CNN Hyperparameter Optimization Based on CNN Visualization and Perception Hash Algorithm

Yifeng Wang, Yang Wang, Hongyi Li, Zhuoxi Cai, Xiaohan Tang, Yin Yang
School of Science, Harbin Institute of Technology
Shenzhen, China
wangyifeng_ai@163.com

***Abstract*—In this paper, the network structure and the optimal hyperparameter selection which affect the performance of the model are obtained through the analysis of the convolutional neural network model with mathematical interpretation and visualization. In the study, we used visual methods such as deconvolution and Guided Grad-CAM to display the network structure, parameter changes, and the learning process of the model convolutional layer. Simultaneously, we developed CNN hyperparameters optimization strategy based on the perceptual hash algorithm according to its training characteristics. This method significantly improves the accuracy of image classification of the model and the generalization ability of the model and also provides certain theoretical support for the optimization and understanding of deep learning models in practical application. In addition, the hyperparameter optimization method based on the deep learning model feature map reconstruction visualization proposed in this paper also provides a good idea for the formulation of model training strategies.**

Keywords-convolutional neural network; visualization; interpretation; hyperparameters optimization; perceptual hash algorithm;

I. INTRODUCTION

The convolutional neural network has shown superior performance in the field of computer vision. As an end-to-end learning mode, it essentially uses a large amount of labeled data to backpropagate errors, thereby optimizing the parameters to be trained and improving the model performance. This way of learning is like a black box. Although the model performs well, we cannot understand the working principle of the model, so it is difficult to formulate a unified and effective training strategy. The uninterpretability of the model also increases the difficulty in the selection of hyperparameters greatly. However, if we can explain the working principle and effectiveness of the model, such as analyzing the feature extraction process and feature integration process of the model in the training process, it will be helpful for us to adjust parameters based on the working mechanism of the model, to improve the training efficiency and even obtain some universal structural design and parameter adjustment strategies.

Therefore, we used deconvolution [1-2], grad-CAM [3-4], and Guided grad-CAM [5] to visualize the model structure and training process, and reconstructed The feature extraction process of CNN. Taking Guided grad-CAM as an example, the method combines backpropagation with the method of using gradient global average to calculate the corresponding weight of each pair of feature maps，and finally calculate the weighted sum to create a high-resolution category visualization. By analyzing the visualization results, we found some laws of CNN feature extraction. For example, the deeper the structure in the CNN network is, the more abstract the feature is extracted, and the features extracted by different filters may be redundant. Based on these characteristics of the CNN model, we established some CNN training strategies combined with the methods such as the perceptual hash algorithm, and finally effectively improved the accuracy of the model on MNIST, CIfar-10, CIfar-100, and other data sets.

II. FEATURE VISUALIZATION BASED ON DECONVOLUTION

A. Deconvolution Network Structure

The deconvolution network can be regarded as the reverse process of the convolutional network. To visualize the convolutional network, the deconvolution layer is added to each layer of the convolutional network. The detailed structure is shown in Fig. 1 below.

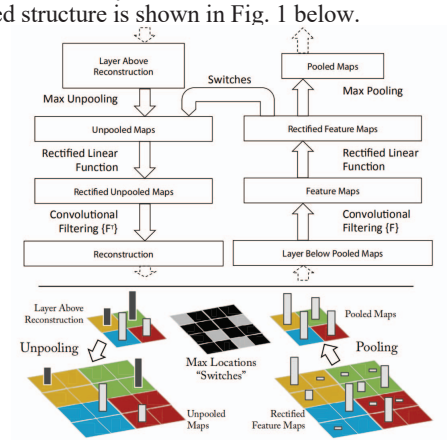


Figure 1. Deconvolution neural network diagram.

This figure shows a method path to return the feature map to the image pixel. The deconvolution layer (left) is connected to the convolutional layer (right). The deconvolution network will reconstruct the feature map in the convolutional layer. The bottom part of the figure shows an operation diagram of an unpooling operation in a deconvolution network.

