Neuronale Netze
Convolutional Neural Networks (CNNs)

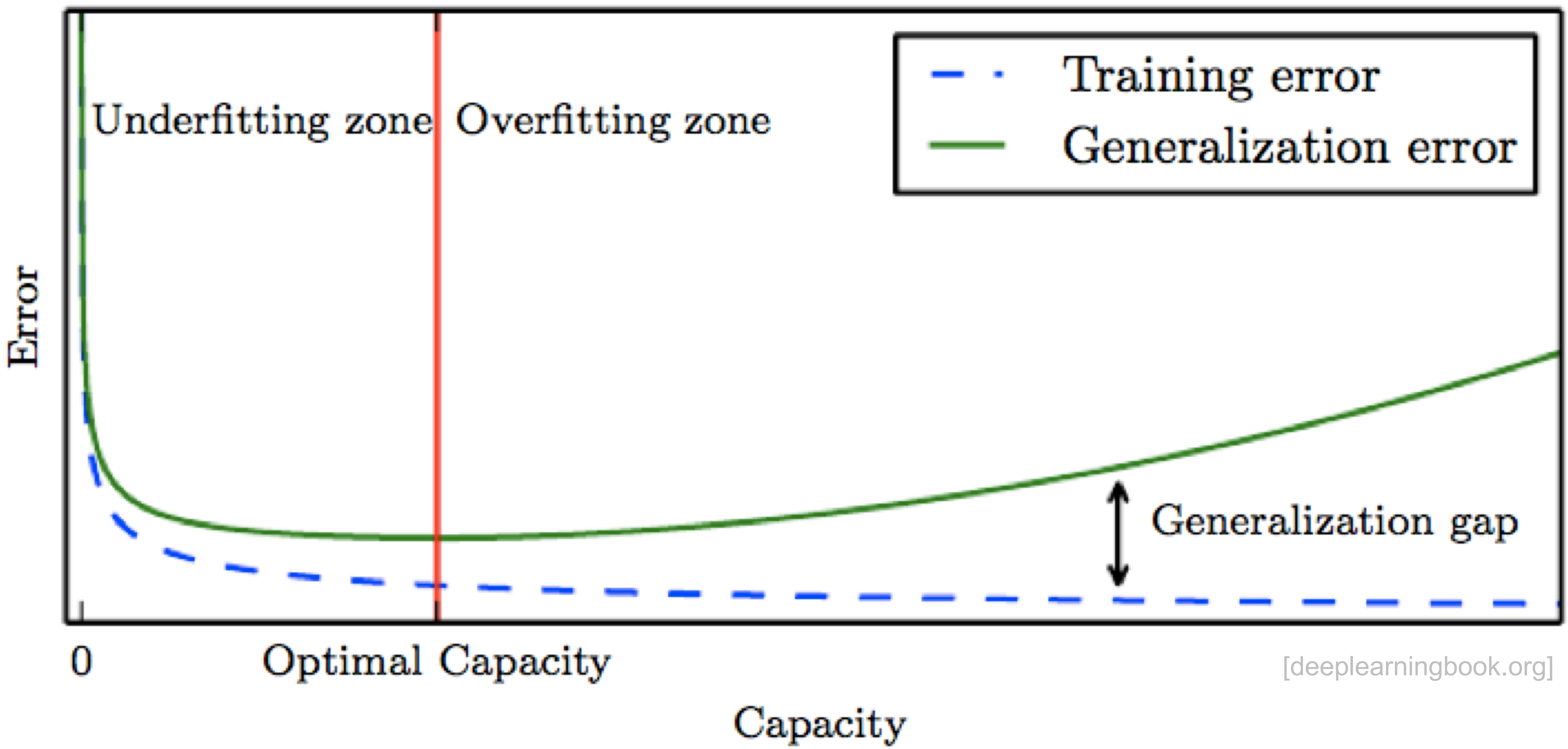
Prof. Dr.-Ing. Sebastian Stober

Artificial Intelligence Lab
Institut für Intelligente Kooperierende Systeme
Fakultät für Informatik
stober@ovgu.de

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
FACULTY OF COMPUTER SCIENCE
AILAB

Recap

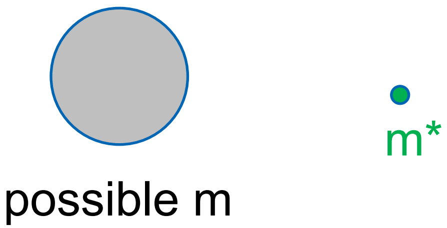
Modell-Kapazität



Bewertung & Selektierung von Modellen nur auf der Basis bisher ungesehener Daten!

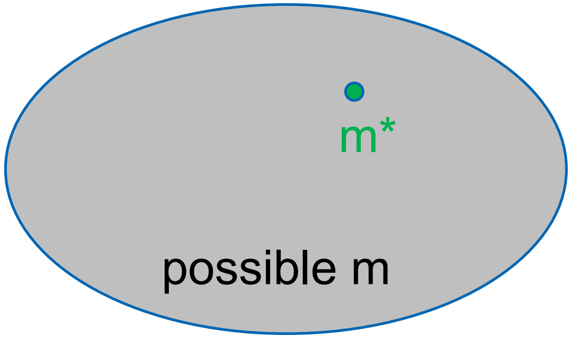
Bias-Variance Trade-Off

- high bias, low variance:

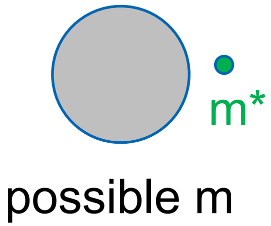


- low bias, high variance:

regularize!



- good trade-off:



Regularization Techniques

- parameter norm (L1/L2)
- early stopping
- dropout
- more data / data augmentation
- adding noise / denoising
- semi-supervised learning
- multi-task learning
- parameter tying & sharing
- sparse representations
- bagging / ensembles
- DropConnect = randomly set weights to zero
- (layer-wise) unsupervised pretraining
- adversarial training
- …

Practical Methodology

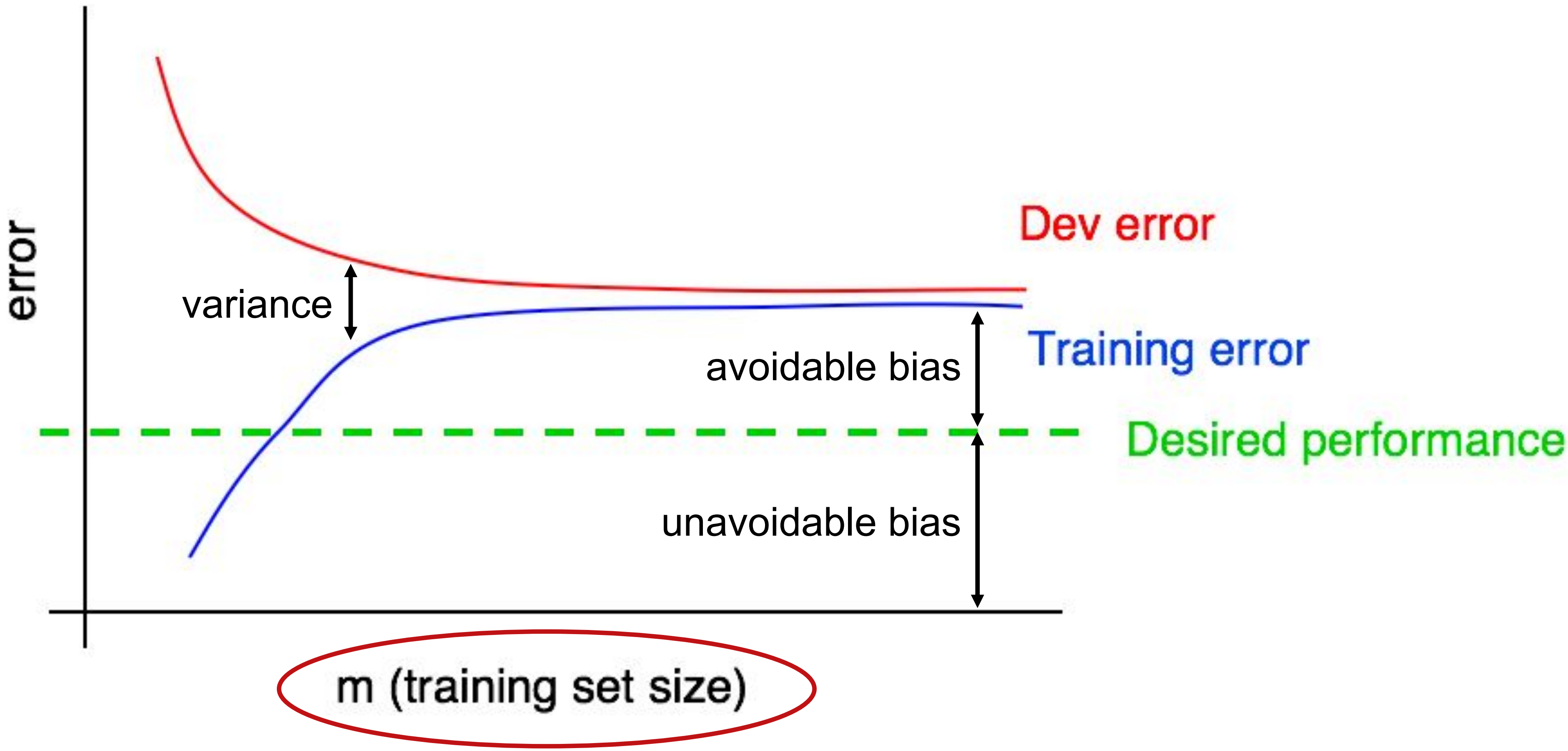
adapted from Andrew Ng. “Machine Learning Yearning” (draft), 2018

Bias & Variance (continued)

- optimal error rate (“unavoidable bias”)
  - needs to be estimated somehow (e.g. human error)
- avoidable bias (training error – optimal error rate)
- “variance” (generalization error)

- high avoidable bias (underfitting)
  - try to reduce training set error first: increase model size (capacity), modify input features, reduce regularization
- high variance (overfitting)
  - regularize, add more data, decrease model size, decrease number/type of input features (selection)
- both: modify model architecture

