VECA: A Method for Detecting Overfitting in Neural Networks (Student Abstract)

Liangzhu Ge,[1] **Yuexian Hou,**[1] * **Yaju Jiang,**[1] **Shuai Yao,**[1] **Chao Yang**[2]
[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Alibaba Group
{geliangzhu123, yxhou, jiangyaju, shuaiyao}@tju.edu.cn,
xiuxin.yc@alibaba-inc.com

Abstract

Despite their widespread applications, deep neural networks often tend to overfit the training data. Here, we propose a measure called VECA (**V**ariance of **E**igenvalues of **C**ovariance matrix of **A**ctivation matrix) and demonstrate that VECA is a good predictor of networks' generalization performance during the training process. Experiments performed on fully-connected networks and convolutional neural networks trained on benchmark image datasets show a strong correlation between test loss and VECA, which suggest that we can calculate the VECA to estimate generalization performance without sacrificing training data to be used as a validation set.

Introduction

In recent years, deep neural networks with a large number of parameters have achieved great success on many AI tasks. However, overfitting is a serious problem in such networks. It raises an important question: how should we detect overfitting when training a neural network? To predict the tendency of the test error, validation-based early stopping is a usually used one in practice. However, in real AI application scenarios, few training samples are usually available which makes it difficult to know when to stop training the deep neural network.

By analyzing the covariance matrix of the hidden representations of trained networks, we construct a statistic metric to monitor the training dynamics of neural networks and find that the metric is highly correlated with the test loss both in fully connected neural networks and convolutional neural networks, thus it could serve as an alternative method for early stopping.

Motivation

Our motivation comes from the researches with regards to the units' importance. Some researches think that a unit's importance is influenced by the variance of its outputs with respect to the neural network's inputs (Guyon and Elisseeff 2003; Dhamdhere, Sundararajan, and Yan 2019). Other researches argue that its correlation with other units is an important factor for avoiding overfitting (Cogswell et al. 2016). It is natural that we propose to study the property of the empirical covariance matrix of the activation matrix which encodes both the variance term and covariance term into a single matrix.

Morcos et al found that as networks begin to overfit, they become more reliant on the single directions (Morcos et al. 2018). Since the outputs of the last hidden layer provide the features for the output layer to make the final decision, we divide any neural network into two parts: the layers before the output layer and the output layer. We focus on the outputs of the last hidden layer and regard them as a random vector.

We calculate the Variance of Eigenvalues of Covariance matrix of Activation matrix (VECA) to characterize the reliance. Our metric VECA essentially characterizes variance of the variances of the principal components of the original activation matrix. In the sense of PCA, VECA can be thought of as a simple measure for measuring feature space's redundance. The more the network relies on single directions, the larger the VECA will be. A small VECA means that the units in the hidden layer cannot be summarized well by just few first components. It's known that the irredundant features benefit the final classification performance while redundant features harm the performance.

Proposed Method

Let l denote the last hidden layer of a neural network, which has n units. Given m data points $S = \{x_1, \cdots, x_m\}$, which are randomly drawn from the **training set**. For the layer l, let $\boldsymbol{Z}^l_{i\cdot}$ be the activation outputs on x_i,

$$\boldsymbol{Z}^l_{i\cdot} = [Z^l_{i1}, \cdots, Z^l_{in}] \tag{1}$$

where $i = 1, 2, \cdots, m$ and $x_i \in S$. Supposing the outputs of the j^{th} unit in the hidden layer is a random variable, let Z_{ij} be the i^{th} independently drawn observation on the j^{th} random variable ($j = 1, \cdots, n$). These observations can be arranged into m column vectors, each with n entries, with

*Corresponding author:Yuexian Hou (yxhou@tju.edu.cn), This work is funded in part by the National Key R&D Program of China (2017YFE0111900), the National Natural Science Foundation of China (61876129) and the European Unions Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No. 721321.

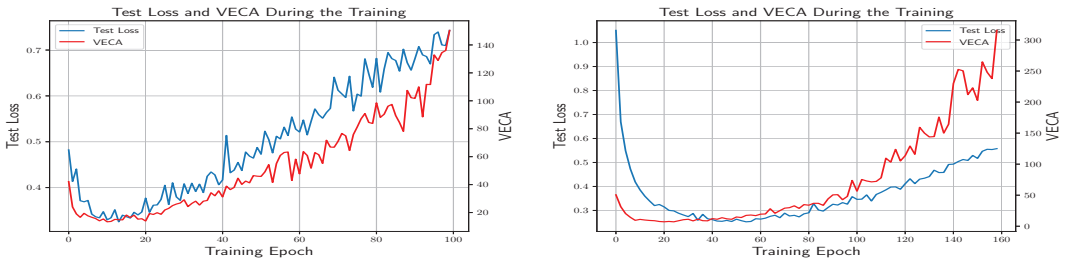

Figure 1: Left: The training dynamic of the test loss and VECA trained on Fashion-MNIST in fully connected network. Right: The training dynamic of the test loss and VECA trained on Fashion-MNIST in convolutional neural network.

the $n \times 1$ column vector giving the i^{th} observations of all variables being denoted $\boldsymbol{Z_i}$. $(i = 1, \cdots, m)$.

