Role Taxonomy of Units in Deep Neural Networks

Yang Zhao **Hao Zhang** **Xiuyuan Hu**
Department of Electronic Engineering
Tsinghua University
{zhao-yang, haozhang}@tsinghua.edu.cn, huxy22@mails.tsinghua.edu.cn

Editors: Marco Fumero, Emanuele Rodolà, Clementine Domine, Francesco Locatello, Gintare Karolina Dziugaite, Mathilde Caron

Abstract

Identifying the role of network units in deep neural networks (DNNs) is critical in many aspects including giving understandings on the mechanisms of DNNs and building basic connections between deep learning and neuroscience. However, there remains unclear on which roles the units in DNNs with different generalization ability could present. To this end, we give role taxonomy of units in DNNs via introducing the retrieval-of-function test, where units are categorized into four types in terms of their functional preference on separately the training set and testing set. We show that ratios of the four categories are highly associated with the generalization ability of DNNs from two distinct perspectives, based on which we give signs of DNNs with well generalization.

1 Introduction

Deep neural networks (DNNs) are computing system modeled in an artificial way mimicking the biological neural networks, which have shown extraordinary ability in tackling real-world problems such as vision tasks [7, 16], natural language processing [5, 18] and so on. Being the primary element in both the biological and artificial neural networks, units are activated to perceive the underlying features for a given task. Similar to the biological neural networks, units in DNNs could learn to present strong specialization and high function preference to specific subtasks in the task.

However, for each basic unit, its content is quite simple and abstract. It seems impossible to decode the meaning of units from a theoretical view. So research is mostly carried out from discussing what the functionality may present by each unit [20, 6, 2, 3]. Generally, the functionality of a unit describes its role in the neural network at task level, and is formed by the natural selection of neural unit groups during task learning. At the same time, each unit has different functional characteristics. When faced with different tasks, due to the distributed working mode in the connectionist neural network models, units will respond to the task according to their own functional characteristics, and thus play distinct roles [9, 4].

In this paper, we are going to give a kind of role taxonomy of network units in DNNs. Before the taxonomy, we introduce a useful tool, called retrieval-of-function test, for identifying the group of units that could present the highest contribution on the given subtask when being activated jointly. In general, this group of active units may perform differently on the training set and the testing set. Based on this fact, we find that the units at a layer could be divided into four categories in terms of their function on the two sets: core units, overfitted units, generalizing units and confusing units. By illustrating with network models with different generalization ability, we show that the ratios of the four types of units could be closely associated with the model generalization. Finally, we give further

Proceedings of the I edition of the Workshop on Unifying Representations in Neural Models (UniReps 2023).

discussions from the perspective of role shares of individual units to show the different behaviors they may present in different network models.

2 Related works

Focusing on the investigation of network units, one line of research is to indicate the working status of units to the given task with certain attribute. Typical attributes include L1-norm of units [11] and the Average Percentage of Zeros (APoZ) of units when using Relu activation function [8]. In addition, [13] introduce the class selectivity from neuroscience to investigate the selectivity over classes for a specific unit, on the basis of calculating the mean feature maps. [21] propose a kind of topological entropy for indicating the chaotic degree of feature pattern for individual units. Studying on the indication of the unit working status could tell quantitative understanding of the contribution of a unit to a given task. But, it is still unclear what role the units could play in this task.

Another related research trace is to find and visualize the elements in the input that are important to the units. In this way, the critical element found in the input could be considered as the features that the unit could perceive, which somehow unveils the functionality of a unit. Typical works [20, 23, 12, 17] search for the image patches that maximize the score of a given unit. Alternatively, instead of searching, [15, 6] generate related images considered the most important for a unit. Bau et al. measure the degree of conceptual alignment between units and human-visual concepts (Mountain, River, etc.) predicted by another vision segmentation model in their works [2, 22, 3]. However, [19] argue that functionalities of units in CNNs may be beyond the recognition to human vision.

