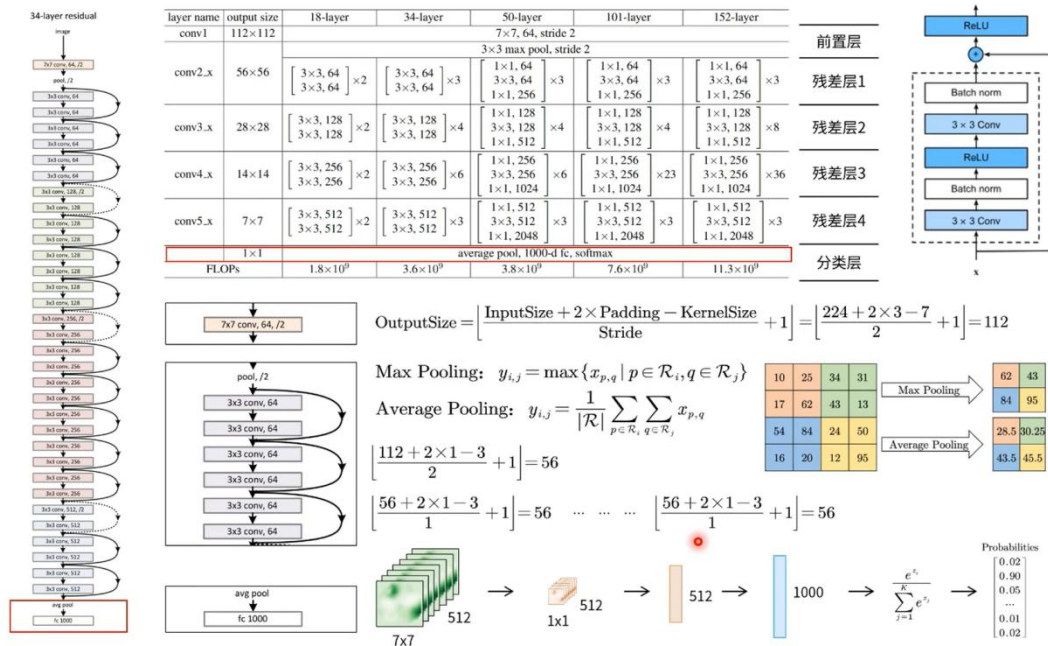


## ResNet 34 图像分类



### Residual blocks

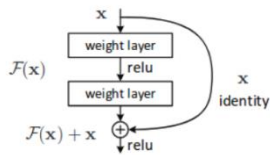


Figure 2. Residual learning: a building block.

### Code Walkthrough

The architecture is based on 34 layer sample (snippet from paper)

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112	7x7, 64, stride 2				
		3x3 max pool, stride 2				
conv2.x	56x56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3.x	28x28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4.x	14x14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5.x	7x7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1x1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

```
F:\anaconda\envs\mypytorch\python.exe F:\pycharmproject\ResNet\train.py
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to cifa\cifar-10-python.tar.gz
18% | 31.3M/170M [00:08<00:21, 6.53MB/s]
```

```
Epoch 24, test acc: 0.7640
Epoch 25, test acc: 0.7704
Epoch 26, test acc: 0.7693
Epoch 27, test acc: 0.7663
Epoch 28, test acc: 0.7682
Epoch 29, test acc: 0.7726
Epoch 30, test acc: 0.7686
```

跑通/问题：跑通

硬件平台：CPU 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz

软件环境：pytorch

算法应用效果：图像分类、目标检测、图像分割

输入数据：ResNet18(ResBlk, [2, 2, 2, 2], 10)

数据的维度：

对于 ResNet18 来说，输入图像的尺寸是 (batch\_size, 3, 224, 224)，即：

batch\_size 是每个批次的样本数量。3 是图像的通道数，表示 RGB 图像。

224, 224 是图像的高度和宽度，ResNet 模型要求输入图像的大小是 224x224。

彩色图像， 224x224 像素的彩色图像。

[2, 2, 2, 2]：这通常表示 ResNet 的各个阶段（stage）中残差块的数量。

这个参数通常表示网络的输出类别数，通常在分类任务中使用。

数据精度：float 32

测试性能（对比）：77%

网址：<https://github.com/maomaol688/ResNet>

代码本地保存：

√

测试数据集保存：

√ <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>

CIFAR-10 包含 10 个不同的类别，每个类别包含 6000 张图像。具体的类别如下：

飞机 (airplane)、汽车 (automobile)、鸟 (bird)、猫 (cat)、鹿 (deer)、狗 (dog)、青蛙 (frog)、马 (horse)、船 (ship)、卡车 (truck)