Photogrammetry & Robotics Lab

Convolutional Neural Networks

Cyrill Stachniss

The slides have been created by Cyrill Stachniss. I took a lot of inspiration from lectures given by Justin Johnson and Fei-Fei Li.

5 Minute Preparation for Today



https://www.ipb.uni-bonn.de/5min/

NN Part 1 on MLPs

- What are neurons and neural networks
- Activations, weights, biases
- Multi-layer perceptron (MLP)
- MLP for simple image classification



NN Part 2 on Training NNs

- Training multi-layer perceptrons
- Parameters are the weights and biases
- Learning/training NNs means estimating the weights and biases
- Minimization of a loss function
- Stochastic gradient descent for parameter optimization
- Backpropagation to compute gradients
- End-to-end: no manual features

Convolutional Neural Networks

In image-related learning tasks, CNNs play an important role



Convolutional Neural Networks

In image-related learning tasks, CNNs play an important role



[Image courtesy: van Veen] 6

Convolutional Neural Networks

In image-related learning tasks, CNNs play an important role



pixel intensities



An image consists of individual pixels.

Each pixel is an intensity value.





An image consists of individual pixels.

We have N+1 such intensity values.







Arrange all the intensity values in a N+1 dim vector.









Problem: the approach destroys the spatial information as it ignores the locations of the pixels in the image!

CNNs Overcome this Problem

- CNNs maintain the 2D image structure
- Neighborhoods are maintained
- Network layers can learn features that also encode spatial information
- Convolutions are local operators (see lecture on local operators)
- CNNs use convolutions & subsampling (called pooling)

Convolutions

Let's Start With the Input





Let's Start With the Input

channels/depth







depth=1



depth=3















example for blurring through a convolution













kernel size: $3 \times 5 \times 5 = 75$ \rightarrow dot product of 75 dim. vectors





We Can Use Multiple Kernels



Sizes

Size of the activation map depends on

- Size of the input (W, H)
- Kernel size (K)

$$W' \times H' = (W - K + 1) \times (H - K + 1)$$

output size input size and kernel size

Padding

- Convolutions slightly shrink the image
- We can solve this by creating a border around the input image



zero

wrap

clamp

mirror

Image courtesy: Szelinsky

CNNs often use zero-padding

Sizes

Size of the activation map depends on

- Size of the input (W, H)
- Kernel size (K)
- Padding (P)

 $W' \times H' = (W - K + 1 + 2P) \times (H - K + 1 + 2P)$

Tensor

- Vector is a 1 dimensional array
- Matrix is a 2-dimensional array
- Voxelgrid is a 3-dimensional array

Tensor = generalization of the "array" (matrix) concept with a flexible number of dimensions $\sqrt{2}$

$$\epsilon_{ijk} =$$



[Image courtesy: A. Kriesch] 30

Each Layer is a Tensor



Convolutional Layer Parameters



Stacking Convolutional Layers



kernel weights

Stacking Convolutions

 For such linear shift-invariant kernels, we know that concatenations of convolutions are again a convolution

$$f \ast (g_1 \ast \ldots \ast g_n) = f \ast g$$

- Multiple layers can be combined into a single convolution
- Property breaks when introducing non-linear activation function

Stacking Convolutions Layers With Activation Functions



Pooling
Pooling

- Besides convolutions, CNNs also use pooling layers
- Pooling combines multiple values into a single value to reduce the tensor sizes and combine information
- Prominent examples are:



max-pooling

avg-pooling

Stride

- Stride defines by how many pixels we shift the filter forward each step
- Larger stride reduces overlaps and makes the resulting image smaller



stride=1

Stride

- Stride defines by how many pixels we shift the filter forward each step
- Larger stride reduces overlaps and makes the resulting image smaller



stride=2

Size tells us how many values to combine and stride define by how much to shift the mask



 2×2

10	23	31	2	3	34			
8	35	44	33	1	45	35		
2	13	0	2	1	7			
12	3	8	22	9	88			
22	88	3	0	2	0	W/	$2 \times$	H
1	9	33	3	4	4	••/2		·· /

 $W \times H$



10	23	31	2	3	34			
8	35	44	33	1	45	35	44	
2	13	0	2	1	7			
12	3	8	22	9	88			
22	88	3	0	2	0	W/	$\frac{1}{2 \times 1}$	
1	9	33	3	4	4	v v /		11

