

Variable-Length Chromosome for Optimizing the Structure of Recurrent Neural Network

Marwa Hasna Aliefa

School of Computing

Telkom University

Bandung, Indonesia

punicamarwa@student.telkomuniversity.ac.id

Suyanto Suyanto

School of Computing

Telkom University

Bandung, Indonesia

suyanto@telkomuniversity.ac.id

Abstract—Designing a structure of Recurrent Neural Network (RNN) can be performed either manually or automatically using a Neural Architecture Search (NAS). Automatically approach is more preferred because manually approach to designing an RNN structure can be a time consuming and error-prone process, whereas NAS uses optimization algorithms such as Evolutionary Algorithm (EA) to find the optimal structure. As one of the NAS methods, a Neuro Evolution of Augmenting Topology (NEAT) searches a neural network structure constructively by adding a new neuron through mutation, which is time-consuming for the RNN structures with large neurons. Genetic Algorithm (GA) is one of the EA that is commonly used to solve optimization problems. Fix length chromosome representation dominates GA field, this representation suitable for fix length solution. Finding the optimal structure of RNN, the number of nodes and its connections cannot be determined. In this research, a Variable-Length Chromosome Genetic Algorithm (VLC-GA) is exploited to represent RNN structure with a different number of nodes. VLC-GA able to evolve the structure of RNN constructively and destructively. Therefore, the processing time for a larger structure can be reduced. Evaluation for a language modeling task shows that it is capable of reaching a lower perplexity than NAS with Reinforcement Learning (RL).

Keywords—genetic algorithm, neural architecture search, variable-length chromosome, recurrent neural network

I. INTRODUCTION

Recurrent Neural Network (RNN) is a class of Artificial Neural Network (ANN) that can receive input and provide output in the form of sequential data for pattern recognition and prediction [1]. It is widely used for many applications, such as image captioning [2], speech processing [3], music generation, time series prediction, natural language [4], [5], [6], [7], [8], [9], video analysis, and information retrieval.

The nodes on RNN have recurrent connections, which tend to have complex structures, and hyper-parameters. RNN structures can be designed and developed manually by a domain expert. However, it is time-consuming. Another approach to designing RNN structures is by Neural Architecture Search (NAS) [10], which is a process of automating the search for neural network structure using an optimization algorithm.

Many algorithms has been proposed and widely used to tackle optimization problems, such as Evolutionary Algorithm (EA) [11], [12], [13], [14] and Swarm Intelligence (SI) [15], [16], [17], [18], [19], [20], [21], [22].

Neuro Evolution of Augmenting Topology (NEAT) is a method for finding neural network structures using an EA. It can outperform the Reinforcement Learning (RL) method to optimize the neural network structures with a fixed number of nodes [23]. It is a constructive algorithm that takes much time

if the structure has many nodes with complex connectivity[24]. Therefore, a destructive operation is necessary to design RNN with a larger structure.

Genetic Algorithm (GA) is another EA method that can solve optimization problems [25]. It uses natural and genetic selection mechanism by simulating species evolution [26]. Fixed length chromosome representation dominates the GA field, this representation suitable for a problem with a fixed-length solution. Unfortunately, not all problems have fixed length solutions [27].

Finding the optimal RNN structure, the number of nodes and its connections cannot be determined. Problems like this can be solved using Variable-Length Chromosome Genetic Algorithm (VLC-GA). In this paper, we propose VLC-GA as the search method to find the optimal structure for RNN. VLC-GA able to evolve the structure for RNN constructively and destructively.

VLC-GA is a GA concept where each node and its connection on RNN being represented by a gene within the chromosome. The number of genes in a chromosome determines the number of connections in a structure. One chromosome represents one structure. In theory, the length of the chromosome will evolve in quality and quantity [27]. Chromosomes in a population will add genes if additional connections are needed to find the optimum structure.

