The linear representation of CNN for single image

Wei Yu W.YU@HIT.EDU.CN
Kuiyuan Yang KUYANG@MICROSOFT.COM
Yalong Bai YLBAI@MTLAB.HIT.EDU.CN
Tianjun Xiao TIXIAO@MICROSOFT.COM
Hongxun Yao H.YAO@HIT.EDU.CN

Abstract

CNN can model the complex underline mappings between images and categories through several layers via non-linear activation function. However, it is hard to analyze the non-linear relation learned in the CNN. In this paper, we show that a set of well-performed CNNs (composed of convolutional layers, max-pooling layers and ReLU) are piecewise linear, i.e., linear at every single image. The nice property means that the output/score of a neuron is a linear combination of outputs of any lower layer for an image. With the property, we can distribute the score of a neuron to every position of a lower layer to probe where contributes more for the score of the neuron.

1. Introduction

Deep Convolutional Neural Networks (CNN) have become an essential machine learning method for object recognition, since it was proposed in computer vision (LeCun et al., 1989). Continuous efforts have been involved to improve CNN's performance in image classification. Meanwhile, the CNNs with more weight layers have made significant progress in many research areas, such as face detection (Sun et al., 2013) (Taigman et al., 2014), object segmentation (Girshick et al., 2014) (Hariharan et al., 2014), object detection (Zhang et al., 2014) (Erhan et al., 2013) and human pose estimation (Toshev & Szegedy, 2014) (Chen & Yuille, 2014). Currently, more researchers attempt to gain insights into the learned CNN in detail, and some approaches are proposed to understand CNN through visualizing CNN model using feedback information. The core idea is visualizing feature to gain intuition about CNN on any layers, while traditional approaches are limited to the first convolutional layer.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

The feedback process can be realized implicitly through showing the filters on any layers (Girshick et al., 2014). The patches covered by a specific neuron can be found in original image based on the mapping between layers. The non-parametric method captures the patches corresponding to high activations on a large-scale set of candidate patches, and shows the invariance computed by the selected neuron. This simple but effective approach regards the unit as a pattern detector, which directly shows the visual modes on different layers.

A visually attractive deconvolutional approach (Zeiler & Fergus, 2014) is presented to reveal the input individual feature maps at any layer in a learned model. The feedback information is transferred along with a multi-layered deconvolutional network, and the selected feature activations can be projected back to the input layer. Although the reconstructed patches are not projected back to the original image space, the visualization ability is a great complementary tool to show the pattern for specific filters.

The gradient can be utilized as feedback information (Simonyan et al., 2013) to visualize the convolutional networks. The specific class model can be visualized through maximizing the corresponding neuron of last full-connected layer. For a specific image, the gradient of a selected class can be easily computed using a single back-propagation pass through CNN. Then the gradient based image-specific class saliency map can be used for weakly supervised object localization. To some extent, the gradient based visualization approach is equivalent or similar to the reconstruction based deconvolutional network as the discussion in (Simonyan et al., 2013). Thus, the gradient based visualization and reconstruction based visualization are unified and complementary.

In this paper, we propose to translate the image-specific prediction to linear form, and name it as score map. On one hand, score map can search the sub-image with high probabilities to be foreground objects. On the other hand, score map is produced to highlight the local patterns with discriminative content,

