Adaptive Learning of Linguistic Hierarchy in a Multiple Timescale Recurrent Neural Network

Stefan Heinrich, Cornelius Weber, and Stefan Wermter

University of Hamburg, Department of Informatics, Knowledge Technology
Vogt-Kölln-Straße 30, D - 22527 Hamburg, Germany
{heinrich,weber,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

Abstract. Recent research has revealed that hierarchical linguistic structures can emerge in a recurrent neural network with a sufficient number of delayed context layers. As a representative of this type of network the Multiple Timescale Recurrent Neural Network (MTRNN) has been proposed for recognising and generating known as well as unknown linguistic utterances. However the training of utterances performed in other approaches demands a high training effort. In this paper we propose a robust mechanism for adaptive learning rates and internal states to speed up the training process substantially. In addition we compare the generalisation of the network for the adaptive mechanism as well as the standard fixed learning rates finding at least equal capabilities.

Keywords: MTRNN, Adaptive Learning, Language

1 Introduction

Research in cognitive intelligent systems recently demonstrated the effectiveness of *continuous time recurrent neural networks* (CTRNNs) as a method to explain higher cognitive functions, namely functions that can extract and evaluate information from time series like non-verbal gestures or verbal utterances [8]. Complex CTRNNs based on Elman or Jordan networks have been proposed and investigated to classify and generalise complex linguistic sequences [9, 10, 12].

However the training of complex sequences with long-term dependencies is difficult [3], for example the training of a *multiple timescale recurrent neural network* (MTRNN) with reasonable network size is computationally intensive, requiring up to one million training epochs for complex sequences [4]. One of the issues for these networks with a large number of parameters is to identify good learning rates for the weight updates during the learning depending on the given problem respective the specific network parameters and the shape of the sequences. With large learning rates the training of a network exhibits a faster convergence to smaller mean errors, but the probability is higher to miss minima in the error landscape and arrive at oscillating or divergent behaviour. With small learning rates the convergence becomes prohibitively slow.

For multi-layer feed forward networks it has been shown that the *resilient propagation* (RPROP) algorithm can address these problems by adapting the learning rate directly and independently for every weight and bias based on the gradient information [5]. In this paper we transfer a variant of the RPROP algorithm to the promising MTRNNs for weights and biases. Furthermore we propose an adaptation of learning rates of the *context controlling units* proportionally to the learning rates of the weights. We show the effectiveness of these methods in an analysis based on the training of a large number of very complex linguistic sequences.

2 MTRNN Architecture

The MTRNN network is composed of an *input- and output-layer* (IO) and an arbitrary number of hidden context layers. Each layer is connected with itself and with the preceding and succeeding adjacent layers. For each layer a time constant τ is used, which increases from one layer to the next, leading to an increasingly slower adaptation of the layer's neurons to a processed sequence. Fig. 1 visualises the scheme for an MTRNN with the IO layer and two context layers, called *Context fast* (Cf) and *Context slow* (Cs). Within the Cs layer, a subset of the neurons is defined as the *Context slow controlling* (Csc) units, whose initial states influence a sequence. A detailed description of the architecture can be found in the work of Yamashita and Tani 2008 [12].

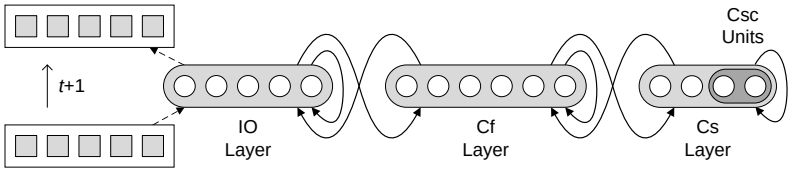


Fig. 1. Schema for a Multiple Timescale Recurrent Neural Network

For the training a variant of the *real-time backpropagation through time* (RTBPTT) algorithm is used [11], which is a temporal extension of the error propagation rule [6]. In the first step of every training epoch n, the activations are calculated and stored for all time steps in a *forward pass*, accumulating the error E between the activation values and the desired activation on the IO units. In the second step the partial derivatives for the internal states u of the neurons are determined in a *backward pass* (BP) considering the respective transfer function. Finally with the determined gradients the weights w and biases b are updated:

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

$$b_i^{n+1} = b_i - \beta_i \frac{\partial E}{\partial b_i} = b_i - \beta_i \sum_t \frac{\partial E}{\partial u_{t,i}} \quad , \tag{2}$$

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.

