View Meta-Reviews

Paper ID
5394

Paper Title
FastGRNN: A Fast, Accurate, Stable and Tiny Kilobyte Sized Gated Recurrent Neural Network

META-REVIEWER #1

META-REVIEW QUESTIONS

1. Please recommend a decision for this submission.
Accept (Poster)

3. Please provide a meta-review for this submission. Your meta-review should explain your decision to the authors. Your comments should augment the reviews, and explain how the reviews, author response, and discussion were used to arrive at your decision. Dismissing or ignoring a review is not acceptable unless you have a good reason for doing so. If you want to make a decision that is not clearly supported by the reviews, perhaps because the reviewers did not come to a consensus, please justify your decision appropriately, including, but not limited to, reading the submission in depth and writing a detailed meta-review that explains your decision.
This paper presents a variant of RNN that matches the performance of standard approaches like LSTM with a reduced memory footprint obtained with sparsity and quantization. However I appears that the paper discard state-of-the-art results on many standard benchmarks (LSTM is not soa on PTB for example) and they should be included in the final version of the paper. All the reviewers are in favor of accepting this paper though.

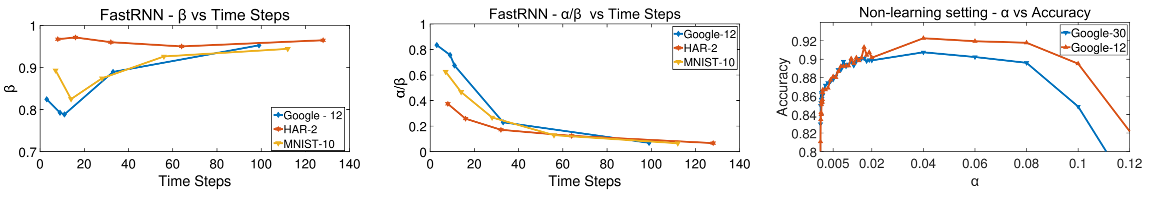
We thank the reviewers for their helpful comments. We will make our code publicly available if our paper is accepted for publication.

Reviewer 1

The Effects of Regularization as studied in Merity et al. (2017): Merity et al. (2017) achieve state of the art results by stacking LSTMs and applying 7 different regularization techniques during optimization. Most of these techniques can be applied straightforwardly to our FastGRNN architecture. While we showed that FastGRNN achieves the same accuracies as a vanilla LSTM with significantly lower computational costs, it will be interesting to investigate the effects of these techniques subject to the RAM and latency constraints of IoT microcontrollers. We performed initial experiments along these lines on the PTB word level language modeling task and observed that the weight-decay regularizer reduced FastGRNN's test perplexity from 116.11 to 111.57 while stacking another FastGRNN layer reduced test perplexity to 106.23. We will update our paper with a detailed study of the effects of these regularizers.

Comparison to Quasi-RNN: Thanks for sharing the reference to the Quasi-RNN paper. Quasi-RNN seems to eliminate state transition matrices in the vanilla RNN and instead relies on convolutions. A theoretical analysis indicates that applying the FastGRNN principles to Quasi-RNNs might result in a 3-20x speedup (on IoT microcontrollers) and model size reduction depending on the number of hidden units, the rank of the convolution mask and input dimensionality. We agree that this would be an interesting extension to our paper and will explore this further in the camera-ready version.

The effects of β and its relation to α: Thanks for the helpful suggestions. Table 11 in Appendix B of our paper shows the values of α and β learnt on three datasets. Table 11 shows that the relative error between β and $1 - \alpha$ for Google-12 with 99 timesteps is 2.15%, HAR-2 with 128 timesteps is 3.21% and MNSIT-10 with 112 timesteps is 0.68%. Also, α/β value for Google-12 with 99 timesteps is 0.069, HAR-2 with 128 timesteps is 0.067 and MNIST-10 with 112 timesteps is 0.065, which reflects the trend suggested by our analysis. Figure 1(a) & 1(b) show the variation of β and α/β with number of timesteps T respectively. For all three datasets, we observe that $\alpha/\beta << 1$ and for large enough timesteps, $\beta \rightarrow 1 - \alpha$. For short sequences, where there is a lower likelihood of gradients exploding or vanishing, it is helpful to let β be a free parameter, different from $1 - \alpha$ (see Table 11, Appendix B). Enforcing the constraint $\beta = 1 - \alpha$ on short sequences can often lead to a drop in accuracy. For example, on the Google-12 dataset with 33 timesteps, one observes a drop of upto 1.5%.



