The Transfomer Architecture*

Karl Stratos

1 Model

Each boldfaced function denotes a layer with its own set of trainable parameters (Appendix A), except **Drop** which is just dropout with rate 0.1. The only other parameters are the input $E^I \in \mathbb{R}^{512 \times |V^I|}$ and output $E^O \in \mathbb{R}^{512 \times |V^O|}$ word lookup matrices (if the input and output vocabularies are the same, they can be tied).

1.1 The Encoder

Let $x_1 \ldots x_n \in V^I$ denote an input sentence. The initial input represenations $z_1^{(0)} \ldots z_n^{(0)} \in \mathbb{R}^{512}$ are obtained by $z_i^{(0)} = \textbf{Drop}\left(E^I_{x_i} + \pi_i\right)$ where $\pi_i \in \mathbb{R}^{512}$ is an encoding of the i-th position. At each $l = 0 \ldots 5$, we will induce $z_1^{(l+1)} \ldots z_n^{(l+1)} \in \mathbb{R}^{512}$ from $z_1^{(l)} \ldots z_n^{(l)}$ as follows.

$$\begin{aligned}
\tilde{z}_1^{(l)} \ldots \tilde{z}_n^{(l)} &= \textbf{Multi-Head-Attention}\left(z_1^{(l)} \ldots z_n^{(l)},\; z_1^{(l)} \ldots z_n^{(l)}\right) \\
\bar{z}_i^{(l)} &= \textbf{Layer-Normalization}\left(z_i^{(l)} + \textbf{Drop}\left(\tilde{z}_i^{(l)}\right)\right) && \forall i \in [n] \\
z_i^{(l+1)} &= \textbf{Layer-Normalization}\left(\bar{z}_i^{(l)} + \textbf{Drop}\left(\textbf{Fat-ReLU}\left(\bar{z}_i^{(l)}\right)\right)\right) && \forall i \in [n]
\end{aligned}$$

1.2 The Decoder

Let $y_0 \in \mathbb{R}^{512}$ denote a reserved BOS embedding. At each step $j = 0 \ldots$ of generation, the decoder auto-regressively uses $y_0 \ldots y_j \in V^O$. The initial input represenations $o_1^{(0)} \ldots o_j^{(0)} \in \mathbb{R}^{512}$ are obtained by $o_i^{(0)} = \textbf{Drop}\left(E^O_{y_i} + \pi_i\right)$. At each $l = 0 \ldots 5$, we will induce $o_1^{(l+1)} \ldots o_j^{(l+1)} \in \mathbb{R}^{512}$ from $o_1^{(l)} \ldots o_j^{(l)}$ as follows.

$$\begin{aligned}
\tilde{o}_1^{(l)} \ldots \tilde{o}_j^{(l)} &= \textbf{Multi-Head-Attention}\left(o_1^{(l)} \ldots o_j^{(l)},\; o_1^{(l)} \ldots o_j^{(l)}\right) \\
\bar{o}_i^{(l)} &= \textbf{Layer-Normalization}\left(o_i^{(l)} + \textbf{Drop}\left(\tilde{o}_i^{(l)}\right)\right) && \forall i \in [j] \\
\underline{o}_1^{(l)} \ldots \underline{o}_j^{(l)} &= \textbf{Multi-Head-Attention}\left(\bar{o}_1^{(l)} \ldots \bar{o}_j^{(l)},\; z_1^{(6)} \ldots z_n^{(6)}\right) \\
\hat{o}_i^{(l)} &= \textbf{Layer-Normalization}\left(\bar{o}_i^{(l)} + \textbf{Drop}\left(\underline{o}_i^{(l)}\right)\right) && \forall i \in [j] \\
o_i^{(l+1)} &= \textbf{Layer-Normalization}\left(\hat{o}_i^{(l)} + \textbf{Drop}\left(\textbf{Fat-ReLU}\left(\hat{o}_i^{(l)}\right)\right)\right) && \forall i \in [j]
\end{aligned}$$

*Readable transforamtion of Attention is All You Need (Vaswani et al., 2017)

The distribution over words at position $j+1$ is given by (note the tying on E^O)

$$p(y_{j+1} = y | x_1 \ldots x_n,\ y_1 \ldots y_j) := \text{softmax}_y \left((E^O)^\top o_j^{(6)} \right) \qquad \forall y \in V^O$$

2 Other Details

Details of how residual connection and dropout can be implemented seem to be negotiable (e.g., see here).

