In: In Koprinkova-Hristova, P.D. et al. eds., Artificial Neural Networks - Methods and
Applications, SSBN 4, pp. 149-174, Springer International Publishing, 2014.

Analysing the Multiple Timescale Recurrent Neural Network for Embodied Language Understanding

Stefan Heinrich, Sven Magg, and Stefan Wermter

Abstract How the human brain understands natural language and how we can exploit this understanding for building intelligent grounded language systems is open research. Recently, researchers claimed that language is embodied in most – if not all – sensory and sensorimotor modalities and that the brain's architecture favours the emergence of language. In this chapter we investigate the characteristics of such an architecture and propose a model based on the Multiple Timescale Recurrent Neural Network, extended by embodied visual perception, and tested in a real world scenario. We show that such an architecture can learn the meaning of utterances with respect to visual perception and that it can produce verbal utterances that correctly describe previously unknown scenes. In addition we rigorously study the timescale mechanism (also known as hysteresis) and explore the impact of the architectural connectivity in the language acquisition task.

1 Introduction

Natural language is the cognitive capability that clearly distinguishes humans from other living beings and is often called the key to intelligence. Humans not only utter short sounds to indicate an intention, but also describe procedural activities or may even completely think in natural language [10, 16]. However, language processing in the human brain and the acquisition of language has not yet been fully understood at the level of neural architectures. While hypotheses and models for innateness and universal grammars dominated the last 60 years (see [14] for a review on generativism), new neuroscientific findings and computational approaches led to alternative views towards emergence and constructivism (see [2, 26] for reviews).

Stefan Heinrich · Sven Magg · Stefan Wermter
University of Hamburg, Department of Informatics, Knowledge Technology
Vogt-Kölln-Straße 30, D - 22527 Hamburg, Germany
e-mail: {heinrich,magg,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

With the tool sets of neural simulations and behavioural robotics we now can approach the following research question: what are the architectural characteristics for models of the human brain that favour the emergence of language?

1.1 Binding and Grounding in Computational Models

In the past researchers have suggested valuable models to explain the binding of language to experience or learned instances of certain roles, but also to ground language in embodied perception and action based on recent neuroscientific data and hypotheses. Recent computational models aimed at mimicking certain abstractions of circuits in the brain and tested them for instances of the binding and the grounding problem [21, 29].

To investigate systematicity in language processing, Frank studied empirically to what extent a neural architecture can bind learned words to novel roles (trained grammatical roles for which these words have not been trained) [17, 18]. For an *Echo State Network* (ESN) with an additional hidden layer a corpus of sentences was tested that stems from a small context free grammar, which allows to include recursions of relatives clauses. Compared to other *Recurrent Neural Network* (RNN) the ESN has a similar complexity in processing, but allows for easier training on the one hand and a more difficult in-depth analysis on the other hand. In the study it was found that language can be learned compositionally and that RNNs show strong systematicity, or in other words: generalisation for structural coarsely related sentences, both syntactically and semantically.

In various experiments Cangelosi et al. investigated the grounding of symbols in a computational model [5, 6, 7]. With the hypothesis that language can emerge from embodied interaction within an environment and a simultaneous exposure to words or “symbols”, a number of simulations were conducted. Firstly, stick-figure robots were supposed to perform actions with a number of proto-objects, for which they also perceived names. The study showed that the underlying neural feed-forward architecture can be trained to ground the label in the sensori-motor perception to produce a name for a perceived action or vice versa. Additionally, an analysis revealed that the architecture self-organised to a semantic representation in the hidden layer. Secondly, an iCub humanoid robot was set up to perform similar interaction tasks with increased complexity. In this study a similar neural architecture was tested, and it was shown that the labels for an object can be grounded in the visual perception. The robots in these approaches do not have full linguistic and compositional abilities, but can enrich their lexicon with simple mechanisms mimicking compositionality. These models are inspired by from research in developmental psychology and neuroscience to provide a better understanding of the emergence of complex cognitive and perceptual structures. Also, they provide a basis to test novel algorithms and methodologies for the development of effective interaction between humans and autonomous robotic systems. Both sets of studies emphasised the importance of integrating language and embodied perception.

In addition, early models captured the fusion of language and multi-modal perceptions or aimed at bridging the gap between formal linguistics and bio-inspired systems. For these approaches the idea is a certain abstraction of the environment and its representation in testing for language learning.

For instance, with the Cross-Modal Early Lexical Learning (CELL) framework, Roy and Pentland proposed a model of embodied word acquisition [40]. CELL is based on a multi-modal learning scheme, where semantic categories and object labels are learned simultaneously. Sequences of phonemes that are detected in a short time window are interpreted as words and associated to visual prototypes, represented by a histogram for the object's shape. The learning takes place in a semi-supervised fashion using a short-term memory for identifying the reoccurring pairs of acoustic and visual sensory data, which later are passed to a long-term representation of extracted audio-visual objects. In the experiment with data from caregiver-infant interactions it was shown that the system is able to pick up the ideal link of sounds forming a word (or in rare cases a onomatopoeic sound) to an object shape and thus associated a meaning to certain chains of phonemes. Although the model shows that the learning of language is much more effective, if the learning is grounded in visual perception, the study was constrained to the abstraction of words from input phonemes and the association of the words with shapes.

Furthermore, Rhode proposed a model for language comprehension and prediction based on an *Elman Recurrent Neural Network* (ERNN or more often called SRN for Simple RN) [38, 39]. The semantic part of the model was trained to abstract the meaning or "the message" of a sentence from a set of linguistic propositions, while the comprehension part of the network learned to extract this meaning from a sequence of words, which includes the distribution of the propositions. The network can be used also in the opposite direction in a way that it can predict the first word for a given meaning and can then predict the next words based on the feedback of the previous word and the meaning. The underlying claim of the model is that humans may learn to produce language based on the previously learned capability to formulate predictions as well as the simultaneous comprehension of language. In this architecture the RNN is used as a statistical tool that can predict a sequence based on a training with structured representation (predefined role binding) and does not attempt to capture embodied representations of the human cortex.

However, due to the vast complexity of language some models rely on well-understood Chomskyan formal theories, which are difficult to maintain in the light of recent neuroscientific findings, e.g. of non-infinite-recursive mechanisms and the evident involvement of various – if not all – functional areas in the human brain in language [35, 36]. A substantial number of studies indicate that the cognitive processes – including language processing – originate in multi-modal interactions with the environments and are encoded in terms of the overall goal involving all the relevant effectors [1, 4]. Other integrating or constructive models are constrained to single words, neglecting the temporal aspect of language, e.g. that both the representation on the level of speech sounds and the processing with a multi-time resolution are important [11, 24].

1.2 Language Acquisition in a Recurrent Neural Model

In a recent study Hinoshita et al. claimed that for human language acquisition just an "appropriate" architecture is sufficient and provided a model based on a *Multiple Timescale Recurrent Neural Network* (MTRNN) [25]. The RNN model learns language from continuous input of sentences composed of words and characters that stem from a small grammar. For the model no implicit information is provided on word segmentation and on roles or categories for words. Instead the input is modelled as streams of spike-like activities on character level. During training the architecture self-organises to the decomposition of the sentences hierarchically based on the explicit structure of the inputs and the specific characteristic of some layers. The authors found that the characteristics, e.g. the information processing on different timescales, indeed leads to a hierarchical decomposition of the sentences in a way that certain character orders forms words and certain word orders forms the sentences. Although in the study the model was reproducing learned symbolic sentences quite well, generalisation was not possible to test, because the generation of sentences was initiated by the internal state of some neurons, which had to be trained individually for every sentence.

In this chapter we incorporate embodied perception based on real world data in an MTRNN model and show that such a novel system is able to generalise to completely new situations by recomposing learned elements, and also self-organises towards the meaning of the learned verbal utterances. For both, the verbal utterances and the perception, we employ representations that are biologically inspired and avoid to provide structural information on the language. To acquire real world data and test the model in a language acquisition task in an embodied and situated agent, we employ an humanoid robot NAO that is supposed to learn language in interaction with different shaped and coloured objects. This work is an extension of the previous ICANN contribution [23], and in addition includes in-depth analyses of the roles of the network connectivity and the timescale concept in language acquisition.

1.3 Chapter Organisation

This chapter is organised as follows: With the related work in mind from the introduction, in Section 2 we will provide a detailed description of our model of an MTRNN extended by embodied perception. We include a complete formalisation to ease re-implementation. In Section 3 we will specify the scenario of the language learning robot as well as a complete description of the used representations of verbal utterances and embodied perception and the preceding encoding mechanisms. Then, in Section 4, follows our evaluation and the analysis. We report on the studies for generalisation capabilities as well as for the network behaviour and the impact of some key characteristics. Finally, in Section 5 we will discuss our findings, conclusions, and future prospects.

2 Extended MTRNN Model

To test for plausible characteristics for the semantic processing of verbal utterances, we incorporate both hypotheses into one model: a) speech is processed on a multiple-time resolution [24], and b) semantic circuits are involved in the processing of language [36]. We model the neural circuit as an RNN to achieve a reasonable biological plausibility, but also be able to analyse the networks behaviour on cortex level. More precisely, for our proposed model we employ the MTRNN to process verbal utterances over time [46], extended by several feed-forward layers to integrate embodied perceptions during the processing of utterances.

