

# リカレント型ニューラルネットの実用的な学習則である 離散時間PRL学習法

大分大学 Mohamad Faizal Samsudin 廣瀬武志 柴田克成

## Practical Recurrent Learning (PRL) in the Discrete Time Domain

Mohamad Faizal Samsudin, Hirose Takeshi and Katsunari Shibata, Oita University

Abstract: One of the authors has proposed a simple learning algorithm for recurrent neural networks, which requires computational cost and memory capacity in a practical order  $O(n^2)$ [1]. The algorithm was formulated in the continuous time domain, and it was shown that a sequential NAND problem was successfully learned by the algorithm. In this paper, the authors name the learning “Practical Recurrent Learning (PRL)”, and the learning algorithm is simplified and converted in the discrete time domain for simplicity. It is shown that sequential 3-bit parity problem as non linearly-separable problem can be learned by PRL even though the learning performance is often inferior to BPTT that is one of the most popular learning algorithms for recurrent neural networks. Furthermore, the output of each neuron is observed in the learning process and the character of PRL is shown.

### 1. まえがき

我々は日常生活において、あることがらを短期間だけ覚えておかなければならないことが多い。たとえば、繰り上がりのある暗算問題を解く場合などもその一つの例である。繰り上がった数字を覚えておかなければ正解に達することはできない。このように、人間は現在の情報のみを使って認識や行動しているのではなく、過去の情報を記憶しておき、それらを利用している場合が多い。ロボット等においても同様に、適切な行動を行わせるためには、現在の情報だけではなく過去の情報の考慮が重要であると考えられる。

一方、ニューラルネットで、過去の必要な情報を記憶させるためにはネット内にリカレント構造が必要となる。このリカレントネットワーク(RNN)を学習させるための学習アルゴリズムとして BPTT (Back Propagation Through Time)[2] や RTRL (Real Time Recurrent Learning)[3] といった学習則が提案されてきた。しかし、この二つの学習アルゴリズムは「計算量」と計算のために必要とされる「メモリ量」の点に、問題がある。

BPTT 学習法では過去のニューラルネットの状態を記憶し、時間をさかのぼって学習しなければならない上、さかのぼる時間が大きくなればなるほど、ニューラルネット数を  $n$ 、時間区間を  $T$  とした時に計算量のオーダーが  $O(Tn^2)$  やメモリ量のオーダーが  $O(nT)$  と多くなるという問題がある。一方、RTRL 学習法は時間をさかのぼる必要はなく、リアルタイムで学習すること可能であるが、計算に必要なメモリのオーダーが  $O(n^3)$ 、

計算量のオーダーが  $O(n^4)$  と非常に大きくなるので大規模なネットワークに対して実用的ではない。

これに対して、筆者らの一人は、時間をさかのぼって学習する必要がなく、かつメモリ量と計算量とも  $O(n^2)$  ですむような実用的リカレント学習 (Practical Recurrent Learning, PRL) 法というアルゴリズムを提案し、連続時間上で定式化した[1]。しかしながら、この方法では時間がずれて入力が入るシーケンシャル EXOR 問題を学習することができなかった。本論文では、解析を容易にするため、学習を簡素化するとともに、連続時間での PRL 学習法を離散時間での学習アルゴリズムに変換し、学習性能を観察したので報告する。

### 2. PRL 学習法

#### 2.1 リカレントニューラルネットの構造

本研究ではリカレントニューラルネットとして、一般的によく用いられる Elman ネットを用いる。Elman ネットは Fig.1 に示すような、階層構造ニューラルネットに中間層から中間層へのフィードバック結合を加えたものである。

