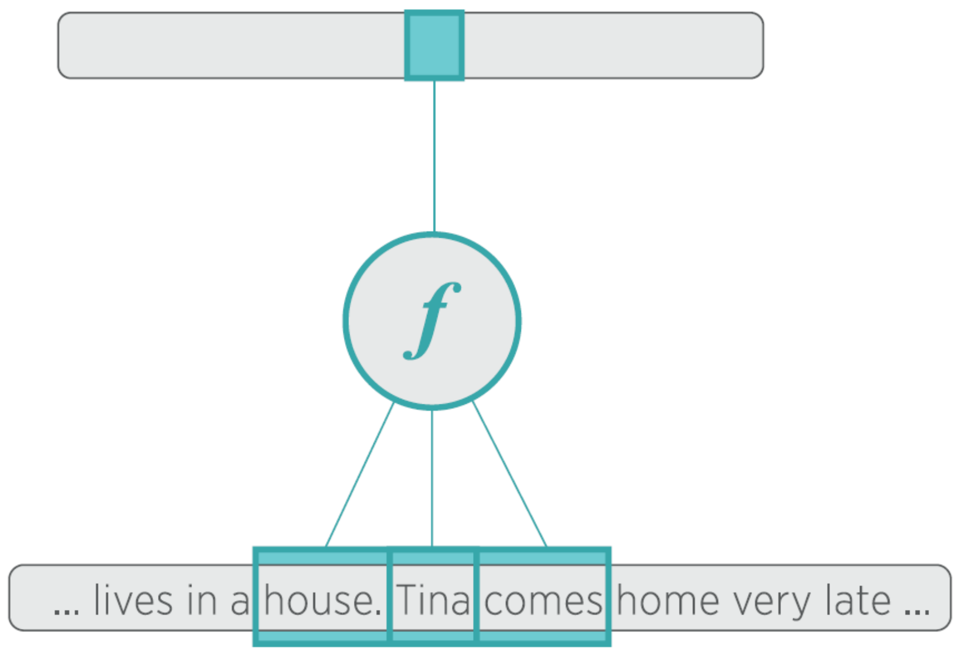
University of Ljubljana, Faculty of Computer and Information Science

