Paper Summary of:

Going Deeper with Convolutions

codename: Inception

By Anthony Martinez

Goal

- Primary focus was not on accuracy
- Motivated by mobile and embedded computing
- Self imposed limitation of 1.5 billion add and multiplies
  - Want adaptation

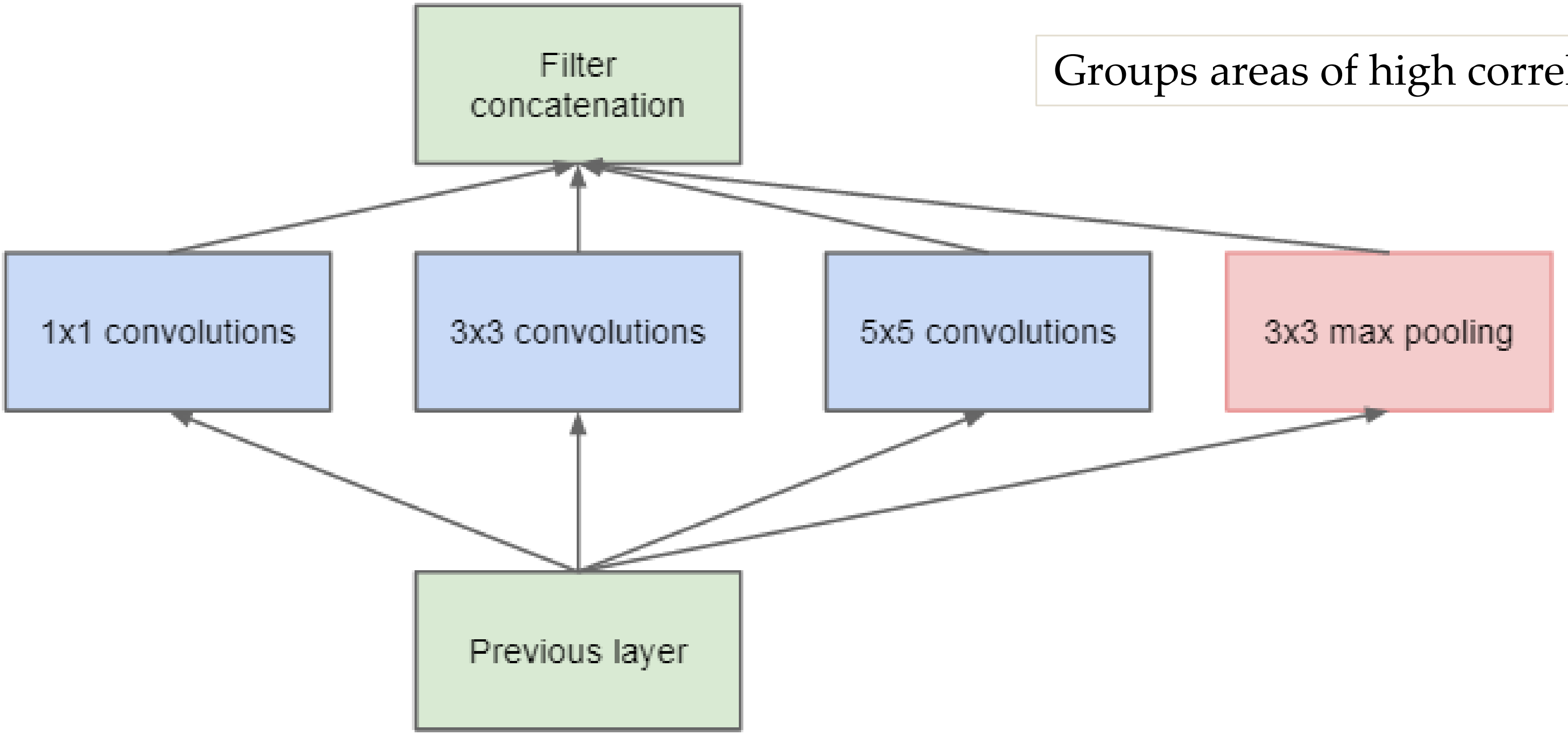
At the time:

