Analyzing and Quantifying Generalization in Convolutional Neural Networks

Yang Zhao  
Tsinghua University  
Beijing, China  
zhao-yanl8@mails.tsinghua.edu.cn

Hao Zhang  
Tsinghua University  
Beijing, China  
haozhang@tsinghua.edu.cn

Abstract

Generalization is the key capability of convolutional neural networks (CNNs). However, it is still quite elusive for differentiating the CNNs with good or poor generalization. It results in the barrier for providing reliable quantitative measure of generalization ability. To this end, this paper aims to clarify the generalization status of individual units in typical CNNs and quantify the generalization ability of networks using image classification task with multiple classes data. Firstly, we propose a feature quantity, role share, consisting of four discriminate statuses for a certain unit based on its contribution to generalization. The distribution of role shares across all units provides a straightforward visualization for the generalization of a network. Secondly, using only training sets, we propose a novel metric for quantifying the intrinsic generalization ability of networks. Lastly, a predictor of testing accuracy via only training accuracy of typical CNN is given. Empirical experiments using practical network model (VGG) and dataset (ImageNet) illustrate the rationality and effectiveness of our feature quantity, metric and predictor.

1. Introduction

Due to excellent ability of generalization to unseen data, convolutional neural networks (CNNs) achieve great success in solving many real-world vision-based problems. However, the thorough understanding towards generalization of CNNs is still rather limited. Many valuable works contribute to the topic from various lines \cite{23,17,6,12,20,8,13,21}, but there remains largely unknown mysteries towards the inner structures of CNNs with good or bad generalization.

Specifically, the functions and effects of network units for the overall generalization performance of network models need to be clarified with the potential applications on its design and simplification. It is not strange that much research attention has been drawn on this point from different pathways \cite{22,25,14,19,2,9,18,7}. But the status and usefulness of individual network units still remain vague for lacking in effective quantitative measure for its importance, especially the generalization capability.

The cumulative ablation test is a powerful tool to investigate the contribution of individual network unit to the overall performance of network models. It was utilized to study the average influence of randomly chosen unit sets with different sizes on the generalization of CNNs \cite{15}. Furthermore, the effect of individual unit was examined by cumulative ablation test of unit sets with importance order being obtained by a specific semantic analysis network and certain kind of ‘high level concepts’ created manually \cite{3,4}. Without doubt, the key ingredient of cumulative ablation test is the rational specification of importance for network units. It should be defined objectively without any help of human made concepts and highly match with the network performance on common datasets.

A general framework for quantitative assessment of effectiveness and importance of individual network unit was built via method from algebraic topology \cite{24}. Some ablation experiments with practical network model (VGG) and dataset (ImageNet) was conducted for all the units in the same layer (5-3) on the data of single class. Four kinds of roles was proposed for the units on single class according to their different contributions on the generalization of network. Despite being the valuable attempt to analyze the status of units in the network via completely objective means, it is heavily insufficient to obtain any solid conclusion on the network generalization only using data from single class. Empirical results on substantially cross classes are indispensable. Moreover, quantifying generalization ability of CNNs is a long standing unsolved problem in the field of machine learning. It is worth probing the possibility of giving a preliminary solution to above puzzle based on new results on data from multiple classes. This is just the target of this paper.
2. Methods

2.1. Role share of Individual Units

Consider an image classification task with dataset \( \mathcal{D} \) including \( N \) classes \( \{D_k\}_{k=1}^N \), \( \mathcal{D} = D_1 \cup \cdots \cup D_N \), and unit set \( \mathcal{U} = \{U_k\}_{k=1}^M \) in the specific convolution layer in a network model. For each data class \( D_k \), unit set \( \mathcal{U} \) could be categorized into four subsets undertaking different roles [24], namely core units \( \mathcal{U}_c(k) \), overfitted units \( \mathcal{U}_g(k) \), generalizing units \( \mathcal{U}_o(k) \) and confusing units \( \mathcal{U}_f(k) \). Clearly \( \mathcal{U} = \mathcal{U}_c(k) \cup \mathcal{U}_g(k) \cup \mathcal{U}_o(k) \cup \mathcal{U}_f(k) \) for each \( k \). We define the role share of certain unit \( U_n \) for all the class as the four-dimensional vector

\[
RS_n = (RS_n^{(c)}, RS_n^{(o)}, RS_n^{(g)}, RS_n^{(f)}) \tag{1}
\]

where

\[
RS_n^{(R)} = \frac{1}{N} \sum_{k=1}^{N} I(U_n \in \mathcal{U}_R(k)), \quad R = c, o, g, f \tag{2}
\]

