NOVA INFORMATION MANAGEMENT SCHOOL

NEURAL AND EVOLUTIONARY LEARNING

Models Benchmarking

Professors:

Karina Rebuli

Leonardo Vanneschi

2024/2025

Iris Moreira-20240659

Rafael Borges-20240497

Leonardo di Caterina-20240489

Table of Contents

1. Introduction

This report benchmarks five machine learning models on a regression task to predict crude protein weight in chickens, using Root Mean Squared Error (RMSE) as the core evaluation metric. The study compares three models from the Genetic Programming family (GP, GSGP, and SLIM-GSGP) against two from the Neural Network family (a standard NN and NEAT). The primary objective is to evaluate their respective performance, behavior, and practical applicability for this task.

2. Train Function

To streamline the evaluation of models from different libraries, we created a single, unified training function. This function was built to work with the slim-gsgp library, as most of our models originate from it. When testing a model from a different framework, we first wrote a wrapper to make it compatible with our standard training function. This strategy prevented us from writing and debugging unique training loops for each model, which made our nested cross-validation process more reliable and less susceptible to implementation errors.

2.1. Nested Cross Validation and Hyperparameter Tuning

For robust model evaluation, a nested K-Fold cross-validation (CV) strategy was implemented. This was preferred over Monte Carlo CV as it ensures each data point contributes to an outer test set exactly once, providing a comprehensive assessment of generalization performance across the entire dataset. The nested CV was configured with 10 outer folds for performance estimation and 5 inner folds for hyperparameter optimization. Within each outer loop, hyperparameter tuning was conducted via a grid search on the inner folds; the combination yielding the minimum median performance metric across these inner folds was then used to train and evaluate the model on that outer fold.

2. Genetic Programming

According to the standard GP, general results, solutions converge rapidly and do not improve substantially within the first 100 or 150 iterations, while continuing to grow in size, hinting at bloating. In most cases, the training and validation curves align closely, suggesting minimal overfitting. The hyperparameters chosen to train the outer fold had no specific relevant pattern.

Population diversity varies markedly across combinations, where some fail to converge, showing multiple peaks in fitness standard deviation; others follow a more monotonic trajectory with only a few fluctuations. This could be further studied by adjusting the population size and selection pressure.

3. Geometric Semantic Genetic Programming

The performance trajectory of the GSGP model paralleled that of standard GP, characterized by rapid initial convergence that typically plateaued within the first generations. The model also exhibited significant solution bloat, an anticipated outcome of its unconstrained evolutionary nature. However, a key distinction was GSGP's more pronounced divergence between training and testing performance, indicating a stronger tendency to overfit. Consequently, while GSGP's average test errors were higher than those of GP, they also showed lower variance, suggesting the model converged to more consistent but ultimately suboptimal solutions.

Overall, GSGP yielded inferior results compared to classic GP. This outcome was contrary to expectations, as the geometric semantic operators were expected to improve search efficacy. The limited exploration of certain hyperparameters, such as the 'mutation step', may have contributed to these results and warrants further investigation.

In terms of population diversity, a steep decline was observed during the initial iterations. Future experimental runs should explore the impact of reduced selection pressure to potentially enhance and maintain population diversity.

4. Semantic Learning Algorithm Inflate Deflate Mutations

Like the other GP models tested, the SLIM algorithm's performance quickly plateaued within the first few iterations. Its "inflate and deflate" feature did produce much more compact solutions than GSGP, but the model still struggled with solution bloat.

We also observed that SLIM maintained a more diverse population of solutions. Interestingly, this diversity did not lead to better performance, as its final results failed to outperform the standard GP and GSGP models. This was a surprising outcome, as we had hypothesized that SLIM's geometric semantic properties would steer the evolutionary search toward better solutions as mentioned earlier and with reduced size.

5. Neural Networks

Our wrapper class includes a Net_Arc module that dynamically constructs multi-layer perceptrons with configurable hidden layer sizes and activation functions, handles data normalization, and manages various optimization algorithms including gradient descent variants (GD, SGD, Mini-batch SGD), adaptive methods (Adam, RMSprop), and averaged stochastic gradient descent (ASGD). Our neural network implementation utilized our nested cross-validation framework with grid search optimization across key hyperparameters including hidden layer architectures [(3,4), (3,4,4), (5,6)], six different optimizers, and two learning rates (0.005, 0.01). The models were trained for 250 epochs using RMSE loss with ReLU activation functions. Results revealed that only Adam and SGD optimizers were selected for outer fold training, with the three-layer architecture (3,4,4) never being chosen, suggesting simpler architectures were more effective. The most frequently selected combination featured hidden layers of size (3,4), SGD optimizer, and learning rate 0.005, demonstrating good generalization with training and test loss curves tracking closely across most folds. Some rapid convergence was observed in initial epochs, though consistent improvement continued throughout training, ultimately achieving near-zero errors for both training and test sets, indicating effective learning without significant overfitting.

