Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6170662, 10 pages
https://doi.org/10.1155/2021/6170662

Research Article

VOVU: A Method for Predicting Generalization in Deep Neural Networks

Juan Wang,[1] **Liangzhu Ge**,[2] **Guorui Liu**,[1] **and Guoyan Li**[3]

[1]*Computing Center, Tianjin Chengjian University, Tianjin, China*
[2]*Alibaba Group, Hangzhou, China*
[3]*School of Computer and Information Computer, Tianjin Chengjian University, Tianjin, China*

Correspondence should be addressed to Guorui Liu; lgr@tcu.edu.cn

Received 6 July 2021; Revised 4 October 2021; Accepted 27 October 2021; Published 23 November 2021

Academic Editor: Nadeem Qazi

During the development of deep neural networks (DNNs), it is difficult to trade off the performance of fitting ability and generalization ability in training set and unknown data (such as test set). The current solution is to reduce the complexity of the objective function, using regularization methods. In this paper, we propose a method called VOVU (Variance Of Variance of Units in the last hidden layer) to maximize the optimization of the balance between fitting power and generalization during monitoring the training process. The main idea is to give full play to the predictability of the variance of the hidden layer units in the complexity of the neural network model and use it as a generalization evaluation index. In particular, we take full advantage of the last layer of hidden layers since it has the greatest impact. The algorithm was tested on Fashion-MNIST and CIFAR-10. The experimental results demonstrate that VOVU and test loss are highly positively correlated. This implies that a smaller VOVU indicates that the network has better generalization. VOVU can serve as an alternative method for early stopping and a good predictor of the generalization performance in DNNs. Specially, when the sample size is limited, VOVU will be a better choice because it does not require dividing training data as validation set.

1. Introduction

In recent years, deep neural networks (DNNs) have been major methods for natural language processing [1], computer vision [2], EEG signal analysis [3], etc. More surprisingly, even when the number of parameters is significantly larger than the amount of training data, DNNs can also be generalized well [4]. However, the generalization of well-trained deep neural network models may be good or may not be satisfactory. These observations lead to a series of studies including balancing the bias-variance trade-off when training networks and developing the evaluation index of the generalization ability of DNNs.

There are many factors that affect generalization of deep neural networks where the important one is the relationship between fitting power and generalization. The most important phenomenon of overfitting is the inconsistent performance of the model on the training set and the test set. To overcome this discrepancy, a lot of regularization, optimization methods, and network structures have been proposed, including early stopping [5], dropout [6], batch normalization [7], and residual networks [2]. These methods enhance the network's generalization capacity in different ways. Among these methods, validation-based early stopping [5] is a widely used method in practice, which estimates this discrepancy through a validation set. Another commonly used method when training neural networks is dropout [6] which attempts to prevent coadaptation of neuron activations. There are also many researches trying to open the "black-box" via information theory [8–11]. For example, Shwartz-Ziv [9] investigated the Information Plane, i.e., the plane of the mutual information values that each layer's internal representation preserves on the input and output variables, for a better understanding of the training dynamics. Keskar et al. show that flat minima are better than sharp minima [12]. Novak

et al. [13] found that the norm of the input-output Jacobian of the network correlates well with generalization. Additionally, Bartlett and Mendelson [14] derived generalization bounds in terms of robustness to noise.

Meanwhile it is still challenging to predict generalization of the well-trained deep neural networks because it cannot be explained by traditional statistical theory [4]. For example, VC dimension and Rademacher complexity only give the upper bound of the generalization error in terms of the capacity model to overfit data [14] and the bound is quite loose. What is more, it can be quite difficult to determine the capacity of deep learning algorithms [15]. Some direct analyses have also been done with respect to a specific setting of model parameters to deal with this problem, such as [15, 16].

Units'role in neural networks in supervised learning has been studied in numerous literatures [17–20]. To understand the internal workings of networks, Morcos et al. [19] evaluated the selectivity of neurons, a quantity that measures how strongly a neuron's output to data samples from one class differs from the behavior for data samples from all other classes. In [21], they proposed a new regularizer called DeCov which leads to significantly reduced overfitting and better generalization. In [22], the Neuron Importance Score Propagation (NISP) algorithm was proposed to propagate the importance scores of final responses to every neuron in the network.

In this paper, we propose a method to optimize the balance between fitting power and generalization to the greatest and propose the evaluation index of prediction generalization through exploring the influence of hidden units in the neural networks.

Our method is from the perspective of representation learning [23]. We investigate the question of what kind of intermediate representation attribute is needed to evaluate generalization. This work mainly focuses on the properties of representations learned in image classification applications with neural networks to study neural units' importance discrepancy through the matrix decomposition based method. We introduce a statistic metric characterized by Variance Of Variance of Units in the last hidden layer (VOVU) to describe the discrepancy of units' importance on training samples. Through experiments on different datasets, we infer the dynamic correlation between VOVU and generalization. Additionally, our algorithm does not require validation set to detect when memorization has begun in neural networks. Validation set is only used for comparing with other methods. Besides, we explain why dropout performs well in various deep neural networks through the perspective of our method.

The structure of this paper is organized as follows. In Section 1, we introduce the research background of deep learning network in evaluating generalization ability. In Section 2 we describe a general model decomposition framework in deep neural networks and introduce our proposed statistic metric VOVU, which is applied to the hidden representations to predict the change of test loss during the neural network's training process. In Section 3, we demonstrate and analyze our experiments on Fashion-MNIST and CIFAR-10 and compare VOVU with other early stopping methods. We discuss our work in Section 4 and make the conclusion in Section 5.