3 Identifying active network units via retrieval-of-function test

In neuroscience, to identify the function of units, it is critical to locate the collaborative unit group that are highly active for given a specific task, which could be detected via the functional magnetic resonance imaging. Correspondingly, consider tackling a typical classification task with DNNs, and our investigation begins with identifying this group of units in DNNs that are highly active to a specific class in this task.

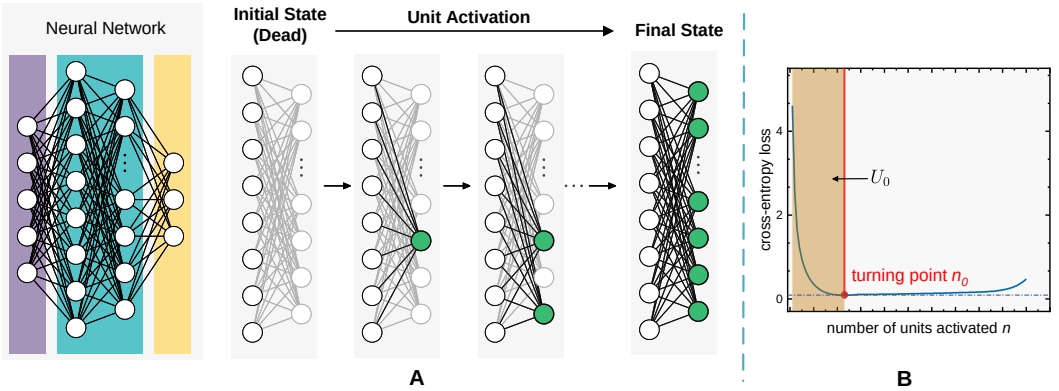

Figure 1: Illustration of retrieval-of-function test (A) and its ROF curve in regards to the cross-entropy loss (B).

Retrieval-of-function Test To identify these highly active units, we will employ the retrieval-of-function (ROF) test, as illustrated in Fig.1A. The core idea of ROF test is to identify the group of units that could recover the full perception of tasks when the networks stay in the "dead" state.

Basically, the ROF test could be summarized into two steps:

- Firstly, deactivate all the units within the layer to force the DNN to stay at the "dead" state, where the network gives no response on any input.
- Secondly, iteratively activate the units according to an ordered list $\boldsymbol{U}_l = \{\mathcal{U}_i\}_{i=0}^{N}$ individually. Typically, the ordered list could be ranked by quantities indicating the activation strength. Here, L1-norm will be used as the target quantity to give a descending rank of

units in the ROF test. In this way, units are activated sequentially from $\mathcal{U}_0$ to $\mathcal{U}_N$ in the initial dead network.

During the process of activation, the evolution of performance with respect to the number of units activated (notated as n) could be recorded as the characterization to indicate the growing effect on network performance when activating the unit. We would call this recorded curve the ROF curve and notate it as $\boldsymbol{E}(n)$. Fig.1B shows a typical evolution of $\boldsymbol{E}(n)$, where the cross-entropy loss is applied for recoding the performance of the DNN.

Turning point Before performing unit activation, as deactivating all the units in the layer, the DNN is unable to perceive any valid feature. With the ordered list in a descending rank, units with strong activation would be activated at the beginning of the evolution process. The performance would experience a continuously rapid increase since the neural network gradually recover its function on perceiving the feature in this class. Notably, the cross-entropy loss would reach at the minimum after activating only a small part of units, after which it would begin dropping steadily until all the units have been activated. We mark the minimum point as the turning point of $\boldsymbol{E}(n)$,

$$n_0 = \arg\min_{n} \ \boldsymbol{E}(n) \tag{1}$$

And correspondingly, the group of units that have been activated could be identified as follows,

$$\boldsymbol{U}_0 = \{\mathcal{U}_i\}_{i=0}^{n_0} \tag{2}$$

It should be pointed out that without loss of generality, $\{\mathcal{U}_i\}_{i=0}^{N}$ denotes the ordered list ranked by the descending order of the activation strength. $\boldsymbol{U}_0$ represents the minimum group of units being activated jointly that could stimulate the most performance on the given class. We call $\boldsymbol{U}_0$ the active unit set, and call its unit the highly active unit.

