How to represent a word and predict it, too: Improving tied architectures for language modelling

Supplemental Material

Kristina Gulordava **Laura Aina** **Gemma Boleda**
Universitat Pompeu Fabra
Barcelona, Spain
{firstname.lastname}@upf.edu

A Training setup

A.1 Language models

On PTB, we trained our language models with stochastic gradient descent for 100 epochs with decaying learning rate based on validation perplexity. We report the best model after the hyperparameter search for dropout (between 0.1 and 0.7 with step of 0.1), learning rate (1, 5 and 20). Batch size was fixed to 20. For the Wiki corpus, we trained for 20 epochs and used Adam optimiser (Kingma and Ba, 2014)[1] to reduce the hyperparameter search. The initial learning rate was 0.001 for batch size of 64. We fixed the dropout to 0.2 after some preliminary experiments.

A.2 Word2vec models

We implemented the standard and tied CBOW models using PyTorch. We trained the CBOW models using batch size of 1024 and report the best tied and non-tied models after the hyperparameter search for learning rate (0.05, 0.1, 0.2). For skip-gram model, we modified a standard TensorFlow implementation[2] and trained the standard and tied models for 20 epochs. We varied the learning rate (0.05, 0.1, 0.2) and fixed other hyperparameters to default values. We used the analogy question dataset from Mikolov et al. (2013) as the validation data for model selection.

References

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. pages 1–12. ArXiv preprint arXiv:1301.3781.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of adam and beyond. In ICLR'18.

[1] We used the version modified as in Reddi et al. (2018).

[2] https://github.com/tensorflow/models/tree/master/tutorials/embedding