2. Materials and Methods

In this section, we first describe a general model decomposition framework in neural networks, which divides the deep neural networks into two parts. Second, we introduce our proposed statistic metric VOVU, which is then applied to the hidden representations to predict the change of test loss during the neural network's training process.

2.1. Model Decomposition. On one hand, deep learning models are often composed of multiple layers. A lot of efforts have been made to identify the role played by each layer, but it can be hard to find a meaning for individual layers [24]. In this work, we simplify this problem through model decomposition. On the other hand, it is well known that the hierarchical hidden layers before output layer perform the feature extraction roles [25], while the high layers tend to learn abstract representations for the final tasks. Particularly, the last hidden layer is assumed to be the most important one as it provides features for discriminating the training samples directly. Thus, it is reasonable for us to focus on the transformed space of the last hidden layer in neural networks. Note that we also conduct experiments on other layers to confirm the effectiveness of model decomposition, the results of which further support our claim.

Let us consider a parametric mapping function for classification: $M(.;\theta)$: $X \longrightarrow Y$, represented here by a neural network model, where X is the input space and Y is the label space. This neural network can be arbitrarily decomposed into two parametric subfunctions: (1) $G(\cdot,\theta_G)$: $X \longrightarrow Z$, a representation function parameterized with the set θ_G. This subfunction projects an input sample x into a representation space Z. (2) $F(\cdot;\theta_F)$: $Z \longrightarrow Y$, a decision function parameterized with the set θ_F. It performs the classification decision over the representation space Z.

In this case, the network decision function can be written as follows:

$$M(x_i;\theta) = F(G(x_i;\theta_G);\theta_F), \tag{1}$$

where $\theta = \{\theta_G, \theta_F\}$.

In this way, we can divide any neural networks into two parts: the hidden layers before output layer performing the transformation function while the output layer uses the transformed feature space (the outputs of the last hidden layer) to make the final decision. Such a decomposition of a neural network with 4 layers is presented in Figure 1. For the discussed reasons, the decision function $F(\cdot)$ is composed of solely the output layer while the rest of the hidden layers form the representation function $G(\cdot)$.

2.2. Motivation for VOVU. Our motivation comes from the researches with regard to units' importance. The problem of attributing a deep network's prediction to its hidden units is not well-studied [17, 18, 26]. Roles of units within a layer for classification can be split into redundant, unique, and

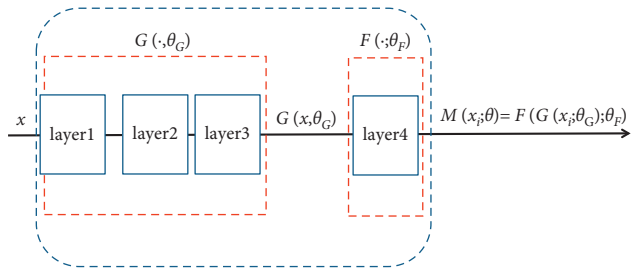

Figure 1: Decomposition of a 3-hidden layer neural network $M(\cdot)$ into a representation function $G(\cdot)$ and a decision function $F(\cdot)$. This is a general training framework in this paper.

synergistic parts [18]. It means that units' behaviors within the same layer can be complex and different.

It is a common rule to think that a larger variance corresponds to a more important feature for tasks in traditional feature selection studies [27], which has also recently been used as an assumption to study the deep learning dynamics [28]. In this paper, we believe that a unit's importance is defined by variance of its outputs with respect to a set of neural network inputs. We use the variance of units' importance to describe the discrepancy between them and study whether this discrepancy affects generalization of the network.

In probability theory and statistics, a random vector is a random variable with multiple dimensions. Each element of the vector is a scalar random variable. Each element has a either a finite number of observed empirical values or a finite or infinite number of potential values. In deep neural networks, we can assume that a neuron in the last hidden layer computes a real-valued function over the network's input domain [28, 29]; that is, we regard the outputs of the last hidden layer as a random vector.

We will use a matrix decomposition based method to examine the relationship between the generalization performance of a network and the behavior of the last hidden units. For example, for a fully connected network, given m data points and the last hidden layer consisting of n units, we regard the activation outputs in the last hidden layer over the m data points as realizations of a random vector and these realizations can be put into a matrix, namely, activation matrix.

We consider the variance of the sample on the neural unit as a unit's importance which can be calculated by the activation matrix. Then we compute the variance of the vector consisting of the unit's importance. The algorithm of calculating VOVU in neural networks is listed in Algorithm 1.

Based on above analysis and common practice in literatures, we conjecture that feature representation learned by the networks in the transformed space $G(\cdot)$ should not be redundant and inefficient for the classification tasks. VOVU characterize the discrepancy of principal components of the original activation matrix. Therefore we expect to see that the smaller the VOVU is, the better the performance of generalization will be. A more detailed explanation is left in Section 3.4.

2.3. VOVU for Fully Connected Networks. This subsection aims at introducing VOVU in fully connected networks. As discussed by the above sections, by considering neuron's outputs in the last hidden layer with respect to neural network inputs as realization of random variables, one can define and calculate statistic quantities to study the hidden units' behavior. The proposed metric, VOVU, amounts to the empirical variance of the unit's importance associated with the last hidden layer's output, considered as a vector of the dimension equaling the layer's number of units.

More concretely, we give a subset of m data points sampled from training set $T = \{x_1, \ldots, x_m\}$ and a neural network with the last hidden layer consisting of n units. VOVU is computed by calculating the unit's importance over the m data points. Note that the m data points were randomly drawn from the training set and the sampling set can be either the whole training set or a subset of the training set. Recall that we do not require validation sets to provide the sampling data. As for the number of sampling sets, we make a further discussion in later section.

