International Congress on Evolutionary Methods for Design,
Optimization and Control with Applications to Industrial Problems
EUROGEN 2003
G. Bugeda, J. A- Désidéri, J. Periaux, M. Schoenauer and G. Winter (Eds)

EVOLUTIONARY FEED-FORWARD NEURAL NETWORKS FOR TRAFFIC PREDICTION

M. Annunziato[*], I. Bertini[*], A. Pannicelli[†], S. Pizzuti[†*]

[*] Environment New technologies and Energy Agency (ENEA)
'Casaccia' Research Centre
Via Anguillarese, 301, 00060 Rome, Italy
e-mail: {mauro.annunziato, ilaria.bertini, alessandro.pannicelli, stefano.pizzuti}@casaccia.enea.it

[†] Communication and Systems s.r.l
Piazza della Repubblica, 32, 20124 Milan, Italy

Key words: evolutionary neural networks, short term urban traffic prediction, adaptive systems.

Abstract. In this paper we show different evolutionary algorithms in order to optimise on-line weights of feed-forward neural networks when applied to short term (20 min.) urban traffic prediction. We compare the evolutionary methods with the classical back-propagation algorithm and we show results when weights are off-line and on-line evolved. Preliminary results are very promising and show the effectiveness of the proposed approach in order to get neural models capable to dynamically adapt to environmental changes.

1 INTRODUCTION

Neural computing is now considered a mature technology and indeed its uptake, particularly in exploratory work. Yet the most popular technique described above has only been available for more than 10 years. Set against this background, the use of neural computing for transportation applications began much more recently and work has largely been of an exploratory nature. Applications which have been addressed using *artificial neural networks* (ANN) include forecasting/classification of traffic flow parameters/traffic states [i,ii,iii,iv,v], incident detection [vi,vii,viii], driver behaviour/vehicle control [ix,x,xi,xii], traffic control [xiii,xiv] and traffic monitoring [xv]. In nearly all of the applications reviewed the back-propagation learning algorithm was used. Despite some encouraging results its main drawback is the lack of on-line adaptation to changing conditions. For artificial neural networks to be viable for on-line applications in transportation they will need to be able to function in real time. Recently another interesting area is the one concerning the application of evolutionary computation based methodologies applied to traffic control [xvi,xvii,xviii], traffic management [xix], traffic signal operation [xx]. What we propose is an innovative approach for traffic prediction which combines these two methodologies in order to carry out evolutionary neural models capable to on-line and dynamically adapt to changing conditions. In fact some recent investigations in combining evolutionary algorithms and ANN have shown very promising results [xxi] giving rise to a new class of ANN called *evolutionary neural networks* (ENN) [xxii,xxiii]. In this paper we show two different evolutionary algorithms applied to the off-line and on-line weights optimisation of feed-forward neural networks and we report results when applied to short term (20 min.) urban traffic prediction. Moreover we compare these methods with the classical back-propagation algorithm.

2 DATA PROCESSING

Several tests have been carried out in order to decide the inputs of the neural network. The basic idea has been derived by the non linear dynamic analysis of the traffic flow signal. In figure 1 an attractor reconstruction in the 2D pseudo-embedded space is reported taking into account several days of the week. This plot is built using two time shifted samples in the traffic flow signals: $x(t)$ and $x(t+T)$, where T is the characteristic delay which allows the unfolding of the attractor. In order to compute the real attractor we should compute the exact number of dimensions we need for the description of the problem. In this case we use this figure only to explain the approach [xxiv,xxv]. Using delay times ranging from 5 to 15 minutes in the attractor reconstruction, we obtain a good unfolding. This means that a set of variables built in this way should be able to give a correct minimal description of the signal dynamics. On this base we choose a T delay time of 10 minutes. The original data are supplied in measurements of the traffic flow every 1 minute. Considering a delay time of 10 samples it is possible to proceed to an under-sampling in order to reduce the computation time and the noise connected with the statistical bias. For this reason we apply a 5 minutes linear moving average filter and we re-sample the signal with a period of five minutes. Finally we use as inputs for the network a series of samples taken every 10 minutes as the T delay time. We carried out several tests to decide a reference neural architecture. The result was a feed forward with 8 input neurons, one single hidden layer of 3 neurons and one output neuron. In this way the 8 inputs correspond to the last 80 minutes of traffic flow.

