リカレントニューラルネットを用いた強化学習による予測機能の創発

大分大学　○後藤健太　松本康生　柴田克成

Emergence of the prediction ability by reinforcement learning using a recurrent neural network

Kenta Goto, Yasuo Matsumoto, Katsunari Shibata, Oita University

Abstract: In order to develop a robot that behaves flexibly in the real world, it must be better that a human do not give much knowledge to the robot in advance, but it learns various necessary functions autonomously. Among them, learning of "prediction" is focused on in this paper. The authors pointed out that it is important to learn not only how some future information can be predicted, but also what information should be predicted. It is suggested that reinforcement learning with a recurrent neural network enables such learning. In a task in which an agent can get a reward when it catches a moving object that disappeared on its way, it was observed that the agent learned to detect the velocity of the object before it disappeared and to catch the object at the correct timing using the velocity information.

１．はじめに

近年、計算機技術や制御技術などの進歩に伴い、ロボットはさまざまな方面で活躍の場を広げている。これらのロボットの多くは、プログラムとして人間が与えた知識に従って行動している。しかし、工場などの限定された環境と違い、多様に変化する実環境では、予め与えたプログラムで適切に行動させることには限界がある。人間のように実環境でも適切に行動できるロボットを開発するためには、ロボットがさまざまな環境で自ら学習して、必要となる知識、機能を身につけていくことが有効であると考えられる。

われわれ生物はさまざまな機能を有し、それらを駆使して目的を達成していく。そのような能力の一つに「予測」という機能がある。これは、現在までの状態変化を元に未来の状態を推測する高次の機能である。人間が「予測」を行う際には、たとえば打球を追いかける時には、膨大な視覚センサ信号の中から、必要な情報を抽出して「ボールが落下する位置」を予測し、そこに先回りしているように見える。このとき、「ボールをキャッチする」ためには「ボールが落下する位置」を予測するべきである、ということを知ること自体が非常に知的な作業であると言える。

「予測」を学習させる研究はすでに数多く行われているが、通常、「何を予測するか」は設計者が予め与えるか、もしくは、次のステップのセンサ信号そのものを予測させており（[1]など）、「何を予測するか」まで含めて学習させた研究は筆者らが知る限りではない。

本研究では、「予測」が必要なタスクを、記憶やダイナミクスを扱うことができるリカレントニューラルネットを用いた強化学習[2]で学習すると、報酬を得るためには何を予測すべきかというところまで含めて「予測」の機能が自律的に獲得されることを主張するとともに、シミュレーションで実際に「予測」機能が創発することを示す。

２．学習方法

本研究では、リカレントニューラルネットを強化学習のアルゴリズムで学習させることにより、自律的に予測機能が獲得されることを確認する。学習には、強化学習のアルゴリズムとして Q-learning[2]を用い、リカレントネットとして中間層ニューロンの出力を次の時刻の入力に付加する Elman 型のリカレントネットを用いた。Q-learning のアルゴリズムに基づいて、リカレントネットの教師信号を求め、教師あり学習を行うことで、強化学習によってリカレントネットを学習させた。

まず、現在の状態をニューラルネットに入力する。リカレントネットを用いることで、過去の状態を保持しなくても、学習を通して、必要な情報を抽出して中間層に保持するようになることが期待される。出力については、離散行動の数と同じ数の出力層ニューロンを用意し、出力値を各行動の行動価値である Q 値として扱う。

エージェントは、ネットワークの計算で得られた Q 値の各々に微小な乱数を付加し、その後 ε-greedy[2]で行動選択をするという二段階の確率的な行動選択を行った。これによって Q 値の高い行動が優先されると共に、Q 値の低い行動もある程度もある程度確率的に選択される。この二つの乱数要素は学習回数と共に小さくしていき、学習終了時には、ほぼ 0 となるよう設定した。学習時は、選択した行動 a_t に該当する出力のみに教師信号を与えて学習させる。現在の状態 s_t における行動 a_t の教師信号 $T_{a_t,t}$ は、その行動をした後の状態における状態 s_{t+1} をニューラルネットに入力したときに最大となる Q 値を使って次式のように自動作成する。

$$T_{a_t,t} = r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') \quad (1)$$

r_t は報酬を、γ は割引率を表す。γ の値の範囲は($0 \leqq \gamma < 1$)で与えられる。（1）式により求められた教師信号を用い、BPTT 法[3]に基づき時間をさかのぼって教師あり学習を行う。

