

Evolution of cooperating ANNs through functional phenotypic affinity

F. Bellas, J.A. Becerra, R. J. Duro

Grupo de Sistemas Autónomos, Universidade da Coruña, Spain
fran@udc.es, ronin@udc.es, richard@udc.es

Abstract. This work deals with the problem of automatically obtaining ANNs that cooperate in modelling of complex functions. We propose an algorithm where the combination of networks takes place at the phenotypic operational level. Thus, we evolve a population of networks that are automatically classified into different species depending on the performance of their phenotype, and individuals of each species cooperate forming a group to obtain a complex output. The components that make up the groups are basic ANNs (primitives) and could be reused in other search processes as seeds or could be combined to generate new solutions. The magnitude that reflects the difference between ANNs is their affinity vector, which must be automatically created and modified. The main objective of this approach is to model complex functions such as environment models in robotics or multidimensional signals.

1 Introduction

The researchers dealing with modelling real environments tend to dedicate great efforts to the automatic modularization of the solutions, searching for models as simple and reusable as possible. In the field of evolutionary learning, an element of paramount importance is how to make efficient use of the learned structures to obtain other structures that can be applied in more complex modelling processes without performing a new learning process.

A very powerful technique for the instantiation of the models, when the learning process consists in modelling from input-output pairs, is the application of artificial neural networks (ANNs). In the case of evolutionary learning, these models adapt to the task through evolution and the modularity of solutions can be obtained creating complex ANN sets as a composition of simpler ones. In this sense, a very common approach in the literature [1], [2], [3] is to apply an evolutionary algorithm over two different populations, one made up of simple neuronal units and the other of patterns that indicate how those neural units must be combined. These approaches have been successful, for example, in classification problems.

A different approach is proposed here, where the basic structures participating in the process are complete networks evolving in the same population and they are combined at the phenotypic level into groups that collaborate in the solution of the problem. The work of Xin Yao and Paul Darwen [4] must be mentioned here. They propose an evolutionary learning approach to design modular systems automatically. To do this,

they use the concept of speciation introduced by Goldberg [5] through a fitness sharing technique [6]. The main novelty of this work lies in the fact that within each species they use coevolutionary techniques where the concept of just one individual solution does not exist. The authors apply this technique to the prisoner's dilemma and the algorithm works successfully.

This kind of modular approaches imply the existence of collaboration between individuals of the population to achieve a successful result. This is called in the literature symbiotic evolution and is a background problem in multiobjective optimization [7]. In this field, the development of fitness sharing techniques, where the fitness of a given individual is scaled through some similitude measure with respect to the other individuals, is common [8]. These approaches start from the knowledge of the Pareto front to be obtained, this is, they start from the knowledge of the desired objectives and search for individuals specialized in these objectives and that collaborate towards their achievement.

The problem we try to solve in this work is quite different because we look for an automatic decomposition of the model into simple and basic networks (primitives) that combined provide the desired solution. No knowledge about how to do it is assumed (we don't know the best decomposition into primitives). Each one of these simple networks is functionally different and, consequently, they belong to different species that evolve together.

A recent algorithm that is based on coevolution and species formation is COVNET [9], where authors use a subnetworks population (nodules) that is divided into species that evolve independently, and another network population that combine these nodules. The main difference with our approach lies at the species formation level. The number of species in COVNET is prefixed, whereas in our algorithm it arises automatically from the evolution itself.

In the next section we will introduce the main concepts used in this algorithm such as groups and affinity. The remaining sections provide an overview of the operation.

2 Groups and Affinity

The main objective of this work is very simple: to develop an algorithm capable of modelling a complex function using ANNs. Because of this complexity, we assume it will be simpler to find the solution using more than one ANN. So our approach is based on providing a search algorithm with the capability of obtaining the solution by aggregation of simple ANNs.

We do not want to impose knowledge about the way to carry out this combination because, in most cases, it is not possible to know how to extract simple parts (that could be modeled separately) from a complex function. This is, we know what we want to obtain from our ANNs (phenotype) but we don't know how to achieve this (genotype). This way, our efforts in the design of the algorithm are concentrated in obtaining a procedure to efficiently combine the outputs provided by different ANNs to reach a desired result.