Just as market share, role share of unit \( U_k \) indicates the relative ratios of its roles (Core, Overfitted, Generalizing, Confusing) over all the data classes. The elements of \( RS_n \) is not independent because

\[
RS_n^{(c)} + RS_n^{(o)} + RS_n^{(f)} + RS_n^{(g)} = 1 \tag{3}
\]

Cooperation and competition among units during training processes inevitably lead to the overall state of network model, which is the basis of network efficiency and generalization. Role share defined in Eq(1) and Eq(2) reflects certain kinds of characteristics of a network unit. The distribution of role shares over all the units is an effective indication for overall performance of network model. It could be used to discriminate network models with diverse generalization capacities. Due to Eq(3), we use the three-dimensional vector

\[
(RS_n^{(c)}, RS_n^{(o)}, RS_n^{(f)}) \tag{4}
\]

instead of Eq(1) to represent role share for the convenience of illustration and without any information loss (Here generalized share is neglected for its extreme rareness).

2.2. Quantitative Metric of Generalization

It is quite hard to quantify the generalization ability of CNNs, especially in the training stage without any information of testing data. However, with the help of algebraic topology tools, we can give the feature entropy \( H_{i,j,k} \) that is critical for importance assessment of individual unit \( U_j \) [24]. In brief, \( H_{i,j,k} \), the topological quantity obtained through calculating the \( k \)-th Betti number of certain simplicial complex, so called clique topology [10], extracted from remarkable components in the feature map as convolution output of unit \( U_j \) with respect to data class \( D_i \). It was shown that all units could be sorted into an ordered list according to the importance derived from their feature entropy. This list is the prior for the cumulative ablation test, which is the powerful measure for analyzing the generalization only using training data on SINGLE class. Hence it is necessary to extend our method to the multiple classes case.

Let’s start with feature entropy \( H_{i,j,k} \) of unit \( U_j \) with respect to class \( D_i \). We give the importance rank on class \( D_i \) for unit set \( \mathcal{U} \) based on sorting of the feature entropy set \( \{H_{i,j,k}\}_{j=1}^M \).

\[
H_{i,(1),k} \geq H_{i,(2),k} \geq \cdots \geq H_{i,(M),k} \tag{5}
\]

Afterwards, the cumulative ablation tests are conducted in both ascent and descent order based on the importance rank Eq(5), just as illustrated in Fig[1].

![Figure 1. Sketch figure of \( E(n,D_j) \) (blue) and \( E_r(n,D_j) \) (green), separately resulted from the descent and ascent cumulative ablation process. The marks represent for \( n_{min}(D_j) \) and \( n_{rmax}(D_j) \).](image)

For specific data class \( D_j \), the accuracy curve \( E(n,D_j) \) along with descent cumulative ablation process and corresponding \( E_r(n,D_j) \) along with ascent (reverse) cumulative ablation process are shown. There are two key points deserve more attention, which are \( n_{min}(D_j) \) and \( n_{rmax}(D_j) \),

\[
n_{min}(D_j) = \inf \{ n | E(n,D_j) \leq CL \} \tag{6}
\]

and

\[
n_{rmax}(D_j) = \arg \max_n E_r(n,D_j) \tag{7}
\]

where \( CL \) is a constant called ‘Chance Level’ [4, 15]. Its typical value is \( 1/N \).

It should be remarked that the accuracy in the curves \( E(n,D_j) \) and \( E_r(n,D_j) \) are build only on training data and network intrinsic structure. In other words, \( n_{min}(D_j) \) and \( n_{rmax}(D_j) \) both could be calculated without any testing data and any human-made matter. Therefore, a novel concept, Generalized Rate \( \zeta(D_j) \), is introduced to account for quantifying generalization ability of CNNs. That is,

\[
\zeta(D_j) = \frac{1}{2} - \frac{n_{min}(D_j) - n_{rmax}(D_j)}{2M} \tag{8}
\]
here \( M = |\mathcal{U}| \) is the size of unit set.

