

A Modified Recurrent Neural Network with Parametric Bias and its Application to Action Learning of a Humanoid Robot

Takashi Kuremoto^{a,*}, Keita Morisaki^a, Kunikazu Kobayashi^b, Shingo Mabu^a, Masanao Obayashi^a

^aYamaguchi University, Tokiwadai 2-16-1, Ube, 755-8611, Japan

^bAichi Prefectural University, Ibaragabasama 1522-3, Nagakute, 480-1198, Japan

*Corresponding Author: wu@yamaguchi-u.ac.jp

Abstract

Tani et al.'s recurrent neural network with parametric bias (RNNPB) is able to learning different time series patterns, and it has been successfully applied in action learning of robots. In this paper, we propose a novel type RNNPB using Elman type model instead of Jordan type in the conventional model. The proposed structure makes it easy to use error back-propagation (BP) learning algorithm which has lower computational cost than back-propagation time through (BPTT) method used in Tani et al.'s model. The effectiveness of the modified RNNPB was confirmed by its application to gesture learning experiment using a humanoid robot.

Keywords: RNNPB, dynamical attractor, humanoid robot.

1. Introduction

Artificial neural networks (ANNs) have been studied since 1950s and many models of them have been successfully applied to adaptive control, time series forecasting, pattern recognition, and many other fields. Among models of ANNs, recurrent neural networks (RNNs) are suitable to simulate dynamic systems or control unknown systems as "inverse models". Jordan type RNN⁽¹⁾⁽²⁾ and Elman type RNN⁽³⁾⁽⁴⁾ are the most well-known feed forward multi-layer models of RNNs. However, as supervised learning models, these RNNs are usually train to be identified models to certain systems. In other words, different dynamics need to be described by different connectionist models. To deal with a complex system containing multiple attractor dynamics, Tani et al. proposed a RNN with parametric bias (RNNPB)⁽⁵⁾⁻⁽⁸⁾, and the novel RNN has been widely applied to behavior learning of

robots recently⁽⁹⁾⁽¹⁰⁾⁽¹¹⁾.

In RNNPB, units named "parametric bias" (PB) are added in to a Jordan type RNN⁽¹⁾⁽²⁾ and they play a role of the bifurcation vectors of nonlinear dynamical systems. After the RNNPB is trained by different teacher signals using different PB values, the network is able to generate different time series patterns according to the PB values. The original RNNPB uses Jordan type RNN which has a connected "context layer". "back-propagation through time" (BPTT)⁽¹²⁾⁽¹³⁾, which is a learning algorithm usually used in the training of RNN, is used.

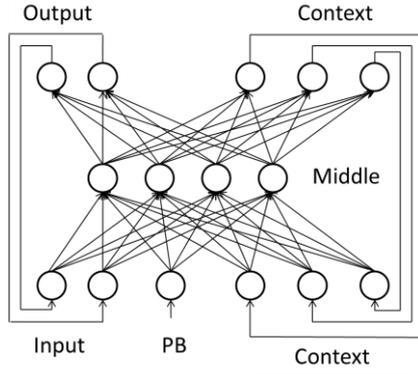
In this paper, we propose to modify RNNPB using Elman type RNN⁽³⁾⁽⁴⁾ instead of Jordan type used in the original RNNPB. The main merit of this modification is the simplicity of Elman model and the training method can be the well-known "error back-propagation" (BP) algorithm⁽¹²⁾, which is more simple than BPTT.

The modified RNNPB was applied to multiple actions learning of a humanoid robot PALRO (product of Fujisoft, Inc., 2010). Experiment results showed the learning performance of the proposed model was higher than the conventional RNNPB.

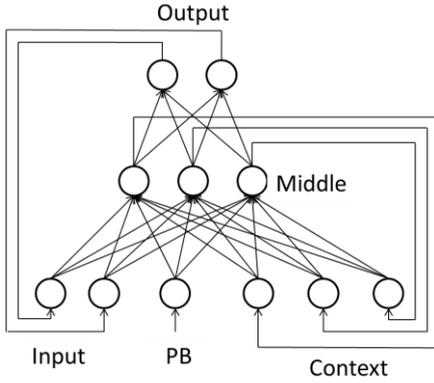
2. A Modified RNNPB

2.1 Structure

The structure of Tani et al.'s RNNPB and a modified RNNPB proposed here is shown in Fig. 1. The original RNNPB shown in Fig.1 (a) is a Jordan-type recurrent feed forward neural network with three kinds of internal layers: Middle layer, context layer and parametric bias (PB) layer. Units of Middle layer and Context layer output values by sigmoid function:



(a) A RNNPB proposed by Tani et al. ⁽⁵⁾⁻⁽⁸⁾.



