Music Transformer

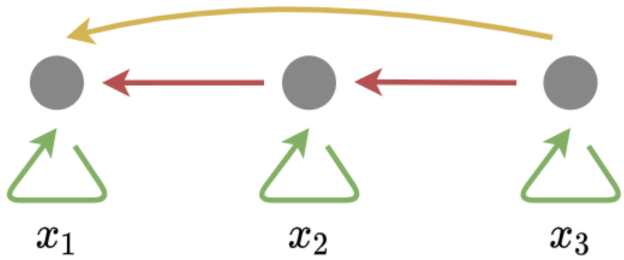
Table of Contents

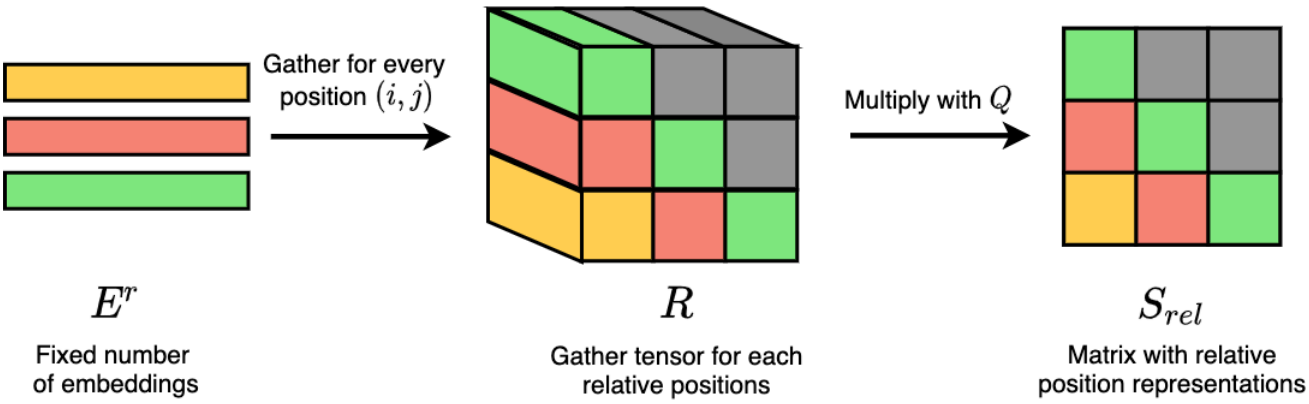
Understanding Relative Positional Attention in Transformers

Suppose we have a musical sequence of notes of length L, then we find the pairwise relative distance between the i^{th} and j^{th} elements of the sequence and obtain a matrix of shape $L * L$. Now we can limit the relative distance between two elements such that the distances are within manageable range. We clip the values of this matrix to the range `[-max_relative_distance, max_relative_distance]`, where `max_relative_distance` can be thought of as the range or window till which an element in the sequence can see. Since embedding indices of the matrix should be non-negative, we shift the clipped distances by adding `max_relative_distance`.

Our shifted matrix will be of shape $L * L$, then if we want we can expand this matrix to have H number of heads, such that the new shape of the matrix can be $H * L * L$. Next, we initialize a learnable matrix $\mathbb{E}$ with `[2*max_relative_distance+1, D]`, where D is the embedding dimension. This matrix maps the relative distances to their corresponding embedding vectors. The relative positional embedding matrix $\mathbb{R}$ is constructed by indexing $\mathbb{E}$ into the initialized matrix based on the relative distances in the shifted matrix. The shape of this matrix comes out to be $L * L * D$. We use the shifted matrix E as a look up table, retrieve the relative distance k from the pair (i, j) to index and retrieve the embedding $\mathbb{E}[k, :]$. This gives us our relative positional embedding matrix $\mathbb{R}$.

