AIST4010 Tutorial5 --More Efficient CNN

LIANG HONG

Outline

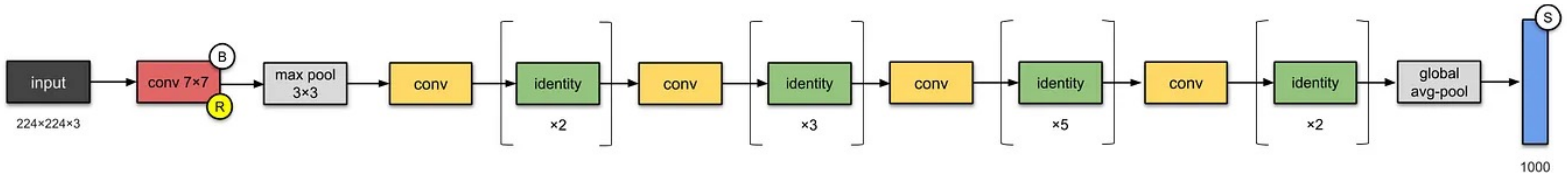
1. Attempts to achieve better performance
2. Attempts to reduce param num
    1. Kernel: ResNet -> ResNeXt
    2. Dim size: Max-Pooling, stride conv

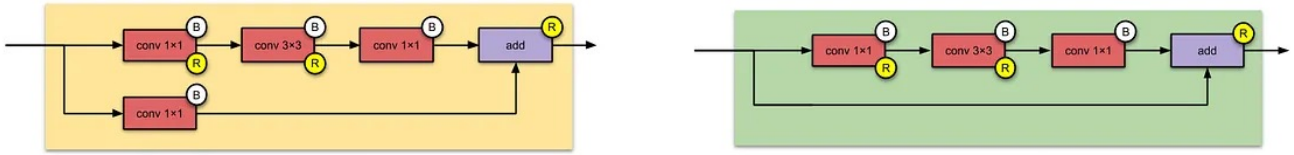
Attempts to achieve better performance

1. Larger model
2. Deeper network
3. Higher quality data
4. Large scale pre-training

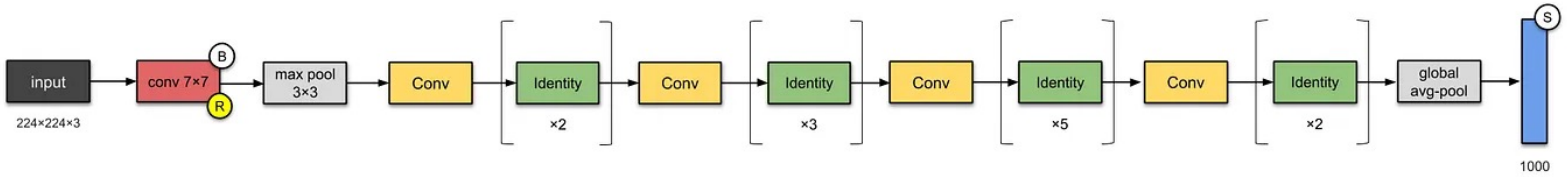
How to achieve similar result with limited resource? E.g. we need it in production scene(on surveillance camera, run on phone chip) or simply do not have enough graphic cards to train.

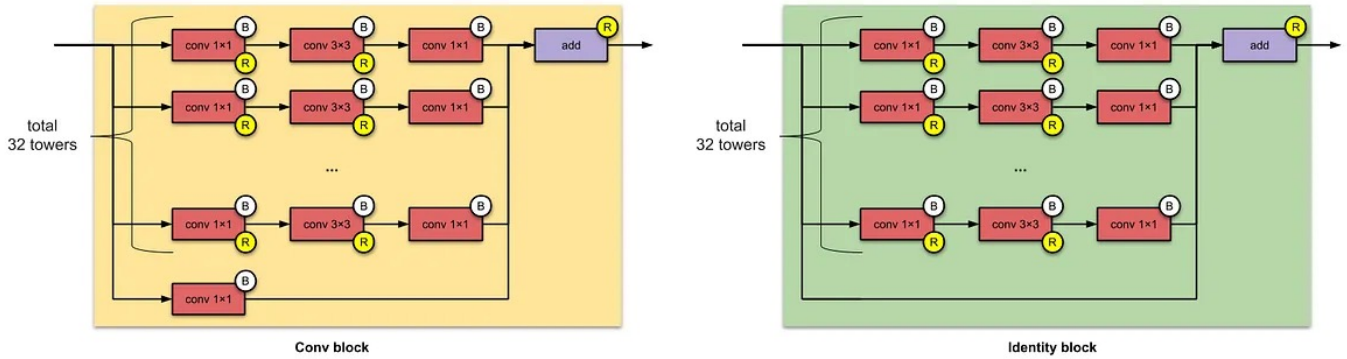
Reduce param num (kernel)





ResNet





ResNeXt

Reduce param num (kernel)

