



A Hybrid Neural Network for Predictive Model in A Plastic Injection Molding Process

Tossapol Kiatcharoenpol^{1*} Sakon Klongboonjit¹

¹*Industrial Engineering Department, School of Engineering,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand*

* Corresponding author's Email: tossapol_k@yahoo.com

Abstract: A reliable and sensitive technique for predicting quality of a plastic work-piece produced in injection molding process is essential help for practicing engineers. A system based on the process parameters that can estimate both two prime characteristics, %volume shrinkage and warpage of work-piece before it produced is significantly beneficial. In this paper, a fast feed forward network, Hybrid Neural Network (HNN), is proposed to construct the predictive model for those two quality characteristics. The unique algorithm of HNN based on the optimization of the weights of each layer is changed to a linear problem by linearization of the sigmoid functions. As iteration procedure used in Backpropagation algorithm is eliminated, the network training time is significant reduced. With this fast convergence of using HNN, the intelligent predictive model for injection molding process that can learn online is possible for further study. To entitle the network to cater for various process parameter conditions, a knowledge base as training and testing data have to be generated on the experimental data in a comprehensive working range of a plastic injection molding process. Consequently, the experiments were performed in 256 conditions based on the combination of nine basic process parameters. The neural networks were trained and the architecture of networks was appropriately selected by benchmarking the Root Mean Square error (RMS). The results of the novel network, HNN, have shown the ability to accurately predict the percentage of volume shrinkage with the 1.02% and 4.87% error at training and testing stages, respectively and for warpage with the 3.76% and 2.47% error at training and testing stages, respectively. These accuracy results are similar to those of backpropagation neural network (BPNN), but HNN has shown the superior fast converging about 38.5% and 66.7% over than those of BPNN.

Keywords: Backpropagation neural network, Hybrid neural network, Injection molding.

1. Introduction

It is commonly known that the poor quality of product is unacceptable in a competitive industrial world. The high quality of production usually comes from the good process control. However, since a lot of process parameters to be optimized in order to set the suitable production condition, it is one of complicated work. There is a need to establish an intelligent system to predict the product quality before it produced. In this work, a neural network approached is proposed to create a predictive model that could handle a variety of process parameters in a case of a plastic injection molding process and the percentage of volume shrinkage and warpage are key

quality characteristics of work-piece. To establish knowledge bases for the predictive model to catering for various process setting, a number of experiments must be performed under the working conditions of nine major parameters of injection molding process. Such parameters are filling time, melt temperature, mold temperature, maximum injection pressure, Packing time, Maximum packing pressure profile value, cooling time, air temperature and eject temperature.

To construct predictive neural network model, the backpropagation algorithm is one of the most powerful and popular algorithms of the neural network. It is used in many other disciplinary for the pattern recognition and modelling purpose, however according to the research carried out in [1] and [2]

report that the backpropagation neural network (BPNN) convergence rate is very poor due to local minima. In the work of liang et al. [3] also try to improve another disadvantage of BPNN which possibly be sensitive to noisy data and irregularity. Many efforts are worked out to modify BPNN in adjusting learning rate such as variable stepsize learning [4] and using dynamically optimal learning rate [5] to reduce training time. Moreover, a highly complicate mathematic technique in various studies such as a hybrid quantum [3] and TAO-robust Backpropagation learning [6] are added to improve the BPNN to robust the neural network output. Some studies suggest that using other fast algorithm or hybrid algorithm is also a better alternative to avoid a complicate of modify BPNN if the convergence time is primarily focus in the application. Then the aim of this study is that a new neural network, hybrid neural network (HNN) [7], has been proposed to be a predictive model for a plastic injection molding process with a fast training time. HNN is known not only for being fast in learning, but also for being superior in accurate modelling. A brief introduction of HNN is initially discussed, followed by the development of a knowledge base of in case of a plastic injection molding process. Therefore, the architectures of neural networks are optimized to yield the best results. The performance benchmarking of HNN is tested and compared with backpropagation neural network (BPNN) in this application in both accuracy of information and the speed of convergence time.

