A Smart-Adaptive-System based on Evolutionary Computation and Neural Networks for the on-line short-term urban traffic prediction

M. Annunziato, S. Pizzuti
E.N.E.A. – Environment New technologies Energy Agency
'Casaccia' R.C. –Via Anguillarese 301, 00060 Rome Italy
Phone: +39-0630484411, Fax: +39-0630484811
e-mail:{mauro.annunziato, stefano.pizzuti}@casaccia.enea.it

ABSTRACT: in this paper we show a *Smart-Adaptive-System* (SAS) for the real time short-term urban traffic forecast. The system combines evolutionary computation and neural networks by optimising on-line the weights of feed-forward neural networks when applied to short-term (24 and 60 minutes) urban traffic prediction. We compare the proposed on-line approach with the classical off-line back-propagation algorithm. Results are very promising and show the effectiveness of the proposed SAS in order to get predictive neural models capable to dynamically adapt to environmental changes.

KEYWORDS: smart adaptive systems, evolutionary neural networks, on-line prediction, urban traffic forecast

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 [11][12][19][31][33][8], incident detection [10][27][28], driver behaviour/vehicle control [17][18][25][26], traffic control [24][36][21] and traffic monitoring [35]. In nearly all of the applications reviewed the Back-Propagation (BP) 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 [22][29][30], traffic management [16], traffic signal operation [34]. Moreover evolutionary computation techniques have already been applied to solve optimisation problems in non-stationary environments showing interesting results [7][13][32]. Furthermore the training and design of ANN using evolutionary algorithms is not new and considerable research, since [15][20][23][9] up to [37][14] and [3], on this topic has already been carried out giving rise to a branch of ANN known as *Evolutionary Neural Networks (ENN)* [5].
These ideas inspired this and the preliminary work proposed in [4] where ENN have been for the first time applied to the problem of short-term (20 minutes) urban traffic flow forecast in real time.
The main contribution of this paper is the application of a *Smart-Adaptive-System (SAS)* based on ENN to the on-line traffic prediction. In fact the proposed system can be defined as '*smart*' because it combines evolutionary computation and neural networks and '*adaptive*' because it performs a real time adaptation to environmental changes (changes in the traffic conditions). In this work we applied such a system to 24 and 60 minutes predictions in order to carry out on-line adaptive forecast models. These kinds of models are very important in order to develop on-line traffic regulation systems and to perform an early detection of potential critical situations.

DATA ANALISYS AND NEURAL SET-UP

The input of the neural model was chosen by applying the non-linear dynamic analysis methodology to the traffic flow signal. The detailed explanation of such an approach is out of the scope of the present paper and the reader interested in the topic can refer to [1][2]. However the basic idea is to use *n* samples of the signal x(t), x(t-T),…,x(t-*n*T) where T is the characteristic time needed to preserve the required information about the dynamics. The choice of the optimal *n* and T depends on the signal itself.

In this work the signal represents the traffic flow rate of the last hour with data sampled every 6 minutes. The analysis showed the optimal T to be in the range between 10-15 minutes and *n* to be in the range 6-10. We set T=12 (which is exactly two samples) and *n*=8.
In this way the final neural architecture (Figure 1:) was chosen to have 8 input neurons, 3 hidden neurons and 1 output neuron.

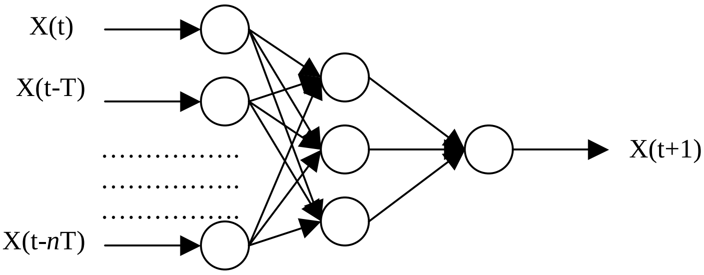

Figure 1: Neural topology (*n*=8, T=12)