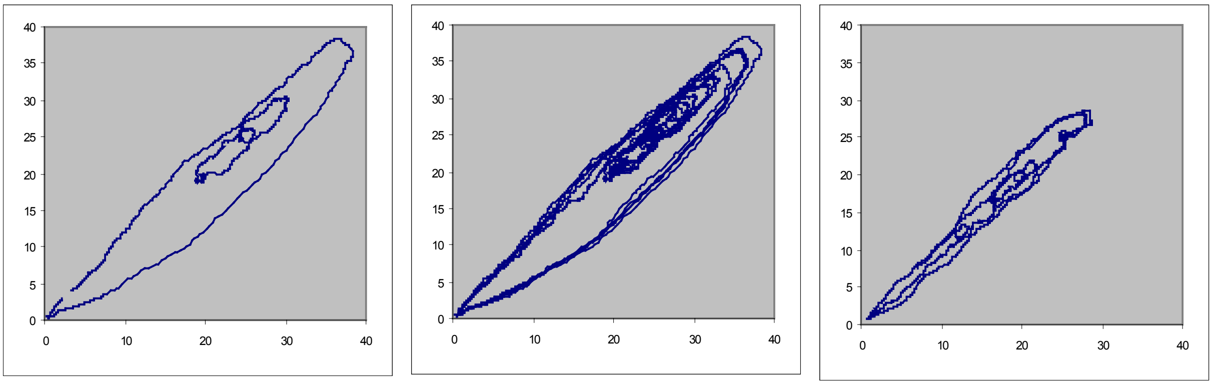

Figure 1 : pseudo attractor of traffic flow signal on Monday (left), Monday to Friday (center) and from Saturday to Sunday (left)

3 THE EVOLUTIONARY ALGORITHMS

We conducted tests with two different types of evolutionary algorithms we carried out. Their basic principle consists in leaving the system a certain degree of freedom in order to develop an emergent behaviour by combining genetics with other peculiar aspects of life (interaction, competition, co-operation, food quest, etc.). In the proposed algorithms each individual represents a feed forward neural network in competition with the others by means of the proper fitness, which depends on the capability of reconstructing the training database having as genotype the synaptic weights. In both the algorithms the only parameters to be set are the number of generations and the max population size. All the other parameters, like crossover and mutation rates, depend on the internal dynamics. Fitness is measured referring to the global error in modelling the training database with the following formula:

$$Fitness = 1 - RMSE \tag{1}$$

Where *RMSE* is the classical Root Mean Squared Error normalised in the lattice [0,1] used by the back-propagation (BP) algorithm. This cost function has been chosen in order to directly compare the results with those obtained with BP methodology.

$$RMSE = \sqrt{\frac{1}{n}\sum_{1}^{n}(0.5*\sum_{1}^{m}(y-y_t)^2)} \tag{2}$$

Where n is the dimension of the training data set, m is the number of output neurons (in this situation m=1), y is the estimated output and y_t is the corresponding target (real value). All the inputs and outputs of the networks are normalized between 0 and 1. As measure of the reached level in the training we take the fitness of the best individual (corresponding to the best performing neural network). The algorithms we applied, which are going to be outlined in the next paragraphs, are Partial Emulation (PE) [xxii], and Chaotic Populations (CP) [xxvi].