II. RELATED WORK

A Recurrent Neural Network Language Model (RNNLM) [28] has produced state-of-the-art in perplexity and error rates for language model [29], [30]. There are several approaches to designing the structure of RNN. The manual approach is developed by humans, but this process is time-consuming and tends to be an error-prone process. The other approach is by using NAS.

NAS is the process of automating the search of neural network structure. NAS finds optimum structure from all possible structures by following a search strategy that will maximize the performance [10]. NAS has outperformed the structure designed manually in several aspects, such as image classification and natural language understanding [31]. NAS uses many algorithms to find the optimum structure.

Barret Zoph uses an RL algorithm to find the optimum structure. Action agents in RL become the generation of neural structure with action space as search space. A sequence of actions that have a good performance on unseen data will be rewarded [31]. Efficient Neural Architecture Search (ENAS) represents the search space using a single directed acyclic graph (DAG). Controllers in ENAS are trained by using gradient policy for the sub-graph selection process. The

selected sub-graph model will be trained to minimize entropy losses [32].

Neural Architecture Optimization (NAO) encodes the neural network structure into continuous space. In continuous space, gradient-based optimization can find new structures with the potential to get good accuracy [33].

Differentiable Architecture Search (DARTS) is a method for searching network structure that is differentiated based on bi-level optimization that applies to Convolutional Neural Network (CNN) and RNN structures [34].

Miller uses GA, a class of EA, to build neural network structures and uses backpropagation to search for weights [35].

Neuro Evolution (NE) is a class of Artificial Intelligence that uses EA. NE has shown good potential for complex RL problems [36]. NE looks for structures that perform well on the given problem. NE is a promising approach to solving the RL problem for several reasons. Previous studies have shown NE to be faster and more efficient than RL methods such as Adaptive Heuristic and Q-learning in single-pole balancing and robot arm control [37], [38]. NE becomes an effective method in problems that have a continuous and high-dimensional state space.

NEAT can outperform the best fixed-length method in RL. To find the best-fixed topology from the neural network, NEAT uses the EA approach is a constructive algorithm that uses GA with minimal structure. With GA, the neural network structure can evolve into a larger structure [23]; this will take a long time to find the optimum solution if the solution area is large [24]. Therefore, a chromosome that can represent a structure with a different number of neurons is needed to design larger and more complex neural networks in the search for optimum neural network structures.

III. SYSTEM MODEL

VLC-GA method will be used in this paper to find the optimum structure of RNN. Fig. 1 provides the block diagram of the model. First, the chromosomes will be generated to create a population. The chromosomes represent neurons and their connections. Then, each chromosome will generate a structure of RNN. Each RNN is estimated by its fitness value. Genetic operations, such as selection, crossover, and mutation, are implemented to make offspring. The newly generated chromosome can replace the least fit chromosome in the population.

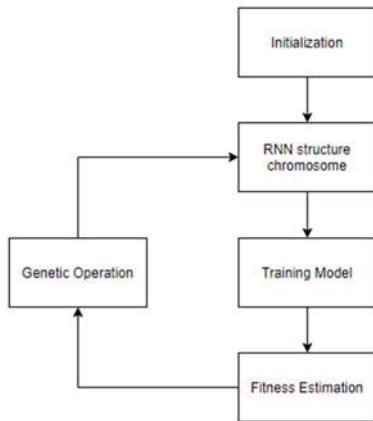


Fig. 1. Block diagram of the proposed model.

A. Dataset

The dataset used here is Penn Treebank, which is a dataset with over four million and eight hundred thousand annotated words that corrected manually by humans. The experiment conducted here uses the non-annotated words.

B. RNN structures based on VLC-GA

A chromosome represents the neurons and their connections. Using a VLC-GA, the chromosome length on each individual can be different so that the number of neurons on each individual can also be different. A chromosome represents one structure of RNN. Each chromosome will be applied to the genetic operation to evolve. Fig. 2 shows an example of one chromosome that represents an RNN structure.