In the process of maximum pooling of the convolutional layer, *variable switches* record the *Max Locations* of each maximum pooling area. The black/white bars in the diagram represent negative/positive activation in the feature map. First, input the image into the convolutional neural network, and the image features are calculated in the whole network layer. To judge a given network activation, it is necessary to set all other activation values in that layer to zero, and take the feature map as the input of the deconvolution network, and return to the previous layer successively, through the process of unpooling, de-activation, and deconvolution, and finally

2473-3636/20/$31.00 ©2020 IEEE
DOI 10.1109/DCABES50732.2020.00029

returns the pixel space to obtain the visual image. The specific decomposition process of model training is as follows:

Unpooling: In the forward propagation training process of the convolutional neural network, the max-pooling method is generally adopted when the training data passes through the pooling layer, which is irreversible. To reverse the process, a set of variables is needed to record the coordinate of the maximum in each pooling region, and then in the unpooling process of the deconvolution network, the reconstructed value from the previous layer is placed to the max location recorded previously, and the rest is set to zero. This is only an approximation because information stored at all locations except the one where the maximum value is located has been lost.

Deactivation: The convolutional neural network used the Relu nonlinear activation function to correct the obtained feature mapping, to ensure that the weight value in the feature mapping is always positive. In the deconvolution network, the Relu function is also used to activate features in order to ensure effective feature reconstruction in each layer.

Deconvolution: The process in which the convolution kernel acts on the network feature map of the previous layer is a convolution operation. To reverse the process, the deconvolution uses the transposed form of the same filter and applies it to the feature mapping.

B. Visualization results

Based on the above principle, a deconvolution network is constructed, which consists of two convolutional layers, two 2×2 maximum pooling layers, and the fully connected layer. The loss function of the output layer is Softmax. The model was trained on the CIFAR-10 data set. After the training, the features learned by the convolution kernel were projected to the pixel space through the deconvolution structure to obtain the visualization results. The experimental model contains two convolutional layers, which are respectively visualized as follows:

1) Visualization of features in the first layer.

We sent pictures of dogs, planes, and horses to the network, and visualized the features extracted by 32 filters in the first layer. We found that each filter extracts a different feature, as shown in Fig. 2, Fig. 3 and Fig. 4.

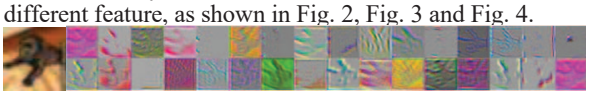


Figure 2. Visualization results of the first convolutional layer of dogs.

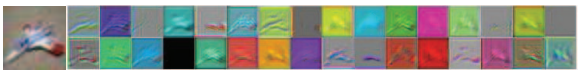


Figure 3. Visualization results of the first convolutional layer of planes.

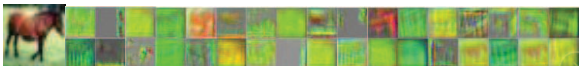


Figure 4. Visualization results of the first convolutional layer of horses.

From Fig. 2 to Fig. 4 above, we can see parts of the outline of the original image in each visual image, and the parts of the restored image are different. It can be seen that the parts and features of the image extracted by different filters of each convolutional layer are different during the learning and training of the model, that is, various features of the image are extracted from multiple angles. From the visualization results of the feature maps, the features of the main parts of the original image can be seen clearly, and the outlines of dogs, airplanes, and horses can be distinguished clearly.

Multiple pictures are input into the neural network structure designed by us, and the seventh feature mapping of the first layer is visualized by the deconvolution method and returned to the pixel space in Fig. 5(a). As shown in the figure below, the images are all red, but every image shows the outline and the shape of the main part of the original picture. Since each convolution kernel extracts the same image features for different images, which can determine that, different convolution kernels for the extraction of image features should have obvious functionality and specificity.