- For larger data sets
  - Simple solution : Increase network size.
- Drawbacks
  - Explosion of parameter size
  - Overfitting
  - Increase in computational resources

Go Deeper

- Don’t just add more layers
- Design a smarter architecture
  - Sparse matrix
    - Can’t do that
    - Dense matrix have a similar property
    - You can build it in modules
  - Network in Network

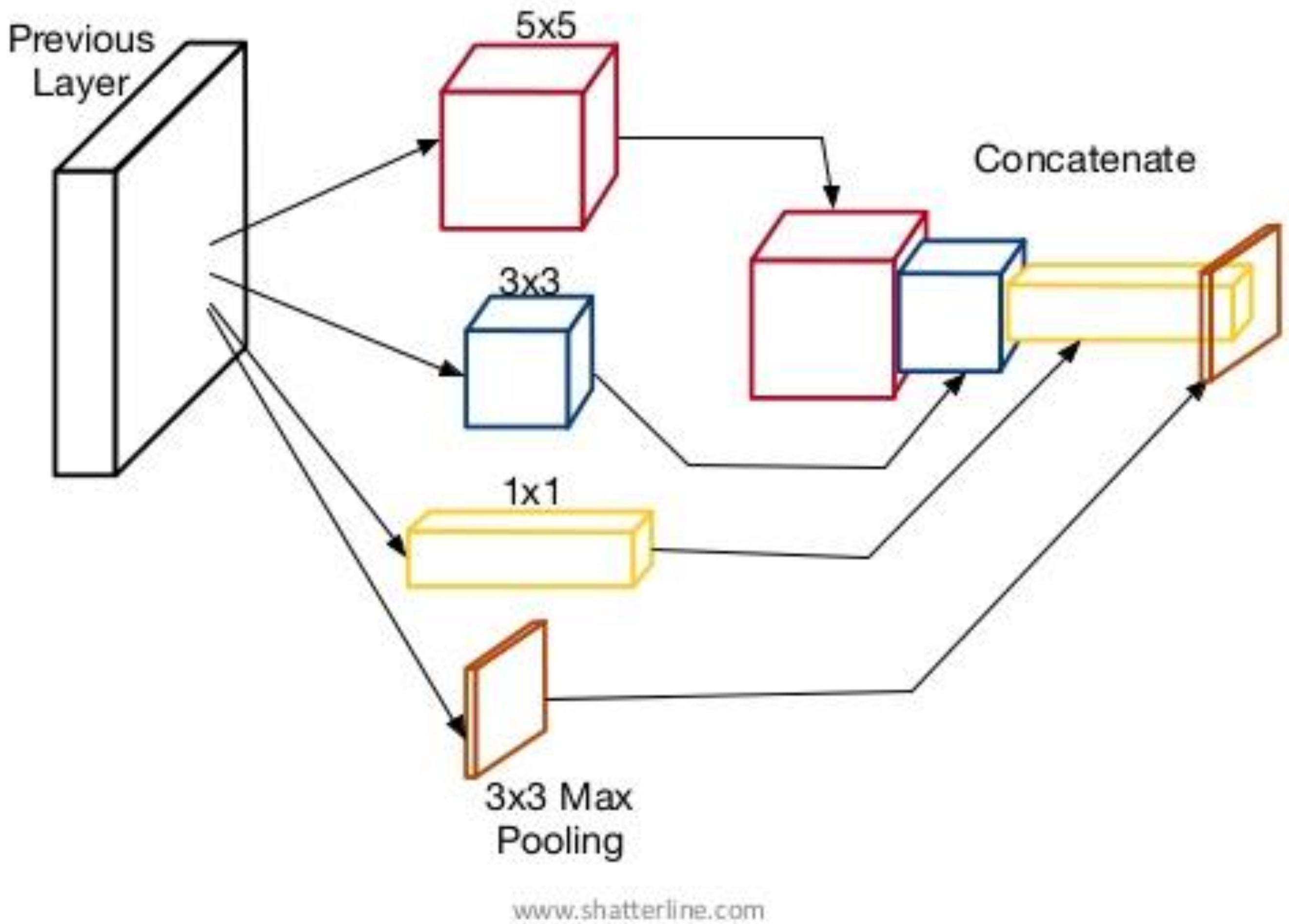
Inception Module



Groups areas of high correlated statistics

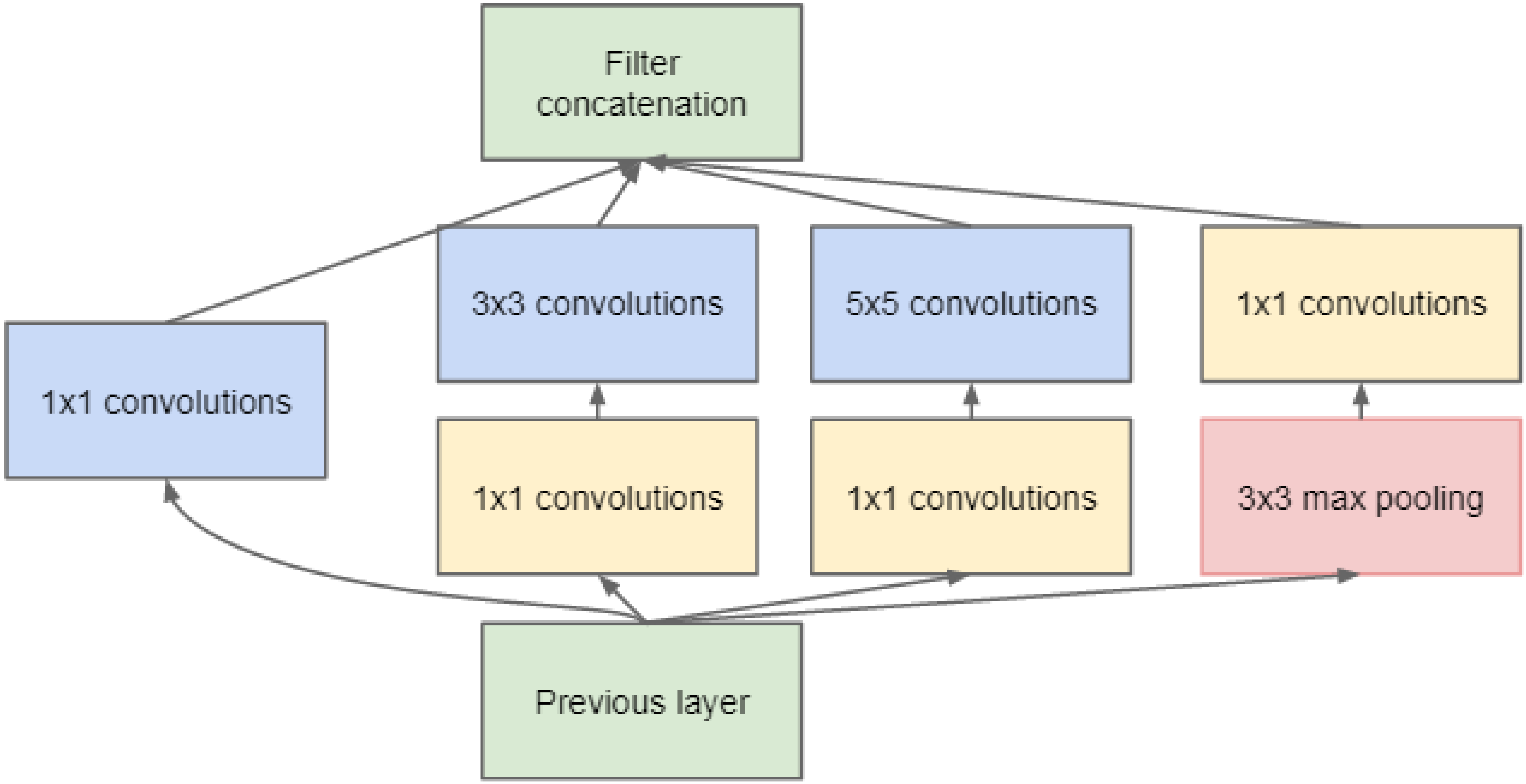
(a) Inception module, naïve version

Conceiving the Inception Module



Why 1x1, 3x3, and 5x5?