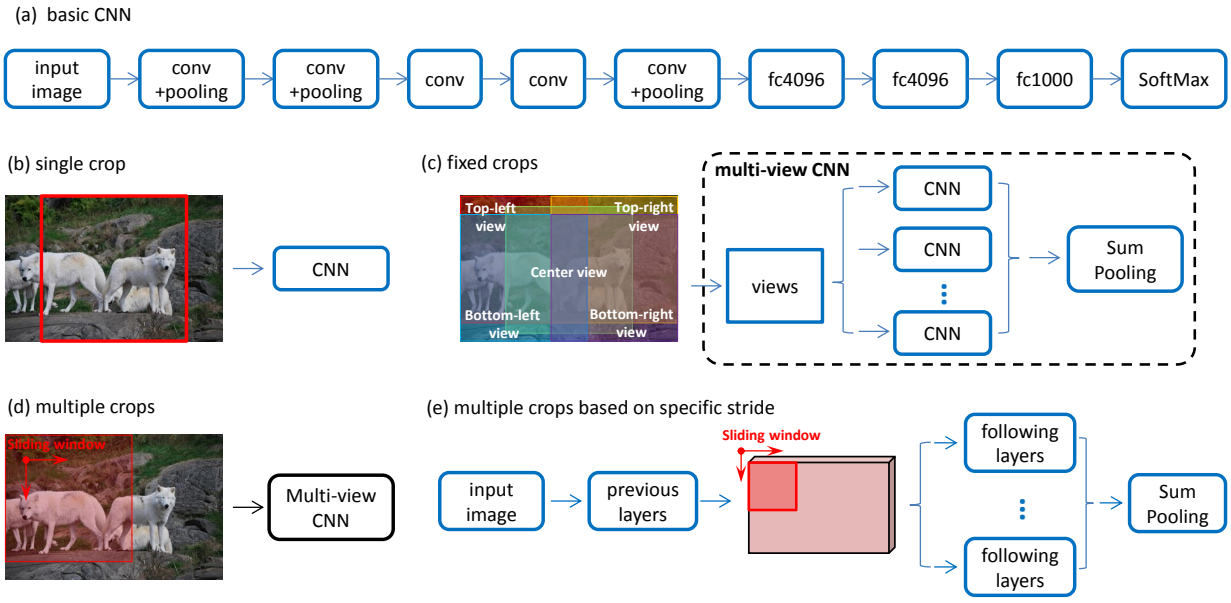

Figure 1. The summary of prediction strategies. (a): A received structure of CNN introduced by (Krizhevsky et al., 2012). (b): Single corp based prediction just capture one patch with fixed size as input view. (c): Multiple crops are cropped from fixed locations, such as four corner and center (Krizhevsky et al., 2012). (d): Multi-crop based prediction extracts views using sliding windows (Simonyan & Zisserman, 2014). (e): Multi-crop with specific stride based prediction extracts views from the output of one layer, where previous layers work on original image size.

2. Score map for single input image

2.1. The prediction architecture

Currently, most outstanding CNN are designed with fixed input size due to the technical issue as perviously mentioned. However, the actual images and most popular datasets, such as ImageNet (Russakovsky et al., 2014) and PASCAL VOC (Everingham et al.), provide the test images with arbitrary size. Thus, many classification strategies were introduced for the learned CNN model with fixed input image size predicting the test images with arbitrary size, as shown in Fig. 1.

Generally, a trained CNN model (Fig. 1 (a)) requires input view with fixed size (Fig. 1(b)). In practice, CNN predicts multiple crops at test time, since single crop is less persuasive for the whole image. For instance, CNN predicts specific crops in fixed locations and average the final classification results over the predictions of all crops (Fig. 1(c)). Moreover, multi-crop prediction is implemented using sliding windows (Fig. 1(d)), rather than fixed location. In order to improve efficiency, more tasks slide window over one mediate layer to share the previous layers, since convolutional layers and pooling layers can be operated on arbitrary size (Figure 1(e)). However, this way only is equivalent to slide windows over input image with specific stride, considering the stride of convolutional filter or pooling kernel.

Although multi-crop prediction is more reasonable compared with single crop, the prediction with more views doesn't mean more accuracy. We implement multi-crop prediction based on the basic CNN structure through sliding feature window on the last pooling layer, where the sizes of previous convolutional and pooling layers both depend on the size of input image. In this paper, all processes are realized based on the multi-crop prediction framework.

