INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

LSTM, GRU, Highway and a Bit of Attention: An Empirical Overview for Language Modeling in Speech Recognition

Kazuki Irie, Zoltán Tüske, Tamer Alkhouli, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition,
Computer Science Department, RWTH Aachen University, Aachen, Germany
{irie,tuske,alkhouli,schlueter,ney}@cs.rwth-aachen.de

Abstract

Popularized by the long short-term memory (LSTM), multiplicative gates have become a standard means to design artificial neural networks with intentionally organized information flow. Notable examples of such architectures include gated recurrent units (GRU) and highway networks. In this work, we first focus on the evaluation of each of the classical gated architectures for language modeling for large vocabulary speech recognition. Namely, we evaluate the highway network, lateral network, LSTM and GRU. Furthermore, the motivation underlying the highway network also applies to LSTM and GRU. An extension specific to the LSTM has been recently proposed with an additional highway connection between the memory cells of adjacent LSTM layers. In contrast, we investigate an approach which can be used with both LSTM and GRU: a highway network in which the LSTM or GRU is used as the transformation function. We found that the highway connections enable both standalone feedforward and recurrent neural language models to benefit better from the deep structure and provide a slight improvement of recognition accuracy after interpolation with count models. To complete the overview, we include our initial investigations on the use of the attention mechanism for learning word triggers.

Index Terms: language modeling, speech recognition, long short-term memory, gated recurrent unit, highway network, word trigger, attention

1. Introduction

The language model (LM) is a crucial component for the automatic speech recognition, including newly emerging end-to-end systems [1, 2]. The combination of two complementary approaches, n-gram count-based [3, 4] and neural network-based modeling [5, 6, 7], achieves current state-of-the-art in language modeling. Recent advances are made on the improvements of the latter. Specifically, the language model based on the long short-term memory (LSTM) recurrent neural network (RNN) has been shown to be very effective [8]. The main innovation of the LSTM [9, 10, 11] is the use of soft gates in the architecture, themselves modeled by the activation of RNNs. In fact, the LSTM stores an internal memory cell activation in addition to its output activation. Each type of access to its internal memory cell (namely write, reset and read actions) is regulated by its corresponding gate via multiplication between the gate activation and the activation related to the corresponding action. While the gating makes it look complex at first sight, a simple stochastic gradient descent has been empirically shown to be effective to train such a model: after all, the original motivation of its architecture is to ease the training by alleviating the vanishing gradient problem that the standard RNN suffers from during training under the backpropagation through time.

The success of the LSTM has opened room for creativity in designing neural networks with multiplicative gates. While earlier works [12] motivated multiplications to make higher-order neural networks, recent works use them as a means to control the information flow inside the network. As a result, many concepts have been emerged. The gated recurrent unit [13] was proposed as a simpler alternative to the LSTM. The highway network [14, 15] has gates to ensure the unobstructed information flow across the depth of the network. Also, more elementary gating can be found in tensor networks [16], also known as the lateral network [17].

The objective of this paper is to evaluate the effectiveness of these concepts for language modeling with application to large vocabulary speech recognition. To be more specific, we first investigate the effect of the highway connection in feedforward models. An evaluation of the lateral network is also included in this analysis. Second, we carry out an exhaustive comparison between LSTM and GRU with respect to the number of hidden nodes and layers. A number of investigations have been already done previously to compare LSTM and GRU [18, 19] for multiple tasks including language modeling. Nevertheless, the experiments were often carried out on small tasks, typically on Penn Treebank. While such results are already insightful, further investigations on larger tasks, that involve the full speech recognition pipeline together with an n-gram count model, would provide a better practical overview.

Furthermore, as will be shown, the motivation underlying highway connections also applies to recurrent networks. The extension of LSTM with a linear connection between the memory cells of adjacent layers has been proposed by several works [20, 21]. Such a technique is specific to the LSTM. Instead, we investigate the direct application of the highway to the recurrent network, by substituting the transformation operation in the highway layer by a gated RNN. Such an extension can be used for both LSTM and GRU.

However, gating is not a unique way to make the intention explicit in neural networks. Recently, the attention mechanism has been designed to select the relevant parts of its inputs for a specific prediction. This has been shown to be successful in many applications [1, 2, 22]. Therefore we investigate the learning of a simple neural language model from which word triggers [23, 24] can be explicitly visualized.