A detailed description is as follows. Let l denote the last hidden layer which has n units after the model decomposition by $G(\cdot)$. For each $i \in (1, 2, \ldots, m)$, we get the activation outputs on $x_i \in T$, i.e.,

$$A_{i\cdot} = [A_{i1}, \ldots, A_{in}]. \tag{2}$$

Note that we suppose the outputs of the j^{th} unit in l are a random variable; then we get A_{ij} to describe the i^{th} independently drawn observation on the j^{th} random variable.

When collecting m observations, the output of this layer on the m inputs can be regarded as a set of neuron vectors, which can be arranged as columns of the activation matrix $\mathbf{A}$, i.e.,

$$A = [A_1, \ldots, A_m]. \tag{3}$$

For each $i \in (1, 2, \ldots, m)$, we calculate the unit's importance I_i by the variance of $\mathbf{A}_i$, so that

$$I_i = \frac{1}{m} \sum_{i=1}^{m} \left(A_i - \widetilde{A}\right)\left(A_i - \widetilde{A}\right)^T = \mathrm{Var}\left(A_i\right), \tag{4}$$

where $\widetilde{A}$ is the sample mean vector of $\mathbf{A}$. Then we get the vector of unit's importance:

$$I = [I_1, \ldots, I_m]. \tag{5}$$

Finally, we calculate the variance of I that can be written as $\mathrm{Var}(I)$ and get our proposed statistic metric VOVU, which is denoted as

$$\mathrm{VOVU} = \mathrm{Var}(I). \tag{6}$$

In the following sections, as the test error is commonly used as a proxy for the generalization error in determining when overfitting has begun, we will investigate the connection between this statistic metric VOVU and the test loss on classification task in full connected networks.

Input:
(1) The activation matrix of neural network's last hidden layer, $A \in R^{n \times m}$
(2) The fixed subset of the training set, T
Output:
Variance Of Variance of Units in the last hidden layer, $\mathrm{Var}(I)$.
(1) Calculating the activation matrix vector A.
(2) Calculating the vector of unit's importance I , for each $I_i = \mathrm{Var}(A_i), i \in (1, m)$
(3) Calculating the empirical variance of I
(4) Return VOVU = $\mathrm{Var}(I)$

Algorithm 1: Algorithm of calculating VOVU in neural networks.

2.4. VOVU for Convolutional Neural Networks. As for the convolutional networks, similar approach can also be used. However, since there are various types of structures in convolutional networks, the problems need to be divided into two different cases.

In the first case, if the last hidden layers in convolutional networks are fully connected layers, we compute the proposed measure VOVU same as in the fully connected neural networks, which has been already discussed in the above sections.

While in the other case, when the last hidden layer is a convolutional layer or other types of layer, this case is different from fully connected layers. If we regard each feature in the feature map of a convolutional layer as a unique neuron, it will cost a very large amount of computation resource when calculating VOVU.

In this work, we ignore the latter case because the last hidden layer in current image classification neural networks is mainly fully connected layers. Then our method can also be feasible in these networks. How to degrade the convolutional layer to a fully connect layer or how to find the really useful features in other types of layers is left for future work.

3. Results

Our method can be a general method for predicting test loss during the training process in various neural networks. In this paper, we test our method on two kinds of networks: one is fully connected networks trained on commonly used benchmark datasets Fashion-MNIST, and the other one is convolutional networks trained on CIFAR-10 datasets. In all experiments, ReLU nonlinearity is applied to all layers but the output and pooling layer. We use cross-entropy loss as the evaluation metric for classification performance. The loss value implies how well or poorly a certain model behaves during each iteration of optimization.

3.1. Experiment on Fully Connected Neural Network

3.1.1. Dataset. We conduct experiments on Fashion-MNIST datasets, both of which have a training set of 55000 samples and a test set of 10000 samples. 5000 samples are left for validation. Note that we do not need the validation sets to detect memorization in neural networks.

3.1.2. Experimental Settings. To evaluate the effectiveness of VOVU used as a tool for predicting the generalization performance, we choose to set a relatively small network structure, a fully connected neural network with three hidden layers. The size of the hidden units, the learning rate, and standard variance of the initialized weights are all varied to test the generality of the proposed method. The batch size is set to 128. The maximum epoch for each net is 100.

We use the SGD with momentum algorithm to train the network. Note that it is only chosen for its better training process. Other optimization methods are also suitable for our experiments. Experiments are implemented by TensorFlow.

We conduct experiments to show the relationship between VOVU and test loss during the training. We train 50 fully connected networks models with different hyper parameter settings, including the size of hidden units, learning rate, and different initialization of weights. We calculate the VOVU of the last hidden layer every two epochs during the training process and plot the dynamic change of test loss and VOVU in the same plots.

3.1.3. Experimental Results. Interestingly, we can see from Figures 2 and 3 that the point when VOVU begins to rise is nearly the same point that the test loss starts to rise. Furthermore, we find that test loss and VOVU are highly positively correlated in fully connected networks trained on Fashion-MNIST datasets.

Besides, we observe that VOVU has potential to be an excellent method for early stopping and predicting generation. For any two well-trained networks (for example, any two points with small training cost) with a large probability, we can conclude that the larger the VOVU of the network, the worse the generalization ability of the network. It means VOVU may serve as a good proxy for the selection of hyperparameter and early stopping. It can also be used to predict and evaluate network generalization.

3.2. Experiment on Convolutional Neural Network

3.2.1. Dataset. We use CIFAR-10 datasets in Convolution Neural Networks. The CIFAR-10 datasets consist of 60000 images with 32×32 in 10 classes. It has been split into 55000 training images, 5000 validation images, and 10000 test images, respectively. Note that in fact our method does not need the validation sets to detect memorization in neural networks.

