ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence
and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

Combining Back-Propagation and Genetic Algorithms to Train Neural Networks for Start-Up Time Modeling in Combined Cycle Power Plants

I. Bertini[1], M. De Felice[1,2], S. Pizzuti[1]

1- ENEA (Italian Energy New Technology and sustainable Economic development Agency)
Via Anguillarese 301, 00123 Rome, Italy
2- University of Rome ``Roma Tre'', Department of Informatics and Automation
Via della Vasca Navale 79, 00146 Rome, Italy

Abstract. This paper presents a neural networks based approach in order to estimate the start-up time of turbine based power plants. Neural networks are trained with a hybrid approach, indeed we combine the Back-Propagation (BP) algorithm and the Simple Genetic Algorithm (GA) in order to effectively train neural networks in such a way that the BP algorithm initializes a few individuals of the GA's population. Experiments have been performed over a big amount of data and results have shown a remarkable improvement in accuracy compared to the single traditional methods.

1 Introduction

Combined Cycles (CC) power plants are highly complex systems and in general most studies on such plants are based on simulations because with the availability of high powerful processors and advanced numerical solutions there is a great opportunity to develop high performance simulators for modeling energy systems. At present the main drawback of such simulators is that every single simulation needs a certain amount of time (up to 10 minutes) and this is a very hard problem when dealing with designing and optimizing such systems. Therefore, there is a strong need for good estimating tools in order to use simulators only in few cases. Artificial Neural Networks (ANNs) [1][2] have proved [3][4] to be powerful tools to solve complex modelling problems for non-linear systems and an usual 3 layered MLP neural network, with m inputs and n outputs, can approximate any non-linear mapping from R^m to R^n using an appropriate number of neurons in the hidden layer. Due to this approximation and classification ability, neural networks can also be successfully used for CC applications and particularly ANNs have been mainly applied to fault detection [5][6], diagnosis [7][8] and control [9]. Back-Propagation (BP)[10] and Genetic Algorithms (GA)[11][12] are among the most used algorithms to train ANNs. Despite the success of the algorithms, each has its own drawback [13][14]. As a deterministic gradient-descent algorithm and stochastic technique respectively, there might exist a balance between their advantages and disadvantages. Among the possible combinations of the two methods the most interesting ones are basically three: the first one uses the BP as initialisation for the GA (BPGA)[15][16], the second uses the GA as initialisation for the BP (GABP) [17] and the third one uses the

GA for best training set generation for the BP [18]. In this context, we used a BPGA method in order to train ANN as CC models. The presented work does not actually make any significant advancement with respect to state-of-the-art in the theoretical knowledge of such training methods. The main novelty lies in the underlying application because one of the most critical operation in CC management is the start-up optimization. This problem is very hard and it is studied by means of heavy complex simulators. Thus, having fast, robust and accurate CC models is vital for this task and that is why we approached this problem through ANNs.

2 The combined cycle power plants

Gas and steam turbines are an established technology available in sizes ranging from several hundred kilowatts to over several hundred megawatts. Industrial turbines produce high quality heat that can be used for industrial or district heating steam requirements. Alternatively, this high temperature heat can be recovered to improve the efficiency of power generation or used to generate steam and drive a steam turbine in a combined-cycle plant. Therefore, industrial turbines can be used in a variety of configurations:

- Simple cycle (SC) operation which is a single gas turbine producing power only
- Combined heat and power (CHP) operation which is a simple cycle gas turbine with a heat recovery heat exchanger which recovers the heat in the turbine exhaust and converts it to useful thermal energy usually in the form of steam or hot water
- Combined cycle (CC) operation in which high pressure steam is generated from recovered exhaust heat and used to create additional power using a steam turbine (fig.1)