2. Networks with Multiplicative Gates

In this section, we shortly review the classical model architectures based on gates. In the rest of the paper, $\odot$ denotes the element-wise product and $\boldsymbol{x}$ (also $\boldsymbol{x}_t$ or $\boldsymbol{x}_t^{(\ell)}$) denotes the input to the layer, while $\boldsymbol{h}$ (or $\boldsymbol{h}_t$) denotes the output. $\boldsymbol{W}_*$ and $\boldsymbol{R}_*$ are weight matrices and $\boldsymbol{w}_*$ are weight vectors, $\boldsymbol{b}_*$ denote biases. Sigmoid activation function (σ) and hyperbolic tangent ($\tanh$) are applied element-wise to its argument vector.

http://dx.doi.org/10.21437/Interspeech.2016-491

2.1. Highway network

The highway network was introduced in [14, 15]. The commonly used highway layers are defined as:

$$\boldsymbol{y} = \sigma(\boldsymbol{W_y}\boldsymbol{x} + \boldsymbol{b}_y) \tag{1}$$

$$\boldsymbol{g} = \sigma(\boldsymbol{W_g}\boldsymbol{x} + \boldsymbol{b}_g) \tag{2}$$

$$\boldsymbol{h} = \boldsymbol{g} \odot \boldsymbol{y} + (\boldsymbol{1} - \boldsymbol{g}) \odot \boldsymbol{x} \tag{3}$$

The transformed feature $\boldsymbol{y}$ (Eq. 1) is interpolated (Eq. 3) to the untransformed feature $\boldsymbol{x}$ using weights which are learned as a neural network (Eq. 2). The original motivation of this architecture is to ensure an unobstructed information flow between adjacent layers via linear connection, called highway connection (the second term in the right-hand side of Eq. 3). In [15], it has been shown that such an architecture effectively enables the training of very deep networks (up to 900 layers). However, in practice for language modeling, the benefit for such a connection has been reported for models with much fewer layers. In [25], the highway is used in language modeling as a means to combine the word-level feature with character-level local features; while using only two layers of the highway, the improvements in perplexity were reported. After all, the highway can also be seen as a pure feature combination operation between the features from different stages of transformation. We denote this highway layer Sigm-HW in the experimental section.

2.2. Lateral network (Tensor network)

The equations for a lateral network can be obtained by using Eq. 1-2 and:

$$\boldsymbol{h} = \boldsymbol{g} \odot \boldsymbol{y} \tag{4}$$

First of all, this can be seen as a variant of maxout networks [26] with two groups, which is obtained by redefining the $\odot$ operation as an element-wise maximum operation instead of product. Another way to interpret this architecture is to consider $\boldsymbol{g}$ as a relevance gate and $\boldsymbol{y}$ as a simple transformation of $\boldsymbol{x}$ (a highway network without highway connection). In [17], this model has been evaluated for language modeling and has been shown to outperform its variant based on the maximum operation.

2.3. Long short-term memory (LSTM)

The standard LSTM-RNN is defined by:

$$\boldsymbol{y}_t = \tanh(\boldsymbol{W_y}\boldsymbol{x}_t + \boldsymbol{R}_y\boldsymbol{h}_{t-1} + \boldsymbol{b}_y) \tag{5}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W_i}\boldsymbol{x}_t + \boldsymbol{R}_i\boldsymbol{h}_{t-1} + \boldsymbol{b}_i + \boldsymbol{w}_i \odot \boldsymbol{c}_{t-1}) \tag{6}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W_f}\boldsymbol{x}_t + \boldsymbol{R}_f\boldsymbol{h}_{t-1} + \boldsymbol{b}_f + \boldsymbol{w}_f \odot \boldsymbol{c}_{t-1}) \tag{7}$$

$$\boldsymbol{c}_t = \boldsymbol{i}_t \odot \boldsymbol{y}_t + \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} \tag{8}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W_o}\boldsymbol{x}_t + \boldsymbol{R}_o\boldsymbol{h}_{t-1} + \boldsymbol{b}_o + \boldsymbol{w}_o \odot \boldsymbol{c}_t) \tag{9}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t) \tag{10}$$

While this is not the unique variant of LSTM (in particular, the peephole connections are often removed for efficiency), in this work, we stick to the LSTM with these standard equations. We refer to [27] for an overview.

