### LAB 1: Convolutional neural network

Tai-Hsiang Huang

MediaTek Inc.

1. **Introduction**

In this work, the Residual Network (ResNet) [1] is implemented to recognize the images in the Cifar-10 data set. Comparing to the traditional convolutional neural network (i.e. Vanilla CNN), the ResNet

1. **Experiment**

This section describes the environment of the experiment conducted in this work. The dataset is described in section 2.1, the architecture of the CNN models are described in section 2.2, and the training hyper-parameters are described in section 2.3.

## **Dataset**

The Cifar-10 dataset is used as the training and test dataset in this work. It consists of 60000 32x32 color images (RGB) in 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each of the class has 6000 images. Among them, 50000 images are training images, and 10000 images are used as test images. Fig. 1 shows 10 images of each of the classes.

## **CNN Models**

The models implemented in this work are not exactly the same as those described in [1]. Specifically, only the basic residual block is adopted to construct the CNN networks, and the sizes of the filters are different from that of the original. For experimental purpose, three different CNN networks are implemented: ResNet, VanillaNet, and Pre-active ResNet. The results of each network would be reported and compared between each other in Section 3.

Three different network depths are implemented for each of the CNN networks: 20 layers, 56 layers, and 110 layers. All of the networks are constructed with a similar high-level structure. The first layer is a convolutional layer that generates 16 output feature maps with resolution of 32x32. Behind the first layer, three sets of 2n-layer basic blocks would be cascaded. After that, a global average pooling would applied before the final linear layer. The value of n is computed depends on the depth of the CNN networks. Specifically, the n values would be 3, 9, and 16 for the 20-layer, 56-layer, and 110-layer networks respectively. The details of the output map size and filter size of each block sets are listed in Table I.

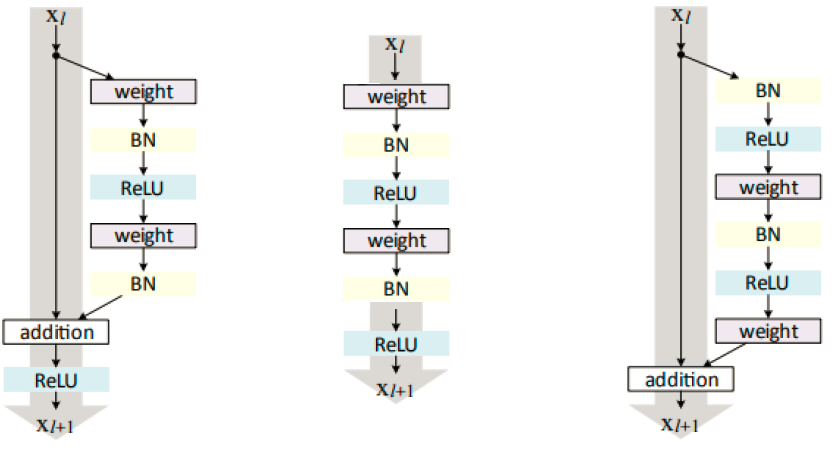


**Fig. 1.** Examples of the images contained in the Cifar-10 dataset.

TABLE I

Details of The Networks

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1st Block set | 2nd Block set | 3rd Block set |
| Layer depth | 1+2n | 1+2n | 1+2n |
| Filter size | 16x16 | 32x32 | 64x64 |
| Output map size | 32x32 | 16x16 | 8x8 |



1. (b) (c)

**Fig. 2.** Basic blocks of the CNNs implemented in this work: (a) ResNet (b) VanillaNet (c) Pre-active ResNet

Fig. 2 shows the basic blocks of the CNN networks implemented in this work: (a)ResNet, (b)VanillaNet, and (c)Pre-active ResNet. It can be seen that both (a) and (c) have a shortcut from the input to the output. This shortcut makes the networks learn the residual instead of the real data. On the other hand, (b) is a traditional CNN network structure. It serves as the reference to the ResNet and Pre-active ResNet in my experiment.

## **Training Hyper-Parameters**

The hyper-parameters used for training are the same for the ResNet, Vanilla Net, and the Pre-active ResNet. The details are listed in Table I.

TABLE II

Training Hyper-Parameters

|  |  |
| --- | --- |
|  | ResNet/ Vanilla Net/ Pre-active ResNet |
| Method | SGD |
| Momentum | 0.9 |
| Mini-batch size | 128 |
| Total epochs | 164 |
| Learning rate | 0.1, epoch < 81,  0.01, 81<= epoch < 122,  0.001, 122<= epoch |
| Weight decay | 0.0001 |
| Weight initialization | Conv2D.weight: KaiMing\_Normal  Conv2D.bias: 0  BatchNorm2D.weight: 1  BatchNorm2D.bias: 0  Linear.weight.std: 0.001  Linear.bias: 0 |
| Loss function | Cross-entropy |

1. **experimental results**

The results of ResNet, VanillaNet, and Pre-active ResNet are shown in this section. For each type of networks, the results of 20-layer, 56-layer, and 110-layer are reported.

TABLE III

Final Test Error

|  |  |  |  |
| --- | --- | --- | --- |
|  | ResNet | VanillaNet | Pre-active ResNet |
| 20 layers | 8.08% | 9.64% | 8.48% |
| 56 layers | 6.81% | 13.02% | 6.90% |
| 110 layers | 6.42% | 53.02% | 6.36% |



(a)



(b)



(c)

**Fig. 3.** Training loss curves: (a) 20 layers, (b) 56 layers, and (c) 110 layers.



(a)



(b)



(c)

**Fig. 4.** Test error curves: (a) 20 layers, (b) 56 layers, and (c) 110 layers.

1. **Discussion**

In this paper, we have described a method based on the

* Importance of initial parameter

1. **References**
2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
3. He, K., Zhang, X., Ren, S., & Sun, J. (2016, October). Identity mappings in deep residual networks. In European Conference on Computer Vision (pp. 630-645). Springer International Publishing.