Compression of Neural Machine Translation Models via Pruning

Abigail See*, Minh-Thang Luong*, Christopher D. Manning

*equal contribution

The Problem

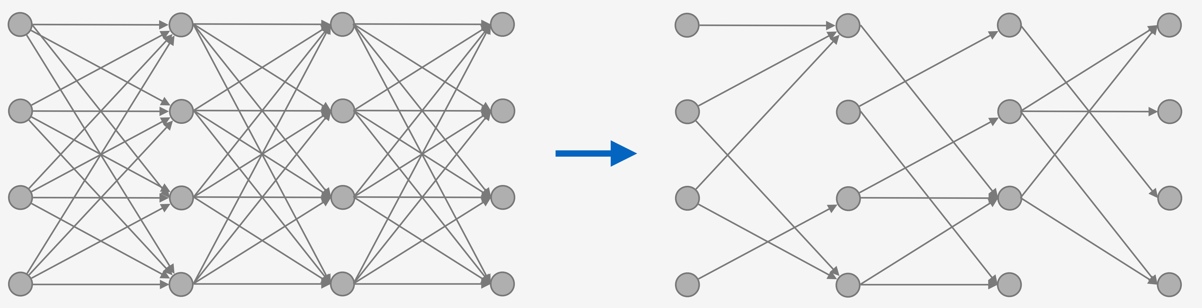
- NMT models (and neural networks in general) are getting **bigger** and **bigger**.
- Advantages: performance improvements!
- Disadvantages: **over-parameterization** leads to long running times, large storage size and overfitting.

This is an obstacle for NMT on mobile devices.

How can we reduce over-parameterization?

The Solution

Magnitude-based parameter pruning is simple: delete weights (connections) that are close to zero.



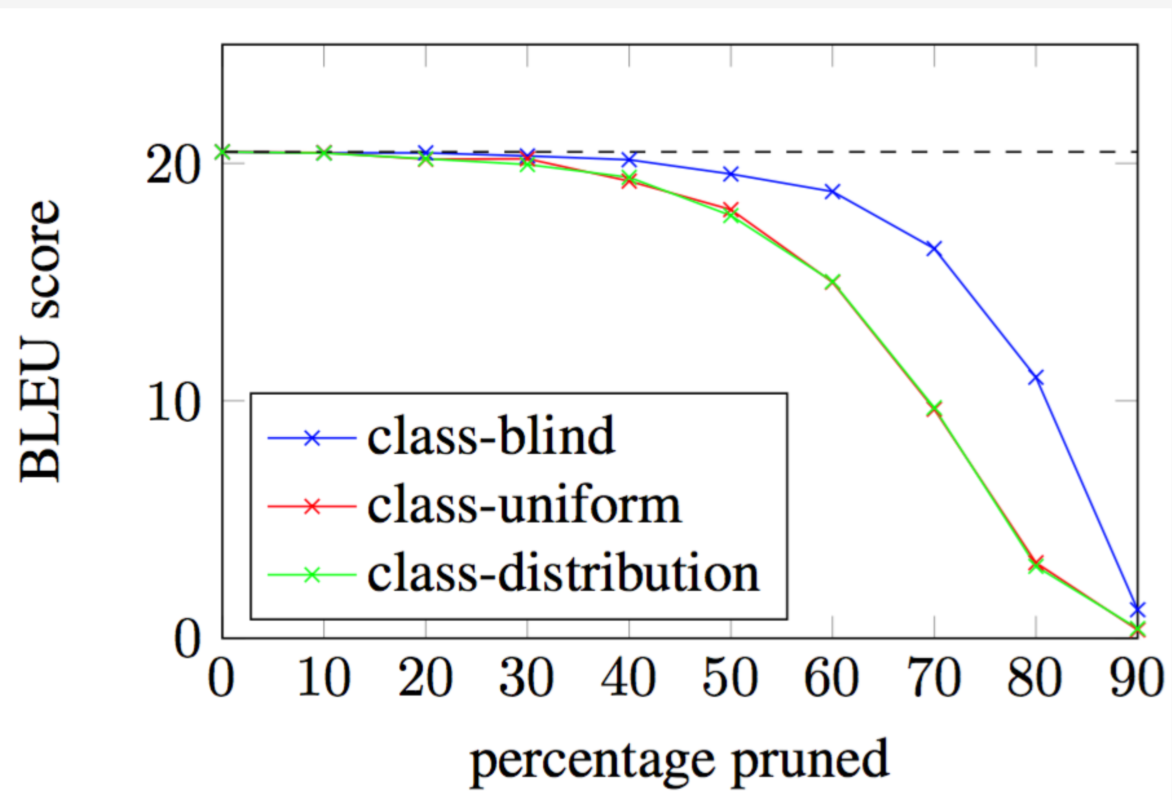
The remaining weights must be retrained to recover performance [1].

