Robot Trajectory Prediction and Recognition based on a Computational Mirror Neurons Model

Junpei Zhong, Cornelius Weber, and Stefan Wermter

Department of Computer Science, University of Hamburg,
Vogt Koelln Str. 30, 22527 Hamburg, Germany
{zhong,weber,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

Abstract. Mirror neurons are premotor neurons that are considered to play a role in goal-directed actions, action understanding and even social cognition. As one of the promising research areas in psychology, cognitive neuroscience and cognitive physiology, understanding mirror neurons in a social cognition context, whether with neural or computational models, is still an open issue [5]. In this paper, we mainly focus on the action understanding aspect of mirror neurons, which can be regarded as a fundamental function of social cooperation and social cognition. Our proposed initial architecture is to learn a simulation of the walking pattern of a humanoid robot and to predict where the robot is heading on the basis of its previous walking trajectory.

Keywords: Recurrent Neural Network, Parametric Bias, Mirror Neurons, Robot Walking Pattern

1 Introduction

Mirror neurons are a kind of premotor neurons that exist in primates (e.g. monkeys). They have been detected in monkey's premotor area (F5), but the existence of mirror neurons in humans is also evident in the premotor cortex and the inferior parietal lobule (IPL) [8]. According to the experiments done with monkeys, mirror neurons are involved in learning by imitation and social cognition. Moreover, in cooperation with environment affordances, they have the ability to indicate motor actions by activating corresponding neurons in the superior temporal sulcus (STS), indicating the code of action [10].

One class of computational models of mirror neurons focuses on the action imitation property, for instance, the modular action approaches by Demiris [4,3] and Wolpert [18,9]. These models emphasize *how* to generate motions by decentralized automatic modules of the action parts using mirror neurons. The core of these architectures are multiple forward-inverse models, which compete for control based on the selection of likelihoods to imitate actions. Also the selection of a controller can be regarded as action recognition to some extent.

With the further discovery that the mirror neuron system takes part in the recognition of *what* the agent is doing by both the observations from the executors the executions of own actions by the primate itself, further implementations refer to the charateristics incorporating different network patterns to deal with multimodal inputs, in parallel to the vision, somatosensory or auditory stimuli in the human brain, so that the network discharges as a goal-directed action layer. This idea was realized by an association network based on the Helmholtz machine, in which goal-directed codes were associated with vision and language representations, as well as the output of motor actions [7,17]. The learned association enabled neurons of the hidden layer to behave like mirror neurons.

Another network model called RNNPB advocated by Tani et al. [15] can generate and recognize temporal actions by the self-organizing property of an additional layer, called parametric biases units (PB Units). This model deals with temporal inputs with the PB units acting as additional biases to compensate the back-propagation error (Figure 1(a)). Three running modes (learning, observing and action generation) functionally simulate different imitation stages of mirror neurons in the human brain. With the ability of recognizing and generating the desired action from previous training, the PB units behave in a similar manner as mirror neurons in human brain [2].

Recent research [16,11] further suggests that it is not necessary to split *how* and *what.* In fact, as an integrated process, all mirror neurons receive information from visual inputs and represent both *how* the agent does an action and *what* the agent is doing, supporting that these two levels can be switched flexibly.

To integrate the two levels of understanding of mirror neurons and their related processes, the target of our project is to establish a computational model that enables a robot to recognize and understand the action sequence of another robot without communication, and then to act by mimicking or imitation [1]. Due to PB units' property of recognizing temporal input sequences, PB units should be beneficial to robot action understanding. With a prototype architecture of PB units, we will focus on robot trajectory prediction and recognition in the following sections, and test their ability to understand *what* the robot is doing. In the remainder of this paper, we will present a new combined version of recurrent network with PB units and an experiment for predicting and recognizing robot trajectories.

