Feature selection using evolutionary algorithm

Donghao Wang

December 14, 2021

1 Problem description

During training for clustering classification neural network, the data sets could contain redundant features that increase the calculation cost of the model. Feature selection, as the title of this report says, is the process of finding the most relevant inputs for a model.[1] Feature selection techniques helps to automatically remove the redundant features that are irrelevant to or decreasing the performance of the neural network model.

This project is an extension on an assignment I've done in CISC 452 Neural and Genetic Computing. The assignment consists of two parts: building a network for a K-3 Clustering classification using Kohonen Neural network, feature selection on the 4 features given in the input data set using a PCA neural netowrk. This project aims on establishing a genetic evolutionary algorithm that could achieve the same goal as the PCA network in feature selection.

This project will explore further on the possibility of using genetic approach in feature selection(reduction), in comparison with Principal Component Analysis Neural network, which is a neural network approach on this problem, and also brute-force approach.

2 EA Design

2.1 Why Evolutionary Algorithm

Evolutionary Algorithm is suitable for optimization problems that requires exploration. Feature selection is an optimization problem asking for the optimized combination of features to be selected, but not sure about how many features should be selected. Therefore, EA should be a good candidate for solving this problems as the problem requires optimization and exploration on how many features could be selected(reduced).

2.2 Basic Design

Fig 1 below shows a structure of basic Evolutionary Algorithm Design. It consists of five parts: initialization, parent selection,crossover, mutation,survivor selection. If after the survivor selection the stopping criterion is met, the algorithm will output the optimized result, other wise the updated population will be reused for another round of cross over and mutation.

There is one thing the figure didn't show though, during parent selection, fitness evaluation should be applied, in this case, fitness scores will represented by the accuracy in the model training. Details will be explained in the subsection : Fitness Evaluation.

2.3 Initialization

During initialization,initial population should be established for the problem. In this problem, the population should be a list of individuals that represent selection from 4 features given in the input data set. A simple way to represent those individual would be using Binary representation, meaning the selection is represented by binary strings.For example, 1111 means all 4 features are selected, 1010 means the first and third feature are selected, the second and the fourth feature is reduced.

The population size is fixed to 4. For the initial population, 2 will be selected from the 3-feature selection permutation and 2 will be selected from the 2-feature selection permutation. The population is then shuffled.

The purpose of limiting the population size is to reduce the computational cost for each round of fitness evaluation during parent selection and survival selection. 4 is sufficient for the current problem.

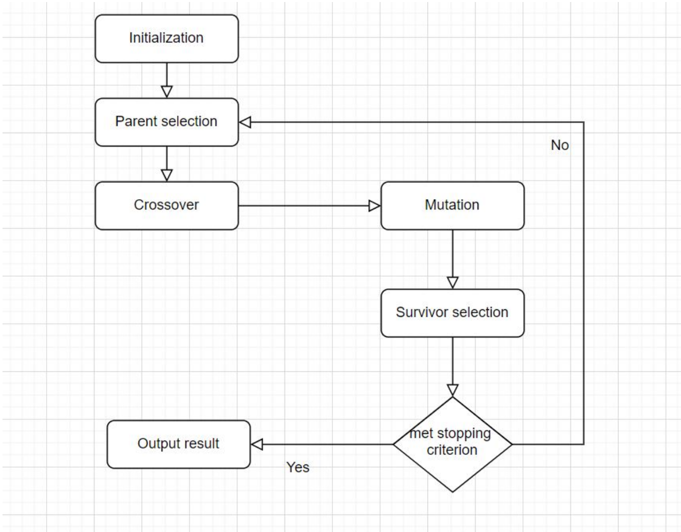

Figure 1: The basic structure of a common evolutionary algorithm

One thing to stress is that only 3-feature and 2-feature selection will be chosen. 1-feature selection is not practical for this clustering classification problem.

