

Mystic Kernel Size in CNN

Lei Shiye

2019
March

Abstract

With the fast development in deep learning filed, there are more and more deep neural networks(DNN) with various architectures. Besides, the number of layers in DNN is becoming larger and larger. Under the circumstance, some features between basic convolutional neural network(CNN) may be overlooked by us, like kernel size. In this report, I focus on the how the size of kernel in CNN have a influence on deep learning process. Through testing a same CNN with different kernel size on different dataset, we can find some interest phenomenons and extract some information from them.

Note: The source codes ralated to the report can be found on github:

https://github.com/LeavesLei/UPC-DL-lab-code/tree/master/lab_1/source_code_lab_1

1 Introduction

After AlexNet [4] showing strong strength in ImageNet competition in 2012, deep learning has received the attention of many people. In a basic image classification task, a DNN with some convolutional layers, pooling layers and full connected layers is used to extract information from image dataset. We often devide the image dataset into two subset: training dataset and validation dataset during training process. Maybe we could do the dataset augmentation before put them into network. After some epochs, we can get DNN weights for the classification task.

At the same time, neural networks became deeper and deeper and the final accuracy is higher and higher. But the most critical part of DNN is still convolutional layer. Lots of beginners including me in the filed of deep learning is a little hard to understand that why the small convolutional kernel could extract huge information from image.

So I decided to do some reseach about convolutional kernel size. More specifically, I want to figure out whether there is a relationship between convolution kernel size and final accuracy in an image classification task.

2 Methods

To focus on convolutional kernel size, I didn't use DNN with complex architecture like VGGNet [8] or ResNet [2] but LeNet-5 [5], a classic CNN. The architecture of LeNet is presented in Fig 1

2.1 CNN architecture

LeNet-5 is often used to deal with MNIST whose figure size is 28×28 . In this report, I used several image datasets which image sizes is far greater than MNIST. So the CNN should have a greater amount of weight than basic LeNet-5. So I changed some details based LeNet-5. The specific structure of the CNN used in the report is as follows:

1. C1: is a convolution layer with 32 filters.

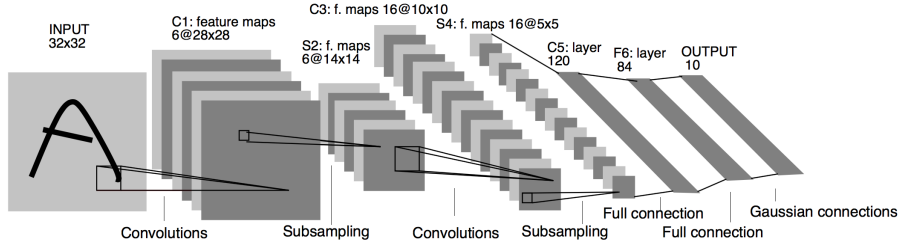


Fig 1. LeNet-5

2. S2: is a max pooling layer with a 2×2 kernel size.
3. C3: is a convolution layer with 62 filters
4. S4: is a max pooling layer with a 2×2 kernel size
5. F5: is a full connected layer with 500 units.
6. Output layer: the number of unit is same as dataset classes.

I used Python with Keras library in the report. The codes of building CNN are as follow:

```
model = Sequential()
    model.add(Conv2D(32, kernel_size=(kernel_size, kernel_size),
                    padding='same', activation='relu', input_shape=
                    input_shape))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Conv2D(64, (kernel_size, kernel_size), padding='same',
                    , activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Flatten())
    model.add(Dense(500, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))

    model.compile(loss=keras.losses.categorical_crossentropy,
                  optimizer=keras.optimizers.
                  SGD(), metrics=['accuracy'])
```

the parameter kernel_size in this report was setted to [3, 5, 7, 9, 11].

2.2 Datasets

I selected 4 iamges datasets in this report: CIFAR-10 [3], 17FLOWERS [6], INDOOR67 [7] and TEXTURES [1]. The details of these datasets are as follows:

- CIFAR-10: consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.
- 17FLOWERS: is a 17 category flower dataset with 80 images for each class. The pixel of images is around 500×500 .
- INDOOR67: contains 67 Indoor categories, and a total of 15620 images. The number of images varies across categories, but there are at least 100 images per category.

- **TEXTURES**: consisting of 5640 images, organized according to a list of 47 terms (categories) inspired from human perception. There are 120 images for each category. Image sizes range between 300×300 and 640×640 , and the images contain at least 90% of the surface representing the category attribute.

I choosed the 4 above image datasets because their categories follows the law from broad to fine.

3 Results

3.1 CIFAR-10

There are the several loss curves in Fig 2 with using same CNN architecture but different convolutional kernel size.

