Evolutionary algorithms for optimizing Neural Network weights

Seyedmojtaba Mohasel MOJTABA.MOHASEL@YAHOO.COM
Department of Mechanical and Industrial Engineering
Montana State University
Montana, USA

Nitasha Fazal NITASHA.FAZAL@MONTANA.EDU
Gianforte School of Computing
Montana State University
Montana, USA

Behzad Karimi BEHKARIMI@GMAIL.COM
Department of Electrical and Computer Engineering
Montana State University
Montana, USA

Editor: -

Abstract

In this project, we implemented and evaluated different population-based approaches (evolutionary or swarm-based methods) for training a feedforward neural network. We compared the performance of these population-based methods with the results from backpropagation, which was used in a previous project. The task involved both classification and regression on six datasets from the UCI Machine Learning Repository (three for classification and three for regression). The focus of this project was on training the weights and biases of all layers using these population-based techniques on a fully connected neural network. Our experiments provided insights into the effectiveness of these methods by comparing their results in terms of accuracy and convergence. The findings showed that backpropagation consistently outperformed the population-based methods in both classification and regression tasks across all layer configurations. However, the results were mixed when it came to convergence, with varying performance across the different datasets.

Keywords: Neural Networks, Population-based algorithms, evolutionary algorithms, Particle Swarm Optimization (PSO), Differential evolution (DE), Genetic Algorithm (GA)

1 Introduction

Population-based algorithms are commonly used in optimization and machine learning, especially when the problem is complex, non-differentiable, or high-dimensional. These algorithms maintain a population of candidate solutions and iteratively improve them based on a fitness function. When the problem environment or data changes over time (e.g., in dynamic optimization problems), population-based algorithms can adapt more flexibly compared to traditional methods.

In this project, we combined population-based algorithms with neural networks to optimize their weights. We implemented three evolutionary algorithms: Genetic Algorithm

(GA), Differential Evolution (DE), and Particle Swarm Optimization (PSO). Unlike the deterministic approach of gradient descent used in backpropagation, these algorithms take a stochastic approach to evolve the best weights from a population. The balance between exploration and exploitation in these algorithms helps prevent the model from getting stuck in local minima.

For the implementation, we used the forward layers of the feedforward neural network and initialized the weights randomly using PSO, GA, and DE. The fitness was evaluated using the neural network's loss function. Over a fixed number of generations, the algorithms iteratively adjusted the weights and biases to optimize the model. We applied this process to six diverse datasets from the UCI Machine Learning Repository, covering both classification and regression tasks.

The main goal of this project is to enhance the performance of neural networks with multiple hidden layers, as developed in the previous project, by incorporating population-based algorithms. We then compared the results with those obtained using backpropagation (in the last project) for classification and regression tasks. The basic structure of the neural network, including the activation function and number of neurons, remained the same as in the previous project. However, other hyperparameters related to the population-based algorithms were tuned during the implementation of this project.

2 Problem Statement and Hypotheses

The problem in this project consists of several components: developing a neural network classifier and regressor, implementing population-based algorithms (PSO, GA, and DE) alongside backpropagation to adjust the weights and biases, training the classifier on three datasets, and reporting its performance using F1-score, precision, recall, and F1-score (considering the macro average for all metrics). The regressor is trained on three datasets, and its performance is reported using Mean Absolute Error (MAE) and Mean Squared Error (MSE). The project also addresses issues such as handling missing values, converting categorical attributes into dummy variables, and using stratified 10-fold cross-validation for optimization and hypothesis testing in both classification and regression tasks.

2.1 Hypotheses

The backpropagation algorithm is expected to converge faster during training due to its deterministic gradient-based optimization approach. However, population-based algorithms, are anticipated to achieve better final performance on certain datasets.

Our first hypothesis is that backpropagation algorithms will achieve higher F1-scores in classification tasks and lower mean squared error (MSE) in regression tasks compared to evolutionary methods for adjusting the weights and biases of neural networks across all layers.

Null Hypothesis (H0): Backpropagation algorithm will not achieve higher F1-scores in classification tasks or lower MSE in regression tasks compared to evolutionary algorithms.

