Distilling Knowledge for Non-Neural Networks

Shota Fukui*, Jaehoon Yu*, and Masanori Hashimoto*
* Osaka University, Osaka, Japan
E-mail: {s-fukui, yu.jaehoon, hasimoto}@ist.osaka-u.ac.jp

***Abstract*—Deep neural networks (NNs) have shown high inference performance in the field of machine learning, but at the same time, researchers require their speeding-up and miniaturization methods due to the computational complexity. Distillation is drawing attention as one of the ways to overcome this problem. NNs usually have better expression power than its learning ability. Distillation bridges the gap between expressive power and learnability by training a small NN with additional information obtained from a larger already trained NN. This gap does not exist only in neural networks but also in other machine learning methods such as support vector machine, random forest, and gradient boosting decision tree. In this research, we propose a distillation method using information extracted from NNs for non-NN models. Experimental results show that distillation can improve the accuracies of other machine learning methods, and especially, the accuracy of SVM increases by 2.80%, 90.15% to 92.95%.**

I. Introduction

Recently, in the field of machine learning, neural networks (NNs) have achieved high inference performance. Especially, convolutional neural networks (CNNs) are attracting attention in image recognition. By convolution and pooling, CNNs are highly resistant to the displacement, and they can represent various spaces that conventional models could not express. In ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2012, AlexNet [1] outperformed existing machine learning methods in accuracy, which is one of the CNN models. Since then, CNN continues to improve its accuracy year by year [2], [3].

However, this improvement entails an increase in computational cost and network scale, which makes it challenging to apply NNs to embedded systems. For accelerating and miniaturizing NNs, there is a method called distillation. NNs usually have better expression power than its learning ability, i.e., a small NN may not be trainable even if it has enough power to express a proper model. Distillation bridges the gap between expressive power and learnability by training a small NN with additional information obtained from a lager already trained NN. However, even a small distilled NN still requires a huge amount of calculation and memory resources, and it is now a challenging problem to apply NNs to embedded systems of practical applications.

On the other hand, the gap between expressive power and learnability exists not only in NNs but also in other machine learning methods such as support vector machine (SVM) [4], random forest (RF) [5], and gradient boosting decision trees (GBDT). This paper proposes a distillation method from NNs to other machine learning methods for exploiting both advantages of NN learnability and smaller computation of other machine learning methods.

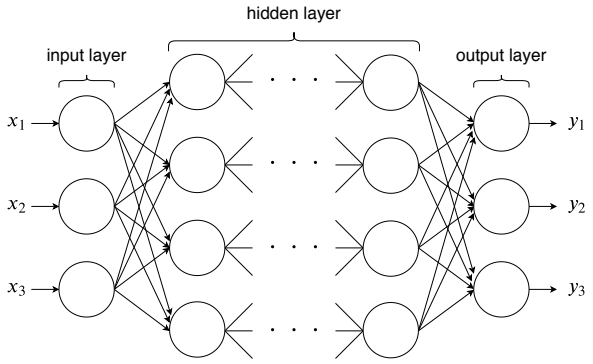


Fig. 1. Schematic of NN.

The rest of this paper is organized as follows. Section II introduces related existing methods. Next, Section III describes the proposed method distilling from a NN into other machine learning methods, and Section IV shows experimental results. Finally, Section V concludes this paper and discusses future work and remaining issues.

II. Related research

This section introduces machine learning methods and the distillation required to understand the proposed method. Section II-A presents a basic structure of NN and describes a typical architecture of CNN. Section II-B explains the conventional distillation method. Then Section II-C and II-D explain SVM, RF, and GBDT.

A. Neural network and convolutional neural network

A NN is a mathematical model in which neurons are connected between multiple layers in various forms as shown in Fig. 1. The first layer is called an input layer, the last layer is called an output layer, and the other layers between input and output layers are called hidden layers. By performing various transformations in the hidden layers, NNs improve their expressive power.

Among various types of NNs, CNNs have achieved good performance in the field of image recognition. A CNN mainly consists of three components: convolutional layer, activation function, and pooling layer. The convolutional layer generates its output by convoluting 3-D filters called kernels with the input. This convolution process makes it possible to overcome displacements of features. Given that A, B, and C are width,

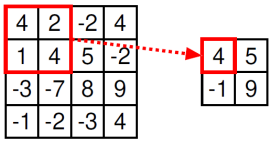


Fig. 2. Max pooling without overlap.

height, and depth of the k-th kernel, and P, Q, and R are width, height, and depth of output, the convolution process is expressed as the following (1):

$$y_{p,q,r} = \sum_{a,b,c} w^{(k)}_{a,b,c} x_{p+a,q+b,c}, \tag{1}$$

where $x_{p,q,c}$, $y_{p,q,r}$ and $w^{(k)}_{a,b,c}$ are a value at coordinate (p,q,c) of input $\mathbf{x}$, a value at (p,q,r) of output $\mathbf{y}$, and a value at (a,b,c) of k-th kernel, respectively.

