# Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition

A.R. Burton and T. Vladimirova
Dept. of Electronic and Electrical Engineering
University of Surrey, Guildford, GU2 5XH, England
e-mail A.Burton@ee.surrey.ac.uk

## Abstract

The aim of this paper is to propose a means by which neural network fitness evaluation can be applied to a genetic algorithm (GA), and an application of this system to musical rhythm composition. An Adaptive Resonance Theory (ART) neural network is trained using binary information representing classification patterns. By comparing new genetically derived individuals to clustered data, a measure of fitness of the new patterns is determined; the patterns of higher fitness values then being used in successive generations to further improve the overall population fitness. A proposed application for this system is described - a genetic composer that utilises clustered representations of rhythm styles to interactively generate rhythm patterns to the user's general stylistic requirements.

## 1 Introduction

Genetic Algorithms (GA's) are computational methods which use procedures based upon the laws of natural selection to increase the worth of a population as a whole by combining those individuals with high worth [4]. GA's have been seen to perform better than other optimisation processes such as random walks or enumerative calculus-based methods [4]. GA's combine the probabilistic nature of random search procedures with the guided nature of deterministic search methods to direct the search towards areas of the search space of higher worth, and utilise encodings of the data through which it is searching, rather than the data itself. This implies that GA's are blind to the application [4], and are thus applicable to a great number of subject areas. One particular subject area is that of musical composition and synthesis, for example [1, 5, 6]. Given an encoded representation of musical phrase or sound, a GA can generate further examples based upon aspects of the original fragment and the means by which fitness is evaluated.

The fitness functions used by GA's in musical applications vary according to the desired results, for example adherance to formal rules [6], or user input to guide population evolution [1]. Both of these methods have produced successful results, but at certain costs. Rule-based fitness regimes tend to limit the scope of compositions, often breaking the rules of composition results in more musically interesting pieces than adhering them. Algorithms that depend on the user to assign fitness to every result are time consuming and sequentially context dependent.

The aim of this paper is to investigate the way in which neural networks can be used as a fitness evaluation procedure, and to propose a means by which neural network fitness evaluation can be applied to a GA based interactive intelligent drum machine.

## 2 Neural Networks and Genetic Algorithm Fitness Evaluation

Neural networks have been applied as fitness evaluators in genetic and evolutive applications, to a varying degree of success [2, 3, 8]. Neural fitness evaluators are used in applications where the knowledge of how fitness should be assigned to any individual is heuristic, uncertain, or in applications where there are many degrees of freedom in the information encoded in each individual. Neural network fitness evaluators require time consuming training, which assumes that knowledge of how the input set will map to a certain output is known in advance. Also, once a training set is learnt, there is no scope for new data to change the way the system evaluates fitness. This may prevent some new genetically produced individuals from being rejected as low fitness, when in actual fact they may be of high fitness, but due to the training method employed, can not be identified as such.

These individuals of potential high fitness may be of use in classifying other similar individuals - of a type that may not have been included in the training set.

An Adaptive Resonance Theory (ART) [7] network can be used as a fitness evaluator to solve these problems. ART neural networks are self-organising networks, utilising unsupervised learning and clustering algorithms to be able to recognise patterns. One advantage of unsupervised learning is that no *a priori* knowledge is assumed regarding the classification of any pattern - the network determines how a specific input maps to an output. There is also no limit to the number of patterns that can be classified - if a pattern is detected that can not be clustered with other patterns, a new cluster is created.

ART systems control the similarity of patterns and the classification clustering using vigilance. This is a threshold which determines whether or not a certain pattern is similar to already classified patterns so that a new cluster can be created if required. One of the most important features of Adaptive Resonance networks is their ability to resolve the stability-plasticity dilemma. This allows the network to record new information as it arrives, yet remain stable with regard to information previously processed - thus data not in the initial training set does not cause loss or corruption of either the new or old training data.

Three attributes of ART networks - unsupervised learning, unlimited pattern classification groupings, and adaptability of vigilance - make this network architecture suitable to genetic algorithm fitness evaluation.

