TU-Delft Deep Learning course 2018-2019

04 CNN part 2

27 Feb 2019

Lecturer: Jan van Gemert

Convolutional versus feed forward

Q: $[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ 0,\ +1] =$

Convolutional versus feed forward

Q: $[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ 0,\ +1] = [-1,\ 99,\ 100,\ 1]$

Convolutional versus feed forward

Q: $[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ 0,\ +1] = [-1,\ 99,\ 100,\ 1]$

- Q: Can you write this as a matrix multiplication? (how many rows and cols?)

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \begin{bmatrix} \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdots \\ \vdots & \ddots & \cdots \end{bmatrix} = [-1,\ 99,\ 100,\ 1]$$

Convolutional versus feed forward

Q: $[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ 0,\ +1] = [-1,\ 99,\ 100,\ 1]$

- Q: Can you write this as a matrix multiplication? (how many rows and cols?)

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \begin{bmatrix} \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdots \\ \vdots & \ddots & \cdots \end{bmatrix} = [-1,\ 99,\ 100,\ 1]$$

- A: Toeplitz matrix (or diagonal-constant matrix):

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [-1,\ 99,\ 100,\ 1]$$

Convolutional versus feed forward

Q: $[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ 0,\ +1] = [-1,\ 99,\ 100,\ 1]$

- Q: Can you write this as a matrix multiplication? (how many rows and cols?)

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \begin{bmatrix} \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdots \\ \vdots & \ddots & \cdots \end{bmatrix} = [-1,\ 99,\ 100,\ 1]$$

- A: Toeplitz matrix (or diagonal-constant matrix):

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [-1,\ 99,\ 100,\ 1]$$

Can be extended to 2D by doubly block-Toeplitz matrices

Convolutional versus feed forward

Chapter 9.2

Toeplitz matrix (or diagonal-constant matrix) is the same as convolution.

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201]\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [-1,\ 99,\ 100,\ 1]$$

- Q: What are three things to notice about this matrix?

Convolutional versus feed forward

Chapter 9.2

Toeplitz matrix (or diagonal-constant matrix) is the same as convolution.

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [-1,\ 99,\ 100,\ 1]$$

- Q: What are three things to notice about this matrix?
  - Sparse (many zeros)
  - Local (non-zero values occur next to each other)
  - Sharing parameters (same values repeat)
- Q: What is the difference between feed-forward and convolutional networks?

Convolutional versus feed forward

Chapter 9.2

Toeplitz matrix (or diagonal-constant matrix) is the same as convolution.

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [-1,\ 99,\ 100,\ 1]$$

- Q: What are three things to notice about this matrix?
  - Sparse (many zeros)
  - Local (non-zero values occur next to each other)
  - Sharing parameters (same values repeat)
- Q: What is the difference between feed-forward and convolutional networks?
- A: CNN is –by design– a limited parameter version of feed forward
- Q: Wait.. Less parameters is less flexibility. Why is this a *good* thing?

Convolutional versus feed forward

Chapter 9.2

Toeplitz matrix (or diagonal-constant matrix) is the same as convolution.

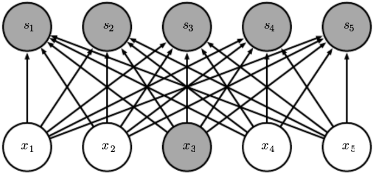
$$[100,\ 101,\ 99,\ 200,\ 199,\ 201]\begin{bmatrix}-1 & 0 & 0 & 0\\ 0 & -1 & 0 & 0\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} = [-1,\ 99,\ 100,\ 1]$$

- Q: What are three things to notice about this matrix?
  - Sparse (many zeros)
  - Local (non-zero values occur next to each other)
  - Sharing parameters (same values repeat)
- Q: What is the difference between feed-forward and convolutional networks?
- A: CNN is –by design– a limited parameter version of feed forward
- Q: Wait.. Less parameters is less flexibility. Why is this a *good* thing?
- A: Curse of dimensionality; each parameter has to be learned from data.

