SCIENCE CHINA
Information Sciences

• LETTER •

March 2021, Vol. 64 139101:1–139101:3
https://doi.org/10.1007/s11432-018-9824-9

Robust encoder–decoder learning framework for offline handwritten mathematical expression recognition based on a multi-scale deep neural network

Guangcun SHAN[1*], Hongyu WANG[1], Wei LIANG[2,3] & Kai CHEN[4]

[1]*School of Instrumentation Science and Optoelectronics Engineering, Beihang University, Beijing* 100191*, China;*
[2]*Institute of Electronics, Chinese Academy of Sciences, Beijing* 100190*, China;*
[3]*School of Electronic, Electrical and Communication Engineering,University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[4]*Boheng Technology (Hangzhou) Co. Ltd, Hangzhou* 310016*, China*

Received 3 July 2018/Revised 11 October 2018/Accepted 11 February 2019/Published online 27 May 2020

Citation Shan G C, Wang H Y, Liang W, et al. Robust encoder–decoder learning framework for offline handwritten mathematical expression recognition based on a multi-scale deep neural network. Sci China Inf Sci, 2021, 64(3): 139101, https://doi.org/10.1007/s11432-018-9824-9

Dear editor,

Mathematical expressions have been widely employed in scientific research, finance, and statistics, and play a significant role in educational activities. For example, if a computer can recognize teachers' handwritten expressions as standard printed mathematical expressions, this will undoubtedly be more conducive and helpful for improving the effectiveness of lectures. Thus, the question of how to make computers automatically recognize mathematical expressions is highly significant.

Mathematical expression recognition was first proposed by Anderson [1] in 1968. Subsequently, many researchers have attempted to investigate this problem in various aspects [2–5]. Lavirotte and Pottier [2] proposed a model based on graph grammars. Chan and Yeung [3] proposed a method employing definite clause grammars. In 2006, Yamamoto [4] presented a system using probabilistic context-free grammars. And several years later MacLean and Labahn [5] developed an improved approach based on relational grammars and fuzzy sets.

Compared to the traditional text recognition, there remain some difficulties in recognizing mathematical expressions, making this a challenging research problem. The main reason for this is that mathematical expressions can contain a large number of mathematical symbols and have a complex two-dimensional structure. First, the character coverage of mathematical expressions includes numbers, operators, English letters, Roman letters, commas, dots, and other types. Second, some symbols are not shown in the expressions, and these symbols need to be recognized by spatial positions of mathematical expressions. For example, 'X^6' denotes the sixth power of 'X', but the operator '∧' is not directly written in this mathematical expression. Finally, some of the characters in mathematical expressions are very similar, but have completely different meanings. For example, the English letter 'O' and the number '0.' Therefore, owing to these challenging issues, handwritten mathematical expression recognition has considerable room for development and improvement in terms of the accuracy and range of recognition.

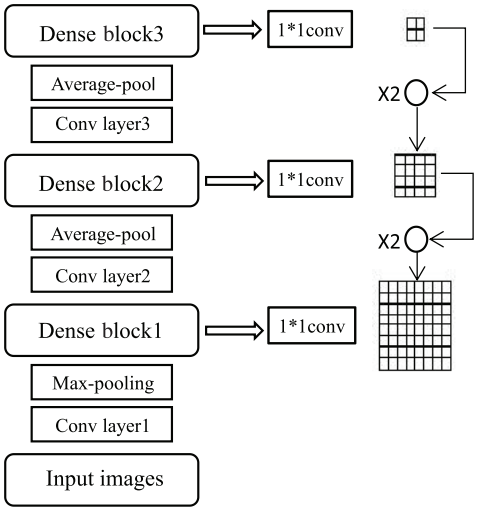
Framework. To recognize handwritten mathematical expressions, an encoder–decoder framework is utilized in this study. This framework consists of three parts: an encoder, a decoder, and a context vector generated by the encoder. In this study, we employ a convolutional neural network (CNN) as the encoder, because inputs are pictures, and a recurrent neural network (RNN) is utilized as the decoder, because outputs are LaTeX sequences.

Encoder model. In this study, dense convolutional network (DenseNet), a recently proposed CNN architecture by Huang [6], is employed as the encoder, and further enhanced in a feed-forward fashion. We designed three dense blocks, and added a multi-scale structure to better extract features from input pictures. The multi-scale CNN used in this study is illustrated in Figure 1(a).

The reason for adopting this structure is that the high-level neural network in the CNN can obtain more advanced abstract features, which are conducive to improving the generalization performance of the network, while the low-level neural network in the CNN contains more edge information, which is conducive to target location. Therefore, the combination of these two-level features can lead to a better performance for the network.

Decoder model. We employ an RNN as the decoder. In

* Corresponding author (email: gcshan@buaa.edu.cn)



(a)

System	Accuracy (%)
Tokyo University	19.97
University Nantes	18.33
Rochester Institute of Technology (RIT)	14.31
Ours	**28.21**

(b)

Figure 1 (a) Schematic structure of multi-scale convolutional neural network; (b) test results.

further detail, the gated recurrent unit (GRU) model [7] is utilized for the design of the RNN in this study. Although the GRU model is employed, the effect is still not ideal for complex sequences such as mathematical expressions. Therefore, the attention model is adopted in the RNN, which is a method that can considerably improve the model's performance. The main design ideas are described as follows.

