UiO : Department of Informatics
University of Oslo

IN5400 Machine learning for image classification

Lecture 5 : Convolutional neural networks

Tollef Jahren

February 13, 2019

About today

- Naming convention: Convolutional neural network, ConvNet, CNN
- What is a convolutional neural network?
- The required computation in a convolutional neural network
- Considerations when designing an convolution neural network architecture

Outline

- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

Readings

- **Text:**
- http://cs231n.github.io/convolutional-networks/

- **Video:**
- https://www.youtube.com/watch?v=bNb2fEVKeEo&index=5&list=PLC1qU-LWwrF64f4QKQT-Vg5Wr4qEE1Zxk

- **Optional text:**
  - Receptive field: http://www.cs.toronto.edu/~wenjie/papers/nips16/top.pdf
  - Visualizing and Understanding CNN: https://cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf
  - Dilated convolutions: https://arxiv.org/abs/1511.07122

- **Optional videos:**
- https://www.youtube.com/watch?v=ghEmQSxT6tw
- https://www.youtube.com/watch?v=SQ67NBCLV98

Progress

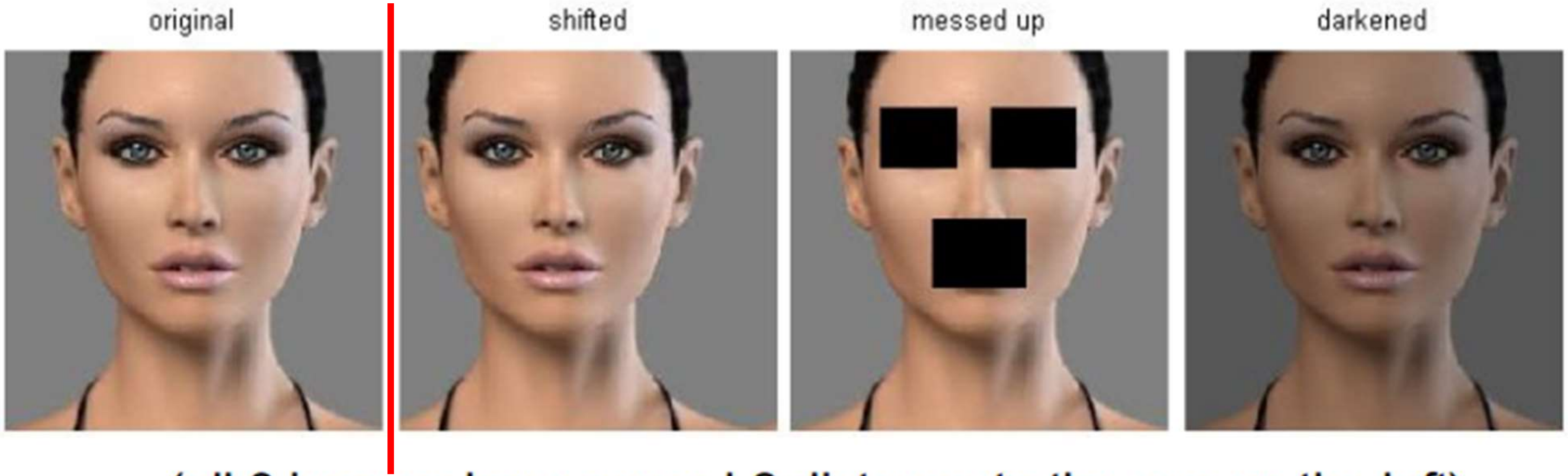
- **Challenges with image classification**
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

Challenges with image classification

- Build invariance:
  - Translation
  - Occlusion
  - Illumination
  - View angle variations
  - Deformation
  - Background Clutter
  - Interclass variation

Building invariance

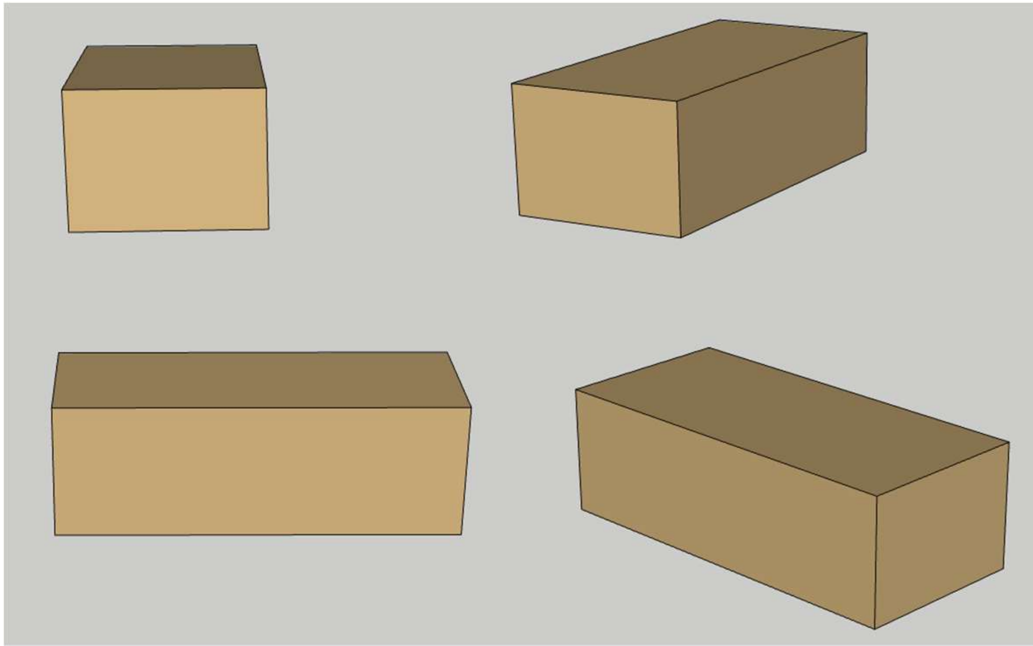
- Translation
- Occlusion
- Illumination



(all 3 images have same L2 distance to the one on the left)

Building invariance

- View angle variations



Building invariance

- Deformation

Building invariance

- Background Clutter

Building invariance

- Interclass variation

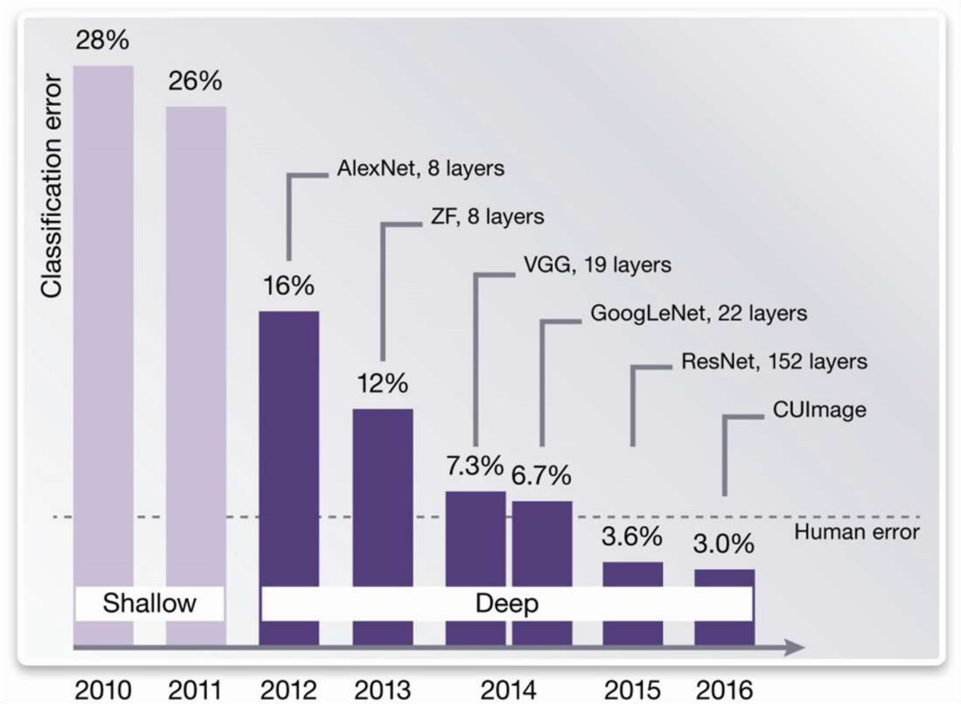
Progress

- Challenges with image classification
- **Benchmark: ImageNet**
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

The ImageNet challenge

- **The images classification challenge**

- **Dataset**
  - 1,431,167 images
  - 1,000 classes



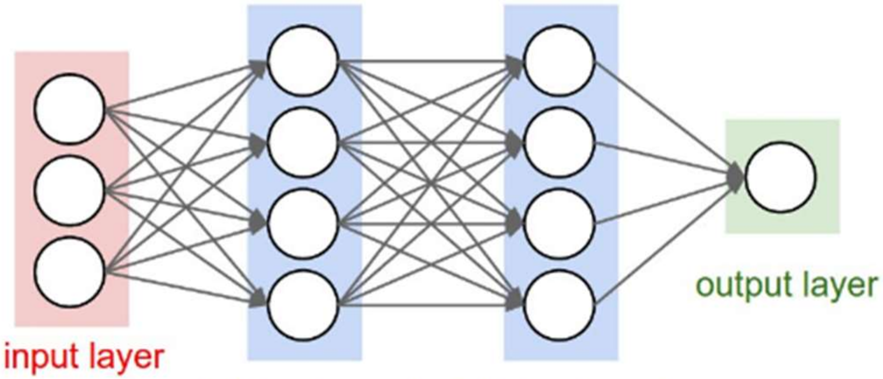
Progress

- Challenges with image classification
- Benchmark: ImageNet
- **Fully connected neural network on images**
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

Fully connected neural network on images

- Most image applications are absolute position invariant.
- A fully connected network will have too many parameters and not able to scale to normal size images and generalize

$$z^1 = W^T x$$
$$a^1 = g(z^1)$$



Progress

- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- **Convolutional layer**
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

Convolutional vs correlation

- **Note**: We will be using **cross correlation**, although we will call it **convolution**. As the network weights are learned there is no real difference.

- 2D cross correlation:

$$z[p,q] = \mathrm{w} \star x = \sum_{r=-K}^{K} \sum_{s=-K}^{K} w[r,s] \cdot x[p+r, q+s]$$

- 2D convolution:

$$z[p,q] = \mathrm{w} * x = \sum_{r=-K}^{K} \sum_{s=-K}^{K} w[r,s] \cdot x[p-r, q-s]$$

https://www.youtube.com/watch?v=C3EEy8adxvc

Convolution example:

- Input image x with shape $[4, 4]$
- Weight matrix w with shape $[3, 3]$
- Output feature map z with shape $[2, 2]$

$$z[p,q] = \mathrm{w} \star x = \sum_{r=-K}^{K} \sum_{s=-K}^{K} w[r,s] \cdot x[p+r, q+s]$$

$x =$

1	2	3	4
2	4	1	2
1	3	2	1
1	2	3	1

$w =$

1	2	1
2	1	2
1	2	1

$z =$

?	?
?	?

Convolution example:

$$x = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 2 \\ 1 & 3 & 2 & 1 \\ 1 & 2 & 3 & 1 \end{bmatrix} \qquad w = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad z = \begin{bmatrix} 27 & \\ & \end{bmatrix}$$

$$\begin{aligned} z[0,0] &= 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 1 \\ &+ \ 2 \cdot 2 + 4 \cdot 1 + 1 \cdot 2 \\ &+ \ 1 \cdot 1 + 3 \cdot 2 + 2 \cdot 1 \\ &= 27 \end{aligned}$$