2. A briefs note of artificial neural network

2.1 Basic of backpropagation (BPNN)

All The standard back propagation network [8-10] comprises three layers of processing elements, fully feed-forward connected shown in Fig. 1. With the sigmoid on the hidden layer, only the basic equations are:

$$y_k = \sum_{j=1}^H v_{kj} z_j \quad (1)$$

$$z_j = \frac{1}{1 + \exp(-net_j)} \quad (2)$$

$$net_j = \sum_{i=1}^M w_{ji} x_i \quad (3)$$

The Least Mean Square error

$$E = \frac{1}{2} \sum_{k=1}^M (y^k - t^k)^2 \quad (4)$$

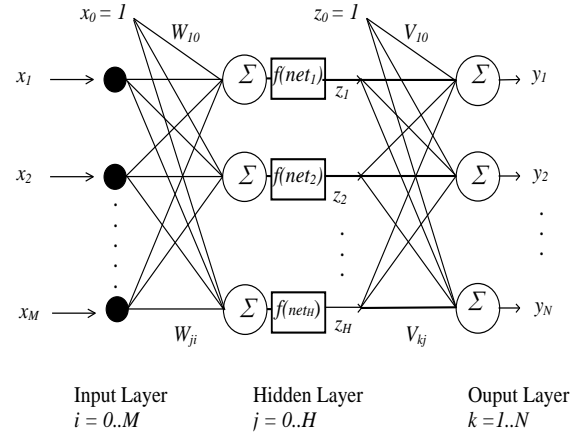


Figure. 1 A feed forward neural network with one hidden layer

where, net_j = weighted sum of inputs to neuron j .

t_k = target value for neuron k

x_i = input of neuron i in input layer

y_k = output value for neuron k

z_i = output of neuron j in hidden layer

w_{ji} = weight connecting input neuron i to hidden layer neuron j

V_{kj} = weight connecting hidden neuron j to output layer neuron k .

All the data are scaled between 0..1 but it can be scaled between -1..1 to standardize all the inputs with various dimensions. The weights W_{ji} and V_{kj} are assigned random numbers in the range -1..1, and a random pair of input / output vectors are picked from the training set. The input vector is fed through the network to get an output vector (feed-forward process); this is then compared with the output vector and an error is found.

This error is then passed back through the neural network (back propagation process) to modify the weights using the following equations

$$v_{kj}^{new} = v_{kj}^{old} + \Delta v_{kj} \quad (5)$$

$$w_{ji}^{new} = w_{ji}^{old} + \Delta w_{ji} \quad (6)$$

The gradient descent optimization technique is used to calculate the change in each weight. This is then repeated by picking another random pair of input / output vectors and continuing until the error is at a minimum.

2.2 A brief of hybrid neural network (HNN)

The basic architecture of an HNN is the same as BPNN, as shown in Fig. 1. It consists of an input layer,

one or more hidden layers and an output layer. All input nodes are connected to all hidden neurons through weighted connections, W_{ji} , and all hidden neurons are connected to all output neurons through weighted connections, V_{kj} .

The basic idea of HNN learning algorithm is to combine iterative and direct weight optimization methods. Based on this concept, the training time is improved and local minima can be avoided [11]. The network is trained by initialising small random weights and then the weights are iteratively adjusted by using Backpropagation algorithm with some initial iteration for training performance enhancement. Then the direct weight optimization method based on Optimization Layer by Layer algorithm [12] is used to finalize weights with a fast converging rate. The fast convergence is resulted from that the optimization of weights in each layer is reduced to a linear problem and the weights in each layer are modified dependent on each other, but separately from all other layers.

This HNN algorithm can be applied to any number of layers. However, the vast majority of practical problems are rarely necessary to use more than one hidden layer [13]. Thus, only the algorithm of network with one hidden layer to reduce the complexity and speed of computing is presented as followings.

