Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English- to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data. *Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research. +Work performed while at Google Brain. ++Work performed while at Google Research. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. arXiv:1706.03762v7 [cs.CL] 2 Aug 2023

Introduction

Recurrent neural networks, long short-term memory and gated recurrent neural networks have been established as state of the art approaches in sequence modeling and transduction problems. Recurrent models typically factor computation along the symbol positions of the input and output elements. The results of this study will be published in the next issue of the journal Machine Learning.The inherently sequential nature of training examples precludes parallelization within training examples. Memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks and conditional co. ut sequences. For example, we can generate a sequence of hidden states ht, as a function of the previous hidden state ht-1 and the input for position t. attention mechanisms have become an integral part of compelling sequence modeling and transduc- tion models in various tasks. In all but a few cases, attention mechanisms are used in conjunction with a recurrent network. The fundamental constraint of sequential computation, however, remains. In this work we use attention mechanisms to model mputation.

Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S. All of these use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions. This makes it moSelf-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions. We counteract with Multi-Head Attention as described in section 3.2.End-to-end memory networks are based on a recurrent attention mechanism instead of sequence- aligned recurrence. They have been shown to perform well on simple-language question answering and language modeling tasks. To the best of our knowledge, however, the Transformer is the first transduction model relying on recurrent attention.

Model Architecture

Most competitive neural sequence transduction models have an encoder-decoder structure. Here, the encoder maps an input sequence of symbol representations to a sequence of continuous representations. Given z, the decoder then generates an output sequence of symbols one element at a time. At each step the model is auto-regressive, consuming the previously generated symbols as additional input when generating the next.

Encoder and Decoder Stacks

The encoder is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position- wise fully connected feed-forward network.The decoder is also composed of a stack of N = 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack.

Attention

Scaled Dot-Product Attention

Multi-Head Attention

Scaled Dot-Product Attention consists of several attention layers running in parallel. The weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The input consists of queries and keys of dimension dk, and values ofdimension dv.In practice, we compute the attention function on a set of queries simultaneously. The keys and values are also packed together into matrices K and V. We compute the matrix of outputs as: Attention(Q, K, V) = softmax(QKT dk)V (1) The two most commonly used attention functions are additive attention and dot-product attention.Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice. Additive attention outperforms dot product attention without sc for small values of dk.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. To counteract this effect, we scale the dot product by 1 dk . 3.2.2 Multi-Head Attention Instead of performing a single attention function with dmodel-dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projection.On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding dv-dimensional 4. To illustrate why the dot products get large, assume that the components of q and k are independent random variables. Then their dot product, q * k = dk i=1 qiki, has mean 0 and variance dk. 4.Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this. In this work we employ h = 8 parallel attention layers, or heads. For each of thes, we employ a different attention layer.

Applications of Attention in our Model

The Transformer uses multi-head attention in three different ways. In "encoder-decoder attention" layers, the queries come from the previous decoder layer. This allows every position in the decoder to attend over all positions in the input sequence. The encoder contains self-attention layers.The keys, values and queries come from the same place, in this case, the output of the previous layer in the encoder. We need to prevent leftward information flow in the decoder to preserve the auto-regressive property. We implement this inside of scaled

Position-wise Feed-Forward Networks

Each of the layers in our encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically. This 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.

Embeddings and Softmax

We use learned embeddings to convert the input tokens and output tokens to vectors of dimension dmodel. We also use the usual learned linear transfor- mation and softmax function. In our model, we share the same weight matrix between the two embedding layers and the pre-softmax linear transformation. ximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Positional Encoding

In order for the model to make use of the order of the sequence, we must inject some information about the relative or absolute position of the tokens in the sequence. To this end, we add "positional encodings" to the input embeddings at the bottoms of the encoder and decoder stacks.We use sine and cosine functions of different frequencies. Each dimension of the positional encoding corresponds to a sinusoid. The wavelengths form a geometric progression from 2p to 10000 * 2p. We chose this function because we hypothesized it would allow the model to easily learn to attend by relative positions.