The activation function nonlinearly transforms the output $y_{p,q,r}$ of each neuron in convolutional layers into $z_{p,q,r}$, and it is the most important part contributing inference accuracy. In recent CNNs, rectified linear unit (ReLU) is the most widely used activation function:

$$z_{p,q,r} = \max(0, y_{p,q,r}). \tag{2}$$

The pooling layer shrinks the output $z_{p,q,r}$ of the activation function while retaining important information. There are various types of pooling methods: max pooling, average pooling, and so on. Fig. 2 shows an example of 2×2 max pooling with a 4×4 input. By extracting the maximum value from regions of interests, max pooling compresses the 4×4 input space to 2×2 and makes it possible to achieve high resistance against displacement and noise of extracted features [6].

B. Distillation

Distillation [7] trains a smaller NN with the additional information obtained from a pretrained larger NN: the smaller and the larger NNs are called student and teacher models, respectively. In contrast to the conventional training process using only training datasets, distillation can provide detail information about the pretrained space of a teacher model and help a student model learn a similar space.

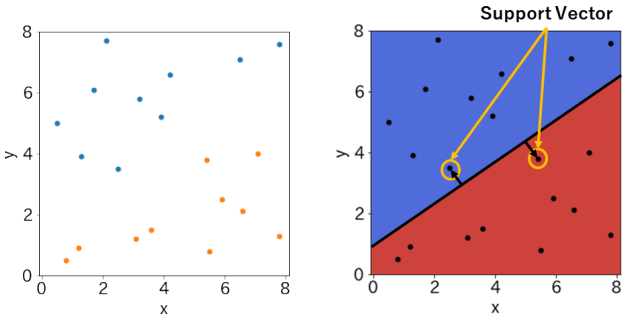
In distillation, temperature softmax plays an important role to transfer the information from a teacher model to a student model. Softmax function generally used in classification is defined as

$$p_i = \frac{\exp(y_i)}{\sum_j \exp(y_j)}, \tag{3}$$

where y_i is the i-th value of the output layer, and p_i is the probability that the current input data belongs to i-th class. Here, a well-trained teacher model provides almost no additional information over its target label since its output through the softmax operation tends to be a one-hot vector. The temperature softmax function, on the other hand, smooths the distribution of output and makes it easy to extract additional information. Temperature softmax is defined as

$$p_i = \frac{\exp(y_i/T)}{\sum_j \exp(y_j/T)}. \tag{4}$$

Note that temperature T is introduced as a new hyperparameter, and the probability distribution becomes smoother as the temperature T increases.



(a) Input data. (b) Divided space.

Fig. 3. An example of space division by SVM.

C. Support vector machine

A SVM separates an input space by finding the maximum margin between two groups of data, where the data closest to the boundary are called support vectors. For example, given two groups of data shown in Fig. 3(a), a SVM divides them into two spaces as in Fig. 3(b), where two data pointed by arrows are support vectors. Even for spaces with nonlinear boundaries, a SVM can divide the spaces by mapping training data to high-dimensional feature spaces. The functions used for the conversion to a high dimensional feature space are called kernels. Representative kernels, a RBF kernel and a polynomial kernel are shown in (5) and (6):

$$K_{\mathrm{RBF}}(\mathbf{x},\mathbf{y}) = \exp(-\gamma\|\mathbf{x}-\mathbf{y}\|^2) \tag{5}$$

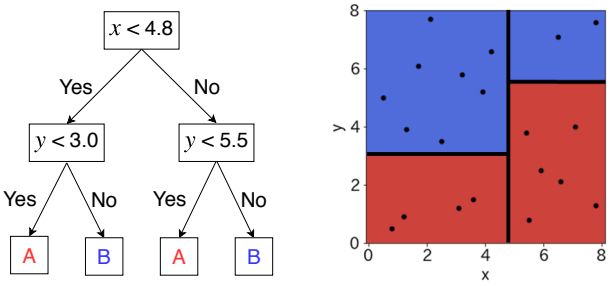
and

$$K_{\mathrm{poly}}(\mathbf{x},\mathbf{y}) = (\gamma(\mathbf{x}\cdot\mathbf{y})+r)^d, \tag{6}$$

where γ, r, d are hyperparameters.

