ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning, pp. 43-48. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/

Image-to-Text Transduction with Spatial Self-Attention

Sebastian Springenberg, Egor Lakomkin, Cornelius Weber and Stefan Wermter *

University of Hamburg - Dept. of Informatics, Knowledge Technology
Vogt-Kölln-Straße 30, 22527 Hamburg - Germany

Abstract. Attention mechanisms have been shown to improve recurrent encoder-decoder architectures in sequence-to-sequence learning scenarios. Recently, the Transformer model has been proposed which only applies dot-product attention and omits recurrent operations to obtain a source-target mapping [5]. In this paper we show that the concepts of self- and inter-attention can effectively be applied in an image-to-text task. The encoder applies pre-trained convolution and pooling operations followed by self-attention to obtain an image feature representation. Self-attention combines image features of regions based on their similarity before they are made accessible to the decoder through inter-attention.

1 Introduction and Background

Sequence-to-sequence learning, involving the mapping of two variable-length sequences, has been an active area of research in the last decade. Recently, neural encoder-decoder architectures have become be the predominant approach to this task, yielding state-of-the-art results while being trainable in an end-to-end fashion. Such architectures usually consist of two sub-networks, namely an encoder and a decoder. The encoder, typically a recurrent neural network (RNN), first encodes the source sentence and produces a context vector. This vector is then decoded by the decoder RNN to predict a target sequence [1]. First compressing source information into an intermediate vector representation and then extracting target information from it allows mapping any type of sequences independently of their representation and length.

However, one often occurring problem in such architectures is a loss of information due to the compression step. This becomes especially evident when dealing with long and complex sequences where all the necessary information cannot be efficiently stored in a single, fixed-length context vector. Attention mechanisms have been shown to improve this issue by considering not only one, but distinct context vectors during decoding [4, 2]. These context vectors are obtained by a weighted combination of the source sentence constituents. The weights applied are often called attention weights, indicating that the model is able to dynamically allocate attention to the most relevant parts of the source sequence at every decoding step.

While such models yield good results, they tend to take a long time to train due to the recurrent computations in both the encoder and the decoder that can-

*This research was partially supported by the German Research Foundation (DFG) under project Crossmodal Learning (TRR 169) and the EU under project SECURE (No 642667).

not be parallelised. This drawback becomes especially severe when facing long sequences. In order to circumvent such limitations, Kalchbrenner et al. proposed the Bytenet model [6], which does not make use of any recurrence during training by applying dilated convolution and stacking a dynamically unfolding decoder on top of the encoder. This allows producing the whole target sequence prediction in one pass, where computation scales linearly with sequence length.

Vaswani et al. recently proposed a model that even surpasses Bytenet in terms of performance and computational efficiency in a neural machine translation task by eschewing recurrence and convolution altogether [5]. This model, called the Transformer, relies only on attention operations to transduce sequences. *Self-attention* serves as a way to obtain a feature representation (encoding) based on the similarity of sequence constituents. *Inter-attention* allows information flow from the encoder to the decoder where, similar to previous attention mechanisms, attention can be allocated to relevant parts of the source sequence.

While the Transformer has only been used in text-to-text learning scenarios, we show that a similar architecture can effectively be applied to images as well in a captioning task where multiple observations from an image are obtained to describe the depicted scene. Self-attention on image features appears to be supportive here as it enables the model to group and combine regions with coherent content before making them accessible to the decoder. We show that an attention-based architecture can be a faster-to-train alternative to approaches that apply a recurrent encoder-decoder architecture with attention mechanism [3].

2 Model Architecture

Our proposed model builds upon the transformer model by Vaswani et al. [5] but has modifications mainly on the encoder side, enabling the processing of images instead of text and allowing to compute spatial self-attention. The overall architecture is depicted in figure 1 and consists of an encoder and a decoder. While the encoder obtains a *source encoding*, the decoder obtains a *target encoding* and combines both encodings via *source-target mapping*. While source and target encoding both involve self-attention, source-target mapping involves inter-attention and thus permits information flow from encoder to decoder.

2.1 Encoder

Source Encoding: The encoder contains two sequentially arranged self-attention layers each involving multi-head self-attention followed by a feed-forward operation. A detailed description of the attention operations can be found in section 2.3. The output of the second self-attention layer is made accessible to the decoder during inter-attention. Before applying self-attention, the input image is first fed to a pre-trained convolutional neural network (CNN) to extract descriptive image features. Since pooling and standard convolution operations diminish the spatial resolution, feature maps of VGG-11 net [7] are used after a convolution layer that still retains a spatial resolution of 14 × 14. Computed image features of these regions are subsequently equipped with a positional encoding and then passed to the self-attention layers. As positional encoding, the same sinusoidal function as presented in [5] is used.

2.2 Decoder

The decoder consists of four layers each involving *target encoding* and *source-target mapping*. An embedding of the target caption (shifted one position to the right) followed by positional encoding is obtained before accessing the first decoder layer. The output of the last decoder layer is passed to a linear operation followed by softmax to obtain the final target word predictions.

