KAIST

AI502 Deep Learning

Homework 2 - Convolutional Neural Networks CIFAR100

Cappa Victor
20206080
School of computing
victor.cappa@kaist.ac.kr

HW2-Intro

The main purpose of this Homework is to implement three different 18 layer CNN architectures based on different building blocks to solve a classification task on the CIFAR100 dataset.

The CIFAR100 dataset consists of consists 60000 32x32 colour images in 100 classes, with 600 images per class. There are 50000 training images and 10000 test images. There are 500 training images and 100 testing images per class (more informations about the dataset can be found here: https://www.cs.toronto.edu/~kriz/cifar.html).

The basic blocks that we will implement are:

- PlainBlock
- ResidualBlock
- MobileBlock

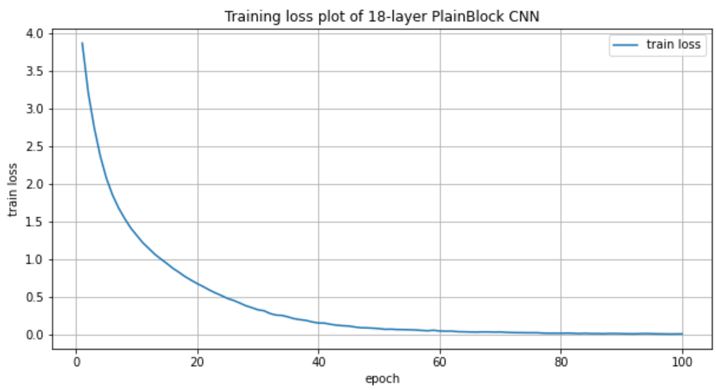
The chosen hyperparameters and configurations are:

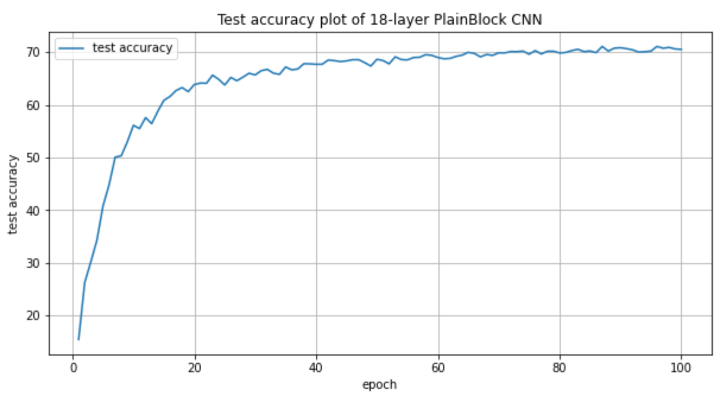
- batch size: 128
- learning rate: 0.01
- number of epochs: 100
- optimizer: SGD with momentum=0.9
- loss function: cross entropy loss
- decay rate scheduler: none
- data augmentation is performed
- regularization techniques like dropout and stochastic depth are not used

HW2-Problem1 - Plot the training loss and test accuracy of 18-layer CNNs for each epoch

In this section the PlainBlock is used. More informations about the provided implementation for the PlainBlock can be found in the [HW2]20206080.ipynb file.

The plotted training loss and test accuracy for each epoch are:

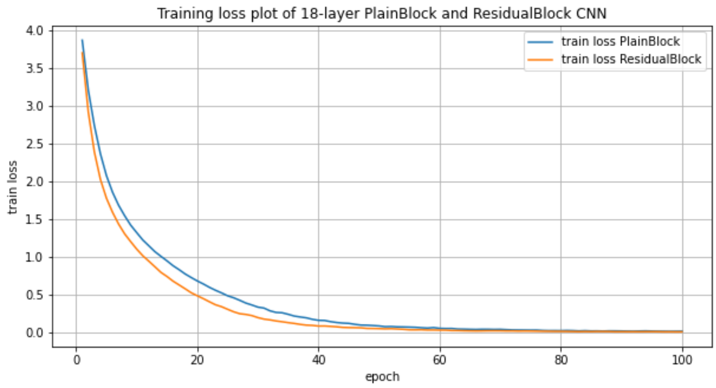


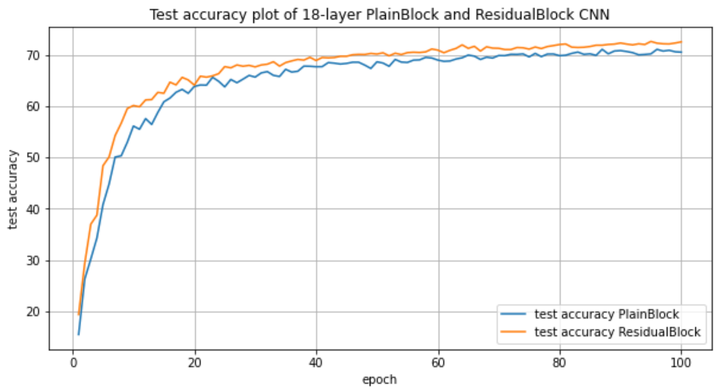


The Training loss decreases monotonically for each epoch, starting form an initial value of 3.872 at epoch 1 and reaching a final loss value of 0.014 at epoch 100. It means the chosen hyperparameters (batch size: 128, learning rate: 0.01, SGD with momentum=0.9) are a valuable solution for the given architecture and CIFAR100 (improvements are still possible). The loss is smooth, doesn't diverge and decreases fast in the first epochs.
The test accuracy achieves a final value of 70.55% at epoch number 100.