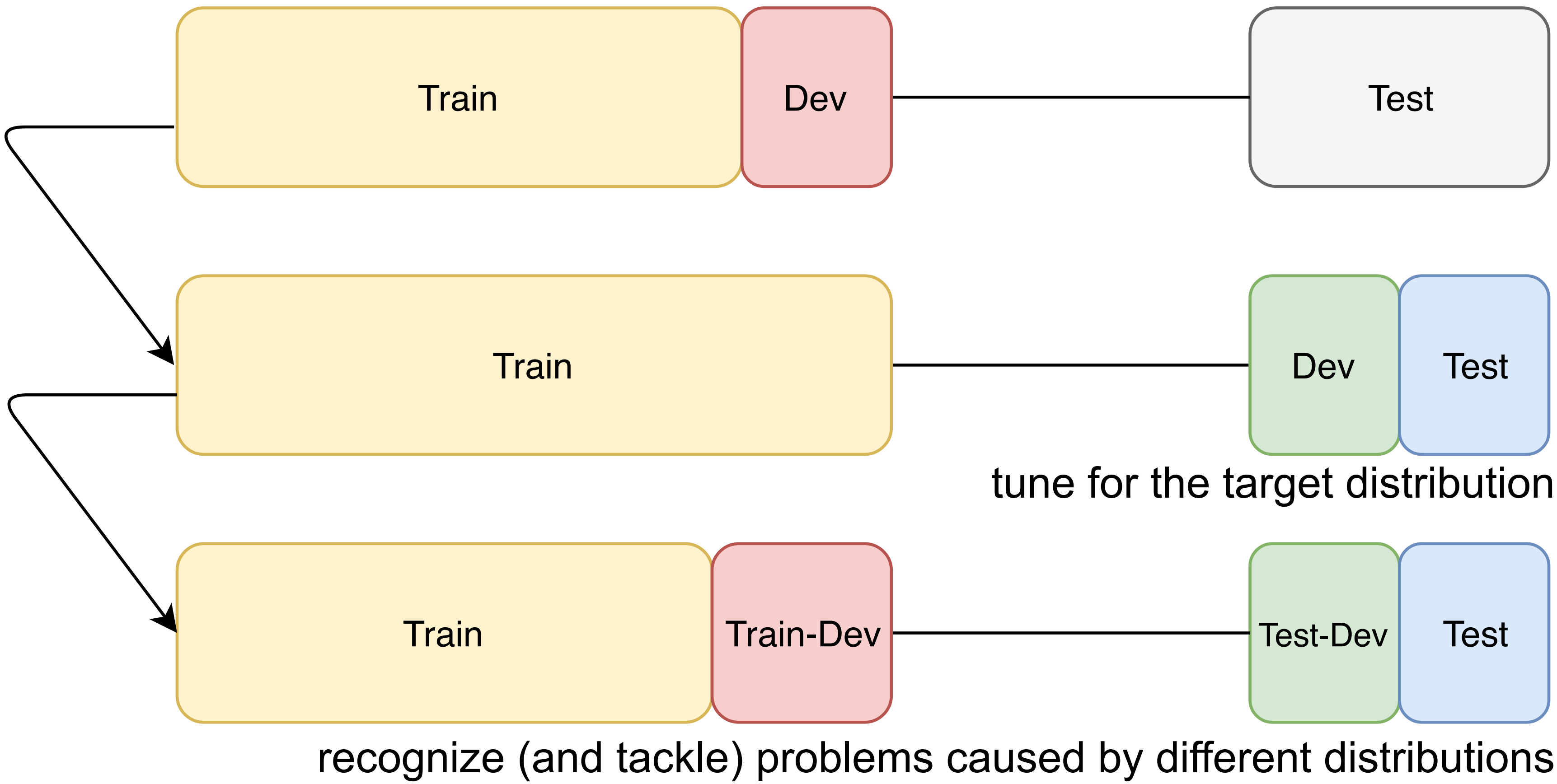
Learning Curves



adapted from A. Ng. “Machine Learning Yearning” (draft), 2018

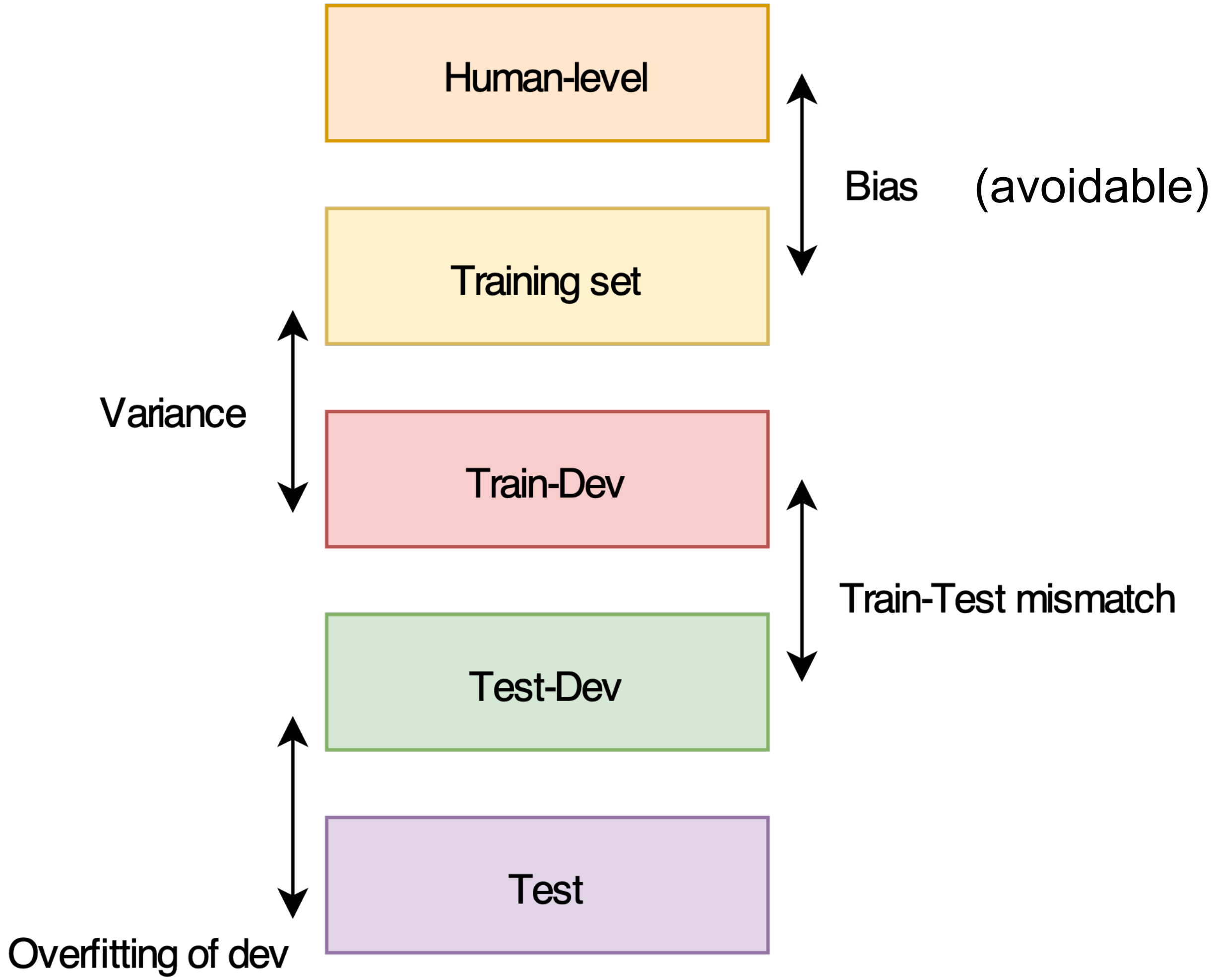
Data Splits for Different Distributions

=> Make dev and test sets come from the same distribution!



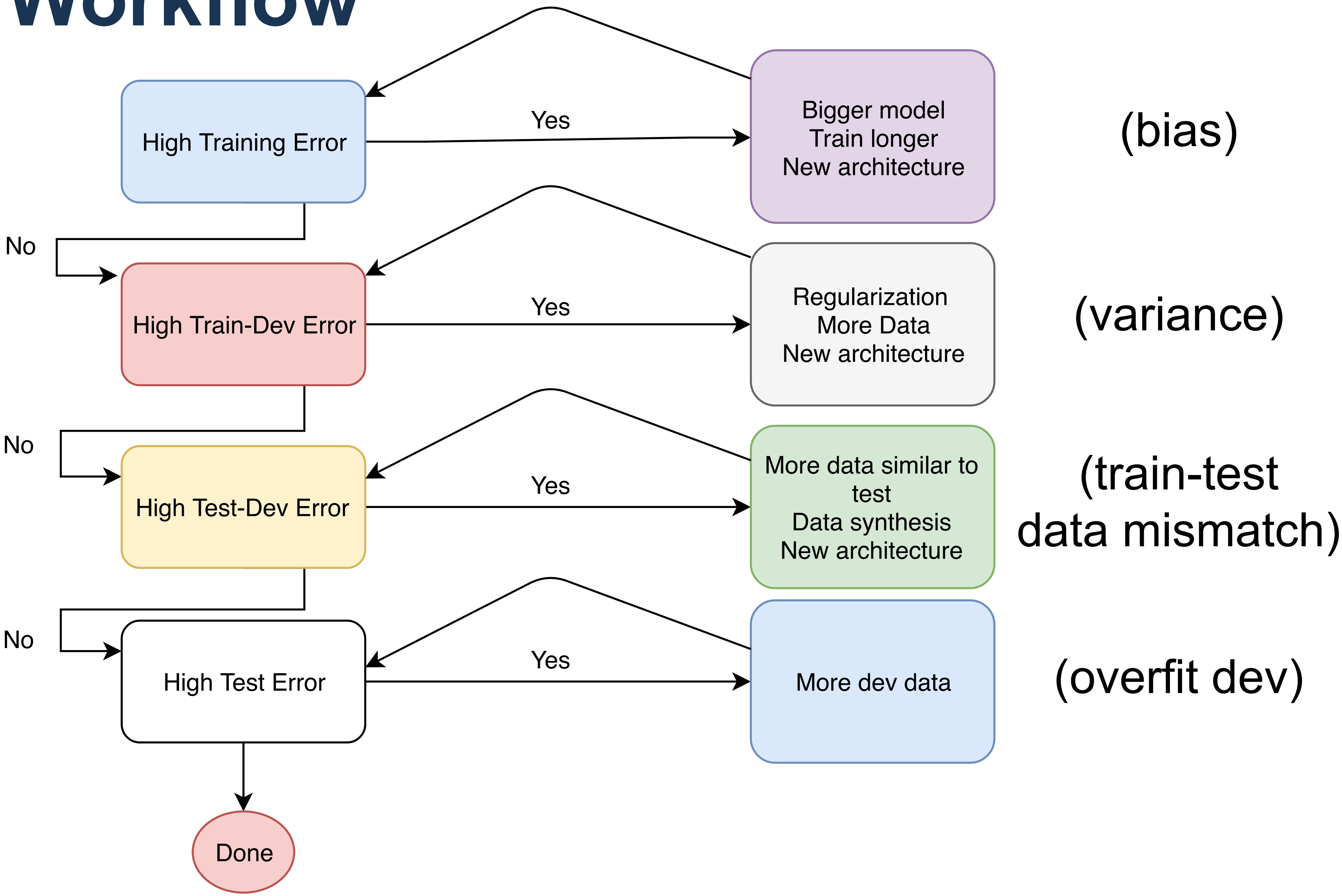
https://kevinzakka.github.io/2016/09/26/applying-deep-learning/

