Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230665

Understanding the class separation process of convolutional neural networks

Yiqi Yang[1,6], Xingyu Lian[2,7], Weihao Liu[3,8], Liyuan Shen[4,9], and Jiayang Lin[5,10]

[1]School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China
[2]College of Letters and Science, University of Wisconsin-Madison, Madison, USA
[3]School of computer science, Auburn University, Auburn. USA
[4]College of Engineering, University of California, Santa Barbara, USA
[5]Private Boarding High School, Trinity-Pawling School, Pawling, USA

[6]yiqi_yang_21@163.com
[7]xlian6@wisc.edu
[8]wzl0059@auburn.edu
[9]lshen@ucsb.edu
[10]kevinlin4854@gmail.com

Abstract. Extensive research shows that deep neural networks (DNNs) capture better feature representations than the previous handcrafted feature engineering, which leads to a significant performance improvement. However, it raises the question of how does the neural network spontaneously learn the intermediate representation, during the training process, to correctly separate the test data set into different categories? In this paper, based on the existing research work, we continue to take a small step towards understanding the dynamics of convolutional neural networks (CNN). Specifically, we model the test data set as a relationship graph, and each intermediate layer representation learned by CNN will transform the relationship graph correspondingly. Through these continuous transformations, we investigate the class separation process of CNN from two perspectives: 1) show the evolution of the degree of correlation between samples through the histogram of similarity between vertices; 2) by visualizing and quantifying the degree of separation (i.e., modularity) of the important relationship subgraph, we can show the contribution of each convolution layer of CNN to class separation. In the preliminary experiment, it can be found that: 1) each convolution layer of CNN gradually reduced the similarity of most edges between the samples to a very low level, that is, the dense relationship graph gradually evolved into a sparse graph; 2) the remaining important relationship subgraph consist of edges with high similarity shows evident modularity; 3) both visualization and quantification results show that the modularity of intra-class subgraph increases when the layers go deeper. Moreover, the degradation and plateau in the modularity curve reveal the existence of redundant layers. In this paper, we reveal some more in-depth dynamics of CNN, and introduced another modularity, which is widely used as the theoretical guidance tool of layer pruning. It can save more model parameters without losing the accuracy of classification.

Keywords: interpretability, modularity, layer pruning, deep learning.

DOI: 10.54254/2755-2721/6/20230665

1. Introduction

Over the years, deep learning has proved to be very efficient in various tasks, such as computer vision and natural language understanding [1-4]. The remarkable achievements of neural networks in these tasks are widely believed to be mainly due to the powerful feature representation learned from data [5-6]. However, how does DNN achieves the final accurate classification through these intermediate representations? The research community knows little about the internal behavior, decision-making process, or the source of good performance of the neural network model. The lack of a clear explanation will definitely limit the development of neural network models to trial-and-error method, which is a major setback. In order to further study the internal mechanism of neural networks and design methods to improve the effectiveness of neural networks, researchers are exploring a variety of methods to explain neural networks. It is generally believed that if you want to open the DNN decision black box, you need to deeply understand some inherent characteristics of DNN feature representation. At the same time, it is helpful to better understand the intermediate representation learned by DNN and the DNN decision-making process, making the DNN prediction results more trustworthy. In addition, this knowledge can also be used as a theoretical tool to guide the design of DNN. Some researchers describe the characteristics of intermediate representation by observing the similarity between neural network layer representation and model [7-13]. Others directly visualize the feature representation of the hidden layer for intuitive understanding. They reveal that the feature representation of the shallow layer is relatively general, while the feature representation of the deep layer is more specific [14-18]. These studies have promoted the understanding of the internal representation of neural networks, but they are still limited, since they ignore the dynamics of DNN, or they can only understand the dynamics of DNN through qualitative visualization rather than quantitative research. Y. Lu et al. studied the internal dynamics of DNN from the perspective of community evolution, and provided a new perspective to observe the relationship between DNN intermediate representation and model decision making [19]. Although Lu et al. have taken a good first step in this direction, they only uncovered a few elementary understandings of the dynamics of DNN, and more internal relationships or characteristics need to be further explored [19].

Therefore, inspired by Y. Lu et al., we conducted a more in-depth study into the dynamics of the CNN interlayer from a slightly different perspective and gained a deeper understanding [19]. Specifically, we treat each sample of the test set as a vertex, and construct a sample relationship graph to observe the continuous effects or dynamic changes of CNN layer representations on the relationship graph. The weight between two vertices is expressed by the similarity of two vertices in the corresponding layers feature representation. In fact, they also represent a certain distance between two vertices. In principle, all vertices are connected to each other by edges, and the weight of edges are different. Then, we construct a complete relationship graph. Using the weight of edges and the community evolution of important relationship sub-graph, we can deeply observe how the sample relationship graph shifts under the effect of each convolution layer. The main contributions of this paper include the following:

1) On the whole data set, the similarity distribution of the representations of each layer is fully counted, and the continuous effect of the representations of each layer of CNN on the degree of correlation between vertices in the graph is observed. A stable trend is found: the representations of each layer of CNN reduce the weight of most (more than 85%) edges layer by layer;

2) The edges with higher weights are used to form an important relationship sub-graph. Then, the continuous effect of each layer representation of CNN on the important relationship sub-graph is analyzed from two aspects: on the one hand, a sub-graph visualization method is proposed to intuitively present the dynamic changes of the important relationship sub-graph; On the other hand, the modularity is redefined on the important relationship sub-graph to quantify the dynamics of the important relationship sub-graph. From two different perspectives, the same stable trend is found: with the

DOI: 10.54254/2755-2721/6/20230665

deepening of the convolution layer, the blocking feature of intra-class vertices is constantly strengthening, while the blocking feature between inter-class vertices is constantly weakening;

3) According to the definition of modularity proposed in this paper, several classical CNN modularity curves on CIFAR-10 are obtained [20]. Compared with the modularity curve of, the quantization curve obtained is more smooth, and more or longer plateau areas and peak areas can be found [19]. Furthermore, using it to guide layer pruning can achieve higher parameter saving rate on almost all models, which fully reflects a more reasonable definition. Therefore, the experiment proves the effectiveness of the proposed method.

In the following section 2, we introduced the related work. In section 3, we described the construction method of our relationship diagram in detail, and gave a new definition of modularity. In section 4, we gave the experimental results, and the summary is in section 5.

2. Related works

Some research works have done a lot of in-depth research in understanding the dynamics of DNN. Hence, in this section, we would like to provide a brief survey of the literature related to our current work.

