Convolution Neural Network Analysis

XinyuYang*
SJTU Mathematics SJTU
XinyuYang-15221861038@163.com

Abstract

This paper analyzes four different CNN include LeNet AlexNet NiN and GoolgNet. we use Fashion-mnist data set to train CNN moudles, and get results to analyse the performance of CNN.

1 Introduction

This paper includes five parts. The first part is data preprocessing, this part will introduce Fashion-Mnist data set and how to make data trainable.The second part is the comparison of LeNet and AlexNet,and introduce a simplified moudle of AlexNet to deal with Fashion-Mnist data set. The next part is about VGG, and its comparison to LeNet and AlexNet. In the fourth part we will improve Googlnet and show these new moulds' proformaces compared to GoolgNet. Finally we will get some conclusions.

2 Data preprocessing

In this paper we use Fashion-Mnist data set to train CNN moudles, this data set has ten kinds of dress include clothes shoes and others.And we devide the data set into train data set and test data set. First of all , we load the data set by a function ,then we transform all the data into Tensor, so that we can train them on GPU. Then we will train our moudles on train data set, and test the finished moudles on test data set to get the preformace of the moudles

The following are ten example pictures and their lables.

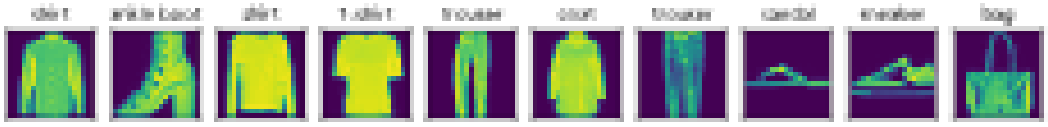

Figure 1: Some examples

3 Comparison of LeNet and AlexNet

In this section we will introduce 2 types CNN :LeNet and AlexNet.We will analyze and compare the structural differences between LeNet and AlexNet from the perspective of data set preprocessing, activation function use, training method improvement and model structure change, and try to explain why alexnet has superior processing performance for computer vision tasks.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

2nd Term Project for Dive into Deep Learning in 14 Days (Boyu Edu. 2020), Shanghai, China.

3.1 LeNet

Lenet as an early convolution neural network used to recognize handwritten digital images, shows that convolution neural network can achieve the most advanced results of handwritten digital recognition at that time through gradient descent training. As shown in the figure below:

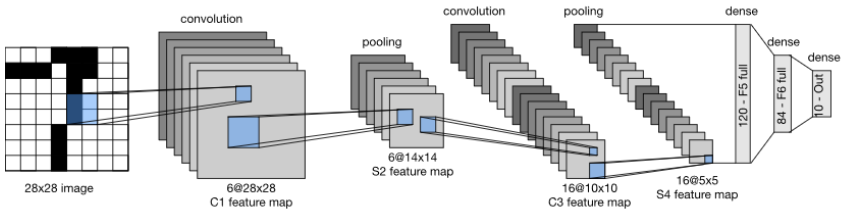

Figure 2: LeNet

The model structure of lenet is divided into two parts: convolution layer block and full connection layer block

- The convolution layer keeps the input shape, so that the correlation of image pixels in the two directions of height and width can be recognized effectively. And the same convolution kernel and input at different positions can be calculated repeatedly through the sliding window, so as to avoid too large parameter size.

- Full connected layer blocks flatten each sample in the output of convolution layer blocks, that is, the input shape will become two-dimensional, in which the first dimension is the sample in small batch, and the second dimension is the vector representation of each sample after flattening, so as to classify.

3.2 AlexNet

In 2012, alexnet came out, using 8-layer convolutional neural network, and won the Imagenet 2012 image recognition challenge with great advantages.

The design concept of alexnet is very similar to that of lenet, but compared with the relatively small lenet, alexnet contains five layers of convolution and two layers of full connection hidden layer, as well as one full connection output layer, and the model parameters are greatly increased. Due to the limitation of early memory, the earliest alexnet used the design of double data flow, so that a GPU only needs to deal with half of the model. As shown in the figure below:

