Going Deeper …

[Slides from Shuicheng Yan (NUS), Christian Szegedy (Google), Karen Simonyan (Oxford)]

Emerging Trend

- Krizhevsky et al. 2012 (AlexNet / SuperVision)
  New benchmark on image classification
- Zeiler & Fergus 2013 (ZFNet – improved AlexNet)
- Dong et al. 2014 (Network In Network – NIN)
  New topology going deeper – 1x1 conv. layers without fully connected layers
- Szegedy et al. 2014-15 (GoogLeNet)
  New topology going deeper - Mixes depth with concatenated inceptions and new topologies
- Simonyan & Zisserman 2014 (VGG Net)
  New topology going deeper – small convolution filters in all layers (3x3)

VGG Net

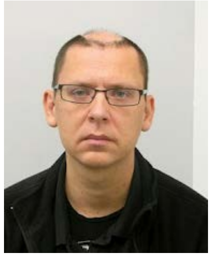
Karen Simonyan
Andrew Zisserman

Oxford University

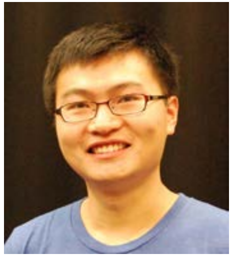
VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

does size matter?

GoogLeNet



Christian Szegedy,
Google



Wei Liu,
UNC

Yangqing Jia,
Google

Pierre Sermanet,
Google

Scott Reed,
University of Michigan

Dragomir Anguelov,
Google

Dumitru Erhan,
Google

Vincent Vanhoucke,
Google

Andrew Rabinovich,
Google

ILSVRC 2014

NIN, Good! (您好)

(Network in Network)

Jian DONG, Min LIN, Yunchao WEI, Qiang CHEN, Shuicheng YAN

NUS
National University
of Singapore

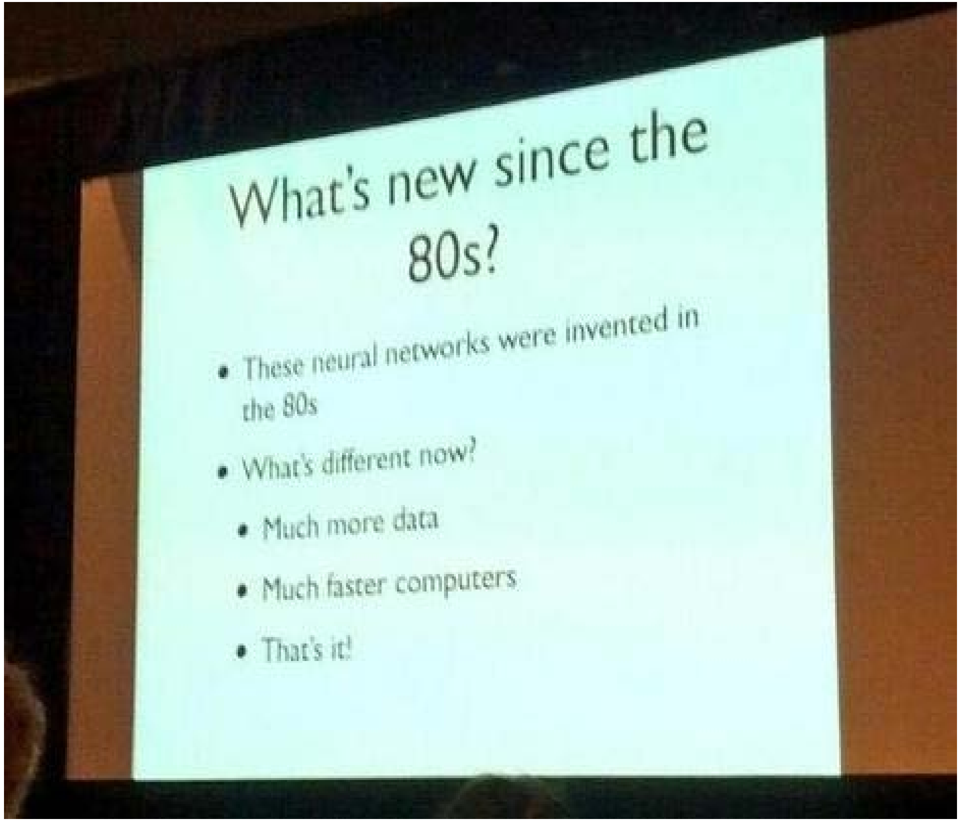
Convolutional Neural Networks

Revolutionizing image classification (computer vision?) since ~~1989~~ 2012

What’s new since 80s?

- Deep learning needs a lot of training data (why?)

- Deep learning needs a lot of computational resources (why?)



Why going deeper (and what stops it)?

- Deep learning needs a lot of training data

- Deep learning needs a lot of computational resources

?

Why going deeper (and what stops it)?

- Deep learning needs a lot of training data
- Deep learning needs a lot of computational resources

Too many parameters to learn

(AlexNet:
60 million parameters /
230 Megabytes memory)

VGG Net

Architecture considerations

- Preprocessing: fixed size image inputs (224x224) and mean subtraction
- Use stacks of small receptive filters (3x3) and (1x1) with 1 pixel convolutional strides
- Spatial preserving padding
- 5 max-pooling layers carried out at 2x2 windows with stride of 2
- Max-pooling only applied to some conv layers

- Observation:
- Drastic change from previous shallower nets with larger receptive fields and strides
- e.g. 11×11 with stride 4 in (Krizhevsky et al., 2012)
- e.g. 7×7 with stride 2 in (Zeiler & Fergus, 2013; Sermanet et al., 2014))

VGG Net

Architecture considerations

- 11 to 19 weight layers
- Conv. layer width increase by factor of 2 after each max-pooling, e.g. 64, 128, 512 …

- Observation:

  Although depth increases, total parameters are loosely conserved compared to a shallower CNN with larger receptive fields (all tested VGG nets <= 144M (Sermanet))

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Observation

Decreases parameters with same effective receptive field

- Consider triple stack of (3x3) filters and a single (7x7) filter
- The two have same effective receptive field (7x7)
- Single (7x7) has parameters proportional to 49
- Triple (3x3) stack has parameters proportional to 3x(3x3) = 27

Going Deeper

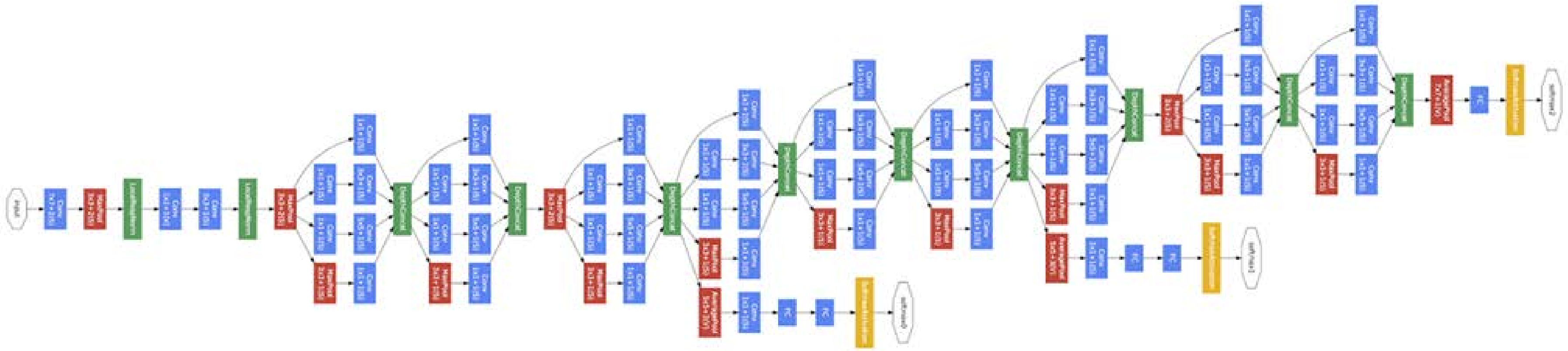
- Additional conv. Layers add non-linearities introduced by the rectification function
- Other small conv. filters: Ciresan et al. (2012), GoogLeNet (Szegedy et al. 2014)

- GoogLeNet going DEEPER – 22 weight layers and more complex topology

- Microsoft Deep Residual Network – 152 weight layers! (8x deeper than VGG but with less complexity / less parameters)

(“Deep Residual Learning for Image Recognition”, Kaiming He, Xiangyu Zhang, Shaoging Ren, Jian Sun, arXiv:1512.03385, December 2015)

