Training Data Reduction using Support Vectors for Neural Networks

Toranosuke Tanio* and Kouyab Takeda†
* Osaka University, Osaka, Japan
E-mail: t-tanio@ist.osaka-u.ac.jp Tel/Fax: +81-090-6964-9088
† Northwestern Polytechnical University, Xi'an, Japan
E-mail: Tel/Fax: +86-29-XXXXXXXX

***Abstract*—In the field of machine learning, deep learning is widely used to improve versatility and accuracy by deepening the network. Deep learning can achieve higher expression ability compared to conventional models but requires large amounts of data and time for training. To tackle this issue, we propose a training data reduction method using support vectors (SVs) that are closest data to the classification boundary obtained by Support Vector Machine (SVM). In this research, we use the training data consisting of support vectors to training neural networks and evaluate the effect. In the evaluation experiment, we confirmed that it is possible to reduce the number of training data by about 12% and reduce the learning time of neural network by about 9.5% by using ResNet, a model of deep learning, and the CIFAR-10 data set.**

I. Introduction

With the advent of deep learning, the development of machine learning centered on neural networks has reached its heyday. While a neural network called a multilayer perceptron performed an approximation of an arbitrary function by combining nonlinear transformation with one hidden layer, in Deep Neural Network(DNN), the expressive ability of the neural network is dramatically improved by increasing the number of hidden layers and performing an iterative nonlinear transformation. In the massive image recognition competition ILSVRC [1], as shown in Fig.1, since the initial DNN AlexNet [2] was proposed, we have achieved logarithmic accuracy improvement every year. This makes it possible to approximate complex functions with the number of parameters that can be learned, and neural networks are expected to be applied to real problems. However, training data in proportion to the number of parameters are required to learn the latest neural network where hundreds of millions of parameters are used, and a huge amount of computing resources and time are required. Furthermore, as can be seen from the change by year for the scale of the network model that won the ILSVRC in Fig.2, the size of the neural network is increasing exponentially, and problems due to the increase in the number of training data are expected to become more serious.

In this research, in order to tackle this problem, we propose a method to reduce training data while maintaining DNN accuracy. Focus on differences in importance in training data as a basic idea of the proposed method, and we will improve the efficiency of learning by performing neural network learning only with training data of high importance. In order

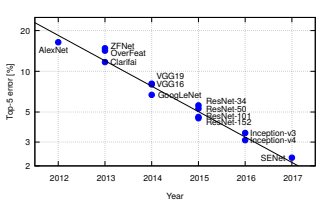


Fig. 1. Yearly error transition for ImageNet.

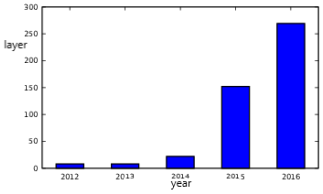


Fig. 2. A scale change according to the year of ILSVRC championship model

to realize this, we focus on the role of support vectors in Support Vector Machine (SVM) [3]. A support vector refers to a group of data used to define boundary hyperplanes in spatial separation of SVM. Therefore, the support vector has a feature located at the outermost part of the training data of the same class and is expected to play an essential role in the learning of neural networks. While existing research on training data reduction based on support vectors has provided initial results for MNIST, which is a handwritten digit recognition data set, it does not clarify the learnability for more complicated problems. In this study, we aim to clarify the effect of support vectors in DNN learning using ResNet [4] which is one of the newest neural networks and CIFAR-10 which is a data set for image classification.

The composition of this paper is as follows. Section 2 explains the concepts of machine learning and deep learning, and the basics of neural networks and SVM used in this research. Next, Section 3 explains the training data reduction method using support vectors, and verifies the effects of support vectors in neural network learning visually by performing preliminary experiments on multiple two-dimensional data. Section 4 shows the results of evaluation experiments using the latest neural network ResNet and the image data set CIFAR-10 and discusses the issues. Section 5 concludes with the conclusions.

II. Related research

This chapter describes machine learning and support vector machines as related research required to understand the proposed method. First, we will explain the components and basic principles of a neural network, which is a network model mainly used in machine learning, and then the mechanism

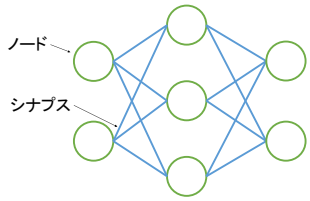


Fig. 3. Structure of neural network.

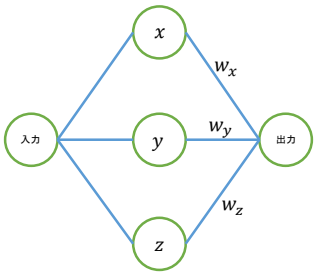


Fig. 4. Model to decide whether to go shopping.

of deep learning, which is one of machine learning methods. Next, we will describe SVM. Finally, we will talk about the existing training data reduction method.

A. Machine learning

Machine learning is a method that enables machines to learn themselves and analyze the regularity and the rule of data, instead of human beings writing all actions in the program in advance. This method makes it possible to perform specific actions by repeatedly learning and training. As an example, consider identifying images of horses and zebras. In order to identify a given image as a horse or a zebra, the machine reads a large number of images of the horse and the zebra and repeats learning. At this time, by giving instructions to pay attention to the presence or absence of the stripe pattern of the body and learning, when reading a new image of a horse or a zebra, they will be noticed and identified. As described above, it is called supervised learning that learning is performed by reading data for which answers are known in advance, which is one of the mainstream machine learning methods. Conversely, there is also unsupervised learning, but we will omit it because it is not relevant to this research.