2.4. Gated recurrent units (GRU)

The equations for the GRU (the version in [18]) are as follows:

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W_z}\boldsymbol{x}_t + \boldsymbol{R}_z\boldsymbol{h}_{t-1} + \boldsymbol{b}_z) \tag{11}$$

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W_r}\boldsymbol{x}_t + \boldsymbol{R}_r\boldsymbol{h}_{t-1} + \boldsymbol{b}_r) \tag{12}$$

$$\boldsymbol{y}_t = \tanh(\boldsymbol{W_h}\boldsymbol{x}_t + \boldsymbol{R}_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}_h) \tag{13}$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{y}_t + (\boldsymbol{1} - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} \tag{14}$$

In contrast to the LSTM, GRU has only two gates (reset r_t and update z_t) and does not have the memory cell.

3. Incorporating Highway into Gated RNNs

The highway network was originally introduced for feedforward multilayer perceptrons (MLP). However, its motivation (Sec. 2.1) also applies to recurrent networks.

3.1. Existing technique: Depth-gated LSTM

Many works [20, 21] suggested the extension of stacked LSTMs with additional linear connections between memory cells of adjacent LSTM layers. This is a natural extension for the LSTM since its memory cell had already linear connection over time (Eq. 8). In [20], such an architecture has been used for acoustic modeling and has been shown to outperform the standard LSTM, especially in the context of discriminative training. The proposed LSTM architecture, depth-gated LSTM or highway LSTM is obtained by replacing Eq 8 by:

$$\boldsymbol{c}_t^{(\ell)} = \boldsymbol{i}_t \odot \boldsymbol{y}_t + \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1}^{(\ell)} + \boldsymbol{d}_t \odot \boldsymbol{c}_t^{(\ell-1)} \tag{15}$$

where

$$\boldsymbol{d}_t = \sigma(\boldsymbol{W_d}\boldsymbol{x}_t^{(\ell)} + \boldsymbol{w}_{d1} \odot \boldsymbol{c}_{t-1}^{(\ell)} + \boldsymbol{b}_d + \boldsymbol{w}_{d2} \odot \boldsymbol{c}_t^{(\ell-1)}) \tag{16}$$

if the predecessor layer $(\ell-1)$ is also an LSTM layer, otherwise:

$$\boldsymbol{c}_t^{(\ell)} = \boldsymbol{i}_t \odot \boldsymbol{y}_t + \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1}^{(\ell)} + \boldsymbol{d}_t \odot \boldsymbol{x}_t^{(\ell)} \tag{17}$$

$$\boldsymbol{d}_t = \sigma(\boldsymbol{W_d}\boldsymbol{x}_t^{(\ell)} + \boldsymbol{w}_d \odot \boldsymbol{c}_{t-1}^{(\ell)} + \boldsymbol{b}_d) \tag{18}$$

By construction, the number of nodes in the layers ℓ and its predecessor $(\ell - 1)$ should match unless a projection layer is inserted in addition.

This is an extension specific to the LSTM. In contrast, we investigated a direct application of the highway operation, which can be used for both GRU and LSTM. The description for GRU follows.

3.2. GRU-Highway: simple substitution

Since the highway layer consists of an interpolation of transformed and untransformed features (Eq. 3), the transformation part (Eq. 1) can be replaced by any other operation, for example by the GRU. The full equations for such a model can be obtained with Eq. 11-13 and in addition:

$$\boldsymbol{h}_t^{(\text{gru})} = \boldsymbol{z}_t \odot \boldsymbol{y}_t + (\boldsymbol{1} - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} \tag{19}$$

$$\boldsymbol{g}_t = \sigma(\boldsymbol{W_g}\boldsymbol{x}_t + \boldsymbol{R}_g\boldsymbol{h}_{t-1} + \boldsymbol{b}_g) \tag{20}$$

$$\boldsymbol{h}_t = \boldsymbol{g}_t \odot \boldsymbol{h}_t^{(\text{gru})} + (\boldsymbol{1} - \boldsymbol{g}_t) \odot \boldsymbol{x}_t \tag{21}$$

It is possible to suggest yet other variants for such a model. In this paper, we limit ourselves to the model described in the above equations. We denote this model GRU-HW in the experimental section. This extension can also be applied to the LSTM in the same manner by using the output of the LSTM (Eq. 10) instead of Eq. 19. In the experimental section, we focus on the GRU version.