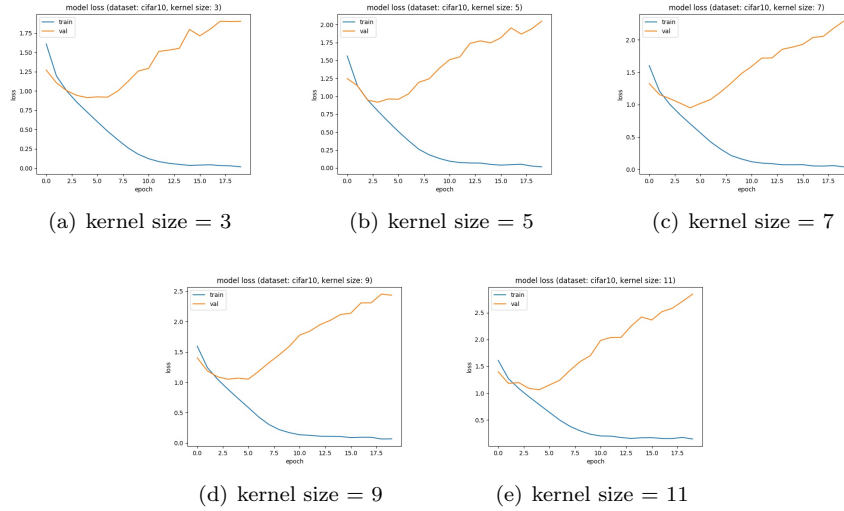


Fig 2. Loss with CIFAR-10

We can see that all of CNNs with different kernel size have been overfitting.

3.2 17FLOWERS

There are the several loss curves in Fig 3 with using same CNN architecture but different convolutional kernel size.

We can see that all of CNNs with different kernel size have been overfitting.

3.3 INDOOR67

There are the several loss curves in Fig 4 with using same CNN architecture but different convolutional kernel size.

We can see that all of CNNs with different kernel size have been overfitting.

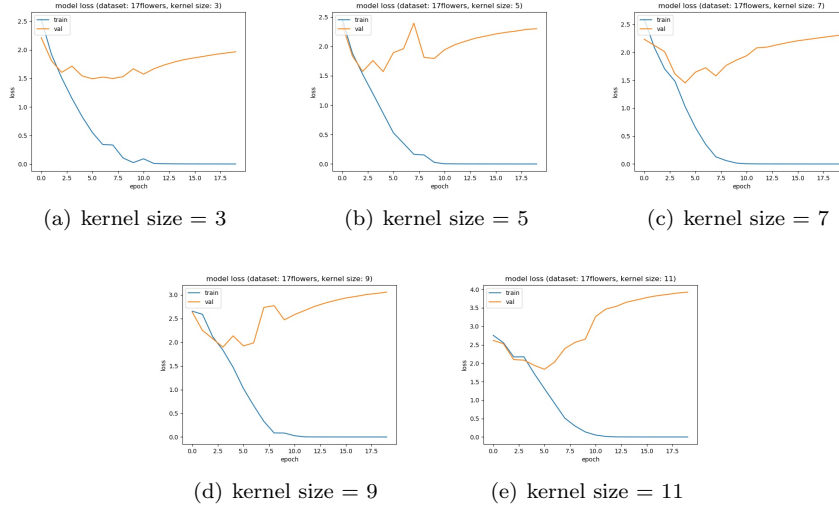


Fig 3. Loss with 17FLOWERS

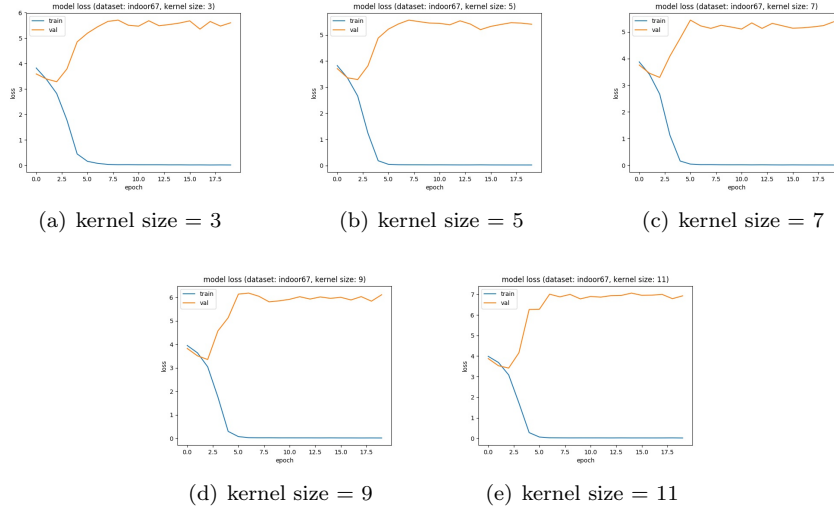


Fig 4. Loss with INDOOR67

3.4 TEXTURES

There are the several loss curves in Fig 5 with using same CNN architecture but different convolutional kernel size.