2) The second layer of feature visualization:

We used the same pictures as the training sample, and the feature map obtained by the convolution of the front layer is used as the input of the second convolutional layer along with the forward propagation of the model. The visualized result after the interaction with 64 convolution kernels is shown in Fig. 5(b).

(a) (b)

Figure 5. (a) Visualization results of the 7th feature map of different pictures. (b) Visualization results of the second layer feature map.

As can be seen from Fig. 5(b), the characteristics of the original image can hardly be seen from the visualization result. Experiments show that the features extracted by the second convolutional layer are obviously different from the convolution results of the previous layer. Compared with the image features extracted by the first convolutional layer, they are mostly image outline or color features. It can be seen that the features learned by this layer are relatively abstract.

For different input images, the 7th feature visualization result of the second layer of the model is shown in Fig. 6.

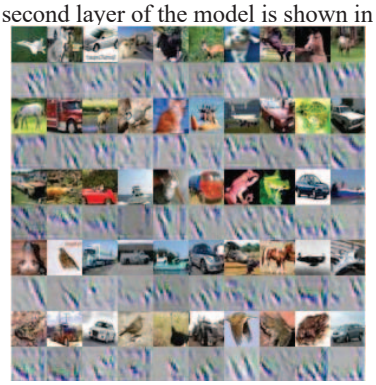


Figure 6. Visualization results of the 7th activation value in the second layer feature map.

It can be seen from Fig. 6 that for different images, the features extracted by the same filter should be the same. Besides, since the input of the subsequent convolutional layer is the feature mapping of the preceding convolutional layer, the features obtained by the subsequent nonlinear activation will be more abstract.

From the above CNN visualization experiment, we can find that model training is very effective, and features learned by different kernels are different in the efficient CNN model. Therefore, in image feature extraction, by setting the reasonable number of convolution kernels, repetition can be avoided, which also effectively reduces the amount of computation in model training and improves the efficiency of model training.

III. FEATURE VISUALIZATION BASED ON CATEGORY ACTIVATION MAPPING

A. CNN Visualization Based on Grad-CAM

The grad-CAM method obtains the weight value by calculating the contribution degree of each feature map to the model decision. α_k^c represents the weight of the influence prediction result of the k-th feature graph, which can be calculated by (1):

$$\alpha_k^c = \frac{1}{z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{1}$$

Where Z is the number of pixels in the feature map, y^c is the value predicted as class C, and A_{ij}^k represents the pixel value at the position of (i,j) in the k-th feature map. After the weights of all feature maps to the category prediction are obtained, the weighted sum can be used to obtain the thermal diagram.

$$L_{Grad-CAM}^c = \mathrm{Re}\,LU(\sum_k \alpha_k^c A^k) \tag{2}$$

After the weighted sum of the weights is obtained, the nonlinear operation is carried out, and a ReLU function is applied to it, to retain the pixel values in the image that have a positive influence on category C. Without ReLU operation, some pixels belonging to other categories will be substituted into the results, thus interfering the effect of model interpretation.

In general, the model of a deep network which extra more information predicts better. The function of the convolutional layer is to extract image information layer by layer, and the shallow convolutional layer extracts simple information, while the deepest convolutional layer can obtain rich content. After the image passes through the full connected layer and the Softmax layer, the spatial information will be lost, and the obtained data will not be able to fully express the original content of the image. To explain the CNN model reasonably, visualization operation should be made for the results of the deepest convolutional layer, because it is a generalization of the extracted features of the front layers.

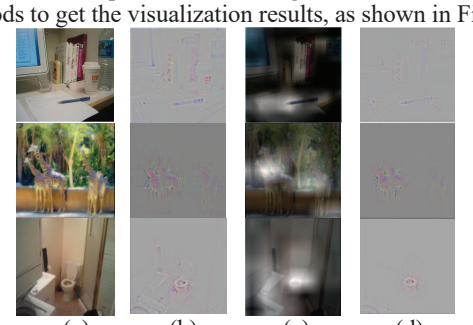
The grad-CAM visualization method is combined with the neural network constructed above, and the visualization results are shown in Fig. 7 below. As can be seen from the figure, After the image is reversely mapped back to the image by the feature map obtained after the first layer convolution, no special activation parts or pixels are shown. However, by observing the image of the last convolutional layer, the key areas in the image are clearly shown, and it can be seen that the head features of these animals are of great importance in the classification problem.