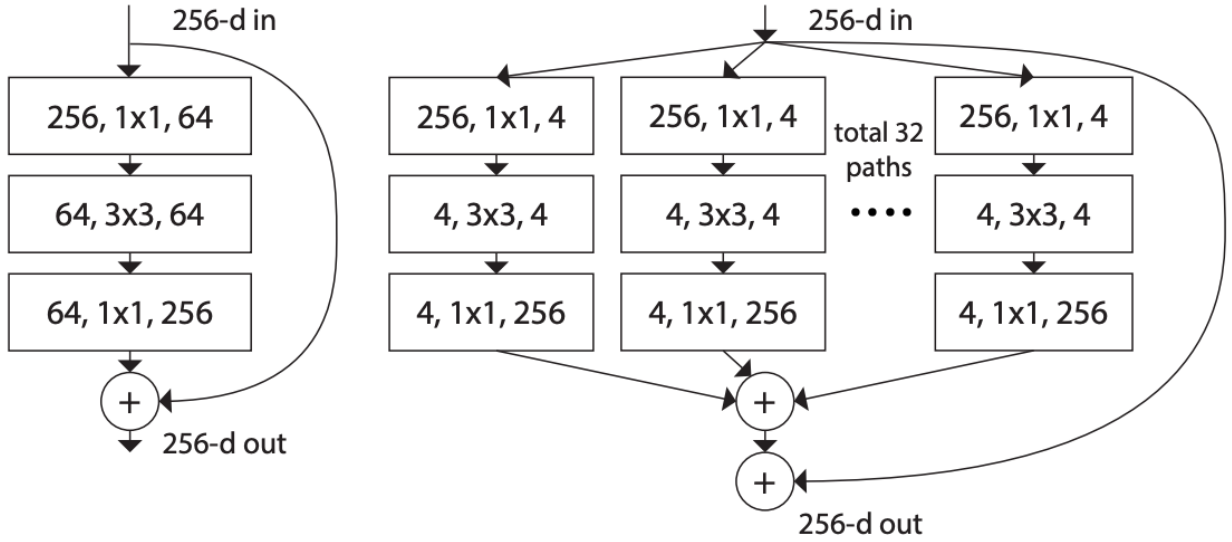
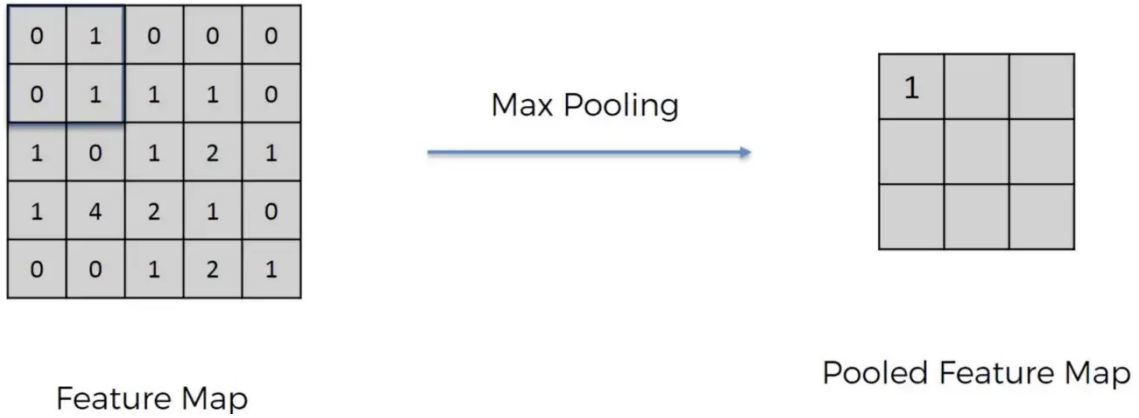
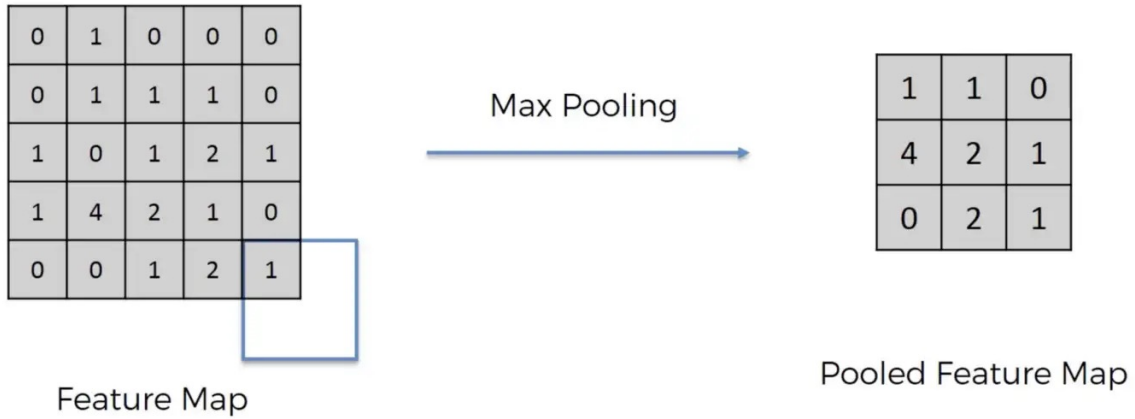


Figure 1. **Left**: A block of ResNet [14]. **Right**: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

Similar to group convolution!

Reduce param num (dim)





Max-pooling:
Squeeze feature map (downscale)
Invariance

Reduce param num (dim)

Simply use conv with stride instead of pooling

1. Conv of 2x2 is very similar to pooling (maybe weighed sum instead of max)
2. Conv of 1x1 itself uses significantly less param and has much higher speed

Hinton: The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster. If the pools do not overlap, pooling loses valuable information about where things are. We need this information to detect precise relationships between the parts of an object. Its true that if the pools overlap enough, the positions of features will be accurately preserved by "coarse coding" (see my paper on "distributed representations" in 1986 for an explanation of this effect). But I no longer believe that coarse coding is the best way to represent the poses of objects relative to the viewer (by pose I mean position, orientation, and scale).