D. Decision tree ensemble

Decision Tree (DT) ensemble builds a classifier or a regressor consisting of multiple DTs, where each DT is a weak learner. Fig. 4 shows an example of a DT and the space partitioned by it. As the depth of a DT increases, the expression power improves. However, it also causes overfitting, and hence it is necessary to set an appropriate depth. Besides, DT ensemble further enhances the expression power by combining relatively shallow multiple DTs. This section briefly explains two types of DT ensembles: RF and GBDT.



(a) DT. (b) Divided space.

Fig. 4. Space represented by a DT.

1) Random forest: RFs train each DT with randomly sampled training data. This random sampling can avoid training similar trees since training data have diversity. Although a RF uses multiple DTs, it can operate faster than NN because each DT can be processed by comparison operation and they are much faster than multiply-accumulate operations.

2) Gradient boosting decision trees: A RF gives diversity to each DT by using randomly sampled training data used. On the other hand, GBDT increases the weights of training data that cannot be classified by the current DT and trains the next DT with the newly-weighted training data. Therefore, GBDTs cannot learn in parallel so that learning takes time compared with RFs.

III. Distillation for other machine learning methods

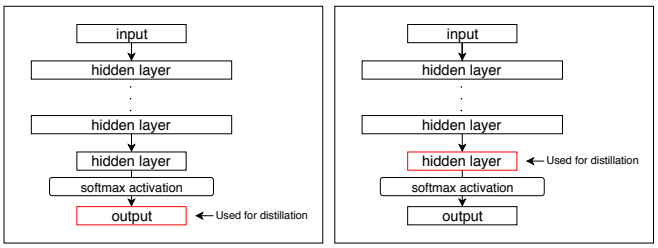
This section explains the proposed distillation method from NNs to other machine learning methods. Section III-A describes the basic framework for applying distillation to other machine learning methods. Then, Section III-B explores variations of the proposed distillation.

A. How to distill to other machine learning methods

The processing flow of the proposed method is as follows:

1) Training a NN as a teacher model,
2) Building a new training dataset consisting of input data and corresponding output from the teacher model, and
3) Training a non-NN model in the manner of regression with the new training dataset.

The proposed method starts with training a teacher model as the conventional distillation does. After preparing a teacher model, it creates a new dataset by associating the training data with the corresponding output of the teacher model. Here, the output of the teacher model used in the proposed method will be discussed in the next subsection. The proposed distillation uses this new dataset for training SVM, RF, and GBDT. The corresponding output of the dataset are real numbers, and hence the proposed distillation trains other machine methods in the training manner of regression for both classification and regression. For multiclass classification, the proposed method trains learners as many as the number of classes and selects the output with the maximum value as the classification result.



(a) using temperature softmax. (b) using activation.

Fig. 5. Two variations of the proposed distillation.

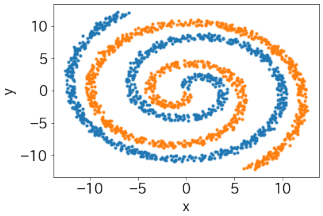


Fig. 6. Spiral dataset.

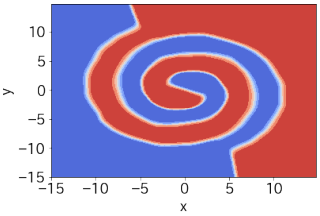


Fig. 7. Teacher model.

B. Variations of the proposed distillation

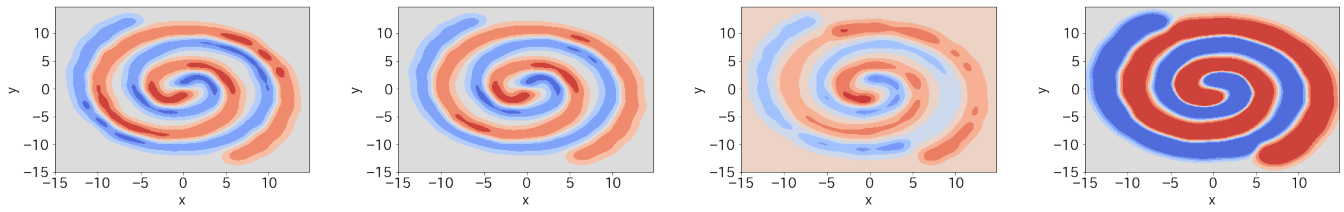
This section explores two variations of the proposed distillation, which are illustrated in Fig. 5: the first one associates the temperature softmax and the other associates the output of activation with the training data. For preliminarily evaluating both the variations, 2-D spiral data in Fig. 6 is used as training data, and a teacher model is trained as shown in Fig. 7. Student models are trained using the output of a NN instead of the target label of the training data.

