Linear-dependent Multi-interpretation Neuro-Encoded Expression Programming

Jun Ma[1], Fenghui Gao[1], Shuangrong Liu[1], Lin Wang[1,*]
[1]Shandong Provincial Key Laboratory of Network based Intelligent Computing,
University of Jinan, Jinan, 250022, China
[*]Corresponding Author, Email: wangplanet@gmail.com

ABSTRACT

Neuro-Encoded Expression Programming (NEEP) implements the continuous coding for the discrete solution through recurrent neural networks (RNNs), and smooths sharpness of the discrete coding. However, the insertion model generating linear coding in NEEP breaks the coherence of linear coding of RNNs, because the resulting symbols tend to be cluttered when RNNs learn the incoherent sequence relationships. Meanwhile, the redundancy phenomenon that different RNNs generate the same code results in that lots of solutions with the same performance exist in the search space, and causes the decrease for search efficiency. To address these problems, the linear-dependent multi-interpretation NEEP(LM-NEEP) is proposed in this research. LM-NEEP tackles the incoherence problem by employing a linear dependence strategy, and the multi-interpretation strategy is adopted to deal with the redundancy problem in search space. The capability of LM-NEEP is estimated on several symbolic regression problems. The experimental results display that the LM-NEEP significantly outperforms NEEP and some classical genetic programming methods.

CCS CONCEPTS

• **Theory of computation** → *Design and analysis of algorithms*; *Probabilistic computation*; • **Computing methodologies** → **Genetic programming**; **Neural networks**.

KEYWORDS

Genetic Programming, Recurrent Neural Network, Neuro-Encoded Expression Programming

ACM Reference Format:
Jun Ma[1], Fenghui Gao[1], Shuangrong Liu[1], Lin Wang[1,*]. 2021. Linear-dependent Multi-interpretation Neuro-Encoded Expression Programming. 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/10.1145/3449726.3459498

1 INTRODUCTION

Neuro-Encoded Expression Programming (NEEP)[2] proposes a continuous neural encoding approach to improve smoothness and stability of the search space[5], which improves the conventional linear representation in genetic programming for symbolic regression problems[1]. In NEEP, the k-expression string as solution is optimized by optimizing the encoder. Thus, the genotype of NEEP represents the encoder and the phenotype represents the string. However, the NEEP still exists two weaknesses. First, NEEP introduces a insert method of symbol nodes, which leads to the inconsistency between the fixed length k-expressions strings generated by encoder in NEEP and the strings evaluated by fitness evaluation function. For example, the encoder should have generated a string S_1 :"$+, -, x, y, z$", but when "$-$" is inserted before the "+", the string will become S_2 :"$-, +, x, y, z$" and S_2 will be evaluated instead of S_1. Second, several different genotypes representing encoder map a phenotype representing string. In the searching process, the phenotype is constant even though the genotype changes. This causes dramatic decrease in search efficiency.

To address these challenges, the novel linear-dependent multi-interpretation NEEP(LM-NEEP) is proposed. LM-NEEP generates the k-expression strings through a linear dependency strategy instead of the original insert method of symbol nodes in NEEP. Then, In order to solve that different genotypes generate the same phenotypes, multi-interpretation strategy is proposed. For the linear dependency, when a linear string is formed, encoder generates one node after another node. Thus, the encoder is considered to learn a dependency relationship between nodes in the linear string. In the linear dependency strategy, the position vector of node in a linear string is added as input of encoder, and the generated node in string are produced in the order. Linear dependency strategy ensures that the evaluated string is a generated string by encoder. The multi-interpretation strategy employs a genotype to map several phenotype by decoder, thus LM-NEEP increases the probability for searching the optimal phenotype.

2 METHODOLOGY

2.1 General Framework

LM-NEEP is composed of the encoder and the decoder. After the initialization of LM-NEEP, the encoder with linear dependency strategy generates a k-expression string that is considered as the chromosome of gene expression programming (GEP)[4]. Then, the decoder with multi-interpretation strategy decodes the string to several expression trees(ETs) and the fitness values of ETs are evaluated. Finally, the optimal value among these fitness values stands for the fitness to evaluate the encoder, and is adopted to update parameters of the encoder by optimization algorithms (*e.g.* Genetic Algorithm).

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

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3459498

2.2 Encoder

The encoder is approximated by an recurrent neural network (RNN). The inputs are positions of generated node in the string coded by the one-hot method. The output are a series of probability values of all alternative symbols that conform to the k-expression syntax at the current node position. It is worth noting that if the current symbolic node position is in the tail of k-expression string, only the probability of the termination symbol is calculated for finishing a linear symbolic string with correct syntax. Linear dependency is reflected in the fact that RNN learns actually how to generate the next node dependent on one generated node in a linear k-expression string.

2.3 Decoder

The decoder translates ETs according with a string, and computes the values of expressions when the data is inputted. In the decoder, the multi-interpretation strategy is used to build ETs from a string. As an example, supposing that the encoder generates a mentioned string S_1, the string is divided into several substrings, such as "$+, -, x, y, z$", "$-, x, y, z$", "x, y, z". These substrings is adopted to build ETs, such as "$x + y - z$","$x - y$","x", and computes values of ETs. Then, the optimal value are selected as the evaluation value the string generated by encoder.

