NLU project exercise lab: 9

Mateo Rodriguez (239076)

University of Trento

md.rodriguezromero@studenti.unitn.it

For the first part of the lab four variations of a model were trained and tested, each variation of the model resulted in an improvement of the perplexity. For the second part of the lab, the best performing model from the first part was taken and tested with three more variations.

1. Introduction

- For the first part of the lab, there are two main models, one that uses RNN and the other LSTM
- For the second part the extra modifications added to best model from the last part were: weight tying, variational dropout and non-monotonically triggered ASGD

To maintain the variables as similar as possible the loss used for all the models was cross entropy and a batch size of 16, alongside a hidden size of 384 and an embedding size of 256 maxing out available computing power. Since the optimizers differ the learning rate cannot be held constant.

2. Implementation details

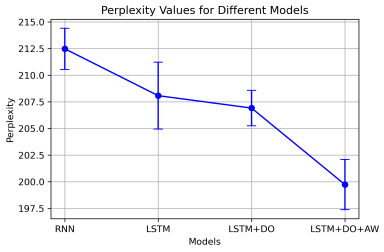
Going more in detail, I utilized the Penn Treebank dataset, and the 4 model variations tested for part 1 were the following: RNN, LSTM, LSTM plus dropout, and LSTM plus dropout with AdamW. All of them used SGD except for the last one.

Regarding the second part of the lab, the best performing model was of course the last configuration, so it was used for this part to add the modifications mentioned previously in an accumulative way. For the first modification, weight tying, it was only necessary to specify that the weights of the output linear layer were the same as the weights of the embedding layer, this has been shown helpful in cases where the input has a similar structure as the output. The next modification was using variational dropout, which involves simply keeping the same dropout mask for both input and output of the LSTM. This can also be used in the between layers of the LSTM, but since the tested models only used one, the dropout was only applied to the input and output of the LSTM, and the main idea behind it is similar to weight tying.

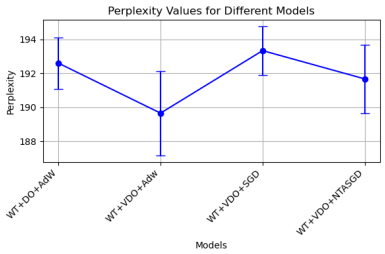
For the final modification two models where tested. Both have weight tying and variational dropout in them but what differs is the optimizer. Up until here the models used AdamW, so to have a baseline comparison one run was trained using SGD and the other NT-ASGD. The implementation of NT-ASGD used is the one provided by the paper [1] which is available in the source code and the idea behind it is to remove the triggering of the averaging as an hyper-parameter and do it automatically after a criteria involving past validation losses is met (another implementation available online with more control over the logging interval (L) was also tested but it didn't seem to be working [2]).

3. Results

For the first part of the lab, the lowest perplexity was obtained by the last model configuration, the LSTM model with dropout and AdamW. Also as expected the LSTM based models performed better than the RNN based one.



For the second part of the lab, results were the following:



Here it's clear that the variations tested were beneficial for the performance of the models, where the model with weight tying and variational dropout reached the lowest perplexity of 189.

To compare the impact of NT-ASGD it was necessary to add another run with weight tying and variational dropout but trained with SGD. In comparison to this run, NT-ASGD performed better, but it still couldn't reach the very good performance of the run using AdamW.

In general results meet expectations, specially confirming the superiority of AdamW. Regarding the other modifications like weight tying and variational dropout, it's possible that they are even more beneficial for larger models and not ones using single layer LSTMs, judging by the theory and logic behind them. It's also important to note that these results are based of training in a single dataset, meaning for a full analysis of the effect of the modifications added to the models it'd be necessary to run similar testing on different datasets and tasks. Compared to the paper the authors use a batch size of 80 and 8000 epochs which are both much higher than the settings tested, so the performance analysis of NT-ASGD might not be representative of it's full capabilities.

Specific scores can be found annexed below.

4. References

[1] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and Optimizing LSTM Language Models," *arXiv preprint arXiv:1708.02182*, 2017.

[2] A. U. Durmus, "Awd-lstm pytorch implementation," https://github.com/ahmetumutdurmus/awd-lstm, Year.

Model	Mean Value	Standard Deviation
RNN	212.47	$\pm$ 1.92
LSTM	208.08	$\pm$ 3.14
LSTM+DO	206.92	$\pm$ 1.66
LSTM+DO+AW	199.74	$\pm$ 2.35

Table 1: *Results of part 1*

Model	Mean Value	Standard Deviation
WT+DO+AdW	192.60	$\pm$ 1.51
WT+VDO+AdW	189.65	$\pm$ 2.50
WT+VDO+SGD	193.34	$\pm$ 1.43
WT+VDO+NTASGD	191.67	$\pm$ 2.02

Table 2: *Results of part 2*