2.1. Research methods based on model feature analysis

Different DNNs are composed of neural network layers with different quantities and structures. The output of each layer is used as the input of the next layer. Therefore, the output of each layer is represented by a feature. These intermediate representations must hide more dynamic factors of DNN. The research work on understanding feature representation can be divided into two categories. One is to quantitatively calculate the similarity between feature representations at layer level or model level [7-13]. For example, Kornblith et al. introduced central kernel alignment (CKA) to measure the relationship between intermediate representations [7]. Feng et al. proposed a result oriented measurement method, called transfer difference, which uses the performance of downstream tasks to quantify the difference between the two representations [13]. While Tang et al. used both eigenvectors and gradients to define differences between features [21]. The other method attempts to understand the feature representation by explaining the semantics expressed by the feature [14-18]. Wang et al. and Zeiler et al. found the hierarchy of neural network features: the basic and general features are extracted at the shallow level, while the deep level is more specific and comprehensive [15,17].

2.2. Empirical research methods for community detection

Some previous empirical studies found that neural networks showed modularity and community, so more researchers began empirical research based on community discovery and analysis. Some research work attempts to study the modularity and community structure of neural networks at the neuron level or sub-network level [22-29]. In other research work, the architecture with modular characteristics is trained through parameter isolation or regularization during training, or the degree of modularity is improved, so as to develop a neural network with better modularity [30-33]. Literature does not focus on the modularity of the neural network itself and the discovery and analysis of the community structure, but discusses the evolution of the community at the feature representation level, providing a new perspective for describing the dynamic characteristics of DNN [19]. However, the exploration of is still preliminary, which is mainly shown in the following aspects: First, its research on the dynamics of neural networks is conducted on a randomly selected sub-graph, and there is no convincing basis to prove that the sub-graph is representative for the whole [19]. Therefore, there is some doubt about the generality of the conclusions drawn; Secondly, its definition of modularity is coarse-grained. It does not distinguish the edges within and between classes. In fact, we found in experiments that the modularity within and between classes is not consistent. In other words, we need to consider these two factors when defining modularity. Therefore, its modularity curve jitters greatly.

Therefore, inspired by, this paper conducts further in-depth research in this direction [19]. We modeled the test data set as a graph. First, we conducted a large number of comprehensive evaluations on CIFAR-10 and found an important feature: most (more than 85%) of the edges in the graph are similar,

DOI: 10.54254/2755-2721/6/20230665

or the correlation is very low. Therefore, on this basis, this paper extracts the first N edges with the closest relationship to construct an important relational sub-graph, and studies the dynamic characteristics of modularity on its basis. Secondly, through the visualization and quantification of the relationship graph, this paper more comprehensively observes the dynamic characteristics of the CNN middle layer representation, and draws more reliable conclusions. Moreover, a large number of experiments show that the more fine-grained definition of modularity given in this paper can find more redundant layers in the model, and can achieve higher parameter savings when used for layer pruning.

3. Method

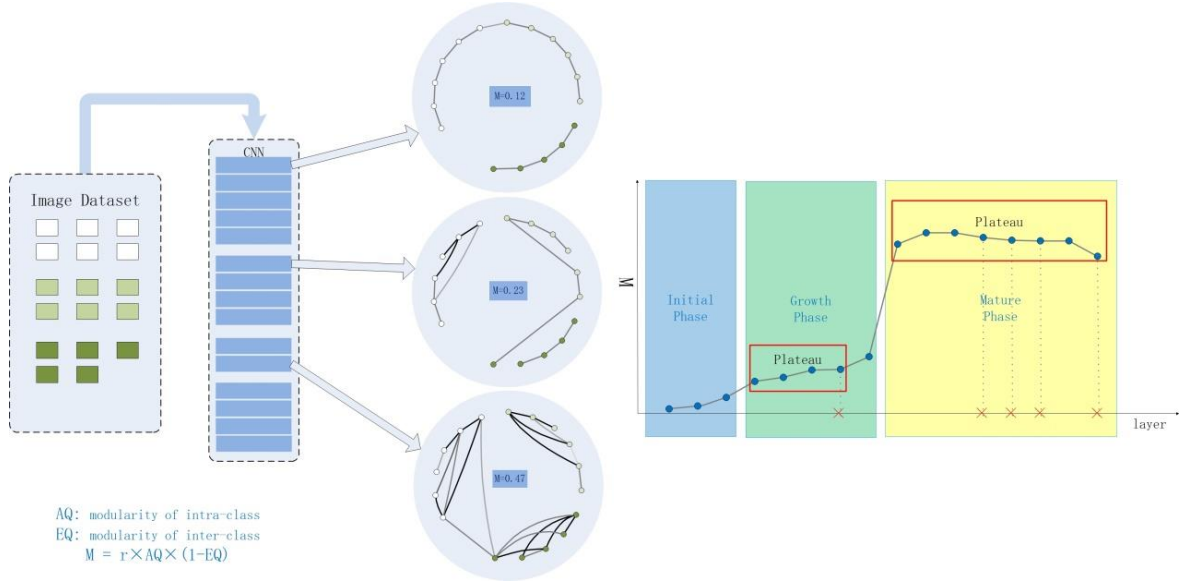

Figure 1. Pipeline for the dynamic graph construction and the application scenarios of the modularity metric.

3.1. Modularity of CNN

Graph $G = (V, E)$ is composed of vertex set and edge set, where V is vertex set and E is edge set. Clustering coefficient γ_v of vertex v in graph G describes the degree of proximity between the vertex and its neighbors [34]:

$$\gamma_v = \frac{|E(\Gamma_v)|}{\binom{k_v}{2}} \tag{1}$$

Among them, Γ_v represents the neighborhood of vertex v, that is, the subgraph formed by all connected vertices (excluding v), $E(\Gamma_v)$ and k_v represents the edge set and vertex number of Γ_v respectively, $\binom{k_v}{2}$ representation the number of all possible edges in Γ_v. In Γ_v there are most $\binom{k_v}{2}$ edges, and γ_v represents the net ratio of edges really existed in Γ_v. Similarly, in social networks, γ_v refers to the degree of familiarity between acquaintances, and this value measures the closeness of members' communication networks. γ_v can be regarded as the probability that any two vertices in Γ_v are connected. The clustering coefficient of graph G is defined as γ_v, the mean or median of γ_v. This definition assumes that all vertices have the same weight. Another clustering coefficient is defined as:

$$\gamma = \frac{\sum_v |\Gamma_v|}{\sum_v \binom{k_v}{2}} \tag{2}$$

in which, vertices with higher degrees have more weight than vertices with lower degrees.

Some previous studies have shown that the convolutional neural network can be viewed as a graph with neurons as vertices, showing a block characteristic [22-27]. A training or test sample corresponds

DOI: 10.54254/2755-2721/6/20230665

to a vertex in the graph, and the edges between vertices are defined by the similarity between samples. Convolutional neural network is composed of each convolution layer. The processing results of each convolution layer can build a graph, and the clustering coefficient of vertices in the graph can reflect the block characteristics in the graph. The clustering coefficient in this paper is determined by the sum of similarities between the current vertex and other vertices.

