Representing Compositionality based on Multiple Timescales Gated Recurrent Neural Networks with Adaptive Temporal Hierarchy for Character-Level Language Models

Moirangthem Dennis Singh, Jegyung Son, Minho Lee
School of Electronics Engineering
Kyungpook National University
Daegu, South Korea
{mdennissingh,wprud4,mholee}@gmail.com

Abstract

A novel character-level neural language model is proposed in this paper. The proposed model incorporates a biologically inspired temporal hierarchy in the architecture for representing multiple compositions of language in order to handle longer sequences for the character-level language model. The temporal hierarchy is introduced in the language model by utilizing a Gated Recurrent Neural Network with multiple timescales. The proposed model incorporates a timescale adaptation mechanism for enhancing the performance of the language model. We evaluate our proposed model using the popular Penn Treebank and Text8 corpora. The experiments show that the use of multiple timescales in a Neural Language Model (NLM) enables improved performance despite having fewer parameters and with no additional computation requirements. Our experiments also demonstrate the ability of the adaptive temporal hierarchies to represent multiple compositonality without the help of complex hierarchical architectures and shows that better representation of the longer sequences lead to enhanced performance of the probabilistic language model.

1 Introduction

Language Modeling is a fundamental task central to Natural Language Processing (NLP) and language understanding. A character-level language model (CLM) can be interpreted as a probability estimation method for the next character given a sequence of characters as input. From the perspective of sequence generation, predicting one character at a time has higher importance since it allows the network to invent novel words and strings. CLMs are commonly used for modeling new words and there have been successful techniques that use generative language models (LMs) based on characters or phonemes (Sutskever et al., 2011).

Recurrent neural networks have been applied to CLMs (Sutskever et al., 2011; Graves, 2013). Recently Kim et al. (2016b) introduced a LM with explicit hierarchical architecture to work at character levels and word levels. Cooijmans et al. (2017) introduced recurrent batch normalization into CLMs which significantly improved the performance. However, since the population statistics are estimated separately for each time step, the model is computationally intensive particularly for a CLM where the number of steps are more than conventional word level LMs. Similarly, Krueger and Memisevic (2016) introduced regularization in CLMs using a norm-stabilizer and reported an increased training time for higher levels of regularization.

In spite of the recent successes, CLMs still have inferior performance compared to its equivalent word-level models (Mikolov et al., 2012) since these LMs need to consider longer history of tokens to properly predict the next one. In order to improve the performance of the CLMs, there is a need for better representation of the additional levels of compositionality and the richer discourse structure found in CLMs.

Heinrich et al. (2012) used multiple timescale RNNs to learn the linguistic hierarchy for speech related tasks and Ding et al. (2016) demonstrated that, during listening to connected speech, cortical activity of different timescales concurrently tracked the time course of abstract linguistic compositionality at different hierarchical levels, such as words, phrases and sentences. In this work,

Proceedings of the 2nd Workshop on Representation Learning for NLP, pages 131–138,
Vancouver, Canada, August 3, 2017.

we propose a character-level recurrent neural network (RNN) LM that employs an adaptive multiple timescales approach to incorporate temporal hierarchies in the architecture to enhance the representation of multiple compositionalities. Our proposed model includes a novel timescale update mechanism which enhances the adaptation of the temporal hierarchy during the learning process. We build the temporal hierarchical structure using fast and slow context units to imitate different timescales. This temporal hierarchy concept is implemented based on the multiple timescales gated recurrent unit (MTGRU) (Kim et al., 2016a) that incorporates multiple timescales at different layers of the RNN.

Our model, inspired by the concept of temporal hierarchy found in the human brain (Botvinick, 2007; Meunier et al., 2010), demonstrates the ability to capture multiple compositionalities similar to the findings of Ding et al. (2016). This better representation learning capability enhances the ability of our model to handle longer sequences for the CLM. The resulting LM is a much simpler model that does not incorporate explicit hierarchical structures or normalization techniques. We show that our CLM with the biologically inspired temporal hierarchy is able to achieve performance comparable to the existing state-of-the-art CLMs evaluated over the Penn Treebank (PTB) and Text8 corpora.