(b) A modified RNNPB proposed here.

Fig. 1. The structure of RNNPBs.

$$f(\mathbf{z}) = \frac{1}{1 + e^{-\alpha \mathbf{z}}} . \quad (1)$$

Where α is a positive constant deciding the gradient of the function, and \mathbf{z} is the input vector.

Specially, the input vector \mathbf{z}_h for the Middle layer:

$$\mathbf{z}_h = \mathbf{u}_i \mathbf{v}_i^T + \mathbf{u}_{pb} \mathbf{v}_{pb}^T + \mathbf{u}_c \mathbf{v}_c^T . \quad (2)$$

Where $\mathbf{u}_i = \mathbf{x}(t)$, \mathbf{u}_{pb} , \mathbf{u}_c are the input to the Middle layer from the input vector, PB units, and units of Context layer respectively, \mathbf{v}_i , \mathbf{v}_{pb} , \mathbf{v}_c are the connection weights.

The input vectors \mathbf{z}_o and \mathbf{z}_c for Output layer and Context layer are given by:

$$\mathbf{x}(t+1) = \mathbf{z}_o = \mathbf{u}_h \mathbf{w}_o^T , \quad (3)$$

$$\mathbf{z}_c = \mathbf{u}_h \mathbf{w}_c^T . \quad (4)$$

Where $\mathbf{u}_h = f(\mathbf{z}_h)$ is the output of Middle layer given by

Eq. (1), and \mathbf{w}_o , \mathbf{w}_c are the connection weights between the Middle layer and the Output layer, the Context layers, respectively.

The learning rule, i.e., the modification to connection weights is given by the back-propagation through time (BPTT) ^{(12) (13)} algorithm. Let $\mathbf{y}(t)$ is the output of the network, $\mathbf{y}_d(t)$ is the teacher signal:

$$\Delta \mathbf{w}_o = (\mathbf{y}(t) - \mathbf{y}_d(t)) f'(\mathbf{z}_o) \mathbf{u}_h \quad (5)$$

$$\Delta \mathbf{w}_c = (\mathbf{y}(t) - \mathbf{y}_d(t)) f'(\mathbf{z}_c) \mathbf{u}_h \quad (6)$$

$$\Delta \mathbf{v}_i = \Delta \mathbf{w}_o \mathbf{w}_o^{old} f'(\mathbf{z}_h) \mathbf{u}_i + \Delta \mathbf{w}_c \mathbf{w}_c^{old} f'(\mathbf{z}_o) \mathbf{u}_i \quad (7)$$

$$\Delta \mathbf{v}_{pb} = \Delta \mathbf{w}_o \mathbf{w}_o^{old} f'(\mathbf{z}_h) \mathbf{u}_{pb} + \Delta \mathbf{w}_c \mathbf{w}_c^{old} f'(\mathbf{z}_o) \mathbf{u}_{pb} \quad (8)$$

$$\Delta \mathbf{v}_c = \Delta \mathbf{v}_i \mathbf{v}_i^{old} f'(\mathbf{y}(t-1)) \mathbf{z}_c(t-1) \quad (9)$$

For the units of PB layer, the internal state \mathbf{u}_{pb} changes according to the delta errors $\Delta \mathbf{v}_{pb}(t)$ during a period (a time series window) l ⁽⁵⁾⁻⁽⁸⁾:

$$\Delta \mathbf{u}_{pb}^t = \eta_1 [k_1 \sum_{t-1/2}^{t+1/2} \Delta \mathbf{v}_{pb}(t) + k_2 (\mathbf{u}_{pb}^{t+1} - 2\mathbf{u}_{pb}^t + \mathbf{u}_{pb}^{t-1})] + \eta_2 \Delta \mathbf{u}_{pb}^{t-1} . \quad (10)$$

Where η_1, η_2, k_1, k_2 are learning coefficient, learning rate, and internal coefficient of PB units.

In BPTT learning algorithm, it can be imagined that units of RNNPB at time t are connected to units of RNNPB at time $t-1$, and the same structure in the case of $t-1$ and $t-2$, ..., till the period l . So error back-propagation (BP) learning



Fig. 2. A humanoid robot PALRO was used in the experiment to verify the effectiveness of the proposed method.

rule⁽¹³⁾ is used repeatedly in BPTT.

