MIOwAD - Sprawozdanie 3 - Genetic Algorithm

Michail Legczylin

June 2024

Contents

1 Introduction

Genetic Algorithm is kind of algorithm that is used to find minimum of a given function.

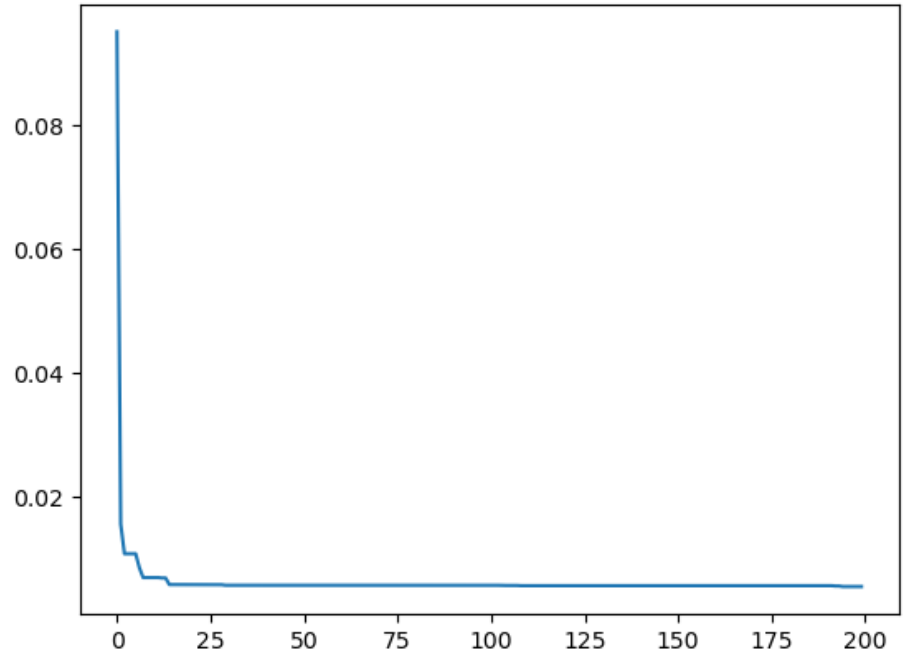
It's main idea is said to be derived from biology and works as follows. The solution, or rather candidate for solution, is nothing more than a vector of input arguments to function that minimizes the latter. The population is list of such candidates. There were two methods implemented that allow for creating new candidates based on existing ones — these are crossover of parents and mutation of specific candidate, just like in nature. Crossover of parents is often swaping respective subsets of parents' values, while mutation is modification by some amout of said value. At the end, as I named them, proximities are evaluated, to determine the output of optimized function of all candidates created by now. After this only those which performed the best a.k.a. had the lowest value, since we minimizing problem, are left in population. This proccess concludes one iteration, which more commonly is known as generation.

Goal is to find suitable number of generations and population size, as well as crossover and mutation probabilities.