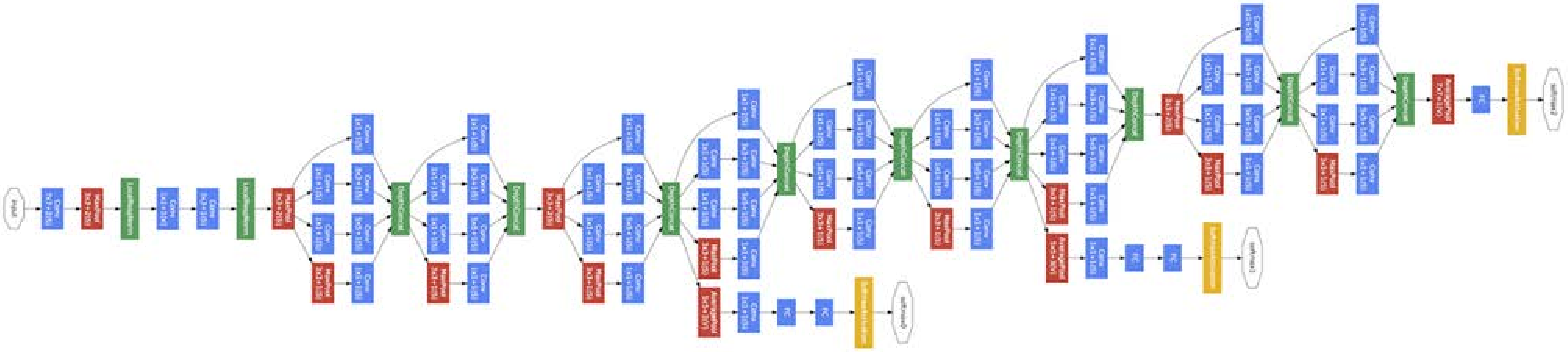
GoogLeNet vs. shallow nets



GoogLeNet

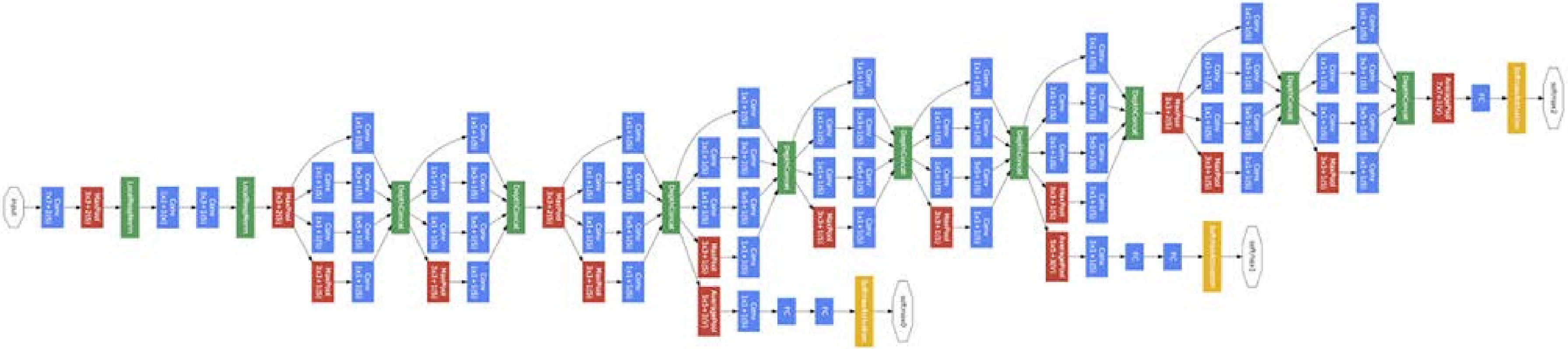
Zeiler-Fergus Architecture (1 tower)

Convolution
Pooling
Softmax
Other



Why does it have so many layers???

Problems with training deep architectures?



Vanishing gradient?
Exploding gradient?
Tricky weight initialization?

Two major challenges for deeper networks

1. Overfit -> Bigger net, more parameters to learn, prone to overfit if not enough data

2. Sparse weights -> Uniformly increase size, introduces lots zeor weights, waste of computation quadratically to the number of weights

"While the theoretical benefits of deep networks in terms of their compactness and expressive power have been appreciated for many decades, until recently researchers had ***little success training deep architectures.****"*

… snip …

"How can we train a deep network? One method that has seen some success is the ***greedy layer-wise training*** *method."*

… snip …

"Training can either be supervised (say, with classification error as the objective function on each step), ***but more frequently it is unsupervised*** *"*

Andrew Ng, 2010 UFLDL tutorial (Unsupervised Feature Learning and Deep Learning)

The rational

- It used to be hard and cumbersome to train deep models due to **sigmoid** nonlinearities, **expensive**.

The rational

- It used to be hard and cumbersome to train deep models due to **sigmoid** nonlinearities, **expensive**.

- Learning deep neural networks are highly non-convex optimisations, no optimality guarantees nor a nice **theory** for architecture design principles.

The rational

- It used to be hard and cumbersome to train deep models due to **sigmoid** nonlinearities, **expensive**.

ReLU

- Learning deep neural networks are highly non-convex optimisations, no optimality guarantees nor a nice **theory** for architecture design principles.

Hebbian Principle

The cost – **Re**ctified **L**inear **U**nit

Glorot, X., Bordes, A., & Bengio, Y. (2011).
Deep sparse rectifier networks
Proceedings 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume (Vol. 15, pp. 315-323).

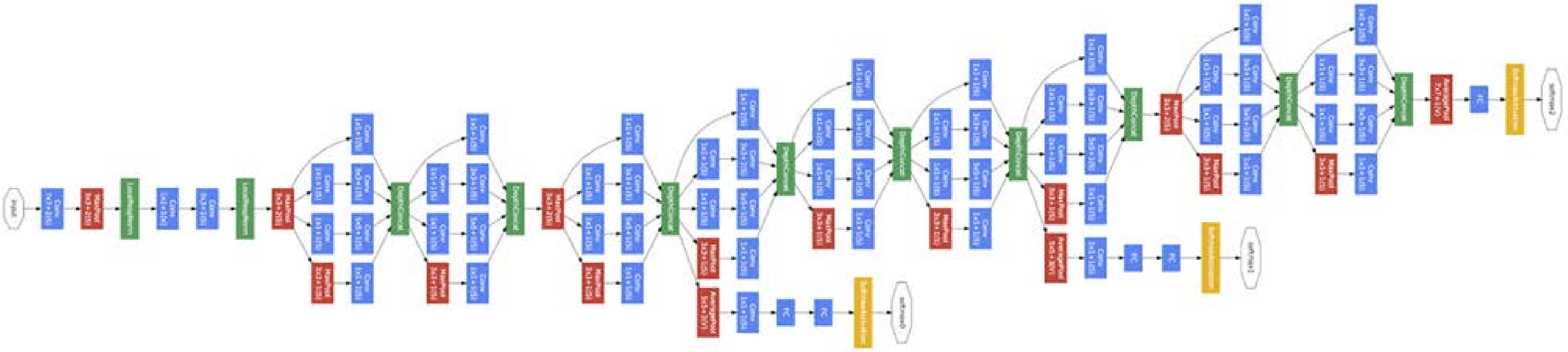
The Theory – Hebbian Principle

Arora, S., Bhaskara, A., Ge, R., & Ma, T.
Provable bounds for learning some deep representations. *ICML 2014*

The Theory – Hebbian Principle

Arora, S., Bhaskara, A., Ge, R., & Ma, T.
Provable bounds for learning some deep representations. *ICML 2014*

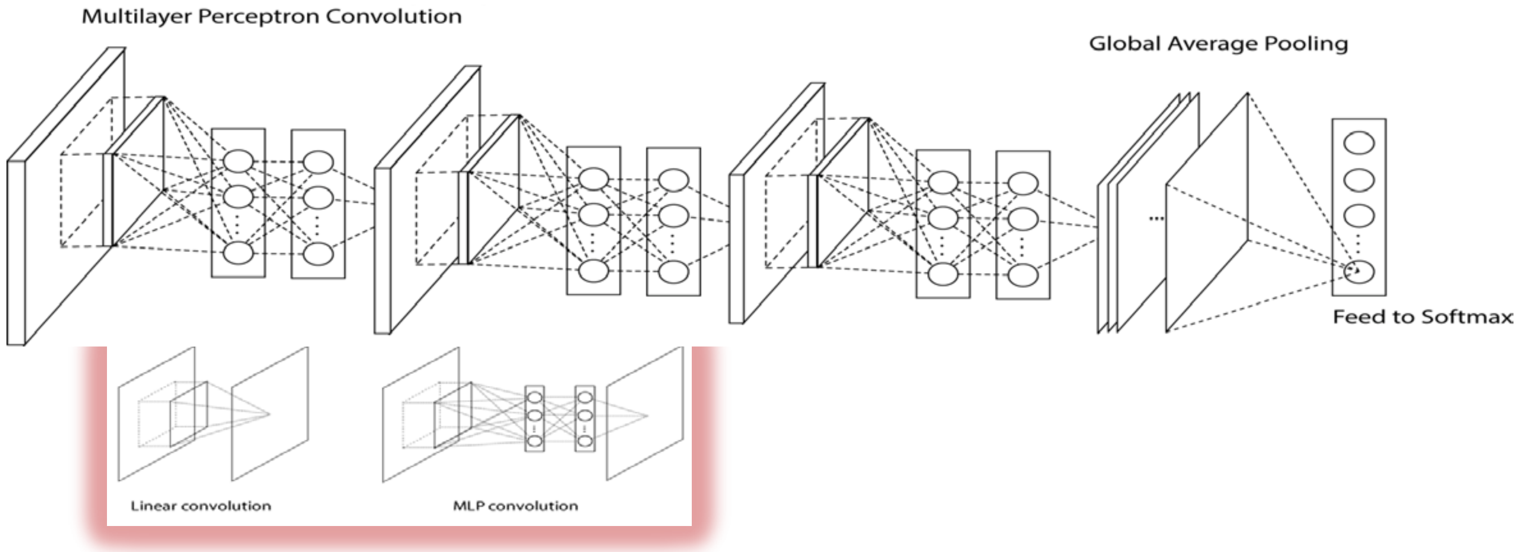
Even non-convex ones!