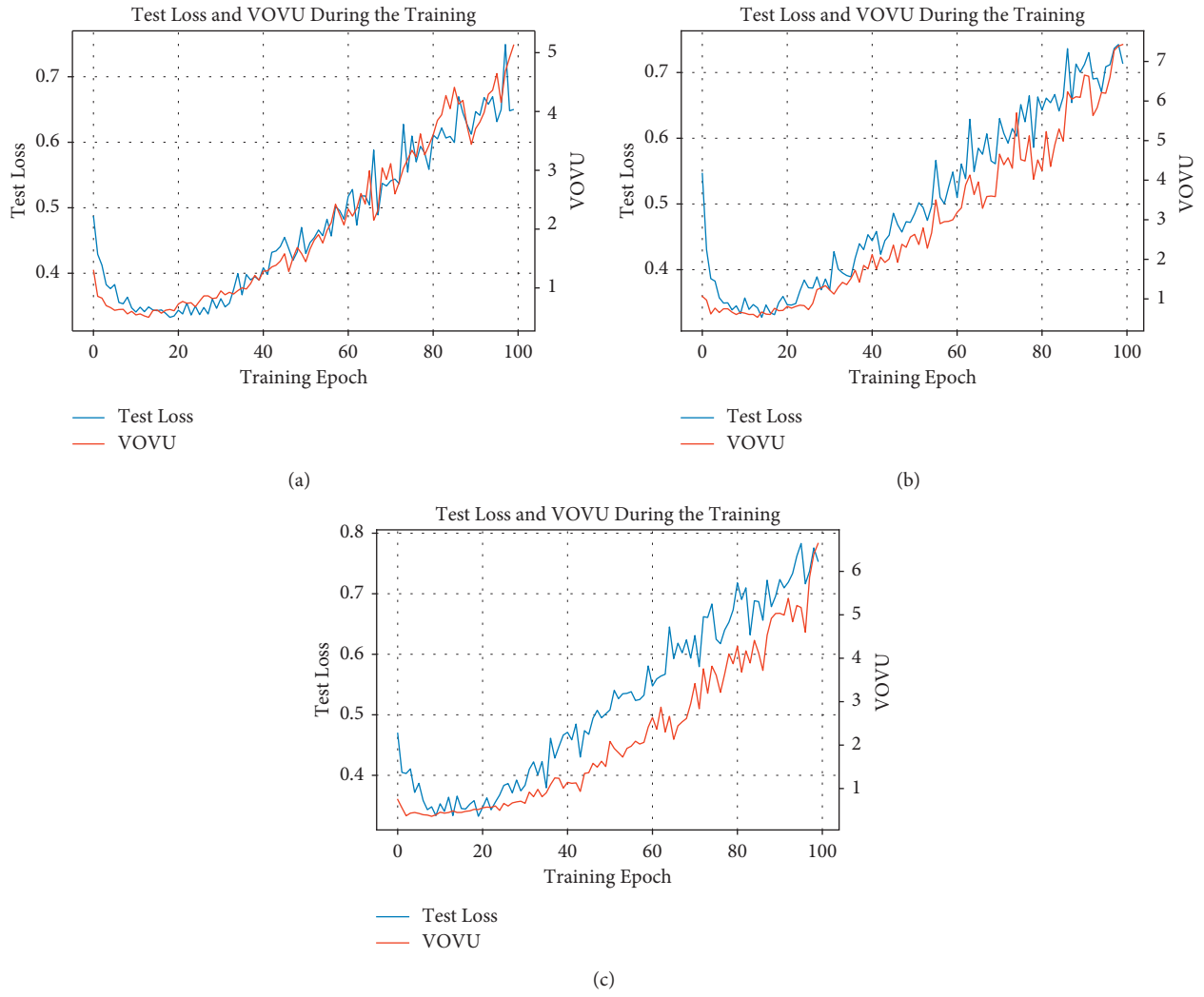

FIGURE 2: The dynamic trend of the test loss and VOVU trained on Fashion-MNIST in fully connected networks. The point when VOVU begins to rise is nearly the same point that the test loss starts to rise (the learning rate is set to 0.008 and 0.01 and 0.012 from left to right, respectively; the variance of the initialized weights is fixed at 0.1 in each network). (a) Learning rate = 0.008. (b) Learning rate = 0.01. (c) Learning rate = 0.012.

3.2.2. Experimental Settings. We choose to use a standard CNN architecture Quick-CIFAR-10, which composes two convolutional layers and two fully connected layers and a softmax output layer. Each convolutional layer is followed by a pooling layer. We use SGD with momentum algorithm to train these networks, the initial learning rate is 0.01, and the batch size is 128. The maximum epoch for each net is 160. We conduct experiments to show the correlation between VOVU and test error during the training. Networks are trained with different learning rate. Each experiment is repeated 5 times. We calculate the VOVU of the last hidden layer every two epochs and plot the dynamic change of test error and VOVU in the same plots.

3.2.3. Experimental Results. Results can be seen from Figure 4. For CIFAR-10 dataset, through testing on different learning rates, although the first part of the VOVU curve does not decrease much, the point when test loss starts to rise is the same as which VOVU starts to rise. We attribute this phenomenon to the initialization which is better for CIFAR-10, so it is also reasonable to stop at the point when VOVU starts to rise.

3.3. Experiment Analysis. From Figures 2–4, we can preliminarily conclude that the test loss and VOVU are highly positively correlated. In order to give a quantitative result of the correlation between test loss and VOVU during the training, we employ t-test to perform the significance test in the fully connected networks, which further demonstrate the effectiveness of our proposed VOVU metric.