3.1 Chaotic Populations

This algorithm is inspired by the fact that it is known that in natural environments population sizes, reproduction and competition rates, change and tend to stabilise around appropriate values according to some environmental factors. In this way it has been carried out a technique for setting the genetic parameters during the course of a run by adapting the population size and the operators rates on the basis of the environmental constrain of maximum population size. The main features of the algorithm are : meeting, mono-sexual and bi-sexual reproduction, and competition. In this evolutionary algorithm the roulette wheel selection is replaced with the meeting concept. At each iteration we pick the i-th individual of the population, for i from 1 to the current population size, up and then we randomly look for a second individual. Therefore the meeting probability is defined as the population density. In this stage if someone is met then interaction (bi-sexual reproduction or competition) will start, else mono-sexual reproduction of the current individual might occur. Bi-sexual reproduction is performed according to an adaptive rate and if it occurred then the resulting sons would not replace their parents, they would simply be added to the population. In this situation population increases of two new elements. Mono-sexual reproduction is performed according to an adaptive reproduction rate and when it occurs an individual first clones itself and then mutates. The mutated individual doesn't replace the original one, it is simply added to the population and the population size increases of one unit. Competition starts according to an adaptive rate and it means that when two individuals meet and they do not mate through bi-sexual reproduction then they'll fight for survival, the stronger will kill the weaker and this one will be kicked from the population off. The resulting population dynamics are chaotic since the algorithm is a particular instance of a chaotic map similar to the well known logistic map. The reader interested in further details can refer to [xxvi].

3.2 Partial Emulation

In this algorithm, many individuals or agents move in a two-dimensional space. An individual is a point moving over the 2D lattice. The space is divided in cells and one cell can be occupied by only one individual. When an agent tries to enter into an occupied cell an interaction occurs. All the parameters for the dynamics, reproduction, life, interaction are recorded in a genetic map that is defined at the birth of an individual and remain unchanged throughout the individual's life. At every life cycle all the individuals are moved into one of the surrounding cells. The dynamics of the individuals and the probability of interaction depends on several variables like number of agents, modalities of interaction, size of the discretization of the space. This algorithm is inspired by a cooperative-competitive society where optimisation is obtained through a communication-emulation mechanism. The metaphors of evolution and genetics are not included because population is fixed and the agents cannot neither die nor reproduce. The optimisation consists in a sort of development of the best adaptive behaviour. At the beginning, a fixed number of individuals are placed in the space, they start moving around the space and every individual is initialised with a bonus of initial energy. At every life cycle a consumption of energy is paid by each agent. Upon the meeting of two individuals, a mechanism of competition is activated. Competition is based on the value of fitness of the two individuals: the individual with lower fitness transfers a quantum of its energy to the other individual. At the same time a cooperative mechanism is applied: the losing individual applies a partial emulation of the winner one. The partial emulation consists in the modification of its network's weights as in the following formula:

$$W_{li} = a * W_{wi} + (1-a) * W_{li} \quad (3)$$

where W_{wi} is the generic weight (or threshold) of the agent with lower fitness, W_{li} is the generic weight of the agent with higher energy and a is the emulation factor (0.05). When an agent loses energy up to a minimal zero level, its neural network is completely reinitialised and a new bonus of initialisation energy is assigned to it. The complete re-initialisation is important in order to have a basin of completely new solutions continuously improving the biodiversity of the environment. The initial bonus of energy is important to give the re-initialised agents enough time to evolve through the self mutation and the partial emulation mechanism. During this time agents lose energy because of the interaction with well evolved agents. Most of the re-initialised agents come back at zero energy, but a few number of them (emergent agents) reach a good fitness level and are able to pass to the class of well evolved agents. When an individual gains a high fitness, it increases its incoming of energy competing with the other agents. The other individuals try to emulate and learn from him. When the others reach its level and someone becomes better, the first individual starts having an attenuation of the incoming energy and then a drastic energy reduction up to finish its energy. At this point it is forced to change its behaviour to come back to a positive energy incoming. This form of learning is conceptually very different from the other case because of its feature of collective learning and volatility. Knowledge is a product of the whole society and it is moved dynamically between the various individuals and it is generated during the life of the individuals, it can be transmitted through the generations. To conclude, the most interesting feature of this algorithm is the mechanism of the balance between the competitive and the cooperative behaviour. This mechanism enhances the social component of the learning and the formation of niches of evolution. Also a good mechanism to promote new solution and protect emergent solution in included. The general drawback is the need of a wide number of generations to reach the maximum level.