Error Factors



https://kevinzakka.github.io/2016/09/26/applying-deep-learning/

Workflow



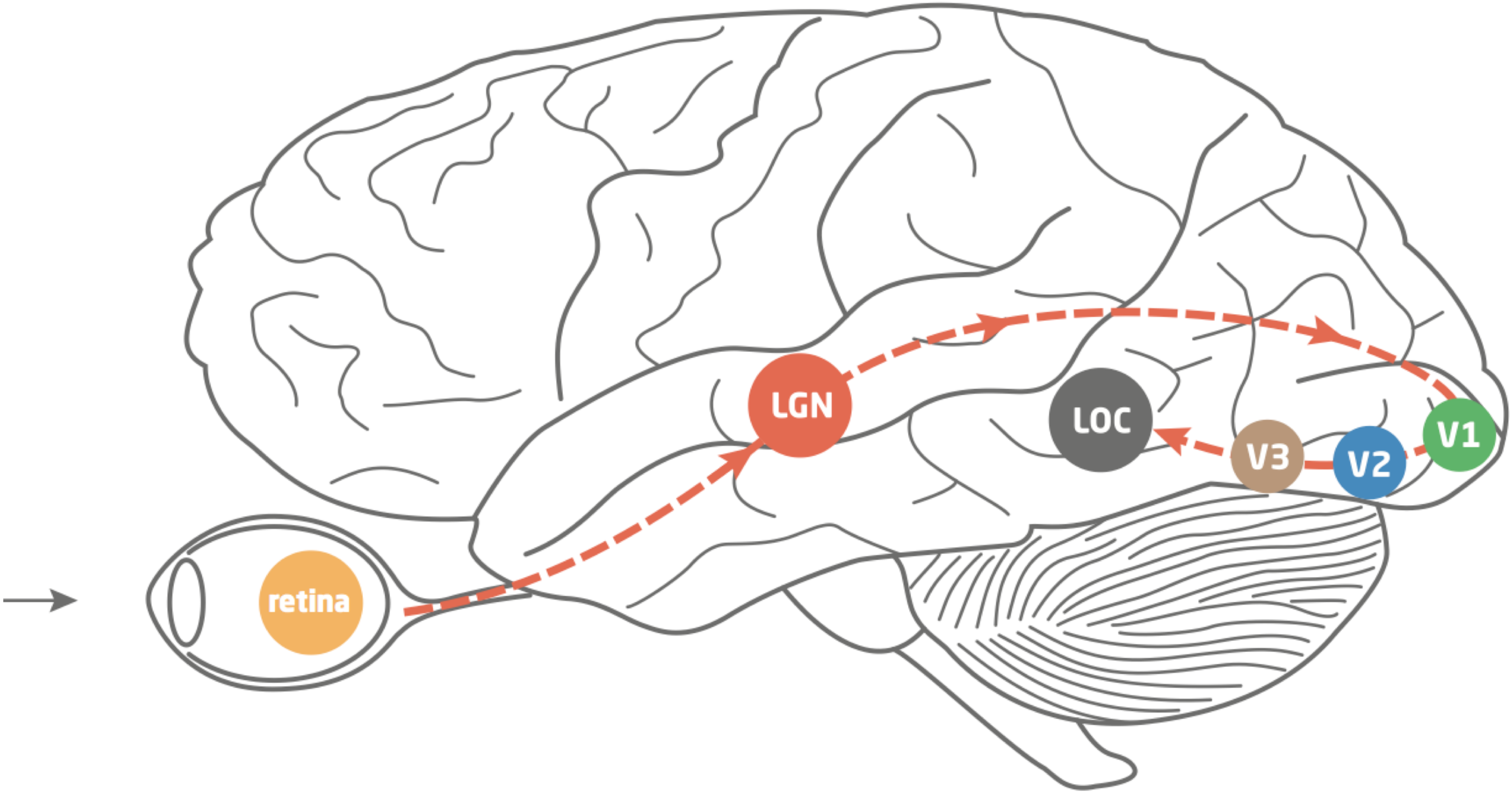
https://kevinzakka.github.io/2016/09/26/applying-deep-learning/

Further Reading

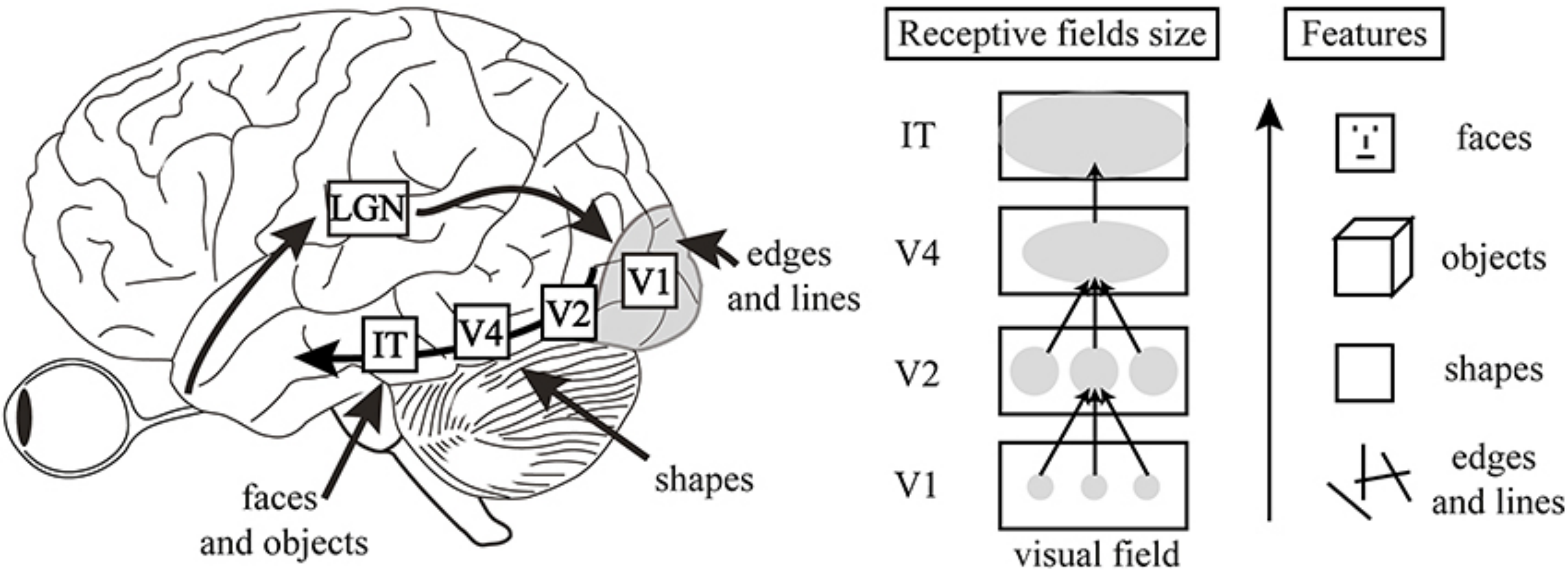
- http://mlyearning.org/
- http://mlexplained.com/2018/04/24/overfitting-isnt-simple-overfitting-re-explained-with-priors-biases-and-no-free-lunch/
- https://karpathy.github.io/2019/04/25/recipe/
- https://lilianweng.github.io/lil-log/2019/03/14/are-deep-neural-networks-dramatically-overfitted.html

CNNs

Modelling the Visual System

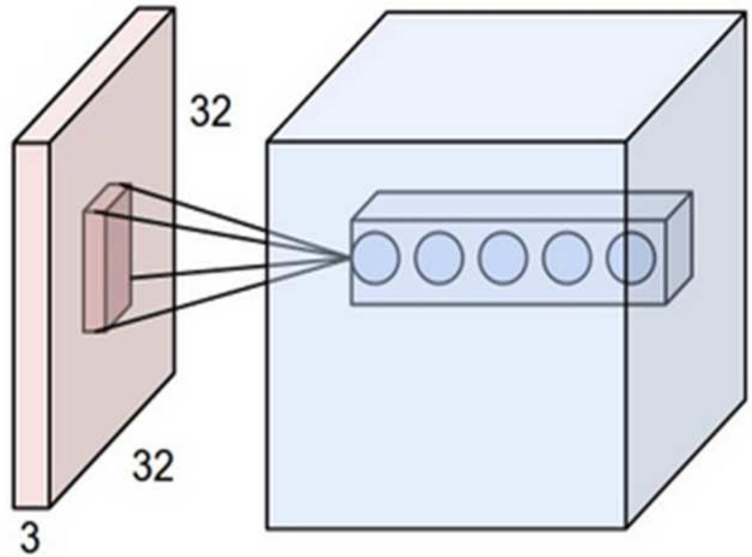


Modelling the Visual System



https://neurdiness.wordpress.com/2018/05/17/deep-convolutional-neural-networks-as-models-of-the-visual-system-qa/

Faltung (Convolution)



Motivation: Egal wo auf dem Bild ein Objekt ist, soll es erkannt werden

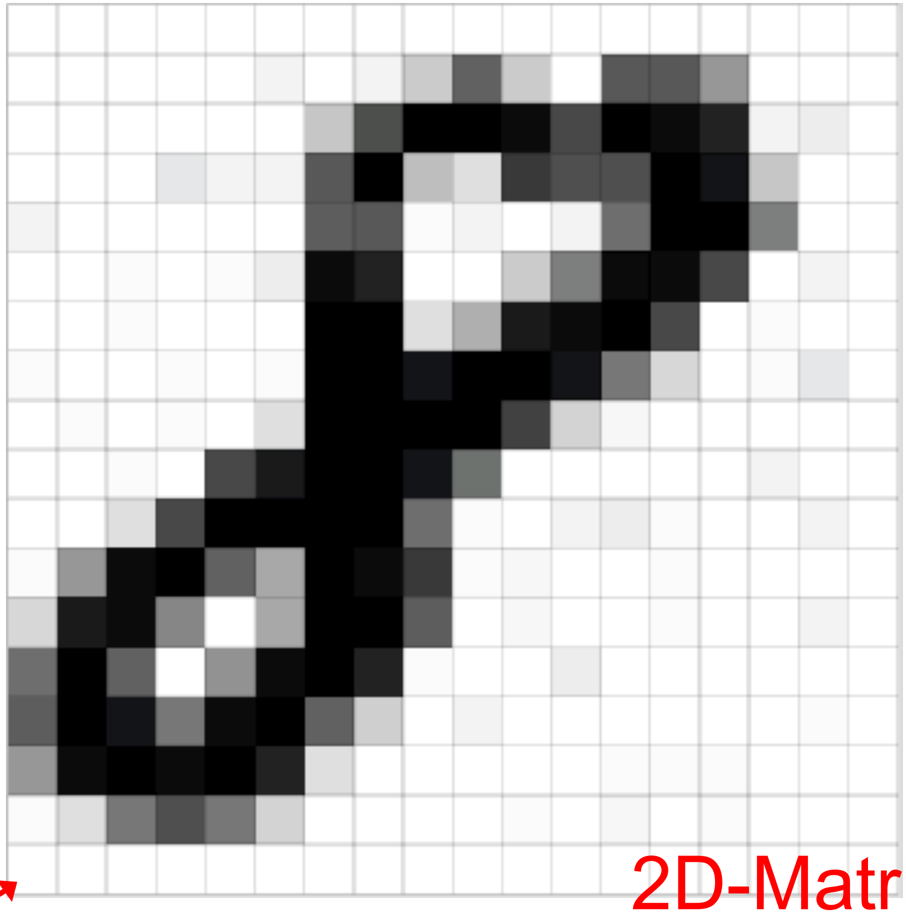
Idee: Verwende die selben Features auf dem gesamten Bild

Umsetzung: Filter / Kernel werden auf jedem Teil des Bildes angewandt und teilen sich die Gewichte

Ein Kurzer Exkurs in die Digitale Bildverarbeitung

Bilder sind Zahlen!

