A Compositionality Assembled Model for Learning and Recognizing Emotion from Bodily Expression

Junpei Zhong[1] and Chenguang Yang[2]

Abstract— **When we are express our internal status, such as emotions, the human body expression we use follows the principle of compositionality, which asserts that the single components of the bodily expression as well as the rules used to combine them are the major parts to finish this process. In this paper, such principle is applied to the process of expressing and recognizing emotional states through body expression, in which certain key features can be learned to represent certain primitives of the internal emotional state in the form of basic variables. This is done by a hierarchical recurrent neural learning framework (RNN) because of its nonlinear dynamic bifurcation, so that variables can be learned to represent different hierarchies. In addition, we applied some adaptive learning techniques in machine learning for the requirement of real-time emotion recognition, in which a stable representation can be maintained compared to previous work. The model is examined by comparing the PB values between the training and recognition modes. This hierarchical model shows the rationality of the compositionality hypothesis by the RNN learning and explains how key features can be used and combined in bodily expression to show the emotional state.**

I. Introduction

Identifying the user's emotional state raises interest in human-computer interaction (HCI) applications because it is necessary to capture the nuances of emotional expression in these applications. Basically, these recognition techniques can interpret emotions from features captured by different modalities such as facial expressions, body expressions or speech. The role of emotional state in humans is often one of the important clues to better understanding social interaction applications.

Similar to facial expressions citedarwin1998expression, ekman1992argument, psychological studies have shown why some body postures/actions are related to emotional expression. Some literature (e.g. [1, 2]) supports the perception of emotions stemming from the key features of body shape and body movement. This is in line with our intuition between head angle and emotional state, and it provides us with a direct way of identification. For example, Patterson's work [3] is to discover such key features as the angle of the head and the movement of the arm, and the connection to the emotional space. He found that not only static features (poses), but also the dynamics of features, such as the speed of movement, also play a major role in distinguishing emotions.

* Corresponding author: zhong@junpei.eu

[1]Nottingham Trent University, Nottingham, United Kingdom

[2]Key Laboratory of Autonomous Systems and Networked Control, School of automation science and engineering, South China University of Technology, Guangzhou, China

On the other hand, during the recognition process, those temporal integration occurs and forms the global form information [4]. According dual pathway studies from neuroscience, the human brain perceives visual information with form and motion information in two separate visual pathways, the ventral and dorsal pathways, within which the ventral pathway is greatly influenced by the emotional stimuli [5, 6]. Such emotional stimuli, for example, stimuli from bodily expression may have more rapid access to the brain's processing resources.

On the other hand, during the recognition process, those temporal integration take place and form a global form information [4]. According to neuroscience research, the human brain perceives visual information through two independent visual pathways (ventral and dorsal pathways), where the ventral pathway is greatly affected by emotional stimuli cessoa2004neuroimaging, adolphs2008fear. Such emotional stimuli, for instance, stimuli from body expression, can enter the brain's processing resources more quickly through both pathways. Therefore, both the static postures and the dynamic movements contribute to the understanding of the emotion. This is similar to the principle of compositionality, which supports that the meaning of one complex expression can be only traced by the meanings of its constituent parts and the way that they assembles.

The above-mentioned psychological and neuroscience research on combination provides a possible relationship between emotion and body expression. In this way, the two-level hierarchical model we construct in the latter sections explains the hierarchical learning process for parameter combination (such as body movements or "key features") for emotion recognition. In this model, higher level parameters (i.e, basic variables) also represent internal states as basic variables for dynamic interaction. The hierarchical learning structure is extended by the hierarchical perceptron structure [7] and hierarchical perception-action model (PAM) model [8].