NIN



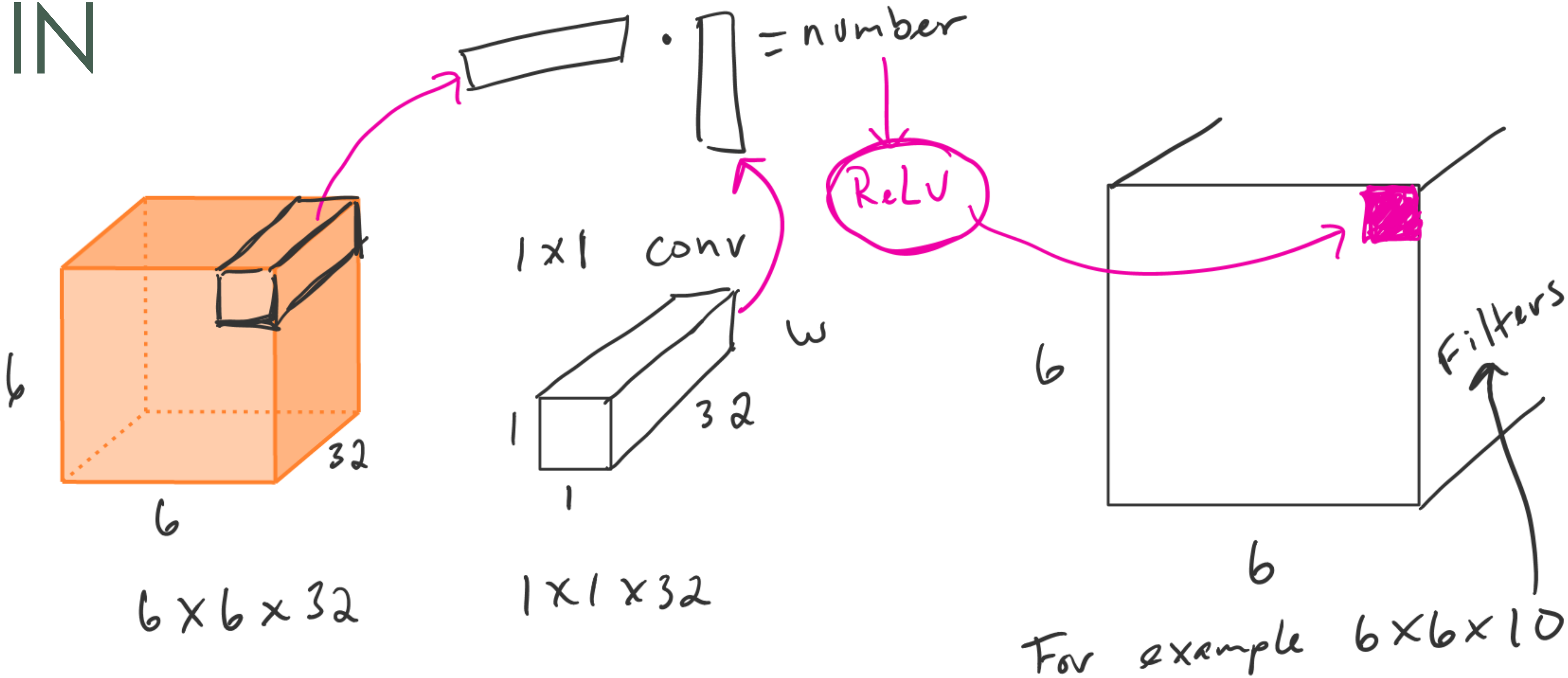
(b) Inception module with dimension reductions

1 x 1 convolution

- Mainly used as dimension reduction modules
  - Remove computational bottlenecks
- Allows increase in
  - Depth
  - Width