The MTRNN part is composed of an *Input- and Output* layer (IO) and two context layers called *Context fast* (Cf) and *Context slow* (Cs). Our extension part consists of an *Embodied Input* layer (EI), an *Embodied Fusion* layer (EF), and an *Embodied Controlling* layer (EC). Fig. 1 provides an overview of our architecture.

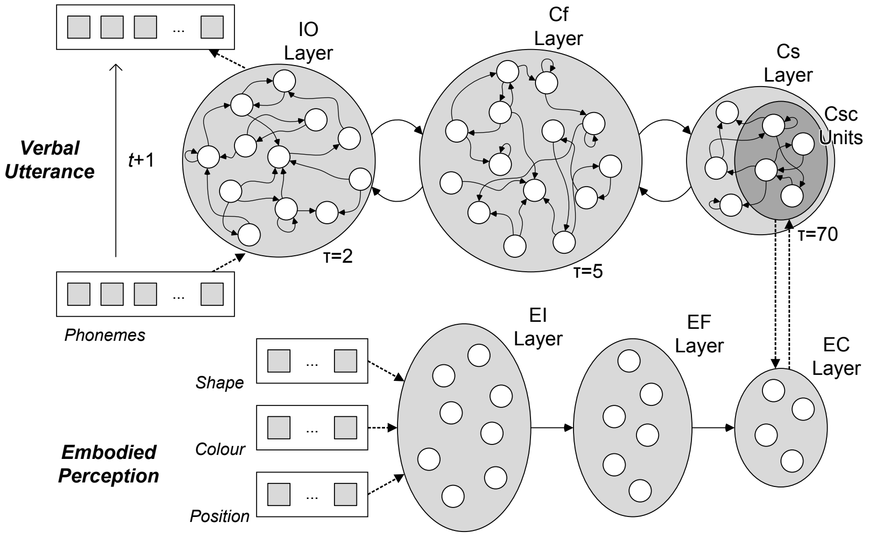


Fig. 1 Architecture of a Multiple Timescale Recurrent Neural Network extended by embodied perception from the scene. A sequence of phonemes (utterance) is processed over time, while the perceived embodied and situated information is constantly present.

During learning the MTRNN layers self-organise to the decomposition of a semantic meaning into a verbal utterance on phoneme level over time, while the feed-forward layers associate the meaning with the embodied perception. For production of utterances the feed-forward layers have the role of abstracting the meaning from the embodied input, whereas the MTRNN functions as a predictor of the next phoneme based on the context information and the previous sequence of phonemes.

2.1 RNN Schematics

The MTRNN is composed of an *Input- and Output* layer that continuously produces an output from an input and from recurrent connections, as well as of an arbitrary number of coupled context layers. In general, the MTRNN is an extended *Elman Recurrent Neural Network* (ERNN) on the one hand and a special case of the *Plausibility Recurrent Neural Network* (PRNN) on the other hand [15, 43].

In contrast to the ERNN, the MTRNN allows for full connectivity of neurons to all neurons of the same and of adjacent layers. Also, all neurons process information based on incoming connections as well as their previous internal state. In principle the neurons maintain a fraction of previous information and process new information slower, based on a lagging parameter [33]. We call the parameter *hysteresis* φ, if we denote, which fraction of new information is taken into account (where $1-\varphi$ denotes, which fraction of old information is kept), or *timescale* τ, if we denote to which magnitude new information is processed slower and we can relate:

$$\varphi = \frac{1}{\tau} \quad . \tag{1}$$

Compared with the PRNN the MTRNN restricts this concept of hysteresis to an increasing slowness from the first to the last layer and also restricts the architecture to one horizontal set of layers only. A schematic comparison of the network architectures is shown in Fig. 2.

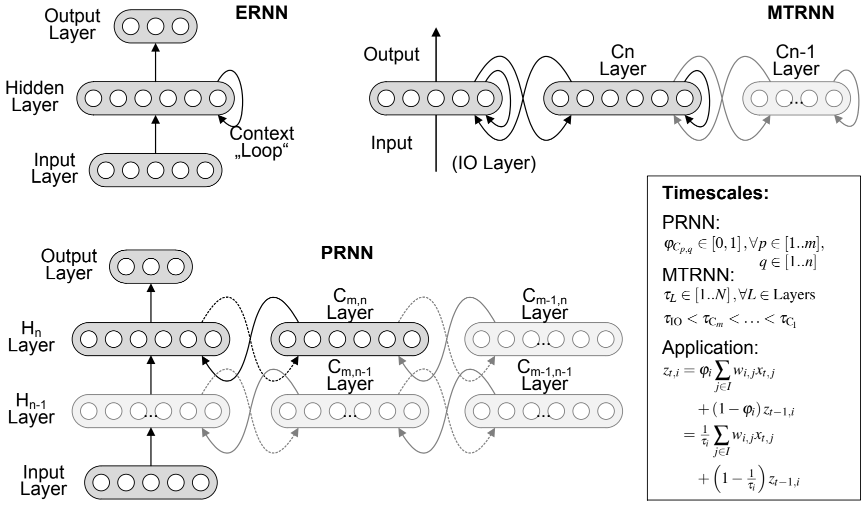

Output Layer
ERNN
Hidden Layer
Context „Loop“
Input Layer
MTRNN
Output
Input
(IO Layer)
Cn Layer
Cn-1 Layer
Output Layer
PRNN
H_n Layer
$C_{m,n}$ Layer
$C_{m-1,n}$ Layer
H_{n-1} Layer
$C_{m,n-1}$ Layer
$C_{m-1,n-1}$ Layer
Input Layer
Timescales:
PRNN:
$\varphi_{C_{p,q}} \in [0,1], \forall p \in [1..m], q \in [1..n]$
MTRNN:
$\tau_L \in [1..N], \forall L \in \text{Layers}$
$\tau_{\text{IO}} < \tau_{\text{C}_m} < \ldots < \tau_{\text{C}_1}$
Application:
$z_{t,i} = \varphi_i \sum_{j \in I} w_{i,j} x_{t,j} + (1-\varphi_i) z_{t-1,i} = \frac{1}{\tau_i} \sum_{j \in I} w_{i,j} x_{t,j} + \left(1 - \frac{1}{\tau_i}\right) z_{t-1,i}$

Fig. 2 Schematic comparison of Elman Recurrent Neural Network, Plausibility Recurrent Neural Network, and Multiple Timescale Recurrent Neural Network. The concept of timescales τ in the MTRNN is equivalent to the concept of hysteresis φ in the PRNN, but is restricted to increasing values τ for increasing numbers of layers to the right.

Based on the idea of introducing a parametric bias to the network [42], some neurons in the slowest context layer Cs of the MTRNN are also designated as *Context controlling units* (Csc units). The internal state of these units at the initial time step ($t = 0$) can be stored during training and can be used to initialise the generation of a meaningful sequence. By modulating these internal states, different other sequences can be generated.

2.2 Information Processing

In the MTRNN information is processed continuously with a constant firing rate as a sequence of T discrete-time steps. A sequence $s \in S$ is represented as a discretised flow of activations of the neurons in the IO layer $i \in I_{\mathrm{IO}}$. The input activation x of a neuron $i \in I_{\mathrm{all}} = I_{\mathrm{IO}} \cup I_{\mathrm{Cf}} \cup I_{\mathrm{Cs}}$ at time step t is calculated as:

$$x_{t,i} = \begin{cases} 0 & \text{iff } t = 0 \\ (1-\alpha) y_{t-1,i} + (\alpha) d_{t-1,i} & \text{iff } t \geq 1 \wedge i \in I_{\mathrm{IO}} \\ y_{t-1,i} & \text{iff } t \geq 1 \wedge i \notin I_{\mathrm{IO}} \end{cases} , \tag{2}$$

where $\alpha \in]0,1[$ is the feedback rate reflecting a *teacher forcing* (TF) signal of the desired output y^* to the input together with the generated output y of the last time step (see [12, 45]). The feedback is only given during training the MTRNN.

The internal state z of a neuron i at time step t is determined by:

$$z_{t,i} = \begin{cases} 0 & \text{iff } t = 0 \wedge i \notin I_{\mathrm{Csc}} \\ c_{0,i} & \text{iff } t = 0 \wedge i \in I_{\mathrm{Csc}} \\ \left(1 - \dfrac{1}{\tau_i}\right) z_{t-1,i} + \dfrac{1}{\tau_i} \sum\limits_{j \in I_{\mathrm{all}}} w_{i,j} x_{t,j} & \text{otherwise} \end{cases} , \tag{3}$$

where $c_{0,i}$ is the initial internal state of the Csc units $i \in I_{\mathrm{Csc}} \subset I_{\mathrm{Cs}}$ (at time step 0), and $w_{i,j}$ are the weights from the jth to the ith neuron. In our model the MTRNN is specified by timescale values of $\tau = 2$, $\tau = 5$, and $\tau = 70$ for the IO, Cf, and Cs layers respectively, based on previous work [25, 46] and experiments in this study (see Section 4.3), indicating that these settings work well for language learning scenarios.