2.2. The linear form of multi-view prediction architecture

The CNN is piecewise linear function, which divides the image space into massive of linear regions (Montufar et al., 2014). Thus, CNN shows the linearity in one linear regions. In feedforward process, the nonlinearity of basic CNN is caused by max-pooling layer and rectifier units, while both convolutional layer and fully-connected layer are linear. Through recording the pooling location and rectified information, the image-specific CNN can be represented as linear form, which also is treated as the feedback process from one label. As mentioned in 2.1, we will discuss the linear form of the multi-crop prediction architecture, which is built to predict the input image with arbitrary size.

There are three differences between basic CNN and multi-view prediction architecture: (1) **the convolutional and pooling layers with unfixed size**, the previous layers all

are adjusted to fit the size of input image; (2) **sliding layer**, an additional sliding layer is introduced, since the operation of window sliding can be treated as a special convolution operation, and the size of feature window is same with the first following layer in basic CNN; (3) **sum-pooling layer**, one additional sum-pooling layer is introduced to average the predictions of the extracted views. All these three additional layers are linear, which the feedback process of multi-crop prediction architecture for one specific image is linear.

The operation of fully-connected layer is inherently linear:

$$Y = R(FX + b) \tag{1}$$

where Y and X are the output and input of the fully-connected layer with the vector form. F is the weight matrix. b is the bias. R corresponds to the operator of ReLUs, where the diagonal record the rectified information.

Similar to Eq. 1, other layers also can be reformulated as linear form through transforming the weights. For example, in the linear representation of convolutional layer, the output and input are resized as vectors. The weights of filters are assigned to corresponding position of each row of F. In the linear representation of max-pooling layer, each row of F only contains one weight 1 corresponding to the pooling position, and weights 0 on other positions. R is the identity matrix and b is the zeros vector.

Therefore,the multi-view prediction architecture can be formulated as linear form based on Eq. 1:

$$Y_l = R_l(F_l Y_{l-1} + b_l), l = 1, ..., L \tag{2}$$

where Y_l is the output of the l^{th} layer, and Y_0 is the input image. F_l is the weight matrix of the l^{th} layer. L is the count of layers. For example, our multi-view prediction architecture is composed of 13 layers, including 5 convolutional layers, 3 pooling layers, 3 fully-connected layers, 1 sliding layer and 1 sum-pooling layer.

2.3. Score map of specific label

In practice, the output of last fully-connected layer are the predicted scores of all labels, while the soft-max layer translates the predicted scores into the probability Based on Eq. 2, the predicted scores of label c can be represented as the linear form with output of any layer:

$$S_{I,c} = A_{I,c,l} Y_{I,l} + B_{I,c,l}, l = 0, ..., L \tag{3}$$

where $S_{I,c}$ is the c^{th} label's predicted score of image I, $Y_{I,l}$ is the l^{th} layer's output of image I resized as vector. $A_{I,c,l}$ and $B_{I,c,l}$ are produced based on iterative Eq. 2.

In order to define score map, we first consider the product of one activation and corresponding coefficient in Eq. 3, which is treated as sub-score of the activation to corresponding label. These sub-scores are able to measure the degree of one activation predicting as specific label. Further, the score map of one layer can be produced through summing all channels of sub-score, which measures the degree of locations predicting as specific label. In order to capture score map, we resize $A_{I,c,l}$ and $Y_{I,l}$ as 3-D matrix with the size of the output of l^{th} layer. Then, the image I's score map of l^{th} layer for label c can be represented as:

$$M_{I,c,l}(x, y) = \sum_{1 \le h \le H} A_{I,c,l}(x, y, h) Y_{I,l}(x, y, h) \tag{4}$$

where $M_{I,c,l}$ is image I's score map of l^{th} layer for label c. (x, y) is the grid coordinate of the l^{th} layer's output. H is the channels' number of l^{th} layer and h the index of channel.

In particular, the score maps of different labels are distinguishing, even produced on the output of same layer. The score maps represent the regions responding to corresponding label, which also can be treated as saliency maps. The score maps of an example image are shown in Fig. 2.

