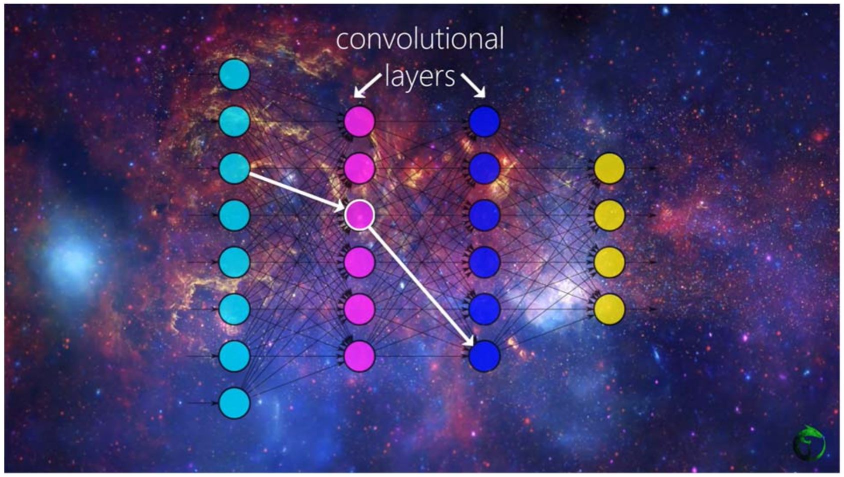
RETHINKING THE INCEPTION ARCHITECTURE FOR COMPUTER VISION

By C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna

PRESENTATION BY EMILY DA

CONVOLUTION NETWORKS

- Filters that detect patterns
- Convolutional layers
- Deeper layers can detect more specific objects like hair, eyes etc, even deeper = full objects like animals, etc
- Need to specify amount of filters
- Image recognition
- Convolution layers : Inputs information, transforms it and outputs



BACKGROUND

- Convolution networks started to become mainstream in 2014- significantly improved
- Success of “AlexNet”, winning entry of an ImageNet competition has helped a large amount of computer vision tasks including:
  -object detection, segmentation, human pose estimation, video classification, object tracking, and superresolution
- The success inspired new research for Convolutional Neural Networks (CNN)
- How do we scale up networks, utilize computation as efficiently as possible?

COMPARISONS

- AlexNet (2012), GooGleNet, VGGNet (2014) all had high performance results
- + classification performance = significant quality gains
- VGGNet has a strong feature of architectural simplicity but the downside is that evaluating the network requires a lot of computation.
- Inception architecture of GooGleNet performed well even with limited memory and a computation limit

INCEPTION ARCHITECTURE

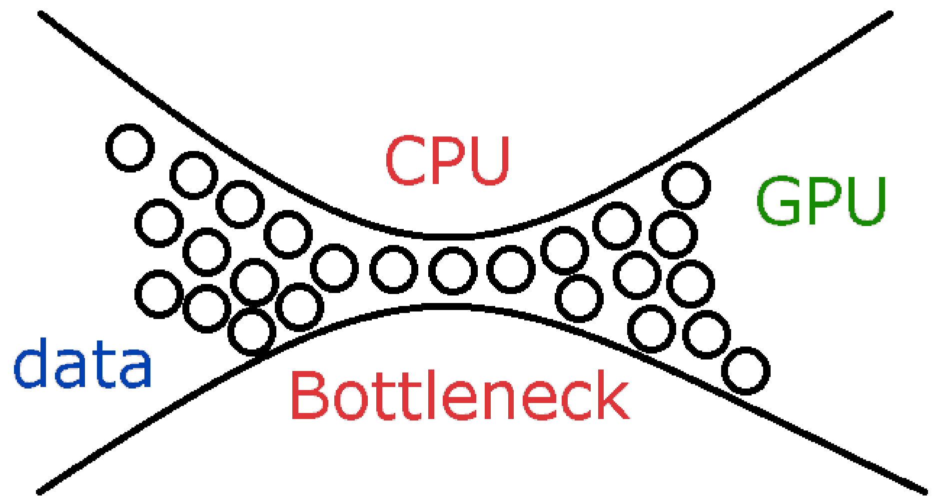
- Computation cost of inception is much lower than VGGNet
- Makes it attractive to use in a big data scenario
- Complex and difficult to make changes
- Double filter bank sizes = 4x computational cost and parameter number

- Goal? We want to find efficient ways to scale up convolution networks
- What are the general principles and optimization ideas? -→ To find out