３．タスク設定

予測なしでは目的の達成が困難なタスクとして、毎回軌道がランダムに決められ、かつ途中で見えなくなることがある物体を捕獲するタスクを採用した。エージェントは物体の壁への跳ね返りを考慮しながら、物体の的確な時間と位置の両方を予測し、捕獲を行うことで報酬を得られる。

このタスクはFig.1のような7.5×3.0の大きさのフィールドで行われる。移動物体の初期位置は (0.0,1.5) で固定とし、速度は0.40/stepから0.70/stepの間で、角度はx軸方向から-45°から+45°の間で毎回ランダムに設定する。移動を開始してからの物体は等速で移動するが、壁にぶつかると、入射角と反射角は変わらず、現在の速度のみ0.9倍の速度に減速する。

エージェントは常にx=6.0の位置におり、y方向の初期位置のみy=0.25からy=2.75の範囲で毎回ランダムで決められる。これによって、最適な行動をとれば、どのような条件でも報酬を得られる可能性があるように設定した。また、エージェントはy方向のみに移動できる。そして毎ステップ「捕獲」「上移動」「下移動」「待機」の4つのうちのどれかの行動をとる。「上移動」「下移動」を選んだ場合は、エージェントは、それぞれy方向に+0.20、-0.20移動する。「待機」の行動をとった場合、エージェントはそのステップでは、捕獲も移動も行わないものとする。

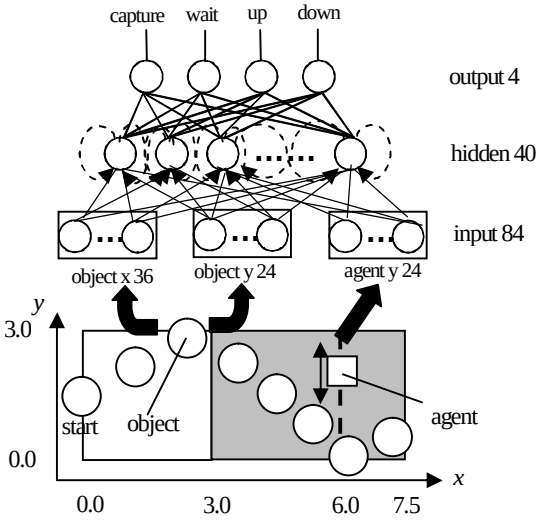

Fig.1 Object capture task and the neural network

移動物体とエージェントの相対距離dが1.0未満のときに「捕獲」行動を選択すると報酬が与えられて、その試行は終了する。報酬rの値は

$$r = 0.45 \times \left(1.0 - \frac{d}{1.0}\right) + 0.45 \qquad (2)$$

と設定されており、rの大きさは相対距離dの大きさに応じて変化し、dが0のとき最大値0.9、dが1のとき最小値0.45となる。また、dが1.0を越えた状態で「捕獲」行動をとった場合と、フィールドを越えても「捕獲」行動を取れなかった場合は、その試行は失敗として強制的に打ち切る。特に前者の場合は、罰として、（1）式にr_{t+1}=−0.1を代入した上で、「捕獲」のQ値の更新を行う。

エージェントには移動物体のx,y座標、自身のy座標の値を、それぞれ局所化して入力する。つまり個々のニューロンは、その信号が特定の値の周辺にあるときのみ発火し、複数のニューロンで一つの信号を表現する。そのため、一つのニューロンで連続値を表現するよりも、入力に対して出力を急激に変化させることが容易となる。具体的には移動物体のx座標の0.0から7.5までの値を36個のニューロンを使って表現している。そしてFig.2のように、そのうち常に4個のニューロンが発火し、他のニューロンの入力値は0となる。移動物体のy座標、エージェントのy座標も同様の方法で、それぞれ24個のニューロンで表現し、計84個のニューロンを通して状態が入力される。