Convolutional neural networks



Prof Dr Marko Robnik-Šikonja

Natural Language Processing, Edition 2022

Contents

- convolutional neural networks
- specifics of text

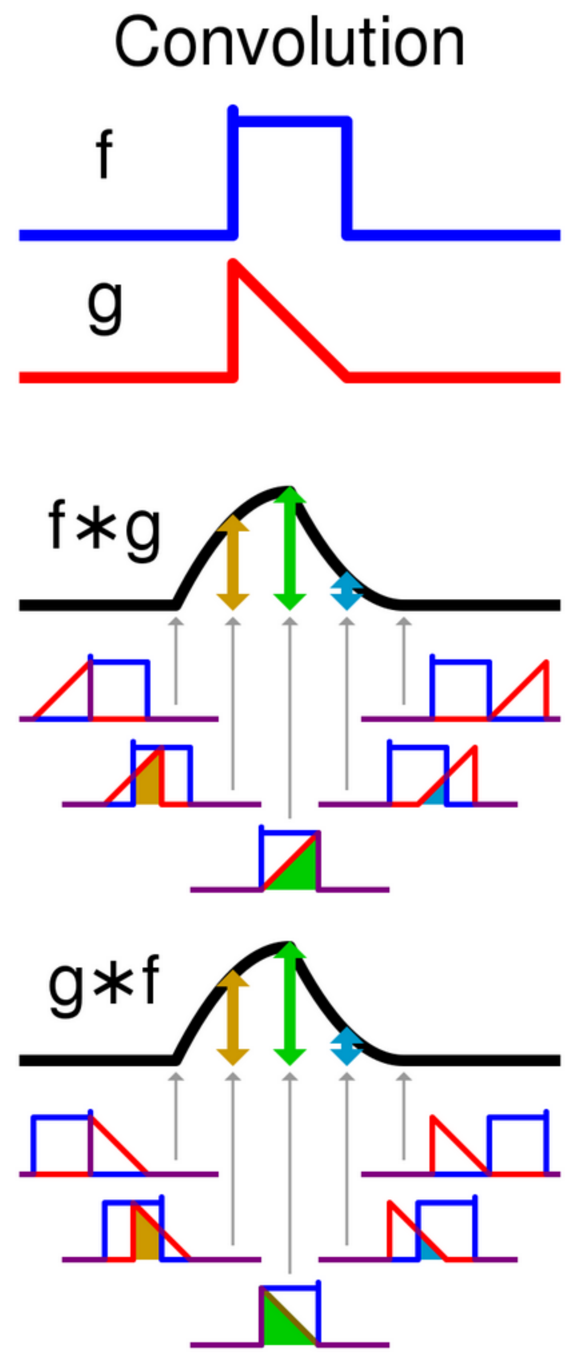
Convolution

- an operation on two functions (f and g) that produces a third function expressing how the shape of one is modified by the other.

$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau.$$

- for discrete functions

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m].$$

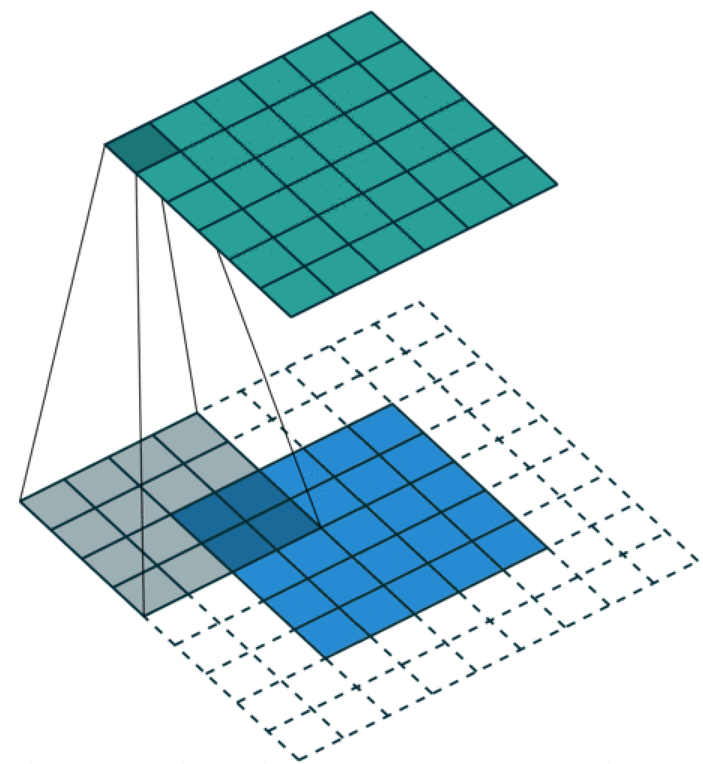


Convolutional Neural Network (CNN)