Pixel (Bildpunkt)



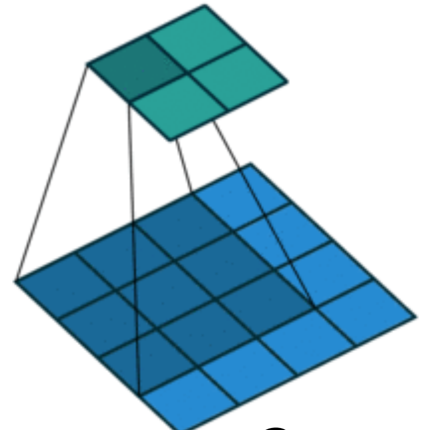
2D-Matrix

Ein Kurzer Exkurs in die Digitale Bildverarbeitung

original

grayscale

Otsu threshold



Convolution

jitter

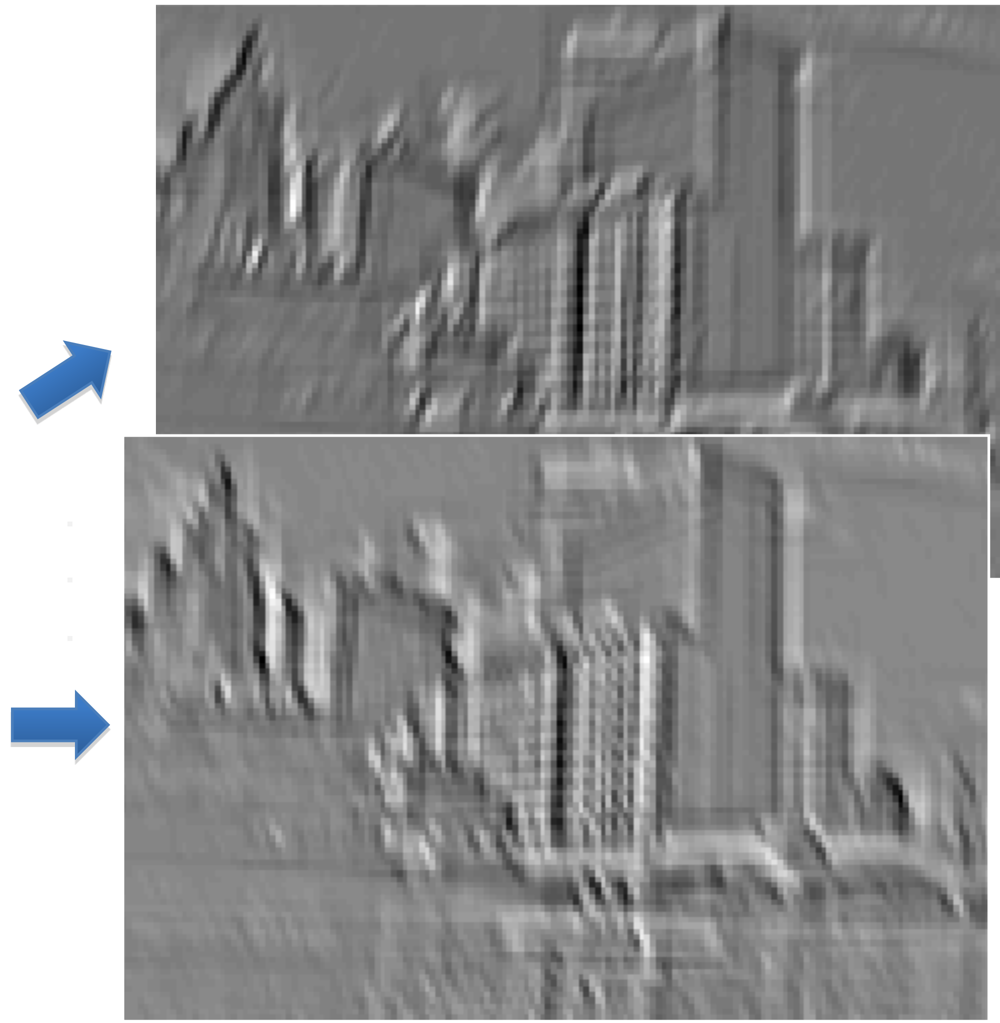
median filter

dilation

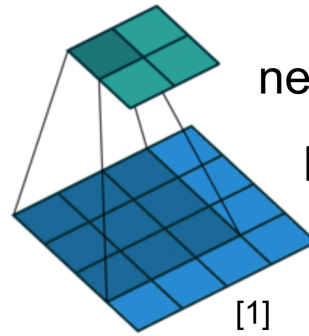
erosion

Ein Kurzer Exkurs in die Digitale Bildverarbeitung

Kantenerkennung
mit mehreren Filtern

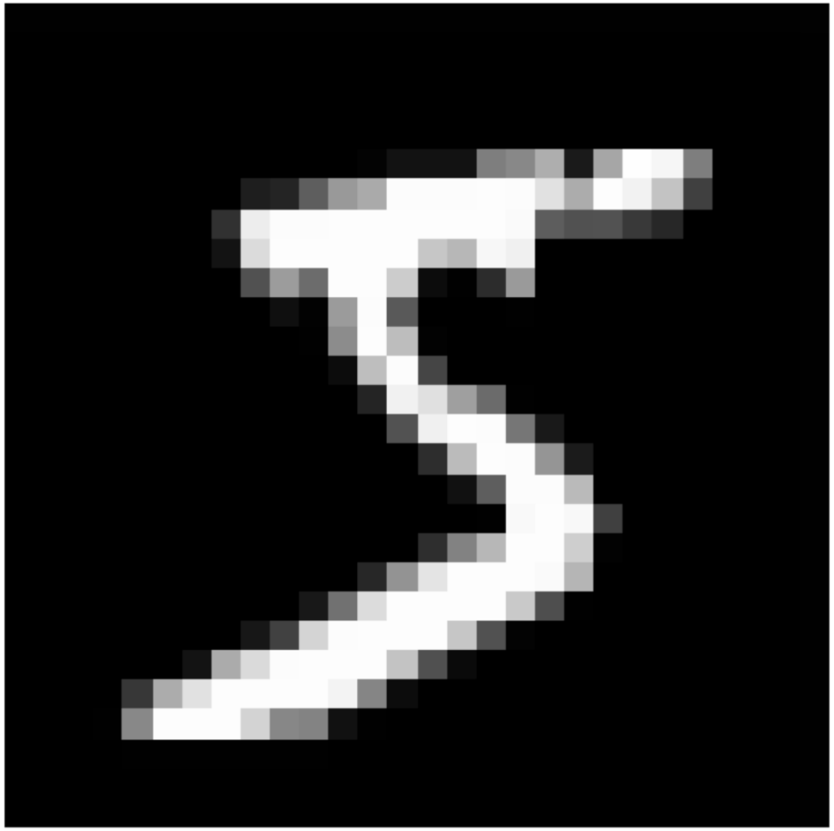


Convolutional Neural Nets (CNNs)



specialized group of neurons in visual cortex

limited & overlapping receptive fields with same filter



example from MNIST dataset
http://yann.lecun.com/exdb/mnist/

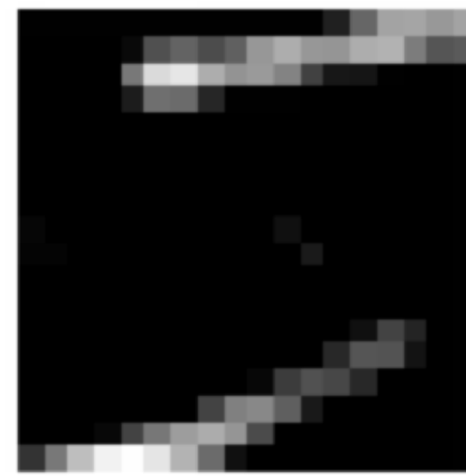
[1] http://deeplearning.net/software/theano/tutorial/

2D input

×

learnable filter (feature)

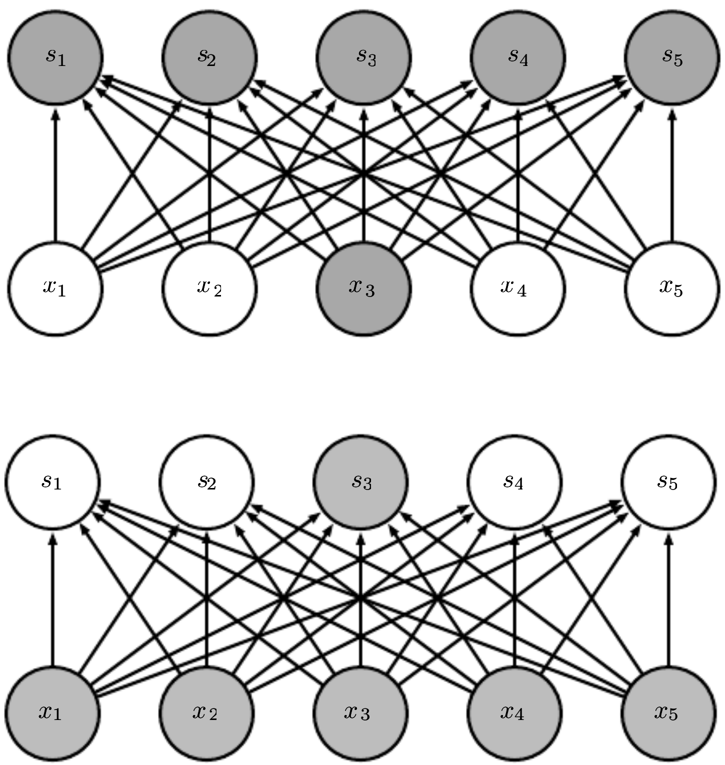
=



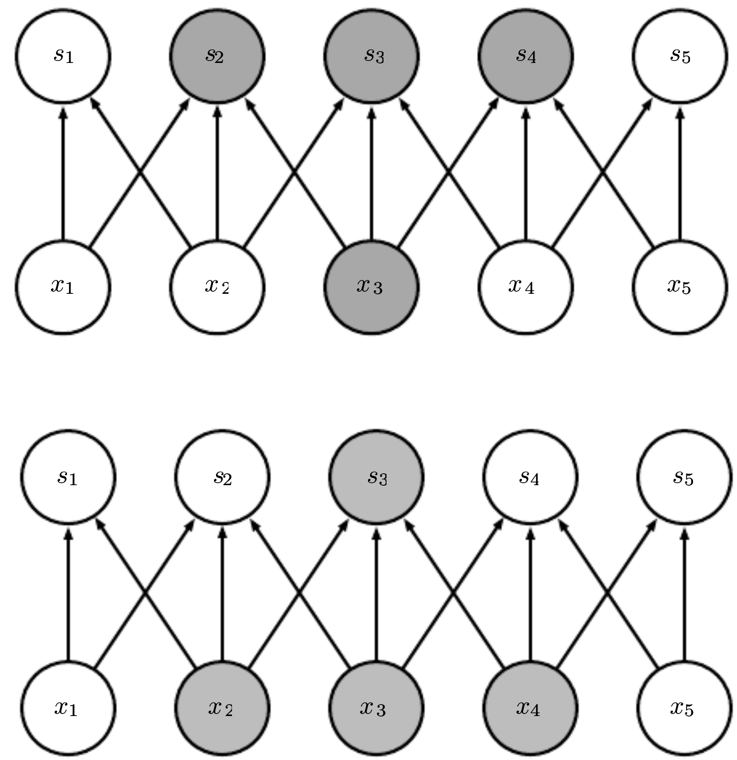
2D output (feature map)