2 Related Works

Recent advances in distributed representation learning have demonstrated promising results in language modeling. Distributed representation learning approaches are a group of methods in which real-valued vectors are trained to capture the underlying meaning in the input. Bengio et al. (2003) demonstrated that the probabilistic language models can achieve much better generalization.

Out-of-vocabulary words have always been a problem in language tasks. To address the rare word problem in language generation, Alexandrescu and Kirchhoff (2006) represented a word as a set of shared factor embeddings. In another approach, Sutskever et al. (2011) introduced NLMs that incorporated a character-level model with both input and output as characters. CLMs are also capable of generating novel words and are suitable for addressing the rare word problem.

Training a CLM has been a difficult task and its performance has been lower than the word-level LMs. Handling longer sequences is critical to improve the performance of CLMs and it remains a challenge. Mikolov et al. (2012) proposed an alternative approach, where subword-level models are trained to benefit from the word-level LMs. Pachitariu and Sahani (2013) proposed impulse-response LMs for improving RNN LMs. Recently, several studies on RNN based CLMs have been proposed, mostly using the Long short-term memory (LSTM) (Graves, 2013; Cooijmans et al., 2017; Zhang et al., 2016), where numerous regularization and normalization techniques have been applied to enhance the performance of CLMs. Additionally, weight generation networks (Ha et al., 2017), hierarchical architectures (Chung et al., 2017) in conjunction with the normalization techniques have been proposed to achieve state-of-the-art performance.

We propose a different approach for our CLM by using a simpler biologically inspired temporal hierarchy model. Recently, Ding et al. (2016) demonstrated strong evidence for a neural tracking of hierarchical linguistic structures in the brain. The study performed experiments to determine whether neural representation of language (speech) tracks hierarchical linguistic structures, rather than prosodic and statistical transitional probability cues. In simpler terms, the brain tracks and represents linguistic structures hierarchically. Similar findings have been shown in Meunier et al. (2010) and Botvinick (2007), but Ding et al. (2016) was the first to confirm the same in the domain of language. The authors hypothesized that concurrent neural tracking of hierarchical linguistic structures provides a mechanism for temporally integrating smaller linguistic units into larger structures. Therefore, our knowledge of the hierarchical nature of linguistic structures and the theory of linguistic compositionality have been shown to be biologically plausible. Previous works have applied this hierarchical structure to RNNs in movement tracking (Paine and Tani, 2004), sensorimotor control systems (Yamashita and Tani, 2008) and speech recognition (Heinrich et al., 2012). Based on the above conclusions, we adopt the multiple timescales concept to implement the temporal hierarchy architecture for representing multiple compositionalities which will help in handling longer sequences for our CLM.

Moreover, Yamashita and Tani (2008) had tested their model performance over different values of the timescale τ to investigate the impact of multiple timescales in RNNs. They showed that different setting of the timescales significantly affects the performance and a higher τ-ratio(τ-slow/τ-fast) improved the performance. In this spirit, we implement an adaptive timescale update method for better performance compared to a model with static timescales.

3 Proposed Character-Level Neural Language Model

A character-level language model (CLM) estimates a probability distribution over w_{t+1} given a sequence $w_{1:t} = [w_1, \ldots, w_t]$. We propose a recurrent neural network based CLM with temporal hierarchies using a multilayer gated recurrent neural network. Gated RNNs such as LSTM (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014) address the problem of learning long range dependencies. GRU and LSTM have been shown to yield comparable performance (Chung et al., 2014). These gated recurrent architectures are known to address the vanishing gradient problem efficiently and multilayer architectures are known to be able to learn expressive and complex features. However, the phenomenon responsible for these algorithms to approach human performance on speech and language tasks cannot be ascertained owing to our lack of understanding or insight into the actual representations that are being learned. However, due to our lack of understanding or insight into the actual representations being learned, it is difficult to ascertain the phenomenon responsible for these algorithms to approach human performance on speech and language tasks. Therefore, concurrent to the studies of temporal hierarchy in neuroscience (Ding et al., 2016), we formulated a hypothesis that multilayer gated recurrent neural networks can represent compositional hierarchies in the learning process that involves a monotonically increasing timescale hierarchy. We argue that hierarchical temporal representations capture the linguistic hierarchy of the input and are primarily responsible for better performance of multilayer gated recurrent architectures.