B. Neural Network

A Neural network is a network created by mimicking the structure of the human brain. As shown in Fig. 3, it consists of nodes and synapses. Nodes are called artificial neurons, which are mathematical models created by imitating neurons in the human brain and can perform arbitrary operations. Nodes can pass computation results by connecting them, and it is the synapse used for this connection.

To illustrate that neural networks can mimic the human brain's thinking circuit, as an example, we consider the question "Do you go shopping on the weekend." It is assumed that there are the following three factors as to whether to go shopping.

1) x: Closeness from home
2) y: Weather
3) z: With or without discount sale

These outputs can be expressed as 0 or 1. For example, the output of x is one if the store is near the house, and 0 if it is far. The other two are the same. It should be noted that each factor is not an equivalent value. For example, no matter how close the store is from the house, you may never go if the weather is terrible, or even if no matter how far the store is from the house and bad the weather is, you may go if the store is making a discount sale. Such the importance of a factor is called a weight. Here, assuming that the weights of x, y and z are $w_x = 2$, $w_y = 4$ and $w_z = 8$, respectively, Fig. 4 shows a model that determines whether to go shopping. At this time, an expression indicating whether to go shopping is defined as

$$xw_x + yw_y + zw_z > 5 \tag{1}$$

The number 5 is called the threshold, and the criterion can be changed by changing the threshold. If the threshold is 5, the shop will go shopping if it is near the house and the weather is good and if the shop is making a discount sale. In other words, if the store is near the house and the weather is good, it will go even if the store does not have a discount sale, and if it has a discount sale, other factors will be ignored. Next, assuming that the threshold is 10 and

$$xw_x + yw_y + zw_z > 10 \tag{2}$$

if the weather is good and the store is making a discount sale, you will go shopping, and whether the store is near the house does not affect on the result. In this way, the threshold changes the criteria for deciding whether to go shopping. Although a simple network is considered in this example, if nodes are increased and more complex networks are used, more advanced judgment can be made.

C. Deep Learning

Deep learning is one of the machine learning methods using Deep Neural Networks (DNN). The difference between DNN and conventional neural networks is that the layers are deeply overlapped. Whereas ordinary neural networks have only one middle layer, DNN has two or more middle layers, which is a feature of DNN. Compared to ordinary neural networks, learning data, learning time, and power consumption are more massive, so that learning can be performed with higher accuracy. Usually, in machine learning, it is necessary to set in advance a factor called a feature that is an index of discrimination. However, in deep learning, features can be extracted automatically by updating the equation of nonlinear transformation performed in each layer. Here, as an example, Fig. 6 shows the results of identifying the spiral model. The figure represents that the orange background is positive, blue is negative, and white is 0, and the point is the data at that

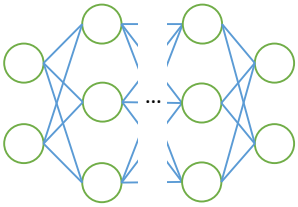


Fig. 5. Structure of Deep Neural Network.

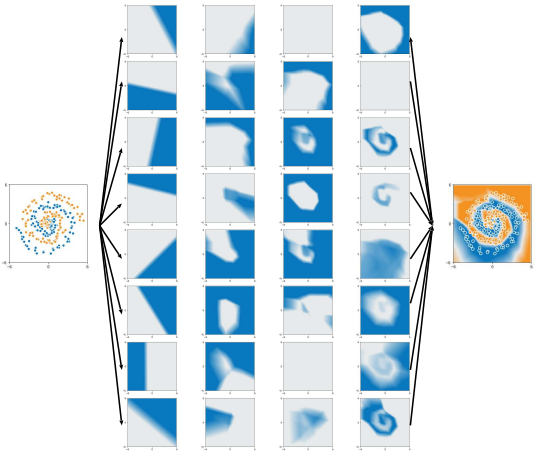


Fig. 6. General flow of feature detection.

coordinate. From the input on the left side, it is understood that the division of the space is performed with a complicated curve each time the layer is advanced, and it is finally expressed as a spiral model. When analyzing data such as images and sounds in deep learning, the data contains many features. In order to extract this feature without loss, it is ideal for dividing it into each, but it is very defficult because the features are not always arranged at equal intervals and we do not even know what the features are at first. Therefore, it is essential to divide data without loss of features as much as possible. Here, image identification is considered as an example. If the criterion for determining whether it is kangaroo is the presence or absence of a pouch, the judgment can not be made if the portion of the pouch is on the boundary of the data division and divided into two. Thus, depending on the method of data division, features may be lost. In order to solve this, we use an overlapping method. This is a method of dividing the data by overlapping them to some extent. If it explains in the character string ”abcde” for the sake of simplicity, this method is not divided into ”abc” and ”de” but divided into ”abc”, ”bcd” and ”cde.” This can reduce the loss of features, which contributes to the increase in data analysis accuracy.