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

$$c_{0,i}^{n+1} = c_{0,i} - \zeta_i \frac{\partial E}{\partial u_{0,i}} \quad \text{iff } i \in I_{\text{Csc}} \quad , \tag{3}$$

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

3 Adaptive Learning Rates

In our approach the learning rates η, β, and ζ are adaptive based on the local gradient information inspired by the RPROP algorithm [5]. 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:

$$\eta_{i,j} = \begin{cases} \min\left(\eta_{i,j}^{n-1}\xi^+, \eta_{\max}\right) & \text{iff } \left(\frac{\partial E}{\partial w_{i,j}} \cdot \frac{\partial E}{\partial w_{i,j}}^{n-1}\right) > 0 \\ \max\left(\eta_{i,j}^{n-1}\xi^-, \eta_{\min}\right) & \text{iff } \left(\frac{\partial E}{\partial w_{i,j}} \cdot \frac{\partial E}{\partial w_{i,j}}^{n-1}\right) < 0 \\ \eta_{i,j}^{n-1} & \text{otherwise} \end{cases} \quad , \tag{4}$$

$$\beta_i = \begin{cases} \min\left(\beta_i^{n-1}\xi^+, \eta_{\max}\right) & \text{iff } \left(\frac{\partial E}{\partial b_i} \cdot \frac{\partial E}{\partial b_i}^{n-1}\right) > 0 \\ \max\left(\beta_i^{n-1}\xi^-, \eta_{\min}\right) & \text{iff } \left(\frac{\partial E}{\partial b_i} \cdot \frac{\partial E}{\partial b_i}^{n-1}\right) < 0 \\ \beta_i^{n-1} & \text{otherwise} \end{cases} \quad , \tag{5}$$

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:

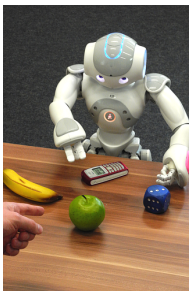
$$\zeta_i \propto \frac{1}{|I_{\text{Cf}}| + |I_{\text{Cs}}|} \sum_{j \in (I_{\text{Cf}} \cup I_{\text{Cs}})} \eta_{i,j} \quad . \tag{6}$$

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.

4 Scenario

Our research is focused on communication between humans and robots and on grounding of language in a teaching scenario [8]. A humanoid robot is supposed to learn cues about its environment based on natural language and visual information, and to recognise and generalise unheard utterances correctly (see Fig. 2a for an overview). The MTRNN is one of the cognitive modules processing the linguistic part. For the current work we investigated the hierarchical self-organisation and recognition of known and unknown linguistic utterances that stem from a grammar as presented in Fig. 2b. The small grammar is based on a vocabulary of 34 words and can generate exactly 100 different utterances. Every utterance is represented as a symbolic sentence $s = (c_1, \ldots, c_l)$, where the characters c are taken from the alphabet $\Sigma = \{\text{'a'}, ..., \text{'z'}, \text{' '}, \text{'.'}, \text{'!'}, \text{'?'}\}$ with $|\Sigma| = 30$ symbols.

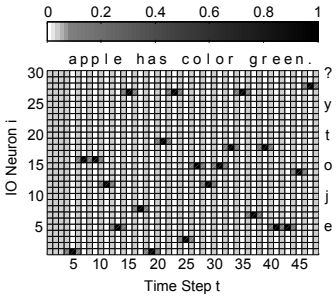
To encode the symbolic sentences into neural activation, we follow closely the encoding scheme suggested by Hinoshita et al. 2011 [1]: The occurrence of a character c_k is reflected by a spike-like neural activity of a specific neuron at relative time step r. In addition some activity is spread backward in time (rising phase) and some activity is spread forward in time (falling phase) represented as a Gaussian over the interval $[r - \omega/2, \ldots, r - 1, r, r + 1, \ldots, r + \omega/2]$. All activities of spike-like peaks are normalised by the soft-max function for every absolute time step. A detailed description of the encoding scheme can be found in [1]. For our scenario we set the constants according to their work to $\omega = 4, \sigma^2 = 0.3, \mu = 4, \upsilon = 2$. The ideal neural activation for a sample sentence is illustrated in Fig. 2c. For our corpus we obtained encoded sequences of lengths between 34 and 54 time steps.