Alternative Hypothesis (H1): Backpropagation algorithm will achieve higher F1-scores in classification tasks or lower MSE in regression tasks compared to evolutionary algorithms.

We present our second hypothesis as follows:

Data set	# Features	# Instances	Missing Values	Continuous Data	Imbalance	Curse of Dimensionality Risk	Size
Abalone	8	4177	No	Yes	Moderate	Low	Large
Breast Cancer	10	699	Yes	Yes	Moderate	Low	Large
Glass	9	214	No	Yes	High	Moderate	Moderate
Computer Hardware	10	209	No	Yes	Moderate	Moderate	Small
Soybean (small)	35	47	Yes	No	Moderate	High	Small
Forest Fires	12	517	No	Yes	Moderate	Low	Large

Table 1: Dataset Characteristics

Null Hypothesis (H0): Population-based algorithms will not result in a slower convergence rate compared to backpropagation.

Alternative Hypothesis (H1): Population-based algorithms will result in a slower convergence rate compared to backpropagation.

We will use an ANOVA test to compare the performance of all four algorithms. If a performance difference is detected, we will apply a t-test for pairwise comparisons. We will also use visualizations to compare the convergence of the models to evaluate our second hypothesis.

3 Experimental Approach

This work uses six datasets from the UCI Machine Learning Repository, with a brief description provided in Table 1. Class imbalance occurs when one class outnumbers others, leading to biased model performance that favors the majority class. To assess the risk of the curse of dimensionality, we considered both the number of features and training samples. For instance, the Soybean dataset, with 35 features and only 46 instances, is particularly susceptible to this issue. Additionally, we evaluated the dataset size based on the number of training samples.

4 Program design

The dataset undergoes preprocessing through specific classes implemented for each dataset (e.g., BreastCancer and Glass). These classes handle tasks such as removing irrelevant columns, filling missing values with the mode, and generating dummy variables for categorical attributes. Numerical values are normalized using min-max scaling.

We applied stratified 10-fold cross-validation for both regression and classification tasks. Before this, the dataset was split into a tuning set. Neural network hyperparameters were inherited from the previous project, while population-based hyperparameters were tuned in this project using the tuning set.

Four models were developed for each dataset, Model 1 uses a neural network with Genetic Algorithm (GA) for weight optimization. Model 2 uses Differential Evolution (DE). Model 3 uses Particle Swarm Optimization (PSO). Model 4 uses backpropagation. The neural network implementation is flexible, allowing adjustments to the number of layers and neurons per layer.

The DE and GA algorithms initialize the population with random values within a bound of -0.5 to +0.5. For PSO, particles are initialized with both positions and velocities. Each individual in the population represents a complete solution to the neural network, i.e., the weights and biases of each layer. These weights and biases are assigned to the neural network, and the fitness of the population is evaluated using the corresponding loss function: binary cross-entropy for binary classification, multi-class cross-entropy for multi-class classification, and mean squared error (MSE) for regression. In the next step, the population is updated using different rules depending on the algorithm. For DE, a mutant vector $v - i$ is created for each individual x_i by combining the weighted differences of three randomly chosen distinct individuals:

$$v_i = x_r 1 + F(x_{r2} - x_{r3}) \tag{1}$$

A trial vector u_i is created by mixing the mutant vector v_i and the original individual x_i using a crossover probability.

$$u_i^j = \begin{cases} v_i^j & \text{if } \mathrm{rand}_j \leq CR \text{ or } j = j_r \\ x_i^j & \text{otherwise} \end{cases}$$

if u_i performs better than the original individual it replaces the x_i

For GA, pairwise parents are selected using tournament selection to perform a two-point crossover.

$$Individual = argmax_{i \in T} F_i \tag{2}$$

10% of the genes in a chromosome are randomly mutated for exploration, and the least fit 20% of the population are replaced with offspring that have better fitness to form the next generation

For PSO, the velocity and position of particles are updated based on their personal best and the global best positions using the standard PSO formula.

$$v_i(t+1) = wv_i(t) + c_1 r_1 (p_i - x_i) + c_2 r_2 (g - x_i) \tag{3}$$

$$x_i(t+1) = x_i(t) + v_i(t+1)) \tag{4}$$

where v_i is velocity, x_i is position, p_i is personal best, g is global best, and w, c_1, c_2, r_1, r_2 are tuning parameters All the population-based models iterate for a fixed number of generations.

