The 2021 World Congress on
Advances in Structural Engineering and Mechanics (ASEM21)
GECE, Seoul, Korea, August 23-26, 2021

Integrated framework for efficient topology optimization using the convolutional LSTM network

Younghwan Joo[1)], Yonggyun Yu[1)] and *In Gwun Jang[2)]

[1)] *Korea Atomic Energy Research Institute, Daejeon 34141, Korea*
[2)] *Korea Advanced Institute of Science and Technology, Daejeon, 34141, Korea*
[2)] *igjang@kaist.edu*

ABSTRACT

An iteration-reduction method for 2D structural topology optimization is proposed. The acceleration model is combined with the conventional topology optimization method to provide a reliable optimal design with the boosted convergence. The acceleration model predicts a near-optimal design from designs in early stages using the spatial-temporal deep neural network. The training data are obtained by performing 2D SIMP-based topology optimization under randomly given boundary conditions and the filtering radius. To validate the method, the acceleration performance and the accuracy are evaluated from 200 test cases. Design histories before and after acceleration model are presented to investigate the geometric variation caused by the proposed method. The prosed method accelerates convergence by predicting a drastic change in geometry such as the removal of the supporting structures or formation of clear shapes from grey regions.

1. INTRODUCTION

Topology optimization (Bendsoe and Sigmund 2013) is a design method determining the optimal layout of a structure under a given boundary condition. It offers a feasible concept design through the iterative update of the geometry to maximize the structural stiffness under specifically given design constraints. As each iteration involves finite element and design sensitivity analysis procedures, the computational load becomes extensive as the number of elements increases. Thus, it is required to reduce the computational cost of topology optimization for its practical use in industries.

Due to the recent developments in machine learning technologies, their engineering application has received increasing attention. Several authors (Carleo and Troyer 2017; Mills et al. 2017; Singh et al. 2017) have applied machine learning techniques to reduce the computational cost of topology optimization. Aulig and Olhofer

---

[1)] Senior Researcher
[2)] Professor

(2014) developed a neuro-evolutionary topology optimization where an artificial neural network substituted the analytic design sensitivity. Sosnovik and Oseledets (2019) implemented a CNN-based auto-encoder network to predict the final 0-1 binary image from a design in an intermediate iteration. This modeling strategy outperformed the thresholding method. Yu et al. (2019) proposed a deep learning-based method to predict a near-optimal topological design without iterations using a CNN-based encoder-decoder network and generative adversarial network.

In the previous studies where the deep learning models are used to reduce the number of iterations, the method of inferring a near-optimal solution was mainly used. However, regarding the robustness of the solution, the results predicted by a deep learning model are only an approximate solution. Some disadvantages are also present; for instance, boundary conditions are challenging to generalize for generating learning datasets and require relatively substantial amounts of data. Thus, a method that can accelerate topology optimization while providing an optimal design rather than an approximate solution is required. In this regard, a framework including the conventional topology optimization method and the deep neural networks is proposed in this study. A near-final solution is rapidly predicted only with a few local density fields in the early stages through a deep neural network that learns the relationship between the design history and the result. For estimating the acceleration performance, the number of iterations required for obtaining a final design between the proposed method and the original topology optimization method is compared.

2. Method

2.1 Problem definition and data generation

The compliance minimization problem was considered in this study. A unit module of a 2D rectangular domain consisting of 64 × 64 finite elements was defined. To make the deep learning model respond to arbitrary geometries encountered in topology optimization, a multi-load problem which can generate more complicated geometries than a single-load problem was defined. A fixed boundary condition was established at a randomly selected wall among the four surfaces. Two point loads were located at a randomly selected surface except for the fixed boundary surface. The magnitude and direction of the load were randomly selected. All the random sampling associated with the load location and direction relied on the uniform distribution.