We propose a Multiple Timescales Gated Recurrent Unit (MTGRU) with adaptive timescales for our language model. The MTGRU, which has a temporal hierarchical architecture, is implemented in the framework of a language model as shown in Figure 1. We find that the multiple timescales model is primarily responsible for explicitly guiding each layer of the neural network to facilitate in learning of features operating over increasingly slower timescales, corresponding to subsequent levels in the compositional hierarchy. The temporal hierarchy in the network is implemented by applying a timescale constant at the end of a conventional GRU unit, essentially adding another constant gating unit that modulates the mixture of past and current hidden states. In an MTGRU, each step takes as input $\mathbf{x}_t, \mathbf{h}_{t-1}$ and produces the hidden $\mathbf{h_t}$. The time constant τ added to the activation $\mathbf{h}_t$ of the MTGRU is shown in Eq. (1). τ is used to control the timescale of each GRU cell. Larger τ results in slower cell outputs but it makes the cell focus on the slow features of a dynamic sequence input. The MTGRU model is illustrated in Figure 2.

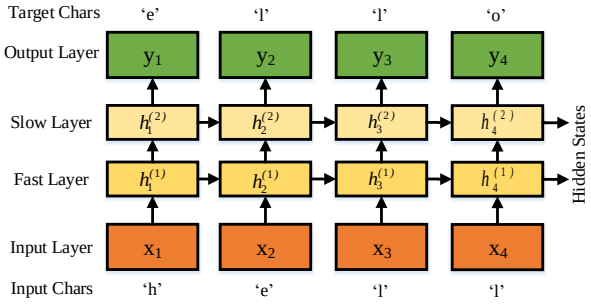

Figure 1: Proposed character-level neural language model.

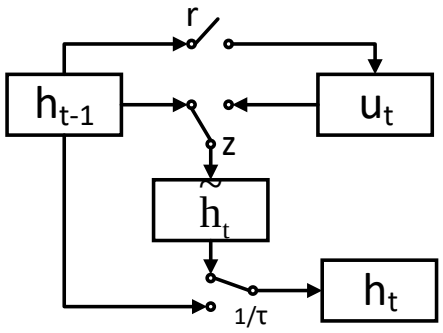

Figure 2: A Multiple Timescales Gated Recurrent Unit.

$$
\begin{aligned}
r_t &= \sigma(W_{xr}x_t + W_{hr}h_{t-1}) \\
z_t &= \sigma(W_{xz}x_t + W_{hz}h_{t-1}) \\
u_t &= \tanh(W_{xu}x_t + W_{hu}(r_t \odot h_{t-1})) \\
\tilde{h}_t &= z_t h_{t-1} + (1 - z_t)u_t \\
h_t &= \tilde{h}_t \frac{1}{\tau} + (1 - \frac{1}{\tau})h_{t-1}
\end{aligned} \tag{1}
$$

where $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid and tangent hyperbolic activation functions, $\odot$ denotes the element-wise multiplication operator, and $\mathbf{r}_t$, $\mathbf{z}_t$ are referred to as *reset*, *update* gates respectively. $\mathbf{u}_t$ and $\tilde{\mathbf{h}}_t$ are the candidate activation and candidate hidden state of the MTGRU.

We build the multilayer MTGRU-CLM with a different timescale τ for each layer. Based on our hypothesis that later layers should learn features that operate over slower timescales, we set larger τ as we go up the layers. We use the bits-per-character (BPC) as the evaluation metric. The timescale τ is initialized for each layers at the start of the training.