Training algorithm of HNN with one hidden layer

- Step 1 Initialize weights
 - Set all weights (W_{ji}, V_{kj}) to small random values
 - Set weight factor $\mu=0.0001$,
- Step 2 Weight adjustment by using Backpropagation algorithm
 - Set the number of initial iterations
 - All weights (W_{ji}, V_{kj}) are iteratively modified according to the number of initial iterations
- Step 3 Optimization of output-hidden layer weights
 - The gradient of cost function with respect to V is calculated to derive the optimal weight V for all training patterns. Thus,

$$V_{jk} = A^{-1} \cdot b \tag{7}$$

where the A and b matrix are given by:

$$A_{(j,l)} = \text{matrix}[a_{j,l}]; a_{j,l} = \sum^p [z_j z_{j,l}]; j, l = 0..H$$

$$b_{(j,k)} = \text{matrix}[b_{j,k}]; b_{j,k} = \sum^p [t_k z_j]; k = 1..N$$

p = number of training patterns
 t_k = target output of output neuron k

- Step 4 Optimization of the input-hidden layer weights
 - Transform non-linear part into linear problem. Then the linearized weights in each output layer node can be calculated as follow:

$$Vlin_{kj} = \sum^j [f'(net_j) V_{kj}] \tag{8}$$

where $f'(net)$ is derivative of the sigmoidal function

- Calculate weight correction term (ΔW_{opt}) for all training patterns.

$$\Delta W_{opt} = Au^{-1} \cdot bu \tag{9}$$

where Au = matrix [$a_{(ij, hm)}$];

bu = vector [$b_{(j,i)}$]

$$a_{(ji, hm)} : \text{for } (j \neq h) = \sum^p \sum^k [(Vlin_{kj} x_i)(Vlin_{kh} x_m)]$$

$$: \text{for } (j = h) = \sum^p \sum^k [(Vlin_{kj} x_i)(Vlin_{kh} x_m) + \mu/H * abs(V_{kj}) f'(net_j) x_i x_m]$$

$$b_{ji} = \sum^p \sum^k [(t_k - y_k) Vlin_{kj} x_i]$$

H = no. of neurons in hidden layer

- Calculate weight test (W_{test})

$$W_{test} = W_{old} + \Delta W_{opt} \tag{10}$$

- Step 5 Update of the input-hidden layer weights
 - Base on W_{test} , the new RMS error is calculated
 - If (New RMS > RMS) then go back to Step 3 and increase μ ($\mu = \mu * 1.2$)
 - Else update weights

$$W_{new} = W_{test} \{W_{old} + \Delta W_{opt}\} \tag{11}$$

and decrease μ ($\mu = \mu * 0.9$)

for next iteration

- Step 6 Do step 3 - 5 until test stop condition is true.
 - (RMS error < target or End of number of iterations)

2.3 A brief of a plastic injection molding process

A case study conducted in this place is injection molding process by using injection molding simulation software. Basically, the injection molding processes in Fig. 2 are separated into four steps: 1) Plasticizing the resin, 2) Injecting the resin, 3) Cooling the part and 4) Ejecting the part. At the beginning state, the melting process is started by rotating and retracting a screw in order to transform plastic resin into ready liquid phase, which is then injected into the molding. The injection molding is hold for a while in the packing process to maintain pressure to compensate for shrinkage of the setting plastic. When the temperature of work-piece is reduced to meet the eject temperature by the cooling process, the unload process is started to free work-piece. [14].