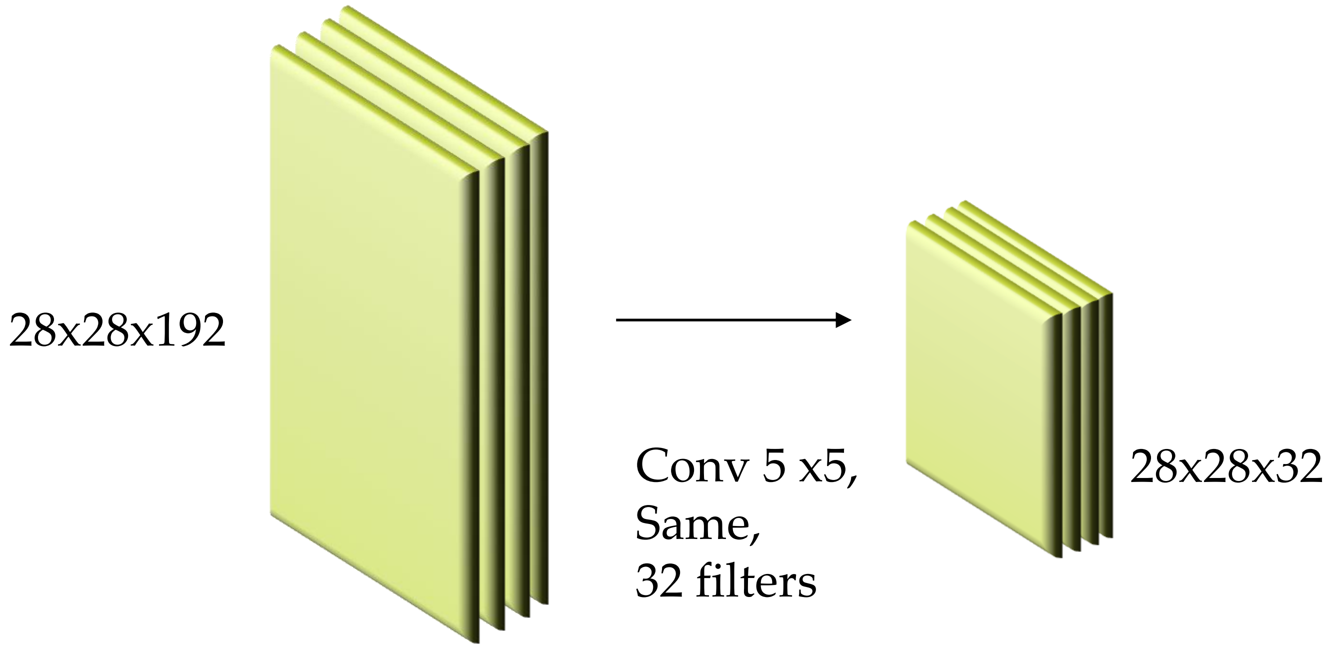
- A way to reduce C = Channels
  - Keep the size down

