Transformer

目标任务：sequence transduction

use learned embeddings to convert the input tokens and output tokens to vectors of dimension dmodel.

use the usual learned linear transformation and softmax function to convert the decoder output to predicted next-token probabilities.

创新点

在网络架构中完全基于 attention mechanisms

注意力机制可以不受序列中的距离限制进行建模

摒弃一般研究所用的 recurrence and convolutions

RNN中固有的顺序性质限制了并行化

CNN中长程依赖也是个问题

优势

be superior in quality

确实在结果上有巨大的进步

be more parallelizable

require significantly less time to train

这一点不好说，训练依旧耗时

encoder-decoder structure

The encoder maps an input sequence of symbol representations (x1, ..., xn) to a sequence of continuous representations
z = (z1, ..., zn).
Composed of a stack of N = 6 identical layers, each layer has two sub-layers

The first is a multi-head self-attention mechanism

The second is a simple, position-wise fully connected feed-forward network

consists of two linear transformations with a ReLU activation in between.

While the linear transformations are the same across different positions, they use different parameters from layer to layer.

Each sub-layer's output is:
LayerNorm(x + Sublayer(x))

Employ a residual connection

Followed by layer normalization

All sub-layers in the model, as well as the embedding layers, produce outputs of dimension dmodel = 512.

At each step the model is auto-regressive, consuming the previously generated symbols as additional input when generating the next

Given z, the decoder then generates an output sequence (y1, ..., ym) of symbols one element at a time.
Composed of a stack of N = 6 identical layers, each layer has THREE sub-layers

Addition inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack

Also modify the self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent position(防止位置关注后续位置).

by masking out (setting to −∞) all values in the input of the softmax which correspond to illegal connections.

Scaled Dot-Product Attention

While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.

We suspect that for large values of dk, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients 4. To counteract this effect, we scale the dot products by 1√dk

To illustrate why the dot products get large, assume that the components of q and k are independent random variables with mean 0 and variance 1. Then their dot product, q · k = ∑qiki, has mean 0 and variance dk

Multi-Head Attention

it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to dk, dk and dv dimensions, respectively.

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions.

$$\mathrm{MultiHead}(Q,K,V)=\mathrm{Concat}(\mathrm{head}_1,\ldots,\mathrm{head}_{\mathrm{h}})W^O$$
$$\text{where head}=\mathrm{Attention}\left(QW_i^Q,KW_i^K,VW_i^V\right)$$

$$W_i^Q\in\mathbb{R}^{d_{\text{model}}\times d_k},W_i^K\in\mathbb{R}^{d_{\text{model}}\times d_k},W_i^V\in\mathbb{R}^{d_{\text{model}}\times d_v},W^O\in\mathbb{R}^{hd_v\times d_{\text{model}}}$$

Positional Encoding

add "positional encodings" to the input embeddings at the bottoms of the encoder and decoder stacks. The positional encodings have the same dimension d-model as the embeddings, so that the two can be summed.

$$PE_{(\text{pos},2i)}=\sin\left(\text{pos}/10000^{2i/d_{\text{model}}}\right)$$
$$PE_{(\text{pos},2i+1)}=\cos\left(pos/10000^{2i/d_{\text{model}}}\right)$$

where pos is the position and i is the dimension

We chose this function because we hypothesized it would allow the model to easily learn to attend by relative positions, since for any fixed offset k, PEpos+k can be represented as a linear function of PEpos.

it may allow the model to extrapolate to sequence lengths longer than the ones encountered during training.

组件训练结果评估

While single-head attention is 0.9 BLEU worse than the best setting, quality also drops off with too many heads.

replace our sinusoidal positional encoding with learned positional embeddings, observe nearly identical results to the base model.

reducing the attention key size dk hurts model quality. This suggests that determining compatibility is not easy and that a more sophisticated compatibility function than dot product may be beneficial.

bigger models are better, and dropout is very helpful in avoiding over-fittin