INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Internal Memory Gate for Recurrent Neural Networks with Application to Spoken Language Understanding

Mohamed Morchid

LIA, University of Avignon (France)
mohamed.morchid@univ-avignon.fr

Abstract

Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN) require 4 gates to learn short- and long-term dependencies for a given sequence of basic elements. Recently, "Gated Recurrent Unit" (GRU) has been introduced and requires fewer gates than LSTM (reset and update gates), to code short- and long-term dependencies and reaches equivalent performances to LSTM, with less processing time during the learning. The "Leaky integration Unit" (LU) is a GRU with a single gate (update) that codes mostly long-term dependencies quicker than LSTM or GRU (small number of operations for learning). This paper proposes a novel RNN that takes advantage of LSTM, GRU (short- and long-term dependencies) and the LU (fast learning) called "Internal Memory Gate" (IMG). The effectiveness and the robustness of the proposed IMG-RNN is evaluated during a classification task of a small corpus of spoken dialogues from the DECODA project that allows us to evaluate the capability of each RNN to code short-term dependencies. The experiments show that IMG-RNNs reach better accuracies with a gain of 0.4 points compared to LSTM- and GRU-RNNs and 0.7 points compared to the LU-RNN. Moreover, IMG-RNN requires less processing time than GRU or LSTM with a gain of 19% and 50% respectively.

Index Terms: Recurrent neural network, Long short-term memory, Spoken language understanding

1. Introduction

Recurrent neural networks (RNNs) encountered a wide success in the last decay in many Speech and Language Processing related applications, such as statistical parametric speech synthesis [1, 2], speech emotion recognition [3, 4] and speech recognition [5]. RNNs with gates such as Long Short-Term Memory [6] (LSTM) avoid the vanishing gradient problem of the RNN [7]. Even if the results of LSTM-RNNs are promising, the processing time required to treat large data sets is quite huge due to the different gates activation sub-processes. A recent addition to the RNN set of models called the Gated Recurrent Unit (GRU) has been proposed by [8] to address this issue. GRU is very similar to LSTM, in that it uses a combination of gates to adjust exposure from input to the hidden states. The good results obtained by the GRU-RNN (almost equivalent to LSTM) are coupled to a gain in terms of processing time. The Leaky Integration Unit (LU) [9] is equivalent to the GRU with a single gate and requires less processing time for learning than GRU and LSTM. Nonetheless, the LU codes mostly long-term dependencies but reveals little in way of short-term dependencies. This paper proposes a novel recurrent unit called "Internal Memory Gate" (IMG) that takes advantage from the hitherto-proposed RNN units (LSTM and GRU) as well as from the LU. Indeed, IMG-RNN is able to code both long-term dependencies with a single gate and short-term dependencies with an internal recurrent loop in this gate. The paper shows the effectiveness of the proposed IMG-RNN during a theme identification task of a small set of spoken dialogues from the DECODA corpus [10] that reveals short-term dependencies (long-term dependencies are managed in the same way as LSTM/GRU). Moreover, the experiments demonstrate that the proposed IMG-RNN requires less processing time than LSTM and GRU.

The rest of the paper is organized as follows: Section 2 introduces the RNNs models and Section 3 details the proposed IMG-RNN. The experimental protocol is presented in Section 4 and the results are discussed in Section 5. Section 6 concludes this paper.

2. Recurrent Neural Networks

Figure 1 presents the recurrent neural network (RNN) folded (a) and unrolled (b). The next sections introduce different recurrent units to evaluate the hidden state h_t: $\tanh$, the Long Short-Term Memory (LSTM), the Gated Recurrent Units (GRU) and the Leaky Integration Unit (LU). For brevity, the biases are ignored.