When the data is divided and passed to each node in the first layer, the processing is performed for each node, and the data is propagated to the nodes in the next layer. At this time, each node is still in the condition that it has features required and unnecessary features(errors) to produce accurate results. If all the information is passed to the next layer, all

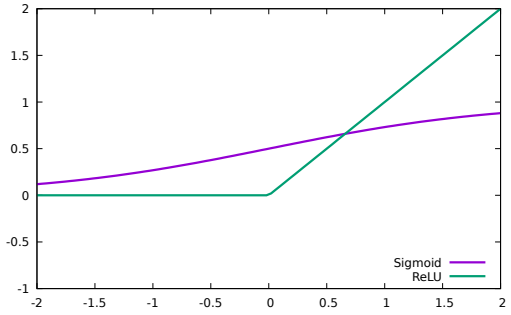


Fig. 7. Activation function.

the errors will be transmitted, and if the information passed is too small to pass the errors, the features will be lost. Here, the calculation formula applied when passing data to the next layer is called the activation function. The activation function can convey features to the next layer more clearly by slightly changing the original data, and has the effect of suppressing overfitting. There are several types of activation functions, and two primary sigmoid functions and ReLU functions are described.

The sigmoid function is shown in Fig. 7 is a function that approaches one as the value of the input increases and approaches 0 as it decreases. The characteristic is that it is a monotonically increasing function, and it does not become one regardless of how big the original input is, and does not become -1 regardless of how small it is. The sigmoid function is shown in (3).

$$h_{sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The sigmoid function can maintain the original input characteristics to some extent but has the disadvantage that the computational complexity increases. The characteristic is that the change in value becomes gentler as it gets closer to -1 and 1, and the gradient when differentiating the sigmoid function is

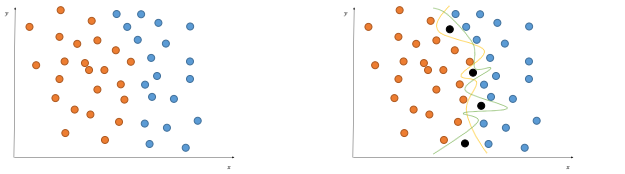
$$h'_{sigmoid}(x) = (1 - h_{sigmoid}(x))h_{sigmoid}(x) \tag{4}$$

and is almost 0 in the vicinity of those values. In other words, features tend to be flat at these values, and as the layer gets deeper, features tend to be lost, and learning may not progress. Therefore, along with the problem of computational complexity, it is often used in ordinary neural networks with few layers, but it is not often used in DNN.

In light of the shortcomings of sigmoid functions, DNN often uses the ReLU (Rectified Linear Unit) function shown in Fig. 7. The ReLU function is a function that outputs 0 when the input is 0 or less and outputs the input value as it is when 0 or more. The process is simpler than the sigmoid function, and the equation is as follows.

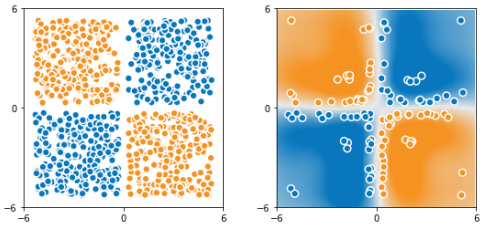
$$h_{ReLU}(x) = max(0, x) \tag{5}$$

Because the ReLU function is linear, differentiation does not result in a zero gradient. As a result, there is no problem that



(a) training data (b) Classification result by difference of discrimination hyperplane.

Fig. 8. Example of different classification for the same data.



(a) training data (b) support vector

Fig. 9. Data reduction by support vector for XOR.

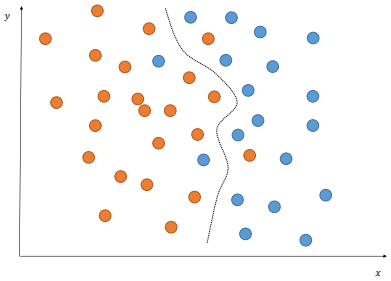


Fig. 10. Example of misclassification.

learning will not progress due to the loss of features, which has been an issue with sigmoid functions. The calculation is also straightforward, and DNN often uses the ReLU function.

D. Support Vector Machine

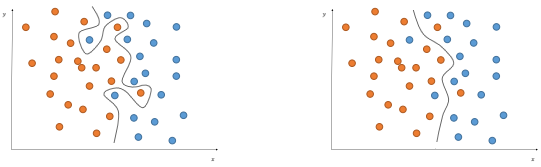
Support Vector Machine (SVM) [3] is one of pattern recognition models using supervised learning and is a method that can be used for classification and regression problems. In describing the concept of SVM, consider the problem of classifying the distribution map, as shown in Fig. 8(a). This figure is a graph representing the distribution with x as the horizontal axis and y as the vertical axis. We consider classifying orange and blue data into one hyperplane. There are many division methods, but arbitrary division methods are not possible. As an example, Fig. 8(b) shows two ways of drawing below. The two divisions are shown in Fig. 8(b) can not be regarded as incorrect because both can correctly classify orange and blue. Assuming that both lines are suitable for classification, we consider the input data shown in black. Because these input data belong to the orange side in one line and to the blue side in the other line, these lines are not suitable for classification. In this way, the data near the boundaries to be classified are vague data. SVM is a method to classify this ambiguous data as correctly as possible. SVM components have support vectors and margins. The support vector is the data near the classified boundary, and the margin is the distance between the classified boundary and the data. The above vague data is data that is close to the boundary; in other words, data with a small margin. If there are many data with a small margin, the classification accuracy will be reduced, and such data can be reduced by making the margin as large as possible. This is called margin maximization. In order to prevent incorrect classification, it is only necessary to classify the data near the boundary correctly. Therefore, classification is performed by focusing on the support vectors. This concept is used in the proposed method described later. Fig. 9 shows how support vectors reduce data.