THE SMART ADAPTIVE SYSTEM ENVIRONMENT

In this paper we show how evolutionary algorithms can be applied to the on-line weights optimisation of feed-forward neural networks and we report results when applied to short term (24 and 60 minutes) urban traffic prediction. Moreover we compare these methods with the classical Back-Propagation (BP) algorithm used to train ANN. The on-line optimisation acts as follows (Figure 2:). The basic concept consists in the realization of an artificial environment running in parallel with the process and that synchronously communicates with it. We suppose to always measure from the process the data which continuously update a data set, representing the objective of our optimisation, since in this situation performance is defined as the capability of each neural network of reconstructing the underlying dataset. The update strategy is a first-in-first-out (FIFO) policy since we want to make our prediction on a travelling window whose length depends on the problem. In this situation every time the data set changes then a new network is dynamically found, therefore we have an evolutionary neural model capable to adapt and to follow in real-time the process evolution. In this way the evolutionary environment continuously provides the best neural network, corresponding to the fittest individual, and the related traffic flow rate prediction.

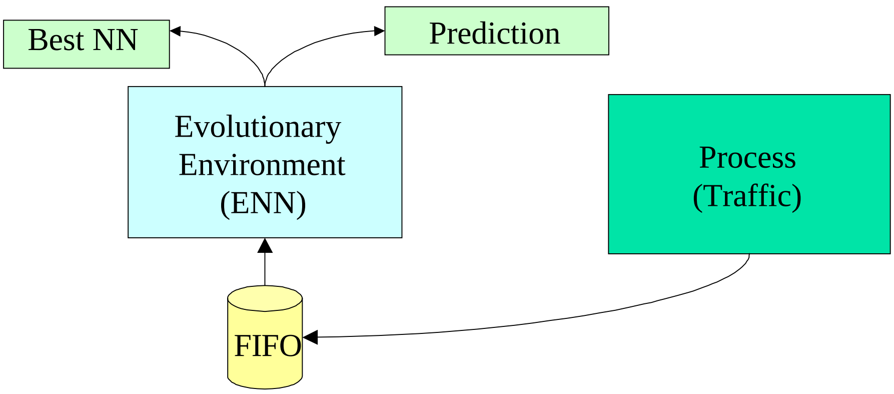

Figure 2: Layout of the on-line system

In this context each individual of the evolutionary environment represents a feed forward neural network, in competition with the others by means of their fitness, having as genotype the synaptic weights. Fitness is measured referring to the global error in modelling the training database with the following formula:

$$\text{Fitness} = 1 - \text{RMSE} \quad (1)$$

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

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

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 most performing neural network). The algorithm we used to develop ENN is *Chaotic Populations* (CP).

THE 'CHAOTIC POPULATIONS' ALGORITHM

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 stabilize 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, haploid and diploid 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 (diploid reproduction or competition) will start, else haploid reproduction of the current individual might occur. Diploid 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 the population size increases of two new elements. Haploid 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 diploid 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 [6].

EXPERIMENTAL RESULTS

Experimentation concerned the 24 and 60 minutes prediction of traffic flow rates using ANN, off-line trained with the BP algorithm, and ENN on-line trained. The goal is to dynamically optimise the weights of a neural network structured with 8 input nodes, see paragraph 2, three hidden nodes and one output node (the traffic flow rate forecast). In all situations we used as transfer functions for all the nodes the classic sigmoid.

$$y = 1/(1+e^{-x}) \quad (3)$$

The data set consisted of one week observations, 1650 real data sampled every 6 minutes representing the traffic flow rate of the last hour, of four different significant points placed in the town of Terni, a small city about 100 km away from Rome with about 100000 citizens and with a network of sensors composed by 87 measurement points (Figure 3:). The first bunch of experiments concerned the off-line optimisation with the classical BP algorithm (ANN). In such an experimentation one million cycles were set. The second group of tests involved the on-line optimisation (Figure 2:). This stage was carried out using a one hour travelling window, representing the memory of the past, 50 evolutionary generations, corresponding to the time needed to adapt to the new situation, and a maximum population of 50 units. The CP evolutionary algorithm [6] doesn't need any further setting. In order to compare the two approaches the data set was partitioned into two sets of respectively 1500 points (the training set) and 150 points (the testing set). In the off-line situation the neural network was fully trained with the training set and afterwards the resulting model was directly tested on the whole testing set. In the on-line situation the evolutionary algorithm iteratively adapts the neural model during the travelling window and then the prediction of the next point is made.