2 SRNPB for Robot Trajectory Recognition

The Simple Recurrent Network with Parametric Biases (SRNPB) is based on an Elman network [6] with additional PB units. In a SRN, the hidden layer is fully connected with its previous state, so that it is not only updated with the external input but also with activation from the previous forward propagation. With the recurrent input from the hidden layer, rather than from the output layer as in the RNNPB [15], the short term memory processed in the Elman Network is attractive for sequential processing, because the error which is back-propagated in the recurrent process is smaller. Furthermore, the recurrent connections in the hidden layer can be seen as related to the horizontal connections in the human cortex. By directly delivering the past history of the hidden layer instead of relying only on the output memory as in the RNNPB, the SRNPB should learn the temporal data with less effort from the network dynamic uncertainty and the learning capability is expected to improve. A layer of parametric bias units is also connected to the hidden layer. Figure 1(b) shows the architecture of the SRNPB. It is based on an Elman Network [6], with an additional layer connecting to the hidden layer as the parametric biases, which serve as adaptable biases learned by the back-propagation through time (BPTT) algorithm [13]. There exist three running modes in SRNPB. In the learning mode, all connection weights and PB values are updated by BPTT. In the recognition mode, only the PB units are updated. The PB value and manually set in generation mode. The determined synaptic weights are common for all learning patterns, but the parametric biases are the sigmoid function values (Equation 2) of the interval values of the PB units. The outputs of the PB units also act as a compensation for the network output error.

Learning mode: The learning is performed supervised and off-line; when providing the training stimulus (positions) for each pattern, the weights are updated with BPTT from layer to layer. Similarly, the internal values of the PB units are also updated. In the updating of PB units, we refer to one entire learning cycle (all sequences) as an epoch e. In each epoch, the kth PB unit u updates its internal value based on the summation of the back-propagation error from the whole sequence (Equation 1 and 2).

$$u_{k,e+1}^{PB} = u_{k,e}^{PB} + \eta_l \sum_{t=1}^{T} \delta_{k,t}^{PB} \tag{1}$$

$$p_{k,e} = \frac{1}{1 + e^{u_{k,e}^{PB}}} \tag{2}$$

$\delta_{k,t}^{PB}$ represents the back-propagation error of the PB unit u_k at time-step t, and η_l is the learning rate of the PB unit, $p_{k,e}$ is the output value of parametric units which are transfering into the hidden layer via full connectivity. T represents the length of the whole training sequence, so the update value of the internal value is the summation of the error based on the whole sequence multiplied by a specific updating rate.

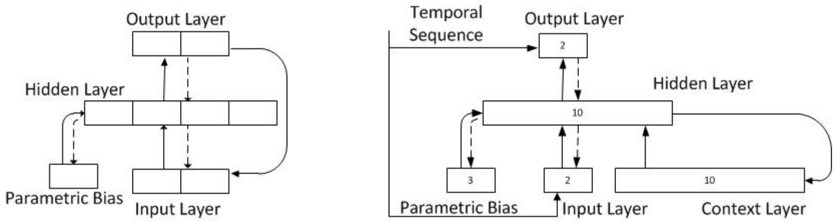
Action recognition mode: This mode is responsible for the recognition by updating the PB units according to the past observation. The information flow is mostly the same as in the learning mode, except that the synaptic weights are not updated. The generated error between target and prediction is only back-propagated and updated into the PB units. If a trained sequence is presented to the network, the activation of the PB units will converge to the values that were previously shown in the learning mode.

The internal values of PB unit are updated by:

$$u_{k,t+1}^{PB} = u_{k,t}^{PB} + \eta_r \sum_{step=t-l}^{t} \delta_{k,step}^{PB} \tag{3}$$

where η_r is the updating rate. The internal values of the PB units are integrated with a specific time length l. The reason of doing a summation is that we expect it can smooth the local fluctuations of the error to decrease its effect and maintain the PB unit at a stable value.

Action generation mode: After learning and after the synaptic weights are determined, the SRNPB can act in a closed-loop, in which the output prediction can be applied as the input for the next time step. The network can automatically generate a trajectory by externally setting the PB unit, representing action codes (*what* to do). Although action generation has not been tested in the experiment, we regard it as an important feature to give commands for robot behavior in future experiments.