(a) Learner

```
S → INFORM | INSTRUCT | QUESTION
INFORM → (this | DIR) is a OBJ.
INFORM → OBJ feels PRT.
INFORM → OBJ can be AFF.
INFORM → OBJ has color COL.
INSTRUCT → AFF the OBJ!
QUESTION → where is a OBJ?
QUESTION → how feels OBJ?
QUESTION → what can be OBJ?
QUESTION → which color has OBJ?
OBJ → apple | banana | dice | phone
DIR → above | below | left | right
AFF → pick | pull | push | touch
PRT → heavy | light | soft | hard
COL → blue | green | red | yellow
```

(b) Grammar



(c) Encoded sequence

Fig. 2. Scenario of language learning in human-robot interaction.

For the decoding of the sentences at every designated time step t the ith symbol from Σ is selected for the neuron $i \in I_{\text{IO}}$ with the highest activity $y_{t,i}$. If a punctuation mark ('.', '!', and '?') is the winning symbol, the end of the sentence is found and no further activity has to be taken into account.

5 Empirical Analysis

For our scenario we employed the same network parameter settings as in [1] except that we doubled the number of Cf neurons: The authors stated that the Cf neurons represent the *words*, though in their corpus 17 different words were used while in our corpus 34 words are incorporated. Specifically we used $I_{IO} = 30$, $I_{Cf} = 80$, and $I_{Cs} = 11$ neurons with timescale parameters $\tau_{IO} = 2$, $\tau_{Cf} = 5$, and $\tau_{Cs} = 70$. In addition we used a feedback rate $\varphi = 0.1$ and initialised the weights in the interval $[-0.025, 0.025]$.

In the experiments we analysed four different setups: three with fixed learning rates $\eta_{i,j} = \beta_i = \zeta_i \in \{0.05, 0.005, 0.0005\}, \forall i, j \in I_{\text{all}}$ and one with the adaptive mechanism. The choice for these settings is based on the original MTRNN study and recent applications [7, 12]. For the adaptive learning we set the parameter $\xi^+ = 1.2$, $\xi^- = 0.5$, $\eta_{\max} = 10.0$, and $\eta_{\min} = 10^{-8}$. Because of the parallel architecture of the MTRNN we implemented the network in OpenCL. In addition to the speed up of around 3x for a network with a size of 121 neurons in comparison to a conventional implementation, the parallel implementation allows for an only linear increase of the computational demands if the number of neurons is increased.

With different seeds for every setup we randomly initialised 10 networks and conducted two studies: First, we ran a preliminary experiment where we randomly selected 5 out of the 100 sentences and trained for $\theta = 60,000$ epochs. Second, we performed the main experiment where we randomly selected 50 out of the 100 sentences for training over $\theta = 600,000$ epochs and kept the other 50 sentences for testing. In both studies we trained for the reason of comparability over the same number of epochs and used no other termination criteria.

For testing the recognition capabilities we used the standard measures precision $p_{\text{precision}}$ and recall p_{recall}, defined as follows:

$$p_{\text{precision}} = \frac{tp}{tp + fp} \quad , \quad p_{\text{recall}} = \frac{tp}{tp + fn} \quad , \tag{7}$$

while we defined all correct and matching sentences as tp (true positives), all grammatically correct but not matching sentences as fp (false positives), and all incorrect sentences as fn (false negatives).

