Time Series Signal Prediction by Neural Model with Modified Neurons

Pin-Hsuan Weng[1] Tzu-Te Liu[1] Yu-Ju Chen[2] Rey-Chue Hwang[1]

***Abstract*—In this paper, the time series signal prediction held by NN3 neural forecasting competition was presented. Eleven reduced time series datasets, NN3_101 to NN3_111, were studied and analyzed. After several trials, the neural models with modified neurons were used to perform the prediction works. One-to-eighteen-step-ahead forecasting was executed for each dataset.**

I. Introduction

Due to the powerful learning and adaptive capabilities, neural network (NN) has been applied into many scientific areas, such as signal prediction, decision-making, system modeling, control, image processing and so on. [1-6]. In general, the neural network can automatically develop the mapping model between the historical input/output pairs through a simple training process.

After several trials, the NN models with modified neurons were decided and used to predict the time series signals provided by this competition. Its learning algorithm will be described in the following section. Based on the NN structure and its learning mechanism, each neuron of NN model has its individual proper learning step in accordance with the immediate learning error. Therefore, we expect that the learning speed and the modeling accuracy of NN used could have a better performance in comparison with the traditional NN model.

In the experiments, eleven monthly time series datasets, NN3_101 to NN3_111, were studied and simulated. For each dataset, one-to-eighteen-step-ahead signal prediction was performed. Due to capturing the proper inputs for each NN model, the simple data analysis was done firstly for the construction of each prediction model. In other words, the number of inputs taken by each NN for individual *k*-step-ahead signal prediction might be different based on the analysis we did.

II. NN Model and Its Learning Algorithm

A four-layered feed-forward NN structure with size *n-p-p*-1 is the basic model used for all NN prediction models. The modified hyperbolic tangent function, $f(x) = a(1-\exp(-bx))/(1+\exp(-bx))$ [8], is adopted as the node's transfer function in all NN models. a and b are two parameters that can be automatically tuned based on model's performance. Based on the neuron's structure, the learning rule of NN can be presented as follows.

Let $z_j(k)$ be the output of unit j on the kth pattern and ω_{ij} be a weight association with the connection from the ith unit to the jth unit. If j is an input unit, $z_j(k) = x_j(k)$. If j is an output unit, $z_j(k) = y_j(k)$. For each unit j, the output can be expressed as

$$z_j(k) = a_j(k)(1-\exp(-b_j(k)\sum_i \omega_{ij}(k)x_i(k))) \tag{2-1}$$

where, i is over all the inputs having connections to unit j. The objective of network's training is to map given input patterns $X(k) = [x_0(k), x_1(k), .. x_n(k)]$ onto the desired outputs, $S(k) = [S_1(k), S_2(k), ..., S_m(k)]$.

Let $y(k) = [y_1(k), y_2(k), ..., y_m(k)]$ be the actual outputs of network on the presentation pattern k. The cost function is then set as

$$\mathrm{E} = \sum_{p=1}^{m}\sum_{k}(S_p(k) - y_p(k))^2 \tag{2-2}$$

To achieve the desired mapping, we need to minimize the cumulative sum-square error E. This can be achieved by adjusting the weights and parameters as

$$\Delta\omega_{ij}(k) \propto -\frac{\partial \mathrm{E}}{\partial \omega_{ij}(k)} \tag{2-3}$$

$$\Delta a_j(k) \propto -\frac{\partial \mathrm{E}}{\partial a_j(k)} \tag{2-4}$$

$$\Delta b_j(k) \propto -\frac{\partial \mathrm{E}}{\partial b_j(k)} \tag{2-5}$$

By the chain rule, the increments of $\omega_{ij}(k)$, $a_j(k)$ and $b_j(k)$ are derived by

$$\frac{\partial \mathrm{E}}{\partial \omega_{ij}(k)} = \frac{\partial \mathrm{E}}{\partial V_j(k)}\frac{\partial V_j(k)}{\partial \omega_{ij}(k)} \tag{2-6}$$

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

$$\frac{\partial \mathrm{E}}{\partial a_j(k)} = \frac{\partial \mathrm{E}}{\partial y_j(k)}\frac{\partial y_j(k)}{\partial a_j(k)} \tag{2-7}$$

Pin-Hsuan Weng is with the Electrical Engineering Department, I-Shou University, Kaohsiung County, Taiwan. (e-mail: wph36661@seed.net.tw)

Tzu-Te Liu is with the Electrical Engineering Department, I-Shou University, Kaohsiung County, Taiwan (e-mail: chichimvp@yahoo.com.tw)