4 EXPERIMENTAL RESULTS

Experimentation concerned the 20 minutes prediction of traffic flow rates using neural networks. The goal is to optimise the weights of a neural network structured with 8 input nodes, described in paragraph 2, three hidden nodes and one output node (the traffic flow rate forecast) using the standard Back-Propagation algorithm (BP) and the above mentioned evolutionary algorithms in order to compare off-line and on-line approaches. In both situations the we used as transfer functions for all the nodes the classic sigmoid (4).

$$y = 1/(1+e^{-x}) \quad (4)$$

The data set consists of one week observations, 1980 data as result of a 5 minutes moving average filter, of the vehicles flow rate of the Genoa's urban freeway downloaded from "The EUNITE SAS data repository". Training and testing sets are exactly the same for all experimentations and the whole data set has been partitioned into 1800 training data and 180 testing data. In the first bunch of experiments a direct comparison with the BP algorithm can be done, in the second one the network is optimised on a travelling window of the last ten data (50 min.). In this situation every time the data set changes different weights are dynamically found, in this way the neural model is capable to adapt in real-time to changes. Table 1 and 2 show a comparison of the main experimental training and testing results. Results are calculated according to the *RMSE* formula (2). In tables 3 and 4 the main experimental settings for off-line and on-line optimisation are shown.

	BP	CP	PE
Off-line	0.1	0.043	0.038
On-line		0.013	0.005

Table 1 : training results

	BP	CP	PE
Off-line	0.11	0.042	0.035
On-line		0.031	0.033

Table 2 : testing results

	Generations/Cycles	Max population size
BP	3000000	
CP	500	100
PE	2000	256

Table 3 : Experimental settings used for off-line training

	Generations	Max population size
CP	50	100
PE	100	256

Table 4 : Experimental settings used for *on-line* training

Tables 1 and 2 show very encouraging results. In the off-line situation evolutionary methods show a remarkable increase of accuracy performance compared to the BP algorithm. The on-line experimentation shows a significant improvement with respect to the previous case as well. The reason for this is pretty simple because in the off-line situation a difficult overall global model is built up while in the on-line case several easy local models are dynamically made. These results clearly show the effectiveness of using evolutionary methodologies to build up adaptive neural models overcoming the off-line drawbacks imposed by the BP based methodologies.

5 CONCLUSION

In this paper we showed the effectiveness of evolutionary neural networks when applied to short time urban traffic prediction in order to develop real time adaptive models. In this work we compared the evolutionary methods with the classical back-propagation algorithm and we showed results when weights are off-line and on-line optimised. Experimental results are very promising. In the off-line situation evolutionary methods showed a remarkable increase of accuracy performance compared to the traditional BP algorithm. The on-line experimentation showed a significant improvement with respect to the previous case as well. The reason for this is pretty simple because in the off-line situation a difficult overall global model is built up while in the on-line case several easy local models are dynamically made. These results showed the effectiveness of using evolutionary methodologies to build up neural models able to on-line adapt to environmental changes, in the this application the changes of the vehicle flow rate, overcoming the off-line drawbacks imposed by BP based methodologies.

ACKNOLEDGEMENTS

Authors wish to thank Prof. D.Anguita of the Department of Biophysical and Electronic Engineering of the University of Genova (Italy) for providing the data.

REFERENCES

[i] M.S. Dougherty, H.R. Kirby, and R.D. Boyle, *The use of neural networks to recognise and predict traffic congestion,* Traffic Engineering + Control, 34(11), 311-314 (1993).

[ii] M.S. Dougherty, and M. Cobbett, , *Short term inter-urban traffic forecasts using neural networks.* Proc. Second DRIVE-II Workshop on Short-Term Traffic Forecasting, Delft, The Netherlands, 65-79 (1994).

[iii] C. Taylor, and D. Meldrum, *Freeway traffic data prediction using neural networks*. Proc. 6th VNIS Conference, Seattle, WA, 225-230 (1995).

[iv] J.W.C. van Lint, S.P. Hoogendoorn, H.J. van Zuylen, *Freeway Travel time Prediction with State-Space Neural Networks*, Transportation Research Record 1811, pp30-39, National Academies Press, Washington D.C. (2003)

[v] G.D. Lyons, M. McDonald, N.B. Hounsell, B. Williams, J. Cheese, and B. Radia, *Urban Traffic Management; The Viability of Short Term Congestion Forecasting Using Artificial Neural Networks.* Proc. 24th European Transport Forum, Seminar H(P407) (1996)

[vi] S.G. Ritchie, and R.L. Cheu, *Simulation of freeway incident detection using artificial neural networks*. Transportation Research C, 1(3) 203-217 (1993).