GOAL

- Increase depth by stacking more convolution layers mean the network can learn more complex features
- Cons to it though..
- Scaling could help it learn more defined features
- We want to find efficient ways to scale up the convolution layers

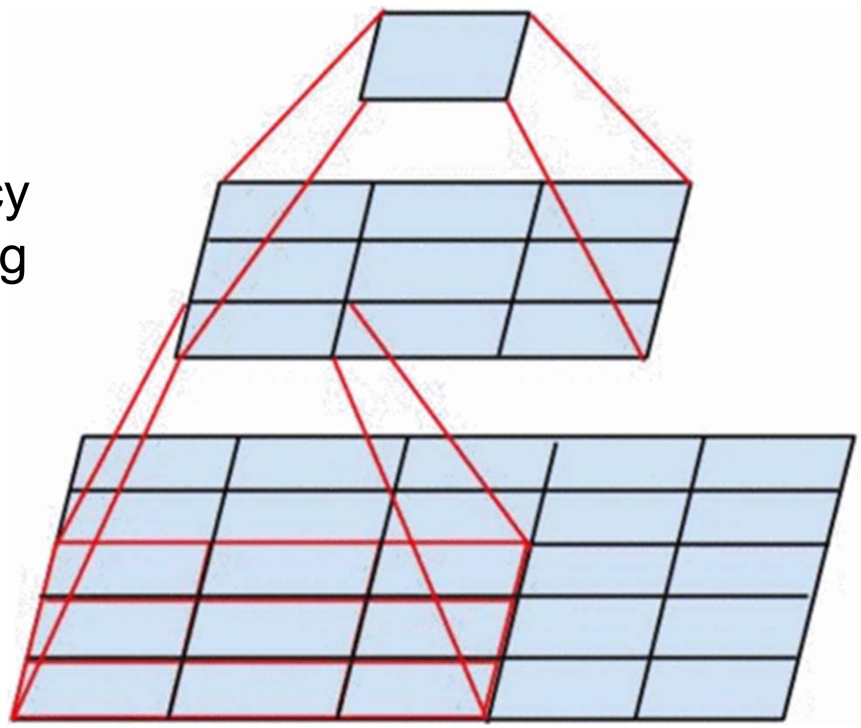
GENERAL DESIGN PRINCIPLES



- 1. Avoid Representational Bottlenecks
- 2. Higher dimensional representations
- 3. Spatial Aggregation
- 4. Balance the Width and Depth of the network

FACTORISING

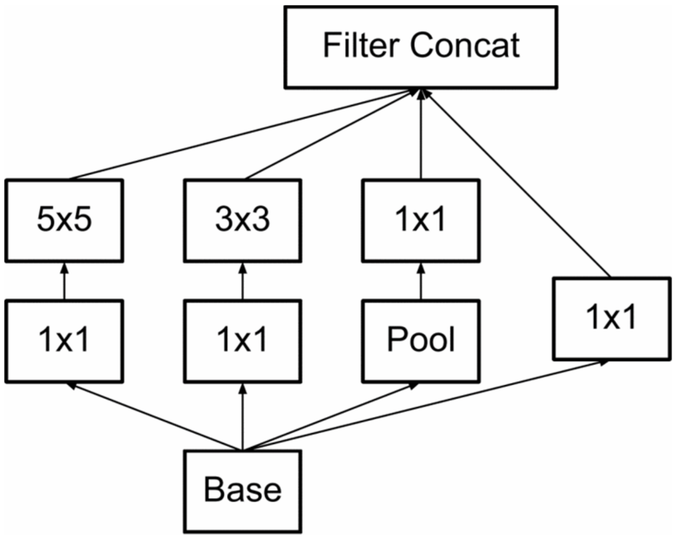
-Increase computational efficiency
-Potentially result in faster training
-Could have more disentagled parameters

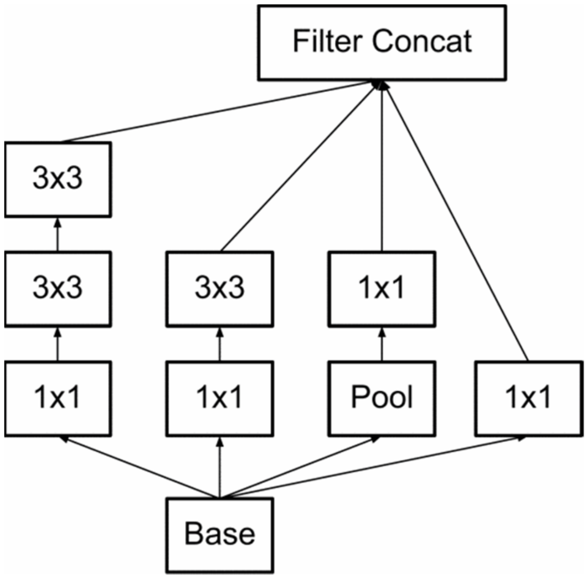


FACTORIZING INTO SMALLER CONVOLUTIONS

- Larger special filters much more expensive. E.g. 5x5 convolution is 2.78 times more expensive than a 3x3
- What about 2 layers of 3x3?
- 18/25x reduction = computational savings + 28% relative gain with that factorization