HW2-Problem2 - Plot the training loss and test accuracy of ResNet

In this section the ResidualBlock is used. The residual connections have been implemented according to the paper "Deep residual learning for image recognition" and when the dimension of the output block increases we use 1x1 convolutions with stride=2.
The plotted training loss and test accuracy for each epoch are:





The final Training loss of the residual network is 0.005, meaning it is approximately three times smaller than the final one of the PlainBlock network. Moreover, the loss decreases faster compared to the PlainBlock network especially in the first epochs of the training phase.
The Test accuracy, though, is similar to the one of the PlainBlock network and the final test accuracy is 72.56%, very similar to the previous one.
As a result we can conclude that the shortcut connections can be useful to ensure a faster convergence of the training loss but they don't allow us to have a significant improvement of the classification performances (test set accuracy) of the neural network. Maybe it can be due to the fact that the residual connections have been designed to allow the training of very deep neural networks and they don't provide significant benefits in our configuration (with 18 layers).

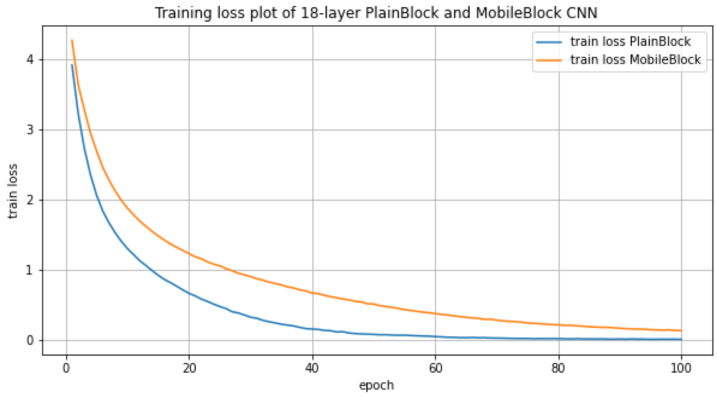
HW2-Problem3 - Plot the training loss, test accuracy, number of parameters, and number of FLOPs of MobileNet and PlainNet

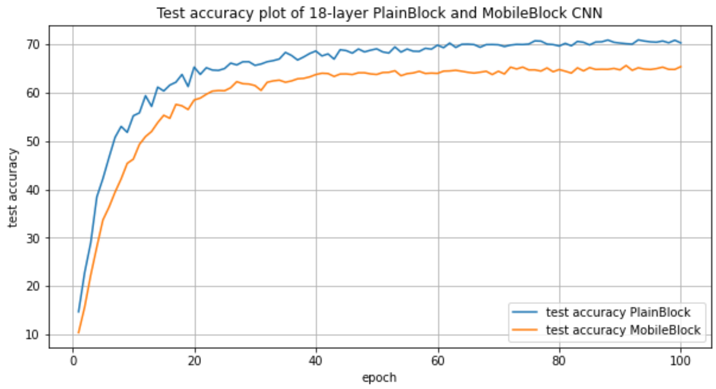
Number of parameters of a standard convolutional layer and a depthwise separable convolutional layer calculated with respect to N, M, and Dk:

#totParamStdConvLayer=Dk*Dk*M*N (+N if bias is True)
#paramDepthwiseConv=Dk*Dk*M (+M if bias is True)
#paramPointwiseConv=M*N (+N if bias is True)
#totParamDepthwiseSeparableConv=#paramDepthwiseConv + #paramPointwiseConv

The program prints the number of parameters and the model's computational complexity expressed in Gmac for the PlainBlock and the MobileNet network:
PlainBlock network:
Computational complexity: 0.55 GMac
Number of parameters: 11.05 M
MobileNet network:
Computational complexity: 0.07 GMac
Number of parameters: 1.33 M
(Flops (Floating point operations per second) are another measure that can be used to assess the performances of an algorithm requiring floating point operations. Flops is approximately double the Macs).

The plotted training loss and test accuracy for each epoch are:





The first plot shows the training loss of the MobileBlock network compared to the PlainBlock one. The MobileBlock network loss decreases more slowly compared to the PlainBlock scenario and the final loss at epoch 100 is 0.100, so it is approximately seven times bigger compared to the one achieved with the former architecture (0.014). Also the final Test accuracy is 61.49%, while with PlainBlock it was 70.55%.
The main advantage of this architecture is that it has much less parameters (1.33 M compared to 11.05 M of the PlainBlock architecture) and its computational complexity is also much lower (0.07 GMac compared to 0.55 GMac). So it could be a valuable solution in resource-constrained environments like mobile and robotic applications.