The convolution graph $\mathrm{CG} = (\mathrm{V}, \mathrm{E})$ formed by all samples, where V and E is the vertex and edge sets respectively. The vertex in V is a mapping vector of the sample. Let $e(i, j) \in \mathrm{E}$ represents the edge between vertices i and j, and the similarity between the two sample vectors $\mathrm{sim_{ij}}$ represents the weight of the edge. Clustering coefficient $\mathrm{s_i}$ of arbitrary vertex i is defined as the sum of the weights of all edges connected to it, namely $\mathrm{s_i} = \sum_{\mathrm{j}} \mathrm{sim_{ij}}$, and $\mathrm{S} = \sum_{\mathrm{i}} \mathrm{s_i}$. We refer to the connection between two samples belonging to the same category as the correctly connected edge, the set of correctly connected edges as *AS*, the connection between two samples belonging to different categories as the wrongly connected edge, and the set of wrongly connected edges as *ES*. The sum of weights of all correctly connected edges is the correct-*modularity*:

$$\mathrm{AQ} = \frac{1}{2\mathrm{S}} \sum_{(\mathrm{i,j}) \in \mathrm{AS}} (\mathrm{sim_{ij}} - \frac{\mathrm{s_i s_j}}{2\mathrm{S}}) \tag{3}$$

The sum of the weights of all incorrectly connected edges is the *wrong-modularity*:

$$\mathrm{EQ} = \frac{1}{2\mathrm{S}} \sum_{(\mathrm{i,j}) \in \mathrm{ES}} (\mathrm{sim_{ij}} - \frac{\mathrm{s_i s_j}}{2\mathrm{S}}) \tag{4}$$

The sum of the weights of all wrongly connected edges is the probability of wrong modularity and correct connection, which is defined as $\mathrm{r} = \frac{|\mathrm{AS}|}{|\mathrm{E}|}$, finally, the modularity of convolution graph is defined as:

$$M = \mathrm{r} \times \mathrm{AQ} \times (1 - \mathrm{EQ}) \tag{5}$$

3.2. Dynamic graph construction

Our proposed dynamic graph construction framework to understand the dynamics of a given DNN is visually summarized in Fig 1.

In general, a convolutional neural network CNN is composed of K convolutional layers, $\mathrm{CNN} = \{\mathrm{L_1}, \mathrm{L_2}, \ldots, \mathrm{L_K}\}$. The output of the previous layer is used as the input of the next layer. We regard a convolution layer L_i as a mapping $\mathrm{f_i}: \mathrm{C}^{\mathrm{N}\times\mathrm{w}\times\mathrm{h}\times\mathrm{m}} \rightarrow \mathrm{C}^{\mathrm{N}\times\mathrm{w}\prime\times\mathrm{h}\prime\times\mathrm{m}\prime}$, the input multi-channel feature is mapped to the output multi-channel feature. A CNN is a composite mapping $\mathrm{f_1} \circ \mathrm{f_2} \circ \ldots \circ \mathrm{f_m}$. Convert input X into a series of intermediate representations until the final output. CNN training process is the process of learning this composite mapping. The three sub layers *Conv-BatchNormal-Relu* of ResNet are regarded as one convolution layer, and so as to in VGG-bn, whose convolution layer also contains such three sub layers.

A sample taken from the data set (training data set or test data set) is put into the CNN, and we will get a group of feature maps in layer Li , then perform flatten operation on them to obtain a two-dimensional matrix $\mathrm{SIM^l_{N,w\times h\times m}} = (\mathrm{h_1}, \mathrm{h_2}, \ldots, \mathrm{h_N})^{\mathrm{T}}$. Therefore, a sample $\mathrm{X_j}$ be mapped into a vector $\mathrm{h^i_j}$ by convolution mapping $\mathrm{f_i}$ of i^{th} layer, $\mathrm{h^i_j}$ corresponds to the vertex $\mathrm{v^i_j}$ in convolution graph. In the convolution graph $\mathrm{CG_i}$, the weight of the edge between two vertices $\mathrm{v^i_s}$ and $\mathrm{v^i_t}$, is defined by the cosine similarity between their corresponding two vectors $\mathrm{h^i_s}$ and $\mathrm{h^i_t}$ as follow:

$$\mathrm{sim^i_{s,t}} = \frac{\mathrm{h^i_s} \cdot \mathrm{h^{i\,T}_t}}{||\mathrm{h^i_s}|| * ||\mathrm{s^i_t}||} \tag{6}$$

In this way, the vector corresponding to all samples in the dataset constitutes convolutional graph $\mathrm{CG_i}$, in which any two vertices are connected by edges, and the weight of edges is defined by similarity. Edge $\mathrm{e(s,t)}$ in $\mathrm{CG_i}$ is called *correct edge* if two vertices of which belong to the same category, an if the two vertices of which belong to different categories, it are called *wrong edge*.

DOI: 10.54254/2755-2721/6/20230665

Thus, a convolution neural network CNN is composed of K convolution layers, $\mathrm{CNN} = \{\mathrm{L}_1, \mathrm{L}_2, \dots, \mathrm{L}_\mathrm{K}\}$, and the convolution dynamic graph $\mathrm{CG} = \{\mathrm{CG}_1, \mathrm{CG}_2, \dots, \mathrm{CG}_\mathrm{K}\}$ composed of K graphs is obtained on the dataset, where, CG_i is regarded as the state of convolution dynamic graph CG at the i^{th} moment. The vertices of this dynamic graph are fixed, and the connection weight of edges between vertices is changed. A convolutional neural network constantly changes the connection weight between vertices through layer by layer convolution mapping, and at the same time, continuously improves the weight of *correct edges* and reduces the weight of *wrong edges*, so as to finally obtain high classification accuracy.

4. Results

Our goal is to intuitively understand the dynamics in well-optimized DNNs. Reflecting this, our experimental setup consists of a family of VGGs [35], ResNets[1] and MobileNet[36] trained on standard image classification datasets CIFAR-10[20]. Specifically, we leverage stochastic gradient descent algorithm with an initial learning rate of 0.01 to optimize the model. The batch size, weight decay, epoch and momentum are set to 256, 0.005, 150 and 0.9, respectively. All experiments are conducted on one NVIDIA RTX3060 GPU.

In this paper, VGG, ResNet and MobileNet, three classical convolutional neural networks, have been used in a large number of experiments on CIFAR-10. The super parameter settings in all experiments are consistent. This paper mainly studies the dynamic characteristics of convolutional neural network from three aspects:

1) The similarity between vectors obtained by mapping each convolution layer for all samples of the complete statistical data set, and observe the dynamic changes of the similarity distribution among these samples;

2) By visualizing the convolution map of each convolution layer, the dynamic characteristics of convolution neural network can be observed directly;

3) According to the segmentation degree of the convolution map defined by the formula (5) proposed in this paper, the segmentation characteristics of the convolution map of each layer are quantified, and the dynamic characteristics of the convolution map are quantitatively analyzed via the quantitative results. Finally, after obtaining the modularity curve of convolutional neural network, a preliminary application of it is shown: taking the peak point and platform area of the modularity curve as the important basis for layer pruning.