The last combination produces electricity more efficiently than either gas or steam turbine alone because it performs a very good ratio of transformed electrical power per CO_2 emission. CC plants are highly complex systems but being available highly powerful processors and advanced numerical solutions, there is a great opportunity to develop high performance simulators for modeling energy systems in order to consider various aspects of the system. In particular, one of the most studied problems of CC operation is the start-up optimization with the goal to minimize the start-up time. Most studies on it are based on complex simulators, therefore any optimization system would need a reliable but fast start-up model in order to use the heavy simulators only in few cases. The start-up scheduling is as follows (fig.1). From zero to time t_0 (about 1200 sec) the rotor engine velocity of the gas turbine is set to 3000 rpm. From time t_0 to t_1 the power load is set to 10 MW and then the machine keeps this regime up to time t_2. All this initial sequence is fixed. From time t_2 to t_3 (about 3600 sec) the machine must achieve a new power load set point which has to be set optimal and then the machine has to keep this regime up to time t_4. The time lag $t_4 – t_3$ is variable and during this interval the steam turbine starts with the rotor reaching the desired velocity. Then the turbines have to reach at time t_5 the normal power load regime (270 MW for the gas turbine) according to two load gradients which are variable depending on the machine.

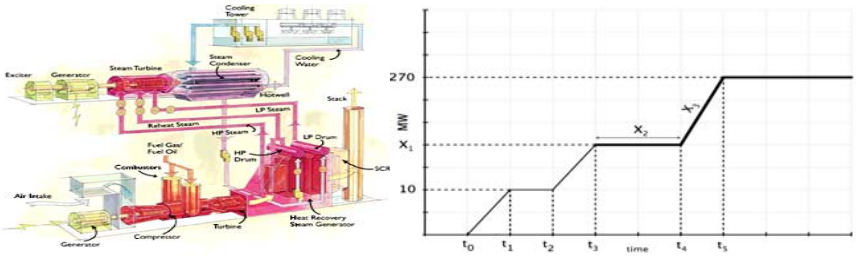

Fig. 1: combined cycle power plant (left) and start-up operations (right)

Variable	Meaning	Operating range	Unit measure
X_1	Intermediate power load set point	[20, 120]	MW
X_2	Intermediate waiting time	[7500, 10000]	Sec
X_3	Gas turbine load gradient	[0.01, 0.2]	MW/s
X_4	Steam turbine load gradient	[0.01, 0.2]	%/s

Table 1: start-up variables.

Therefore, these variables will also be the input of the neural model.

3 The BPGA combination

The BP is a gradient-descent algorithm used to find the neural network weights so that the estimate error function reaches the minimum. It can be proved that BP can reach the extreme within a limited number of epochs. The merits of BP are that the adjustment of weights is always towards the descending direction of the error function and that only some local information is needed. On the other hand, BP has its disadvantages too. For example, the error curve is generally so complex that there are a lot of local minima making the convergence of the algorithm very sensitive to the initial values. GA is a parallel stochastic optimization algorithms. As compared with BP, GA is more qualified for neural networks if only the requirement of a global searching is considered. However, the price one pays for GA is the slowness which is mainly due to the random initialization of the genes and to the slow but crucial exploration mechanisms employed, which has three basic arithmetic operators: reproduction, crossover and mutation. Another shortcoming of GA is that the method is not theoretically perfect and it cannot ensure convergence and achievement of the optimum. From the analysis above, it is easy to observe the complementarity between BP and GA. The proposed BPGA can learn from their strong points to offset their weaknesses. It can be typically applied to MLP (Multi-Layer Perceptron) supposed that the activation functions of the hidden layers and the output layer are all sigmoidal.

In the proposed BPGA, BP is firstly used to train several neural networks, a small fraction (about 5-10%) of the total GA's population size, for approximately 10^6 cycles with no early stopping criterion. Then, the weights of the BP computations are encoded into several chromosomes of the GA's initial population together with other randomly generated chromosomes. From this initial population, subsequent populations are being computed for a small number of fitness requests (about 1000). The GA we implemented is Holland's Simple Genetic Algorithm with 1-elitism.

