Research on the Extraction of Image Edge Information in Convolutional Neural Networks

Guanru Zou, Yulin Luo*, Zefeng Feng
Beijing Institute of Technology Zhuhai, China

*Corresponding author e-mail: luoyulin@stu.shmtu.edu.cn

Abstract. Convolutional neural network is an important neural network model in deep learning and a common algorithm in computer vision problems. From the perspective of practical application scenarios, this paper studies whether padding in convolutional neural network convolution layer weakens the image edge information. In order to eliminate the background factor, this paper select MNIST dataset as the research object, move the 0-9 digital image to the specified image edge by clearing the white area pixels in the specified direction, and use OpenCV to realize bilinear interpolation to scale the image to ensure that the image dimension is 28×28. The convolution neural network is built to train the original dataset and the processed dataset, and the accuracy rates are 0.9892 and 0.1082 respectively. In the comparative experiment, padding cannot solve the problem of weakening the image edge weight well. In the actual digital recognition scene, it is necessary to consider whether the core recognition area in the input image is at the edge of the image.

Keywords: Convolutional Neural Network; Padding; MNIST Dataset; OpenCV.

1. Introduction
Deep learning is a new field with rapid development in recent ten years, which has attracted more and more researchers' attention. With the rapid rise of deep learning, the most important one is artificial neural network. Based on the physiological research results of the brain, the purpose is to simulate some mechanisms and mechanisms of the brain and realize some functions [1]. Using artificial neural network, scientists have created multi-layer perceptron, but it is very limited because it can only be used to deal with linear problems, so neural network has been ignored by scientists. It was not until 2012 that the emergence of convolutional neural network [2] broke the previous situation. Convolutional neural network was first used in computer vision. It is mainly composed of convolution layer, pooling layer and full connection layer. This paper mainly discuss whether the pooling operation in the convolution layer balances the weight distribution between the image edge and the image center.

2. Dataset Introduction
MNIST is a dataset of handwritten digits for in-depth learning built by Yann LeCun based on the National Institute of standards and technology. MNIST is composed of 70000 gray-scale images handwritten by different people, each of which is written with any number from 0 to 9. There are 60000
training images and 10000 test images, and the image data is 28 × 28 pixels grayscale image. The model built in this paper is to classify 10 numbers according to the pictures in the dataset. [3]

3. Conversational Neural Network

3.1. Overview of Convolutional Layer

The convolutional layer [4] is a key layer of the convolutional neural network. Its main function is to extract features of various scales in the image through different convolution kernels and to ensure that the image size does not change. In functional analysis, convolution generates a third function through two discrete or continuous functions. Convolution in image processing is usually a discrete form.

The convolution operation in the convolutional neural network is completed by the convolution kernel. The convolution kernel is also called a filter. Assuming that the length and width of the convolution kernel are \( k_h \) and \( k_w \) respectively, it is called \( k_h \times k_w \) convolution. Common convolution methods include single-channel and multi-channel convolution, 3D convolution, transposed convolution, and 1×1 convolution. Taking single channel convolution as an example, the convolution process is to use convolution kernel \( W \) (size n × m) scan the element \( x \) of the image pixel matrix row by row and column by column, multiply each weight \( w \) in the convolution kernel by each elements \( x \) in the matrix, and then add the products to obtain a dimension reduced matrix.

![Figure 1. Convolution process](image)

The specific calculation formula is as follows:

\[
z = w_1 x_1 + w_2 x_2 + \cdots + w_{nm} x_{nm} = \sum_{k=1}^{nm} w_k x_k = W^T X
\]

Assuming that the dimension of the single-channel image is \( n \times n \), the size of the convolution kernel is \( f \times f \), the step size is \( s \), and the padding size is \( p \), the size of the output image pixel matrix after convolution is:

\[
\left(\frac{n + 2p - f}{s} + 1\right) \times \left(\frac{n + 2p - f}{s} + 1\right)
\]

3.2. Padding of Image Convolution

Since the dimensionality of the input image will be reduced after each convolution operation, if the model network layer is very deep, the image will become smaller and smaller, causing a lot of loss of important information; in addition, due to the volume in the unit step size in the product calculation, some edge pixels will only be scanned once, and the edges of the input image play a small role, so most of the information at the image boundary is lost. This is because the pixels on the edge are never in the center of the convolution kernel, and the convolution kernel cannot extend beyond the edge area.