Their "big" models are still 6 layers but twice the dimensions (1024 instead of 512, 4096 instead of 2048) and 16 heads, with dropout rate 0.3. The source and target vocabularies are shared/tied (vocab size 37k for En-De). They use 8 P100 GPUs (single machine). The base models take $\sim$ 1 day to train, the big models take 3-4 days to train (negative log likelihood loss). They use Adam with a learning rate scheduling. They use dropout on certain layers and attention weights—dropout is important for generalization. The decoding is done with beam search (4 beams, length penalty). Keeping dimensions large is also important for good performance. Their big models achieve SOTA MT results with fast training time. They also do constituency parsing experiments and show competitive results.

Acknowledgement. Thanks to Mohammad Rasooli for pointing out residual connection and dropout.

A The Layers

A.1 Multi-Head-Attention

There are 8 different types of attention ("heads").

Parameters

- Query/key/value matrices $W_{h,Q}, W_{h,K}, W_{h,V} \in \mathbb{R}^{64\times 512}$ for each head $h \in [8]$
- Matrices $U_h \in \mathbb{R}^{512\times 64}$ for each head $h \in [8]$

Input

- Attender sequence $x_1 \dots x_n \in \mathbb{R}^{512}$
- Attendee sequence $y_1 \dots y_m \in \mathbb{R}^{512}$

Output

- $\mu_1 \dots \mu_n \in \mathbb{R}^{512}$: a transformation of $x_1 \dots x_n$ *proactively* sensitive to $y_1 \dots y_m$

Forward pass

1. Embed all vectors into 64-dimensional query/key/value representations:

$$\begin{array}{llll} x_i^{(h,q)} = W_{h,Q}x_i & x_i^{(h,k)} = W_{h,K}x_i & x_i^{(h,v)} = W_{h,V}x_i & \forall i \in [n]\ h \in [8] \\ y_i^{(h,q)} = W_{h,Q}y_i & y_i^{(h,k)} = W_{h,K}y_i & y_i^{(h,v)} = W_{h,V}y_i & \forall i \in [m]\ h \in [8] \end{array}$$

2. An attender uses its "query" on the "key" of each attendee

$$\alpha_{i,j}^{(h)} = \frac{x_i^{(h,q)} \cdot y_j^{(h,k)}}{\sqrt{64}} \qquad \forall i \in [n],\ j \in [m]$$

3. And uses $\left(\beta_{i,1}^{(h)} \dots \beta_{i,m}^{(h)}\right) = \text{softmax}\left(\alpha_{i,1}^{(h)} \dots \alpha_{i,m}^{(h)}\right)$ to combine the "values":

$$\mu_i^{(h)} = \sum_{j=1}^{m} \beta_{i,j}^{(h)} y_j^{(h,v)} \qquad \forall i \in [n]$$

4. Combine heads to get final 512-dimensional attender representations:

$$\mu_i = \sum_{h=1}^{8} U_h \mu_i^{(h)} \qquad \forall i \in [n]$$

A.2 Layer-Normalization

Parameters

- "Gain" $g \in \mathbb{R}^{512}$ and bias $b \in \mathbb{R}^{512}$

Input

- Vector $x \in \mathbb{R}^{512}$

Output

- "Whitened" vector $\bar{x} \in \mathbb{R}^{512}$

Forward pass

1. Compute the mean μ and standard deviation σ of $\{x_1 \ldots x_{512}\}$.
2. Return

$$\bar{x} = g \odot \frac{x - \mu}{\sigma + 0.001} + b$$

A.3 Fat-ReLU

Parameters

- Matrices $F_1 \in \mathbb{R}^{2048 \times 512}$ and $F_2 \in \mathbb{R}^{512 \times 2048}$
- Vectors $b_1 \in \mathbb{R}^{2048}$ and $b_2 \in \mathbb{R}^{512}$

Input

- Input vector $x \in \mathbb{R}^{512}$

Output

- Same-dimensional output vector $y \in \mathbb{R}^{512}$

Forward pass

$$y = F_2 \max\{0, F_1 x + b_1\} + b_2$$