Sparse vs full connectivity

Chapter 9.2, fig 9.2 and 9.3

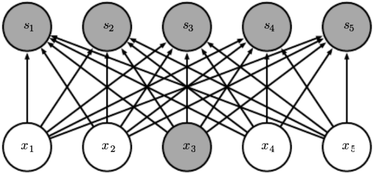
Full connectivity viewed from below:



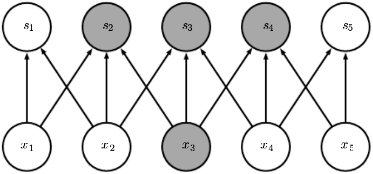
Sparse vs full connectivity

Chapter 9.2, fig 9.2 and 9.3

Full connectivity viewed from below:



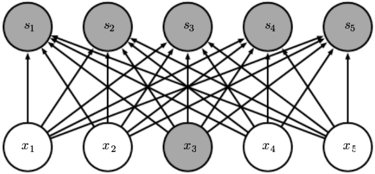
Sparse connectivity viewed from below:



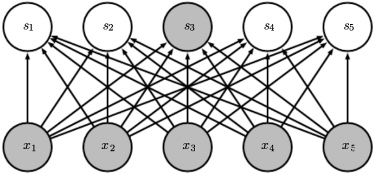
Sparse vs full connectivity

Chapter 9.2, fig 9.2 and 9.3

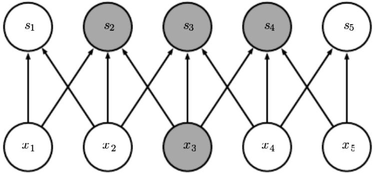
Full connectivity viewed from below:



Full connectivity viewed from above:



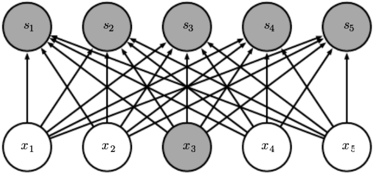
Sparse connectivity viewed from below:



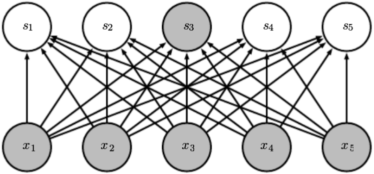
Sparse vs full connectivity

Chapter 9.2, fig 9.2 and 9.3

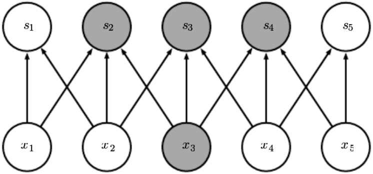
Full connectivity viewed from below:



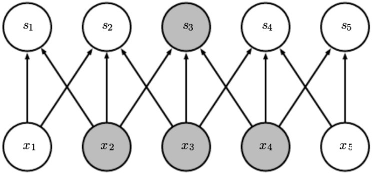
Full connectivity viewed from above:



Sparse connectivity viewed from below:



Sparse connectivity viewed from above:



Questions?

Equivariance

Chapter 9.2

- Q: What do you notice here?

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ +1] = [1,\ -2,\ 101,\ -1,\ 2]$$
$$[101,\ 99,\ 200,\ 199,\ 201,\ 201] \star [-1,\ +1] = [-2,\ 101,\ -1,\ 2,\ 0]$$

Equivariance

Chapter 9.2

- Q: What do you notice here?

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ +1] = [1,\ -2,\ 101,\ -1,\ 2]$$
$$[101,\ 99,\ 200,\ 199,\ 201,\ 201] \star [-1,\ +1] = [-2,\ 101,\ -1,\ 2,\ 0]$$

- A: If the input shifts to the left, the output shifts to the left (equivariance)

Equivariance

Chapter 9.2

- Q: What do you notice here?

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ +1] = [1,\ -2,\ 101,\ -1,\ 2]$$
$$[101,\ 99,\ 200,\ 199,\ 201,\ 201] \star [-1,\ +1] = [-2,\ 101,\ -1,\ 2,\ 0]$$

- A: If the input shifts to the left, the output shifts to the left (equivariance)
- Function $f()$ is equivariant to $g()$: $f(g(x)) = g(f(x))$.

Equivalance

Chapter 9.2

- Q: What do you notice here?

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ +1] = [1,\ -2,\ 101,\ -1,\ 2]$$
$$[101,\ 99,\ 200,\ 199,\ 201,\ 201] \star [-1,\ +1] = [-2,\ 101,\ -1,\ 2,\ 0]$$

- A: If the input shifts to the left, the output shifts to the left (equivariance)
- Function $f()$ is equivariant to $g()$: $f(g(x)) = g(f(x))$.
- Q: Let g be a translation, and f be a convolution, what does this equivariance mean?

Equivariance

Chapter 9.2

- Q: What do you notice here?

