NLU project exercise lab: 9

Giovanni Scialla (239181)

University of Trento

giovanni.scialla@studenti.unitn.it

1. Introduction

The aim of this work is to use a language model based on recurrent neural networks to train word embeddings. In the first part of the laboratory, i have substituted the Recurrent Neural Network(RNN) with the Long-Short Term Memory(LSTM) and also searched for the best hyperparameter of the optimizers with a grid-search approach [1]. In the second part i have implemented some of the optimization following the work of Stephen Merity [2]. In particular i have implemented:

- Weight Tying
- Variational Dropout
- Non-monotonically Triggered Averaged-SGD

All of these models will be evaluated using the perplexity score

2. Implementation details

In order to achieve a good perplexity on the first task, one of the game changer modifications was to substitute the SGD optimizer with ADAMW and also increase the learning rate from 0.0001 to 0.1. In addition i've tried several parameters for the optimizer using a grid search searching approach. In details i have searched for the nesterov momentum value (0.99) and weigth decay (0.0001). Another improvement was to use the LSTM network instead of the plain RNN with the addition of two dropout layers in correspondence of the embedding layer and last output layer. The loss function used in the training was the Cross Entropy loss and the model has been trained for 100 epoches. In the second task the major improvements have been achieved by using the weight tying optimization, in order to share the weights of the embedding layer with the output layer. To implement the non-monotonically triggered Av-SGD i've added a trigger variable in the training loop that activates as soon as the standard SGD has converged. In this way we can keep decreasing the loss for a bit more with the Averaged SGD optimizer. Unfortunatly i have obtained no improvement by adding the variational dropout with a bernoulli distribution. All of the three models have been trained for roughly 12 epoches each due to early stopping patience.

3. Results

since we are modeling the probaiblity distribution over a sequence. Our task is, given some previous words, guess which is the next word. In this case, the loss function will be the perplexity, which is roughly how likely the model will miss the true word : the higher the perplexity the worse is the model.

Part 1

	PPL
RNN	231.30
LSTM	221.05
LSTM (Dropout)	201.04
LSTM (AdamW)	164.12

Part 2

	PPL
LSTM (Weight Tying)	235.74
LSTM (Variational Dropout)	237.84
LSTM (NT-AvSGD)	221.44

4. References

[1] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5528–5531.

[2] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing lstm language models," 2017.