Thus, the first important feature of our algorithm is that the solution is provided by a combination of ANNs (*a group*) that, in the simplest case, will be formed by just one

network but it could contain several. All the ANNs in a group combine their outputs to provide a complete solution to the problem.

To conform the groups, we have developed a selection criteria based on a magnitude called *affinity* that classifies the ANNs depending on their phenotype. The main idea is that initially all the networks have a random affinity and through a self organization process these affinities become different automatically. Then, the groups that provide the solution to the problem are made up by combining ANNs with high affinity to each other. The details of this process will be explained in section 4.2.

3 Evolution

From the previous basic concepts of the algorithm two main requirements can be extracted: first that our approach is based on the use of a set of ANNs that can be combined and second that to improve the networks we must apply a search algorithm. Taking into account these two points, we propose the use of an evolutionary algorithm as search technique because it is based on a population (a set) of solutions and it is an established technique for adjusting the parameters of ANNs. Furthermore, evolutionary algorithms are very adequate for this approach because of their capabilities to create specialized individuals (*species*) in the population. This way, once some species appear in the population, we can make up the groups that conform the solutions using individuals from these species.

The differentiation between species must arise from the phenotypic capability of each ANN. It is the collaboration towards a common goal by a group what determines the different species through the association of individuals that are specialized in the different parts of the solution. This association is established through some type of affinity criterion based on the functional performance of the phenotype of the ANNs, so that individuals with higher affinity towards each other tend to collaborate in the tasks to be carried out.

We have developed a process of life interaction in order to evaluate the individuals in such a way that the survival of an individual is determined by the fitness obtained by the whole group in which the individual participates and not only on its particular fitness. The most important part of this group formation strategy is establishing the affinity among individuals, which must be a part of the evolutionary process and progressively adapt to the needs and virtues of each individual autonomously.

4 Details of the algorithm

From the point of view of an evolutionary algorithm, the objective of group creation is to obtain individuals made up of subindividuals through the combination of their particular actuations, thus permitting a much more versatile way of reusing acquired knowledge.

As mentioned in the previous section, the proposed strategy is a part of any standard evolutionary algorithm where the individuals represent ANNs and where the solution is not a simple individual but an aggregation of some (a group) at a phenotypic level.

Consequently, individuals will be evaluated (fitness calculation) as a part of a group and not individually. Evolution, on the other hand, takes place over the individuals as usual and the groups are formed in the fitness calculation stage of the algorithm.

The operation is simple and can be summarized in the following 6 steps:

1. Initial creation of a random population of ANNs, as usual in general evolutionary algorithms.
2. Initial creation of a random affinity value (represented by a vector in general) associated to each individual.
3. Fitness calculation of all the ANNs in population. To do this, we apply 5 basic steps:
 - a. Each individual in the population creates its own group by choosing other individuals from a window of the population using its and their affinity values.
 - b. Each group calculates a fitness value, resulting from the common application of its individuals to the modelling problem.
 - c. Each individual creates a new group and the fitness in these new groups is calculated. The best groups remain.
 - d. The fitness value obtained in step c) permits adjusting the affinity of the individuals depending on the increase or decrease of fitness.
 - e. Steps c) and d) are repeated a given number of iterations. At the end, we assign each individual the fitness of its own group.
4. Selection over the population of ANNs (like in an ordinary evolutionary algorithm) including the affinity value of each individual as a part of the selection criteria favoring reproduction of individuals with similar functional characteristics (same species).
5. Crossover and mutation as usual but including the affinity vector that is transmitted from parents to offspring.
6. Steps 3, 4 and 5 are repeated a given number of generations.

As we can see, this is the operation schema of a simple evolutionary algorithm where the fitness calculation is more complex and where the reproduction is directed towards the formation of species by mixing operationally similar individuals.