The output (activation value) y of a neuron i at time step t is defined by:

$$y_{t,i} = \begin{cases} \dfrac{\exp(z_{t,i} + b_i)}{\sum\limits_{j \in I_{\mathrm{IO}}} \exp(z_{t,j} + b_j)} & \text{iff } i \in I_{\mathrm{IO}} \\ \mathrm{sig}(z_{t,i} + b_i) & \text{iff } i \notin I_{\mathrm{IO}} \end{cases} , \tag{4}$$

where b_i is the bias of neuron i. For the IO layer we employ a soft-max function due to the problem-specific representation, while for the neurons in the remaining layers we use a sigmoidal transfer function:

$$\mathrm{sig}\left(z_{t,i}+b_i\right)=\frac{1+2\kappa_a}{1+\exp\left(-\kappa_b\left(z_{t,i}+b_i\right)\right)}-\kappa_a \quad , \tag{5}$$

which is a logistic function with parameters κ_a for range and κ_b for slope. For our model we modulated the function with $\kappa_a = 0.35795$ and $\kappa_b = 0.92$ to capture the characteristics of the synchronic transfer function that has been proposed by LeCun for faster convergence in association tasks [32]. Although the choice of transfer functions is vast and well studied, more complex, but also more flexible functions exist [13]. In particular, the asynchonic tangens-hyperbolicus function and its synchronic equivalent – the logistic function – are used more often. We favour the use of a simple function since we are more interested in the general characteristics and also the comparability of the model and less in an optimal solution.

For the feed-forward extension the information processing follows analogously. Input perception p_s for a sequence is constantly present, while the output constitutes the activity for the initial internal states $c_0(s)$ of the Csc units for the sequence s. For all layers of the extension we use the same modified logistic transfer function.

2.3 Learning

During learning of the system the MTRNN is trained with verbal utterances, and self-organises the weights and also the internal state values of the Csc units. These self-organised values are then transferred to the EC layer and associated with the present embodied perception (EI layer). For training the MTRNN we use an adaptive variant of the *real-time backpropagation through time* (RTBPTT) algorithm [22, 45].

In the *forward pass* (FP) the error E is accumulating the error between the activation values (y) and the desired activation values (y^*) of the IO neurons at every time step as follows:

$$E(W)=\sum_t \sum_{i\in I_{\mathrm{IO}}} y^*_{t,i}\cdot\log\left(\frac{y^*_{t,i}}{y_{t,i}}\right) \quad , \tag{6}$$

where we use the *Kullback–Leibler divergence* as error function on IO neurons [31].

In the second step the partial derivatives of the calculated activation (y) and the desired activation (y^*) are derivated in a *backward pass* (BP):

$$\frac{\partial E}{\partial z_{t,i}}=\begin{cases} y_{t,i}-y^*_{t,i}+\left(1-\frac{1}{\tau_i}\right)\frac{\partial E}{\partial z_{t+1,i}} & \text{iff } i\in I_{\mathrm{IO}} \\ \mathrm{sig}'\left(y_{t,i}\right)\sum\limits_{k\in I_{\mathrm{all}}}\frac{w_{k,i}}{\tau_k}\frac{\partial E}{\partial z_{t+1,k}}+\left(1-\frac{1}{\tau_i}\right)\frac{\partial E}{\partial z_{t+1,i}} & \text{otherwise}\end{cases} \quad , \tag{7}$$

where the gradients are 0 for the time step $T+1$. The derivative for the sigmoidal transfer function is calculated as follows:

$$\mathrm{sig}'(y_{t,i}) = \frac{\kappa_b}{1+2\kappa_a}\left(y_{t,i}+\kappa_a\right)\left(1-y_{t,i}+\kappa_a\right) \quad . \tag{8}$$

Finally, with the determined gradients the weights w and biases b are updated:

$$w_{i,j}^{n+1} = w_{i,j} - \eta_{i,j}\frac{\partial E}{\partial w_{i,j}} = w_{i,j} - \eta_{i,j}\sum_t \frac{1}{\tau_i} x_{t,j}\frac{\partial E}{\partial z_{t,i}} \quad , \tag{9}$$

$$b_i^{n+1} = b_i - \beta_i\frac{\partial E}{\partial b_i} = b_i - \beta_i\sum_t \frac{1}{\tau_i}\frac{\partial E}{\partial z_{t,i}} \quad , \tag{10}$$

where the partial derivatives for w and b are the sums of weight and bias changes over the whole sequence respectively, and η and β denote the learning rates for the weight and bias changes. Here, we use individual learning rates for all weights and biases because we adapt them with respect to the gradient as described below.

The initial internal states $c_{0,i}$ of the Csc units define the behaviour of the network and are also updated as follows:

$$c_{0,i}^{n+1} = c_{0,i} - \zeta_i\frac{\partial E}{\partial c_{0,i}} = c_{0,i} - \zeta_i\frac{1}{\tau_i}\frac{\partial E}{\partial z_{0,i}} \quad \text{iff } i \in I_{\mathrm{Csc}} \quad , \tag{11}$$

where ζ_i denotes the learning rates for the initial internal state changes.

For training the association of the EC layer with the EI layer, we apply the *least mean square* (LMS) rule as error function [44]:

$$E(W) = \frac{1}{2}\sum_{i\in I_{\mathrm{EI}}}\left(y_i - y_i^*\right)^2 \quad , \tag{12}$$

$$\frac{\partial E}{\partial z_i} = \begin{cases} (y_i - y_i^*)\mathrm{sig}'(y_{t,i}) & \text{iff } i \in I_{\mathrm{EC}} \\ \mathrm{sig}'(y_i)\sum\limits_{k\in I_{\mathrm{EC}}} w_{k,i}\dfrac{\partial E}{\partial z_k} & \text{iff } i \in I_{\mathrm{EF}} \end{cases} \quad , \tag{13}$$

where the desired output y^* corresponds to the activity derived from the $c_0$values:

$$y_i^* = \mathrm{sig}(c_i) \quad \forall i \in I_{\mathrm{EC}} \quad . \tag{14}$$

The adaptation of the weights and biases follows analogously.

2.4 Adaptive Learning Rates

In our approach the learning rates η, β, and ζ are adaptive, based on the local gradient information inspired by the *Resilient Propagation* (RPROP) algorithm [37]. In contrast to the original RPROP, learning rates are adapted and multiplied directly with the partial derivatives instead of only using the sign of the partial derivatives, to determine the change of the learning step:

$$\eta_{i,j}^{n} = \begin{cases} \min\left(\eta_{i,j}^{n-1}\xi^{+}, \eta_{\max}\right) & \text{iff } \left(\frac{\partial E}{\partial w_{i,j}} \cdot \frac{\partial E}{\partial w_{i,j}}^{n-1}\right) > 0 \\ \max\left(\eta_{i,j}^{n-1}\xi^{-}, \eta_{\min}\right) & \text{iff } \left(\frac{\partial E}{\partial w_{i,j}} \cdot \frac{\partial E}{\partial w_{i,j}}^{n-1}\right) < 0 \\ \eta_{i,j}^{n-1} & \text{otherwise} \end{cases} , \tag{15}$$

$$\beta_{i}^{n} = \begin{cases} \min\left(\beta_{i}^{n-1}\xi^{+}, \eta_{\max}\right) & \text{iff } \left(\frac{\partial E}{\partial b_{i}} \cdot \frac{\partial E}{\partial b_{i}}^{n-1}\right) > 0 \\ \max\left(\beta_{i}^{n-1}\xi^{-}, \eta_{\min}\right) & \text{iff } \left(\frac{\partial E}{\partial b_{i}} \cdot \frac{\partial E}{\partial b_{i}}^{n-1}\right) < 0 \\ \beta_{i}^{n-1} & \text{otherwise} \end{cases} , \tag{16}$$

where $\xi^{+} \in]1, \infty]$ and $\xi^{-} \in]0, 1[$ are the increasing or decreasing factors respectively and $\eta_{\max} > \eta_{\min}$ are upper and lower bounds for both learning rates η and β. If the partial derivative of the current epoch n is pointing to the same direction as in the former epoch $n-1$, then the learning rate is increased. If the direction of the partial derivative is pointing to the other direction, then the minimum has been missed and the learning rate is decreased.

For the update of the initial internal states $c_{0,i}$ the learning rates ζ are adapted proportionally to the average learning rates η of all weights that are connected with unit i and neurons of the same (Cs) and the adjacent (Cf) layer:

$$\zeta_{i} \propto \frac{1}{|I_{\mathrm{Cf}}| + |I_{\mathrm{Cs}}|} \sum_{j \in (I_{\mathrm{Cf}} \cup I_{\mathrm{Cs}})} \eta_{i,j} \quad . \tag{17}$$

Since the update of the $c_{0,i}$ depends on the same partial derivatives (time step 0) as the weights, we do not need additional parameters in this adaptive mechanism.

2.5 Production

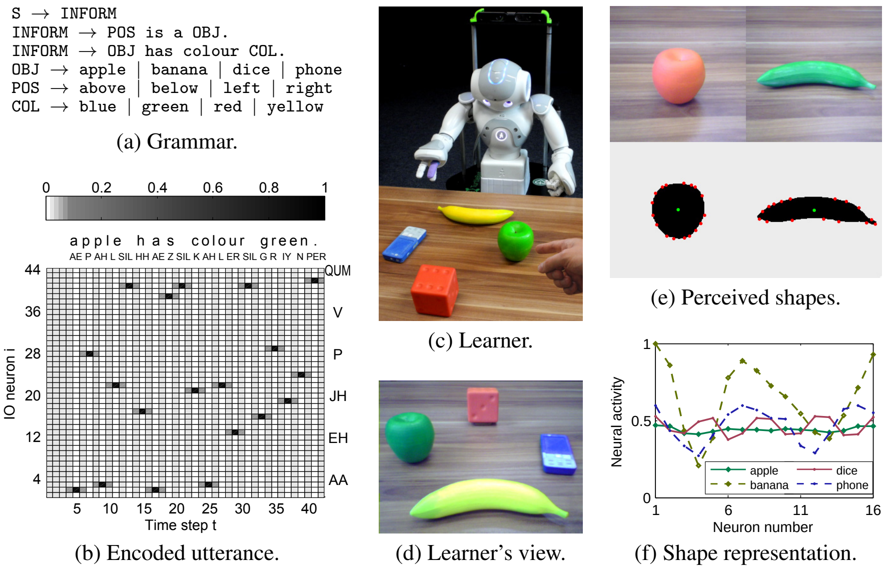
During testing the system approximates EC values from the embodied perception input at the EI layer. From the EC values the corresponding values of Csc units are calculates using the inverse of Eq. 14, which in turn initiate the generation of a corresponding verbal utterance. These processing steps are done in a single set of computation – no additional training or adaptation is necessary.

