ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Generalisation of action sequences in RNNPB networks with mirror properties

Raymond H. Cuijpers[1,2] Floran Stuijt[2]
Ida G. Sprinkhuizen-Kuyper[2]*

[1]Eindhoven University of Technology - Human Technology Interaction
Den Dolech 2, P.O. Box 5600 MB, Eindhoven - The Netherlands

[2]Radboud University Nijmegen - Donders Institute for Brain, Cognition and Behaviour
Montessorilaan 3, P.O. Box 9104, 6500 HE Nijmegen - The Netherlands

Abstract. The human mirror neuron system (MNS) is supposed to be involved in recognition of observed action sequences. However, it remains unclear how such a system could learn to recognise a large variety of action sequences. Here we investigated a neural network with mirror properties, the Recurrent Neural Network with Parametric Bias (RNNPB). We show that the network is capable of recognising noisy action sequences and that it is capable of generalising from a few learnt examples. Such a mechanism may explain how the human brain is capable of dealing with an infinite variety of action sequences.

1 Introduction

The human mirror neuron system (MNS) is active both during observing and performing actions [1]. Because of this *mirror* property various authors have suggested the involvement of the MNS in understanding actions of others [2, 3, 4]. Theoretically, mirror neurons could be involved in simulating the perceptual consequences of actions[5] through forward modelling [6, 7]. In this view mirror neurons can be thought of as representing a particular action from an action repertoire. As soon as an action is observed the corresponding mirror neurons will fire. An immediate implication is that it is easiest to recognise one's own actions [8].

Tani and colleagues [9, 10] constructed a Recurrent Neural Network with Parametric Bias (RNNPB) that was capable of learning, recognising and generating observed actions. This clearly gives the network mirror properties. One very interesting aspect of the RNNPB architecture is that the same network can represent multiple actions, in contrast to the assumption that different mirror neurons represent different actions. But can such a system learn a large variety of actions? Note that Ito and Tani [11] also considered the problem of generalizability. However, our experimental setup differs from their's. In Section 2, we present the RNNPB architecture. In Section 3 we will analyse the generalising capabilities RNNPB model . We conclude with discussion and conclusions in Section 4.

*The present study was supported by the EU-Project "Joint Action Science and Technology" (IST-FP6-003747).

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

2 The RNNPB architecture

The RNNPB architecture is a modified version of the Jordan RNN [12]. Once trained these networks produce the next time step of a learnt time series. This property enables the RNNPB architecture to learn action sequences as was demonstrated in [10]. The network consists of several layers of neurons (Fig. 1). The hidden layer receives inputs from the context, input and the parametric bias (PB) layer. The context layer stores previous activations of the hidden layer. All layers except the input layer have sigmoid activation functions. The nodes in the PB layer correspond to the mirror neurons because they encode the sequences generated by the network. The output is copied to the input layer in a one-to-one fashion. There are three modi operandi for the RNNPB architecture: learning mode, recognition mode and generation mode, which we will discuss next.

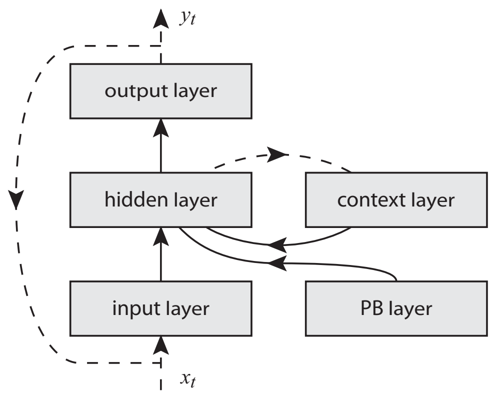

Figure 1: RNNPB architecture. The context layer is a copy of the previous activations of the hidden layer, and the ouput layer is copied to the input layer (dashed arrows). The solid arrows denote fully connected layers.

