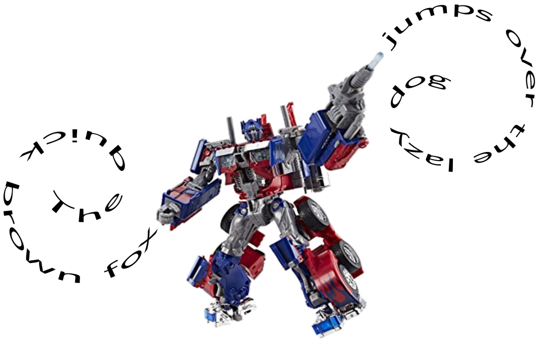


Papers & Cookies VII: Transformers

Vaswani et al.'s *Attention Is All You Need*

Luis Oala
ML Group @ Fraunhofer HHI

November 6, 2018

Overview

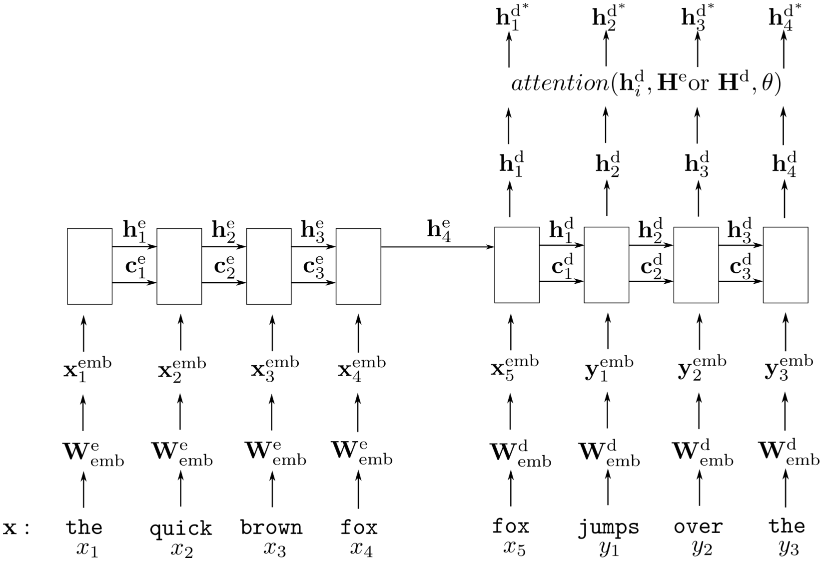


Figure: Vanilla seq2seq LSTM with attention mechanism.

Attention as we (or I) know it

attention$(\mathbf{h}_i^{\mathrm{d}}, \mathbf{H}^{\mathrm{e}}\text{or } \mathbf{H}^{\mathrm{d}}, \theta)$ steps:

(a) Calculate *attention scores* $\mathbf{e}_i$, e.g. as $\mathbf{e}_i = \mathbf{v}^T \tanh(\mathbf{W}_1\mathbf{H} + \mathbf{W}_2\mathbf{h}_i^{\mathrm{d}} + \mathbf{b}_{\mathrm{attn}})$,

(b) normalize the *attention scores* to an *attention distribution* $\mathbf{a}_i = \mathrm{softmax}(\mathbf{e}_i)$ via softmax

(c) and finally combine the attention values, $\mathbf{H}$, into an attention weighted representation $\mathbf{h}_i^{\mathrm{d}^*} = \mathbf{H}\mathbf{a}_i$.

(c) Then do what you please with $\mathbf{h}_i^{\mathrm{d}^*}$, often we see *concatenate*$(\mathbf{h}_i^{\mathrm{d}}, \mathbf{h}_i^{\mathrm{d}^*})$ before going to the output FC layer.

Terminology

- Intra-temporal (regular) attention
  - $\mathit{attention}(\mathbf{h}_i^{\mathrm{d}}, \mathbf{H}^{\mathrm{e}}, \theta)$
- Intra-decoder attention/self-attention
  - $\mathit{attention}(\mathbf{h}_i^{\mathrm{d}}, \mathbf{H}^{\mathrm{d}}, \theta)$

- $\mathbf{h}_i^{\mathrm{d}}$: query $\mathbf{q}$
- $\mathbf{H}^{\mathrm{e}}$: keys $\mathbf{K}$
- $\mathbf{H}^{\mathrm{e}}$: values $\mathbf{V}$

