**Technical Writing HW** – First draft of final research paper

[ Sohee Kim - 20220344 ]

Topic : Models of Convolutional Neural Network

[Outline]

1. **Introduction**
   1. Computer vision Tasks – Image Classification
   2. About Convolutional Neural Network
2. **Convolutional Neural Network**
   1. **Basic CNN components**
      1. Convolution layer
      2. Pooling layer
      3. Fully-connected layer
   2. **Models (Architectures)**
      1. AlexNet
      2. VGGNet
      3. ResNet
   3. **Training Models**
      1. Dataset - CIFAR10
      2. Details of learning
3. **Result**
   1. Accuracy & Loss
   2. Comparing each model
   3. Discussion
4. **Conclusion**
5. **References**
6. **Abstract**
7. **Introduction**

Computer vision has become increasingly important and effective in recent years due to its wide-ranging applications in areas as diverse as smart surveillance and monitoring, health and medicine, sports and recreation, robotics, drones, and self-driving cars. Visual recognition tasks, such as image classification, localization, and detection, are the core building blocks of many of these applications, and recent developments in Convolutional Neural Networks (CNNs) have led to outstanding performance in these state-of-the-art visual recognition tasks and systems. As a result, CNNs now form the crux of deep learning algorithms in computer vision.[1]

CNN is useful in many applications, especially in image related tasks. Applications of CNN include image classification, image semantic segmentation, 2 object detection in images, etc. Image classification plays an important role in computer vision. We will focus on image classification (or categorization) in this paper. In image categorization, every image has a major object which occupies a large portion of the image. An image is classified into one of the classes based on the identity of its main object, e.g., dog, airplane, bird, etc. One key ingredient of deep learning in image classification is the use of Convolutional architectures

Convolutional neural network design inspiration comes from the mammalian visual system structure [2]. Convolutional neural network is first introduced by LeCun in [2]. Since 2006, many methods have been developed to overcome the difficulties encountered in training deep neural networks. Krizhevsky propose a classic CNN architecture Alexnet [3] and show significant improvement upon previous methods on the image classification task. With the success of Alexnet [3], several works are proposed to improve its performance. ZFNet [4], VGGNet [5], GoogleNet [6] and ResNet [7] are proposed. In this work, the architecture and training methods of Convolutional neural networks, specifically Alexnet[3], VGGNet[5] and ResNet[7] will be introduced.

1. **Convolutional Neural Network**
   1. **Basic CNN components**

Convolutional neural network layer types mainly include three types, namely Convolutional layer, pooling layer and fully-connected layer. When these layers are stacked, a CNN architecture has been formed. A simplified CNN architecture for MNIST classification is illustrated in Figure 1[9].

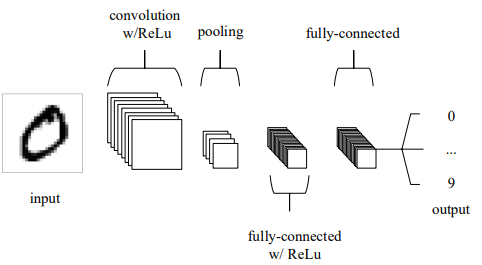


Figure 1 : A simple CNN architecture [9]

* + 1. **Convolution layer**

Convolutional layer is the core part of the Convolutional neural network, which has local connections and weights of shared characteristics. The aim of Convolutional layer is to learn feature representations of the inputs. As shown in above, Convolutional layer is consisting of several feature maps. Each neuron of the same feature map is used to extract local characteristics of different positions in the former layer, but for single neurons, its extraction is local characteristics of same positions in former different feature map. In order to obtain a new feature, the input feature maps are first convolved with a learned kernel and then the results are passed into a nonlinear activation function. We will get different feature maps by applying different kernels.[8] The rectified linear unit (commonly shortened to ReLu) aims to apply an ’elementwise’ activation function such as sigmoid to the output of the activation produced by the previous layer.

Convolutional layers are also able to significantly reduce the complexity of the model through the optimization of its output. These are optimized through three hyperparameters, the depth, the stride and setting zero-padding. The depth of the output volume produced by the convolutional layers can be manually set through the number of neurons within the layer to the same region of the input. We are also able to define the stride in which we set the depth around the spatial dimensionality of the input in order to place the receptive field. For example, if we were to set a stride as 1, then we would have a heavily overlapped receptive field producing extremely large activations. Alternatively, setting the stride to a greater number will reduce the amount of overlapping and produce an output of lower spatial dimensions. Zero-padding is the simple process of padding the border of the input and is an effective method to give further control as to the dimensionality of the output volumes.