NIN



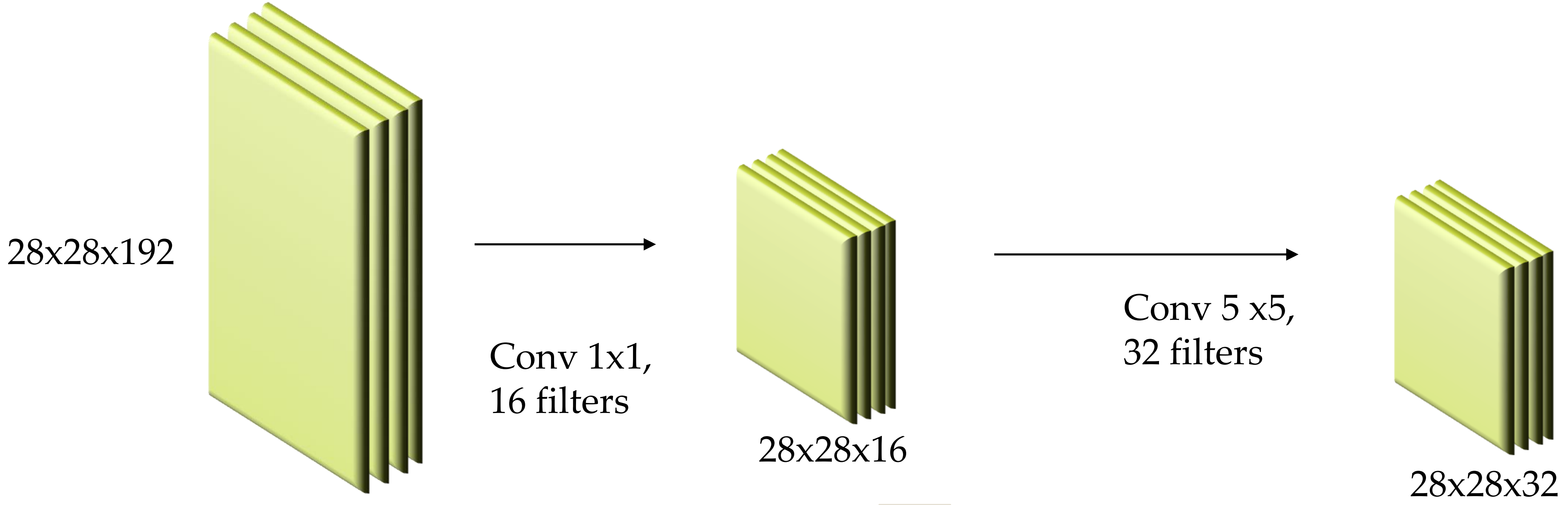
Computation size if we want 10 filters: 6x6x10x1x1x32 = 11520