We treat the value of VOVU and test loss during training as two sets of realizations of two random variables. The hypothesis test lets us decide whether the value of the population correlation coefficient is close to zero or significantly different from zero. The t-test returns two values: the correlation coefficient r and p value, in which r tells us

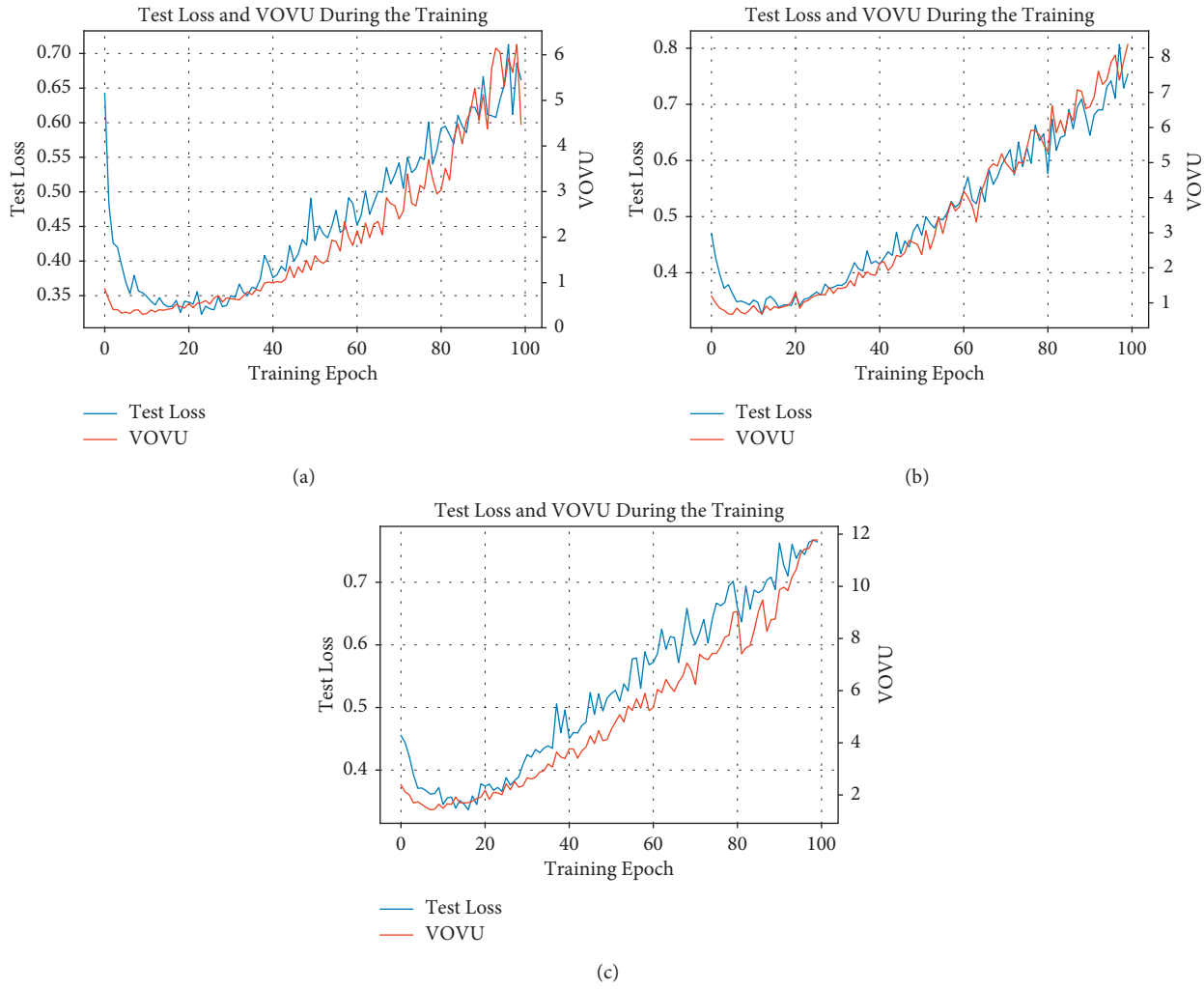

Figure 3: The dynamic trend of the test loss and VOVU trained on Fashion-MNIST in fully connected networks.The point when VOVU begins to rise is nearly the same point that the test loss starts to rise (the variance of truncated Gaussian distribution of the initialized weight matrix in each nets is 0.05 and 0.1 and 0.15 from left to right, respectively). (a) Variance = 0.05. (b) Variance = 0.1. (c) Variance = 0.15.

about the strength and direction of the linear relationship between test loss and VOVU. If the p value is less than the prechosen significance level α ($\alpha = 0.05$ in our paper), we conclude that there is sufficient evidence that there is a significant linear relationship between test loss and VOVU.

Both the correlation coefficient r and its corresponding p value are summarized in Table 1. The network has 400, 200, and 100 hidden units in each layer. More experimental results are summarized in Table 1.

3.4. Comparison with Other Methods. We compare VOVU with other early stopping methods. One is the commonly used validation-based early stopping and the other is single direction reliance proposed by [19].

In practice, early stopping is the most commonly employed method in the training of neural networks. There are various kinds of early stopping rules which are worked by splitting the original training set into a new training set and a validation set. The error on the validation set is used as a proxy for the generalization error in determining when overfitting has begun. Among these methods, validation-based early stopping is the most widely used method for detecting memorization (overfitting). Another promising method proposed by Morcos et al. [19] is single direction reliance, which we refer to as SDR. However, the number of the samples used has an obvious impact on the final estimation precision performance of both methods.