Why does it have so many layers???

“Network in Network” (NIN)

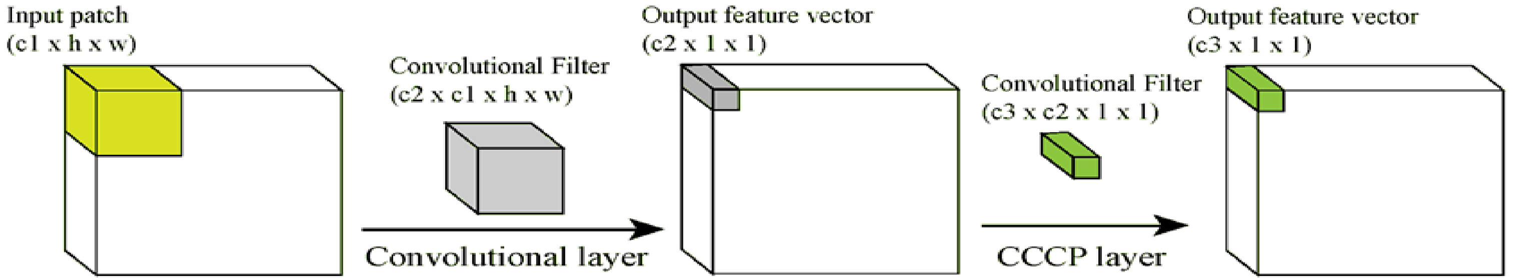
NIN: CNN with non-linear filters, but without fully-connected layers



NIN

Lin, Min, Qiang Chen, and Shuicheng Yan. "Network In Network." arXiv preprint arXiv:1312.4400 (2013) & ICLR-2014

Better Local Abstraction ≈ Cascaded 1x1 Convolution

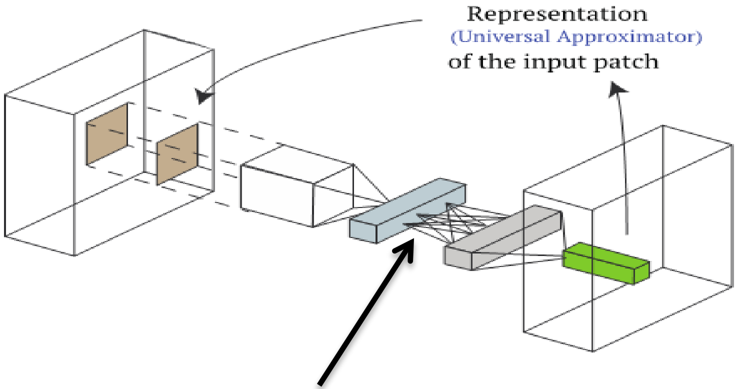


Efficient implementation of CCCP

Local patch is projected to its value in a feature map using **a small network**

$$y_i = \phi(w_i^T y_{i-1} + b_i)$$

$$y_0 = x$$

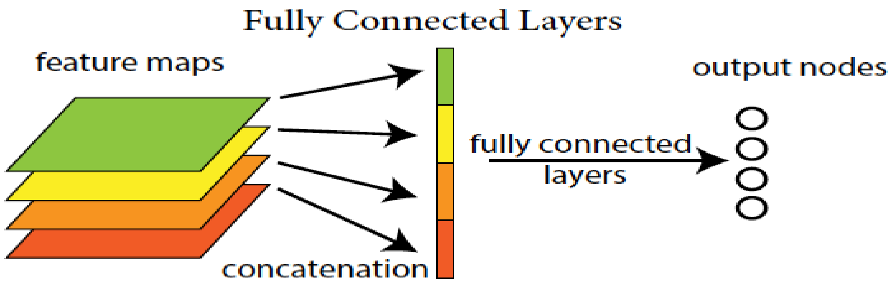


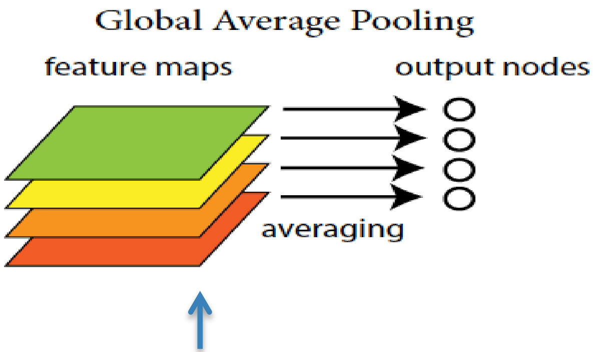
Cascaded Cross Channel Parametric Pooling (CCCP)

Global Average Pooling

CNN

NIN

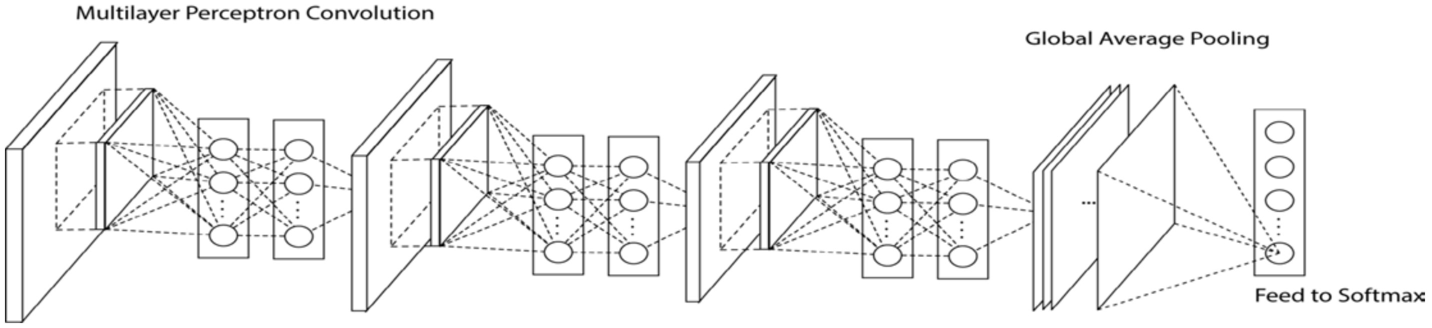




Explicitly confidence map of each category

Save a large portion of parameters

"Network in Network" (NIN) - Overview



NIN

Better local abstraction, **less** global overfitting, and **much less** parameters

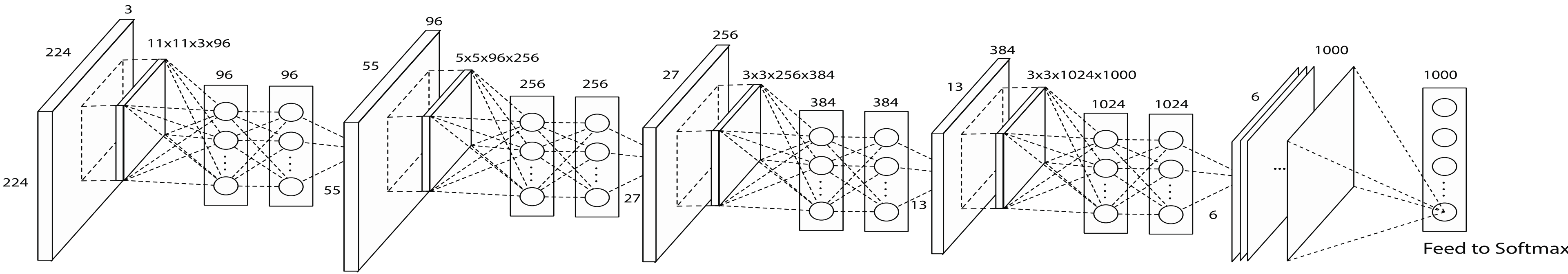
	Cifar-10	Cifar-100
Previous Best performance (Maxout) [1]	11.68%	38.57%
Our method	10.41%	36.30%