As such, to recognise emotions from one's behaviour or to generate a bodily expression, the model should encode the knowledge about the bodily expression with the form of essential variables (i.e. the PB values on the upper hierarchy). Thus the compositionality in this network also explains the memorized critical features from the bodily expression. This work extends our previous work [9] with the following novel adaptive learning methods:

1) an improved learning (Adagrad) method to speed up the learning process.
2) an adaptive learning rate (Adadelta) method to ensure

real-time recognition;

The rest of this paper will be organized as follows. The next section will see how the hierarchical compositionality theory of the critical features can be learnt with the recurrent neural network model. At the third section, we will show how we apply it in both emotion recognition and expression learning. Lastly, brief discussion and summary will be given.

II. Recurrent Neural Networks

The joint angles in the skeletal motion information of human is usually non-linear time series with noise. In general, recurrent neural networks (RNN) use directional connections between neurons. Allowing neurons to maintain a temporal relationship during activation by transmitting internal states to each other, it has proven to be useful to model the non-linear sequences. In our model we use a kind of RNNs that are able to capture the temporal relationship of the datasets, or any length of temporal sequences [10]. In terms of learning ability to learn temporal processes, they therefore stand out from other machine learning methods.

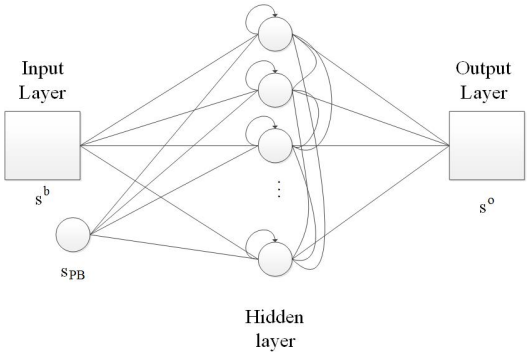


Fig. 1: RNNPB with Elman-like connections
The S^b and S^o are input and output units, while S_{PB} is the parametric units.

Similar as the original RNN, [11, 12] introduced Recurrent Neural Network with Parametric Bias Units (RNNPB). It consists of an additional bias unit with the RNN model, acting as bifurcation parameters which change the non-linear dynamics of the whole RNN model. It can be constructed by either Elman network [13] (Fig. 1) or Jordan network [14]. In both of these models, the adjustable and learn-able additional bias units provide additional term in the hidden layer (Eq.1). For instance, the input of the hidden layer in an Elman based RNNPB can be expressed as:

$$h_t = f(W_h x_t + U_h h_{t-1} + V_h PB + b_h) \qquad (1)$$

where the x_t, h_t and b_h are original terms of the RNN: x_t is the input vector, h_t is the hidden layer vector and b_h is the bias. The additional term of $V_h PB$ is the parametric bias term, which is the feature of the RNNPB. Considering as additional memories, the adjustable PB value can determine the non-linear patterns within the bifurcation dynamics in the model, so a single model can store multiple sequences. Based on this, it endows the following functions compared with the conventional RNNs:

- the PB units work as additional memories to modulate the non-linear dynamics of the network so as it is able to store multiple sequences;
- the PB units are update slowly to exhibit the slow features of the trained sequences;
- in terms of the generation, the PB units also act as a top-down generator to recover the pre-trained sequences by recompute the non-linear dynamics. [1]

In the context of learning, the critical features of the bodily expression, the PB unit can also be regarded as the essential variables of the basic components which follow the compositionality assembly rule for the bodily expression. Due to the above features, an RNNPB network can be regarded as a two-layer hierarchical structure for cognitive systems, where the PB layer acts as a higher level representation which determines/modulates the lower level of representation. The similar concept of hierarchical learning structure has been employed to study the feasibility in both mathematical and biological perspectives to reconstruct the compositionality principle in language acquisition [15, 16], intention understanding [17], etc.

In terms of the training methods, these units are trained from the data sequences with back-propagation through time (BPTT). The slow changing feature of the PB units results from the relatively slower updating rate of the parametric bias than that of ordinary neurons. Therefore, it is able to memorise and capture the feature of the whole temporal sequences. The role of learning the PB units is to do self-organize a representation to memorise the spatio-temporal patterns of the sequences . It is important to note that the PB vector for each learning pattern is self-determined in a non-supervised manner, without any training signals.