Why Self-Attention

In this section we compare various aspects of self-attention layers to the recurrent and convolu- tional layers commonly used for mapping one variable-length sequence of symbol representations (x1, ..., xn) to another sequence of equal length (z1,..., zn) such as a hidden layer in a typical sequence transduction encoder or decoder.The shorter the path length between any combination of positions in the input and output sequences, the shorter these paths can be parallelized. Learning long-range dependencies is a key challenge in many sequence transduction tasks. One key factor affecting the ability to learn such dependencies is the length of the paths forward and backward signals have to traverse.Self-attention layers connect all positions with a constant number of sequentially executed operations. A recurrent layer requires O(n) sequential operations. In terms of computational complexity, self-att attention layers are faster than recurrent layers when the sequence 6 length n is smal.Self-attention could be restricted to considering only a neighborhood of size r in the input sequence centered around the respective output position. This would increase the maximum path length to O(n/r). We plan to use this technique to improve computational performance for tasks involving very long sequences.A single convolutional layer with kernel width k < n does not connect all pairs of input and output positions. Convolutional layers are generally more expensive than recurrent layers, by a factor of k. It is possible to investigate this approach further in future work. The complexity of a separable convolution is equal to the combination of a self-attention layer and a point-wise feed-forward layer. We inspect attention distributions from our models and present and discuss examples in the appendix. Not only do individual attention heads clearly learn to perform diffs, but they also learn to act on each other. As side benefit, self-Attention could yield more interpretable models.

Training

Training Data and Batching

We trained on the standard WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs. Sentences were encoded using byte-pair encoding [3], which has a shared source- target vocabulary of about 37000 tokens. For English-French, we used the significantly larger WMT2014 English- French dataset consisting. of 36M sentences and split tokens into a 32000 word-piece vocabulary.

Hardware and Schedule

Optimizer

Regularization

The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English- to-French newstest2014 tests at a fraction of the training cost. We employ three types of regularization during training: 7We apply dropout [33] to the output of each sub-layer, before it is added to the sub- layer input and normalized. In addition, we apply drop out to the sums of the embeddings and the positional encodings in both the encoder and decoder stacks.

Results

Machine Translation

On the WMT 2014 English-to-German translation task, the big transformer model (Transformer (big) in Table 2) outperforms the best previously reported models (including ensembles) by more than 2.0 BLEU. The configuration of this model is listed in the bottom line of Table 3. Training took 3.5 days on 8 P100 GPUs.The Transformer (big) model trained for English-to-French used dropout rate Pdrop = 0.1, instead of 0.3. For the base models, we used a single model obtained by averaging the last 5 checkpoints, which were written at 10-minute intervals. The big model achieves a BLEU score of 41.0, outperforming all of the previous single models.We used beam search with a beam size of 4 and length penalty a = 0.6. We set the maximum output length during inference to input length + 50, but terminate early when possible. We estimate the number of floating point operations used to train a model by multiplyin.

Model Variations

We used values of 2.8, 3.7, 6.0 and 9.5 TFLOPS for K80, K40, M40 and P100. All metrics are on the English-to-German translation development set, newstest2013. Listed perplexities are per-wordpThe steps are based on the byte-pair encoding, and should not be compared to per-word perplexities. N dmodel dff h dk dv Pdrop ls train PPL BLEU params steps (dev) (Dev) x106 base 6 512 2048 8 64 64 0.1 0. 1 100K 4.92 25.8 65 (A) 1 512 512 5.29 24.9 4 128 128 5.00 25.5 32 16 16 5.01 25.4 60 (C) 2 6.11 23.7 36 4 5.19 25.3 50 8 4.88 24.5 28 1024 128 128 4.66 26We used beam search as described in the previous section, but no checkpoint averaging. We present these results in Table 3. In Table 3 rows (A), we vary the number of attention heads and the attention key and value dimensions, keeping the amount of computation constant. While single-head attention is 0.9 BLEU worse than the best, it is still better than the worst.In Table 3 rows (B), we observe that 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. We further observe in rows (C) and (D) that, as expected, bigger models are better.

English Constituency Parsing

To evaluate if the Transformer can generalize to other tasks we performed experiments on English constituency parsing. This task presents specific challenges: the output is subject to strong structural constraints and is significantly longer than the input. We trained a 4-layer transformer with dmodel = 1024 on the Wall Street Journal (WSJ) portion of the Penn Treebank.We trained it in a semi-supervised setting using the larger high-confidence and Berkley corpora from with approximately 17M sentences. We used a vocabulary of 16K tokens for the WSJ only setting and 32K token for the semi- supervised setting. We performed only a small number of experiments to select the dropout, both attention and residual (section 5.4)The Transformer generalizes well to English constituency parsing (Results are on Section 23 of WSJ) Parser Training WSJ 23 F1 Vinyals & Kaiser el al. anged from the English-to-German base translation model. During inference, we 9 Table 4:We used a beam size of 21 and a = 0.3 for both WSJ only and the semi-supervised setting. Our results in Table 4 show that despite the lack of task-based learning, we were able to achieve a high level of learning. We used the same beam size but increased the maximum output length from 21 to 300.