4. Speech Recognition Experiments

Our experiments were conducted on the English broadcast news and conversation speech recognition task from the Quaero project [28].

4.1. Baseline system description

The baseline ASR system used for this work is the same as in our previous work [29]. For acoustic modeling, a hybrid 12-layer rectified linear unit based feedforward network was trained. The model was first multilingually initialized [30] on 4 languages (French, English, German, Polish) on the total amount of 800 hours of speech, then fine-tuned with the 250 hours of English data. The minimum phone error sequence level discriminative training criterion was used in the final step. The baseline n-gram count language model is the same as in [29, 31]: 4-gram model with Kneser-Ney smoothing (KN4) was trained on the total of 3.1B of running words, with a vocabulary size of 150k. The 3.1B data was composed of 11 sub-corpora. Small LMs were trained on each of the subcorpora and combined into a single model. The interpolation weights were optimized on the development text using the SRILM toolkit [32]. The development and evaluation texts contain 40k and 36k running words, respectively. For further details, we refer to [29].

4.2. Neural network-based language models (NLMs)

All language models based on neural networks were trained on 50M running words. The 50M data are the in-domain subsets of the full 3.1B data. The resulting lexicon size for NLMs is 128k; a renormalization is therefore done for interpolation with the KN4 [33]. Again, this setup is the same as in [29, 31]. In the 50M corpus, the 2M most in-domain set is included. Following the recipe from [29], we also performed a fine-tuning on that 2M data (indicated when used). The models were trained using the stochastic gradient descent with mini-batches of size 64 for feedforward models, while recurrent models were trained with backpropagation through time without truncation with mini-batch size of 4. The output layer of NLMs was always factorized with 1000 word-classes as in [31]. All models presented in this paper were implemented as extension to the `rwthlm` toolkit [34]. Except the largest LSTM (3-layer 600-node model in Sec. 4.3.2) which was trained on GPU with a batch size of 8, models were trained using multithreading on CPU.

4.3. Text-based Results

4.3.1. Gates in MLP based models

We trained 20-gram MLP models with projection layer of 300 units per word and multiple stacked hidden layers of 600 units. Unlike in [29], we used neither layer-wise training nor low-rank factorization. The logistic function (Sigm) was used for all models as an activation function. The exponential linear unit (ELU) [35] was also tested for the baseline MLP. All models were fine-tuned (Sec 4.2). Table 1 shows the performance of different layer types for models with 2 layers. The first layer after the projection layer of Sigm-HW is a standard MLP layer. The perplexities of all layer types were about the same except the lateral network which performed slightly better. In order to assess the effect of the highway connection in deep models, we increased the number of layers until five: Table 2 shows perplexities on the development set. First of all, we observed that the performance of baseline MLP (Sigm) saturated at four layers, while no degradation was observed for highway models (Sigm-HW) until five layers. Furthermore, the highway models performed 4% relative better than the baseline. The lateral network (Lateral) saturated with 3 layers and its best perplexity was slightly worse than that of the highway model. This result shows the importance of the highway linear connection, since the lateral network only differs from the highway network in that connection.

Table 1: *Comparison of different feedforward layer types. Perplexities are reported with 2-layer models on development set.*

	Sigm	Sigm-HW	ELU	Lateral
PPL	126.4	126.5	126.3	**123.4**

Table 2: *Effect of the depth. Perplexities on development set.*

Layer	2	3	4	5
Sigm	126.4	124.9	**124.6**	126.7
Sigm-HW	126.5	120.4	119.8	**119.7**
Lateral	123.4	**122.0**	122.2	-

4.3.2. LSTM and GRU

We carried out a comparison between LSTM and GRU. The results are shown in Table 3. The first observation is that for models with a hidden layer size of less than 200, GRUs performed slightly better than LSTMs. However, the LSTMs were found to benefit clearly better from larger widths than the GRUs: in the end, the best perplexity was obtained for a stacked, 2-layer LSTM. Besides, we observed that the improvements saturated after 2 layers for both architectures. The perplexities of 3-layer LSTM of size 300 and 600 are 110.0 and 103.0, respectively. The results for GRU can be seen in the first row of Table 5. To further improve the perplexities, a fine-tuning (Sec. 4.2) was applied to the best models: the results are shown in Table 4.