There exist three running modes of hierarchical architecture, enabling us to recognize emotions from bodily expression, and to generate bodily expression from our emotions.

A. Three Modes

The three modes of operation are learning, recognition and generation. They functionally mimic the different stages between sensory motion sequences (as body expression) and the high level internal states of these sequences. The neural dynamics and algorithms of these modes are as following:

- Learning mode: The learning algorithm is derived from the BPTT (back-propagation through time). The learning is usually done off-line, and the weights of the network is trained in a supervised way. When the training sample with a new pattern is provided, the weights connecting neurons are updated with BPTT. Moreover, in terms of the training of the PB units, part of the error comes from the BPTT, after shaping the weights of the connection weights, also updates the internal values in PB units. It follows the rule that the

[1] This report will omit this function of generating bodily expression for robots, but we will report it in the future publications.

internal values of the PB units are updated in a self-organising way.

- Recognition mode: In the recognition mode, the type of sequences is being recognized by observing the internal values of PB units and comparing them with the pre-trained values. The information flow in this mode is mostly the same as in the learning mode, i.e. the error is back-propagated from output neurons to the hidden neurons, but the synaptic weights are not updated. Instead, the back-propagated error only contributes to the updating of the PB units. By this mean, if a trained sequence is presented to the network, the values of the PB units converge to the values which were previously shown in the learning mode, which indicates that the sequences are learnt before.
- Prediction mode: After learning and after the synaptic weights are determined, the RNNPB can act in a closed-loop way: the output prediction can be used as an input for the next time step. In principle, the network can generate a trained sequence by providing initial value of the input and externally setting the PB values.

B. Algorithms

The detailed algorithms of the three modes are introduced below. Note that our previous publication [9] also proposed a similar network about a learning model between internal affective states and motor actions, but the novelties of this work mostly contribute on the algorithms we use to ensure a robust convergence in learning and real-time PB values for recognition.

1) Learning Mode: : During learning mode, if the training progress is basically determined by this cost function:

$$C = \frac{1}{2}\sum_{t}^{T}\sum_{k}^{N}(s_k^b(t+1) - s_k^o(t))^2 \tag{2}$$

where $s_i^b(t+1)$ is the one-step ahead input (as well as the desired output), $s_k^o(t)$ is the current output, T is the total number of available time-step samples in a complete sensorimotor sequence and N is the number of output nodes which is equal to the number of input nodes. Following gradient descent, each weight update in the network is proportional to the negative gradient of the cost with respect to the specific weight w that will be updated:

$$\Delta w = -\eta_t \frac{\partial C}{\partial w} = -\eta_t g \tag{3}$$

where η_{ij} is the adaptive learning rate that calculated by Adagrad algorithm [18], g is defined as the gradient of the objective function w.r.t. the weights w. This adaptive learning rate method is used to avoid the uncertainties in the sequences that are brought from the unstable movements. In Adagrad, it adapts the learning rate based on the sparsity of the data: it increases the learning rate with infrequent information (i.e. error) and decreases its value for frequent information. This is done by multiplying a base learning rate with the elements of a vector G which is the diagonal of the outer product matrix.

$$\eta_t = \frac{\eta_i}{\sqrt{G_t + \epsilon}} \cdot g_t \tag{4}$$

where G_t contains the sum of the squares of the past gradients w.r.t. to all the weights along its diagonal as shown up to time t in Eq. 5, ϵ is a smoothing term.

$$G_t = \sum_{\tau=1}^{t} g_\tau g_\tau^T \tag{5}$$

where g_τ is the gradient of the objective function, which has been defined in Eq. 3.