(a) Network Architecture of RNNPB

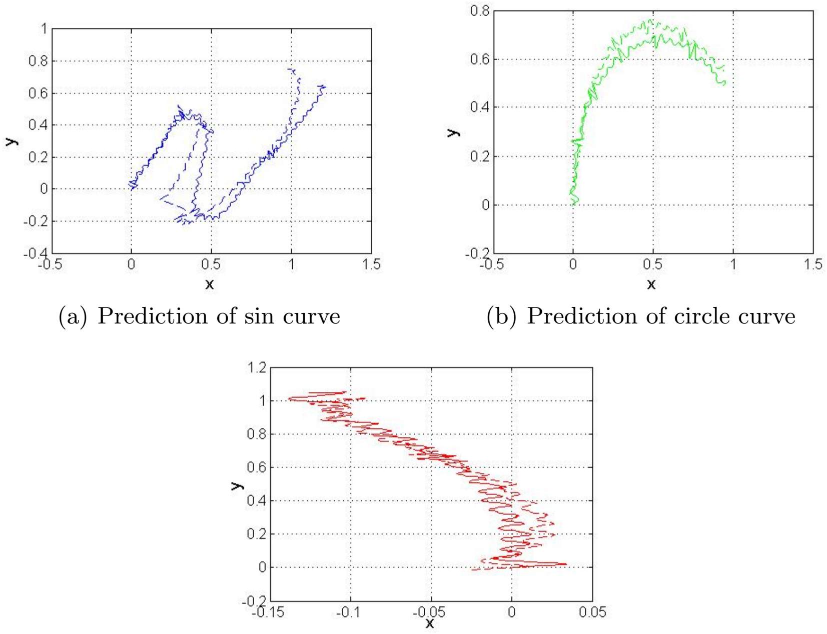
(b) Network Architecture of SRNPB

Fig. 1. Comparison of architectures of RNNPB and SRNPB

3 Experimental Results

As a foundation of robot action understanding, the recognition and prediction of robot walking trajectories are the objectives of the following experiments. For effectiveness, we use the Webots simulator [12] to collect the trajectory data. Our NAO robot is controlled in the Webots simulator to walk along predefined trajectories. From the supervisor function within Webots, three kinds of trajectories, that is a straight line, a sine curve and a half circle were recorded using x and y coordinates. Different combinations of these curves with different parameters make the robot walk in various trajectories. As an initial controlled experiment with known trajectories, we select three trajectories: 1. sine curve: $y = 0.5sin(\frac{2\pi}{3} * x)$; 2. a half circle curve with 0.57m radius; 3. straight line.

The reason why we use these kinds of trajectories is that they can be combined into different kinds of trajectories, e.g. into a walking trajectory when



(a) Prediction of sin curve

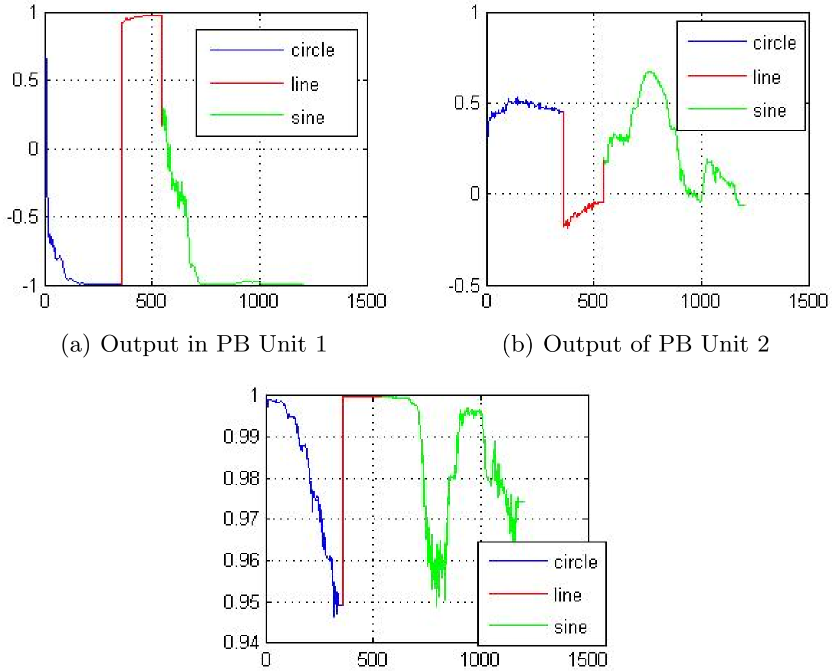
(b) Prediction of circle curve

(c) Prediction of straight line

Fig. 2. Prediction of three curves: Prediction experiments were done for three types of trained sequences. Solid lines represent the true positions and the dashed line represent the predictions. We can see the predicted sequence and the target sequence were quite close in the above figures.