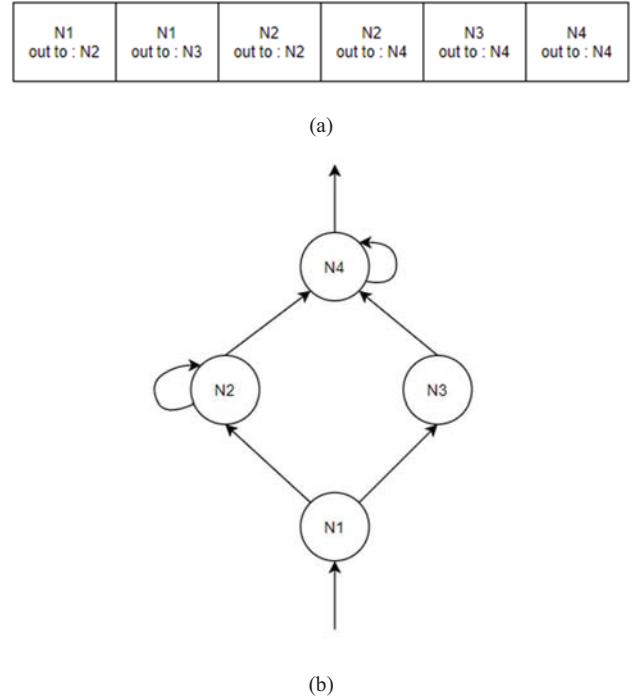


Fig. 2. (a) Example of chromosome (b) Example of RNN structure.

The steps of searching the RNN structure can be explained in detail as follow:

1) *Chromosome based RNN structure*: First, the VLC-GA initializes the population by generating population chromosomes with a random number of neurons between one to eight with random connections, the number of neurons from one chromosome can be added from crossover or mutation. At the initialization, each chromosome has a different structure from the other chromosome.

2) *Training Model*: In the training process, RNN will train to predict the next word. The data, which is a sequence of words, is fed into RNN to compute output predict the probability of every word, given words so far.

3) *Fitness estimation*: The fitness estimation process is a process to evaluate the RNNs model. Perplexity is the common evaluation metric for the language model. It measures how well the proposed probability model represents the target data. Perplexity on RNNLM is the exponential of the cross-entropy loss [39].

$$perplexity = \exp\left(\frac{1}{T} \sum_{t=1}^T -\log \hat{y}_{xt+1}^{(t)}\right) = \exp(J(\theta)) \quad (1)$$

Where T is the number of words, \hat{y} is the predicted probability distribution, and J is the cross-entropy loss.

Since a lower perplexity is the better, the fitness value is formulated as:

$$fitness = \frac{1}{perplexity} \quad (2)$$

4) *Genetic operation*: The genetic operations applied are roulette selection, single-point crossover, and mutation. Roulette selection is a selection process of population-based on fitness value to form the parent population for reproduction. Roulette selection used a roulette wheel to select chromosome parents; the bigger the fitness, the bigger the place in the roulette wheel.

Mutation and crossover can change the genes of chromosomes in order to increase the diversity of the population. Crossover is the process of recombining two chromosomes from the parent population to create new chromosomes that will have some genes from each parent. The Single-Point crossover is the crossover at one random point from 1 to n , n being the length of the shortest chromosomes [40]. For example, Fig. 3 shows a crossover between two parents.

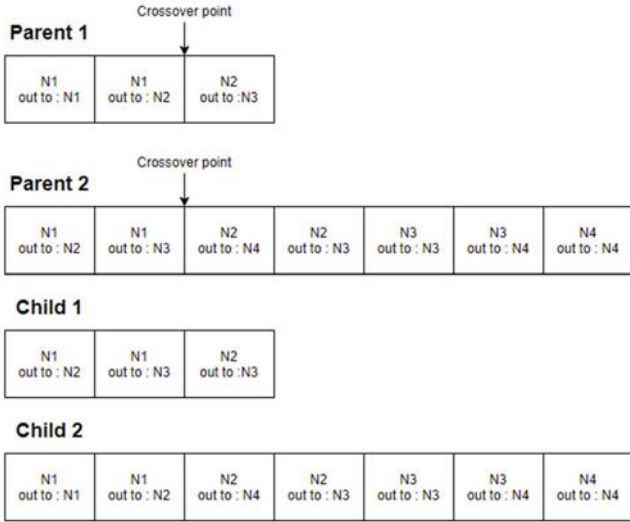


Fig. 3. Example of crossover between two different length chromosomes.