3 Scenario

Our scenario for this model is the interaction between a human teacher and a robotic learner, which is supposed to learn language from scratch by grounding utterances in its embodied experience, but also is supposed to use its learned language to describe novel situations. We believe it is important to test the learning in a real environment to face the influence of natural noise and uncertainty of perception.

The robot is placed in a scene and receives an utterance from the teacher, who describes the scene, e.g. "THE APPLE HAS COLOUR GREEN". The system should learn, in a self-organised way, how to bind the visual scene information with this verbal expression to be able to describe another scene like "THE BANANA HAS COLOUR GREEN" correctly. The focus of this study is on generalisation using possibly learned components.

To control our setup, all verbal utterances stem from a small symbolic grammar as presented in Fig. 3a. However, every symbolic sentence is transformed into a phonetic utterance based on phonemes from the ARPAbet and four additional signs to express pauses and intonations in propositions, exclamations, and questions: $A = \{\text{'AA'}, \ldots, \text{'ZH'}\} \cup \{\text{'SIL'}, \text{'PER'}, \text{'EXM'}, \text{'QUM'}\}$, with size $|A| = 44$.



```
S → INFORM
INFORM → POS is a OBJ.
INFORM → OBJ has colour COL.
OBJ → apple | banana | dice | phone
POS → above | below | left | right
COL → blue | green | red | yellow
```

(a) Grammar.

(b) Encoded utterance.

(c) Learner.

(d) Learner's view.

(e) Perceived shapes.

(f) Shape representation.

Fig. 3 Representations and scenario of language learning in human-robot interaction.

3.1 Utterance Encoding

To encode an utterance $u = (p_1, \dots, p_{|u|})$ into neural activation over time, we adapted the encoding scheme suggested by Hinoshita et al. [25], but we use a phoneme-based instead of a symbol-based representation: The occurrence of a phoneme p_k is represented by a spike-like neural activity of a specific neuron at relative time step r. In addition, some activity is spread backward in time (rising phase) and some activity is spread forward in time (falling phase) represented as a Gaussian over the interval $[-\omega/2, \dots, -1, 0, +1, \dots, \omega/2]$. On absolute course of time t the peaks mimic priming effects in articulatory phonetic processing. For example the previous occurrence of the phoneme "P" could be often related to the occurrence of the phoneme "AH" leading to an excitation of the respective neuron for "AH", when the neuron for "P" was activated.

All activities of spike-like peaks are normalised by a soft-max function for every absolute time step t over the set of input neurons. A sketch of the utterance encoding is shown in Fig. 4.

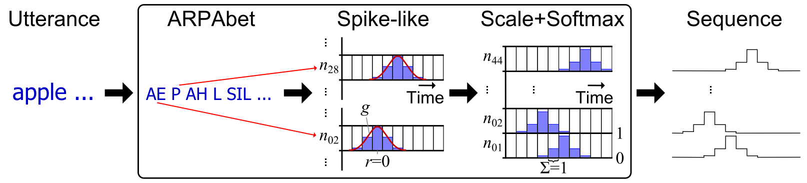


Fig. 4 Schematic process of utterance encoding. The input is a symbolic sentence, while the output is the neural activity over N neurons times T time steps.

The Gaussian g for p_k is defined by:

$$g_{k,r,i} = \begin{cases} \exp\left(\frac{-r^2}{2\sigma^2}\right) & \text{iff } p_k = A_i \\ 0 & \text{otherwise} \end{cases} \quad , \tag{18}$$

where $r = 0$ is the mean and σ the filter sharpness factor. A peak occurs for the neuron $i \in I_{\mathrm{IO}}$ with $|I_{\mathrm{IO}}| = |A|$, if the phoneme p_k is equal to the ith phoneme in the phoneme alphabet A. From the spike-like activities the internal state z of a neuron i at time step t is determined by:

$$z_{t,i} = \begin{cases} \lambda \cdot \max\left(g_{k=1\dots|u|,r=-\omega/2\dots\omega/2,i}\right) & \text{iff } t = \mu + k\upsilon + r \\ 0 & \text{otherwise} \end{cases} \quad , \tag{19}$$

where ω is the filter width, μ is a head margin to put some noise to the start of the sequence, υ is the interval between two phonemes, and λ is a scaling factor for the neuron's activity y^*.

The scaling factor depends on the number of IO neurons and scales the activity to $d \in]0, 0.9]$ for the specified soft-max function:

$$\lambda = \ln\left(\frac{0.9}{1.0 - 0.9}\left(|I_{\mathrm{IO}}| - 1\right)\right) \quad , \tag{20}$$

$$y^{*}_{t,i} = \frac{\exp\left(z_{t,i}\right)}{\sum\limits_{j \in I_{\mathrm{IO}}} \exp\left(z_{t,j}\right)} \quad . \tag{21}$$

For our scenario we set the constants to $\mu = 4$, $\omega = 4$, $\sigma^2 = 0.3$, and $\upsilon = 2$. The ideal neural activation for an encoded sample utterance is visualised in Fig. 3b.

3.2 *Visual Perception Encoding*

To encode the visual shape perception into sustained neural activity, we aim at capturing a representation that is biologically plausible, but on a level of abstraction of shapes as found in the *posterior infero-temporal* (PIT)/V4 area [34]. On an image taken by the NAO robot we employ the mean shift algorithm for segmentation [9], and the Canny edge detection as well as the contour finder for object discrimination [8, 41]. Subsequently, we calculate the centre of mass and 16 distances to salient points around the contour. Finally, we scale the distances by the square root of the object's area and order them clockwise – starting with the largest – to determine the characteristic shape, which is scale- and rotation-invariant. Fig. 3e provides two example results of this process, and Fig. 3f visualises typical characteristics for the employed object shapes (scaled to $[0, 1]$). Encoding of the perceived colour is realised by averaging the three R, G, and B values of the shape, while the perceived position is encoded by the two values of the centroid coordinate in the field of view. A sketch of the visual perception encoding is shown in Fig. 5.

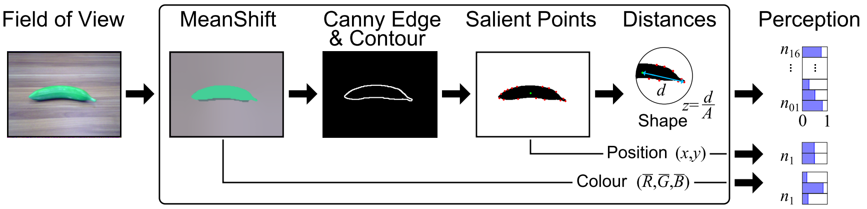


Fig. 5 Schematic process of visual perception encoding. The input is a single frame taken by the NAO camera, while the output is the neural activity over N neurons, with N is the sum over shape + colour + position features.

4 Evaluation and Analysis

To understand the dynamics of the architecture in this study, we are interested in evaluating the generalisation capabilities, and the role of some key characteristics like connectivity and timescales. We also aim at analysing the network behaviour in generating utterances for known as well as for novel scenes.

To test and analyse our model, we collected a data set consisting of all possible scenes and their respective verbal description. From the grammar we obtained 32 different combinations, which we set up as scenes and in turn used for collecting different examples. The corresponding verbal utterances were reasonably complex sequences with a length of 32 to 46 time steps (compare Fig. 3b). Subsequently, we ran a series of experiments, for which we carefully, but randomly divided the data into a training set and a test set (50:50) – making sure that every scene is included only in one of these sets – and trained ten randomly initialised systems. For every setup we repeated this process ten times with different distributions of data in training and test set to arrive at 100 runs for analysis. Compared to the previous ICANN contribution [23], the results are based on twice the number of runs per setup and per experiment. The parameters of the network and the meta-parameters were mostly chosen based on the experience in [22] and [25] and are detailed in Tab 1. The number of neurons in the input layers $|I_{\mathrm{IO}}|$ and $|I_{\mathrm{EC}}|$ are given by the input representations. The size of EC depends on and is equal to the size of Csc, which we determined with $|I_{\mathrm{Csc}}| = \lceil |I_{\mathrm{Cs}}|/2 \rceil$. As the termination criteria for the learning, we used a maximum number of epochs with $\theta = 50,000$ and minimal average errors on the IO and EI layers with $\varepsilon_{\mathrm{IO}} = 5.0 \times 10^{-4}$ and $\varepsilon_{\mathrm{EI}} = 5.0 \times 10^{-6}$. We favoured the use of fixed termination criteria over the use of a validation set to allow for comparisons on the meta-parameters.

Table 1 Standard parameter settings for evaluation.