2.1. Basics of Recurrent Neural Networks

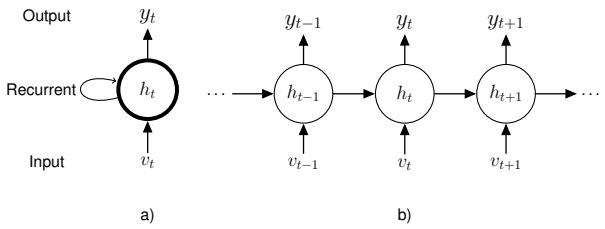


Figure 1: *Illustration of (a) an Recurrent Neural Network (RNN) with loops and (b) an unrolled RNN presented as a chain-like that reveals the link between sequences throughout the time.*

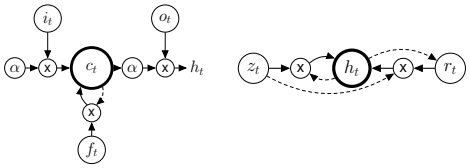
The Recurrent Neural Network (RNN) is an extension of a mere feedforward neural network that can handles a variable-length sequence as input vector v. The activation of the j-th hidden state h_j depends on the previous hidden state at each time-step t, and is computed as a state-to-state transition function ϕ (point-wise activation function, such as the logistic function or the Rectified Linear Unit [11, 12]). $h_j(t)$ and the output

http://dx.doi.org/10.21437/Interspeech.2017-357

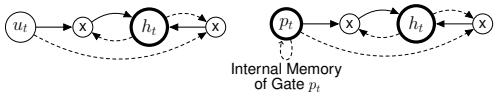
unit $y_k(t)$ (same $y_k(t)$ for all units) are computed as:

$$h_j(t) = \phi(\sum_m s_{h_j m} v_m(t) + w_{h_j m} h_m(t-1)) \quad (1)$$
$$y_k(t) = \sigma(\sum_m s_{y_k m} v_m(t) + w_{y_k m} h_m(t-1))$$

where w_{lm} and s_{lm} (from unit m to l) are the state-to-state recurrent and input-to-hidden weight connections respectively; σ is the logistic sigmoid function. Then, a RNN is trained to predict the next symbol v_{t+1} given the hidden states h_t (equation (1)). Given a current symbol v_t and the previous ones $v_1, \ldots, v_{t-1}$: $p(v_{t+1}|v_1, \ldots, v_t) = \phi(h_j(t))$. [13, 7] expose the difficulty for a RNN to capture long-term dependencies due to the vanishing/exploding gradient problem addressed by the LSTM [14].



(a) Long Short-Term Memory (b) Gated Recurrent Unit



(c) Leaky Integration Unit (d) Internal Memory Gate

Figure 2: *Graphical illustrations of (a) LTSM, (b) GRU, (c) Leaky Unit and (d) Internal Memory Gate (IMG). Dashed lines is connections with time-lag.*

2.2. Long Short-Term Memory (LSTM)

The goal of LSTM [14] is to create an internal cell state which allows it to exhibit dynamic temporal behavior and to process arbitrary sequences of inputs, such as sequences of words [6] for language modeling or time series [15]. The basic unit is a memory block that replaces the hidden unit in the mere RNN. Each cell has a forget gate f_j to learn to reset memory blocks ; an input gate i_j that controls how much old content has to be memorized; an output gate o_j that controls the degree the memory content is exposed for the j-th memory block (see Figure 2 (a)):

$$i_j(t) = \sigma(\sum_m s_{i_j m} v_m(t) + w_{i_j m} h_m(t-1))$$
$$o_j(t) = \sigma(\sum_m s_{o_j m} v_m(t) + w_{o_j m} h_m(t-1))$$
$$f_j(t) = \sigma(\sum_m s_{f_j m} v_m(t) + w_{f_j m} h_m(t-1)) \, .$$

The cell c_j and the hidden state h_t (eq. (1)) are:

$$c_j(t) = f_j(t) c_j(t-1) + i_j(t) \tilde{c}_j(t)$$
$$\tilde{c}_j(t) = \phi(\sum_m s_{c_j m} v_m(t) + w_{c_j m} h_m(t-1))$$
$$h_j(t) = o_j(t) \phi(c_j(t)) \, .$$

2.3. Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) [16] depicted in Figure 2 (b), is a LSTM-like adaptive "reset" and "update" memory unit. The LSTM input gate i is replaced in the GRU by an update gate z, and the forget gate f by a reset gate r for the GRU computed at time step t by:

$$r_j(t) = \sigma(\sum_m s_{r_j m} v_m(t) + w_{r_j m} h_m(t-1))$$
$$z_j(t) = \sigma(\sum_m s_{z_j m} v_m(t) + w_{z_j m} h_m(t-1))$$

where h_t defined in equation (1) and $\tilde{h}_j(t)$ are the hidden and the candidate hidden states defined as :

$$h_j(t) = \underbrace{(1 - z_j(t))}_{\text{long-term}} h_j(t-1) + \underbrace{z_j(t)}_{\text{long-term}} \tilde{h}_j(t)$$
$$\tilde{h}_j(t) = \phi(\sum_m s_{h_j m} v_m(t) + w_{h_j m} h_m(t-1) \underbrace{r_j(t)}_{\text{short-term}})$$

The main idea behind the GRU unit is to expose the memory content at each time step and to balance the previous and new hidden state using leaky integration.

