DLLAB WEEK 2 Pre-Report

Hyunsoo Cha[1], Kwanghoon Sohn[2]
[1]Yonsei University. [2]Yonsei University.

1 Very Deep Convolutional Networks for Large-Scale Image Recognition

With the advent of AlexNet, the CNN Model began to gain attention in the field of Image Classification. VGG Network has better performances by using deep layers compared to AlexNet. For summary of a paper, Very Deep Convolutional Networks for Large-Scale Image Recognition, there are some keywords helping to understand the paper. First keyword is a large scale image recognition. In this paper, the network which the paper mentioned can recognize the large scale image. In the past, only 224×224 image could be recognized. Second keyword is the convolution filter. VGG Network has 3×3 convolution filter for reducing the number of parameters. AlexNet used 11×11 convolution filters. Last keyword is the depth of increasing. In a word, this model adopted a method that increases the number of layers of existing models by using convolution filters more deeply. After doing this, researchers got a good result in large-scale image recognition. In previous models, convolution filter size was slightly different, but in network, it is equally applied as 3×3 convolution filters. I describe the structure of the ConvNet and describe how the training method and classification task of the VGG model were carried out.

Architecture of ConvNet configuration is as follows. First, Input image has a fixed size of 224×224. Preprocessing for input image at training dataset only applies to subtracting RGB mean value which means the average of each value of R, G, and B held by pixels on the image. Second, ConvNet uses convolutional layer with 3×3 convolution filter. This is because 3×3 filter size is the minimization of receptive field on the network. Interestingly, the network also uses 1×1 filter. The convolution filter has a stride of 1, and padding is applied to the operation. The third is about the pooling layer. Max pooling is applied after the convolutional layer, and consists of a total of five max pooling layers. The pooling operation consists of 2×2 size and stride has 2. The fourth is the fully-connected layer. The abbrevation of fully-connected layer is FC Layer. The first two FC Layers have a total of 4096 channels. Finally, ReLU activation function is applied to all Hidden Layers. LRN techniques used in AlexNet did not have performance improvements when tested by VGGNet, and were not used due to increased memory consumption or computation.

The configuration of ConvNet is figured at Figure 1. In briefly, conv3-64 means that the number of channels in the convolutional layer is 64 and the size of the receivable field is 3. For simplicity, the ReLU function is not written separately. In the table, the deeper the configuration is as the alphabet A to E. The bolded layer means that it is added to the previous step. In addition, Figure 2 shows that the number of parameters does not increase even as the depth increases, but rather decreases.

Then I wonder why researchers used 3×3 convolution filters also known as small size filter. The reason is simpler than I thought. Using two 3×3 convolution filters has the same effect as using one 5×5 convolution filters. In other words, there are any difference between two filters. ReLU is a function that goes through when a convultion layer is performed. The use of more convolution filters makes it possible to use multiple nonlinear functions, ReLU. More nonlinear functions allow the decision function to be more identifiable. That is, to be more accurate. It can also reduce the number of parameters.

I will briefly summarize the test method to learn VGG Model. First, the paper explain how set up the hyper-parameters. First, Cost Function set Multinomial Logistic Regression Objective to Cross Entropy. The Mini-Batch set the Batch size to 256 and the Optimizer set the momentum to 0.9. Optimizer momentum is simply a methodology created to solve the problem of optimizing the gradient method, which is implemented to find the exact minimum. Regularization is set to L2 Regularization. The value is 5.10^{-4}. And researchers intentionally tried to prevent overfitting by setting dropout to 0.5. The running rate is 10^{-2}, which decreases by 0.1 as the validation

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 1: The configuration of ConvNet[2].

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Figure 2: The number of parameters with the kind of networks. The unit is million[2].

error rate increases. This allowed the user to record fewer epochs, although it has more depth and parameters than AlexNet. The reason for this is simply because the impulse regulation was first done by reducing the number of parameters. By replacing 7×7 convolution filter with 3×3 convolution filter, the number of parameters was reduced at the same time as having the same receptive field. In addition, we implemented it by learning VGG Layer A through pre-initialization first and then bringing the learned layers to the first learned model when constructing B, C, D models. It used the first four convolutional layers of its first model, A model, and the last three fully-connected layers.

