Attention based image to caption using a Transformer based network

Willem Diepeveen, Niek van der Laan, Daphne van Tetering, Paul Verkooijen, Kasper Wendel

TU Delft

Introduction

The canonical method for sequential modelling is the Recurrent Neural Network (RNN). In recent years, other architectures have been proposed and were shown to perform better in several cases. Especially attention to model recurrence got more popular after the paper by Vaswani et al. [1] that proposes the Transformer network. The Transformer network is something special, as it does not use any recurrences nor convolutions making it very attractive due to its high degree of parallelism. The network is purely built up from attention mechanisms and fully connected layers. Although these networks have been tested against the accepted RNN variations, these transformer networks are not the standard yet.

Research objective

Xu et al. [2] used a VGGNet and a LSTM to make an algorithm that turns an image into a caption in their paper *Show, Attend and Tell*. Inspired by this image-to-caption task, our main objective is:

- Implement a trainable VGGNet-Transformer combination for the image-to-caption task;
- Compare the performance with the VGGNet-LSTM combination.

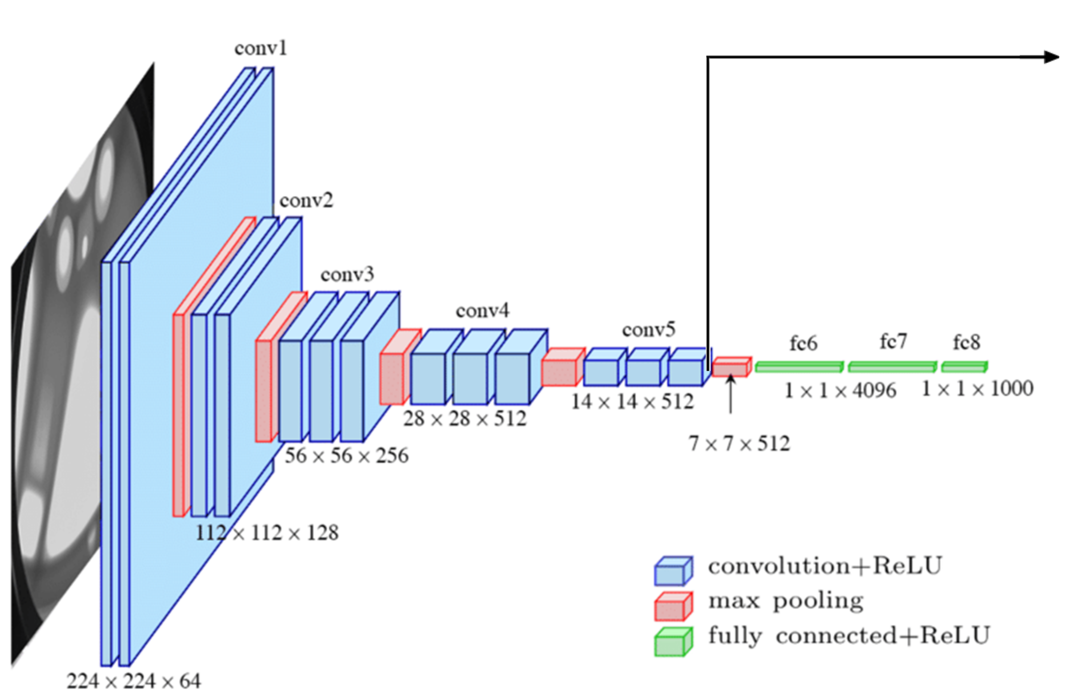

Figure 1: The feature maps obtained after the 5th convolutional layer will be flattened to get the shape 512×196 used as the VGG output of the proposed network.