With less parameter #

[1] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C. Courville, Yoshua Bengio: Maxout Networks. ICML (3) 2013: 1319-1327

NIN for ImageNet Object Classification

A simple 4 layer NIN + Global Average Pooling:

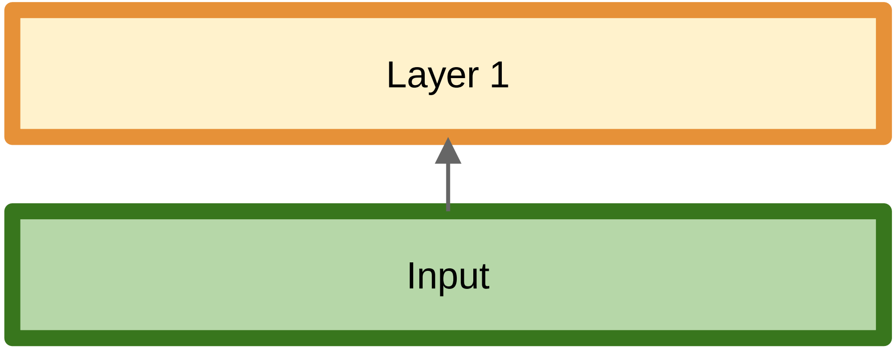


	Parameter Number	Performance	Time to train (GTX Titan)
AlexNet	60 Million (230 Megabytes)	40.7% (Top 1)	8 days
NIN	7.5 Million (**29 Megabytes**)	39.2% (Top 1)	4 days

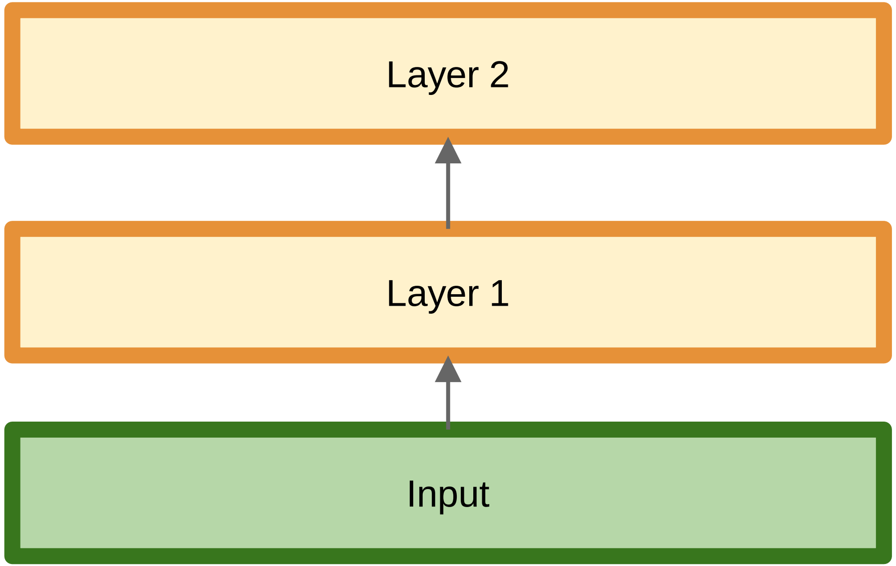
Hebbian Principle

Input

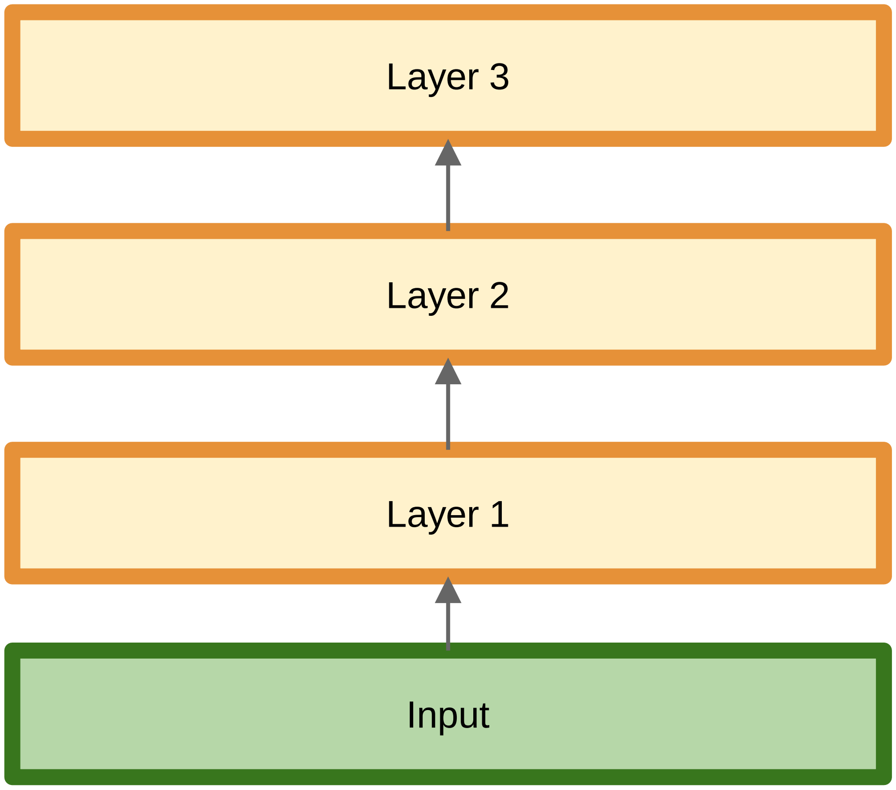
Cluster according activation statistics



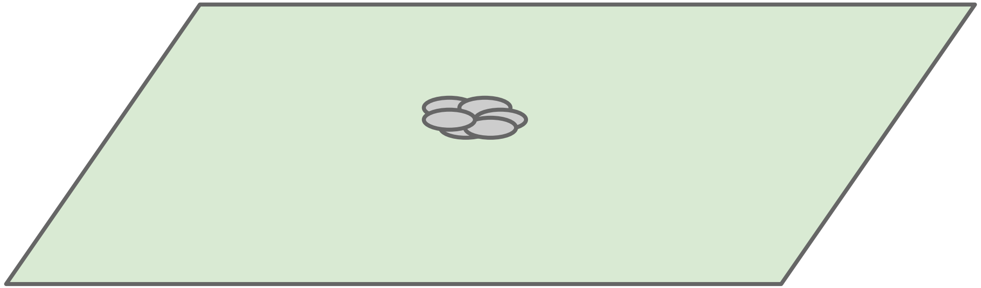
Cluster according correlation statistics



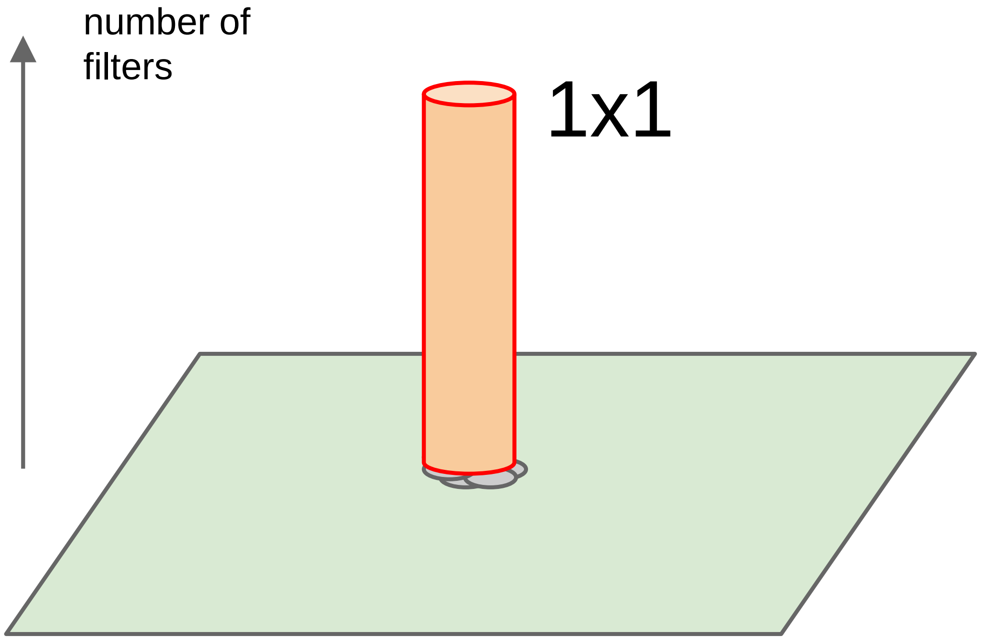
Cluster according correlation statistics