In our model, because the outputs of the CNN consist of three three-dimensional matrixes, and each of these can be expressed as $C \times H \times W$, the inputs of the RNN can be written as follows:

$$a = \{a_1, a_2, a_3, \ldots, a_n\}, \quad a_i \in \mathbb{R}^L, \tag{1}$$

where $L = H \times W$. Thus, each a_i is a C-dimensional vector. Then, we define the function f_{att} given in (2) and (3) as our attention function. Therefore, the RNN with attention in our model can be calculated by the function f_{att}:

$$e_{ti} = v_a^{\mathrm{T}} \times \tanh(w_a h_{t-1} + U_a a_i), \tag{2}$$

$$\alpha_{ti} = \frac{\exp e_{ti}}{\sum_{k=1}^{L} \exp e_{tk}}. \tag{3}$$

In (2), h_{t-1} is the value of the previous hidden layer in the GRU model, a_i is the i-th vector in (1), and v_a is a random initialization matrix, which is updated continuously following the training of the network. Its main role is to adjust the dimensions of e_{ti}. In (3), α_{ti} is the value of the next attention model.

Coverage model. By incorporating attention, the performance of the RNN model can be improved compared to the model without attention. However, the problem of a lack of coverage remains in the attention model. Coverage refers to the parts of the input images that have been translated. If we do not add a coverage model, then the computer may repeatedly translate one part. The coverage model requires all past values of the attention model to be summed. Therefore, after including the coverage model the function f_{att} will be computed as follows, where we now denote it as $f_{\text{att\_conv}}$:

$$\beta_t = \sum_{l}^{t-1} \alpha_l, \tag{4}$$

$$F = Q\dot{\beta}_t, \tag{5}$$

$$e_{ti} = v_a^{\mathrm{T}} \times \tanh(w_a h_{t-1} + U_a a_i + U_f f_i), \tag{6}$$

$$\alpha_{ti} = \frac{\exp e_{ti}}{\sum_{k=1}^{L} \exp e_{tk}}. \tag{7}$$

In (4), α is the value calculated by the attention model in the previous step, and its initial value is 0. In (5), Q is a random initialization matrix, which is updated continuously following the training of the network. The main role of Q is also to adjust the dimensions. Furthermore, F is the value of the coverage model, and f_i is the i-th vector in F. Then, the attention value can be calculated using (6) and (7).

Output. Because the output from the encoder is multi-scale, the context vector of each scale will be computed using the following equation:

$$c_{tk} = \sum_{i}^{L} \alpha_{ti} a_i \qquad k = 1, 2, 3, \tag{8}$$

$$c_t = \text{Concat}(c_{t1}, c_{t2}, c_{t3}). \tag{9}$$

Each c_{tk} will be computed from (8). Then, we can combine these as c_t by simply applying the cat function in (9). Subsequently, the next hidden layer will be calculated as follows:

$$h_t = \text{GRU}(x_{t-1}, h_{t-1}, c_t), \tag{10}$$

where h_{t-1} is the value of the previous hidden layer and x_{t-1} is the value of the previous input, whose initial value is $\langle \text{sos} \rangle$. We can obtain the next hidden layer h_t according to (10). Finally, the output from the model can be computed as follows:

$$y_t = g(W_0(Ey_{t-1} + W_h h_t + W_c c_t)). \tag{11}$$

where g is the softmax function, and W_0, W_h, W_c, and E are random initialization matrixes, which can be continuously updated following the neural network training.

Experiment. In the experiment, all tests were performed on the CROHME 2014 dataset. Furthermore, we choose Adam as the optimizer and CrossEntropy as the loss function. The initial learning rate was 0.00010. Moreover, the experiments in this study were all performed on an NVIDIA Tesla K80. The CUDA version utilized in the tests was 8.0,

the cuDNN version was 7.0, the Python version was 3.6, and the Pytorch version was 0.3. The results are illustrated in Figure 1(b), and our model achieved the best results compared with other approaches under the same conditions. Detailed experimental data can be found in Appendix.

Conclusion. Based on the state-of-the-art approach of combining a multi-scale CNN with an attention-based RNN, a new neural network model is proposed to identify two-dimensional handwritten mathematical expressions as one-dimensional LaTeX sequences. Using the public benchmark of the CROHME 2014 dataset, our experimental test results demonstrate that the WER loss and ExpRate on the testing dataset are 25.715% and 28.216%, respectively. So, our approach achieved a more advanced level of accuracy than previous methods under the same test conditions.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2016YFE0204200) and National 1000-Talent Youth Program. The authors want to thank Dr. Jianshu ZHANG for insightful comments and suggestions.

Supporting information Appendixes A–F. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Anderson R H. Syntax-directed recognition of hand-printed two-dimensional mathematics. In: Proceedings of the Association for Computing Machinery Inc. Symposium. New York: ACM, 1968. 436–459
2 Lavirotte S, Pottier L. Mathematical formula recognition using graph grammar. In: Proceedings of SPIE, 1998. 3305: 44–52
3 Chan K F, Yeung D Y. Error detection, error correction and performance evaluation in on-line mathematical expression recognition. Pattern Recogn, 2001, 34: 1671–1684
4 Yamamoto R. On-line recognition of handwritten mathematical expression based on stroke-based stochastic context-free grammar. In: Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, 2006
5 MacLean S, Labahn G. A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. Int J Document Anal Recogn, 2013, 16: 139–163
6 Huang G, Liu Z, Weinberger K Q. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 2261–2269
7 Cho K, Merrienboer B, Bahdanau D, et al. On the properties of neural machine translation: encoderdecoder approaches. 2014. ArXiv: 1409.1259