As mentioned before, besides the usual weight update according to back-propagation through time, the accumulated error over the whole time-series also contributes to the update of the PB units. The update for the i-th unit in the PB vector for a time-series of length T also follows a similar adaptive updating rate:

$$\rho_{t+1} = \rho_t + \frac{\gamma_t}{\sqrt{G_t + \epsilon}}\delta^{PB} \tag{6}$$

where δ^{PB} is the error back-propagated to the PB units, t is t-th time-step in the whole time-series (e.g. epoch), γ_t is PB units' initial adaptive updating rate which is similar as the adaptive rate η_t:

$$\gamma_t = \frac{\gamma_i}{\sqrt{G_t + \epsilon}} \cdot g_t \tag{7}$$

Although essentially the original version of PB update is quite similar as Adagrad, the empirical studies we have done showed that it results in a better convergence performance in the PB values. One of the reasons is that every term $\sqrt{G_t}$ stays positive, which results in a stable values in the accumulated PB units.

2) Recognition Mode:: The recognition mode is executed with a similar information flow as the learning mode: given a set of spatio-temporal sequences, the error between the target and the real output is back-propagated through the network to the PB units. However, the synaptic weights remain constant and only the PB units will be updated, so that the PB units are self-organized as the pre-trained values after certain epochs.

Different from the learning mode, we used a sliding-window-like[2] adaptive learning rate method called Adadelta [19]. The update rule of the PB value is defined as:

$$\rho_{t+1} = \rho_t + \frac{RMS[\Delta\Theta]_{t-1}}{RMS[g]_t}\delta^{PB} \tag{8}$$

where δ^{PB} is the error back-propagated from a certain sensory information sequence to the PB units and RMS is root mean square of the training at time $t-1$. And the root mean squared error of updates is:

$$RMS[\Delta\Theta]_t = \sqrt{E[\Delta\Theta^2]_t + \epsilon} \tag{9}$$

[2]Here we do not need to define the sliding window size, but it is hidden in the parameter λ which we define next.

where the $E[\Delta\Theta^2]_t$ is the running average at time step t. It can be updated as follows:

$$E[\Delta\Theta^2]_t = \lambda E[\Delta\Theta^2]_{t-1} + (1-\lambda)\Delta\Theta_t^2 \qquad (10)$$

In Adadelta, it adapts the learning rate depends on the previous average and the current gradient $E[g^2]_t$. This is determined by the fraction λ. In this application, we set the λ equals to 0.9.

3) Generation Mode:: The values of the PB units can also be manually set or obtained from recognition, so that the network can generate the upcoming sequence with one-step prediction, while the current output becomes the input of next time-step:

$$s_k^i(t+1) = s_k^o(t) \qquad (11)$$

When given a fixed PB vector, the RNNPB generates the corresponding dynamic patterns. On the other hand, when given target patterns to be recognized, the corresponding PB vectors are obtained through an iterative inverse computation.

C. Performance and Training

During training, in order to balance the training process among the multiple temporal sequences and to avoid the over-fitting, a threshold is defined to stop the iteration in training for one sequence. One epoch here includes a few iterations for training all the sequence using stochastic gradient descent (SGD) [20, 21], as showed in Algorithm 1:

Algorithm 1 Time-sequences Training

procedure ONE EPOCH($data$) ▷ $data$ contains multiple time sequences.
  for $seq \in data$ **do**
    while $error > threshold$ or $iteration > maximum\_iteration$ **do**
 ▷ Repeat iteration for one sequence until threshold is achieved
      Run SGD(seq)
    end while
  end for ▷ Choose the next sequence
end procedure

We fed the dimension-reduced data by PCA into the RNNPB for training. We presented the two RNNPBs with two sets of training data from two behaviours: walking and standing. To make the training more efficient and practical to map the generated movements into humanoid robots, the sampling rate of the processed sequence was reduced to 6Hz. Each data set for one network included 10 sequences of the spatio-temporal reduced-dimensional data from 5 basic emotions (i.e. 2 sequences for each emotion) were used for training. As the self-organizing property of the PB values, we ran the training for each behaviour three times, the main target of the experiment is to find whether we can extract similar PB values with the same category of emotion. Because of the Adagrad algorithm, the PB units converge to a stable values after a few epochs. The result of PB values after training is shown in Tab. I and II.

