Transfer Learning Through Deep Learning: Application to Topology Optimization of Electric Motor

Jo Asanuma[1], Shuhei Doi[1], Hajime Igarashi[1], *Member, IEEE*

[1]Graduate School of Infomation and Science Technology, Hokkaido University, Sapporo, 060-0814, Japan
asanuma@em.ist.hokudai.ac.jp

This paper proposes use of transfer learning through deep learning to reduce the computing cost of the topology optimization of electric motors. It is shown that the training of the convolutional neural network (CNN) can be made effective using the transfer learning. Moreover, the recursion is realized by CNN. The computing cost is shown to be reduced by about 80% by the proposed method to obtain the pareto solutions of the multi-objective problem with respect to the average torque and torque ripple of a permanent magnet motor.

Index Terms—**Deep learning, transfer learning, topology optimization, electric motor**

I. Introduction

Recently, development of high-performance motors has become considerably important because the fuel fossil vehicles are expected to be replaced by plug-in hybrid and electric vehicles. In development of such motors, it is necessary to consider various the motor characteristics as well as constraints relevant to average torque, torque ripple, loss, demagnetization, mechanical vibration, mechanical strength and so on. To develop an electric motor considering these complicated conditions, topology optimization based on the genetic algorithm (GA) and finite element method (FEM) is effective especially for the initial design phase [1]. However, it sometimes needs unacceptably long computing time to perform optimization because a number of fitness evaluations using FEM are involved.

It has been shown by the authors that introduction of deep learning as a surrogate model to the topology optimization can reduce the computational cost [2, 3]. However, large input data has to be prepared to train the convolutional neural network (CNN) prior to the optimization process. In this paper, we propose a method using the fine tuning for CNN with relatively small training data. In this method, CNN preliminary trained by the general image data is re-trained by the data composed of the cross-sectional image of electric motors and their torque performances. Moreover, the recursion is realized by the proposed CNN for acceleration of the topology optimization.

II. Proposed Optimization based on deep learning

A. Transfer learning

In the shallow layers of CNN, the local features represented by stripes with various directions are extracted while the global features are extracted in the deep layers. Because the local features are expected to be common for general images of things, the shallow layers trained by a large data set containing various images would be used for other image data set. On the other hand, the deep layer is tuned to the image data of interest. This is the principle of the transfer learning [4] that is a technique to adapt a neural network learned for a certain problem separately to a different problem. In this study, we use the classifier VGG 16 [5] trained by ILSVRC 2012 in ImageNet which is a data set composed of 1000 different classes and 1.2 million learning data. We re-train the layers deeper than 9th layer of VGG16 using the data composed of the cross-sectional image and the corresponding average torque and torque ripple of an inner permanent magnet (IPM) motor.

Moreover, in this study, we realize the deep neural network which performs recursion of the torque performances, while classification is realized in the previous papers [2, 3]. To do so, the full connected layer is replaced by the four layers shown in Fig.1. The merit of recursion over classification in the optimization will be mentioned in the following section.

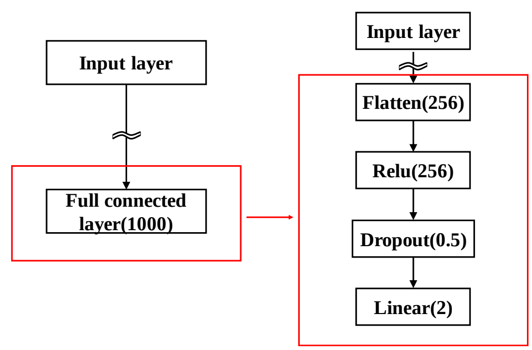

Fig. 1. Realization of recursion by CNN

B. Optimization Method

In this study, we consider the multi-objective topology optimization of an IPM motor using GA and Gaussian basis functions [1] where the average torque T_{ave} is maximized and torque ripple T_{rip} is minimized. To construct the learning data, we perform the preliminary optimization with small population setting which is summarized in the left column in Table I. In the main optimization process, for which the parameters in the right column of Table I are used, all the individuals are evaluated by the trained CNN, and only the individuals on the Pareto front are re-evaluated by FEM. The pseudo code of the proposed method for the main optimization problem is summarized in Fig.2.

III. Optimization Results

A. Accuracy of CNN

We train CNN by the data obtained from the optimization for the small population, shown in the left column of Table I is carried out by FEM analysis. To verify the transfer learning, CNN is trained in the different conditions: (a) training is performed for all the layers (usual training) and (b) the deep layers are trained (transfer learning), as summarized in Table II. The mean square errors for T_{ave} and T_{rip} for both learning are plotted against the number of epochs in Fig.3. It is evident that the transfer learning has the better convergence. The accuracy of recursion performed by CNN that is trained by the transfer learning is shown in Fig.4.

