Algerian Journal of Research and Technology
https://doi.org/10.58681/ajrt.21050101
A.J.R.T Volume 5 (N°1) (2021) 1 - 11

A Hybrid Evolutionary Neural Networks Training applied to Phonetic Classification

R. Tlemsani, N. Neggaz

Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, USTO-MB, BP 1505, EL M'naouer, 31000 Oran-Algérie - Laboratoire Signal Image PArole (SIMPA) Département d'informatique Faculté des Mathématiques et Informatique,

** Corresponding author. tlemsani.redouane@gmail.com; nabil.neggaz@univ-usto.dz*

Received. March 08, 2020. Accepted. December 04, 2020. Published. March 22, 2021.

DOI: https://doi.org/10.58681/ajrt.21050101

Abstract. In machine learning, the most challenge is the learning process of artificial neural network that aims to determine the optima set of weights and biases. In general, gradient descent methods are the most employed as training algorithm. However, this category of algorithms converges to local optima with slow convergence. For this reason, a great number of biological and swarm inspiration are developed in the literature for avoiding the shortcomings of gradient descent algorithms. Basically, genetic algorithm (GA) is inspired from Darwin theory and more recently, evolution strategies (ES) is developed. This paper proposes a new combination between multi-layer perceptron MLP and evolutionary algorithms (EA). Two algorithms of EA are exploited known as GA and ES for training strategy by optimizing the weights and biases. This improvement leads to accelerate the speed convergency and minimize the risk of getting by local optima. The proposed methods treat the continuous speech recognition field by assessing exactly a sub-corpus of the TIMIT datasets. The experimental results shown that the ES-MLP achieves high performance compared to other algorithms including GA-MLP and Back -propagation gradient (BP) in terms of overall classification rate with 58.81%.

Keywords. Evolutionary Algorithms, Evolution Strategies, Genetic Algorithm, Neural Networks, Speech Recognition.

INTRODUCTION

Biological evolution has resulted in extremely complex living systems. It is the result of a gradual and continuous alteration of living beings over generations and takes place in two steps: selection and reproduction. Natural selection is the central mechanism that operates at the population level, selecting the individuals best suited to their environment. Reproduction

involves a memory: heredity, in the genes form. This hereditary material undergoes, at the molecular level, constant changes by mutations and recombination, resulting in a great diversity (Tang et al., 2014).
These principles, presented for the first time by Darwin, inspired much later computer researchers. They gave birth to a class of algorithms grouped under the generic name Evolutionary Algorithms (EA) (Spalanzani, 1999). They are a class of probabilistic search optimization algorithms based on the natural evolution model. They model a population of individuals by points in a space.
An individual is coded in a genotype (chromosome) composed of genes that correspond to the values of the parameters of the problem to be treated. The genotype of the individual corresponds to a potential solution to the problem posed with the goal of finding optimal solution using EA. EA are derived directly from nature's ability to adapt to the environment by evolving through selection and reproduction. Neural networks are also a simplified way of simulating the abilities of living organisms to adapt to their environment through learning. Just because nature works that way, and successfully, it has been a source of inspiration for many of the work on neuron network (NN) hybridization with evolutionary algorithms, hoping that this combination can solve problems. In a more efficient way than the two methods taken independently. In this category, we find the work of (Jalal et al., 2019) which aims to learn MLP using butterfly optimization algorithm for data classification. The experimental results shown that the proposed method BOAMLP outperformed other optimizer as GOAMLP, FAMLP, PSOMLP, DEMLP, GAMLP on Parkinson datasets with 88.21% in terms of accuracy.
More recently, Bansal et al., 2020 proposed a new hybrid swarm model called simple - matching grasshopper new cat swarm optimization algorithm (SM-GNCSOA) for selecting the optimal MLP and relevant features from the datasets. The proposed method gives a good behaviour in terms of accuracy and the size of selected features on most datasets. In the same context, proposed a novel swarm optimizer called ant lion optimizer for tuning the weights and bias of multi-layer perceptron (Heidari et al., 2020).
For speech recognition based neuro-evolutionary techniques, proposed genetic algorithm for learning the structure of deep learning (Anwar and Ali, 2019). The obtained results are successfully tested on google speech commands datasets with a superior accuracy of 91.4%
Speech is the main means of communication in every human society. Its appearance can be considered as concomitant with the appearance of the tools, the man then needing to reason and to communicate to shape them.
Speech is one of the first modes of communication of the simplest man but also the most sophisticated. The mastery of this mode of communication has allowed the emergence of automatic systems of synthesis, compression or even speech recognition. For several decades, many teams have been working on this last point: speech recognition.
The special importance of speech processing is explained by the privileged position of speech as a vector of information in our human society (Miclet and Haton, 1984).
The extraordinary singularity of this science is due to the fascinating role played by the human brain both in the production and in the comprehension of the word, and to the extent of the functions, it unconsciously implements in order to achieve it. Virtually instantaneous way. Constantly control through the motor cortex, speech is produced by the vocal tract. The study of phonation mechanisms makes it possible to determine, to a certain extent, what is speech and what is not. Like most biological signals, speech is a non-stationary signal, which explains the complexity of its treatment (study of relevant parameters, transmission ...), which has led researchers to improve the techniques applied to this signal.
An Automatic Speech Recognition System can be represented as follows: (Haton and Haton, 1989)

