Classifying Data from Ionosphere Using an Evolutionary Algorithm

Yong Wei Lim

Australian National University

Commented [S1]: Chinese authors should write their first names in front of their surnames. This ensures that the names appear correctly in the running heads and the author index.

Abstract. This study is an extension from the previous publication which studied the effect of Bimodal Distribution Removal on the ionosphere dataset. This study aims to investigate the effect of Genetic Algorithm on the said dataset which serves to filter data that is considered unfit by the network.
The filtered data is passed through a neural network at 250 epochs and 34 hidden neurons, achieving a mean accuracy of 87.83% on the training set and 88.24% on the testing set. In terms of training accuracy, the network did not perform as well as the one implemented in the source paper published by Vincent G. Sigillito, Simon P. Wing, Larrier V. Hutton and Kile B. Baker.

Keywords: Neural Network, Evolutionary Algorithm, Genetic Algorithm.

1 Introduction

Evolutionary algorithms are biologically inspired as they carry the underlying concept of natural selection, establishing a call for fitness. Provided a fitness function we can deduce the functional domains of a particular dataset, filtering out data considered unfit for training/testing (a.k.a. noise). The reasoning behind the use of Genetic Algorithm on a non-biological dataset is the curiosity of whether biological algorithms can predict noisy data as accurate as other non-biological methods such as Bimodal Distribution Removal and LSTM.

Neural networks are commonly used in signal processing, including but not limited to "audio signal recovery, speech quality enhancement, nonlinear transducer linearization, learning based pseudo-physical sound synthesis" (INFOCOM, University of Rome).

To investigate the effect of the Genetic Algorithm on the chosen dataset, the favorable candidates (data inputs) are passed through functions which manipulate and outputs 'better' child data to feed the neural network for training and testing. The implemented network is trained to classify good and bad radar data, where good data indicates the existence of some structure (particularly free electrons). As mentioned above, the network achieved a mean accuracy of 88.24% when tested on the testing data set with the Genetic Algorithm applied.

The Goose Bay Radar System transmits 17 pairs of pulses into the ionosphere, where each pair consists of a Real and Imaginary reading. (Vince Sigillito), and these readings are fed into the neural network after the application of the Genetic Algorithm.

2 Method

The Genetic Algorithm is initialized with an output size of 34 (matching the dataset input size), a crossover probability of x, a mutation rate of y, generation size and data/population size that is equal to the randomized input dataset (80 training : 20 testing).

The dataset consists of 351 instances, in which classification is either 1 (good) or 0 (bad). For in depth information on the captured data used in this study, read Sigillito, V.G., Wing, S.P., Hutton, L.V., Kile B. Baker (1989). The targeted Doppler velocity is measured by the phase shift of the captured radar signal.

The neural network used consists of 34 input neurons as the first layer, 34 hidden neurons as the second/hidden layer, and two output neurons as the third layer. This approach is commonly known as a feedforward network.

The Genetic Algorithm for fitness measurements involves converting the initial dataset to values between -1 and 1 for consistency. In the study, two input data is used, one of which consists of data post-random-splitting (TRAIN), while the other consists of the full dataset with no splitting (FULL). The converted values are then passed through respective target functions: $\sin(10x)x + \cos(2x)\,x$ and the Sigmoid function $1/(1 + e^{-y})$. Their fitness rating is then computed with a non-zero fitness function in which outputs are fed into a selector function based on individual's fitness rating. The higher the computed fitness for a given datapoint, the better the chance of being selected for the neural network. A crossover function takes the *better* (relatively speaking) data and randomly selects another datapoint and produces a new datapoint. This process is biologically inspired by biological reproduction. The newly produced 'child' then goes through a mutation function which serves as an unbiased randomizer, alike genetic mutation, and ultimately replaces its 'parent' data.