5 Results and Discussion

For classification datasets (Table 2), statistical tests indicated a significant difference (p<0.05) in the performance of neural networks between the Backpropagation algorithm and evolutionary methods on the Breast Cancer and Glass datasets. Backpropagation attained higher F1-scores. However, there was no difference in the performance of different optimizers on the soybean dataset when 0 and 1 hidden layer were evaluated. In the soybean dataset with two hidden layers, Backpropagation and DE attained higher F1-scores (p<0.05) compared to GA and PSO.

For regression datasets (Table 3), there was a significant difference in MSE between PSO and other optimizers ($p < 0.05$) in Abalone 0. In Abalone 1 and Forest Fires 0, backpropagation performed better than the other methods ($p < 0.05$). However, no significant difference was observed in other model configurations across the datasets between different optimizers.

Table 2: Comparison of Accuracy and F1 scores across different datasets and algorithms.

	Backpropagation		GA		PSO		DE	
Dataset	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
Breast cancer 0	0.95	0.96	0.79	0.80	0.75	0.76	0.87	0.87
Breast cancer 1	0.96	0.96	0.87	0.87	0.88	0.88	0.86	0.86
Breast cancer 2	0.96	0.96	0.73	0.68	0.65	0.51	0.79	0.76
Glass 0	0.62	0.60	0.41	0.36	0.36	0.33	0.46	0.36
Glass 1	0.68	0.64	0.46	0.37	0.35	0.18	0.35	0.21
Glass 2	0.67	0.65	0.34	0.18	0.35	0.19	0.35	0.18
Soybean 0	0.57	0.46	0.55	0.44	0.44	0.57	0.55	0.44
Soybean 1	0.55	0.44	0.57	0.46	0.46	0.55	0.57	0.46
Soybean 2	0.57	0.46	0.36	0.19	0.19	0.36	0.55	0.44

Table 3: Comparison of MSE and MAE across different datasets and algorithms.

	Backpropagation		GA		PSO		DE	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Abalone 0	0.10	0.15	0.09	0.17	0.30	0.26	0.09	0.16
Abalone 1	0.57	0.22	0.82	0.34	0.93	0.39	0.84	0.33
Abalone 2	1.59	0.64	0.98	0.36	1.12	0.42	0.90	0.30
Machine 0	0.11	0.15	0.08	0.15	0.47	0.25	0.14	0.17
Machine 1	0.55	0.20	0.87	0.35	0.97	0.39	0.79	0.34
Machine 2	1.58	0.63	1.20	0.51	1.06	0.49	1.00	0.40
Forest fires 0	1.30	0.41	5.33	1.38	1.74	0.68	2.65	0.88
Forest fires 1	1.24	0.33	1.26	0.37	1.24	0.32	1.23	0.32
Forest fires 2	1.24	0.33	1.26	0.31	1.24	0.31	1.24	0.32

5.1 Breast Cancer Dataset

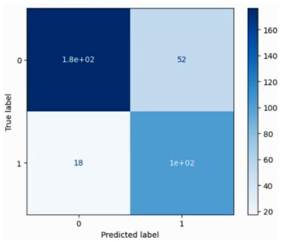
The confusion matrices for different models on the breast cancer dataset (Figure 1) indicate that Backpropagation (BP) consistently delivers the best classification accuracy, minimizing false positives and false negatives. GA performs reasonably well but has slightly more misclassifications compared to BP. PSO shows the highest number of misclassifications, while DE's performance varies.

In Figure 4 (a), (b), and (c), which represent neural networks with 0, 1, and 2 hidden layers respectively on the breast cancer dataset, PSO and GA exhibit faster convergence compared to BP and DE, particularly in the early generations. BP shows more fluctuations.

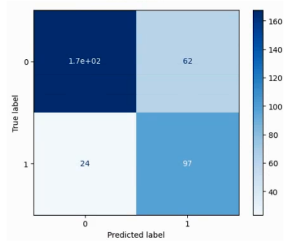
5.2 Glass Dataset

The confusion matrices for the glass dataset (Figure 2) indicate that BP achieves better classification compared to other methods. GA and PSO generally show a higher rate of misclassifications, while DE performs inconsistently, often misclassifying certain classes, suggesting BP's superior performance in handling the complexity of the glass dataset.