Broadening our view from a observation to the collection of m observations, the layer's outputs over m inputs can be thought of as a set of neuron vectors, which could be arranged as the columns of a activation matrix $\boldsymbol{Z}$, so that

$$\boldsymbol{Z} = [\boldsymbol{Z_1}, \boldsymbol{Z_2}, \cdots, \boldsymbol{Z_m}] \tag{2}$$

$$\bar{\boldsymbol{Z}} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{Z}_{i\cdot} \tag{3}$$

where $\bar{\boldsymbol{Z}}$ denote the sample mean vector, which is a column vector whose j^{th} element $\bar{Z}_j$ is the average value of the m observations of the j^{th} unit. Then the covariance matrix $\boldsymbol{C_Z}$ is defined via:

$$\boldsymbol{C_z} = \frac{1}{m-1} \sum_{i=1}^{m} (\boldsymbol{Z}_{i\cdot} - \bar{\boldsymbol{Z}})(\boldsymbol{Z}_{i\cdot} - \bar{\boldsymbol{Z}})^T \tag{4}$$

the $(j,k)^{th}$ element of this covariance matrix C_z is given by

$$C_{jk} = \frac{1}{m-1} \sum_{i=1}^{m} (Z_{ij} - \bar{Z}_j)(Z_{ik} - \bar{Z}_k) \tag{5}$$

where $j, k = 1, 2, \cdots, n$.

According to the definition, the covariance matrix is a symmetric matrix with the variances on its diagonal and the covariances off-diagonal. $\boldsymbol{C_z}$ is also positive semi-definite, thus the eigenvalues of this matrix are non-negative.

Once we get the covariance matrix $\boldsymbol{C_z}$, we perform eigenvalue decomposition of the $\boldsymbol{C_z}$, obtaining a set λ which contains n eigenvalues $\lambda = \{\lambda_1, \cdots, \lambda_n\}$, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n \geq 0$, where n is the number of units in the last hidden layer. Then, the empirical variance of the n eigenvalues can be written as

$$Var(\lambda) = \frac{1}{n} \sum_{i=1}^{n} (\lambda_i - \bar{\lambda})^2 \tag{6}$$

where $\bar{\lambda}$ is the average value of these n eigenvalues, i.e., $\bar{\lambda} = \frac{1}{n} \sum_{i=1}^{n} \lambda_i, i = 1, \cdots, n$

Finally, we get a statistic called VECA, whose value equals to $Var(\lambda)$. A $n \times n$ covariance matrix captures the spread of n-dimensional extracted feature. We expect the feature spread uniformlly across the n dimensions rather than just few first principal dimensions.

Experimental Results

To evaluate the effectiveness of VECA for detecting overfitting, we test our method on two kinds of networks: fully connected networks and convolutional networks. Both are trained on the Fashion-MNIST datasets.

We train 50 fully-connected network models with different hyperparameter settings, including the size of hidden units, learning rate, and different initialization of weights. We calculate the VECA of the last hidden layer every two epochs during the training process and plot the dynamic change of test loss and VECA in the same plots.

Interestingly, we can see from Figure 1 that the point when VECA begins to rise is nearly the point when the test loss starts to rise. In addition, we find that test loss and VECA are highly positively correlated both in fully connected networks and convolutional networks. Furthermore, for any two points with small training loss, with a large probability, we can conclude that the larger VECA is, the worse the net's generalization ability will be.

Conclusion

Our results suggest that we can calculate the VECA to estimate generalization performance without sacrificing training data to be used as a validation set, especially when labeled training data is sparse. Another clear extension of this work is to construct a regularizer to decrease VECA during the network's training process. Dropout is an obvious candidate since it can reduce the network's reliance on single directions.

References

Cogswell, M.; Ahmed, F.; Girshick, R. B.; Zitnick, L.; and Batra, D. 2016. Reducing overfitting in deep networks by decorrelating representations. In *ICLR*.

Dhamdhere, K.; Sundararajan, M.; and Yan, Q. 2019. How important is a neuron? In *ICLR*.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3:1157–1182.

Morcos, A. S.; Barrett, D. G. T.; Rabinowitz, N. C.; and Botvinick, M. 2018. On the importance of single directions for generalization. In *ICLR*.