Figure 3: Layout of the city of Terni with the locations of the sensors (circles and pentagons)

The next two tables show the averaged results, according to (2), of the four considered sensors, while in tables III to X the results of each sensor are reported. Finally in Figure 4: a graph comparing the real forecast capability between ANN and ENN is shown.

	Training	Testing
ANN (BP trained)	0.040	0.063
ENN (on-line evolved)	0.033	0.054

Table I:. Average RMSE (2) results of the 24 minutes prediction

	Training	Testing
ANN (BP trained)	0.061	0.169
ENN (on-line evolved)	0.036	**0.068**

Table II:. Average RMSE (2) results of the 60 minutes prediction

24 MINUTES PREDICTION

	Training	Testing
ANN (BP trained)	0.04	0.06
ENN (on-line evolved)	0.03	0.04

Table III:. Sensor 1 RMSE (2) results of the 24 minutes prediction

	Training	Testing
ANN (BP trained)	0.036	0.065
ENN (on-line evolved)	0.030	0.058

Table IV:. Sensor 2 RMSE (2) results of the 24 minutes prediction

	Training	Testing
ANN (BP trained)	0.046	0.06
ENN (on-line evolved)	0.038	0.06

Table V:. Sensor 3 RMSE (2) results of the 24 minutes prediction

	Training	Testing
ANN (BP trained)	0.037	0.065
ENN (on-line evolved)	0.032	0.059

Table VI:. Sensor 4 RMSE (2) results of the 24 minutes prediction

60 MINUTES PREDICTION

	Training	Testing
ANN (BP trained)	0.059	0.118
ENN (on-line evolved)	0.035	0.064

Table VII:. Sensor 1 RMSE (2) results of the 60 minutes prediction

	Training	Testing
ANN (BP trained)	0.059	0.216
ENN (on-line evolved)	0.035	0.066

Table VIII:. Sensor 2 RMSE (2) results of the 60 minutes prediction

	Training	Testing
ANN (BP trained)	0.071	0.180
ENN (on-line evolved)	0.039	0.078

Table IX:. Sensor 3 RMSE (2) results of the 60 minutes prediction

	Training	Testing
ANN (BP trained)	0.054	0.163
ENN (on-line evolved)	0.035	0.064

Table X:. Sensor 4 RMSE (2) results of the 60 minutes prediction

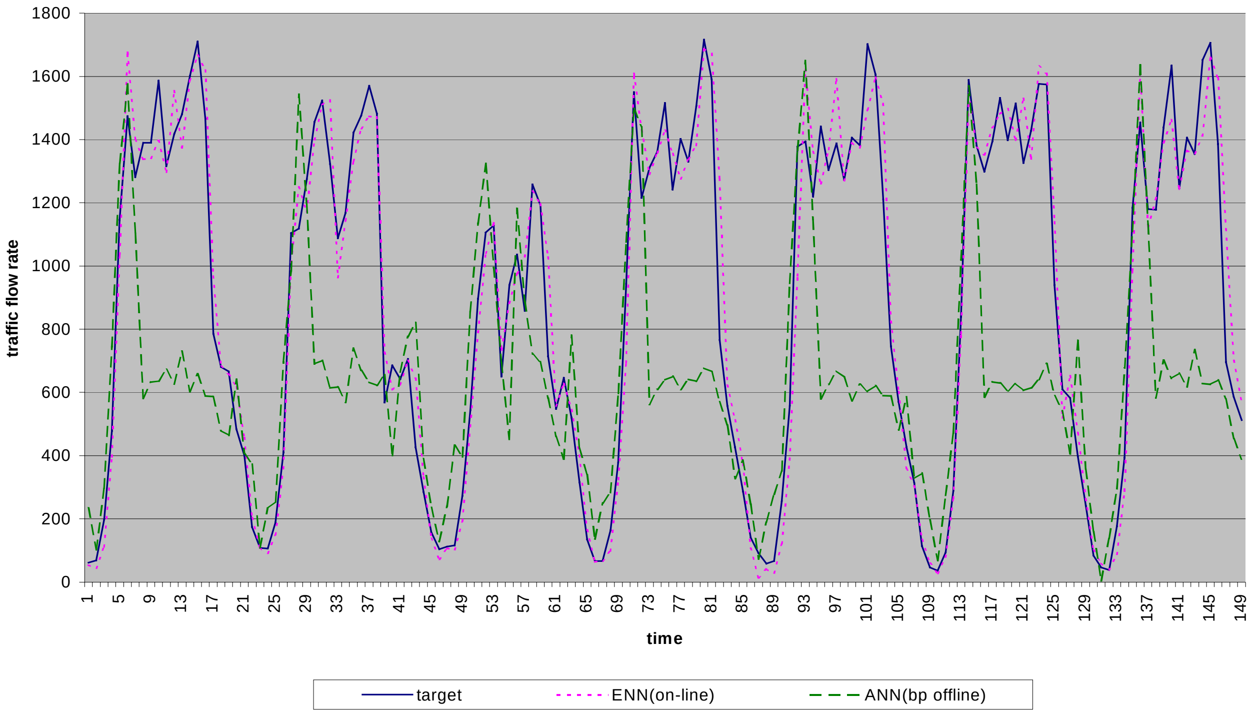

Figure 4: Comparison of testing results for sensor 2 (60 min. prediction)

These results show the effectiveness of the proposed approach to solve the problem of on-line traffic flow rate prediction. The 24 minutes forecast (Table I: and tables III to VI) shows the proposed methodology performs slightly better than ANN BP trained. The 60 minutes prediction (Table II: and tables VII to X) shows a remarkable improvement, with respect to traditional ANN, by cutting the average accuracy error of the real forecast capability from 0.169 to 0.068.

ACKNOWLEDGMENT

This work has been carried out in the framework of the *MERLINO* project supported by E.N.E.A.

CONCLUSION