We implement the proposed timescale update mechanism by adaptively increasing the τ during the training process as it is known that just higher τ-ratio, without timescale adaptation, leads to improved performance (Yamashita and Tani, 2008). As we proceed with the training epochs, whenever the validation negative log-likelihood (NLL) (shown in Eq.(2)) stopped decreasing, the timescale τ is updated. A $growth\_factor$ is used to determine the growth rate of the timescales. We update the timescales only after training has completed for a particular number of epochs (max_epoch). The timescale update mechanism is presented in Algorithm 1. In order to prevent deteriorated performance over large increases in the timescales, smaller growth factors are set for the experiments.

$$
-\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\log P\left(w_t^n \mid w_1^n \ldots w_{t-1}^n; \theta\right) \tag{2}
$$

where N is the number of training sequences, T is the length of the n^{th} sequence, θ is the model parameter, and w_t^n is the token at time t of sequence n and so on.

4 Experiments and Results

We evaluate our CLM on the Penn Treebank (PTB) corpus (Marcus et al., 1993) and on the much larger Text8 corpus (Mahoney, 2009). We use orthogonal initialization for all the weight matrices and use stochastic gradient descent with gradient clipping at 1.0 and step rule determined by Adam (Kingma and Ba, 2014). We report the hyperparameter values that were explored in our experiments in Table 1. The timescale for the fast layer is initialized to 1 in all the experiments as $\tau = 1$ defines the default or the input timescale.

We also conduct an additional experiment for the comparison of computational efficiency of our model with normalization based techniques. The details of all the experiments are described in the sections below.

Input: Current Timescale τ
Output: Updated Timescale τ
if *current_epoch* $>$ *max_epoch* **then**
  read $growth\_factor$;
  if *the validation NLL did not decrease* **then**
    τ = τ * $growth\_factor$;
    return τ;
  else
    return τ;
  end
end

Algorithm 1: Timescale adaptation in MTGRU

τ	$\{1.2, 1.3, 1.35, 1.4\}$
$growth\_factor$	$\{1.01, 1.05, 1.1, 1.15\}$
learning rate	{1e-2, 1e-3, 1e-4, 2e-3}
minibatch size	{32, 64, 128}

Table 1: Grid of hyperparameters explored in the experiments.

4.1 Penn Treebank (PTB) Corpus

The PTB corpus is divided to train, valid, and test sets following Mikolov and Zweig (2012). For this experiment we use 600 units in each layer of the 2 layer MTGRU network. We train on non-overlapping sequences of length 100 with a mini-batch size of 64 and a learning rate of 0.002. We initialize the timescales τ as $\{1, 1.3\}$ for the fast and the slow layers respectively. We set the $growth\_factor$ to 1.05 with a max_epoch of 25. The size of our model is 3.7M parameters.

We also implemented the batch normalized gated recurrent unit (BN-GRU) following Cooijmans et al. (2017) as a baseline to compare with

Model	BPC	Size
Zhang et al. (2016)	1.49	-
Mikolov et al. (2012)	1.41	-
Krueger and Memisevic (2016)	1.39	4.25M*
BN-GRU	1.39	4.1M
Mikolov et al. (2012)	1.37	-
Cooijmans et al. (2017)	1.32	4.25M*
Ha et al. (2017)	1.31	4.25M
MTGRU-CLM	1.27	3.7M
Ha et al. (2017)	1.27	4.91M
Chung et al. (2017)	1.24	5.35M*
MTGRU-CLM-Adaptive	1.24	3.7M
Ha et al. (2017)	1.22	14.41M

Table 2: Bits-Per-Character on PTB test and size of the models. **MTGRU-CLM** and **MTGRU-CLM-Adaptive** correspond to our CLMs with a constant timescale and an adaptive timescales respectively. *These are estimated model sizes as the actual number of parameters is not available in the literature.