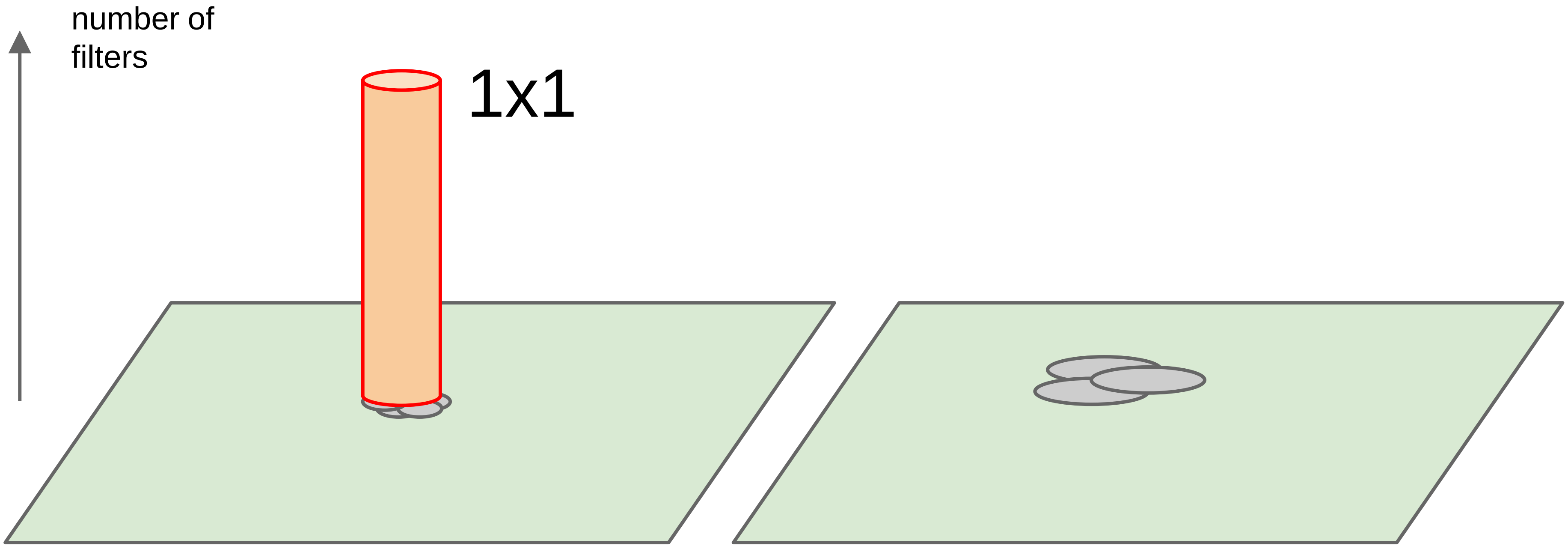
In images, correlations tend to be local



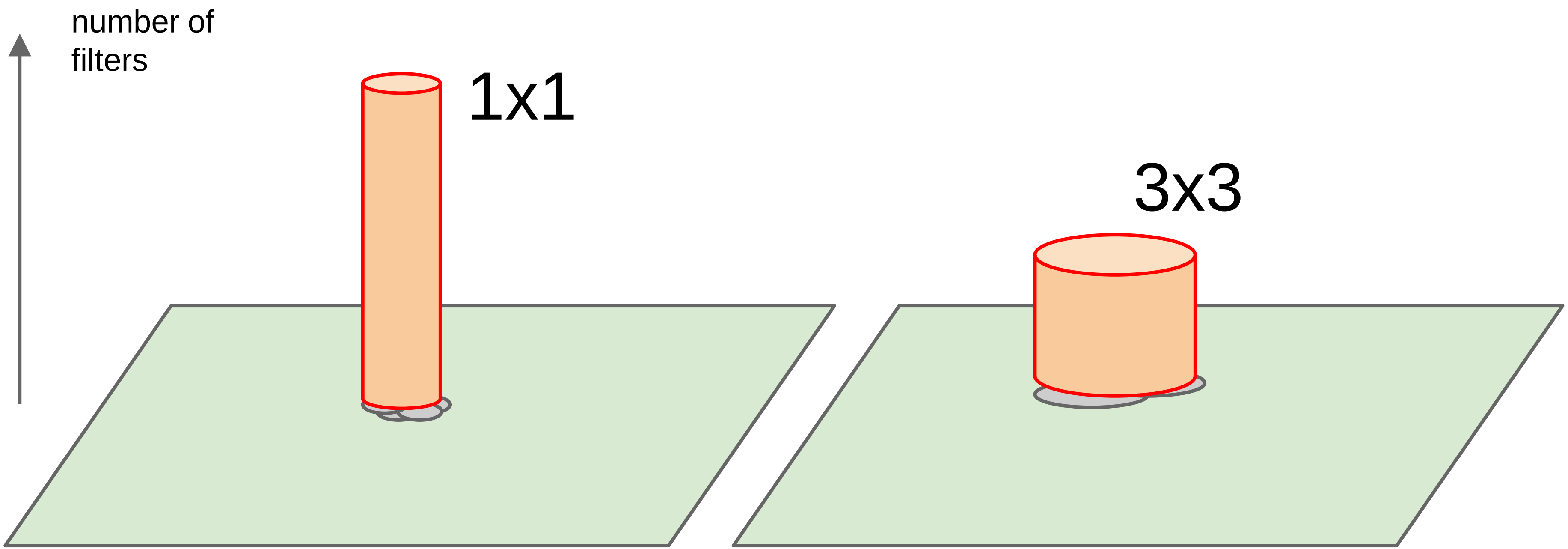
Cover very local clusters by 1x1 convolutions