The first thing to do before training is to change the training image to fit the input size of the VGG model. The first thing to do before training is to change the training image to fit the input size of the VGG model. For example, if the $S = 256$, it reduces the width or height of the training image to 256. At this time, while maintaining the aspect ratio, the rest of the parts are also rescheduled, which is said to have been an isotropic-rescaled method. Researchers have to set the S value after cutting the image randomly. There are two methods: single-scale training and multi-scale training. First, for single-scale training, SSS is fixed at 256 to 384. Based on the weight values of the VGG model, which was initially trained by setting the S to 256 , the S to set the S to 384 and retrain. Since a lot of learning has already been done at 256, setting the learning to 384 reduces the running rate and makes it learn. Multi-scale training methods do not fix the S and randomly set the values from 256 to 512. Usually, objects can be different from each other, not the same size. Therefore, learning effects can be better if taught at random multi-scale. This method of data alignment is called scale-jittering.

The classification experimental results simply used the ILSVRC dataset.

Experiments show that the D model using 3×3 convolution filter performs better than the C model using 1×1 convolution filter. Authors explained that this is due to better extraction of features of space and location information compared to 1×1 model.

2 Deep Residual Learning for Image Recognition

In Deep Learning, the deeper the Neural Network gets, the better the performance, but the more difficult training is. In this paper, Researchers use the Residual Learning Framework to show that training can be easily done even in deep neural networks and present a methodology. It reconstructs the Layer by using Residual Function for Learning without creating a new function. In this paper, Researchers show how to make Optimization with Residual easier with the Imperial Evidence Showing method, and focus on how to construct deeper layers while increasing accuracy.

As a result, 152 layers were stacked to provide better performance than traditional VGG networks while reducing complexity. It achieved 3.57% of errors, winning first place in ILSVRC 2015, and analyzed CIFAR-10 data from 100 to 1,000 layers in the paper. Idea is simple. It can be summarized to overcome the Degradation Problem. Degradation is a problem in which the deeper the network, the lower the Accuracy, i.e., the lower the performance. This problem is not a matter of overfitting. In the case of overfitting, the train accuracy of the deep layer should be high and the test accuracy should be low, but both of them are low in this problem. The problem of degradation of training accuracy means that the system is not easily optimized, so researchers want to compare shallow structures with deep architectures. In the paper, degradation problem is viewed and solved as a side effect that occurs because the deeper layers build up, the more complicated the optimization becomes. More details are showed as Figure 3.

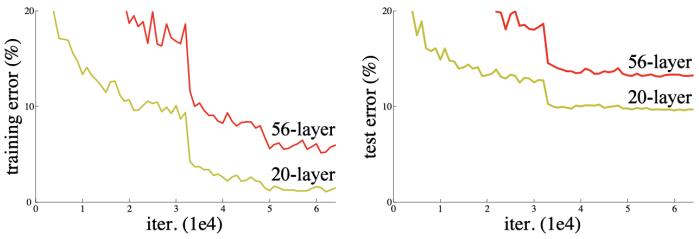

Figure 3: Training and Testing Errors on CIFAR-10 Dataset.[1].

From now on, I will summarize the methodology briefly. The Deep Resistant Learning Network is designed so that stacked layers are fitted to the Residual mapping, not directly to the next layer. For example, if the conventional immediate mapping is $H(x)$, then in this paper author present $F(x) = H(x) - x$, a junction of nonlinear layers. That is, $H(x) = F(x) + x$. This assumption assumes that Residual Mapping is easier to optimize than conventional Mapping.