(a) (b) (c)

Figure 1: (a) and (b) plot the learnt values of β, α/β with varying number of timesteps T respectively. (c) Accuracy vs α in non-learning setting where the parameters of the classifier was learnt and evaluated for a range of fixed α values (using 99 timesteps).

Reformatting of paper: Thanks for the feedback. We will make the recommended changes.

Reviewer 2

The effects of β and its relation to α: We refer the reviewer to the response above to Reviewer 1 about a similar issue.

Bias due to the initial hidden states: In order to understand the bias induced at the output by the initial hidden state h_0, we evaluated a trained FastRNN classifier on the Google-12 dataset with 3 different initializations sampled from a standard normal distribution. The resulting accuracies had a mean value of 92.08 with a standard deviation of 0.09, indicating that the initial state does not induce a bias in FastRNN prediction in the learning setting.
In the non-learning setting, the initial state can bias the final solution for very small values of α. Indeed, setting $\alpha = 0$ and $\beta = 1$ will bias the final output to the initial state. However, as Figure 1(c) indicates, such an effect is observed only for extremely small values of $\alpha \in (0, 0.005)$. In addition, there is a large enough range for $\alpha \in (0.005, 0.08)$ where the final output of FastRNN is not biased and is easily learnt by FastRNN.

FastRNN vs Residual Networks: The reviewer correctly points out that setting $\alpha = \beta = 1$ would indeed convert FastRNN into a Residual Network with fresh input coming in at every layer. However, when $\alpha/\beta = 1$ in the recurrent setting, the condition number of the gradient becomes large and such a network becomes very difficult to train. Instead, we use a weighted residual connection, and as shown in Figure 1(b), the values of α/β learnt by FastRNN are very small as compared to the value of 1 for typical Residual Networks.

Reviewer 3

We thank the reviewer for the kind comments.

View Reviews

Paper ID
5394

Paper Title
FastGRNN: A Fast, Accurate, Stable and Tiny Kilobyte Sized Gated Recurrent Neural Network

Reviewer #1

Questions

1. Please provide an "overall score" for this submission.
9: Top 15% of accepted NIPS papers. An excellent submission; a strong accept. I will fight for accepting this submission.

2. Please provide a "confidence score" for your assessment of this submission.
4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.

3. Please provide detailed comments that explain your "overall score" and "confidence score" for this submission. You should summarize the main ideas of the submission and relate these ideas to previous work at NIPS and in other archival conferences and journals. You should then summarize the strengths and weaknesses of the submission, focusing on each of the following four criteria: quality, clarity, originality, and significance.
This paper presents an adaptation to the peep-hole connection from [22] to create fast, accurate, and, small RNNs. There are two variants introduced:

(1) FastRNN which adds two extra learnable parameters to the vanilla RNN to regulate the computation flow between the hidden state and the nonlinear mapping of inputs and states

(2) FastGRNN which replaces the two parameters with gating functions which share input/state matrices with the nonlinear projection but have separate biases. Furthermore, the FastGRNN utilises low-rank sparse representation for matrices with constraints that helps with compressing the size of the model.

Theoretical analysis studies the convergence and stability of these models vs normal RNNs while the extensive experimental setup shows that these models are capable of achieving comparable results to state-of-the-art or comparable models (e.g. LSTMs) with much smaller network sizes.