In the proposed RNNPB, the output of Middle units become to the input of Context units without connection weights. So the learning rule for connections between Input layer with PB layer, context and Middle layer, and connections between Middle layer and Output layer can be given by standard back-propagation (BP) algorithm⁽¹³⁾ as



(a) Waving a hand (b) Raising two hands (c) Clapping hands
Fig. 3. Three actions to be learned by PALRO.

same as it is used in multi-layer perceptron (MLP).

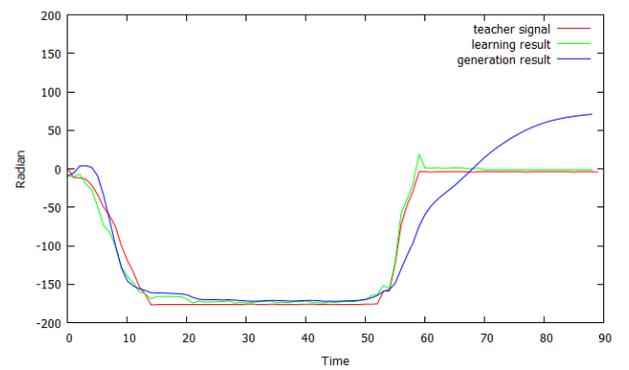
When BP learning algorithm is adopted into the proposed Elman type RNNPB, except Eq. (8) and Eq. (10) for the PB units, the modification of connections between Output layer and Middle layer, Middle layer and Input layer with PB and Context layer can be executed offline, i.e., batch learning. Additionally, the modification of connections between Middle layer and Context layer is not needed. So the computational cost of the modified RNNPB is lower than the original model.

To verify the learning performance of the modified RNNPB, experiment of multiple actions learning of a humanoid robot was executed and the results are reported in the next Section.

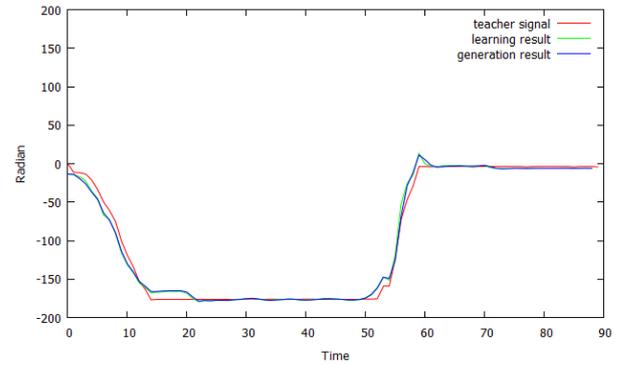
3. Experiments

PALRO, a humanoid robot (Fujisoft Inc., 2010 product) was used in the experiment (Fig. 2). Three kinds of actions, i.e., (a) waving a hand; (b) raising two hands; (c) clapping hands, were learning as different dynamic patterns (Fig. 3). The actions were realized by changing angles of 8 joints of robot: 2 joints of neck, 6 joints of 2 arms. Table 1 shows the values of parameters used in the experiment.

In Fig. 4, a case of angles of right shoulder pitch joint in Y direction is shown to compare the difference of learning results, generation results between different RNNPBs. It can be confirmed that the proposed method resulted more precise time series patterns than the conventional model, especially comparing the generation



(a) A case of the conventional RNNPB



(b) A case of the proposed RNNPB

Fig. 4. Learning results and Generation results of RNNPBs. For Action 1, waving a hand, angles of right shoulder pitch joint Y direction.

results.

The generation errors of different RNNPBs are compared in Table 2. Average errors of 3 actions showed that the proposed method (3.72 radian) had a better results than the conventional RNNPB (6.15 radian).

Table 1. Parameters used in the experiment.

Description	Symbol	Value
The number of units in Input layer	N	8
The number of units in Output layer	N	8
The number of units in Middle layer	H	30
The number of units in PB layer	P	1
The number of units in Context layer	M	30
Learning rates of output, context layer and input layer connections	-	0.01, 0.01, 0.9
Learning coefficient of PB units	η_1	0.1
Learning rate of PB units	η_2	0.9
Length of time series (width of time window)	l	20
Internal coefficients f PB units	k_1, k_2	0.5, 0.8
Gradient of sigmoid function	α	5.0
Gradient of sigmoid function for PB	α_{pb}	5.0

Table 2. Comparison of generation error after learning for different RNNPBs.

Description	Jordan type ⁽⁵⁾⁻⁽⁸⁾	Elman type (proposed)
Waving a hand	6.71	3.21
Raising two hands	6.18	4.35
Clapping hands	5.57	3.61
Average	6.15	3.72

Unit: radian

Additionally, the PB values of different actions learned by different RNNPBs are shown in Fig. 5. Similar values of the PB unit in different models were observed.