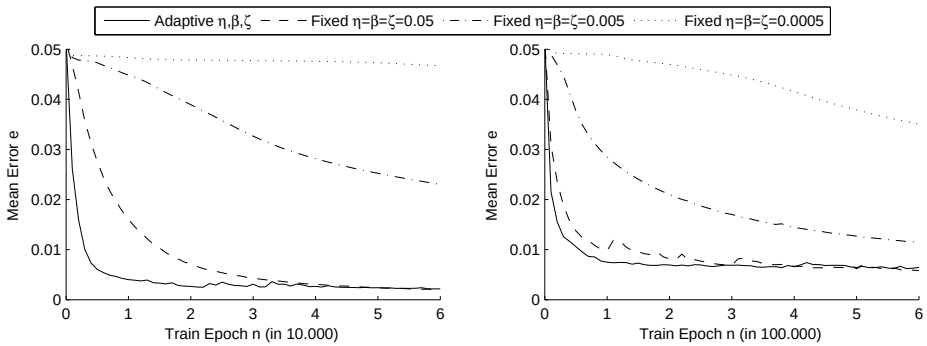
5.1 Training Effort

Results of the preliminary study revealed the faster convergence for training the networks using the adaptive mechanism (see Fig. 3a). On average after $14,000$ training epochs a mean error value of 0.003 was achieved and the networks started to oscillate slightly. The training with fixed learning rates $\eta = \beta = \zeta = 0.05$ showed the second fastest convergence after $39,000$ training epochs towards the same error value. For the setups with lower learning rates no convergence values were reached during $60,000$ training epochs.

The learning rates show a very diverse development during training (see Fig. 3c). On the one hand, the learning rates for the weights quickly reach very

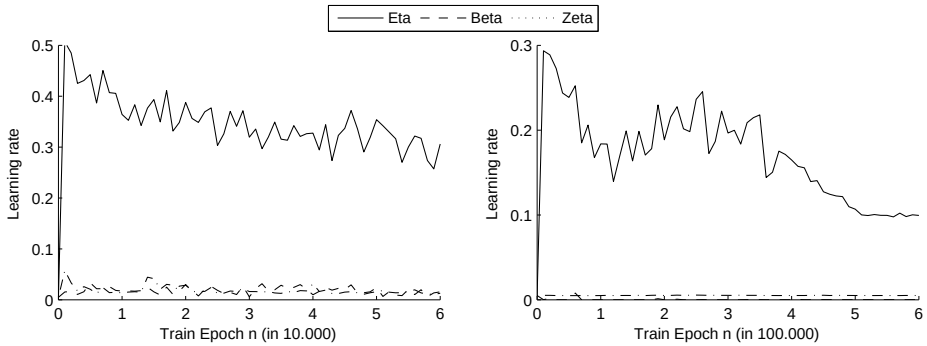
high values up to 0.5 on average and standard deviations up to 3.8: An inspection of single $\eta_{i,j}$ values revealed higher learning rates for neurons in the IO layer and lower on the Cs layer. On the other hand, learning rates for biases and the Csc states alter to very low values down to 0.005 and 0.01 respectively. The overall trend of the development of the learning rate is descending but shows large jumps indicating a gradient descent along another direction.

For the main experiment the convergence behaviour is similar to some extent: Training of 50 sentences with the adaptive mechanism shows the fastest convergence to a mean error of 0.006 after $120,000$ training epochs (see Fig. 3b). With fixed learning rates of $\eta = \beta = \zeta = 0.05$ the convergence value was reached after $380,000$ epochs. However some differences are important: First, with the fixed learning rates, three out of the ten networks diverged during the training with some weights moving infinite. Second, some of the remaining networks achieved slightly smaller mean error rates towards 0.005 while other reveal devastating jumps of the error, emphasising a high degree of problem-dependence of specific fixed learning. Networks trained with smaller fixed learning rates achieved no convergence during $600,000$ training epochs, but did not lead to divergence.



(a) 5 sequences: Average mean error

(b) 50 sequences: Average mean error



(c) 5 sequences: Average learning rates

(d) 50 sequences: Average learning rates

Fig. 3. Comparison of mean error and learning rates during training.

The learning rates for the adaptive mechanism also show very high diversity during the training: The rates for the weights developed quickly to an average mean value up to 0.3 with standard deviation up to 2.5 (see Fig. 3d). In contrast to the preliminary study, in the main experiment the learning rates for the changes of the biases decreased much more to average values of 10^{-6}.

5.2 Recognition Capabilities

The recognition results for the setups which converged during training are presented in Tab. 1. Although the networks show equal shortcomings for unknown utterances, the recognition of known utterances is good. Note that for comparison of the convergence behaviour of the training error, we did not use early stopping and obtained over-fitted networks. Additionally we strictly judged a sentence as incorrect even if only minor substitution errors occurred. Overall the networks show similar recognition capacity for both setups.