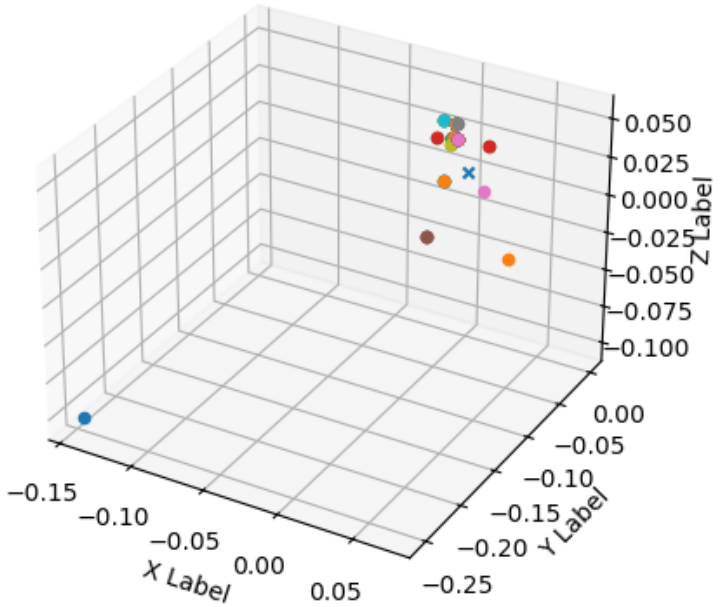
2 AE1

This task was centered on basic iplementation of Genetic Algorithm and its evaluation on simple problems — optimizing $f(x, y, z) = x^2 + y^2 + z^2$ and 5D Rastrigin's function.

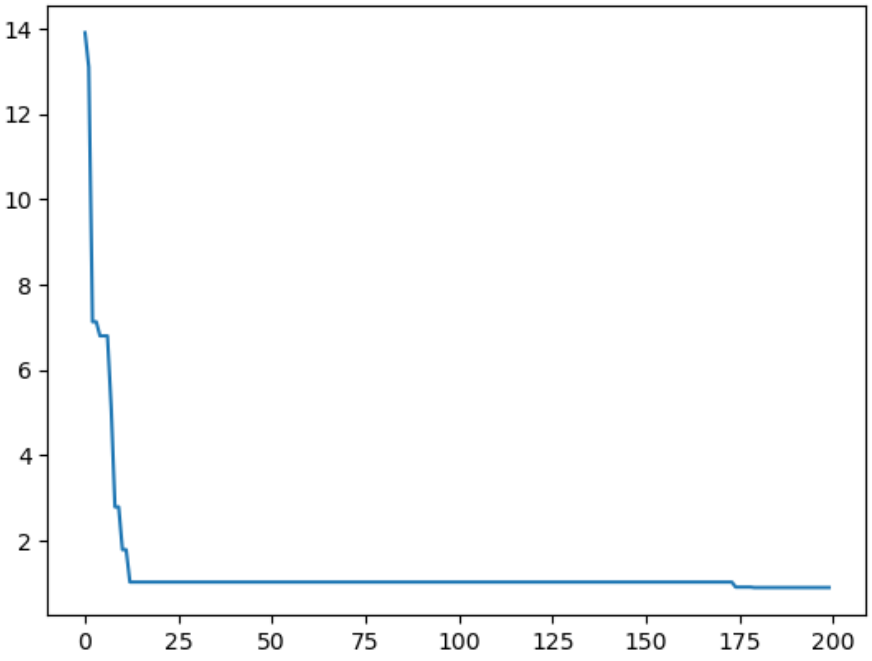
2.1 Sum of squares




```
last proximity=0.005457344497938764 for solution=[ 0.00442964 -0.0506424   0.05360103]
```



2.2 5D Rastrigin




```
last proximity=0.9014607583714884
    for solution=[ 0.00977728  0.03498694 -0.00139709 -0.05702451 -0.00313031]
```

There will be no visualisation, unfortunately, as it is impossible to visualize 5D data on 2D screen.

2.3 Sum-up

In both cases minimum is achieved at vector zero (points [0, 0, 0] and [0, 0, 0, 0, 0] respectively) and as shown, output is close enough to the true one. It can be influenced by problem simplicity, though it is more likely that initial population initialization has bigger part in it, since it comes from Normal(0, 1) distribution having close to 0 values from the beginning.

3 AE2

This task was using Genetic Algorithm solve the cutting-store problem. You are given limited storage and set of objects different shapes with their prescribed values. Goal is to fill the storage with object maximizing their total value (sum).