Table 3: *Perplexities on development set for LSTM and GRU. *The perplexities for 1- and 2-layer LSTMs are taken from* [31].

		LSTM		GRU	
	depth	1*	2*	1	2
size	100	147.0	139.6	143.9	136.4
	200	127.7	117.7	121.9	116.8
	300	117.6	109.1	115.7	110.7
	400	112.8	104.6	114.7	110.0
	500	109.2	101.8	112.6	**108.1**
	600	107.8	**100.5**	112.2	108.9

Table 4: *Effect of fine-tuning. Perplexities on development text.*

fine-tuning	LSTM	GRU
No	100.5	108.1
Yes	**98.3**	**104.7**

4.3.3. Highway network based on GRU

To assess the effect of highway connections in RNNs, we evaluated the highway network with GRU transformation (Sec 3.2). We stacked until four such layers. The perplexities are presented in Table 5. The standard GRU got degradations with more than two layers while GRU-HW allowed deeper structures and achieved a 4% rel. improvement from 110.7 to 106.3 for model with 300 nodes. Further improvements of 5% rel. were obtained with 500 nodes and 4 layers, from 104.7 to 99.1.

Table 5: *Perplexities on development set for GRU-HW.*

size	layer type	fine-tuning	depth		
			2	3	4
300	GRU	No	**110.7**	114.5	116.4
	GRU-HW		109.1	**106.3**	106.6
		Yes	105.5	**102.9**	103.3
500	GRU-HW	Yes	101.5	100.3	**99.1**

4.4. Lattice Rescoring Results

We performed the lattice rescoring [36] (implemented in `rwthlm` [34]) with neural language models, linearly interpolated with the baseline count model KN4. The word error rates (WER) were obtained after confusion network based decoding. Table 6 shows the word error rates and the perplexities of models after the interpolation. For the MLP based models, the improvements in perplexity due to the highway connections were preserved after interpolation on both development and evaluation sets. However, WER improvement was observed only on the development set. Despite comparable perplexities, the lateral network gave a slightly worse evaluation WER. For the recurrent models, the LSTM was found to be better than the GRU in terms of both perplexity and WER. Although the GRU achieved a close WER to the LSTM on the development set, the LSTM significantly outperformed the GRU on the evaluation set. Similar to MLP-based models, the improvements in perplexity related to the highway remained for GRU after the interpolation with the count model. Moreover, it achieved an improvement on the evaluation WER from 9.4% to 9.2%, which is more noteworthy than the effect of the highway in the MLP.

Table 6: *PPL and WER results of interpolated models. The best configurations found on the development set are indicated in parentheses.*

	DEV		EVAL	
	PPL	WER[%]	PPL	WER[%]
KN4	132.7	12.3	131.2	10.5
Sigm-MLP (600x3)	106.1	11.3	106.0	9.5
Sigm-HW (600x5)	103.9	11.2	103.1	9.5
Lateral (600x3)	104.8	11.3	104.5	9.7
LSTM (600x2)	**89.8**	**10.7**	**90.5**	**9.0**
GRU (500x2)	93.0	10.8	94.2	9.4
GRU-HW (500x4)	**90.7**	**10.6**	**91.4**	**9.2**

5. Attention for Learning Word Triggers

So far, we focused on the networks based on the gating mechanism. However, the use of multiplicative gate is not a unique method to give explicit meaning to parts of a neural network. The attention mechanism has been recently proposed to learn the relevance of its inputs at each prediction. An application of such an idea for language modeling also makes sense: certain words in context can be particularly relevant to predict some words, like (multi-)word triggers [23, 24] in the count-based approach. In this section, we present our initial work on learning word triggers with an attention mechanism.