2.4. Leaky Integration Unit (LU)

The Leaky Integration Unit (LU) [9] depicted in Figure 2 (c), is a GRU with the reset gate fixed to 1. The leaky gate u_t is computed at time step t by:

$$u_j(t) = \sigma(\sum_m s_{u_j m} v_m(t) + w_{u_j m} h_m(t-1))$$

and the hidden state h_t of eq. (1) is computed as:

$$h_j(t) = \underbrace{(1 - u_j(t))}_{\text{long-term}} h_j(t-1) + \underbrace{u_j(t)}_{\text{long-term}} \tilde{h}_j(t) \quad (2)$$
$$\tilde{h}_j(t) = \phi(\sum_m s_{h_j m} v_m(t) + w_{h_j m} h_m(t-1))$$

The LU learns long-term dependencies (see eq.(2)) such as the GRU-RNN. Indeed, the u_t gate plays the same role than the update gate z_t of the GRU by controlling how much information from the previous hidden state will carry over to the current hidden state.

3. Internal Memory Gate (IMG)

The LU reveals little in way of short-term dependencies and the proposed Internal Memory Gate (IMG) codes both short- (recurrent loop with the immediate previous gate state $p_j(t-1)$ in eq.(3)) and long-term ($p_j(t)$ in eq.(4)) dependencies during the learning. The gate p_t is computed at time step t by:

$$p_j(t) = \sigma(\sum_m s_{p_j m} v_m(t) + w_{p_j m} h_m(t-1) + g_{p_j m} \underbrace{p_j(t-1)}_{\text{short-term}}) \quad (3)$$

g is the weight matrix and the hidden state h_t (eq. (1)):

$$h_j(t) = \underbrace{(1 - p_j(t))}_{\text{long-term}} h_j(t-1) + \underbrace{p_j(t)}_{\text{long-term}} \tilde{h}_j(t) \quad (4)$$
$$\tilde{h}_j(t) = \phi(\sum_m s_{h_j m} v_m(t) + w_{h_j m} h_m(t-1))$$

Overall, the IMG (Figure 2 (d)) codes short-term dependencies and the next sections evaluate the effectiveness of the IMG-RNN during a theme identification task of a small corpus of spoken dialogues.

4. Experimental protocol

The effectiveness of the proposed IMG-RNN is evaluated in the application framework of the DECODA corpus [10] containing a small set of spoken dialogues to better-reveal short-term dependencies. It is composed of only $1,514$ telephone conversations, corresponding to about 74 hours of signal, split into a train set (740 dialogues), a development set (175 dialogues) and a test set (327 dialogues), and manually annotated with 8 conversation themes: *problems of itinerary*, *lost and found*, *time schedules*, *transportation cards*, *state of the traffic*, *fares*, *infractions* and *special offers*. Transcription of dialogues has been made by the LIA-Speeral ASR system [17] and detailed in [18]. The LSTM, GRU, LU and the IMG are composed with 3 layers: an input layer **v** of size equal to the training corpus vocabulary $(5,782)$, a hidden layer **h** of size varying from 10 to 100 and an output layer **y** with a size equals to the number of themes (8). To better underline the difference between RNNs in terms of processing time (small inputs and hidden layers), RNNs are implemented on a CPU card by the author.

5. Experiments and Results

Section 5.1 evaluates the accuracies reached by the RNN models, Section 5.2 compares the processing times during the learning of the RNNs and Section 5.3 sums up the results observed.