6. NeuroEvolution of Augmenting Topologies

For NEAT integration, we developed a custom NEATWrapper class that maintains compatibility with our nested cross-validation framework while handling data format translation, evolutionary process management, and RMSE-based fitness evaluation. We conducted two experiments investigating speciation parameters: conservative coefficients (disjoint: 0.8, weight: 0.3) producing 1-3 species, and higher coefficients (0.9, 0.8) promoting greater diversity. Using our cross-validation with 100 individuals over 100 generations, results showed reduced speciation achieved superior fitness while increased diversity degraded performance due to fitness sharing across smaller niches. Models demonstrated good generalization, typically converging around generation 70 with minimal hyperparameter sensitivity

7. Models Comparison

7.1. Overfitting and Premature Convergence

To compare all models, it was grouped all the outer results for each one. Based on the boxplots, Figure 1a, we can observe that the GSGP and SLIM-GSGP show a tendency to overfit, since the test has a much higher spread than the train. Looking at the training results of GP, it can be observed that it has a higher spread than all the other train results, which seems to have an instability pattern. Compared in an overall view, the NN and the NEAT seem to stand with the best results. Additionally, both have a low spread, with the NEAT having the most similarity between train and test, suggesting that it does not overfit, while the NN seems to show a higher tendency to this problem.

7.2. Model Evolution

In the evolution plot (Figure 1b), it can be observed that the train and test lines are generally aligned with no strong detour. Regarding fitness, there is a visible stagnation of the GSGP and SLIM-GSGP over iterations. On the other hand, NN and NEAT seem to have a more evolution-like topology, starting with high RMSE and converging to a better solution, even if by small steps. Because of this, we consider the latter to not have such marked premature convergence as the GSGP and the GSGP-SLIM. Additionally, none of the models do not get meaningfully better past the 150th iteration.

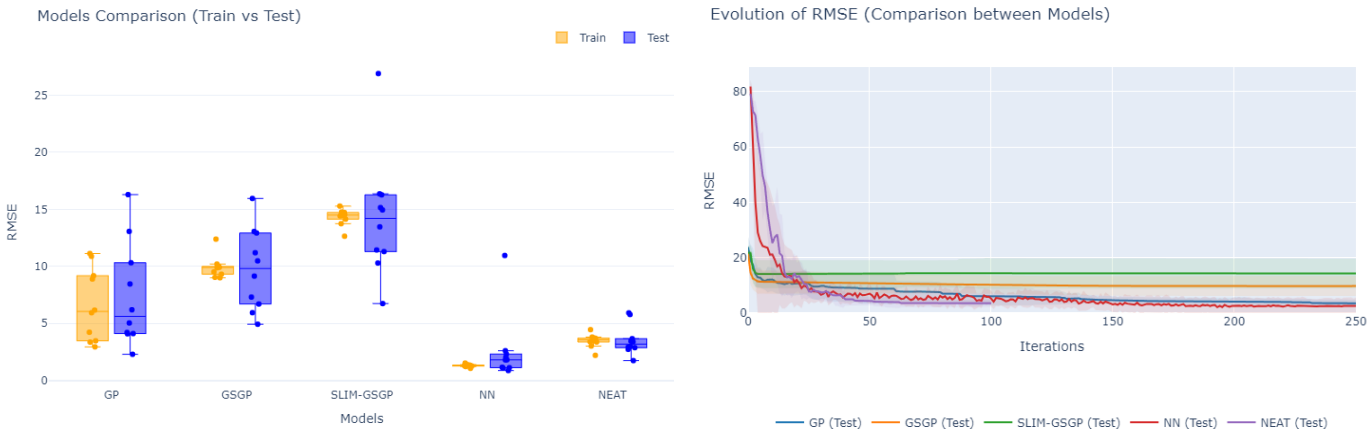

Figure 1: Models Comparisons. (a) Boxplots with train vs test data points from all the outer folds of each model; (b) Evolution plot of the RMSE throughout iterations for train and test for each model.