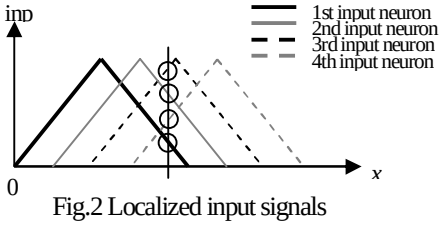

Fig.2 Localized input signals

移動物体のx座標が3.0を超える領域で、不可視領域を設けた。物体が不可視領域内に入ると、その物体はエージェントには見えないものとして、その際はネットワークへの入力を全て0とした。この領域は始点と終点をx座標の3.0から7.5の範囲で毎回ランダムに設定される。また、エージェントが予め不可視領域を知る術はない。そのため、エージェントは、移動物体が不可視領域に入るより前に、それまでの情報から移動物体の到着の位置とタイミングを予測しなければ、大きな報酬を得られるような行動をとることができない。

ネットワークの構造は3層のElman型リカレントネットで、中間層の数は40個とし、ニューロンの出力は-0.5から+0.5の範囲のシグモイド関数を用い、その値に+0.5を加えてQ値とした。また、フィードバック部の初期重みは、自身への重みを4.0、他への重みを0.0とした。これは、BPTT時に効率よく過去への誤差伝播を行うためである。

４．学習結果と考察

Fig.3に学習曲線を示す。横軸は試行回数で、縦軸は平均取得報酬を表した。エージェントは学習を重ねることで、より的確な捕獲行動をとれるようになったことが分かる。また、式（2）より、報酬の最大値は0.90となるが、移動物体、エージェント共に離散のステップでシミュレーションを行っているため、エージェントが理想的に行動したと

しても、Table1 で示したように両者の平均相対距離は 0.13 となる。またエージェントが理想的に行動した場合と比較した学習後のパフォーマンスを合わせて Table1 に示す。

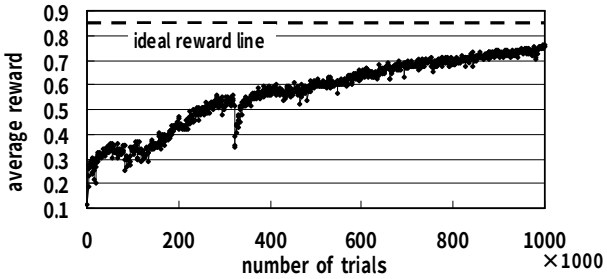

Fig.3 Learning curve

Table1 The agent's peformance after learning

	不可視領域の範囲			理想値
	ランダム	なし	最大	
平均報酬	0.760	0.770	0.760	0.841
報酬取得率[%]	99.1	99.5	99.1	100.0
エージェントと物体の平均相対距離	0.30	0.28	0.30	0.13

学習後のエージェントの行動の一例として、不可視領域が最大となる x=3.0 から x=7.5、速度 0.40/step、角度-30°だった場合を Fig.4 に示す。エージェントは常に x 座標を 6.0 固定で、y 方向のみで移動を行うが、Fig.4 では移動の様子を見やすくするよう移動物体の x 方向の位置に対してエージェントの y 座標をプロットした。

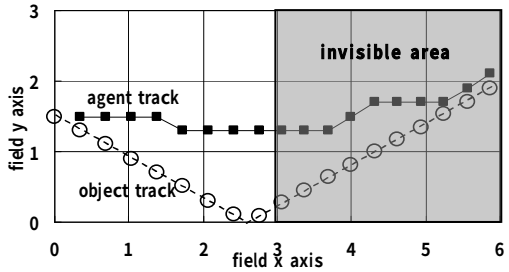

Fig.4 The behaviore of the object and agent (angle -30° 0.40/step)

また、この試行中の Q 値の変化を Fig.5 に示す。Fig.5 より、捕獲以外の行動の Q 値は、似たような変化をし、ゴールに近づくにしたがって徐々に上昇しているが、15step 目 (x=4.5 付近) から捕獲行動の Q 値が急激に上昇し、18step 目に捕獲している。これによって報酬領域に入る前には、お手つきをせず、正しいタイミングで捕獲を行うことができると考えられる。そこで Fig.4 と同じ角度で、速度を変化させ報酬取得までのステップを変えて検証したところ、後半の step では物体が見えていないにも関わらず、捕獲の Q 値は Fig.6 に示したように、報酬領域直前で急激に上昇していることが分かった。このことは速度や軌道を変化させても同様であった。