(a) (b) (c)

Figure 7. Grad-CAM visualization result diagram (a) Input image. (b) The first layer convolution. (c) The last layer convolution.

B. CNN visualization based on Guided Grad-CAM

Combined with the backpropagation and Grad-CAM methods, the Guided Grad-CAM method can not only explain the image classification task but also be of great help to target detection and location. We will use this method to study the concerned area of the model in an image and the extraction methods for features of different filters. This method uses the feature mapping method based on gradient signal combination and combines with the backpropagation method, so the model architecture does not need to be modified. Therefore, it can be applied to any CNN-based architecture with stronger generalization ability. Besides, this method only requires forward propagation and partial backpropagation of each image to complete positioning at one time, so it is more efficient. We input different images and use the three methods to get the visualization results, as shown in Fig. 8.



(a) (b) (c) (d)

Figure 8. Experimental comparison results (a) The original image. (b) Backpropagation. (c) Grad-CAM. （d）Guided Grad-CAM.

As is shown, the visualization results for the Guided Grad-CAM method only clearly depict some of the key pixels for correctly predicting the class, ignoring the secondary and background pixels. For example, when making predictions for desk images, the Guided grad-CAM visualization method only clearly depicts the articles such as books, pens, and computers that are closely related to the desk, while ignoring the articles such as water cups that have no obvious relationship with the desk.

Compared with the grad-CAM method, the Guided grad-CAM method has stronger generalization ability, which can obtain a clearer visualization graph for distinguishing image categories and can be used without modifying the structure of the model. It can be seen from the above that this method combines the advantages of the

backpropagation method and the Grad-CAM method, which can clearly show part of the pixel regions that affect the classification results and provides a visual interpretation for the model prediction results.

IV. CNN HYPER-PARAMETER OPTIMIZATION STRATEGY BASED ON THE PERCEPTUAL HASH ALGORITHM

In the previous part, the features learned at each layer in the convolutional neural network are expressed in reverse by deconvolution, and the feature map is expressed in the form of pixels in reverse. The learning essence of the CNN model can be understood by using network visual images.

Through the CNN visualization results based on Grad CAM and Guided Grad CAM, we can obtain some analysis results as follows:

- Each convolution kernel extracts a specific feature, and the shallow convolutional layer of CNN mainly learns the simple structural information of the image, while the content learned by the deep convolutional layer is more complex information, to form more specific target characteristics.
- When CNN is used for classification tasks, features extracted by the deeper convolutional layer are the most abstract, and features extracted by the deepest convolutional layer play a key role in classification tasks.
- Visualization results based on the Guided Grad-CAM method can locate the target more clearly, which helps the model to classify correctly and explains the decision-making of the model.
- Based on the above analysis results, we try to optimize CNN's training strategy and select more appropriate hyperparameters without changing the deep learning framework.

First of all, we reconstructed the feature maps of each layer of CNN. Taking the 64 channels of layer 0 as an example, we obtained 64 feature maps as shown in Fig. 9.

Since each convolution kernel extracts a different image feature, the redundancy of the convolution kernel of this layer is measured by the similarity between the reconstructed feature maps. If the redundancy is low, we will appropriately increase the number of convolution kernels of this layer, which will help the model to extract more effective features. However, if the redundancy is high, we will consider formulating the hyperparameter optimization strategy from two aspects:

1) The high redundancy of the convolution kernel may be due to inadequate training. In this case, we will increase the training times and adjust the learning rate.

2) Assuming that the model has been fully trained, we will consider reducing the number of convolution kernels, which will help to reduce the network scale and reduce the time and space complexity of the model, which will have an obvious effect in practical application.

