NLU Lab: Exercise 9

Nicola Debole
University of Trento
nicola.debole@studenti.unitn.it

1 Models

In the "models" folder there 6 files for each part, 2 versions for each one of the 3 models: one is the trained model checkpoint, the second one is the fine-tuned model. When running the code, it will load the fine-tuned models without training. Set "*train*" to *True* if you want to train the model and then show the results, and use the variable "*load_model_checkpoint*" to decide if the model has to be trained from scratch or from the checkpoint.

2 Part 1

2.1 Implementation

For the first part we developed 3 models:

- A vanilla LSTM model with a Stochastic Gradient Descent optimizer.
- LSTM with 2 dropout layers (on embeddings and on output) with a SGD optimizer.
- LSTM with 2 dropout layers (on embeddings and on output) with AdamW optimizer (Loshchilov and Hutter, 2019).

As for the parameters we opted for embedding size of 256 and hidden layer size of 512 ((Mikolov et al., 2010) suggested to use a value between 20 and 500 for the hidden units).

To make it faster we decided to first train all the models from scratch with a high learning rate (0.5 for SGD and 0.01 for ADAMW) and also with high patience (20) for 1000 epochs. Then the models are fine-tuned with a smaller learning rate (10 epochs, 3 patience, 0.05 lr for SGD and 0.001 lr for AdamW). After this step we checked the test PPL score to make sure the models were not overfitted. The probability for the embedding dropout layer is 0.4 and 0.2 for the output dropout layer.

2.2 Results

Model	Perplexity
LSTM + SGD	182.42
LSTM + SGD + Dropout	211.44
LSTM + AdamW + Dropout	184.67

3 Part 2

3.1 Implementation

For the second part we first implemented weight tying and checked with "*torch.equal()*" if the weights were truly tied. The model has embedding size and hidden size of 256 (they must be equal) and for the training we used lr = 0.5 and patience = 10 (the max epochs were set to 1000).

Then for Variational Dropout we took part of the code[1] from an implementation of the paper (Gal and Ghahramani, 2016) and adapted it to our model. The learning rate is set to 0.45, patience = 10 and n_epochs = 300. Then it has been finetuned with lr = 0.05.

Lastly the model trained with the NT AvSGD optimizer had lr = 4.5 and patience 30 for 300 epochs, then finetuned with lr = 0.45 and patience 3 (effectively trained for 16 epochs).

3.2 Results

Model	Perplexity
LSTM + SGD + Weight Tying	226.36
LSTM + SGD + Variational Dropout	140.46
LSTM + NT AvSGD	121.96

References

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

[1] https://github.com/keitakurita/Better_LSTM_PyTorch

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. volume 2, pages 1045–1048.