7.3. Statistical Testing

To determine if the observed performance differences between the models were statistically significant, a two-step non-parametric analysis was performed using a significance level (α) of 0.05.

First, a Friedman test was conducted as a global assessment across all five models. The test rejected the null hypothesis ($p < 0.05$), confirming that at least one model's performance was significantly different from the others.

Following this, a post-hoc analysis was performed using the pairwise Wilcoxon Signed-Rank test with a Holm-Bonferroni p-value adjustment to identify which specific pairs of models differed. The key finding from this analysis was that no statistically significant difference exists between the two top-performing models, NEAT and the standard NN (Figure 2). This suggests they are statistically equivalent in performance within our experimental framework. Detailed results of all statistical tests are available in the supplementary notebook.

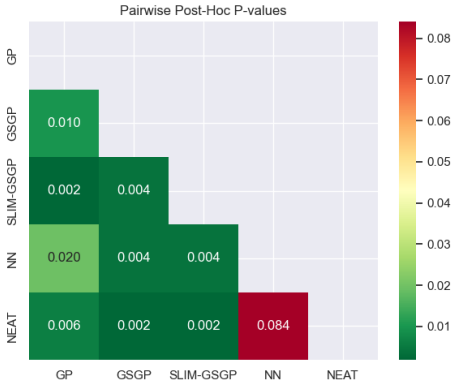

Figure 22: Pairwise Post-Hoc P-values Heatmap

7.4. Computational Cost

Furthermore, we paved the way for the comparison of computational time between models. The results of the average total time to train each model are shown in Table 1. It can be concluded that the time was not

similar throughout the models, which could present a limitation to our analysis. One could consider that this could be determined by taking *n* iterations proportional to the time complexity. However, we are tuning in a various number of combinations depending on the complexity of the model, so the comprehensive time is scaled and turns out to be much more computationally expensive to run all at once.

Nonetheless, when comparing the NN and NEAT, we should pay attention to the fact that not only did NEAT show the less tendency to overfit and an outlier to be less distant, but also that it took less computational time.

Table 1: Average Total Time in seconds that each model takes to train.

	SLIM-GSGP	**GSGP**	**NEAT**	**NN**	**GP**
Total time (sec)	2.48	4.46	6.21	11.24	14.49

8. Conclusion and Discussion

This project conducted a comparative evaluation of five distinct machine learning models, employing a rigorous framework of modular implementation, performance visualization, and statistical analysis. The results conclusively demonstrate that the Neural Network (NN)-based models significantly outperformed the Genetic Programming (GP)-based architectures in both predictive accuracy and generalization ability. This outcome aligns with expectations, given the historical effectiveness of NNs.

A counter-intuitive secondary finding was the outperformance of the standard GP model over its variants, which were theoretically augmented with a semantic operator to improve performance. Furthermore, statistical tests confirmed a significant difference between the performance of most models. The notable exception was the comparison between the standard NN and NEAT, whose performance was not found to be statistically distinguishable.

It is important to acknowledge that this statistical validation was conducted using a 10-fold cross-validation procedure. This sample size was a necessary compromise due to computational resource limitations. While our findings are robust within this framework, future work could aim to corroborate these results with a larger number of validation folds to increase statistical power. In summary, this work identifies NN-based architecture as the most effective solution for the problem.

9. Future Work

Consequently, we bring forward some proposals for future work: (1) The time cost component as the principal basis for comparison – With this, would the NEAT surpass the performance of the NN significantly?; (2) Given our results in GP, is GP an unstable model and would the tuning of crossover probabilities (or the usage of mutation) diminish the effect shown and contribute to escape the stagnation seen in the graphs?. These questions would be interesting for further investigation.

10. References

Vanneschi, L., & Silva, S. (2023). Lectures on Intelligent Systems. Springer Nature.

DALabNOVA. (2024, October 12). GitHub - DALabNOVA/slim: The first Python library to bring SLIM-GSGP to life — faster, smarter genetic programming! GitHub. https://github.com/DALabNOVA/slim

NEAT Overview — NEAT-Python 0.92 documentation. (n.d.). Neat-Python.readthedocs.io. https://neat-python.readthedocs.io/en/latest/neat_overview.html

LLMs tools were used to assist with routine technical tasks such visualizations improvement and other mundane computational processes. All analysis, conclusions, and interpretations remain our original work.