Each individual can form its own group selecting the other individuals from a candidate's window instead of the whole population, trying to reduce the number of possible combinations. In addition to its own group, every individual can participate in other groups without limitation of number.

4.1 Fitness calculation

Once a group is formed, its fitness is calculated in general as the fitness provided by a set of outputs that are the result of some combination of the individual outputs of its members. In the simplest case the combination is carried out through the addition of these outputs that must be clearly marked to discriminate which ANN affects which output. The group must provide values for all the outputs of the problem, otherwise it is penalized.

After calculating the fitness of all the groups, we carry out a self organization stage (steps from a) to e) in the algorithm), which allows each individual to create more affine groups. On each step of this stage, the individuals form new groups and their fitness is calculated. If the fitness of the new group is better than that of the old one, the group is maintained; otherwise the older group is preserved. After a set number of recombination steps, each individual has formed its most affine group using the technique explained in the next section.

4.2 Affinity

Affinity is used in our algorithm as a label to classify the ANNs depending on the function they perform. It is represented by a numerical vector with fixed upper and lower limits, in which the dimension determines the complexity of the groups that may be formed.

Each element in the affinity vector is represented by a real number. When an individual forms a group it searches for other individuals in the population with *complementary* affinity vectors. For example, if each element of the vector is represented by a real number between $-V$ and V , we can establish that the desired global affinity value is the zero vector. This way, a component value of $-V_1$ is the complementary of a value of V_1 and a value of $-V_2$ is the complementary of a value of V_2 . In most cases, an individual will not find another individual with a complementary affinity and it will be necessary to use two or three colleagues to reach the desired affinity.

As we can see, the number of individuals that conform a group depends on the dimension of the affinity vector. The higher the dimensionality the larger (on average) the number of individuals in the group necessary to obtain complementary affinities. For example, if we have an individual with a 3-dimensional affinity vector $(V_1, V_2, -V_3)$ it could select one only individual with affinity $(-V_1, -V_2, V_3)$ to form the group. But it is simpler to find two individuals of vectors $(-V_4, -V_5, V_6)$ and $(-V_7, -V_8, V_9)$ that verify $-V_1 = -V_4 + (-V_7)$, $-V_2 = -V_5 + (-V_8)$ and $V_3 = V_6 + V_9$. As we can see, higher dimensionalities imply a more complicated process to find individuals with complementary affinities.

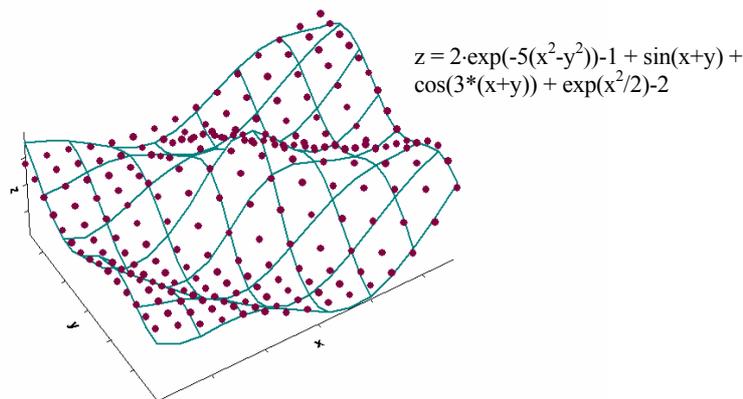


Fig 1. Modelling result (points) of a 3D function (surface) made up by the addition of 4 basic functions

Initially, the affinity vectors are random as no knowledge of the characteristics of the individuals and/or their shortcomings is available. As mentioned before, every time a group is evaluated (fitness calculation), if the group improves the fitness of the previous group in which the individual participated, the distance of the affinity vector of the individual (in Euclidean terms) with respect to the affinity vectors of the rest of the individuals in the group is increased. In addition, from this instant, this will be the new group for the individual. On the other hand, if the fitness of the group is worse than that of the previous group, it is the distance with respect to the affinity of the members of the previous group that is increased. This method favours the formation of species, that is, of clusters of individuals with similar affinity vectors.