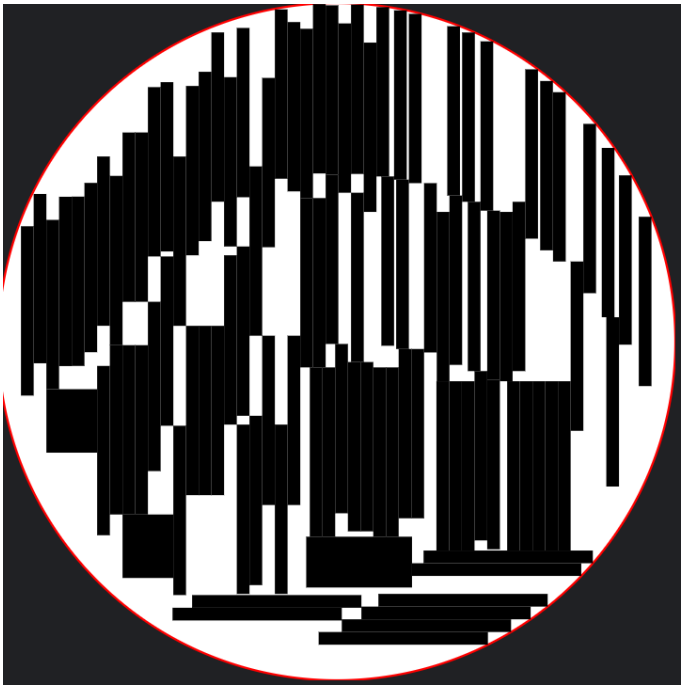
Here we had specified the circle as storage and rectangles as object to fill with.

3.1 Idea

Idea was that in every generation try to insert rectangle in random place in circle. Then goes the use of mutation i.e. moving the said rectangle to the highest rightest possible place in circle. And so we do this for every available rectangle type and for set amount of rectangles per type. Rectangles are also available to be placed rotated by 90 deg.

3.2 Results

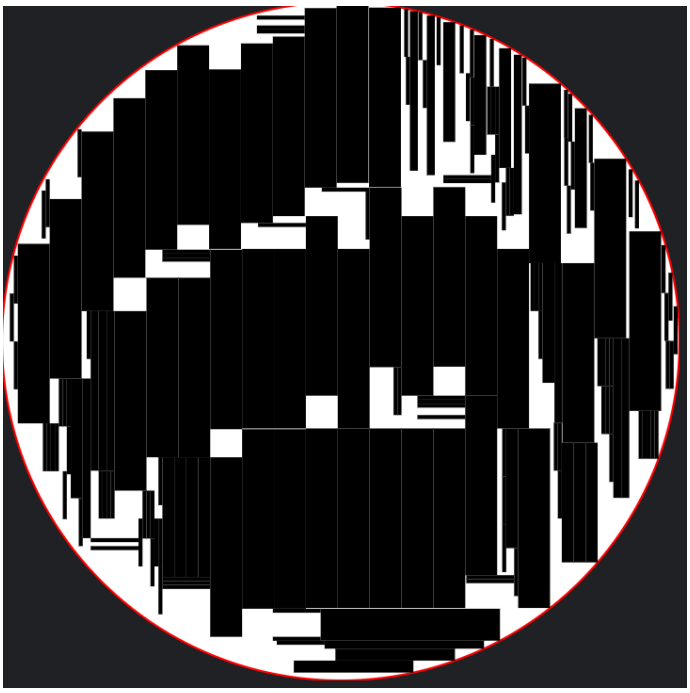
3.2.1 800 radius



```
model evaluated on average in 0h 0m 3s
average score: 36887.2

best score evaluated in 0h 0m 3s
best score: 40180
expected score: 30000
```