Computational Cost



28x28x32x5x5x192 = 120 million

1x1 to 5x5

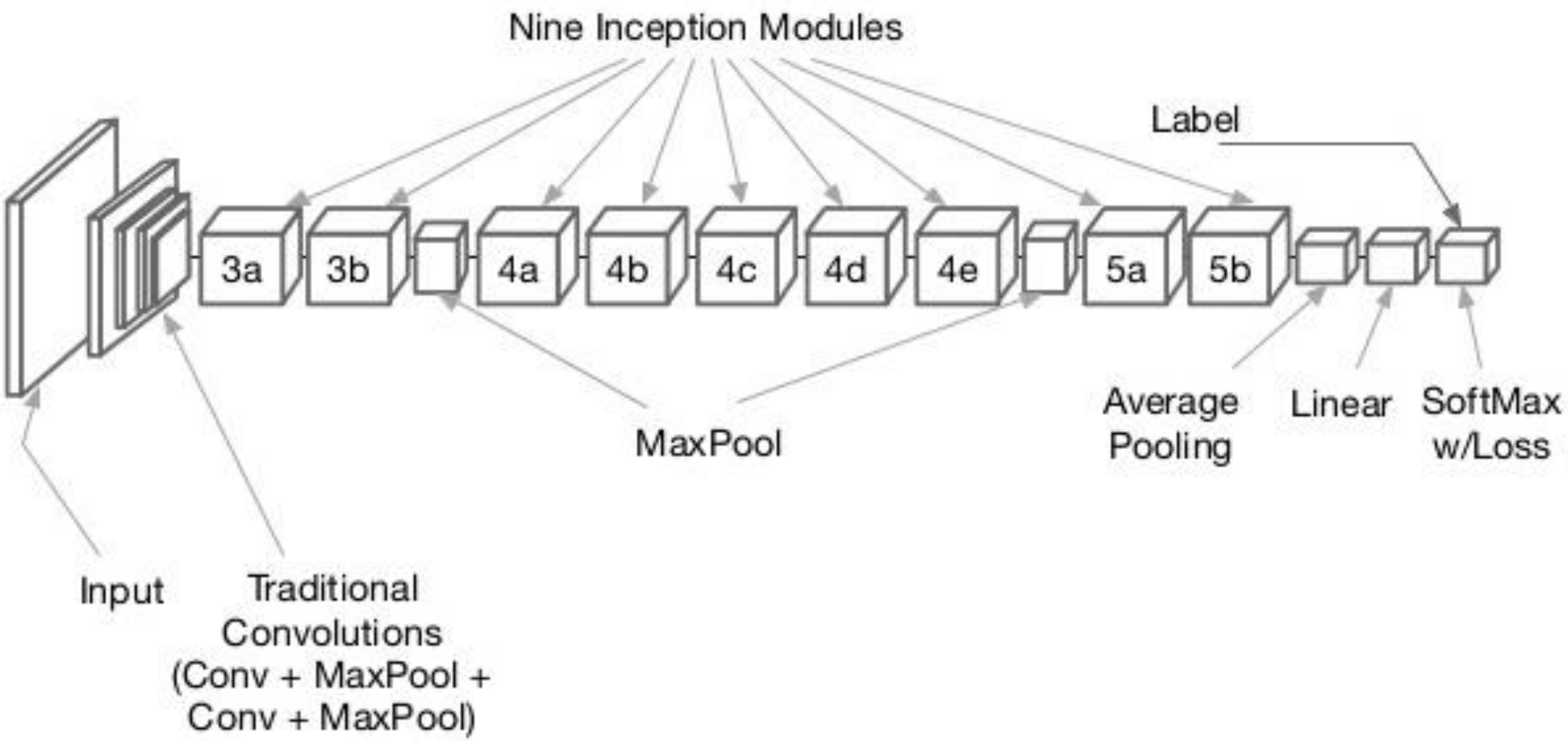


28x28x192x16 = 2.4 million + 28x28x16x5x5x32 = 10million

About 12.4 million

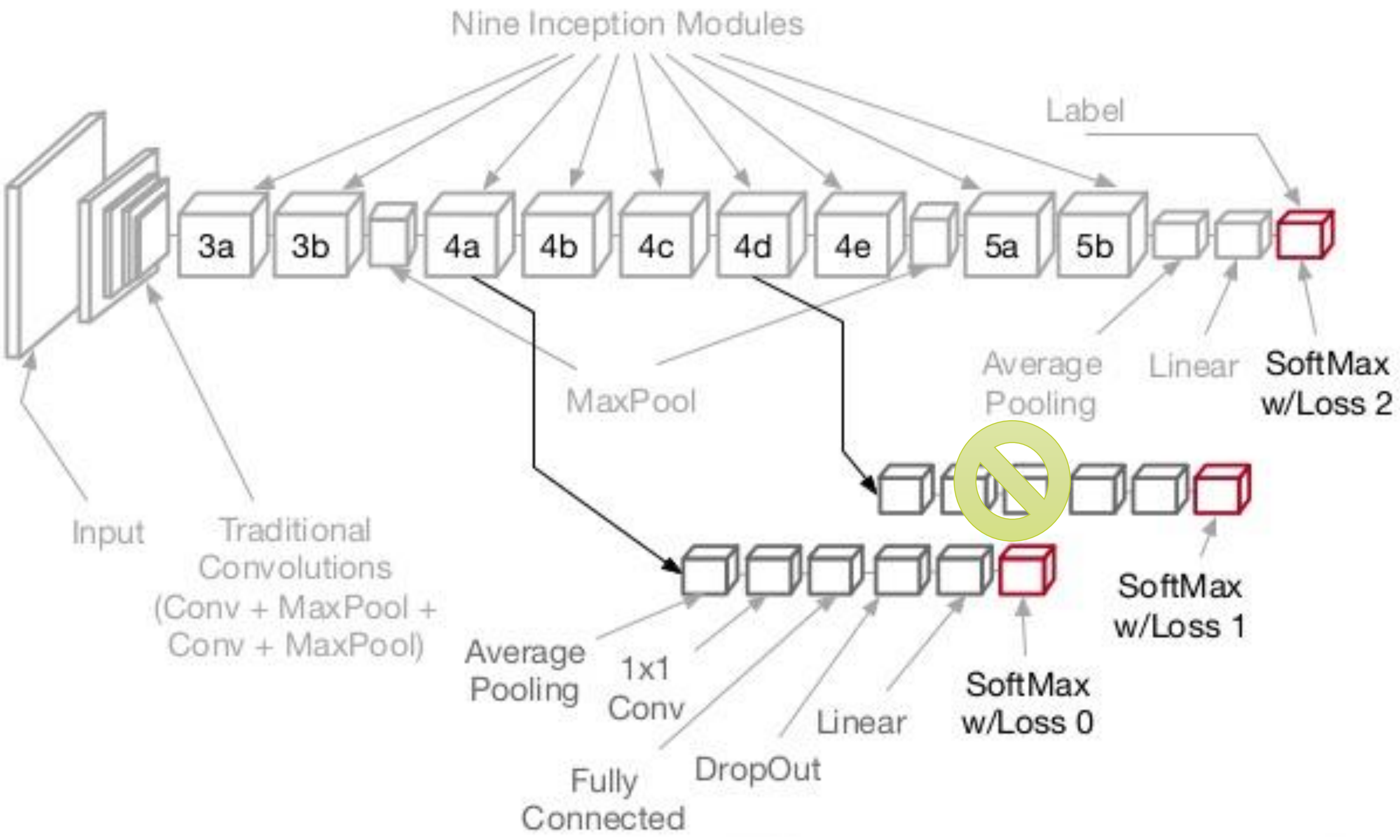
GoogLeNet Components