After each self organization stage, all the individuals have automatically adjusted their affinity vector depending on the function they carry out in the group. The individuals that are included in more than one group adjust their affinity vector in each one, maximizing the available information. As time progresses, each individual will have an affinity vector representing its preferences that has been automatically obtained.

In the evolutionary process, we tend to select for reproduction individuals with similar affinity vectors, which are transmitted in the crossover from the parents to the offspring. This way, we are promoting the formation of species in the population. The feature that distinguishes these species (the affinity vector) has not been imposed by the designer, but it is adjusted in “lifetime” through the evolution.

5 Operation example

Just to display the results the mechanism can produce and the way it works, we consider here a simple problem consisting of an objective function that was constructed by adding 4 different functions: $2 \cdot \exp(-5(x^2-y^2))-1$, $\sin(x+y)$, $\cos(3 \cdot (x+y))$ and $\exp(x^2/2)-2$ limited to a range from $[-1.5, 1.5]$ in x and y . In this experiment, the affinity vector considered is two dimensional and the values of each dimension can vary between $[-1, 1]$. Thus, it seems that an optimal solution for this problem would be the generation of four species, each one specialized in modelling each one of the known functions or any other combination of functions that lead to the perfect modelling of the global function.

The networks that will be evolved in this case are standard multilayer perceptrons which have two inputs (the x and y coordinates) and one output (the z coordinate). The objective is obviously to obtain the network or combination of networks that best approximate the objective function.

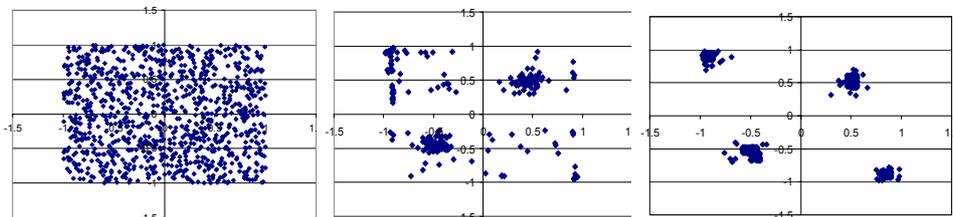


Fig. 2. Evolution of the affinity vectors for the whole population. Left graph represents the initial distribution, middle graph the distribution after 2000 generations and right graph the final distribution

The results of modelling obtained with the best group are presented in Fig. 1, where we have represented the prediction (points) and a sampling of the objective function (surface). The results are very satisfactory taking into account the complexity of the expected signal. We are going to take a closer look at what have happened with the affinity values for the individuals and what types of species arose.

Fig. 2 displays the evolution of the affinity vectors for the whole population (1000 individuals in this case). In the left graph we have represented the initial random affinities distributed from -1 to 1 in both axes. In the middle graph we show the affinity vectors in generation 2000. As we can see, the values are still not clustered, and some individuals do not belong clearly to a species, although four affinity areas are starting to be delimited. Finally, in the right graph we show the affinity vectors in generation 3600 (final) where 4 clear species have been formed with affinity vectors $(-1, 1)$, $(1, -1)$, $(0.5, 0.5)$ and $(-0.5, -0.5)$.

The final group taken as solution of the modelling problem (that provides prediction shown in Fig. 1) is made up of 4 individuals, one from each species. In Fig. 3 we present the functions provided by each one of these individuals in the final group, as compared to the original four functions that were combined in order to produce the test problem. We can see that the decomposition into primitives *is different* from the original one which is normal because there are lots of possible mathematical combinations to reach the desired functions. What is important here is that these primitives have been obtained automatically and could be reused in other search processes.