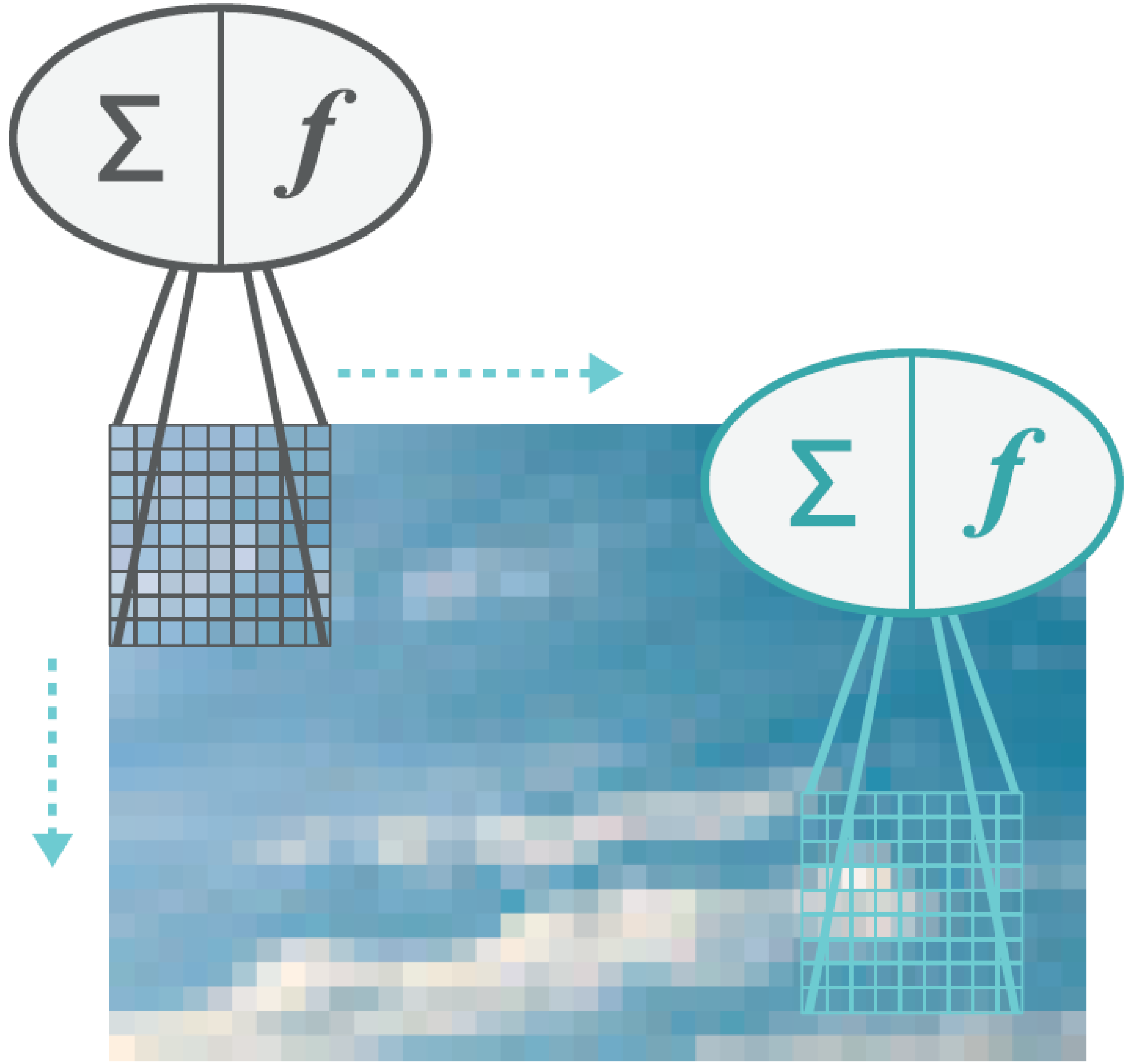
- Convolutional Neural Networks are inspired by mammalian visual cortex.
- The visual cortex contains a complex arrangement of cells, which are sensitive to small sub-regions of the visual field, called a receptive field. These cells act as local filters over the input space and are well-suited to exploit the strong spatially local correlation present in natural images.
- Two basic cell types:
    - Simple cells respond maximally to specific edge-like patterns within their receptive field.
    - Complex cells have larger receptive fields and are locally invariant to the exact position of the pattern.

Convolutional neural networks (CNN)

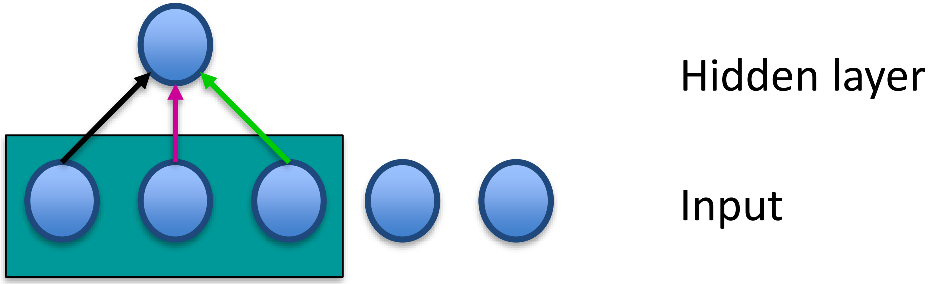
- currently, the most successful approach in image analysis, successful in language
- idea: many copies of small detectors used all over the image, recursively combined,
- detectors are learned, combination are learned