B. Result of main optimization

The pareto solutions obtained by solving the preliminary and main optimization problems are shown in Fig. 5. The cross section of an IPM motor on the pareto front, marked P in Fig.5, is shown in Fig.6. The corner of the L-shaped curve composed by the pareto solutions is improved by the main optimization. The solutions obtained by the main optimization are, however, partially dominated by the solutions obtained by the preliminary optimization. This might be due to the stochastic nature of GA.

When the individuals are classified into several classes according to the torque performance as in [2][3], the number of individuals on the Pareto front increases as the generation of GA because of the round error caused by the classification. This problem is expected to be overcome by introducing the recursion performed by CNN. Indeed, the number of execution of FEM, normalized by the FEM computations for conventional optimization, is suppressed for all the generations when using the proposed recursion as shown in Fig. 7. This point will be discussed in the long version in detail.

```
for max generation
    if initial generation
        generate initial population with random genes
        evaluate T_ave and T_rip of population by classifiers
        rank population by NSGA-Ⅱ
        for each individuals of population
            compute probability  for FE analysis
            if
                evaluate T_ave and T_rip by FEM
            end if
        end for
    else
        select parents
        generate children
        evaluate T_ave and T_rip of children by VGG16
        rank children by NSGA-Ⅱ
        for each individuals of children
            if rank==1
                evaluate T_ave and T_rip by FEM
            end if
        end for
        rank individuals of population + children by NSGA-Ⅱ
        choose next population
    end if
end for
```

Fig. 2. Pseudocode of main multi-objective optimization

References

[1] T. Sato, K. Watanabe and H. Igarashi, "Multimaterial Topology Optimization of Electric Machines Based on Normalized Gaussian Network," *IEEE Trans. Magn.*, vol. 51, no. 3, Art. no. 7202604, 2015.

[2] H. Sasaki, H. Igarashi, "Topology optimization accelerated by deep learning", presented at CEFC2018, submitted to *IEEE Trans. Magn.*

[3] S. Doi, H. Sasaki, H. Igarashi, "Multi-objective topology optimization of rotating machines using deep learning", presented at CEFC2018, submitted to *IEEE Trans. Magn.*

[4] Y.Bengio, I.Goodfellow andA.Courville, "DEEP LEARNING",Masachusetts Institute of Technology,2016.

[5] K.Simonyan, A.Zisserman, "Very deep convolutional networks for large-scale image recognition" CoRR, abs/1409.1556.

TABLE I
PARAMETERS FOR GA

	Preliminary optimization	Main optimization
Number of genes	28	28
Population size	352	704
Number of children	160	320
Number of generation	100	200

TABLE II
VGG16 PARAMETER

	(a) Training all layers	(b) Transfer Learning
Number of training images	16030	16030
Number of trainable parameter	21137986	20582658
Epoch	100	100
Weight	Random	ImageNet

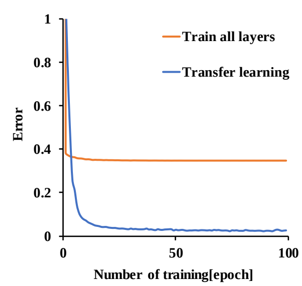

Fig. 3 Errors in CNN

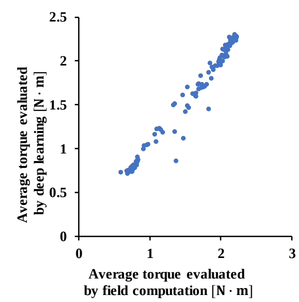

Fig. 4 (a) Accuracy of T_{ave}

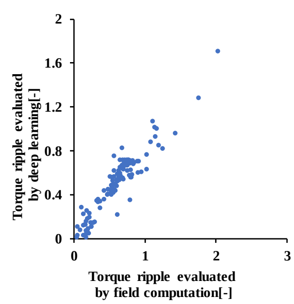

Fig. 4 (b) Accuracy of T_{rip}

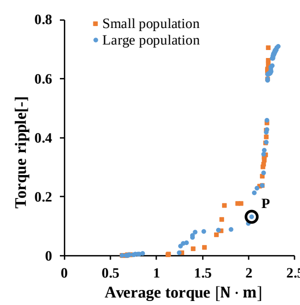

Fig. 5 Pareto solutions

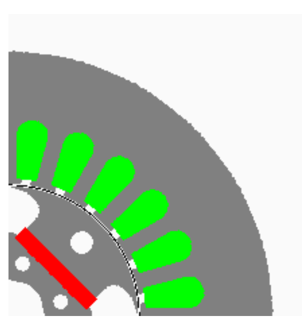

Fig. 6 IPM motor located at P in Fig.5

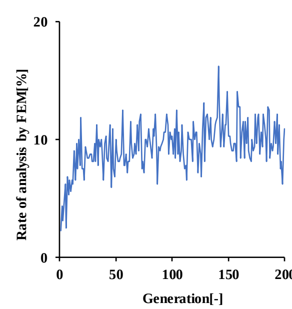

Fig. 7 Number of execution of FEM