It is important to understand that through using these techniques, we will alter the spatial dimensionality of the convolutional layers output. To calculate this, you can make use of the following formula[9]:

Where V represents the input volume size (height×width×depth), R represents the receptive field size, Z is the amount of zero padding set and S referring to the stride. If the calculated result from this equation is not equal to a whole integer, then the stride has been incorrectly set, as the neurons will be unable to fit neatly across the given input.

* + 1. **Pooling layer**

Pooling layers aim to gradually reduce the dimensionality of the representation, and thus further reduce the number of parameters and the computational complexity of the model.[9] It is usually placed between two Convolutional layers. The size of feature maps in pooling layer is determined according to the moving step of kernels. The typical pooling operations are average pooling and max pooling. We can extract the high-level characteristics of inputs by stacking several Convolutional layer and pooling layer.[8]

* + 1. **Fully-connected layer**

The fully-connected layers will then perform the same duties found in standard ANNs (Figure 2) and attempt to produce class scores from the activations, to be used for classification. In general, the classifier of Convolutional neural network is one or more fully-connected layers. They take all neurons in the previous layer and connect them to every single neuron of current layer. There is no spatial information preserved in fully-connected layers. The last fully-connected layer is followed by an output layer. For classification tasks, softmax regression is commonly used because of it generating a well-performed probability distribution of the outputs. Another commonly used method is SVM, which can be combined with CNNs to solve different classification tasks.[8]

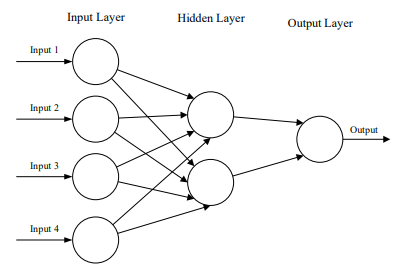


Figure 2 : A simple three-layered feedforward neural network, comprised of an input layer, a hidden layer, and an output layer. [9]

* 1. **Models (Architectures)**

테이블이(가) 표시된 사진

자동 생성된 설명

Figure 3 : Part of classic CNN models [9]

Compared with other methods, CNNs can achieve better classification accuracy on large scale datasets due to their capability of joint feature and classifier learning. The breakthrough of large-scale image classification comes in 2012. Krizhevsky et al. [3] develop the AlexNet and achieve the best performance in ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2012. After the success of AlexNet, several works have made significant improvements in classification accuracy by either reducing filter size or expanding the network depth. Part of well-known models can be seen in Figure 3. We will discuss about AlexNet and two more significant models, VGGNet and ResNet.

* + 1. **AlexNet**

The architecture of our AlexNet is summarized in Figure 4. It contains eight learned layers - five convolutional and three fully-connected. The output of the last fully-connected layer is sent to a 1000-way softmax layer which corresponds to 1000 class labels in the ImageNet dataset.

Before AlexNet, sigmoid and tanh were usually used as activation function, but AlexNet used Rectified Linear Units (ReLUs) activation function which are non-saturating nonlinearity. The formula of ReLU is . It avoids vanishing gradients for positive values and more computationally efficient to compute and has better convergence performance than sigmoid and tanh.

In AlexNet, 1.2 million training parameters are too big to fit into the NVIDIA GTX 580 GPU with 3GB of memory. Therefore, they spread the network across two GTX 580 GPUs. It is just for memory limitation. AlexNet used overlapping max pooling of size 3x3 with stride 2. This scheme reduced the top-1 and top-5 error rates by 0.4% and 0.3%. Local Response Normalization (LRN) is used in AlexNet to help with generalization. LRN reduces the top-1 and top-5 error rates by 1.4% and 1.2%. Nowadays, batch normalization is used instead of LRN. A regularization technique called Dropout was used in AlexNet. It randomly set the output of each hidden neuron to zero with the probability of 0.5. Dropout reduces complex co-adaptations of neurons since a neuron cannot rely on the presence of particular other neurons.

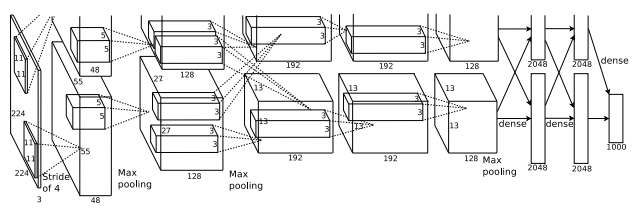


Figure 4 : An illustration of the architecture of AlexNet [3]

* + 1. **VGGNet**

K. Simonyan et al. [5] address another important aspect of AlexNet architecture design – its depth. To this end, they fix other parameters of the architecture, and steadily increase the depth of the network by adding more convolutional layers, which is feasible due to the use of very small (3 × 3) convolution filters in all layers.

This showed experimentally that the parallel assignment of these small-size filters could produce the same influence as the large-size filters. In other words, these small-size filters made the receptive field similarly efficient to the large-size filters (7×7 and 5×5). By decreasing the number of parameters, an extra advantage of reducing computational complication was achieved by using small-size filters. These outcomes established a novel research trend for working with small-size filters in CNN.

