ResNet on CIFAR10

Pablo Ruiz – Harvard University – August 2018

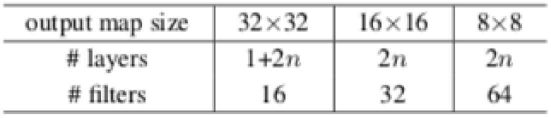
Introduction

This work is a continuation of the previous tutorial, where we demystified the ResNet following the original paper [1]. However, this structure is built to perform well on ImageNet dataset.

ImageNet dataset consist on a set of images (the authors used 1.28 million training images, 50k validation images and 100k test images) of size (224x224) belonging to 1000 different classes. However, CIFAR10 consist on a different set of images (45k training images, 5k validation images and 10k testing images) distributed into just 10 different classes.

Because the sizes of the input volumes (images) are completely different, it is easy to think that the same structure will not be suitable to train on this dataset. We cannot perform the same reductions on the dataset without having dimensionality mismatches.

We are going to follow the solution the authors give to ResNets to train on CIFAR10, which are also tricky to follow like for ImageNet dataset. On the paper [1], section ***4.2 CIFAR-10 and Analysis***, we find the following table:



output map size	32×32	16×16	8×8
# layers	$1{+}2n$	$2n$	$2n$
# filters	16	32	64

Figure 1. Schema for ResNet from the paper

Let's follow then the literal explanation they give to construct the ResNet. We will use n=1 for simplification, leading to a ResNet20.

Structure

Following the same methodology of the previous work on ResNets, let's take a look at the overall picture first, to go into the details layer by layer later.

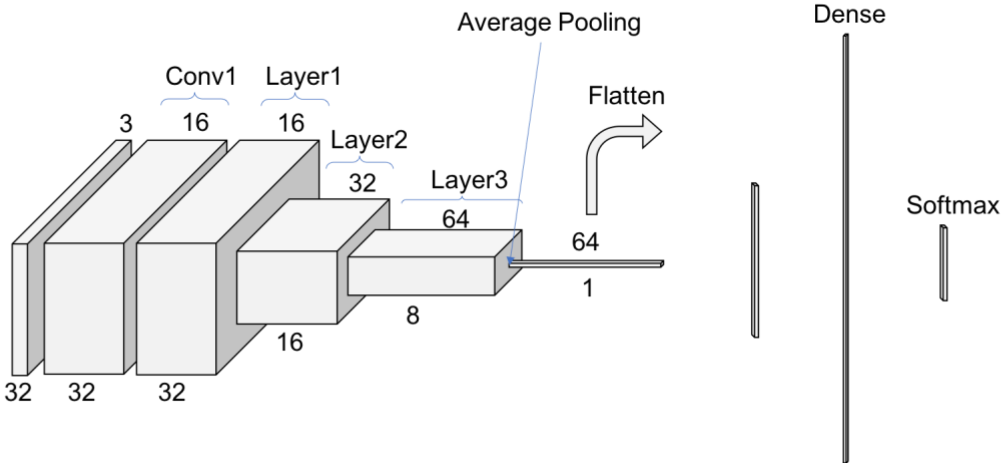


Figure 2. Scheme for ResNet Structure on CIFAR10

Convolution 1

The first step on the ResNet before entering into the common layer behavior is a 3x3 convolution with a batch normalization operation. The stride is 1 and there is a padding of 1 to match the output size with the input size. Note how we have already our first big difference with ResNet for ImageNet, that we have not include here the max pooling operation in this first block.

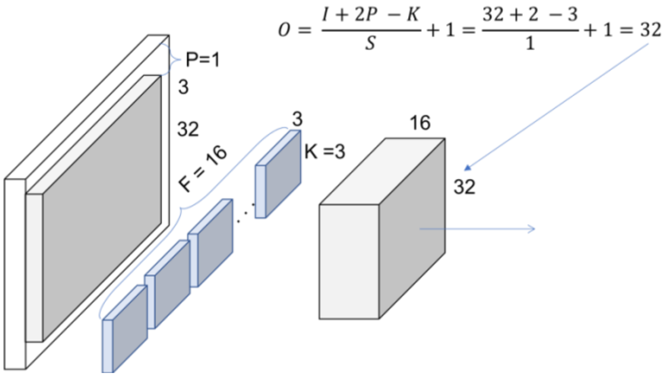

$O = \frac{I + 2P\ - K}{S} + 1 = \frac{32 + 2\ - 3}{1} + 1 = 32$
P=1
3
32
F = 16
3
K =3
16
32