Up to this point, data that can be classified finely for simplicity is taken as an example, but let us consider the case where somewhat specific data, as shown in Fig. 10, is given. If this data is classified in the same line as before, an incorrect identification will occur. Forcing correct classification of incorrectly identified data will lead to loss of these valuable data. It is called overfitting [5] that it overfits the original data and the prediction accuracy for the newly given data falls. It is said that there is generalization [6] that can be flexibly predicted for given data without causing overfitting, and tolerating an incorrect identification to some extent leads to an increase in generalization. In generalization, a technique that does not allow false identification is called a hard margin, and a technique that allows false identification to some extent is called a soft margin [7]. The soft margin allows some misidentification, but it balances the margin maximization and the allowance of the misidentification by giving a penalty. This penalty is also called cost. By adjusting the value of the cost parameter C, it is possible to decide how much misidentification is permitted. The following minimization problem replaces margin maximization.

$$\min\{\frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i\} \tag{6}$$

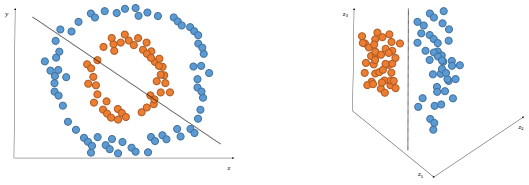
ξ is a slack variable, $\xi = 0$ if the data is correctly identified, $0 < \xi \leq 1$ if correctly identified but within the margin, and $\xi > 1$ if false. That is since the second term becomes larger as there are more misidentifications, if (6) is to be minimized, an adjustment is made to reduce the value of ξ. It has been stated that it is possible to decide how much to allow erroneous identification by adjusting the value of the cost parameter C. The reason for this is the larger the value of C, the larger the value of the second term. Also, the case of $C \to \infty$ is called the hard margin. In the case of soft margin, adjusting the value of C can change how much incorrect identification is allowed.

In SVM, two kernel methods, RBF (Radial Basis Function) kernel and polynomial kernel, are widely used for classification. The kernel method is a data analysis method developed from the 1990s when the proposal of SVM started, and it is



(a) Hard margin (b) Soft margin

Fig. 11. Hard and soft margin.



(a) Example of data that can not be classified linearly (b) Convert to data that can be classified linearly

Fig. 12. Transformation of nonlinear data by kernel method.

a convenient method when dealing with nonlinear data. As an example, circularly distributed nonlinear data, as shown in Fig. 12(a) cannot be classified linearly.

$$(z_1, z_2, z_3) = (x^2, y^2, \sqrt{2}xy) \tag{7}$$

By using the kernel method for coordinate conversion, data that cannot be classified linearly can be converted to data that can be classified linearly. The following equation represents the RBF kernel of the two kernels used in my evaluation.

$$K_p(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{8}$$

The RBF kernel is a frequently used kernel function and often uses C and γ as hyperparameters of SVM. Next, the equation for the polynomial kernel is as follows.

$$K_p(x_i, x_j) = x_i x_j + r^d \tag{9}$$

This is a kernel represented by a polynomial of order d, with r added as a hyperparameter. Sections 3 and 4 show how effective these two kernel functions are for reducing training data.

E. Training data reduction method

We consider two existing studies on training data reduction. The first is a study by Nguyen et al. using support vectors to examine how training data reduction and classification accuracy change with two-dimensional data [8]. This study describes the trade-off between classification accuracy when data reduction is performed using support vectors, and the problem of classifying three different data distributions: Gaussian, sine, and an ellipse, which are two-dimensional data. This derives an upper limit on how much data reduction is possible by the support vector, and it has been shown that training data can be reduced by an appropriate amount within a predetermined tolerance range. It is found that the reduction of training data by support vectors is effective in classification problems of 2D data.

Fig. 13. MNIST dataset.

Next, as the second existing study, we describe training data reduction using support vectors for neural networks conducted by Dahiya et al. [9]. In this study, support vector is used to reduce training data while maintaining accuracy for the problem of classifying image dataset of handwritten numbers from 0 to 9 called MNIST shown in Fig. 13 [10]. Table 1 shows the experimental results. In this research, it was reported that accuracy degradation did not occur even if the training data of 60,000 were reduced to 20,000 using SVM of the RVM kernel. Also, when using SVM of the polynomial kernel, this research succeeded in reducing training data to 10,000, and there was almost no deterioration in accuracy. As a control experiment, this research also experimented when extracting the same number randomly instead of the support vector, and compared with the case of extracting the support vector, and the accuracy is reduced by 2 to 3%. From this, it is clear that although accuracy decreases if the number of training data is reduced at random, it is possible to classify without degrading the accuracy by extracting data using support vectors.

III. Training Data Reduction in Deep Learning

In this section, we describe the reduction of training data for deep learning using support vectors. As mentioned in Section 2, since deep learning requires a large amount of training data and learning time, we aim to reduce training data by extracting training data efficiently using support vectors and excluding relatively less critical data.

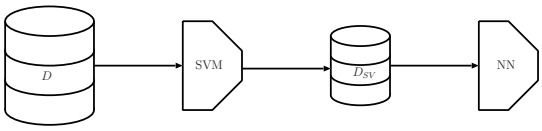
A. Training data reduction using support vectors

This method reduces training data using support vectors to reduce the learning time of the neural network. Fig. 14 shows the proposed method and the learning of a conventional neural network. The procedure of the training data reduction method using support vectors is as follows.