Figs. 8, 9, and 10 compare results of a genuine training method and the variations of the proposed method for SVM, RF, and GBDT, respectively. In each figure, (a) is trained using intact training data, (b) and (c) are trained by the proposed distillation with temperature softmax, and (d) is trained by the proposed distillation with activation output. Here, we provide two results of temperature softmax with $T = 1$ and $T = 7$ to investigate the influence of hyperparameter T.

As shown in (a) and (b) of each figure, the spaces trained using temperature softmax with a small T are not much different from the spaces trained using intact training data. In this case, the proposed distillation hardly contributes to better inference performance than genuine training. As hyperparameter T increases, the proposed distillation can extract more information from the teacher model, and the spaces become smoother as shown in (c) of each figure. On the other hand, the proposed distillation using activation output shows similar spaces with the teacher model as shown in (d) of each figure. The following section will evaluate these two variations with more complicated dataset and discuss the details.

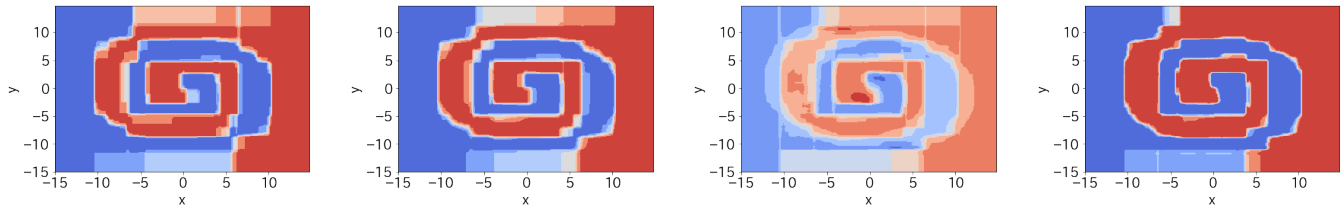
IV. Evaluation

This section evaluates and compares the effectiveness of the proposed distillation under various setups. First, Section IV-A



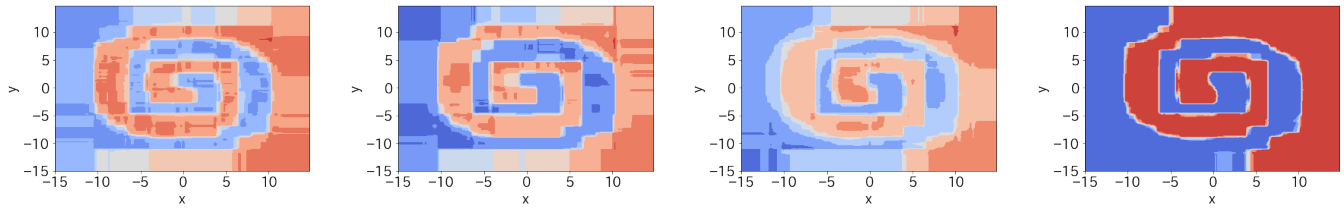
(a) genuine SVM. (b) temperature softmax ($T = 1$). (c) temperature softmax ($T = 7$). (d) activation output.

Fig. 8. Comparison of spaces trained by SVMs: (a) is trained using intact training data, (b) and (c) are trained by the proposed distillation using temperature softmax, and (d) is trained by the proposed distillation using activation output.



(a) genuine RF. (b) temperature softmax ($T = 1$). (c) temperature softmax ($T = 7$). (d) activation output.

Fig. 9. Comparison of spaces trained by RFs: (a) is trained using intact training data, (b) and (c) are trained by the proposed distillation using temperature softmax, and (d) is trained by the proposed distillation using activation output.



(a) genuine GBDT. (b) temperature softmax ($T = 1$). (c) temperature softmax ($T = 7$). (d) activation output.

Fig. 10. Comparison of spaces trained by GBDTs: (a) is trained using intact training data, (b) and (c) are trained by the proposed distillation with temperature softmax, and (d) is trained by the proposed distillation with activation output.

describes the evaluation environment, and then Section IV-B shows the distillation results.