The training data were constructed by capturing series of the local density distribution and the corresponding final design. Fig. 1 shows how to construct the training data from the design histories. At a randomly given boundary condition, local density distributions are arranged according to the iteration number. The optimal solution at the final iteration is used as the target output data. A randomly determined filtering radius (r_{min}) is given at each optimization case. To construct the training data, three parameters were defined in this study: a window size (s_{window}), starting iteration number (n_{start}), and the number of window at each optimization case (n_{window}).

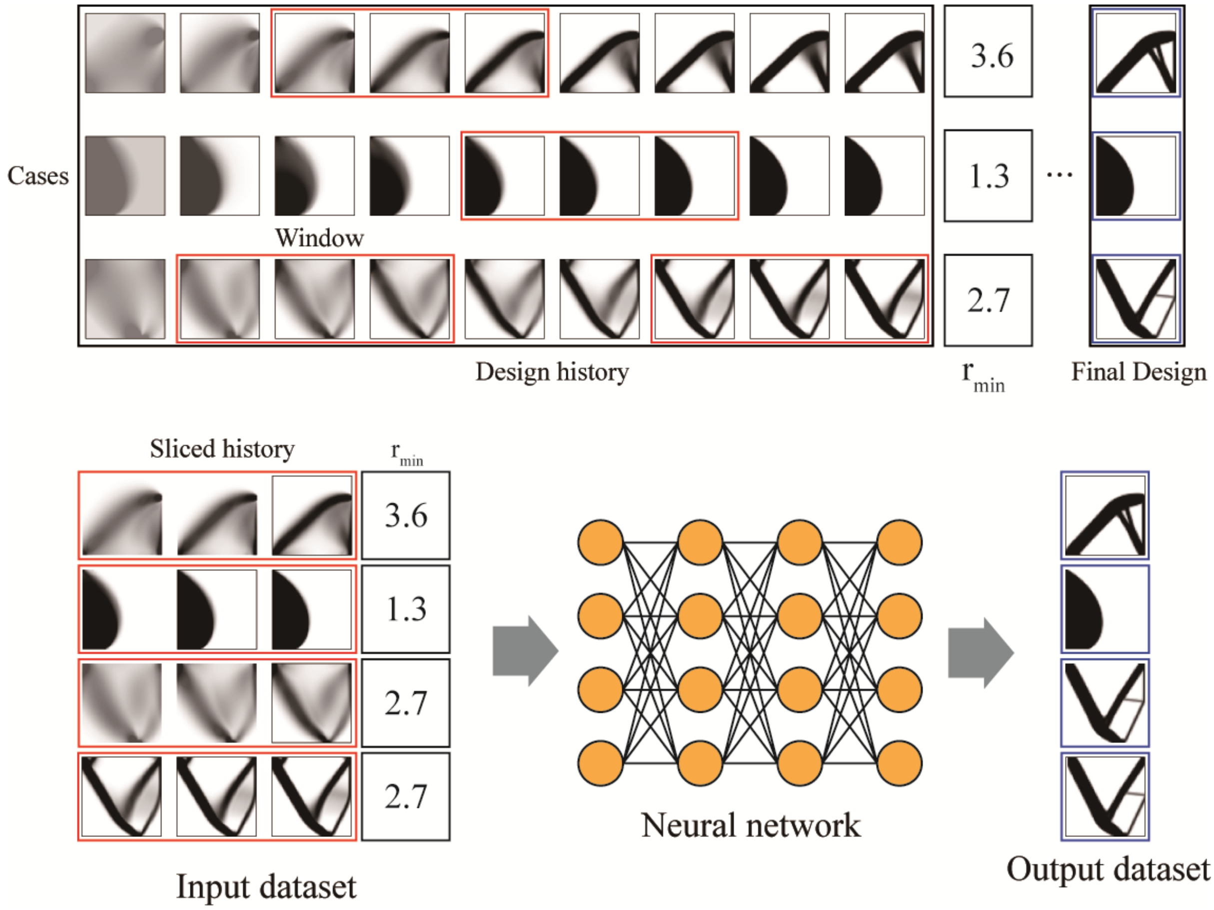

Fig. 1 Design history and training dataset

2.3 Acceleration framework with convolutional LSTM model

Fig. 2 depicts the overall process of deep-learning-accelerated topology optimization. The acceleration model is placed within the iterative process of topology optimization. When a boundary condition is given, topology optimization is performed for a predefined number of iterations. Using the last s_{window} local density fields, acceleration model returns a predicted design. From this design, topology optimization is performed until it converges.