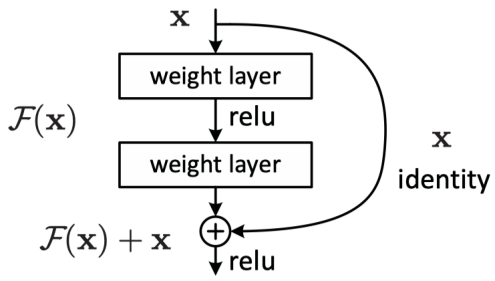

Figure 4: Shortcut Connection.[1].

$F(x) + x$ is the same as Shortcut Connection. This makes one or more layers skip. In other words, it creates a Skip while making a Shortcut Connection with Identity Mapping. The advantage of this Identity Short Connection is that it does not require additional parameters and does not require multiplication operations. x is an input, which is a series of processes called Model $F(x)$ and adds identity, x, and produces $F(x) + x$ as output. For reference, $F(x)$ is called Model and ReLU is function. Assuming that $H(x)$ is considered a basic mapping and that a number of nonlinear layers can asymptotically approximate complex functions when x is input, the residual function, i.e., $H(x) - x$, can be unconsciously approximated. In other words, if complex functions can be approximated by multiple nonlinear layers, the Residual function can also be approximated. To sum up, $H(x)$ is the predicted value by inserting an input into the model. x is the actual value. Surprisingly, only binomial, but $F(x)$ Residual Function is easy to learn.

Next, I organize about Evaluation. Experimental methods reveal the degradation problem and evaluate the methods in this paper. The two objectives of this paper are: Simply put, it is about easy optimization methods and increasing accuracy. First, unlike the plain net, there is a goal that ResNet seems to be more easily optimized. Secondly, it is shown that the Residual Net increases accuracy more easily. The network structure is largely described as three network structures, which are the VGG-19 network, a Plain Network consisting of 34 Parameter Layer, a Residual Network consisting of 34 Parameter Layer. Dotted shortcut increases the dimension. The photo was omitted due to the paper size.

$$y = F(x, W_i) + x \tag{1}$$

The Plain Network is inspired by the philosophy of VGGNet. The convolutional layer is usually composed of 3×3 filters and there are two simple design laws. First, authors construct layers with the same number of filters so that the output feature map size is the same. Second, if the feature map is half the size, double the number of filters so that the time complexity per layer is constant. The researchers directly downsampling the convolutional layer with stride 2. The profile of the network consists of a global average pooling layer. It also configures the 1000-way full-connected layer as softmax. The total weighted layer is 34. For ResNet, we construct a small number of filters to have a lower complexity than VGGNet. The 34 Layer Baseline is 3.6 Billion FLOPs (multiple-adds), accounting for 18% of VGG-19. This translates to 19.6 billion FLOPs.

$$y = F(x, W_i) + W_s x \tag{2}$$

Residual Network added a shortcut connection (Figure 4) based on the Plain Network. This short cut connection makes the network a counterpart residual version. Identity shortcut is used when the input and output have the same Dimension. There are two options when Dimension increases, such as dotted lines. There are two options when Dimension increases, such as dotted lines.

1. Padding Extra Zero Entries to increase Dimension makes Shortcut the same mapping. These options do not require additional parameters.

2. Projection Shortcut which figured by equation (2) was used as a match dimension. This is a 1×1 convolution layer. The two options consisted of stride 2.

The Experimental Result is similar to the foregoing. To execute, the image was resized. Samples were sampled randomly at [256, 480]. The researchers adopted Batch Normalization to ensure that it takes place before each convolution and activation. The researchers used SGD and set the mini-batch size to 256. The Learning Rate started at 0.1 and was set to divide by 10 for Error Plateaus. The model was trained to 60×10^4 Iteration. The researchers used weight decay as 0.0001 and set the momentum to 0.9. Dropout was not done separately. Standard 10-Crop testing was adopted for testing. For the best results, researchers adopted the Fully-Convolution form. Furthermore, researchers derive the mean value of the multi-scale.

[1] 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.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.