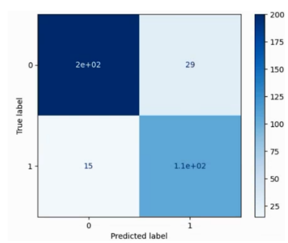
In Figures 5; GA, PSO, and DE converge quickly to low error values, while BP shows slower convergence and more oscillations throughout the iterations. This suggests that the



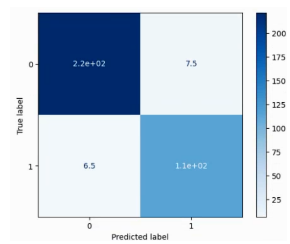
(a) GA 0



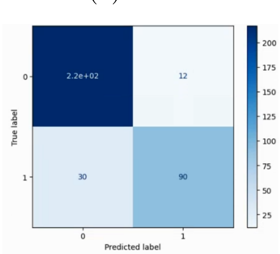
(b) PSO 0



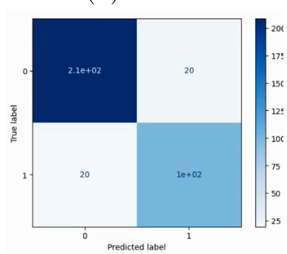
(c) DE 0



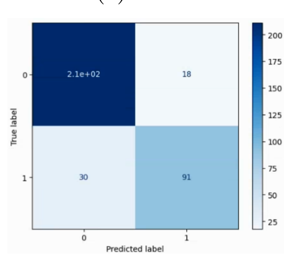
(d) Backprop 0



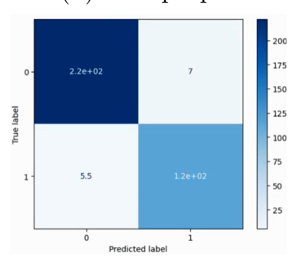
(e) GA 1



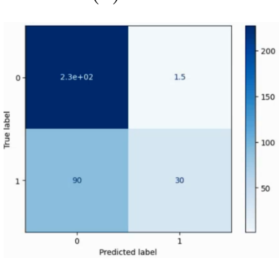
(f) PSO 1



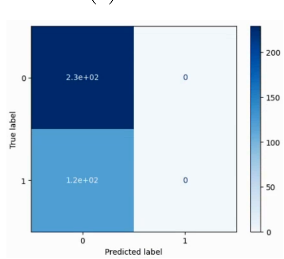
(g) DE 1



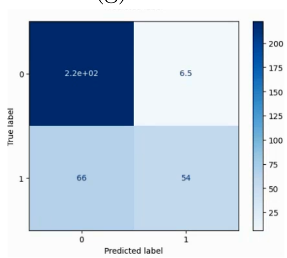
(h) Backprop 1



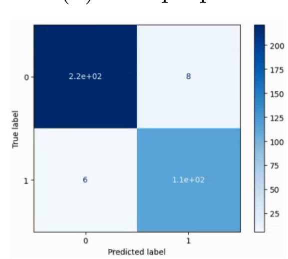
(i) GA 2



(j) PSO 2



(k) DE 2



(l) Backprop 2

Figure 1: Confusion Matrices for Breast Cancer dataset

optimization algorithms (GA, PSO, DE) are more efficient at minimizing the error for the glass dataset compared to BP, which struggles with stability.

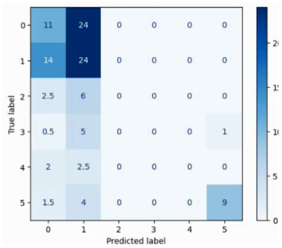
5.3 Soybean Dataset

The confusion matrices for the soybean dataset (Figure 3) reveal that most models—GA, PSO, DE, and BP—struggle with classification accuracy, with frequent misclassifications across all iterations. While BP performs slightly better in distinguishing between classes in some iterations, GA, PSO, and DE exhibit similar misclassification patterns.