Param.	Description	Value	Param.	Description	Value
$\|I_{\mathrm{IO}}\|$	Number of IO neurons	44	τ_{IO}	Timescale of IO neurons	2
$\|I_{\mathrm{Cf}}\|$	Number of Cf neurons	80	τ_{Cf}	Timescale of Cf neurons	5
$\|I_{\mathrm{Cs}}\|$	Number of Cs neurons	23	τ_{Cs}	Timescale of Cs neurons	70
$\|I_{\mathrm{Csc}}\|$	Number of Csc neurons	12	α	Teacher forcing	0.1
$\|I_{\mathrm{EC}}\|$	Number of EC neurons	12	η_{max}	Maximal learning rate	1.0
$\|I_{\mathrm{EF}}\|$	Number of EF neurons	16	η_{min}	Minimal learning rate	1.0×10^{-6}
$\|I_{\mathrm{EI}}\|$	Number of EI neurons	21	ξ^{+}	Increasing factor	1.01
$\mathbf{W}^0$	Initial weights range	±0.025	ξ^{-}	Decreasing factor	0.96
$\mathbf{C}_0^0$	Initial Csc values range	±0.01	η^0, β^0, ζ^0	Initial learning rates	0.05

4.1 Generalisation

To be able to compare the generalisation capabilities, we use the standard measure F_1-score determined by precision and recall, and defined as follows:

$$p_{\text{precision}} = \frac{tp}{tp+fp} \quad , \quad p_{\text{recall}} = \frac{tp}{tp+fn} \quad ,$$

$$F_1\text{-score} = 2 \cdot \frac{p_{\text{precision}} \cdot p_{\text{recall}}}{p_{\text{precision}} + p_{\text{recall}}} \quad , \tag{22}$$

where we specify all correct and matching utterances as tp (true positives), all correct, but not matching utterances as fp (false positives), and strictly all incorrect utterances as fn (false negatives).

Table 2 Parameter variation in the generalisation experiment.

Dimension	Parameter	Description	Values
1	$(\lvert I_{\text{Cf}}\rvert, \lvert I_{\text{Cs}}\rvert)$	Number of Cf, Cs neurons	$\{(40,11),(80,23),(160,47)\}$
2	$\lvert I_{\text{EF}}\rvert$	Number of EF neurons	$\{8,16,24\}$

The results in Tab. 3 show that the system can be trained perfectly in most cases, and also produces correct utterances for new scenes on a moderate level: For a suitable parameter setting, networks reach an F_1-score of up to 1.0 on the training set and 0.545 on the test set, with an average over all random seeds of 0.999 on the training set and 0.185 on the test set. From the chart in Fig. 6 for the same results for the test set only, we can learn that the size of the network dimension is important for ideal generalisation capabilities. The F_1-score on utterance level is clearly stricter than on word or on phoneme level, but we aim at evaluating, if the complete meaning of the scene was uttered correctly.

Table 3 Comparison of F_1-score for different network dimensions.

$\lvert Cf\rvert/\lvert Cs\rvert$	40/11	40/11	40/11	80/23	80/23	80/23	160/47	160/47	160/47
$\lvert EF\rvert$	8	16	24	8	16	24	8	16	24
training set best	1.000	1.000	1.000	1.000	1.000	**1.000**	1.000	1.000	1.000
test set best	0.316	0.316	0.316	0.476	0.476	**0.545**	0.476	0.400	0.400
training set best avg *	1.000	1.000	1.000	1.000	1.000	**1.000**	1.000	1.000	1.000
test set best avg *	0.200	0.229	0.207	0.322	0.333	**0.336**	0.277	0.254	0.264
training set average	0.913	0.927	0.928	0.948	0.999	**0.999**	0.988	0.998	0.996
test set average	0.068	0.072	0.076	0.133	0.177	**0.185**	0.098	0.119	0.115

* Averaged over all best networks of all data set distributions.

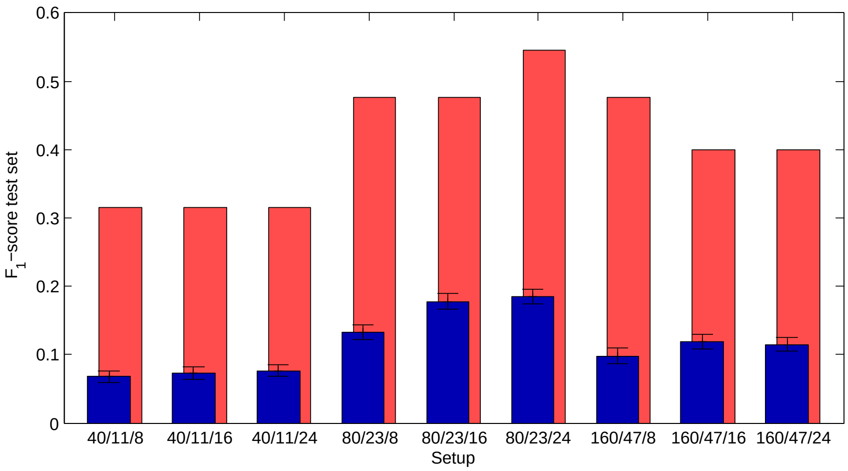


Fig. 6 Comparison of the F_1-score on the test set for the generalisation experiment. The dark/blue bars and the error bars present the average F_1-score and the standard error of means respectively, while the bright/red bars show the F_1-score of the best network for the respective setup.

Note that due to the random selection the system had to describe a scene in several cases, for which it had not seen any aspect (shape, colour, or position) before. This was intended to keep the scenario realistic and observe the effects.

In experiment we observed for incorrect utterances three types of errors: a) Minor substitution errors in terms of a single wrong phoneme or a pause that was too long ("SIL SIL" instead of "SIL"), b) word confusion errors, and c) phoneme chains without any meaning. Tab. 4 provides example results for observed errors. Errors of type (a) occurred often for networks in which the MTRNN part did not converge well to small average errors. For errors of type (b) we only found very few instances, and in these cases found the confused word mostly in the end of the sentence. A reason for this error was not found in this experiment, but further experiments (compare Sec. 4.3) indicates a link to the timescale parameter. The type (c) error appeared often in cases in which the training set and the test set are structural very different, e.g. when the test scene consisted of unknown aspects, as described above.

Table 4 Examples for different correct and incorrect utterances for errors (a), (b), and (c). Incorrect phonemes are emphasised bold.

correct	B AH N AE N AH SIL HH AE Z SIL K AH L ER SIL B L UW PER
substitution error (a)	R AY T SIL IH Z SIL AH SIL ***B*** AY S PER
substitution error (a)	B IH L OW SIL IH Z ***SIL SIL*** AH SIL AE P AH L PER
word confusion (b)	B AH N AE N AH SIL HH AE Z SIL K AH L ER SIL ***G R IY N*** PER
phoneme babbling (c)	***AE P AH AE SIL AH SIL AE AE Z K P L ER EH*** R EH D . . .

4.2 The Role of Connectivity and Pathways

During training of the system we found that the connection weights from the Cs to the Cf layer as well as from the Cf to the IO layer converged towards zero in many cases. This means that the highly dynamic networks organised themselves towards a directed flow of information from the context to the phonetic output instead of a mutual exchange of information.

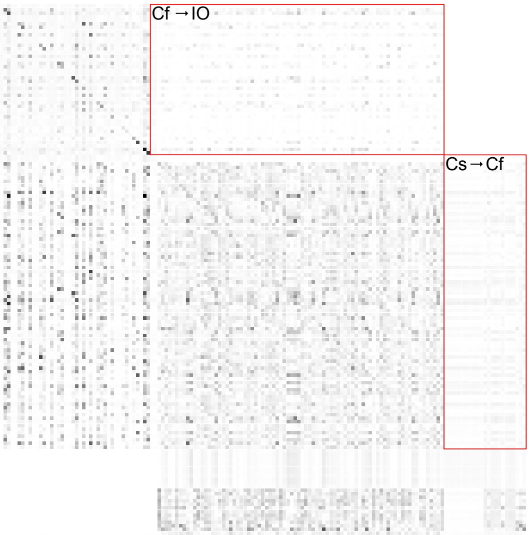
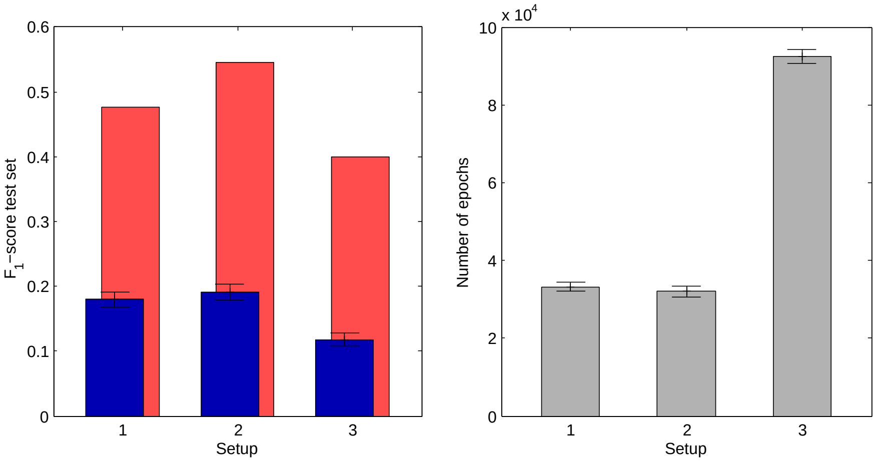


Fig. 7 Connectivity for an example network trained with the standard parameters and visualised as a Hinton diagram, where a square represents a connection weight from a neuron (horizontal dimension) to another neuron (vertical dimension). The diagram has been modified in a way that the strong connections are shown towards black (omitting the sign of the weights to increase readability), while weak connections are shown towards white.