Although being somewhat obscure, generalized rate can be explained intuitively. Firstly, the core of definition Eq(3) is \( n_{\min}(D_j) - n_{\max}(D_j) \). The objective of the normalization operation is just to make \( \zeta(D_j) \) lie in \([0, 1]\) for convenience of application. Secondly, the basic logic behind generalized rate is: the larger difference between \( n_{\min}(D_j) \) and \( n_{\max}(D_j) \), the better generalization. In fact, it is plausible that CNNs with good generalization should have a few core units, abundant confusing units and few overfitted units. So for curve \( E(n, D_j) \), \( n_{\min}(D_j) \) exactly is impacted by the size of core unit set and overfitted unit set, which should be limited in a narrow scope. On the other hand, the slight rise of curve \( E_r(n, D_j) \) indicates that the units being deleted are basically confusing units because the deletion of overfitted units should lead to decline of \( E_r(n, D_j) \). Thus \( n_{\max}(D_j) \) is the sign of size of confusing unit set to a large extent. That is why we choose Eq(8) as the quantitative metric for generalization ability of CNN.

Until now, generalized rate defined in Eq(8) is still available only for certain class \( D_j \). It’s time to extend the definition to the scenario of multiple classes.

For the set \( \{D_k\}_{k=1}^N \) of all classes, the total generalized rate is defined as

\[
\zeta = \frac{1}{N} \sum_{j=1}^{N} \zeta(D_j) \quad (9)
\]

Here we use simply the average to make fusion for generalized rate on various data classes to achieve the total generalized rate. There is hardly any necessity to incorporate extra complexity for little performance benefit. We claim that the total generalized rate Eq(9) is a purely computable quantitative metric of generalization of CNNs without any human factor, more important, without any information on testing data. Using the total generalized rate, we can infer the level of generalization of a network in the training stage. Needless to say, it is of great value for the adjustment, optimization and simplification of CNNs. Our empirical results strongly support its rationality and effectiveness.

Furthermore, the accuracy performance of a CNN even could be predicted in some extent with only training data. Although being a bit incredible, the prediction model can be formulated as

\[
\text{acc}_\text{gen} = \alpha \cdot (\zeta \cdot \text{acc}_\text{train})^\gamma + \beta \quad (10)
\]

here \( \alpha, \gamma \) and \( \beta \) are parameters just depend on the network model and common training data. Extra introduction of any human-created concept and label is no need. Hence the prediction model Eq(10) has substantial potential of practical usage.

| Model     | Training Accuracy | Testing Accuracy |
|-----------|-------------------|------------------|
| Pretrained| 0.754             | 0.700            |
| Model A   | 0.633             | 0.556            |
| Model B   | 0.779             | 0.662            |
| Model C   | 0.864             | 0.598            |
| Model D   | 0.892             | 0.490            |
| Model E   | 0.975             | 0.411            |

Table 1. Training and testing accuracies of 6 models.

3. Experiment Results

3.1. Experiment Setup

We focus experiments on the Imagenet classification task using the VGG16 network architecture. Keras pretrained model and five other VGG16 models trained from scratch by us on the whole Imagenet training set with diverse generalization are prepared for the demonstration of role share and our quantitative metric. We choose all the \( M = 512 \) units in the last convolution layer “conv5_3” as the unit set \( \mathcal{U} \). A randomly chosen subset of Imagenet including 200 classes is set as \( \mathcal{D} \) and \( N = 200 \). Each class has a training set and a testing set. The training set including 100 images for each class is randomly sampled. And the testing set is just the Imagenet validation set which contains 50 images for each class. All the images involved in our demonstration (such as all the reported accuracies and unit categorization) are simply resized to \( 224 \times 224 \) without using any extra data augmentation. Table 1 shows the training and testing accuracies of all the 6 models used in our experiments.

3.2. Visualization of Role Share

The role share of each network unit \( U_i \in \mathcal{U} \) in distinct network models are calculated and the results are depicted in Fig 2.

Each of 3D figures on the left side contains totally \( 512 \) points (512 units) in the six network models with coordinates just as Eq(4). Three components in role share are shown separately on the right side and their average values are also computed via Eq(11) and tagged under the corresponding figures.

\[
\overline{\text{RS}^{(R)}} = \frac{1}{M} \sum_{j=1}^{M} \text{RS}_j^{(R)}, \quad R = c, o, f \quad (11)
\]