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ +1] = [1,\ -2,\ 101,\ -1,\ 2]$$
$$[101,\ 99,\ 200,\ 199,\ 201,\ 201] \star [-1,\ +1] = [-2,\ 101,\ -1,\ 2,\ 0]$$

- A: If the input shifts to the left, the output shifts to the left (equivariance)
- Function $f()$ is equivariant to $g()$: $f(g(x)) = g(f(x))$.
- Q: Let g be a translation, and f be a convolution, what does this equivariance mean?
- A: First translating, and then convolving, is the same as first convolving and then translating.

Equivalance

Chapter 9.2

- Q: What do you notice here?

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ +1] = [1,\ -2,\ 101,\ -1,\ 2]$$
$$[101,\ 99,\ 200,\ 199,\ 201,\ 201] \star [-1,\ +1] = [-2,\ 101,\ -1,\ 2,\ 0]$$

- A: If the input shifts to the left, the output shifts to the left (equivariance)
- Function $f()$ is equivariant to $g()$: $f(g(x)) = g(f(x))$.
- Q: Let g be a translation, and f be a convolution, what does this equivariance mean?
- A: First translating, and then convolving, is the same as first convolving and then translating.
- Q: How does equivariance relate to images?

Equivariance

Chapter 9.2

- Q: What do you notice here?

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ +1] = [1,\ -2,\ 101,\ -1,\ 2]$$
$$[101,\ 99,\ 200,\ 199,\ 201,\ 201] \star [-1,\ +1] = [-2,\ 101,\ -1,\ 2,\ 0]$$

- A: If the input shifts to the left, the output shifts to the left (equivariance)
- Function $f()$ is equivariant to $g()$: $f(g(x)) = g(f(x))$.
- Q: Let g be a translation, and f be a convolution, what does this equivariance mean?
- A: First translating, and then convolving, is the same as first convolving and then translating.
- Q: How does equivariance relate to images?
- A: Camera position is accidental, objects may appear anywhere

Equivariance

Chapter 9.2

- Q: What do you notice here?

$$[100,\ 101,\ 99,\ 200,\ 199,\ 201] \star [-1,\ +1] = [1,\ -2,\ 101,\ -1,\ 2]$$
$$[101,\ 99,\ 200,\ 199,\ 201,\ 201] \star [-1,\ +1] = [-2,\ 101,\ -1,\ 2,\ 0]$$

- A: If the input shifts to the left, the output shifts to the left (equivariance)
- Function $f()$ is equivariant to $g()$: $f(g(x)) = g(f(x))$.
- Q: Let g be a translation, and f be a convolution, what does this equivariance mean?
- A: First translating, and then convolving, is the same as first convolving and then translating.
- Q: How does equivariance relate to images?
- A: Camera position is accidental, objects may appear anywhere

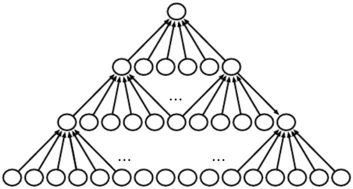
Add prior knowledge (convolution) to deep nets: huge gain in params and compute

Questions?

Padding

Chapter 9.5, fig 9.13

Lets do a convolution with size 6.

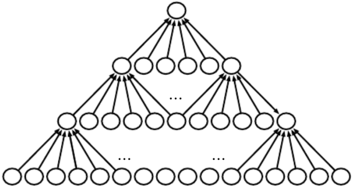


- Q: What do you see?

Padding

Chapter 9.5, fig 9.13

Lets do a convolution with size 6.

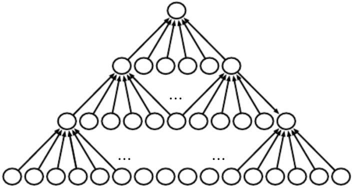


- Q: What do you see?
- A: Shrinks

Padding

Chapter 9.5, fig 9.13

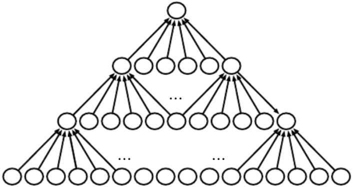
Lets do a convolution with size 6.



- Q: What do you see?
- A: Shrinks
- Q: How to prevent?

Padding
Chapter 9.5, fig 9.13

Lets do a convolution with size 6.

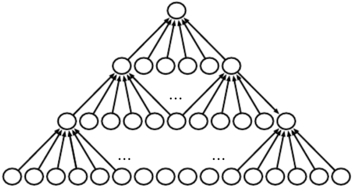


- Q: What do you see?
- A: Shrinks
- Q: How to prevent?
- A: Padding

Padding

Chapter 9.5, fig 9.13

Lets do a convolution with size 6.