4.1. The change of similarity distribution of each layer shows a stable trend

In order to observe the influence of each convolution layer on the similarity, we made a complete statistics of 10000 test pictures on the CIFAR-10 test data set. After obtaining the similarity matrix of each convolution layer, divide the similarity in [0,1] into 10 sub ranges on average, account the number of similarities falling in these intervals, and finally obtain the similarity histogram of each convolution layer. The results are shown in figure2-5, and the similarity histograms of other models are given in Appendix 1 (figure 11-15). It can be seen from the figure: (1) Similarity is mainly distributed in a few intervals; (2) After the treatment of each convolution layer, the section with the largest distribution gradually moves to the left; (3) After the last layer of processing, most of the similarities are very small, that is, (in other words), in the last convolution graph CG, most of the edge weights become weak. Therefore, we only select the first $|\mathrm{E}|$ (=40000) edges with the greatest similarity to construct the convolution graph in the experiments in the following section 4.2 and 4.3.

4.2. The convolution dynamic graph shows a stable trend

In ring convolution graph at each time, vertices belonging to the same category are arranged consecutively. The correct connected edges between vertices of the same category are represented by the same color, and the wrong connected edges are represented by red color. In order to visualize the dynamic graph, firstly, N=4000 samples are randomly selected as the vertices of the circular graph of the convolution layer, and the first $|\mathrm{E}|$=40000 edges with the largest similarity between them are taken

DOI: 10.54254/2755-2721/6/20230665

as the edges of the circular graph of the convolution layer. A convolutional neural network has K convolution layers, corresponding to K ring convolution graphs. Each ring convolution graph is the state of the convolutional neural network dynamic graph at a time. This paper has done a lot of experiments on ResNet18, ResNet34, ResNet50, ResNet101, resnet56, VGG11-bn, VGG16-bn, VGG19-bn and MobileNet. The figure 6-9 only show the dynamic diagrams of ResNet34, VGG19-bn, resnet56 and MobileNetV2. The dynamic diagrams of other models are given in Appendix 2 (figure 16-20).

From the dynamic graphs of all models, it can be seen that: (1) with the deepening of the convolution layer, the number of wrong edges decreases and the number of correct edges increases; (2) The whole process is roughly divided into three stages, from the initial stage (many wrong edges) to the development stage (fewer wrong edges), and finally to the mature stage (few wrong edges); (3) The effect of some layers is more obvious, while that of other layers is weaker.

4.3. The modularity curve of convolution dynamic graph shows a stable trend

The experiment in this paper is carried out on CIFAR-10. A comprehensive statistical calculation of the modularity of the dynamic diagrams of ResNet18, ResNet34, ResNet50, ResNet101, resnet56, VGG11-bn, VGG16-bn, VGG19-bn and MobileNet was carried out, and finally the following modularity curve was obtained. Without losing generality, the calculation of modularity is carried out on the dynamic diagram with N=4000 and |E|=40000.

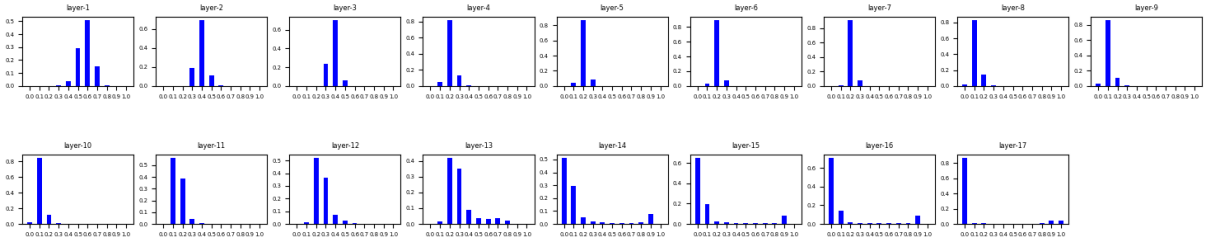

Figure 2. Similarity distribution of ResNet34.

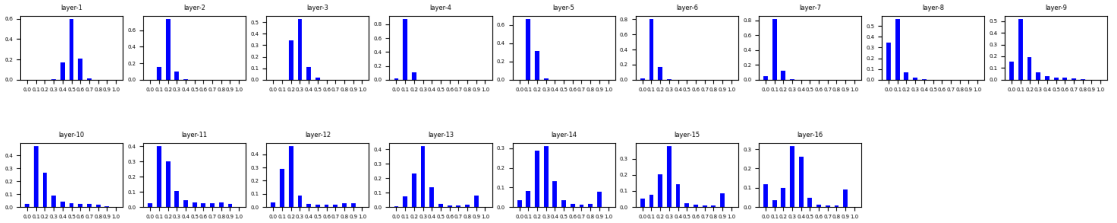

Figure 3. Similarity distribution of VGG19-bn.

DOI: 10.54254/2755-2721/6/20230665

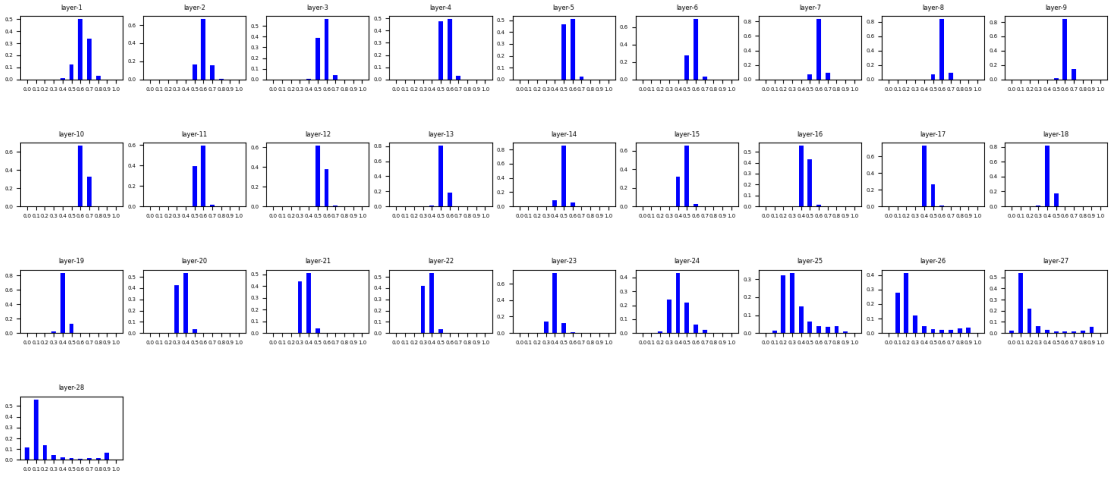

Figure 4. Similarity distribution of resnet56.

