Evolutionary Neural Architecture Search for Image Classifier
Mid-term Report

Bowen Zheng, Shijie Chen, Shuxin Wang
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, Guangdong, China

***Abstract*—EvoNAS project**

I. Introduction

II. Proposed Framework

Variable	Name
N_{total}	size of the total population

TABLE I: Notation

A. Fuzzy Rules

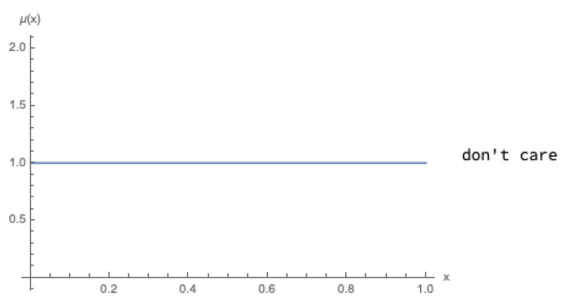

Fig. 1: The $don'tcare$ function

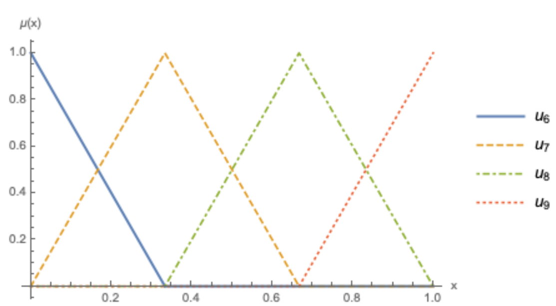

Fig. 2: Membership Functions with 4 intervals

The antecedent part of R_q is a set of antecedent fuzzy sets:

$$A_q = \{A_{qi} | i = 1, 2, ..., n\} \quad (1)$$

B. Fuzzy Classifier

C. Generate Fuzzy Rule From Training Patterns

III. Hybrid Genetics-based Machine Learning Framework

A. NSGA-II

1) *Elite-preserving:*

2) *Pareto ranking:*

3) *Crowding measure:*

B. Michigan-style GBML

IV. Asynchronous Parallel Distributed System Design

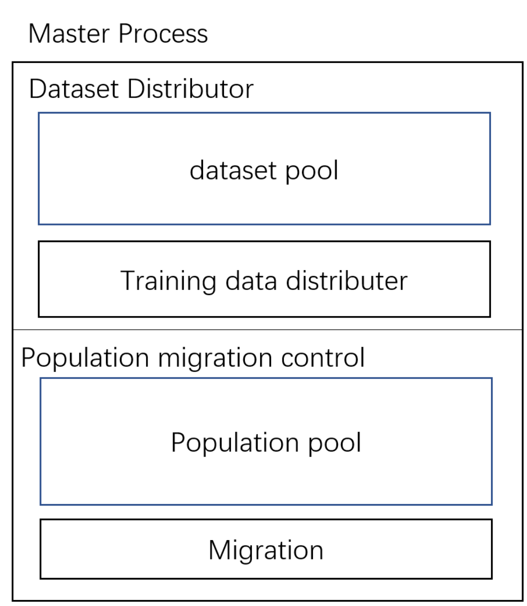

Fig. 3: The master process

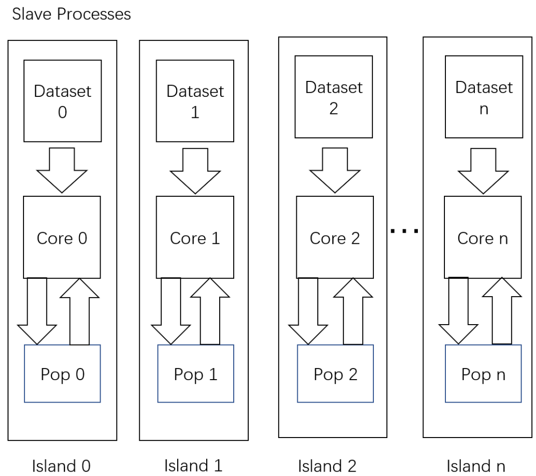

Fig. 4: The slave process

A. Master and Slave process

1) *Master Process:*

2) *Slave Processes:*

B. Dataset Distributor:

C. Population Migration Control:

D. Asynchronous Population Migration

V. Computational Experiments

A. Pareto Front

1) With different cores: As is shown by Fig.5 and Fig.6 ($I_{update} = 100, N_{total} = 264$), our model is able to get results that is similar to non-parallel models. Note that with the increase of the number of cores, the ability of the model to obtain classifiers with more rules and antecedent fuzzy sets is decreasing, due to the decrease of sub-population. It's acceptable since our objective is to minimize the two objectives.