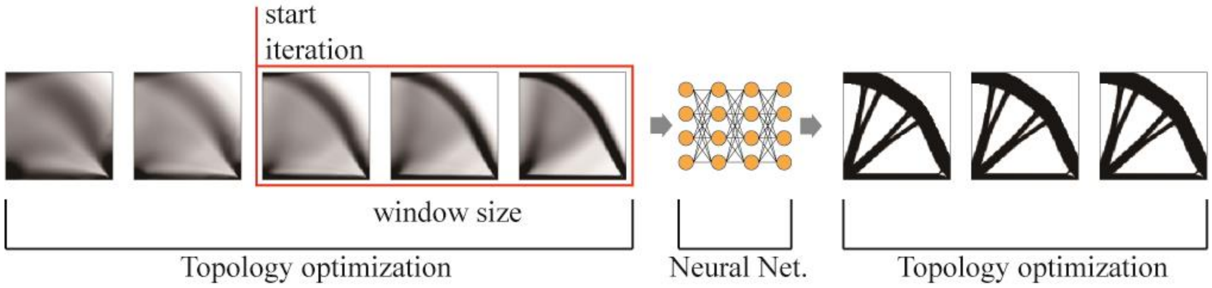

Fig. 2 Process of deep-learning-accelerated topology optimization

In order to deal with the data where time series and spatial characteristics coexist, the convolutional long short-term memory (convLSTM) (Shi et al. 2015) was used in this study. For image data arranged in time series, the CNN contributes to capturing the feature of each image and the LSTM contributes to capturing time-series variation characteristics of these images. The convLSTM layers can be stacked to build a deep learning model. As introduced in Fig. 2, this model receives s_{window} density fields with 64 × 64 resolution and the filtering radius (r_{min}) as input data and returns a single density field with 64 × 64 resolution as output. The overall structure of the model is based on the U-Net (Ronneberger et al. 2015), a widely used network for image segmentation. In this network, the input data converged into the latent space (encoder) and then diverged to its original resolution (decoder). The number of convolutional filter (n_{filter}) was set to 32. Recurrent activation function for a 2D convolutional LSTM layer was set to “hard sigmoid.”

3. Result

The deep learning model was trained by varying the hyper parameters (n_{window}, s_{window}) related to training data generation and the model structure to find the best training condition. During the training process, the mean absolute error between the target and predicted density field was used as the loss function. Adam optimizer (Kingma and Ba 2014) was used for model training. The training is performed for the maximum 500 epochs and the training process ends if the validation loss does not decrease for latest 50 epochs.

After training process, topology optimization was performed for additional 200 cases at two distinct design domain. One domain has the same resolution as the unit module. The other domain has 128 × 64 resolution. The boundary conditions were randomly given. Since this domain has the resolution different to the unit module, the target domain was divided into unit modules before model application and integrated into the original resolution after prediction.

Table 1 presents the result of model training according to the hyper parameters (n_{window}, s_{window}). The starting iteration number (n_{start}) was set to 1 for these cases. To estimate the performance of the model, the poor prediction ratio and the iteration number ratio were defined. The poor prediction ratio means the ratio of cases where the difference in the compliance between the accelerated and the original topology optimization is larger than 10 %. Since the suggested framework for acceleration has a goal to obtain a reliable optimal solution while enhancing the computational efficiency, the lower value of the poor prediction ratio is preferred. The iteration number ratio means the ratio of the total iteration number required for the original method to that for accelerated method (n_{origin} / n_{acc}). The higher the iteration number ratio, the better acceleration performance. From the results, model #10 was selected in this study to obtain a reliable design with a reasonable acceleration performance.