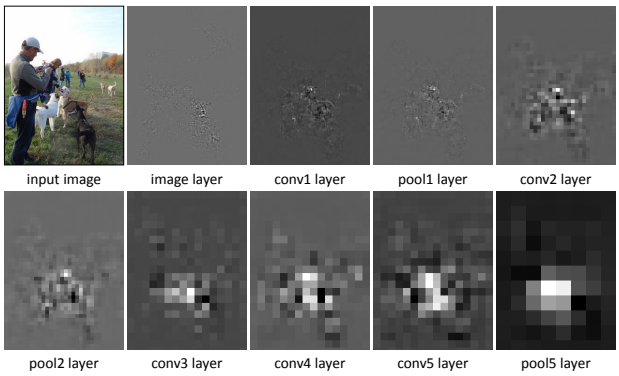

Figure 2. The score maps of different layers for groundtruth label. Left-top shows the example of input image. Others are the score maps of different layer, including convolutional layers, max-pooling layers and image layer. We adjust the size of score maps to the size of input using interpolation.

3. Score map based saliency map

The score map based saliency map can be captured based on the linear form of multi-view prediction architecture. Different from low-level feature driven saliency map, the score map is guided by high-level class label and rich semantic information, which makes score map focus on the objects of the training classes. As shown in Fig. 3, high score value of score map tends to be located surrounding the regions with help for prediction. In top example, high score value tends to locate the discriminative region, such as the head of dog. In middle example, the people is of rich semantical, but the class of people is not to be predicted. Thus, the region with people is treated as background and

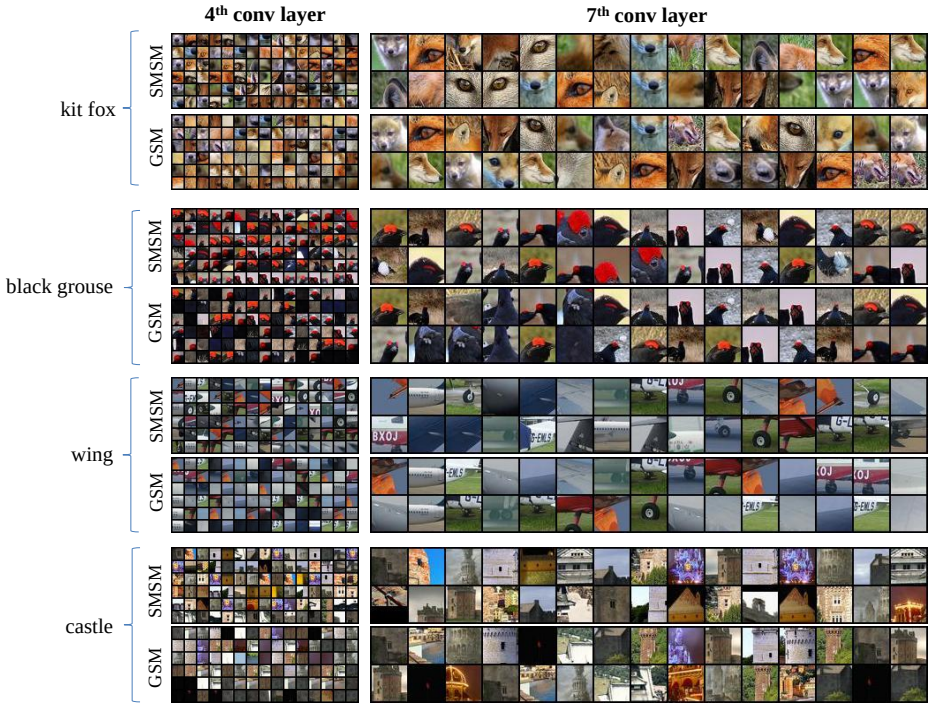

Figure 4. The semantical patches extracted from two convolutional layers based on SMSM and GSM. The saliency value of GSM is captured following the procedure of (Simonyan et al., 2013).

