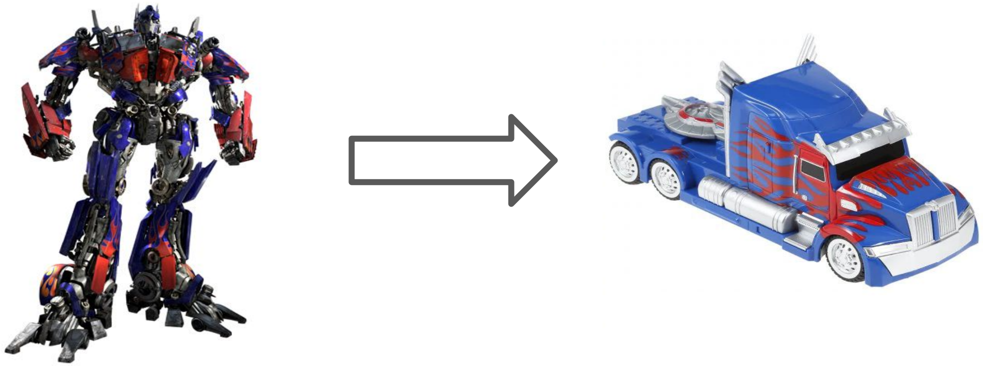
Compressed Transformer



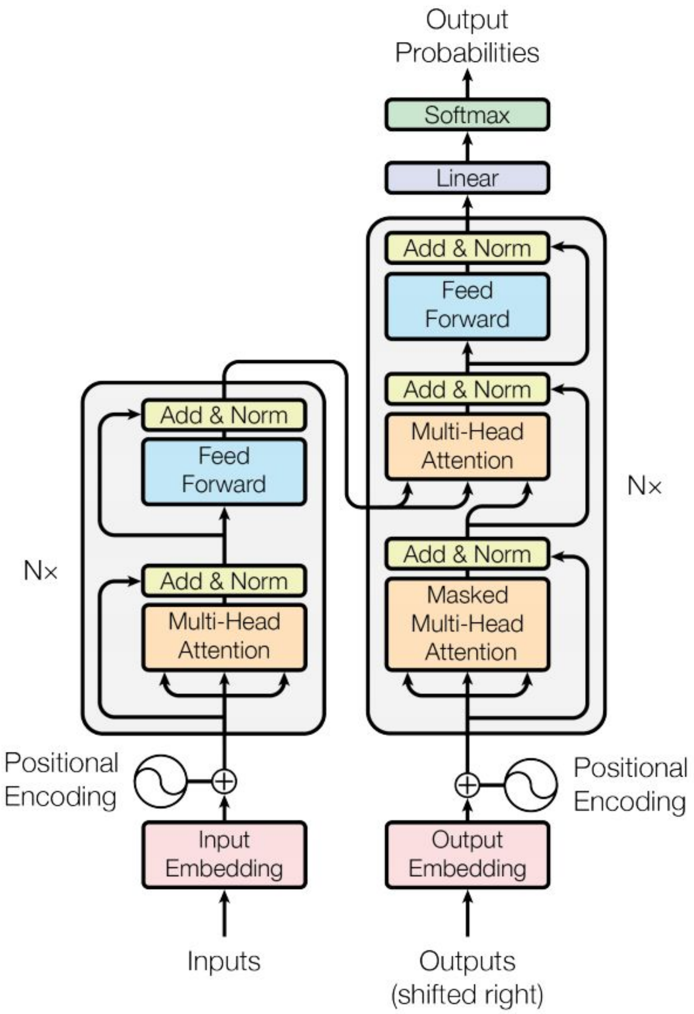
Taras Khakhulin
Irina Saparina
Aleksandr Shevchenko
Michael Konobeev

Transformer

model for **state-of-the-art** results in NLP:

- neural machine translation
- Q&A
- NER
- POS-tagging

Up to **213×10^6** parameters



Vaswani et al. "Attention Is All You Need", 2017.

