

Power Signal Predictions by Several Neural Models

Rey-Chue Hwang¹ Yu-Ju Chen² Shang-Jen Chuang³ Huang-Chu Huang³ Chuo-Yean Chang⁴

¹Electrical Engineering
Department
I-Shou University
Kaohsiung, Taiwan, 840
rchwang@isu.edu.tw

²Information Management
Department
⁴Electrical Engineering
Department
Cheng-Shiu University
Kaohsiung, Taiwan, 833
yujuchen@csu.edu.tw

³Electric Communication
Department
National Kaohsiung
Marine University
Kaohsiung, Taiwan, 814

Abstract:

In this paper, a comparison study for the non-stationary power signal prediction by using several neural models is presented. A reliable and accurate neural forecasting model is trying to be found and concluded. This study is expected to provide some suggestions and could be treated as a reference for the researchers in this area. All simulations are executed by using several neural networks with different structures and then the results are reported.

Keywords: non-stationary, power signal, prediction, neural models

1. Introduction

Last two decades, neural network (NN) has been widely applied into different areas due to its powerful nonlinear mapping property and learning capability. In most of NN's studies, system identification and signal prediction could be concluded as the most popular applications. Through a simple learning process in the training examples, the mathematic mapping model could be automatically developed by NN to capture the very complex and nonlinear relationships between input/output pairs of training data.

Recently, NN technique has been widely used in the studies and applications of power load forecasting. In general, load forecasting is a way adopted by utility company in the power business management [1-6]. Accurate power forecasting not only can help company to make the schedules of business running, such as fuel purchasing, unit commitment and maintenance, but also can help company to have a planning for the development of power capacity in the future. Generally, NN model could very precisely execute the forecasting work desired, if it was well-trained and the signal of NN model dealt with is stationary. However, many signal processing problems in real world, the signal data collected

indeed involves many unknown or uncertain factors and disturbances that could heavily affect the behavior of signal. In this case, the signal behaviors are mostly dynamic and varied with time. Therefore, such a type of signal processing could be concluded as the non-stationary signal processing problem. Usually, such an environment of signal information is generally complex and ill defined. It might make NN have an ill learning and then cause NN to have an unsatisfactory performance.

In this paper, several neural models with different structures are used in the power load forecasting studies. From the simulation results, a reliable and accurate neural forecasting model in this field is expected to be found and concluded. All NN models and their structures are reported in Section 2. Section 3 presents our simulations and results, and then a conclusion is made in Section 4.

2. NN Models and Node Structures

In our studies, all NN forecasting models are composed of a four-layered feed-forward fully connected NN. Three types of transfer functions are used in the nodes of NN models respectively, including sigmoid function, modified hyperbolic tangent transfer function and n_s -level sigmoid function. Their math forms are expressed as follows.

$$f(x) = 1 / (1 + \exp(-x)) \quad (2-1)$$

$$f(x) = a(1 - \exp(-bx)) / (1 + \exp(-bx)) \quad (2-2)$$

$$sgm(x) = (1 / n_s) \sum_{r=1}^{n_s} \left(1 / \left(1 + \exp \left(- \left(x - \theta^r \right) \right) \right) \right) \quad (2-3)$$

In the neural models we constructed, each layer is composed of the nodes with same transfer function. In each model, different layers could be composed of different types of nodes. At here, for simplifying the description of the models we proposed, the layer is formed by sigmoid function nodes will be named as NN layer, the layer is formed by modified hyperbolic tangent transfer function node will be named as MNN layer, and the layer is formed by n_s -level sigmoid

function node will be named as QNN layer. Due to the node's structure at each layer, the learning rule of neural models can be described as follows [7-10]. If we denote $S(k)$ as the desired output and $y(k)$ as the actual output of network on the presentation pattern k , then the cost function can be set as

$$E = \sum_k (S(k) - y(k))^2. \quad (2-4)$$

For minimizing E , the weights of NN could be adjusted according to the node's structure.

The layer with sigmoid nodes:
for an output node

$$\begin{aligned} \Delta\omega_{ij}(k) &\propto (S_j(k) - y_j(k)) \\ &*(y_j(k)(1 - y_j(k))x_i(k)) \end{aligned} \quad (2-5)$$

for a hidden layer node

$$\begin{aligned} \Delta\omega_{ij}(k) &\propto \sum_l \delta_l(k)\omega_{jl}(k) \\ &*(y_j(k)(1 - y_j(k))x_i(k)) \end{aligned} \quad (2-6)$$

where, $\delta_l(k)$ is the error term feedback from all nodes in the layer above node j .