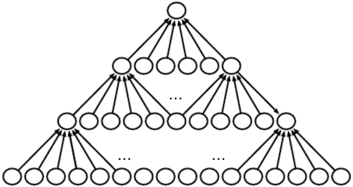


- Q: What do you see?
- A: Shrinks
- Q: How to prevent?
- A: Padding
- Q: What do you notice?

Padding

Chapter 9.5, fig 9.13

Lets do a convolution with size 6.

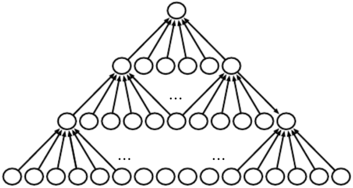


- Q: What do you see?
- A: Shrinks
- Q: How to prevent?
- A: Padding
- Q: What do you notice?
- A: Unequal border: 3 left, 2 right.

Padding

Chapter 9.5, fig 9.13

Lets do a convolution with size 6.



- Q: What do you see?
- A: Shrinks
- Q: How to prevent?
- A: Padding
- Q: What do you notice?
- A: Unequal border: 3 left, 2 right.
- A: Kernel is even, has no center.

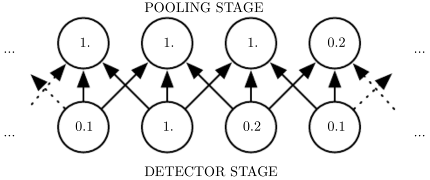
Questions?

Pooling

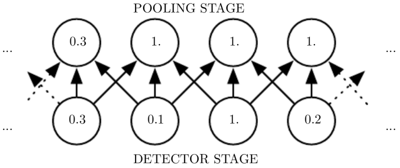
Chapter 9.3, fig 9.8

Pooling summarizes the outcome over a region.

Lets pool the maximum detector response for a width of 3:



Original



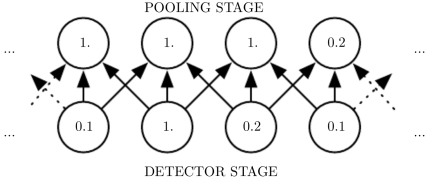
Shifted 1 pixel

Pooling

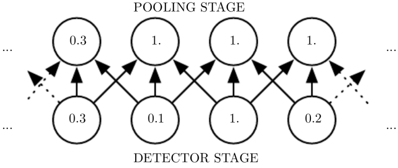
Chapter 9.3, fig 9.8

Pooling summarizes the outcome over a region.

Lets pool the maximum detector response for a width of 3:



Original



Shifted 1 pixel

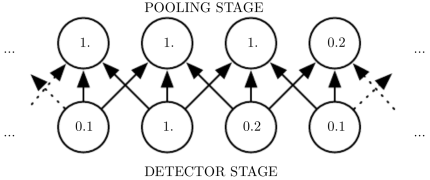
Q: What do you notice?

Pooling

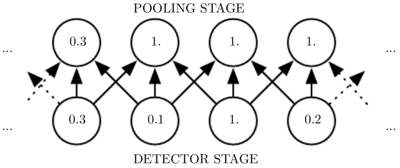
Chapter 9.3, fig 9.8

Pooling summarizes the outcome over a region.

Lets pool the maximum detector response for a width of 3:



Original



Shifted 1 pixel

Q: What do you notice?

A: All input values have changed; only 2 output values have changed.

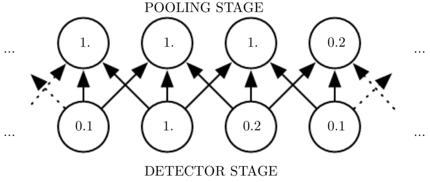
Pooling is approximately invariant to local translations.

Pooling

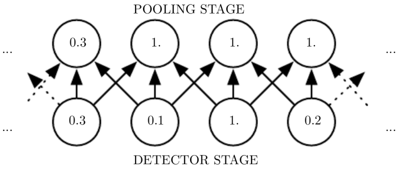
Chapter 9.3, fig 9.8

Pooling summarizes the outcome over a region.

Lets pool the maximum detector response for a width of 3:



Original



Shifted 1 pixel

Q: What do you notice?

A: All input values have changed; only 2 output values have changed.