Pruning Schemes

The NMT architecture includes several *classes* of weights (see 'Our NMT Architecture').

Should we prune:

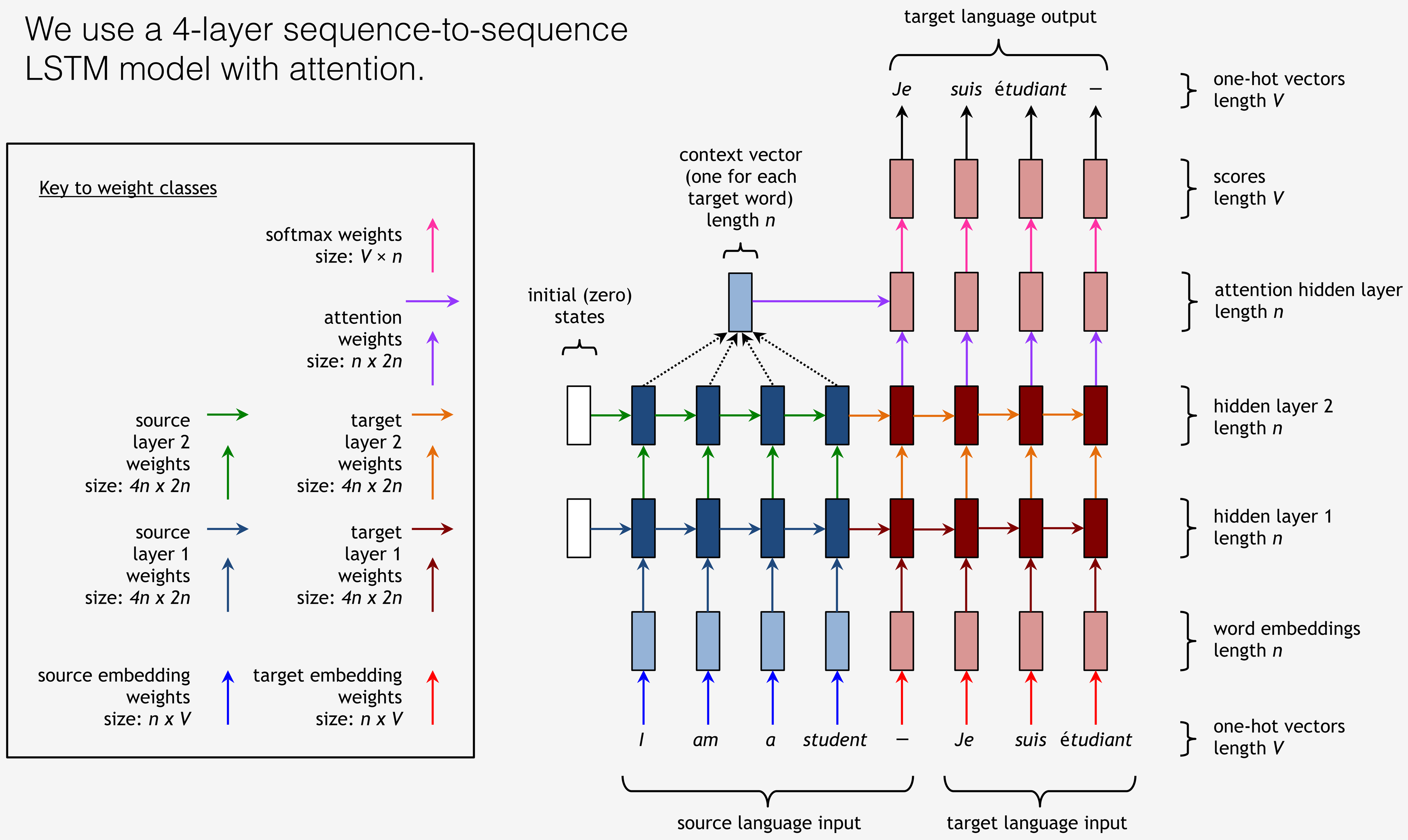
- *proportionally* from each class (class-uniform pruning), or
- in proportion with the *standard deviation* of each class (class-distribution pruning), or
- *without regard* to class (class-blind pruning)?