$x =$

1	2	3	4
2	4	1	2
1	3	2	1
1	2	3	1

$w =$

1	2	1
2	1	2
1	2	1

$z =$

27	

1	2	3	4
2	4	1	2
1	3	2	1
1	2	3	1

1	2	1
2	1	2
1	2	1

27	33

1	2	3	4
2	4	1	2
1	3	2	1
1	2	3	1

1	2	1
2	1	2
1	2	1

27	33
29	

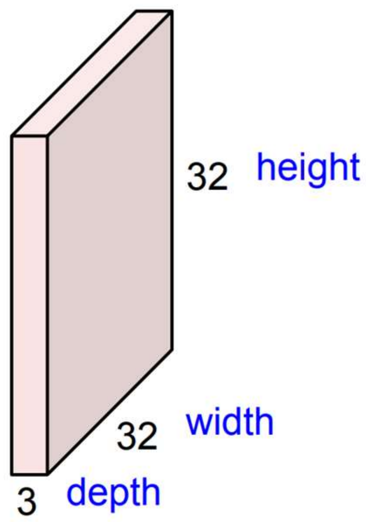
1	2	3	4
2	4	1	2
1	3	2	1
1	2	3	1

1	2	1
2	1	2
1	2	1

27	33
33	29

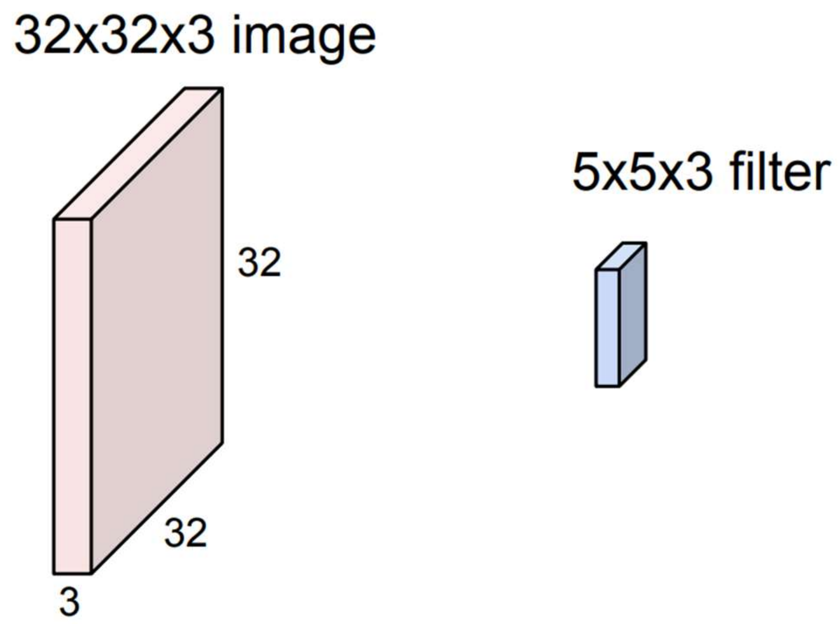
Convolutional layer

32x32x3 image -> preserve spatial structure



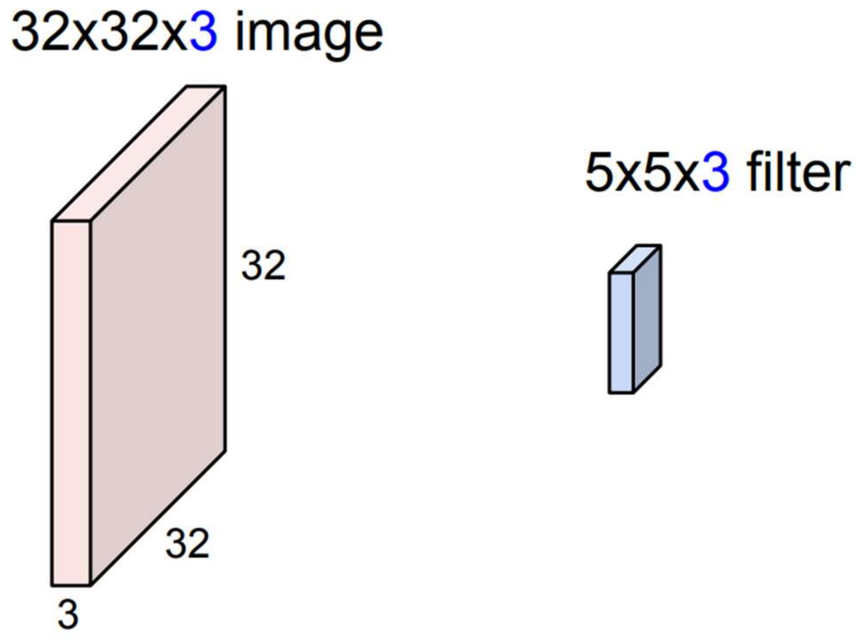
Convolutional layer

- We are convolving /sliding the filter spatially across the input image and computing the dot product.



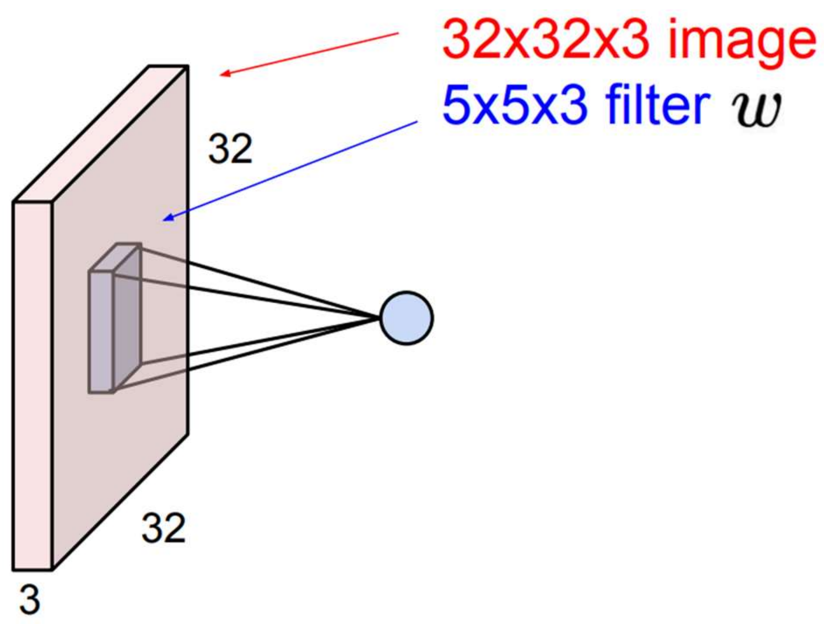
Convolutional layer

- The input volume and the filer has always the same depth (blue value).

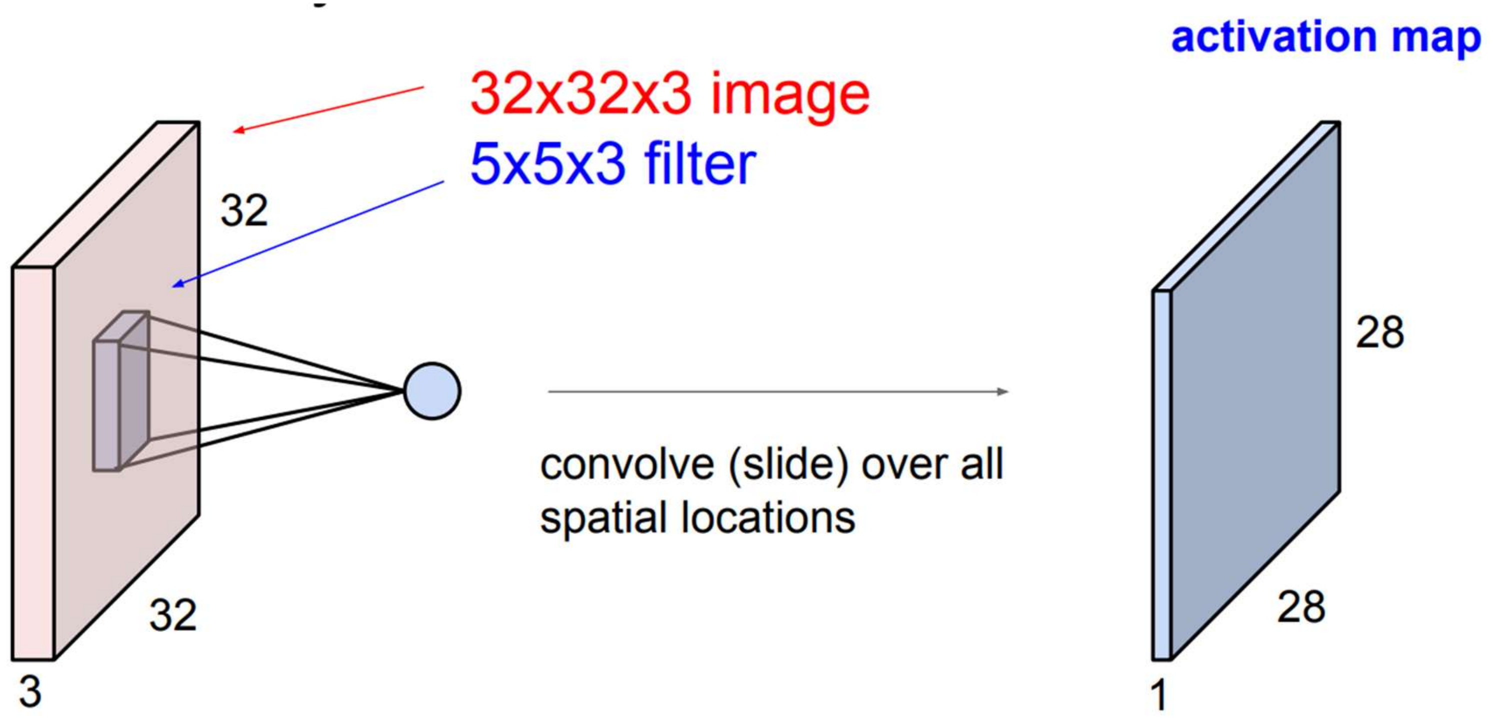


Convolutional layer

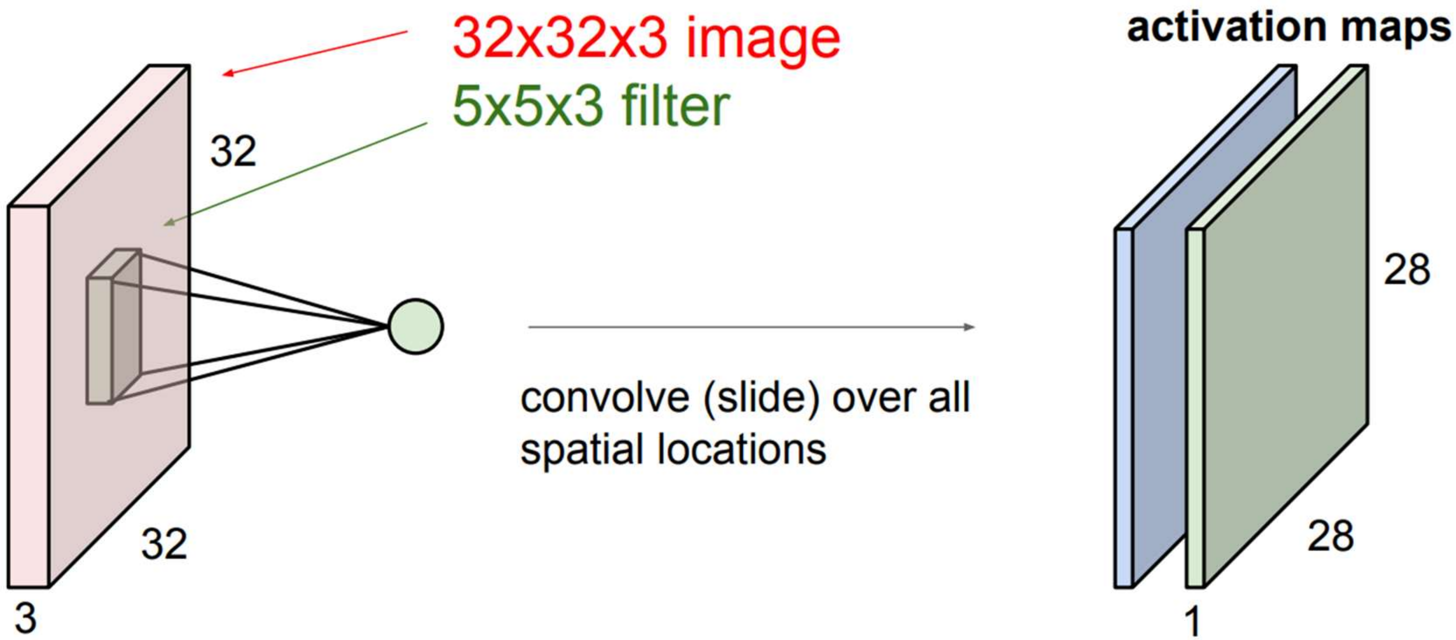
- The activation from a local region is computed:
- $z = w^T x + b$



Convolutional layer

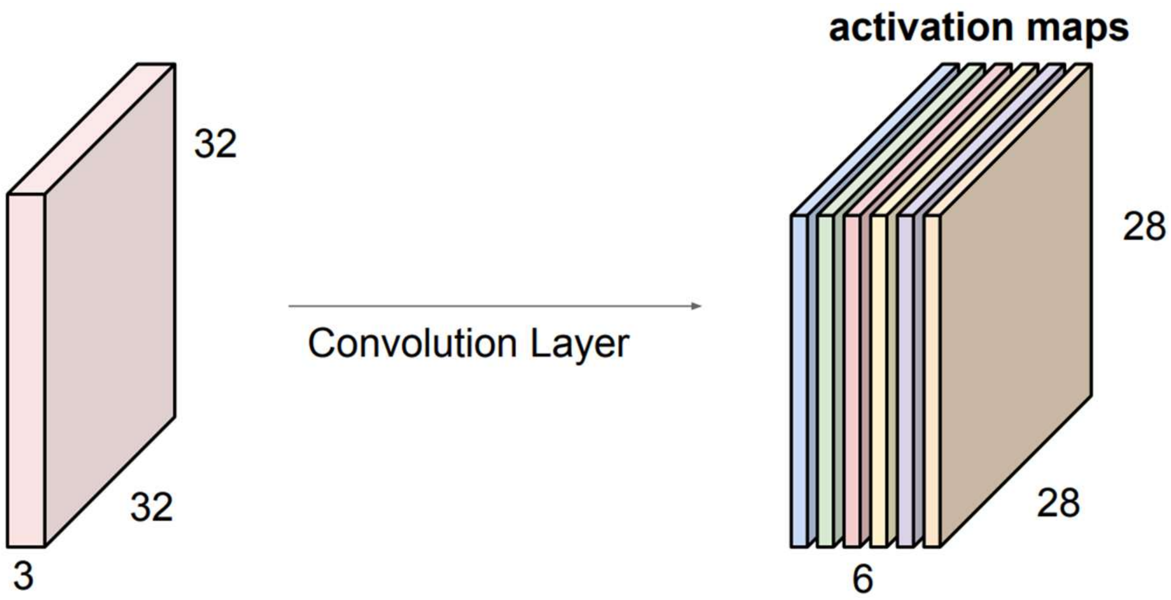


Convolutional layer



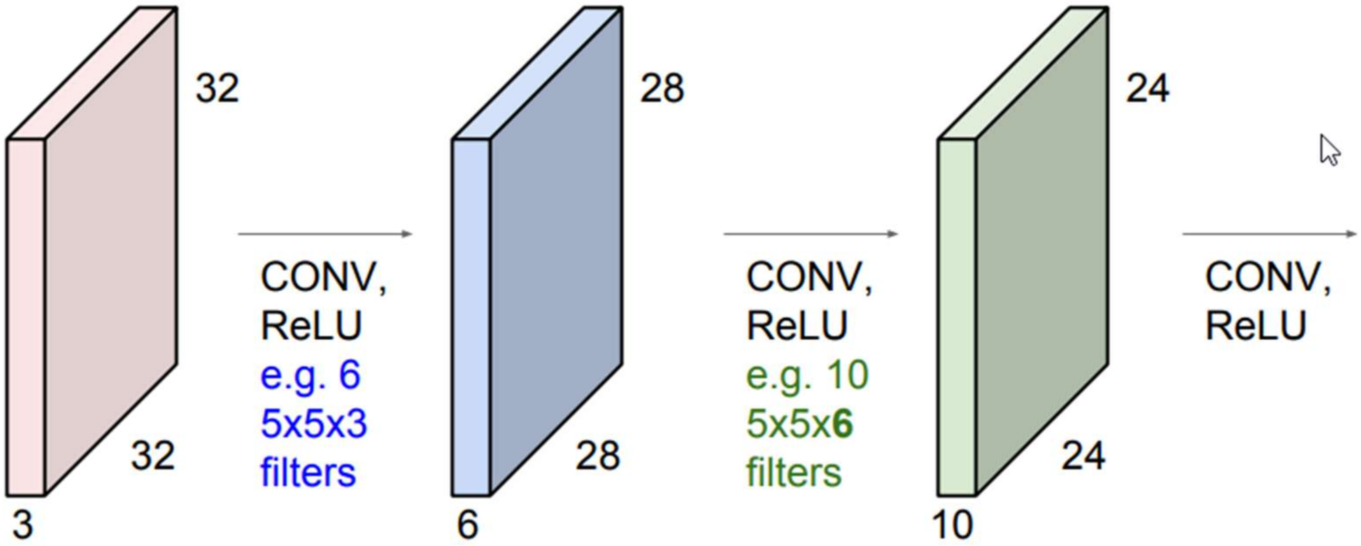
Convolutional layer

- If we filter the input volume 6 times using 5x5x3 filters, we get an output volume with 6 channels (depth)