2.4 Fitness Evaluation

The fitness for each individual is the average accuracy score it gets in the model training. For evaluation on each individual, a kohonen neural network will be trained on a data set with less features, as the individual indicates, and the final accuracy score will be the fitness value.

For example, for individual 0111, the input data set will have the first feature reduced, then used for training a kohonen model for 10 epochs,which should output a final training accuracy.

2.5 Parent selection

The parent selection in this problem is a probability selection. 50 percent of time the parent selection will be done by multi-pointer selection, another 50 percent of the time it will be done by Best selection. The selection size is 2.

Multi-pointer selection is a variant of round roulette selection. As shown in Figure 2, each line segment represents (the fitness of the individual)/(the total fitness) , each pointer represents a selection. The space between each pointer is 1/ selection-size. That means, the 2 pointer could land on any individual based on the position of the first pointer. Every individual gets some chance to be selected, the probability depends on its fitness.

Best selection is a simplified way to say picking 2 individuals among the population with higher fitness than others.

2.6 Crossover

During crossover, the two parents will swap two random digits to each other. for example, 1100 swap digits at position 2,3 with 1001 will form 2 off-springs, 1000 and 1101. Since mutation is implemented, recombination is not implemented but replaced with extension on mutation.

2.7 Mutation

In 75 percent of chance the individual will have a random digit flopped and 25 percent of chance the individual will have the two digits in random position swapped internally.

After the mutation, if the individual only have one feature selection such as 1000 or 0100, the algorithm will bit-flop one or two of its 0 bits in random position. The chance for it to become a 2-feature or 3-feature selection will be 50 to 50. Same all applies to full selection(1111),the algorithm will bit-flop one or two of its 1 bits in random position. This ensures only 3-feature and 2-feature binary strings will survive.

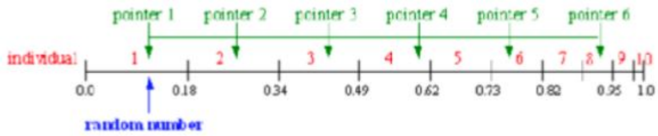

Figure 2: Multi-pointer selection

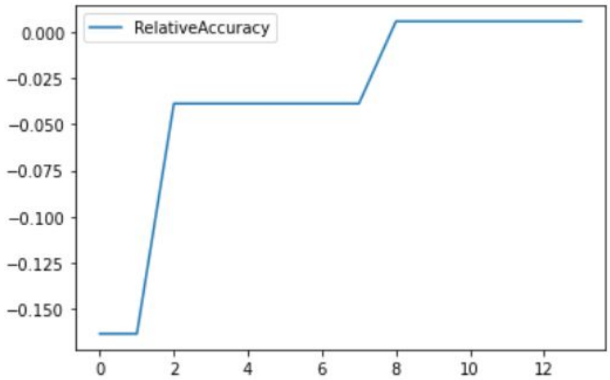

Figure 3: Result visualization: Relative Accuracy vs steps

2.8 survivor selection

All 4 off-springs generated from the parent-selection-crossover-mutation cycle will be considered as the new population. The new population will replace the old population entirely. This is to ensure there is enough exploration on as much possible combination of feature selection as possible.

The new population will then be evaluated and the best individual with highest fitness is recorded. If it is better than the optimum fitness recorded before, the current fitness will be the optimum fitness and this feature selection will be the optimum result for now.

2.9 Stopping criterion

The algorithm will continue to evolve until all below conditions are met

- reduced-feature accuracy is greater than (the accuracy for full features -0.01)
- the optimum result is not replaced for 5 rounds of evolution

These criterion ensures the optimum result is the global optimum of this problem.

The reason for the reduced feature accuracy can be slightly lower than the original accuracy is that the reduced feature in the optimum situation could still contribute slightly to the model training.