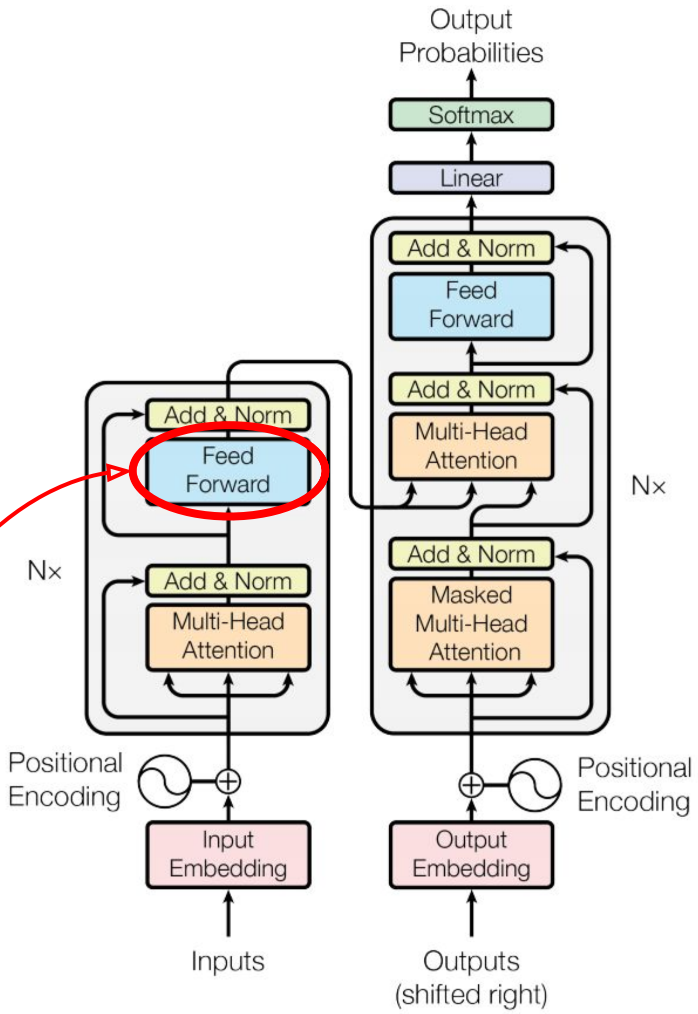
Transformer

model for **state-of-the-art** results in NLP:

- neural machine translation
- Q&A
- NER
- POS-tagging

Up to **213×10^6** parameters

$2^{21} \approx 2 \times 10^6$ parameters in **one** Feed Forward block!



Vaswani et al. "[Attention Is All You Need]", 2017.

Problem Formulation

The goals are:

- to study of different compression methods for Feed Forward NN
- to compress Feed Forward NN in Transformer

Methods

Explicit structure:

- Tensor-train
- SVD + finetune

Sparsification:

- Magnitude pruning $w_i = \mathbb{I}[|w_i| > \lambda] w_i$
- Sparse Variational Dropout[1]

  Idea: ELBO maximization, with variational posterior induced by:

  $$w_{ij} = \theta_{ij} + \sigma_{ij}\varepsilon_{ij}$$
  $$\varepsilon_{ij} \sim \mathcal{N}(0,1)$$

  and prior $p(w_{ij}) \propto \frac{1}{|w_{ij}|}$

[1] Molchanov et al. "Variational Dropout Sparsifies Deep Neural Networks", 2017.

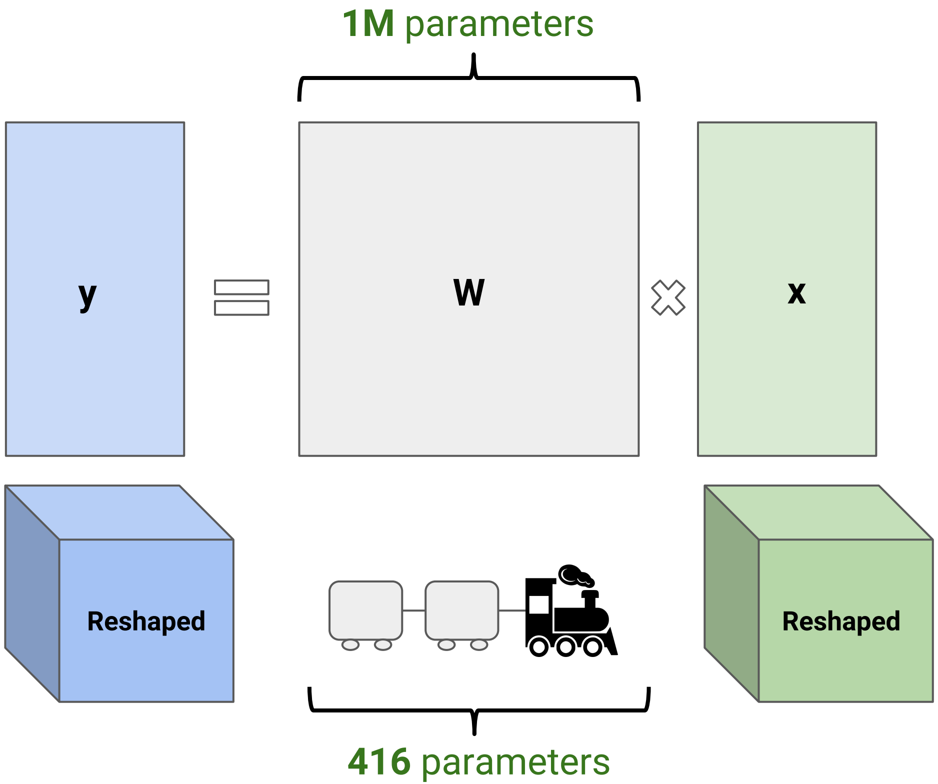
Results on MNIST

Network: one hidden layer NN, a.k.a. FeedForward block of Transformer

Method	Accuracy	Compress ratio
Original Model	0.9770	1.00
Magnitude Pruning	0.9546	8.15
SVD	0.9669	7.54
SVD + fine tuning	0.9641	12.68
Variational Dropout	0.9841	23.39
Tucker	0.9247	28.82
Tensor Train	0.9643	56.95
Tensor Train + fine tuning	0.9620	**72.02**