1. Local Connectivity

fully-connected (dense) layer



convolutional layer

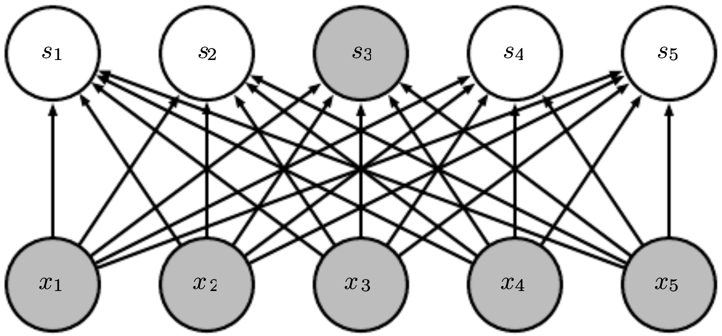


receptive field

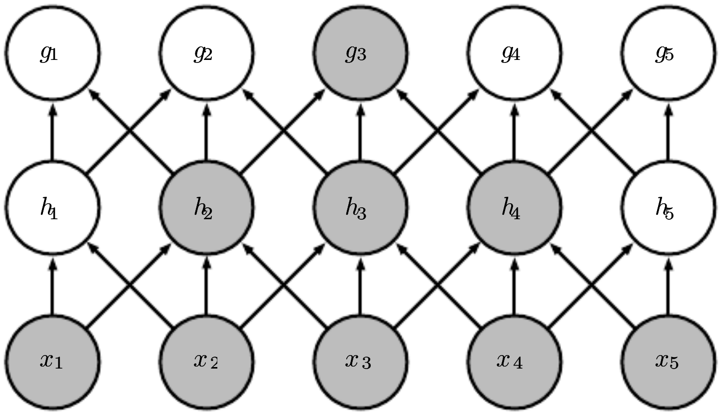
[http://www.deeplearningbook.org/contents/convnets.html]

1. Local Connectivity

1 fully-connected (dense) layer



2 convolutional layers

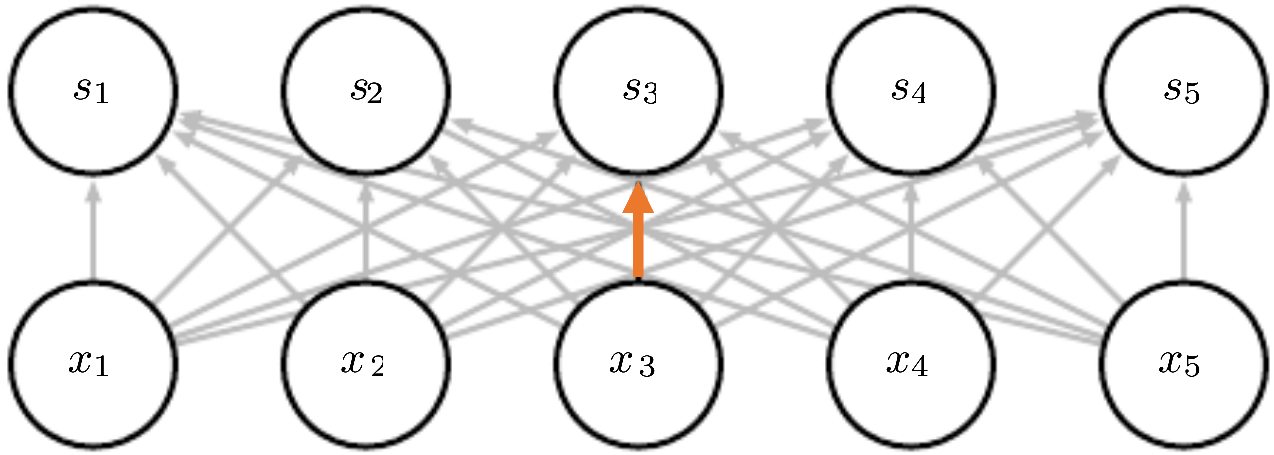


growing receptive field

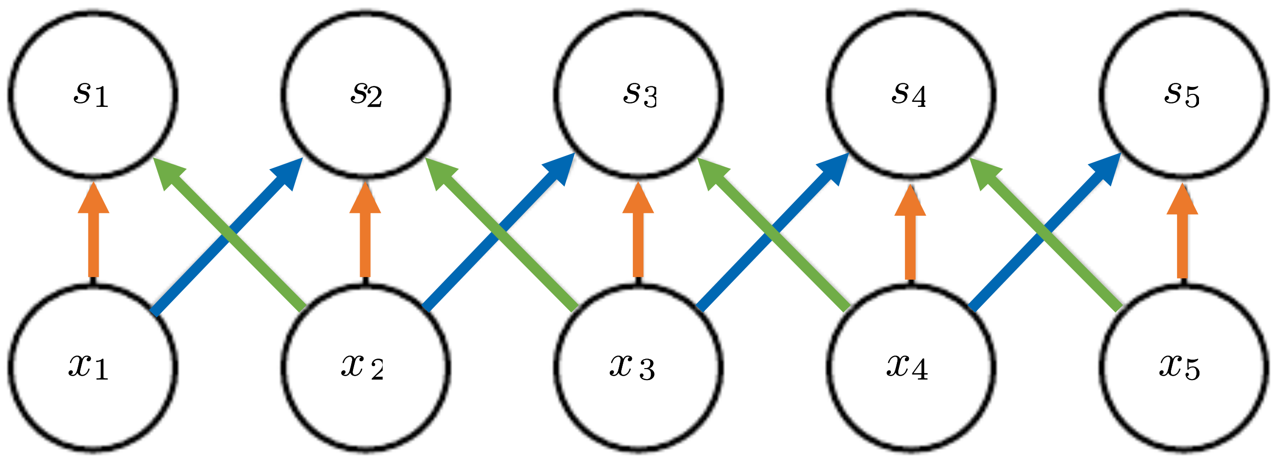
[http://www.deeplearningbook.org/contents/convnets.html]

2. Shared Weights

fully-connected



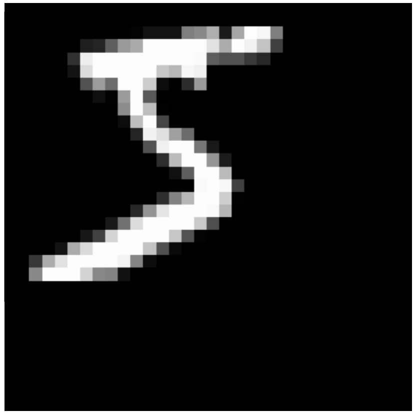
convolutional



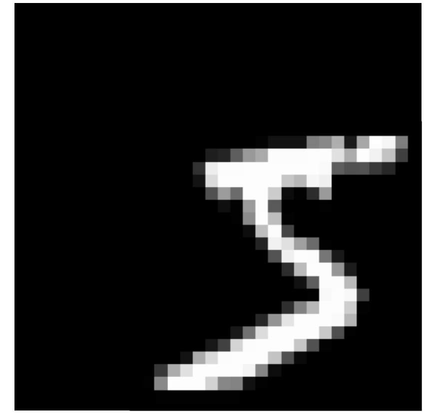
[http://www.deeplearningbook.org/contents/convnets.html]

3. Translation Equivariance

input

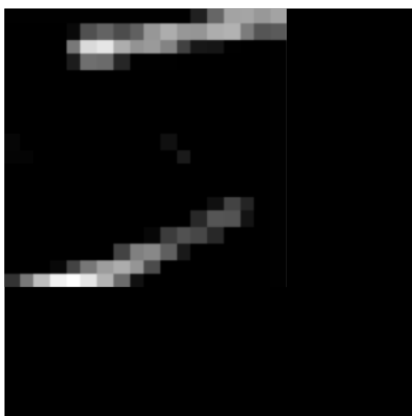


translations
in the input

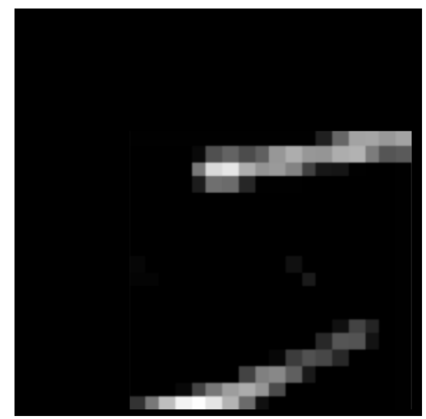


result in

feature map

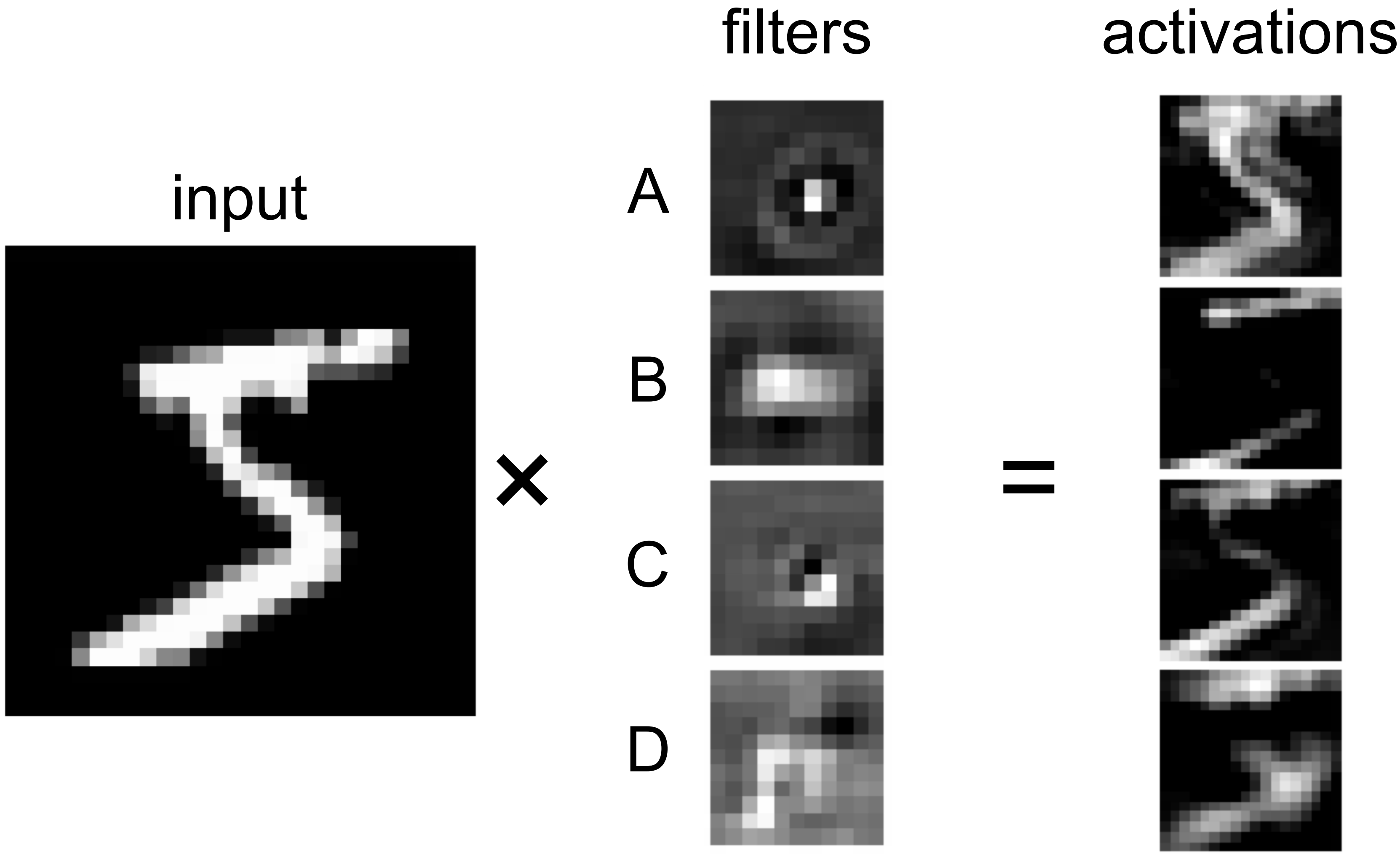


translations
in the output
(activations)