3 EA result

In general, the evolutionary algorithm could get the optimum solution in around 3-5 seconds within 15 steps of iteration under google colab GPU environment. In CPU environment, the time cost would increase to 12-15 seconds

Figure 3 shows a run of the algorithm that converges at step8. The y axis is the value of (Reduced-feature Accuracy -Original accuracy of the 4-feature model). The 4-feature accuracy was maintained around 0.88. The final accuracy of reduced feature varies from 0.87 to 0.89,which is normal fluctuation for neural network training.

The final result is 1101 meaning reducing the third feature would not affect the performance of model.

4 Comparison to other implementations

All the results are obtained in GPU environment, time cost will increase in CPU environment.(PCA:1s, Brute-force:over 20 mins)

Method	Time cost(GPU)
PCA network	0.5s
EA	3-5s
Brute-force	over 4m

Table 1: An example table.

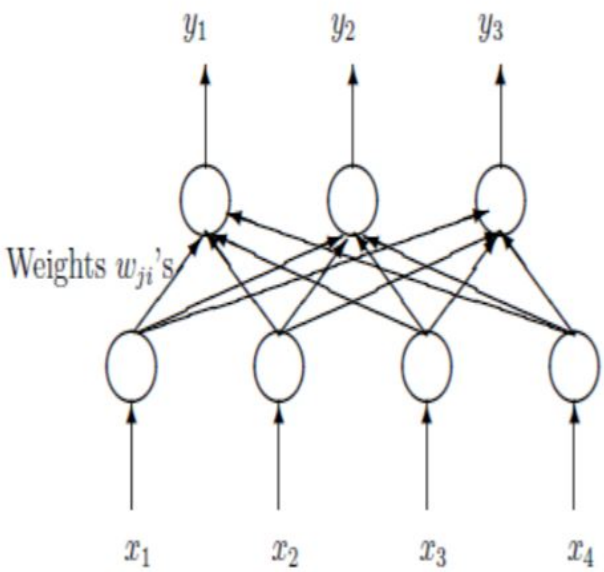

Figure 4: PCA network simple illustration

4.1 PCA neural network

It is an neural network approach in feature reduction. The implementation of the network was a 4 to 3 feature reduction neutral network. As Figure 4 shows, the 4 input nodes corresponds to 4 features in the input data set and 3 output node in corresponds to a 3 feature combination, meaning one feature is reduced.

4.2 Overview

We can see both PCA neural network and genetic evolutionary algorithm perform much better than brute force algorithm. In terms of time cost, PCA neural network is better than EA. However, the PCA neural network in this problem could not adapt to do 4 to 2 feature reduction training,which means in terms of degree of exploration, it is not better than genetic evolutionary Algorithm.

5 Discussion

5.1 PCA NN and EA

The reason for PCA to outperform than EA in this problem is that is dedicated on reducing 4 features to 3 features, which makes it more exploitative. EA algorithm in this implementation is an exploratory approach.

I implemented the PCA algorithm in the code by myself while writing the assignment in 452. The professor specifies there is only one feature needed to be reduced, thus makes PCA NN a good choice for feature reduction. If there is no specification on number of feature to be reduced, EA should be better since it explores more state space in this problem without huge time cost like brute-force.

Therefore one conclusion could be drawn that. If the number of feature reduced is specified, PCA neural network should be a suitable approach; if it is not, EA will perform better.

5.2 Further extension

The code could be reused for other clustering classification problem by tweaking the parameters. One possible extension on this project would be trying to reform the code to adapt to other neural network types, such as the self-organizing-map. Feature reduction is a common problem for all kinds of neural network implementations, by solving this for cluster classification model, the project has proved that improving neural network performance using genetic algorithm techniques is effective.

References

[1] Artelnics Gomez Fernando, Lopez Roberto and Quesada Alberto. Genetic algorithms for feature selection. https://www.neuraldesigner.com/blog/genetic$_algorithms_for_feature_selection, 2021. Accessed : 2021 - 12 - 10$.