Recurrent Neural Networks for Natural Language Inference

Peter Mitura (miturpet@fit.cvut.cz)

December 19, 2017

1 Introduction

Natural Language Inference (NLI) is a class of classification problems, where we are given two natural language sentences, and our goal is to find out whether the first one (called hypothesis) could be deduced from the second one (called premise).

Three possible outcomes are considered:

a) **Entailment** The hypothesis can be deduced from the premise. Example premise and hypothesis:

   `An old man with a package poses in front of an advertisement.`

   `A man poses in front of an ad.`

b) **Contradiction**: The hypothesis is contradicted by the premise. Example:

   `A land rover is being driven across a river.`

   `A sedan is stuck in the middle of a river.`

c) **Neutral**: Both sentences have an independent meaning. Example:

   `A man in a black shirt is overlooking bike maintenance.`

   `A man learns bike maintenance.`

The advancement in this field was significantly accelerated by the Stanford Natural Language Inference (SNLI) dataset [1], which contains over 570 000 sentence pairs, each labeled by a majority vote of five independent human annotators, as the classification might be subjective to some extent.

We have used this dataset in order to compare several recurrent neural network (RNN) setups. A baseline model is given by authors of the dataset, which has 77.6% testing accuracy [1] and uses separate LSTM encoders for both sentences. The current state of the art models are able to achieve up to 86% accuracy [2], or even 89.1% [3] if we allow attention models and ensembles.

2 Used models

Since feeding the individual letters into the network would not be computationally feasible, we have decided to use the word-by-word approach. Words are represented by pretrained word embeddings with 300 parameters, gained from the GloVe Common Crawl Corpus [4]. It has a vocabulary of nearly 1 900 000 words, found by crawling various internet sources and trained on co-occurrence statistics.

During our preprocessing stage, words found in the embedding database are transformed into corresponding vectors. Of 36 273 distinct words in the SNLI dataset, 32 033 are represented in the GloVe. A more detailed look into the unmatched words has shown, that most of them are typos, compound words, or generally uncommon. Sentences containing unknown words are not used for training, reducing the size of the input dataset to 535 392.

Our model does not further train these weights, as it would add an unnecessary bulk of additional parameters. Instead, the first layer of all networks we use is always a dense layer, applied to each time step with same weights. This allows the network to shift and turn off parameters in embeddings as needed, without the need to adjust each value separately.

On top of that, and based on a multitude of runs with a smaller input set (limited to first 10 000 or 100 000 pairs), we have selected two RNN setups for comparison on the full dataset.

The first one is similar to the baseline used in [1], using two separate recurrent encoders for sentences, and then feeding them to three non-recurrent layers. We have used a larger number of parameters than the baseline, employing 300 neurons in each recurrent layer and 600 neurons in dense layers, in order to match the size of embeddings. For recurrent layers, gated recurrent units (GRU) are used instead of LSTM cells, as lengths of the sequences are generally short and GRU cells have been shown to achieve similar results as LSTMs, but with a better time performance [5]. The model is visualized in Figure 1.

The second model operates in an analogous fashion, but uses two recurrent layers instead of one, and feeds their output into two non-recurrent dense layers. This could in theory allow the network to encode more complicated patterns inside recurrent encoders. The model is visualized in Figure 2.

Both models use a three-way softmax classifier at the end, and are trained with respect to cross entropy cost function by a RMSProp optimizer with learning rate of 0.001. As overfitting was generally low thanks to large data size and variance, we have tried to run both networks without any dropout layers, and with dropouts on output of every layer using the keep rate of 0.9.

3 Results

Both models have been trained in at most 20 epochs (with early stop after 5 epochs if the testing accuracy does not improve) using the full dataset. The training and testing accuracy of both used models are shown in Table 1. As dropout was applied during the training but turned off in testing phase, training accuracy may appear lower than it should when it is used. The progression of accuracy during training in both single GRU models is visualized in Figure 3. All trainings have been performed on a single NVIDIA GeForce GTX 1050 Ti GPU, with one training session lasting around 6 hours in case of single GRU setup and 9 hours with double GRU.

4 Conclusion

As the runs have shown, fitting on the full dataset was so difficult, that models without any dropout managed to achieve best results and did not exhibit any signs of overfitting. Our single GRU setup has matched results of the baseline in terms of testing accuracy, although we had to use much more parameters, meaning there is still a lot of room for improvement. The configuration with a second GRU layer has been shown to have no additional benefits in comparison with the first model, while being slower to train, meaning the future improvements should probably adopt different direction.

The source code is publicly available at `https://github.com/PMitura/snli-rnn/` under MIT license.

Model	Number of parameters	Training accuracy	Testing accuracy
Single GRU, 0.1 dropout	2 346 003	72.39%	74.86%
Single GRU, no dropout	2 346 003	**77.37%**	**77.12%**
Double GRU, 0.1 dropout	1 985 403	70.60%	74.00%
Double GRU, no dropout	1 985 403	76.70%	76.47%

Table 1: Training and testing accuracy for all tested models.