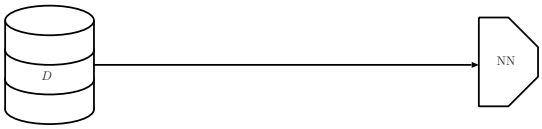
1) Train dataset D with SVM and extract support vector D_{SV}
2) Train neural networks using D_{SV} as training data

TABLE I
Results of experiments conducted by Dahiya et al.

training data	method	SVM kernel	accuracy(%)
60,000	Original	-	97.62
20,000	Suport vector	RBF	97.66
10,000	Suport vector	Polynomial	97.48
20,000	Random	-	95.32
10,000	Random	-	94.67

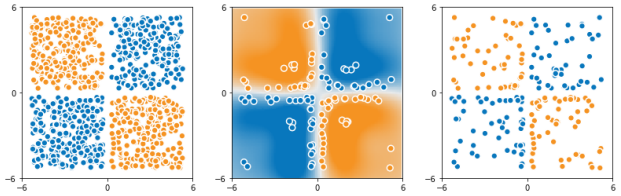


(a) Reduction of training data of neural network using support vector



(b) Normal neural network learning

Fig. 14. Reduction of training data by support vector.



(a) original data (b) support vector (c) support vector + random

Fig. 15. Data reduction by support vector and increase of support vector by random sampling data.

An example of applying data reduction by support vectors to two-dimensional data is shown in Fig. 15. Focus on the following points when using support vectors.

- How much training data could be reduced
- How much learning time decreased
- What is the impact on classification accuracy
- What happens if the same number of data is randomly extracted for the above three items

When using support vectors, if the training data is only slightly reduced, it will be meaningless. It is even more meaningless if degradation in classification accuracy occurs that does not meet the reduction in training data. Also, even if training data can be significantly reduced, classification accuracy must be maintained at a certain level. we will consider the reduction of training data while focusing on these points. However, even if we clear these conditions and confirme that only the support vectors are extracted to reduce the training data and the accuracy hardly falls, this is the undesirable result because there is no meaning using support vectors if the result is almost the same as when data reduction is performed by extracting data at random. It should be kept in mind that this may well occur if the problem to be dealt with is simple, or the training data given originally is excessive. Also, if the classification accuracy is not maintained and degraded when the support vector is used, we consider reinforcing the training data by supplementing the data not extracted as the support vector, as shown in Fig. 15(c). Since part of the data that forms the original space is supplemented, accuracy can be expected to be improved compared to when the training data is only the support vector.

B. Preliminary experiment

In order to confirm the effectiveness of the training data reduction method using support vectors, we evaluate using a program with the same function as TensorFlow Playground[1] as a preliminary experiment. In the program used, two-dimensional data can be learned with a neural network, and the activation function, the number of intermediate layers, and the number of neurons can be changed arbitrarily. The two-dimensional data used in the evaluation are Gaussian (R_GAUSS) and straight (R_PLANE) as two types of regression problems, and Gaussian (C_GAUSS), spiral (C_SPIRAL), circle (C_CIRCLE), and XOR (C_XOR) as four types of classification problems. We evaluate the amount of data reduction and accuracy and analyze the classification results visually. In the evaluation, we used ReLU as the activation function. The middle layer is three layers, and the number of neurons in each layer is five. When we tried to reduce the data using support vectors for six types of two-dimensional data, almost no deterioration in accuracy occurred in all six. Detailed results are described in Section 4.

In the preliminary experiment, we evaluate the classification accuracy, regression accuracy, and learning time using training data extracted by the following three methods.

1) Learning using all training data
2) Learning using only support vectors
3) Learning extracting randomly as many data as support vectors

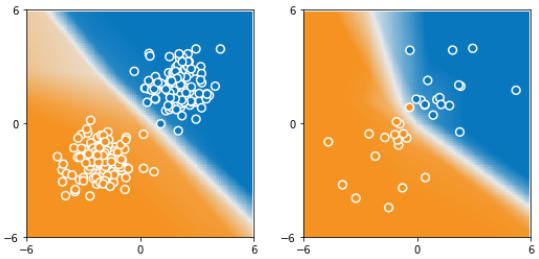
The number of epochs is set to 100. However, because C_SPIRAL is more complicated than the other functions and it takes time, it is set to 500 epochs. Classification accuracy is evaluated by loss, and the smaller the loss, the higher the accuracy.

In C_GAUSS, the accuracy was not degraded even if learning was performed using only the support vector. However, because the classification is relatively simple, even if it is randomly extracted, the accuracy is hardly degraded, and it can not be determined whether the support vector is capable. As shown in Fig. 16, although it is not classified as a clean straight line like the original data, it can be said that there is no problem because the classification itself is properly performed.

