Supplementary material for Towards efficient end-to-end speech recognition with biologically-inspired neural networks

Thomas Bohnstingl[*]
IBM Research, Zurich
Graz University of Technology

Ayush Garg
IBM Research, Zurich
ETH Zurich

Stanisław Woźniak
IBM Research, Zurich

George Saon
IBM Research AI, Yorktown Heights

Evangelos Eleftheriou†
IBM Research, Zurich

Angeliki Pantazi
IBM Research, Zurich

1 Comparison of biologically-inspired units to LSTMs

As discussed in Section 1 of the main paper, LSTM units are commonly used in state-of-the-art machine learning networks for speech recognition. They have three distinct gates which mediate the input to the units, the decay of the internal state as well as the output of the units. In particular, a layer of n LSTM units with m inputs is governed by the following equations

$$\mathbf{i}^t = \boldsymbol{\sigma}(\mathbf{W}_i \boldsymbol{x}^t + \mathbf{H}_i \boldsymbol{y}^{t-1} + \mathbf{b}_i), \tag{1}$$

$$\boldsymbol{c}^t = \boldsymbol{\sigma}(\mathbf{W}_c \boldsymbol{x}^t + \mathbf{H}_c \boldsymbol{y}^{t-1} + \mathbf{b}_c) \tag{2}$$

$$\mathbf{f}^t = \boldsymbol{\sigma}(\mathbf{W}_f \boldsymbol{y}^t + \mathbf{H}_f \boldsymbol{y}^{t-1} + \mathbf{b}_f), \tag{3}$$

$$\boldsymbol{s}^t = \mathbf{f}^t \odot \boldsymbol{s}^{t-1} + \mathbf{i}^t \odot \mathbf{tanh}(\mathbf{W}_s \boldsymbol{y}^t + \mathbf{H}_s \boldsymbol{y}^{t-1} + \mathbf{b}_s), \tag{4}$$

$$\boldsymbol{y}^t = \mathbf{c}^t \odot \mathbf{tanh}(s^t). \tag{5}$$

We found empirically that the standard LIF dynamics lacks such functionalities and thus has intrinsic limitations in tackling ASR in comparison with LSTMs. For example, the dynamics of the output gate in LSTMs is governed by Equation 2, where $\mathbf{W}_c \in I\!R^{n \times m}$, $\mathbf{H}_c \in I\!R^{n \times n}$, $\mathbf{b}_c \in I\!R^n$ are trainable parameters. The novel variants of SNUs introduced in Section 2 of the main paper, exploit additional dynamics beyond the common LIF model, achieved via the axo-somatic and axo-axonic synapses. In particular, the threshold adaptation of the SNU-a, as represented in Equation 4 of the main paper, controls the output of the neuron by increasing or decreasing the firing threshold. By comparing the output mechanisms of the LSTM to the SNU-a

$$\text{Output gate LSTM:} \quad \boldsymbol{\sigma}(\mathbf{W}_c \boldsymbol{x}^t + \mathbf{H}_c \boldsymbol{y}^{t-1} + \mathbf{b}_c),$$

$$\text{Adaptive threshold SNU-a:} \quad \beta \boldsymbol{b}^t,$$

with

$$\boldsymbol{b}^t = \boldsymbol{\rho} \odot \boldsymbol{b}^{t-1} + (1 - \boldsymbol{\rho}) \odot \left(\mathbf{W}_a \boldsymbol{x}^t + \mathbf{H}_a \boldsymbol{y}^{t-1}\right),$$

one can see that the adaptive threshold mechanism is different from the LSTM gate. The threshold adaptation presents a low-pass filter, with a constant decay of $\boldsymbol{\rho}$, of the input activity and the recurrent activity multiplied with a trainable matrix. Moreover, its relation to the neuronal output is additive, whereas for the output gates in LSTM units it is multiplicative, see Equations 5 of the main paper and Equation 5 of these notes.

[*]Correspondence to boh@zurich.ibm.com
† Currently with Axelera AI, Zurich

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

In contrast, the output modulating mechanism of the SNU-o, mimicking the axo-axonic synapses, is more closely related to the LSTM output gate. This becomes apparent when comparing the relevant equations for the LSTM units and SNU-o units:

$$\begin{aligned} \text{Output gate LSTM:} \quad & \boldsymbol{\sigma}(\mathbf{W}_c \boldsymbol{x}^t + \mathbf{H}_c \boldsymbol{y}^{t-1} + \mathbf{b}_c), \\ \text{Ouput modulation SNU-o:} \quad & \boldsymbol{\sigma}\left(\mathbf{W}_o \boldsymbol{x}^t + \mathbf{H}_o \boldsymbol{y}^{t-1} + \boldsymbol{b}_o^t\right). \end{aligned}$$

It is important to mention that although the output modulating mechanism of the SNU-o resembles closer the behavior of the LSTM output gate, the former mechanism has a different background and is inspired from the existence of different synapse types in the human brain.