Yu-Ju Chen is with the information and Management Department, Cheng Shiu University, Kaohsiung County, Taiwan. (e-mail: yjchen@csu.edu.tw)

Rey-Chue Hwang is with the Electrical Engineering Department, I-Shou University, Kaohsiung County, Taiwan (e-mail: rchwang@isu.edu.tw)

$$\frac{\partial \mathrm{E}}{\partial b_j(k)} = \frac{\partial \mathrm{E}}{\partial y_j(k)} \frac{\partial y_j(k)}{\partial b_j(k)} \quad (2\text{-}8)$$

For the output nodes, the increments can be listed as

$$\Delta\omega_{ij}(k) \propto (b_j(k)/a_j(k))(S_j(k) - y_j(k))$$
$$*(a_j(k) + y_j(k))(a_j(k) - y_j(k))x_i(k) \quad (2\text{-}9)$$

$$\Delta a_j(k) \propto (S_j(k) - y_j(k))(y_j(k)/a_j(k)) \quad (2\text{-}10)$$

$$\Delta b_j(k) \propto (V_j(k)/a_j(k))(S_j(k) - y_j(k))$$
$$*(a_j(k) + y_j(k))(a_j(k) - y_j(k)) \quad (2\text{-}11)$$

For the hidden nodes,

$$\Delta\omega_{ij}(k) \propto (b_j(k)/a_j(k))\sum_l \delta_l(k)\omega_{jl}(k)$$
$$*(a_j(k) + y_j(k))(a_j(k) - y_j(k))x_i(k) \quad (2\text{-}12)$$

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

$$\Delta a_j(k) \propto (y_j(k)/a_j(k))\sum_l \delta_l(k)\omega_{jl}(k) \quad (2\text{-}13)$$

$$\Delta b_j(k) \propto (V_j(k)/a_j(k))(a_j(k) + y_j(k))$$
$$*(a_j(k) - y_j(k))\sum_l \delta_l(k)\omega_{jl}(k) \quad (2\text{-}14)$$

III. Inputs Selection

Owing to the one-to-eighteen-step-ahead signal prediction is asked for this competition, therefore, the proper inputs for each forecasting model need to be determined. In our studies, we decided that each *k*-step-ahead signal prediction for every dataset was performed by individual model. There are totally 198 forecasting models were used to execute this work. The size of each model is *m*-12-12-1, where *m* is the number of inputs used for each forecasting model. Table 1 to Table 11 present the detailed inputs used for each dataset.

IV. Prediction Results

The prediction values of the next 18 signals for each dataset are reported as follows.

NN3_101:
5434.55, 5335.241, 5114.969, 5414.379, 5105.043, 5239.098, 5264.585, 4992.333, 5237.783, 5073.278, 5169.461, 5229.307, 4899.031, 5581.51, 5072.088, 5659.568, 4884.518, 5245.283

NN3_102:
3648.357, 7148.444, 8575.931, 8829.302 8028.43, 7990.048, 7226.604, 5602.557, 6429.619, 4410.137, 4354.102, 3183.829, 1364.766, 8771.438, 8556.851, 8831.204, 8728.084, 9118.975

NN3_103:
2052.908, 58117.595, 58151.791, 19329.626, 5754.497, 10375.911, 7535.829, 5843.526, 2698.747, 1921.551, 3379.476, 1601.779, 4556.096, 57417.23, 45722.672, 26717.333, 5059.824, 5059.885

NN3_104:
7127.844, 7002.469, 6701.325, 7632.714, 6291.623, 2897.807, 2559.683, 2817.189, 5010.918, 6887.053, 7120.197, 7303.079, 7248.066, 7086.272, 6900.521, 6634.745 6551.949, 2252.288

NN3_105:
4329.656, 4074.622, 4527.898, 4632.609, 4878.288, 4809.153, 4538.278, 4736.277, 5285.76, 4839.489, 4159.17, 3605.074, 4591.692, 4579.83, 4648.947, 4058.547, 4863.611, 4179.317

NN3_106:
6090.765, 5118.506, 4395.458, 4428.781, 4476.37, 4764.877, 5022.543, 5777.125, 5036.525, 3901.06, 4793.094, 5947.609, 5310.471, 4967.257, 4582.81, 4984.734, 5598.096, 5585.334

NN3_107:
3676.499, 3670.892, 3371.532, 4005.184, 3708.164, 3583.756, 3886.553, 3838.325, 3371.79, 3368.108, 3902.77, 3433.812, 3741.426, 3679.079, 3789.273, 3888.858, 3394.421, 3660.524