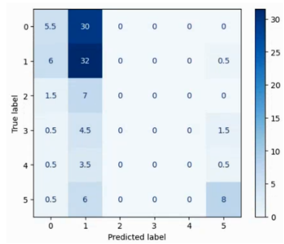
In Figures 6; GA, PSO, and DE converge faster and achieve significantly lower error values compared to BP. BP exhibits almost no convergence throughout the generations, maintaining high error values. This indicates that GA, PSO, and DE are more effective at reducing error in the soybean dataset, while BP struggles to make meaningful improvements.

5.4 Regression datasets (Abalone, Machine, Forest Fires)

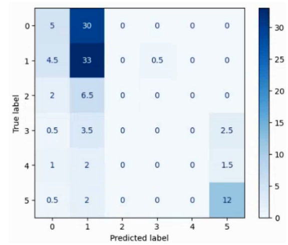
For regression datasets, Backpropagation consistently outperformed evolutionary algorithms, including PSO, GA, and DE (Figures 7, 8, 9).



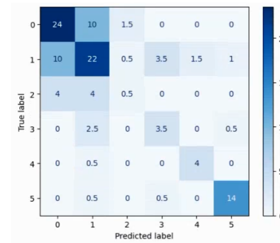
(a) GA 0



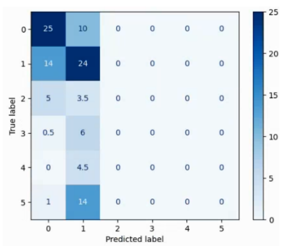
(b) PSO 0



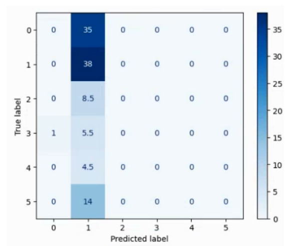
(c) DE 0



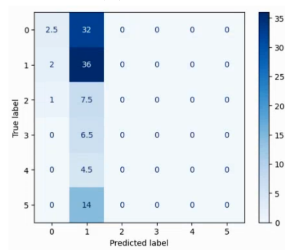
(d) Backprop 0



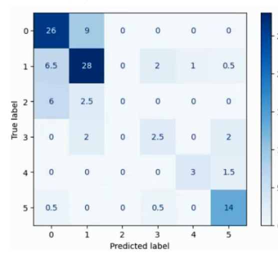
(e) GA 1



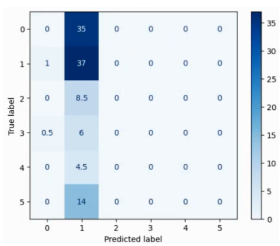
(f) PSO 1



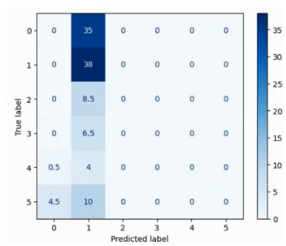
(g) DE 1



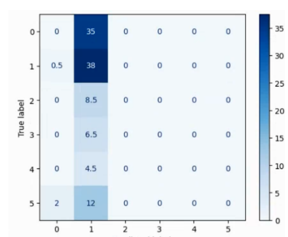
(h) Backprop 1



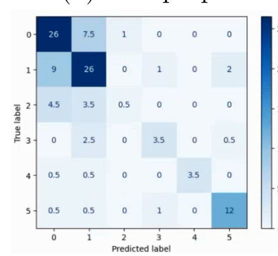
(i) GA 2



(j) PSO 2