The layer with modified hyperbolic tangent nodes:
for an output node

$$\begin{aligned} \Delta\omega_{ij}(k) &\propto (b_j(k)/a_j(k))(S_j(k) - y_j(k)) \\ &*(a_j(k) + y_j(k))(a_j(k) - y_j(k))x_i(k) \end{aligned} \quad (2-7)$$

$$\Delta a_j(k) \propto (S_j(k) - y_j(k))(y_j(k)/a_j(k)) \quad (2-8)$$

$$\begin{aligned} \Delta b_j(k) &\propto (V_j(k)/a_j(k))(S_j(k) - y_j(k)) \\ &*(a_j(k) + y_j(k))(a_j(k) - y_j(k)) \end{aligned} \quad (2-9)$$

where, $V_j(k) = \sum_i \omega_{ij}(k)x_i(k)$,

for a hidden layer node,

$$\begin{aligned} \Delta\omega_{ij}(k) &\propto (b_j(k)/a_j(k))\sum_l \delta_l(k)\omega_{jl}(k) \\ &*(a_j(k) + y_j(k))(a_j(k) - y_j(k))x_i(k) \end{aligned} \quad (2-10)$$

$$\Delta a_j(k) \propto (y_j(k)/a_j(k))\sum_l \delta_l(k)\omega_{jl}(k) \quad (2-11)$$

$$\begin{aligned} \Delta b_j(k) &\propto (V_j(k)/a_j(k))(a_j(k) + y_j(k)) \\ &*(a_j(k) - y_j(k))\sum_l \delta_l(k)\omega_{jl}(k) \end{aligned} \quad (2-12)$$

where, $\delta_l(k)$ is the error term feedback from all nodes in the layer above node j .

In all NN models we constructed, no output layer is composed of the node with n_s -level sigmoid transfer function, therefore, for the hidden layer with n_s -level sigmoid node:

for a hidden layer node,

$$\begin{aligned} \Delta\omega_{ij}(k) &\propto ((1/n_s)\sum_{r=1}^{n_s} y_j^r(k)(1 - y_j^r(k))) \\ &*\sum_l \delta_l(k)\omega_{jl}(k). \end{aligned} \quad (2-13)$$

The quantum interval adjustment:

In each training cycle, we calculate the following outputs for each hidden node j .

$$\bar{h}_j = \sum_{i=0}^{n_i} \omega_{ij} X_i, \quad (2-14)$$

n_i is the number of input signals.

$$\begin{aligned} h_j^r &= \sum_{r=1}^{n_s} \text{sgm}(\bar{h}_j - \theta_j^r), \quad \tilde{h}_j = \frac{1}{n_s} \sum_{r=1}^{n_s} h_j^r, \\ v_j^r &= h_j^r(1 - h_j^r). \end{aligned} \quad (2-15)$$

Take the average values for each class, $\langle \tilde{h}_{j,C_m} \rangle$ and

$$\langle v_{j,C_m}^r \rangle.$$

For m -th class C_m ,

$$\langle \tilde{h}_{j,C_m} \rangle = \frac{1}{|C_m|} \sum_{x_m, x_m \in C_m} \tilde{h}_{j,m}, \quad (2-16)$$

and

$$\langle v_{j,C_m}^r \rangle = \frac{1}{|C_m|} \sum_{x_m, x_m \in C_m} v_{j,m}^r, \quad (2-17)$$

where, $|C_m|$ denotes the cardinality of C_m .

The quantum interval can be adjusted by

$$\begin{aligned} \Delta\theta_j^r &= \alpha_\theta \frac{1}{n_s} * \\ &\sum_{m=1}^{C_{m\theta}} \sum_{x_m \in C_m} (\langle \tilde{h}_{j,C_m} \rangle - \tilde{h}_j) \times (\langle v_{j,C_m}^r \rangle - v_j^r) \end{aligned} \quad (2-18)$$

where, α_θ is the learning rate.

$$\theta_j^r = \theta_j^r + \Delta\theta_j^r. \quad (2-19)$$

In the weight adjustment process, we denote that α is the learning rate and η is the momentum for all NN models. α_a and α_b are the step sizes for the adjustment of parameters a and b in the MNN layer.

3. Simulations

In our simulations, the predictions of one-day-ahead daily total load are studied and simulated. Daily loads and relevant temperatures data from years 1992 to 1996 provided by the Taipower utility company and the Central Weather Bureau are implemented. The data from years 1992 to 1995 are used for model's training and that of year 1996 data is used as a real-line testing. Each neural model with size 12-12-12-1 is used for all simulations. Denote the desired forecasting daily load of day k by $L(k)$. The forecasting value is the incremental amount ($\Delta y(k)$) of power load for next day. Thus, the real forecasting load value is taken by $y(k) = \Delta y(k) + L(k-1)$.

According to our experiences, the following twelve parameters are generally used as inputs for

network training.