Activations

- We use an activation function separately on all elements of the output volume



Progress

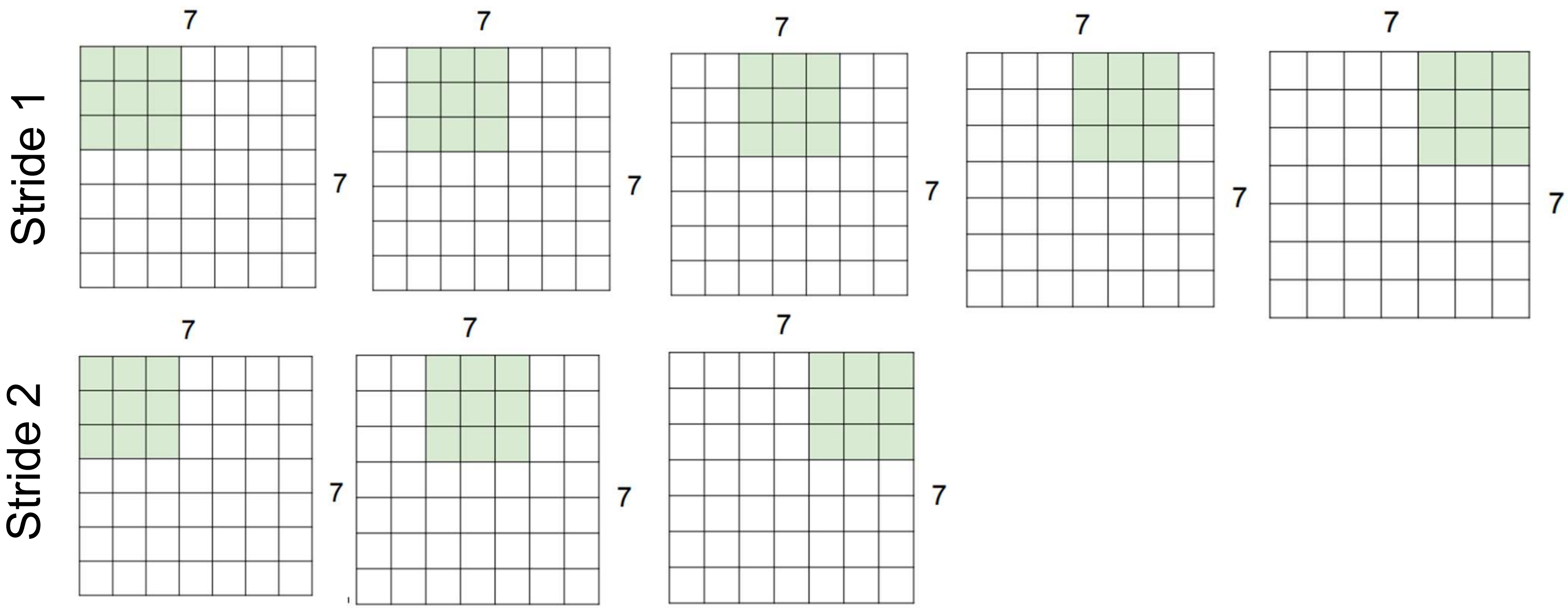
- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- **Convolutional layer hyperparameters**
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

Convolution neural network hyper-parameters

- Stride
- Padding
- Kernel (filter/weights) size

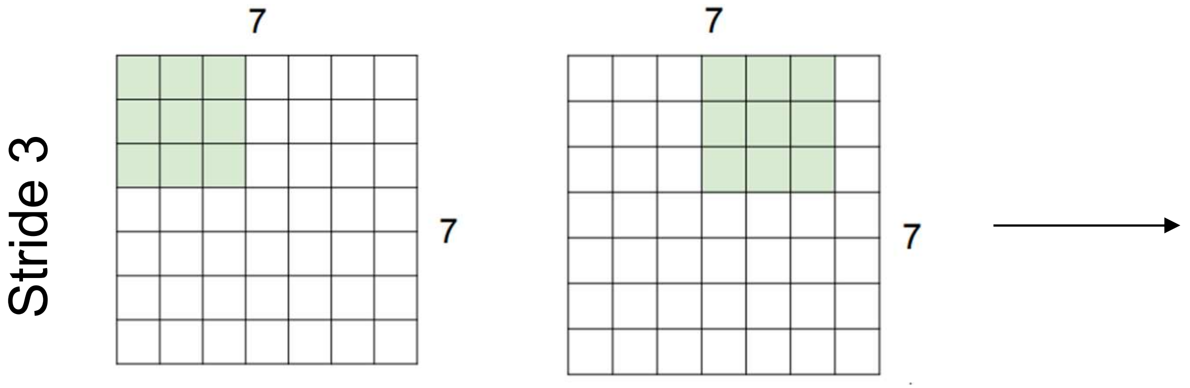
Stride

- Stride is the **spatial step length** in the convolution operation.
- Example: Input volume 7x7x1, kernel (filter) size 3x3x1
- The stride is an important parameter for determining the spatial size of the output volume



Stride

- What about stride equal to 3?



Using a 3x3x1 filter on the 7x7x1 volume with stride 3 does not fit!

Padding

- The output volume can get a lower spatial dimension compared to the input volume. We can solve this by padding the input volume. Common to use zero padding.

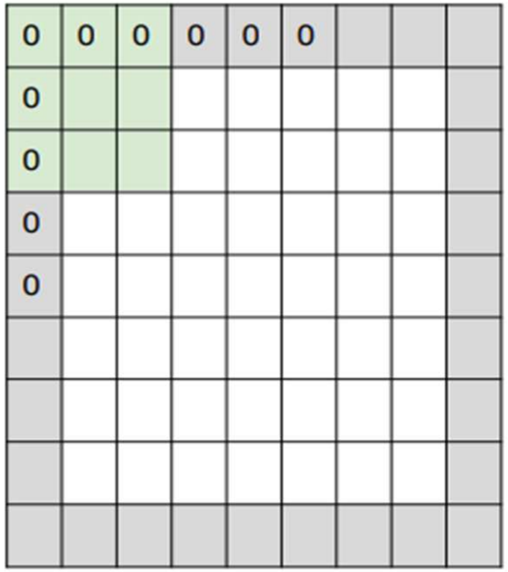
- Abbreviations: Stride (S), spatial filter size (F), input spatial size (N^i), output spatial size (N^{i+1}) and padding (P)

- For $S = 1$, we can achieve $N^0 = N^1$ by selecting P equal to:

$$P = \frac{(F-1)}{2}$$

- Calculation of the spatial output size:

$$N^{i+1} = \frac{N^i - F + 2P}{S} + 1$$



Padding examples

- Remember: $N^{i+1} = \frac{N^i - F + 2P}{S} + 1$

- Parameters:
  - $N^0 = 7$
  - $P = 0$
  - $F = 3$

- Stride 1 $\rightarrow$ $\frac{7-3+2\cdot 0}{1} + 1 = 5$
- Stride 2 $\rightarrow$ $\frac{7-3+2\cdot 0}{2} + 1 = 3$
- Stride 3 $\rightarrow$ $\frac{7-3+2\cdot 0}{3} + 1 = 2.33$

Padding examples

- Remember, to keep $N^i = N^{i+1}$ with $S = 1$ use:

$$P = \frac{(F-1)}{2}$$

- $F = 3 \quad \rightarrow$ zero pad with 1
- $F = 5 \quad \rightarrow$ zero pad with 2
- $F = 7 \quad \rightarrow$ zero pad with 3

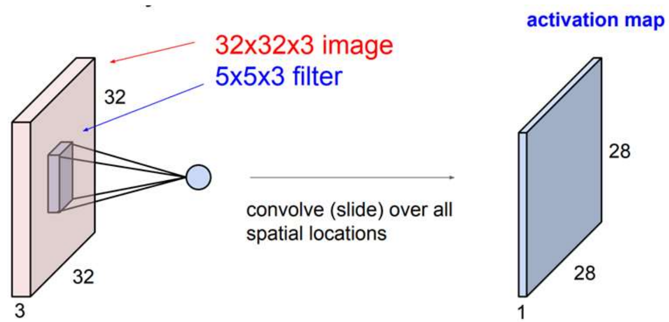
Kernel size (filter bank)

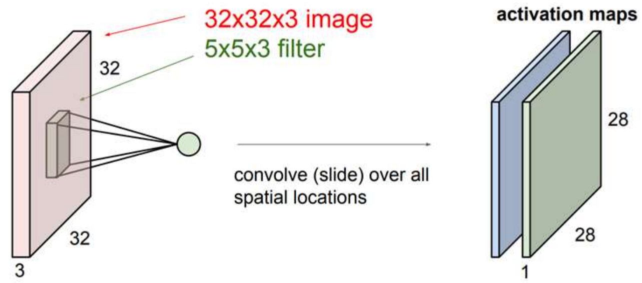
- Each filter has a size of $[F_c, F_h, F_w]$ e.g. $[3, 5, 5]$

- Multiple filters (F_N) can be applied at each layer and the filter bank are represented by a 4-D tensor
    - $[F_N, F_c, F_h, F_w]$

- F_N corresponds to the depth of the next layer

- This is a practical representation and used by many deep learning frameworks.