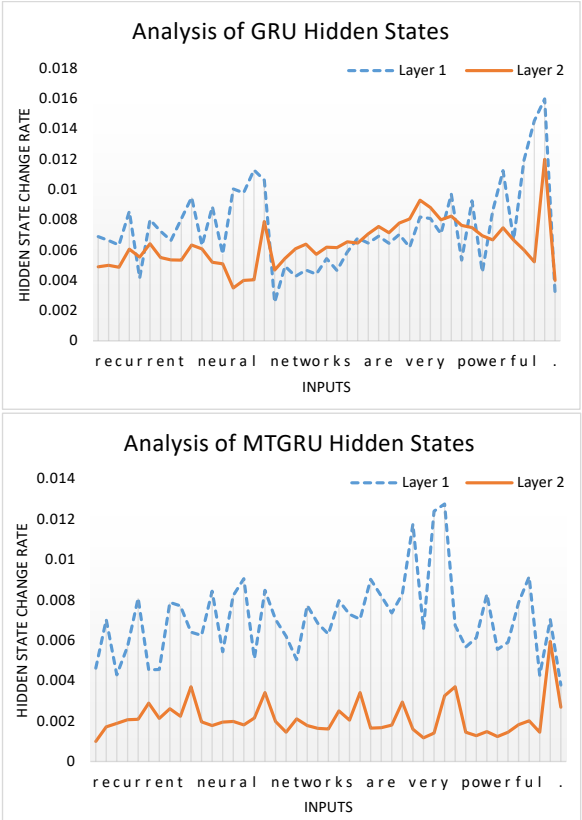

Figure 3: Hidden state representation of GRU-CLM and MTGRU-CLM-Adaptive.

our model. We use early-stopping on validation-set performance and the resulting model is evaluated over the test set and the results are summarized in Table 2. Our model performed comparable to the current state-of-the-art models with a test BPC of 1.24. We also illustrate the performance of our model in Table 4 under the different settings of τ and $growth\_factor$ given in Table 1.

For further analysis, we graph the hidden state change rate of each layer by measuring the $L2$ distance between the past and current hidden states of the MTGRU-CLM-Adaptive and the GRU CLM, over the input time steps as shown in Figure 3. The models are trained on the PTB set with the same set of parameters. In this graph, "spikes" can be interpreted as events where the input representation at each layer is varying significantly. The first(fast) layer, where the layer of the recurrent unit is exposed directly to the input (as illustrated in Fig. 1), the spiking corresponds to each character of the input. Looking at the second(slow) layer graph of MTGRU, we can observe the larger spikes correspond to the end of each word, while the same is not true in the case of GRU. It is evident that MTGRU learns better representation of each word by utilizing the temporal hierarchy without any explicit architectural hierarchy. These findings are consistent with the previous studies on speech where in speech sounds, syllable-level information on short time scale is integrated into word-level information over a longer time scale (Yamashita and Tani, 2008; Ding et al., 2016).

4.2 Text8 Corpus

The Text8 corpus consists of 100M characters extracted from the Wikipedia corpus. We follow Mikolov and Zweig (2012) and divide the train, valid, and test sets accordingly in order to compare with previous works. Since Text8 contains only alphabets and spaces, the total number of symbols is just 27. We use 1200 units in each layer of the 2 layer MTGRU network and it is trained over non-overlapping sequences of length 180. The mini-batch size is set to 128 and the learning rate is 0.001. The timescales τ is initialized as $\{1, 1.3\}$ for the fast and the slow layers respectively with a $growth\_factor$ of 1.05. The max_epoch parameter is set to be 25. The size of this model is 14.5M parameters. The performance of the resulting model with early-stopping on validation-set is shown in Table 3. We also report the performance of the BN-GRU baseline model on the Text8 corpus. The MTGRU-CLM-Adaptive model obtains a test BPC of 1.29 which is comparable to the

Model	BPC	Size
Mikolov et al. (2012)	1.54	-
Zhang et al. (2016)	1.49	-
Pachitariu and Sahani (2013)	1.48	-
BN-GRU	1.39	16.1M
Cooijmans et al. (2017)	1.36	16.22M*
MTGRU-CLM	1.34	14.5M
Chung et al. (2017)	1.32	21M*
Chung et al. (2017)	1.29	21M*
MTGRU-CLM-Adaptive	1.29	14.5M

Table 3: Bits-Per-Character on Text8 test and size of the Models. **MTGRU-CLM** and **MTGRU-CLM-Adaptive** correspond to our CLMs with a constant timescale and an adaptive timescales respectively. *These are estimated parameter sizes as the actual value is not available in the literature.