4 Experimentation

The experimentation concerned the CC start-up time estimation which is one of the most important parameters in the overall CC start-up optimization. The inputs of the neural networks are those reported in table 1, therefore the neural models have four inputs and one output (the start-up time length in seconds). All the neural networks are feed-forward MLP models with four hidden neurons and sigmoidal activation function for all the hidden and output nodes. The data set is made of 14641 simulator generated points* (since no real data were available yet) partitioned as 70% for training and 30% for testing and in the following table we report the main features of the training set.

	Input1	Input2	Input3	Input4	Output
Min	20	7500	0.017	0.017	11702
Max	120	10000	0.217	0.367	29416
Avg	70	8750	0.117	0.192	15601
Std Dev.	31.6	790.6	0.063	0.119	3318

Table 2: training set info.

In the proposed work we compared the performances of the neural models trained with three different methods : Back-Propagation (BP), Genetic Algorithms (GA) and the composition of the two as described in par.3 (BPGA). In the following tables we report the main settings of the methods.

Algorithm	Learning Rate	Momentum	Performance Requests
BP	0.3	0.9	10^6

Table 3: BP settings.

Algorithm	Population size	Crossover probability	Mutation probability	Performance Requests
GA	50	0.9	0.1	10^6

Table 4: GA settings.

The BPGA method has the same settings of the single methods except for the GA fitness requests which have been set to 1000. Therefore, the total number of BPGA fitness requests is 10^6 (BP) +1000 (GA) and the number of chromosomes BP-initialized is set to 5 (10% of the total population). In the following table we report the experimental results (averaged over ten runs with standard deviations in brackets) on the validation set. In the table we compare the three training methods and the improvement of the best method (BPGA) with respect to the others. The mean absolute error is reported in seconds.

* Reference: ivo.torre@aen.ansaldo.it

	GA	BP	BPGA	BPGA vs. GA	BPGA vs. BP
Mean absolute error	214 (±16)	178 (±23)	118 (±3)	-45%	-34%
Mean relative error	1.39% (±0.10%)	1.17% (±0.15%)	0.78% (±0.02%)	-44%	-33%
Normalised RMSE	1.89% (±0.16%)	1.55% (±0.2%)	1.02% (±0.02%)	-46%	-34%

Table 4: Experimental results (testing).

Experimentation shows the effectiveness of the proposed BPGA approach. In fact, it clearly outperforms the other methods and it is interesting to point out that the BPGA average standard deviation is very little, highlighting the robustness of the method. This is very important because when dealing with model based optimization problems (like the CC start-up) the more accurate and reliable the model is the more successful the optimization is. The reasons for this are mainly due to the fact that the searching domain of the GA is cut down by the BP initialization and that the GA's parallel optimization gets the BP out of the local minima which gets stuck in.

5 Conclusions

In this paper we tackled the issue of start-up time modeling in combined cycle power plants (CC) since it is one of the most important parameters when optimizing the start-up operations of such plants. To solve this problem we proposed a hybrid approach based on soft computing techniques. Indeed, we combined the Back-Propagation algorithm and the Simple Genetic Algorithm (BPGA) in order to effectively train neural networks in such a way that the BP algorithm initializes several individuals of the GA's initial population. Experiments were performed over a big amount of artificial data and results showed a remarkable improvement in accuracy compared to the single (BP and GA) traditional training methods. In particular, with respect to the BP algorithm the average modeling error is cut of about one third. Moreover, The BPGA method showed very high robustness (very low standard deviation). The reason for this success is due to the fact that the BPGA algorithm combines BP and GA in such a way that the virtues of the single methods are enhanced. Indeed, the BP is first applied so that the searching domain of GA is trimmed down, reducing therefore the GA convergence time, and then the parallel GA optimization extricates the BP from the local minima which plunges into. Therefore, the main advantage of this method is that we have a non linear interpolation tool capable to provide a reliable start-up time estimate, and this is a critical point to effectively optimize the start-up operations without the intense use of heavy simulations. Future work will first concern the validation of these results on real data, and then we’ll compare the proposed method to more supervised learning techniques and architectures, like Radial Basis Function Networks and Support Vector Machines, as well as different evolutionary algorithms and ensembling methods. Moreover, the proposed technique is going to be applied to model other important CC parameters like gas consumption, pollutant emissions, produced energy and thermal stress.