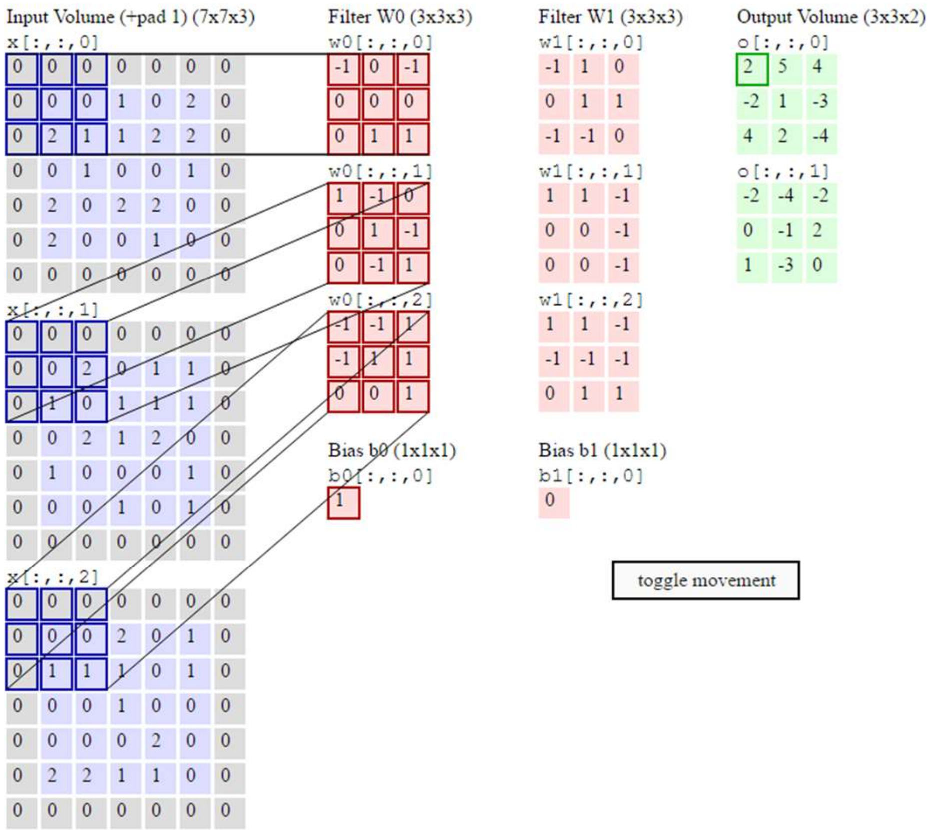


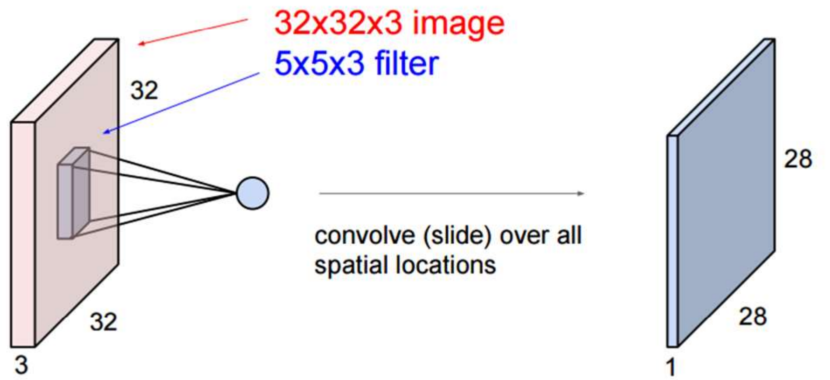


Progress

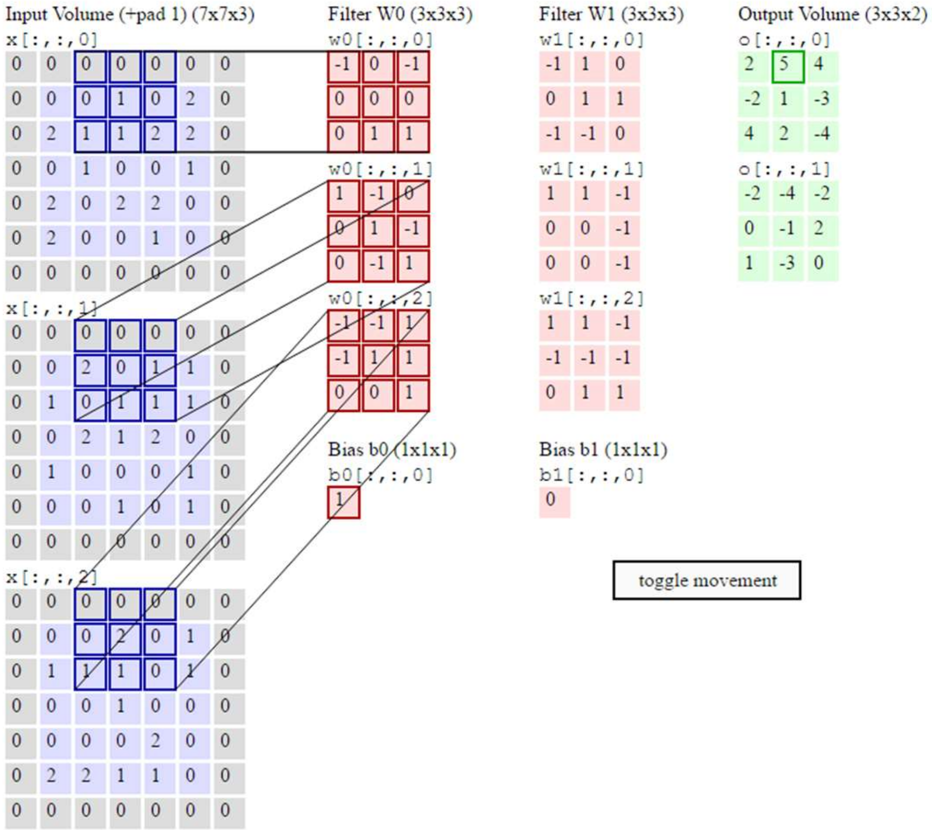
- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- **Convolutional layer example**
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

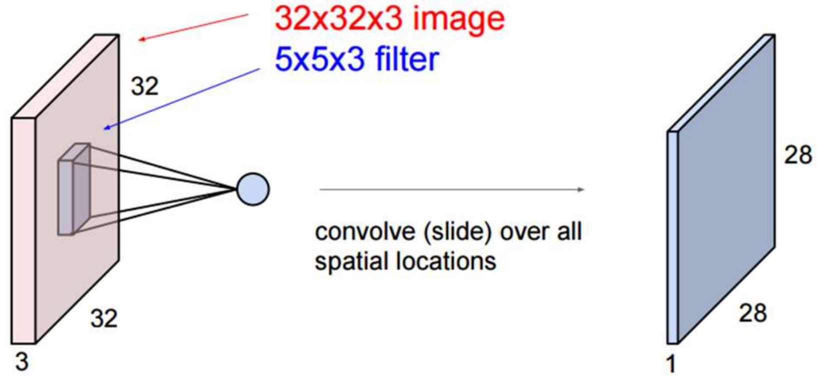
A one-layer, two-filter network



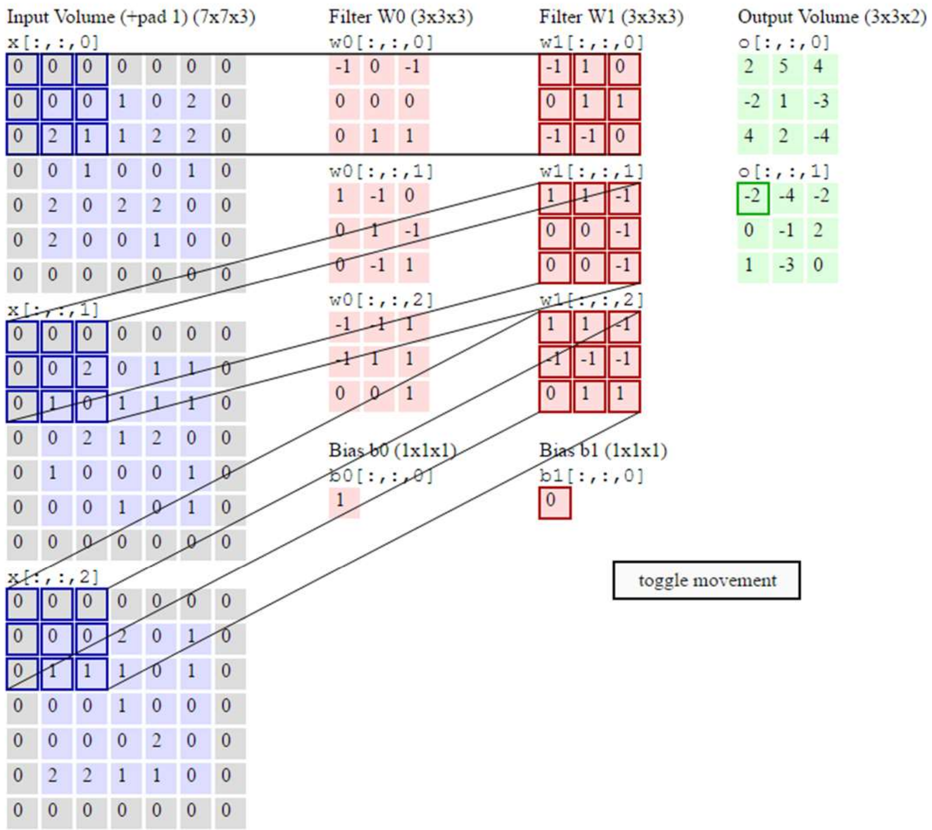


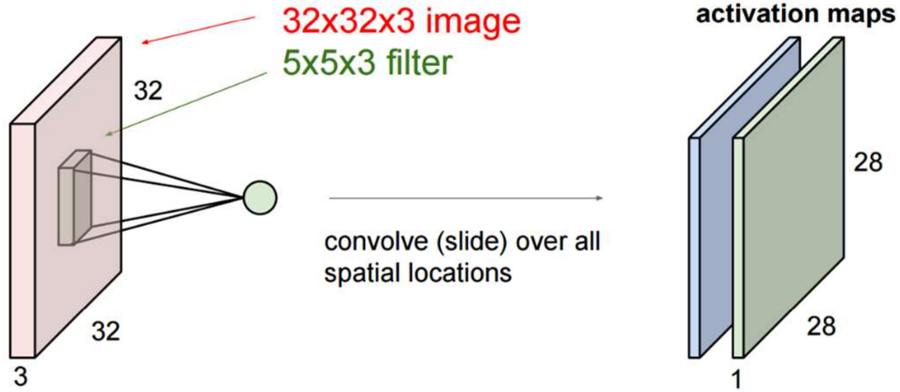
A one-layer, two-filter network



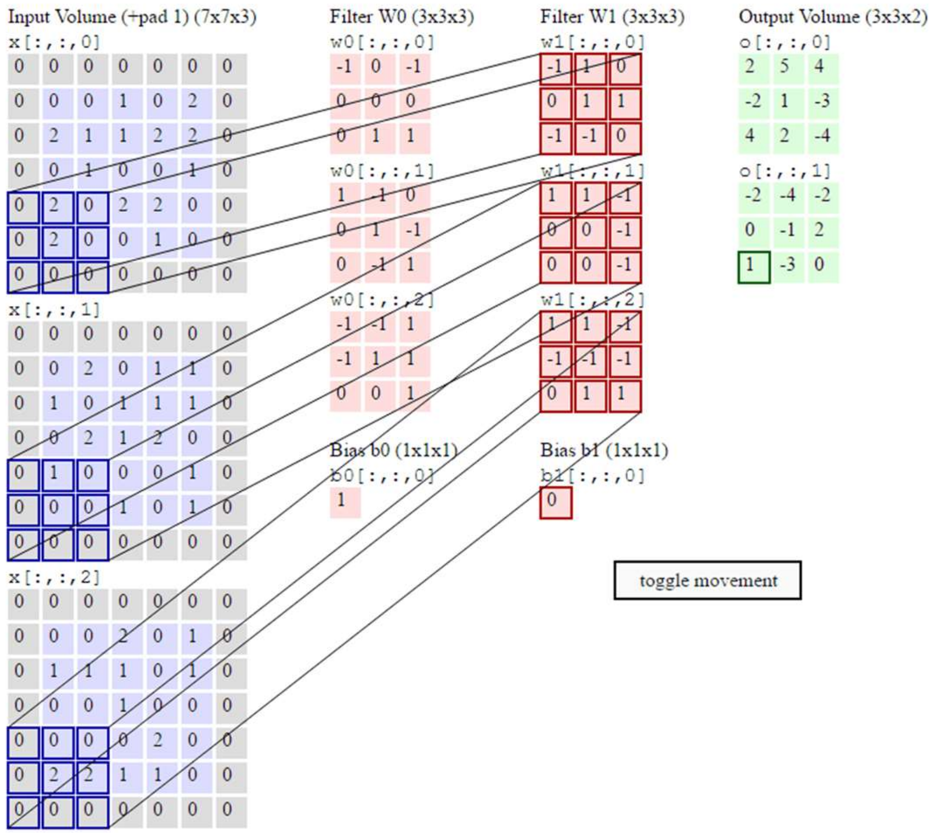


A one-layer, two-filter network





A one-layer, two-filter network



Progress

- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- **Receptive field (Field of View)**
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

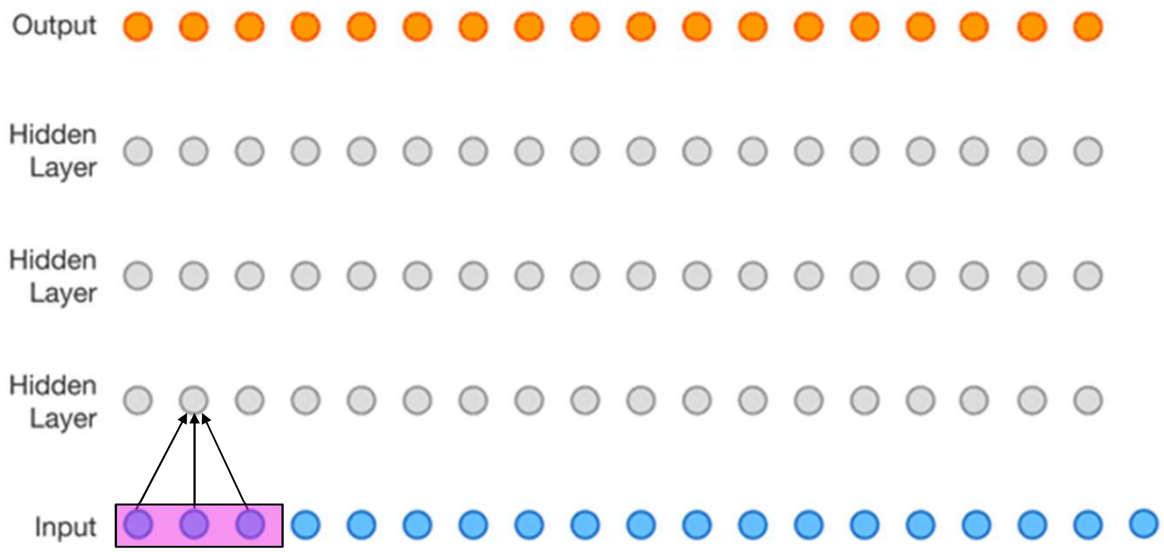
Receptive field (Field of View)

- How much of the input image is available for a particular neuron?

How large area influence the end result?