Figure 3. Conv1

We can check with Figure 2 that the output volume of Conv1 is indeed 32x32x16

Layer 1

The rest of the notes from the authors to construct the ResNet are:

- Use ***a stack of 6n layers of 3x3 convolutions***. The choice of n will determine the size of our ResNet.
- The feature map sizes are {32, 16, 8} respectively with $2n$ convolutions for each feature map size. Also, the number of filters is {16, 32, 64} respectively.
- The ***down sampling*** of the volumes through the ResNet is ***achieved increasing the stride to 2***, for the first convolution of each layer. Therefore, no pooling operations are used until right before the dense layer.
- For the ***bypass connections***, no projections will be used. In the cases where there is a different in the shape of the volume, the input will be ***simply padded with zeros***, so the ***output size matched the size of the volume before the addition***.

This would leave Figure 4 as the representation of our first layer. In this case, our bypass connection is a regular ***Identity Shortcu***t because the dimensionality of the volume is constant thorough the layer operations. Since we chose n=1, 2 convolutions are applied within the layer 1.

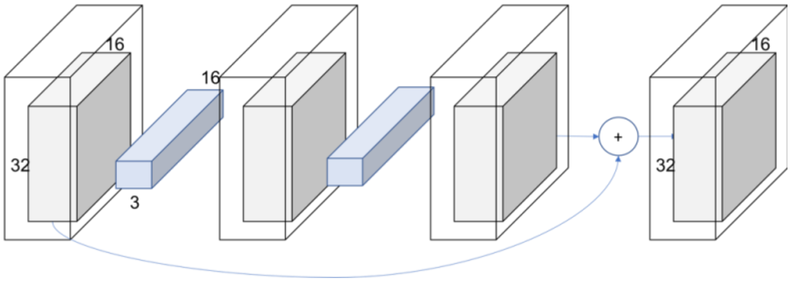


Figure 4. Layer 1

We still can check from Figure 2 that the output volume of Layer1 is indeed 32x32x16. Let’s go deeper!

Layer 2

We are going to see now how to deal with the down sampling of the input volume. Remember we are following the same structure and notation as the work on ImageNet dataset. Make sure you take a look if not follow any step, as there is explained in more detail.

For both layer 2 and next layer 3 the behavior is equivalent to layer 1, with the exception that the first convolution uses a stride of 2, and therefore the size of the output volume is half of the input volume (with the padding of 1). This implies that also the shortcut connection will require an extra step, to adjust the volumes' sizes before the summation.

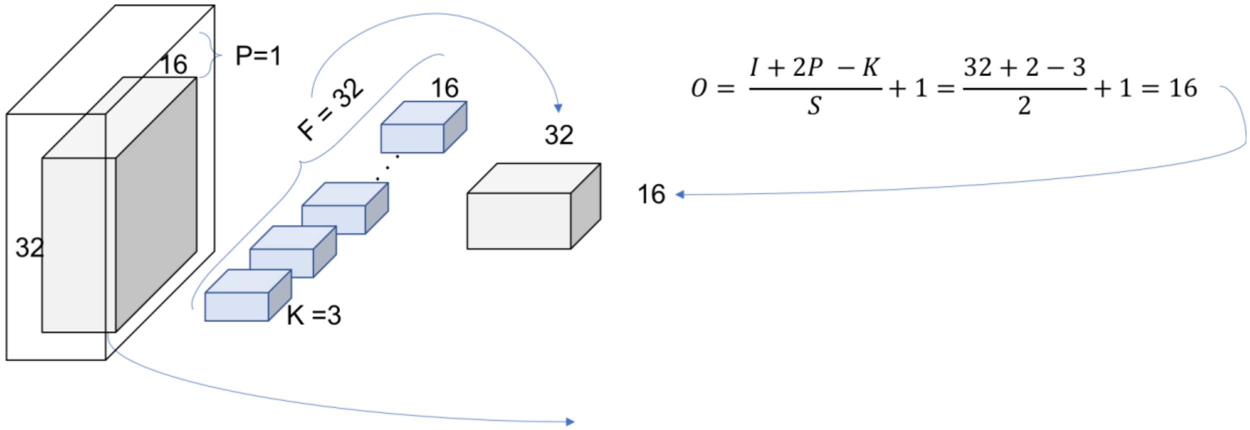

16
P=1
32
F = 32
16
32
K =3
$O = \frac{I + 2P \ - K}{S} + 1 = \frac{32 + 2 - 3}{2} + 1 = 16$
16

Figure 5.Layer 2, Block 1, Convolution

The final look of the entire layer 2 is shown in Figure 6Figure 2. We can see how the convolution with stride 2 is used in the skip connection for the down sample as well as in the first convolution of the layer. Also, we can check with the table from the paper that we have indeed a 16x16x32 volume.