- Acquisition and pretreatment: The acquisition corresponds to the extraction of digital data. Signal sampling allows you to switch from analog data to digital data. Pre-processing reduces the quality of speech signal information and facilitates classification.
- Encoding: Encoding reduces the amount of data to be processed, transmitted, etc. It is important during this step to keep the relevant information about the speech signals processed. In our case, we use the MFCC coding.
- Training: Where a model or a reference of a form is built from several occurrences of this form. A classification method is used to memorize the shapes. The shapes are represented by vector sequences corresponding to phonetic segments.
- Recognition: Where one declares recognized the form most likely or closest to that presented in the sense of a distance. It corresponds to the identification phase of unknown forms compared to the knowledge stored by one of the classification methods.

The performance of MLPs depend on the learning technique that we use to train the model. In addition, the unique random initialization used in learning method as back propagation increases the possibility to converge to local optima (LO). This paper aims to integrate evolutionary algorithm in training process of MLP-based models to enhance the chance of LO avoidance and mitigate the stagnation problems.

The paper is organized as follow. The subsequent section describes the basic concept of some type of EA including genetic algorithm and evolution strategies. Section 3 explains in details the model of training MLP using EA. The results and discussion are shown in section 5. Finally, the conclusion and future direction are drawn in section 6.

EVOLUTIONARY ALGORITHMS

Biological evolution has resulted in extremely complex living systems. It is the result of a gradual and continuous alteration of living beings over generations and takes place in two stages: selection and reproduction.

Natural selection is the central mechanism that operates at the population level, selecting the individuals best suited to their environment. Reproduction involves a memory: heredity, in the form of genes. This hereditary material undergoes, at the molecular level, constant changes by mutations and recombination, resulting in a great diversity.

These principles, presented for the first time by Darwin, inspired much later computer researchers. They gave birth to a class of algorithms grouped under the generic name Evolutionary Algorithms (or Evolutionary Algorithms (EA)).

There are two main axes in these methods: Genetic Algorithms (GA) and Evolutionary Strategies (ES). These methods are implemented according to the principle of selection and replacement of individuals in the population:

Genetic Algorithms

Genetic Algorithms are systems that rely on Darwin's principles of selection and gene combination methods introduced by Mendel to address optimization problems.

They can surpass other classical methods with their robustness and are fundamentally different according to four main axes (Goldberg, 1994):

1. AGs use parameter coding, not the parameters themselves.
2. They work on a population of points, instead of a single point.
3. They only use the values of the function, not its derivative, or other auxiliary knowledge.
4. They use probabilistic and non-deterministic transition rules.

Evolution occurs on chromosomes that each represent individuals in a population. The natural selection process ensures that the most suitable individuals reproduce more often and contribute more to future populations (Pal and Wang, 2017).
During reproduction, the information contained in the individuals of the parents is combined and mixed to produce the individuals of the children. The crossing result can in turn be modified by random disturbances.
Evolution strategies
ES, is applied in a content domain of the search space Ω that is included in Rn. In ES, the individual contains not only his position in the search space but also some information about his mutation.
In the general case of structure, by adding a random vector that follows a normal to zero mean distribution, this information is incorporated into each individual (Hellwig and Beyer, 2018)
The parameters of the mutation space S are composed of nσ number of standard deviations σ and nα number of covariance of the rotation angles α so the individual X is represented in a space Ω * S by:
$X= ((x_1, x_2,\ldots, x_n)\ ,\ (\sigma_1, \sigma_2, .,\sigma_n)\ ,\ (\alpha_1, \alpha_2, .., \alpha_n))$ (1)
There are several types of mutations in the ES according to the values of nσ and nα:
$n_\sigma= n;\ n_\alpha = n(n-1)/2$ (2)
The correlated mutations take place as follows (Hansen and Ostermeier, 2001):