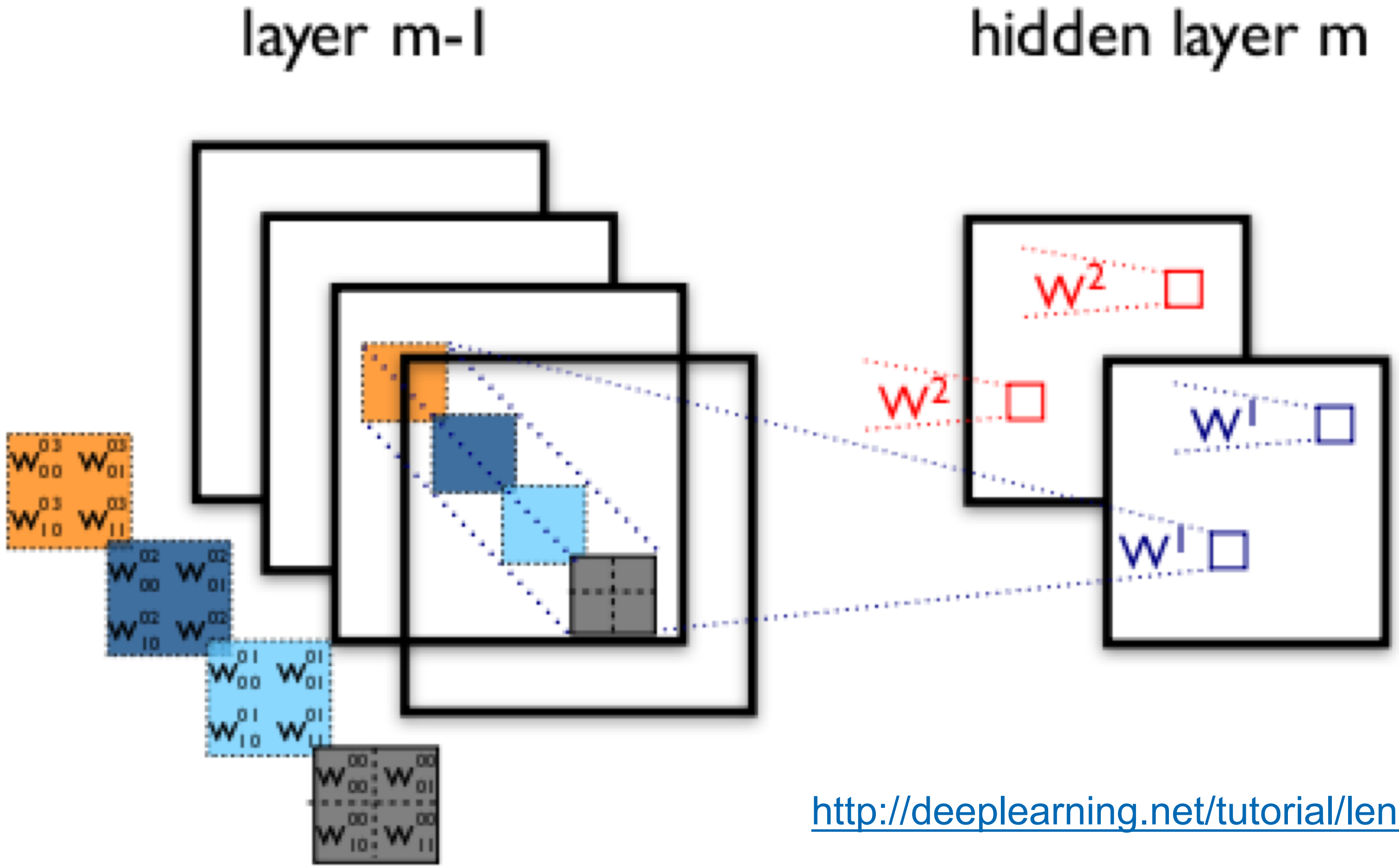


This is NOT invariance!

Filters & Activations



Multi-Channel Filter Input



http://deeplearning.net/tutorial/lenet.html

Generally no convolution along channel axis!

Filter Output Size

Image

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Filter

1	0	1
0	1	0
1	0	1

Convolved Feature

4	3	4
2	4	3
2	3	4

Featuretransformation

Schiebe einen „Filter“ über die Features und betrachte die „gefilterten“ Features

Multipliziere Originalfeature mit Filter und Summiere

Originalraum: 5x5

Filtergröße: 3x3

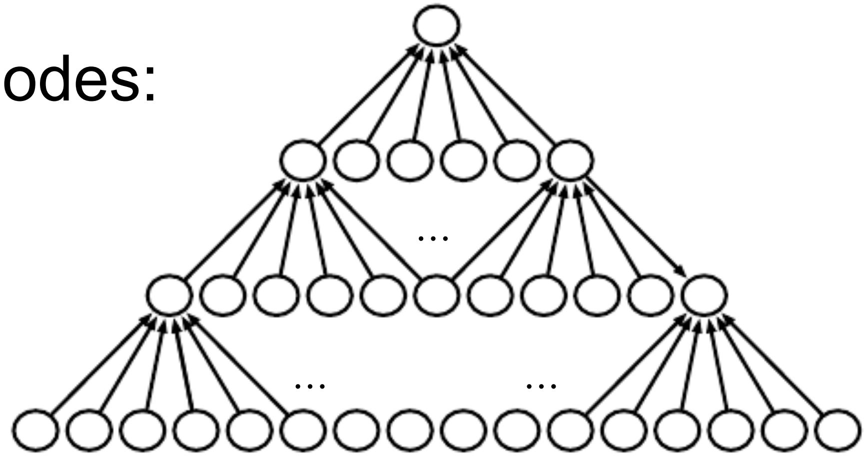
Neue Featuregröße: 3x3

Featureraum wird kleiner

To Pad or Not to Pad?

popular convolution modes:

- valid
- same/half
- full

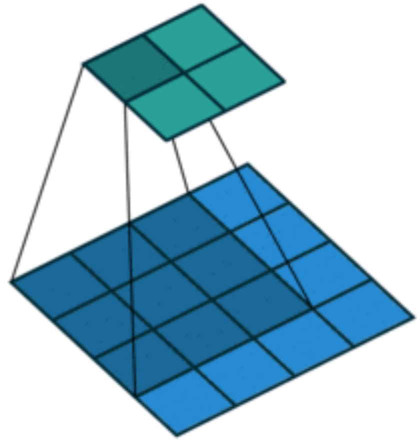


[http://www.deeplearningbook.org/contents/convnets.html]

Padding in “valid” Mode

- rationale:
  only apply filters in actual (valid) data, i.e. no padding

- given a 1D-input with length *n* and a convolutional filter with length *k*, the resulting output size is *n*-*k*+1



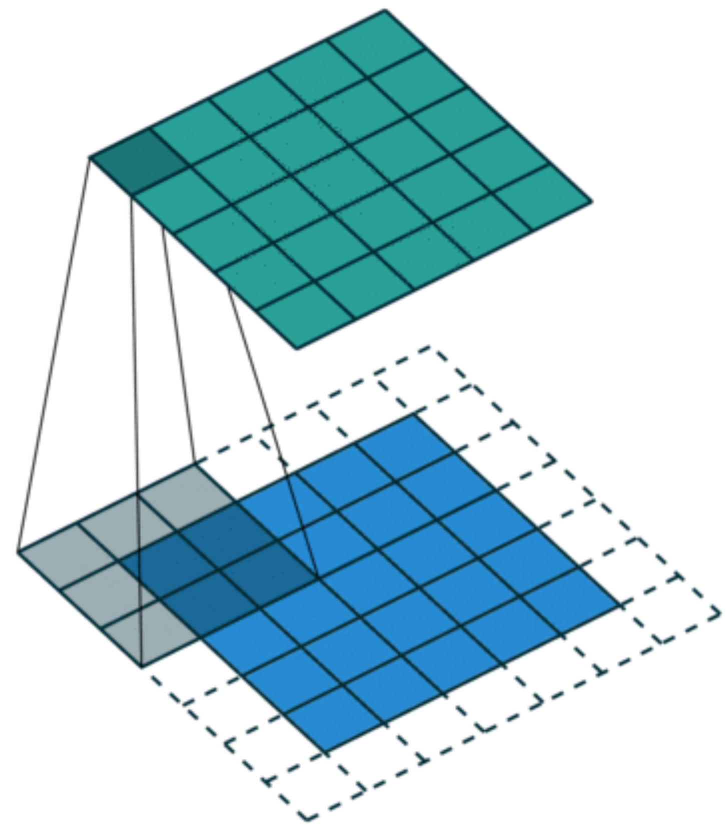
2D valid mode padding example

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Padding in "same" Mode

- rationale:
  output has the same size as the input

- given a 1D-input with length *n* and a convolutional filter with length *k*, add (*k*-1) / 2 zeros at each end of the input



2D same mode padding example

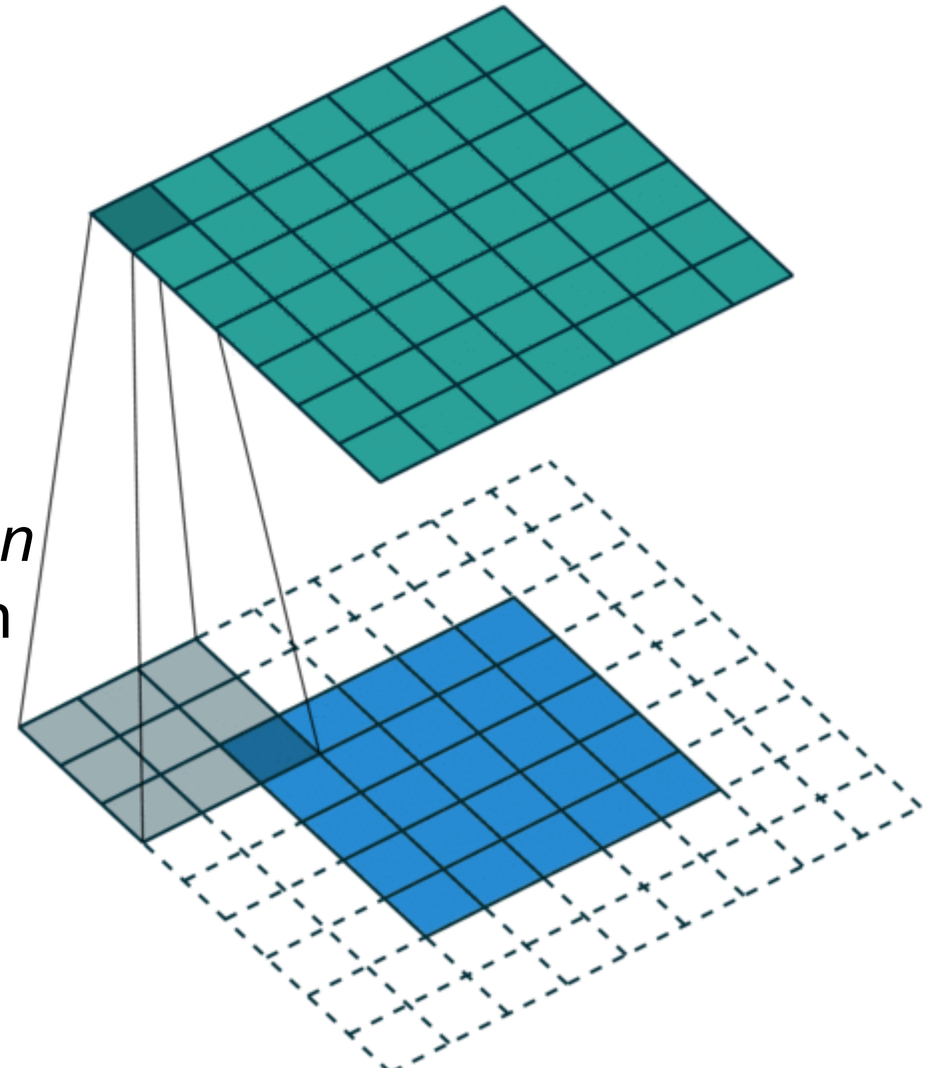
http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Padding in “full” Mode