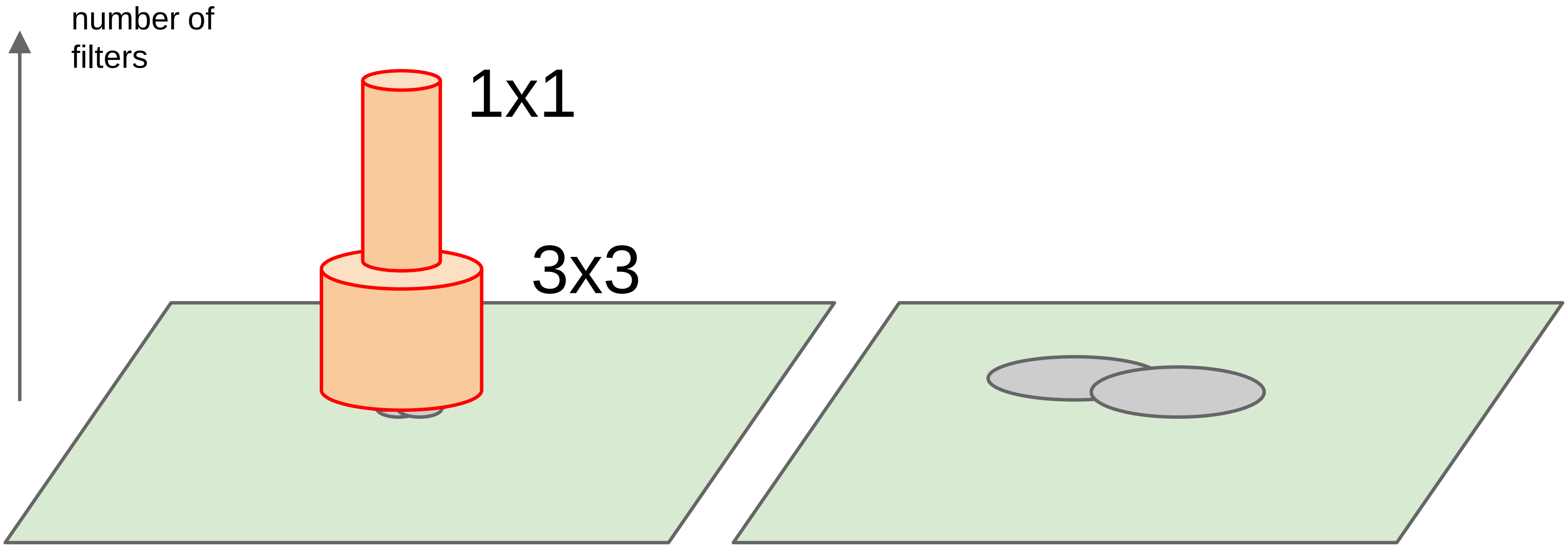
Less spread out correlations



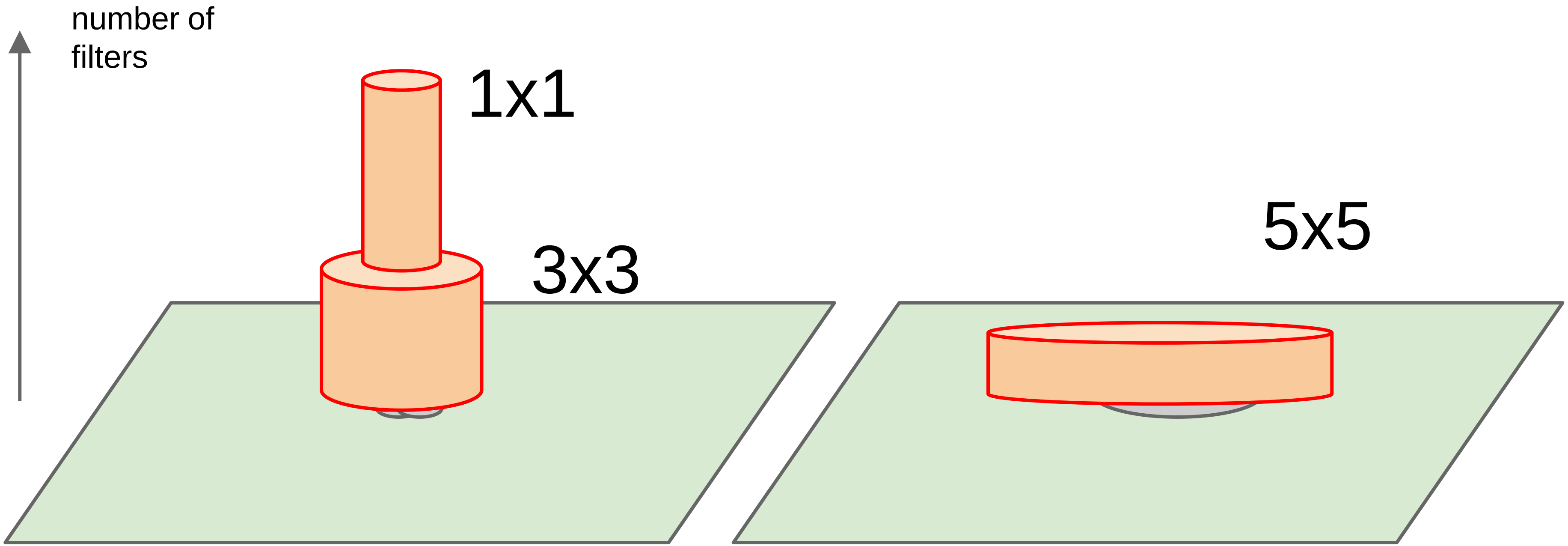
Cover more spread out clusters by 3x3 convolutions



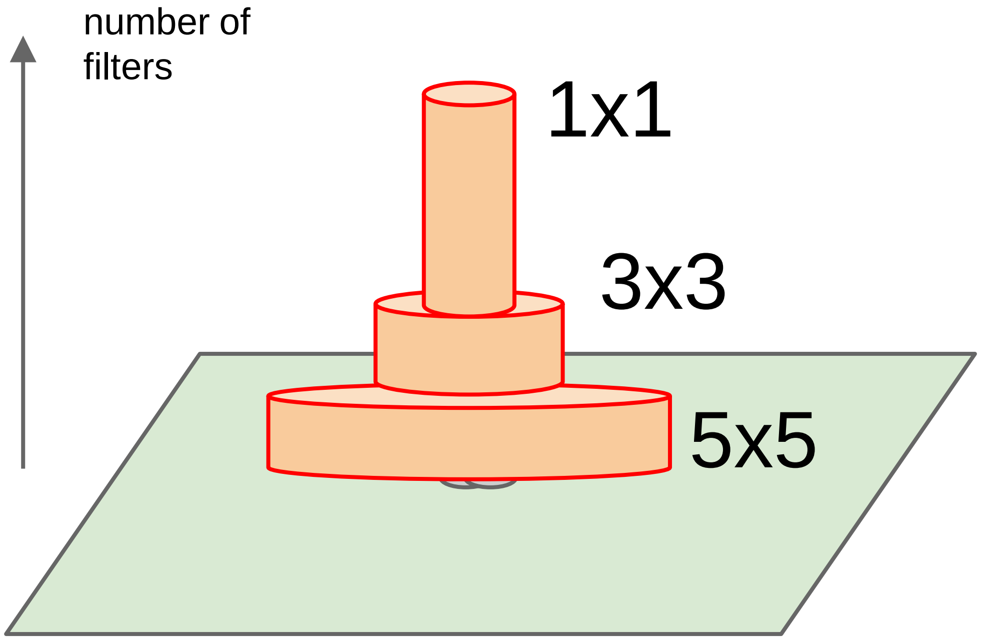
Cover more spread out clusters by 5x5 convolutions



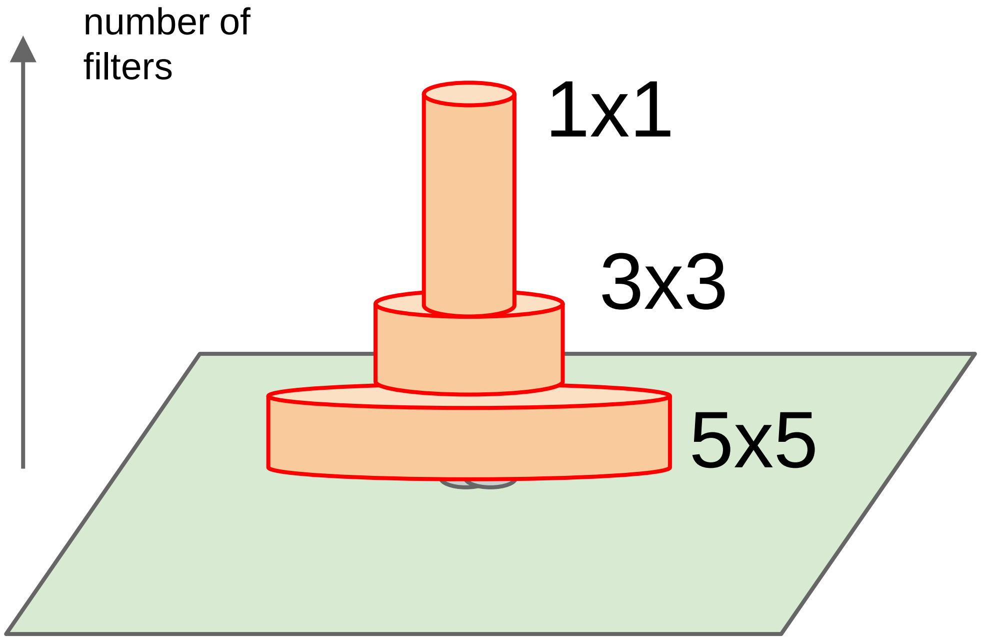
Cover more spread out clusters by 5x5 convolutions