$$\begin{cases} \sigma_i' = \sigma_i \,.\, \exp(\tau', N(0,1) + \tau.\, N_i(0,1)) \\ \alpha_j' = \alpha_j + \beta \,.\, N_i(0,1) \\ x' = x + N(0,C') \end{cases} \quad (3)$$

Where:

$$\tau = \frac{1}{\sqrt{2\sqrt{n}}}\ ;\ \tau' = \frac{1}{\sqrt{2n}} \quad (4)$$

The β value is equal to 0.0873 (corresponding to 5° of rotation). The matrix C' is the inverse of the covariance matrix and its elements can be calculated using the parameters αij (the vector α is transformed into a matrix):

$$C_{ij} = \begin{cases} \sigma_i^2 & \text{if } i = j \\ \frac{1}{2}(\sigma_i^2 - \sigma_j^2)\tan(2\alpha_{ij}) & \text{Else} \end{cases} \quad (5)$$

The vector Z_c = N (0,C') is created first by obtaining a vector:
Z_u = N (0, σ) (σ represents a diagonal matrix with the σi), and then we use the rotation matrix R_{ij}:

$$Z_c = \prod_{i=1}^{n-1}\prod_{j=j+1}^{n} R(\alpha_{ij}) Z_u \quad (6)$$

The rotation matrix:
R(αij) = [rij] i, j = 1 ... n are the unit matrices modified by:
$r_{ii} = \cos(\alpha_{ij})$ et $r_{ij} = r_{ji} = -\sin(\alpha_{ij})$ (7)
The modification of the covariance values retains a positive definite matrix (Rudolph, 2001).
In ES, selection (μ, λ) represents the next population that will be made up of μ best offspring from the offspring of the λ population.
The main difference between ES and GA can be illustrated in the encoding type of agents' i.e ES used real encoding while GA employed binary encoding. In addition, the main operator in GA is crossover while in ES the main operator is mutation. Furthermore, the mutation in ES is more complicate than GA because in ES, the mutation process is realized by adding a normal random vector while in GA the mutation consists to change the value of the bit (0 to 1 or 1 to

0). Finally, the selection operator used in ES is deterministic, while in GA used stochastic selection including RWS, SUS.

ELITIST SELECTION POLICY

We tested different selection policies; the one that proved to be the best is elitist selection. It simply means that the rule is applied: the strongest survive. A large portion of the best individuals survive from one generation to the next. In practice, at a time t, we select a large part of the population (90% for example), this part is made up of the best individuals, in the sense of the criterion sought (the mean squared error) (Trivedi et al., 2017)

The rest of the population will be replaced by other individuals at the time t+1. These new individuals are obtained by combining the selected individuals (Goldberg, 1989).

THE PROCESS OF TRAINING MLP USING OPTIMIZER ALGORITHMS

This section represents a detailed description for training process of MLP network using EA-based optimizer algorithm. This approach is named EA-MLP which is divided to two models GA-MLP and ES-MLP. In this study, The MLP network contains only one single hidden layer. To achieve this goal, two key aspects are taken into consideration for constructing EA-MLP algorithm:

the encoding representation of individuals in the EA algorithm and selecting the formulation of the fitness function. In ES-MLP, all individuals are encoded as one-dimensional vectors of random real numbers inside the interval [-1, 1], whereas, in GA-MLP a binary encoding is employed. Each generated solution by the encoding schema represents an MLP candidate. The designed vectors include three key parts: a set of weights connecting the input layer to hidden layer, the connection weights between the hidden layer and the output layer, and a set of bias weights (Abusnaina et al., 2018). The structure of agents in the proposed EA-MLP is shown in figure 1.