In learning mode, the input at time t to the input layer is the weighted average of the external input and the recurrently connected output. If we use vectors to denote the activity of nodes within a single layer, we can write:

$$\mathbf{u}_t^{\text{input}} = \beta \mathbf{y}_{t-1} + (1-\beta)\mathbf{x}_t \,, \tag{1}$$

where $\mathbf{x}_t$ is the external input, $\mathbf{y}_t$ the output, $\mathbf{u}_t^{\text{input}}$ the internal activation of the input layer at time t, and β the relative strength of the recurrent output and the external input. The error of the network's output is given by the difference between actual and desired output (i.e. the next time step of the external input):

$$\delta_t^{\text{output}} = \mathbf{y}_t - \mathbf{x}_{t+1} \,.$$

The connection weights between layers (solid arrows in Fig. 1) are updated using the back-propagation through time (BPTT) algorithm [13]. In order to

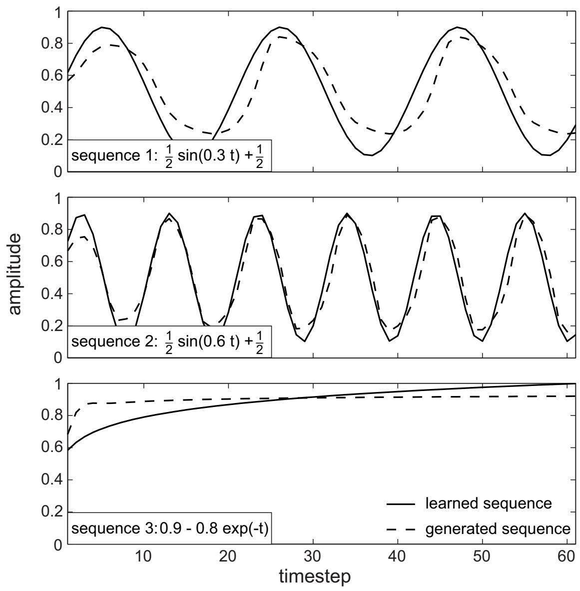

Figure 2: The network's ability to generate previously learnt sequences.

let the network predict multiple, sequentially presented time series, a different PB vector is updated for each time series. All PB vectors are initialised at zero. The updating of these PB vectors and the internal connection weights takes place after all time series have been presented. From here on we refer to this entire cycle as one epoch, denoted by e. The internal values of the PB vector of the k^{th}time series ($\mathbf{u}^{\text{PB}}_{k,e}$) are updated according to[1]:

$$\mathbf{u}^{\text{PB}}_{k,e+1} = \mathbf{u}^{\text{PB}}_{k,e} + \eta \sum_{t=1}^{T} \delta^{\text{PB}}_{k,t} \,,$$

$$\mathbf{p}_{k,e} = \text{sigmoid}(\mathbf{u}^{\text{PB}}_{k,e}) \,,$$

where $\delta^{\text{PB}}_{k,t}$ represents the back-propagated error for the PB layer at time step t of the k^{th} time series, η is the learning rate of the PB layer, and T is the duration of the each time series.

The recognition phase only differs from the learning phase in that the internal weights of the network are not updated. Only the PB vectors are updated.

In the generation phase, the network generates the previously learnt time series by setting the PB vector to the appropriate value and running the network in closed loop ($\beta = 1$ in Eq. 1). During this phase, no updating takes place.

[1]In [10] the symbol ρ was used to refer to internal values the PB layer and p to refer to output values of the PB layer.

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

3 Generalising capabilities of the RNNPB architecture

In order to verify that the RNNPB architecture is capable of learning multiple time series, we trained the network with three different time series. For all our simulations we used the following architecture: 1 input node, 1 output node, 5 hidden nodes, 2 context nodes, and 2 PB nodes. The network's learning parameters were: $\eta = 0.01$, $\beta = 0.1$, $\eta_{BP} = 0.02$ and $\alpha = 0.9$, where η_{BP} and α respectively denote the learning rate and momentum parameter for the BPTT algorithm. The learnt sequences and the network's output in generation mode are shown in Fig. 2. It is clear that the network has captured the amplitude and periodicity of the learnt sequences.

