DEEP LEARNING LAB(EEE4423-01) Week 3 : VGGNet and ResNet

Hyeon Si Eun[1], 2019142214
[1] Department of Electric and Electronic Engineering, Yonsei University

1 VGG-Net

The paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" introduces a new neural network architecture called VGGNet[2], which achieves state-of-the-art performance on the ImageNet dataset, a large-scale image classification task.

This model won first place in ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) (Russakovskyet al., 2014) for localization and second in classification. The googleNet, the another model in ILSVRC, also showed good performance. But the VGGnet has become more popular because of its simpler structure. This paper investigates the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Therefore the experiment was conducted by setting a convolution kernel size and increasing the number of convolutions.

The VGGNet architecture consists of a series of convolutional layers with small filters (3x3), followed by max pooling layers. Max pooling kernel size is 2x2, stride is 2 to resize the image in half. And the padding size is 1. Also VGGNet have activation function, ReLU and 3 linear layers. After two hidden layers with 4096 nodes, an output layer with 1000 nodes follow because ImageNet dataset has 1000 classes. Due to these linear layers, the number of parameters is too large. To draw out the probability of each class, softmax activation is used. The model architecture is shown in Figure 1.

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 depth of the configurations increases from the left (A) to the right (E) as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv (receptive field size)-(number of channels)". The ReLU activation function is not shown for brevity. It is designed for ImageNet classification.

Theoretically, using a 5x5 kernel or two 3x3 kernels are the same in performance, but experiments show that overlapping 3x3 kernels has better performance. The reason for this is as follows: First, the non-linear function ReLU is more used, so it makes the decision function more discriminative. Second, it reduces the number of parameters that need to be learned. it also speeds up the model.

The authors also propose a new training method that uses small batches and data augmentation to prevent overfitting.

The authors show that VGGNet outperforms previous state-of-the-art methods on the ImageNet dataset, achieving a top-5 error rate of 7.3%. They also demonstrate the generality of the VGGNet architecture by applying it to other image recognition tasks, such as object localization and detection, with promising results.

Overall, the paper highlights the effectiveness of deep convolutional neural networks for large-scale image recognition tasks and provides a new architecture that achieves excellent performance.

2 ResNet

ResNet (short for residual network) is a deep neural network architecture that was introduced in the paper "Deep Residual Learning for Image Recognition"[1] in 2016. ResNets are designed to address 'the problem of vanishing gradients' that can occur in very deep neural networks and make difficult to train them effectively. The paper addresses the challenge of training deep convolutional neural networks (CNNs) by proposing a novel residual learning framework.

Deep learning is basically believed to improve performance as the layers are deeper. In VGGnet, image classification performance was tested up to 19 layers. But the difference between 16 layers and 19 layers was little. If we make the layers more deeper, is the performance better? The researchers who make the research paper experimented to make the layers deeper than 19. They conducted comparative tests on 20-layer and 56-layer. The results are shown in Figure 2.

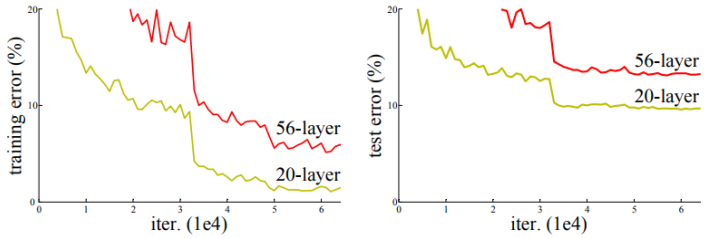

Figure 2: Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error.

Looking at Figure 2, the results get worse as the layers are deeper. The authors tried to eliminate one of the causes, the vanishing gradient problem. They argue that training very deep CNNs is difficult because of the problem of vanishing gradients, where the gradients of the loss function become increasingly small as they propagate through the network, leading to slow convergence and poor performance. The backpropagation and gradient-based learning methods all have this problem. Because if you use the backpropagation or gradient method, you update the weight of the neural network, and in the process of updating, there is a vanishing problem that the gradient itself becomes very small.

Residual neural network (ResNet) was created to solve these problems. The ResNet architecture addresses this problem by introducing shortcut connections, or skip connections, that allow gradients to bypass one or more layers in the network. In the ResNet architecture, each layer is replaced by a residual block, which contains multiple convolutional layers followed by a shortcut connection. The authors show that ResNets can be deeper than

This is an extended abstract. The full paper is available at the this paper and Computer Vision Foundation

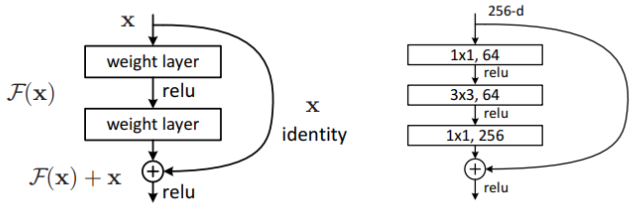

Figure 3: A deeper residual function F for ImageNet. Left: a building block (on 56×56 feature maps) as in Residual learning for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

traditional CNNs while still being easier to train and achieving better performance on various image recognition benchmarks.