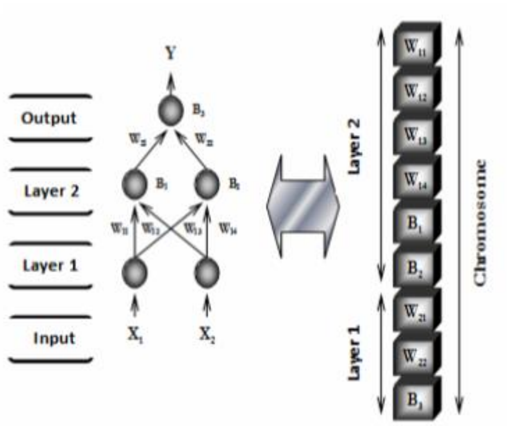

Fig. 1. The evolutionary representation of a neural network.

To evaluate the fitness value of EA-MLP approach, the vector of biases and weights is passed to the MLP network. In this work, the mean squared error (MSE) is utilized as the fitness function. This evaluation metric calculates the difference between the actual and predicted values obtained by the generated individuals (MLPs) using training samples. MSE metric is attained by equation (8):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \quad (8)$$

Where y_i and $\hat{y}_i$ are the actual and predicted values, and *n* indicates the number of instances in the training datasets. The workflow of the EA-based MLP algorithm is shown in figure 2.

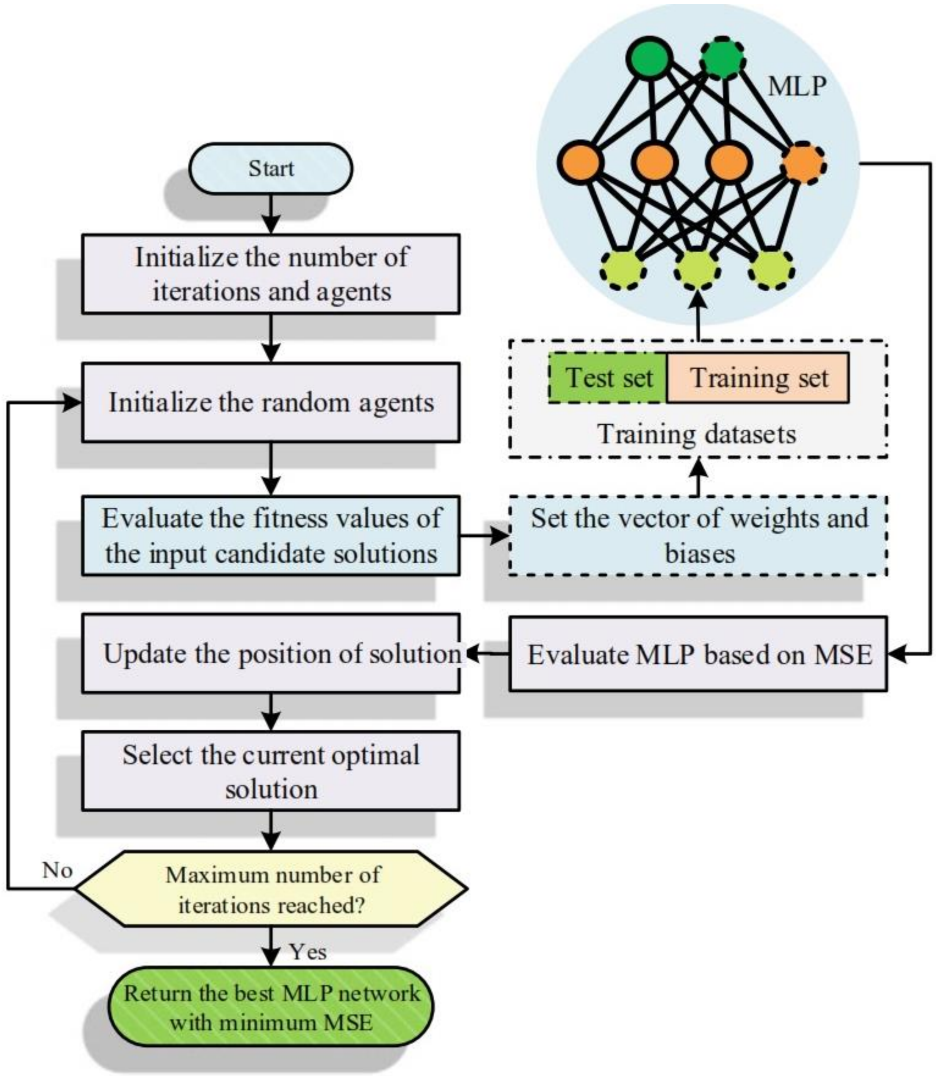

Fig.2. The framework of EA-MLP training process.