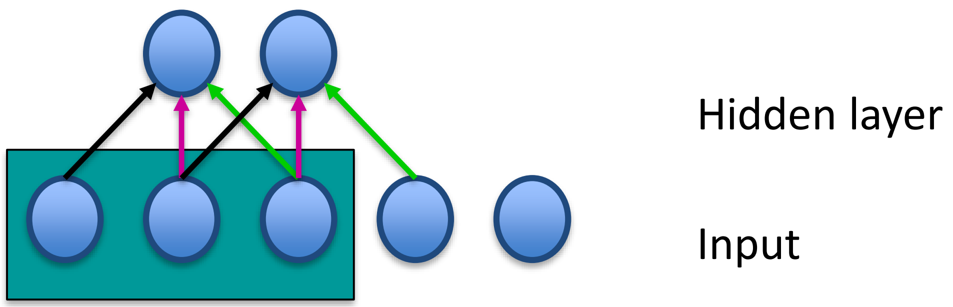
2d convolution for images



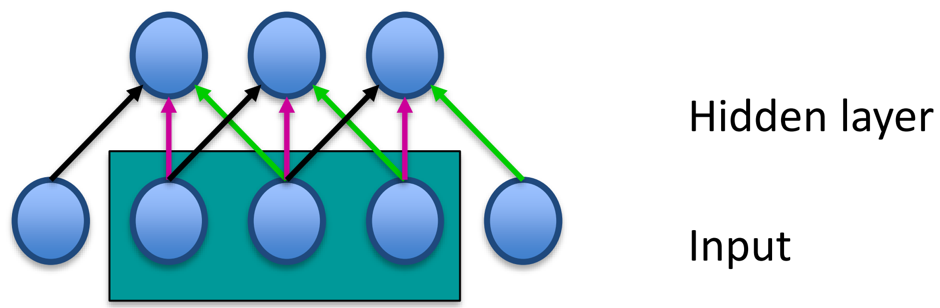
Basic Idea of CNNs



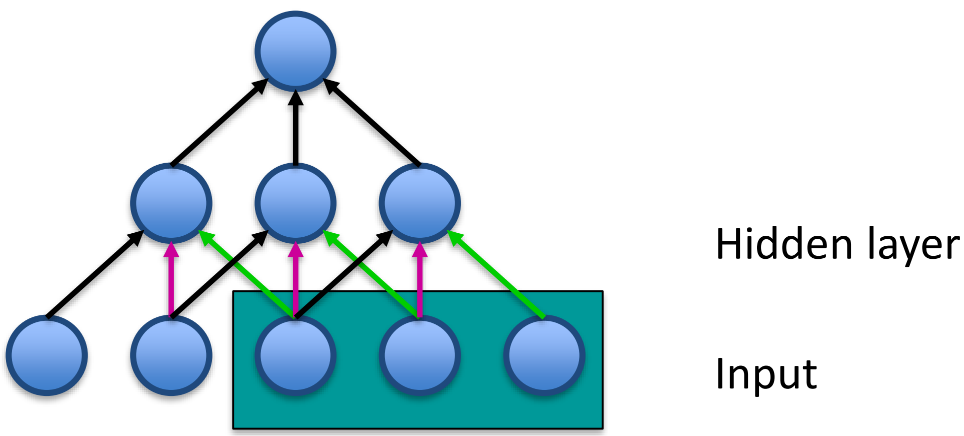
Basic Idea of CNNs



Basic Idea of CNNs

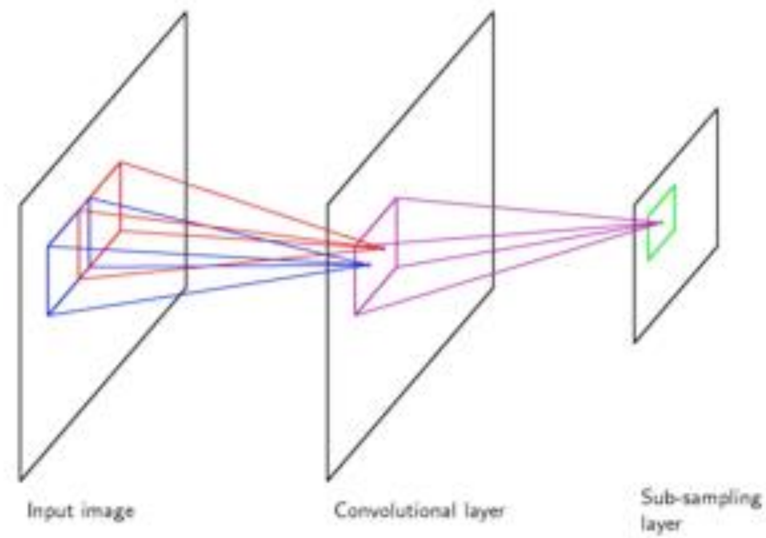


Basic Idea of CNNs



Convolutional Network

- The network is not fully connected.
- Different nodes are responsible for different regions of the image.
- This allows for robustness to transformations.
- Convolution is combined with subsampling.



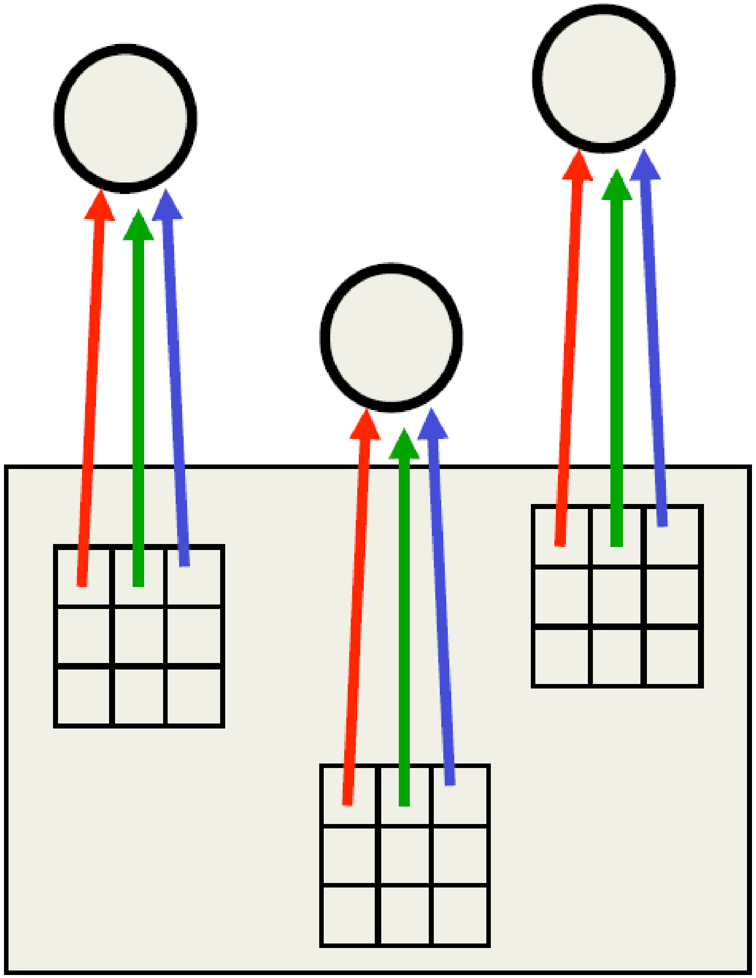
CNN Architecture: Convolutional Layer

- The core layer of CNNs
- The convolutional layer consists of a set of filters.
  - Each filter covers a spatially small portion of the input data.
- Each filter is convolved across the dimensions of the input data, producing a multidimensional feature map.
  - As we convolve the filter, we are computing the dot product between the parameters of the filter and the input.
- Intuition: the network will learn filters that activate when they see some specific type of feature at some spatial position in the input.
- The key architectural characteristics of the convolutional layer is local connectivity and shared weights.