Fig. 2a and Fig. 2b show the PB values of two behaviours. In these figures, the same shape of the markers (except the asterisks) indicates the same emotion. We can observe that PB values from the same emotion are located closer to each other in the PB space after the network was learned, which suggests the PB space became a continuous space for different emotions.

Moreover, Tabs. I and II give a quantitative measurement of the distance between each PB values in the PB space. The number in these two table indicate the distance of the PB values between these two sequences (the row and the column) after the training is getting stable.

D. Experiment 2 - Recognition

In this subsection, we mainly investigated the performance of recognition mode of the network. As introduced before, we adopted the Adadelta for the recognition mode to achieve the real-time requirement for the recognition as well as the maintain the stable status for recognition.

Using the same motion capture device to capture the sequences, the detailed algorithm is in Algorithm 2.

Algorithm 2 Time-sequences Recognition

procedure RECOGNITION($data$ from Kinect)
  while do$data$ is updated
    $input$ = Sliding_Window($data$)
  end while
  while $error > threshold$ or $iteration > maximum\_iteration$ **do** ▷ Repeat iteration for one sequence until threshold is achieved
    Recognition($input$)
    Update PB
  end while
end procedure

Similar to the training, 30 new sequences that captured from five emotions and walking/standing behaviours were fed into the pre-trained walking/standing-trained networks, respectively, to test whether the networks can distinguish the emotion correctly.

Due to the usage of Adadelta algorithm, a stopping criterion is usually not necessary to be set to keep the PB values dynamically.

In Figs. 2, we compared the trained PB values and the recognized ones (asterisk markers). Together with the training sets, markers with the same colour imply the same emotions. Tabs. IV and V show the distance in the PB space between the recognized values and the average training values we obtained in the previous experiment, in which we can identify they were locating closer the same emotions in the PB space. In other words, inherited from generic RNN network, the RNNPB network owns generalisation capability to predict/classify untrained sequences.

TABLE I: PB Values after Training (Standing Behaviour)

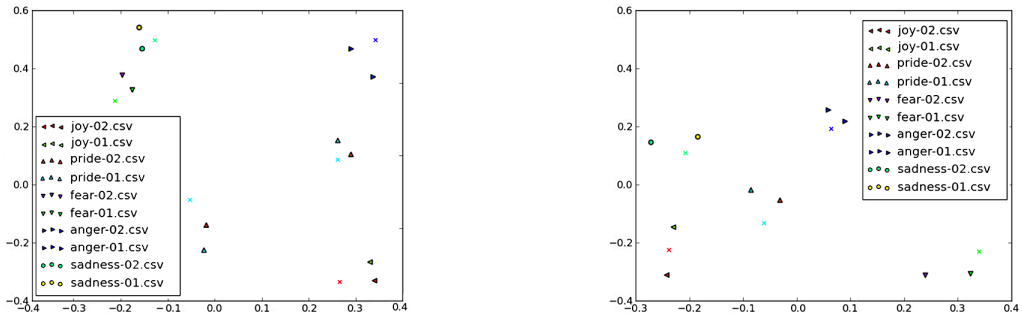
The values represent the distances between two emotions after training. Numbers in bold represent the smallest one in a column.