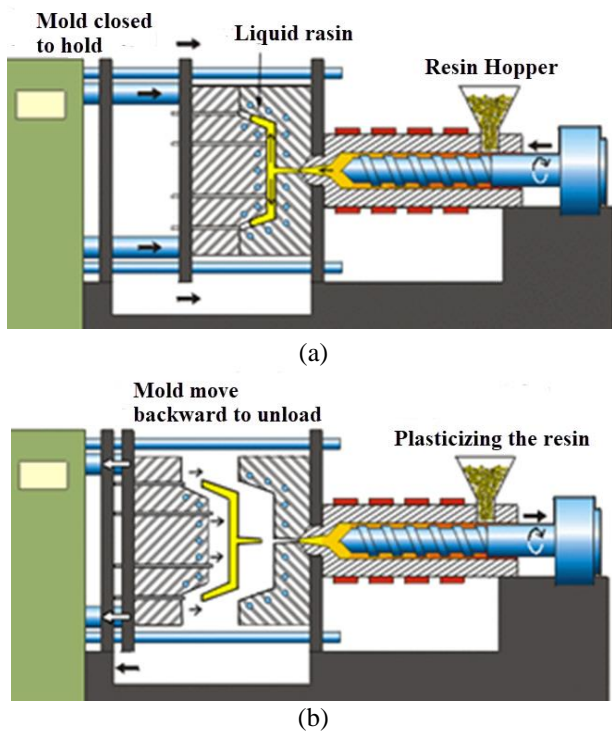


Figure. 2 System model: (a) injecting and (b) ejecting the part

Table 1. Input process parameters and settings

Inputs	Level		Units
	Low (-)	High (+)	
1. Filling time	2	3	sec.
2. Melt temperature	215	230	°C
3. Mold temperature	35	60	°C
4. Maximum injection pressure	50	80	%
5. Packing time	4	6	sec.
6. Maximum packing pressure profile value	30	50	%
7. Cooling time	12	14	sec.
8. Air temperature	15	40	°C
9. Eject temperature	50	99	°C

3 A development of a knowledge base for HNN training

The experiments are carried out to create a data base, which came from the work of our research group. A numerical set of process parameters as network inputs and both of % volume shrinkage and warpage (displacement) as network outputs are generated. Such nine process parameters are 1) Filling time, 2) Melt temperature, 3) Mold temperature, 4) Maximum injection pressure, 5) Packing time, 6) Maximum packing profile value, 7) Cooling time, 8) Air temperature, and 9) Eject temperature. Based on the principle of 2-level

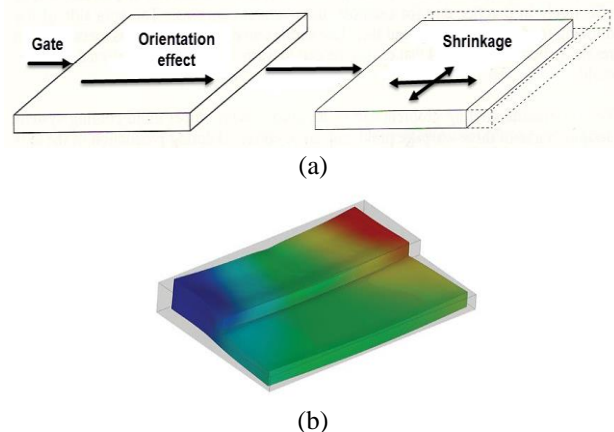


Figure. 3 Characteristic of: (a) volume shrinkage and (b) warpage

factorial design, the detail conditions of each parameter are demonstrated in Table 1.

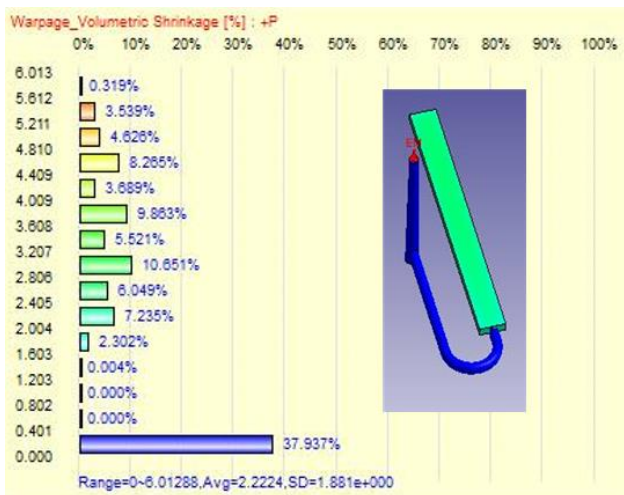
The outputs for neural network are % volume shrinkage and warpage or total displacement of a plastic work-piece. This volume shrinkage is measured in percentage (%) by using a formula in Eq. (12). And the characteristic of this response is shown in Fig. 3 (a). Another key output, total displacement is measured in millimetres (mm.) by using a formula in Eq. (13) with the characteristic of this response as illustrated in Fig. 3 (b).

$$\begin{aligned} \%Volume\ shrinkage \\ = \frac{Size\ of\ mold\ cavity - size\ of\ workpiece}{size\ of\ mold\ cavity} \% \end{aligned} \quad (12)$$