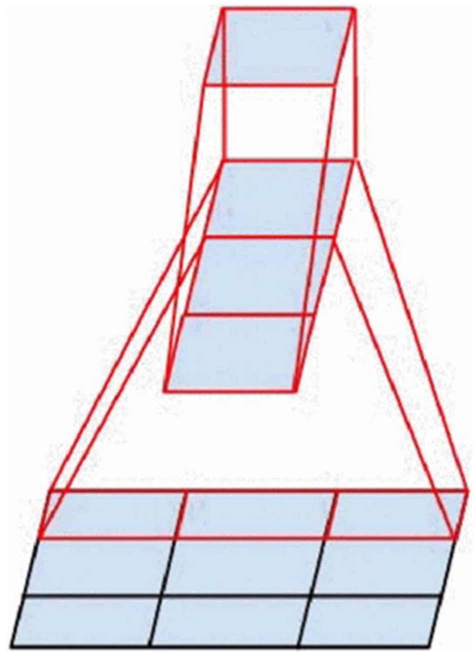
EXAMPLE OF 2 LAYER CONVOLUTION

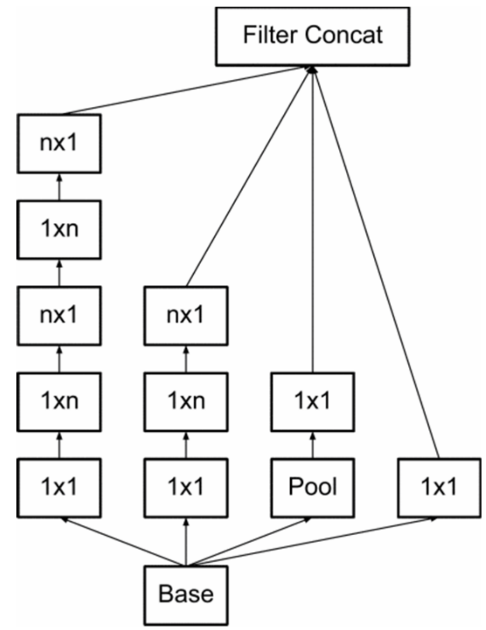




SPACIAL FACTORIZATION INTO ASYMMETRIC CONVOLUTIONS

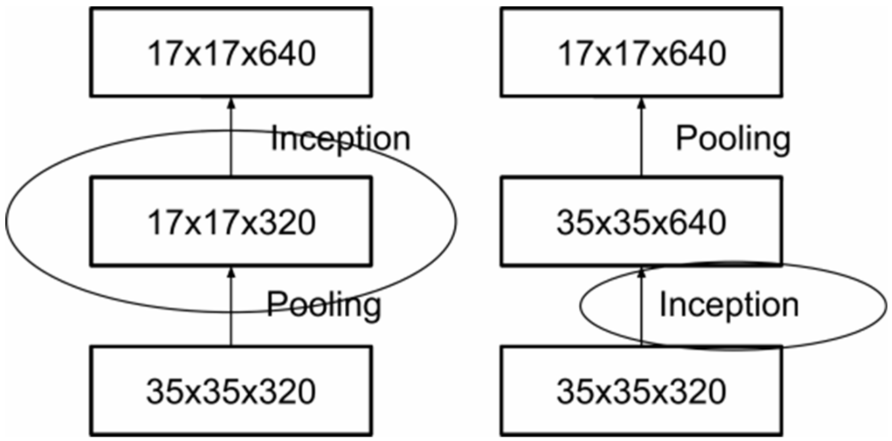
- -Can always be reduced to 3x3
- -2x2? nx1?
- n x1 is very good for medium sized grids.





HOW DO WE REDUCE GRID SIZE EFFICIENTLY?