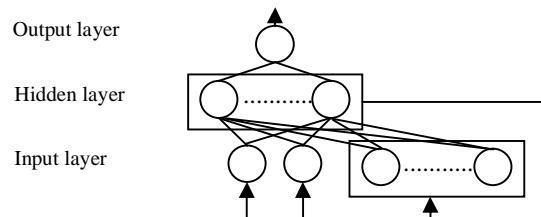


Fig.1 Elman-type recurrent neural network

## 2.2 連続時間の PRL 学習法[1]

始めに、[1]で提案された連続時間の実用的リカレント学習(PRL)法の学習アルゴリズムの要点を述べる。PRL 法でも通常の BP 法と同様に誤差信号をネットワーク内に逆伝播させる。過去にさかのぼらないで学習を行うために、各結合部に過去の入力信号の情報を保持する変数を設け、それを使って結合重み値の更新を行う。しかし、過去のすべての時間の情報を保持せず、以下の3つの情報を保持する。

- A) 現在および近い過去の入力信号の情報
- B) 結合先ニューロンへの入力の中で、最近変化した入力信号の情報
- C) 結合先ニューロンの出力値が最近変化した時の入力信号の情報

ここで、三つの情報に対応する変数  $p(t)$ 、 $q(t)$ 、 $r(t)$  を導入して、適切な初期重み値を与えた後、次式のように計算する。

$$\tau_j \frac{d}{dt} p_{ji}(t) = -p_{ji}(t) + x_i(t) f'(S_j(t)) \quad (1)$$

$$\frac{d}{dt} q_{ji}(t) = (x_i(t) f'(S_j(t)) - q_{ji}(t)) \sum_i \left| \frac{d}{dt} x_i(t) \right| \quad (2)$$

$$\frac{d}{dt} r_{ji}(t) = (x_i(t) f'(S_j(t)) - r_{ji}(t)) \left| \frac{d}{dt} x_j(t) \right| \quad (3)$$

ここで、 $x_i$ : 結合先ニューロンの入力、 $x_j$ : 結合先ニューロンの出力、 $\tau_j$ : 時定数、 $S_j$ : 結合先への総和入力、 $f(S_j)$ : シグモイド関数である。そして、たとえば三つの変数を用いて、重み値を

$$dw_{ji}(t) = \eta (p_{ji}(t) + q_{ji}(t) + r_{ji}(t)) \delta_j(t) \quad (4)$$

と更新する。ここで、 $\eta$  は学習係数である。

この中で、特に重要な変数とされている  $r_{ji}(t)$  は、Fig.2 の例のように、出力  $x_j(t)$  に変化がない限り過去の入力  $x_i(t)$  を保持し続けることができ、時間をさかのぼらずに計算を行うために重要な役割を果たすと考えられる。

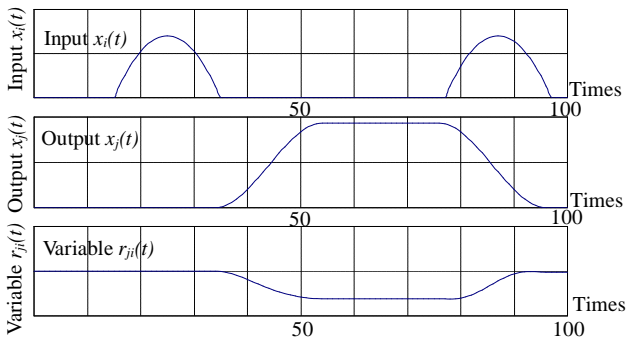


Fig.2 The example of the variable  $r_{ji}(t)$  transition.

## 2.3 離散時間の PRL 学習法

前に述べた連続時間の PRL 学習法を離散時間に変換したものを示す。また、文献[1]ではニューロンの出力関数は特殊なものを用いているが、簡単のため、本研究では出力関数は-0.5から0.5の値域をもつ通常のシグモイド関数とする。

学習の方法は BP 法と同様に誤差を逆伝播させ、その伝播した誤差を用いてニューロン間の結合を更新する。出力層の伝播誤差  $\delta_k^{(3)}$  を

$$\delta_k^{(3)} = \frac{\partial E(t)}{\partial S_k^{(3)}(t)} = \frac{\partial E(t)}{\partial x_k^{(3)}(t)} \cdot \frac{\partial x_k^{(3)}(t)}{\partial S_k^{(3)}(t)} = (Tr_k(t) - x_k^{(3)}(t)) f'(S_k^{(3)}(t)) \quad (5)$$