5.1. IMG *vs.* LSTM and GRU in terms of accuracy

Figure 3 presents the accuracies reached by each RNN units and the X-axis represents the number of neurons contained in the hidden layer (from 10 to 100).
It is worth underlining first that the best accuracy reached whatever the RNN employed, is equal to 86.9% (86.3% for the LU) on the development data set. This observation is verifiable with the minimum accuracies reached for all RNN-based models ($\simeq 76\%$). The right column of Figure 3 ((b-d-f-h) figures) depicts the accuracies obtained with the test data set. In green is presented the best accuracies and in blue the "real" accuracies [1]. One can easily point out that the best accuracies reached by all RNNs are equivalent for the test data set ($\simeq 82.6$). In spite of the fact that the best accuracy observed for the proposed IMG moves down (82.3%), this accuracy is appears three times ($|h_t| = \{70; 75; 100\}$) on the development data set (Figure 3 (h)). Therefore, the proposed IMG is more robust to RNN configurations variation than other RNNs, to code short-term dependencies with a high accuracy. In "real" conditions, the LSTM and GRU obtain a better accuracy (81.9% in blue in Figure 3) than the LU (81.6%). The small corpus of spoken dialogues is better-classified with the proposed IMG-RNN model with an accuracy of 82.3%. Moreover, the IMG requires fewer neurons in the hidden layer than the LSTM/GRU/LU-RNNs. LU reaches the worst accuracy but requires less basic operations during the learning than the other RNNs.

[1] Best configuration observed on development set (number of neurons in the hidden layer) applied to test data set.

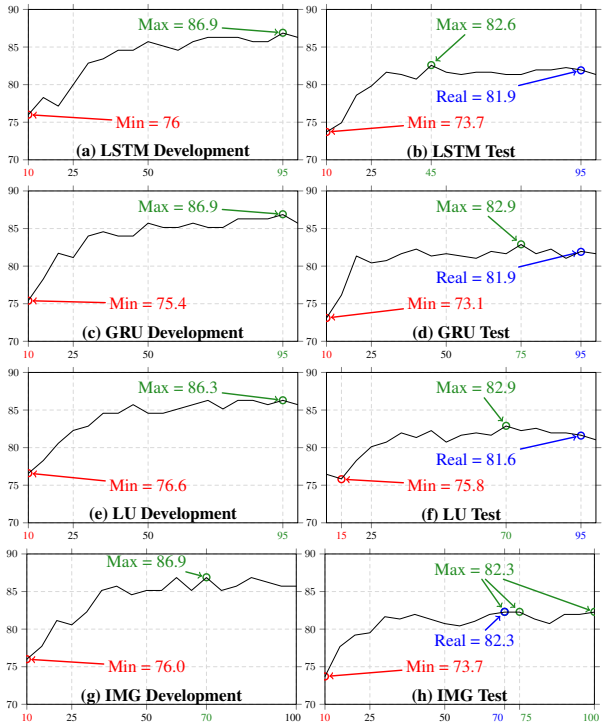


Figure 3: *Spoken dialogues classification* **accuracies** *(%) from the* **DECODA** *corpus by varying the number of neurons in the hidden layer (from 10 to 100 in the X-axis) with LSTM, GRU, LU and the proposed IMG.*

5.2. Processing time required for learning RNNs

Figure 4 presents the processing time (PT) for LSTM, GRU, LU and IMG-RNNs by varying (X-axis) the number of neurons in the hidden layer. It is worth emphasizing that the

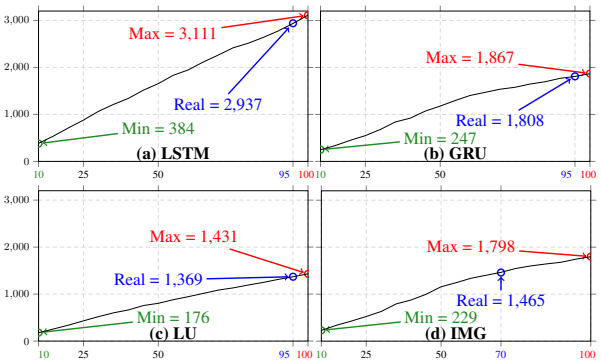


Figure 4: **Processing time** *(seconds) during the learning of LSTM, GRU, LU and IMG for the theme identification task of dialogues from the* **DECODA** *corpus.*

LSTM requires the largest PT with a maximum$= 3,111$ seconds and minimum=384 sec. with 100 and 10 hidden neurons respectively. The GRU is the second slowest RNN model (max$= 1,867$ sec.), then the IMG (max$= 1,798$ sec.) and finally the LU (max$= 1,431$ sec.). In real conditions (blue in

Figure 4), the LU-RNN requires the shortest PT ($1,369$ sec.) compared to the LSTM ($2,937$ sec. with a gain of 54%), GRU ($1,808$ sec. with a gain of 24%) and IMG ($1,465$ sec. with a small gain of only 7%). Nonetheless, the IMG requires a shorter processing time compared to the LSTM and GRU with a gain in "real" conditions of 50% and 19% respectively.
Overall, the proposed IMG unit needs more PT than the LU but less than LSTM or GRU that also code both short and long-term dependencies. Moreover, the IMG requires fewer neurons in the hidden layer than the LSTM-, GRU- and even the LU-RNNs.