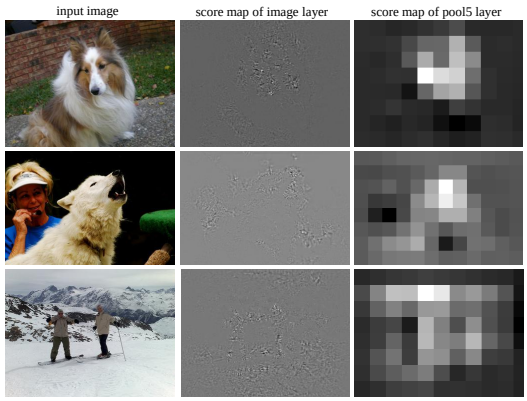

Figure 3. The score map with respect to groundtruth label. The high score tends to locate the discriminative content for prediction. The size of pool5's score map also are interpolated as the size of input image

with low score value. In bottom example, high score value tends to locate surrounding regions of label *alp*, where the contextual information is helpful for prediction. Obviously, the score map is help helpful to crop discriminative regions, which keeps from introducing artificial occlusion. We will introduce the algorithm of score map based crops selection in following section.

4. Score map based semantical pattern

Score map can be treated as the distribution of the predicted score of specific label. Thus, the high score value is of strong semantic with respect to specific class. We attempt to compare the score map based saliency map (SMSM) and gradient based saliency map (GSM). As shown in Fig. 4, we present the patches with high values of SMSM and GSM respectively. Compared with the GSM, the patches of SMSM contain more discriminative content, which is help for prediction. For instance, for the *kit fox* class, SMSM highlights the eye and nose on 4^{th} conv layer, while GSM highlights fur. For the *black grouse* class, SMS highlights the comb on 4^{th} conv layer, while GSM highlights feather and even grass. The obvious difference is presented on the 4^{th} conv layer, since the receptive field of 7^{th} conv layer owns larger size and contains more context. However, SMSM still highlights the semantical patches on 7^{th} conv layer, such as *wing* class and *castle* class.

References

Chen, Xianjie and Yuille, Alan L. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, pp. 1736–1744, 2014.

Erhan, Dumitru, Szegedy, Christian, Toshev, Alexander, and Anguelov, Dragomir. Scalable object detection using deep neural networks. *arXiv preprint arXiv:1312.2249*, 2013.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Girshick, Ross, Donahue, Jeff, Darrell, Trevor, and Malik, Jitendra. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 580–587. IEEE, 2014.

Hariharan, Bharath, Arbeláez, Pablo, Girshick, Ross, and Malik, Jitendra. Simultaneous detection and segmentation. In *Computer Vision–ECCV 2014*, pp. 297–312. Springer, 2014.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Yann, Boser, Bernhard, Denker, John S, Henderson, Donnie, Howard, Richard E, Hubbard, Wayne, and Jackel, Lawrence D. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4): 541–551, 1989.

Montufar, Guido F, Pascanu, Razvan, Cho, Kyunghyun, and Bengio, Yoshua. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2924–2932, 2014.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

Sun, Yi, Wang, Xiaogang, and Tang, Xiaoou. Deep convolutional network cascade for facial point detection. In *Computer Vision and Pattern Recognition, 2013*, pp. 3476–3483, 2013.

Taigman, Yaniv, Yang, Ming, Ranzato, Marc'Aurelio, and Wolf, Lior. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1701–1708. IEEE, 2014.

Toshev, Alexander and Szegedy, Christian. Deeppose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1653–1660. IEEE, 2014.

Zeiler, Matthew D and Fergus, Rob. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision, 2014*, pp. 818–833. 2014.

Zhang, Ning, Donahue, Jeff, Girshick, Ross, and Darrell, Trevor. Part-based r-cnns for fine-grained category detection. In *European Conference on Computer Vision, 2014*, pp. 834–849. Springer, 2014.