딥 CNN 에서의 Different Scale Information Fusion (DSIF) 의 영향에 대한 이해

Understanding the Effect of Different Scale Information Fusion in Deep Convolutional Neural Networks

Kai Liu, Usman Cheema and Seungbin Moon*
Department of Computer Engineering, Sejong University
e-mail : kailiu@sju.ac.kr, sbmoon@sejong.ac.kr

Abstract

Different scale of information is an important component in computer vision systems. Recently, there are considerable researches on utilizing multi-scale information to solve the scale-invariant problems, such as GoogLeNet and FPN. In this paper, we introduce the notion of different scale information fusion (DSIF) and show that it has a significant effect on the performance of object recognition systems. We analyze the DSIF in several architecture designs, and the effect of nonlinear activations, dropout, sub-sampling and skip connections on it. This leads to clear suggestions for ways of the DSIF to choose.

1. Introduction

The importance of analyzing images at different scales originates from the nature of images themselves [1]. Scenes in the world not only contain many different sized objects, but also these objects contain many different sized features. Moreover, from the perspective of the viewer objects can be at various distances. As a result, any analysis procedure that is applied only at a single scale may miss information at other scales. The solution is to go through analyses at all scales simultaneously.

Recently, in many deep convolutional neural networks [2] studies, combining features of different scales has received significant attention, in part because the lower layers features have higher resolution, contains more position and local detail information, but due to less convolution it has less global semantics and more noise. High-level or deep layer features have stronger global semantic information, however due to the low resolution, vast local information is lost. How to represent multi-level information efficiently is one of the keys to improving the deep model for classifying objects at different scales.

Image pyramids [1] were heavily used in the era of hand-crafted features. Image pyramids are scale-invariant in the sense that an object's scale change is offset by shifting its level in the pyramid. Intuitively, this property enables a model to detect objects across a large range of scales by scanning the model over both positions and pyramid levels.

Convolution is another basic operation of most image processing systems. In a multi-level system, such as GoogLeNet [3], they performed convolutions with many different sized kernels, ranging from very small to very large (1x1, 3x3, 5x5 and 7x7), of which small sized kernels are designed to represent local features and global features are represented by large sized kernels.

In general, the purpose of feature fusion is to combine the different scale features extracted from the image into one high-dimensional feature vector that is more discriminative than the input. How to properly and effectively fuse different scale features remains a challenge. Specifically, current deep models pose two major feature fusion methods: addition and concatenation.

Addition: Parallel strategy [4][5], which combines different scales feature vectors into one complex vector. The dimension of input feature vector, X_i, is equal to the dimension of output Y_j, and have the same height and width. For $0 <= i < N$:

$$Y_j = \sum_{i=0}^{N} X_i \qquad (1)$$

where N is the number of scales.

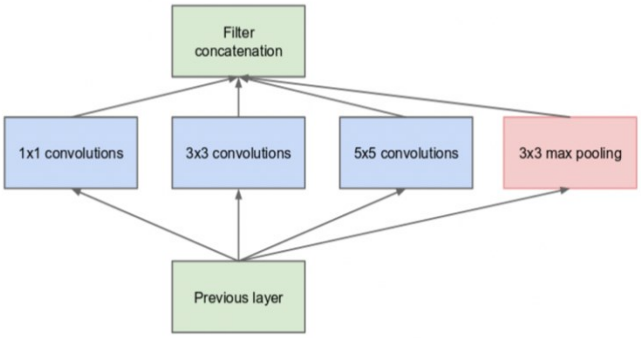
Concatenation: Series feature fusion [4][5], that different scales features are connected directly. Same height and width are the prerequisites of X_i, but the dimension of each X_i can be different. So that,

$$Q_j = \sum_{i=0}^{N} P_i \qquad (2)$$

where Q_j is the dimension of Y_j, P_i is the dimension of X_i

In this paper we aim to study the impact of fusion methods on the different scale features, focusing on the architecture of inception networks. We conduct extensive experiment on the large-scale image dataset (Places 365 standard [6]), and empirically, we discover the effect of DSIF.

The rest of the paper is as follows. In Section 2, the original version of inception network is introduced. In Section 3, our experimental details and results are showed. Finally, the conclusions are given.



