Neural Language Model

Davide De Martini

University of Trento

davide.demartini@studenti.unitn.it

1. Introduction

This report outlines the improvement made to the baseline LM RNN through the integration of several regularization techniques. For the first part some minor changes were performed in order to regularize the model.

For the second part, starting from the LSTM baseline created in part 1, other techniques from the paper of Merity et al. [1] were added. The regularization methods implemented from this work are: weight tying, the implementation of variational dropout layer (Gal et al.[2]) and Non-monotonically Triggered AvSGD.

The next section will provide more information about the implementation details starting from the first enhancement.

2. Implementation details

For the first part, the implementation was pretty straightforward. At first the Recurrent Neural Network was substituted with a Long Short Term Memory network. Then two dropout layers were added: one after the embedding and the other before the last linear layer. The last thing done was to substitute the SGD optimizer with the AdamW one.

The listed techniques have been implemented incrementally, monitoring the model's performance at each step. As expected, these leads the model to perform better, so all three regularization were maintained (the results can be consulted at table 1).

For the second part, other improvements were added starting from the LSTM baseline. All the techniques used are described in the paper of Merity et al. [1], in which the authors explain some regularization that lead LSTM models to perform better.

The first method implemented is *weight tying*. It's a form of regularization that consists in sharing the weights across different part of a model. Inan et al. [3] explained that the technique has theoretical motivation and prevents the model to learn multiple parameters, resulting in a one to one correspondence between input and output. The idea is to share the weights of the embedding and the last linear layer. In order to do this, the two layers have to have compatible dimensions, then the two can share the same weights parameters.

The second technique applied is *variational dropout* (Gal et al. [2]). The intuition of this is that with standard dropout the mask is changed through time, but with the variational one the mask is the same at each time step (see figure 1). So, in order to be consistent with the recurrent nature of the LSTM, the mask will be the same for all the batch length. The implementation of this required more effort since by default is not in PyTorch, so an ad hoc class has to be created.

The last thing to implement was one of the key contribution of the Merity et al. [1] work: the Non-monotonically Triggered AvSG (NT-AvSGD). The intuition is to switch to the AvSGD when the SGD converge to a steady-state distribution. This can be seen as triggering it when it reaches a neighborhood of the solution. A first idea could be to trigger it whenever the model perform worse, they proposed that the AvSGD is triggered when the validation metric fails to improve for multiple cycles. This could be done in multiple ways, a first solution could be using the patience mechanism for early stopping. When the patience become negative, it will trigger the AvSGD and then the patience will be reinitialized at its initial value in order to early stop the training. Another solution is one proposed by the code of Merity et al. [1] consultable here: https://github.com/salesforce/awd-lstm-lm. The switch is performed when the validation loss is higher than the minimum of the past losses windowed by a non-monotone interval (see figure 3 for a clarification).

3. Results

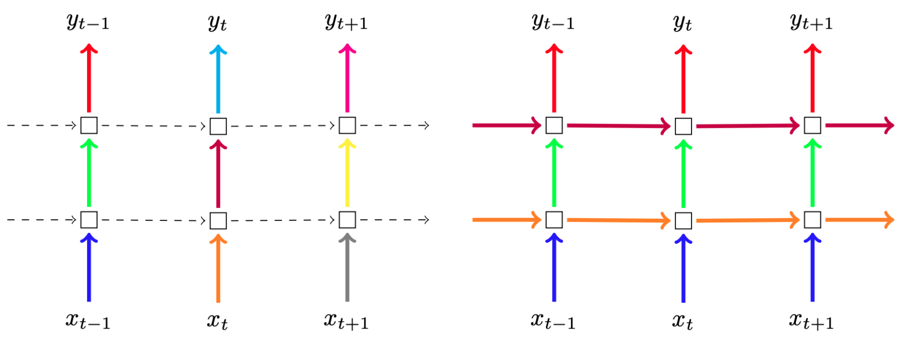
For the first part, the regularization techniques have been incrementally added and tested in order to recognize if the model was performing better or not. The results of this first part can be found in table 1. The model was trained and evaluated with the PennTreeBank dataset.

For the second part, all the methods implemented outperformed the previous model with a big margin. The results achieved are not far from the one presented by Merity et al. [1]. The best training achieved a final perplexity of 93.41, the parameters are the following: lr 5, hidden size and embedding size 800, scheduler with a gamma of 0.75 that step every 5 epochs. The batch size used are 32 for the training set, 64 for the validation set and for the test set. As before, the model was trained and evaluated with the PennTreeBank dataset.

The only concerning part is that, when the AvSGD is triggered, the performance of the model are very pour and the loss strangely increase as reported in figure 3. This is not an isolated problem, since other colleagues had the same issue. Maybe the implementation of AvSGD in PyTorch has some bugs, as now it seems broken.

Regularization technique	Final PPL
LSTM	177.20
LSTM + Dropout layers	140.88
LSTM + Dropout layers + AdamW	112.35

Table 1: *Results of the first part, the left column reports the regularization technique, the right one the perplexity score. For all the test, the size of the hidden layers and the embedding layers is 400. The learning rate is 5 for the first two experiments, instead for the last one is 10^{-4}. The batch sizes used are 10 for the training set and 32 for the validation set and for the test set.*



(a) Naive dropout RNN (b) Variational RNN

Figure 1: *Difference from the Naive Dropout RNN (a) and the Variational dropout RNN (b). Image taken from the work of Gal et al. [2]*

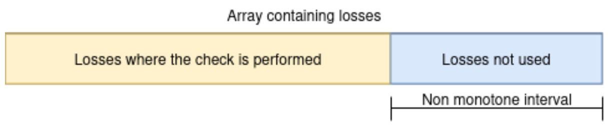


Figure 2: *How the AvSGD is triggered by the non-monotonic Interval*

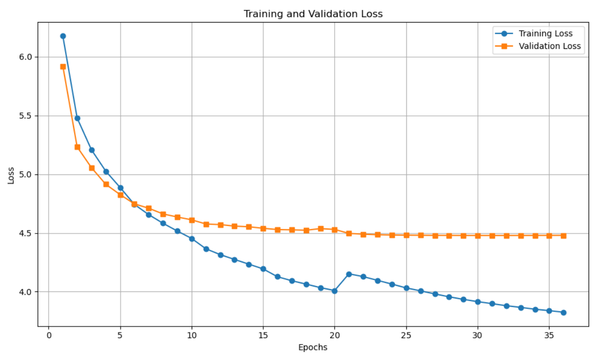


Figure 3: *Strange behavior of the AvSGD. The trigger point is when the loss increase*

4. References

[1] S. Merity, N. S. Keskar, and R. Socher, “Regularizing and optimizing lstm language models,” 2017.

[2] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” 2016.

[3] H. Inan, K. Khosravi, and R. Socher, “Tying word vectors and word classifiers: A loss framework for language modeling,” 2017.