(k) DE 2



(l) Backprop 2

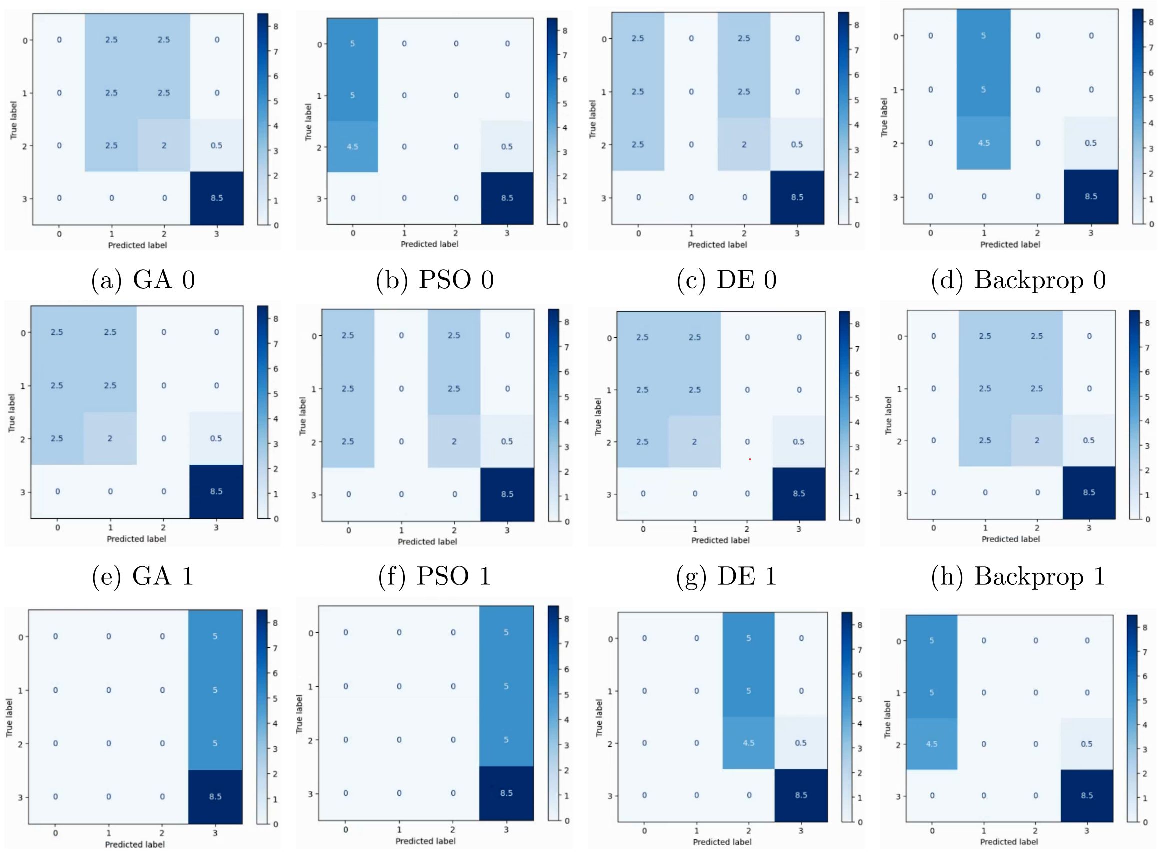
Figure 2: Confusion Matrices for Glass dataset

In the Abalone dataset, Backpropagation showed rapid convergence, while PSO, GA, and DE exhibited slower progress and plateaued early, particularly for Loss 1 and Loss 2. This pattern highlights the difficulty evolutionary algorithms face in efficiently exploring the solution space compared to gradient-based optimization.

For the Machine dataset, Backpropagation continued to show superior performance. While GA and PSO initially competed well, they quickly stagnated, suggesting difficulties in fine-tuning towards the global optimum. DE was consistently the slowest in reducing the loss, further highlighting its limitations in this context.

The Forest Fires dataset proved challenging for all evolutionary algorithms. Backpropagation showed robustness, converging effectively, while PSO, GA, and DE largely failed to reduce the loss significantly, especially for Loss 1 and Loss 2. This suggests that the dataset's complex error landscape hindered the ability of these algorithms to find optimal solutions.