4 Role taxonomy of units in DNNs

In practice, to check the generalization ability, units would be trained on the training set $\mathcal{D}^{(train)}$ and tested for generalization on a held-out testing set $\mathcal{D}^{(test)}$. Notably, all the samples in this class should be drawn from the same distribution, sharing the same common features. For ideal networks, units should be able to perceive the common intrinsic features in the class. In this way, units could form substantial representations for unseen samples. This implies that units that are highly active on $\mathcal{D}^{(train)}$ are expected to perform in the same way on $\mathcal{D}^{(test)}$.

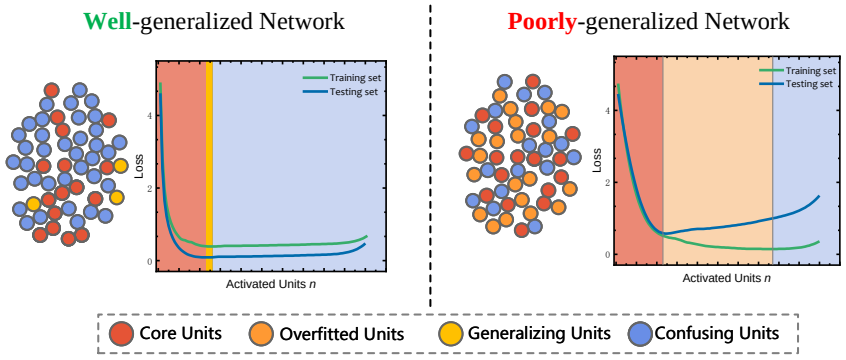

Figure 2: Illustration of role taxonomy of units in networks with different generalization abilities.

However, due to the excessive capability of the DNNs, networks may be over-fitted, presenting perfect performance on $\mathcal{D}^{(train)}$ while poor performance on $\mathcal{D}^{(test)}$. These poorly-generalized networks seem to simply memorize each labels of the training samples, and their units are not able to perceive any valid feature shared in the class. In addition, [10] finds that only some of the features could be reliably learned in networks with different initialization. So, highly active units could behave quite differently between the $\mathcal{D}^{(train)}$ and the $\mathcal{D}^{(test)}$. In this case, the units that are highly active on $\mathcal{D}^{(train)}$ may be able to perceive some common features to some extent. But meanwhile, they may

also be trained to perceive the features that **only** belong to $\mathcal{D}^{(train)}$ not $\mathcal{D}^{(test)}$. Since these features are not the intrinsic features in the class, the corresponding units could cause severe interference to the perception of $\mathcal{D}^{(test)}$.

Apparently, different units can present different roles on $\mathcal{D}^{(train)}$ and $\mathcal{D}^{(test)}$. Based on the functional discrepancy between $\mathcal{D}^{(train)}$ and $\mathcal{D}^{(test)}$, units can be categorized into 4 roles: core units, overfitted units, generalizing units and confusing units, as illustrated in Fig.2.

Core units Core units are the units that perform well on both $\mathcal{D}^{(train)}$ and $\mathcal{D}^{(test)}$,

$$\boldsymbol{U}_c = \boldsymbol{U}_0^{(train)} \cap \boldsymbol{U}_0^{(test)} \tag{3}$$

Typically, core units are the cornerstone units in the layer, which is colored light red in Fig. 2. They are usually sparse and could perceive the common features in this class for both the trained data and unseen data, so deactivating core units would cause the performance deteriorates seriously on both the two dataset. For well-generalized networks, activating only a small amount of most active units could be suffice to promote the network performance to the peak, which can be even higher than the original model. Then, the performance declines gradually along with the addition of other units, which clearly shows the difference on functionality between core units and others. For over-fitted networks, although activating the units other than core units could improve the performance on the training set, yet it could be found that these units could decrease the testing performance rapidly.