NN3_108:
1673.379, 7514.384, 1872.069, 2763.152, 3121.338, 1731.802, 1029.326, 1350.628, 2947.602, 3148.951, 3024.757, 1827.89, 1009.878, 2554.873, 3212.205, 2323.816, 1021.025, 2398.322

NN3_109:
3552.38, 2681.64, 2918.438, 3101.475, 2857.232, 3243.75, 2649.913, 2939.517, 2739.644, 3234.823, 3192.334, 2854.022, 3021.847, 3339.432, 3414.694, 3191.141, 3205.44, 3206.984

NN3_110:
1727.818, 175.305, 2585.625, 1923.612, 2188.624, 2660.498, 2816.389, 2597.095, 2788.893, 2581.579, 3257.069, 2961.769, 2632.349, 3186.95, 2364.339, 2807.933, 2691.972, 2664.929

NN3_111:
2786.515, 2759.038, 2343.897, 2275.073, 2311.196, 2671.254, 2622.385, 2765.577, 2525.427, 2774.96, 2720.998, 2702.876, 2210.779, 2677.007, 2201.852, 2200.098, 4318.736, 2612.977

V. Conclusion

In this paper, the time series signal prediction held by NN3 neural forecasting competition was presented. Eleven datasets, NN3_101 to NN3_111, were studied and analyzed. The neural models with modified neurons were used to perform our prediction works. For each dataset, One-to-eighteen-step-ahead, i.e. the next 18 forecasting values were executed and reported.

References

[1] J. A. Anderson, M. T. Gately, P. A. Penz, and D. R. Collins. (1990). Radar Signal Categorization Using a neural Network. *Proceedings of the IEEE*, 78, 1646-1657.

[2] S. Chen, S. Billings, and P. Grant. (1990). Non-linear System Identification Using Neural Networks. *International Journal of Control*, 51, 1191-1214.

[3] W. H. Schiffman and H. W. Geffers. (1993). Adaptive Control of Dynamic Systems by Backpropagation Networks. *Neural Networks*, 6, 517-524.

[4] A. Khotanzad, R. C. Hwang, A. Abaye, and D. Maratukulam. (1995). An Adaptive Modular Artificial Neural Network: Hourly Load Forecaster and Its Implementation at Electric Utilities. *IEEE Transactions on Power Systems*, 10, 1716-1722.

[5] C. Wohler and J. K. Anlauf. (1999). An Adaptive Time-Delay Neural-Network Algorithm for Image Sequence Analysis. *IEEE Transactions on Neural Networks*, 10, 1531-1536.

[6] P. J. Edwards, A. F. Murray, G. Papadopoulos, A. R. Wallace, J. Barnard, and G. Smith. (1999). The Application of Neural Networks to the Papermaking Industry. *IEEE Transactions on Neural Networks*, 10, 1456-1464.

[7] D. E. Rumelhart, G. E. Hinton and R. J. Williams. (1986). Learning Internal Representations by Error Propagation. *Parallel Distributed Processing*, 1, MIT Press.

[8] W. D. Chang, R. C. Hwang, J. G. Hsieh (1999), A Single Auto-tuning Neural Controller for Nonlinear Dynamic Systems. P*roceedings of IEEE International Conference on Systems, Man, and Cybernetics*, (IEEE SMC'99), I-461-465.

Table 1: The inputs used for NN3_101 data

Step Ahead	Inputs	No. of Input
1	y(k-1), …, y(k-14)	14
2	y(k-2), …, y(k-19)	18
3	y(k-3), …, y(k-12)	10
4	y(k-11), …, y(k-21)	11
5	y(k-11), …, y(k-22)	12
6	y(k-6), …, y(k-12)	7
7	y(k-18), …, y(k-24)	7
8	y(k-8), …, y(k-12)	5
9	y(k-9), …, y(k-12)	4
10	y(k-10), …, y(k-12)	3
11	y(k-11), …, y(k-21)	11
12	y(k-12), …, y(k-20)	9
13	y(k-13), …, y(k-23)	11
14	y(k-20), …, y(k-31)	12
15	y(k-15), …, y(k-17)	3
16	y(k-22), …, y(k-33)	12
17	y(k-30), …, y(k-34)	5
18	y(k-18), …, y(k-24)	7