In this paper we showed a Smart-Adaptive-System (SAS) based on Evolutionary Neural Networks for the real time short-term urban traffic forecast. The proposed system can be defined to be a real SAS because it combines evolutionary computation and neural networks in order to optimise on-line the weights of feed-forward neural networks and because it performs a real time adaptation to changing conditions. In this work we applied such a system to 24 and 60 minutes urban traffic flow rate predictions in order to carry out real time adaptive forecast models.
We compared the proposed real-time approach with the classical one based on Artificial Neural Networks (ANN) off-line trained with the Back-Propagation (BP) algorithm. Experimental results were very promising because the proposed methodology showed a significant improvement with respect to ANN. In particular the most remarkable result concerned the 1-hour prediction where the average root mean squared prediction error is cut off from 16.9% to 6.8%. These results show the success of the proposed SAS methodology in order to get predictive neural models capable to dynamically adapt to environmental changes (changes in the traffic conditions) overcoming the drawbacks imposed by traditional off-line gradient based methodologies (the BP algorithm).
Future work will concern the experimentation on longer predictions (6, 12 and 24 hours) in order to understand the prediction horizon of the proposed system.

REFERENCES

[1] Abarbanel H.D.I., 1996, Analysis of Observed Chaotic Data, Springer Verlag, New York, NY

[2] Annunziato M., Abarbanel H.D.I., 1999, Non Linear Dynamics for Classification of Multiphase Flow Regimes, Proc. of Int. Conf. Soft Computing, SOCO, Genoa, Italy

[3] Annunziato M., Bertini I., Lucchetti M., Pizzuti S., 2003, Evolving Weights and Transfer Functions in Feed Forward Neural Networks, Proc. EUNITE2003, Oulu, Finland

[4] Annunziato M., Bertini I., Pannicelli A., Pizzuti S., 2003, Evolutionary feed-forward neural networks for traffic prediction, Proc. of EUROGEN03, Barcelona, Spain

[5] Annunziato M., Lucchetti M., Pizzuti S., 2002, Adaptive Systems and Evolutionary Neural Networks: a Survey, Proc. EUNITE2002, Albufeira, Portugal

[6] Annunziato M., Pizzuti S., 2002, Adaptive Parameterisation of Evolutionary Algorithms and Chaotic Population Dynamics, International Journal of Knowledge-Based Intelligent Engineering Systems, 170-177

[7] Branke J., 1999, Evolutionary Approaches to Dynamic Optimisation Problems – A Survey, GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, A. Wu (ed.), 134-137

[8] Canca D., Larrañeta J., Lozano S. and Onieva L., 1997, Traffic intensity forecast in urban networks using a multilayer perceptron. Joint International Meeting EURO XV - INFORMS XXXIV. Barcelona, Spain

[9] Chalmers, D., 1991, The evolution of learning: An experiment in genetic connectionism. In: D. Touretzky et al. (Eds.), Connectionist Models: Proceedings of the 1990 Summer School. San Mateo, Morgan Kaufmann.