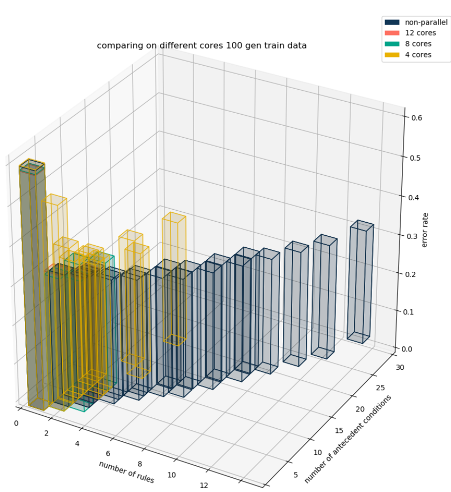

Fig. 5: Pareto Front on training data

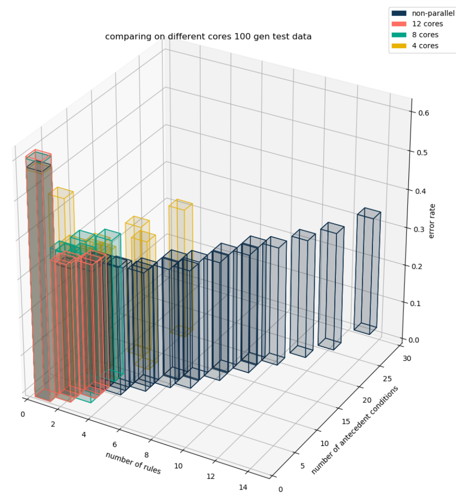

Fig. 6: Pareto Front on test data

2) With different exchange interval: We can see from Fig.7, Fig.8 and Fig.9 ($N_{total} = 264, N_{island} = 8$) that the choice of exchange interval also impacts the performance of our model. A longer exchange interval leads to less communication between master and slave processes, thus reduces running time.