PB	joy1	joy2	pride1	pride2	fear1	fear2	anger1	anger2	sadness1	sadness2
joy1	—									
joy2	**0.1327**	—								
pride1	0.4520	0.6386	—							
pride2	0.5784	0.5932	**0.1093**	—						
fear1	1.2333	1.1395	0.7479	0.7481	—					
fear2	1.6351	1.2324	0.6813	0.7083	**0.0831**	—				
anger1	0.7909	0.8133	0.7864	0.7591	1.2818	1.3682	—			
anger2	0.7591	0.9591	0.6004	0.6136	1.3136	1.2773	**0.1727**	—		
sadness1	1.1134	1.1220	0.7753	0.7236	0.2018	0.1936	0.7351	0.7691	—	
sadness2	1.1248	1.1981	0.7933	0.7512	0.2492	0.2409	0.7817	0.8102	**0.2727**	—

TABLE II: PB Values after Training (Walking Behaviour)

The values represent the distances between two emotions after training. Numbers in bold represent the smallest one in a column.

PB	joy1	joy2	pride1	pride2	fear1	fear2	anger1	anger2	sadness1	sadness2
joy1	—									
joy2	**0.2109**	—								
pride1	0.5127	0.5983	—							
pride2	0.3424	0.3979	**0.1342**	—						
fear1	0.8211	0.9878	0.6198	0.6083	—					
fear2	0.9825	1.2192	0.6591	0.7065	**0.1986**	—				
anger1	0.7079	0.6129	0.6989	0.4897	0.8954	0.6510	—			
anger2	0.7546	0.7318	0.6448	0.4799	0.7364	0.6108	**0.1101**	—		
sadness1	1.4182	0.4193	0.4899	0.5611	1.5502	1.2219	0.6481	0.6522	—	
sadness2	1.0545	0.3880	0.5221	0.4321	1.2344	1.1710	0.4752	0.5077	**0.1912**	—



(a) Coordinates of the PB Vectors of Standing Behaviour

(b) Coordinates of the PB Vectors of Walking Behaviour

Fig. 2: PB values of different emotions under two behaviours in learning and recognition modes: the same shape of markers indicate the same emotion; PB values in recognition mode were shown with asterisk markers.

TABLE III: Network parameters

Parameters	Parameter's Descriptions	Value
η_i	Initial Learning Rate	2.0×10^{-6}
γ_i	Initial Updating Rate of PB	1.0×10^{-6}
λ	Fraction of PB in recognition	0.9
$n_{b/o}$	Size of Input/Output Layer	4
n_h	Size of Hidden Layer	100
n_{PB}	Size of PB Unit	2

From the video[3], we can also find that the algorithms fulfil the real-time requirement and can be used for emotion recognition for indoor environment.

III. Discussion and Summary

The proposed model applies the idea of the hierarchical system with the perception-action model (PAM). Using RNNPB, we further realises the hierarchical PAM model by the bifurcation functions of non-linear dynamics. During this process, a small number of (higher-level) variables (i.e. PB values), which can be considered as the internal states in our emotion status, mediate the sensorimotor behaviours of bodily expression. Such connections can be learnt as the essential variable in the compositionality assembly model by the hierarchical recurrent neural model. This model is

[3]https://www.youtube.com/watch?v=JusCuKvHg44

TABLE IV: Distance between Recognized Value and Training Value in the PB Space in Standing Behaviour

The row represents the PB location of the training values and the column represents the PB locations of the recognised values.

T. \ R.	joy	pride	fear	anger	sadness
joy	**0.1322**				
pride	0.4844	**0.0918**			
fear	1.1884	0.6413	**0.1301**		
anger	0.6980	0.4101	0.8534	**0.1419**	
sadness	1.3579	0.8298	0.2144	0.8421	**0.0984**

TABLE V: Distance between Recognized Value and Training Value in the PB Space in Walking Behaviour

The row represents the PB location of the training values and the column represents the PB locations of the recognised values.