To test the hypothesis that the MTRNN architecture might already be more complex than necessary and should be studied with less initial connectivity, we set up an experiment with modified connectivity and compared the following setups:

1. No modification (baseline): all neurons of a layer are connected to all neurons of the same and of adjacent layers.
2. All neurons of a layer are connected to all neurons of the same and of adjacent layers, but the connection weights from Cs to Cf and from Cf to IO are initialised with 0.0 instead of ±0.025.
3. Connections from Cs to Cf and from Cf to IO are removed.

We trained the networks with the procedure and the standard parameters as described above (see Tab. 1), but increased for the training the maximum number of epochs to $\theta = 100,000$, to ease the comparison of the training effort for the modifications. The results presented in Fig. 8 show that on the test data the F_1-score is slightly, but not significantly higher for setup 2 compared to setup 1, whereas the F_1-score is significantly ($p < 0.001$) lower for setup 3 compared to setup 1. However, the training effort for setup 2 is a bit but significantly ($p < 0.01$) smaller, and for setup 3 vastly larger (significant, $p < 0.001$) than for setup 1.



(a) F_1-score on test set.

(b) Training effort.

Fig. 8 Comparison of generalisation capability and training effort for modifications of the MTRNN connectivity. For (a) the dark/blue bars represent the average F_1-score, while the bright/red bars show the F_1-score of the best network for the respective setup. The error bars denote the respective standard error of means.

Note that for setup 2 we do not expect a higher F_1-score compared to setup 1, since in the training process all weights self-organise with respect to the partial derivatives. However, the results indicate that the introduced "bias" of having low connectivity from Cs to Cf and from Cf to IO leveraged the training process and led to faster convergence. For setup 3 the results show that having no backward connectivity makes the language acquisition problem much harder, indicating that backward connections are indeed necessary.

In terms of types of errors for the incorrect utterances we did not find considerable differences between setup 2 and setup 1, but a larger number of substitution errors for setup 3 compared to setup 1.

4.3 The Role of the Timescale Parameter

In preliminary experiments we confirmed that simpler RNNs cannot reproduce the generalisation capabilities of the MTRNN on the language acquisition. We tested both the ERNN with additional *parametric bias* units attached to the hidden layers (RNNPB, for details see [42]), as well as the MTRNN architecture with no timescale mechanism. Although the networks could learn the training data to some extent, the generation of utterances for novel scenes led to meaningless phoneme babbling. Basically the networks did not self-organise to the decomposition of the training sequences, but to reproduce them in whole, and thus generalisation ability was not evident.

Table 5 Parameter variation in the timescale experiment.

Dimension	Parameter	Description	Values
1	τ_{Cf}	Timescale of Cf neurons	$\tau_{\mathrm{IO}} \cdot k, k \in \{1,2,3,4,5,6\}$
2	τ_{Cs}	Timescale of Cs neurons	$\tau_{\mathrm{Cf}} \cdot l, l \in \{2,6,10,14,18,22\}$

Because the concept of hysteresis or timescales seems crucial for the language acquisition task, we investigated the influence of the timescale parameter. In a rigorous experiment we systematically varied the combination of timescale values of the neurons in the Cf and in the Cs layer. More precicely we tested the 2-fold up to 6-fold of the timescale for Cf with respect to the timescale for IO (fixed to $\tau_{\mathrm{IO}} = 2$) and also the 2-fold up to 22-fold of the timescale for Cs with respect to the timescale for Cf. For every combination as shown in Tab. 5 we trained 100 networks in the procedure as described above and kept all other parameters fixed. In sum we tested for 36 combinations leading to 3600 trained networks. Since we are interested both in the influence on the convergence of the networks for the given data set as well as in the generalisation capabilities we define a mixed score:

$$\begin{aligned} F_{1,\mathrm{mixed}}\text{-score} = &(F_1\text{-score(training set average)} + F_1\text{-score(test set average)} \\ &+ F_1\text{-score(training set best avg)} + F_1\text{-score(test set best avg)})/4 \,. \end{aligned}$$

The result of the experiment is visualised in Fig. 9, where high (desired) scores are shown in red and low scores are shown in blue. From the map we can obtain that using increasing timescales for the different layers increases the score. However, the scores do not differ much on a certain plateau: Networks for timescale ratio $\tau_{\mathrm{Cf}}/\tau_{\mathrm{IO}}$ of 2 and $\tau_{\mathrm{Cs}}/\tau_{\mathrm{Cf}}$ of 6 or higher reached a score of > 0.6, but this score does not

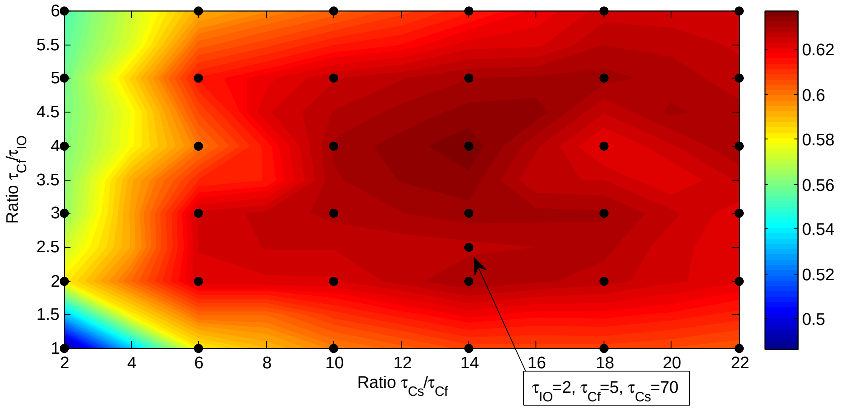


Fig. 9 $F_{1,\mathrm{mixed}}$-score for different combinations of timescale values of the Cf and the Cs neurons. Desired scores (high) are shown in red.

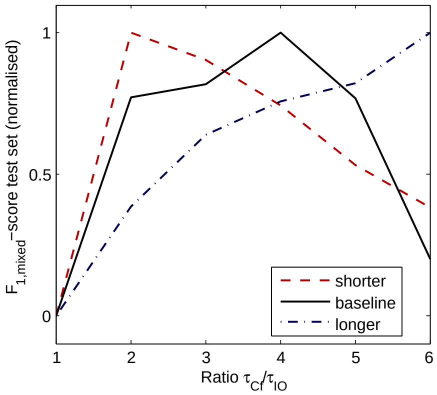
increase considerably for larger timescale ratios. In the results we can find some peaks e.g. for $\tau_{\mathrm{Cf}}/\tau_{\mathrm{IO}} = 4, \tau_{\mathrm{Cs}}/\tau_{\mathrm{Cf}} = 14$ ($\tau_{\mathrm{Cf}} = 8, \tau_{\mathrm{Cs}} = 112$), but the differences in the score values compared to e.g. the baseline ($\tau_{\mathrm{Cf}} = 5, \tau_{\mathrm{Cs}} = 70$) are not significant.

To investigate the differences in the results for networks with smaller timescale ratio (both $\tau_{\mathrm{Cs}}/\tau_{\mathrm{Cf}}$ and $\tau_{\mathrm{Cf}}/\tau_{\mathrm{IO}}$) we looked at the erroneous utterances that these networks produced on IO level. For both cases we noticed that the number of incorrect words as well as substitution errors in the end of the utterances occurred more often. The networks with smaller $\tau_{\mathrm{Cs}}/\tau_{\mathrm{Cf}}$ ratio generated syntactically correct but semantically not matching words more often for longer utterances, while networks with smaller $\tau_{\mathrm{Cf}}/\tau_{\mathrm{IO}}$ in general started to generate meaningless phoneme babbling more often. In summary, the results indicate that:

- The timescale for neurons in Cf is ideally of the length of the number of time steps for an average word length. For example the average word length in our scenario is 3.156 phonemes or 6.313 time steps, while the average inter-word distance (distance between the beginning of words including pauses) is 4.208 phonemes or 8.417 time steps.
- The timescale for neurons in Cs is ideally equal to or larger than the number of time steps of the longest sequence for a high score. However, very large timescales increase the training effort significantly. Recall, in our scenario we used sequences with length up to 46 time steps.

In an additional test we investigated the first indication further. We modified our corpus of utterances in a way that we changed all translations from words to phonemes to half the number of phonemes for the first setup and to double the number of phonemes for the second setup. Again, we trained the networks with different ratios $\tau_{\mathrm{Cf}}/\tau_{\mathrm{IO}}$ (compare Tab.5), while keeping the ratio fixed for the first setup with $\tau_{\mathrm{Cs}}/\tau_{\mathrm{Cf}} = 7$ and for the second setup with $\tau_{\mathrm{Cs}}/\tau_{\mathrm{Cf}} = 28$ due to the halved and doubled sequence lengths respectively. From the results in Fig 10 we can take that the estimate holds also for shorter and longer average word lengths as well.

