Evolving Transformer Architecture for Neural Machine Translation

Ben Feng
College of Computer Science,
Sichuan University
Chengdu, China
fengben97@foxmail.com

Dayiheng Liu
College of Computer Science,
Sichuan University
Chengdu, China
losinuris@gmail.com

Yanan Sun
College of Computer Science,
Sichuan University
Chengdu, China
ysun@scu.edu.cn

ABSTRACT

The transformer models have achieved great success on neural machine translation tasks in recent years. However, the hyper-parameters of the transformer are often manually designed by expertise, where the layer is often regularly stacked together without exploring potentially promising ordering patterns. In this paper, we propose a transformer architecture design algorithm based on genetic algorithm, which can automatically find the proper layer ordering pattern and hyper-parameters for the tasks at hand. The experimental results show that the models designed by the proposed algorithm outperform the vanilla transformer on the widely used machine translation benchmark, which reveals that the performance of transformer architecture can be improved by adjusting layer ordering pattern and hyper-parameters by the proposed algorithm.

CCS CONCEPTS

• **Computing methodologies** → **Genetic algorithms**; **Machine translation**.

KEYWORDS

Genetic algorithm, Machine translation, Transformer

ACM Reference Format:
Ben Feng, Dayiheng Liu, and Yanan Sun. 2021. Evolving Transformer Architecture for Neural Machine Translation. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459441

1 INTRODUCTION

Transformer [10] constructs the encoder-decoder structure [11] by stacking interleaved Multi-Head Attention(MHA) layers and Feed-Forward Networks (FFN) layers with fixed hyper-parameters. Raganato *et al.* [7] observe that when using transformer models to perform Neural Machine Translation (NMT) tasks, the bottom blocks in the encoder tend to learn more about the syntax while the top blocks tend to learn more about the semantics. Therefore, in principle, different layer ordering patterns should be investigated and the hyper-parameters should be automatically designed for these blocks at different positions. Though, researchers have proposed a series of variant models to improve the performance of transformer on NMT [3, 6]. There are few works to explore the layer ordering patterns and the automatically design of the hyper-parameters. In order to address the above issues, we propose a Genetic Algorithm (GA)-based algorithm [1, 8, 9], denoted as Trans-GA, to automatically design a promising transformer architecture with proper layer ordering pattern and the optimal hyper-parameters in each block.

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459441

2 THE PROPOSED ALGORITHM

The framework of the Trans-GA algorithm is shown in Algorithm 1. Firstly, the population is initialized with the proposed gene encoding strategy (line 1). Secondly, the fitness of population is evaluated (line 2). Thirdly, the evolutionary search runs until the stopping criterion is satisfied (lines 4-9). Finally, the best individual in the last population is returned for the final training and evaluation (line 10).

In order to explore potentially promising ordering patterns and hyper-parameters of transformer model, new gene encoding strategy and genetic operators are designed for the Trans-GA algorithm.

Algorithm 1: Framework of Trans-GA

1 $P_0 \leftarrow$ Initialize individuals in the first population with ***the proposed gene encoding strategy***;
2 Evaluate the fitness of P_0;
3 $i \leftarrow 1$;
4 **while** *stopping criterion is not satisfied* **do**
5 $O_i \leftarrow$ Choose parent individuals from P_i to generate the offspring with ***the proposed genetic operators***;
6 Evaluate the fitness of O_i;
7 $P_{i+1} \leftarrow$ Select next population with the environment selection algorithm from $P_i \cup O_i$;
8 $i \leftarrow i + 1$;
9 **end**
10 **Return** the best individual in P_i

In the new gene encoding strategy, multiple blocks with diverse layer ordering patterns and customizable hyper-parameters are designed to represent the individuals in Trans-GA. Specifically, four possible blocks are designed for the encoder, and each block contains two layers. Fig. 1a shows the four blocks, where the MHA rectangle denotes the MHA layer and the FFN rectangle denotes the FFN layer. E_0, E_1, E_2 and E_3 are the identification numbers for the four blocks. The two variables in the bracket are the hyper-parameters of the layers, where h denotes the number of heads