In digital picture classification, this situation is unacceptable, so padding is proposed in convolution, that is, extra pixels are added to the edge of the picture matrix (this pixel is usually 0, so it is often called
"0 padding"). In this way, the convolution kernel makes the edge pixels participate in the center of the convolution kernel in the process of moving and scanning.

4. Issues Raised
In real life, people cannot control that the numbers written by the user on the panel or on the paper are in the middle of the entire picture. Therefore, it is proved whether the matrix supplement 0 at the edge of the picture is effective in solving the weakening problem of edge information, which affects the accuracy and generalization ability of artificial intelligence to identify numbers.

Based on the application scenario of digital recognition, this paper discusses whether the padding operation in the convolutional layer is beneficial to the information extraction of the edge of the picture.

5. Related Works
This paper uses the MNIST dataset to test for two reasons: First, in actual application scenarios, the numbers people write on panels or paper may not be in the middle, may be in the corner of the entire sampled picture. In the MNIST dataset, the background is uniform, and only a single number appears in a single picture. Such picture features are very helpful to the realization of our experimental purpose. In more complex scenarios, the experiment in this paper is used as a baseline. Make different adjustments according to the actual situation.

5.1. Adjust the Original MNIST Dataset
In the original MNIST dataset, the numbers are in the middle of the picture, so it is necessary to move the numbers and move the numbers to different corners. In this operation, this paper follows the principle of maximizing the information that each number falls on the edge of the picture and chooses a different corner for each number. For example, move the numbers 3 and 9 to the upper right corner of the picture, move the number 4 to the upper right corner, move the numbers 5 and 6 to the lower left corner of the picture.

This paper uses NumPy and OpenCV to move the number to the specified corner. The steps are explained here with the designated corner as the upper right corner. The first step is to clear the blank space below the picture (In the MNIST dataset, the background of each images are white, so the pixel is 0. Using this image feature to clear the blank). The second step is to clear the left margin of the picture. The third step, after the first and second steps, the blank area around the images are removed. Therefore, the blank area needs to be supplemented in the lower left corner so that the number is in the upper right corner of the picture. Theoretically, as much area pixels are removed as necessary, but in order to avoid some errors, the picture is not restored to the original 28×28 size, so in this paper, the bilinear interpolation in OpenCV is used to scale the image to 28×28.

Figure 2. The picture on the left is the picture of MNIST dataset, and the picture on the right is the adjusted picture

5.2. Bilinear Interpolation [6]
Supposing the mapping function of the image is f, at any point (i, j), the pixel value is \( y = f(i, j) \), where \( 0 \leq i \leq w, 0 \leq j \leq h \), w, h are the images respectively width and height. Now given \( f(x_1, y_1), f(x_2, y_1), f(x_2, y_2) \), solve \( f(x, y) \).

Linear interpolation in the x axis direction:
\[
\frac{f(x_2, y_1) - f(x_1, y_1)}{x_2 - x_1} = \frac{f(x_2, y_2) - f(x_1, y_2)}{x_2 - x_1} = \frac{f(x, y_2) - f(x, y_1)}{x - x_1}
\]

Get:

\[
f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} f(x_1, y_1) + \frac{x - x_1}{x_2 - x_1} f(x_2, y_1)
\]

\[
f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} f(x_1, y_2) + \frac{x - x_1}{x_2 - x_1} f(x_2, y_2)
\]