Overfitted units Overfitted units are the units being good at $\mathcal{D}^{(train)}$ but bad at $\mathcal{D}^{(test)}$,

$$\boldsymbol{U}_o = \boldsymbol{U}_0^{(train)} \cap (\boldsymbol{U}_l \setminus \boldsymbol{U}_0^{(test)}) \tag{4}$$

Overfitted units come after core units in the ordered list, colored orange in Fig. 2. The performance on training set keeps being improved, but its turning point appears on the test set obviously when overfitted units enter the network. This indicates that the functional status of these units has changed, from the previous positive effect on the network to a "false" positive effect. These units can only perceive features in the training samples. However, these features are not common features and do not exist in the test samples. Thus, overfitted units are harmful to the network because they offset the effect of core units. The existence of overfitted units implies the continuous rising of the training performance. It is a significant sign of overfitting. Remarkably, this provides a criterion to detect overfitting using only training sets.

Generalizing units Generalizing units are the units that play well on $\mathcal{D}^{(test)}$ but not on $\mathcal{D}^{(train)}$,

$$\boldsymbol{U}_g = \boldsymbol{U}_0^{(test)} \cap (\boldsymbol{U}_l \setminus \boldsymbol{U}_0^{(train)}) \tag{5}$$

Just like overfitted units, they come after core units in the important rank, colored light yellow in Fig. 2. Generalizing units are the separatrix of core units and confusing units. Normally, they are rare in the network because test sets are not involved in training. Emergence of generalizing units generally implies the model is well trained.

Confusing units Confusing units are the units important on neither $\mathcal{D}^{(test)}$ nor $\mathcal{D}^{(train)}$, but may be core units to other classes,

$$\boldsymbol{U}_f = (\boldsymbol{U}_l \setminus \boldsymbol{U}_0^{(train)}) \cap (\boldsymbol{U}_l \setminus \boldsymbol{U}_0^{(test)}) \tag{6}$$

They normally lie on the end part of the importance rank, colored light blue in Fig. 2. Confusing units are the products of performance balancing on diverse classes. Opposite to the overfitted units, if we observe the continuous slight decline in the training performance along with the addition of units, the model probably might not suffers from seriously overfitting.

Additionally, it is apparent that

$$\begin{aligned} \boldsymbol{U}_c \cap \boldsymbol{U}_o \cap \boldsymbol{U}_g \cap \boldsymbol{U}_f &= \emptyset \\ \boldsymbol{U}_c \cup \boldsymbol{U}_o \cup \boldsymbol{U}_g \cup \boldsymbol{U}_f &= \boldsymbol{U}_l \end{aligned} \tag{7}$$

which means that all units have one and only one role in the networks.