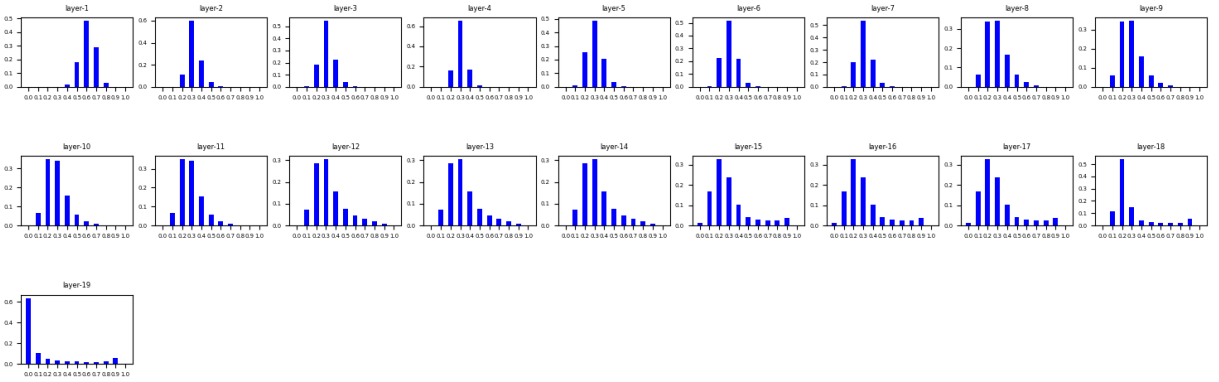

Figure 5. Similarity distribution of MobileNetV2.

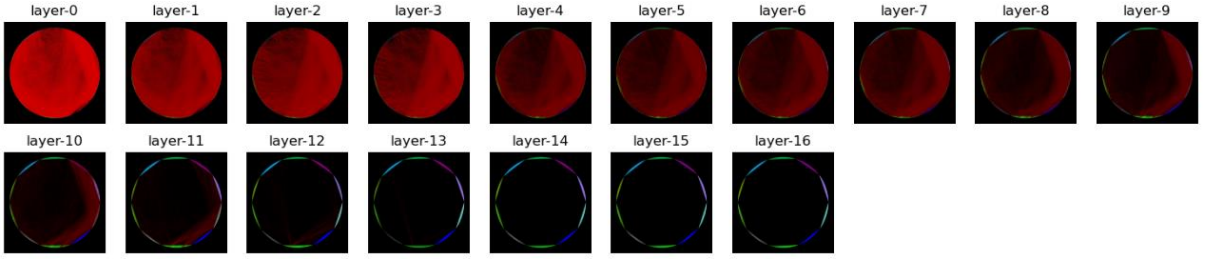

Figure 6. Dynamic graph of ResNet34.

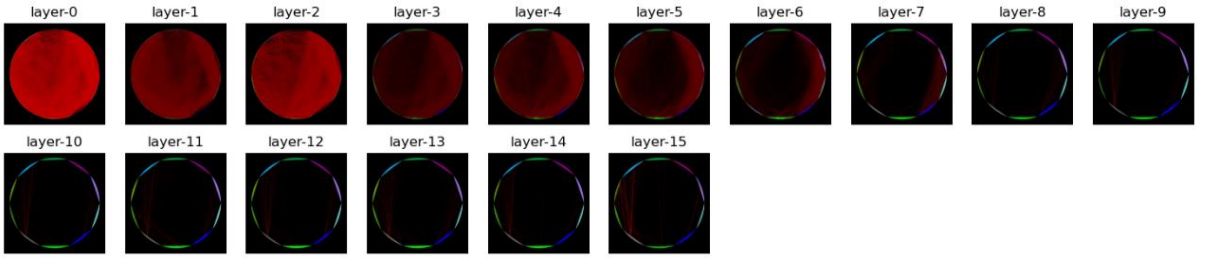

Figure 7. Dynamic graph of VGG19–bn.

DOI: 10.54254/2755-2721/6/20230665

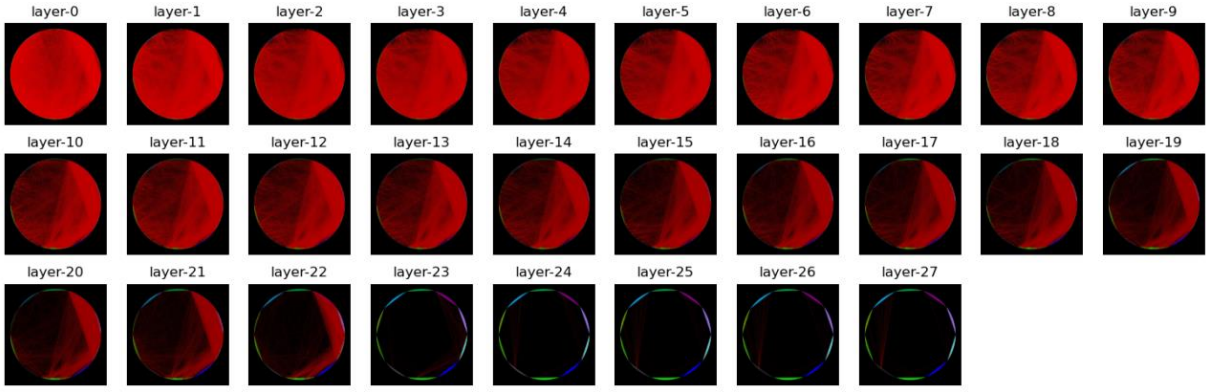

Figure 8. Dynamic graph of resnet56.

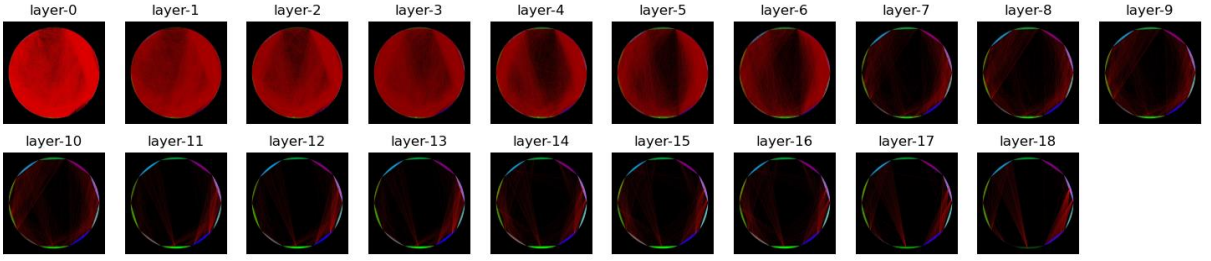

Figure 9. Dynamic graph of MobileNetV2.