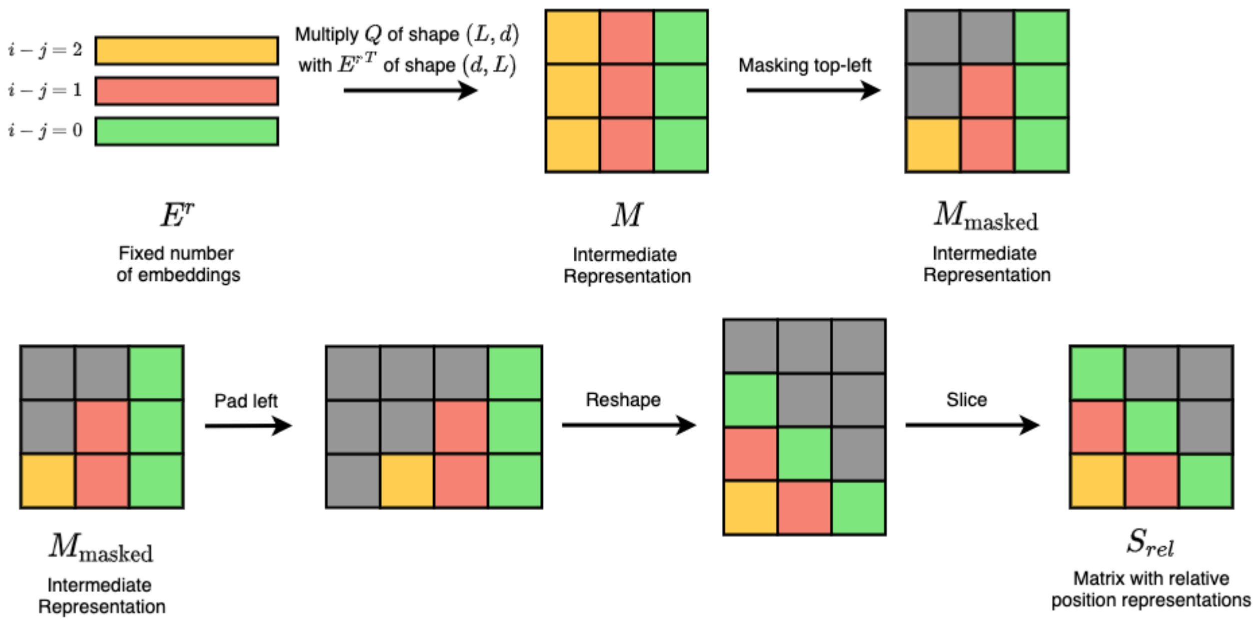
For each head, we have the $\mathbb{R}$ matrix which we multiply with $\mathbb{Q}$ to obtain S_{rel}. The overhead is that this total computation takes $O(L^2.D)$ space complexity and hence restricting its use for longer sequences.





Relative Global Attention - Music Transformers

The main problem in computing S_{rel} comes from explicitly calculating the relative distances for all pairs of i and j in the sequence. This involves indexing into the embedding matrix $\mathbb{E}$ for every pair (i, j). Instead in the paper, they implement an interesting trick by initialized the embedding matrix $\mathbb{E}$ with the embedding dimension D , which results in a matrix , assume $\mathbb{M}$, of shape $L * D$ and instead of gathering the relative information for every i and j as in the previous method, they directly multiply $\mathbb{M}$ with $\mathbb{Q}$ to obtain an $L * L$ matrix. The trick is that they don't calculate the relative position for each i and j, instead we do it for i and a relative offset$r$ and then using the skew procedure in the paper we map r to j. This results in the reduced complexity due to the shapes of the matrices.



Skewing Procedure

The algorithmic approach to obtain the equivalent matrix S_{rel} is to first pad the the masked matrix M with a buffer so that no values are cut out or fall off the matrix resulting in preserving the relative alignment. We reshape the matrix and map the indices using the formula $j_k = r - (L-1) + i_q$ from a absolute-relative indexing to absolute-absolute indexing. We then take the last L rows of the matrix to form the matrix S_{rel} of shape $L * L$.

References

1. Music Transformers
2. Self-Attention with Relative Position Representations
3. Hao Hao Tan's Blog Post
4. Self-Attention-with-Relative-Position-Representations Github
5. Music Transformer Pytorch