Corpus	BPC
PTB Corpus	1.26±0.0158
Text8 Corpus	1.31±0.0187

Table 4: Performance of the proposed model under different timescale updates. These are the performance of the model under different settings of τ and $growth\_factors$ as shown in Table 1.

current state-of-the-art. The performance of this model under different τ and $growth\_factor$ settings given in Table 1 is shown in Table 4.

4.3 Comparison of Computing Efficiency

In order to compare the computation efficiency of MTGRU with the baselines, we replicated the Sequential MNIST experiment from Cooijmans et al. (2017) using our implementation of LSTM, GRU, Batch Normalized (BN)-LSTM, BN-GRU, MTGRU, and MTGRU-Adaptive following the same experimental conditions. Despite the faster convergence of BN-LSTM, BN-GRU, MTGRU, and MTGRU-Adaptive as illustrated in Figure 4, the total time taken to train 100K steps of BN-LSTM, BN-GRU significantly increased compared to LSTM and GRU while the training time of MTGRU and MTGRU-Adaptive remained comparable to GRU as shown in Table 5. The experiments were performed on one machine with an Nvidia Titan-X GPU. Moreover, Kim et al. (2016a) already demonstrated much faster convergence in the case of MTGRU when compared to GRU in sequence-to-sequence tasks.

Model	Accuracy	Train Time
LSTM	98.89%	14 hours
GRU	98.56%	15 hours
BN-LSTM	99.01%	41 hours
BN-GRU	98.97%	43 hours
MTGRU	99.26%	17 hours
MTGRU-Adaptive	99.37%	17 hours

Table 5: Sequential MNIST classification results with training duration of 100K steps.

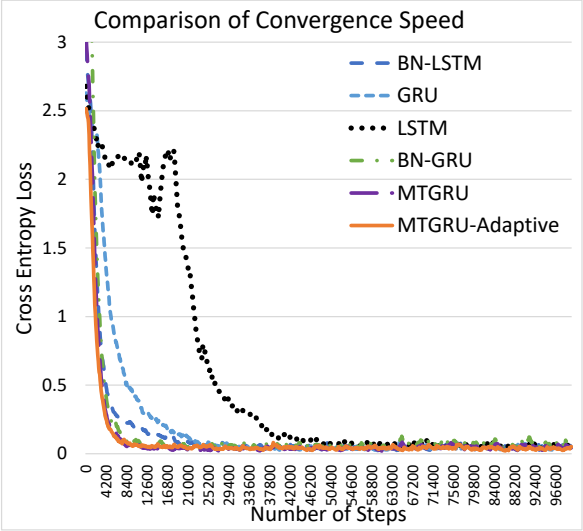

Figure 4: Comparison of convergence speed of the various models for Sequential MNIST classification task.

5 Discussion

The proposed method based on a biologically inspired hierarchical structure can represent multiple compositions of language by virtue of the adaptive multiple timescales in each layer. Sensitivity of the human brain to the compositional structure of language was recently confirmed by Ding et al. (2016). By recording the activity of listeners brains using magnetoencephalography (MEG), it was found that the brain activates or spikes when it is presented with individual words, phrases, or a whole sentence. We successfully replicated this property in our model and it achieves significant performance gains despite having a simpler structure and lesser number of parameters. Our model's ability to represent compositions of language at the word level is illustrated in Figure 3. This illustration validates our hypothesis that multilayer gated recurrent neural networks can represent compositional hierarchies similar to the human brain.

The enhanced performance of the proposed CLM illustrates that our approach can give better generalization performance without the help of complex hierarchical architectures. The results indicate that the temporal hierarchies with the adaptive timescale approach can represent the compositonality better and increases the capability of the model to handle longer sequences for the CLM. The results also demonstrate that our multilayer MTGRU model with adaptive timescale performs comparable to the current state-of-the-art models despite having fewer parameters and a simpler architecture. The temporal hierarchy approach eliminates the need for complex structures and normalization techniques (Cooijmans et al., 2017; Krueger and Memisevic, 2016; Chung et al., 2017; Ha et al., 2017) for the LM task, thereby increasing the computational efficiency of our model.