The modularity measurement method proposed in this paper has been compared with the reference [19], and the experimental results are shown in the figure 10 below. From the experimental results, it can be seen that: (1) The rule is consistent with that found in the dynamic chart in the previous section, and the modularity shows a nonlinear upward trend, which can be divided into the beginning, development and maturity stages; (2) The modularity curve proposed in this paper is more smooth and more conducive to finding the curve plateau area; (3) From the experiments in Section 4.4, we can see that the modularity measure proposed in this paper is more conducive to layer pruning, which shows that the modularity measure proposed in this paper is more effective.

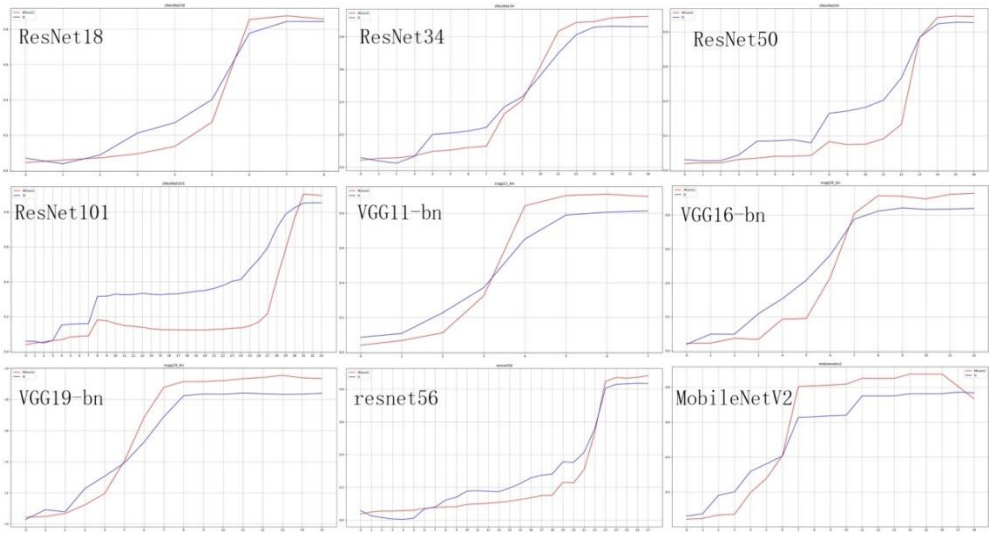

Figure 10. The red curve is our modularity curve, and the blue one is another modularity curve in [19].

DOI: 10.54254/2755-2721/6/20230665

4.4. Application on layer pruning

Table 1. Comparison results of layer pruning performance.

model	Accuracy (%)	pruning with modularity curve	accuracy of new model (%)	ratio of saving parameters (%)+△
ResNet18	94.57	our	**95.03**	**42.25+**42.25
		[19]	-	0
ResNet34	94.89	our	95.09	**55.46**+33.28
		[19]	**95.10**	22.18
ResNet50	95.29	our	95.30	**47.45**+23.73
		[19]	**95.31**	23.72
VGG11-bn	92.05	our	**92.59**	**48.40**+48.40
		[19]	-	0
VGG16-bn	93.5	our	**94.15**	**50.30**+19.34
		[19]	93.80	30.96
VGG19-bn	93.64	our	**93.83**	**57.40**+34.44
		[19]	93.81	22.96
resnet56	93.27	our	93.26	**60.76**+19.52
		[19]	**93.30**	41.24
MobileNet V2	88.11	our	**88.43**	**70.10**+22.63
		[19]	88.08	47.47

Criterion I: When the modularity curve reaches the maximum value or a plateau period occurs or starts to decline, the corresponding convolution layer can be cut off.

Similar to the method in the literature, the modularity curve can guide the layer pruning of the model [19]. In order to compare the modular definition method proposed in this paper with the reference, we conducted layer pruning experiments on several classical convolutional neural network models, such as ResNet18, ResNet34, ResNet50, VGG11-bn, VGG16-bn, VGG19-bn, resnet56 and MobileNetV2, according to the Criterion I [19]. In experiments the super parameters during training are all consistent. The experimental results are shown in table1, from which it can be found that using the same pruning criteria, layer pruning based on the modularity curve proposed in this paper saves more parameters, that is, greater benefits are obtained. Therefore, the definition of modularity proposed in this paper is more scientific or more effective. This is mainly because the modularity curve defined in the literature is not smooth enough, the plateau area is less or shorter, and the peak area on some models is shorter [19]. Therefore, fewer redundant layers can be found when pruning layers according to Criterion I.

5. Conclusions

Deep neural network (DNN) has achieved high accuracy in many classification tasks by learning feature representation automatically. However, it is not clear how it correctly separates test data sets into different categories. Based on the existing research work, we model the test data set as a relational graph to reveal the class separation process of CNN from two perspectives: 1) observe the evolution of the degree of correlation between samples through the statistical similarity distribution between vertices; 2) By visualizing and quantifying the degree of separation (i.e., modularity) of important relationship subgraph, we can observe the contribution of each convolution layer of CNN to class separation. We have taken a small step towards understanding the dynamic characteristics of CNN. First of all, we find that the convolution layers of CNN gradually reduce the similarity of most edges between the absolute samples to a very low level, that is, the dense graph gradually evolves into a sparse graph; Secondly, the remaining few important relationship subgraph composed of highly similar edges show obvious modularity, and both visualization and quantification results show the same trend: the modularity of

DOI: 10.54254/2755-2721/6/20230665

intra–class subgraph increases with the deepening of layers. Moreover, the degradation and platform in the block degree curve reveal the existence of redundant layers. Experiments show that this paper reveals some new dynamics of CNN, and according to the new definition of modularity given in this paper, the obtained modularity curve is smoother and more effective as a theoretical guidance tool for layer pruning.