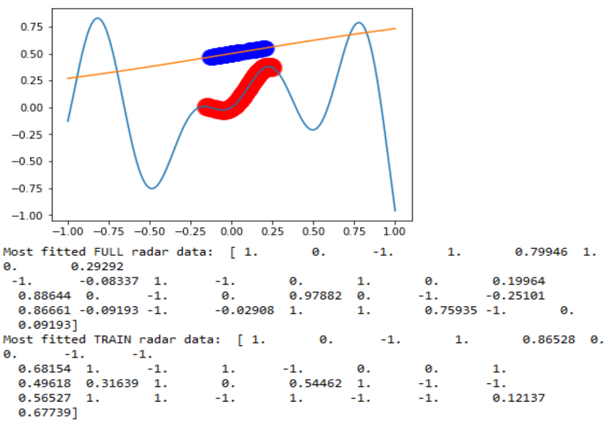
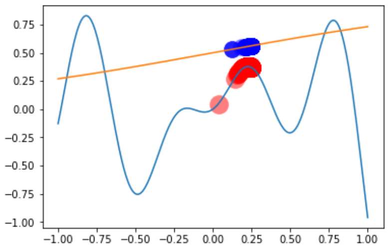


Figure 1: GA applied to FULL (Blue) and TRAIN (Red) datasets. This snapshot is the initial execution of the algorithm.




```
Most fitted FULL radar data:  [ 1.  1. -1.  1.  1.  0.  1.  0.  0.  0.  1.  0.  0.  0.  1.  1.  0.  1.
  0.  1.  0.  0.  1.  1.  0.  1.  1.  1.  0.  1.  0.  0.  0.  1.]
Most fitted TRAIN radar data:  [ 1.       1.       -1.       1.       1.       1.       1.       1.
  0.80365  1.       1.       1.       1.       1.       0.       1.       0.
 -0.16555  0.       1.       1.       0.       1.       1.       0.       0.
  1.      -1.       0.       1.       0.       0.       0.       1.      ]
```

Figure 2: GA upon completion, the fitness algorithm has effectively filtered out the ‘weaker’ candidates (Manipulation 1 – Table 1)

During the GA phase, mutation rates and crossover rates were slightly manipulated to investigate if there exists a significant change as provided in the following table:

Manipulation no.	MUTATION_RATE	CROSS_RATE
1	0.008	0.7
2	0.004	0.7
3	0.008	0.8
4	0.004	0.8

Table 1 consists of the manipulated variables during the application of GA.

3 Results and Discussion

Based on the results drawn from the neural network, the error distribution of the training set has significantly improved when the dataset is processed via GA. When compared to the results without GA processing, the error peak came out significantly better. However, when compared between the manipulated rates for GA, a pattern can be observed, as shown in Figure 4, Figure 6 and Figure 7.

The results shown that there is a negligible difference when comparing between the same crossover rate, but there is a notable difference of error peak and fitness when compared between 0.7 and 0.8 crossover rates. Manipulation 3 and 4 produced almost identical fitness diagram, though a slightly smaller error distribution was observed with a smaller mutation. Likewise, manipulation 1 and 2 produced the same fitness diagram. However, there was a significant difference between the error distribution charts, as there was hardly any error peak observed in manipulation 2. Furthermore, a slightly decrease in overall network accuracy was observed when the mutation rate is set to 0.08 (85.53%).

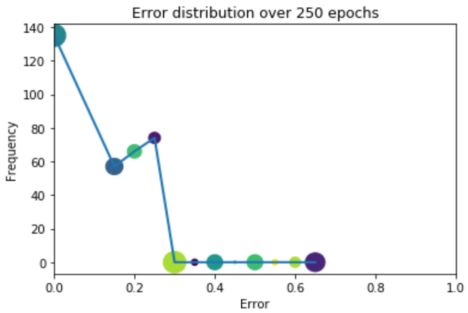


Figure 3: Post-GA processed dataset fed into a BMD neural network. (Manipulation 1 – Table 1)

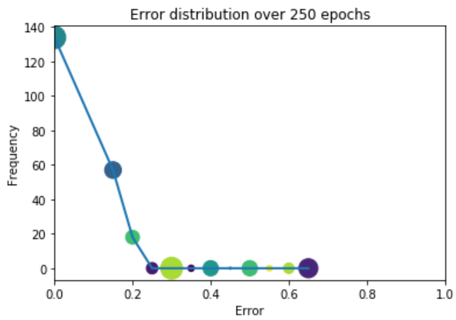


Figure 4: Post-GA processed dataset fed into a BMD neural network. (Manipulation 2 – Table 1)

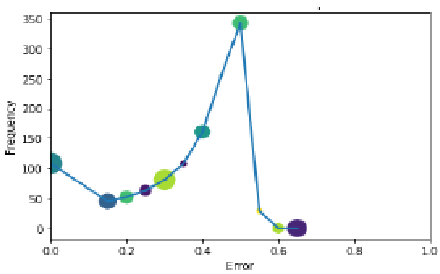


Figure 5: BMD approach with no GA processing.

