Comparing Convolutional Neural Networks on Handwritten Digits

Bochen Dong, Luke Yao, Walter Mao
{b5dong, h22yao, w7mao}@uwaterloo.ca
University of Waterloo
Waterloo, ON, Canada

Introduction

Convolutional neural networks (CNNs) are a special type of neural networks designed notably for image recognition. Unlike other neural networks, CNNs have convolutional layers with filters that detect patterns in images. These convolutional layers are the basis of the CNN. Since LeNet, the pioneering 7-level convolutional neural network by Yann LeCun in 1998, the world has seen major developments in CNNs such as ALEXNET in 2012, GOOGLENET in 2014, and VGGNet in 2014.

In our project, we would like to compare different CNN architectures such as ALEXNET, VGGNet, and GOOGLENET on MINST data set, a large database of handwritten digits. After trying each of the architectures, we would compare each of their results based on training set size, test error, and efficiency. We would then perform data augmentation to compare based on robustness. In addition, we will observe the properties of each architecture that lead it to its results. Finally, we will try to implement our own program to achieve a similar result. CNNs are used today for many applications such as facial recognition and document analysis.

CNNs are used today for many applications such as facial recognition and document analysis. From the iPhone's FaceID to Google Translate's instant camera translation, CNNs have changed the way we live our lives. If our project is successful and we are able to pinpoint features of a CNN that make it effective, we can create more powerful CNNs. These power CNNs can then be applied to impactful applications such as those mentioned.

Related Work

AlexNet[1]**:** On the ImageNet LSVRC-2010, in the classification task of a total of 1.2 million high-resolution images containing 1000 categories, AlexNet's top-1 and top-5 error rates on the test set were 37.5% and 17.0%. AlexNet has 600 million parameters and 650,000 neurons, including 5 convolutional layers, some layers are followed by a max-pooling layer, and 3 fully connected layers. In order to reduce overfitting, dropout is used in the fully connected layer, and Use ReLU function as activation function. The main method it used was the Non-linear ReLU function. At that time, the standard neuron activation function was the tanh() function. This saturated nonlinear function is much slower than the unsaturated nonlinear function during gradient descent. Therefore, using the ReLU function in a 4-layer convolutional network as the activation function in AlexNet to achieve a training error rate of 25% on the CIFAR-10 dataset is 6 times faster than using the tanh function under the same network and the same conditions. Moreover, Local Response Normalization was being used to reduce AlexNet's top-1 and top-5 error rates by 1.4% and 1.2%, respectively and the Overlapping Pooling scheme reduces the top-1 and top-5 error rates by 0.4% and 0.3%. Last but not least, to prevent overfitting in the algorithm, dropout and data augmentation were the two main methods that were being used.

VGG[2]**:** The main work of the network is to prove that increasing the depth of the network can affect the final performance of the network to a certain extent. VGG has two structures, namely VGG16 and VGG19. There is no essential difference between the two, but the network depth is different. An improvement of VGG16 compared to AlexNet is to use several consecutive 3x3 convolution kernels to replace the larger convolution kernels in AlexNet (11x11, 7x7, 5x5). For a given receptive field, the use of stacked small convolution kernels is better than the use of large convolution kernels, because multiple nonlinear layers can increase the depth of the network to ensure that learning is more complicated Mode, and the cost is relatively small. While VGG consumes more computing resources and uses more parameters, resulting in more memory usage. Most of the parameters are from the first fully connected layer. In conclusion, the author found through the network A and A-LRN that the local response normalization (LRN) layer used by AlexNet had no performance improvement. Moreover, with the increase of depth, the classification performance will also get better. Lastly, the author found that multiple small convolution kernels have better performance than single large convolution kernels. The author did an experiment to compare B with one of his shallower networks not in the experimental group. The shallower network uses conv5x5 instead of B's two conv3x3, and multiple small convolution kernels perform better than one single large convolution kernel.