References

[1] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770-778 (2016).
[2] Maqueda, A.I., Loquercio, A., Gallego, G., Garc?a, N., Scaramuzza, D.: Eventbased vision meets deep learning on steering prediction for self-driving cars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5419-5427 (2018).
[3] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, realtime object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 779-788 (2016).
[4] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems. pp. 3104-3112 (2014).
[5] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science (1985).
[6] Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A., Olah, C.: Multimodal neurons in artificial neural networks. Distill 6(3), e30 (2021).
[7] Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning. pp. 3519-3529 (2019).
[8] Nguyen, T., Raghu, M., Kornblith, S.: Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In: International Conference on Learning Representations (2021).
[9] Raghu, M., Gilmer, J., Yosinski, J., Sohl-Dickstein, J.: SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In: Advances in Neural Information Processing Systems. pp. 6076-6085 (2017).
[10] Morcos, A.S., Raghu, M., Bengio, S.: Insights on representational similarity in neural networks with canonical correlation. In: Advances in Neural Information Processing Systems. pp. 5732-5741 (2018).
[11] Wang, L., Hu, L., Gu, J., Hu, Z., Wu, Y., He, K., Hopcroft, J.E.: Towards understanding learning representations: To what extent do different neural networks learn the same representation. In: Advances in Neural Information Processing Systems. pp. 9607-9616 (2018).
[12] Tang, S., Maddox, W.J., Dickens, C., Diethe, T., Damianou, A.: Similarity of neural networks with gradients. arXiv preprint arXiv:2003.11498 (2020).
[13] Feng, Y., Zhai, R., He, D., Wang, L., Dong, B.: Transferred discrepancy: Quantifying the difference between representations. arXiv preprint arXiv:2007.12446 (2020).
[14] Das, A., Rad, P.: Opportunities and challenges in explainable artificial intelligence (xai): A survey. arXiv preprint arXiv:2006.11371 (2020).
[15] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. pp. 818-833 (2014).
[16] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Gradcam: Visual explanations from deep networks via gradient-based localization. In: International Conference on Computer Vision. pp. 618-626 (2017).
[17] Wang, F., Liu, H., Cheng, J.: Visualizing deep neural network by alternately image blurring and deblurring. Neural Networks 97, 162-172 (2018).
[18] Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5188-5196 (2015).

DOI: 10.54254/2755-2721/6/20230665

[19] Yao Lu, Wen Yang, Yunzhe Zhang, Zuohui Chen, Jinyin Chen, Qi Xuan, Zhen Wang, Xiaoniu Yang, Understanding the Dynamics of DNNs Using Graph Modularity, ECCV2022.
[20] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009).
[21] Tang, S., Maddox, W.J., Dickens, C., Diethe, T., Damianou, A.: Similarity of neural networks with gradients. arXiv preprint arXiv:2003.11498 (2020).
[22] Watanabe, C., Hiramatsu, K., Kashino, K.: Modular representation of layered neural networks. Neural Networks 97, 62-73 (2018).
[23] Watanabe, C., Hiramatsu, K., Kashino, K.: Understanding community structure in layered neural networks. Neurocomputing 367, 84-102 (2019).
[24] Watanabe, C.: Interpreting layered neural networks via hierarchical modular representation. In: International Conference on Neural Information Processing. pp. 376-388 (2019).
[25] Davis, B., Bhatt, U., Bhardwaj, K., Marculescu, R., Moura, J.M.: On network science and mutual information for explaining deep neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 8399C-8403 (2020).
[26] Hod, S., Casper, S., Filan, D., Wild, C., Critch, A., Russell, S.: Detecting modularity in deep neural networks. arXiv preprint arXiv:2110.08058 (2021).
[27] You, J., Leskovec, J., He, K., Xie, S.: Graph structure of neural networks. In: International Conference on Machine Learning. pp. 10881-10891 (2020).
[28] Csordas, R., van Steenkiste, S., Schmidhuber, J.: Are neural nets modular? inspecting their functionality through differentiable weight masks. In: International Conference on Learning Representations (2021).
[29] Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behavioral and Brain Sciences 40 (2017).
[30] Alet, F., Lozano-Perez, T., Kaelbling, L.P.: Modular meta-learning. In: Conference on Robot Learning. pp. 856-868 (2018).
[31] Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., Scholkopf, B.: Recurrent independent mechanisms. In: International Conference on Learning Representations (2021).
[32] Kirsch, L., Kunze, J., Barber, D.: Modular networks: Learning to decompose neural computation. In: Advances in Neural Information Processing Systems. pp. 2414-2423 (2018).
[33] Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).
[34] Duncan J. Whatts, Small WorldsThe Dynamics of Networks between Order and Randomness, pp.33, 1999, Princeton University Press.
[35] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015).
[36] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520.

Appendix 1:

The similarity histogram of other models: ResNet18, ResNet50, ResNet101, VGG11-bn, and VGG16-bn.

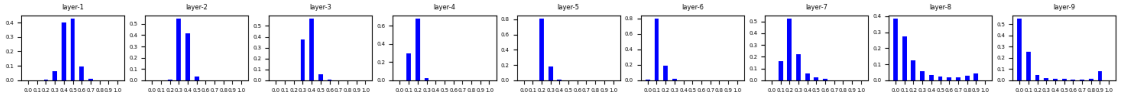

Figure 11. Similarity distribution of ResNet18.

DOI: 10.54254/2755-2721/6/20230665

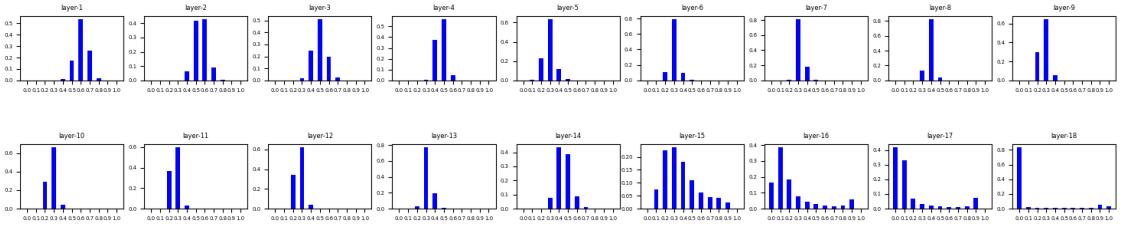

Figure 12. Similarity distribution of ResNet50.

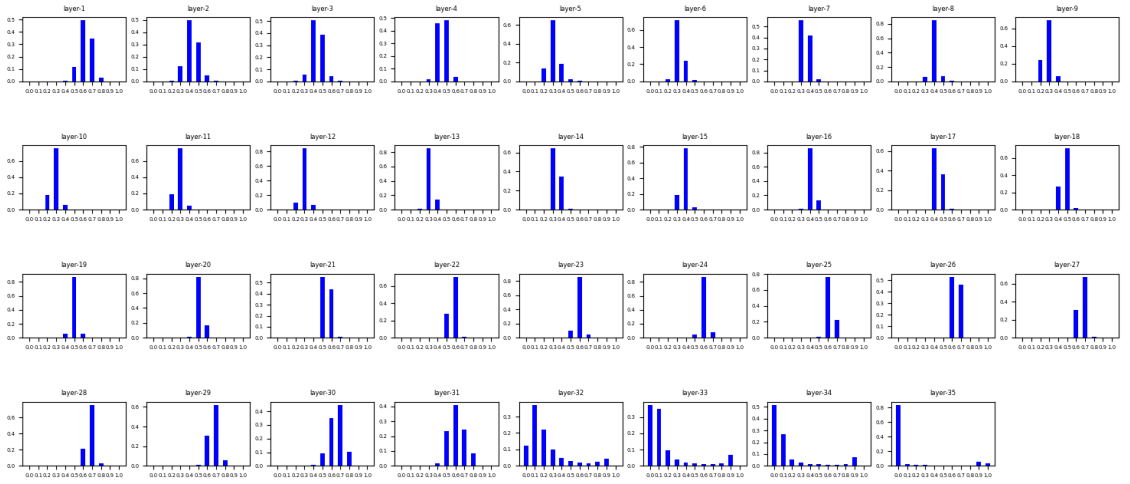

Figure 13. Similarity distribution of ResNet101.