Table 1. Comparison of testing results.

(a) Validation on training data

	Adaptive η, β, ζ	Fixed $\eta = \beta = \zeta = 0.05$
$p_{\text{precision}}$	1.0	1.0
p_{recall}	0.8255	0.8486

(b) Validation on test data

	Adaptive η, β, ζ	Fixed $\eta = \beta = \zeta = 0.05$
$p_{\text{precision}}$	0.6667	0.1429
p_{recall}	0.0134	0.0029

6 Conclusion

The MTRNN is a promising CTRNN able to classify and generalise complex sequences. Trained on complex sequences, for example on linguistic utterances, this network showed high computational demands, though. To overcome this issue, Peniak et al. implemented in 2011 the MTRNN in CUDA, showing moderate to good speed ups for standard to large network sizes [4]. With OpenCL as a successor to CUDA, we were able to transfer these advantages from computers with superior graphic cards to standard computers with up-to-date CPUs.

On the algorithmic side the speed up of learning of a related *recurrent neural network with parametric bias* (RNNPB) was achieved by Kleesiek et al. in 2012 [2]. However, the *context controlling units* in a MTRNN depend differently on the sequence than the *parametric bias neurons*, making it impossible to transfer the mechanism to MTRNNs.

The adaptive mechanism for learning rates proposed in this paper is practicable and highly effective. For complex sequences and data sets the convergence of networks during training can be achieved after 100k epochs instead of several 100k or even a million epochs. Furthermore the adaptive process for the learning rates of weights, biases, as well as initial Csc units helps to avoid the search for 'good' learning rates for a given problem and a given network setup.

In the future we plan to analyse in detail the influence of the increasing and decreasing factors of the adaptive mechanism. A function based only on the proportion of the local gradient could speed up the learning even further and could spare another parameter. Furthermore we plan to integrate the vision information to the network to investigate the recognition capabilities of mixed linguistic sequences and visual object references.

Acknowledgments. The authors would like to thank J. Kleesiek and S. Magg for inspiring and very helpful discussions as well as E. Strahl for assistance with the numerous OpenCL computations.

References

1. 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)
2. Kleesiek, J., Badde, S., Wermter, S., Engel, A.K.: What do objects feel like? - Active perception for a humanoid robot. In: Proc. 4th Int. Conference on Agents and Artificial Intelligence (ICAART2012). vol. 1, pp. 64–73. SciTePress, Vilamoura, PL, (Feb. 2012)
3. Kolen, J.F., Kremer, S.C.: A Field Guide to Dynamical Recurrent Networks. Wiley-IEEE Press (2001)
4. Peniak, M., Marocco, D., Tani, J., Yamashita, Y., Fischer, K., Cangelosi, A.: Multiple time scales recurrent neural network for complex action acquisition. In: Proceedings of ICDL-Epirob 2011. IEEE, Frankfurt, GER (Aug. 2011)
5. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the rprop algorithm. In: Proc. IEEE Int. Conference on Neural Networks (ICNN93). vol. 1, pp. 586–591. IEEE, San Francisco, USA (Mar. 1993)
6. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representation by error propagation. In: Parallel Distributed Processing. MIT Press, Cambridge, MA, USA (1986)
7. Sano, S., Nishide, S., Okuno, H.G., Ogata, T.: Predicting listener back-channels for human-agent interaction using neuro-dynamical model. In: Proc. 2011 IEEE/SICE Int. Symp. on System Integration (SII2011). pp. 18–23. Kyoto, JP (Dec. 2011)
8. Steels, L., Spranger, M., van Trijp, R., Höfer, S., Hild, M.: Emergent action language on real robots. In: Language Grounding in Robots, chap. 13, pp. 255–276. Springer, New York (2012)
9. 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 (Jul. 2003)
10. Wermter, S., Panchev, C., Arevian, G.: Hybrid neural plausibility networks for news agents. In: Proc. National Conference on Artificial Intelligence. pp. 93–98. AIII Press, Orlando, USA (Jul. 1999)
11. Williams, R.J., Zipser, D.: Gradient-based learning algorithms for recurrent networks and their computational complexity. In: Backpropagation: Theory, Architectures, and Applications. Lawrence Erlbaum Associates, NJ, USA (1995)
12. 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 (11 2008)