Pooling is approximately invariant to local translations.
Function $f()$ is invariant to $g()$: $f(g(x)) = f(x)$

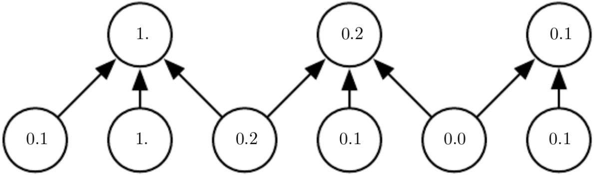
Q: Why is this useful for image classification?

A: Feature presence is more important then feature location.

Pooling and subsampling

Chapter 9.3, fig 9.10

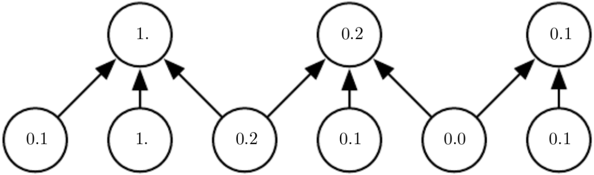
- Q: What do you notice here?



Pooling and subsampling

Chapter 9.3, fig 9.10

- Q: What do you notice here?



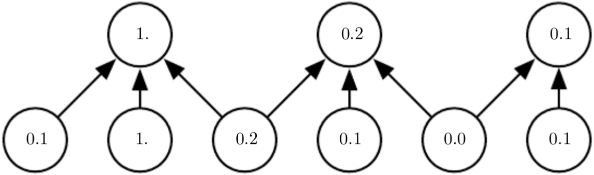
- A: Max-pooling width 3, Subsampling stride of 2.
- A: Highest response is kept.
- A: Size is reduced by 2^d, where d is dimensionality
- A: Summary statistics may be unbalanced

Reducing memory is a great advantage of pooling.

Pooling and subsampling

Chapter 9.3, fig 9.10

- Q: What do you notice here?



  - A: Max-pooling width 3, Subsampling stride of 2.
  - A: Highest response is kept.
  - A: Size is reduced by 2^d, where d is dimensionality
  - A: Summary statistics may be unbalanced

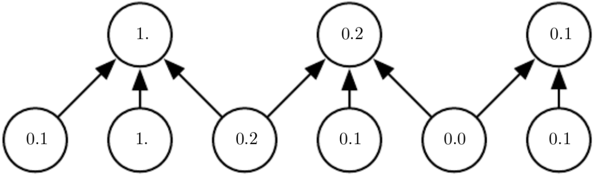
  Reducing memory is a great advantage of pooling.

- Q: For varying input size n; how to get a fixed size output m?

Pooling and subsampling

Chapter 9.3, fig 9.10

- Q: What do you notice here?



  - A: Max-pooling width 3, Subsampling stride of 2.
  - A: Highest response is kept.
  - A: Size is reduced by 2^d, where d is dimensionality
  - A: Summary statistics may be unbalanced

  Reducing memory is a great advantage of pooling.

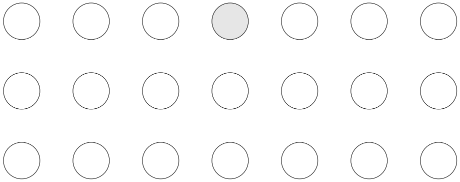
- Q: For varying input size n; how to get a fixed size output m?
- A: Adapt pool width depending on input size: $\frac{n}{m}$

Questions?

Receptive field: Convolution

Example: two size 3 convolution layers and one input image. (bottom)

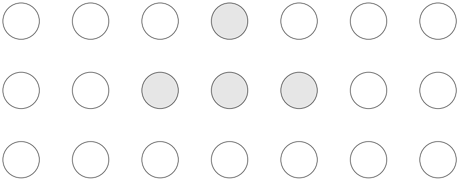
- Q: how much does the shaded neuron 'see' of the image?



Receptive field: Convolution

Example: two size 3 convolution layers and one input image. (bottom)

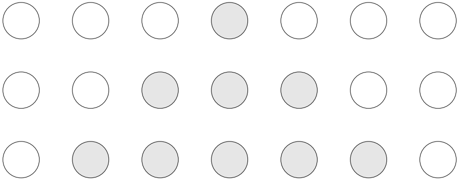
- Q: how much does the shaded neuron 'see' of the image?



Receptive field: Convolution

Example: two size 3 convolution layers and one input image. (bottom)

- Q: how much does the shaded neuron 'see' of the image?