Target Encoding: Multi-head self-attention is also applied to the target caption. Making use of an encoding of the whole target sequence might first appear counterintuitive as the model is desired to produce the target itself in an autoregressive fashion. However, producing one word at a time would prevent parallelisation in the decoder. For this reason, the actual target sequence is fed to the decoder during training instead of the previous emitted output. This resembles a form of permanent teacher forcing. In order to still assure the autoregressive property needed later during inference, attention masking is applied to the self-attention weights. Masking restricts words of the caption to only attend to those at a previous position in the sequence.

Source-Target Mapping: After having obtained a source and a target encoding, multi-head inter-attention between both is applied to combine information. This is followed by a feed-forward operation on the attention result.

2.3 Attention

Attention weights between constituents of two representations can be obtained by first projecting both to a new feature space and then calculating the dot product. Vaswani et al. describe this process as comparing a query to key-value pairs, where queries, keys and values are all linear projections:

$$Q = Q_{input} \cdot W_Q, \quad K = K_{input} \cdot W_K, \quad V = V_{input} \cdot W_V \tag{1}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{h \times d_{feat} \times d_{attn}}$. The number of attention heads is hereby represented by h, d_{feat} represents the number of initial image or text features and $d_{attn} = d_{feat}/h$ is the new feature dimension used to calculate attention. We set $h = 8$, $d_{feat} = 512$ and $d_{attn} = 64$.

After linear projection, attention weights $a_{Q,K}$ are calculated by applying a soft-max operation to the scaled dot product between queries and keys:

$$a_{Q,K} = softmax(\frac{Q \cdot K^T}{\tau}), \quad \tau = \sqrt{d_{len} + d_{attn}} \tag{2}$$

where the scaling factor τ is chosen to account for d_{attn}, as well as for d_{len} which represents the spatial (in case of an image) or temporal (in case of text)

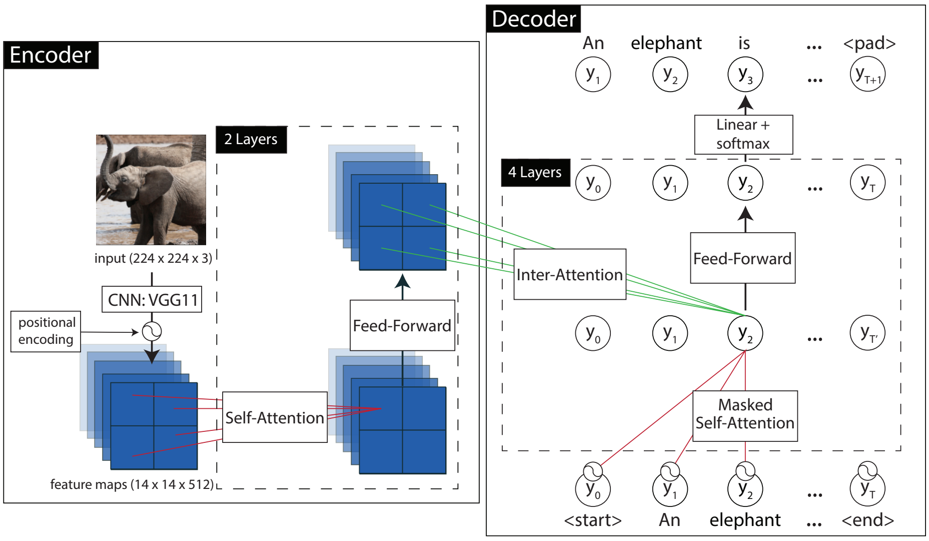

Fig. 1: Model architecture: Self-attention and inter-attention weights for a single query position shown in red and green respectively. For simplicity, image feature maps of $14 \times 14 \times 512$ are depicted as $2 \times 2 \times 5$. Word embeddings also have a feature depth of 512.

dimensionality of the keys. This is different to the implementation of Vaswani et al. which does not involve adding d_{len} when calculating τ. We found increasing τ this way to be beneficial when facing the allocation of soft-attention to many positions as it enforces to spread attention instead of focusing only on a single entry.

Multi-head attention results are subsequently obtained by multiplying attention weights with the values V: $MultiHead = a_{Q,K} \cdot V$, which results in $Multihead \in \mathbb{R}^{h \times d_{len} \times d_{attn}}$. Appending all multi-head attention results along the h dimension and applying a further linear projection results in the final attention output that has the same dimensionality as the initial input to the attention operation:

$$Attn = cat(Multihead_1^{(d_{len} \times d_{attn})}, ..., Multihead_h^{(d_{len} \times d_{attn})}) \cdot W_O \qquad (3)$$

where $W_O \in \mathbb{R}^{d_{feat} \times d_{feat}}$. Finally, a residual connection is applied, followed by layer normalisation [10]: $AttnOut = LayerNorm(Attn + V_{input})$.

Self-attention: In *self-attention*, the similarity between constituents of a single input is to be calculated. Thus, query, keys and values all take the same values: $Q_{input} = K_{input} = V_{input}$.

Inter-attention: In *inter-attention*, the similarity of a target to a source is to be calculated. Thus: $Q_{input} = target$ and $K_{input} = V_{input} = source$.