Transformer - Basic mechanics

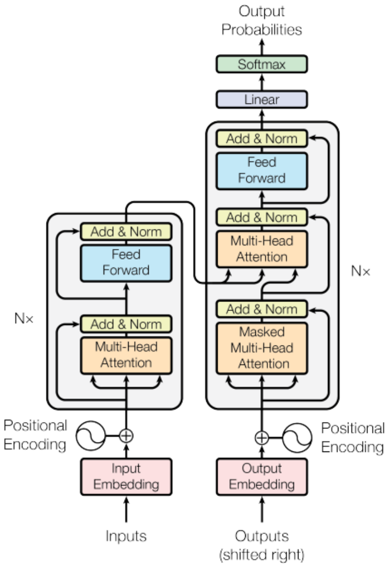
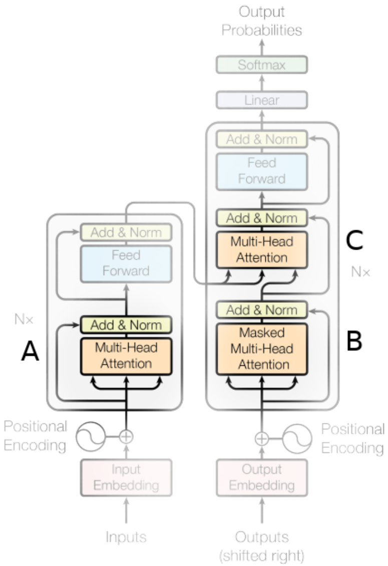
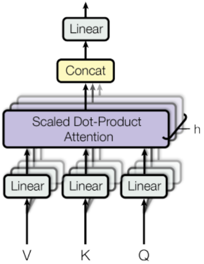


Figure: Transformer illustration, graph taken from [3]

Transformer - Basic mechanics



Transformer - Basic mechanics



Scaled dot-product attention: $\text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}$

With $\mathbf{Q}$.shape = [seq-positions, d_k], $\mathbf{K}$.shape = [seq-positions, d_k] and $\mathbf{V}$.shape = [seq-positions, d_v]

Transformer - Basic mechanics

Tracing the dimensions - step by step

- $\mathbf{QK}^T$.shape $=$ [seq-positions, seq-positions], interpretation: one row represents dots of one query with all keys like the attention scores $\mathbf{e}_i$ from before
- softmax$(\frac{\mathbf{QK}^T}{\sqrt{d_k}})$.shape $=$ [seq-positions, seq-positions], interpretation: one row represents attention distribution $\mathbf{a}_i$ from before
- softmax$(\frac{\mathbf{QK}^T}{\sqrt{d_k}})\mathbf{V}$.shape $=$ [seq-positions, d_v], interpretation: one row represents attention weighted interpolation of all values w.r.t one query

Transformer - Bells and whistles

- Positional encoding as $PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$
- Residual connections
- Layer normalization

Transformer - Results

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Universal Transformer - Additions

Motivation

- Empirical: Improve generalization
- Theoretical: Expanding computational expressivity

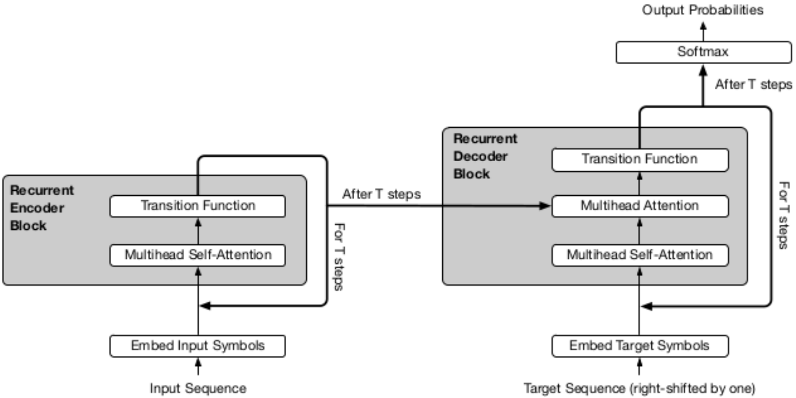


Figure: Illustration of Universal Transformer, graph taken from [1]

Universal Transformer - Results

Model	BLEU
Universal Transformer *small*	26.8
Transformer *base* [31]	28.0
Weighted Transformer *base* [1]	28.4
Universal Transformer *base*	**28.9**

Table 7: Machine translation results on the WMT14 En-De translation task trained on 8xP100 GPUs in comparable training setups. All *base* results have the same number of parameters.

BERT - Usage

- Only use encoder part of transformer to pretrain a LM
- Pretraining on masked inputs and next sentence prediction
- Use encoder outputs as input for downstream task, fine-tune all parameters

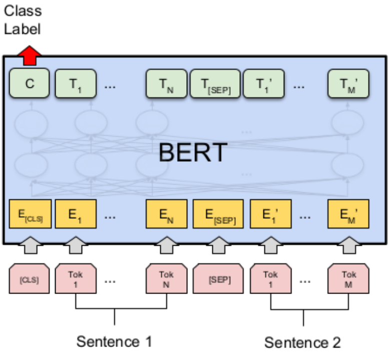


Figure: Illustration of BERT for sequence classification task, graph taken from [2]

BERT - Results

Let us check the paper, too much stuff

Observations and questions

- Path length reduction vis-a-vis LSTM
- Computational elegance and impressive performance

- What is the point of positional encoding?
- Other ways to decode? (e.g. like in the Universal Transformer)
- BERT hyperparameters: how to validate the masking scheme?

Bibliography I

M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser.
Universal Transformers.
arXiv preprint arXiv:1807.03819, 2018.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova.
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
arXiv preprint arXiv:1810.04805, 2018.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, \. Kaiser, and I. Polosukhin.
Attention is all you need.
In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.
read.