短期記憶のためのリカレントネット簡易学習則の基本構想

Simple Learning Algorithm for Recurrent Networks to Realize Short-Term Memories

柴田 克成(PY)*、岡部 洋一**、伊藤 宏司*　*：東京工業大学大学院 総合理工学研究科 知能システム科学専攻
Email：shibata@ito.dis.titech.ac.jp　　**：東京大学 先端科学技術研究センター
Katsunari SHIBATA(PY)* , Yoichi OKABE** and Koji ITO*　* : Tokyo Inst. of Technology　** : Univ. of Tokyo

Abstract-A simple learning algorithm for recurrent neural networks is proposed. This learning is similar to normal static back-propagation learning and needs only $O(n^2)$ memories and $O(n^2)$ calculations, but the target function of a recurrent neural network achieved by this learning is limited to a delayed recognition problem (short-term memory).

1．はじめに

我々生物がセンサから得る情報量は非常に大きいが、実世界の情報量はさらに大きく、その全てを見ることは不可能である。従って、現在のセンサ信号からだけでは状態を区別できず、過去の履歴から動作、認識を決定している場合が多いと考えられる。

環境に適応した柔軟な動作、認識の獲得に、ニューラルネットの学習機能が有効である。ニューラルネットで過去の記憶を扱うためには、リカレント構造にする必要がある。この学習が実現できれば、過去の膨大な情報の中から必要な情報のみを保持し、動作、認識に利用するといった効果が期待される。

ところが、現在存在するリカレントニューラルネットの学習則は、計算量、メモリ量といった観点から非現実的である。そこで、ここでは、ニューロンの内部状態として記憶を行う短期記憶のための現実的なレベルの学習則の基本構想とそれに基づく簡単な遅延認識課題のシミュレーション結果を示す。

2．従来の学習則

従来の代表的なリカレントネットの学習則に、BPTT (Back Propagation Through Time) と RTRL (Real Time Recurrent Learning) がある[1]。BPTT は、時間をさかのぼって誤差を逆伝搬させる必要がある上、さかのぼった時間より前の信号に対して学習できない。一方、RTRL では必要なメモリ量が $O(n^3)$、計算量が $O(n^4)$と非常に大きい[1]。そこで、ここでは、メモリ量は各結合上にメモリを持つ場合の $O(n^2)$以下、計算量はニューロン間のローカルなデータ転送だけで計算を行う場合の $O(n^2)$以下で、かつ時間をさかのぼらない学習則の構築を目指す。また、その代償とし、任意の連続値関数の近似を対象とせず、Fig. 1 のように、入力の演算結果を中間層ニューロンで保持し、ある時間経過後のトリガ信号の後に出力として反映させる遅延認識課題（短期記憶）を対象とする。

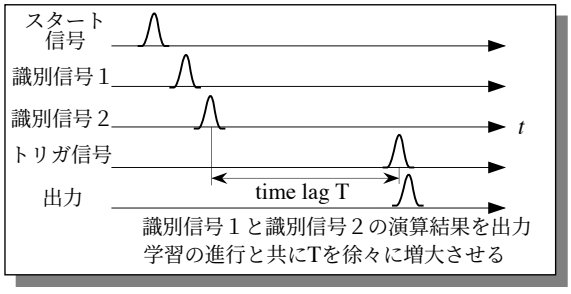

Fig. 1　A sample of delayed recognition problem

3．順方向計算

順方向計算は、通常の連続時間モデルとほぼ同じであるが、(ア) 内部状態 $u_j(t)$ を負にしない、(イ) 出力関数を値域が-1 から 1 のシグモイド関数（内部状態が 0 以上のため、出力も 0 以上）とした。つまり、

$$\tau_j \frac{d}{dt}u_j(t) = -u_j(t) + \sum_i w_{ji}\, x_i(t) + \theta_j(t) \tag{1}$$

$$x_j(t) = f\left(u_j(t)\right) = \frac{2.0}{1+\exp\left(-u_j(t)\right)} - 1.0 \tag{2}$$

ただし、x:出力値、u:内部状態、θ:バイアス入力
w:結合の重み値、τ:時定数、f:出力関数
とした。これには、以下の利点が考えられる。

(a) 出力 0 を容易に実現できるため、遅延認識課題の際に、通常時の教師信号を 0 とすることができる。
(b) 出力値が 0 の時に、誤差の吸収ができる。
(c) 自己フィードバックの重み値を制御することにより、2 値の安定平衡状態を容易に実現できる。
(d) 出力値が 0 の時に、出力関数の微分値が大きい値となるため、学習が容易である。
(c) 入力が 0 の時に出力が 0 なので、エネルギー消費の面から合理的である。
(f) 出力値が正なので、パルス密度として捉え易い。