2 Details of the RNN-T architecture

As described in the main paper, we utilize a state-of-the-art RNN-T network and redesign it incorporating biologically-inspired units. Broadly speaking, the RNN-T consists of two main network components leveraging RNNs, the encoding network and the prediction network.

The encoding network is responsible for the feature encoding and processes the MFCCs, denoted with $\boldsymbol{x}^t = \boldsymbol{x}^0, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^{T-1} \in I\!R^{n_{MFCC}}$, where n_{MFCC} is the number of mel frequency cepstral coefficients and T is the length of the input sequence. The encoding network uses k bidirectional layers, we use $k = 6$ in our simulations. The second part, the prediction network, acts as a language model and processes the output produced thus far by the RNN-T without the blank symbols, i.e., $\boldsymbol{y}^u = \boldsymbol{y}^0, \boldsymbol{y}^1, \ldots, \boldsymbol{y}^U \in I\!R^{(n_{voc}+1)}$. Here n_{voc} is the size of the vocabulary which in our case consists of 45 individual characters, U is the length of the final output sequence, which might be different from the input sequence length T and the initial input to the prediction network $\boldsymbol{y}^0$ is always the blank symbol. Note that the prediction network is composed of an embedding layer followed by a layer of recurrent neural units. The outputs of the encoding network, the acoustic embedding, $\boldsymbol{h}_{enc}^t \in I\!R^{n_{enc}}$, where n_{enc} is the number of units in the last layer of the encoder, and the output of the prediction network, the prediction vector $\boldsymbol{h}_{pred}^u \in I\!R^{n_{pred}}$ where n_{pred} is the number of units in the last layer of the prediction network, are expanded, combined together via a Hadamard product and then further processed by a $\mathbf{tanh}$ activation and softmax operation. The final result, $\boldsymbol{h}_{joint}^{t,u} \in I\!R^{T \times U \times (n_{voc}+1)}$, is then used to compute the output distribution and in turn the most probable input-output alignment using the Forward-Backward algorithm as proposed in [1]. Note that the output of the joint network may contain a special blank symbol that allows for alignment of the speech signal with the transcript. This symbol gets removed from the final prediction of the RNN-T network.

3 Data preprocessing and simulation details

In our work we investigate the Switchboard speech corpus, which is a widely adopted dataset of roughly 300 hours of English two-sided telephone conversations on predefined topics. In particular, the dataset contains speech from a total of 543 speakers from different areas of the United States and is licensed under the Linguistic Data Consortium [2].

Initially, four data augmentations are applied to the original dataset in which the speed as well as the tempo of the conversation are increased and decrease by a value of 1.1 and 0.9, respectively. Then, a 40-dimensional MFCC vector is extracted every 10 ms and extended with the first and second order derivatives, yielding a 120-dimensional vector. Next, a time reduction technique is applied that involves stacking consecutive pairs of frames, resulting in a 240-dimensional vector. Additionally, the extracted features are combined with speaker-dependent vectors, called i-vectors [3], to form a 340-dimensional input used for neural network training.

As highlighted in the main paper, the two network components of the RNN-T are responsible to carry out different tasks, hence different sSNU variants might be better suited for application in one or the other. Therefore, we investigated various configurations of our novel biologically-plausible variants.

The training is accomplished using the AdamW [4] optimizer with a one-cycle learning rate schedule [5], where the maximum learning rate η_m has been determined for each run individually. We trained for 20 epochs wherein the learning rate was linearly ramped up from an initial value to a maximum value within the first six epochs and linearly ramped down to a minimum value in the

Table 1: RNN acronyms used in the result tables and details on the included parameters.

RNN *Suffix*	Comment	Thr.	Axo-dendritic	Axo-somatic	Axo-axonic
sSNU	Feedforward	$\mathbf{b}$	$\mathbf{W}$		
sSNU *R*	Recurrent	$\mathbf{b}$	$\mathbf{W}, \mathbf{H}$		
sSNU-a	Adaptive thr. feedforward	$\mathbf{b_0}$	$\mathbf{W}$		
sSNU-a *R*	Adaptive thr. recurrent	$\mathbf{b_0}$	$\mathbf{W}, \mathbf{H}$		
sSNU-a *Ra*	Adaptive thr. axo-somatic recurrent	$\mathbf{b_0}$	$\mathbf{W}, \mathbf{H}$	$\mathbf{H_a}$	
sSNU-o	Output modulating feedforward	$\mathbf{b}$	$\mathbf{W}$		$\mathbf{W_o}, \mathbf{b_o}$
sSNU-o *R*	Output modulating recurrent	$\mathbf{b}$	$\mathbf{W}, \mathbf{H}$		$\mathbf{W_o}, \mathbf{H_o}, \mathbf{b_o}$