- rationale:
  consider every possible superimposition of filter and input

- given a 1D-input with length *n* and a convolutional filter with length *k*, add *k*-1 zeros at each end of the input

- size of output increased by *k*-1



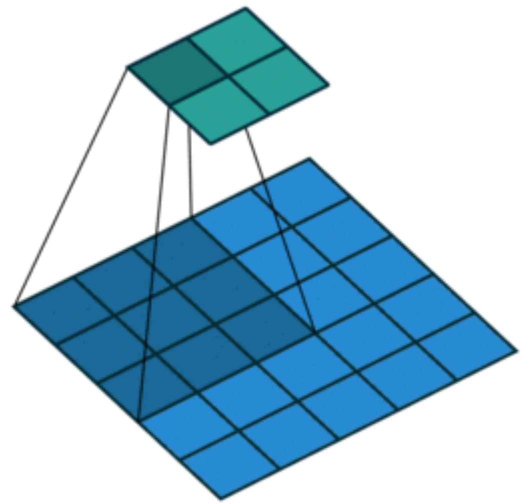
2D full mode padding example

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Strided Convolution

- rationale:
  decrease resolution (and thus dimensionality)
  reduce computation

- same effect as down-sampling



2D strided convolution example

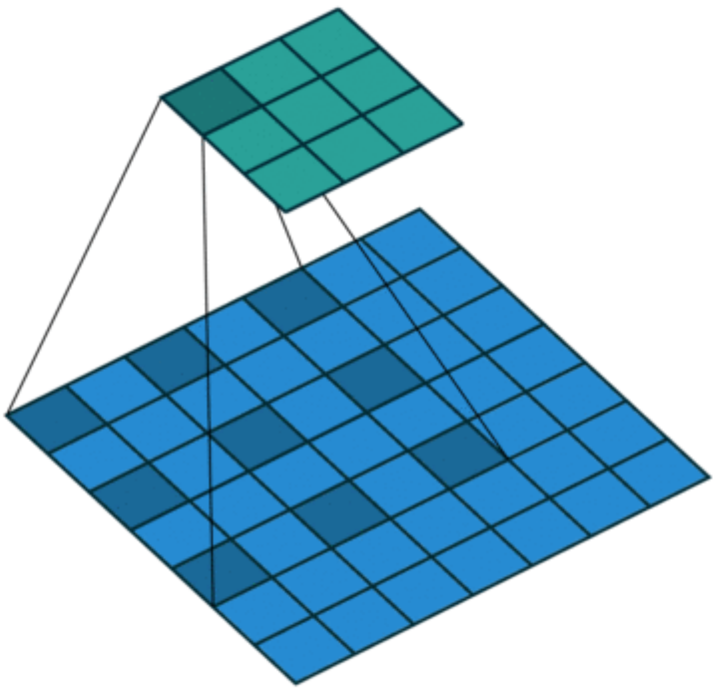
http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Dilated Convolution

- rationale:
  increase receptive field size

- "inflate" filter by inserting
  spaces between filter elements

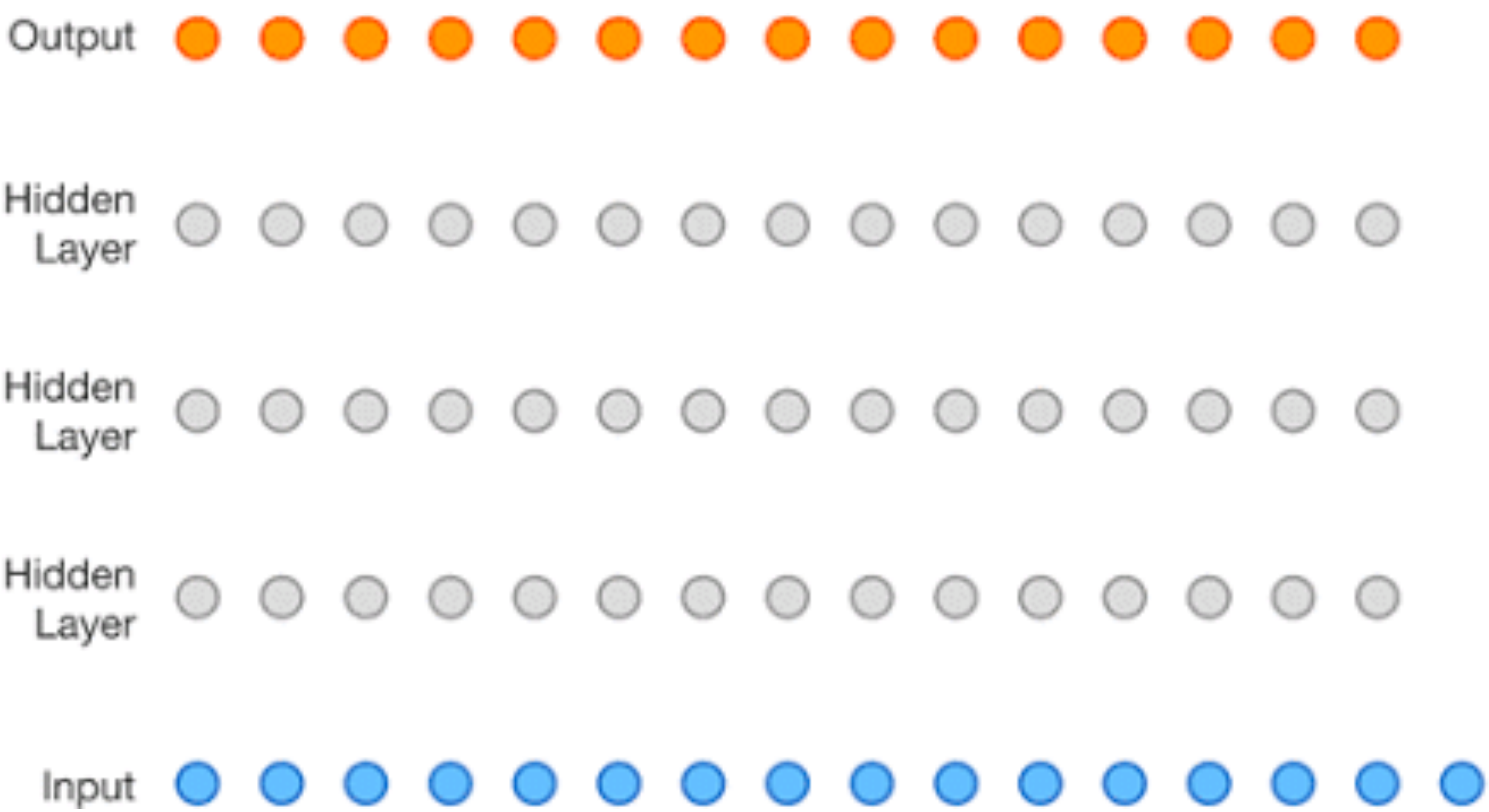
- dilation rate *d* corresponds to
  d-1 spaces



2D dilated convolution example

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Dilated Convolution Stacks

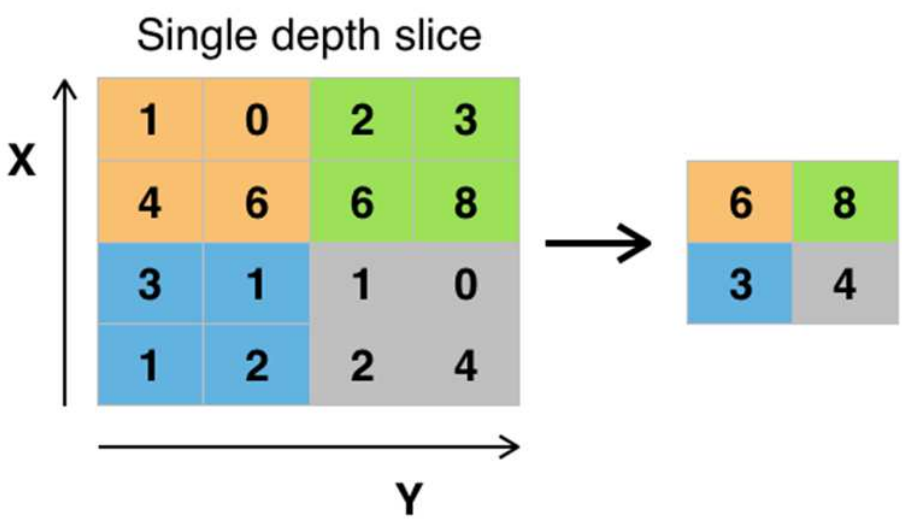


https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Convolution & Pooling

- convolution
  - equivariance: if the input changes, the output changes in the same way
- pooling
  - approximate invariance to small translations
  - trade-off: whether? vs. where?
  - special case: maxout-pooling (pooling over several filters => learn invariance)

Pooling



Featuretransformation

Schiebe einen „Filter“ über die Features und betrachte die „gefilterten“ Features