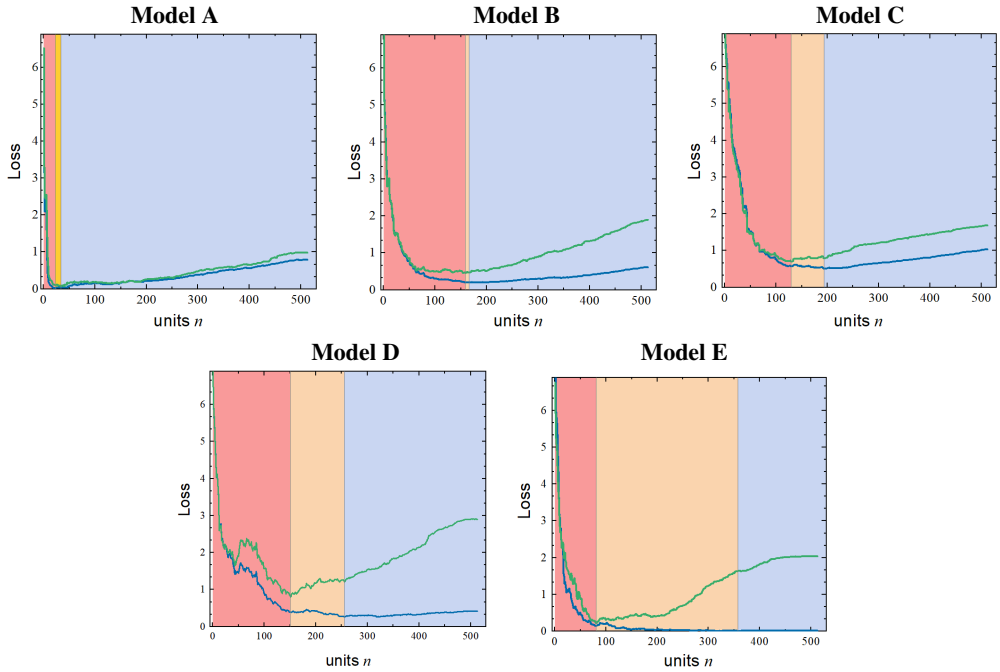

Figure 3: ROF curves and unit role taxonomy of 5 VGG16 network models on the same class

5 Experimental Analysis of Different Unit Roles

In our experiment, we train 5 VGG16 models on ImageNet dataset, each of which applies a separate hyper-parameters to reach at different generalization ability. Table 1 shows the performance of models ranking in the descending order of the generalization gap. In general, generalization gap denotes the difference between the training accuracy and the testing accuracy. It is usually used as the indicator of the generalization ability of a DNN.

Table 1: Classification accuracies for five different models.

MODELS	TRAINING ACC	TESTING ACC	GENERALIZATION GAP
MODEL A	0.732	0.657	0.075
MODEL B	0.730	0.600	0.130
MODEL C	0.818	0.543	0.275
MODEL D	0.828	0.444	0.384
MODEL E	0.978	0.374	0.604

We would first study the unit roles in the 5 different networks. Fig.3 shows the results of the same class for units at the last convolutional layer in the 5 VGG16 models. From the figure, we could found that for these well-generalized models, there are similar ROF curves on the training set and the test set. But for these poorly-generalized models, there are large differences in the ROF curves of the two data sets.At the same time, it can be seen that there are less than 30 core units in Model A, but it can give the best network model performance, indicating that these units in Model A have presented highly functional specialization which could perceive the intrinsic features in the class. As the performance of the network model continues to deteriorate, the number of overfitting units in the model will gradually increase, and the number of confusing units will continue to decrease. In addition, it can also be seen that the number of core units could keep within a certain range and will not increase as the model performance deteriorates. But generally, when there are few overfitting units, fewer core units mean that the units present better functional specialization.

Apparently, units with different roles in network models with different performances will show different proportions. Next, the research will be extended from a single class to all classes, and the proportion of role classification of units in different classes by these models will be summarized. In