doing obstacle avoidance, by changing their parameters, i.e. frequency and amplitude in sine curve and radius in half circle. We train the network with three types of input sequences. The expectation is that the generalization ability of PB units can recognize similar trajectories with different parameters. As denoted in the remarks in Figure 1(b), for all simulations we use the same network: 2 input nodes, 10 hidden nodes, 10 context nodes, 2 output nodes. Additionally, we use 3 PB nodes in the experiment. The empirically determined network parameters are: $l = 30, \eta_l = 0.01, \eta_r = 0.5$, and the learning rate of connection weights in back-propagation is defined by $\eta_{BP} = 0.01$.

After training, we input the walking records from the above three trajectories respectively and attempt to predict position one step ahead given the previous inputs. As shown in Figure 2, after several time-steps, the network can basically predict the learned trajectories. Furthermore, we inspect the values in the PB units while we continuously feed three types of input sequences into it. As shown in Figure 3 internal values in the PB units can reflect different input patterns of the whole learning sequence.



(a) Output in PB Unit 1

(b) Output of PB Unit 2

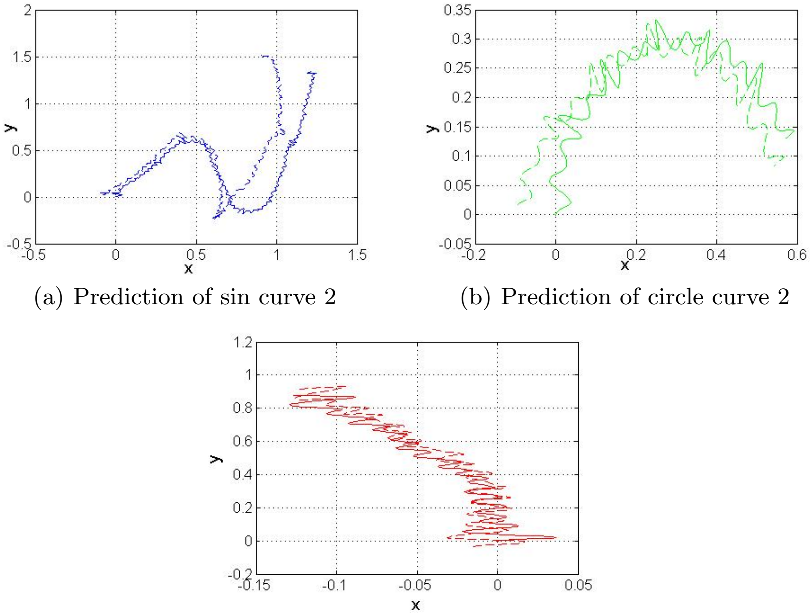
(c) Output of PB Unit 3

Fig. 3. Three sequences were fed into the network to demonstrate the recognition in PB units.

Secondly, we attempt to input another three different types of similar patterns, but with different parameters in order to test the generalization ability for other untrained trajectories. Figure 4 shows the prediction results of the network. Although some errors occur, the generalization of the network still successfully predicts the trend of the curves: 1. sine curve: $y = sin(\frac{\pi}{2} * x)$; 2. half circle curve with 0.3m radius; 3. straight line. The Table 1 below shows the RMS error between the true value and prediction.

RMSE	sine	line	circle	sine2	line2	circle2
x coordinate	0.0714	0.0052	0.0077	0.2655	0.0187	0.0427
y coordinate	0.0829	0.0066	0.0108	0.1884	0.0094	0.0744

Table 1. Root Mean Square Error of two curve sets predictions



(a) Prediction of sin curve 2

(b) Prediction of circle curve 2

(c) Prediction of straight line 2

Fig. 4. Prediction of three untrained curves: The errors between the predicted curves and untrained curves were larger than those in Fig. 2, but the trend of the similar curves can also be predicted.