Table 2: The inputs used for NN3_102 data

Step Ahead	Inputs	No. of Input
1	y(k-1),…, y(k-18)	18
2	y(k-2),…, y(k-12)	11
3	y(k-3),…, y(k-9)	7
4	y(k-6),…, y(k-21)	16
5	y(k-6),…, y(k-22)	17
6	y(k-12),…, y(k-23)	12
7	y(k-10),…, y(k-24)	15
8	y(k-8),…, y(k-23)	16
9	y(k-9),…, y(k-22)	14
10	y(k-10),…, y(k-24)	15
11	y(k-11),…, y(k-23)	13
12	y(k-12),…, y(k-23)	12
13	y(k-13),…, y(k-24)	12
14	y(k-14),…, y(k-26)	13
15	y(k-15),…, y(k-25)	11
16	y(k-16),…, y(k-27)	12
17	y(k-24),…, y(k-34)	11
18	y(k-18),…, y(k-24)	7

Table 3: The inputs used for NN3_103 data

Step Ahead	Inputs	No. of Input
1	y(k-1),…, y(k-13)	13
2	y(k-2),…, y(k-12)	11
3	y(k-3),…, y(k-16)	14
4	y(k-4),…, y(k-9)	6
5	y(k-5),…, y(k-12)	8
6	y(k-6),…, y(k-10)	5
7	y(k-7),…, y(k-13)	7
8	y(k-8),…, y(k-17)	10
9	y(k-9),…, y(k-15)	7
10	y(k-10),…, y(k-15)	6
11	y(k-11),…, y(k-13)	3
12	y(k-12),…, y(k-23)	12
13	y(k-13),…, y(k-29)	17
14	y(k-14),…, y(k-27)	14
15	y(k-15),…, y(k-24)	10
16	y(k-16),…, y(k-33)	18
17	y(k-18),…, y(k-34)	17
18	y(k-18),…, y(k-34)	17

Table 4: The inputs used for NN3_104 data

Step Ahead	Inputs	No. of Input
1	y(k-5),…, y(k-18)	14
2	y(k-2),…, y(k-15)	14
3	y(k-5),…, y(k-20)	16
4	y(k-12),…, y(k-21)	10
5	y(k-5),…, y(k-18)	14
6	y(k-6),…, y(k-15)	10
7	y(k-21),…, y(k-24)	4
8	y(k-9),…, y(k-25)	17
9	y(k-9),…, y(k-15)	7
10	y(k-19),…, y(k-27)	9
11	y(k-21),…, y(k-28)	8
12	y(k-21),…, y(k-29)	9
13	y(k-21),…, y(k-30)	10
14	y(k-21),…, y(k-31)	11
15	y(k-21),…, y(k-32)	12
16	y(k-24),…, y(k-33)	10
17	y(k-17),…, y(k-31)	15
18	y(k-23),…, y(k-35)	13

Table 5: The inputs used for NN3_105 data

Step Ahead	Inputs	No. of Input
1	y(k-1),…, y(k-16)	16
2	y(k-2),…, y(k-18)	17
3	y(k-11),…, y(k-20)	10
4	y(k-4),…, y(k-13)	10
5	y(k-5),…, y(k-15)	11
6	y(k-6),…, y(k-16)	11
7	y(k-18),…, y(k-24)	7
8	y(k-21),…, y(k-25)	5
9	y(k-9),…, y(k-13)	5
10	y(k-10),…, y(k-20)	11
11	y(k-11),…, y(k-20)	10
12	y(k-12),…, y(k-16)	5
13	y(k-23),…, y(k-30)	8
14	y(k-22),…, y(k-31)	10
15	y(k-23),…, y(k-32)	10
16	y(k-26),…, y(k-33)	8
17	y(k-25),…, y(k-34)	10
18	y(k-18),…, y(k-24)	7

Table 6: The inputs used for NN3_106 data

Step Ahead	Inputs	No. of Input
1	y(k-12),…, y(k-18)	7
2	y(k-7),…, y(k-19)	13
3	y(k-8),…, y(k-20)	13
4	y(k-4),…, y(k-17)	14
5	y(k-5),…, y(k-22)	18
6	y(k-6),…, y(k-19)	14
7	y(k-7),…, y(k-19)	13
8	y(k-8),…, y(k-20)	13
9	y(k-9),…, y(k-17)	9
10	y(k-10),…, y(k-20)	11
11	y(k-24),…, y(k-28)	5
12	y(k-12),…, y(k-18)	7
13	y(k-13),…, y(k-20)	8
14	y(k-22),…, y(k-31)	10
15	y(k-18),…, y(k-32)	15
16	y(k-16),…, y(k-26)	11
17	y(k-17),…, y(k-32)	16
18	y(k-18),…, y(k-32)	15