2.4 Feed-Forward Operation

A feed-forward operation with a Relu activation function is applied to the result of attention in both the encoder and the decoder:

$$FeedForward(x) = Relu(x \cdot W_1 + b_{b1}) \cdot W_2 + b_2 \tag{4}$$

where $W_1 \in \mathbb{R}^{d_{feat}, d_{hidden}}$ and $W_2 \in \mathbb{R}^{d_{hidden}, d_{feat}}$. We set $d_{hidden} = 2 \cdot d_{feat} = 1024$. Afterwards, a residual connection is applied, followed by layer normalisation: $FFOut = LayerNorm(FeedForward(x) + x)$.

3 Results

We train and evaluate the proposed model on image-captions of the MS COCO 2014 dataset [8]. Captions are restricted to have a maximum length of 14 words during training. Optimisation is achieved by reducing the crossentropy loss between predicted and actual target captions using the Adam optimiser [9] with a learning rate of $l = 0.0003$. As a regularisation method, we apply dropout with a probability of $P = 0.1$ to the caption embeddings and image features as well as to the attention weights [5]. The model is trained for only 11 hours on a single NVIDIA Tesla K80 GPU. In our experiments we found training a simple recurrent model to take at least 20 hours to obtain reasonable results.

Table 1 shows quantitative results. Qualitative results of learned attention weights are depicted in figure 2. Spatial inter-attention assigns higher weight to pixel regions that contain relevant information when producing a word. Attention can be distributed independently over the 14×14 regions.

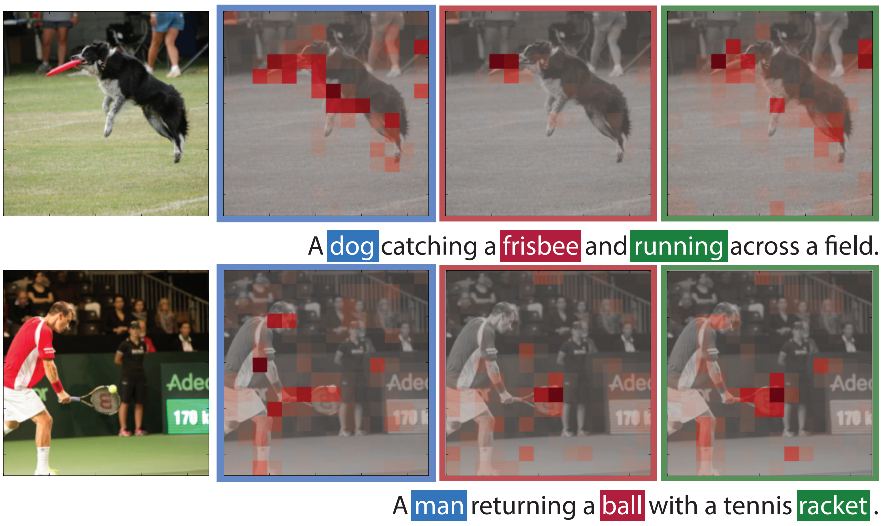
A dog catching a frisbee and running across a field.
A man returning a ball with a tennis racket.

Fig. 2: Inter-attention weights attending to relevant parts of the input image associated with query words of the captions.

Model	Bleu-1	Bleu-2	Bleu-3	Bleu-4	METEOR
Show, Attend and Tell [3]	71.8	50.4	35.7	25	23.9
Proposed Model	66.7	46.4	31.2	22.6	21.8

Table 1: Bleu precision and METEOR score obtained on the MS-COCO 2014 dataset.

4 Conclusion

In this paper we have shown that a network relying primarily on attention operations can efficiently be applied to image captioning. Self- and inter-attention can serve as powerful tools to obtain a rich feature representation while being faster to train than recurrent operations due to parallelisable computation. Further work has to be done to improve the model's performance. Architectural choices encouraging the model to rely more heavily on image features than on previously produced words might improve the quality of generated captions. Also, it might be interesting to investigate the performance when solely applying self-attention to the input images and not making use of convolution at all. However, our interpretation is that convolution serves as a good basis to obtain descriptive image features that can then be combined with self-attention. Applying strided instead of regular convolution could avoid diminishing the spatial dimension, allowing to apply self-attention on a pixel level rather than on pixel regions.

References

[1] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks.", *Advances in Neural Information Processing Systems*, 2014.

[2] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *International Conference on Learning Representations*, 2015.

[3] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." *International Conference on Machine Learning*, 2015.

[4] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation.", 2015.

[5] Vaswani, Ashish, et al., "Attention is all you need.", arXiv preprint arXiv:1706.03762, 2017.

[6] Kalchbrenner, Nal, et al. "Neural machine translation in linear time.", arXiv preprint arXiv:1610.10099, 2016.

[7] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556, 2014.

[8] Lin, Tsung-Yi, et al. "Microsoft COCO: Common objects in context." European Conference on Computer Vision. Springer, Cham, 2014.

[9] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[10] Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer normalization." arXiv preprint arXiv:1607.06450, 2016.