Simplest method works best!
(class-blind pruning)

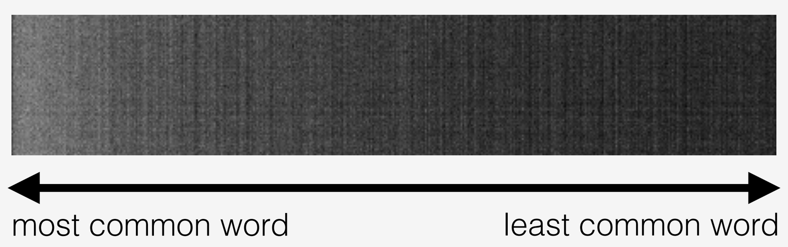
Our NMT Architecture

We use a 4-layer sequence-to-sequence LSTM model with attention.



Areas of Redundancy

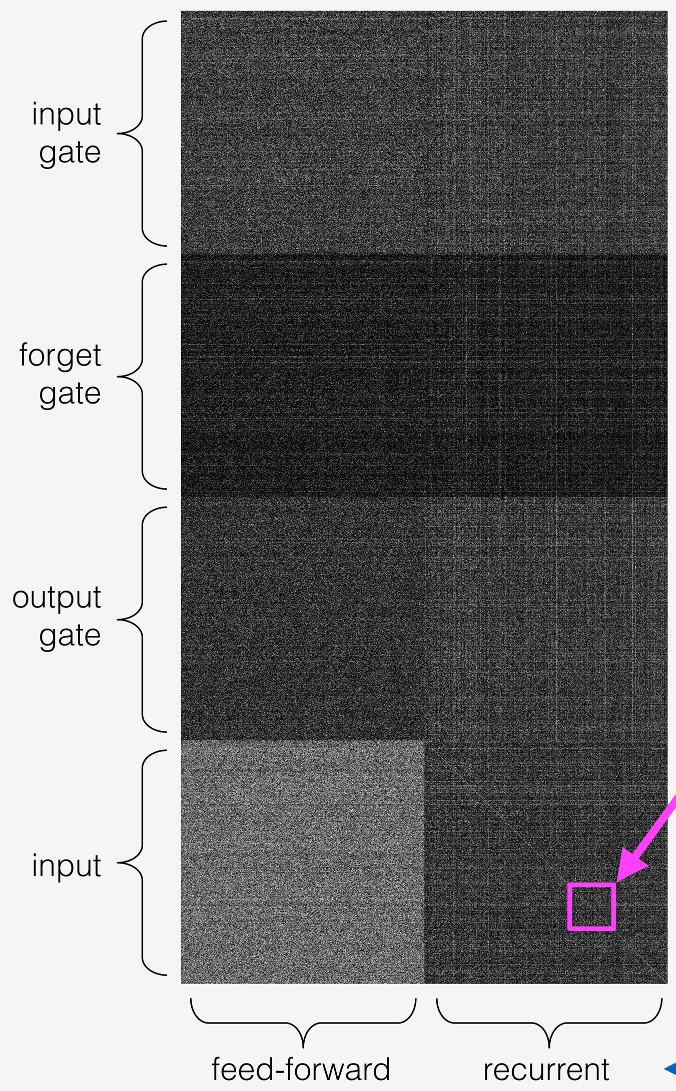
The location of the pruned weights reveals the areas of redundancy in the network.