5.3. Summary of the experiments

Table 1 shows that the IMG reaches the best accuracy during the theme identification task of a small corpus of spoken dialogues (short-term dependencies).

Table 1: *Theme classification accuracies, number of iterations and processing time (seconds).*

RNN model	hidden layer size	Real Test acc. %	Proc. time in seconds
LSTM	95	81.9	2,937
GRU	95	81.9	1,808
Leaky Unit	95	81.6	**1,369**
IMG	**70**	**82.3**	1,465

The few number of hidden neurons (70) for IMG-RNN compared to the other RNNs (95) validates the initial intuition that the IMG better-catch short-term dependencies with a more robust latent representation of dialogue content (three best accuracies reached in Figure 3-(h)). LU needs less PT ($1,369$ sec.) compared to the IMG-RNN but with a worse accuracy (81.6%).

6. Conclusions

Summary. This paper presents a novel RNN unit called "Internal Memory Gate" (IMG) based on the assumption that a single gate can deal with both short- and long-term dependencies. The long-term dependencies are managed in the same manner as the GRU, while the short-term dependencies are considered with a internal recurrent loop. IMG reaches better results than state-of-the-art RNNs during a theme identification task of a small corpus of spoken dialogues that reveals short-term dependencies, and validates the initial assumption that IMG-RNNs better manages short-term dependencies than GRU/LSTM. Moreover, we demonstrate that the IMG-RNNs are more robust to unseen documents and require fewer hidden units and less processing time compared to the other RNNs (except for the LU but with a worse accuracy).
Limitations and Future Work. Our results reveal the promise of IMG-RNN to code short-term dependencies. Our model is far from perfect, and more work is needed to explore the impact of long-term dependencies. Future work will use a much larger corpus of documents such as the 20-NewsGroups [19] to exhibit long-term dependencies and evaluate also the impact of the IMG in terms of processing time. We hope to show similar results on such larger datasets.

7. References

[1] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks." in *Interspeech*, 2014, pp. 2268–2272.

[2] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks." in *Interspeech*, 2014, pp. 1964–1968.

[3] L. Chao, J. Tao, M. Yang, Y. Li, and Z. Wen, "Long short term memory recurrent neural network based multimodal dimensional emotion recognition," in *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*. ACM, 2015, pp. 65–72.

[4] S. Chen and Q. Jin, "Multi-modal dimensional emotion recognition using recurrent neural networks," in *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*. ACM, 2015, pp. 49–56.

[5] A. Graves, "Neural networks," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012, pp. 15–35.

[6] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling." in *INTERSPEECH*, 2012, pp. 194–197.

[7] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[8] 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."

[9] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8624–8628.

[10] F. Bechet, B. Maza, N. Bigouroux, T. Bazillon, M. El-Beze, R. De Mori, and E. Arbillot, "Decoda: a call-centre human-human spoken conversation corpus." LREC'12, 2012.

[11] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 142–150.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[13] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

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

[15] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying lstm to time series predictable through time-window approaches," in *Artificial Neural NetworksICANN 2001*. Springer, 2001, pp. 669–676.

[16] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[17] G. Linarès, P. Nocéra, D. Massonie, and D. Matrouf, "The lia speech recognition system: from 10xrt to 1xrt," in *Text, Speech and Dialogue*. Springer, 2007, pp. 302–308.

[18] M. Morchid, R. Dufour, and G. Linarès, "Impact of word error rate on theme identification task of highly imperfect human–human conversations," *Computer Speech & Language*, vol. 38, pp. 68–85, 2016.

[19] S. D. Kamvar, D. Klein, and C. D. Manning, "Incremental spectral classification for weakly supervised text learning." [Online]. Available: http://www.ai.mit.edu/jrennie/20Newsgroups/.