Show, Attend and Tell

Paul Liebenow, Philip Fürste, Berend Brandt

February 2018

Contents

1 Introduction

Making Computers able to understanding images scenes is one of the key problems in computer vision. There have been different approaches to this problem. The classic approach was to use object detectors which scan the input image for specific shapes or colors. The modern approach is to use deep neural networks and the encoder - decoder framework. We want to define those terms shortly. Deep neural networks (DNN) are artificial neural network with multiple layers where every layer extracts different features and the output is a composition of those features. There exist different architectures for DNNs. For the encoder-decoder-framework we used recurrent neural network (RNN) and convolutional neural networks (CNN). The encoder-decoder-framework is a composition of DNNs. The encoder extracts low level features and compresses those in a feature vector and the decoder learns with this information to output. The origins of this framework can be found in neural machine translation [2]. In neural machine translation the task is to translate sentences of one language into another. When people tried to solve this task they faced the problem that the encoder-decoder-framework wasn't capable of translating very long sentences because of the information loss through the compression into a feature vector. In 2014 an approach to solve this problem proposed by Bahdanau,Cho and Bengio was to use attention [1]. Attention is the encoding of the input into a fixed-length vector and the learning of joint alignation and translation. Since the task of neural machine translation and generating image caption are similiar in their nature Bengio applied attention to the problem of generating image caption using the encoder-decoder-framework [4]. The framework consists of a CNN as an encoder and a RNN as decoder. The low level features extracted by the CNN are encoded into an vector which is then fed into the RNN to generate the image caption.

2 Methods

In this part we will first explain how we preprocessed the kaggle handwritten-math-symbol data set. Secondly we will present the algorithm that we implemented using Tensorflow.

2.1 Data preprocessing

The data preprocessing was done in two steps. First we implemented a class glue.py which glued randomly the different math symbols on to each other. We choosed 20 different symbols. Secondly we implemented a batch generator which merged glued images together and generated one-hote-vectors which we used as labels.

2.2 The Algorithm

2.2.1 Theoretical Basis

We will now try to show the mathematical formalism of the model that we implemented. We will try to be especially cautious to derive a clear definition of attention. In the paper 2 kinds of attention were introduced: hard and soft attention. We focused on soft attention only.
We will use the notation from paper [2]. Let small letters be scalars (only if explicitly told small letters will be sets),bold small letter vectors and big letters matrices. We start by defining some terms:
Let X be the input image. Let s_t be the position where the model focus its attention.

Encoder: Convolutional Network

Let the feature vector be

$$a = \{\mathbf{a_1}, ..., \mathbf{a_L}\}, \mathbf{a_i} \in \mathbf{R^D}$$

L D-dimensional vectors representing the corresponding part of X. We generate this vector through the convolutional network.
Let

$$y = \{\mathbf{y_1}, ..., \mathbf{y_C}\}, \mathbf{y_i} \in \mathbf{R^K}$$

be the caption where K is the size of vocabulary and C is the length of the caption.

Decoder: Long-Short-Term-Memory (LSTM)

The paper used a Long-Short-Term-Memory cell to generate a caption. At every timestep t the framework produced one word depending on context

vector $\hat{\mathbf{z}}_\mathbf{t}$, previous hidden state $\mathbf{h_{t-1}}$ and previously generated word $E\mathbf{y_{t-1}}$. The exact gating of the LSTM cell can be found in the paper.

Context vector $\hat{\mathbf{z}}$: heart of attention

The context vector is defined in the paper as the position at time t where the model directs its attention. We state shortly how it is computed. First we compute $\mathbf{e_{t,i}}$

$$e_{t,i} = f_{att}(\mathbf{a_i}, \mathbf{h_{t-1}})$$

where f_{att} is a Multilayer Perceptron. Now we plug the $\mathbf{e_{t,i}}$ into the softmax function to normalize and therefore reduce the influence of outliers:

$$\alpha_{t,i} = \frac{exp(e_{t,i})}{\sum_{k=1}^{L} exp(e_{t,k})}$$

since the $\alpha_{t,i}$ add up to 1 after normalization we can see them as the probability that the model has choosen the right position s_t at time t or as the i-th weight for the feature vector a_i at a time t. We arrive at $\hat{\mathbf{z}}_\mathbf{t}$ with:

$$\hat{\mathbf{z}}_\mathbf{t} = \phi(\{\mathbf{a_i}\}, \{\alpha_{\mathbf{t,i}}\})$$

Attention mechanism: Deterministic "Soft" Attention

In the case of soft attention $\phi(\{\mathbf{a_i}\}, \{\alpha_{\mathbf{t,i}}\})$ is simply given as:

$$\phi(\{\mathbf{a_i}\}, \{\alpha_{\mathbf{t,i}}\}) = \sum_{\mathbf{i=1}}^{\mathbf{L}} \alpha_{\mathbf{t,i}} \cdot \mathbf{a_i}$$

which is the expected value of the attented position at time t. The attention mechanism consists now in maximizing the expected value for all timesteps t with the help of the backpropagation algorithm.

Summary

To summarize first the encoder computes $\{\mathbf{a_1}, ..., \mathbf{a_L}\}$. Secondly $\{\mathbf{a_1}, ..., \mathbf{a_L}\}$ is passed to the decoder which is doing the following in every timestep t:

1. compute probability $\alpha_{t,i}$
2. compute contex vector $\hat{\mathbf{z}}_\mathbf{t}$
3. generate new output vector $\mathbf{y_t}$
4. compute new decoder state $\mathbf{h_{t+1}}$

The training of the LSTM cell is done after every timestep with the help of stochastic gradient descent. More precisly in the training session we are feeding one hot vectors with the feature vectors a_t into the LSTM cell and applying the backpropagation algorithm. With this procedure training of the weights of the MLP and of the RNN is done at the same time.
Question: Why does the gradient not explode?

Evalution of the Theoretical Basis

We will evaluate this framework by comparing it to the NIC from [3] which used the same framework without attention. On the Flickr8k Data set the NIC scored 63 under the BLEU-4 Metric and the model that we partly implemented 67. The BLEU metric indicates how similiar a machine translational is to a human with value 100 being identical. Why does the framework from Xu et. al perform better? The weakness of the encoder-decoder framework without attention comes from the fact that all information from the encoder is compressed into the feature vector which is then fed into the decoder. The framework from Bahdanau et. al ,which was applied to images by our paper, learns to jointly align the feature vector from the encoder. Therefore it becomes possible for the network to catch semantic connections on a larger scale than it is possible for the framework from Oriol et. al. This is done because of the $\alpha_{t,i}$ jointly aligning the $\mathbf{a_i}$ by weighting them. More precisly the $\alpha_{t,i}$ are weighting the feature vectors $\mathbf{a_i}$ according to the currently attended $\mathbf{s_t}$ or in mathematical terms maximizing the expected value $\sum_{i=1}^{L} \alpha_{t,i} \cdot \mathbf{a_i}$ given conditional probability of $\mathbf{s_t}$ given $\{\mathbf{a_1}, ..., \mathbf{a_L}\}$.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Kyunghyun Cho, Bart Van Merrinboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[3] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition*, 2015.

[4] 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.