Conclusion

The Transformer is the first sequence transduction model based entirely on attention. It replaces the recurrent layers most commonly used in encoder-decoder architectures. For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers.We plan to extend the Transformer to problems involving input and output modalities other than text. We plan to investigate local, restricted attention mechanisms to efficiently handle large inputs and outputs such as images, audio and video. We are excited about the future of attention-based models and plan to apply them to other tasks.We are grateful to Nal Kalchbrenner and Stephan Gouws for their fruitful comments, corrections and inspiration. The code is available on GitHub at: http://www.tensorflow.com/ tensorflow/tensor2tensor. ls.

Figures

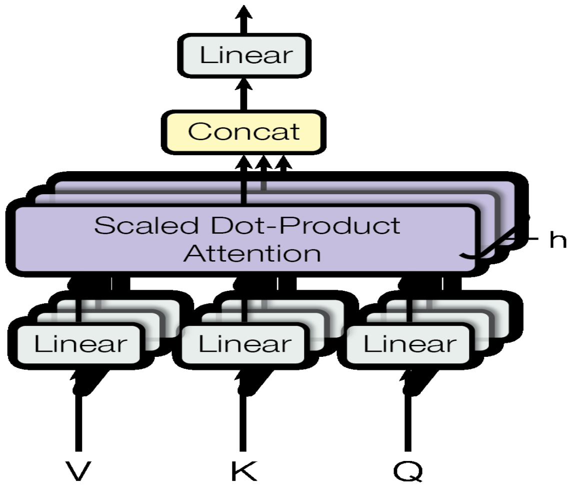

Figure 1

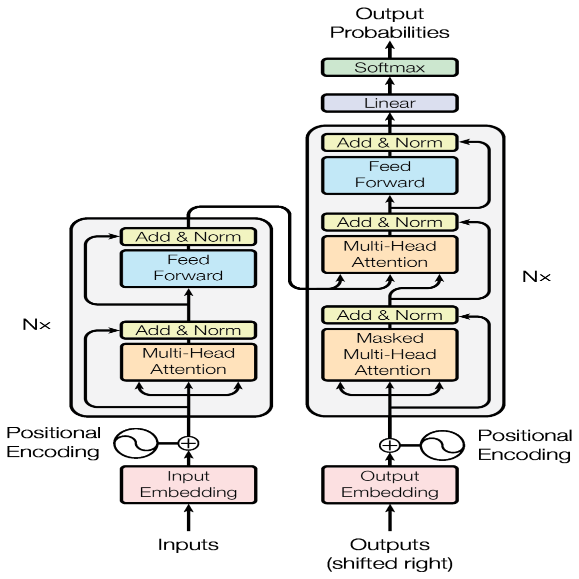

Figure 2

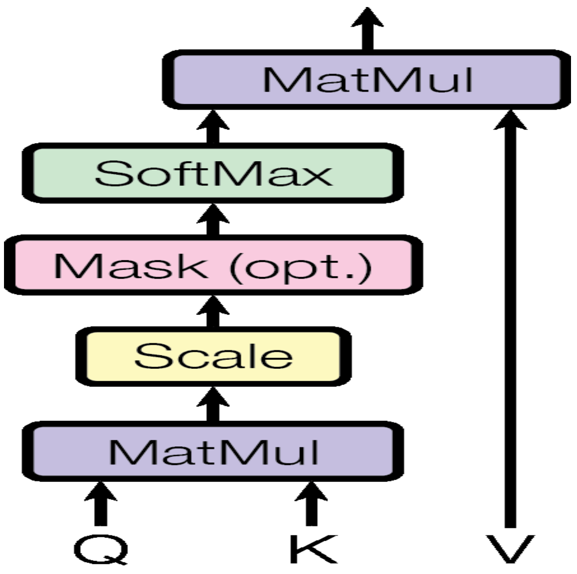

Figure 3