Primitives of transformers

GLyC

October 9, 2025

Embedding with flags

Assume there is a transformer T with embedding size d. It's possible to add flags expanding the embedding size. The flags are added using the positional embedding. It's needed to extend the dimensions of all the matrix and the token embedding, but doing it with zeros works.

Since the positional embedding from T is just a mapping from n_max to $\mathbb{R}^d$ for doing the flagging it's enough to define the positional embedding of the transformer with flags as:

$$(\theta'_{TE}(n))_j = \begin{cases} (\theta_{TE}(n))_j & \text{if } j \neq d+1 \\ 1 & \text{if } j = d+1 \text{ and } n = i \\ 0 & \text{if } j = d+1 \text{ and } n \neq i \end{cases}$$

Add one specific vector to the ith

The i is fixed from the construction of the transformer. We can use a coordinate (lets pick the last one) for flagging witch is the ith vector. This vector will have a 1 in this coordinate and the others will have a 0.

Now lets define a FF layer such that $FF(h_j) = \mathbb{I}(j = i) * v$.

$$W_1 = id \qquad W_2 = \begin{pmatrix} 0 & \dots & 0 & v_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & v_d \end{pmatrix} \qquad b_1 = b_2 = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

If we were to pick the kth coordinate for the flag then we have to put v in the kth column.

Make the ith vector zero

For making a vector zero we will use the limits of the finite representation.

For all x it happens that $(x + \infty) + \infty = \infty$. Since $x \geq -\infty$, it happens that $x + \infty \geq 0$, then $(x + \infty) + \infty \geq \infty$, so $(x + \infty) + \infty = \infty$. Therefore if we want to get zero, it's enough to subtract ∞.

$$h_i \leftarrow \left(\left(h_i + \begin{pmatrix} \infty \\ \vdots \\ \infty \end{pmatrix} \right) + \begin{pmatrix} \infty \\ \vdots \\ \infty \end{pmatrix} \right) + \begin{pmatrix} -\infty \\ \vdots \\ -\infty \end{pmatrix}$$

ATTN y FF controlling which coordinates are affected

Later we will see that we want to go trough layers of attention and feed forward ignoring some coordinates. That means without taking them into consideration and without affecting them in the output.

For example, if we want to ignore the ith coordinate trough a layer of attention, it's enough to put zeros in the ith column and row of W_Q, W_K, W_V, W_O. This can be done with as many coordinates as desired. Note that because the ith row of W_O is zero, the output vector of the ATTN layer will have a zero in the ith coordinate. This is the point that makes it possible to not just ignore the value of the ith coordinate but to not affect it. This happens because the result of the attention is added to the original vector.

The same applies in the feed forward layer, putting zeros in the ith column and row of the matrix and vectors of the FF makes it ignore the value of the ith coordinate. Also in the result of the FF will appear a zero in the ith coordinate.

Linear transformations

In this section we will see how to apply a linar transformation to one of the vectors and save the output in another more to the right, in other words: $h_j^{l+c} = Mh_i^l$ for some $i < j$ previously chosen.

Note that after this procedure all the vector will have their values corrupted with the exception of h_j.

First we will make $h_j = 0$ at the coordinates we are interested. Then with an ATTN layer we will add Mh_i.

Thanks to the previous sections we can assume that h_i is the only vector which has $(1, 0)$ in the lasts coordinates and that the rests end in $(0, 1)$. These numbers would have been set by the encoding and preserved trough the previous layers.

Assuming this will allow us to construct W_K and W_Q such that:

$$\langle q_j, k_{i'} \rangle = \begin{cases} -\infty & \text{if } i' \neq i \\ 0 & \text{if } i' = i \end{cases}$$

Note that the W_Q and W_K that we are looking for are the following:

$$W_Q = \begin{pmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & -\infty \end{pmatrix} \qquad W_K = \begin{pmatrix} 0 & \dots & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

since

$$q_j = W_Q\ h_j = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -\infty \end{pmatrix} \qquad k_{i'} = W_K\ h_{i'} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ (h_{i'})_d \end{pmatrix}$$

so $\langle q_j, k_{i'} \rangle = -\infty * (h_{i'})_d$

Taking $W_O = M$ y $W_V = id$ we get what we were looking for. This happens because the fact that $e^{-\infty} = 0$ makes that $softmax(-\infty, \dots, -\infty, 0) = (0, \dots, 0, 1)$, which makes $(s_j)_{i'} = \mathbb{I}(i' = i)$ so finally:

$$W_O \sum_{i'=0}^{n} (s_j)_{i'} v_{i'} = W_O \sum_{i'=0}^{j} (s_j)_{i'} v_{i'} = W_O \sum_{i'=0}^{j} (s_j)_{i'} h_{i'} = W_O \sum_{i'=0}^{j} \mathbb{I}(i' = i) h_{i'} = W_O\ h_i$$