Neural implementation of convolution

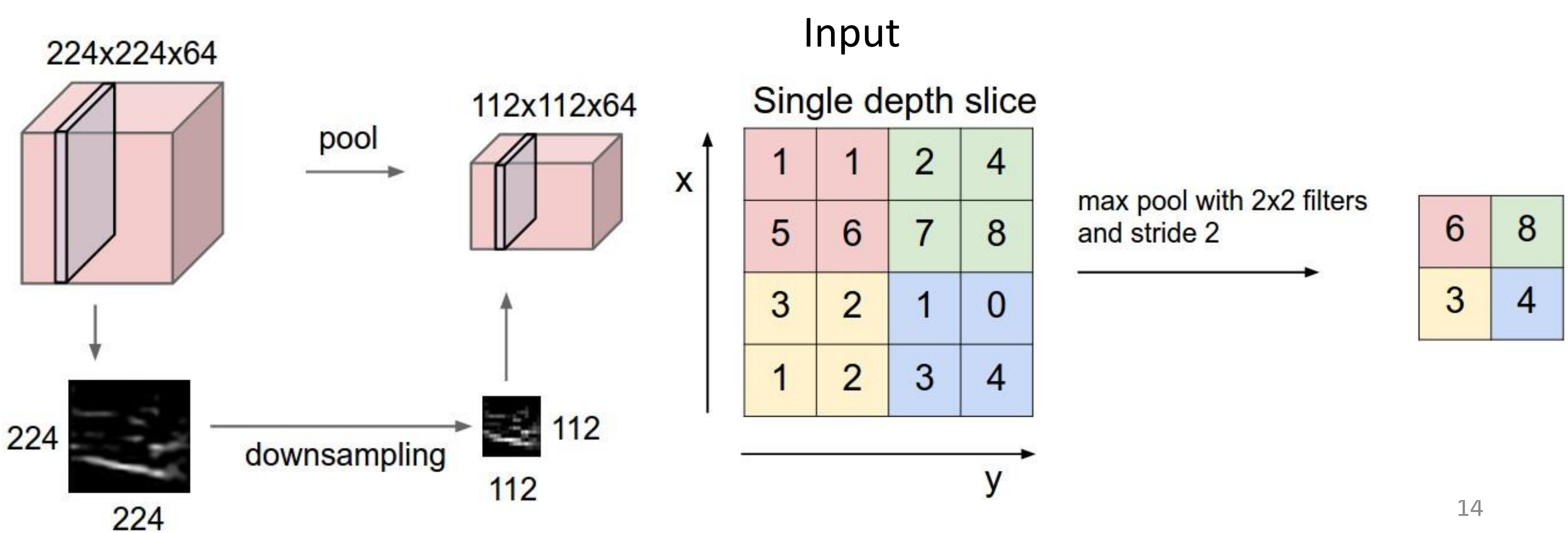
- weights of the same colors have equal weights
- adapted backpropagation

- images: 2d convolution
- languages: 1d convolution



CNN Architecture: Pooling Layer

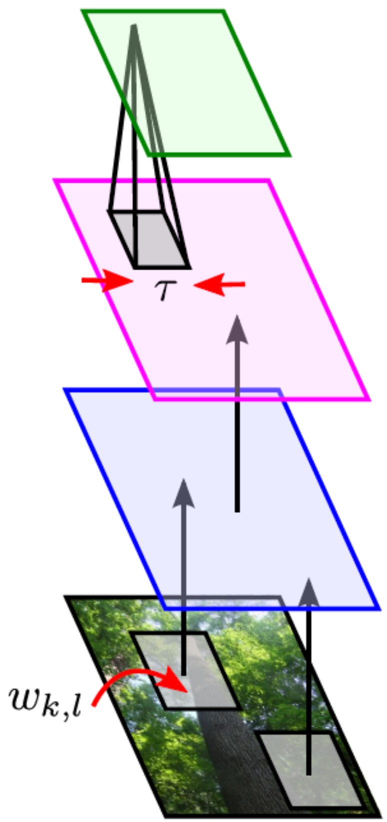
- Intuition: to progressively reduce the spatial size of the representation, to reduce the amount of parameters and computation in the network, and hence to also control overfitting
- Pooling partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum value of the features in that region.