$\Delta L(k-1), \Delta L(k-2), \Delta L(k-7)$
:= the increments of load value of days $k-1$,
 $k-2$, and $k-7$, respectively;
 $T_{\max}(k-1), T_{\max}(k-2), T_{\max}(k-7)$
:= maximum temperature of days $k-1$, $k-2$,
and $k-7$, respectively;
 $T_{\min}(k-1), T_{\min}(k-2), T_{\min}(k-7)$
:= minimum temperature of days $k-1$, $k-2$,
and $k-7$, respectively;
 $T_{\max}(k), T_{\min}(k)$
:= forecasted maximum and minimum temperatures
of day k , respectively;
 $Day(k)$
:= day number of day k (Sunday=0.1,
Monday=0.2, ..., Saturday=0.7).

In the model with MNN layer, we set the number of class (C_m) of data 7, i.e. seven days of a week respectively. The learning rate, $\alpha_\theta = 0.0005$ and the number of level $n_s = 10$ are adopted. Actual temperature readings are used for all our simulations. Moreover, the loads of weekday holidays are not included in either training or testing. Mean absolute percentage error (MAPE) over all testing data is taken as a measure of performances.

In our studied, four NN forecasting models, named Model 1, Model 2, Model 3 and Model 4, are studied and simulated. Figure 1 to Figure 4 present the structure of each layer in these four models. The error statistics of daily total load forecasts by using these NN models are listed in Table 1, Table 2, Table 3 and Table 4, respectively. Each model was executed by different learning rates. Through a comprehensive observation to the results shown, it could be obviously found that the better prediction accuracies can be obtained by using Model 2 and Model 4 those models with MNN output layer.

4. Conclusion

In this paper, the non-stationary power signal prediction by using several NN models are studied and presented. From the simulation results, we clearly find that the models with MNN output layer have better performances as compared with other NN models. In addition, the simulation results also show that the performances by Model 1 and Model 3 are highly correlated with the learning rate used. These two models easily plunge into local minima, if an improper learning rate was selected in the training process. Therefore, we could conclude that Model 2 and Model 4 those models with MNN output layer are more reliable and accurate than other models with NN output layer when NN model is used in the

applications of non-stationary signal prediction.

5. Acknowledgment

This work is supported by the National Science Council of Republic of China under contract No. NSC-94-2213-E-214-047.

Reference

- [1] S. T. Chen, D. C. Yu, and A. R. Moghaddamjo, "Weather Sensitive Short-Term Load Forecasting Using Nonfully Connected Artificial Neural Network," *IEEE Transaction on PWRs*, Vol. 7, No. 3, pp. 1098-1105, Aug., 1992.
- [2] C. N. Lu, N. T. Wu, S. Vemuri, "Neural Network Based short-Term Load Forecasting," *IEEE Transaction on PWRs*, Vol. 8, No. 1, pp. 336-342, Feb., 1993.
- [3] A. Kaotanzad, R. C. Hwang, A. Abaye, and D. Maratukulam, "An Adaptive Modular Artificial Neural Network Hourly Load Forecaster and Its Implementation at Electric Utilities," *IEEE Transaction on Power System*, Vol. 10, No. 3, pp. 1716-1722, Aug. 1995.
- [4] R. C. Hwang, H. C. Huang, J. G. Hsieh, "A New Artificial Intelligent Peak Power Load Forecaster Based on Non-fixed Neural Networks", *International Journal of Electrical Power and Energy Systems*, Vol. 24, Issue: 3, pp. 245-250, 2002.
- [5] Y. J. Chen, T. C. Huang, R. C. Hwang, "Power Signal Prediction by Neural Network with A New Fuzzy BP Learning Algorithm," *2002 IEEE International Conference on Industrial Technology*, Vol. 2, pp. 845-849, 11-14 Dec. 2002
- [6] Yu-Ju Chen, Tsung-Chuan Huang, R. C. Hwang, "Reliable Power Load Forecaster by Neural Networks with a Modified FBP Learning Algorithm," *Engineering Intelligent Systems for Communication and Electrical Engineering* (CRL Publishing LTD), Vol. 12, No. 4, pp. 211-216, Dec., 2004.
- [7] G. Purushothaman and N. B. Karayiannis, "Quantum Neural Networks (QNN's): Inherently Fuzzy Feed-Forward Neural Networks," *IEEE Transactions on Neural Network*, Vol. 8, No. 3, 1997.
- [8] Chang-Der Lee, Yu-Ju Chen, Huang-Chu Huang, Rey-Chue Hwang, Gwo-Ruey Yu, "The Non-Stationary Signal Prediction by Using Quantum NN," *Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics (SMC04)*, pp. 3291-3295, Oct.10-13, The Hague, The Netherlands, 2004.
- [9] Xin-Yi Tsai, Yu-Ju Chen, Huang-Chu Huang, Shang-Jen Chuang, Rey-Chue Hwang, "QNN

- vs. NN in Signal Recognition,” Proceedings of IEEE 2005 *International Conference on Information Technology and Applications*, pp. 308-312, July 4-7, Sydney, Australia, 2005.
- [10] Rey-Chue Hwang, Yu-Ju Chen, Shang-Jen Chuang, Huang-Chu Huang, Wei-Der Chuang, “Fast Learning Neural Network with Modified Neurons,” 2005 *International Conference on Information Technology and Applications*, pp. 313-318, July 4-7, Sydney, Australia, 2005.