と計算する。ここで、 $E$ : 誤差信号、 $Tr$ : 教師信号である。重み値の更新は

$$\Delta w_{kj}^{(3)} = \eta \delta_k^{(3)} x_j^{(2)} \quad (6)$$

と通常の BP 法の場合と同様にする。ここで、

$w_{kj}^{(3)}$ :  $j$  番目の中間層ニューロンと  $k$  番目の出力層ニューロンとの重み値。

中間層ニューロンの伝播誤差  $\delta_j^{(2)}$  は以下の式にしたがって出力層ニューロンの  $\delta_k^{(3)}$  から計算する。

$$\delta_j^{(2)} = \left( \sum_k \delta_k^{(3)} \cdot w_{kj}^{(3)}(t) \right) \quad (7)$$

ここでは、通常の BP 法と違い、入出力の微分係数  $f'(x)$  を掛けていないが、これは変数  $r_{ji}(t)$  を式(8)で計算する際に、入力を取り込む時の  $f'(S)$  を含めて計算しているからである。そして、出力  $x_j(t)$  が変化した時の入力  $x_i(t)$  を  $r_{ji}(t)$  として保持するように、変数  $r_{ji}(t)$  を次のように計算する。

$$r_{ji}(t) = r_{ji}(t-1) \left( 1 - \left| \Delta x_j^{(2)}(t) \right| \right) + x_i^{(2)}(t) f'(S_j^{(2)}(t)) \left| \Delta x_j^{(2)}(t) \right| \quad (8)$$

ここで、 $\Delta x_j(t) = x_j(t) - x_j(t-1)$  である。これを用いて入力層-中間層の重み値の更新を、

$$\Delta w_{ji}(t)^{(2)} = \eta \delta_j^{(2)} r_{ji}(t) \quad (9)$$

と計算する。

## 3. シーケンシャル 3-bit パリティ問題のシミュレーション

### 3.1 タスクの設定

本論文では離散時間の PRL 法と BPTT 法を用いて、シーケンシャル 3-bit パリティ問題を学習させた。

3-bit パリティ問題は 3 つの 2 値入力の中に入力値 1

が偶数個あれば教師信号が-0.4 になり、奇数個あれば教師信号が 0.4 になる。このタスクは EXOR 問題と比較して入力数が多くなることで、多くの情報を保持しなければならないのでより難しいと考えられる。

入力パターンのシーケンスを Table 1 に示す。入力 1 は学習開始を判別するための信号で  $t=0$  の時に常に 1 が入力される。ランダムに選んだ 0 または 1 の値を  $t=5$  の時には入力 2 に、 $t=10$  の時には入力 3 に、 $t=15$  の時には入力 4 に入力する。それ以外の場合の入力は常に 0 である。また、入力 2 から 4 の中の 1 の個数より求めた教師信号を  $t=20$  の時に与えた。

Table 1 Timing of inputs and the training signal of one pattern.

Time, $t$	0	0~4	5	6~14	15	16~20	20
Input1	1	0	0	0	0	0	training signal is given
Input2	0	0	0,1	0	0	0	
Input3	0	0	0	0	0,1	0	

ニューラルネットの構造は、Fig.1 に示すように層数を 3 とし、入力層ニューロン数を 4、中間層ニューロン数を 20、出力層ニューロン数を 1 とした。ニューラルネット内の初期重み値に関しては、BPTT 法で伝播された誤差が発散したり、0 に近い値になってしまわないように、シグモイド関数の微分値の最大が 0.25 となることを考慮し、セルフフィードバックの重み値を 4.0、それ以外のフィードバックの重み値を 0.0 とした。また、入力層と中間層との重み値を 0 から 1 までの乱数とし、中間層と出力層との重み値を 0 とした。また、PRL 法では  $t=0$  の時、すべて変数  $r$  を 0 にリセットした。