Seems like TT is the best choice

TT in Transformer



- Number of blocks: N = **6**
- Number of Linear blocks in FeedForward NN **2×N = 12**
- Target embeddings: 15M
- Source embeddings: 22M

Total in full model: **65M**

multi-gpu mode for training

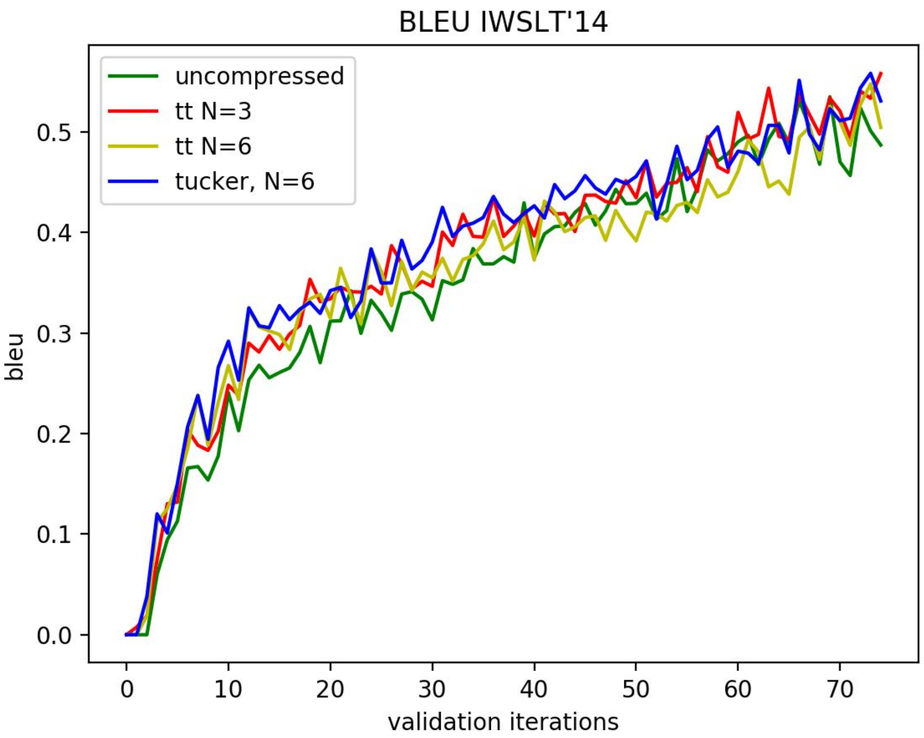
Novikov et al. "[Tensorizing Neural Networks]", 2015.

Results: Multi30k

Method	**BLEU**	**Compression ratio**	**Time**
original model	0.442	1.0	41.38
all tt-compressed	0.434	**1.64**	76.89
small-transformer	0.403	1.6	-
⅚ tt-compressed	**0.489**	1.484	75.24
½ tt-compressed	0.412	1.243	71.36
⅙ tt-compressed	0.414	1.069	40.97
all Tucker-compressed	0.447	**1.64**	43.83

Feed-forward block:
2×TTLayer(ranks=[1, 2, 2, 2, 2, 1])
Tucker ranks equal to 2

Results: IWSLT'14



Model	BLEU	Time
transformer	0.291	154.23
all tt-compressed	0.292	241.24
½ tt-compressed	**0.297**	180.81
Tucker-compressed	0.283	198.3

Conclusion

We did:

- Comparison of various algorithms for compression of FeedForward NN (SVD, pruning, VarDrop, TT and Tucker) on small data: **TT works better**
- Implementation an original and compressed Transformer model with multi-gpu training

We archived:

- Compressed Transformer converges **faster** than original
- TT-compressed Transformer **slower** for inference
- TT-compressed Transformer is **bad** for multi-gpu mode (but not other option)
- Tucker-compressed Transformer hasn't these disadvantages, but BLEU is **worse**

References

1. Vaswani, Ashish, et al. "Attention is all you need." *Advances in Neural Information Processing Systems.* 2017.
2. Novikov, Alexander, et al. "Tensorizing neural networks." *Advances in Neural Information Processing Systems.* 2015.
3. Dmitry, Molchanov et al. "Variational Dropout Sparsifies Deep Neural Networks." *International Conference on Machine Learning*. 2017.