The PB vectors that were obtained during training are $\mathbf{p}_1 = (0.6147, 0.546)$, $\mathbf{p}_2 = (0.3905, 0.206)$ and $\mathbf{p}_3 = (0.4553, 0.6511)$. To quantify how well the network could recognise the first sequence (Fig. 2), we used the Euclidean norm $\varepsilon_k = ||\mathbf{p}_{\text{actual}} - \mathbf{p}_k||$ as an error measure. We found that all three learnt signals were successfully recognised in the absence of noise ($\varepsilon_k < 0.02$ for correct k and $\varepsilon_k > 0.18$ for incorrect k). In the presence of Gaussian noise the network could reliably recognise all three time series for $\sigma < 0.18$. This is shown for the recognition of sequence 1 in Fig. 3. The error ε_1 is smaller than the other errors until the noise level exceeds $\sigma > 0.18$. Similar results were obtained for recognition of the other sequences.

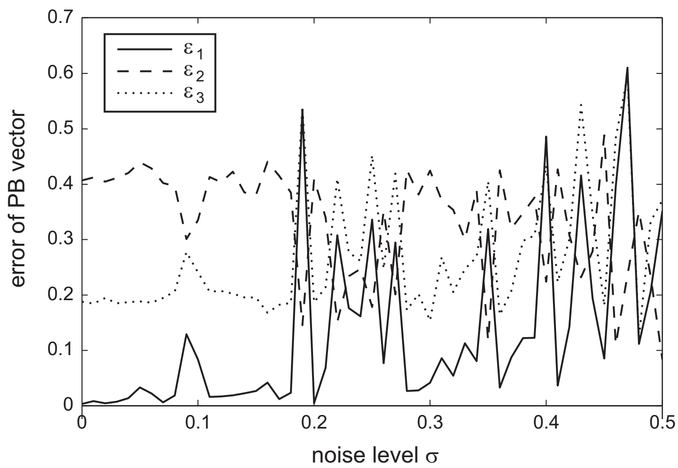

Figure 3: Ability to recognise sequence 1 when it is corrupted with Gaussian noise. The error ε_k of the recognised PB vector with respect to each of the learnt PB vectors $\mathbf{p}_k$ is plotted as a function of noise level σ.

We were also interested in whether the RNNPB network could generalise across frequencies and amplitudes of the learnt signals. If so, we would expect that the PB vector depends systematically on the amplitude and frequency of sequence that the network tries to recognise. First, we varied the angular fre-

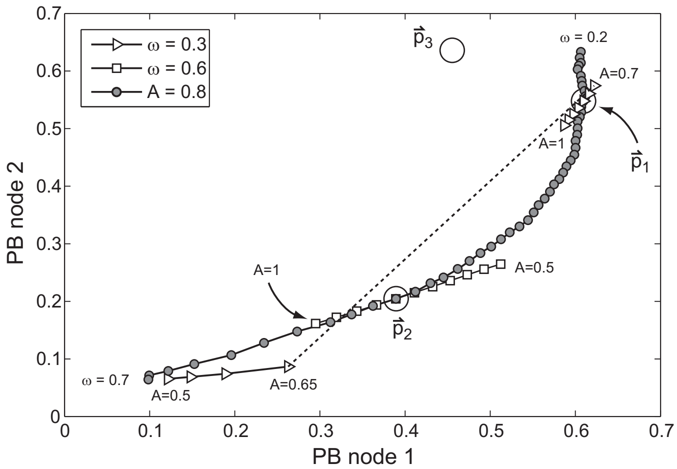

Figure 4: Coordinates of the PB vector after recognising sinusoidal sequences of varying amplitude an frequency. The learnt PB vectors are indicated by large circles. The angular frequency was varied from 0.2 to 0.7 in steps of 0.01 for a fixed amplitude of 0.8 (grey circles). Amplitude was varied from 0.5 to 1 in steps of 0.05 for angular frequencies of 0.3 (squares) and 0.6 (triangles).