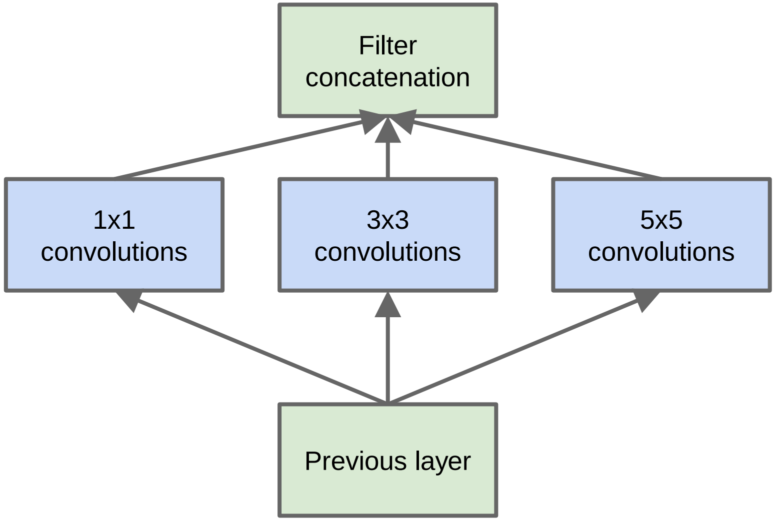


A heterogeneous set of convolutions

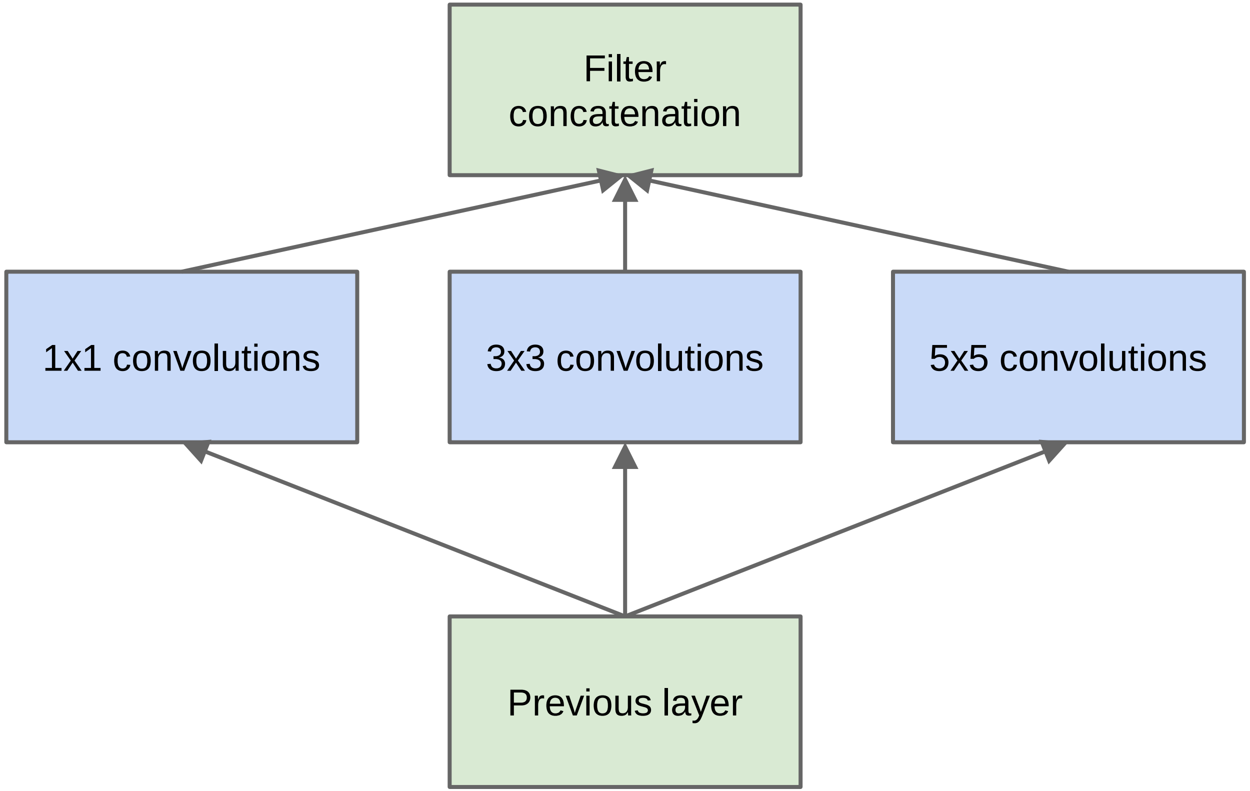


Schematic view (naive version)

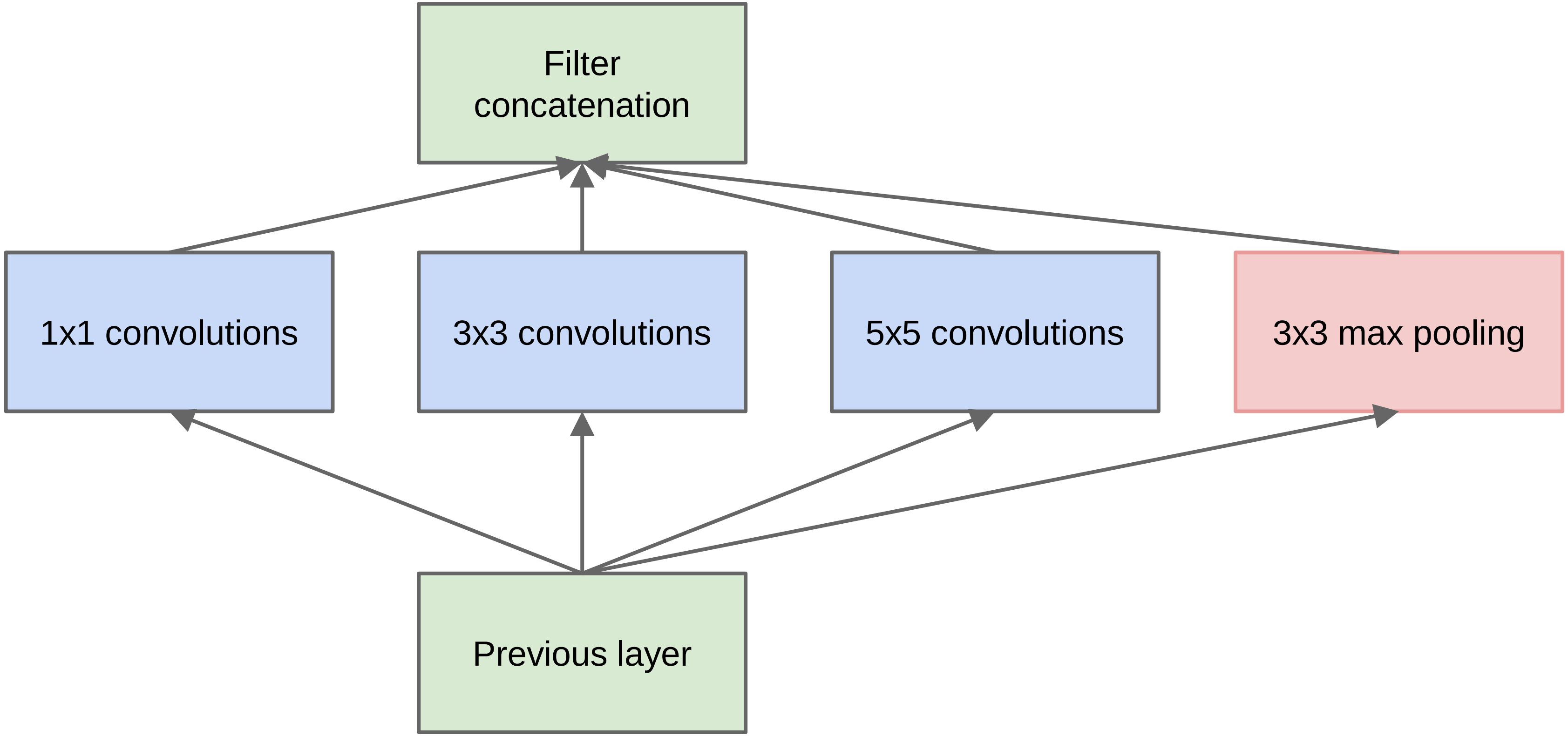




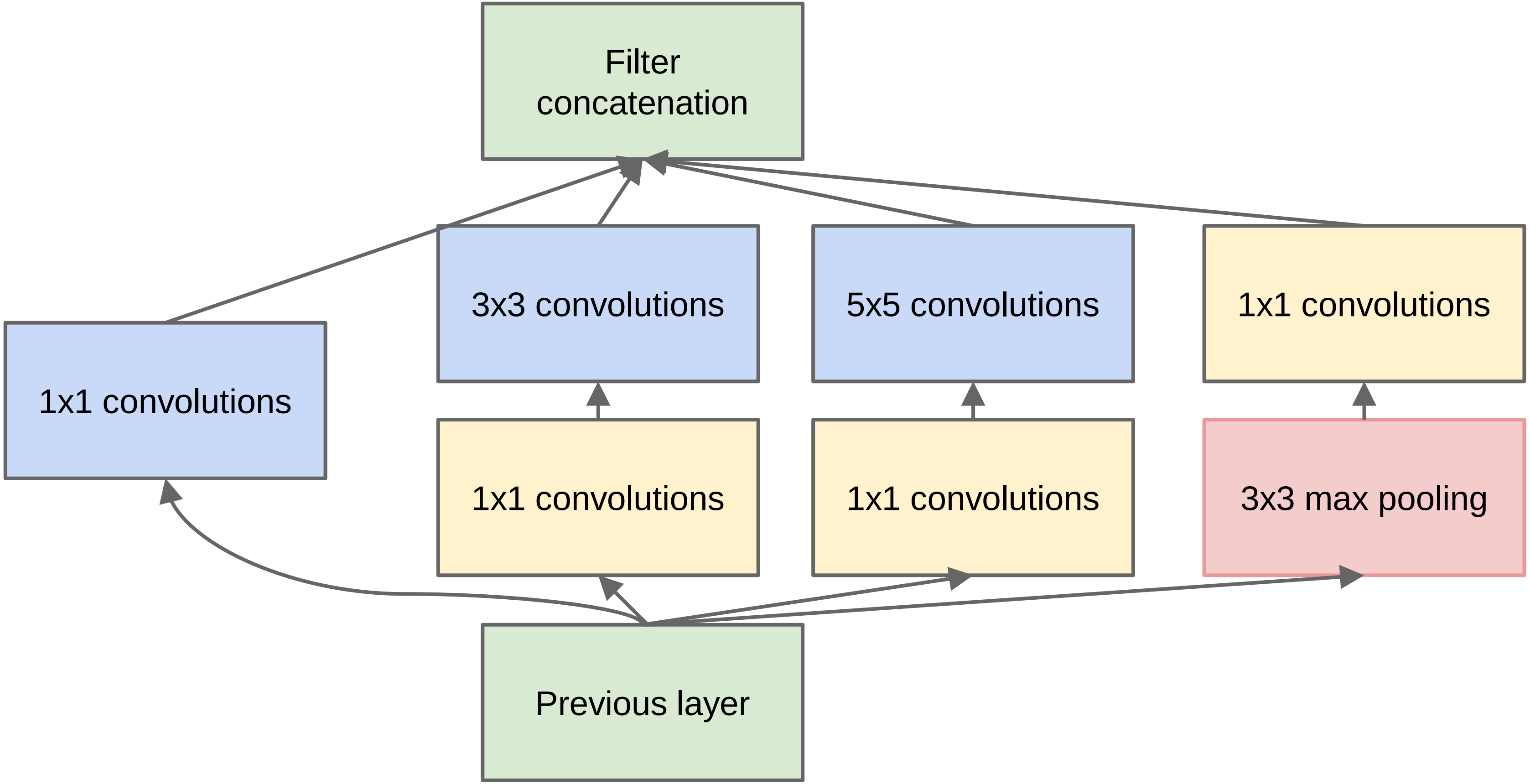
Naive idea


Filter concatenation
1x1 convolutions
3x3 convolutions
5x5 convolutions
Previous layer