T. \ R.	joy	pride	fear	anger	sadness
joy	**0.0137**				
pride	0.4121	**0.1429**			
fear	0.9961	0.6721	**0.2011**		
anger	0.6210	0.6352	0.6138	**0.0772**	
sadness	0.5449	0.4118	1.2149	0.4922	**0.1725**

able to explain how the bodily expression associates with emotion status. Based on revised adaptive learning methods, we further propose algorithms for an emotion learning and recognition model from bodily expression based on this hypothesis. Two different adaptive learning rates methods were also used to maintain a stable and convergent learning/recognition results. During training, we used data from motion capture sensor while we used the Kinect sensor to realise real-time emotion recognition

ACKNOWLEDGMENT

The research was supported by Japan New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] M. A. Giese and T. Poggio. "Neural mechanisms for the recognition of biological movements". In: *Nat. Rev. Neurosci.* 4.3 (2003), pp. 179–192.

[2] L. M. Vaina et al. "Intact biological motion and structure from motion perception in a patient with impaired motion mechanisms: A case study". In: *Visual Neurosci.* 5.04 (1990), pp. 353–369.

[3] H. M. Patterson, F. E. Pollick, and A. J. Sanford. "The role of velocity in affect discrimination". In: (2001).

[4] J. Lange and M. Lappe. "The role of spatial and temporal information in biological motion perception". In: *Advances in Cognitive Psychology* 3.4 (2007), pp. 419–428.

[5] L. Pessoa and L. G. Ungerleider. "Neuroimaging studies of attention and the processing of emotion-laden stimuli". In: *Progress in brain research* 144 (2004), pp. 171–182.

[6] R. Adolphs. "Fear, faces, and the human amygdala". In: *Current opinion in neurobiology* 18.2 (2008), pp. 166–172.

[7] J. Zhong. "Artificial Neural Models for Feedback Pathways for Sensorimotor Integration". In: (2015).

[8] J. Zhong et al. "A Hierarchical Emotion Regulated Sensorimotor Model: Case Studies". In: *The 5th International Conference on Data-Driven Control and Learning Systems*. 2016.

[9] J. Zhong and L. Canamero. "From continuous affective space to continuous expression space: Non-verbal behaviour recognition and generation". In: *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*. IEEE. 2014, pp. 75–80.

[10] H. T. Siegelmann and E. D. Sontag. "On the computational power of neural nets". In: *Journal of computer and system sciences* 50.1 (1995), pp. 132–150.

[11] J. Tani. "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes". In: *Neural Networks* 16.1 (2003), pp. 11–23.

[12] J. Tani and M. Ito. "Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment". In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 33.4 (2003), pp. 481–488.

[13] J. L. Elman. "Finding structure in time". In: *Cognitive science* 14.2 (1990), pp. 179–211.

[14] M. I. Jordan. "Serial order: A parallel distributed processing approach". In: *Advances in psychology* 121 (1997), pp. 471–495.

[15] Y. Sugita and J. Tani. "Learning semantic combinatoriality from the interaction between linguistic and behavioral processes". In: *Adaptive Behavior* 13.1 (2005), pp. 33–52.

[16] J. Zhong, A. Cangelosi, and S. Wermter. "Towards a self-organizing pre-symbolic neural model representing sensorimotor primitives". In: *Front. Behav. Neurosci.* 8 (2014), p. 22.

[17] J. Zhong, C. Weber, and S. Wermter. "Robot Trajectory Prediction and Recognition based on a Computational Mirror Neurons Model". In: *Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN 2011)*. Ed. by T. Honkela et al. Vol. 2. Espoo, Finland: Springer, 2011, pp. 333–340.

[18] J. Duchi, E. Hazan, and Y. Singer. "Adaptive subgradient methods for online learning and stochastic optimization". In: *Journal of Machine Learning Research* 12.Jul (2011), pp. 2121–2159.

[19] M. D. Zeiler. "ADADELTA: an adaptive learning rate method". In: *arXiv preprint arXiv:1212.5701* (2012).

[20] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu. "Advances in optimizing recurrent networks". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 8624–8628.

[21] R. Pascanu, T. Mikolov, and Y. Bengio. "On the difficulty of training recurrent neural networks." In: *ICML (3)* 28 (2013), pp. 1310–1318.