6 Conclusion

Our approach incorporates temporal hierarchies in a character-level NLM to improve the performance of the language model without introducing additional parameters. The proposed approach takes into account the need for a biologically plausible structure and a model to implement simpler hierarchies for handling different level of language compositions in order to tackle the longer sequences problem in CLMs. Our approach with adaptive timescales enables a simpler model with a better representation of language to achieve significant performance gains over existing models with larger complexities and also alleviates the need of additional computations.

Acknowledgment

This research was supported by ICT R&D program of MSIP/IITP [R7124-16-0004, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding] (70%) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2016M3C1B6929647) (30%).

References

Andrei Alexandrescu and Katrin Kirchhoff. 2006. Factored neural language models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, pages 1–4.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *journal of machine learning research* 3(Feb):1137–1155.

Matthew M Botvinick. 2007. Multilevel structure in behaviour and in the brain: a model of fuster's hierarchy. *Philosophical Transactions of the Royal Society B: Biological Sciences* 362(1485):1615–26. https://doi.org/10.1098/rstb.2007.2056.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR* abs/1406.1078. http://arxiv.org/abs/1406.1078.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2017. Hierarchical multiscale recurrent neural networks. In *Proceeding of the International Conference on Learning Representations*.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* abs/1412.3555. http://arxiv.org/abs/1412.3555.

Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülçehre, and Aaron Courville. 2017. Recurrent batch normalization. In *Proceeding of the International Conference on Learning Representations*.

Nai Ding, Lucia Melloni, Hang Zhang, Xing Tian, and David Poeppel. 2016. Cortical tracking of hierarchical linguistic structures in connected speech. *Nature neuroscience* 19(1):158–164.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* .

David Ha, Andrew Dai, and Quoc V Le. 2017. Hypernetworks. In *Proceeding of the International Conference on Learning Representations*.

Stefan Heinrich, Cornelius Weber, and Stefan Wermter. 2012. Adaptive learning of linguistic hierarchy in a multiple timescale recurrent neural network. In *International Conference on Artificial Neural Networks*. Springer, pages 555–562.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Minsoo Kim, Moirangthem Dennis Singh, and Minho Lee. 2016a. Towards abstraction from extraction: Multiple timescale gated recurrent unit for summarization. *ACL 2016* pages 70–77.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016b. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, pages 2741–2749.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

David Krueger and Roland Memisevic. 2016. Regularizing rnns by stabilizing activations. In *Proceeding of the International Conference on Learning Representations*.

Matt Mahoney. 2009. Large text compression benchmark. *URL: http://www. mattmahoney. net/text/text. html* .

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics* 19(2):313–330.

D. Meunier, R. Lambiotte, A. Fornito, K. D. Ersche, and E. T. Bullmore. 2010. Hierarchical modularity in human brain functional networks. *ArXiv e-prints* .

Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. 2012. Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)* .

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *SLT*. pages 234–239.

Marius Pachitariu and Maneesh Sahani. 2013. Regularization and nonlinearities for neural language models: when are they needed? *arXiv preprint arXiv:1301.5650* .

Rainer W. Paine and Jun Tani. 2004. Motor primitive and sequence self-organization in a hierarchical recurrent neural network. *Neural Networks* 17(89):1291 – 1309. New Developments in Self-Organizing Systems.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. pages 1017–1024.

Yuichi Yamashita and Jun Tani. 2008. Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment. *PLoS Comput Biol* 4(11):1–18. https://doi.org/10.1371/journal.pcbi.1000220.

Saizheng Zhang, Yuhuai Wu, Tong Che, Zhouhan Lin, Roland Memisevic, Ruslan R Salakhutdinov, and Yoshua Bengio. 2016. Architectural complexity measures of recurrent neural networks. In *Advances in Neural Information Processing Systems*. pages 1822–1830.