The key idea behind ResNets is the introduction of residual blocks, which contain shortcut connections that allow the network to skip over one or more layers, like in Figure 3. These shortcut connections add an identity mapping to the output of a layer and allow gradients to be propagated more easily through the network. The residual block can be expressed mathematically as follows:

$$H(x) = F(x) + x \tag{1}$$

After the input value x has passed through the neural network, the result is H(X) = F(x)+x, which is the sum of the weighted value F(x) and the x created through identity mapping. The addition of the residual function to the identity mapping forms the residual connection that allows information to flow through the block without loss. Even if the gradient of F(x) becomes almost zero, the final output value H(x)'s gradient is guaranteed to be near one.

In the above case, this is the default ResNet when the input and output values are in the same dimension. This time, let's think about the difference between the weight layer and the identity mapping. For example, in the case of CNN, if you put the input into the weight layer, the input image size decreases, which means that the matrix size of the reduced image F(x) and the input image x are different. So you cannot add it. Therefore, in this case, we need to post-process x with identity mapping to fit the reduced matrix F(x). This is expressed as an equation:

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

W_i is a weight belonging to the CNN layer, and W_s multiplies the identity mapping so that matrix addition is possible.

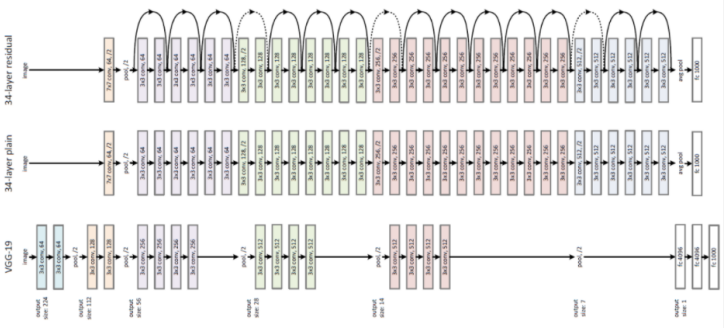

Figure 4: The bottom structure is VGG-19. The intermediate structure is a 34-layer plane network where VGG-19 is deeper. The top structure is a 34-layer reserved network (ResNet), and skip/short connection has been added to the plain network.

To add skip/short connection, the added value x must be the same dimension as the output value. ResNet has two types of skip/shortcut connections when the input dimension is smaller than the output dimension. (A) Shortcut performs identity mapping by applying zero padding in addition to increasing dimensions. Therefore, there are no additional parameters. (B) Use the projection shortcut only when the dimension is increasing. The other shortcut is identity. Additional parameters are required. (C) All shortcuts are projections. It requires more parameters than B. C is not used in this paper. This is because the amount of computation in the model increases. The bottleneck structure introduced below uses the A option.

The paper also introduces the ResNet architecture with bottleneck blocks, which reduces the number of parameters and computation time required to train very deep networks. The principle is shown in Figure 3.

1x1 conv layers are added at the beginning and end of the network as shown on the right of Figure 3. This technique was proposed by GoogLeNet. The 1x1 conv reduces the number of parameters without reducing the performance of the neural network. By reducing the amount of computation with bottleneck design, the 34-layer becomes the 50-layer ResNet, and there are deeper neural networks with bottleneck design, ResNet-101 and ResNet-152. ResNet-152 has less computation than VGG-16. The overall structure is shown in Figure 5.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 5: Resnet Architectures for ImageNet classification

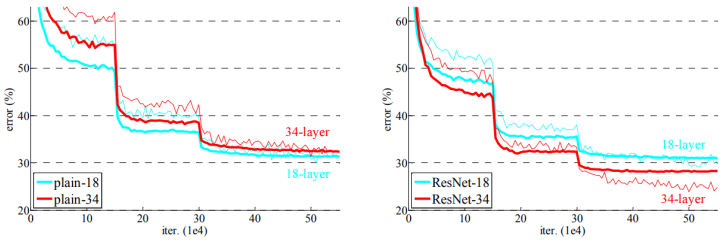

Figure 6: Training on ImageNet. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

In plain networks, 18-layers outperform 34-layers because of the vanishing gradient problem. In ResNet, the gradient vanishing problem is solved by skip connection, so the performance of 34-layer is superior to that of 18-layer. There is no difference between 18-layer plane network and 18-layer ResNet network. This is because the problem of gradient vanishing does not appear in small neural networks.

Overall, the paper shows that residual learning is an effective technique for training very deep neural networks for image recognition tasks. ResNets have become a popular choice for various computer vision applications and have significantly advanced the state-of-the-art in image classification, detection, and segmentation.

[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.