http://dx.doi.org/10.35596/1729-7648-2021-19-8-40-44

Original paper

UDC 004.855.5:004.932.75

DEVELOPING A SEQ2SEQ NEURAL NETWORK USING VISUAL ATTENTION TO TRANSFORM MATHEMATICAL EXPRESSIONS FROM IMAGES TO LATEX

PAVEL A.VYAZNIKOV, IGOR D. KOTILEVETS

MIREA – Russian Technological University (Moscow, Russian Federation)

Submitted 19 November 2021

Abstract. The paper presents the methods of development and the results of research on the effectiveness of the seq2seq neural network architecture using Visual Attention mechanism to solve the im2latex problem. The essence of the task is to create a neural network capable of converting an image with mathematical expressions into a similar expression in the LaTeX markup language. This problem belongs to the Image Captioning type: the neural network scans the image and, based on the extracted features, generates a description in natural language. The proposed solution uses the seq2seq architecture, which contains the Encoder and Decoder mechanisms, as well as Bahdanau Attention. A series of experiments was conducted on training and measuring the effectiveness of several neural network models.

Keywords: im2latex, seq2seq, NLP, neural network.

Conflict of interests. The authors declares no conflict of interests.

For citation. Vyaznikov P.A., Kotilevets I.D. Developing a seq2seq neural network using visual attention to transform mathematical expressions from images to LaTeX. Doklady BGUIR. 2021; 19(8): 40-44.

Introduction

In the modern world, Optical Character Recognition technology finds an incredible number of applications (text recognition and rapid document scanning). The progress in this area is due to the emergence of advanced deep learning algorithms and neural network models which learn from a huge number of examples, can make very accurate predictions.

The seq2seq architecture (Sequence to Sequence) is very effective in OCR neural networks. It consists of two main parts – Encoder and Decoder, these are two connected neural networks training simultaneously, but performing different tasks. The encoder reads the input sequence (input data) and summarizes the information in a form called “the internal state vector”. In turn, the decoder uses these vectors and trains to generate the correct output sequence based on them.

The task of im2latex, known thanks to the OpenAI company, is to create an OCR neural network for converting math images into LaTeX language. The presented solution uses the seq2seq architecture and involves extracting features from images using convolutional layers (Encoder) and passing them to the decoder, which contains an RNN layer and trains to produce a caption in LaTeX for any mathematical image. The principle of operation of the neural network is shown in Fig. 1.

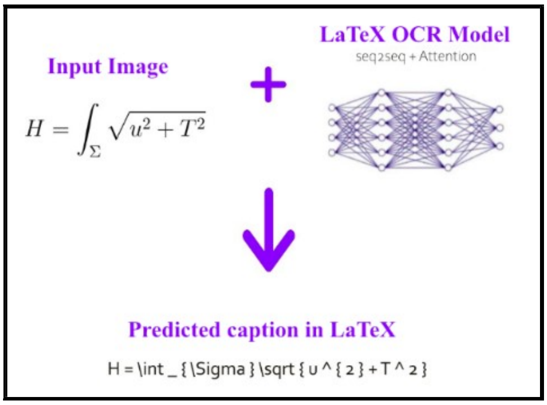
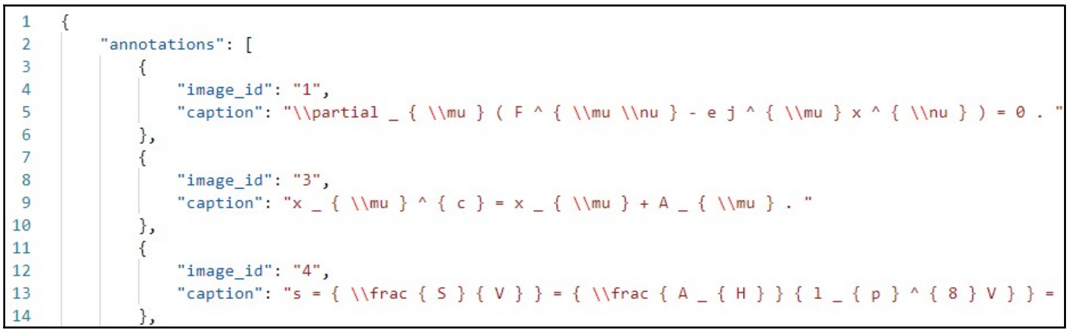