subsequent 14 epochs, similar to [6]. Table 1 lists explicitly the trainable parameters for different sSNU variants along with their abbreviations. We trained with a batch size of 64 on two V100-GPUs for approximately 10 days. To avoid overfitting we employed gradient clipping, in which the gradients are combined to a single vector $\boldsymbol{w}$ and the individual components are then computed as

$$\tilde{\boldsymbol{w}} = \boldsymbol{w} \odot \frac{c}{||\boldsymbol{w}||_2},$$

with $c = 1$ or $c = 10$. In addition, we use dropout with a dropout probability of $p_W = 0.25$ for all the input weights and $p_E = 0.05$ for the embedding layer.

The final speech transcript was produced using beam search with a beam width of 16. For the evaluation of our models, we followed the common procedure to report the word error rates (WER) on the Hub5 2000 Switchboard and the CallHome test set jointly [7; 6]. As the baseline for our benchmark, we re-implemented the very recent state-of-the-art results from [6]. We focused primarily on the dynamics of the neurons and thus used a basic model implementation without any external language model. Such an LSTM-based RNN-T architecture achieves a WER of $12.7\,\%$, which we consider as our baseline. However, it is worth mentioning that the research field of speech recognition is very active and although our selected baseline is representative of the state-of-the-art, it can potentially be improved with the features mentioned in [6].

As mentioned in the main paper, we followed a three-step approach to integrate our biologically-inspired units into the RNN-T network architecture. In addition to Section 4 of the main paper, the Tables 2, 3 and 4 contain more detailed hyperparameter settings of our simulations.

Table 2: Hyperparameters of the prediction network

RNN	Cfg.	η_m	c	Additional
LSTM	1x768	$5 \cdot 10^{-4}$	10	
sSNU	1x768	$5 \cdot 10^{-4}$	10	$d = 0.9$
sSNU *R*	1x768	$5 \cdot 10^{-4}$	10	$d = 0.9$
sSNU-a	1x768	$5 \cdot 10^{-4}$	10	$d = 0.9, \beta = 0.1, \rho = 0.9$
sSNU-a *R*	1x768	$5 \cdot 10^{-4}$	10	$d = 0.9, \beta = 0.1, \rho = 0.9$
sSNU-o	1x768	$5 \cdot 10^{-4}$	10	$d = 0.9$
sSNU-o *R*	1x768	$5 \cdot 10^{-4}$	10	$d = 0.9$
SNU	1x768	$5 \cdot 10^{-4}$	10	$d = 0.9$

Table 3: Hyperparameters of the encoding network

RNN	Cfg.	η_m	c	Additional
LSTM	6x640	$5 \cdot 10^{-4}$	10	
sSNU-a *Ra*	6x640	$5 \cdot 10^{-4}$	1	$d = 0.9, \beta = 0.1, \rho = 0.9$
sSNU-o	6x640	$5 \cdot 10^{-4}$	10	$d = 0.9$
sSNU-o *R*	6x640	$9 \cdot 10^{-4}$	1	$d = 0.9$

Table 4: Hyperparameters of the full RNN-T network

Encoding network			Prediction network				
RNN	Cfg.	Additional	RNN	Cfg.	Additional	η_m	c
LSTM	1x768		LSTM	6x640		$5 \cdot 10^{-4}$	10
sSNU-o *R*	6x640	$d = 0.9$	sSNU-a *R*	1x768	$d = 0.9, \beta = 0.1, \rho = 0.9$	$9 \cdot 10^{-4}$	10
sSNU-o *R*	1x768	$d = 0.9$	sSNU-o *R*	6x640	$d = 0.9$	$5 \cdot 10^{-4}$	1

References

[1] A. Graves, "Sequence Transduction with Recurrent Neural Networks," *arXiv*, Nov 2012.

[2] "Switchboard-1 Release 2 - Linguistic Data Consortium," May 2021. [Online; accessed via `https://catalog.ldc.upenn.edu/LDC97S62` on 2. May 2021].

[3] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Twelfth annual conference of the International Speech Communication Association*, 2011.

[4] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *arXiv*, Nov 2017.

[5] L. N. Smith and N. Topin, "Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates," *arXiv*, Aug 2017.

[6] G. Saon, Z. Tueske, D. Bolanos, and B. Kingsbury, "Advancing RNN Transducer Technology for Speech Recognition," *arXiv*, Mar 2021.

[7] "2000 HUB5 English Evaluation Transcripts - Linguistic Data Consortium," Jan 2002. [Online; accessed via `https://catalog.ldc.upenn.edu/LDC2002T43` on 2. May 2021].