A. Environment

For evaluation, we use the CIFAR-10 dataset[1]. CIFAR-10 consists of 60,000 32×32 color images of ten classes: 6,000 images per class. There are 50,000 training images and 10,000 test images. Only two of ten classes are used in the evaluation for simplicity. A CNN is adopted as a teacher model, and Fig. 11 shows the network configuration. Accuracy of the teacher model is 98.75% in two-class classification. The Keras library is used for implementing the CNN, and scikit-learn is used for implementing SVMs, RFs, and GBDTs.

B. Experimental results

Section IV-B1, IV-B2, and IV-B3 evaluate the proposed distillation with SVMs, RFs, and GBDTs in order. For temperature softmax, each section uses the following three temperatures as a hyperparameter: $T = 50$, $T = 100$, $T = 250$.

[1]https://www.cs.toronto.edu/~kriz/cifar.html

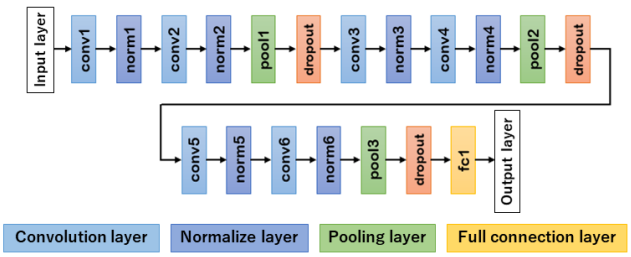


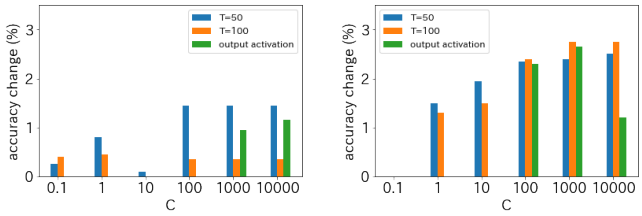
Fig. 11. Schematic of convolutional neural network.

1) Distillation to SVM: TABLE I shows accuracy changes due to distillation for SVMs using activation output. This experiment uses RBF kernel and polynomial kernel and changes C, which is a parameter that penalizes errors. SVMs achieved remarkable accuracy improvements when $C = 1,000$. Besides, SVMs trained with distillation break the accuracy record of conventional training in both the RBF kernel and the polynomial kernel.

We also perform distillation using output after tempera-

TABLE I
ACCURACY COMPARISON OF DISTILLING SVM WITH ACTIVATION(%)

		kernel function					
		no distill		distill w/ act.		diff.	
		RBF	poly	RBF	poly	RBF	poly
C	0.1	89.00	88.90	73.90	76.45	−15.10	−12.45
	1	92.55	91.80	82.20	82.05	−10.35	−9.75
	10	94.20	91.55	88.15	87.10	−6.05	−4.45
	100	93.25	90.35	93.00	92.85	−0.25	2.50
	1,000	93.30	90.15	94.40	92.95	**1.10**	**2.80**
	10,000	93.33	89.50	94.20	91.45	0.90	1.95



(a) Using RBF kernel. (b) Using Polynomial kernel.

Fig. 12. Accuracy change of distilling to SVM (%).

ture softmax function at temperature $T = 1$, $T = 50$, $T = 100$, and $T = 250$. As the case of using activation output, distillation using temperature softmax achieved high accuracy, especially at temperature $T = 50$ and $T = 100$. Fig. 12(a) and Fig. 12(b) summarize the change in accuracy when $T = 50$ and $T = 100$ and when using activation output. We omit the part where the inference accuracy degraded. These figures show that distillation using a temperature softmax also provides the same or better effect. Distillation to SVM achieved remarkable accuracy improvements, especially when C is large. This result can be explained by the fact that a large C tends to overfit the SVM model to outputs of NN.

2) Distillation to RF: TABLE II shows accuracy changes of distillation for RFs using activation output. We perform distillation by changing the number of trees and the depth of trees. RFs achieved accuracy improvements when the number of trees was small and the trees were deep. On the other hand, unlike SVMs, RFs trained with distillation failed to break the accuracy record of the conventional training. As in the case of the SVM, we performed distillation using a temperature softmax at temperature $T = 1$, $T = 50$, $T = 100$, and $T = 250$, and Fig. 13(a) to Fig. 13(d) show the results. From these figures, RFs improve the accuracy when the number of trees is small and the trees are deep regardless of the used output. This result also can be explained in the same manner with the above. It is by the fact that the small number of deep trees tends to overfit the RF model to outputs of NN.