Mutation can change the gene from the child chromosome. It can change the number of neurons or connectivity. The modification gene on mutation can be shown in Fig. 4.

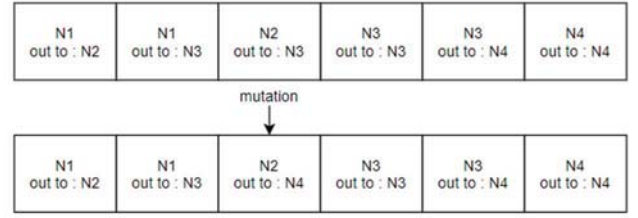


Fig. 4. Example connectivity mutation.

IV. RESULTS AND DISCUSSION

We conduct an experiment to test the effectiveness of VLC-GA to search optimum RNN structure on the language model task. For the searching structure of RNNLM with VLC-GA, we used the following parameters: learning rate was set at 0.001 with 80 epochs, the generations set at 100 with 50 chromosomes for each generation and mutation rate at 0.49, for the RNNMLs cell we used Long Short Term Memory (LSTM) cell. Data for searching the optimum structure was 5% of Penn Treebank dataset, a well-known benchmark for language model. We used 5% data of the corpus to reduce computational time.

Fig. 5 plots the best perplexity for RNNLM structure that is found by VLC-GA for 100 generations. The lowest perplexity starts from 24.745335 and decreases to 19.13463. It takes 68 generations to find the optimum solution.

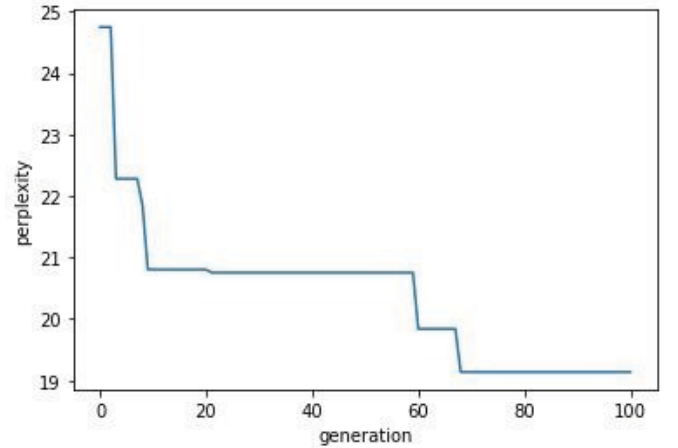


Fig. 5. Simulation result.

Once the optimum structure is found by VLC-GA, RNNLM is then trained using LSTM cell for 250 epoch and learning rate at 0.001 for 100% data from Penn Treebank corpus. The perplexities produced by NAS, ENAS, DARTS, NAO, and VLC-GA for RNNLM on the Penn Treebank dataset are listed in Table 1. The proposed VLC-GA method achieved comparable results with NAS. VLC-GA achieved perplexity of 57.4 while NAS achieved 62.4 perplexities. The perplexity score of VLC-GA is close with the perplexity score from DARTS, who achieved 56.7 perplexities. However, RNNLM with structure from ENAS, NAO is better with perplexity score 55.8 and 56.6.

