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

[ Sohee Kim - 20220344 ]

Topic : Models of Convolutional Neural Network

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

CNN is useful in a lot of 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[1]. Convolutional neural network is first introduced by LeCun in [1]. Since 2006, many methods have been developed to overcome the difficulties encountered in training deep neural networks. Krizhevsky propose a classic CNN architecture Alexnet [2] and show significant improvement upon previous methods on the image classification task. With the success of Alexnet [2], several works are proposed to improve its performance. ZFNet [3], VGGNet [4], GoogleNet [5] and ResNet [6] are proposed. We will focus on the architecture and training methods of Convolutional neural networks, specifically Alexnet[2], VGGNet[4] and ResNet[6].

1. **Body**
   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.

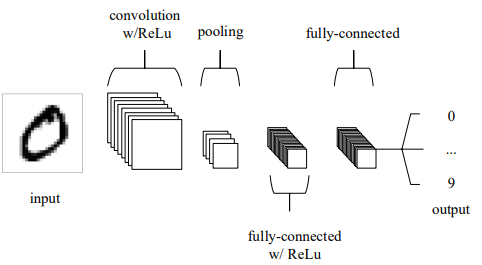


Figure 1 : An simple CNN architecture,

* + 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 consists 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. 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 optimisation of its output. These are optimised 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:

(V − R) + 2Z / S + 1

* + 1. Pooling layer

The pooling layer will then simply perform downsampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation and increase the robustness of feature extraction. 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[16] and max pooling[17].We can extract the high level characteristics of inputs by stacking several Convolutional layer and pooling layer.

* + 1. Fully-connected layer

The fully-connected layers will then perform the same duties found in standard ANNs 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.

* + 1. Models (Architectures)
       1. AlexNet
       2. VGGNet
       3. ResNet
  1. **Training Models**
     1. Dataset
        1. CIFAR10
     2. Details of learning
        1. Loss function
        2. Optimizer
        3. Batch size, epoch

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

[1] LecunY, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

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

[3] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in ECCV, 2014.

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

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with Convolutionals,” Co RR, vol. abs/1409.4842, 2014

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

<https://www.sas.com/en_in/insights/analytics/neural-networks.html>

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