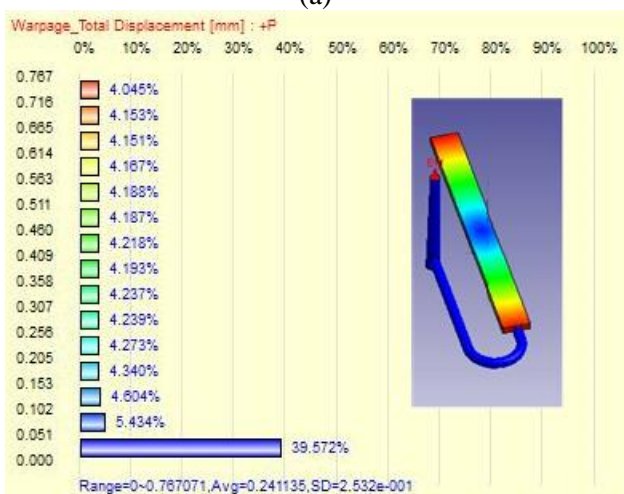
$$\begin{aligned} Warpage\ (total\ displacement\ in\ mm.) \\ = \frac{Dimension\ of\ mold\ cavity}{Dimension\ of\ workpiece} \end{aligned} \quad (13)$$

The computer program is applied to analyze the injection molding process for finding the best value of work-piece design and both outputs, % volume shrinkage and warpage, as shown in Fig. 4.

After the experimentations in simulation of injection molding process, 256 data patterns for training and testing a neural network were created. Twenty percent of all data is separated as testing data. Thus, 204 data patterns were used as training data to build the model and the remaining 52 data patterns were used as extensive testing data to validate the model.



(a)



(b)

Figure 4 Measurement of: (a) % volume shrinkage and (b) warpage

4 Results and discussion

4.1 A selection of HNN architectures

The optimization of the neural networks is done by systematically training and testing the networks with varied parameter settings. As a benchmark, the root mean square (RMS) error of the test data set is recorded and compared. The RMS error easily combines the results from all the test data patterns into one number without any cancellation of negative and positive errors. The formula of RMS [15] is given by

$$RMS\ error = \sqrt{\frac{\sum (target - output)^2}{no.\ of\ data}} \quad (14)$$

In HNN, the number of neurons in the hidden layer has to be optimized. Therefore, HNN with one hidden layer containing two to nine neurons was tested. The maximum number of iterations is limited

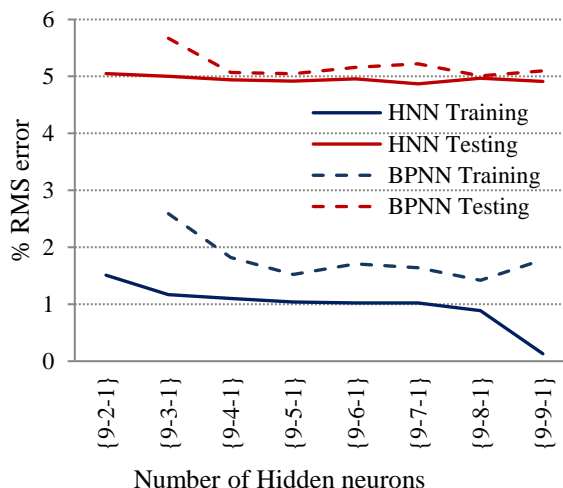


Figure 5 RMS error of HNN and BPNN with number of hidden neurons for % volume shrinkage

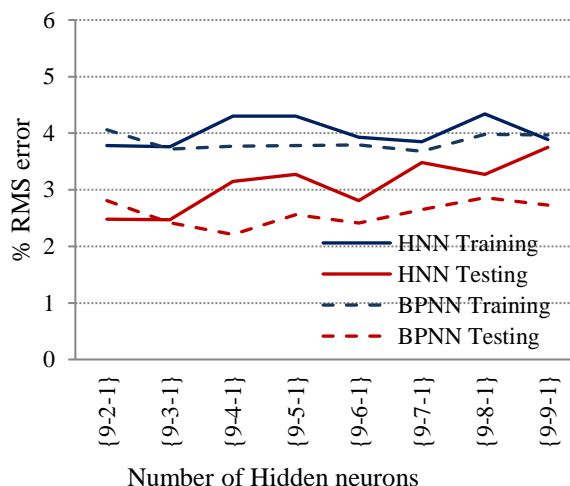


Figure 6 RMS error of HNN and BPNN with number of hidden neurons for warpage

to ten iterations in order to keep the advantage of HNN in fast computing time. The results have shown the trend of RMS errors of training and testing data as illustrated in the Fig. 5 and Fig. 6.