EXPERIENCES AND RESULTS

The phoneme classification experiments were carried out on a subset of the TIMIT database consisting of 6 vowels, 6 fricatives and 6 plosives (Table 1).

The signals were sampled at 16 kHz with cepstral analysis under the Mel scale, taken every 20ms in 25ms Hamming windows each giving 12 Mel Frequency Cepstral Coefficients (MFCC) coefficients and the corresponding residual energy (Al-Kaltakchi et al.,2017).

The selected TIMIT corpus has a large number of data to process 31,514 occurrences at the learning level and 12055 occurrences at the test phase. For this, we made a classification in order to compress the data using the LBG algorithm with the variant of k-means relating to vector quantization (Ramesh et al., 2017)

Transcripts of TIMIT corpora were generally verified. Assay and training subsets, balanced for phonetic and dialectal assurance, are indicated. Computer-readable information is included as well as written documentation.

The 18 phonemes, which are each characterized by several vectors of 13 MFCC coefficients, will be processed globally at the vector quantization level with 64 prototypes (Figure 3). The use of vector quantization has had the effect of reducing the learning time by assigning to each treated phoneme interval a vector of 64 binary components.

Table 1. TIMIT sub-corpus.

		Classes		
		Phonemes	*Train*	*Test*
Phonetic	Vowels	/ah/	2200	879
		/aw/	700	216
		/ax/	3352	1323
		/ax-h/	281	95
		/uh/	502	221
		/uw/	536	170
	Fricatives	/dh/	2058	822
		/f/	2093	911
		/sh/	2144	796
		/v/	1872	707
		/z/	3574	1273
		/zh/	146	74
	Plosives	/b/	399	182
		/d/	1371	526
		/g/	1337	546
		/p/	2056	779
		/q/	3307	1191
		/t/	3586	1344

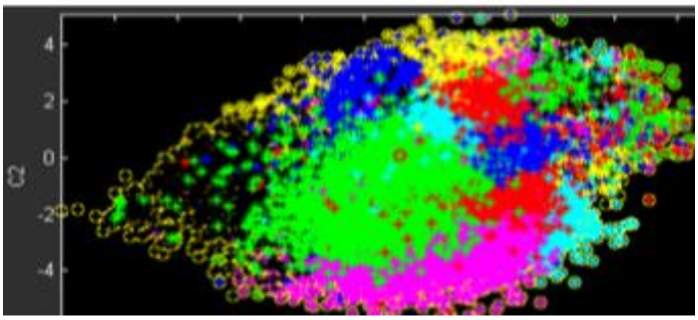

Fig. 3. The representation according to the first two cepstral coefficients (C1, C2).

For learning, we chose the use of a single layer NN hidden, to which we have varied the size of the input layer, and the size of the hidden layer, to finally keep the most efficient after several attempts and which was a NN at an input layer of 64 neurons (relative to the number of prototypes), a hidden layer of 36 neurons and an output layer of 18 neurons (relative to the number of phonemes).

The two hybrid models GA-MLP (Neural network Genetic Algorithm) and ES-MLP (Neural network Evolution Strategies) were implemented with the classical BP (gradient retro-propagation approach) to compare them, and for this reason the parameters given in table 2 were chosen after several test.

Table 2. Training parameters.

Parameters	Values
BP	
Neuron Activation	sigmoid
Iteration	1000
Desired error	0.001
Learning step	0.7

GA-MLP	
Selection operator	roulette
1 locus Crossing	Pc=0.7
Elitism	el=0.9
Mutation	Pm=0.01 Gaussian
Population size	10
Generation number max	1000
ES-MLP	
Selection operator	Determinist $(\mu,\lambda)=(1,10)$
Mutation	Correlate (x,σ,α)
Population size	1 parent
Generation number max	1000

The use of both hybrid models, for a medium-sized TIMIT speech corpus, resulted in overall classification terminal rates ranging from 45.54% to 52.98% using the first evolutionary GA-MLP approach but for the second ES-MLP approach. it leads to an even higher rate 59.24%. The results obtained from the experiments conducted are detailed in figure 4.