Model

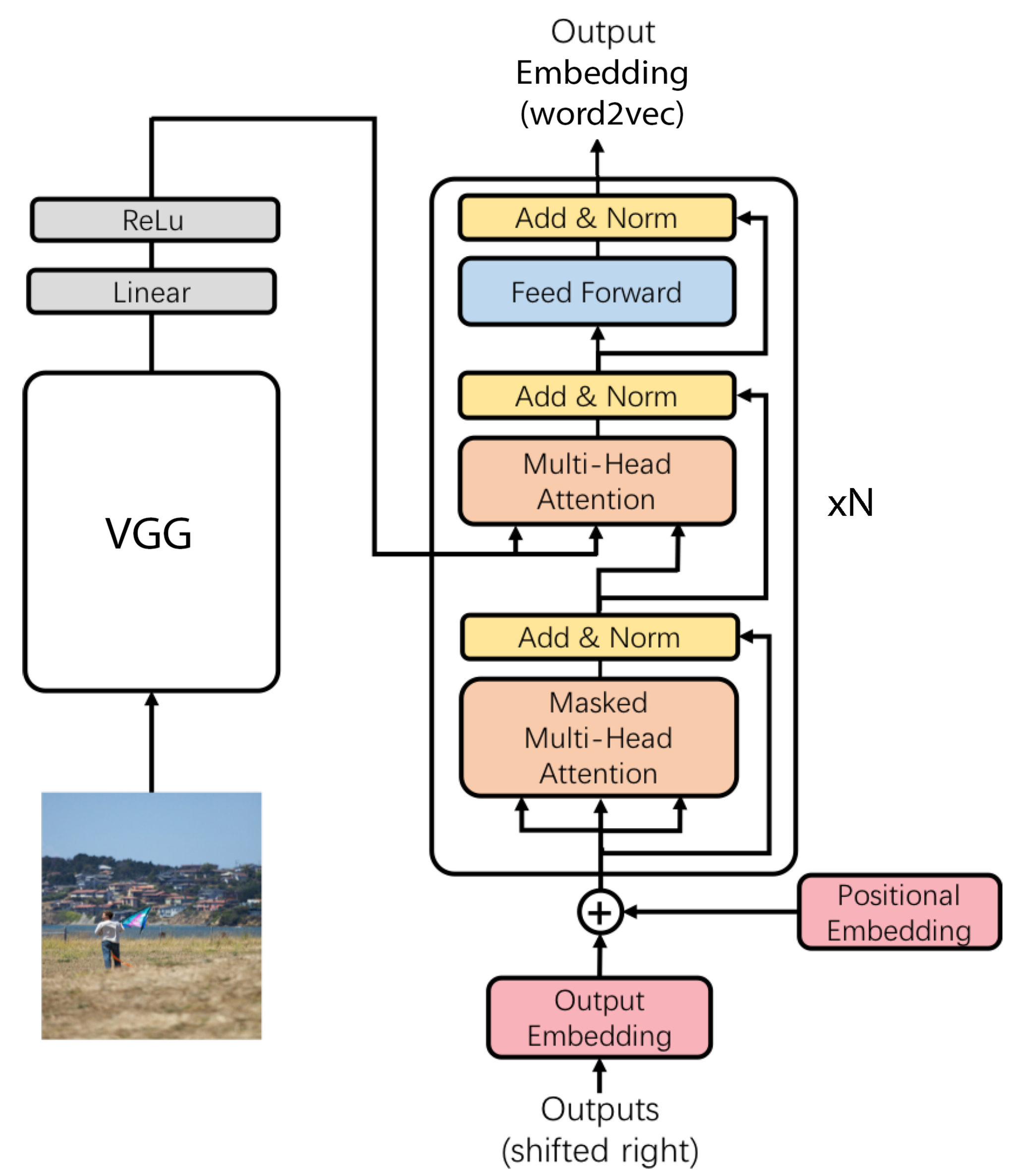

Figure 2: The proposed network inspired by the network of Zhu et al [3]. The VGG will output an array size 512×196. The fully connected layer will reshape the image as input for the transformer. The Transformer initiates with the <start> vector and keeps predicting the next word until it guesses <stop>.