- Option 1: Use pooling and avoid bottleneck- activation dimensions of the network filters are expanded.
- For d/2 grid size and 2k filter, computational cost is expensive
- Option 2: Pooling with convolution.
- Reduces computational cost by ¼ but creates a bottleneck



MORE EFFICIENT OPTION?

- New architecture proposal- Inception V3
- 3 traditional inception modules : 35 x 35 with 288 filters become 17x 17 with 768 filters
- 5 instances of factorization- reduces to 8x8x1280 grid
- Results in network 42 layers deep, 2.5x more computation cost than GooGleNet but much more efficient, quality stable

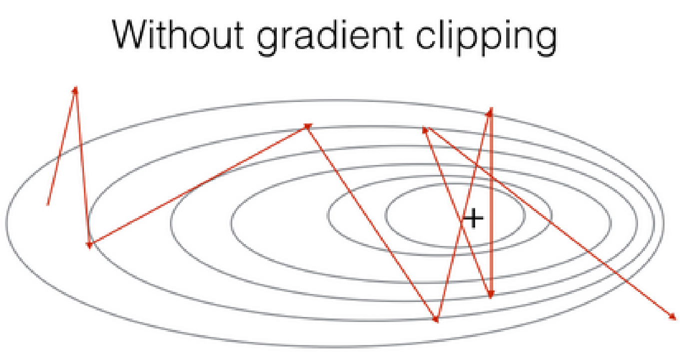
type	patch size/stride or remarks	input size
conv	$3\times3/2$	$299\times299\times3$
conv	$3\times3/1$	$149\times149\times32$
conv padded	$3\times3/1$	$147\times147\times32$
pool	$3\times3/2$	$147\times147\times64$
conv	$3\times3/1$	$73\times73\times64$
conv	$3\times3/2$	$71\times71\times80$
conv	$3\times3/1$	$35\times35\times192$
$3\times$Inception	As in figure 5	$35\times35\times288$
$5\times$Inception	As in figure 6	$17\times17\times768$
$2\times$Inception	As in figure 7	$8\times8\times1280$
pool	8×8	$8\times8\times2048$
linear	logits	$1\times1\times2048$
softmax	classifier	$1\times1\times1000$