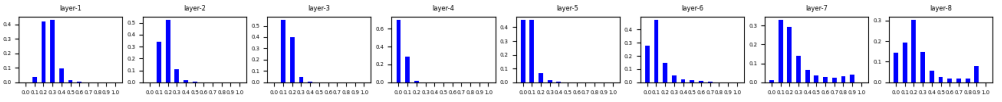

Figure 14. Similarity distribution of VGG11-bn.

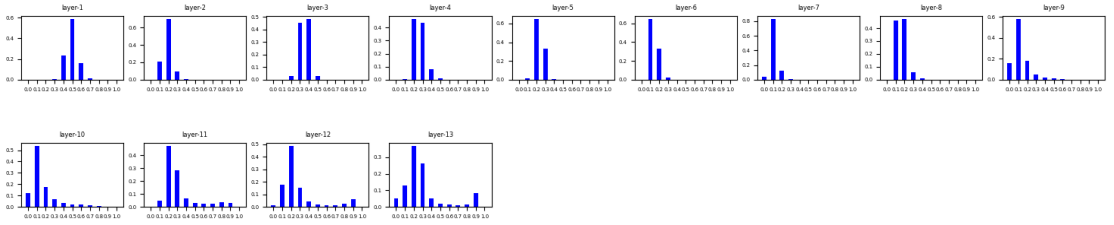

Figure 15. Similarity distribution of VGG16-bn.

Appendix 2:

The dynamic graphs of other models: ResNet18, ResNet50, ResNet101, VGG11-bn, and VGG16-bn.

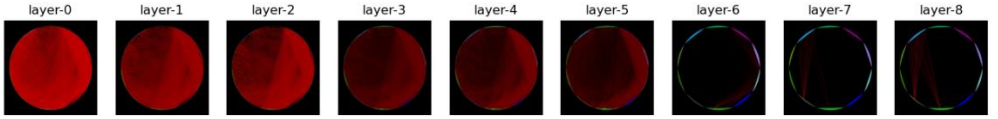

Figure 16. Dynamic graph of ResNet18.

DOI: 10.54254/2755-2721/6/20230665

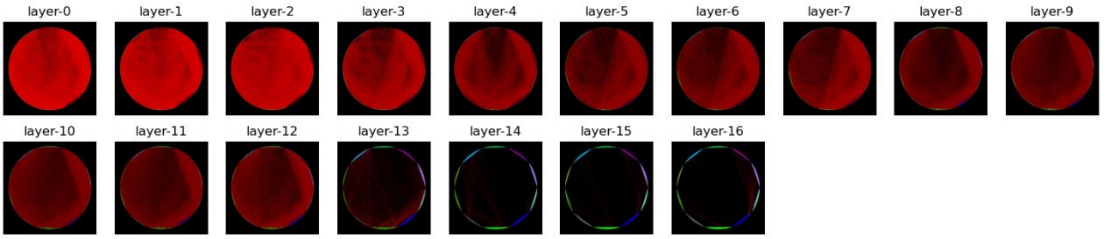

Figure 17. Dynamic graph of ResNet50.

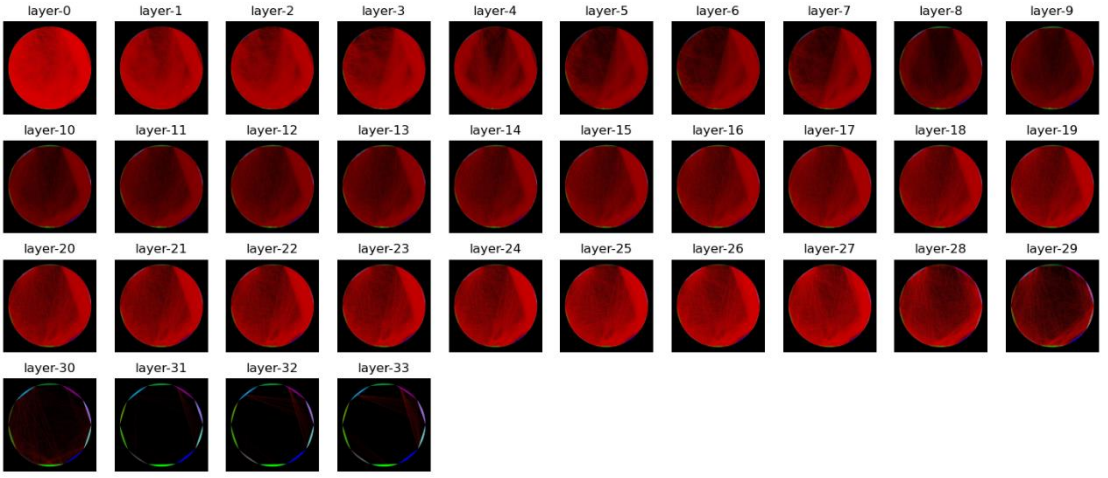

Figure 18. Dynamic graph of ResNet101.

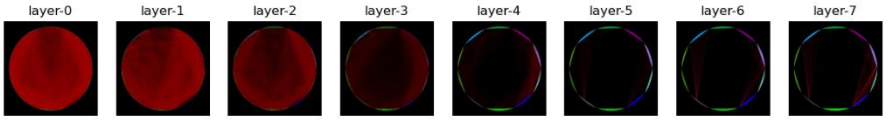

Figure 19. Dynamic graph of VGG11-bn.

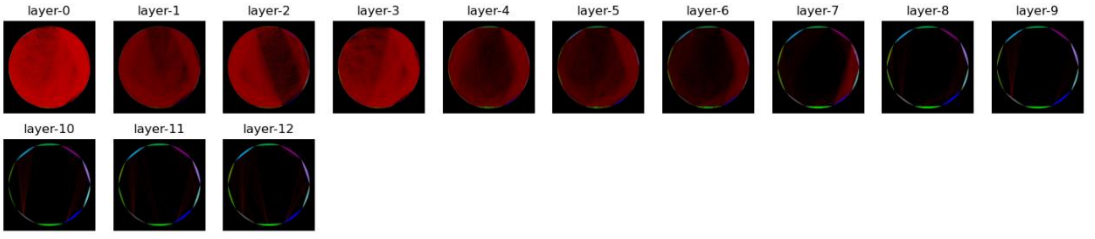

Figure 20. Dynamic graph of VGG16-bn.