NLU project exercise lab: 9

Matteo Minardi (238789)

University of Trento

matteo.minardi@studenti.unitn.it

1. Introduction

In the first part, the goal was to modify the baseline "LM_RNN" to add a set of improvements and see how these affect the performance. I had to play with the hyperparameters to minimise the PPL.
The set of modifications was the following:

- Replace RNN with LSTM
- Add two dropout layers
- Replace SGD with AdamW

The second part was about the implementation of additional regularizations, in particular:

- Weight Tying
- Variational Dropout
- Non-monotonically Triggered AvSGD

All the necessary information to implement anything was taken from the papers cited in the references.

2. Implementation details

The best results were when the implementations were added one on top of the other in cascade.
A key point in the finding of the optimal performances was the learning rate. In fact, that alone was able to impact the PPL for different orders of magnitude. To find the best configuration, I manually fine-tuned the hypervalue myself, because an automatic search kept getting stuck on local optimas with much higher PPLs, and thus were worse.
The dropout layers made the network more stable and the AdamW optimizer played a crucial role in the decrease of the final PPL. The dropout addition was very effective given its simplicity: just satisfy the dimensionality of the layers.
Mechanisms like weight decay were explored but they have been discarded because the performance didn't show constant improvements.

In the second task, ulterior major improvements have been achieved by using the weight tying optimization, decreasing furthermore the PPL sharing the weights of the embedding layer with the output layer with a very low learning rate.
Even the variational dropout with a Bernoulli distribution as masking technique led to some improvements, although minor ones this time but still acceptable even with much higher learning rates.
For the AvSGD, I've set the number of minimum computed losses to be 5 by default, but obtained the best result with 2, so that the counter t can surpass it easily and AvSGD can be computed as soon as possible and be kept for the whole execution.
The previous models have been trained with a small patience of just 3, but for this final case about AvSGD I had to increase the patience to 10 in order to let it have the chance to be actually activated. Unfortunately this was the only case when the PPL didn't decrease.

3. Results

Model	LR	PPL
RNN base	0.1	249.39
LSTM base	1.0	231.43
LSTM + dropout	1.0	222.64
LSTM + dropout + AdamW	0.01	181.03

Table 1: *Best configs of part 1*
Good to see that the results keep improving when adding the enhancements on top of one another, decreasing the PPL each time even with higher learning rates.

Model	LR	(base) optimizer	n	PPL
Weight Tying	0.01	AdamW	N/A	162.67
WT + Variational Dropout	0.08	AdamW	N/A	161.53
WT + VD + AvSGD	1.8	SGD	2	189.71

Table 2: *Best configs of part 2*
Notice how the drop of PPL still kept going with the first two solutions, bringing it to the lowest ever registered in my case.
Unluckily, the AvSGD solution couldn't decrease it any more, but still was able to return very satisfactory results even with a very high learning rate.

4. References

[1] Mikolov, Tomas, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. 'Recurrent Neural Network Based Language Model'. ISCA. https://www.fit.vutbr.cz/research/groups/speech/publi/2010/mikolov_interspeech2010_IS100722.pdf

[2] Mikolov, Kombrink, Burget, Cernocky, Khudanpur. 2011. 'EXTENSIONS OF RECURRENT NEURAL NETWORK LANGUAGE MODEL'. IEEE. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5947611&tag=1.

[3] Merity, Stephen, Nitish Shirish Keskar, and Richard Socher. 2018. 'REGULARIZING AND OPTIMIZING LSTM LANGUAGE MODELS'. ICLR. https://openreview.net/pdf?id=SyyGPP0TZ