Hence, the HNN (9-7-1) with seven hidden neurons giving the lowest RMS in the study for % volume shrinkage prediction model was chosen and the RMS error of testing data for such a configuration was 4.87%, while HNN (9-3-1) with three hidden neurons was chosen for warpage prediction model with the lowest RMS of 2.47%. For backpropagation neural network (BPNN), since it is a well-established neural network, the detailed algorithm is not discussed in this work, but appropriate references are provided in [16-18]. The lowest RMS error of 5.01% and 2.41% was obtained for % volume shrinkage and warpage when using the BPNN (9-8-1) and BPNN (9-6-1), respectively.

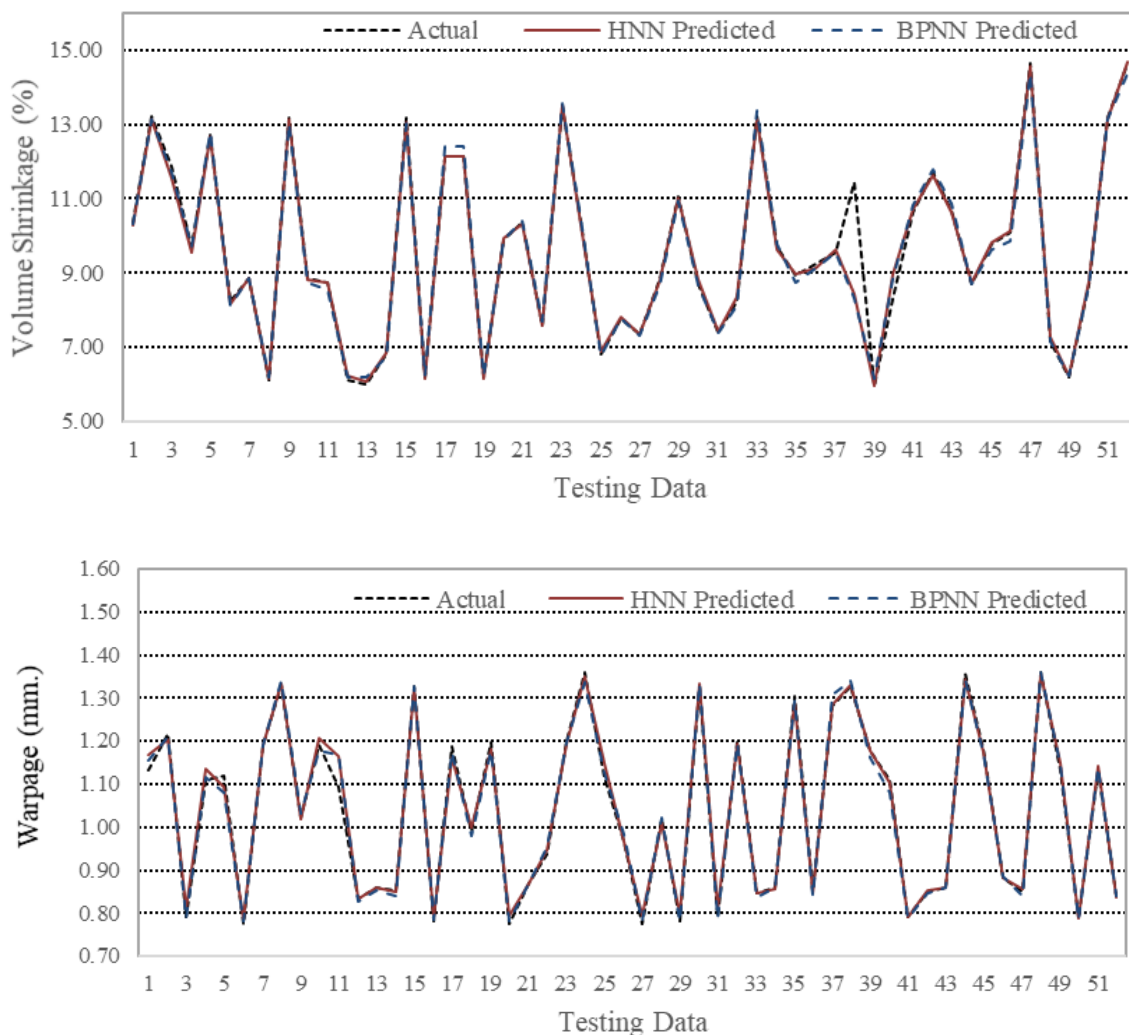


Figure. 7 Actual vs HNN and BPNN predicted of % volume shrinkage and warpage for testing data