Stacking Inception Modules

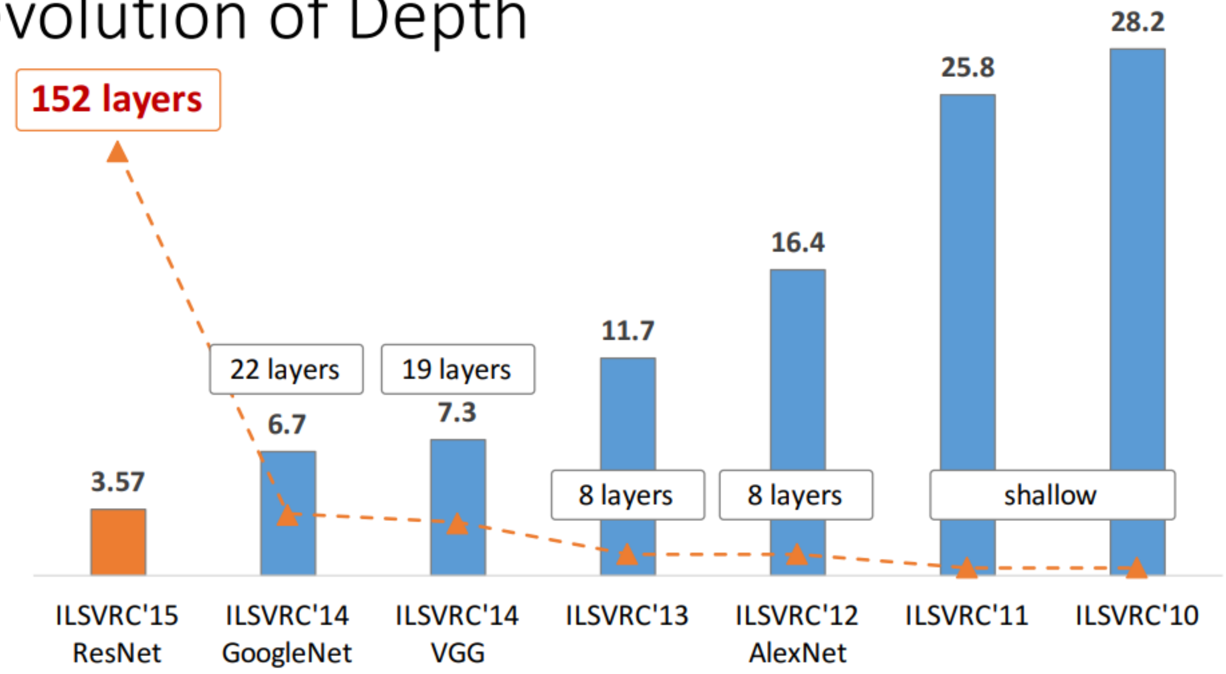


Two Additional Loss Layers

for Training to Depth



Revolution of Depth



ImageNet Classification top-5 error (%)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.