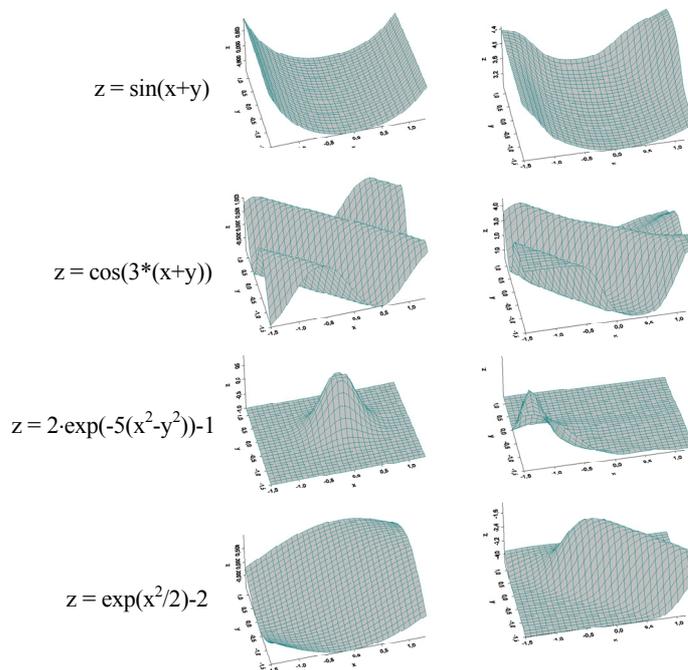


Fig. 3. The left graphs show the four theoretical functions that make up the target in the operation example. The right graphs show the functions provided by each individual in the best group (phenotype corresponding to each specie)

Conclusions

In this work we have presented an affinity based strategy for obtaining groups of cooperating artificial neural networks as a solution for evolutionary learning problems. This strategy may be applied in general to any type of evolutionary algorithm where the objective is to obtain a group of solution points which, through their combination and cooperation in a phenotypic level produce a modular or decomposable solution to the problem so that parts of these solutions may be later reused in other problems. The main idea behind the phenotypic affinity based process is that during the evaluation phase of the evolutionary algorithm the individuals can try out how fit they are through collaboration with other individuals by forming groups. The results of these trials result in modifications of each individual's affinity vector. This process leads to the formation of species of individuals specialized in different subtasks of the global one, thus allowing for the modularity we desire. The arising of these species implies certain modifications in the general evolutionary algorithm, especially during the reproduction stage so that functionally similar individuals are chosen for reproduction. The results applying this strategy have been very fruitful. Here we have presented a quite complex case with the arising of four species into the population.

Acknowledgements

This work was funded by the MCYT of Spain through project VEM2003-20088-C04-01 and Xunta de Galicia through project PGIDIT03TIC16601PR.

References

1. Smalz, R., Conrad, M.: Combining evolution with credit apportionment: A new learning algorithm for neural nets, *Neural Networks*, vol 7, num 2, 1994.
2. Opitz, D.W., Shavlik, J.W.: Actively searching for an effective neural network ensemble, *Connection Sci*, vol 8, num 3, 1996.
3. Moriarty, D.E., Miiikkulainen, R.: Forming neural networks through efficient and adaptive coevolution, *Evolutionary Computation*, vol 4, num 5, 1998.
4. Darwen, P., Yao, X.: Speciation as automatic categorical modularization, *IEEE Transactions on Evolutionary Computation*, 1(2), 1997.
5. Goldberg, D., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization, *Proceedings of 2nd International Conf. on Genetic Algorithms*, 1987.
6. Deb, K. and Goldberg, D. E.: An investigation of niche and species-formation in genetic function optimization, *Proc. 3rd Int. Conf. Genetic Algorithms*, pp 42-50, 1989.
7. Coello, C., Van Veldhuizen, D., Lamont, G.: *Evolutionary algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, 2002.
8. DellaCioppa, A., DeStefano, C., Marcelli, A.: On the Role of Population Size and Niche Radius in Fitness Sharing, *IEEE Trans on Evolutionary Computation*, vol 8, num 6, 2004.
9. García-Pedrajas, N., Hervás-Martínez, C., Muñoz-Pérez, J.: COVNET: A Cooperative Coevolutionary Model for Evolving Artificial Neural Networks, *IEEE Transactions on Neural Networks*, vol 14, num 3, 2003.