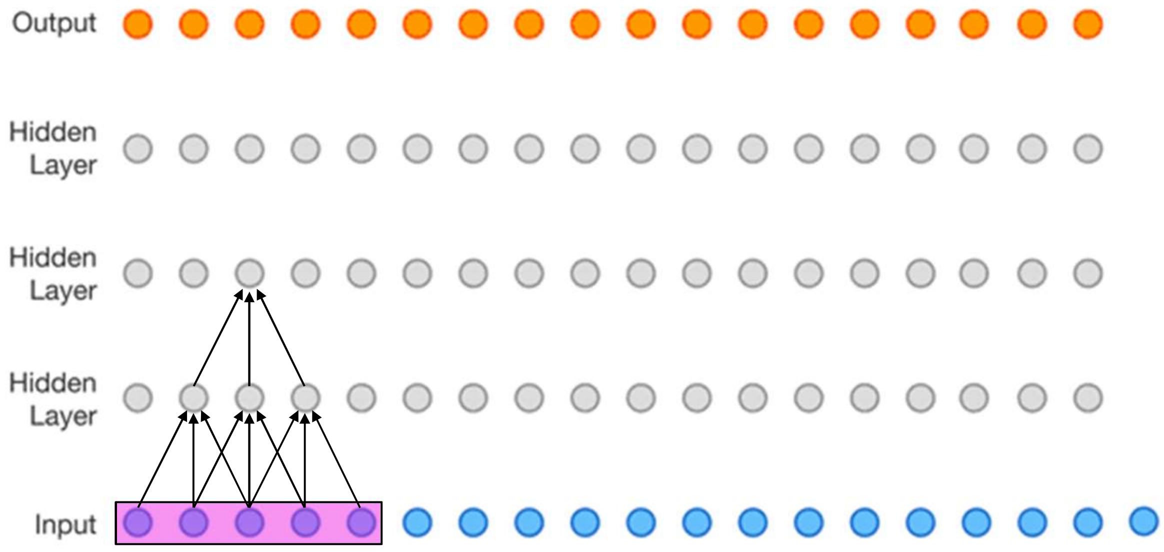
- With a convolutional network the receptive field increase with each layer

- 3 inputs influence each node in the first hidden layer



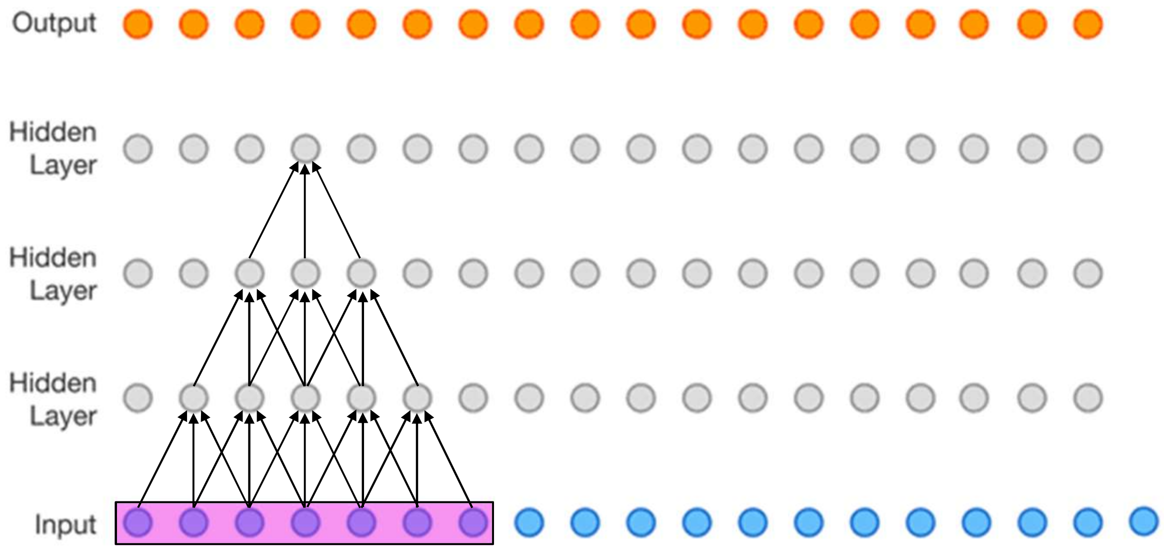
How large area influence the end result?

- With a convolutional network the receptive field increase with each layer
- 3 inputs influence each node in the first hidden layer
- 5 influence the next



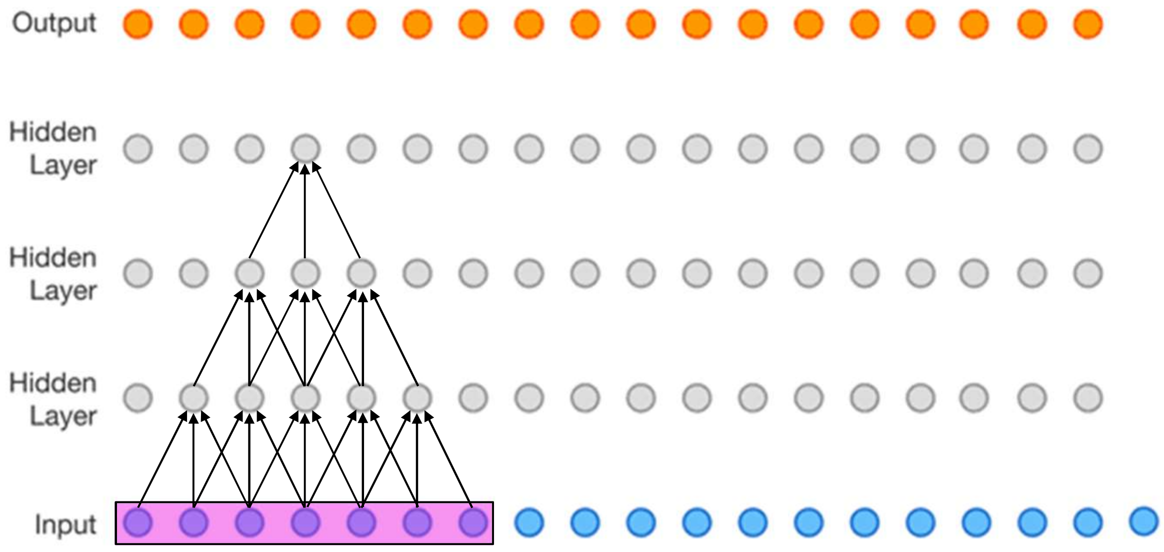
How large area influence the end result?

- With a convolutional network the receptive field increase with each layer
- 3 inputs influence each node in the first hidden layer
- 5 influence the next
- 7 influence the next



The receptive field grow with k-1 for each layer

- Two 3x3 filters give equal receptive field as one 5x5 filer
- Should we use 3x3 or 5x5?

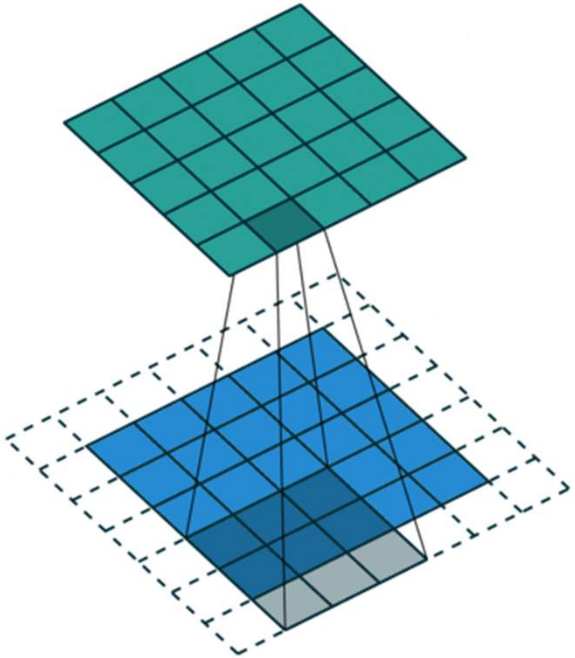


Parameter efficiency

- Two 3x3 filters give equal receptive field as one 5x5 filter
- Should we use 3x3 or 5x5 filters?
- Assumption:
  - The filter count in all layers are ($F_c = F_c^i = F_c^{i+1}$) and we don't account for biases.
- Number of parameters:
  - 3x3 filter → $(3 \cdot 3 \cdot F_c) \cdot F_c + (3 \cdot 3 \cdot F_c) \cdot F_c = 18{F_c}^2$
  - 5x5 filter → $(5 \cdot 5 \cdot F_c) \cdot F_c = 25{F_c}^2$
- Note: Many 3x3 filters will lead to a larger memory footprint during training as the system must store the values for backpropagation.

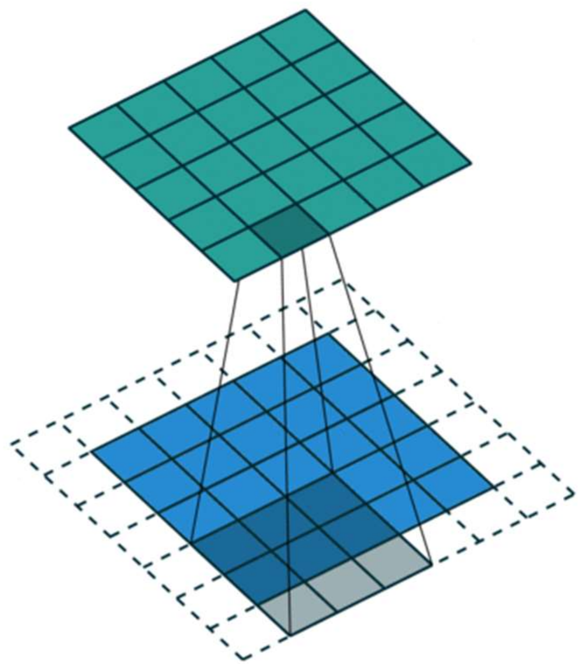
Smaller spatial filter size is more parameter efficient

- A network with many parameters generally need more training data and computation time
- A larger receptive field per parameter is good
- More layers can give more reuse



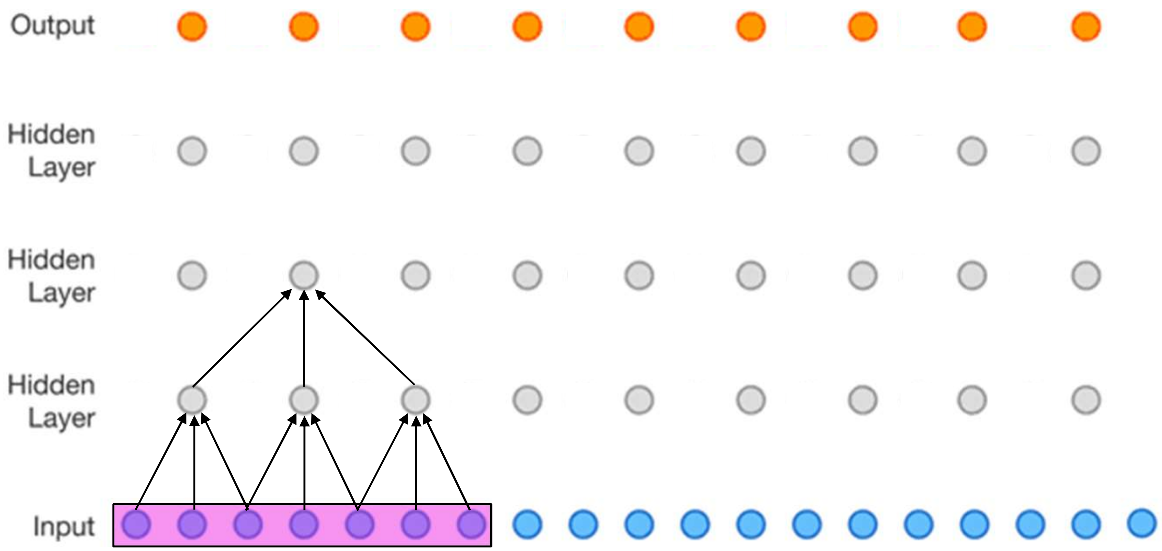
Strided convolutions

- By skipping positions we can cover a larger area with less computation
- The effect of the receptive field for the next layer is important



The effect of strided convolutions

- We still cover the whole input
- With stride of two we have increased the receptive field from 5→7 in layer 2

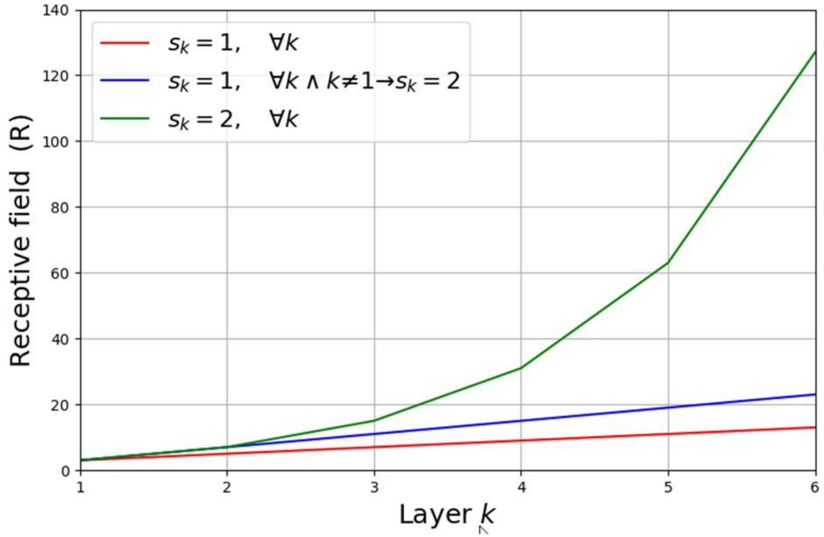


The effect of strided convolutions

- Receptive field : R
- Spatial filter size: F
- Stride: S
- Layer index: $\mathrm{k} \in \{1,2,3,\dots,n\}$

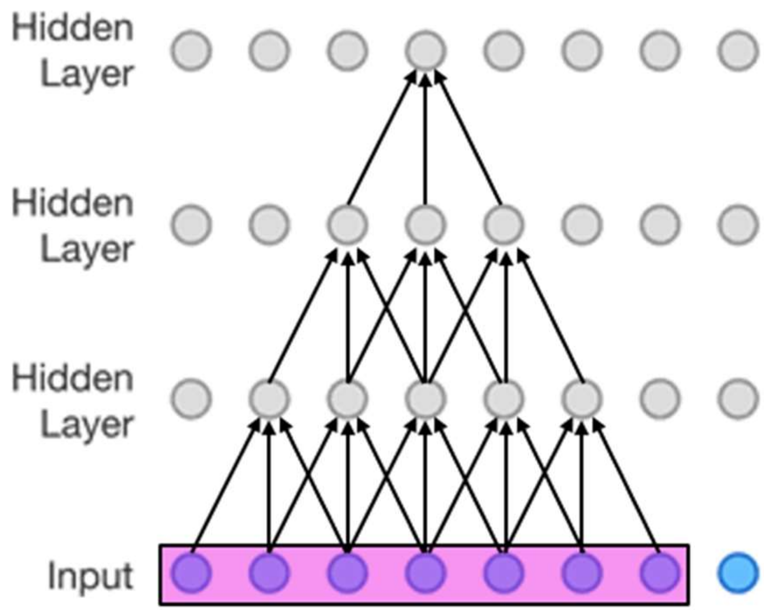
$$R^k = R^{k-1} + \left[\left(F^k - 1 \right) \cdot \prod_{i=1}^{k-1} S^i \right]$$

- Essentially all the following layers will have a receptive field multiplied by S^k



