

Homework 5

Due: Friday, March 15 at 5:00 PM

1. Homework Goal

You will learn how to build very deep convolutional networks, using Residual Networks (ResNets). In theory, very deep networks can represent very complex functions; but in practice, they are hard to train. Residual Networks, introduced by He et al., allow you to train much deeper networks than were previously practically feasible. In this assignment, you will implement the basic building blocks of ResNets, and then put together these building blocks to implement and train the neural network for image classification on CIFAR100 dataset via Pytorch. Moreover, in this assignment, you will also learn how to load the pre-trained ResNets which was trained on ImageNet dataset and train it on CIFAR100.

2. Homework Requirements

- Build the following specified Residual Networks as shown in Figure 1 and achieve at least 60% testing accuracy.

First of all, your input has go through one single convolution layer named as *conv1*, which has a filter size 3×3 , input channel size of 3, output channel size of 32, padding of 1, and stride of 1. The output of *conv1* go through a spatial batch normalization defined as *nn.BatchNorm2d* in Pytorch, and then go through a ReLu function. After that the output goes to a dropout layer, and then it goes a sequence of resnet basic blocks. In the homework, you should define your basic block, and the structure of it is shown in Figure 2. For each weight layer, it should contain 3×3 filters for a specific number of input channels and output channels. The output of a sequence of resnet basic blocks go through a max pooling layer with your own choice of filter size, and then goes to a fully connected layer (you need to vectorize the output of pooling layer). The parameter specification for each component is in Figure 1. Note, the parameter notation follows the notation in reference 1 (Kaiming, 2015).

- Fine-tuning pretrained ResNet-18 model and achieve at least 70% testing accuracy.

3. Dataset and Pretrained Model

We provide the CIFAR100 in two different formats as previous homework. The data is located in the following directory:

/projects/training/bawc/CIFAR100/Dataset/

For the second part of your homework, you need to load the pre-trained model, and it is located in the following directory:

/projects/training/bawc/CIFAR100/model/

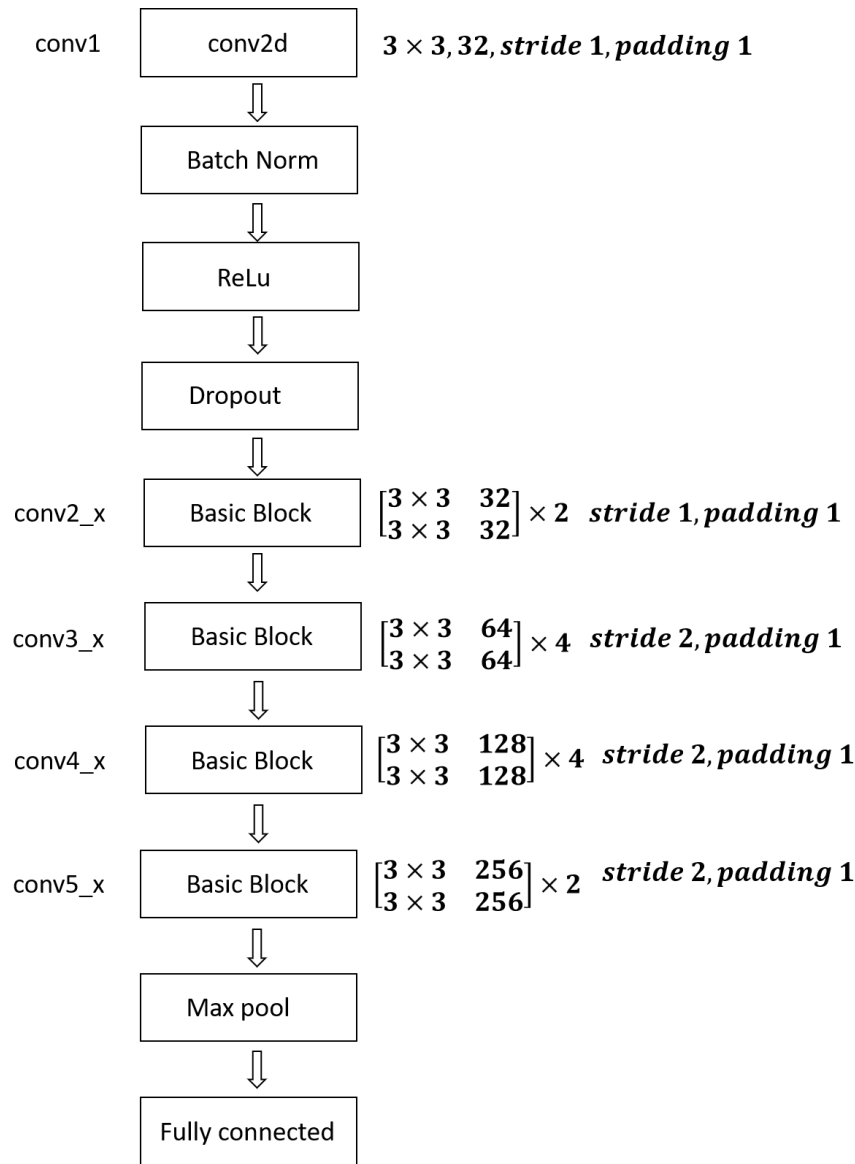


Figure 1: ResNet Structure

Copy the pre-trained model file into your code directory and you can load the pre-trained model with the following command

```
def resnet18(pretrained = True) :
    model = torchvision.models.resnet.ResNet(torchvision.models.resnet.BasicBlock, [2, 2, 2, 2])
    if pretrained :
        model.load_state_dict(model_zoo.load_url(model_urls['resnet18'], model_dir = './'))
    return model
```

Note that the input size for the pre-trained model is 224×224 , and you have to upsample

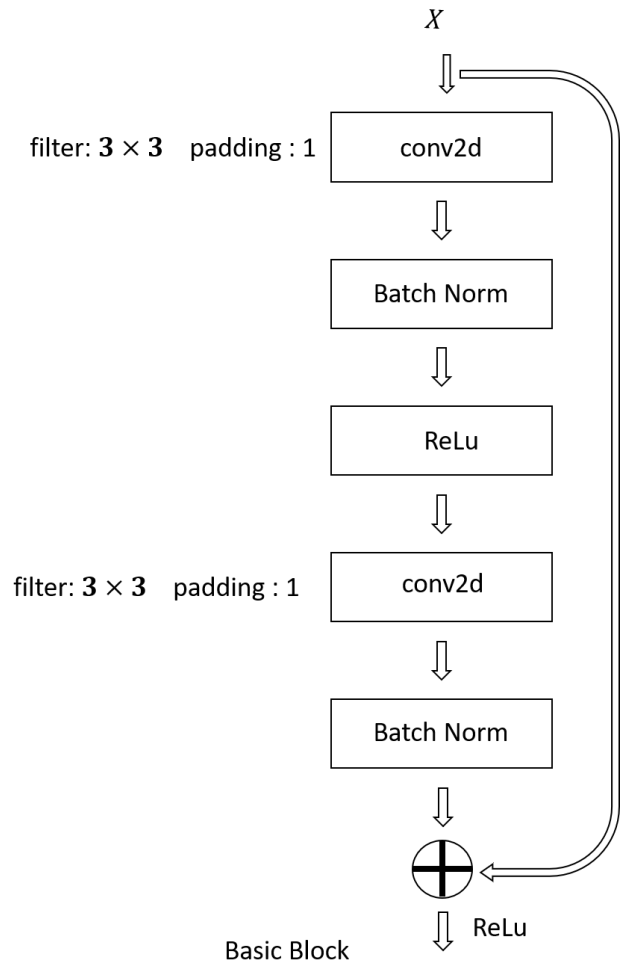


Figure 2: Basic Block

the images in CIFAR100 dataset.

4. Running Time Estimation

For Bluewater users, a reasonable running time for the first part is about 3 hours and the second part is about 3 to 4 hours.

5. Reference

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun - Deep Residual Learning for Image Recognition (2015)
- Pytorch torchvision.models <https://pytorch.org/docs/stable/torchvision/models.html>