❶ Embedding with Word2Vec

- Pretrained Word2Vec network
- Lemminized words to vectors size 50
- Vector space is extended to 52 for words:
  - ❶ <start>as $[\mathbf{1}_{50}, 1, 0]$
  - ❷ <stop>as $[\mathbf{1}_{50}, 0, 1]$
  - ❸ <pad>as $[\mathbf{1}_{50}, 1, 1]$
- So for the Transformer we set $d_{model} = 52$

❷ VGG encoder

- Pretrained VGG16 network (Figure 1)
- Takes a 224×224 image as input
- Outputs a 512 feature maps size 14×14
- Linear layer resizes the features to $512 \times d_{model}$

❸ Transformer decoder

- Transformer decoder from *Attention is all you need* [1] (without softmax)
- Takes the <start>embedding as input
- Based on the input the next word in the sentence is guessed
- The output will be given to the bottom of the Transformer decoder until <stop>is guessed
- Positional sine encoding for information about semantic order

Results

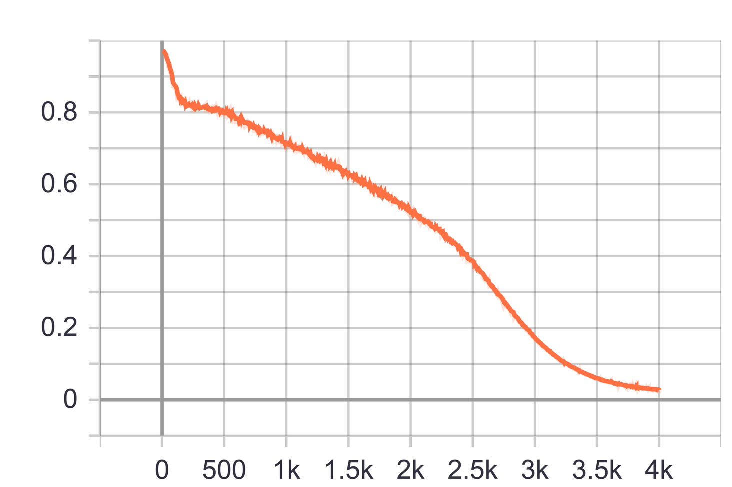

Figure 3: The cosine distance loss for the dev set.

$\boxed{k = 2024}$
Real: START man at use tire shop have white beard END
Pred: pail man boy boy girl girl girl girl girl boy boy boy boy boy boy boy boy boy boy boy baby baby my you you you you pack pack walk walk walk walk walk walk walk walk walk sit sit sit sit sit sit sit sit sit sit

$\boxed{k = 2676}$
Real: START man in black leath jacke be sleep in subwa car END
Pred: START $(100\times)$

Discussion

We overfit on the test set for $k > 2500$, but the result before that is not very good. We observe that <stop>is never guessed and the algorithm overfits by guessing <start>. Although this is not desired, we do know what happens:

- Most of the sentences are short and most of the training input is padded
- For Word2Vec <pad>is non-zero and has similarity
- Guessing <start>has cosine distance $1 - 51/52 \approx 0.02$ with <pad>, which is the loss we observe

Proposed solution: different embeddings for <start>, <stop>and <pad>promoting dissimilarity

- ❶ <start>as $[\mathbf{rand}([-1,1])_{50}, 1, 0]$
- ❷ <stop>as $[\mathbf{rand}([-1,1])_{50}, 0, 1]$
- ❸ <pad>as $[\mathbf{rand}([-1,1])_{50}, -1, -1]$

Method

We tried to **overfit** on a **dev** set for **trainability** and **compare** on a **full** set for **performance** testing.

❶ Data

- Sampled Flickr8 dev (training) set of size 100
- Full Flickr8 dataset using a training, test and validation set with official split (6000/1000/1000) for performance testing

❷ Training

- Hardware - Google Cloud CPU with capacity 7.5 GB & NVIDIA Tesla V100 GPU.
- Optimizer - Adam with Noam scheme learning rate schedule
- Regularization - Dropout
- Loss - Cosine distance loss

❸ Validation

- BLEU scores
- SPICE scores

Conclusion

We implemented a trainable Transformer-VGGNet combination. The major drawback is probably the choice for embedding the special tokens <start>, <stop>and <pad>. We proposed a solution to this problem, which has not yet been tested.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[2] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio.
Show, attend and tell: Neural image caption generation with visual attention.
In *International conference on machine learning*, pages 2048–2057, 2015.

[3] Xinxin Zhu, Lixiang Li, Jing Liu, Haipeng Peng, and Xinxin Niu.
Captioning transformer with stacked attention modules.
Applied Sciences, 8(5):739, 2018.