### 3.2 シミュレーション結果

BPTT 法と PRL 法をそれぞれについて、シーケンシャル 3-bit パリティ問題を学習させた時のフィードバック部の学習係数を変化させた時の学習成功率と学習成功までの平均繰り返し回数を Table 2 に示す。100 回続けて二乗誤差が  $10^{-3}$  以下の値であることを、学習成功の条件とした。ただし、試行回数の制限は 20 万試行である。

Table 2 をみると、PRL 学習では時間をさかのぼって学習することはしないにもかかわらず、この問題をある程度学習ができることがわかる。連続時間では、EXOR すら学習することができなかった。それは連続時間のため教師信号の時間波形を与えており、ある特定の時刻のみに教師信号の値を与える今回の問題に比べ、学習が難しくなっていたことが原因ではないかと

Table 2 The comparison result of learning success rate and average number of iterations.

Learning rate	Learning rate for feedback connections	Learning success rate (/100times)		Average number of iterations	
		PRL	BPTT	PRL	BPTT
1	0.001	75	100	133003	17494
	0.003	81	100	119909	11393
	0.01	62	100	107270	7929
	0.03	26	96	119020	7710
	0.1	30	1	108248	6033

考えている。しかし、PRL 法は BPTT 学習法より、学習が非常に遅かった。そこで、その理由を調べるため、両者の学習の状況を観察した。

始めに、Fig.3 に PRL 法と BPTT 法の学習開始直後(約 1000 試行学習後)の出力層の出力を示す。ここで、比較するために、両者の重み値の初期値とパターン提示順序を PRL と BPTT の場合で同じにした。Fig.3 を見ると、両者の場合では入力 2 の値が 1 である時(図中の をつけたところなど)、出力層の出力の変化に大きな違いがあるという現象が見られた。

その出力の変化の違いを調べるため、パターン P5 の  $t=5$  の時の中間層から出力層への重み値と中間層ニューロンの出力をそれぞれ Fig.4, Fig.5 に示す。Fig.4 を見ると、PRL の場合では、hidden20 と出力層の出力との重み値が正であるが BPTT の場合では負になっており、PRL の場合は入力 2 が入る時に負に出力をしている hidden20 が抑制をかけていると考えられる。

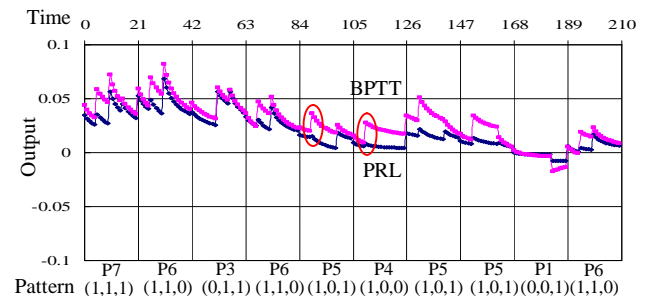


Fig.3 The transition of output in PRL and BPTT method.

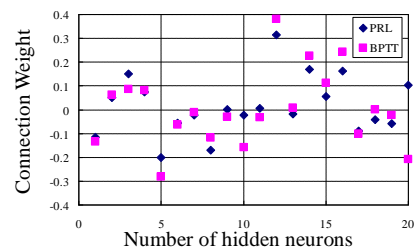


Fig.4 Connection weight from hidden neuron to output

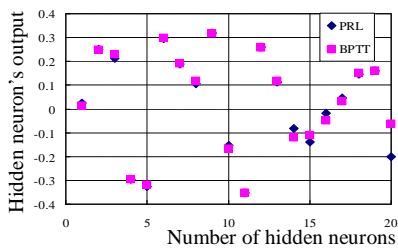


Fig.5 Output of hidden neurons when  $t=5$

次に、Fig.6 に input1,2,3,4 と hidden20 との重み値の時間変化を示す。PRL の場合では重み値の変化が非常に小さいことがわかった（特に input2 の場合）。