TABLE I. PERPLEXITY SCORE

Method	Perplexity	Search Method	GPU days
NAS	62.4	RL	1e4 CPU day
ENAS	55.8	RL	-
DARTS	56.7	Gradient-based	1
NAO	56.6	Gradient-based	0.4
VLCGA based	57.4	VLCGA based	3

VLC-GA cannot achieve better perplexity results since it does not find a better structure for RNNLM within 100 generations and 50 chromosomes for each generation. If the evolution of the chromosome is longer, it can find a better RNN structure.

V. CONCLUSION

The proposed VLC-GA gives a comparable result with NAS to find the optimum structure of RNN. The different lengths of the chromosome can represent a different number of neurons and their connections. First, the VLC-GA uses a small portion (5%) of the dataset to reduce computational time. Once the optimum structure is found, then RNNLM will be trained using a full (100%) dataset. The optimum structured RNNLMs found by VLC-GA produce a perplexity of 57.46 on the Penn Treebank dataset. This result is better than NAS that gives a perplexity of 62.4. It proves that the proposed VLC-GA is sufficient to optimize the RNN structure.

REFERENCES

- [1] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent Advances in Recurrent Neural Networks," pp. 1–21, 2017.
- [2] A. A. Nugraha, A. Arifianto, and Suyanto, "Generating Image Description on Indonesian Language using Convolutional Neural Network and Gated Recurrent Unit," in *2019 7th International Conference on Information and Communication Technology (ICICT)*, 2019, pp. 1–6.
- [3] A. Prayitno and S. Suyanto, "Segment Repetition Based on High Amplitude to Enhance a Speech Emotion Recognition," *Procedia Comput. Sci.*, vol. 157, pp. 420–426, 2019.
- [4] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," pp. 1–38, 2015.
- [5] R. Adelia, S. Suyanto, and U. N. Wisesty, "Indonesian Abstractive Text Summarization Using Bidirectional Gated Recurrent Unit," *Procedia Comput. Sci.*, vol. 157, pp. 581–588, 2019.
- [6] P. M. Hanunggul and S. Suyanto, "The Impact of Local Attention in LSTM for Abstractive Text Summarization," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 54–57.
- [7] H. S. Nugraha and S. Suyanto, "Typographic-Based Data Augmentation to Improve a Question Retrieval in Short Dialogue System," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 44–49.
- [8] M. H. Toding Bunga and S. Suyanto, "Developing a Complete Dialogue System Using Long Short-Term Memory," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 326–329.
- [9] Y. W. Chandra and S. Suyanto, "Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model," *Procedia Comput. Sci.*, vol. 157, pp. 367–374, 2019.
- [10] T. Elsken, J. H. Metzen, and F. Hutter, "Neural Architecture Search," pp. 63–77, 2019.
- [11] A. Thengade and R. Dondal, "Genetic Algorithm – Survey Paper," *MPGI Natl. Multi Conf. Int. J. Comput. Appl.*, no. March, pp. 975–8887, 2012.
- [12] Suyanto, *An informed genetic algorithm for university course and student timetabling problems*, vol. 6114 LNAI, no. PART 2, 2010.
- [13] R. M. Karim and S. Suyanto, "Optimizing Parameters of Automatic Speech Segmentation into Syllable Units," *Int. J. Intell. Syst. Appl.*, vol. 11, no. 5, pp. 9–17, 2019.
- [14] N. R. Emilia, S. Suyanto, and W. Maharani, "Isolated word recognition using ergodic hidden markov models and genetic algorithm," *Telkomnika*, vol. 10, no. 1, pp. 129–136, 2012.
- [15] H. R. Ahmed and J. I. Glasgow, "Swarm Intelligence: Concepts, Models and Applications," *Queen's Univ. Sch. Comput. Tech. Reports*, vol. 585, no. February, pp. 1–50, 2012.
- [16] A. P. Pertiwi and S. Suyanto, "Globally Evolved Dynamic Bee Colony Optimization," in *KES*, 2011, no. 1, pp. 52–61.
- [17] V. Clarissa and S. Suyanto, "New Reward-Based Movement to Improve Globally-Evolved BCO in Nurse Rostering Problem," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 114–117.