The differences among the first three models (Pretrained, A and B) are not significant. As the models with relatively good generalization, these models have similar characteristics that can be seen from figures: a small share of core units, abundant share of confusing units and few share of overfitted units. The slight discrepancy lies in the means and variances of components in role share. The variance would rise along with the decline of generalization.
| Model   | role share | \( RS^{(3)} \) | \( RS^{(e)} \) | \( RS^{(f)} \) |
|---------|------------|----------------|----------------|----------------|
| Pretrained | \[ RS^{(3)} = 0.161 \] | \[ RS^{(e)} = 0.031 \] | \[ RS^{(f)} = 0.791 \] |
| Model A | \[ RS^{(3)} = 0.175 \] | \[ RS^{(e)} = 0.032 \] | \[ RS^{(f)} = 0.779 \] |
| Model B | \[ RS^{(3)} = 0.259 \] | \[ RS^{(e)} = 0.034 \] | \[ RS^{(f)} = 0.689 \] |
| Model C | \[ RS^{(3)} = 0.286 \] | \[ RS^{(e)} = 0.115 \] | \[ RS^{(f)} = 0.594 \] |
| Model D | \[ RS^{(3)} = 0.254 \] | \[ RS^{(e)} = 0.339 \] | \[ RS^{(f)} = 0.407 \] |
| Model E | \[ RS^{(3)} = 0.177 \] | \[ RS^{(e)} = 0.641 \] | \[ RS^{(f)} = 0.209 \] |

Figure 2. Role shares of the six network models (separated), where each row stands for the results of a distinct model. The left column presents 3D visualizations of the distribution of role shares across all the 512 units, and its corresponding projection in the coordinate planes. The right column shows the distribution of each component in the role share for the same 512 units.
This phenomena becomes even more remarkable in the last three models (C, D and E), which have poor generalization. Moreover, the changes of average values also can not be ignored. The number of core unit become larger and more fluctuant. The size of overfitted units expands obviously. The mean quantity of confusing units drops sharply and the overall distribution becomes chaotic. All of the above appearances have explicit conflict with the indicator of good generalization. So the role share analysis of network units could give us some kinds of auxiliary criterion for goodness of generalization ability of CNNs.

Results of all the six models are depicted together in Fig.3 for the sake of intuitive comparison. It can be seen clearly that good separability exists for these models in the three-dimensional space of role shares.

3.3. Discussion on Role Share

With different generalization, trained models always present diverse distributions of units’ role shares. It is interesting how should the role share be specified for a CNN with good generalization?

For core unit share set \( \{ RC_n^{(c)} \}_{n=1}^{512} \), its element \( C_n^{(c)} \) is the share of classes that the unit \( U_n \) really plays a decisive part. The determination of ideal distribution for \( \{ RC_n^{(c)} \}_{n=1}^{512} \) is essentially a problem of multi-agent system design [1]. In particular, one of the design objective for multi-agent system is allocating multiple tasks to multiple agents. Analogously, the unit here corresponds to the agent and the classification of images in multiple classes corresponds to the tasks. How to coordinate individual agents (units) to efficiently finish complex tasks (classification)? Typically, the agents are initialized randomly with no functionality preference. By gradually training, they automatically learn to become specialized to only a small subset of tasks [16]. Broad research show that specialization could reduce both the physical and virtual interference that occurs in agents and lead to overall increase in productivity of whole system [1]. So, \( RC_n^{(c)} \) should not be too large in the scenario of multiple classes for each unit ought to be in charge of only a small portion of classes. Meanwhile, when tackling the same task, several agents (units) cooperate simultaneously could improve system efficiency. Therefore, too small \( RC_n^{(c)} \) should also be avoided. Hence most elements of \( \{ RC_n^{(c)} \}_{n=1}^{512} \) should be limited in a small neighborhood of zero, just as shown in Fig.2. The model having too many units with overly large \( RC_n^{(c)} \) probably could be detriment to generalization due to the bad specialization.

Since overfitted units are only effective on the training set, it is trivial that \( RC_n^{(o)} \) should be as few as possible for each unit \( U_n \). Fig.2 shows that set \( \{ RC_n^{(o)} \}_{n=1}^{512} \) lies close to zero for well generalized models. When a CNN is poorly generalized, large number of units will present high \( RC_n^{(o)} \), meaning that for many classes, these units are overfitted, that is to say, being only effective on the training set, not on the testing set. As ablation operation being performed, removing these units will impact the CNN training accuracy on the classes regarding these units as either core units or overfitted units. In a word, the more units with high \( RC_n^{(o)} \), the more severe decline on the training accuracy in a cumulative ablation test. This results in a more dramatically drop of the overall training accuracy. This is consistent with the observations in [15]: ”training accuracy drops more rapidly for less generalized CNNs when performing cumulative ablation operation”.