Fig. 10 Comparison of $F_{1,\mathrm{mixed}}$-score for different timescale values over shortened and prolonged average word lengths. The timescale ratio are varied for $\tau_{\mathrm{Cf}}/\tau_{\mathrm{IO}}$ layer only. For the first setup, all words have been artificially halved in length (to a minimal length of one phoneme) and for the second setup, all words have been doubled in length. Results have been normalised for each setup to increase readability.



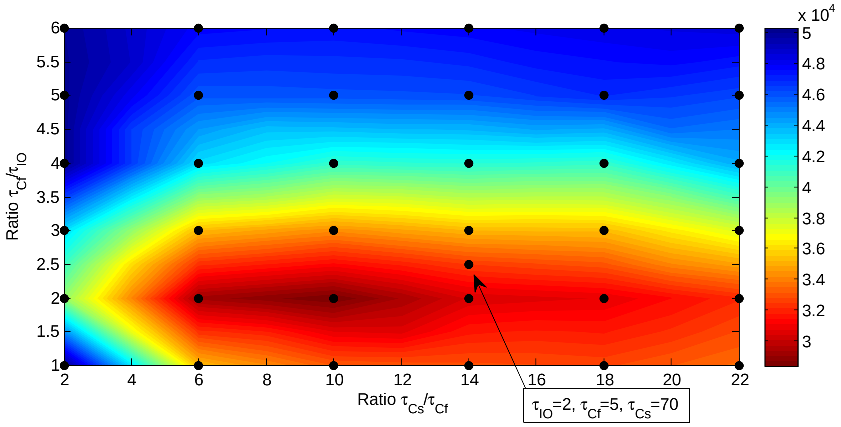


Fig. 11 Training effort (number of training epochs until termination) for different combinations of timescale values of the Cf and the Cs neurons. Desired (low) numbers are shown in red.

To also compare the difficulty to train the networks we looked at the average number of epochs until the training reached one of the termination criteria. The numbers are presented in Fig. 11, where low (desired) numbers are shown in red and high numbers are shown in blue.

For some combinations of timescale values around $\tau_{\text{Cf}}/\tau_{\text{IO}} = 2, \tau_{\text{Cs}}/\tau_{\text{Cf}} = 10$ ($\tau_{\text{Cf}} = 4, \tau_{\text{Cs}} = 40$) we found the smallest training effort, while for larger timescales both for Cf and Cs neurons the effort increases.

Combining both results, the scores on training and test data as well as the training effort can provide a rough estimate of good parameter values for practical applications. For example in Fig. 12 a possible combination is shown, where we weighted the proportion of the score five times over the proportion of the effort.

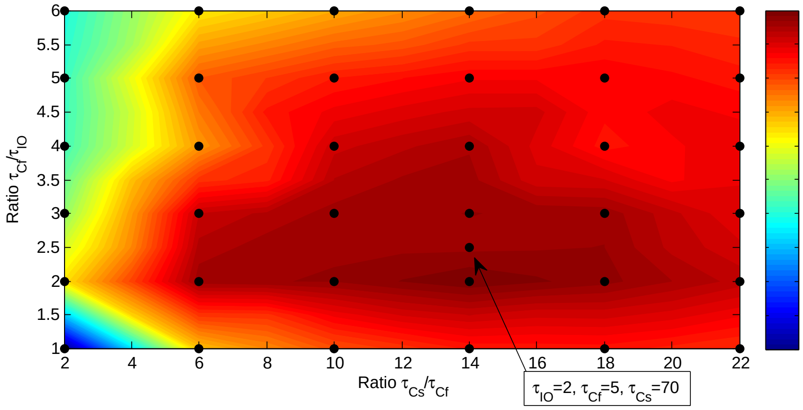
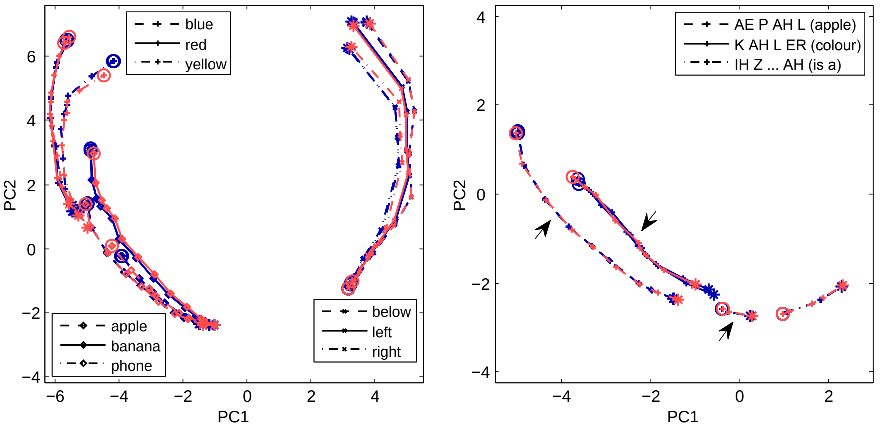


Fig. 12 Combination of $F_{1,\text{mixed}}$-score and training effort (5:1) for practical applications. Desired values are shown in red and may indicate good parameters.

4.4 Network Behaviour

To provide a better understanding of the system, we analysed the neural activity of the Cf layer for the trained networks. We aimed to test whether this layer had organised itself to represent the words in the utterances (compare [25]). Using *principle component analysis* (PCA), we reduced the dimensionality to visualise trajectories over time for specific words. The start and end point of the trajectory were defined as the first highest activity for the first phoneme and the last highest activity for the last phoneme of the word in the IO layer.

The results reveal several characteristics (see Fig. 13 for the trajectories of a typical network): Firstly, the neural activity in the Cf layer is nearly identical for the same words from trained utterances. Secondly, the same words from untrained utterances have a quite similar activity pattern. Thirdly, words of the same type (shape, colour, or position words) have very related activity patterns. From the data we can observe that the networks self-organise to patterns for words about shapes, colours, and positions. Fourthly, words with similar phonetic representations have different activities, if the type of the word is different. Low correlation was found of activity for phonetically similar but semantically different words.



(a) Words of similar type.

(b) Words with similar phonetics.

Fig. 13 Comparison of neural activation in the Cf layer for different words. The dimensionality has been reduced from $|I_{\text{Cf}}|$ to two dimensions (PC1 and PC2) and the beginning ($*$) as well as the end ($\circ$) of the words have been marked. The dark/blue lines represent words from utterances of the training set and the bright/red lines show words from utterances of the test set. Arrows indicate the same phoneme “AH”.

In addition, we found the tendency that the activation of a word primes the activation of other grammatically related words. In terms of trajectories it can be observed that the end point of the word “COLOUR” is close to the starting point of all colour words, and the end point of a position word is close to the starting point of “IS A ...” (compare Fig. 13a and b).

5 Discussion

The combination of visual perception and an architecture that includes different timescales in processing verbal sequences provides a system that self-organises towards the perceptual meaning of learned utterances in a real world scenario. Our experiments have shown that such a system apparently is able to understand verbal utterances and describe novel scenes with the correct corresponding verbal utterances. The analysis revealed that novel scenes are described by recomposing the correct words, which have been grounded in the perception of different shapes, colours, or positions.

Analysis of the errors for incorrect utterances revealed a) minor substitution errors, b) word confusion errors and c) phoneme babbling errors. In cases of type (a) listening humans would presumably consider this a normal inaccuracy and automatically correct the error. Errors of type (b) may indicate effects of the memorising capacity. For the trained networks we observed the word confusion error mostly in such cases, where timescale parameter values have been chosen sub-optimally. Neural activity in the Cs layer revealed that the networks seemingly could not produce the correct word, because they “forgot” the meaning of the scene at a certain time step and initiated the production of the most probable next word. Further research in the brain ’s information habituation could clarify this observation. Case (c) clearly shows that generalisation was sometimes difficult. It is open to clarify, whether this degree of difficulty is inherent, e.g. if the error rate is comparable to certain learning stages in young children during early language learning [30].

During training we also observed that connectivity plays an important role for the behaviour of the network. Although we found that the connection weights from the Cf to the Cs layer as well as from the IO to the Cf layer in many cases converged towards zero, we learned in additional experiments that we cannot leave out the backward connections. We found that a more directed flow of information from the context to the phonetic output was the result of the training, but a certain feedback seems to be important as well. In the light of neuroscientific evidence the directed information flow from the conceptual network (reflected by Cs) to the articulatory areas (reflected by IO) is plausible [24]. Also, in computational studies, researchers found that network architectures of biological plausible integrate-and-fire-neurons tend to form a mostly feed-forward structure out of initial randomly connected network for recurring input patterns [27, 28]. However, for many cortical regions of the human brain, for example in vision, it was also reported that certain proportions of backward (feedback) connections exist and play an important role [19, 20].

The examination of the timescale parameter revealed that the hysteresis mechanism (the timescales) is a key element for learning complex sequences like longer phoneme chains. Firstly, our results confirm that the hysteresis mechanism may be a required architectural characteristic that favours the emergence of language. Secondly, our results suggest that ideal parameter values are indeed problem-dependent, but less ideal values still lead to good performances. For the language acquisition problem we suggest to choose the average word-length in time steps as timescale value for the fast context layer (Cf) and to choose the maximal length of the sequences as timescale value for the slow context layer (Cs). These results can perhaps get transferred to other problems, where one uses the average length of the fast dynamics and maximal length of the slow dynamics as the respective values.