Table 1 Results of model training

Model #	**n_{window}**	**s_{window}**	**64 × 64 module domain**		**128 × 64 domain**	
			Poor prediction ratio (%)	Iteration number ratio	Poor prediction ratio (%)	Iteration number ratio
1	1	5	4.00	2.183	15.00	2.072

2	2	5	3.00	2.229	8.00	1.837
3	3	5	3.00	2.263	7.50	1.921
4	1	10	1.03	1.837	2.00	1.644
5	2	10	1.03	1.902	1.00	1.689
6	3	10	1.03	1.906	1.50	1.638
7	1	15	0.53	1.736	0.53	1.564
8	2	15	0.53	1.715	0.00	1.562
9	3	15	0.53	1.778	0.53	1.592
10	1	20	0.00	1.674	0.00	1.657
11	2	20	0.57	1.622	0.00	1.530
12	3	20	0.57	1.603	0.00	1.519

Figs. 3 and 4 show the design histories of the original and accelerated topology optimization. For these figures, the cases showing the features of the acceleration model were selected subjectively among the 200 optimization cases at each design domain. At each optimization case in Fig. 3, the results on top and bottom present the design history of the original and accelerated topology optimization, respectively. In the original topology optimization, first six density fields are results at successive iterations. In accelerated topology optimization, first three density fields are the latest three iterations before the acceleration model is used. The next three density fields are the first three iterations after the model prediction. Therefore, by concentrating on the geometric variation between the third and fourth density fields in accelerated topology optimization, we can investigate how the acceleration model works. The last three density fields in each design history are the last three designs reaching at convergence. Table 2 presents the number of iterations and the compliance for the optimization cases in Figs. 3 and 4.

In Fig. 3(a), the grey region became clearer after the acceleration model. The predicted design had almost the same layout as the optimal design. In Fig. 3(b), a supporting structure was removed after the acceleration model. This kind of a drastic change in geometry was also found in the domain with a higher resolution in Fig. 4(a). In Fig. 4(b), it can be seen that some grey regions were removed after model prediction. Through the iterations after the model prediction, the optimized design for the acceleration optimization became to be different with that for the original method. However, as shown in Table 2, the compliance error was about 0.1%, and this means the optimized designs from the acceleration model have the same level of structural performance as the original method. The iteration number ratio varies according to boundary conditions, but it was found that the acceleration method still works in the higher resolution.

