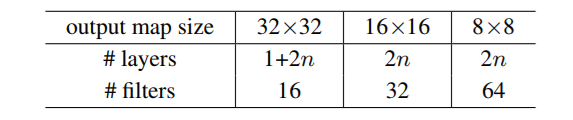
The network inputs are 32×32 images, with the per-pixel mean subtracted

The first layer is 3×3 convolutions. Then we use a stack of 6n layers with 3×3 convolutions on the feature maps of sizes {32, 16, 8} respectively, with 2n layers for each feature map size. The numbers of filters are {16, 32, 64} respectively. The subsampling is performed by convolutions with a stride of 2. The network ends with a global average pooling, a 10-way fully-connected layer, and softmax. There are totally 6n+2 stacked weighted layers



1+2n+2n+2n+1, n=9

1: 3x3 con2d

2n: filter size: 16, conv kernel size: 3x3

2n: filter size: 32, conv kernel size: 3x3

2n: filter size: 64, conv kernel size: 3x3

1: average pooling + 10-way fc + softmax

参数个数：3\*3\*16 + 18\*3\*3\*16 + 18\*3\*3\*32+18\*3\*3\*64 + 8\*8\*64\*10

Mini-batch size = 128

start with a learning rate of 0.1, divide it by 10 at 32k and 48k iterations, and terminate training at 64k iterations, which is determined on a 45k/5k train/val split

data augmentation: 4 pixels are padded on each side, and a 32×32 crop is randomly sampled from the padded image or its horizontal flip

(A) zero-padding shortcuts are used for increasing dimensions, and all shortcuts are parameterfree

(B) projection shortcuts are used for increasing dimensions, and other shortcuts are identity;

(C) all shortcuts are projections

Similar phenomena are also shown on the CIFAR-10 set

[20]

We present comprehensive experiments on ImageNet

[36]

On the ImageNet classification dataset [36], we obtain

excellent results by extremely deep residual nets

Our implementation for ImageNet follows the practice

in [21, 41]

We conducted more studies on the CIFAR-10 dataset

[20]

We initialize the weights

as in [13] and train all plain/residual nets from scratch

The identity

shortcuts (Eqn.(1)) can be directly used when the input and

output are of the same dimensions (solid line shortcuts in

Fig. 3). When the dimensions increase (dotted line shortcuts

in Fig. 3), we consider two options: (A) The shortcut still

performs identity mapping, with extra zero entries padded

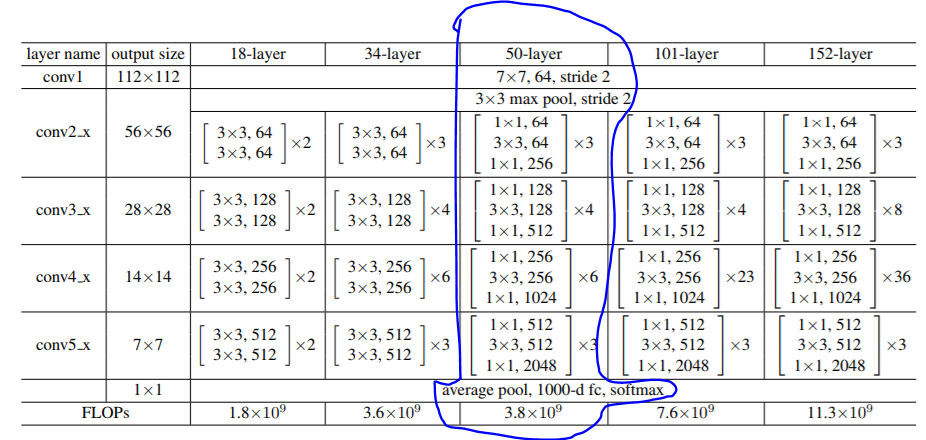
for increasing dimensions. This option introduces no extra

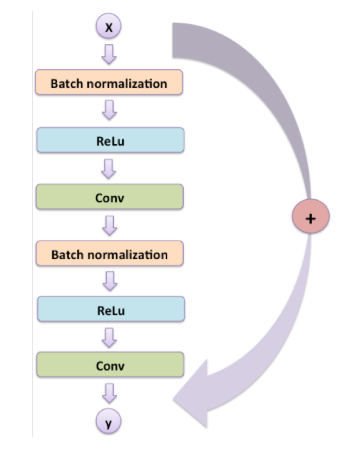
parameter; (B) The projection shortcut in Eqn.(2) is used to

match dimensions (done by 1×1 convolutions). For both

options, when the shortcuts go across feature maps of two

sizes, they are performed with a stride of 2.





def conv2d\_same(inputs, num\_outputs, kernel\_size, stride, scope = None):

if stride == 1:

return slim.conv2d(inputs, num\_outputs, kernel\_size, stride = 1, padding = 'SAME', scope = scope)

else:

pad\_total = kernel\_size - 1

pad\_beg = pad\_total // 2

pad\_end = pad\_total - pad\_beg

inputs = tf.pad(inputs, [[0, 0], [pad\_beg, pad\_end], [pad\_beg, pad\_end], [0, 0]])

return slim.conv2d(inputs, num\_outputs, kernel\_size, stride = stride, padding = 'VALID', scope = scope)

tf.image.per\_image\_xxx

kernel\_initial

initializer=tf.contrib.layers.xavier\_initializer()

|  |
| --- |
| regularizer = tf.contrib.layers.l2\_regularizer(scale=FLAGS.weight\_decay) |
|  |  |
|  | new\_variables = tf.get\_variable(name, shape=shape, initializer=initializer, |
|  | regularizer=regularizer) |