4. Conclusions

A modified recurrent neural network with parametric bias (RNNPB) was proposed. Instead of the Jordan type RNN proposed by Tani et al., Elman type RNN, which has a more simple structure, was used in the proposed model. The training algorithm of the modified RNNPB adopted error back-propagation (BP) method which has lower computational cost than back-propagation through time (BPTT) used in the conventional network. Actions realized by different time series patterns of joint angles of a humanoid robot were learned using RNNPBs, and the experiment results showed the priority of the proposed model.

The future work of this study is to apply the model to more complex actions learning of the robot. In fact, there are 20 joints in the humanoid robot PALRO, and it is possible to generate or to learn more time series patterns to deal with more difficult works, such as dancing, guiding, etc. The learning time also needs to be reduced to realize these works.

Acknowledgment

A part of this work was supported by Grant-in-Aid for Scientific Research of JSPS (No. 25330287 and No. 26330254).

References

(1) Michael I. Jordan. "Attractor dynamics and parallelism in a connectionist sequential machine", IEEE Computer Society Neural Networks Technology

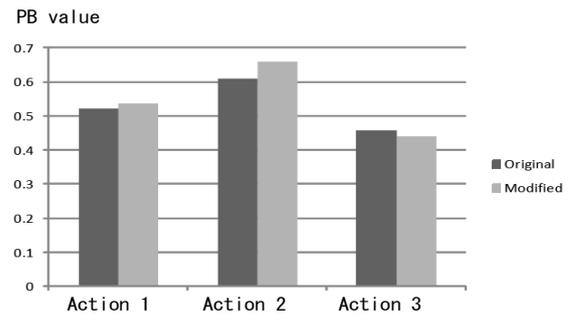


Fig. 5. PB values of different actions after learning in different RNNPBs. "Original" means the case of conventional RNNPB, and "Modified" indicates result of RNNPB proposed here.

Series, pp. 112–127, 1990.

- (2) Michael I. Jordan, and David E. Rumelhart: "Forward models: Supervised learning with a distal teacher", *Cognitive Science*, Vol. 16, pp. 307–354, 1992.
- (3) Jeffrey L. Elman: "Finding structure in time", *Cognitive Science*, Vol.14, pp179-211, 1990.
- (4) Jeffrey L. Elman: "Distributed representation, simple recurrent networks, and grammatical structure", *Machine Learning*, Vol. 7, pp. 195-225, 1991
- (5) Jun Tani: "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes", *Neural Networks*, Vol. 16, Issue 1, pp. 11–23, 2003.
- (6) Jun Tani and Masato Ito: "Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 33, No. 4, pp. 481-488, 2003.
- (7) Jun Tani, Masato Ito, and Yuuya Sugita: "Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB", *Neural Networks*, Vol. 17, Issue 8-9, pp. 1273–1289, 2004.
- (8) Masato Ito and Jun Tani: "Generalization in learning multiple temporal patterns", *Lecture Note in Computer Science*, Vol. 3316, pp. 592–598, 2004.
- (9) Tetsuya Ogata, Hayato Ohba, Jun Tani, Kazunori Komatani and Hiroshi G. Okuno: "Extracting multimodal dynamics of objects using RNNPB", *Journal of Robotics and Mechanics*, Vol. 117, No. 6, pp. 681-688, 2005.
- (10) Raymond H. Cuijpers, Floran Stuijt, and Ida G. Sprinkhuizen-Kuyper: "Generalization of action sequences in RNNPB networks with mirror

properties”, Proceedings of European Symposium on Artificial Neural Networks – Advances in Computational Intelligence and Learning, pp. 251-256, 2009.

- (11) Takashi Kuremoto, Kouichi Hashiguchi, Keita Morisaki, Shun Watanabe, Kunikazu Kobayashi and Masanao Obayashi: “Multiple action sequence learning and automatic generation for a humanoid robot using RNNPB and reinforcement learning”, Journal of Software Engineering and Applications, Vol. 5, pp.128 -133, 2012
- (12) David E. Rumelhart, Geoffrey E. Hinton and Renald J. Willimas: “Learning internal representations by error propagation”, in “Parallel Distributed Processing: Explorations in the Microstructure of Cognition” (eds: David E. Rumelhart and James L. McClelland), MIT Press, pp. 318-362, 1986.
- (13) Ronald J. Williams and David Zipser: “A learning algorithm for continually running fully recurrent neural networks”, Neural Computation, Vol. 1, No. 2, pp. 270-280, 1989.