(Figure 1) The basic inception module

2. Inception network

In the development of convolutional neural networks (CNNs) the Inception network [3] was an important milestone. Prior to its inception, most common CNNs just stacked convolution layers deeper and deeper, hoping to lead to a better performance. While very deep networks are not only prone to overfitting, but it also becomes hard to pass gradient updates through the entire network. Furthermore, simply stacking large convolution operations is computationally expensive. Instead of going "deeper", inception network decided to go "wider" with different sized kernels (1x1, 3x3 and 5x5). Additionally, max pooling is also performed. The outputs are concatenated and sent to the next inception module. Figure 1 shows the basic inception module.

3. Experiments

In this section, we introduce the experimental setting and declare the performance of DSIF on the Places365 standard [6] database. We first describe the dataset and our implementation details. Then, we discover the effect of DSIF by performing extensive experiments on the large-scale dataset.

The Places 365 standard [6] is a large-scale scene-centric benchmark, containing 365 common scene categories. In total the dataset includes 1.8 million training images where 5000 training images per class, and 50 images per category for validation. The evaluation criteria of the Places 365 standard is based on top5 accuracy [7].

For training, we employ the mini-batch stochastic gradient descent algorithm [8] to optimize network, where the batch size is set as 128 and momentum [9] set to 0.9. The learning rate [10] is initialized as 0.1 and decreases according to a fixed schedule as shown in table 1. For data augmentation [11] we only use randomly horizontal flipping. The experimental results are shown in Table 2.

As the result shows that concatenation is more effective and achieves a higher accuracy, when Q_j<1000, which means staking multi-level information all together is not prone to have problems with gradient vanishing or exploding. While when Q_j > 1000, the vanishing/exploding issue is more likely to happen. Therefore, addition method that plays the role of skipping connections is more appropriate. Finally, using both addition and concatenation separately at different depth of layers based on the number of neurons is the most effective way.

<Table 1> Learning Rate Schedule

Epoch	Learning rate
0-15	1e-1
15-30	1e-2
30-45	1e-3

<Table 2> Results on DISF

Fusion method	Top-5 Validation (Q_j<1000)	Top-5 Validation (Q_j>1000)
Add	82.25%	84.79%
Concat	84.21%	83.52%
Add + Concat	83.07%	85.37%

4. Discussions & Conclusions

In this paper we have conducted three kinds of DISF methods and analyzed the effect of them based on the classical architecture inception network. As one of the widely recognized CNN models, every version of inception has scored tremendous achievements in recent years. The concatenation of different sized kernels which is the only way they used. However, as shown in the Table 2, when the number of neurons exceed 1000, addition achieves better accuracy than concatenation. Though, combining of addition with concatenation gets the best performance. In the future we will explore more available ways to represent multi-level information and fusion methods.

References

[1] H. Adelson, H. Anderson, R. Bergen, J. Burt, and M. Ogden, "Pyramid methods in image processing", *RCA engineer* 29, no. 6 (1984): 33-41.

[2] Krizhevsky, Alex, I. Sutskever, and. G. Hinton, "Imagenet classification with deep convolutional neural networks", in *advances in neural information processing systems*, pp. 1097-1105. 2012.

[3] Szegedy, Christian, W. Liu, Y. Jia, P. Sermanet, S.Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich", Going deeper with convolutions." in *Proc. of IEEE conf. computer vision and pattern recognition*, pp. 1-9. 2015.

[4] J. Yang, J.-Y. Yang, D. Zhang, J.-F. Lu, "Feature

fusion: Parallel strategy vs. serial strategy", *Pattern Recognit.*, vol. 36, no. 6, pp. 1369-1381, Jun. 2003.
[5] J. Yang, J. Y. Yang, "Generalized K–L transform based combined feature extraction", *Pattern Recognit.*, vol. 35, no. 1, pp. 295-297, 2002.
[6] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition." *IEEE trans. on pattern analysis and machine intelligence* 40, no. 6 (2017): 1452-1464.
[7] Iandola, N. Forrest, S. Han, W. Moskewicz, K. Ashraf, J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." *arXiv preprint arXiv:1602.07360* (2016).

[8] Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." In *Proceedings of COMPSTAT'2010*, pp. 177-186. Physica-Verlag HD, 2010.
[9] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.
[10] E. Fahlman, "An empirical study of learning speed in back-propagation networks." (1991).
[11] L. Wang, G. Sheng, W. Huang, X. Yuanjun, and Q. Yu. "Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs." *IEEE Transactions on Image Processing* 26, no. 4 (2017): 2055-2068.