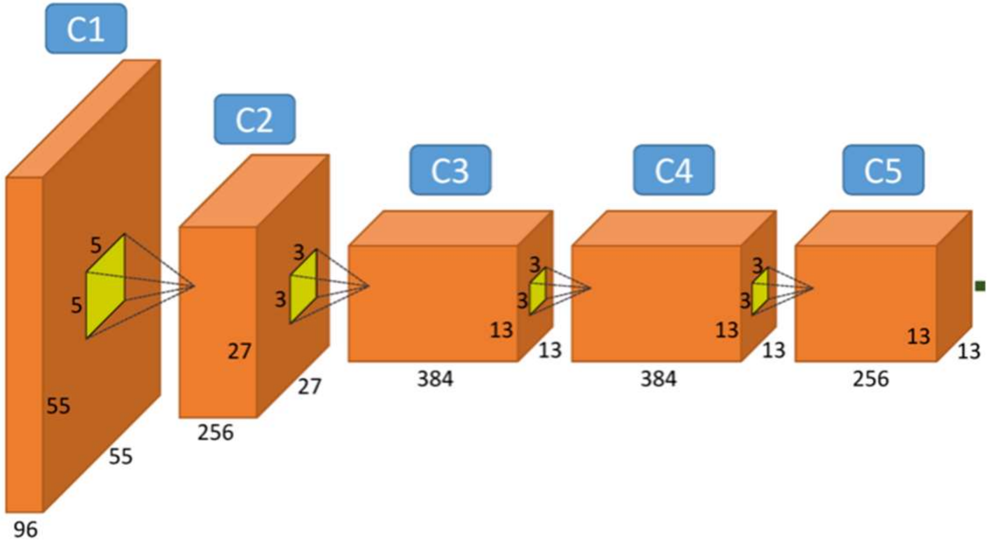
Theoretical vs effective receptive field

- “Effective receptive field only takes up a fraction of the full theoretical receptive field”



With strides, spatial dimensions will become smaller

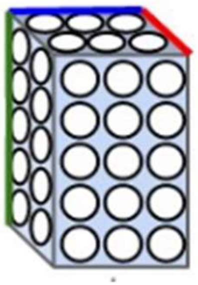
- Usually some of the of the network capacity is preserved through an increasing number of channels



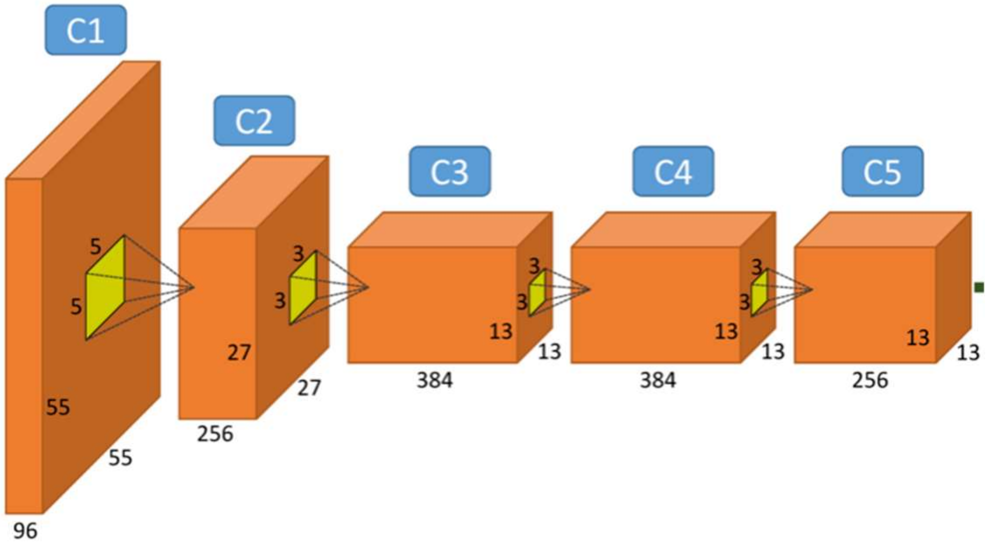
Can the network still remember positions?

- Yes, the network can still encode positional information in the **depth** dimension

- A network can pass positional information (right, left etc.) to different channels



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

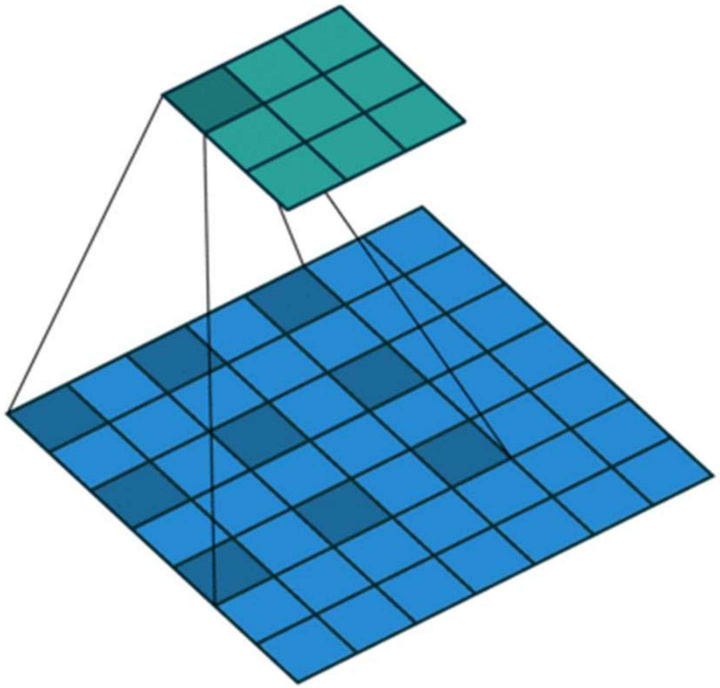


Progress

- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- **Dilated convolutions**
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

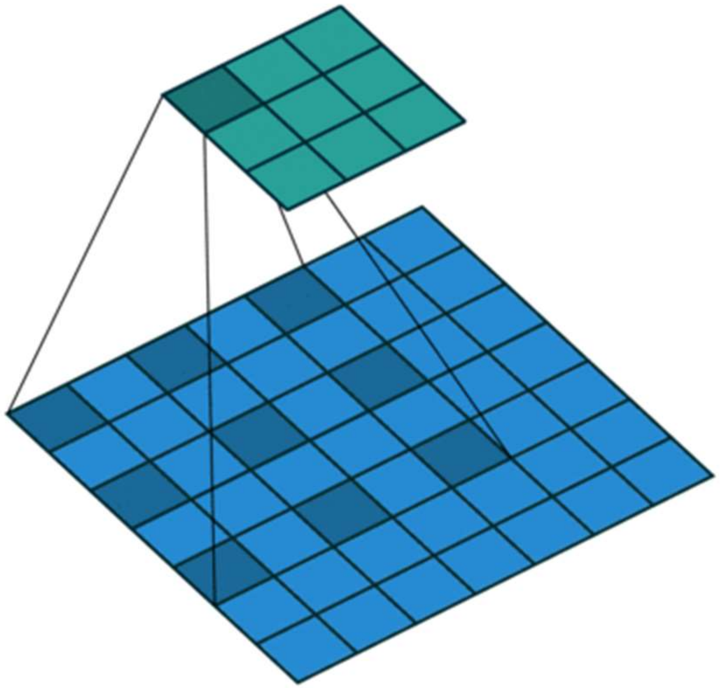
Dilated convolutions

- Larger receptive field, without reducing spatial dimension or increasing the parameters



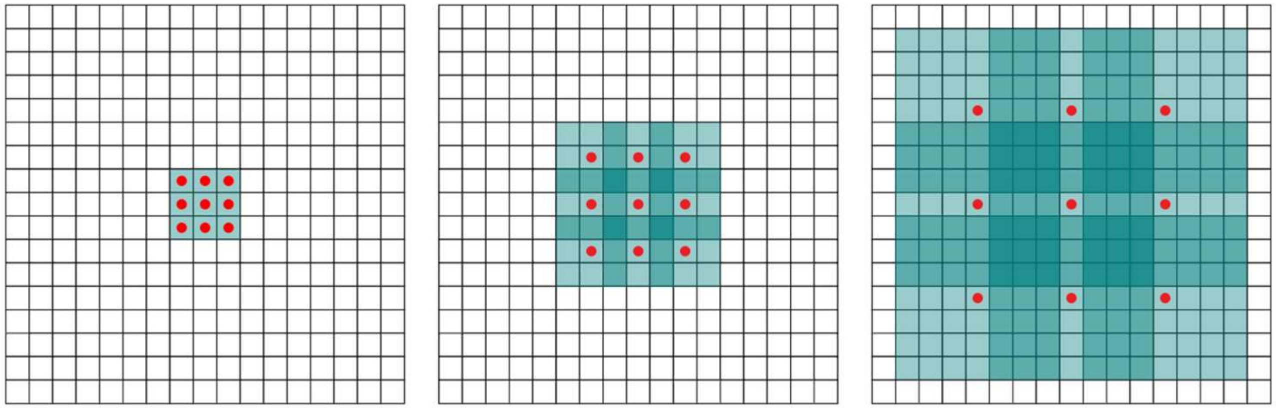
Dilated convolutions

- Skipping values in the kernel
- Same as filling the kernel with every other value as zero
- Still cover all inputs
- Larger kernel with no extra parameters



A growing dilation factor can give similar effect as stride

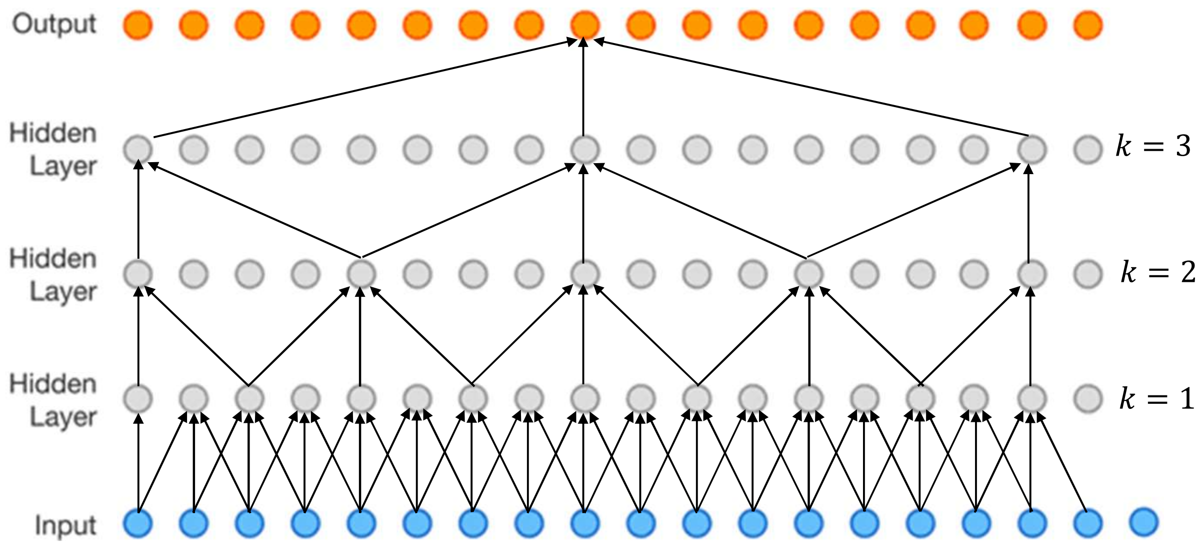
- With a constant dilation factor you get the similar effect as using a larger kernel
- With growing dilation factor you can get an even larger receptive field, while still covering all inputs



Fisher Yu, Vladlen Koltun (2016) Multi-scale Context Aggregation by Dilated Convolutions

Growing dilation factor

- 1-D example:
  - Filter size: $\mathrm{F} = 3$
  - Layer: $\mathrm{k} \in \{1,2,3, \dots, n\}$
  - Receptive field : $R^k = 2^{k+1} - 1$
  - Dilation factor: $l = 2^{k-1}$



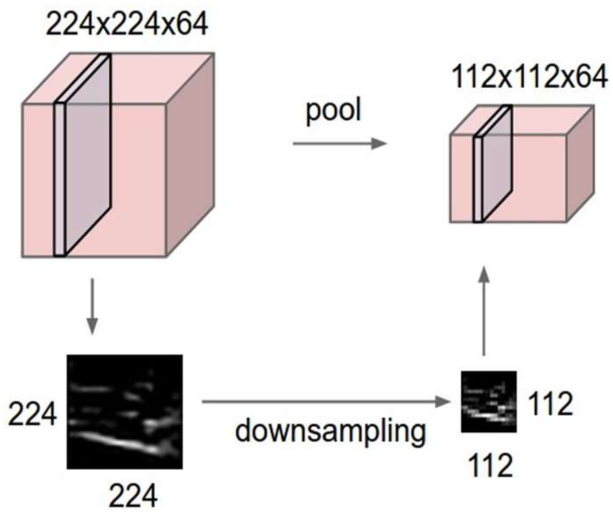
Progress

- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- **Pooling**
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

Pooling

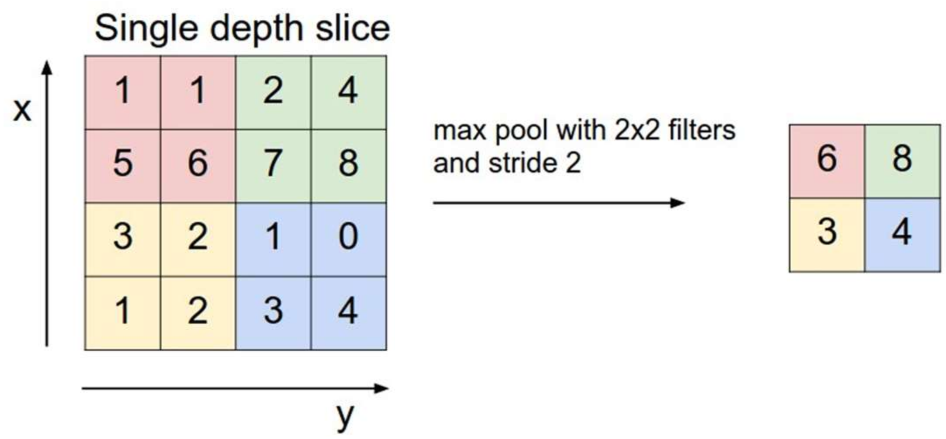
- Spatial reduction and forcing invariance
- Operates over each activation map (channel) independently
- No learnable weights