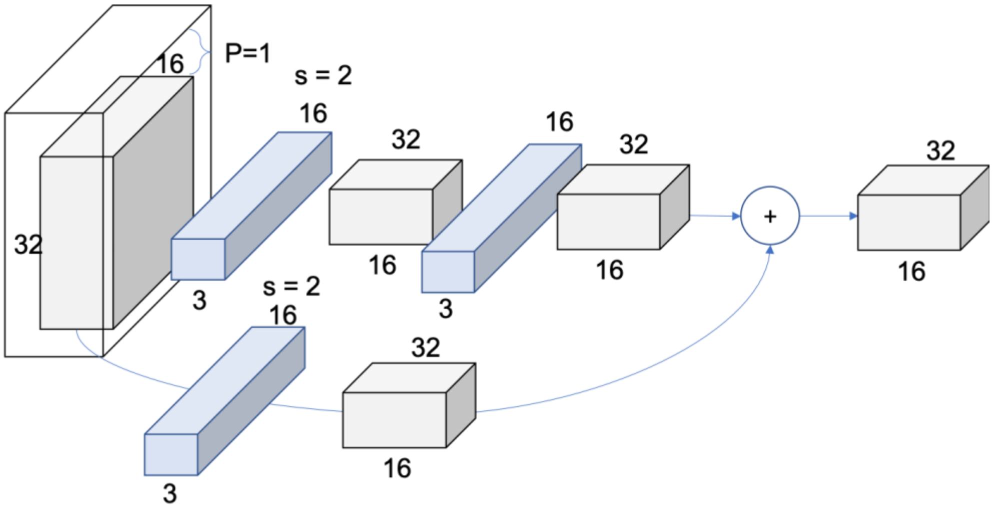


Figure 6. Layer 2

Layer 3

Layer 3 will apply the exact same principles as layer 2, leading to Figure 7.

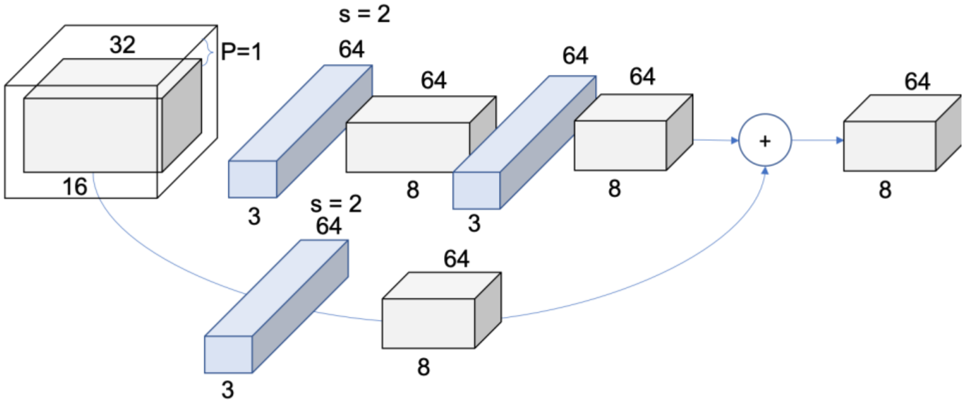


Figure 7. Layer 3

We can check with the table from the paper that we have indeed an 8x8x64 volume.

Summary

The ResNets following the explained rules built by the authors yield to the following structures, varying the value of *n* in Figure 1:

Table 1. ResNets architectures for CIFAR-10

Number of Layers	Number of Parameters
ResNet 20	0.27M
ResNet 32	0.46M
ResNet 44	0.66M
ResNet 56	0.85M
ResNet 110	1.7M
ResNet 1202	19.4M

Note that, intuitively, these architectures do not match the architectures for ImageNet showed at the end of the work on ImageNet.

Find here the code to build these architectures:

Bibliography

[1] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016.