As shown in Table 1, in order to take a further step to analyze the influence of the number of sampling data, we compare our method with other two methods in terms of the robustness of correlation coefficient with test loss when using different numbers of samples. We observe that the dynamic change of the correlation coefficient r with different number of the training samples is used in the three methods. In addition, we measure the absolute difference value d between the extrema of test loss curve and the extrema of comparison method, which characterize the quality of each early stopping method.

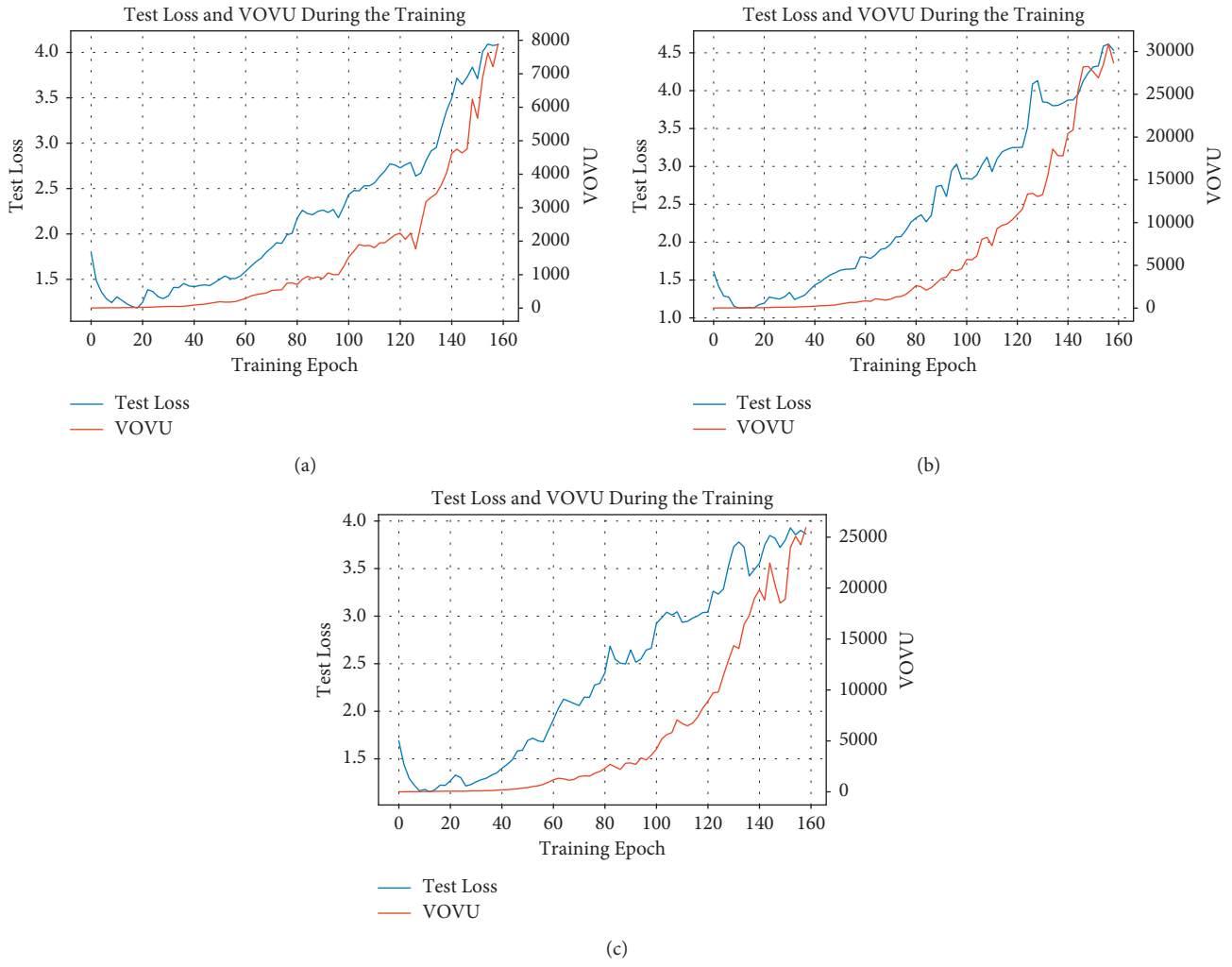

Figure 4: The dynamic trend of the test loss and VOVU trained on CIFAR-10 in convolutional neural networks. The first part of the VOVU curve does not decrease much, but the point when test loss starts to rise is the same point VOVU starts to rise (the learning rate is set to 0.0008, 0.001, and 0.0012 from left to right, respectively). (a) Learning rate = 0.0008. (b) Learning rate = 0.001. (c) Learning rate = 0.0012.

Table 1: Comparison results of VOVU and other early stopping methods.

Number of samples	VOVU			Validation-based early stopping			SDR		
	r	p value	d	r	p value	d	r	p value	d
$N = 5000$	0.981	$4.48e-72$	4	0.994	$3.45e-97$	0	0.788	$2.64e-63$	6
$N = 3000$	0.972	$9.45e-64$	4	0.991	$8.60e-88$	0	0.757	$5.23e-56$	6
$N = 2500$	0.971	$9.45e-64$	4	0.991	$8.60e-88$	0	0.714	$4.50e-33$	9
$N = 2000$	0.976	$2.08e-66$	1	0.987	$4.17e-81$	2	0.674	$2.32e-45$	11
$N = 1000$	0.974	$1.01e-65$	2	0.988	$6.64e-83$	5	0.641	$3.32e-23$	12
$N = 500$	0.969	$6.96e-62$	6	0.983	$3.33e-74$	13	0.608	$2.23e-23$	15
$N = 250$	0.978	$2.77e-69$	3	0.953	$1.78e-52$	18	0.554	$9.23e-16$	23

It can be seen from Figure 5(a) that though all three methods show correlation with generalization error, VOVU robustly has a high correlation coefficient with test loss, while decreasing the number of data used has a negative impact on the two other methods especially SDR. As shown in Figure 5(b), VOVU robustly detects the transition between overfitting and underfitting during training, while reducing the amount of data used will negatively affect the other two methods.