In conclusion, Backpropagation's gradient-based approach allowed it to exploit the differentiable loss landscapes effectively, leading to faster and more reliable convergence. In contrast, the evolutionary algorithms, though useful for exploration, struggled with premature convergence and lacked the precision needed for complex regression problems. Future



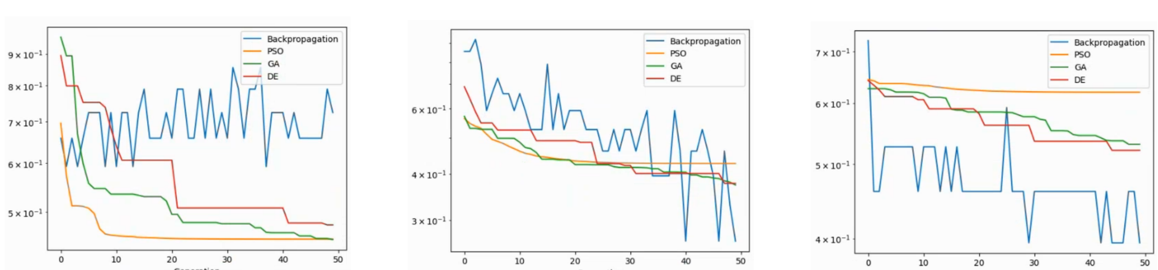
(a) GA 0 (b) PSO 0 (c) DE 0 (d) Backprop 0

(e) GA 1 (f) PSO 1 (g) DE 1 (h) Backprop 1

(i) GA 2 (j) PSO 2 (k) DE 2 (l) Backprop 2

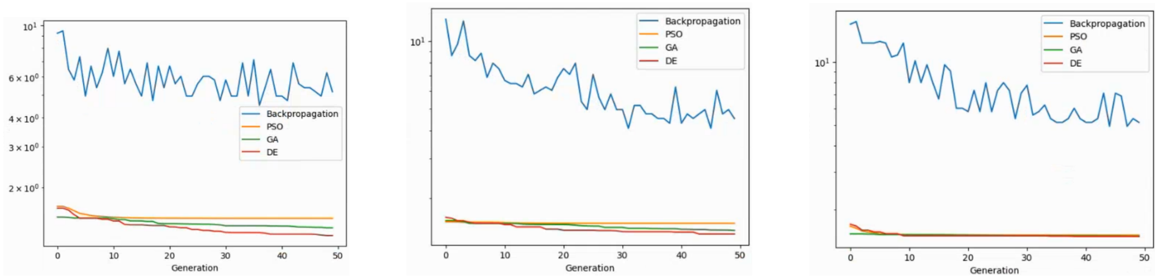
Figure 3: Confusion Matrices for Soybean dataset

improvements may come from hybrid methods that combine the exploration strengths of evolutionary approaches with the exploitation capabilities of gradient-based algorithms.



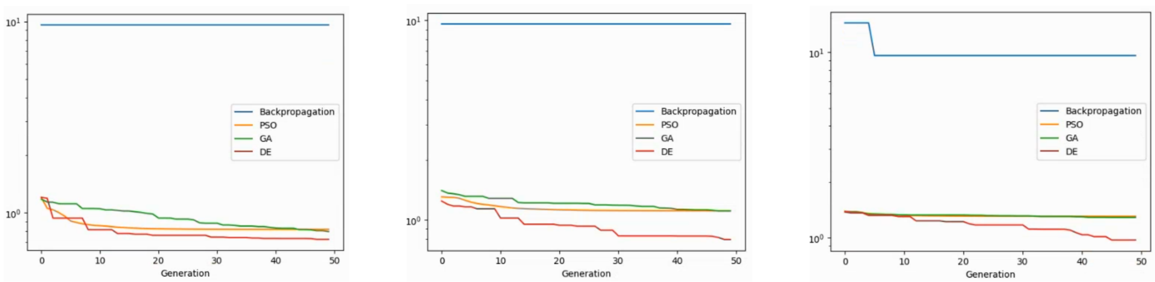
(a) Breast Cancer 0 (b) Breast Cancer 1 (c) Breast Cancer 2