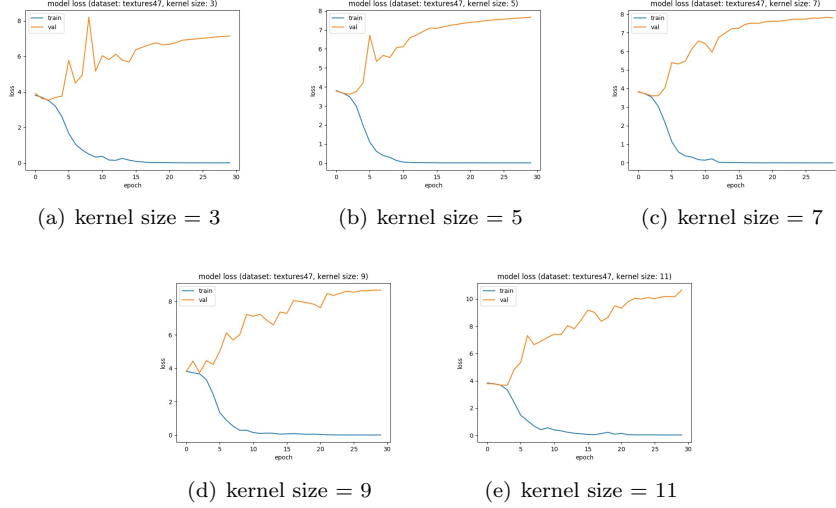


Fig 5. Loss with TEXTURES

We can see that all of CNNs with different kernel size have been overfitting.

4 Discussion

I selected the minimum loss from every loss figure to denote the best effect of CNN with these kernel size. The minimum loss with different kernel size is drawn in Fig 6:

In the CIFAR-10 figure, the loss increases from 0.9145 to 1.0632 with kernel size increasing. When kernel size of convolutional layer is equal to 3(or 3×3), CNN has a minimum loss that I called optimum kernel size. When we focus on 17FLOWERS figure, there is a little different from CIFAR-10 because optimum kernel size is 5(or 5×5) but not 3(or 3×3). Then looking at INDOOR67, its optimum kernel size is up to 7(or 7×7), same as the TEXTURES dataset.

From the above phenomenon, we can draw conclusions that different dataset has different optimum kernel size though their CNN architecture are same. I have some explanations for the phenomenon as follows:

- **regularization:** bigger kernel size means more weights and the CNN model will become more complex. In the report, all of these CNN models with different datasets and kernel sizes are have been trained to overfitting. So according to Occam's rezor, smaller kernel size will increase the model generation. That is to say smaller kernel size can make contributions to reduce validation loss.
- **category of dataset:** We can distinguish the class of images in CIFAR-10 easily. But it may be a little difficult to identifying the category of pic-

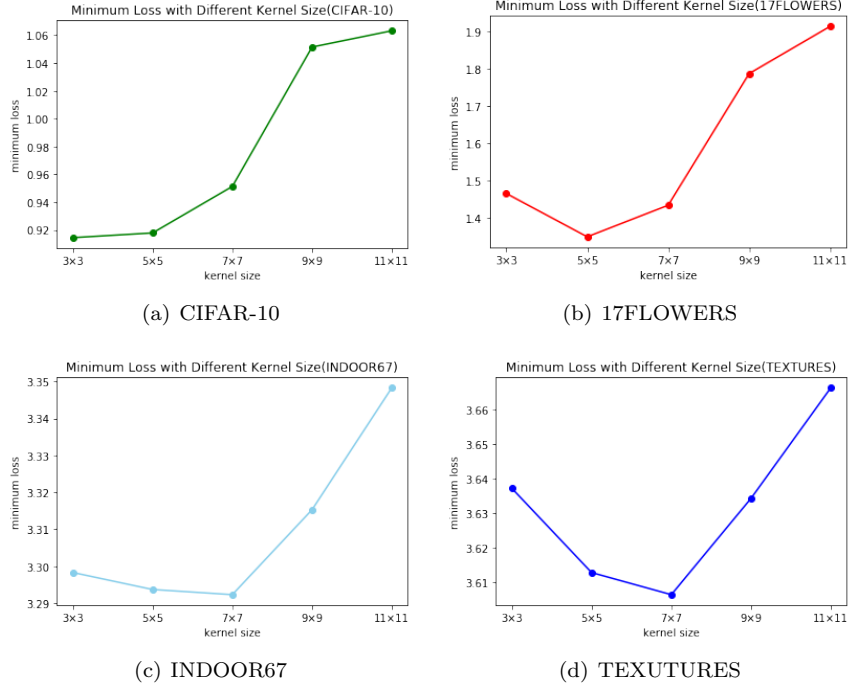


Fig 6. minimum loss with different kernel size

tures in TEXTURES dataset. When the category of dataset is widely distributed like CIFAR-10 from horse to airplane, small kernel size is enough to capture information from dataset. If the classes between dataset are very closed each other, we need bigger kernel size to extract more information to recognize them.

On the one hand, bigger kernel size will increase model complexity leading to higher validation loss. On the other hand, bigger kernel size can capture more information from dataset especially for the datasets which categories are very closed so that the validation loss could be reduced. Optimum kernel size for a specific dataset should achieve balance between the two.

5 Conclusions

In the report, I do some research on the convolutional kernel size. There is a connection between optimum kernel size and different datasets for same CNN architecture. Small kernel size cannot extract enough information but bigger kernel size will increase model complexity. We need to balance them to find optimum kernel size.

Reference

- [1] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [3] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- [4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [5] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [6] M-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.
- [7] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009.
- [8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.