However, the statement in [15], ”a more generalized CNN relies less on single units”, deserves further query. In a sense, it implies that removing a single unit will be less impact for a more generalized CNN. It seems that the difference between core and confusing units is somewhat improperly ignored. Seeing Fig.2 for a well generalized CNN, removing a unit will surely heavily decrease the training accuracy on the classes for which the unit is a core unit. But for many other classes where this unit exists as a confusing unit simultaneously, it will slightly increase the training performance. The total effect of removing this unit on the overall training accuracy would present less drop. That is
the more comprehensive explanation for that the more generalized model seems less reliance on single units.

For confusing unit share set \( RC_n \) of \( n = 1 \), it distributes around some large value close to 1, being contrary to that of core unit share set, just as shown in Fig. 2. In fact, individual confusing units in a well generalized network actually present little key contribution on the majority of classes. Our observation here somehow coincides with the actual state of famous model BERT [5]. Being heavily trained, BERT finally presents no task preference, providing an excellent base for solving further specific tasks with only small effort of fine-tuning. Seemingly, most units in BERT exhibit confusing functionality. Fine-tuning seeks a few units suitable for the certain task and features them to be more task-specific. That is, transforming these units from confusing units to core units. As for most unsuitable units, they keep as confusing units. It is the remarkable characteristic of well generalized CNN models.

3.4. Evaluation of Quantitative Metric of Generalization

In this section, the effectiveness of quantitative metric of generalization proposed in section 2.2 is evaluated empirically for all the six network models. Firstly, we perform the cumulative ablation tests to get the key parameters, \( (E(n, D_j), E_r(n, D_j)) \), of each class \( D_j \) and each generalized rate \( \zeta(D_j) \) is calculated. Secondly, measurements of generalization using training and testing accuracy (Attention: accuracy here is summarized with both training and testing data) are defined. Correlation analysis between these measurements and our quantitative metric is conducted to show its rationality. Thirdly, the accuracy of our prediction model for testing accuracy is estimated to illustrate its usefulness.

Fig. 4A presents some common evolution examples of ablation curves \( E(n, D_j) \) and \( E_r(n, D_j) \) for the six models. It could be observed that difference between \( n_{\text{min}}(D_j) \) and \( n_{\text{rmax}}(D_j) \) roughly increases along with the improvement of generalization of models. For the models with better generalization, \( E(n, D_j) \) drops more sharply and \( E_r(n, D_j) \) holds for a longer time with slight rising until reaching the breakpoint and beginning to decrease drastically.

Fig. 4B makes a scatter plot for all the points \( \{(n_{\text{min}}(D_j), n_{\text{rmax}}(D_j))\}_{j=1}^{200} \) with respect to 200 data classes. According to our statement in section 2.2 for better generalization on \( D_j \), \( n_{\text{min}}(D_j) \) ought to be smaller and \( n_{\text{rmax}}(D_j) \) should be larger. In other words, the points should be concentrate around the top left corner of the figure. The degrees of aggregation of those points in figures could be regarded as some kind of manifestation of generalization. The situation in figures shows that our choice of key parameters, \( (n_{\text{min}}(D_j), n_{\text{rmax}}(D_j)) \), actually reflect the state of generalization of models. The degrees of aggregation of good models (Pretrained, A and B) are apparently higher than bad models (C, D and E). Fig. 4C illustrate the situation using histogram of generalized rate \( \zeta(D_j) \) calculated via \( \theta \) instead of scatter plot. The same impression could be formed from another viewpoint.
For more clarity, the simple average positions

$$\left( \bar{n}_{\text{min}}, \bar{n}_{\text{rmax}} \right) = \frac{1}{200} \left( \sum_{j=1}^{200} n_{\text{min}}(D_j), \sum_{j=1}^{200} n_{\text{rmax}}(D_j) \right)$$

for six models are calculated and depicted in Fig[5]A. Compared to Fig[4]B, the diversity of models with different generalization becomes more penetrating. The models with better generalization lie closer to top left corner of the figure than the models with poor generalization.