CNN: pooling

- reduces the number of connections to the next layer (prevents the excessive number of parameters, speeds-up learning, reduces overfitting)
- max-pooling, min-pooling, average-pooling
- the problem: after several layers of pooling we lose the information about the exact location of the recognized pattern and about spatial relations between different patterns and features, e.g., a nose on a forehead

Building-blocks for CNN's



$$x_{i,j} = \max_{|k|<\tau, |l|<\tau} y_{i-k,j-l}$$

mean or subsample also used

pooling stage

Feature maps of a larger region are combined.

$$y_{i,j} = f(a_{i,j})$$

e.g. $f(a) = [a]_+$

$f(a) = \mathrm{sigmoid}(a)$

non-linear stage

Feature maps are trained with neurons.

$$a_{i,j} = \sum_{k,l} w_{k,l} z_{i-k,j-l}$$

Shared weights

convolutional stage

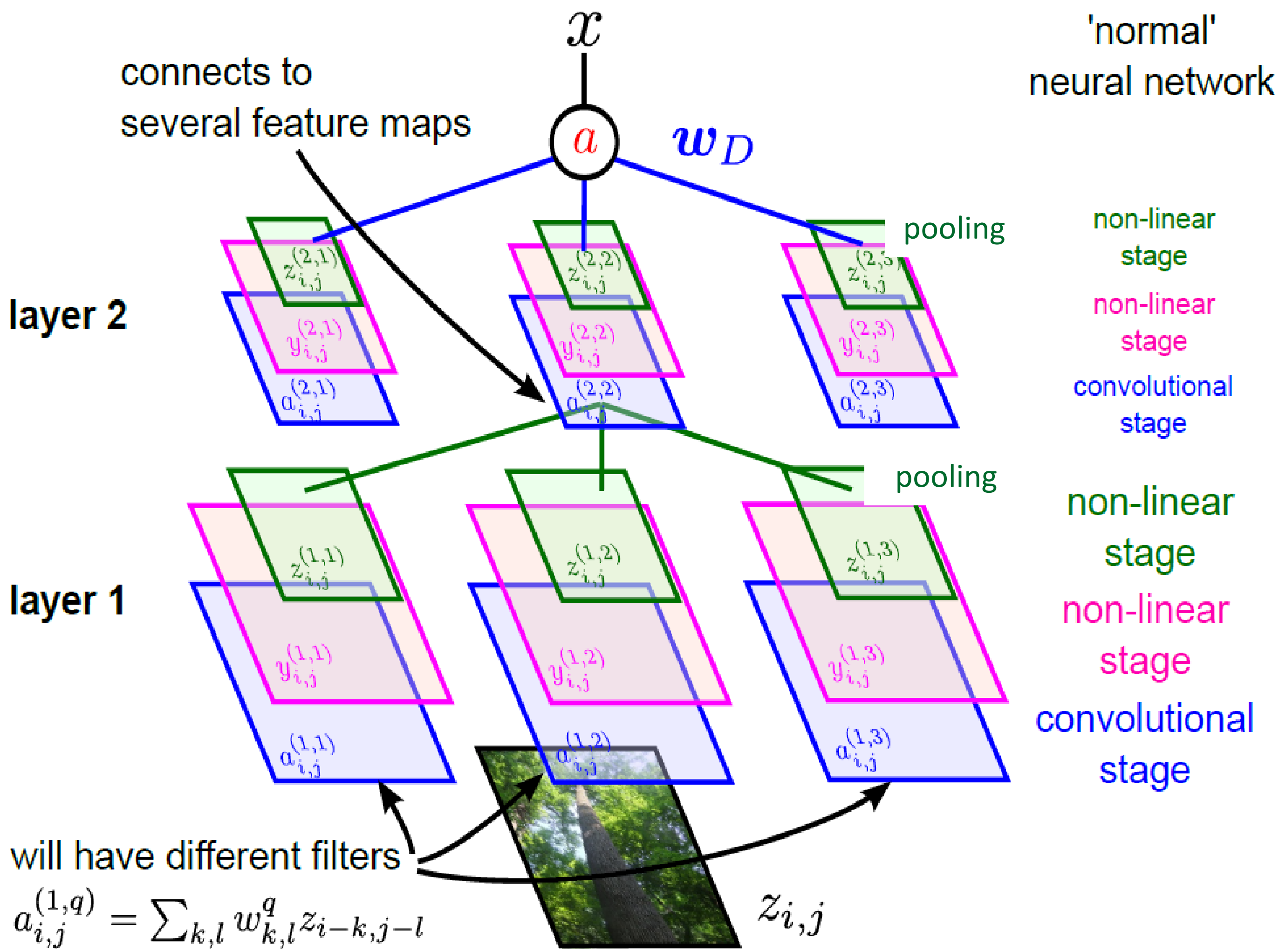
Each sub-region yields a feature map, representing its feature.