References

[1] Arbib M.A.: The Handbook of Brain Theory and Neural Networks. The MIT Press, Cambridge (MA),1995

[2] Haykin S.: Neural Networks, a comprehensive foundation (2nd edition). Prentice Hall, New Jersey,1999

[3] Kosko B. : Neural networks and fuzzy systems. Prentice-Hall, Englewood Cliffs (NJ) ,1992

[4] Nikravesh M., Farell A.E., Stanford T.G. : Model identification of non linear time variant processes via artificial neural network. Computes and Chemical Engineering, Vol. 20 No. 11, 1277-1290,1996

[5] Korbicz, J., Patan, K., Obuchowitcz, O.: Dynamic neural networks for process modelling in fault detection and isolation systems. Journal of Applied Mathematics and Computer Science, no 3, vol. 9, pp. 519-546,1999

[6] Marcu T., Mirea, L. , Frank, P.M. : Development of dynamic neural networks with application to observer-based fault detection and isolation. Journal of Applied Mathematics and Computer Science, no 3, vol. 9, pp. 547-570 ,1999

[7] Ogaji S.,. Singh R : Neural network technique for gas turbine fault diagnosis. International journal of engineering intelligent systems for electrical engineering and communications, ISSN 0969-1170, Vol. 10, Nº 4, pp. 209-214,2002

[8] Xu Qingyang, Meng Xianyao, Wang Ning : Gas turbine fault diagnosis based on ART2 Neural Network. In: 7th World Congress on Intelligent Control and Automation, 2008

[9] Nabney I. T. and Cressy D. C., Neural network control of a gas turbine, Neural Computing & Applications, Volume 4, Number 4 , Springer London, 1996

[10] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. : Learning representations by backpropagating errors. Nature 323, 533-536 ,1986

[11] Holland J. H. : Adaptation in natural and artificial systems, MIT Press, Cambridge (MA) ,1975

[12] Goldberg, D. E. : Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company,1989

[13] Farag W. A., Quintana V. H. and Lambert-Torres G., "Genetic Algorithms and Back-Propagation: a Comparative Study". Canadian Conference on Electrical and Computer Engineering Proceedings of the 1998 llth Canadian Conference on Electrical and Computer Engineering, CCECE. Part 1 (of 2) pp.93-96, 1998

[14] McInerney M., Dhawan A.P., "Use of Genetic Algorithms with Back Propagation in Training of Feed-Forward Neural Networks," IEEE International Conference on Neural Networks, pp. 203-208,1993

[15] Lu C., Shi B. : Hybrid Back-Propagation/Genetic Algorithm for Feedforward Neural Networks. In ICSP2000

[16] Zhang M., Ciesielski V. : Using Back Propagation Algorithm and Genetic Algorithms to Train and Refine Neural Networks for Object Detection. In Computer Science Postgraduate Student Conference, Melbourne,1998

[17] Khan A.U., Bandopadhyaya T.K., Sharma S. : Genetic Algorithm Based Backpropagation Neural Network Performs better than Backpropagation neural Network in Stock Rates Prediction. Intl Journal of Computer Science and Network Security, Vol.8 No.7, 162-166,2008

[18] Sadeq A.M., Wahdan A-M A., Mahdi H.M.K. : Genetic Algorithms and its use with back-propagation network. In AIN Shams University Scientific Bullettin, vol.35 No.3, 337-348,2000