[10]Cheu R.L. and Ritchie S.G., 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

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

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

[13]Grefenstette J. J., 1992, Genetic algorithms for changing environments, R. Maenner and B. Manderick, eds, Parallel Problem Solving from Nature 2, North Holland, 137-144

[14]Hansen, J.V., McDonald, J.B. and Nelson, R.D., 1999, Time Series Prediction with Genetic Algorithm Designed Neural Networks: An Experimental Comparison with Modern Statistical Models. Computational Intelligence, 15(3), pp. 171-184

[15]Harp, S., Samad, T. and Guha, A., 1989, Towards the Genetic Synthesis of neural networks. In: D. Shaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms. San Mateo, Morgan Kaufmann

[16]Hernández J. Z., Ossowski S. and García-Serrano A., 2001, On Multiagent Co-Ordination Architectures: A Traffic Management Case Study, HICSS 34, Maui, Hawaii, USA

[17]Hunt J.G. and Lyons G.D., 1994, Modelling of dual carriageway lane changing using neural networks, Transportation Research C, 2(4), 231-245

[18]Lyons G.D., 1995, Calibration and validation of a neural network driver decision model, Traffic Engineering + Control, 36(1), 10-15

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

[20]Miller, G., Todd, P. and Hedge, S., 1989, Designing neural networks using genetic algorithms. In: D. Shaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms. San Mateo, Morgan Kaufmann

[21]Minesh I. Patel, N. Ranganathan, 1996, An Intelligent System Architecture for Urban Traffic Control Applications, 8th IEEE Symposium on Parallel and Distributed Processing (SPDP '96), New Orleans, LA, USA

[22]Montana D. and Czerwinski S., 1996, Evolving Control Laws for a Network of Traffic Signals, Proceedings of the First Annual Genetic Programming Conference

[23]Montana, D. and Davis, L. , 1989, Training feedforward networks using genetic algorithms. In: N. Sridhara (Ed.), Proceedings of Eleventh International Joint Conference on Artificial Intelligence, pp. 762-767

[24]Nakatsuji T., et al, 1994, Artificial intelligence approach for optimising traffic signal timings on urban road network. Proc. 5th VNIS Conference, Yokohama, Japan, 199-202

[25]Pant P.D. and Balakrishnan P., 1994, Neural Network For Gap Acceptance at Stop-Controlled Intersections, ASCE Journal of Transportation Engineering, 120 (3), 432-446

[26]Pomerleau D.A., 1992, Progress in Neural Network-based Vision for Autonomous Robot Driving, Proc. Intelligent Vehicle Symposium, Detroit, MI, 391-396

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

[28]Ritchie S.G., Cheu R.L., and Khan S.I., 1995, 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

[29]Roozemond D. A. and Rogier J.L.H., 2000, Agent controlled traffic lights, Proc. ESIT2000, Aachen, Germany

[30]Roozemond D.A., 1999, Using autonomous intelligent agents for urban traffic control systems, 6th World Congress on intelligent transport systems, Toronto, Canada

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

[32]Trojanowski K. and Michalewicz Z., 1999, "Evolutionary Algorithms for Non-Stationary Environments'", Proceedings of 8th Workshop: Intelligent Information systems, Ustron, Poland, ICS PAS Press, pp 229-240

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

[34]Vogel A., Goerick C., von Seelen W., 2000, Evolutionary Algorithms for Optimizing Traffic Signal Operation, Proc. ESIT2000, Aachen, Germany

[35]Wan C.L. and Dickinson K.W., 1992, Computer vision and neural networks for traffic monitoring, Proc. 6th International Conference on Road Traffic Monitoring and Control, IEE, London, 143-147

[36]Wei W. and Wang M., 2001, Traffic Signal Control using Fuzzy and Neural Network, 8th International Conference on Neural Information Processing ICONIP, Shanghai, China

[37]Yao X., 1999, Evolving Artificial Neural Networks, in Proceedings of the IEEE, 87(9): 1423-1447