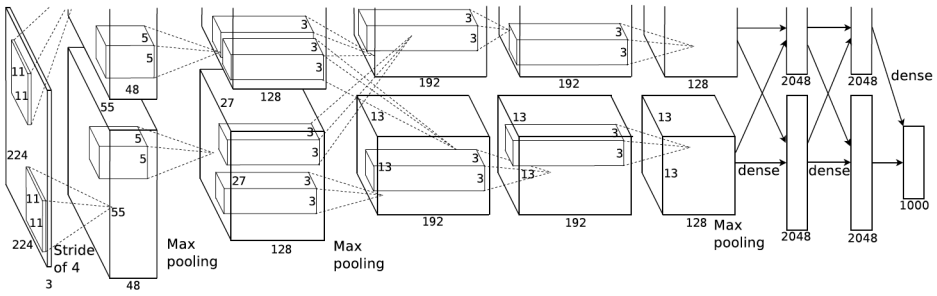

Figure 3: AlexNet

For the first time, alexnet has proved that the features learned by neural network in the end-to-end way can surpass the features of manual design, thus breaking the previous shape of computer vision research.

3.3 Comparison

3.3.1 Data preprocessing

LeNet The convolution neural network in lenet 5 is mainly for handwritten characters, and lenet's data preprocessing is mainly to separate handwritten characters from their neighbors, that is, segmentation. The main method is to use heuristic method to generate a large number of possible segmentation results, and then find the best one based on the score. In this case, the quality of segmentation mainly depends on the heuristic method of generating segmentation and how to find the most matching segmentation results. Here, it is important to manually mark the incorrect segmentation. Obviously, this way is not only time-consuming and laborious, but also very difficult. For example, should the right side of 8 be marked as 3 or 8? In this article, Lecun proposes two ways to solve this problem. One is to optimize the global loss function. The second is to identify every possible segmentation result and select the most central character.

AlexNet Compared with lenet, alexnet network is more complex and can process larger data sets. Alexnet's data preprocessing adopts the means of data enhancement, such as translation transformation, flipping, clipping and color change, so as to further expand the data set to alleviate over fitting:

(1) Random translation: During training, the images of 256×256 are randomly translated to 224×224, and then allowed to flip horizontally, which is equivalent to multiplying the samples to $\left((256-224)^2\right) \times 2 = 2048$.

(2) During the test, the top left, the top right, the bottom left, the bottom right, and the middle were crossed five times, then turned over, a total of 10 crosses, and then the results were averaged. The author says that without random cross-over, large networks are basically under material over fitting.

(3) PCA is applied to RGB space, and then a (0,0.1) Gaussian perturbation is applied to the principal component. As a result, the error rate dropped by 1

3.3.2 Activation function

LeNet Using sigmoid function as activation function can compress the continuous real value of input between 0 and 1. But sigmoid function has two disadvantages: first, when the input is very large or very small, there will be saturation. The gradient of these neurons tends to 0. If the initial value is large, the gradient needs to be multiplied by a sigmoid in the back propagation The second is that the output mean value of sigmoid function is not 0, which will lead the neurons in the later layer to get the non-zero mean signal output from the previous layer as input.

AlexNet Using the relu function instead of sigmoid function requires a threshold value to get the activation value, which greatly reduces the amount of calculation and converges faster.

3.3.3 Improvement of training methods

Alexnet proposed the LRN layer, which creates a competition mechanism for the activities of local neurons, makes the response of larger pairs become relatively larger, and suppresses other neurons with smaller feedback, thus enhancing the generalization ability of the model.

3.3.4 Changes in model structure

LeNet It is composed of two convolution layers, two average pool layers and three full connection layers. The network structure is complete and simple.

AlexNet There are more layers in the network, including 5 layers of convolution and 3 Maximum pooling layers, 2 layers of full connection hidden layer, and 1 layer of full connection output layer.

Compared with lenet, the average pooling layer is changed into the maximum pooling layer. Using dropout technology, the output of each hidden layer neuron is set to 0 with a probability of 0.5, and the neuron being dropout will not participate in forward propagation and back propagation. Each time a sample is input, it is equivalent to that the neural network tries a new structure, but all these structures share the weight. Because neurons can't depend on other specific neurons, this technique reduces the complex adaptive relationship of neurons. Because of this, the network needs to be forced to learn more robust features, which are useful when combining some different random subsets of other neurons. In the test, we only multiply the output of all neurons by 0.5, which is a reasonable approximate method for obtaining the geometric mean of the predicted distribution generated by exponential dropout network.

3.3.5 Summary

Alexnet network is larger, more effective data preprocessing method, activation function, training mode and model structure are adopted, which can effectively prevent over fitting, and can use GPU high-speed parallel computing, which shows superior processing performance in computer vision task.