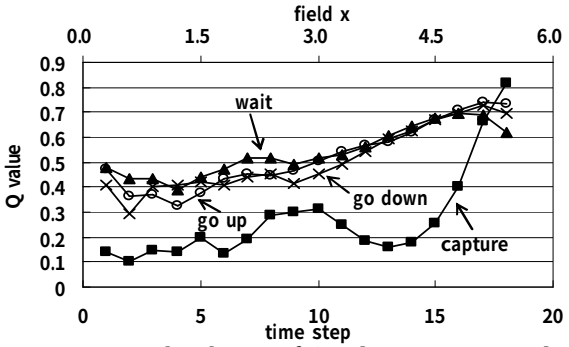

Fig.5 The change of Q value's in one episode

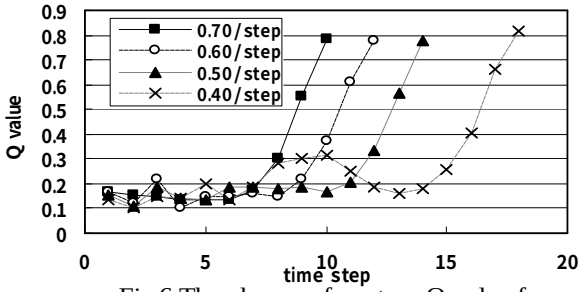

Fig.6 The change of capture Q value for each of 4 velocity conditions

これらのことから、単に強化学習を行うだけで、捕獲のタイミングと位置という必要な情報の予測を行うことができたといえる。

次に物体が見えている間に、どのように捕獲のタイミングを予測し、それを最終的な捕獲の行動に繋げているのかを調べた。多くのニューロンが複雑に影響しながら、予測とそれを利用した捕獲を実現しているようであったが、その中で分かりやすいものについて順を追って説明する。中間層ニューロンの値の変化を示した以下のグラフは、Fig.6 の 4 つのパターンに対応したものである。

まず最初に、Fig.7 に示す 30 番目の中間層ニューロンは移動物体が見えなくなる直前に出力が下がっており、その下がり方が報酬取得までのステップ数を表現しているように見えた。このニューロンの入力層からの結合の重みを観察したところ、Fig.8 に示すように、移動物体の x 座標を表している入力ニューロンのうち、移動物体が見えなくなる前の x=1.75 から x=2.75 の区間に発火の中心があるニューロンから、大きな負の結合をもっていることが分かった。

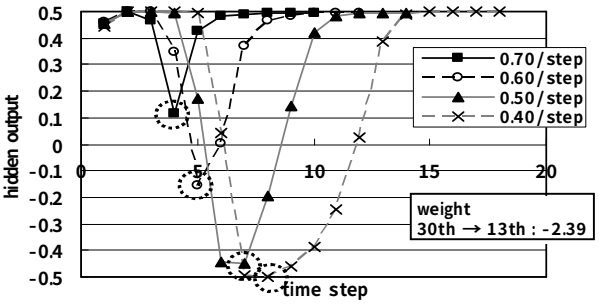

Fig.7 The change of the 30th hidden neuron's output
(The circles indicate the timing just before the object disappeared)

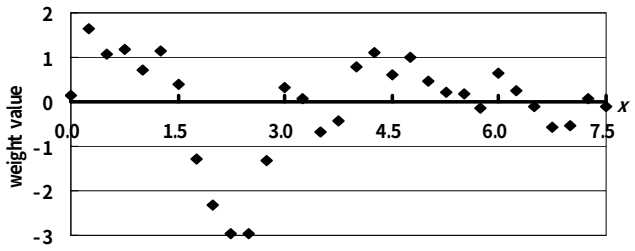

Fig.8 The connection weights from the input neurons that represent x location of the object

(Horizontal axis indicates the x coordinate when each neuron takes the minimum value)

捕獲のタイミングは、移動物体の x 方向の速度と、壁への衝突によって変化するため、移動物体の x 軸方向への速度が遅く、報酬取得まで多くのステップを費やすほど、この区間に留まる時間が長くなり、このニューロンは大きく出力を抑えられる。つまり、このニューロンは不可視領域に入る直前に物体の x 方向の速度を検出することを学習し、捕獲のタイミングの予測に寄与していると考えられる。ここで、このニューロンの不可視領域の入る x=3.0 の直前の出力と、報酬取得までのステップ数に一対一の対応があるかどうかを調べたところ、Fig.9 のように、両者の間には相関はあるものの、一対一の対応は見られなかった。このニューロンは捕獲のタイミングだけではなく他の情報も表現しているためと考えられる。