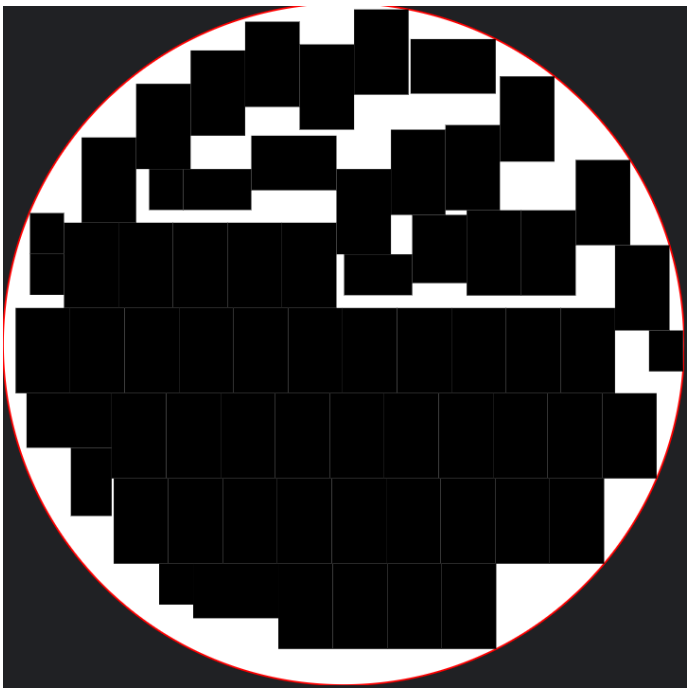
3.2.2 850 radius



```
model evaluated on average in 0h 0m 3s
average score: 457989.0

best score evaluated in 0h 0m 3s
best score: 488810
expected score: None
```

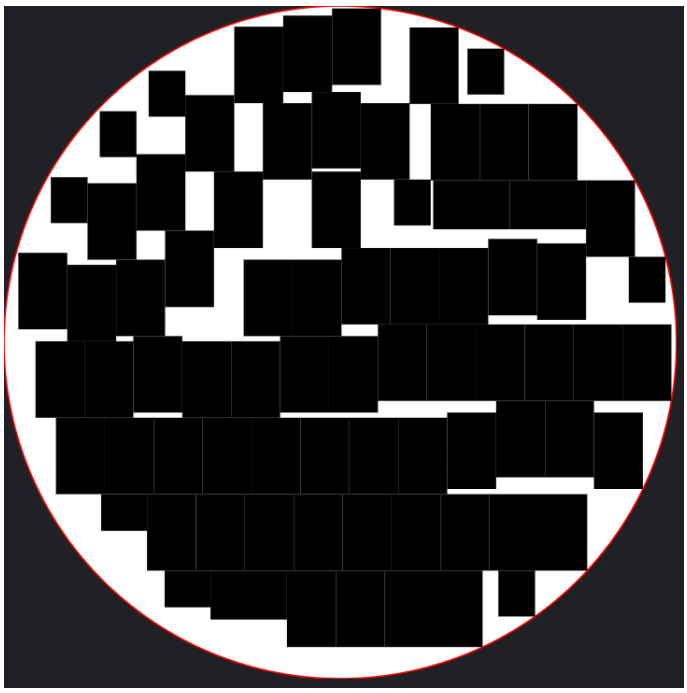
3.2.3 1000 radius



```
model evaluated on average in 0h 0m 2s
average score: 27529.2

best score evaluated in 0h 0m 2s
best score: 29600
expected score: 17500
```

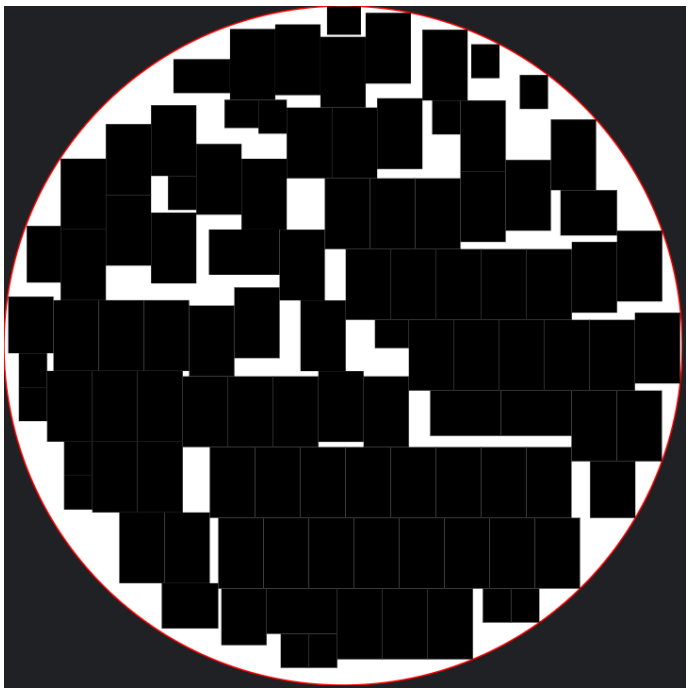
3.2.4 1100 radius



```
model evaluated on average in 0h 0m 2s
average score: 37891.6

best score evaluated in 0h 0m 2s
best score: 41160
expected score: 25000
```