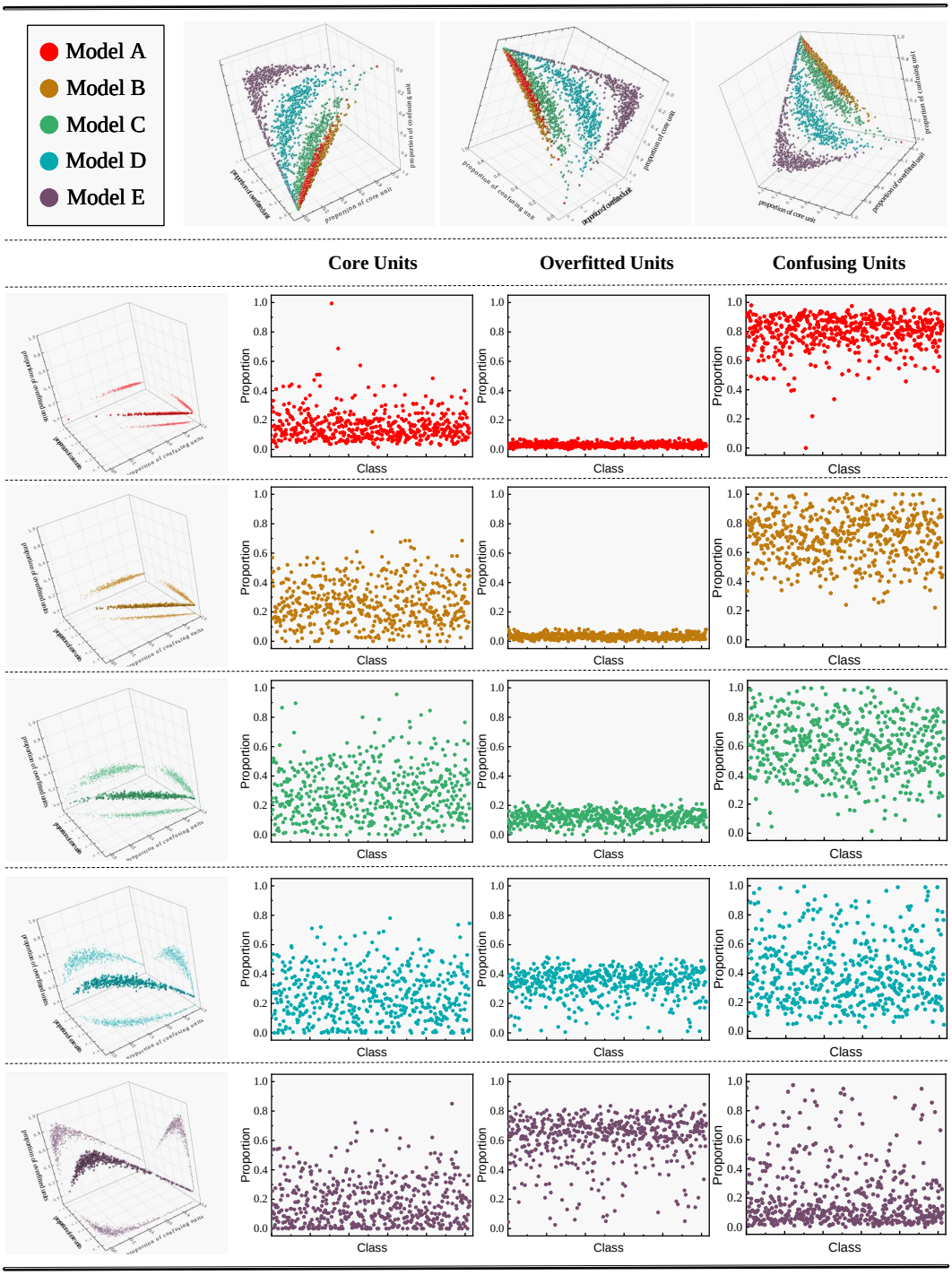

Figure 4: View of the proportion of units with different roles for all classes in ImageNet

Table 2: The mean and standard deviation of the unit role categorization proportions of all classes in the 5 VGG16 models.

Model	Core Units	Overfitted Units	Generalizing Units	Confusing Units
Model A	$16.1\% \pm 9\%$	$3.1\% \pm 1\%$	$1.7\% \pm 1\%$	$79.1\% \pm 10\%$
Model B	$25.9\% \pm 16\%$	$3.4\% \pm 1\%$	$1.8\% \pm 1\%$	$68.9\% \pm 17\%$
Model C	$28.6\% \pm 20\%$	$11.5\% \pm 4\%$	$0.5\% \pm 0\%$	$59.4\% \pm 22\%$
Model D	$25.4\% \pm 17\%$	$33.9\% \pm 11\%$	$0.2\% \pm 0\%$	$40.5\% \pm 24\%$
Model E	$17.7\% \pm 14\%$	$61.4\% \pm 18\%$	$0\% \pm 0\%$	$20.9\% \pm 20\%$

order to give a clear and intuitive comparison, the proportion of unit roles corresponding to each class can be expressed as a three-dimensional vector and drawn in a three-dimensional coordinate system, as shown in Figure 4. In the figure, the x axis, y axis and z axis represent the proportion of core units, overfitted units and confusing respectively. Due to the small proportion of generalizing units, generalizing units will be ignored in the plot. Moreover, each scatter point in the figure represents the proportion of 4 different role units in a class.