Input Image
$H = \int_{\Sigma} \sqrt{u^2 + T^2}$
LaTeX OCR Model
seq2seq + Attention
Predicted caption in LaTeX
H = \int _ { \Sigma } \sqrt { u ^ { 2 } + T ^ 2 }

Fig. 1. Principle of im2latex

Dataset

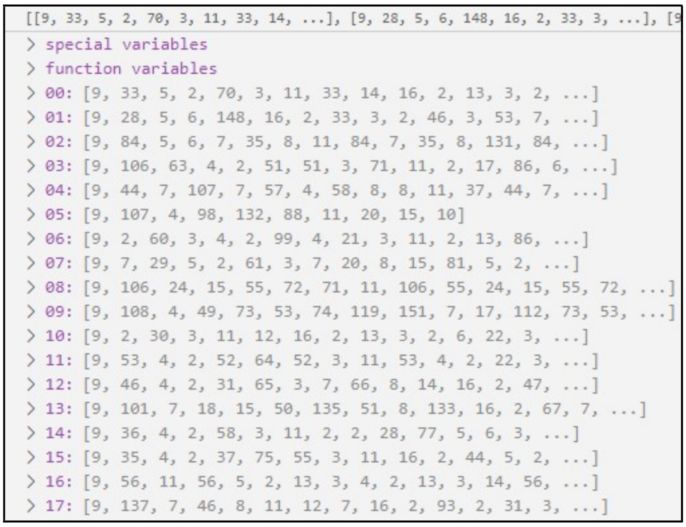
The training data for im2latex contains 100000 images with mathematical expressions, paired with their true LaTeX label. A mandatory part of preprocessing is the separation of all individual LaTeX constructions by spaces. Fig. 2 shows several captions after this step.



```
{
    "annotations": [
        {
            "image_id": "1",
            "caption": "\\partial _ { \\mu } ( F ^ { \\mu \\nu } - e j ^ { \\mu } x ^ { \\nu } ) = 0 . "
        },
        {
            "image_id": "3",
            "caption": "x _ { \\mu } ^ { c } = x _ { \\mu } + A _ { \\mu } . "
        },
        {
            "image_id": "4",
            "caption": "s = { \\frac { S } { V } } = { \\frac { A _ { H } } { 1 _ { p } ^ { 8 } V } } =
        },
```