3.4 Simplify AlexNet

Alex net may be too complex for the fashion MNIST data set. We simplified the model to make the training faster and ensure that the classification accuracy does not decline significantly (no less than 85%). Next, we will give the simplified structure, the saved training time and the decline accuracy and other related indicators, and summarize and analyze them in the form of tables.

Simplified AlexNet1: (reduces one volume and one full connection layer)

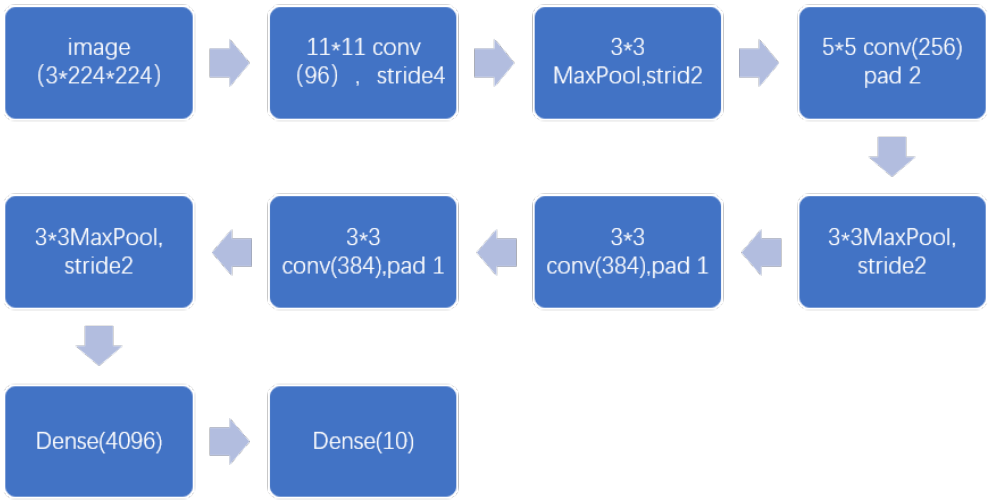

Figure 4: AlexNet1

Training results comparsion:

The accuracy of the simplified network test set is slightly higher than that of the original network. This is because the data set of fashionminst is not complex. Reducing the network complexity properly can reduce over fitting, and the generalization degree of network training is higher. Because of the reduction of two layers, the average training time per round is reduced by about 20 seconds.

Table 1: AlexNet

Part				
epoch	loss	train acc	test acc	time
1	0.6564	0.747	0.861	130.6sec
2	0.3413	0.873	0.885	132.9sec
3	0.2558	0.904	0.894	110.5sec

Table 2: AlexNet1

Part				
epoch	loss	train acc	test acc	time
1	0.5091	0.811	0.871	110.5sec
2	0.3005	0.886	0.891	109.7sec
3	0.2931	0.891	0.896	110.5sec

4 VGG

4.1 Comparison of LeNet AlexNet and VGG

LeNet and AlexNet introduce large-scale convolution kernels in the convolution module close to image input, such as 5×5 or 7×7 convolution window to capture a larger range of image information. Next, we will analyze whether the fixed design of each basic block of VGG will affect the coarse-grained information extraction of image, and compare the characteristic images output by different structure modules.

The fixed design of each basic block of VGG will not affect the coarse-grained information extraction of the image. The VGg network uses multiple 3×3 convolution kernels to stack, while the receptive field of two3×3 convolution kernels is equivalent to a 5×5 convolution kernel receptive field. Under the condition of ensuring the same receptive field, the VGG network has fewer parameters and increased nonlinearity, which makes the network more capable of learning features.

4.2 VGG parameter analysis

We change the height and width of image in fashion MNIST from 224 to 96, analyze the parameter changes of VGg network, and compare the influence of experimental indexes such as model training time and classification accuracy, and summarize and analyze in the form of table.