Table 2. A Comparative result of HNN and BPNN

Method	% Volume Shrinkage				Warpage			
	% RMS Error		No. of Iterations	Computing Time (sec.)	%RMS Error		No. of iterations	Computing Time (sec.)
	Training	Testing			Training	Testing		
HNN	1.02%	4.87%	7	0.16	3.76%	2.47%	6	0.05
BPNN	1.42%	5.01%	437	0.26	3.79%	2.41%	106	0.15

4.2 HNN model for %volume shrinkage and warpage prediction

For prediction, nine inputs were used. The inputs are filling time, melt temperature, mold temperature, maximum injection pressure, packing time, maximum packing profile value, cooling time, air temperature, and eject temperature and the output were volume shrinkage in percentage and warpage in millimetre. The direct comparison between the actual outputs from the experiment and outputs from prediction of testing data is presented in Fig. 7.

A comparative result of the RMS error of both prediction models are calculated and summarized in Table 2. For %volume shrinkage, it can be seen that HNN can predict with a 4.87% error in testing stage and BPNN can predict with 5.01% error, while the computing time of HNN is 0.16 sec. which is faster than that (0.26 sec) of BPNN by 38.5%. Likewise, the result of warpage, it shows that HNN can predict with a 2.47% error in testing stage and BPNN can predict with 2.41% error, while the computing time of HNN is 0.05 sec. which is much faster than that (0.15 sec) of BPNN by 66.7%. It should be noted that a reduction of training time of HNN results from the a

few iterations needed for convergence, but the backpropagation algorithm relies on a number of iteration process to reach the appropriate weights.

HNN is clearly showing the similar predictive capability to the established BPNN, but superior faster computing time. Besides, due to learning algorithm of HNN adjust weights in each layer based upon linear problem calculation, this make HNN can be converged in the less computing time. It is noted that the accuracy and fast computing of HNN reassure a pragmatic use of an intelligent model for predicting a quality of plastic work-piece to help the practitioners for excellent process control work.

5 Conclusion

The need for reliable prediction of quality of plastic work-piece in an injection molding process is presented. It has been shown that a number of process parameters are required together with for the injection molding performance prediction. In this work, the fast learning neural network is a hybrid artificial neural network (HNN), whose architecture is a multi-layer feed forward neural network. Basically, the fast convergence is resulted from the optimization of weights in each layer is reduced to a linear problem and the weights in each layer are modified dependent on each other, but separately from all other layers including the initial weights generated based on Backpropagation algorithm. Experiments were carried out over a range of plastic injection molding process conditions to measure the % volume shrinkage and warpage of a work-piece produced. A range of conditions covering 256 process conditions were selected as a training and test data sets. The HNN algorithm and also BPNN have trained well with excellent quantitative accuracy with about or less than 5% average percentage deviation to the experimental values in training stage. The networks have been tested with 52 different process conditions and showed high predictive capability. The average RMS errors of predicted % volume shrinkage and warpage are 4.87% and 2.47%, respectively, while training times of HNN for both predictions are faster than those of BPNN by 38.5% and 66.7%. It should be noted that the HNN can perform well in this case in term of providing similar predictive capability, but much better fast training time.

For future work, the development of the intelligent system that can updated instantly and efficiently by on-line neural network is on focus, then the input process parameters of the injection molding processes can be studied in order to be eliminated the least contribution inputs. This reduces the size of

neural network architecture, which potentially increases fast convergence time to make possible on-line learning neural network.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

In this work, conceptualization, methodology, validation, formal analysis, investigation, resources, data curation, writing-original draft preparation, writing-review and editing, visualization, and funding acquisition have been done by the 1st author. The supervision and project administration have been done by the 2nd author.

Acknowledgments

The work is supported by King Mongkut's Institute of Technology Ladkrabang [Grant Number 2564-02-01-018].