3 EXPERIMENT

In the experiment, the performance of LM-NEEP is compared with NEEP, GP and GEP on 14 benchmarks and two real-world problems from the paper of NEEP[2]. For a fair comparison, common parameters of these methods are initialized with the same value. The experiment is configured as the suggestion from the study[2]. Among these algorithms, the number of generations is 500, the size of population is 100, the maximum code length is 61. In GP[3], the depth of max tree is 10, the length of maximum tree is 61, the depth of maximum mutation is 4, the depth of maximum crossover is 10, the grow depth is 1, the mutation rate is 0.3, the crossover rate is 0.7, the tournament size is 3. For GEP[3], the header length is 30, IS rate is 0.1, RIS rate is 0.1, the inversion rate is 0.1, the mutation rate is 0.06, the crossover rate is 0.7 and the tournament size is 2. In NEEP and LM-NEEP, the header length is 30, the number of hidden neurons is 40, other parameters of GA, PSO, CMA-ES were specified by default. Note in particular that the set of functions for Sphere5 and Poly10 is $[+, -, *, /]$, and the other benchmarks are $[+, -, *, /, sin(x), cos(x), e^n, ln|x|]$. Moreover, the division is protected by $f = x/(y + \epsilon)$, ϵ is a small number (*e.g., 1e-100*).

50 independent test experiments are conducted for LM-NEEP, GP, GEP, and NEEP, and the mean-square error (MSE) is used as the metric for evaluating the performance. The average performance of each generation is shown in the Figure. 1. As results, we can find that LM-NEEP standouts other algorithms on 10 of the 16 benchmarks, and it is competitive on the remaining problems. For the convergence capacity, it can be found that LM-NEEP converges faster than others.

4 CONCLUSIONS

This paper presents a novel Linear-dependent Multi-interpretation Neuro-Encoded Expression Programming method for enhancing the capability of NEEP. The linear dependence strategy and the multiple interpretation strategy of LM-NEEP are effectively to avoid solutions to fall into local optimums, and endows the ability for LM-NEEP to find optimal solutions quickly. The experimental analysis shows that LM-NEEP has the potential to improve test accuracy and efficiency. There are more interesting research directions in the future. LM-NEEP can insert constants to strings. Also, LM-NEEP is able to offer the solutions for more applications.

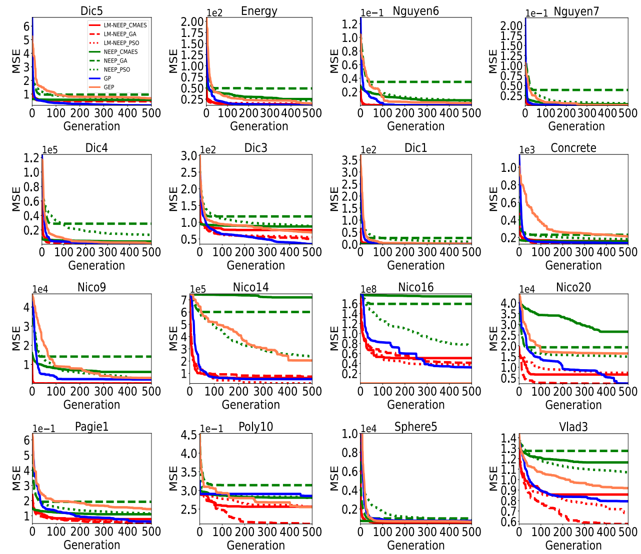

Figure 1: Evolution of the average testing errors of 50 independent trials for compared algorithms. The horizontal axis represents generations, and the vertical axis represents MSE

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China under Grant No. 61872419, No. 61573166, No. 61572230, No. 61873324, No. 81671785, No. 61672262, No. 61903156. Shandong Provincial Natural Science Foundation No. ZR2019MF040, No. ZR2018LF005. Shandong Provincial Key RD Program under Grant No. 2019GGX101041, No. 2018CXGC0706, No. 2017CXZC1206. Taishan Scholars Program of Shandong Province, China, under Grant No. tsqn201812077.

REFERENCES

[1] Baligh Al-Helali, Qi Chen, Bing Xue, and Mengjie Zhang. 2021. A new imputation method based on genetic programming and weighted KNN for symbolic regression with incomplete data. *Soft Computing* (2021), 1–20.
[2] Aftab Anjum, Fengyang Sun, Lin Wang, and Jeff Orchard. 2019. A Novel Neural Network-Based Symbolic Regression Method: Neuro-Encoded Expression Programming. In *International Conference on Artificial Neural Networks*. Springer, 373–386.
[3] François-Michel De Rainville, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. Deap: A python framework for evolutionary algorithms. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. 85–92.
[4] Candida Ferreira. 2001. Gene expression programming: a new adaptive algorithm for solving problems. *arXiv preprint cs/0102027* (2001).
[5] Ying Jin, Weilin Fu, Jian Kang, Jiadong Guo, and Jian Guo. 2019. Bayesian symbolic regression. *arXiv preprint arXiv:1910.08892* (2019).