- [18] F. Ghaisani and S. Suyanto, "Discrete Firefly Algorithm for an Examination Timetabling," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 1–4.
- [19] U. Abdillah and S. Suyanto, "Clustering Nodes and Discretizing Movement to Increase the Effectiveness of HEFA for a CVRP," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 4, pp. 774–779, 2020.
- [20] G. K. Jati and S. Suyanto, "Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem," in *Adaptive and Intelligent Systems: Second International Conference, ICAIS 2011, Klagenfurt, Austria, September 6-8, 2011*, 2011, no. 1, pp. 393–403.
- [21] G. K. Jati and S. Suyanto, "Discrete Cuckoo Search for Traveling Salesman Problem," in *2012 7th International Conference on Computing and Convergence Technology (ICCT)*, 2014, pp. 993–997.
- [22] G. K. Jati, R. Manurung, and S. Suyanto, *Discrete Firefly Algorithm for Traveling Salesman Problem: A New Movement Scheme*. Elsevier Inc., 2013.
- [23] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, 2002.
- [24] K. Park, D. Shin, and S. Chi, "Variable Chromosome Genetic Algorithm for Structure Learning in Neural Networks to Imitate Human Brain," *Appl. Sci.*, vol. 9, no. 15, p. 3176, 2019.
- [25] P. Bajpai, "Genetic Algorithm – an Approach to Solve Global Optimization Problems," vol. 1, no. 3, pp. 199–206.
- [26] A. Bajpai and A. Arya, "International Journal of Computer Sciences and Engineering Open Access Big Data Analytics in Cyber Security," no. October, 2018.
- [27] H. Stringer, "Behavior Of Variable-length Genetic Algorithms Under Random Selection," 2007.
- [28] M. Karafi and J. H. Cernock, "Recurrent neural network based language model.Pdf," no. September, pp. 1045–1048, 2010.
- [29] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. H. Černocký, "Empirical evaluation and combination of advanced language modeling techniques," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, pp. 605–608, 2011.
- [30] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký, "Strategies for training large scale neural network language models," *2011 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2011, Proc.*, pp. 196–201, 2011.
- [31] B. Zoph and Q. V. Le, "Neural Architecture Search With RL," *Icml 2018*, vol. 32, pp. 1–16, 2017.
- [32] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient Neural Architecture Search via parameter Sharing," *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 9, pp. 6522–6531, 2018.
- [33] R. Luo, F. Tian, T. Qin, E. Chen, and T. Y. Liu, "Neural architecture optimization," *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, no. NeurIPS, pp. 7816–7827, 2018.
- [34] I. N. G. P. E. P. Rediction, "a Ccelerating N Eural a Rchitecture S Earch Us -," vol. 2, pp. 1–16, 2018.
- [35] G. Miller and P. Todd, "Neural Networks using Genetic Algorithms," *Int. J. Comput. Appl.*, vol. 77, no. 14, pp. 6–11, 2013.
- [36] J. R. Koza, D. Genetic Programming Conference (1st : 1996 : Stanford University), and L. Pyeatt, "A comparison between cellular encoding and direct encoding for genetic neural networks," *Proc. 1st Annu. Conf. Genet. Program.*, p. 568, 1996.

- [37] D. Moriarty, "Symbiotic evolution of neural networks in sequential decision tasks," no. January, 1997.
- [38] D. E. Moriarty and R. Miikkulainen, "Efficient reinforcement learning through symbiotic evolution," *Mach. Learn.*, vol. 22, no. 1–3, pp. 11–32, 1996.
- [39] R. Socher, L. Neural, and A. See, "Language Modelling and Recurrent Neural Networks."
- [40] S. N. Pawar and R. S. Bichkar, "Genetic algorithm with variable length chromosomes for network intrusion detection," *Int. J. Autom. Comput.*, vol. 12, no. 3, pp. 337–342, 2015.