- Two methods:
  - Max pooling
  - Average pooling



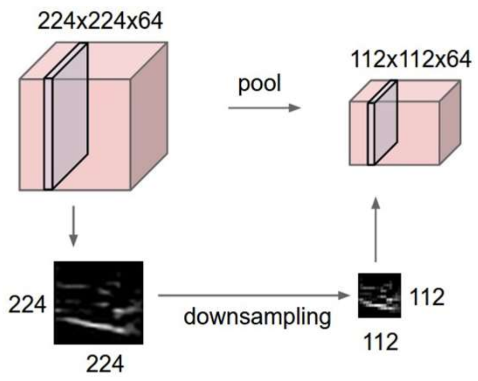
Max pooling

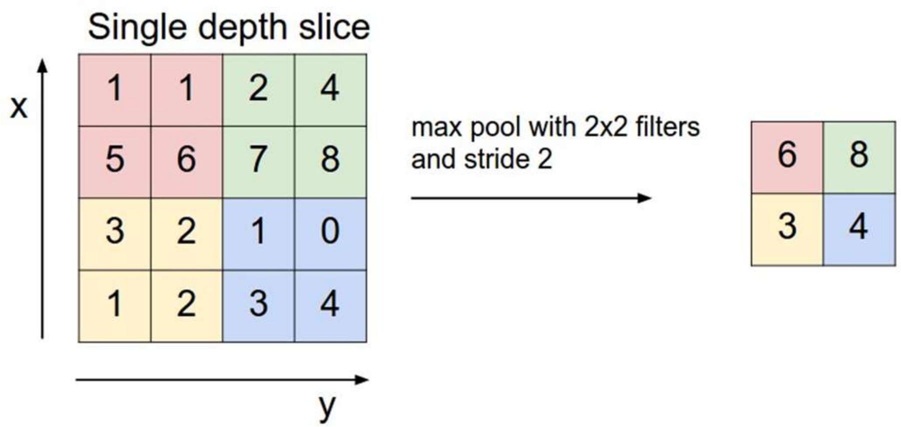
- A strided maximum filtering
- Choosing the maximum value inside the kernel



Max-pooling: invariance built-in

- With max-pooling you explicitly remove some spatial information
- This can help both position and rotation invariance



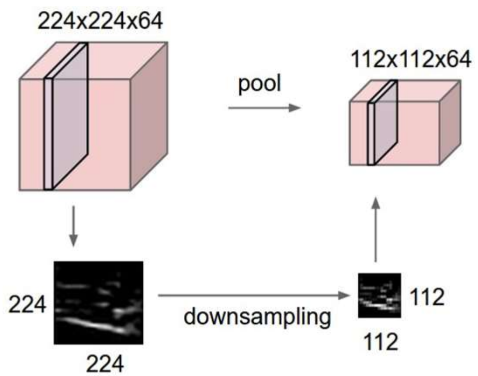


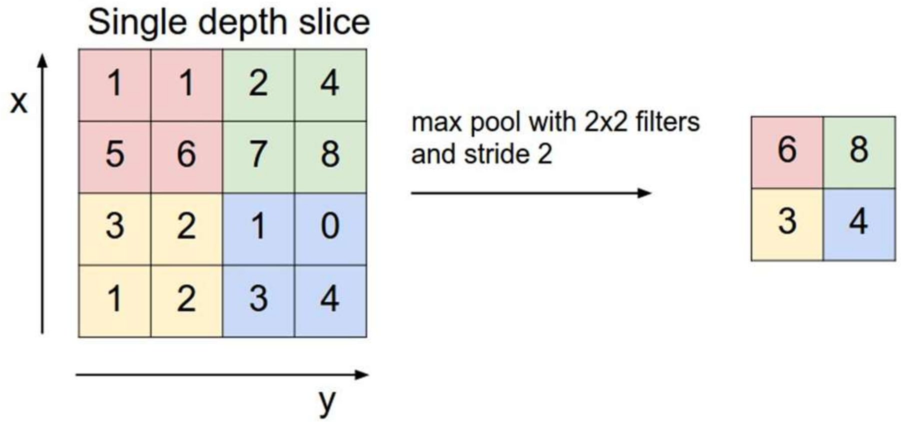
Max-pooling have some important problems

- Even if we want our final results to be positional invariant, we may need positional information in the earlier representations

- Only a small part of the network is updated with gradients each step (learning slower)

- We calculate a lot of values that is not “used”





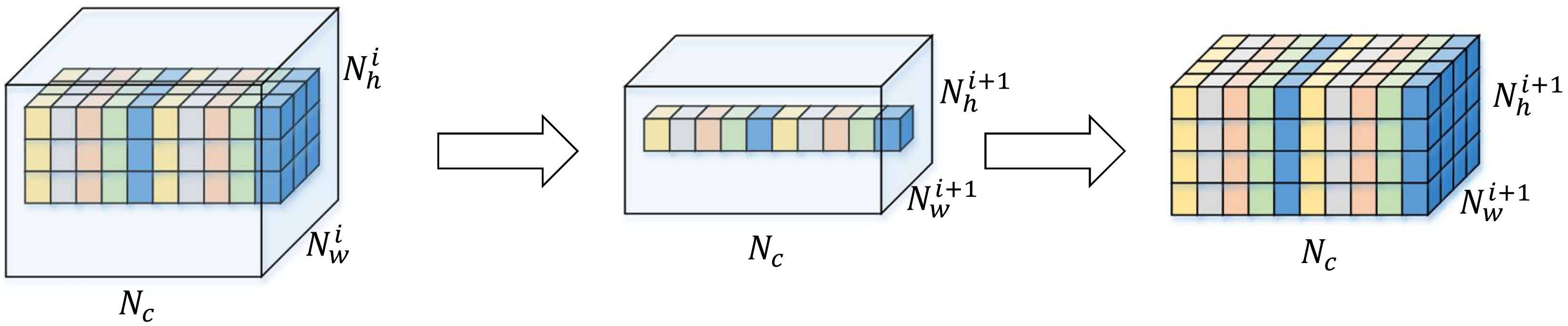
Progress

- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- **Depthwise Separable Convolution**
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

Depthwise Separable Convolution

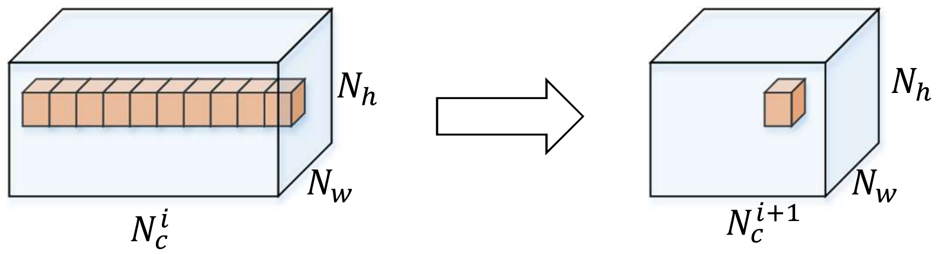
- Depthwise separable convolution is an efficient convolutional layer. It is composed of two steps:
  - Depthwise convolution
  - Pointwise convolution

- **Depthwise convolution** :
  - Input volume of shape $[N_c, N_h^i, N_w^i,]$
  - We use N_c different kernels of shape $[\mathrm{F_c} = 1, F_h, F_w]$ on the input channels individually
  - Output volume $[N_c, N_h^{i+1}, N_w^{i+1},]$



Pointwise convolution

- Pointwise convolutions are ordinary convolutions with :
  - kernels of shape: $[F_c,\ F_h = 1,\ F_w = 1]$
  - Filter bank: $[F_N, F_c, 1, 1]$



Depthwise Separable Convolution – Summary

- Depthwise separable convolution = Depthwise convolution + Pointwise convolutions

- Lets compare the number of parameters in a depthwise separable convolution and a convolutional layer:

  $[F_N = 512, F_c = 256, F_h = 3, F_w = 3]$

- Parameters in a **depthwise separable convolution**:
  - $F_c \cdot 1 \cdot F_h \cdot F_w + F_N \cdot F_c \cdot 1 \cdot 1$ = 133,376

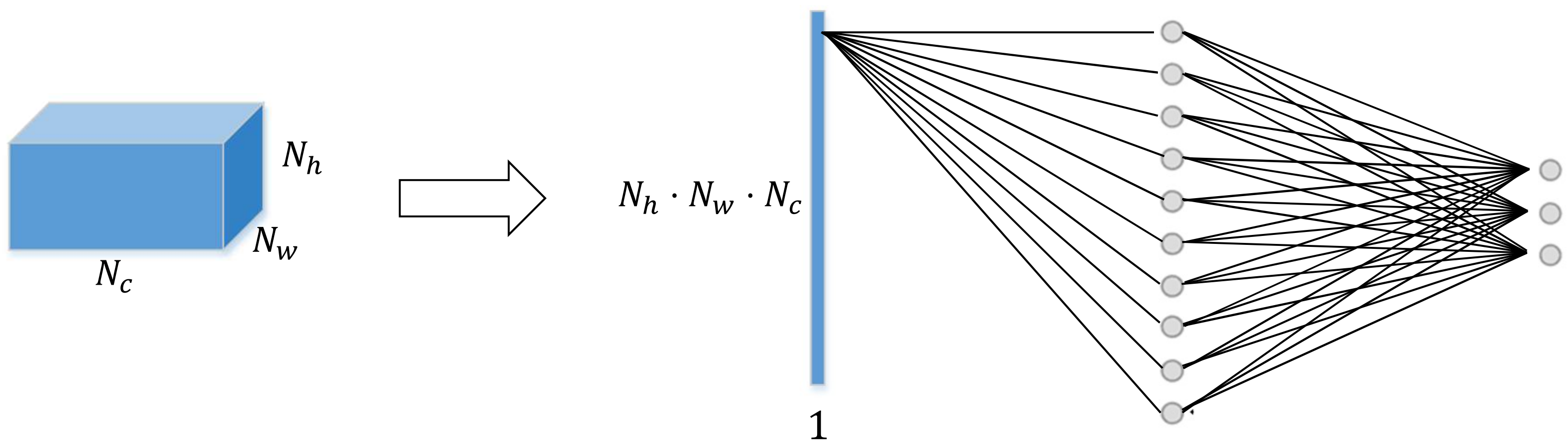
- Parameters in a **convolutional layer**:
  - $F_N \cdot F_c \cdot F_h \cdot F_w$ = 1,179,648

Progress

- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- **Last layer**
- Visualizing and Understanding CNN
- Applications were CNN are used
- Alternative to ConvNet

Structure of the last layer(s) – dense layer

- At the end we normally have a feature map of some spatial size and channels (N_c, N_w, N_h).
- Assume we have a 3 class classification problem and want our output to be a vector of length 3.
- We can flatten the input feature map and stack dense layers

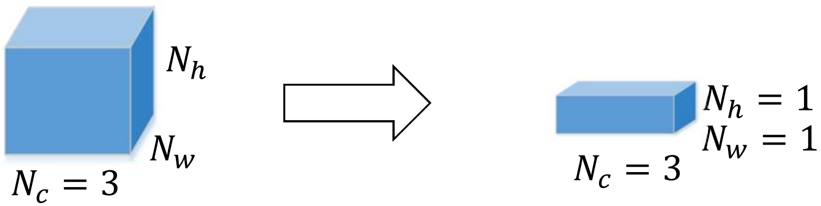


Structure of the last layer(s) – fully convolutional

- We can make sure the last layer has the same number of channels as we have classes.

- A 3 class problem yields $N_c = 3$

- Average over the spatial dimensions N_w and N_h

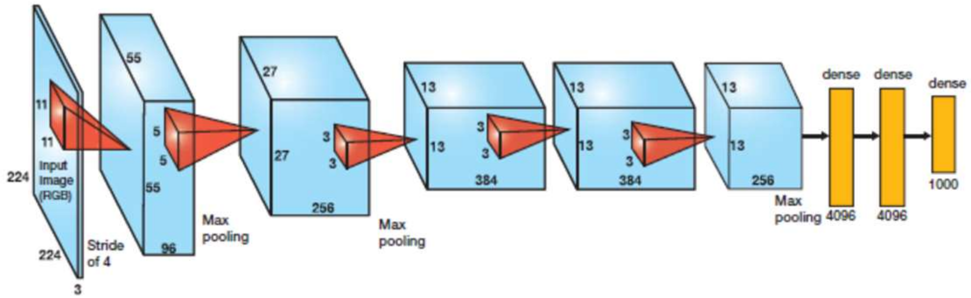


Progress

- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- **Visualizing and Understanding CNN**
- Applications were CNN are used
- Alternative to ConvNet

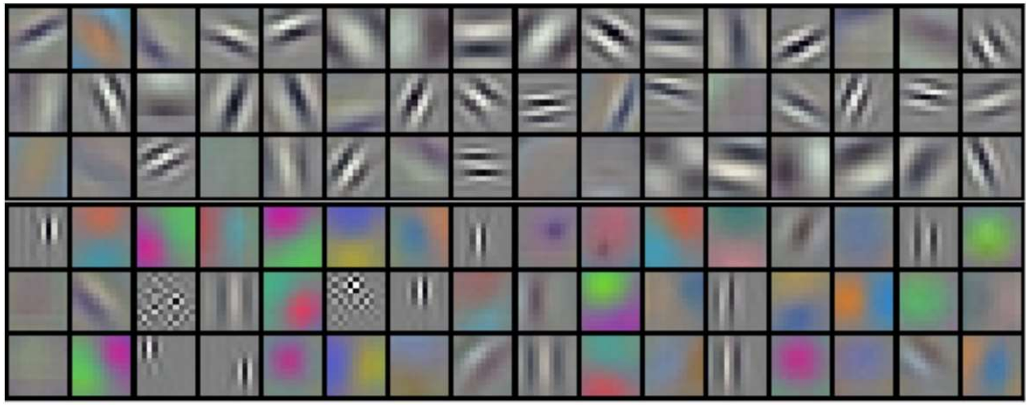
Visualizing and Understanding ConvNets

- AlexNet, the winner of the ImageNet classification challenge 2012.