5.1. Model descriptions

5.1.1. Attention layer

A minimalistic attention mechanism can be defined as a layer. The input at time t is the outputs of the previous layer over time $(\boldsymbol{x}_1, ..., \boldsymbol{x}_t)$. It computes a scalar score s_i for each context $\boldsymbol{x}_i$ (Eq. 22). The resulting score vector $\boldsymbol{s} = (s_1, ..., s_t)$ is then normalized (Eq. 23) and the output is computed as the weighted average of contexts (Eq. 24). The full equations are:

$$\forall i \in \{1, .., t\} \quad s_i = \boldsymbol{w}^{\intercal} \tanh(\boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{R}\boldsymbol{h}_{t-1} + \boldsymbol{b}) \tag{22}$$

$$\boldsymbol{\alpha} = \text{softmax}(\boldsymbol{s}) \tag{23}$$

$$\boldsymbol{h}_t = \sum_{i=1}^{t} \alpha_i \boldsymbol{x}_i \tag{24}$$

5.1.2. Neural word trigger models, a naive approach

We inserted such an attention layer to a simple model composed of 3 layers: projection, GRU and output layers. The attention layer can be inserted either between the projection and GRU or between GRU and output. Experiments showed that the latter model was not suited for the word trigger because in such a model, the attention layer exclusively used the latest output of GRU ($\alpha_t \approx 1$) which had seen the full context. Therefore, we focused on the former case in which the attention directly follows the projection layer. We used the attention limited on a local window [37]. Following the MLP model, we limited the attention on the 19 predecessor words.

^{6} **Thanks**^{10} for^{3} taking^{9} the^{2} time^{4} to^{3} **download**^{22} this^{5} **BBC**^{12} **radio**^{11} five^{4} live^{8} $\boxed{\textit{podcast}}$

\$^{22} In^{4} this^{7} **book**^{17} there^{7} are^{5} **things**^{13} that^{7} are^{5} **very**^{14} $\boxed{\textit{complicated}}$

Figure 1: *Examples of word triggers from development text. The words inside a box are target words. The numbers in exponent of the context words are the scores in percentage given by the model to predict the target word. Top trigger words are highlighted with* **bold** *font. $ denotes sentence begin token.*

5.2. Results

We considered a model with 300 nodes for each layer. The attention-based trigger model achieved a development perplexity of 157.6 after fine-tuning, which is better than the KN4 on the same amount of data (163.0), but much worse than the baseline GRU (110.7, fine-tuned from 115.7 in Table 3). Despite a relatively high global perplexity, qualitatively meaningful triggers could be observed in some sentences: examples are shown in Figure 1. Furthermore, contrary to the tendency of count-based triggers [24], we did not find the self-triggers to be common. While we found these results qualitatively interesting, the performance of this naive model was not satisfactory. The weak dependencies in the score function (Eq. 22) is likely to be the reason. Recently, a more sophisticated approach [38] has been shown to be successful in augmenting the LSTM language model with an attention mechanism.

6. Conclusions

We confirmed that the LSTM seems to be a better default choice for language modeling than the GRU. Besides, we observed two effects of the highway connection: it helps models to benefit better from the depth and it avoids degradations from unnecessary depths. These tendencies were observed for both MLPs and RNNs. For the recognition, the improvements by the highway connection were noteworthy for the GRU: such an extension might be interesting in other fields in which deep GRUs are used. Similar investigations must be conducted for the highway network based on the LSTM. Finally, we presented a simple neural word trigger model based on a minimalistic attention mechanism: it already showed some interesting qualitative results, while the performance was not satisfactory. We also plan to extend our research on more sophisticated attention models.

7. Acknowledgements

Partially supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract no. W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. The research leading to these results has also received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21) and the Marie Skłodowska-Curie grant agreement No. 644283.

8. References

[1] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016.

[2] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: a neural network for large vocabulary conversational speech recognition," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016.

[3] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–393, 1999.

[4] J. T. Goodman, "A bit of progress in language modeling," *Computer Speech & Language*, vol. 15, no. 4, pp. 403–434, 2001.

[5] M. Nakamura and K. Shikano, "A study of english word category prediction based on neural networks," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Glasglow, UK, May 1989, pp. 731–734.