3) Distillation to GBDT: TABLE III shows accuracy changes of distillation for GBDTs using activation output. We perform distillation by changing the number of trees and the depth of trees. When the number of trees was 4 and the depth of trees was 16, GBDTs achieved large accuracy changes. As in RFs, GBDTs failed to update the maximum value of inference accuracy.

We performed distillation using a temperature softmax at temperature $T = 1$, $T = 50$, $T = 100$, and $T = 250$, and achieved high accuracy at $T = 50$, $T = 100$, and $T = 250$. Fig. 14(a) to Fig. 14(d) show the results. Although GBDTs also improve accuracy, it is difficult to identify the condition in which the accuracy improves, and further analysis is necessary.

V. CONCLUSION

This paper proposed a distillation method from a NN to other machine learning methods. The evaluation experiment shows that it is possible for the proposed distillation to transfer the information of a NN to SVMs, RFs, and GBDTs in image classification using CIFAR-10. Especially, the accuracy of SVM increases by 2.80%, 90.15% to 92.95%. The accuracy of RFs and GBDTs also increases, but they failed to break the accuracy record of conventional training. As future work, we are planning to confirm the validity of the proposed method against more difficult classification problems.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP19H04079.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, Dec. 2012, pp. 1097–1105.
[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[3] A. Canziani, E. Culurciello, and A. Paszke, "Evaluation of neural network architectures for embedded systems," in *Proceedings of IEEE International Symposium on Circuits and Systems*, May 2017, pp. 1–4.
[4] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[6] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of International Conference on Machine Learning*, Jun. 2010, pp. 111–118.
[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proceedings of Advances in Neural Information Processing Systems Workshop*, Dec. 2015, pp. 1–9.

TABLE II
ACCURACY COMPARISON OF DISTILLING RF WITH ACTIVATION (%).

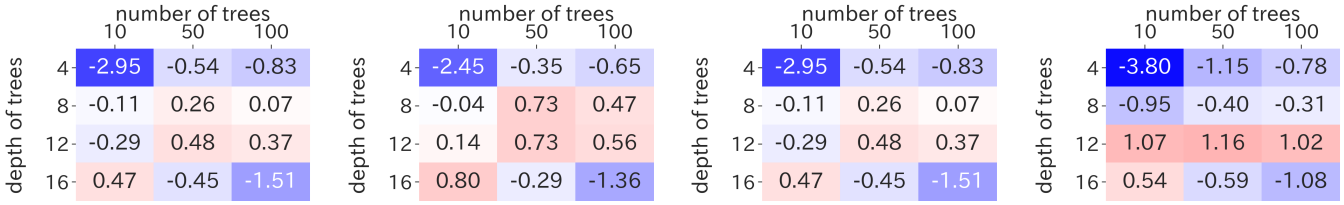
		number of trees								
		no distill			distill w/ act.			diff.		
		10	50	100	10	50	100	10	50	100
depth	4	80.68	81.88	82.06	78.85	78.85	78.95	−1.83	−3.03	−3.11
	8	85.26	87.41	87.77	85.87	86.54	86.91	0.61	−0.87	−0.86
	12	85.30	88.23	88.55	86.92	88.01	88.21	**1.62**	−0.22	−0.34
	16	85.12	88.50	88.82	86.74	88.09	88.20	**1.62**	−0.41	−0.62

(a) $T = 1$. (b) $T = 50$. (c) $T = 100$. (d) $T = 250$.

Fig. 13. Accuracy change of distilling to RF with softmax (%).

TABLE III
ACCURACY COMPARISON OF DISTILLING GBDT WITH ACTIVATION OUTPUT (%).

		number of trees								
		no distill			distill w/ act.			diff.		
		10	50	100	10	50	100	10	50	100
depth	4	84.25	88.80	90.83	80.40	88.55	91.01	−3.85	−0.25	0.18
	8	86.06	89.62	90.32	85.60	89.83	90.49	−0.46	0.21	0.17
	12	84.78	87.05	87.29	84.84	87.91	88.16	0.06	0.86	0.87
	16	82.18	84.56	85.53	83.95	85.37	85.32	**1.77**	0.81	−0.21



(a) $T = 1$. (b) $T = 50$. (c) $T = 100$. (d) $T = 250$.

Fig. 14. Accuracy change of distilling to GBDT with softmax (%).