In the embedding matrix, the weights for rare words are more dispensable than those for common words.

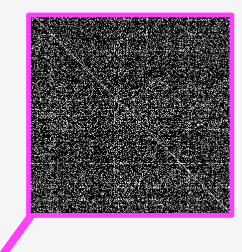
Key
black pixel = pruned
white pixel = remaining

Layer 1 LSTM weight matrix (source)



At layer 1 (left) the input weights are the most important but at layer 4 (right) the gates become important too.

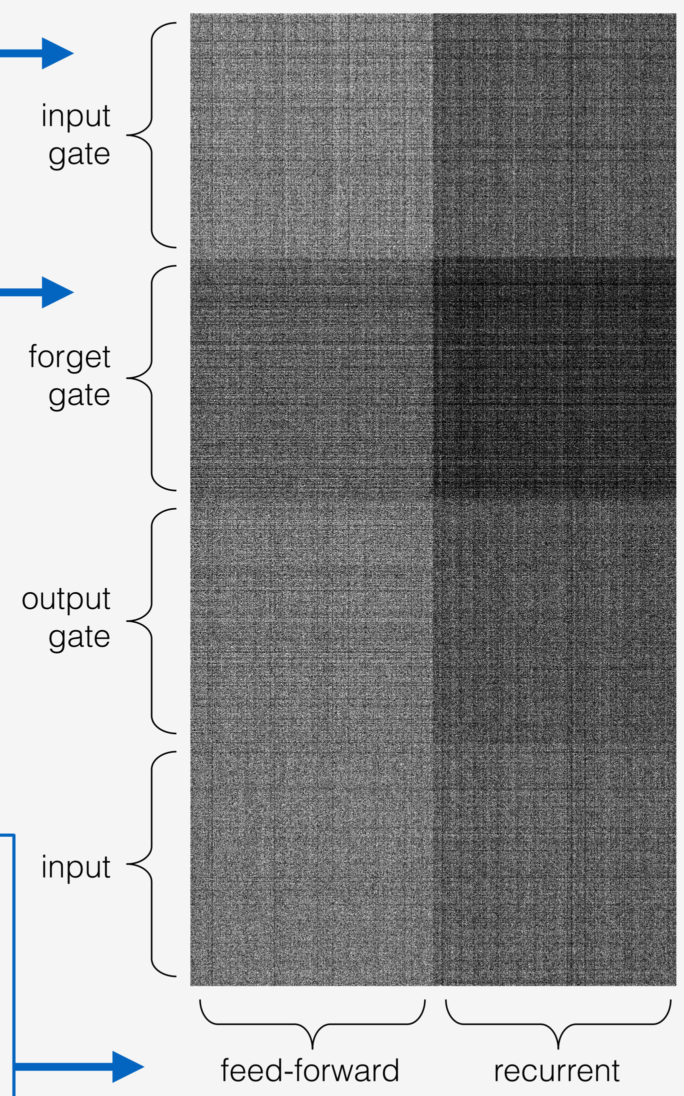
In general, higher layers contain less redundancy.



Diagonal pattern: the network is learning an identity-like transformation.