References

- [1] G. Marco and A. Tesi, "On the Problem of Local Minima in Backpropagation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 1, pp. 76-86, 1992.
- [2] R. Salamon and J. L. V. Hemmen, "Accelerating Backpropagation through Dynamic Self-Adaptation", *Neural Networks*, Vol. 9, No. 4, pp. 589-601, 1996.
- [3] Y. Liang, W. Peng, Z. Zheng, O. Silven and G. Zhao, "A Hybrid Quantum-Classical Neural Network with Deep Residual Learning", *Neural Networks*, Vol. 143, pp. 133-147, 2021.
- [4] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis, "Effective Backpropagation Training with Variable Stepsize", *Neural Networks*, Vol. 10, No. 1, pp. 69-82, 1997.
- [5] X. Yu and G. Chen, "Efficient Backpropagation Learning Using Optimal Learning Rate", *Neural Networks*, Vol. 10. No. 1, pp. 517-527, 1997.
- [6] A. V. P. Espinoza, J. B. O. Mere, F. J. M. D. Pison, and A. G. Marcos, "TAO-rubust Backpropagation Learning Algorithm", *Neural Networks*, Vol. 18, pp. 191-204, 2005.
- [7] V. Karri and T. Kiatcharoenpol, "A Monitoring System of Drill Wear States Using a Hybrid Neural Network", *Materials Science Forum*, Vol. 471-472, pp. 697-701, 2004.
- [8] W. K. Yap and V. Karri, "Emissions Predictive Modelling by Investigating Various Neural Network Models", *Expert Systems with*

- Applications*, Vol. 39, No. 3, pp. 2421-2426, 2012.
- [9] A. M. Hemeida, S. A. Hassan, A. A. Mohamed, S. Alkhalaf, M. M. Mahmoud, T. Senjyu, and A. B. E. Din, "Nature-inspired Algorithms for Feed-forward Neural Network Classifiers: A Survey of One Decade of Research", *Ain Shams Engineering Journal*, Vol. 11, No. 3, pp. 659-675, 2020.
- [10] P. Lanillos, D. Oliva, A. Philippsen, Y. Yamashita, Y. Nagai, and G. Cheng, "A Review on Neural Network Models of Schizophrenia and Autism Spectrum Disorder", *Neural Networks*, Vol. 122, pp. 338-363, 2020.
- [11] B. Verma, "Fast Training of Multilayer Perceptrons", *IEEE Transactions on Neural Networks*, Vol. 8, No. 6, pp. 1314-1320, 1997.
- [12] S. Ergezinger and E. Thomsen, "An Accelerated Learning Algorithm for Multilayer Perceptrons: Optimisation Layer by Layer", *IEEE Transactions on Neural Networks*, Vol. 6, No. 1, pp. 31-42, 1995.
- [13] M. Caudill and C. Butler, *Understanding Neural Networks - Computer Explorations*, Vol. 1 Basic Networks, MIT Press, Cambridge, 1992.
- [14] T. Kiatcharoenpol and T. Vichiraprasert, "Application of Taguchi Method and Shainin DOE Compared to Classical DOE in Plastic Injection Molding Process", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 3, pp. 11-19, 2019, doi: 10.22266/ijies2019.0630.02.
- [15] Fachrurrazi, S. Husin, Munirwansyah, and Husaini, "The Subcontractor Selection Practice using ANN-Multilayer", *International Journal of Technology*, Vol. 8, No. 4, pp. 761-772, 2017.
- [16] A. Dhini, I. Surjandari, M. Riefqi, and M. A. Puspasari, "Forecasting Analysis of Consumer Goods Demand using Neural Networks and ARIMA", *International Journal of Technology*, Vol. 6, No. 5, pp. 872-880, 2015.
- [17] A. Ghatak and P. S. Robi, "Prediction of Creep Curve of HP40Nb Steel using Artificial Neural Network", *Neural Computing and Applications*, Vol. 30, No. 9, pp. 2953-2964, 2018.
- [18] M. Alas and S. I. A. Ali, "Prediction of the High-Temperature Performance of a Geopolymer Modified Asphalt Binder using Artificial Neural Networks", *International Journal of Technology*, Vol. 10, No. 2, pp. 417-427, 2019.