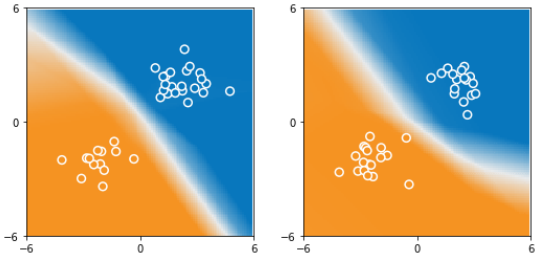
TABLE II
TYPE SIZE FOR PAPERS

dataset	(A) All training data		(B) Support vector		(C) Random	
	data	loss	data	loss	data	loss
C_GAUSS	1,000	0.001	126	0.001	126	0.001
R_PLANE	2,400	0.004	292	0.001	292	0.006
R_GAUSS	2,400	0.192	263	0.011	263	0.563
C_SPIRAL	1,000	0.009	137	0.039	137	0.105
C_CIRCLE	1,000	0.001	129	0.002	129	0.177
C_XOR	1,000	0.003	117	0.001	117	0.005

[1]https://playground.tensorflow.org/

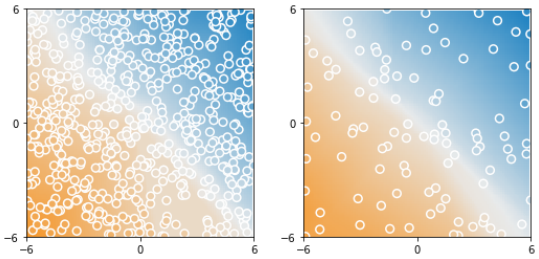


(a) (A) original data (b) (B) support vector

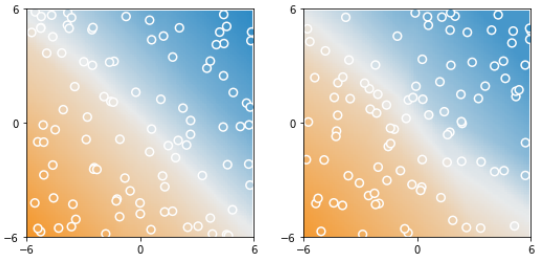


(c) (C) random 1 (d) (C) random 2

Fig. 16. Evaluation in the dataset C_GAUSS.

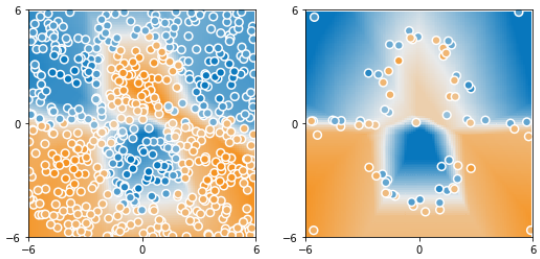


(a) (A) original data (b) (B) support vector

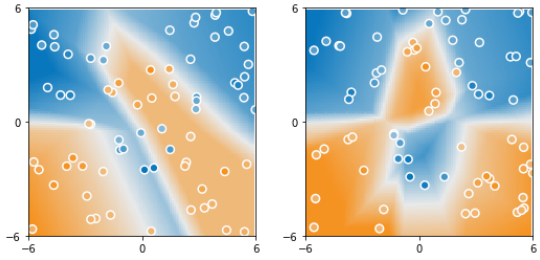


(c) (C) random 1 (d) (C) random 2

Fig. 17. Evaluation in the dataset R_PLANE.

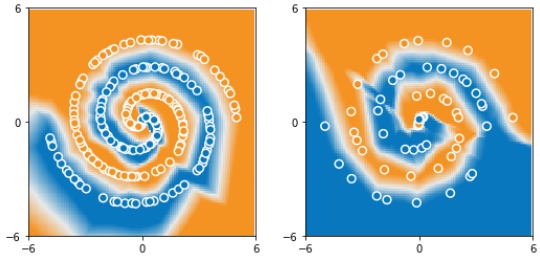


(a) (A) original data (b) (B) support vector

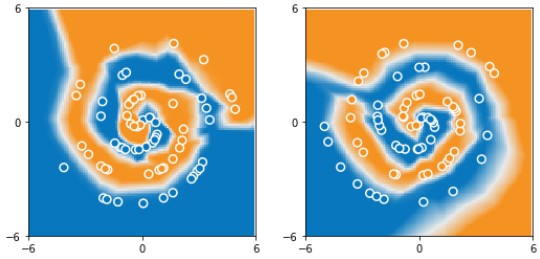


(c) (C) random 1 (d) (C) random 2

Fig. 18. Evaluation in the dataset R_GAUSS.



(a) (A) original data (b) (B) support vector



(c) (C) random 1 (d) (C) random 2

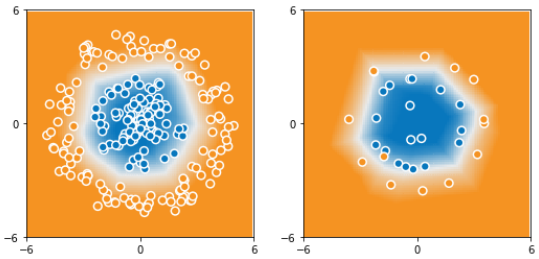
Fig. 19. Evaluation in the dataset C_SPIRAL.

As with C_GAUSS, R_PLANE did not degrade in accuracy. Similarly, the effectiveness of the support vector is unknown because accuracy degradation was hardly confirmed even if it was randomly extracted. As can be seen from Fig. 17, it can be said that there is no problem as well as C_GAUSS because all are classified in the same way.

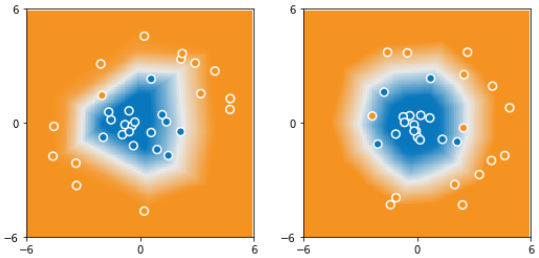
R_GAUSS did not lose accuracy even if learning was performed using support vectors only. Besides, the accuracy was degraded in random extraction, and some did not learn well. In R_GAUSS, a relatively complex data set, the effectiveness of the support vector was shown. Looking at Fig. 18, the support vector can be classified in the same way as the original data, so there is no problem. Also, in some cases, random classification may cause inappropriate classification.