Fig. 2. LaTeX captions splitted by spaces

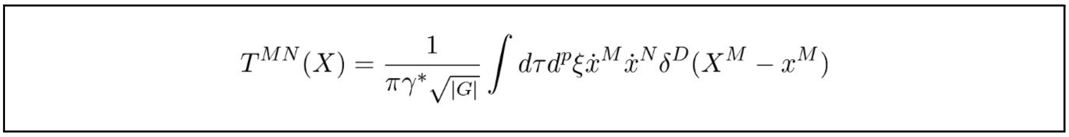
Before training, all captions are being tokenized (represented as numbers) and then a dictionary is assembled from all individual tokens (i.e., individual LaTeX words), which is used during training processes. Fig. 3 shows an array of tokenized labels.



```
[[9, 33, 5, 2, 70, 3, 11, 33, 14, ...], [9, 28, 5, 6, 148, 16, 2, 33, 3, ...], [9
> special variables
> function variables
> 00: [9, 33, 5, 2, 70, 3, 11, 33, 14, 16, 2, 13, 3, 2, ...]
> 01: [9, 28, 5, 6, 148, 16, 2, 33, 3, 2, 46, 3, 53, 7, ...]
> 02: [9, 84, 5, 6, 7, 35, 8, 11, 84, 7, 35, 8, 131, 84, ...]
> 03: [9, 106, 63, 4, 2, 51, 51, 3, 71, 11, 2, 17, 86, 6, ...]
> 04: [9, 44, 7, 107, 7, 57, 4, 58, 8, 8, 11, 37, 44, 7, ...]
> 05: [9, 107, 4, 98, 132, 88, 11, 20, 15, 10]
> 06: [9, 2, 60, 3, 4, 2, 99, 4, 21, 3, 11, 2, 13, 86, ...]
> 07: [9, 7, 29, 5, 2, 61, 3, 7, 20, 8, 15, 81, 5, 2, ...]
> 08: [9, 106, 24, 15, 55, 72, 71, 11, 106, 55, 24, 15, 55, 72, ...]
> 09: [9, 108, 4, 49, 73, 53, 74, 119, 151, 7, 17, 112, 73, 53, ...]
> 10: [9, 2, 30, 3, 11, 12, 16, 2, 13, 3, 2, 6, 22, 3, ...]
> 11: [9, 53, 4, 2, 52, 64, 52, 3, 11, 53, 4, 2, 22, 3, ...]
> 12: [9, 46, 4, 2, 31, 65, 3, 7, 66, 8, 14, 16, 2, 47, ...]
> 13: [9, 101, 7, 18, 15, 50, 135, 51, 8, 133, 16, 2, 67, 7, ...]
> 14: [9, 36, 4, 2, 58, 3, 11, 2, 2, 28, 77, 5, 6, 3, ...]
> 15: [9, 35, 4, 2, 37, 75, 55, 3, 11, 16, 2, 44, 5, 2, ...]
> 16: [9, 56, 11, 56, 5, 2, 13, 3, 4, 2, 13, 3, 14, 56, ...]
> 17: [9, 137, 7, 46, 8, 11, 12, 7, 16, 2, 93, 2, 31, 3, ...]
```

Fig. 3. Tokenized labels

Training images may have different resolutions, may contain noise, etc. These parameters must be carefully selected, since during training processes this can affect the effectiveness both for better or worse. The highest efficiency was achieved with 300dpi images and a formula, padded to the middle (Fig. 4).



$$T^{MN}(X) = \frac{1}{\pi\gamma^*\sqrt{|G|}} \int d\tau d^p\xi \dot{x}^M \dot{x}^N \delta^D(X^M - x^M)$$

Fig. 4. Example of training image

Encoder

Encoder is a mechanism, that contains a set of convolutional and pooling layers (shown in Fig. 5) and is designed to extract features from images, which can then be used by Decoder and Attention to generate an output sequence. The number of layers and their settings are very important and may vary depending on the task. After extracting the features, Encoder summarizes it in a form called the "Internal State Vector", which is passed to Decoder.