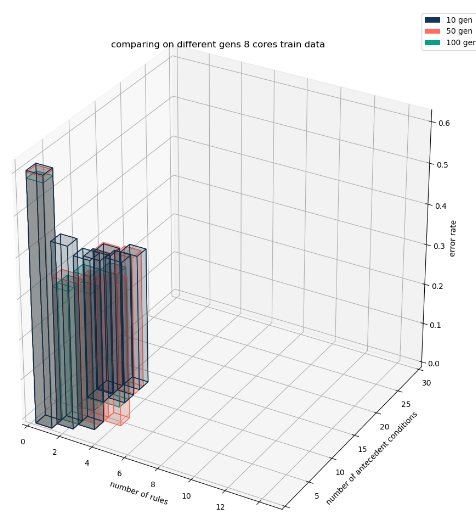

Fig. 7: Pareto Front on training data with different I_{update}

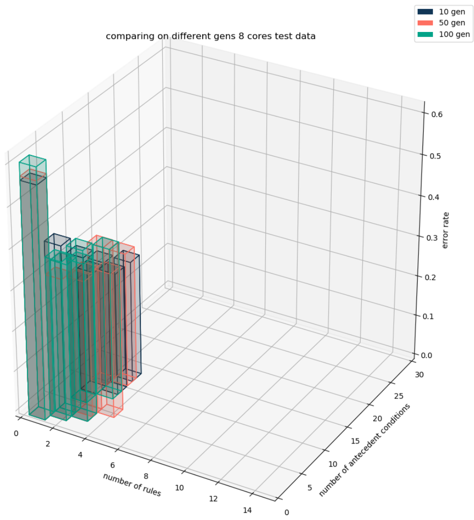

Fig. 8: Pareto Front on training data with different I_{update}

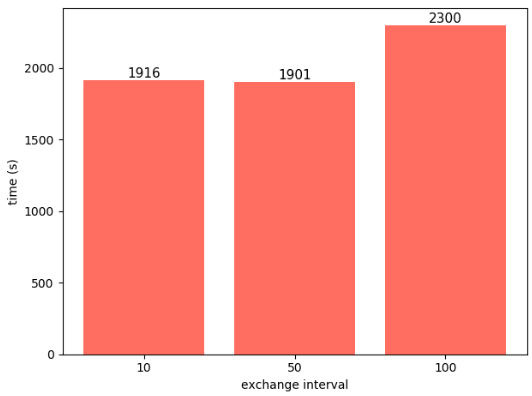

Fig. 9: Computational time with different I_{update}

B. Computation Time

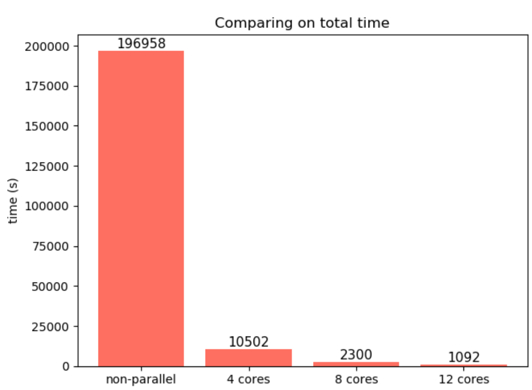

Fig. 10: Computation time of 3000 generations, N_{pop} = 264.

With n cores, we observe a speed up of up to n^2 times in our computational experiments.

C. Compare with other model

We compared our model with the synchronized model by Nojima et al. [1], as is shown in Fig.11, Fig.12 and Fig.13. The experiments are done using 8 cores.

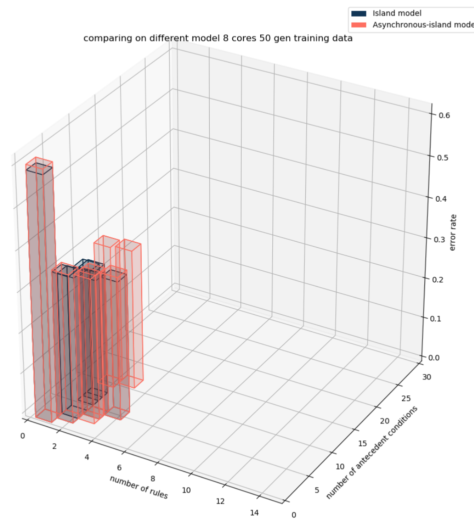

Fig. 11: Comparison on training data.

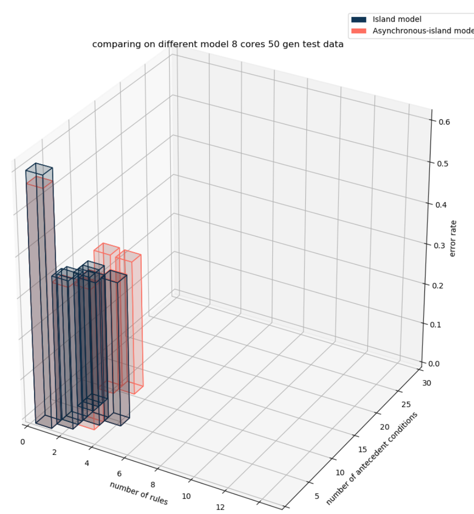

Fig. 12: Comparison on test data.

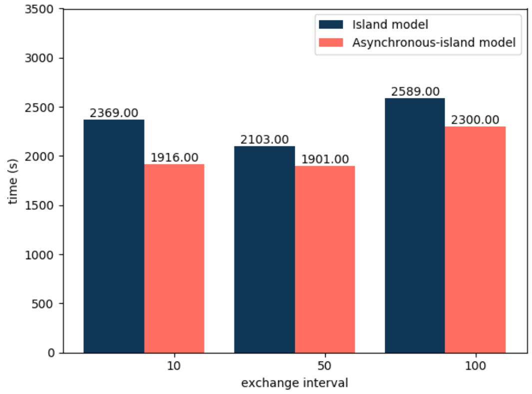

Fig. 13: Comparison on total time.

VI. Conclusion

VII. Contribution

References

[1] Y. Nojima, Y. Takahashi, and H. Ishibuchi, "Application of parallel distributed implementation to multiobjective fuzzy genetics-based machine learning," in *Asian Conference on Intelligent Information and Database Systems*, pp. 462–471, Springer, 2015.