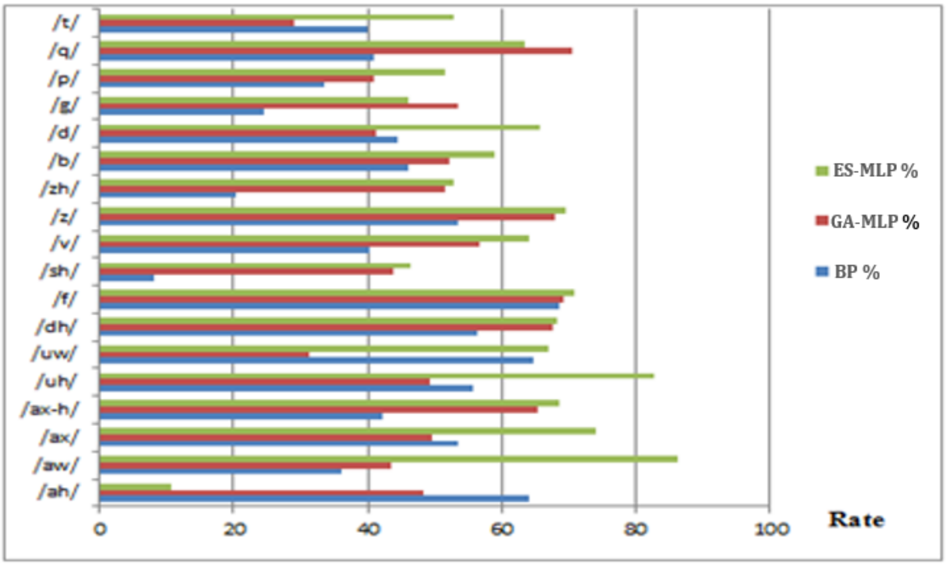

Fig. 4. Recognition scores.

The performance of these methods results in an increase of the scores, more than 11%, especially for the consonants and more particularly for the fricatives (Table 3).

Table 3. Recognition rates obtained for different models by phonetic classes.

	BP (%)	GA-MLP (%)	ES-MLP (%)
Vowels	55.79	48.24	55.72
Fricatives	46.56	61.88	64.46
Plosives	38	47.07	56.26
Overall accuracy	46.78	47.07	58.81

From the results illustrated above, we can a significant improvement in the recognition rate in the two hybrid models GA-MLP and ES-MLP compared to the result obtained by the conventional method BP.

The GA considers that the main genetic operator is that of crossing, while the other ES technique favors the mutation operator. GAs use proportional selection and overall parent replacement by children, while ES use little selection and rely on deterministic replacement.

GAs and ESs have shown their ability to avoid convergence of solutions to local optima, both when combined with other approaches such as the connectionist model (Belew et al., 1992) and when they are alone.
Evolutionary research requires a generally longer learning time, but may, on a case-by-case basis, produce better results than conventional learning methods such as retro-gradient propagation.
An entity is well recognized if the output provided by the network is the same as the desired output.
The recognition function has three parameters of the same type as those of the learning function: the number of entities to be recognized, the input characteristic vectors and the associated desired outputs.
As analysis, we can that GA required more time than other optimizer due to the binary encoding and the use of several operators as Selection, crossover and mutation. For solving this drawback, we propose another type of EA called ES which is more speed and efficient due to the real encoding and the simple operator employed for updating the population.
In addition, all algorithms used the same set of training in order to realize a fair comparison in the same condition (the same number of individuals, the number of generations).
Concerning the results of table .3, we can that BP and ES-MLP provide the same performance for the classification of vowels. This behavior can be interpreted by two reasons: the use of real coding of weights and biases. The second reason, the number of MFCC vectors in vowels category is less than other categories which increase the performance.

CONCLUSION
In the world of artificial intelligence today, we tend to recopy what nature does. And what's more normal than copying the human brain when we talk about intelligence and thinking. Nature tells us that natural selection also brings species improvement. If, then, we want a computer application that reacts like a human, we must try to code the particularities.
Faced with an optimization problem, the comparison of the performances of the different approaches between them is delicate and must be conducted with great care. For EAs, good performance results have been achieved by their performance enhancing properties and their combination with neural networks provides us with a more efficient hybrid model.
Gradient descent methods are subject to variations in performance due to the initial position of NN weights sometimes leading to convergence to local minima. Evolutionary methods, on the other hand, provide research in the complete domain. As generations progress, this search space is refined to potentially performing subspaces. However, it is common for EAs to find a solution close to the best without ever reaching it. It can be assumed that these two methods are complementary.
As future work, the other more recent evolutionary methods implementation, such as EDA (Estimation Distribution Algorithms) and GP (Genetic Programming), is desirable, on the one hand, with the aim of finding higher classification rates and on the other hand, to clearly identify all the currently popular approaches related to Evolutionary Algorithms.