[vii] R.L. Cheu, and S.G. Ritchie, *Neural network model for automated detection of lane-blocking freeway incidents*. Proc. Int. Conference on Advanced Technologies in Transportation and Traffic Management, Singapore, 245-252 (1994).

[viii] S.G. Ritchie, R.L. Cheu, and S.I. Khan, *A comprehensive system for incident detection on freeways and arterials,* Proc. Annual Meeting of ITS America - Intelligent

transportation: serving the user through deployment, Washington D.C., March, 1, 617-622 (1995).
[ix] J.G. Hunt, and G.D. Lyons, *Modelling of dual carriageway lane changing using neural networks*, Transportation Research C, 2(4), 231-245 (1994).
[x] P.D. Pant, and P. Balakrishnan, *Neural Network For Gap Acceptance at Stop-Controlled Intersections*, ASCE Journal of Transportation Engineering, 120 (3), 432-446 (1994).
[xi] D.A. Pomerleau, *Progress in Neural Network-based Vision for Autonomous Robot Driving*, Proc. Intelligent Vehicle Symposium, Detroit, MI, 391-396 (1992).
[xii] G.D. Lyons, *Calibration and validation of a neural network driver decision model*, Traffic Engineering + Control, 36(1), 10-15 (1995).
[xiii] T. Nakatsuji, et al , *Artificial intelligence approach for optimising traffic signal timings on urban road network.* Proc. 5th VNIS Conference, Yokohama, Japan, 199-202 (1994).
[xiv] Wu Wei and Mingjun Wang, *Traffic Signal Control using Fuzzy and Neural Network*, 8th International Conference on Neural Information Processing ICONIP, Shanghai, China (2001)
[xv] C.L. Wan, and K.W. Dickinson, *Computer vision and neural networks for traffic monitoring*, Proc. 6th International Conference on Road Traffic Monitoring and Control, IEE, London, 143-147 (1992).
[xvi] D. Montana and S. Czerwinski, *Evolving Control Laws for a Network of Traffic Signals*, Proceedings of the First Annual Genetic Programming Conference (1996)
[xvii] D.A. Roozemond, *Using autonomous intelligent agents for urban traffic control systems,* 6th World Congress on intelligent transport systems, Toronto, Canada (1999)
[xviii] D. A. Roozemond and J.L.H. Rogier, *Agent controlled traffic lights*, Proc. ESIT2000, Aachen, Germany (2000)
[xix] J. Z. Hernández, S. Ossowski and A. García-Serrano, *On Multiagent Co-Ordination Architectures: A Traffic Management Case Study*, HICSS 34, Maui, Hawaii (2001)
[xx] A. Vogel, C. Goerick, W. von Seelen, *Evolutionary Algorithms for Optimizing Traffic Signal Operation*, Proc. ESIT2000, Aachen, Germany (2000)
[xxi] M. Annunziato, I. Bertini, M. Lucchetti, S. Pizzuti, *Evolving Weights and Transfer Functions in Feed Forward Neural Networks*, Proc. EUNITE2003, Oulu, Finland (2003)
[xxii] X. Yao, *Evolving Artificial Neural Networks*, in Proceedings of the IEEE, 87(9): 1423-1447 (1999)
[xxiii] M. Annunziato, M. Lucchetti, S. Pizzuti, *Adaptive Systems and Evolutionary Neural Networks : a Survey*, Proc. EUNITE2002, Albufeira, Portugal (2002)
[xxiv] H.D.I.Abarbanel, *Analysis of Observed Chaotic Data*, Springer Verlag, New York, NY (1996)
[xxv] M. Annunziato, H.D.I. Abarbanel, *Non Linear Dynamics for Classification of Multiphase Flow Regimes*, Int. Conf. Soft Computing, SOCO, Genova, Italy (1999)
[xxvi] M. Annunziato, S. Pizzuti, *Adaptive Parameterisation of Evolutionary Algorithms and Chaotic Population Dynamics*, International Journal of Knowledge-Based Intelligent Engineering Systems, 170-177, ISSN: 1327-2314 (2002)