Normalization of transformers

GLyC

October 9, 2025

We will define two notions of normalization for transformers. Both concern the value of the last vector before finishing the loop, which the paper calls h_n^L. We will also require that the output matrix be a projection.

0-1 Normalization

Definition

We will say that a transformer T_N is the 0-1 normalization of a transformer T if it accepts exactly the same words and the h_n^L of T_N respects:

$$(h_n^L)_i = \begin{cases} T_{NO} & \text{if } i = 1 \\ T_{YES} & \text{if } i = 2 \\ 0 & \text{cc} \end{cases}$$

where T_{NO} and T_{YES} correspond to the values of the $OUTPUT$ function before the softmax:

$$T_{NO} := (\Theta_{OUTPUT}(h_n^L))_1$$
$$T_{YES} := (\Theta_{OUTPUT}(h_n^L))_2$$

Implementation

Let T be a transformer, we will construct its 0-1 normalized. The idea is to add some layers at the end of T such that they multiply h_n^L by Θ_{OUTPUT}. Since the dimensions doesn't match, it's needed to extend the matrix with zeros obtaining $\Theta'_{OUTPUT} \in \mathbb{R}^{d\times d}$.

$$\Theta_{OUTPUT} h_n^L = \begin{pmatrix} T_{NO} \\ T_{YES} \end{pmatrix} \longrightarrow \begin{pmatrix} \Theta_{OUTPUT} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} h_n^L = \begin{pmatrix} T_{NO} \\ T_{YES} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

If we are capable to apply the OUTPUT matrix of T inside the layers of the transformer, then the matrix Θ_{OUTPUT} of T_N will be just a projection of the firsts two coordinates:

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \end{pmatrix}$$

Since

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} \Theta_{OUTPUT}h & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} h_n^L =$$

$$= \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} T_{NO} \\ T_{YES} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} T_{NO} \\ T_{YES} \end{pmatrix}$$

It's easy to see that the normalized transformer preserves the behavior.

∞ Normalization

Definition

We will say that a transformer T_N is the ∞-normalization of a transformer T if it accepts exactly the same words and the h_n^L of T_N respects:

$$(h_n^L)_i = \begin{cases} x & \text{if } i = 1 \\ -x & \text{if } i = 2 \\ 0 & \text{cc} \end{cases} \qquad \text{where } x = \begin{cases} -\infty & \text{if } T \text{ rejects} \\ \infty & \text{if } T \text{ accepts} \end{cases}$$

Implementation

The $OUTPUT$ matrix of the $\infty-$normalized transformer will be the same as the 0-1 normalized. Lets analyze now how to get h_n^L to be what we want.

Without loss of generality we can ∞-normalize a 0-1 normalized transformer, so we can assume that $h_n^L = (T_{NO}, T_{YES}, 0, \dots, 0)$

Lets call $x' = T_{YES} - T_{NO}$. If T answers yes, then $x' > 0$ because $T_{YES} > T_{NO}$. If T answers no, then the oposite happens, $x' < 0$ since $T_{NO} > T_{YES}$.

Applying the following linear transformation we get a vector with x' and $-x'$:

$$\begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 1 & -1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} T_{NO} \\ T_{YES} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} T_{YES} - T_{NO} \\ T_{NO} - T_{YES} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} x' \\ -x' \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Because of the finite representation, it happens that for every $z \geq 0$:

$$z * \infty = \begin{cases} 0 & \text{if } z = 0 \\ y & \text{for some } y \geq 1 \text{, if } z \geq 1 \text{ since } z \geq \frac{1}{\infty} \end{cases}$$

Then

$$(z * \infty) * \infty = \begin{cases} 0 & \text{if } z = 0 \\ \infty & \text{if } z \geq 1 \end{cases}$$

In the same way, for all $z \leq 0$ it holds that:

$$(z * \infty) * \infty = \begin{cases} 0 & \text{if } z = 0 \\ -\infty & \text{if } z \geq 1 \end{cases}$$

Therefore, applying twice the linear transformation that multiplies by ∞ gives us what we want.

Same as with the 0-1 normalization, using the projection of the first two coordinates as output matrix of T_N preserves the behavior.

- If T answered no then the first coordinate will be ∞ and the second one $-\infty$ before applying softmax. But for the properties of the finite representation, $e^{-\infty} = 0$ and $e^{\infty} = \infty$. Then the output of the softmax will be $(1, 0)$. So the argmax will pick no.

- In the same way if T answered yes, the output of the softmax will be $(0, 1)$. So the argmax will pick yes.