It could be clearly seen from the three-dimensional comparison chart that models with different performance show well separability under the same three-dimensional coordinate system. Comparing the individual model scatter points, we can see that the role proportions between Model A and Model B are quite similar. Both models show the typical nature of well-generalized models, i.e., they have fewer core units and overfitted units, and more confusing units. In the meanwhile, it can also be seen that although Model B has the same proportion of overfitted units as Model A, Model B has more core units than Model A, and the overall variance is larger. From the previous demonstration, when the model has only a small number of overfitted units, too many core units also mean poor specialization, which will damage the performance of the network model to a certain extent. For the latter three models (C, D and E), the corresponding proportion of core units remains relatively unchanged, but as the network performance worsens, the proportion of confusing units will become smaller and the proportion of overfitted units will gradually increase. Compared with models A and B, the latter three models show completely different proportions of unit role categorization. Obviously, the role categorization ratio of units is closely related to the performance of the network.

Further, the mean and standard deviation of the unit role categorization proportions of all classes for each model will be calculated. The results are shown in table2. As can be seen from the table, the pattern of role proportions presented is similar to Figure 4. Among them, the proportion of core units fluctuates within a range. For a well-generalized model, it should have a smaller proportion of core units and a lower standard deviation. However, the opposite is not necessarily true, such as model E. At the same time, for poorly-generalized models, they have more overfitting units. Similarly, if a network model has fewer overfitting units, it does not necessarily mean that it has a very good degree of functional specialization, such as model B. Finally, the proportion of confusing units in the model showed a strong correlation with network performance. It is worth noting that for a well-generalized model, it has a large number of units that have a negative effect on it. On the contrary, for a poorly-generalized model, there are only a small number of confusing units.

6 Role Share of Individual Units

6.1 Role Share of Individual Units

In this section, we would provide discussions of unit roles from another perspective. For each unit, it will play different roles when facing different classes. For example, a given unit will exist as a core unit when faced with some classes, but as a confusing unit when faced with other classes. Here, we call the proportion of roles played by neural units $\boldsymbol{U}_n$ in all classes the *Role Share*,

$$RS_n = \left(RS_n^{(c)}, RS_n^{(o)}, RS_n^{(g)}, RS_n^{(f)}\right) \tag{8}$$

where,

$$RS_n^{(\mathcal{R})} = \frac{1}{N}\sum_{k=1}^{N} \mathbb{I}_{U_n \in \boldsymbol{U}_{\mathcal{R}(k)}},\ \mathcal{R} = o, c, g, f \tag{9}$$

Table 3: The mean and standard deviation of the unit role categorization proportions of all classes in the 5 VGG16 models.

Model	Core Share	Overfitted Share	Generalizing Share	Confusing Share
Model A	$16.1\% \pm 10\%$	$3.1\% \pm 1\%$	$1.7\% \pm 1\%$	$79.1\% \pm 12\%$
Model B	$25.9\% \pm 15\%$	$3.4\% \pm 2\%$	$1.8\% \pm 1\%$	$68.9\% \pm 16\%$
Model C	$28.6\% \pm 18\%$	$11.5\% \pm 4\%$	$0.5\% \pm 0\%$	$59.4\% \pm 20\%$
Model D	$25.4\% \pm 17\%$	$33.9\% \pm 10\%$	$0.2\% \pm 0\%$	$40.5\% \pm 22\%$
Model E	$17.7\% \pm 15\%$	$61.4\% \pm 16\%$	$0\% \pm 0\%$	$20.9\% \pm 22\%$

Here, $\mathbb{I}(\cdot)$ denotes the indicator function.