また、このニューロンが重み値 w の自己フィードバック結合のみを持つ場合、w 及び w と θ の関係が

$$w > 2 \tag{3}$$

$$-\left\{\sqrt{w(w-2)} - \log\left(w-1+\sqrt{w(w-2)}\right)\right\} < \theta < 0 \tag{4}$$

を満たす時に 0 とそれ以外の安定平衡点を持つ。

4．学習則

学習は、通常の静的なＢＰ（バックプロパゲーション）法と同様に、誤差を逆伝搬させ、その伝搬した誤差を用いてニューロン間の結合を更新する方法をとった。ただし、ここでは、伝播誤差 δ と結合元のニューロンの過去の出力に関する情報を持った値との積によって重み値の更新をする。

まず、δ が伝播によって発散しないように、中間層→中間層、中間層→出力層の重み値 w を、変数 $\tilde{w}$ をシグモイド関数に入力して

$$w_{ji} = \frac{2W}{1+\exp\left(-\tilde{w}_{ji}/W\right)} - W \tag{5}$$

と計算し、w が $\pm W$ (ここでは、W=4.0 とした) の範囲に収まるようにし、変数 $\tilde{w}$ を学習によって更新した。入力層→中間層、入力層→出力層の重み値 w は、

キーワード：リカレントニューラルネットワーク、バックプロパゲーション、短期記憶、遅延認識課題

$w_{ji} = \tilde{w}_{ji}$ (6) とした。出力層の δ は、

$\delta_i = tr_j(t) - x_j(t)$ (7) tr: 教師信号 とし、δ の伝播は

$$\delta_i = \sum_j \frac{v_{ji}\,\delta_j(t)}{W} \quad (8)$$

$$\frac{d}{dt}v_{ji} = \left(w_{ji}(t)\,x'_j(t) - v_{ji}\right)\left|\frac{d}{dt}x_j(t)\right| \quad (9)$$

とした。ただし、$x'(t)$は$x(t)$が 0 の時には擬似的に

$$x'_j(t) = \frac{d\tilde{x}_j(\tilde{u}_j(t))}{d\tilde{u}_j(t)} = \frac{(1.0+\tilde{x}_j(t))\,(1.0-\tilde{x}_j(t))}{2.0} \quad if\ x(t) = 0.0$$

$$x'_j(t) = \frac{dx_j(u_j(t))}{du_j(t)} = \frac{(1.0+x_j(t))\,(1.0-x_j(t))}{2.0} \quad otherwise \quad (10)$$

$$\tilde{x}_j(t) = f\left(\tilde{u}_j(t)\right) = \frac{2.0}{1.0+\exp\left(-\tilde{u}_j(t)\right)} - 1.0 \quad (11)$$

$$\tilde{u}_j(t) = \sum_i w_{ji}\,x_i(t) \quad (12) \quad \text{とした。}$$

こうして伝播してきた δ に掛ける変数として、

(a) 現在および近い過去の出力値の情報

(b) 結合元ニューロンへの入力のうち、最近出力が変化したニューロンの情報

(c) 結合元ニューロンの出力値の変化に寄与したニューロンの情報

が必要であると考えられる。そこで、(a)(b)(c)に対応して、以下の p, q, r の 3 つの変数を導入する。

$$\delta_j \frac{d}{dt}p_{ji}(t) = -\,p_{ji}(t) + x_i(t)\,x'_j(t) \quad (13)$$

$$\frac{d}{dt}q_{ji}(t) = \left(x_i(t)\,x'_j(t) - q_{ji}(t)\right)\sum_i \left|\frac{d}{dt}x_i(t)\right| \quad (14)$$

$$\frac{d}{dt}r_{ji}(t) = \left(x_i(t)\,x'_j(t) - r_{ji}(t)\right)\left|\frac{d}{dt}x_j(t)\right| \quad (15)$$

さらに、結合元ニューロンが 0 より大きい時の q を保持した$\tilde{q}$ を以下のように計算し、

$$\frac{d\tilde{q}_{ji}(t)}{dt} = 0 \quad if\ x_i(t) = 0, \quad \tilde{q}_{ji}(t) = q_{ji}(t) \quad otherwise \quad (16)$$

p, q, rおよび$\tilde{q}$ を重み値の更新に用いた。

具体的には、試行錯誤により、各層間の重み値は、

(中→出) $d\tilde{w}_{ji}(t) = \left(p_{ji}(t) + q_{ji}(t) + r_{ji}(t)\right)\delta_j(t)\,\dfrac{dw_{ji}(t)}{d\tilde{w}_{ji}(t)}$ (17)

(入→出) $d\tilde{w}_{ji}(t) = \left(p_{ji}(t) + r_{ji}(t)\right)\delta_j(t)$ (18)