- A: 5 pixels

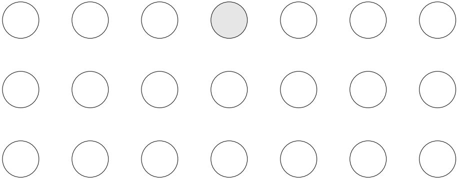
Receptive field: Striding

Chapter 9.5

Example: two size 3 convolution layers and one input image (bottom).

Let layer 1 do a convolution with stride 2 (sub-sample)

- Q: how much does the shaded neuron 'see' of the image?



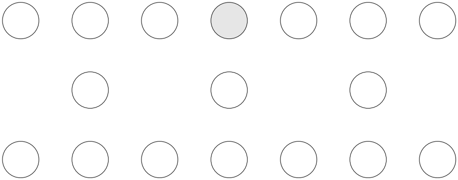
Receptive field: Striding

Chapter 9.5

Example: two size 3 convolution layers and one input image (bottom).

Let layer 1 do a convolution with stride 2 (sub-sample)

- Q: how much does the shaded neuron 'see' of the image?



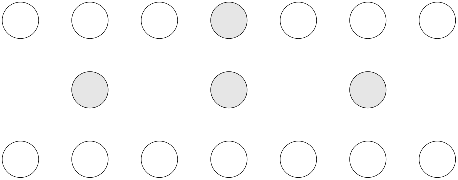
Receptive field: Striding

Chapter 9.5

Example: two size 3 convolution layers and one input image (bottom).

Let layer 1 do a convolution with stride 2 (sub-sample)

- Q: how much does the shaded neuron 'see' of the image?



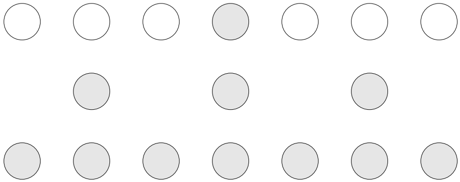
Receptive field: Striding

Chapter 9.5

Example: two size 3 convolution layers and one input image (bottom).

Let layer 1 do a convolution with stride 2 (sub-sample)

- Q: how much does the shaded neuron 'see' of the image?



- A: 7 pixels

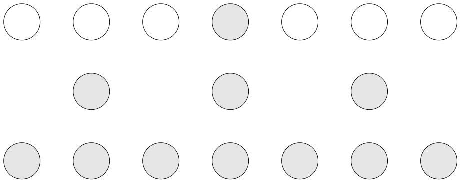
Receptive field: Striding

Chapter 9.5

Example: two size 3 convolution layers and one input image (bottom).

Let layer 1 do a convolution with stride 2 (sub-sample)

- Q: how much does the shaded neuron 'see' of the image?



- A: 7 pixels

- Q: How to characterize the effect of convolution/pooling on receptive field?

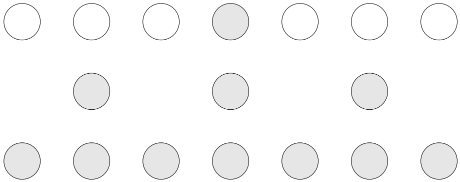
Receptive field: Striding

Chapter 9.5

Example: two size 3 convolution layers and one input image (bottom).

Let layer 1 do a convolution with stride 2 (sub-sample)

- Q: how much does the shaded neuron 'see' of the image?



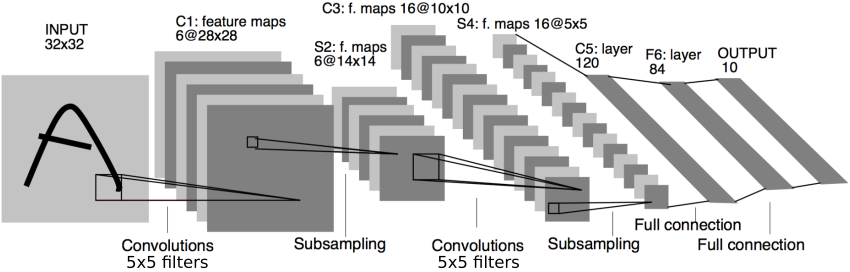
- A: 7 pixels

- Q: How to characterize the effect of convolution/pooling on receptive field?
- A: Convolution increases RF linearly, Pooling increases RF multiplicatively

Pooling allows to quickly 'see' more of the image.

Questions?

Questions?



Q: How to compute the number of parameters?

Q: How to compute the receptive field of a pixel in S4?