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

$$RS_n^{(c)} + RS_n^{(o)} + RS_n^{(f)} + RS_n^{(g)} = 1 \tag{10}$$

The role share offers another dimension to delineate the specialization of network units. Cooperation and competition among units during training processes inevitably lead to the function preference of network units, which is the basis of network efficiency and generalization. The role of a unit and its share is closely associated. Table 3 presents the average role shares across all the neural units. It can be observed that for networks with different performance, each role share shows the same pattern as its role proportion in Table 2. Furthermore, the average value of each role's share is highly consistent with its corresponding role proportion, implying that the neural units maintains a balanced state during training. We would give discussions from the perspective of the role share in the following.

6.2 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 DNN with good generalization?

For core unit share set $\{RS_n^{(c)}\}_{n=1}^N$, 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 $\{RS_n^{(c)}\}_{n=1}^N$ 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 [14]. 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, $RS_n^{(c)}$ 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 $RS_n^{(c)}$ should also be avoided. Hence most elements of $\{RS_n^{(c)}\}_{n=1}^N$ should be limited in a small neighborhood of zero, the same as that of core units (Fig4). The model having too many units with overly large $RS_n^{(c)}$ probably could be detriment to generalization due to the bad specialization.

Since overfitted units are only effective on the training set, $RS_n^{(o)}$ should be as few as possible for each unit U_n. When a DNN is poorly generalized, large number of units will present high $RS_n^{(o)}$, 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 training accuracy on the classes regarding these units as either core units or overfitted units. In a word, the more units with high $RS_n^{(o)}$, the more severe decline on the training accuracy in a ROF test. This results in a more dramatically drop of the overall training accuracy. This is consistent with the observations in [13]: "training accuracy drops more rapidly for less generalized CNNs when performing cumulative ablation operation".

However, the statement in [13], "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. 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 $\{RS_n^{(f)}\}_{n=1}^N$, it should distribute around some large value close to 1, being contrary to that of core unit share set. 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 pre-trained 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-specified. 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 models.

7 Conclusion

In this paper, we have provided a kind of role taxonomy for units in DNNs with different generalization ability. Given a task, units that are highly active at a layer are firstly identified by performing the introduced retrieval-of-function test. We found that they could be categorized into four types in terms of their function on separately the training set and the testing set, which are core units, overfitted units, generalizing units and confusing units. We show that the DNNs with different generalization ability would have distinct ratios of the four types of units, from both the perspective of unit role and role share. We hope our work could be helpful for further how to train a good DNN, and also expect to give connections between biological neural networks and the artificial neural networks.

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] 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.

[3] 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.

[4] Christine A Denny, Mazen A Kheirbek, Eva L Alba, Kenji F Tanaka, Rebecca A Brachman, Kimberly B Laughman, Nicole K Tomm, Gergely F Turi, Attila Losonczy, and René Hen. Hippocampal memory traces are differentially modulated by experience, time, and adult neurogenesis. *Neuron*, 83(1):189–201, 2014.

[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] 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.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[8] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

[9] Michael R Hunsaker and Raymond P Kesner. The operation of pattern separation and pattern completion processes associated with different attributes or domains of memory. *Neuroscience & Biobehavioral Reviews*, 37(1):36–58, 2013.

[10] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John E. Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[11] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017.

[12] 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.

[13] Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In *Iternational Conference on Learning Representations*, 2018.

[14] Antonio Murciano, Jose del R Millan, and Javier Zamora. Specialization in multi-agent systems through learning. *Biological cybernetics*, 76(5):375–382, 1997.

[15] 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.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

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

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[19] 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.

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

[21] Yang Zhao and Hao Zhang. Quantitative performance assessment of cnn units via topological entropy calculation. In *International Conference on Learning Representations*, 2021.

[22] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018.

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