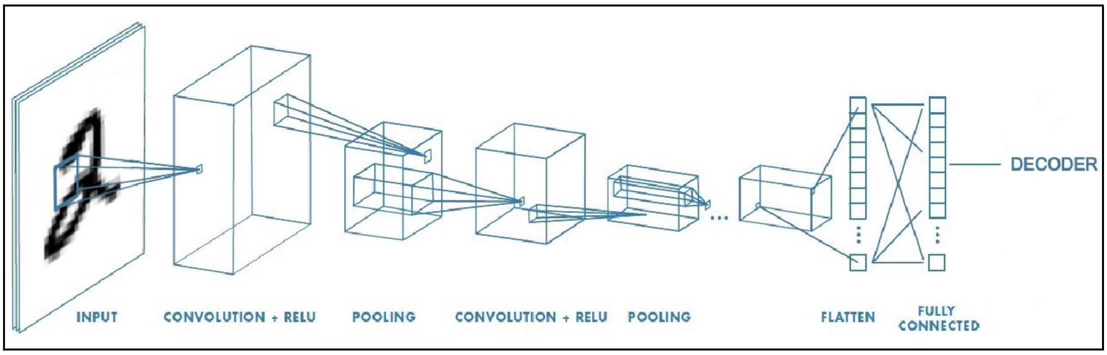


Fig. 5. Encoder architecture

Decoder

After extracting the features from input images, the decoder begins to train, which after the process is done will be able to generate captions for any image. The mechanism is based on a recurrent neural network (RNN), which is able to identify and remember important information and ignore the irrelevant one. There are several recurrent layers, the best of which are proved to be LSTM [1]. Due to the presence of a gate mechanism, LSTM, based on image features and the truth LaTeX labels, computing of which features correspond to certain LaTeX symbols, and subsequently generate an accurate label for the new image.

An important element of the decoder is the Bahdanau Attention mechanism [2]. Its essence is to generate "importance weights", called the "Context Vector", for the output sequences of the encoder (image features), which are then combined with the input data of the decoder, which allow the network to train much more efficiently. Fig. 6 shows an example of how attention mechanism works in NLP. The input data of the decoder is dataset images passed through the encoder and their real captions.

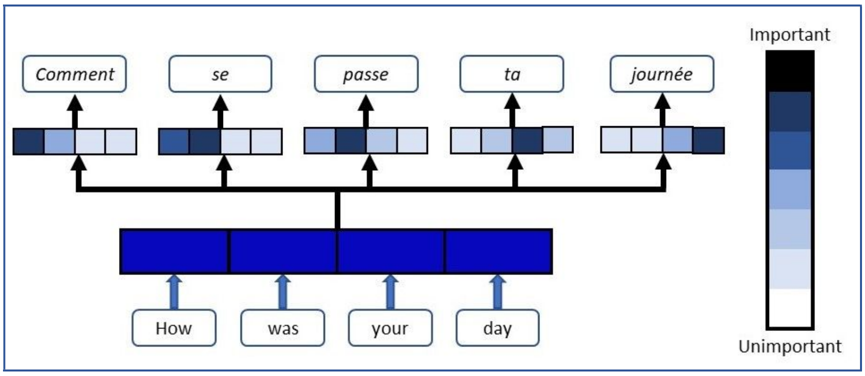


Fig. 6. Attention mechanism

Training loop

The training loop is being done for every LaTeX token of each caption and dataset image. The whole process is shown on Fig. 7 and consists of following parts:

1. Passing input images through Encoder and getting their features.
2. The decoder LSTM hidden state combined with the encoded images is being passed to the attention mechanism, which computes context vector for this training step.
3. Context vector is being concatenated with LaTeX label during this step and passed to LSTM, which generates probability distribution for each unique LaTeX token that is used during training, the higher the value, the greater the probability that the token should be after the previous one.
4. The probability distribution is being passed to the loss function (Categorical Crossentropy), which determines how far the prediction of the network is from the real caption.
5. Optimizer corrects neural network weights according to the obtained loss value.