First of all we changed the parameter in the ordinary paper as follows:

$conv\_arch = ((1, 1, 4), (1, 4, 8), (2, 8, 16), (2, 16, 32), (2, 32, 32))$

$fc\_features = 32 \times 7 \times 7$

$fc\_hidden\_units = 256$

The results as follows:

Table 3: VGG,H,W=224,lr=0.00001

Part				
epoch	loss	train acc	test acc	time
1	1.9261	0.260	0.642	74.8sec
2	1.0502	0.598	0.698	75.1sec
3	0.8939	0.665	0.720	75.0sec
4	0.8243	0.690	0.734	75.1sec
5	0.7792	0.708	0.740	75.1sec
10	0.6578	0.753	0.764	75.0sec

Table 4: VGG,H,W=96,lr=0.0001

Part				
epoch	loss	train acc	test acc	time
1	0.9703	0.624	0.772	75.0sec
2	0.5821	0.781	0.813	74.7sec
3	0.4940	0.817	0.836	74.8 sec
4	0.4440	0.835	0.848	75.1sec
5	0.4105	0.848	0.856	75.1 sec

Through comparison, it is found that when the height width is 96 and the learning rate is 0.0001, the learning time is equivalent, the learning rate is lower and the learning accuracy is higher.

5 NiN and GoogLeNet analysis

5.1 NiN

The common design of lenet, alexnet and VGg introduced before is: firstly, space features are fully extracted by modules composed of volume layer, and then classification results are output by modules composed of full connection layer. Among them, alexnet and VGg's improvement on lenet mainly lies in how to widen (increase the number of channels) and deepen the two modules. The network in the network (NIN) puts forward another idea, that is, to build a deep network by connecting several small networks composed of convolution layer and "full connection" layer in series.

The input and output of convolution layer is usually four-dimensional array (sample, channel, height, width), while the input and output of full connection layer is usually two-dimensional array (sample, feature). If you want to connect the convolution layer after the full connection layer, you need to transform the output of the full connection layer into four dimensions. The 1×1 convolution layer can be seen as the full connection layer, where each element on the spatial dimension (height and width) is equivalent to a sample and the channel is equivalent to a feature. Therefore, NiN uses 1×1convolution layer to replace the full connection layer, so that the spatial information can be naturally transmitted to the later layer. The following figure compares the main differences in structure between NiN and alexnet and VGG networks.

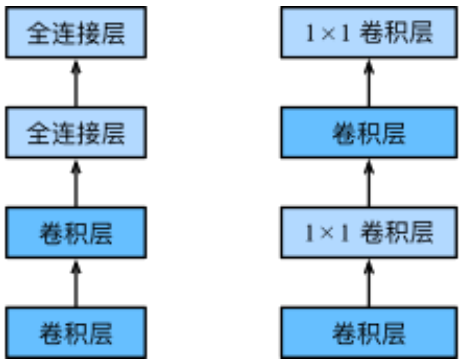

Figure 5: NiN

5.2 GoogleNet

In the 2014 Imagenet image recognition challenge, a network structure named googlenet showed great splendor. Although it pays homage to lenet in name, it is hard to see the shadow of lenet in network structure. Googlenet absorbed the idea of network series network in Nin, and made great improvement on this basis. The basic convolution block in googlenet is called the perception block, which is named after the movie inception of the same name. Compared with the upper Nin block, this basic block is more complex in structure, as shown in the following figure:

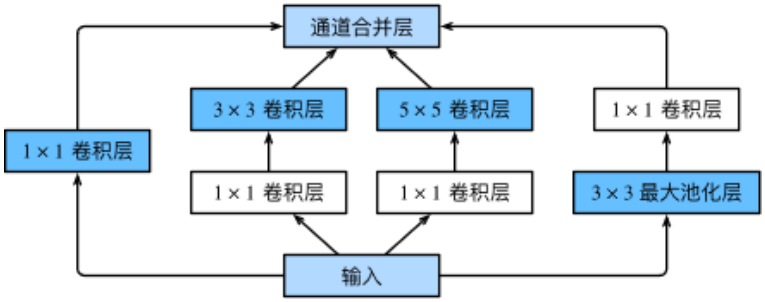

Figure 6: inception

5.3 Dimension analysis of model parameters

Since the convolution operation has fewer parameters, the parameters of a convolution kernel with the shape of $(c_i, c_o, h, w)$are $c_i \times c_o \times h \times w$, independent of the width and height of the input image. If the input and output shapes of a convolution layer are $(c_1, h_1, w_1)$and(c_2, h_2, w_2) , the number of parameters is$c_1 \times c_2 \times h_1 \times w_1 \times h_2 \times w_2$ if full connection layer is used for connection. Using the convolution layer can process larger images with fewer parameters.

NiN uses 1×1 convolution kernel to replace the full connection layer, which greatly reduces the parameter size. GoogleNet has only one full connection layer, and the size of the full connection layer is 1000; in contrast, alexnet and VGG have three full connection layers, with the size of 409640961000, respectively. NiN and Googlenet make full use of the characteristics of small convolution layer parameters, significantly reducing the model parameter size.