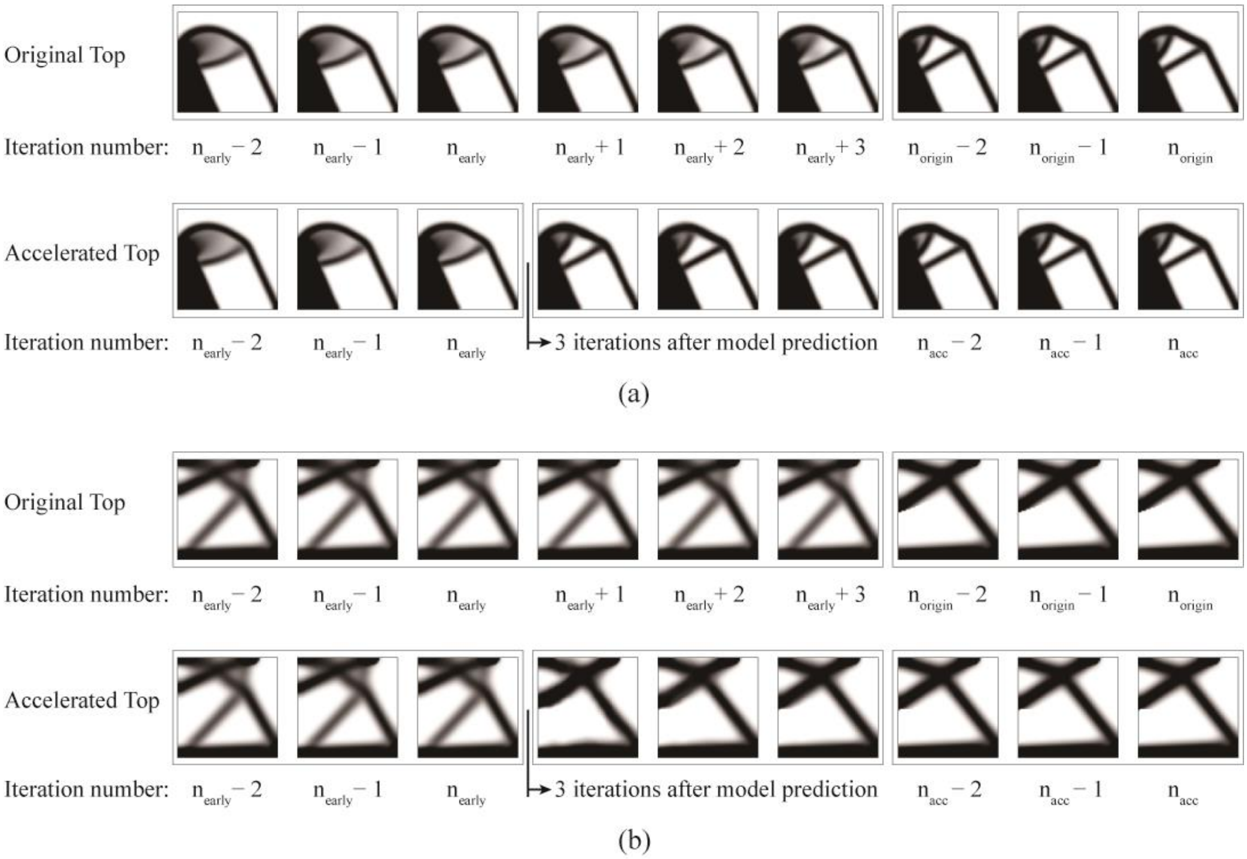

Fig. 3 Design history at domain with 64 × 64 resolution

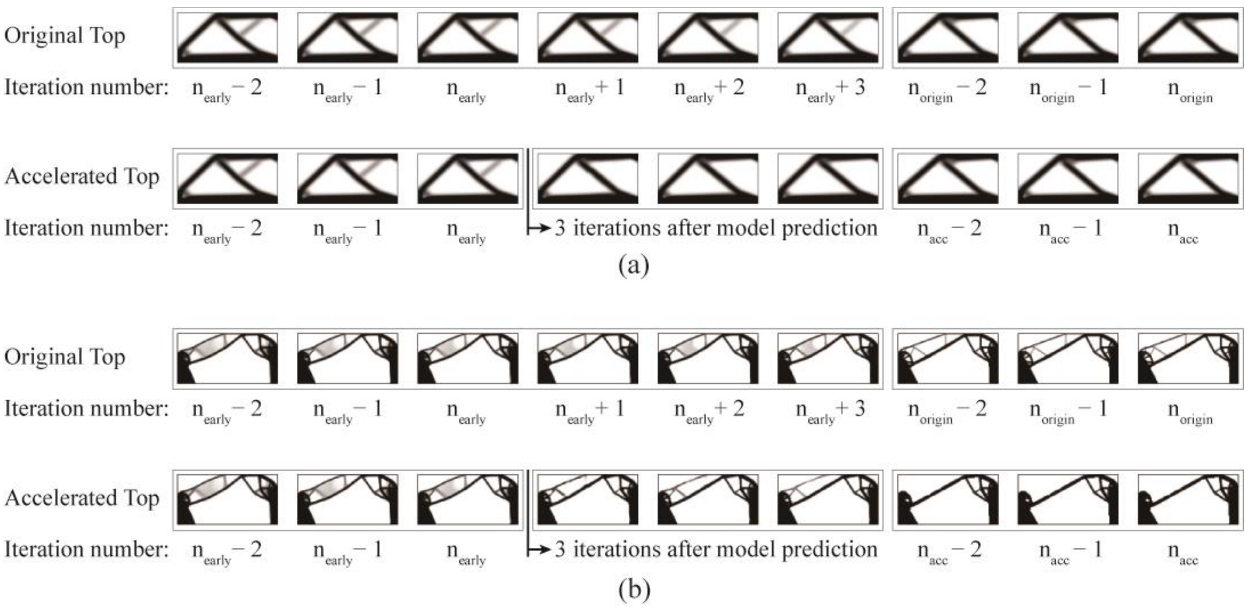

Fig. 4 Design history at domain with 128 × 64 resolution

Table 2 Number of iterations and compliance for results in Fig. 3 and 4

Fig.	**Number of iteration**			**Compliance**		
	Original	Accelerated	Ratio (-)	Original	Accelerated	Error (%)
3 (a)	71	37	1.92	9.90	9.90	0.02
3 (b)	131	32	4.09	31.58	31.72	0.45
4 (a)	96	79	1.22	55.15	55.09	0.11
4 (b)	309	72	4.29	11.52	11.53	0.12

4. CONCLUSIONS

In the study, the framework for accelerating 2D structural topology optimization under various resolutions was proposed. To deal with the design history data that has the spatial-temporal characteristics, convolutional LSTM network was used as the acceleration model. Both the original topology optimization method and the deep learning-based acceleration model were integrated to reduce the total number of iterations required for topology optimization without losing the accuracy caused by the approximation solution. The neural networks were trained to predict the near-optimal design with the latest local density distributions to account for the effect of nonlinear variation of the local densities. To obtain the best acceleration model, the iteration number ratio and the poor prediction ratio were evaluated by varying the training parameters. In the results of 200 optimization cases for model validation, it was found that the proposed method reduces the number of iterations while maintaining the structural performance despite the differences in the domain resolution and boundary conditions.

