Skip RNN: Learning to Skip State Updates in Recurrent Neural Networks

Víctor Campos*, Brendan Jou†, Xavier Giró-i-Nieto‡, Jordi Torres*, Shih-Fu Chang§
*Barcelona Supercomputing Center, †Google Inc, ‡Universitat Politècnica de Catalunya, §Columbia University
{victor.campos, jordi.torres}@bsc.es, bjou@google.com,
xavier.giro@upc.edu, shih.fu.chang@columbia.edu

***Abstract*—Recurrent Neural Networks (RNNs) continue to show outstanding performance in sequence modeling tasks. However, training RNNs on long sequences often face challenges like slow inference, vanishing gradients and difficulty in capturing long term dependencies. In backpropagation through time settings, these issues are tightly coupled with the large, sequential computational graph resulting from unfolding the RNN in time. We introduce the Skip RNN model which extends existing RNN models by learning to skip state updates and shortens the effective size of the computational graph. This model can also be encouraged to perform fewer state updates through a budget constraint. We evaluate the proposed model on various tasks and show how it can reduce the number of required RNN updates while preserving, and sometimes even improving, the performance of the baseline RNN models. Source code is publicly available at https://imatge-upc.github.io/skiprnn-2017-telecombcn/.**

***Keywords*—Deep Learning, Recurrent Neural Networks, Adaptive Computation.**

I. Introduction

Some of the main limitations of Recurrent Neural Networks (RNNs) are their challenging training and deployment when dealing with long sequences, due to their inherently sequential behaviour. These challenges include throughput degradation, slower convergence during training and memory leakage, even for gated architectures [10]. The main contribution of this work is Skip RNN, a novel modification for existing RNN architectures that allows them to skip state updates, decreasing the number of sequential operations to be performed, without requiring any additional supervision signal. The proposed modification is implemented on top of well known RNN architectures, namely LSTM and GRU, and the resulting models show promising results in a series of sequence modeling tasks.

II. Model Description

An RNN takes an input sequence $\mathbf{x} = (x_1, \dots, x_T)$ and generates a state sequence $\mathbf{s} = (s_1, \dots, s_T)$ by iteratively applying a parametric state transition model S from $t = 1$ to T:

$$s_t = S(s_{t-1}, x_t) \tag{1}$$

We augment the network with a binary *state update gate*, $u_t \in \{0, 1\}$, selecting whether the state of the RNN will be updated or copied from the previous time step. At every time step t, the probability $\tilde{u}_{t+1} \in [0, 1]$ of performing a state update at $t + 1$ is emitted. The model formulation implements the observation that the likelihood of requesting a new input increases with the number of consecutively skipped samples:

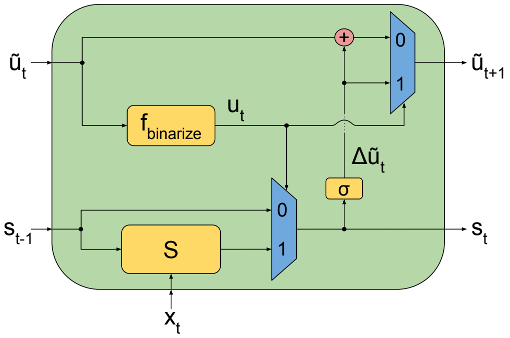

Fig. 1. Model architecture of the proposed Skip RNN, where the computation graph at time step t is conditioned on u_t. In practice, redundant computation is avoided by propagating $\Delta\tilde{u}_t$ between time steps when $u_t = 0$.

$$u_t = f_{binarize}(\tilde{u}_t) \tag{2}$$
$$s_t = u_t \cdot S(s_{t-1}, x_t) + (1 - u_t) \cdot s_{t-1} \tag{3}$$
$$\Delta\tilde{u}_t = \sigma(W_p s_t + b_p) \tag{4}$$
$$\tilde{u}_{t+1} = u_t \cdot \Delta\tilde{u}_t + (1 - u_t) \cdot (\tilde{u}_t + \min(\Delta\tilde{u}_t, 1 - \tilde{u}_t)) \tag{5}$$

where σ is the sigmoid function and $f_{binarize} : [0, 1] \to \{0, 1\}$ binarizes the input value. We implement $f_{binarize}$ as a deterministic step function $u_t = \text{round}(\tilde{u}_t)$ and use the straight-through estimator [5] to propagate gradients through it. The number of skipped time steps can be computed ahead of time, enabling more efficient implementations where no computation at all is performed whenever $u_t = 0$

There are several advantages in reducing the number of RNN updates. From the computational standpoint, fewer updates translates into fewer required sequential operations to process an input signal, leading to faster inference and reduced energy consumption. Unlike some other models that aim to reduce the average number of operations per step [10], [6], ours enables skipping steps completely. Replacing RNN updates with copy operations increases the memory of the network and its ability to model long term dependencies even for gated units, since the exponential memory decay observed in LSTM and GRU [10] is alleviated. During training, gradients are propagated through fewer updating time steps, providing faster convergence in some tasks involving long sequences. Moreover, the proposed model is orthogonal to recent advances in RNNs and could be used in conjunction with such techniques,