Table 7: The inputs used for NN3_107 data

Step Ahead	Inputs	No. of Input
1	y(k-1),…, y(k-3)	3
2	y(k-16),…, y(k-19)	4
3	y(k-14),…, y(k-20)	7
4	y(k-14),…, y(k-21)	8
5	y(k-20),…, y(k-22)	3
6	y(k-6),…, y(k-23)	18
7	y(k-15),…, y(k-24)	10
8	y(k-16),…, y(k-25)	10
9	y(k-9),…, y(k-12)	4
10	y(k-10),…, y(k-15)	6
11	y(k-18),…, y(k-28)	11
12	y(k-12),…, y(k-14)	3
13	y(k-13),…, y(k-20)	8
14	y(k-14),…, y(k-21)	8
15	y(k-15),…, y(k-24)	10
16	y(k-16),…, y(k-19)	4
17	y(k-17),…, y(k-19)	3
18	y(k-18),…, y(k-28)	11

Table 8: The inputs used for NN3_108 data

Step Ahead	Inputs	No. of Input
1	y(k-1),…, y(k-15)	15
2	y(k-2),…, y(k-5)	4
3	y(k-3),…, y(k-9)	7
4	y(k-4),…, y(k-17)	14
5	y(k-5),…, y(k-11)	7
6	y(k-6),…, y(k-13)	8
7	y(k-19),…, y(k-24)	6
8	y(k-8),…, y(k-15)	8
9	y(k-24),…, y(k-26)	3
10	y(k-14),…, y(k-27)	14
11	y(k-26),…, y(k-28)	3
12	y(k-12),…, y(k-14)	3
13	y(k-13),…, y(k-28)	16
14	y(k-14),…, y(k-27)	14
15	y(k-30),…, y(k-32)	3
16	y(k-30),…, y(k-33)	4
17	y(k-17),…, y(k-22)	6
18	y(k-18),…, y(k-20)	3

Table 9: The inputs used for NN3_109 data

Step Ahead	Inputs	No. of Input
1	y(k-1),…, y(k-15)	15
2	y(k-2),…, y(k-16)	15
3	y(k-11),…, y(k-20)	10
4	y(k-11),…, y(k-21)	11
5	y(k-12),…, y(k-22)	11
6	y(k-12),…, y(k-23)	12
7	y(k-19),…, y(k-24)	6
8	y(k-21),…, y(k-25)	5
9	y(k-18),…, y(k-26)	9
10	y(k-12),…, y(k-27)	16
11	y(k-12),…, y(k-28)	17
12	y(k-12),…, y(k-25)	14
13	y(k-18),…, y(k-30)	13
14	y(k-14),…, y(k-30)	17
15	y(k-19),…, y(k-32)	14
16	y(k-16),…, y(k-31)	16
17	y(k-17),…, y(k-30)	14
18	y(k-18),…, y(k-35)	18

Table 10: The inputs used for NN3_110 data

Step Ahead	Inputs	No. of Input
1	y(k-1),…, y(k-16)	16
2	y(k-2),…, y(k-15)	14
3	y(k-3),…, y(k-13)	11
4	y(k-4),…, y(k-16)	13
5	y(k-5),…, y(k-15)	11
6	y(k-6),…, y(k-22)	17
7	y(k-7),…, y(k-11)	5
8	y(k-8),…, y(k-12)	5
9	y(k-9),…, y(k-11)	3
10	y(k-22),…, y(k-27)	6
11	y(k-25),…, y(k-28)	4
12	y(k-23),…, y(k-29)	7
13	y(k-17),…, y(k-30)	14
14	y(k-24),…, y(k-31)	8
15	y(k-26),…, y(k-32)	7
16	y(k-19),…, y(k-33)	15
17	y(k-26),…, y(k-34)	9
18	y(k-26),…, y(k-35)	10

Table 11: The inputs used for NN3_111 data

Step Ahead	Inputs	No. of Input
1	y(k-10),…, y(k-18)	9
2	y(k-8),…, y(k-19)	12
3	y(k-12),…, y(k-20)	9
4	y(k-4),…, y(k-14)	11
5	y(k-9),…, y(k-22)	14
6	y(k-7),…, y(k-23)	18
7	y(k-22),…, y(k-24)	3
8	y(k-8),…, y(k-19)	12
9	y(k-20),…, y(k-26)	7
10	y(k-10),…, y(k-18)	9
11	y(k-11),…, y(k-14)	4
12	y(k-12),…, y(k-19)	8
13	y(k-13),…, y(k-15)	3
14	y(k-14),…, y(k-25)	12
15	y(k-15),…, y(k-23)	9
16	y(k-16),…, y(k-24)	9
17	y(k-17),…, y(k-26)	10
18	y(k-19),…, y(k-35)	17