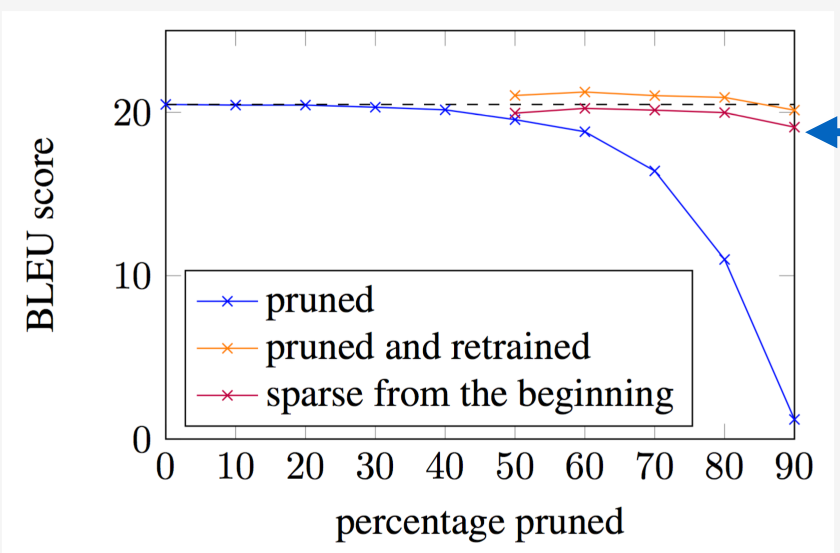
Layer 1 (left): feed-forward connections more important than recurrent connections. Layer 4 (right): recurrent connections also important.

Layer 4 LSTM weight matrix (target)



Results

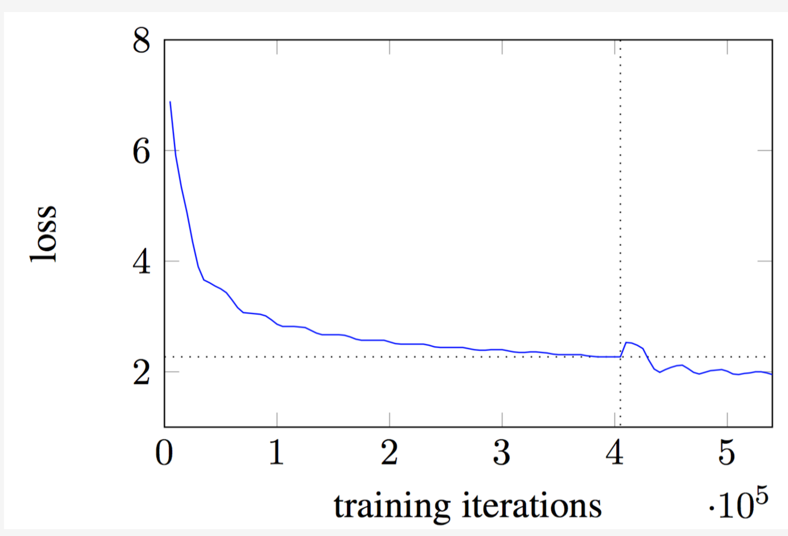
- **Baseline**: state-of-the-art English-German model with 6.1 perplexity and 20.5 BLEU on WMT'14 [2].
- Can prune up to **40%** with negligible effect on performance — a sign of redundancy!
- With retraining, can prune **80%** and *surpass* baseline performance!



Training sparse models is less successful than the train-prune-retrain method.

Pruning also…

- ***regularizes*** the retraining phase.
- ***aids the optimization process***. Pruning helps the model escape its convergence point to find a better one (see below).



Conclusion

- **Weight pruning** is an effective compression method.
- We can make a SOTA model **5 times smaller** with slight performance improvement.
- Pruning seems to aid **optimization** and **regularization**.
- It also gives insights into areas of **redundancy** in the NMT architecture.

Citations
[1] Song Han, Jeff Pool, John Tran, and William Dally. 2015b. *Learning both weights and connections for efficient neural network*. In *NIPS*.
[2] http://nlp.stanford.edu/projects/nmt/

Acknowledgements
- NSF Award IIS-1514268
- Bloomberg L.P.
- NVIDIA Corporation
- DARPA Communicating with Computers (CwC) program under ARO prime contract no. W911NF-15-1-0462