**Abstract:**

Disease prevention has been the focus in health-related research for years from cardiovascular disease prevention to early cancer detection. The aim is to develop new technologies that can process big quantity of data, interpretating features and extracting and creating correlation between characteristics. Tasks that even a doctor with years of experience in the area cannot always do with the precision of a machine learning model. That is why is key to develop machine learning models as Multi-Layer Perceptron (MLP) with the highest precision possible. To reach the best possible performance is important to do a proper hyperparameters tuning, this can be done by heuristic techniques or based on related work. However, finding the best hyperparameters configuration requires time and computer resources, and the hyperparameters set performance can vary depending on the data and the preprocessing applied. Therefore, simulated annealing (SA) as an optimization algorithm is proposed. The purpose of the experiment is to measure the SA algorithm performance as hyperparameter optimization for a MLP classifier in terms of computer complexity and F1-Score evaluation metric.

**Keywords:** Hyperparameter optimization, machine learning, Multi-Layer Perceptron classifier, optimization techniques, disease identification.

**Introduction:**

Neural networks have been used extensively in tasks such as prediction and classification. Therefore, these models are used with a wide range of data, from power amplifiers behavioral modeling [1] to customer churn prediction [2]. Given the high differences between data sets, is not an easy task to find the best hyperparameters for each problem, the characteristics, pre-processing, and number of features are just a couple of the vast differences that can exist between data sets. That is why [1] and [2] use genetic algorithms to optimize the neural network hyperparameters and explore the solution space as much depth as possible.

Moreover, the use of neural networks to detect, predict and classify diseases is key for modern medicine. [3] implements a genetic algorithm to optimize a deep residual neural network for acute lymphoblastic leukemia classification using microscopy images conclude that the accuracy reached was higher than the one presented in the literature. On the other hand, [4] proposes hyperparameter optimization of an artificial neural network and its structure with a genetic algorithm to breast cancer diagnosis/prediction. The results evidence that uses of this optimization algorithms improve the model performance considerably when compared to literature previous research.

Diabetes affects 17% of the UK population and approximately one million people have undiagnosed type 2 diabetes, 40.000 children have diabetes, and more than 3000 children are diagnosed every year [5]. This is not just a problem in the UK, is a global problem where by 2012 around 346 million people [6]. This reality and the related work inspired the experimentation carried out in this report. Considering the question: Could the optimization algorithm simulated annealing improve considerably the performance and computer complexity of neural network classifier for diabetes prediction/classification?

Finding the best hyperparameters for a machine learning model is fundamental to reach a high prediction/classification performance. That in this case is key to reach the highest accuracy possible to detect and classify diabetes as precise as possible. Because of the variability in data sets a Multi-Layer-Perceptron using diabetes Kaggle dataset is proposed to develop, implement, and evaluate the chosen optimization algorithm performance.

**Methodology:**

Sci-kit learn multi-layer perceptron classifier is proposed as a viable solution to the classification problem given the extended use in classification tasks [7][8]. The optimization algorithm chosen to find the best hyperparameters was simulated annealing. Additionally, the fitness function was F1-score. Moreover, the multi-layer perceptron hyperparameters evolved were solver, activation function, hidden layer sizes, alpha and leaning rate. Figure 1 shows the workflow of the experiment.

A diagram of a medical procedure

Description automatically generated

Figure 1. Experiment workflow.

Different values of cooling rate and maximum iterations were tested. A cooling rate of 0.99, 0.95 and 0.93 were evaluated. On the other hand, the maximum iterations tested were 1000, 500 and 300. Since in the first iterations the probability to accept a lower fitness is higher, and as what is wanted is to explore the solution space as deep as possible, testing the model with different generations may help to reach the highest fitness possible.

**Diabetes data set:**

The diabetes data set used in the experiment is originally from the National Institute of Diabetes and Digestive and Kidney diseases. The main goal with this data is to predict if a patient has diabetes based on diagnostic measurements. The samples are from females at least 21 years old.

The features per patient are:

* Number of times pregnant
* Plasma glucose concentration
* Diastolic blood pressure
* Triceps skin fold thickness
* 2-Hour serum insulin
* Body mass index
* Diabetes pedigree function
* Age

The outcome is 0 for a positive diabetes diagnosis and 1 for a negative diagnosis.

The data set is composed of 768 instances and extracted from Kaggle [9].

The data set does not present direct missing values as NaN. However, it has some values a of 0 in some numeric features that may indicate a missing value. Hence, the potential missing values were filled with the column’s median. As the distribution is homogeneous, filling missing values with median or mean values is logical.

**Results:**

The simulated annealing algorithm offers the possibility to evolve a current random solution, that in this experiment is some MLP classifier hyperparameters. The hyperparameters are mutated given a probability that is getting smaller every iteration. If the new fitness is higher than the current one, the changes are accepted, if not the probability that accept negative changes in fitness is evaluated given the difference in fitness and the time.

The algorithm was executed multiple times. In each scenario the algorithm was tested 10 times to obtain characteristic results and build strong solutions.

Table 1 shows the results of using 1000 iterations and a cooling rate of 0.99. Additionally, some final best solutions were not the best solution that was found in the process. An example of a solution that was greater or equal to 0.82 is shown in the table. This evidence shows that there are multiple solutions and that in some cases a change in the hyperparameters in the last iterations may lead to a lower performance, even if there is a small possibility.

In figure 4 the aforementioned behavior is shown. The final solution has in many cases the best performance. However, this peak performance is reached in the 400-300 iterations.

In general, the algorithm evidence an acceptable performance. Multiple runs evidence that the algorithms have a constant best solution fitness. With no less than 0.79 in F1-Score metric.

The results in \ref{fig:SA1000IterationsCoolingRate095} shown the performance of the algorithm with 500 iterations and a cooling rate of 0,95. The average execution time is lower. However, the average best solution fitness is higher than the previous configuration tested.

**Note:**

* Do two SA optimizations.
  + First optimize the performance of the model
  + Without losing high performance, optimize the computational complexity.

**References:**

1. Hyperparameter Optimization of Two-Hidden-Layer Neural Networks for Power Amplifiers Behavioral Modeling Using Genetic Algorithms
2. Hyperparameter Optimization of Artificial Neural Network in Customer Churn Prediction using Genetic Algorithm
3. Leukemia classification
4. Breast cancer diagnosis/prediction
5. Diabetes UK
6. Diabetes in numbers
7. <https://www.sciencedirect.com/science/article/pii/S0925231200003052?casa_token=kk8Qv7Fp5eUAAAAA:NsLqiHwYM8IIHDQalJklEtPAOG0-kH-zaNU0Ly7_ZK-g9odePxjpYGp2ZLfu0PtI5_5tVNqr>
8. <https://ieeexplore.ieee.org/abstract/document/71429?casa_token=mRwpHPn9-H4AAAAA:FCS81XjwYCyb-H2eRXTS8G8je_ZSWph22ku0RbzEj7wkep-pv1lveSgx_FRLIG9nP3VY_eJU>
9. https://www.kaggle.com/datasets/mathchi/diabetes-data-set