4 Conclusion and Future Works

In this paper, the Simple Recurrent Network with Parametric Biases (SRNPB) is developed for trajectory prediction simulation. The SRNPB is an Elman network with parametric biases, which can predict the robot trajectory as temporal sequence and recognize the trajectory type. After training, the network not only shows the prediction and recognition ability of the robot walking trajectory, but also the generalization ability during prediction of unknown trajectories records. The internal values in PB units can be interpreted as the different discharging rate of mirror neurons indicating the action codes. To conclude, the temporal property is an important feature for action understanding and imitation, therefore, a recurrent network could be a basic prototype building block for modeling mirror neurons to understand *what* the action sequence means [14].

In future research, the generalization ability will be further used as information for *how* to generate similar actions. Also the future architecture may incorporate multi-modal inputs as a representation of environment affordances, so that the network will integrate multiple temporal sequences from both auditory and vision inputs.

Acknowledgments. This research has been partially supported by the EU projects RobotDoc under 235065 from the FP7, Marie Curie Action ITN and KSERA under n°2010-248085 for Research and Technological Development.

References

1. Cakmak, M., DePalma, N., Arriaga, R., Thomaz, A.L.: Computational benefits of social learning mechanisms: Stimulus enhancement and emulation. In: Proceedings of the 2009 IEEE 8th International Conference on Development and Learning. pp. 1–7. DEVLRN '09, IEEE Computer Society (2009)
2. Cuijpers, R., Stuijt, F., Sprinkhuizen-Kuyper, I.: Generalisation of action sequences in RNNPB networks with mirror properties. In: Proceedings of the European Symposium on Neural Networks (ESANN) (2009)
3. Demiris, Y., Johnson, M.: Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning. Connection Science 15(4), 231–243 (2003)
4. Demiris, Y., Khadhouri, B.: Hierarchical attentive multiple models for execution and recognition of actions. Robotics and Autonomous Systems 54(5), 361–369 (2006)
5. Dinstein, I., Thomas, C., Behrmann, M., Heeger, D.: A mirror up to nature. Current Biology 18(1), R13–R18 (2008)
6. Elman, J.: Finding structure in time. Cognitive science 14(2), 179–211 (1990)
7. Elshaw, M., Weber, C., Zochios, A., Wermter, S.: A mirror neuron inspired hierarchical network for action selection. Proc. NeuroBotics pp. 89–97 (2004)
8. Fogassi, L., Ferrari, P., Gesierich, B., Rozzi, S., Chersi, F., Rizzolatti, G.: Parietal lobe: from action organization to intention understanding. Science 308(5722), 662 (2005)
9. Haruno, M., Wolpert, D., Kawato, M.: Hierarchical mosaic for movement generation. In: International Congress Series. vol. 1250, pp. 575–590. Elsevier (2003)
10. Jellema, T., Baker, C., Oram, M., Perrett, D.: Cell populations in the banks of the superior temporal sulcus of the macaque and imitation. The imitative mind: Evolution, development, and brain bases pp. 267–290 (2002)
11. de Lange, F., Spronk, M., Willems, R., Toni, I., Bekkering, H.: Complementary systems for understanding action intentions. Current biology 18(6), 454–457 (2008)
12. Michel, O.: Webots: Professional mobile robot simulation. Journal of Advanced Robotics Systems 1(1), 39–42 (2004)
13. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation, pp. 673–695. MIT Press, Cambridge, MA, USA (1988)
14. Sugita, Y., Tani, J.: Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. Adaptive Behavior 13(1), 33 (2005)
15. Tani, J., Ito, M., Sugita, Y.: Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB. Neural Networks 17(8-9), 1273–1289 (2004)
16. Thioux, M., Gazzola, V., Keysers, C.: Action understanding: how, what and why. Current biology 18(10), R431–R434 (2008)
17. Wermter, S., Weber, C., Elshaw, M., Gallese, V., Pulvermuller, F.: A mirror neuron inspired hierarchical network for action selection. Biomimetic neural learning for intelligent robots pp. 162–181 (2005)
18. Wolpert, D., Kawato, M.: Multiple paired forward and inverse models for motor control. Neural Networks 11(7-8), 1317–1329 (1998)