In C_SPIRAL, although a slight deterioration in accuracy was observed, it was not fatal and was within an acceptable range. Moreover, even if it was extracted at random, there was no significant degradation in accuracy, and it is not clear that the support vector was capable. It can be said that there is no problem because all are classified in the same way, as shown in Fig.19.

The accuracy did not deteriorate even if C_CIRCLE was

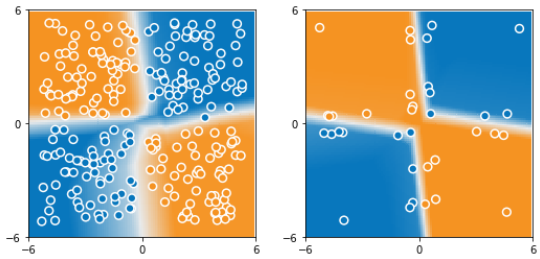


(a) (A) original data (b) (B) support vector

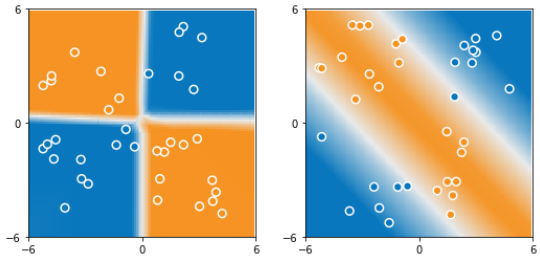


(c) (C) random 1 (d) (C) random 2

Fig. 20. Evaluation in the dataset C_CIRCLE.



(a) (A) original data (b) (B) support vector



(c) (C) random 1 (d) (C) random 2

Fig. 21. Evaluation in the dataset C_XOR.

trained only with support vectors. However, this is also relatively easy to classify, as in C_GAUSS so even if it is randomly extracted, its accuracy is hardly degraded, and it can not be concluded that the support vector is capable. Looking at Fig. 20, although random classification sometimes resulted in somewhat angular classification, all classifications were performed without problems within the allowable range.

The accuracy did not deteriorate in the C_XOR. Besides, even if it was randomly extracted, degradation of accuracy could hardly be confirmed, but there was a case that the classification failed considered once. Therefore, it is considered that the support vector was valid to some extent.

IV. Evaluation

In this section, the training data reduction method for deep learning with support vectors proposed in Section 3 is applied to ResNet 18 [4], which is one of the latest neural networks, and evaluation is performed from the viewpoint of training data reduction and classification accuracy. ResNet is one of the networks achieving high accuracy recognition rate in image recognition. We use the CIFAR-10 [11] data set for evaluation.

CIFAR-10 is an image data set with a total of 60,000 sheets of 50,000 training data and 10,000 test data. As shown in Fig. 22, each image is a color image of 3 channels of RGB with 32×32 pixels. Each image has a label representing a class, and there are 10 types of class labels: airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck. In this case, two patterns, one for classifying 5 labels, and the other for classifying all 10 labels, are performed. The five labels used for classification are airplanes, birds, deer, frogs and, ships. In all cases, the number of learnings is 100 epochs.

In multiclass classification, since different support vectors

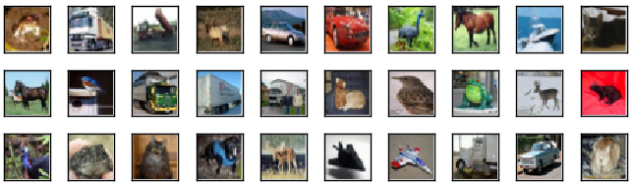


Fig. 22. A part of CIFAR-10 dataset.

are selected for one-to-one classifier and one-to-many classifier, we evaluate both multiclass classification identifications [12]. In the one-to-one classifier, ${}_N C_2$ classifiers are used to classify N classes from class C_1 to class C_N. One-to-many classifiers use N classifiers that solve the two-class classification problem: whether the classification of N classes is classified into a specific class or any other class.

In the evaluation, we confirme the relationship between classification accuracy and learning time for three cases (A) when using all training data, (B-1) when using only support vectors in a one-to-one classifier and (B-2) when using only support vectors in a one-to-many classifier. Table 3 shows the experimental results for five classes. Table 3 shows that the proposed method reduces the calculation time while achieving the same classification accuracy as the previous method using all the training data. For example, focusing on the calculation accuracy, the calculation accuracy of (A) is 78.300%, while the calculation accuracy of (B-1) is 78.580% and (B-2) is 78.340%, respectively. As for the calculation time, (A) requires 1058.243 seconds for learning, while (B-1) require 961.395 seconds and (B-2) requires 957.629 seconds. In other words, using only the support vector reduced the calculation time by 9.16% $(1 - 961.365 \div 1058.243)$ in (B-1) and by 9.50% $(1 - 957.629 \div 1058.243)$ in (B-2). Table 4 shows the

experimental results for 10 classes. Similar to the evaluation for five classes, the proposed method reduced the calculation time while maintaining the same calculation accuracy as the conventional method. According to Table 4, the calculation time was reduced by 2.54% $(1-2062.906 \div 2116.688)$ in (B-1) and by 2.66% $(1-2060.405 \div 2116.688)$ in (B-2). From the above, we confirmed experimentally that the proposed method can reduce the amount of calculation without degradation in accuracy in both 5 label and 10 label evaluations.