## 3 The ART Neural Network as a Fitness Evaluator

Prior to running the GA with the ART fitness evaluator, the neural network is trained using example training data. The training data used for the experiments to verify the ART fitness evaluator were taken from binary array representations of letters of the alphabet. For example the letter 'T' was represented as shown in Figure 1.

```
111110
001000
001000
001000
001000
000000
```

Figure 1: Example of Binary Test Array

Similar representations were used for four more letters P, M, E and Y. Each training pattern was transformed from the array representation as above to a binary string by joining the beginning of one row to the end of the previous row - hence the above array becomes:

111110001000001000001000001000000000.

This is an example of the individuals used to create a cluster that represents one classification. Other examples of training data for a classification differed by one or two bits. Experiments employing a different bit length used a different resolution to represent the same symbols.

The ART fitness evaluator was implemented using a GA based upon Goldberg's Simple Genetic Algorithm (SGA) [4]. Single point crossover and single bit mutation with a user-specified probability was employed. The ART network fitness evaluator operates as follows:

1. Each individual is presented as an input to the ART network.
2. The network determines the 'winning' cluster - the maximum value of the matrix product of the input vector (the chromosome) and the bottom-up connection weights of the ART network.
3. The vigilance test is carried out to determine the degree of match between the individual and the cluster.
4. If the vigilance test passes, the individual is added to the cluster. If the vigilance test fails, then the remaining clusters are tested in a similar way.
5. If no existing cluster represents the individual closely enough, a new cluster is added.
6. Fitness is assigned as the degree of similarity to individuals already represented by the cluster - a high fitness is assigned to individuals which show a closer match. Individuals which form a new cluster are assigned maximum fitness.

The use of an ART neural network as a GA fitness evaluator has been validated by the experimental work carried out so far. The effects of cluster formation has been investigated dependent on network vigilance, and the GA parameters population size, mutation probability, and chromosome length. The number of input units used for the ART network is equal to the chromosome length of any individual.

For each test condition, the genetic algorithm was run several times under the same conditions. This is due to the fact that as the genetic process contains a certain degree of probabilistic operations, then one single result will not be representative of the behaviour of the system. However, if the same conditions are used over a number of runs, then it is possible to average the results to determine trends in the recorded data.

Cluster creation over the first ten iterations of a genetic algorithm as a function of population size, mutation probability and vigilance parameter was investigated. Figure 2 shows one set of results of these experiments - the cluster creation behaviour for a population size of sixteen individuals.

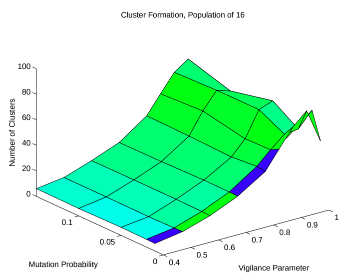


Figure 2: Example of Cluster Creation Results

It can be seen from these results that for higher values of vigilance, the number of clusters that are created by the fitness evaluator is very large compared to the initial number of clusters. This is due to the fact that high vigilance parameters require that a very close, or exact, match be found between individuals being evaluated for fitness and those individuals represented by the cluster. Such a trend was observed for all values of population size, although smaller populations would display a slower rate of cluster creation than higher population sizes. Additionally, the rate and amount of clusters created is independent of mutation probability.

The average and maximum fitness values displayed by a number of individuals in a population of constant size was also investigated. The average fitness is the mean fitness of the population at any given iteration of the algorithm, and the maximum fitness is the greatest fitness observed of any individual in any iteration. These values were observed over the first ten iterations of the genetic algorithm, as a function of chromosome length, mutation probability and vigilance parameter. The average and maximum fitness values for a population of sixteen-bit individuals, with a mutation probability of 5% are shown in Figures 3 and 4.

Use of a high vigilance parameter results in a higher average fitness across the population at the beginning

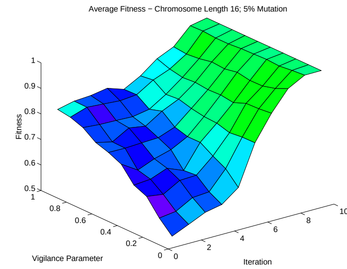


Figure 3: Example of Average Fitness Results

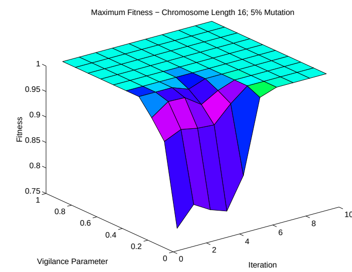


Figure 4: Example of Maximum Fitness Results

of the ten iterations. However, the fact that a high vigilance parameter also results in a very large number of clusters. Maximum fitness values are reached faster given a high vigilance parameter. Again this is the result of a high cluster creation rate for these conditions. A solution to the problem of this attribute of lies in better control of cluster creation, and is the subject of current work. Another attribute of this system observed from the results is that the average and maximum fitness surfaces obtained were independent of the length of chromosome used in the representation.