3.2.5 1200 radius



```
model evaluated on average in 0h 0m 2s
average score: 40108.0

best score evaluated in 0h 0m 3s
best score: 42540
expected score: 30000
```

3.3 Sum-up

As seen algorithm evaluated on every dataset more than enough. Though this modification does not have crossover or classic population generation implemented, it still presented itself well.

4 AE3

Task was to use genetic algorithm to optimize NN parameters i.e. weights and biases. I decided to create a dedicated function that takes vector of size of parameters of optimized network, sets these parameters to said network and then returns cost value of y_train and predicted value of such modified network of X_train. So in this way the cost function is minimized, the same logic that NN training process requires.

Parameters are stored as one vector of first weights per layer and then biases per layer:

```
[1 L W, 2 L W, ..., last layer W, 1 L B, 2 L B, ..., last layer B]
```

Respective function to retrieve and set given parameters were implemented into NN.

Also method tuples() of NN returns positions in parameters vector where the next parameter per layer begins (between weights, between biases or between last layer weights and first layer biases).

I also decided that it has no sense in crossovering two parameters vectors entirely, while having chance of putting subset of first layer's weights into second layer's weights. So crossover() function takes optional parameter of format returned by NN.tuples() that ensures, that crossover occures only in range of subset described by tuples().

4.1 iris dataset

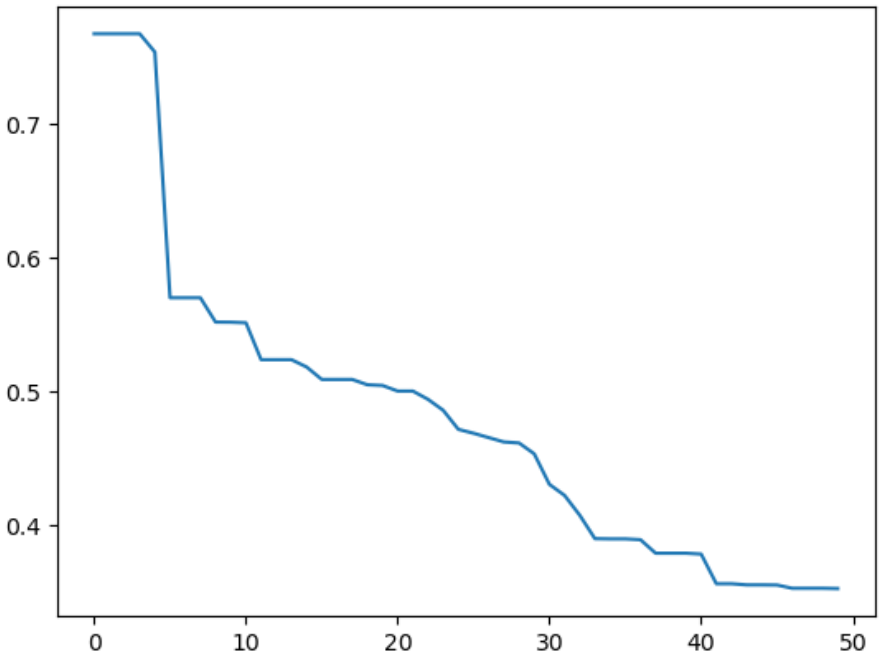


Figure 1: cross-entropy by epochs

```
evaluated in 0h 0m 7s
Cost.Type.CROSS_ENTROPY's last evaluation: 0.3527728918864063
Metric.Type.F_SCORE on test: [1.         0.84615385 0.81818182]
```

Visualization is not possible since X data is 4D.