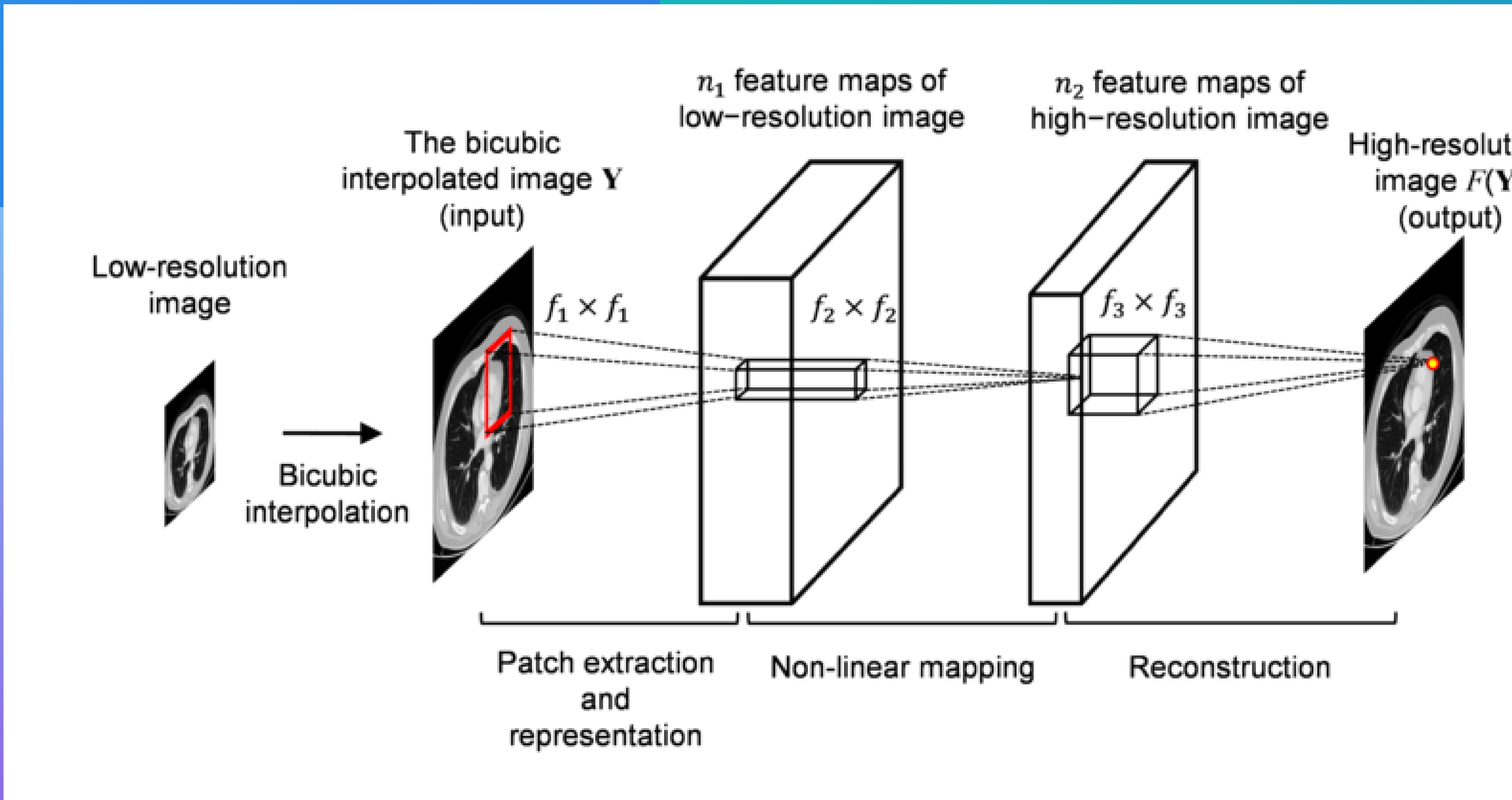
AUXILIARY CLASSIFERS

- Auxiliary classifiers to improve the convergence of very deep networks
- Original plan was to move useful gradients to lower layers so it can be used immediately
- Lee et al (a researcher) claims auxiliary classifies improve learning

TRAINING OUTCOME

- Gradient clipping found useful
- RMSProp an algorithm had the best outcome
- Evaluated using running average computed over time





n_1 feature maps of
low−resolution image
n_2 feature maps of
high−resolution image
The bicubic
interpolated image $\mathbf{Y}$
(input)
High-resoluti
image $F(\mathbf{Y})$
(output)
Low-resolution
image
$f_1 \times f_1$
$f_2 \times f_2$
$f_3 \times f_3$
Bicubic
interpolation
Patch extraction
and
representation
Non-linear mapping
Reconstruction

PROS AND CONS

- Accuracy
- Size and Feature
- Training difficulty
- Computational Cost
- Efficiency

FINDINGS

Network	Top-1 Error	Top-5 Error	Cost Bn Ops
GoogLeNet [20]	29%	9.2%	1.5
BN-GoogLeNet	26.8%	-	**1.5**
BN-Inception [7]	25.2%	7.8	2.0
Inception-v3-basic	23.4%	-	3.8
Inception-v3-rmsprop RMSProp	23.1%	6.3	3.8
Inception-v3-smooth Label Smoothing	22.8%	6.1	3.8
Inception-v3-fact Factorized 7×7	21.6%	5.8	4.8
Inception-v3 BN-auxiliary	**21.2%**	**5.6%**	4.8

Receptive Field Size	Top-1 Accuracy (single frame)
79×79	75.2%
151×151	76.4%
299×299	76.6%

- When computational cost is constant but the receptive field varies- recognition performance

- Compared with the best outcome of GoogLeNet

RELATED WORKS

- Simonyan and Zisserman used deep CNN like Inception. They kept the parameters constant and small sized convolutional filters
- Winner of ILSVRC 2015: They used a residual learning framework
- A new family of CNN: EfficientNet, paper published in 2019

SUMMARY

- Modest computation cost – 2.5x increase
- Less computation compared to some other networks
- Scale up convolutional networks
- Lower parameter count

REFERENCE

- C. Szegedy, V. Vanhoucke, S.Ioffe, J.Shlens, and Z.Wojna, “Rethinking the Inception Architecture for Computer Vision”, Las Vegas, NV, USA. Published in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Available: https://ieeexplore-ieee-org.ezproxy.auckland.ac.nz/document/7780677
- Hashemi, M. “Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation.” *J Big Data* **6,** 98 (2019). Available: https://doi.org/10.1186/s40537-019-0263-7
- C. Szegedy, W.Liu, Y.Jia, P.Sermanet, S. Reed, D. Anguelov, D.Erhan, V.Vanhoucke, A.Rabinovich, “Going Deeper with Convolutions”, University of North Carolina, USA, University of Michigan, USA. Published in 2015 Computer Vision Foundation. Available: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43022.pdf