Figure 4: Comparison of Convergence for Breast Cancer Dataset



(a) Glass 0 (b) Glass 1 (c) Glass 2

Figure 5: Comparison of Convergence for Glass Dataset



(a) Soybean 0 (b) Soybean 1 (c) Soybean 2

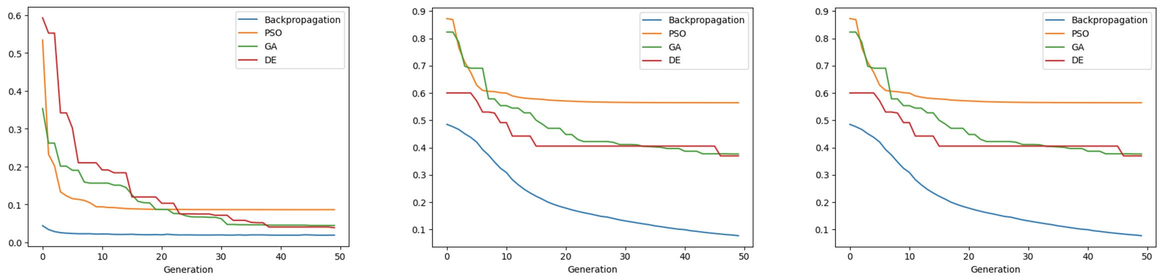
Figure 6: Comparison of Convergence for Soybean Dataset

6 Summary

This paper investigates the effectiveness of population-based GA, DE, and PSO—in optimizing the weights of feedforward neural networks. These methods were applied to six datasets for classification and regression tasks, and compared to backpropagation. The findings indicate that while backpropagation generally achieved better performance in terms of accuracy and F1-scores, population-based methods demonstrated varying convergence results across datasets. The study highlights the limitations of evolutionary algorithms for neural network training, particularly in high-dimensional and non-differentiable problem spaces.

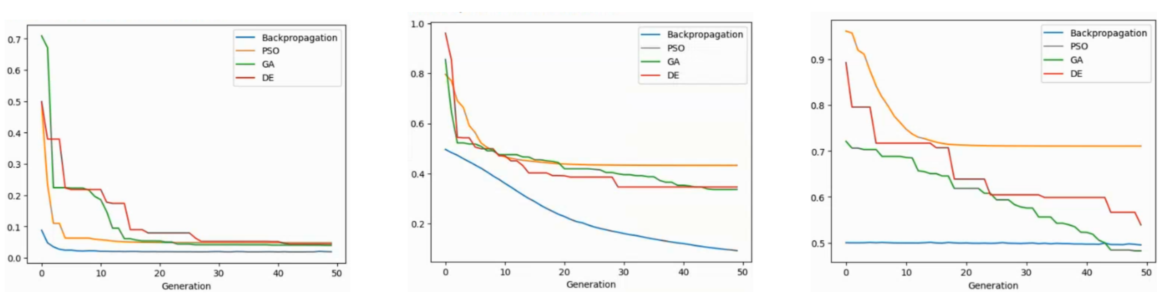
Appendix A.

Overall the workload was distributed equally between group members.



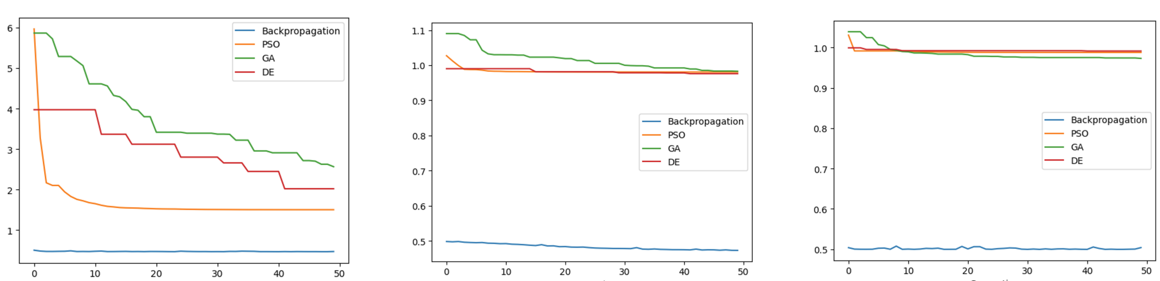
(a) Abalone Loss 0 (b) Abalone Loss 1 (c) Abalone Loss 2

Figure 7: Loss Plots for Abalone Dataset



(a) Machine Loss 0 (b) Machine Loss 1 (c) Machine Loss 2

Figure 8: Loss Plots for Machine Dataset



(a) Forest Fires Loss 0 (b) Forest Fires Loss 1 (c) Forest Fires Loss 2

Figure 9: Loss Plots for Forest Fires Dataset