Table 1: The error statistics of daily total load forecasting by using Model 1.

$\eta = \alpha$		$\eta = \alpha$	
0.1	1.7825%	0.01	1.6952%
0.2	1.7807%	0.02	1.6553%
0.3	1.7450%	0.03	1.6137%
0.4	1.7950%	0.04	1.6193%
0.5	1.6982%	0.05	1.6674%
0.6	1.6905%	0.06	1.7529%
0.7	1.6225%	0.07	1.7335%
0.8	1.6604%	0.08	1.6824%
0.9	1.7580%	0.09	1.7023%
AVG.	1.7259%	AVG.	1.68021%

Table 2: The error statistics of daily total load forecasting by using Model 2.

$\eta = \alpha$	$\alpha_a = \alpha_b = 0.1$	$\alpha_a = \alpha_b = 0.01$	$\alpha_a = \alpha_b = 0.001$
0.01	1.6711%	1.6248%	1.6020%
0.02	1.8066%	1.6123%	1.5823%
0.03	1.6802%	1.6124%	1.6212%
0.04	1.6955%	1.6360%	1.6690%
0.05	1.6694%	1.6237%	1.6241%
0.06	1.6366%	1.5961%	1.6214%
0.07	1.6364%	1.6084%	1.6321%
0.08	1.6471%	1.6116%	1.6297%
0.09	1.6618%	1.6139%	1.6308%
AVG.	1.6783%	1.6155%	1.6236%

Table 3: The error statistics of daily total load forecasting by using Model 3.

$\eta = \alpha$		$\eta = \alpha$	
0.1	1.7116%	0.01	1.9884%
0.2	1.7659%	0.02	1.9697%
0.3	1.7600%	0.03	1.9247%
0.4	1.8131%	0.04	1.9293%
0.5	1.7998%	0.05	1.8375%
0.6	1.8166%	0.06	1.7942%
0.7	1.8241%	0.07	1.8076%
0.8	1.7768%	0.08	1.7578%
0.9	1.7870%	0.09	1.7048%
AVG.	1.7839%	AVG.	1.8571%

Table 4: The error statistics of daily total load forecasting by using Model 4.

$\eta = \alpha$	$\alpha_a = \alpha_b = 0.001$	$\eta = \alpha$	$\alpha_a = \alpha_b = 0.001$
0.1	1.6456%	0.01	1.6681%
0.2	1.6490%	0.02	1.7367%
0.3	1.5775%	0.03	1.7242%
0.4	1.5410%	0.04	1.6772%
0.5	1.7773%	0.05	1.6956%
0.6	1.7314%	0.06	1.7204%
0.7	1.6862%	0.07	1.6727%
0.8	1.6650%	0.08	1.7704%
0.9	1.7619%	0.09	1.5758%
AVG.	1.6705%	AVG.	1.6934%

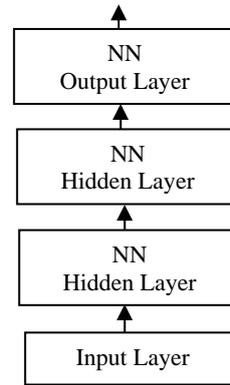


Figure 1: The architecture of Model 1.

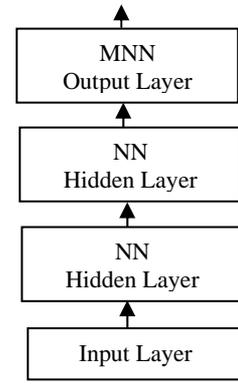


Figure 2: The architecture of Model 2.

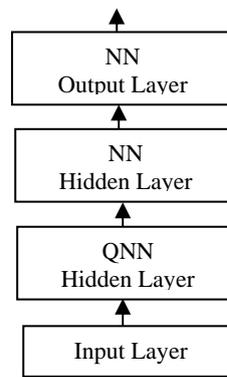


Figure 3: The architecture of Model 3.

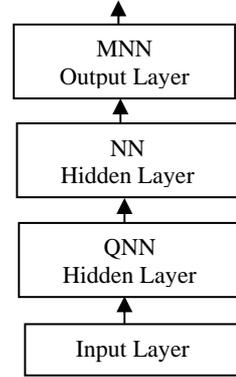


Figure 4: The architecture of Model 4.