This phenomenon shows that VOVU has an advantage over the other two methods in the robustness of the number of data samples. In real AI application scenarios, few training samples are usually available and there is no available data used for validation, which makes the training of deep architectures

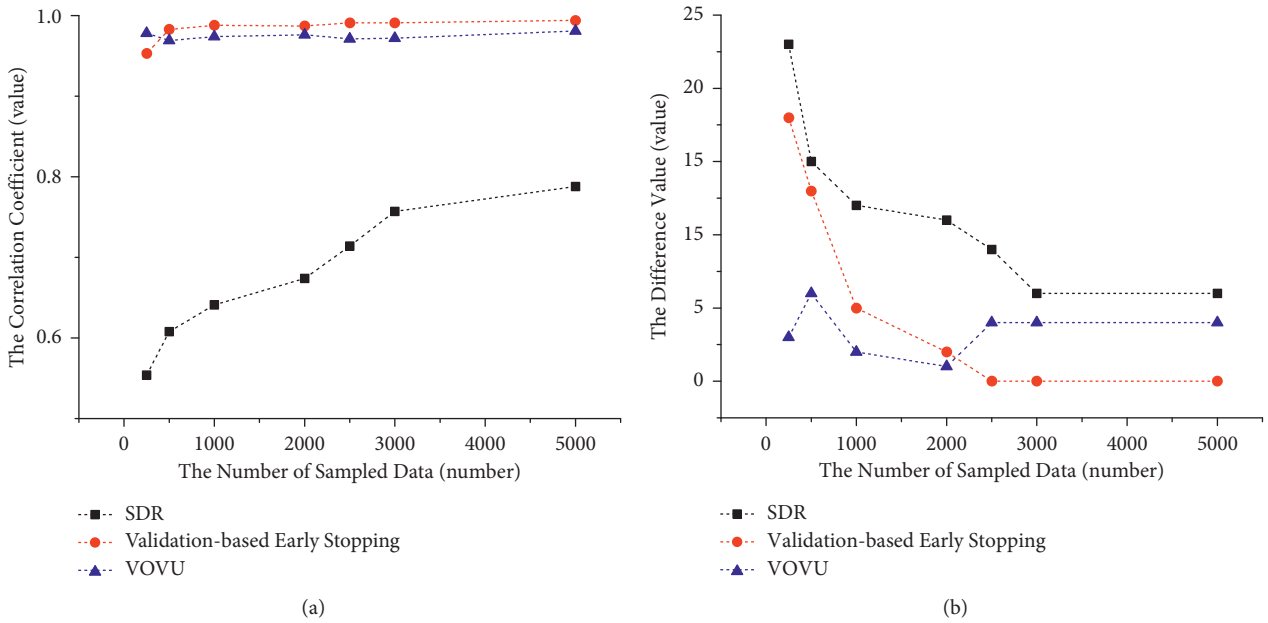

Figure 5: The comparison between VOVU and other early stopping methods in terms of the correlation coefficient r and the difference value d when sampling different number of data. (a) The comparison about r. (b) The comparison about d.

difficult. In this case, compared to other methods, VOVU provides a more practical method to prevent overfitting and predict generalization during training neural networks, since it does not need validation sets.

3.5. The Influence of Dropout on VOVU. In this section, we explain why dropout performs well in various deep neural networks through the perspective of unit's importance. As we have discussed previously, dropout is a regularization method to avoid overfitting by discouraging networks' reliance on single units. It is similar to our intuition that units should provide equal contribution to final decision, though we construct a new statistic to measure this reliance quantitatively in this work.

It is natural to ask how dropout can influence the VOVU during the training process. Based on above analysis, we conjecture that dropout can also decrease VOVU during the training process. Therefore, we perform experiments which implement dropout to confirm our conjecture. The results are exhibited in Figure 6. Specifically we show a comparison between training with dropout and without dropout in fully connected networks and convolutional neural network. In these experiments, whether training with dropout or not, our proposed measure VOVU consists with the trends of test loss, it confirms the generality of VOVU. What is more surprising is that when training the neural networks with dropout, the VOVU at each epoch is lower than that of training without dropout. This phenomenon provides a new perspective for explaining why dropout performs well in various deep neural networks.

Furthermore, these results further validate that our method can predict the generalization. We will discuss the application of this result in the next section.

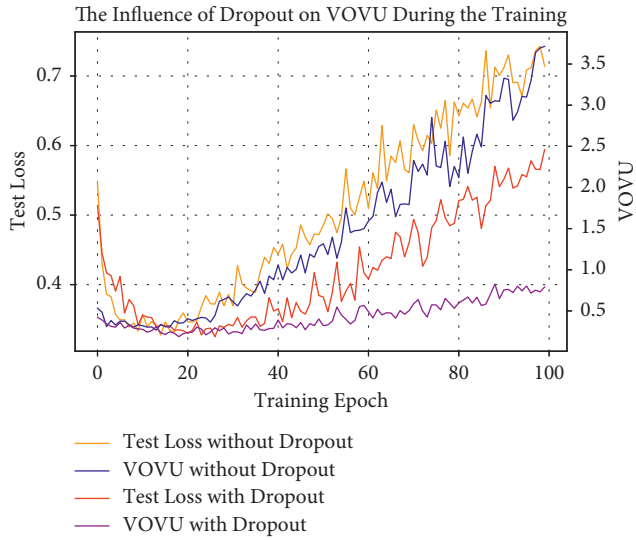

Figure 6: Dropout decrease VOVU during the training process, considering the different scale of VOVU and test loss, we multiply 0.5 with VOVU (without Dropout) and VOVU (with Dropout) for visual purpose.