4.2 multimodal-large

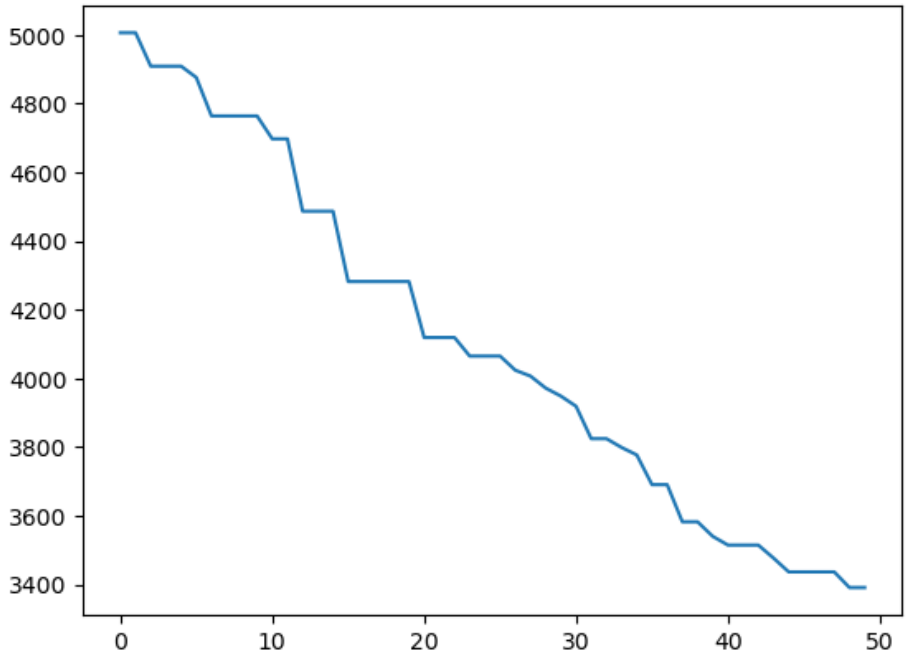


Figure 2: MSE by epochs

```
evaluated in 0h 2m 31s
Cost.Type.MSE's last evaluation: 3391.0661701885297
Metric.Type.MSE on test: 3411.3695596187767
```

