Syst. Mag., vol.15, no.1, pp.49–55, 1995.
[14] C.J.C.H. Watkins and P. Dayan, “Q-learning,” Mach. Learn., vol.8, pp.279–292, 1992.
[15] M. McCloskey and N. Cohen, “Catastrophic interference in connectionistnetworks: The sequential learning problem,” in The Psychology of Learning and Motivation, ed. G.H. Bower, vol.24, pp.109–164, Academic Press, NY, 1989.

（平成 21 年 9 月 4 日受付，12 月 28 日再受付）

新保　智之

平 20 筑波大・工学システム学類卒．現在，同大大学院博士課程システム情報工学研究科在学中．神経回路モデルの研究に従事．

山根　　健　（学生員）

平 17 筑波大・工学システム学類卒．現在，同大大学院博士課程システム情報工学研究科在学中．神経回路モデルの研究に従事．

田中　文英

平 15 東工大大学院総合理工学研究科博士課程了．博士（工学）．ソニー（株）及びソニー・インテリジェンス・ダイナミクス研究所（株）リサーチャー，University of California, San Diego 客員研究員を経て，現在，筑波大大学院システム情報工学研究科准教授．JST さきがけ「情報環境と人」研究員．人間–ロボット間インタラクション，発達学習，社会的相互作用の研究に従事．平 13 人工知能学会全国大会優秀論文賞，平 17 IEEE 国際会議 RO-MAN Best Paper Award など．

森田　昌彦　（正員）

昭 61 東大・工・計数卒．平 3 同大大学院博士課程了．日本学術振興会特別研究員，東京大学工学部助手を経て，平 4 筑波大学電子・情報工学系講師．同大機能工学系助教授などを経て，平 19 より同大大学院システム情報工学研究科教授．脳の情報処理機構及び神経回路網による情報処理の研究に従事．平 5 日本神経回路学会研究費，平 6 同学会論文賞，平 11 日本心理学会研究奨励賞受賞．