The dependency that we found between the size of the architecture and the size of the problem is less desirable, but in line with experience from associator networks [32]. Further investigations should include the consideration of architectures that are dynamic in connectivity as well as in size. In addition, architectures should be tested with more complex scenes and verbal descriptions, including interrelations of multiple objects and embodied experience of a broader set of real world situations.

5.1 Conclusion

In conclusion our study supports that the embodiment of language in perception and a hierarchical structure with hysteresis mechanisms in terms of different timescales are important aspects of an appropriate architecture for language. For such an architecture a feasible constraint can be our mostly feed-forward but compositional structure, also suggested for the (visual) cortex [19].

We believe it is very important to intensively study further the architectural characteristics that both favour or hinder the emergence of language. More specifically, in addition to learning “that” language can be grounded in perception and be bound to experience we need to learn “how”. For this we need to very carefully choose the assumptions that we are willing to make, to avoid to a) run into the *poverty of stimulus* (POS) pitfall [3] and b) research only into the parts instead of looking at the full system, which may result in more than the sum of the parts.

In the future we will further refine the architectural characteristics to identify the most important building blocks for natural language processing. The understanding of the brain’s architecture for language can explain the humans’ most important cognitive capability, but also can inform future software frameworks for service robots that should interact with and understand humans.

Acknowledgements The authors would like to thank A. Saleh and H. Vöcking for support with the real world data acquisition, J. Dávila-Chacón and C. Weber for critical and inspiring discussions, as well as J. Bauer and C. Mönter for valuable revisions of early versions of this chapter.

References

1. Barsalou, L.W.: Grounded cognition. Annual Review of Psychology 59, 617–645 (2008)
2. Behrens, H.: Usage-based and emergentist approaches to language acquisition. Linguistics 47(2), 383–411 (2009)
3. Berwick, R.C., Pietroski, P., Yankama, B., Chomsky, N.: Poverty of the stimulus revisited. Cognitive Science 35(7), 1207–1242 (2011)
4. Borghi, A.M., Gianelli, C., Scorolli, C.: Sentence comprehension: effectors and goals, self and others. An overview of experiments and implications for robotics. Frontiers in Neurorobotics 4(3), 8 (2010)
5. Cangelosi, A.: Grounding language in action and perception: From cognitive agents to humanoid robots. Physics of Life Reviews 7(2), 139–151 (2010)
6. Cangelosi, A., Riga, T.: An embodied model for sensorimotor grounding and grounding transfer: Experiments with epigenetic robots. Cognitive Science 30(4), 673–689 (2006)
7. Cangelosi, A., Tikhanoff, V., Fontanari, Fontanari, J.F., Hourdakis, E.: Integrating language and cognition: A cognitive robotics approach. Computational Intelligence Magazine 2(3), 65–70 (2007)
8. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), 679–698 (1986)
9. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
10. Deacon, T.W.: The symbolic species: The co-evolution of language and the brain. W.W. Norton & Company (1997)
11. DeWitt, I., Rauschecker, J.P.: Phoneme and word recognition in the auditory ventral stream. Proceedings of the National Academy of Sciences 109(8), E505–E514 (2012)
12. Doya, K.: Handbook of Brain Theory and Neural Networks, chap. Recurrent networks: learning algorithms, pp. 955–960. MIT Press (2003)
13. Duch, W., Jankowski, N.: Survey of neural transfer functions. Neural Computing Surveys 2, 163–212 (1999)
14. Eisenbeiß, S.: Generative approaches to language learning. Linguistics 47(2), 273–310 (2009)
15. Elman, J.L.: Finding structure in time. Cognitive Science 14(2), 179–211 (1990)
16. Feldman, J.A.: From Molecule to Metaphor: A Neural Theory of Language. The MIT Press (2006)
17. Frank, S.L.: Strong systematicity in sentence processing by an echo state network. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) Proceedings of the 16th International Conference on Artificial Neural Networks (ICANN 2006), *Lecture Notes in Computer Science*, vol. 4131, pp. 505–514, Springer Heidelberg. Athens, Greek (2006)
18. Frank, S.L., Haselager, W.F., van Rooij, I.: Connectionist semantic systematicity. Cognition 110(3), 358–379 (2009)
19. Friston, K.: A theory of cortical responses. Philosophical Transactions of the Royal Society B: Biological Sciences 360, 815–836 (2005)
20. Gilbert, C.D., Li, W.: Top-down influences on visual processing. Nature Reviews Neuroscience 14, 350–363 (2013)
21. Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena 42, 335–346 (1990)
22. Heinrich, S., Weber, C., Wermter, S.: Adaptive learning of linguistic hierarchy in a multiple timescale recurrent neural network. In: Villa, A.E.P., Duch, W., Érdi, P., Masulli, F., Palm, G. (eds.) Proceedings of the 22nd International Conference on Artificial Neural Networks (ICANN 2012), Lecture Notes in Computer Science, vol. 7552, pp. 555–562, Springer Heidelberg. Lausanne, Swiss (2012)
23. Heinrich, S., Weber, C., Wermter, S.: Embodied language understanding with a multiple timescale recurrent neural network. In: Mladenov, V., Koprinkova-Hristova, P., Palm, G., Villa, A.E.P., Appollini, B., Kasabov, N. (eds.) Proceedings of the 23rd International Conference on Artificial Neural Networks (ICANN 2013), Lecture Notes in Computer Science, vol. 8131, pp. 216–223, Springer Heidelberg. Sofia, Bulgaria (2013)

24. Hickok, G., Poeppel, D.: The cortical organization of speech processing. Nature Reviews Neuroscience 8(5), 393–402 (2007)
25. Hinoshita, W., Arie, H., Tani, J., Okuno, H.G., Ogata, T.: Emergence of hierarchical structure mirroring linguistic composition in a recurrent neural network. Neural Networks 24(4), 311–320 (2011)
26. Hoffmann, T., Trousdale, G.: The Oxford handbook of construction grammar. Oxford University Press (2013)
27. Iglesias, J., Eriksson, J., Pardo, B., Tomassini, M., Villa, A.E.: Emergence of oriented cell assemblies associated with spike-timing-dependent plasticity. In: Duch, W., Kacprzyk, J., Oja, E., Zadrȯzny, S. (eds.) Proceedings of the 15th International Conference on Artificial Neural Networks (ICANN 2005), Lecture Notes in Computer Science, vol. 3696, pp. 127–132. Springer Heidelberg. Warsaw, Poland (2005)
28. Iglesias, J., Villa, A.E.: Recurrent spatiotemporal firing patterns in large spiking neural networks with ontogenetic and epigenetic processes. Journal of Physiology-Paris 104(3–4), 137–146 (2010)
29. Jackendoff, R.: Foundations of language: Brain, meaning, grammar, evolution. Oxford University Press (2002)
30. Karmiloff, K., Karmiloff-Smith, A.: Pathways to language: From fetus to adolescent. Harvard University Press (2002)
31. Kullback, S.: Information Theory and Statistics. John Wiley New York (1959)
32. LeCun, Y., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Orr, G.B., Müller, K.R (eds.) Neural Networks: Tricks of the Trade (NIPS-WS 1996), Lecture Notes in Computer Science, vol. 1524, pp. 9–50, Springer Heidelberg (1998)
33. Mielke, A., Theil, F.: On rate-independent hysteresis models. Nonlinear Differential Equations and Applications NoDEA 11(2), 151–189 (2004)
34. Orban, G.A.: Higher order visual processing in macaque extrastriate cortex. Physiological Reviews 88(1), 59–89 (2008)
35. Pulvermüller, F.: The Neuroscience of Language: On Brain Circuits of Words and Serial Order. Cambridge University Press (2003)
36. Pulvermüller, F., Fadiga, L.: Active perception: sensorimotor circuits as a cortical basis for language. Nature Reviews Neuroscience 11, 351–360 (2010)
37. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the rprop algorithm. In: Ruspini, E.H. (ed.) Proceedings of the IEEE International Conference on Neural Networks (ICNN93), vol. 1, pp. 586–591, IEEE. San Francisco, CA, USA (1993)
38. Rohde, D.L.T.: A connectionist model of sentence comprehension and production. Ph.D. thesis, School of Computer Science, Carnegie Mellon University (2002)
39. Rohde, D.L.T., Plaut, D.C.: Connectionist models of language processing. Cognitive Studies 10(1), 10–28 (2003)
40. Roy, D.K., Pentland, A.P.: Learning words from sights and sounds: A computational model. Cognitive Science 26(1), 113–146 (2002)
41. Suzuki, S., Abe, K.: Topological structural analysis of digitized binary images by border following. Graphical Models and Image Processing 30(1), 32–46 (1985)
42. Tani, J., Ito, M.: Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 33(4), 481–488 (2003)
43. Wermter, S., Panchev, C., Arevian, G.: Hybrid neural plausibility networks for news agents. In: Ford, K., Forbus, K., Hayes, P., Kolodne, J., Luger, G. (eds.) Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99), pp. 93–98, AAAI Pr. Orlando, USA (1999)
44. Widrow, B., Hoff, M.E.: Adaptive switching circuits. IRE WESCON Convention Record 4, 96–104 (1960)
45. Williams, R.J., Zipser, D.: Gradient-based learning algorithms for recurrent networks and their computational complexity. In: Chauvin, Y., Rumelhart, D.E. (eds.) Backpropagation: Theory, Architectures, and Applications. Lawrence Erlbaum Associates, NJ, USA (1995)
46. Yamashita, Y., Tani, J.: Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment. PLoS Computational Biology 4(11), e1000220 (2008)