I really like the paper and the results reported and I'm sure this is going to have a great impact in the field. I can suggest a few improvements that could make the paper even better:

- While the performance of these models have been studied in comparison to competing RNN formulations, the effects of regularisation have not been analysed. For example, the results reported on PTB are far worse than where the SOTA is and recent results on those tasks show that heavy regularisation of LSTMs for example can result in massive improvements in performance ("Regularizing and Optimizing LSTM Language Models", Merity et al 2017). So how would these models behave under heavy regularisation if one wants to attain high performances?
- Also there are other work in this area such as "Quasi-Recurrent Neural Networks", Bradbury et al 2017 which also try to provide faster alternatives to LSTMs. How does FastGRNN perform against these methods?
- While the balance between \alpha and \beta has been discussed briefly in the paper, it would have been better if we had further experimental results on them. Fig 4 in appendix is a good start, but what about \beta, how does that behave? Do we see often that \beta = 1 - \alpha as speculated in the paper? How does the ratio \alpha / \beta change with sequence length or different datasets? There are many interesting results here that would be

good to have a report on
- The paper has considerable redundancy in the introduction Sec 1 and related work Sec 2 which can be condensed. This will provide space to bring forward some of the interesting findings and results from appendix such as Fig 4 which.

4. How confident are you that this submission could be reproduced by others, assuming equal access to data and resources?
3: Very confident

Reviewer #2

Questions

1. Please provide an "overall score" for this submission.
7: A good submission; an accept. I vote for accepting this submission, although I would not be upset if it were rejected.

2. Please provide a "confidence score" for your assessment of this submission.
4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.

3. Please provide detailed comments that explain your "overall score" and "confidence score" for this submission. You should summarize the main ideas of the submission and relate these ideas to previous work at NIPS and in other archival conferences and journals. You should then summarize the strengths and weaknesses of the submission, focusing on each of the following four criteria: quality, clarity, originality, and significance.
This work presents a variant of recurrent neural networks, including its gated version. The key idea is a so-called "peephole" connection which mixes the previous hidden unit with the network hidden output. The proposed method is well supported by theoretical findings and good empirical evidences. The resulting networks are very compact and can be used in embedding products.

Several things need to be clarified:

1) It is unclear how large is the learned beta. Figure 4 in appendex only gives the alpha results. How did you set beta? Was it close to 1-alpha?
2) In the non-learning setting, alpha>0 and 0<beta=1-alpha<1. Applying Eq.2 along a long chain, the mixing can bring bias because early hidden units are involved many times. Please elaborate how to avoid or correct this.
3) By Line 134, beta is close to one. Eq.2 is very close to residual networks. Please elaborate their connections and differences.

The names FastRNN and FastGRNN are not good. It reads like occupying intent and therefore must be discouraged. These names are by no means pertinent because they provide no information about key idea of the method.

4. How confident are you that this submission could be reproduced by others, assuming equal access to data and resources?
2: Somewhat confident

Reviewer #3

Questions

1. Please provide an "overall score" for this submission.
7: A good submission; an accept. I vote for accepting this submission, although I would not be upset if it were rejected.

2. Please provide a "confidence score" for your assessment of this submission.
3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.

3. Please provide detailed comments that explain your "overall score" and "confidence score" for this submission. You should summarize the main ideas of the submission and relate these ideas to previous work at NIPS and in other archival conferences and journals. You should then summarize the strengths and weaknesses of the submission, focusing on each of the following four criteria: quality, clarity, originality, and significance.
In this work, the authors propose a Recurrent Neural Network variant which is both accurate and efficient. FastRNN develops a leaky integrator unit inspired peephole connection which has only two extra scalar parameters. FastGRNN then extends the peephole to a gated architecture by reusing the RNN matrices in the gate to match state-of-the-art accuracies but with a 2-4x smaller model. And after the low-rank sparse and quantized approximation, FastGRNN could make more accurate predictions with up to a 35x smaller model as compared to leading unitary and gated RNN techniques!

Overall, the accuracy and the huge model size reduction ratio is very impressive. The paper gives a very clear review on the various RNN models and illustrate their proposed approach very clearly. I am not an expert on the model size pruning area but from educated guess, the experiment results are extensive, solid and impressive. And it’s appealing that the model is actually evaluated on embedded platform like Arduino and Raspberry Pi.

4. How confident are you that this submission could be reproduced by others, assuming equal access to data and resources?
3: Very confident