 $W \times H$

/2

Max Pooling Example stride=2										
10	23	31	2	3	34					
8	35	44	33	1	45		35	44	45	
2	13	0	2	1	7					
12	3	8	22	9	88	V				
22	88	3	0	2	0		W/2	$2 \times$	H/2	
1	9	33	3	4	4			_	/-	

 $W \times H$

	10	23	31	2	3	34			
	8	35	44	33	1	45	35	44	45
V	2	13	0	2	1	7	13		
	12	3	8	22	9	88	13		
	22	88	3	0	2	0			 ΤΤ / ς
	1	9	33	3	4	4	VV / .	2 ×	$\Pi/2$
	1	9	33	3	4	4			

 $W \times H$

10	23	31	2	3	34			
8	35	44	33	1	45	35	44	45
2	13	0	2	1	7	13	22	88
12	3	8	22	9	88	88	33	4
22	88	3	0	2	0	$W/^{6}$	2 ×	H/2
1	9	33	3	4	4	•• / /		•• / ~

10	23	31	2	3	34							
8	35	44	33	1	45		35	44	45]		
2	13	0	2	1	7		13	22	88			
12	3	8	22	9	88		88	33	4	1		
22	88	3	0	2	0] •		
1	9	33	3	4	4	stride and size determine the						
						1	out	put	siz	e		

Receptive Field

- Stacking multiple pooling operations reduces W, H of the activation maps
- Elements in deeper layer are impacted by larger areas of the inputs



Normalization

Training CNNs

- Similar to MLPs, CNNs are trained using SDG and backpropagation
- Large number of parameters need to be determined
- Fairly large training sets are needed (end-to-end vs. given features)

SGD & Backpropagation for Training Neural Networks



https://youtu.be/4F0_V_0002Q



https://youtu.be/eAIAZIv2m0s

Normalization

- The first CNNs used convolutional and pooling layers in the first part
- Hard to train
- Normalizing the layers makes SGD converge faster
- Normalization of means and variance
- Somewhat unclear why it helps
- Different normalization approaches (batch, layer, instance, ...)

The First CNN [LeCun et al. 1989]

- First convolutional network
- Proposed by Yann LeCun et al.
- Recognition of handwritten digits
- Outperformed all other networks (at that time)
- 5 layers: 2 convolutional and 3 fully connected ones

- 5 layers: 2 convolutional and 3 fully connected ones
- Each convolutional layer combines: convolutions, pooling, activation fct.



[Image courtesy: LeCun et al.] 54




































CNNs Today

- CNNs are the standard approach for image-based tasks
- Networks also also used for other inputs (e.g., point clouds)
- A large number of architectures have been proposed

LeNet-5 (1989)



- The first CNN out there
- Used for character recognition

[Image courtesy: A. Dertat] 74

AlexNet (2012)



- Conceptually similar to LeNet-5
- Larger and deeper architecture

[Image courtesy: A. Dertat] 75

VGG-16/19 (2014)



- Stacks elements from AlexNet using smaller filters
- 16/19 layers
- 138M parameters (16 layer version) [Image courtesy: A. Dertat] 76

ResNet (2015)

- Very deep network:152 layers
- Consists of residual blocks



[Image courtesy: A. Dertat] 77

Convolutional Neural Networks



Convolutional Neural Networks



CNN Summary

- This lecture was a brief overview on convolutional neural networks (CNNs)
- Standard tool today for vision tasks
- Layers combining convolutional blocks, pooling, and normalization
- Classification layers at the end
- End-to-end trainable networks (input image to output)

Literature & Resources

- Goodfellow, Bengio, Courville: "Deep Learning" https://www.deeplearningbook.org/
- Online Book by Michael Nielsen, Chapter 1: http://neuralnetworksanddeeplearning.com/chap1.html
- Online book by Deisenroth, Faisal, Ong: Mathematics for Machine Learning https://mml-book.github.io/
- Alpaydin, Introduction to Machine Learning
- UMich NN Lecture by Johnson
- Standford AI Lectures by Li et al.

Slide Information

- The slides have been created by Cyrill Stachniss as part of the photogrammetry and robotics courses.
- I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.
- I took a lot of inspiration from lectures given by Justin Johnson (UMich, Stanford) and Fei-Fei Li (Stanford).
- If you are a university lecturer, feel free to use the course material. If you adapt the course material, please make sure that you keep the acknowledgements to others and please acknowledge me as well. To satisfy my own curiosity, please send me email notice if you use my slides.

Cyrill Stachniss, cyrill.stachniss@igg.uni-bonn.de