GoogleNet Structures such as AlexNet and VGG all ob-

tain better training results by increasing the depth (number of layers) of the network, but increasing the number of layers could also bring many negative effects, such as overfit, gradient disappearance, gradient explosion, etc. Inception is proposed to improve training results from another perspective: it can use computing resources more efficiently, and more features can be extracted under the same amount of calculation, thereby improving training results.

Going deeper with convolutions[3]**:** The author proposes that the fully connected structure needs to be converted into a sparsely connected structure. There are two methods for sparse connection, one is spatial sparse connection, which is the traditional CNN convolution structure: only a certain part of the input image patch is convolved, rather than convolving the entire image, sharing the parameter reduces the number of total parameters and reduces the amount of calculation; another method is to sparsely connect in the feature dimension and gather the strongly related features together, and the convolution of each size only outputs 256 features Part of this, this is also a sparse connection. Today's computers are very inefficient in calculating sparse data. Even using the sparse matrix algorithm is not worth the cost. Using the sparse matrix algorithm for calculation will reduce the amount of calculation, but will increase the intermediate cache.

Network in Network[4]**:**In this article, we see that the Inception Module can superimpose more convolutions in the same size receptive field, and can extract richer features It is proposed to use the Global Average Pooling (GAP) layer to replace the fully connected layer. The specific method is to average all the points on each feature. If there are n features, output n averages as the final softmax input, which has many benefits. Firstly, regularizing the data on the entire feature can prevent over-fitting. Secondly, we do not need a fully connected layer anymore, which reduces the number of parameters of the entire structure. By applying this improvement, the possibility of overfitting is reduced. Thirdly, we do not need to pay attention to the size of the input image anymore, because no matter how the input is, the same averaging method is going to be applied. While the traditional fully connected layer must choose the number of parameters according to the size, and has no universal Sex.

For Alexnet, we are going to implement the Relu function, overlapping Pooling, LRN, data augmentation and dropout

For VGG, we are going to implement 3x3 convolution kernel and a 16-layer network

For GoogleNet, we are going to implement Global Average Pooling and Inception network

Methodology

To tackle the problem, we are going to construct different Network structures of CNN, such as Alexnet, VGG and GoogleNet. By applying data argumentation on the data set by adding some Gaussian noise and rotating the image, we can then measure the training accuracy and loss on different architectures. In this part, we are planning to use Python package Keras to import the data set, Skimage to add Gaussian Noise to the data set; Tensorflow when constructing the structure and Pillow to perform the data argumentation.

We plan to create different kinds of CNNs. For ALexnet, we are going to implement the Relu function, overlapping Pooling, LRN, data augmentation and dropout. For VGG, we are going to implement a 3x3 convolution kernel and a 16-layer network, and for GoogLeNet, we are going to implement a Global Average Pooling and Inception network. We also plan to create different kinds of data argumentation by using Gaussian Noise and rotating the image.

Results

ALexnet uses dropout, ReLU and other methods to improve performance. VGG proves that the depth affects the performance of the network. However, the deeper network structure brings more calculations. GoogLeNet solves this problem in certain ways using an inception network instead of a fully connected network to capture receptive fields of different sizes increases the efficiency of the network without affecting accuracy. To summarize, the depth of the network is getting larger and fewer convolution kernels can be used to ensure better accuracy. In order to consider the calculation efficiency, the modular structure can reduce the design space of the network. The use of inception network in the module can reduce the amount of calculation

References

(a) Krizhevsky, Alex, et al. ImageNet Classification with Deep Convolutional Neural Networks. University of Toronto.

(b) Simonyan, Karen, and Andrew Zisserman. Very Deep Convolutional Networks For Large-scale Image Recognition. Visual Geometry Group, Department of Engineering Science, University of Oxford.

(c) Szegedy, Christian, and Wei Liu. Going Deeper with Convolutions. Google Inc.

(d) Lin, Min, and Qiang Chen. Network In Network. National University of Singapore, Singapore.