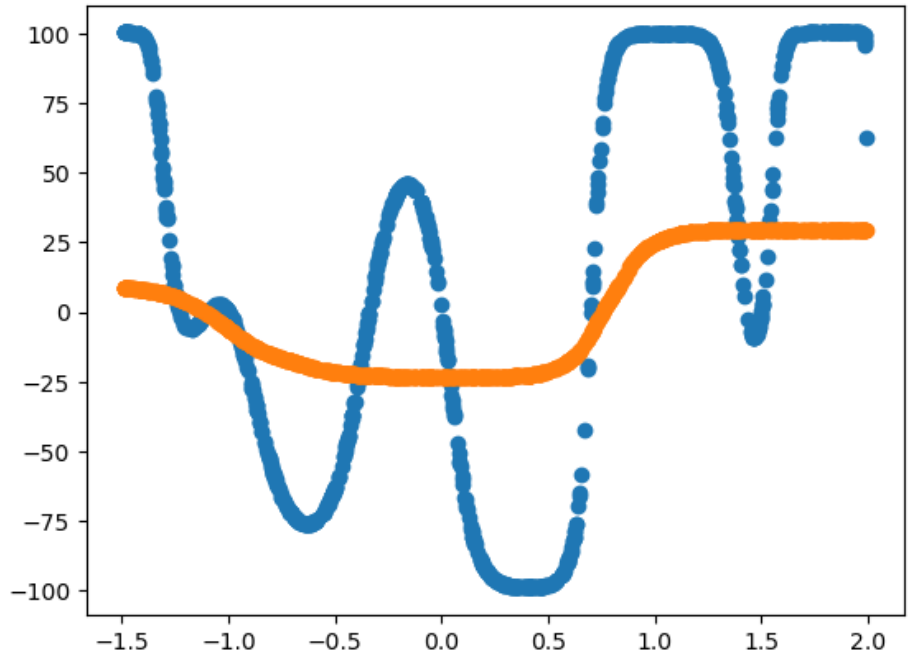


Figure 3: Test true (blue) vs. predicted (orange)

4.3 auto-mpg dataset

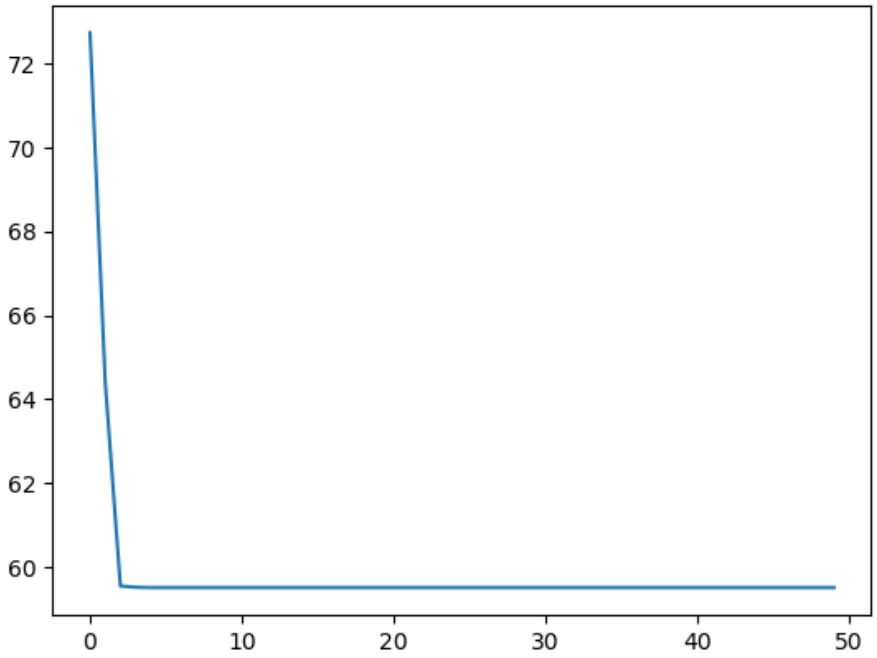


Figure 4: MSE by epochs

```
evaluated in 0h 0m 7s
Cost.Type.MSE's last evaluation: 59.51276348489981
Metric.Type.MSE on test: 63.73829310195137
```

Visualization is not possible since X data is 7D.

4.4 Sum-up

Iris dataset is fairly easy to train, so results are good in terms of metric. Whereas multimodal-large and auto-mpg datasets are regression oriented and their metrics evaluations are far from optimal and by a lot worse than those achieved in regular training.

It can be caused by the fact, that initial population values are generated from Normal(0, 1) distribution, so it means starting weights of NN are in range from -1 to 1 most of the times. Considering that mutation also adds random value from the same distribution, growth of parameters is very slow, while optimal weights and biases contain such values as 40, 50 and even more than 100.

Prehaps it would be wise to implement mutation variance increase over time after no improvements are made.

5 Summary

As shown in this report, genetica algorithms are best used where there is no other method to optimize. For instance looking for function's minimum. But when task has dedicated optimization methods its performance is far better, for example training process of NN.