We use the perceived hash algorithm as a method to measure the similarity of the reconstructed feature map. The image pixel matrix is set as f and the discrete cosine transform is first performed to transform the two-dimensional image from the spatial domain to the frequency domain:

$$F = AfA^T$$

$$A(i,j) = c(i)\cos[\frac{(j+0.5)\pi}{N}i] \tag{3}$$

$$c(i) = \begin{cases} \sqrt{\frac{1}{N}}, & i = 0 \\ \sqrt{\frac{2}{N}}, & i \neq 0 \end{cases}$$

A is the conversion matrix, where i is the horizontal frequency of the two-dimensional wave, j is the vertical frequency of the two-dimensional wave, and the value ranges from 0 to (N-1), and N is the size of the image block.

After the discrete cosine transform result of the image is obtained, the average value is calculated, the pixel gray level is compared, and finally, the comparison result is combined to form the fingerprint of the image. The similarity of different images can be obtained by calculating the Hamming distance between the fingerprints of different images. We calculate the similarity between each feature map and other feature maps in Fig. 9, and obtain an upper triangular matrix 64x64, as shown in Fig. 10.

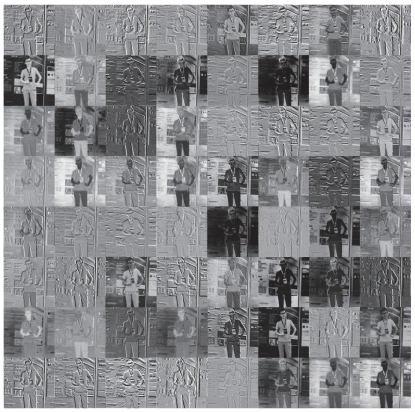


Figure 9. Reconstruction feature map of layer 0.

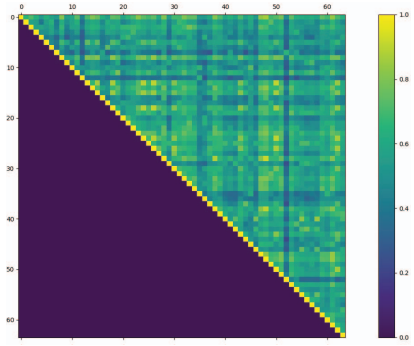
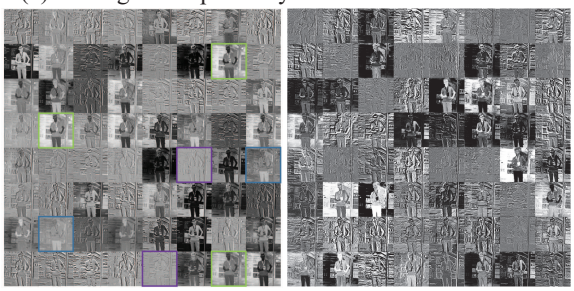


Figure 10. Visualization of the similarity matrix of the feature graph in the 0th layer.

The similarity between some feature maps is very high, which means that the extracted features are very similar. For the convenience of observation, we annotate the 64 feature maps in layer 0 that are partly similar, as shown in Fig. 11(a). It can be seen that the feature map in the same

color box is very similar, so we choose to reduce the number of convolution kernels in this layer. On the contrary, in the second layer, the reconstruction results of the feature map and its similarity matrix are shown in Fig. 11(b) and Fig. 12 respectively.



(a) (b)

Figure 11. (a)Similarity feature map annotation at layer 0. (b) Reconstruction feature map of the second floor.

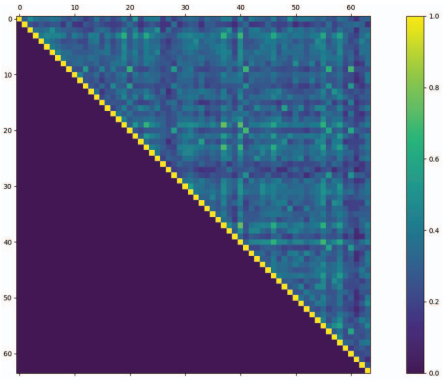


Figure 12. Visualization of the similarity matrix of the second layer feature map.