Naive idea (**does not work!**)



Inception module



Inception filter design (MLP) – the key ideas

- Implement the Hibbian theory: Optimal inter-layer connection is determined by the correlation statistics – clustering units between layers

- Grouping units into filter banks of multi-scales

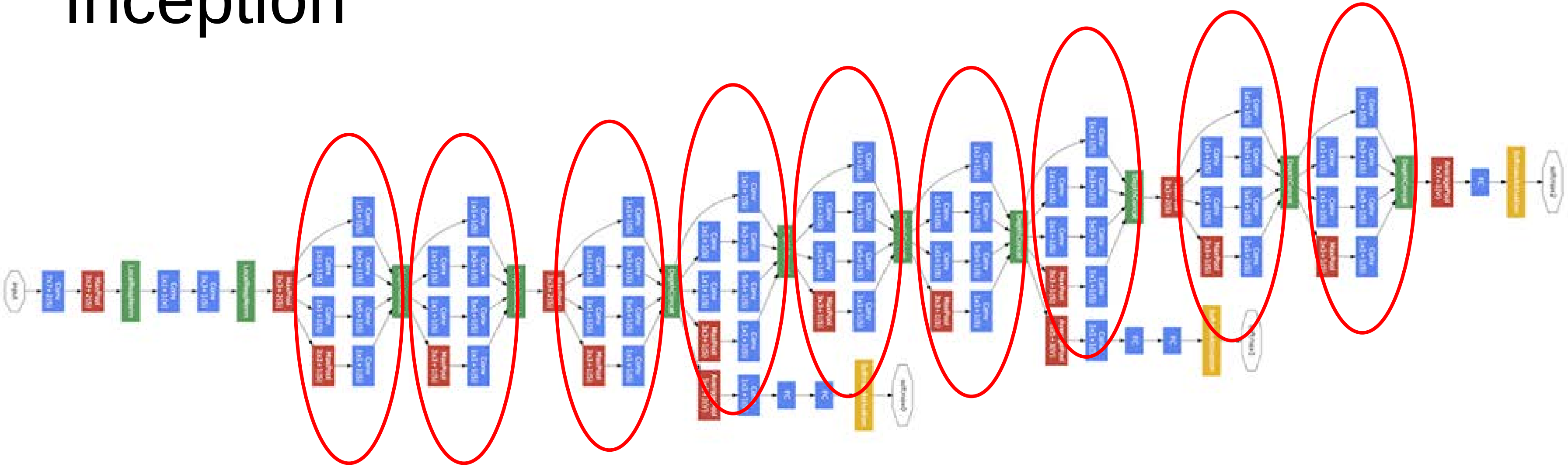
- Reflecting nature of natural images (objects -> small regions / lots of clusters, background -> large regions / less clusters)

- Patch alignment: Earlier layers only 1x1, 3x3; later layers may increase filter size

- 3x3 subsampling (maxpooling) for a single output on the concatenation for combining multi-scale filters

- Dimensionality Reduction (navie combination does not work): To reduce number of parameters when multi-scale, employ 1x1 convolutions before 3x3 and 5x5

Inception



9 **Inception** modules

Network in a network in a network...

Convolution
Pooling
Softmax
Other

Inception

Width of inception modules ranges from 256 filters (in early modules) to 1024 in top inception modules.

Inception

Width of inception modules ranges from 256 filters (in early modules) to 1024 in top inception modules.

Can remove fully connected layers on top completely

Inception

Width of inception modules ranges from 256 filters (in early modules) to 1024 in top inception modules.

Can remove fully connected layers on top completely

Number of parameters is reduced to 5 million
(9 inception layers vs. NIN’s 7.5 million of 4 MLP layers)

Inception

Width of inception modules ranges from 256 filters (in early modules) to 1024 in top inception modules.

Can remove fully connected layers on top completely

Number of parameters is reduced to 5 million
(9 inception layers vs. NIN's 7.5 million of 4 MLP layers)

Computional cost is increased by less than 2X compared to Krizhevsky's network. (<1.5Bn operations/evaluation)

Classification results on ImageNet 2012

Team	**Year**	**Place**	**Error (top-5)**	**Uses external data**
AlexNet	2012	-	16.4%	no
AlexNet	2012	1st	15.3%	ImageNet 22k
Clarifai	2013	-	11.7%	no
Clarifai	2013	1st	11.2%	ImageNet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Detection

- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). **Rich feature hierarchies for accurate object detection and semantic segmentation**. *arXiv preprint arXiv:1311.2524*.

Detection

- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). **Rich feature hierarchies for accurate object detection and semantic segmentation**. *arXiv preprint arXiv:1311.2524*.
- Improved proposal generation:
    - Increase size of super-pixels by 2X
        - coverage 92% $\longrightarrow$ 90%
        - number of proposals: 2000/image $\longrightarrow$ 1000/image

Detection

- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). **Rich feature hierarchies for accurate object detection and semantic segmentation**. *arXiv preprint arXiv:1311.2524*.
- Improved proposal generation:
  - Increase size of super-pixels by 2X
    - coverage 92% → 90%
    - number of proposals: 2000/image → 1000/image
  - Add multibox* proposals
    - coverage 90% → 93%
    - number of proposals: 1000/image → 1200/image

*Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. **Scalable Object Detection using Deep Neural Networks**. *CVPR 2014*

Detection

- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). **Rich feature hierarchies for accurate object detection and semantic segmentation**. *arXiv preprint arXiv:1311.2524*.
- Improved proposal generation:
  - Increase size of super-pixels by 2X
    - coverage 92% → 90%
    - number of proposals: 2000/image → 1000/image
  - Add multibox* proposals
    - coverage 90% → 93%
    - number of proposals: 1000/image → 1200/image
  - Improves mAP by about 1% for single model.

*Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. **Scalable Object Detection using Deep Neural Networks**. *CVPR 2014*

Detection Results

Team	Year	Place	mAP	external data	ensemble	contextual model	approach
UvA-Euvision	2013	1st	22.6%	none	?	yes	Fisher vectors
Deep Insight	2014	3rd	40.5%	ILSVRC12 Classification + Localization	3 models	yes	ConvNet
CUHK DeepID-Net	2014	2nd	40.7%	ILSVRC12 Classification + Localization	?	no	ConvNet
GoogLeNet	2014	1st	43.9%	ILSVRC12 Classification	6 models	no	ConvNet

GoogLeNet – the key ideas

- Going deeper in both depth & width – How?
- Borrowing Network In Network concept – 1x1 conv. for more depth less connectivity to minimise weights / parameters
- Hebbian principle – Learnable convolution filter kernel (not predefined fix kernels), for multiscale and sparsity, the Inception Kernel
- Borrowing R-CNN concept of two-staged processes: CV weak features (cheap) for loci nominations + DL strong features (expensive) for multi-classification

Classification failure cases

Groundtruth: **????**

Classification failure cases

Groundtruth: **coffee mug**

Classification failure cases

Groundtruth: **coffee mug**

GoogLeNet:

- **table lamp**
- **lamp shade**
- **printer**
- **projector**
- **desktop computer**

Classification failure cases

Groundtruth: ???

Classification failure cases

Groundtruth: **Police car**

Classification failure cases

Groundtruth: **Police car**

GoogLeNet:

- **laptop**
- **hair drier**
- **binocular**
- **ATM machine**
- **seat belt**

Classification failure cases

Groundtruth: **???**

Classification failure cases

Groundtruth: hay

Classification failure cases

Groundtruth: **hay**

GoogLeNet:

- **Sorrel (horse)**
- **Hartebeest (deer)**
- **Arabian camel**
- **Warthog (boar)**
- **Gaselle**