To further specify the effectiveness of our quantitative metric of generalization, we use some quantities, Gap [11] and Ratio, derived from training accuracy $\text{acc}_{\text{train}}$ and testing accuracy $\text{acc}_{\text{test}}$ of network models as comparison indices.

$$\text{Gap} = \text{acc}_{\text{train}} - \text{acc}_{\text{test}}$$

$$\text{Ratio} = \frac{\text{acc}_{\text{test}}}{\text{acc}_{\text{train}}}$$

As known, there is no widely recognized quantitative definitions for generalization of CNNs. The quantities we choose have no guarantee to be the most appropriate. But we think that they could represent the ability of network generalization to a large extent. So it is reasonable to consider Gap and Ratio as the yardstick of our metric. Fig[5]B takes bar chart to show the values of averaged generalized rate. Without regard to the normalization operation, very strong trend of linear correlation could be seen easily. This discovery naturally leads us to make use of linear regression to analyzing the relation between Gap (Ratio) and $\zeta$ for given CNN models so as to evaluate our metric.

Fig. [6]A visualizes the linear relation between $\zeta$ and Gap for six models. Surprisingly, $\zeta$ and Gap present an extremely perfect linear correlation, where the Pearson correlation coefficient reaches remarkably $-0.9956$. This gives it very strong support that $\zeta$ is really an effective metric of the generalization of CNN models. Then we use Ratio instead of Gap as the reference for generalization metric $\zeta$. Results in Fig. [6]B show even a bit higher degree of linear correlation between Ratio and $\zeta$ with Pearson correlation coefficient 0.9966. All the empirical evidence above clearly shows the feasibility of our approach to quantifying generalization of CNNs.
generalization of CNNs.

Maybe the climax of this paper is the next result. We try to predict the accuracy of CNN models on testing data only using training data. It seems a crazy idea at first glance. But we believe that the mysteries of generalization of CNN certainly will be revealed and our attempt is worthwhile. Using Eq(10), we construct a curve fitting between training accuracy and testing accuracy. The result in Fig.6C is amazing! The curve we obtained is nearly a straight line! Unexpectedly, Pearson correlation coefficient is even up to 0.9644. This is to say, the parameter $\alpha, \gamma$ and $\beta$ are determined readily. It is quite possible to predict the performance of CNN on testing data in the training process using our predictor Eq.(10). Further validation and accuracy assessments of our prediction model will be the following work.

4. Conclusion

We empirically analyze and quantify the generalization of CNNs from the perspective of individual units. First, role share is proposed to to identify the generalization status of individual units, where each share states a distinct contribution to the generalization performance. Also, we discuss to give a complete description of the typical assignments of role share for a well-generalized CNN. Second, utilizing only the training set, we propose a novel metric, called generalized rate, for quantifying the generalization ability of CNNs. Experiments demonstrate that the accuracy estimated by our metric exhibits a remarkable correlation with the true testing accuracy. This provides chances to measure the generalization in a more reliable manner comparing to conventional performance investigations on artificial testing sets with limited size.

References

[1] Campbell Adam and Wu Annie S. Multi-agent role allocation: issues, approaches, and multiple perspectives. Autonomous agents and multi-agent systems, 22(2):317–355, 2011.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7):e0130140, 2015.

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6541–6549, 2017.

[4] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. Proceedings of the National Academy of Sciences, 117(48):30071–30078, 2020.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[6] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In International Conference on Machine Learning, pages 1019–1028. PMLR, 2017.

[7] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4829–4837, 2016.

[8] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. arXiv preprint arXiv:1703.11008, 2017.

[9] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In Proceedings of the IEEE International Conference on Computer Vision, pages 3429–3437, 2017.

[10] Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. Proceedings of the National Academy of Sciences, 112(44):13455–13460, 2015.

[11] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In International Conference on Machine Learning, pages 1225–1234. PMLR, 2016.

[12] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836, 2016.

[13] Jingling Li, Yanchoo Sun, Jiahao Su, Taiji Suzuki, and Furong Huang. Understanding generalization in deep learning via tensor methods. In International Conference on Artificial Intelligence and Statistics, pages 504–515. PMLR, 2020.

[14] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5188–5196, 2015.

[15] Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In International Conference on Learning Representations, 2018.
[16] Antonio Murciano, Jose del R Millan, and Javier Zamora. Specialization in multi-agent systems through learning. *Biological cybernetics*, 76(5):375–382, 1997.

[17] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

[18] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29:3387–3395, 2016.

[19] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[20] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8684–8694, 2020.

[21] Lei Wu, Zhanxing Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

[22] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[23] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[24] Yang Zhao and Hao Zhang. Quantitative effectiveness assessment and role categorization of individual units in convolutional neural networks. *arXiv preprint arXiv:2103.09716*, 2021.

[25] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.