The similarity between layer characteristic figure is lower, the extracted feature types are rich, therefore, we gradually increase the number of convolution kernels in this layer, finally found that when the number of convolution kernels is near 128, the similarity between feature maps just can be maintained at a suitable level, it shows that different convolution kernels can extract exactly different features.

After adjusting the whole model structure and training strategy with this method, the performance of the model in different training sets was significantly improved.

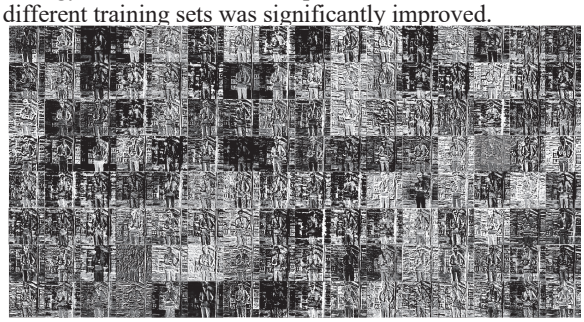


Figure 13. Reconstruction results of the adjusted feature map of the second floor.

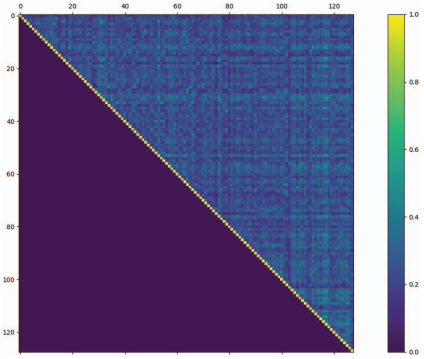


Figure 14. Visualization of layers similarity matrix after adjustment.

TABLE I. MODEL PERFORMANCE ON DIFFERENT DATA SETS BEFORE AND AFTER MODEL ADJUSTMENT

Sets	*Before the adjustment*	*After the adjustment*
MNIST	98.07%	**99.42%**
CIFAR-10	82.25%	**91.61%**
CIFAR-100	77.41%	**87.79%**

V. CONCLUSION

In this paper, the CNN feature map was reconstructed using several model visualization methods such as deconvolution, Grad-CAM, and Guided grad-CAM, and the working mechanism and prediction principle of the CNN model were explored according to the visualization results. We found that the functions of different convolution kernels at different layers in the CNN model are significant specific, which is closely related to the final model effect. Therefore, according to the analysis results, we developed a hyperparameter optimization strategy based on the perceptual hash algorithm, which finally improved the performance of the original CNN model on MNIST, CIFAR-10, and CIFAR-100 data sets significantly.

In addition, the method proposed in this paper to develop the hyperparameter optimization strategy by visualizing the feature maps of the deep learning model has a great room for expansion. This paper focuses on the similarity between feature maps. In the future, the mathematical and statistical properties of feature maps can be studied from more perspectives, and the operating principle of deep learning model can be further understood, to develop more optimized training strategies.

REFERENCES

[1] Zeiler MD, Fergus R. Visualizing and Understanding Convolutional Networks[J]. Springer Verlag, 2014, 8689 (1): 818-833

[2] Zeiler MD, Krishnan D, Taylor G W, et al. Deconvolutional Networks[C]//Proc. IEEE Computer Society Co.Ceoer Vision and Pattern Recognition. IEEE, 2010.

[3] Zhou B, Khosla A, Lapedriza A, et al. Learning Deep Features for Discriminative Localization[J]. IEEE Computer Society, 2016:2921-2929.

[4] Selvaraju R R, Das A, Vedantam R, et al. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization[J]. 2016.

[5] Zaira García, YanaiK, Nakano M, et al. Mosquito Larvae Image Classification Based on Den seNet and Guided Grad-CAM[M]// Pattern Recognition and Image Analysis. 2019.