It can be read that the reduction effect of calculation time at 10 labels is lower than the effect at 5 labels comparing Table 3 and Table 4. The proposed method reduces the number of data required for training by using only supporter vectors, which is considered to contribute to the reduction of execution results. Focusing on the number of training data required by the conventional method, (B-1) and (B-2), according to Table 3, the number of data is reduced by 11.86% $(1 - 22,035 \div 25,000)$ in (B-1) and 12% $(1 - 21,975 \div 25,000)$ in (B-2) at 5 labels, respectively. Similarly, according to Table 4, the reduction rate of the number of training data at 10 labels is 4.26% $(1-47,870 \div 50,000)$ and 4.40% $(1-47,799 \div 50,000)$. Similar to the reduction rate of calculation time, it can be read that the reduction rate of the number of training data at 5 labels is higher than at 10 labels.

Based on the above experimental results, we consider training data reduction when using support vectors for CIFAR-10. The preliminary experiments in Section 3.2 reduced the original data by 86 to 88% when determining the support vector. However, when the support vector was extracted from the training data of CIFAR-10, it was only reduced by about 12% of the original data in the case of 5 labels. It accounted for a large proportion of the original data and did not provide the expected results from the preliminary experiments in terms of reduction of training data. The cause of this problem is that the problem of identifying color images is much more complicated than that of the preliminary experiments. In the case of a relatively simple data set as treated in the preliminary experiment, no deterioration in accuracy was observed even if the number of data extracted as support vectors was 12 to 14%. However, in the case of relatively complex data sets such as CIFAR-10, as there are many data responsible for the formation of the classification plane, the number of training data extracted as support vectors is about 88%, which is the majority of the whole. Compared with the results of the preliminary experiments, although it did not reduce the training data as expected, the training data could be reliably reduced. Moreover, since the accuracy has hardly been deteriorated, it is possible to improve the trade-off between the number of training data and the classification accuracy and to reduce the training data of CIFAR-10 without causing the accuracy deterioration by using the support vector. Compared with the results of the random sampling data performed as a control experiment, the accuracy is also improved, although it is about 1%. Also, the time taken for learning decreased with the reduction of training data, and it can be said that this was also effective. There were more data extracted as support vectors for 10 labels than for 5 labels, and about 95% was extracted. Although only 5%, this was also able to reduce training data. As in the case of 5 labels, almost no degradation in accuracy occurs, so it can be said that the effect of the support vector has been confirmed.

V. Conclusions

In this study, we tried to reduce training data in deep learning by using support vectors. Although several existing studies used data reduction by support vectors, the difference from them is that support vectors are applied to CIFAR-10. First, we conducted preliminary experiments to reduce training data using support vectors for six types of two-dimensional data, and it was confirmed that training data could be reduced by 86 to 88% while maintaining the accuracy. Based on this, when trying to reduce training data using support vectors for the problem of classifying CIFAR-10 color images as well, The training data could be reduced by about 12% in the 5-class classification and about 5% in the 10-class classification with almost no deterioration in accuracy. Although the effect was not obtained like the result of the experiment, it was confirmed that the training data could be reduced by using the support vector for CIFAR-10 as well.

Acknowledgment

References

[1] J. Deng, W. Dong, R. Socher, L. J. Li, , K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 770–778.

[5] S. Lawrence, C. L. Giles, and A. C. Tsoi, "Lessons in neural network training: Overfitting may be harder than expected," in *Proceedings of National Conference on Artificial Intelligence*, Jul. 1997, pp. 540–545.

TABLE III
Result of applying support vector for CIFAR-10 (5 labels)

training data	data	accuracy(%)	time(sec)
(A) all data	25,000	78.300	1058.243
(B-1) support vector(one-to-one)	22,035	78.580	961.394
(B-2) support vector(one-to-many)	21,975	78.340	957.629
(C-1) random	22,035	77.400	962.348
(C-2) random	21,975	77.260	959.526

TABLE IV
Result of applying support vector for CIFAR-10 (5 labels)

training data	data	accuracy(%)	time(sec)
(A) all data	50,000	77.560	2116.688
(B-1) support vector(one-to-one)	47,870	77.420	2062.906
(B-2) support vector(one-to-many)	47,799	76.970	2060.405
(C-1) random	47,870	76.580	2052.179
(C-2) random	47,799	76.290	2057.681

[6] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," in *Mathematics of Deep Learning, Cambridge University Press, to appear*, 2018.
[7] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, Mar. 2001.
[8] X. Nguyen, L. Huang, and A. D. Joseph, "Support vector machines, data reduction, and approximate kernel matrices," in *Proceedings of Machine Learning and Knowledge Discovery in Databases*, Sep. 2008, pp. 137–153.
[9] K. Dahiya and A. Sharma, "Reducing neural network training data using support vectors," in *Proceedings of Recent Advances in Engineering and Computational Sciences*, Mar. 2014, pp. 1–4.
[10] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, Apr. 2012.
[11] A. Krizhevsky, "Convolutional deep belief networks on CIFAR-10," 2010, unpublished.
[12] J. Weston and C. Watkins, "Multi-class support vector machines," Royal Holloway University of London, Tech. Rep., May 1998.