Model	Accuracy	State updates
LSTM	0.910 ± 0.045	784.00 ± 0.00
LSTM ($p_{skip} = 0.5$)	0.893 ± 0.003	392.03 ± 0.05
Skip LSTM, $\lambda = 10^{-4}$	0.973 ± 0.002	379.38 ± 33.09
GRU	0.968 ± 0.013	784.00 ± 0.00
GRU ($p_{skip} = 0.5$)	0.912 ± 0.004	391.86 ± 0.14
Skip GRU, $\lambda = 10^{-4}$	0.976 ± 0.003	392.62 ± 26.48

TABLE I. ACCURACY AND USED SAMPLES ON THE TEST SET OF MNIST. RESULTS ARE DISPLAYED AS $mean \pm std$ OVER FOUR DIFFERENT RUNS.

e.g. normalization [3], [1], regularization [12], [7], variable computation [6], [10] or even external memory [4], [11].

Skip RNN is able to learn when to update or copy the state without explicit information about which samples are useful to solve the task at hand. However, a different operating point on the trade-off between performance and number of processed samples may be required depending on the application, e.g. one may be willing to sacrifice a few accuracy points in order to run faster on machines with low computational power, or to reduce energy impact on portable devices. The proposed model can be encouraged to perform fewer state updates through additional loss terms:

$$L_{budget} = \lambda \cdot \sum_{t=1}^{T} u_t \tag{6}$$

where L_{budget} is the cost associated to a single sequence, λ is the cost per sample and T is the sequence length.

III. EXPERIMENTS: SEQUENTIAL MNIST

The MNIST handwritten digits classification benchmark [9] is traditionally addressed with Convolutional Neural Networks (CNNs) that can efficiently exploit spatial dependencies through weight sharing. By flattening the 28×28 images into 784-d vectors, however, it can be reformulated as a challenging task for RNNs where long term dependencies need to be leveraged [8]. With the goal of studying the effect of skipping state updates on the learning capability of the networks, we introduce a new baseline which skips a state update with probability p_{skip}. We tune the skipping probability to obtain models that perform a similar number of state updates to the Skip RNN models.

Results in Table I show that Skip RNNs solve the task using fewer updates than their counterparts while also showing a lower variation among runs and train faster. We hypothesize that skipping updates make the Skip RNNs work on shorter subsequences, simplifying the optimization process and allowing the networks to capture long term dependencies more easily. However, the drop in performance observed in the models where the state updates are skipped randomly suggests that learning which samples to use is a key component in the performance of Skip RNN. Examples such as the ones depicted in Figure 2 show how the model learns to skip pixels that are not discriminative, such as the padding regions in the top and bottom of images, and the attended samples vary depending on the particular input being given to the network.

ACKNOWLEDGMENT

This work has been accepted as a conference paper at ICLR 2018, and we refer the reader to the full publication for extended experiments and results [2]. This research was partially supported by the Spanish Ministry of Economy and Competitivity and the European Regional Development Fund (ERDF) under contracts TEC2016-75976-R and TIN2015-65316-P, by the BSC-CNS Severo Ochoa program SEV-2015-0493, and grant 2014-SGR-1051 by the Catalan Government. Víctor Campos was supported by Obra Social "la Caixa" through La Caixa-Severo Ochoa International Doctoral Fellowship program. We would also like to thank the technical support team at the Barcelona Supercomputing Center.

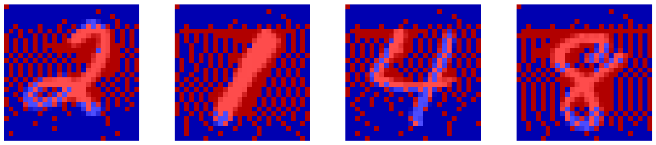

Fig. 2. Sample usage examples for the Skip LSTM with $\lambda = 10^{-4}$ on the test set of MNIST. Red pixels are used, whereas blue ones are skipped.

REFERENCES

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] V. Campos, B. Jou, X. Giró-i Nieto, J. Torres, and S.-F. Chang. Skip rnn: Learning to skip state updates in recurrent neural networks. In *International Conference on Learning Representations*, 2018.

[3] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville. Recurrent batch normalization. In *ICLR*, 2017.

[4] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[5] G. Hinton. Neural networks for machine learning. Coursera video lectures, 2012.

[6] Y. Jernite, E. Grave, A. Joulin, and T. Mikolov. Variable computation in recurrent neural networks. In *ICLR*, 2017.

[7] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, H. Larochelle, A. Courville, et al. Zoneout: Regularizing rnns by randomly preserving hidden activations. In *ICLR*, 2017.

[8] Q. V. Le, N. Jaitly, and G. E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[10] D. Neil, M. Pfeiffer, and S. Liu. Phased LSTM: accelerating recurrent network training for long or event-based sequences. In *NIPS*, 2016.

[11] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[12] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. In *ICLR*, 2015.

Víctor Campos holds a BsC and a MsC degrees on Electrical Engineering from Universitat Politècnica de Catalunya. He is currently pursuing his PhD on the intersection between Deep Learning and High Performance Computing at the Barcelona Supercomputing Center, supported by Obra Social "la Caixa" through La Caixa-Severo Ochoa International Doctoral Fellowship program. His research interests focus on large scale machine learning.