そこで、重み値の変化に影響を与える変数  $r$  の値の変化を観察した。すべての入力が入るパターン P7 での変数  $r$  の時間変化を Fig.7 に示す。Fig.7 をみると変数  $r$  の値は期待通り、各入力のところで値がちゃんと変化し、他の入力が入ると多少減少するもののその値を  $t=20$  まで保っていることがわかった。しかしながら、変数  $r$  の値が全体的に非常に小さく、特に入力 2 に対する  $r$  の値が小さいことがわかる。

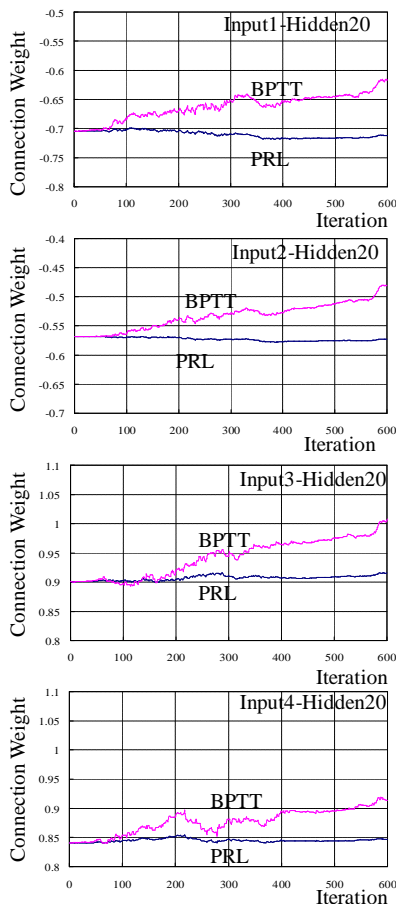


Fig.6 Connection weight from input 1,2,3,4 to hidden20.

その理由としては、 $t=0$  で変数  $r$  をリセットしていること、初期重み値があまり大きくなっていないこと、 $\Delta x_i$  が小さいことが考えられる。特に、 $r_{20,2}$  は他の入力に対する  $r$  と比較して小さいが、初期重み値が小さいため、入力 2 が入った時の hidden20 の  $\Delta x_i$  が小さいこと、さらに、 $t=20$  になるまでに他の入力が入ることで  $r$  の値が小さくなっていることが原因として考えられる。初期重み値が小さいと学習が進まないのは変数  $r$  の限界と考えられるので、今後さらに解析し、改善していきたい。

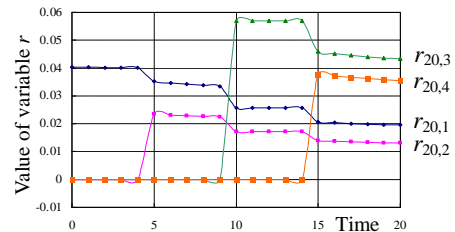


Fig.7 The value of variable  $r$  from input 1,2,3,4 to hidden 20 in the case of PRL.

#### 4. あとがき

PRL 学習法を離散時間に定式化することによって、PRL 学習法で 3-bit のパリティ問題をある程度学習できることを確認した。しかしながら、学習時間の面で PRL の学習性能は BPTT に大きく劣っていることがわかった。そして両者の重み値の時間変化に大きい違いがあることがわかった。PRL 法の性能を改善するために、より多くの実験と解析が必要である。

#### 謝辞

本研究は、日本学術振興会科技学術研究費補助金基盤研究#15300064 and #19300070 の補助を受けた。ここに謝意を表す。

#### 参考文献

- [1] Katsunari Shibata, Yoichi Okabe, Koji Ito, "Simple Learning Algorithm for Recurrent Networks to Realize Short-Term Memories", Proc. of IJCNN(Int'l Joint Conf. on Neural Network)'98, pp.2367-2372(1998).
- [2] Rumelhart D.E., Hinton G.E. and Williams R.J.: "Learning internal representations by errorpropagating", Parallel Distributed Processing, Vol. 1, MIT Press, pp.318-362 (1986)
- [3] Williams, R. J. and Zipser, D, "A learning algorithm for continually running fully recurrent neural network", Neural Computation, Vol.1, pp.270-280(1989)