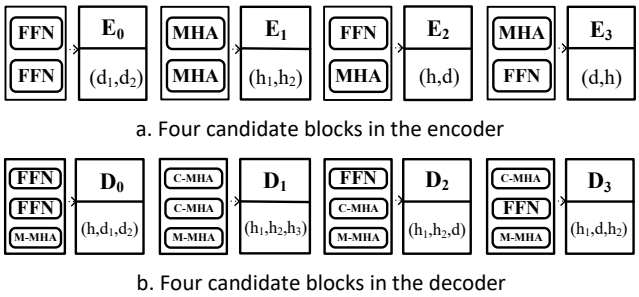

Figure 1: Candidate blocks for encoder and decoder

of MHA layer and d denotes the dimension of FFN layer. For the decoder, another four blocks D_0, D_1, D_2 and D_3 are designed, as shown in Fig. 1b, where the M-MHA rectangle denotes the masked multihead attention layer, the C-MHA rectangle denotes the cross multihead attention layer. Residual connection and layer normalization are employed in the same way as vaswani-transformer [10], which are not shown in the above two figures for the simplicity.

New genetic operators are designed for promoting the information exchange between individuals. The operators consist of two parts: crossover and mutation. During the crossover period, two parents are first selected. Then, the encoders of two parents are randomly divided into two parts and then swapped to form two new encoders. With the similar way, two new decoders are also generated. Finally, two individuals of the next generation are constructed by the new encoders and decoders. After crossover, the individuals have two mutation opportunities, one is the encoder mutation, and the other is the decoder mutation. During each of the mutation period, blocks have the chance to be added, removed or altered in the encoder or the decoder.

3 EXPERIMENT

3.1 Experiment Settings

Three groups of experiments are designed to compare with the baseline model (i.e., vaswani-transformer). The widely used NMT benchmark IWSLT-14 German to English [2] is used in the experiments, and the BLEU [5] is employed as the metrics. The word embedding sizes of the three experiments are 128, 256 and 512, respectively. The number of generations is 5, 5 and 15, respectively. Other settings for the three experiments are the same. The block number range of the encoder and the decoder are set to $5-8$. The crossover and the mutation probability are set to 0.6 and 0.2, respectively. The number of heads in MHA layer is selected from $[2, 4]$ and the FFN dimension candidate list is specified as $[512, 1024]$. The number of individuals in each population is set to 10. The settings of vaswani-Transformer is the same as in [4].

3.2 Results

Table 1 shows the experiment results, where the third and the fourth columns show the number of encoder blocks and decoder blocks in the model. The last two columns provide the number of the parameters of the model and the BLEU score, respectively.

As can be observed from Table 1, when the word embedding size is 512, the number of parameters of Trans-GA model is slightly greater than that of the baseline model, and the BLEU score is 0.3 higher than that of the baseline model. When the word embedding size is 256, the number of parameters of the Trans-GA model is still greater than that of the baseline model, and the BLEU score is 0.2 higher than that of the baseline model. When the embedding size is 128, the number of parameters of the Trans-GA model is less than that of the baseline model, but the BLEU score is 0.2 higher than that of the baseline model.

Table 1: Comparison of Trans-GA models and baseline models under the word embedding sizes of 512, 256, and 128

Embedding Size	Model	# of E	# of D	# of Para	BLEU
512	baseline	6	6	36.7M	34.47
	Trans-GA	6	8	41.0M	34.77
256	baseline	6	6	13.7M	34.79
	Trans-GA	7	7	13.9M	34.99
128	baseline	6	6	5.7M	32.49
	Trans-GA	7	7	5.5M	32.69

4 CONCLUSION

The goal of this paper is to explore the baseline transformer model (i.e., vaswani-transformer) with proper layer ordering pattern and hyper-parameters in an automatic way. Our work proves that the performance of the transformer models can be improved by adjusting the layer ordering patterns. This would benefit to investigation on manually designing promising transformer models to address more challenging NMT tasks.

REFERENCES

[1] Daniel Ashlock. 2006. *Evolutionary computation for modeling and optimization*. Springer Science & Business Media.
[2] Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, Vol. 57.
[3] Surafel M Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. *arXiv preprint arXiv:1806.06957* (2018).
[4] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762* (2019).
[5] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.
[6] Martin Popel. 2018. Cuni transformer neural mt system for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. 482–487.
[7] Alessandro Raganato, Jörg Tiedemann, et al. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. The Association for Computational Linguistics.
[8] Yanan Sun, Bing Xue, Mengjie Zhang, and Gary G Yen. 2019. Evolving deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation* 24, 2 (2019), 394–407.
[9] Yanan Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Jiancheng Lv. 2020. Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification. *IEEE Transactions on Cybernetics* (2020).
[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017), 5998–6008.
[11] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).