NLU course project - Lab 4

Jonathan Fin (256178)

University of Trento

jonathan.fin@studenti.unitn.it

1. Introduction

In this report the **Language Model** task is analyzed and some optimizations are made, such as:

- substituting a baseline RNN with an LSTM model;
- applying different regularization techniques, such as dropout, weight tying, and variational dropout;
- experimenting with different optimizers like SGD, AdamW, and NT-AvSGD;
- optimizing hyper-parameters to find the best configuration.

All these techniques should lower the model's perplexity, for which a value below 250 is required for a good result.

2. Implementation details

For the first part of the project, an LSTM was substituted to the original vanilla RNN, and two dropout layers were added, one after the embedding layer and the other before the last linear layer. Lastly, the original SGD optimizer was replaced by the AdamW one.

After each change a test was made to assert the functionality of that change, also tuning the hyperparameters to find the best ones, that is, the ones that result in the lowest perplexity.

The second part of the project started from the previous LSTM without other regulation techniques. The first modification added is **weight tying**, which forces the weight of the last linear layer to be the same as the embedding layer; it also needed the hidden size equal to the embedding size, so the two weight matrices have the same dimensions. Another added regularization technique is the **variational dropout**, which generates a dropout mask that is the same for all time-steps in an RNN. As will be described later in the results, this does not differ from the normal dropout, as the best LSTM configuration found is the one with only one layer. The last modification was the implementation of a custom optimizer, the **NT-AvSGD** (Non-monotonically Triggered AvSGD). All these techniques can be found in the paper *Regularizing and Optimizing LSTM Language Models* [1]. In the end, a **GRU** (Gated Recurrent Unit) was also tested instead of the LSTM.

3. Results

The base RNN was first tested with different hyperparameters to find the best ones, on the PennTreeBank dataset. The best learning rate for SGD is 0.5, as seen in Table 1, and the best number of layers is 2 (Table 2), but for the LSTM and the following tasks, it is found that the best number of layers is 1.

Learning rate	Perplexity
2	170.34
1	167.07
0.5	157.33
0.3	160.75

Table 1: *RNN learning rates comparison*

Number of layers	Perplexity
1	157.33
2	154.28
3	156.19
4	158.39

Table 2: *RNN layers comparison*

The perplexity is already below 250, so this project's goal, from now on, is to lower the perplexity even more.

After adding the LSTM, the perplexity dropped to 147.18, with 1 layer and a learning rate pf 0.5. Different dropout percentages were tested, and the best one found is 0.1, as seen in Table 3.

Dropout percentage	Perplexity
0.1	122.75
0.3	157.23
0.5	135.05

Table 3: *LSTM dropout comparison*

The *AdamW* optimizer did not yield many results, scoring a perplexity of 121.56 with a learning rate of $5 * 10^{-4}$.

Restarting from the base LSTM (without dropout), the weight tying regularization reduced the perplexity from 147.18 to 113.14. Adding the variational dropout (with probability 0.1) on top the weight tying brought the perplexity to 109.00. The vanilla dropout technique can do the same because the LSTM's number of layer used is 1, so there are no multiple timesteps (for example, with 3 layers, the perplexity is 113.74). Lastly, the NT-AvSGD did not bring any perplexity optimization, as the value with this optimizer is 109.38. This could also be caused by the fact that the local minima is higher than the previous runs, but generally AdamW tend to score better.

Different hidden and embedding sizes were also tested on the best model, to find the better one, the results are displayed in Table 4. 600 as the size seems to give the best results, but for a trade-off of size / performance, a size equal to 400 is fine.

Hidden & embedding size	Perplexity
300	109.00
400	107.59
500	106.99
600	106.94

Table 4: *Best model sizes comparison*

In addition, different batch sizes were tested on the training part, and the best found is 16, which yielded a perplexity of 104.53 (with the hidden and embedding size equal to 400).

The implementation of a GRU, instead of the LSTM, did not bring any meaningful results. It was tested with the best hyperparameters found earlier in the study and it returned a perplexity of 117.70.

In conclusion, the best optimization found is **weight tying**, but also applying other regulation techniques such as dropout help to reduce the perplexity even more. The perplexity value went from 154.28, with a vanilla RNN with hyperparameters tuning, to 104.53, with an LSTM with weight tying and variational dropout.

4. References

[1] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing lstm language models," in *International Conference on Learning Representations (ICLR)*, 2018, OpenReview preprint and conference submission (ID SyyGPP0TZ). [Online]. Available: https://openreview.net/forum?id=SyyGPP0TZ