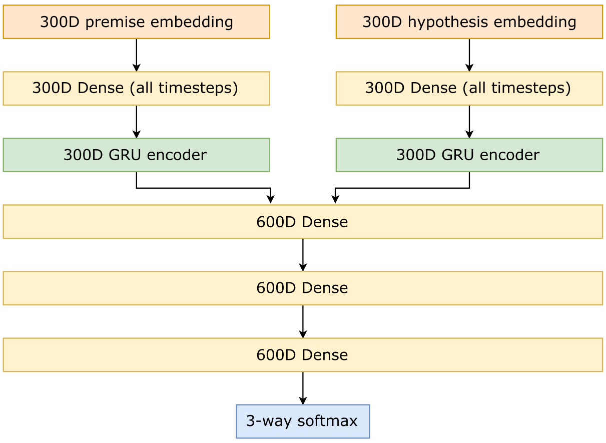

Figure 1: Single GRU setup.

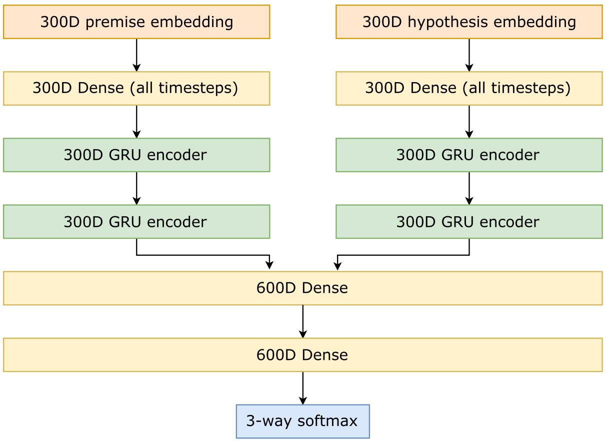

Figure 2: Double GRU setup.

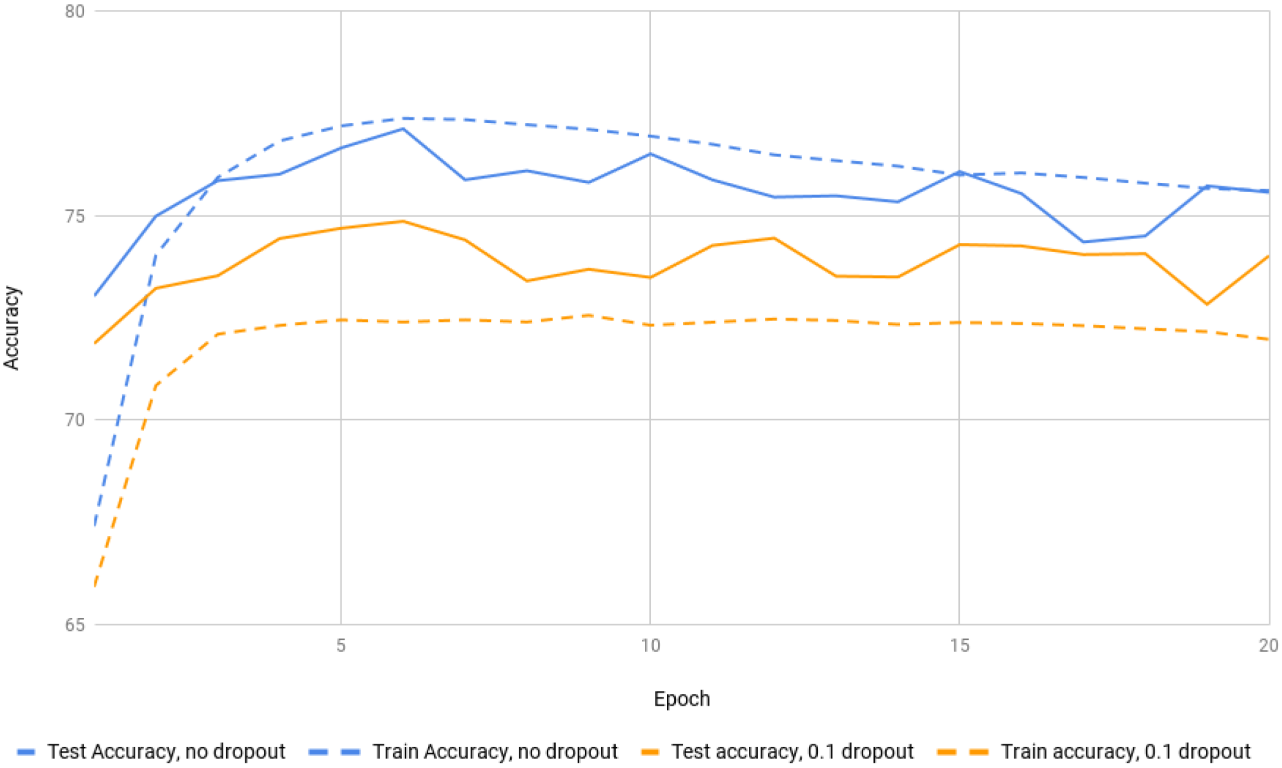

Figure 3: Progression of accuracy during epochs in tested single GRU models.

References

[1] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, “A large annotated corpus for learning natural language inference,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2015.

[2] Y. Nie and M. Bansal, “Shortcut-stacked sentence encoders for multi-domain inference,” *CoRR*, vol. abs/1708.02312, 2017.

[3] Q. Chen, X. Zhu, Z. Ling, D. Inkpen, and S. Wei, “Natural language inference with external knowledge,” *CoRR*, vol. abs/1711.04289, 2017.

[4] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[5] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *CoRR*, vol. abs/1412.3555, 2014.