quency from 0.2 to 0.7 in steps of 0.01 and fixed the amplitude to the trained value ($A = 0.8$). The coordinates of the PB vector (values of the PB nodes) are plotted in Fig. 4 for each frequency (grey dots). As can be seen the PB vectors lie on a continuous curve through the learnt PB vectors (large circles). This shows that the network has captured the notion of angular frequency. In a similar fashion we varied the amplitude for the two learnt frequencies. For $\omega = 0.6$ (triangles) the PB vector varies smoothly for amplitudes $A \geq 0.7$. For smaller amplitudes the PB vector shows a large jump (dashed line). Thus, the network was capable to generalise across amplitudes but for a limited range. For $\omega = 0.3$ (squares) no jumps are observed but the curve practically coincides with the curve that was obtained by varying the angular frequency (grey circles). This means that near $\omega = 0.3$ the network is sensitive to changes of both amplitude and frequency, but it is unable to distinguish between them.

4 Discussion and conclusions

The RNNPB architecture is capable of learning, recognising and generating multiple action sequences. Simulations have shown that the recognition of action sequences is quite robust against noise. In agreement with [11] we found that the RNNPB architecture is capable of generalising the frequency of sinusoidal sequences. Generalisation of the amplitude was limited. Either the network became unstable if the amplitude deviated too much from the amplitude during learning, or the network could not distinguish between changes in amplitude and changes in frequency. The reason for this may be that the network never learnt more than one amplitude in the first place.

The PB nodes are analogous to mirror neurons in a strict sense because their activity is the same during generation and recognition of action sequences. However, human brain imaging is not sensitive to individual neurons, so that the entire RNNPB network would light up in imaging studies. Thus, the number of strict mirror neurons may be quite sparse in the MNS. Generalising action sequences from a few learnt examples could potentially explain how the human brain is capable of dealing with an infinite variety of action sequences. It also implies that neurons in the MNS may be capable of simultaneously representing multiple action sequences depending on the activity of a few strict mirror neurons who parametrically bias the network's dynamics.

References

[1] G. Rizzolatti and L. Craighero. The mirror-neuron system. *Annual Review of Neuroscience*, 27:169–192, 2004.

[2] M. Iacoboni, I. Molnar-Szakacs, V. Gallese, G. Buccino, J.C. Mazziotta, and G. Rizzolatti. Grasping the intentions of others with one's own mirror neuron system. *PLoS Biology*, 3(3):e79, 2005.

[3] U. Castiello. Understanding other people's actions: intention and attention. *Journal of Experimental Psychology: Human Perception and Performance*, 29:416–430, 2003.

[4] H. Bekkering, A. Wohlschläger, and M. Gattis. Imitation of gestures in children is goal-directed. *The Quarterly Journal of Experimental Psychology Section A*, 53(1):153–164, 2000.

[5] V. Gallese and A. Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501, 1998.

[6] E. Oztop, M. Kawato, and M. Arbib. Mirror neurons and imitation: A computationally guided review. *Neural Networks*, 19(3):254–271, 2006.

[7] D. M. Wolpert, K. Doya, and M. Kawato. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of The Royal Society Of London. Series B: Biological Sciences*, 358(1431):593–602, Mar 2003.

[8] B. Calvo-Merino, J. Grèzes, D. E. Glaser, R. E. Passingham, and P. Haggard. Seeing or doing? influence of visual and motor familiarity in action observation. *Current Biology*, 16(19):1905–1910, Oct 2006.

[9] M. Ito and J. Tani. On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behavior*, 12:93–115, 2004.

[10] J. Tani, M. Ito, and Y. Sugita. 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.

[11] M. Ito and J. Tani. Generalization in learning multiple temporal patterns. In N. R. Pal, N. Kasabov, R. K. Mudi, S. Pal, and S. K. Parui, editors, *Neural Information Processing*, LNCS 3316, pages 592–598. Springer Berlin / Heidelberg, 2004.

[12] M.I. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. *IEEE Computer Society Neural Networks Technology Series*, pages 112–127, 1990.

[13] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986.