# 4 A Genetic Algorithm Based Interactive Drum Machine

The specific purpose of the proposed system is to act as an interactive drum machine for a musician or composer. Given initial parameters such as rhythm type, tempo, time signature, the GA constructs a population of individuals which each represents a rhythm pattern. The representation used for these input patterns will consist of a binary array of data, one dimension representing beats, or discrete time steps, the other dimension representing individual voices. For example the rhythm pattern shown in Figure 5 would be represented by the three row array shown in Figure 6.

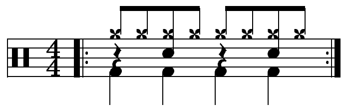

Figure 5: Example of Rhythm Pattern

11111111
00100010
10101010

Figure 6: Binary Rhythm Representation Array

The fitness evaluation network uses information obtained from existing MIDI (Musical Instrument Digital Interface) files. This process creates clusters of stylistically similar rhythm patterns.

Depending on the similarity measure of the genetically produced rhythm patterns to the appropriate rhythm pattern cluster, a certain fitness value can be assigned to that pattern. If the rhythm pattern fails the vigilance test, then a new cluster can be created to represent this pattern.

Patterns with higher fitness values can be used as the starting point for the next generation of individuals, after subjecting the population to typical GA breeding processes. Further iteration of the algorithm will result in populations containing individuals that will approach the requirements of the user. By experimentation, the optimum selection/reproduction and genetic operation schemes will be determined, such that the algorithm can reach the desired result efficiently. More complex crossover operators will be implemented, to allow a more diverse population be created for subsequent generations. Methods including two point crossover, multiple point crossover and crossover masking [4] will be investigated. Mutation will be, as before, single bit, random, bit-inversion mutation. Additionally, the different means of selection and cluster addition will be implemented depending on the results of the tests carried out at the time.

## 5 Concluding Remarks

Results so far suggest that the ART network clusters binary patterns of genetically produced patterns, and can assign fitness values to these patterns as a function of the degree of similarity between an individual pattern and the cluster to which it is closest.

An ART network, when used as a GA fitness evaluator, possesses distinct advantages over other neural methods of fitness evaluation - the ability to add new classifications when existing clusters do not sufficiently represent an individual. This feature will be applied to an evolutive musical composition system. However, high vigilance parameters have been seen to produce large numbers of clusters. Work in progress will introduce a system to control cluster creation.

Application of this hybrid system to the generation of rhythm pattern data for use by an interactive drum machine is currently being investigated. The main areas of focus involved are the size and type of input representation used, weightings of certain members of the input patterns to signify beat strength and variation of the vigilance parameter to determine how close a match between cluster patterns and input patterns is required.

## References


[1] J.A. Biles. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, 1994.

[2] J.A. Biles, P.G. Anderson, and L.W. Loggi. Neural network fitness functions for a musical IGA. Technical report, Rochester Institute of Technology, 1996. http://www.it.rit.edu/ jab/SOCO96/SOCO.html.

[3] C.H. Dagli and S. Sittisathanchai. Genetic neuro-scheduler for job shop scheduling. *Computers and Industrial Engineering*, 25(1-4):267–270, 1993.

[4] D.E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison-Wesley, 1989.

[5] D. Horowitz. Generating rhythms with genetic algorithms. In *Proceedings of the 1994 International Computer Music Conference*, pages 142–143, 1994.

[6] R.A. McIntyre. Bach in a box: The evolution of four-part baroque harmony using the genetic algorithm. In *Proceedings of the IEEE Conference on Evolutionary Computation*, pages 852–857, 1994.

[7] A. Nigrin. *Neural Networks for Pattern Recognition.* MIT Press, 1993.

[8] G. Schneider, J. Schuchhardt, and P. Wrede. Amino acid sequence analysis and design by artificial neural network and simulated molecular evolution - an evaluation. *Endocytobiosis and Cell Research*, 11(1):1–18, 1995.