(中→中) $d\tilde{w}_{ji}(t) = \left(q_{ji}(t) + \tilde{q}_{ji}(t) + r_{ji}(t)\right)\delta_j(t)\,\dfrac{dw_{ji}(t)}{d\tilde{w}_{ji}(t)}$ (19)

(入→中) $d\tilde{w}_{ji}(t) = \left(q_{ji}(t) + \tilde{q}_{ji}(t) + r_{ji}(t)\right)\delta_j(t)$ (20)

と$d\tilde{w}$ を計算し、$\dfrac{d}{dt}\tilde{w}_{ji}(t) = \varepsilon_j\,d\tilde{w}_{ji}(t)$ (21) ε: 学習係数

より$\tilde{w}$ を、(5)式より重み値 w を更新する。また、バイアス θ も重み値 w と同様に学習するが、安定性より、中間層ニューロンのバイアスは最大値を-0.1 とした。出力層間の結合は設けなかった。

5．遅延認識課題のシミュレーション

上記の学習則を用いて、遅延認識課題のシミュレーションを行った。入力層 4 個、中間層 2 個、出力層 1 個それぞれニューロンを設け、入力層の 4 個のニューロンの役割をそれぞれ、スタート信号、トリガ信号、識別信号 1、識別信号 2 とし、サイクルのスタート時にまずスタート信号が入り、その後、識別信号 1、2 の順で信号が入力される。そして、ある時間経過後、トリガ入力が入って、出力を行う。

ここでは、識別信号 1 と 2 の NAND の出力を行うように学習した。学習後のニューラルネットの重み値を Fig. 2 に、各ニューロンの出力の様子を Fig. 3 に示す。中間層ニューロン 1 は、スタート信号の入力で立ち上がり、識別信号 1 に入力が入ると下がり、入力がないとしばらく値を保持していることがわかる。そして、中間層ニューロン 2 は、中間層ニューロン 1 が 0 でかつ識別信号 2 に信号が入ると立ち上がり、値を保持（記憶）し、トリガ信号の後に出力が立ち上がらないように抑制していることがわかる。

さらに、入力数を 1 個増やし、ランダムな信号を入れた場合にはそのニューロンとの結合が 0 に近くなった。また、中間層を 5 個に増やしても学習できた。同様に、AND, OR, NOR, NOT および 2 入力のシーケンスの学習ができることを確認したが、EXOR、EXNOR は学習することができなかった。

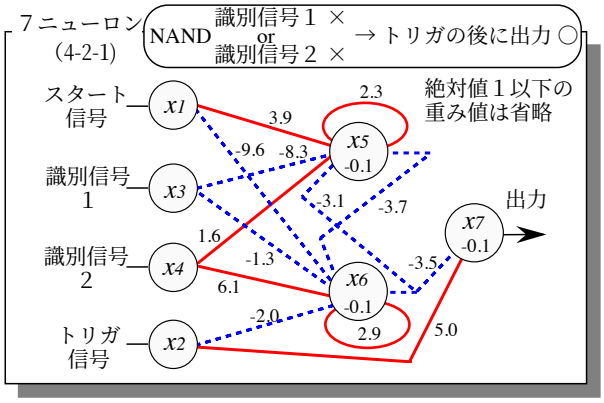

Fig. 2 Weight values after learning

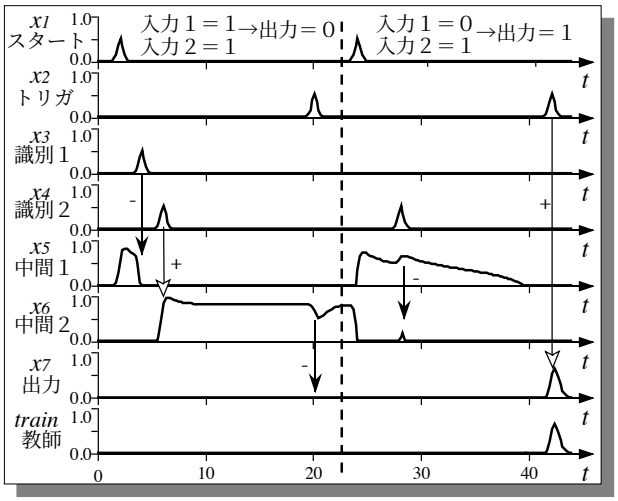

Fig. 3 Transition of each neuron's output after learning

6．結論

簡易リカレントネット学習則を提案し、NAND の遅延認識課題の学習をシミュレーションで確認した。

謝辞 本研究は日本学術振興会未来開拓学術研究推進プロジェクト「生物的適応システム」の一貫として行われた研究である。

参考文献

[1] R. J. Williams (麻生英樹訳), "Real-time recurrent learning algorithm",「脳と学習のメカニズム」, 丸善, pp.102-117 (1992)