4. Discussion

In this work, motivated by the researches related to units' importance and generation in neural networks, we explore the introduction of a generalization evaluation index through taking full advantage of the predictability of the last layer of hidden layer units. We introduce the algorithm to compute VOVU to describe the discrepancy of unit's importance on training samples. The experiments show the positive relationship between VOVU and test loss, which is

commonly used as a proxy for the generalization. VOVU can effectively predict generalization and overfitting in neural network training. In particular, our method is more advantageous when training data is limited.

Obviously, there are infinitely many possibilities for choosing other forms of measure to capture the importance discrepancy among the last hidden units, but in this paper we adopt a simple yet effective metric VOVU to characterize this discrepancy. Surprisingly, it performed well in our experiments. Although more analysis is needed to give a clear guarantee, we find that our approach is successful in predicting generalization across a variety of neural network architectures and image classification datasets.

One clear extension of this work is to use these observations to construct a regularizer which decreases VOVU during the network training. As it happens, the most obvious candidate is dropout (or its variants). As seen from Figure 6, these results suggest that one is able to predict a network's generalization performance without inspecting a held-out validation or test set. This observation could be used in several interesting ways. Firstly, in situations where labeled training data is spare, calculating the VOVU could provide a tool to assess generalization performance without sacrificing training data to be used as a validation set. Secondly, by using VOVU, we can track the VOVU every several epochs; this metric could be used as a signal for early stopping or hyperparameter selection. The specific construction process of the regularizer will be necessary in the future.

5. Conclusion

In this paper, we first take an empirical approach to understand what differentiates neural networks which are generalized from those that do not. We propose to define VOVU to measure the correlations among them with the assumption that the features provided by the last hidden layer should be disentangled for the final classification. Based on this intuitive assumption, we show a series of experimental results to testify the correlation between test loss and VOVU of the last hidden layer in neural networks. All of these results demonstrate the effectiveness of VOVU for predicting generalization and the potential for early stopping and hyperparameter selection in neural network training. This work also makes a potentially surprising observation about the role of independent units and provides a new perspective to improve generalization performance in the deep neural networks.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Project of Grant No. 17JCQNJC00500.

References

[1] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the International Conference on Machine Learning*, pp. 1243–1252, PMLR, Sydney, Australia, August2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[3] Z. Gao, W. Dang, X. Wang et al., "Complex networks and deep learning for EEG signal analysis," *Cognitive Neurodynamics*, vol. 15, no. 3, pp. 369–388, 2021.

[4] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[5] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," *Constructive Approximation*, vol. 26, no. 2, pp. 289–315, 2007.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[7] S. Ioffe and C. Szegedy, "batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.

[8] M. Gabrié, A. Manoel, C. Luneau, N. Macris, F. Krzakala, and L. Zdeborová, "Entropy and mutual information in models of deep neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1821–1831, Montréal, Canada, December2018.

[9] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," 2017, https://arxiv.org/abs/1703.00810.

[10] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proceedings of the 2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, Jerusalem, Israel, April 2015.

[11] A. Xu and M. Raginsky, "Information-theoretic analysis of generalization capability of learning algorithms," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2524–2533, Long Beach, CA, USA, December 2017.

[12] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: generalization gap and sharp minima," in *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, April 2017.

[13] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Sensitivity and generalization in neural networks: an empirical study," 2018, https://arxiv.org/abs/1802.08760.

[14] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2002.

[15] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 6240–6249, Long Beach, CA, USA, December 2017.

[16] A. Achille and S. Soatto, "Emergence of invariance and disentanglement in deep representations," *Journal of Machine Learning Research*, vol. 19, no. 1, pp. 1947–1980, 2018.

[17] K. Dhamdhere, M. Sundararajan, and Q. Yan, "How important is a neuron," in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[18] K. Liu, R. A. Amjad, and B. C. Geiger, "Understanding individual neuron importance using information theory," 2018, https://arxiv.org/abs/1804.06679.

[19] A. S. Morcos, D. G. Barrett, N. C. Rabinowitz, and M. Botvinick, "On the importance of single directions for generalization," in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[20] R. R. Selvaraju, P. Chattopadhyay, M. Elhoseiny et al., "Choose your neuron: incorporating domain knowledge through neuron-importance," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 526–541, Munich, Germany, September 2018.

[21] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra, "Reducing overfitting in deep networks by decorrelating representations," 2015, https://arxiv.org/abs/1511.06068.

[22] R. Yu, A. Li, C.-F. Chen et al., "Nisp: pruning networks using neuron importance score propagation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9194–9203, Salt Lake City, UT, USA, June 2018.

[23] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[24] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," 2016, https://arxiv.org/abs/1610.01644.

[25] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proceedings of the Advances in neural information processing systems*, pp. 3320–3328, Montreal, Canada, December 2014.

[26] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: visualising image classification models and saliency maps," 2013, https://arxiv.org/abs/1312.6034.

[27] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[28] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 6076–6085, Long Beach, CA, USA, December 2017.

[29] A. Morcos, M. Raghu, and S. Bengio, "Insights on representational similarity in neural networks with canonical correlation," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5727–5736, Montréal, Canada, December2018.