REFERENCES
Abusnaina, A. A., Ahmad, S., Jarrar, R., & Mafarja, M. (2018). Training neural networks using Salp Swarm Algorithm for pattern classification. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3231053.3231070

Anwar, M. U., & Ali, M. L. (2019). Boosting Neuro Evolutionary Techniques for Speech Recognition. *2nd International Conference on Electrical, Computer and Communication Engineering, ECCE 2019*. https://doi.org/10.1109/ECACE.2019.8679206

Bansal, P., Kumar, S., Pasrija, S., & Singh, S. (2020). A hybrid grasshopper and new cat swarm optimization algorithm for feature selection and optimization of multi-layer perceptron. *Soft Computing*, *24*(20). https://doi.org/10.1007/s00500-020-04877-w

Belew, R. K., McInerney, J., & Schraudolph, N. N. (1992). Evolving Networks: Using the Genetic Algorithm with Connectionist Learning. *Artificial Life II*, *10*.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989. *NN Schraudolph and J..*, *3*.

Goldberg, D. (1994). Algorithmes génétiques, éditions. *Addison Wesley, juin*, *5*, 5-8.

Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. In *Evolutionary computation* (Vol. 9, Issue 2). https://doi.org/10.1162/106365601750190398

Haton, J.-P., & Haton, M.-C. (1989). L'intelligence artificielle. Presses universitaires de France.

Heidari, A. A., Faris, H., Mirjalili, S., Aljarah, I., & Mafarja, M. (2020). Ant lion optimizer: Theory, literature review, and application in multi-layer perceptron neural networks. In *Studies in Computational Intelligence* (Vol. 811). https://doi.org/10.1007/978-3-030-12127-3_3

Hellwig, M., & Beyer, H. G. (2018). A Matrix Adaptation Evolution Strategy for Constrained Real-Parameter Optimization. *2018 IEEE Congress on Evolutionary Computation, CEC 2018 - Proceedings*. https://doi.org/10.1109/CEC.2018.8477950

Jalali, S. M. J., Ahmadian, S., Kebria, P. M., Khosravi, A., Lim, C. P., & Nahavandi, S. (2019). Evolving artificial neural networks using butterfly optimization algorithm for data classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11953 LNCS*. https://doi.org/10.1007/978-3-030-36708-4_49

Junqua, J.-C., & Haton, J.-P. (1996). Robustness in Automatic Speech Recognition. Fundamentals and Applications. In *Kluwer Academic Publishers*.

Laumanns, M., Rudolph, G., & Schwefel, H. P. (2001). Approximating the Pareto set. *HT014601767*.

Musab, M. T., Woo, W. L., Dlay, S., & Chambers, J. A. (2017). Evaluation of a speaker identification system with and without fusion using three databases in the presence of noise and handset effects. *Eurasip Journal on Advances in Signal Processing*, *2017*(1). https://doi.org/10.1186/s13634-017-0515-7

Miclet, L., & Haton, J. P. (1984). *Méthodes structurelles pour la reconnaissance des formes* (Vol. 61). Paris: Eyrolles.

Pal, S. K., & Wang, P. P. (2017). Genetic Algorithms for Pattern Recognition. In *Genetic Algorithms for Pattern Recognition*. https://doi.org/10.1201/9780203713402

Ramesh, B., Xiang, C., & Lee, T. H. (2017). Multiple object cues for high performance vector quantization. *Pattern Recognition*, *67*. https://doi.org/10.1016/j.patcog.2017.02.024

Spalanzani A., 1999. Algorithmes évolutionnaires pour l'étude de la robustesse des systèmes de reconnaissance de la parole (PhD Thesis). Université Joseph-Fourier-Grenoble I.

Tang, L., Zhao, Y., & Liu, J. (2014). An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production. *IEEE Transactions on Evolutionary Computation*, *18*(2). https://doi.org/10.1109/TEVC.2013.2250977

Trivedi, A., Srinivasan, D., Sanyal, K., & Ghosh, A. (2017). A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, *21*(3). https://doi.org/10.1109/TEVC.2016.2608507

Wang, L., Zeng, Y., & Chen, T. (2015). Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. *Expert Systems with Applications*, *42*(2). https://doi.org/10.1016/j.eswa.2014.08.018