- Filter bank of size (11x11x3)x96 for the first convolutional layer:

- Visualizing the learnt weights

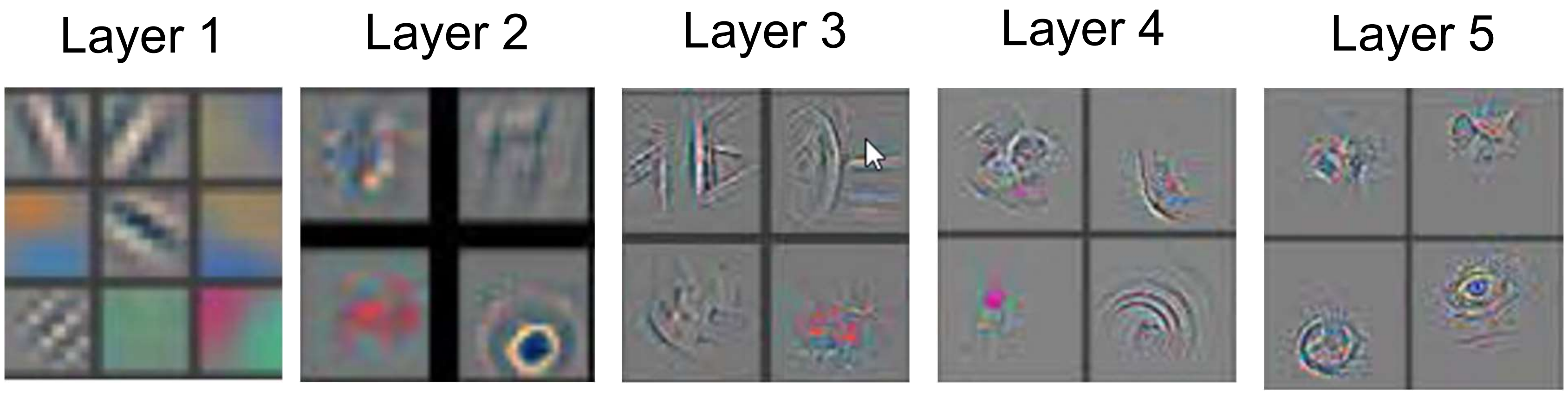


Figures copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012.

Visualizing and Understanding deeper layers

- Looking at the filer coefficient directly at deeper layer is not meaningful.

- Visualization with Deconvnet

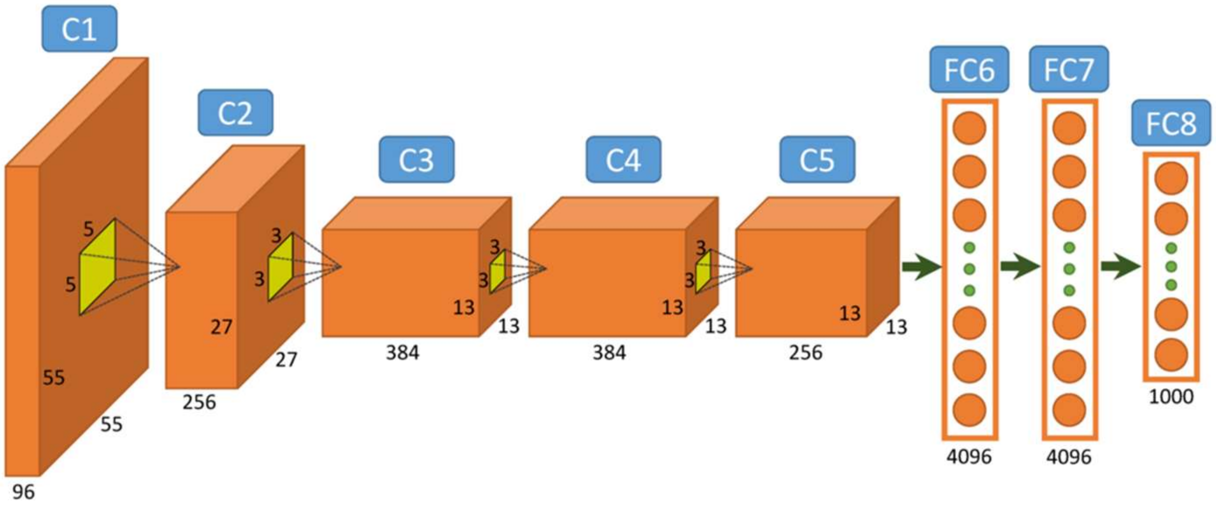


Zeiler M.D., Fergus R. (2014) Visualizing and Understanding Convolutional Networks

Hierarchical learning

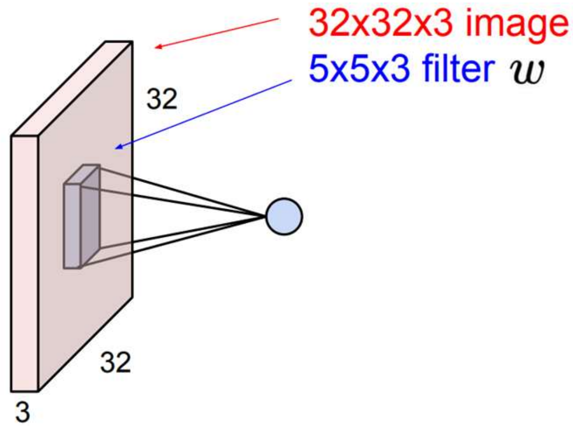
- A convolution neural network is built up as a hierarchy were the complexity (abstraction) is increased by depth.

- A hierarchical structure is parameter efficient



Reuse of features

- Each filter kernel is applied at all spatial positions
- Features are reused:
  – edges, fur, eye, grass
- Reuse instead of retraining many times over



Data driven

- A convolutional neural network still “remembers” shapes, rotation, size.
- No fundamental understanding of the concept “cat”

Progress

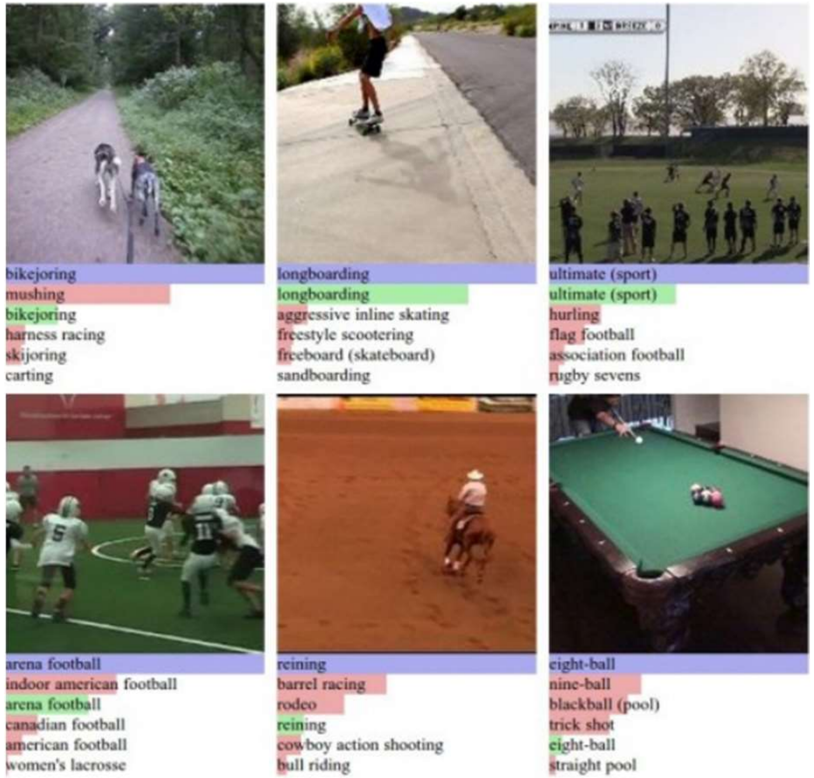
- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- **Applications were CNN are used**
- Alternative to ConvNet

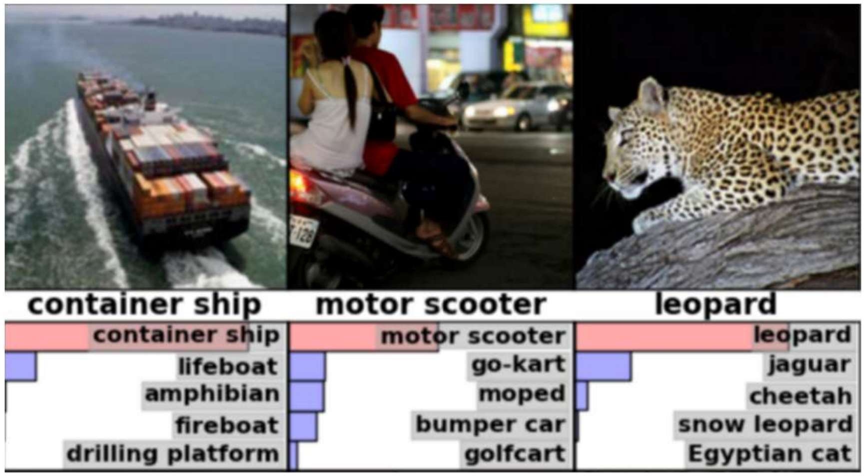
Application of convolutional neural network

- Classification
- Detection
- Segmentation
- Reinforcement learning (game playing)
- Image captioning

Classification

- Images for ImageNet





Detection

Segmentation

Reinforcement learning (game playing)

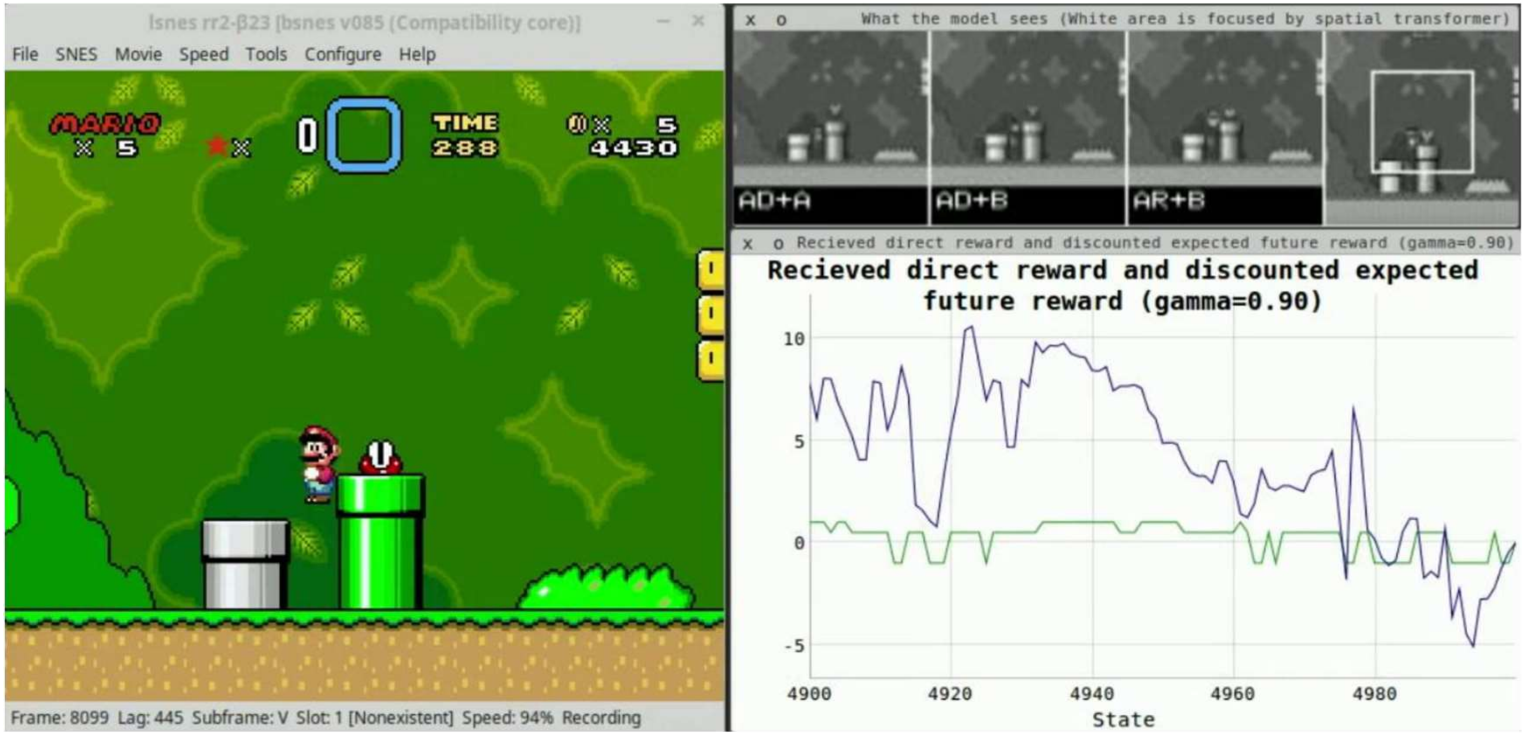


Image captioning

Progress

- Challenges with image classification
- Benchmark: ImageNet
- Fully connected neural network on images
- Convolutional layer
- Convolutional layer hyperparameters
- Convolutional layer example
- Receptive field (Field of View)
- Dilated convolutions
- Pooling
- Depthwise Separable Convolution
- Last layer
- Visualizing and Understanding CNN
- Applications were CNN are used
- **Alternative to ConvNet**

Alternative to ConvNet

Note: Not part of curriculum

- Rotation equivariant vector field networks
  - https://arxiv.org/abs/1612.09346

- Capsule Network
  - https://arxiv.org/abs/1710.09829

CNN vs dense net on cifar10