REFERENCES

Aulig, N., and Olhofer, M. "Topology optimization by predicting sensitivities based on local state features." *Proc., Proceedings of the 5th European Conference on Computational Mechanics (ECCM V), Barcelona, Spain*, 20-25.

Bendsoe, M. P., and Sigmund, O. (2013). *Topology optimization: theory, methods, and applications*, Springer Science & Business Media.

Carleo, G., and Troyer, M. (2017). "Solving the quantum many-body problem with artificial neural networks." *Science*, 355(6325), 602-606.

Kingma, D. P., and Ba, J. (2014). "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*.

Mills, K., Spanner, M., and Tamblyn, I. (2017). "Deep learning and the Schrödinger equation." *Physical Review A*, 96(4), 042113.

Ronneberger, O., Fischer, P., and Brox, T. "U-net: Convolutional networks for biomedical image segmentation." *Proc., International Conference on Medical image computing and computer-assisted intervention*, Springer, 234-241.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." *Advances in neural information processing systems*, 28.

Singh, A. P., Medida, S., and Duraisamy, K. (2017). "Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils." *AIAA journal*, 55(7), 2215-2227.

Sosnovik, I., and Oseledets, I. (2019). "Neural networks for topology optimization." *Russian Journal of Numerical Analysis and Mathematical Modelling*, 34(4), 215-223.

Yu, Y., Hur, T., Jung, J., and Jang, I. G. (2019). "Deep learning for determining a near-optimal topological design without any iteration." *Structural and Multidisciplinary Optimization*, 59(3), 787-799.