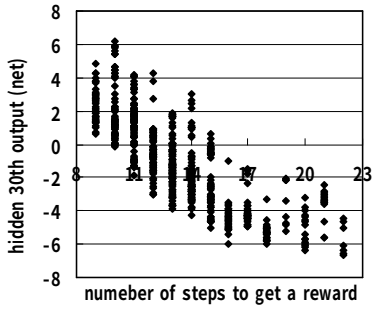

Fig.9 The correlation between the number of steps and the 30th hidden output before the object disappeared

この 30 番目のニューロンは、Fig.10 に示した 13 番目のニューロンに対し大きな負の結合をもっている。そのため、30 番目のニューロンの出力が大きくなると 13 番目のニューロンの出力が少し遅れて落ちてくる。さらに、13 番目のニューロンが、このニューロンと負の結合をもった Fig.11 に示した 38 番目のニューロンをさらに少し遅れて発火させる。38 番目のニューロンは、Fig.6 に示した捕獲行動の Q 値を表すニューロンと大きな正の結合を持っている。そのため、このニューロンの発火が Fig.5 のような捕獲の Q 値の急上昇を引き起こしていると考えられる。

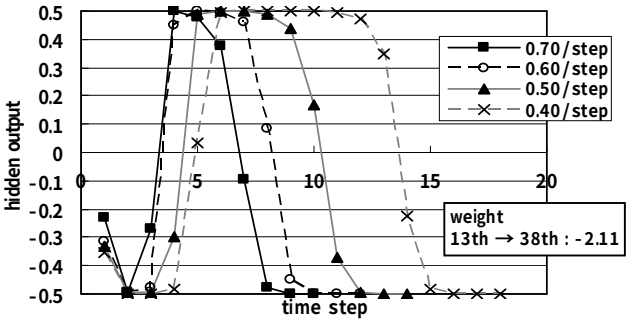

Fig.10 The change of the 13th hidden neuron's output

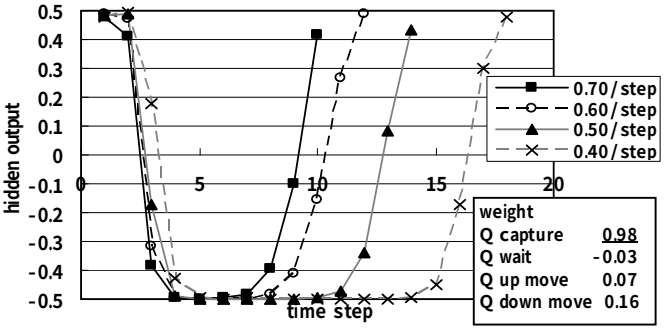

Fig.11 The change of the 38th hidden neuron's output

５．まとめ

本論文では、リカレントネットを用いて強化学習を行うことで、タスク達成のために何を予測すべきか、そしてそれを予測するためにはどうすればよいかという部分を含めた自律的な予測機能が創発することを主張した。

そして、予測なしでは達成が困難であると考えられるタスクの学習が可能であることにより、予測機能が創発したことを示した。また、リカレントネットの働きで、予測に必要な情報を抽出し、さらに複数のニューロンを介して最終的な行動に結びつけ、タスクを達成していることが分かった。

謝辞

本研究は、日本学術振興会科学研究費補助金#19300070 によって補助された。ここに謝意を示す。

参考文献

[1] Tani Jun, Stefano Nolif, “Learning to Perceive the World as Articulated” An Approach for Hierarchical Learning in Sensory-Motor Systems, Neural Networks, vol.12, pp1131-1141(1999)

[2] Watkins C.J.C.H and Dayan P, “Q-earning”, Machine Learning, Vol.8, pp.279-292, 1992.

[3] Rumelhart, D.E., Hinton, G.E., and, Williams, R.J., “Learning Internal Representations by Error Propagation”. Parallel Distributed Processing, The MIT Press, pp. 318-362(1986)