[6] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 13, Denver, CO, USA, 2000, pp. 932–938.

[7] T. Mikolov, S. Kombrink, L. Burget, J. H. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 5528–5531.

[8] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling." in *Proc. Interspeech*, Portland, OR, USA, Sep. 2012, pp. 194–197.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[11] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *The Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2003.

[12] C. B. Miller and C. L. Giles, "Experimental comparison of the effect of order in recurrent neural networks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 04, pp. 849–872, 1993.

[13] K. Cho, B. v. M. C. Gulcehre, D. Bahdanau, F. B. H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. of Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[14] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," in *the Deep Learning workshop at Int. Conf. on Machine Learning (ICML)*, Lille, France, Jul. 2015.

[15] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, Dec. 2015, pp. 2368–2376.

[16] D. Yu, L. Deng, and F. Seide, "The deep tensor neural network with applications to large vocabulary speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 388–396, 2013.

[17] J. Devlin, C. Quirk, and A. Menezes, "Pre-computable multi-layer neural network language models," in *Proc. of Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, Sep. 2015, pp. 256–260.

[18] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Deep Learning workshop at Conf. on Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, Dec. 2014.

[19] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. of Int. Conf. on Machine Learning (ICML)*, Lille, France, Jul. 2015, pp. 2342–2350.

[20] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. Glass, "Highway long short-term memory RNNs for distant speech recognition," *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016.

[21] K. Yao, T. Cohn, K. Vylomova, K. Duh, and C. Dyer, "Depth-gated LSTM," *Presented at Jelinek Summer Workshop, arXiv preprint arXiv:1508.03790*, Aug. 2015.

[22] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[23] C. Tillmann and H. Ney, "Word triggers and the EM algorithm," in *Proc. Special Interest Group Workshop on Computational Natural Language Learning (ACL)*, Madrid, Spain, Jul. 1997, pp. 117–124.

[24] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modelling," *Computer Speech and Language*, vol. 10, no. 3, pp. 187–228, 1996.

[25] Y. Kim and Y. J. D. S. A. Rush, "Character-aware neural language models," in *Proc. AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, Feb. 2016.

[26] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. Int. Conf. on Machine Learning (ICML)*, vol. 28, Atlanta, GA, USA, Jun. 2013, pp. 1319–1327.

[27] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *arXiv preprint arXiv:1503.04069*, Mar. 2015.

[28] M. Nußbaum-Thom, S. Wiesler, M. Sundermeyer, C. Plahl, S. Hahn, R. Schlüter, and H. Ney, "The RWTH 2009 QUAERO ASR evaluation system for English and German," in *Proc. Interspeech*, Makuhari, Japan, Sep. 2010, pp. 1517–1520.

[29] Z. Tüske, K. Irie, R. Schlüter, and H. Ney, "Investigation on log-linear interpolation of multi-domain neural network language model," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016.

[30] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the Use of a Multilingual Neural Network Front-End," in *Proc. Interspeech*, Brisbane, Australia, Sep. 2008, pp. 2711–2714.

[31] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, Mar. 2015.

[32] A. Stolcke, "SRILM-an extensible language modeling toolkit." in *Proc. Interspeech*, Denver, CO, USA, 2002.

[33] M. Sundermeyer, I. Oparin, J.-L. Gauvain, B. Freiberg, R. Schlüter, and H. Ney, "Comparison of feedforward and recurrent neural network language models," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 8430–8434.

[34] M. Sundermeyer, R. Schlüter, and H. Ney, "rwthlm the RWTH Aachen University neural network language modeling toolkit," in *Proc. Interspeech*, Singapore, Sep. 2014, pp. 2093–2097.

[35] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. Int. Conf. on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016.

[36] M. Sundermeyer, Z. Tüske, R. Schlüter, and H. Ney, "Lattice decoding and rescoring with long-span neural network language models," in *Proc. Interspeech*, Singapore, Sep. 2014, pp. 661–665.

[37] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. of Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, Sep. 2015, pp. 1412–1421.

[38] K. Tran, A. Bisazza, and C. Monz, "Recurrent memory network for language modeling," in *Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, San Diego, CA, USA, Jun. 2016.