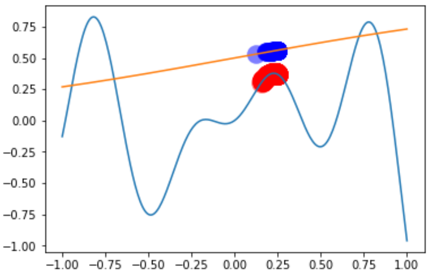


Figure 6: Post-GA processed dataset fed into a BMD neural network. (Manipulation 3 – Table 1)

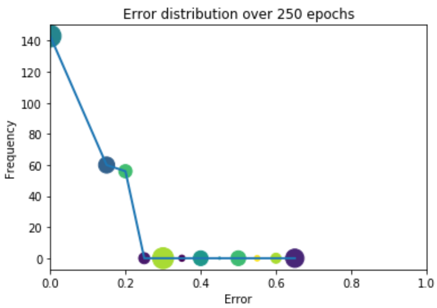


Figure 7: Post-GA processed dataset fed into a BMD neural network. (Manipulation 4 – Table 1)

Despite observing significantly smaller error peaks and better overall distribution of errors with the application of GA to the chosen dataset, the network's accuracy performed slightly worse than without GA. Two main assumptions can be made from this observation:

1. The neural network's results from the initial publication for Bimodal Distribution Removal was poorly implemented which led to better accuracy due to factors such as biased and overfitting.
2. The biologically-inspired GA approach does not work well on non-biological datasets, therefore producing inconsistencies, causing underfitting due to the severe consequence of 'natural selection'.

When compared to the source paper by Sigillito V.G., the accuracy of the said neural network did not perform as well as their implementation which achieved a whopping 100% accuracy in training, and 98% in testing.

Despite the slightly underperforming neural network, the outcome of this study strongly suggests that the application of biologically-inspired algorithms can be used on non-biological studies to some extent, as an accuracy of over 85% is still pretty decent. Moreover, hidden nodes are once again observed to be beneficial for the neural network in terms of training accuracy over time, which is a shared observation from the initial study.

4 Conclusion and Future Work

In conclusion, Evolutionary Algorithms can be implemented alongside other popular methods (LSTM, BDR, etc) used in neural network optimizations for data analysis. This study suggests that an optimized neural network can consistently process radar signals for scientists and military operations (naval and air force research). In the future, I am keen on exploring different Evolutionary Algorithms as well as Deep Learning Algorithms for language analysis, particularly for Japanese text. I am very interested in the capabilities of neural networks in recognizing and pronouncing Kanji(漢字), as they share the same writing with Traditional Chinese characters but not the pronounciations.

References

1. Malkoff, D.B., "The Application of Artificial Intelligence to the Handling of RealTime Sensor Based Fault Detection and Diagnosis," Proceedings of the Eighth Ship Control Systems Symposium, Volume 3, Ministry of Defence, The Hague, pp 264- 276. Also presented at the Hague, Netherlands. Accessed April 26 2018.
2. Johns Hopkins University Ionosphere database. Vince Sigillito.
3. What is an Evolutionary Algorithm? https://www.cs.vu.nl/~gusz/ecbook/Eiben-Smith-Intro2EC-Ch2.pdf
4. Audio Signal Processing by Neural Networks. Aurelio Uncini, Dipartimento INFOCOM, University of Rome "La Sapienza", Via Eudossiana 18, 00184, Rome Italy, https://pdfs.semanticscholar.org/2a79/74d9ad5784ffeb32c490541868de858192a7.pdf