Betrachte den Bereich entsprechend der Filtergröße

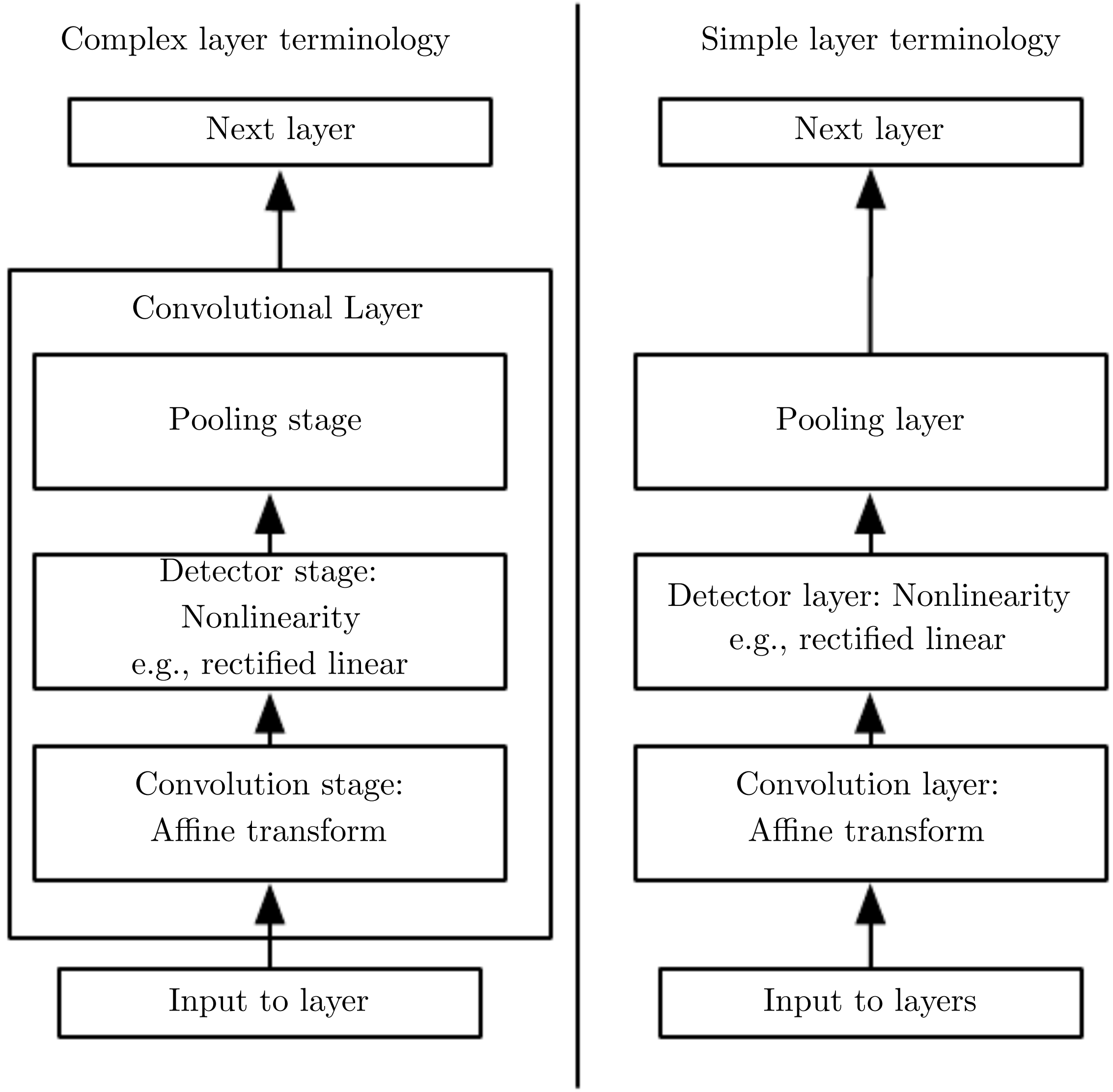
Max Pooling: Nimm maximalen Wert

Mean Pooling: Nimm Mittelwert

Featureraum wird kleiner

Keine trainierbaren Parameter!

Complex vs. Simple Layer Structure



[http://www.deeplearningbook.org/contents/convnets.html]

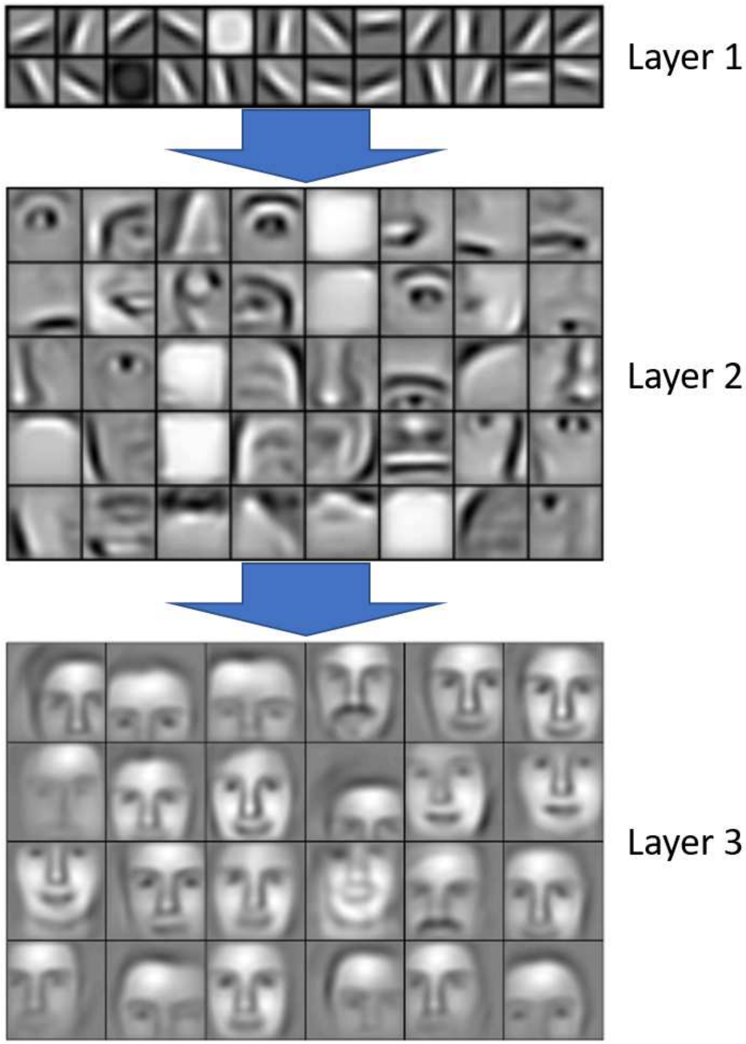
Strong Priors

- CNN = “fully connected net with an infinitely strong prior [on weights]”

- only useful when the assumptions made by the prior are reasonably accurate

- convolution+pooling can cause underfitting

Features learned by CNNs



Gut trainierte Netze haben klar erkennbare Features

Features werden in tieferen Schichten komplexer

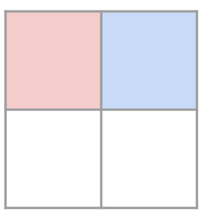
Layer 1:
Kantenzüge

Layer 2:
Augen, Nasen, Augenbrauen, Münder

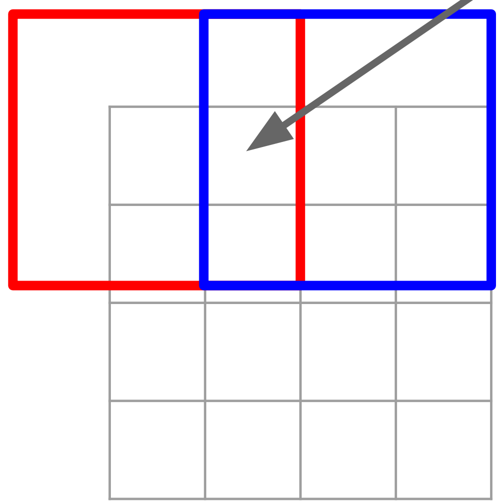
Layer 3:
(abgeschnittene) ganze Gesichter

Deconvolution

3x3 "deconvolution", stride 2, padding 1



Input gives weight for filter



Input: 2 x 2

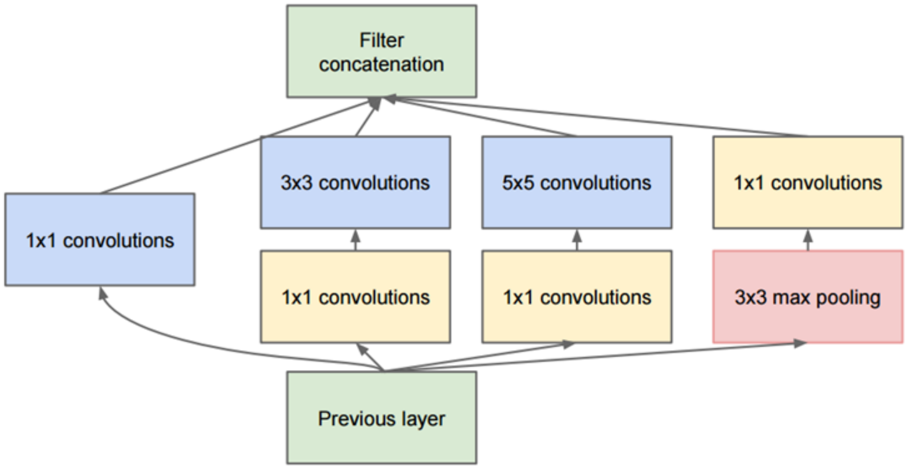
Output: 4 x 4

sum for overlapping output regions

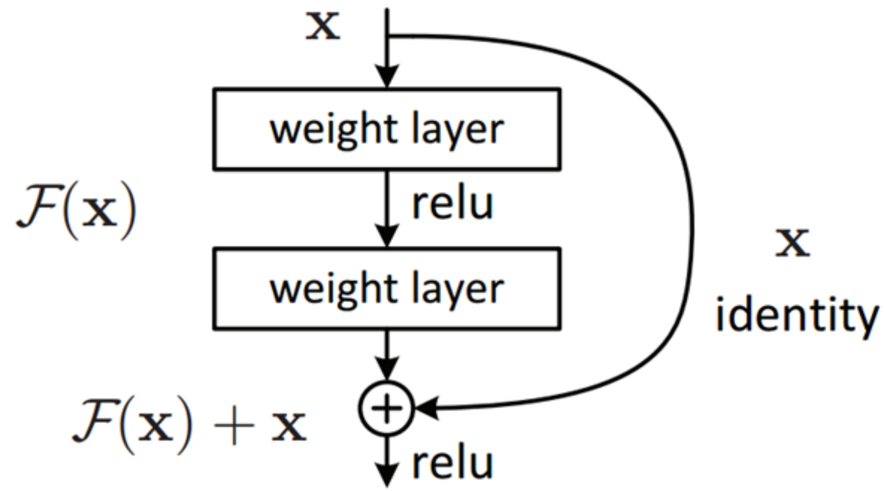
- same as backward pass for normal convolution
- better names:
  inverse of convolution
  convolution transpose
  fractional-stride conv.
  upconvolution

Outlook: Advanced CNN Building Blocks

Inception Module



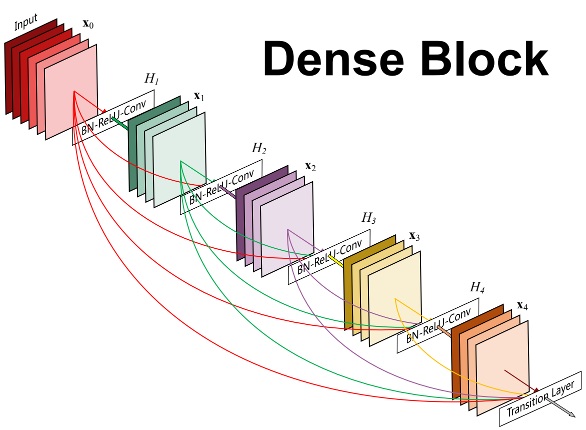
Residual Block



- 1x1 conv to reduce #channels
- multiple filter shapes / parallel computation paths
- <u>concatenation</u> of feature maps

- <u>addition</u> of learned residual

Dense Block



- skip connections
- <u>concatenation</u> of feature maps

U-Net