Linear interpolation in the \( y \) axis direction:

\[
\frac{f(x, y_2) - f(x, y_1)}{y_2 - y_1} = \frac{f(x, y) - f(y, y_1)}{y - y_1}
\]

Get:

\[
f(x, y) = \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2)
\]

Finally, simplify to the form of matrix transformation:

\[
f(x, y) = \frac{y_2 - y}{y_2 - y_1} \left[ f(x_1, y_1) \left( x_2 - x \right) \right] + \frac{y - y_1}{y_2 - y_1} \left[ f(x_2, y_2) \left( x_2 - x \right) \right] + \frac{y_2 - y}{y_2 - y_1} \left[ f(x_1, y_1) \left( x - x_1 \right) \right] + \frac{y - y_1}{y_2 - y_1} \left[ f(x_2, y_2) \left( x - x_1 \right) \right]
\]

If within a unit pixel grid, there are \( x_2 - x_1 = y_2 - y_1 = 1 \), then:

\[
f(x, y) = \left[ f(x_1, y_1) \left( x_2 - x \right) \right] + \left[ f(x_2, y_2) \left( x - x_1 \right) \right]
\]

5.3. Design Convolutional Neural Network Architecture

The purpose of this paper is not to improve the accuracy of digital recognition, so our network structure is mainly built around the core layer. The key design lies in the complete convolution layer in the convolution neural network [7]. The following table is the parameters of each network layer used in this experiment.

| Layer (type)                  | Output Shape | Param |
|-------------------------------|--------------|-------|
| Conv2d (Conv2D)              | (None,28,28,16) | 416   |
| Max_pooling2d (MaxPooling2D) | (None,14,14,16) | 0     |
| Conv2d_1 (Conv2D)            | (None,14,14,36) | 14436 |
| Max_pooling2d_1 (MaxPooling2D)| (None,7,7,36) | 0     |
| Dropout                      | (None,7,7,36) | 0     |
| Flatten                      | (None,1764)  | 0     |
| Dropout_1                    | (None,1764)  | 0     |
| Dense                        | (None,10)    | 0     |

5.4. Methodology

Using the above network architecture, the original MNIST dataset and the MNIST dataset adjusted by padding in different ways are trained separately. Test on the test set and compare the accuracy of the two.
6. Results
In the same GPU environment, the same network structure, and the same test set, we get the accuracy of 0.9892 in the original MNIST dataset, and the accuracy of 0.1082 in the adjusted MNIST dataset.

Experiments have proved that adding 0 to the edge of the image matrix cannot solve the problem of numbers being at the edge of the picture. In digital recognition scenarios in real life scenarios, you cannot add zeros to the edge of the obtained image matrix alone. The subject (number) needs to be centered, so that the requirements of the actual application scenario of artificial intelligence can be met.

References
[1] Zhang Qiaochao, Zeng zhaobing. Overview of artificial neural network [J]. JOURNAL OF LIAONING ECONOMIC VOCATIONAL AND TECHNICAL COLLEGE LIAONING ECONOMIC MANAGEMENT CADRE COLLEGE, 2010, 000 (004): 68-69.
[2] Technicolor T, Related S, Technicolor T, et al. ImageNet Classification with Deep Convolutional Neural Networks [50].
[3] Guo Mengjie, Yang Mengzhuo, Ma Jingjiu. MNIST dataset recognition model based on keras [J]. Modern information technology, 2019 (14).
[4] Zhou Feiyan, Jin Linpeng, Dong Jun. review of convolutional neural networks [J]. Journal of computer science, 2017 (6).
[5] Chang Liang, Deng Xiaoming, Zhou Mingquan, et al. Convolutional neural network in image understanding [J]. Journal of automation, 2016, 42 (9).
[6] Mo K. Spatial Transformer Network.
[7] Long Liangqu. Tensorflow Deep Learning: Deep Understanding of Artificial Intelligence Algorithm Design [M]. Peking University Press.