5.4 GoogleNet improvment

There are several subsequent versions of googlenet, including adding the batch normalization layer, adjusting the perception block and adding the residual connection. We will implement and run them, then observe the experimental results and summarize and analyze them in the form of tables.

The following are results:

(1) Original GoogleNet:

Table 5: GoogleNet

Part				
epoch	loss	train acc	test acc	time
1	1.0111	0.607	0.793	615.1 sec
2	0.3862	0.858	0.857	614.6 sec
3	0.3285	0.876	0.884	614.2 sec
4	0.2912	0.892	0.891	615.8 sec
5	0.2644	0.901	0.905	613.4 sec

(2) Add batch normalization layer GoogleNet-BN:

Table 6: GoogleNet-BN

Part				
epoch	loss	train acc	test acc	time
1	0.4528	0.834	0.874	962.7 sec
2	0.3070	0.889	0.903	960.8 sec
3	0.2626	0.904	0.890	961.4 sec
4	0.2305	0.916	0.900	961.1 sec
5	0.2153	0.921	0.909	962.4 sec

(3) Next,we adjust the concept block based on the paper,the large convolution layer is decomposed into small convolution layer to improve the calculation efficiency.We decompose a convolution convolution layer of 5×5 into two convolution layers of 3×3:

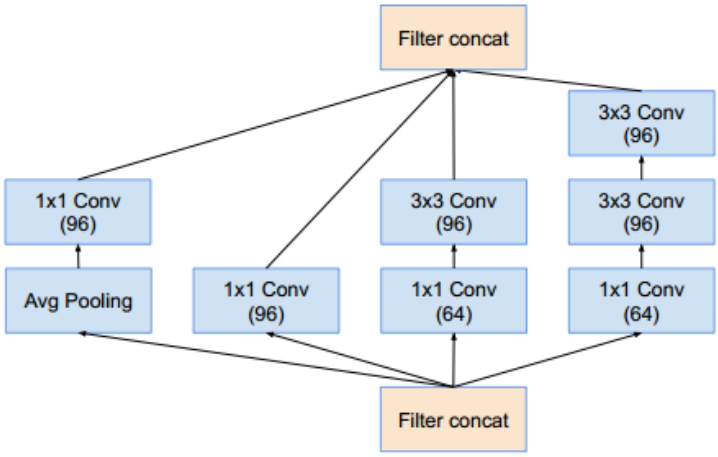

Figure 7: inception-A

Table 7: GoogleNet-Inception1

Part				
epoch	loss	train acc	test acc	time
1	0.9698	0.626	0.781	669.8 sec
2	0.4199	0.842	0.855	672.0 sec
3	0.3421	0.870	0.851	668.5 sec
4	0.3067	0.885	0.879	668.6 sec
5	0.2851	0.893	0.885	668.9 sec

(4) Improvement based on Inception-Resnet:

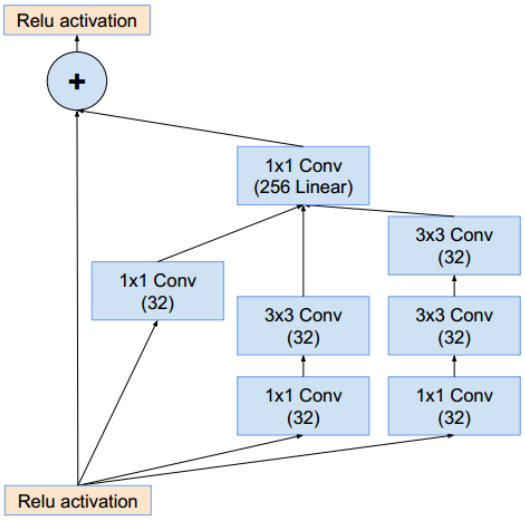

Figure 8: inception-ReaNet-A

Based on the structure, we get the following results:

Table 8: GoogleNet-Inception-ResNet

Part				
epoch	loss	train acc	test acc	time
1	0.9723	0.467	0.678	679.9 sec
2	0.4199	0.780	0.870	672.4 sec
3	0.3421	0.880	0.867	678.5 sec
4	0.3067	0.890	0.889	678.2 sec
5	0.2851	0.904	0.901	668.0 sec

References

[1] Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

[2] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2818-2826).

[3] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 4, p. 12).