In general, VGG obtained significant results for localization problems and image classification. While it did not achieve first place in the 2014-ILSVRC competition, it acquired a reputation due to its enlarged depth, homogenous topology, and simplicity. However, VGG’s computational cost was excessive due to its utilization of around 140 million parameters, which represented its main shortcoming. Figure 5 shows the structure of the network.



Figure 5 : An architecture of VGGNet [11]

* + 1. **ResNet [7]**

For conventional deep learning networks, they usually have conv layers then fully connected (FC) layers for classification task like AlexNet[3], ZFNet[4] and VGGNet[5], without any skip, shortcut connection, we call them plain networks here. When the plain network is deeper (layers are increased), the problem of vanishing/exploding gradients occurs. To solve the problem of vanishing/exploding gradients, a skip/shortcut connection is added to add the input x to the output after few weight layers as Figure 6. Even if there is vanishing gradient for the weight layers, we always still have the identity x to transfer back to earlier layers

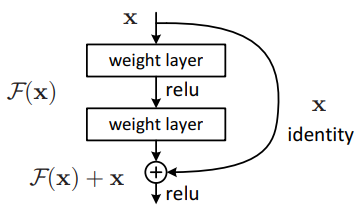
****

Figure 6 Residual learning: a building block [7]

Figure 7 shows the ResNet architecture. In Figure 7, the VGG-19 [5] (bottom) is a state-of-the-art approach in ILSVRC 2014. And 34-layer plain network (middle) is treated as the deeper network of VGG-19. The Resnet, 34-layer residual network (top) is the plain one with addition of skip connection. There are 3 types of skip connections when the input dimensions are smaller than the output dimensions. In addition, Batch Normalization is used for each conv.

Skip connection enables to have deeper network and finally ResNet becomes the Winner of ILSVRC 2015 in image classification, detection, and localization, as well as Winner of MS COCO 2015 detection, and segmentation. By increasing the network depth to 152 layers, 5.71% top-5 error rate is obtained which is much better than VGG-16[5] and GoogleNet[6].

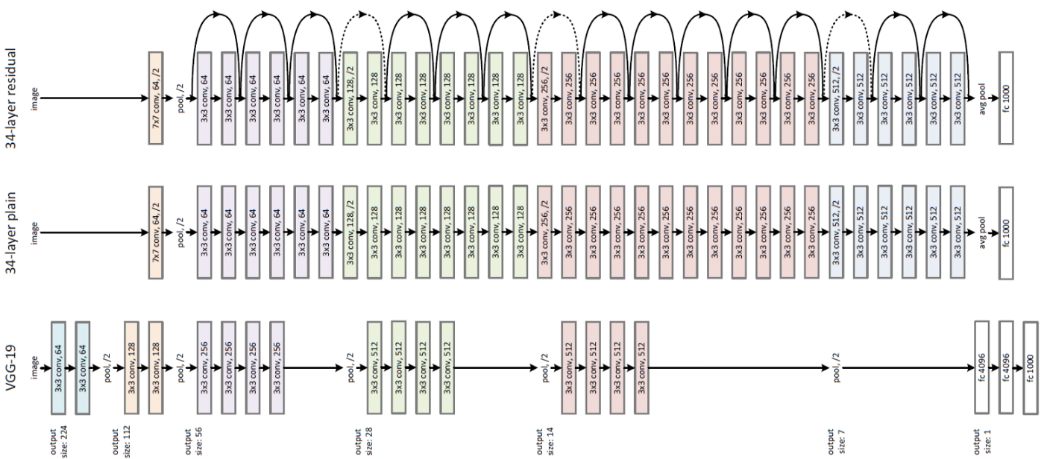


Figure 7 Example network architecture for ImageNet [7].

* 1. **Training Models**
     1. Dataset - CIFAR10
     2. Details of learning

1. **Result**
   1. Accuracy & Loss
   2. Comparing each model
   3. Discussion
2. **Conclusion**
3. **References**

[1] Khan, Salman, et al. "A guide to convolutional neural networks for computer vision." Synthesis Lectures on Computer Vision 8.1 (2018): 1-207.

[2] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012).

[4] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014.

[5] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in ICLR, 2015.

[6] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[7] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[8] Guo, Tianmei, et al. "Simple convolutional neural network on image classification." 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA). IEEE, 2017.

[9] O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." arXiv preprint arXiv:1511.08458 (2015).

[10] Li, Zewen, et al. "A survey of convolutional neural networks: analysis, applications, and prospects." IEEE Transactions on Neural Networks and Learning Systems (2021).

[11] <https://neurohive.io/en/popular-networks/vgg16/>