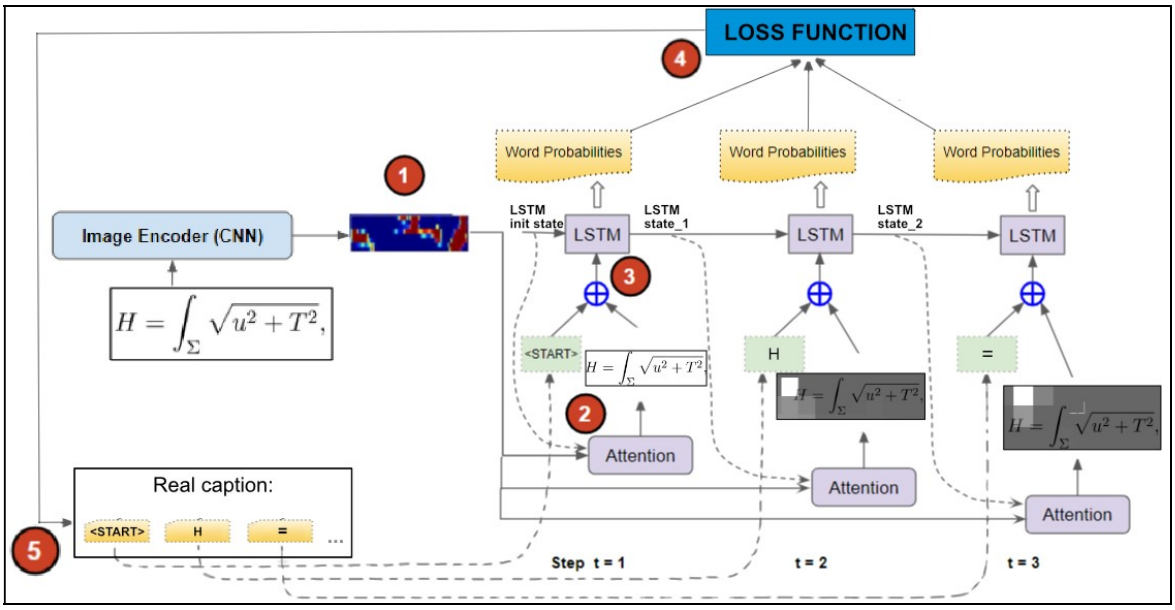


Fig. 7. Training loop

Hyperparameters

The best training results were achieved with the following parameters:

- epochs = 12
- batch_size = 24
- units = 200 (number of cells in decoder's recurrent layer)
- optimizer = Adam.

With such parameters, the network trained for about 14 hours using Nvidia RTX 3080 and 32 gigabytes of RAM. The lowest average loss obtained is 0.02, which is a good result for Image Captioning neural networks.

Evaluation

Measured BLEU [3] score is about 70 % and Levenshtein distance metric is 31. In comparison with similar works [Im2Latex. September 2019. https://github.com/luopeixiang/im2latex (on-line resource)], the obtained metric measurements are quite high. The competing solution has 40 % for BLEU against 70% of the reviewed and 44 for Minimal Edit Distance against 31. Based on the above data, it can be argued that the developed neural network has high efficiency.

Conclusion

The presented neural network with the seq2seq architecture and Attention mechanism successfully solves the im2latex problem, which is confirmed by the results of measuring metrics. Generated captions for images with equations are quite accurate and, in most cases, coincide with the real ones.

Such solution can be used in mathematical programs to automatically translate images into LaTeX and further solve and analyze the resulting equations or expressions.

References

1. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997;9(8):1735-1780.
2. Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. *3rd International Conference on Learning Representations, ICLR 2015.*
3. Papineni K., Roukos S., Ward T., Zhu W. BLEU: a method for automatic evaluation of machine translation. *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002: 311-318.

Authors' contribution

Vyaznikov P.A. completed dataset collection, neural network development in Python, training, and evaluation.

Kotilevets I.D. fulfilled documentation, carried out forming technical specifications, as well as technical and functional requirements.

Information about the authors

Vyaznikov P.A., Bachelor student at the Federal State Budget Education Institution of Higher Education “MIREA – Russian Technological University”.

Kotilevets I.D., Senior Lecturer at the Federal State Budget Education Institution of Higher Education “MIREA – Russian Technological University”.

Address for correspondence

119454, Russian Federation,
Moscow, Vernadsky Ave, 78,
Federal State Budget Education Institution of Higher Education
“MIREA – Russian Technological University”;
tel. +7-499-215-65-65;
e-mail: vyaznikov.p.a@edu.mirea.ru
Vyaznikov Pavel Andreevich