$z_{i,j}$

Images are segmented into sub-regions.

input image

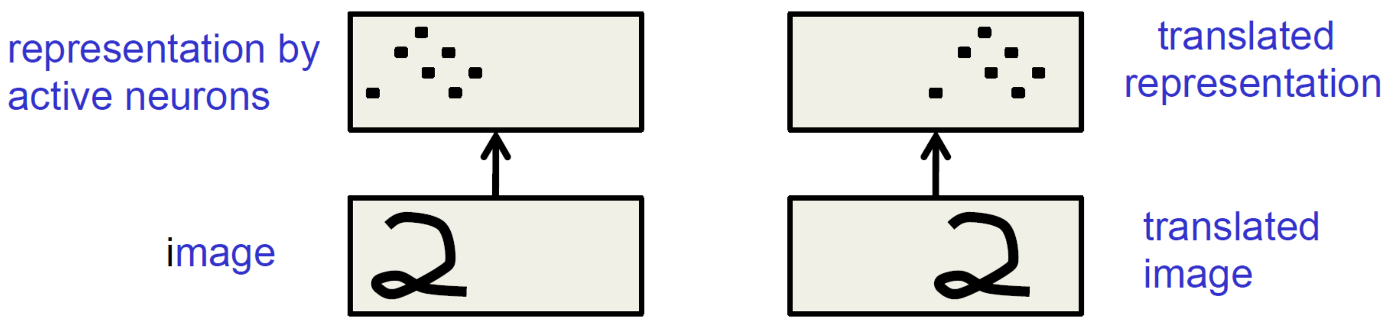
Full CNN



x
'normal'
neural network
connects to
several feature maps
a
w_D
pooling
non-linear
stage
non-linear
stage
convolutional
stage
layer 2
$z_{i,j}^{(2,1)}$
$y_{i,j}^{(2,1)}$
$a_{i,j}^{(2,1)}$
$z_{i,j}^{(2,2)}$
$y_{i,j}^{(2,2)}$
$a_{i,j}^{(2,2)}$
$y_{i,j}^{(2,3)}$
$a_{i,j}^{(2,3)}$
pooling
non-linear
stage
non-linear
stage
convolutional
stage
layer 1
$z_{i,j}^{(1,1)}$
$y_{i,j}^{(1,1)}$
$a_{i,j}^{(1,1)}$
$z_{i,j}^{(1,2)}$
$y_{i,j}^{(1,2)}$
$a_{i,j}^{(1,2)}$
$z_{i,j}^{(1,3)}$
$y_{i,j}^{(1,3)}$
$a_{i,j}^{(1,3)}$
will have different filters
$a_{i,j}^{(1,q)} = \sum_{k,l} w_{k,l}^{q} z_{i-k,j-l}$
$z_{i,j}$

Convolutional networks: illustration on image recognition

- a useful feature is learned and used on several positions
- prevents dimension hopping
- max-pooling

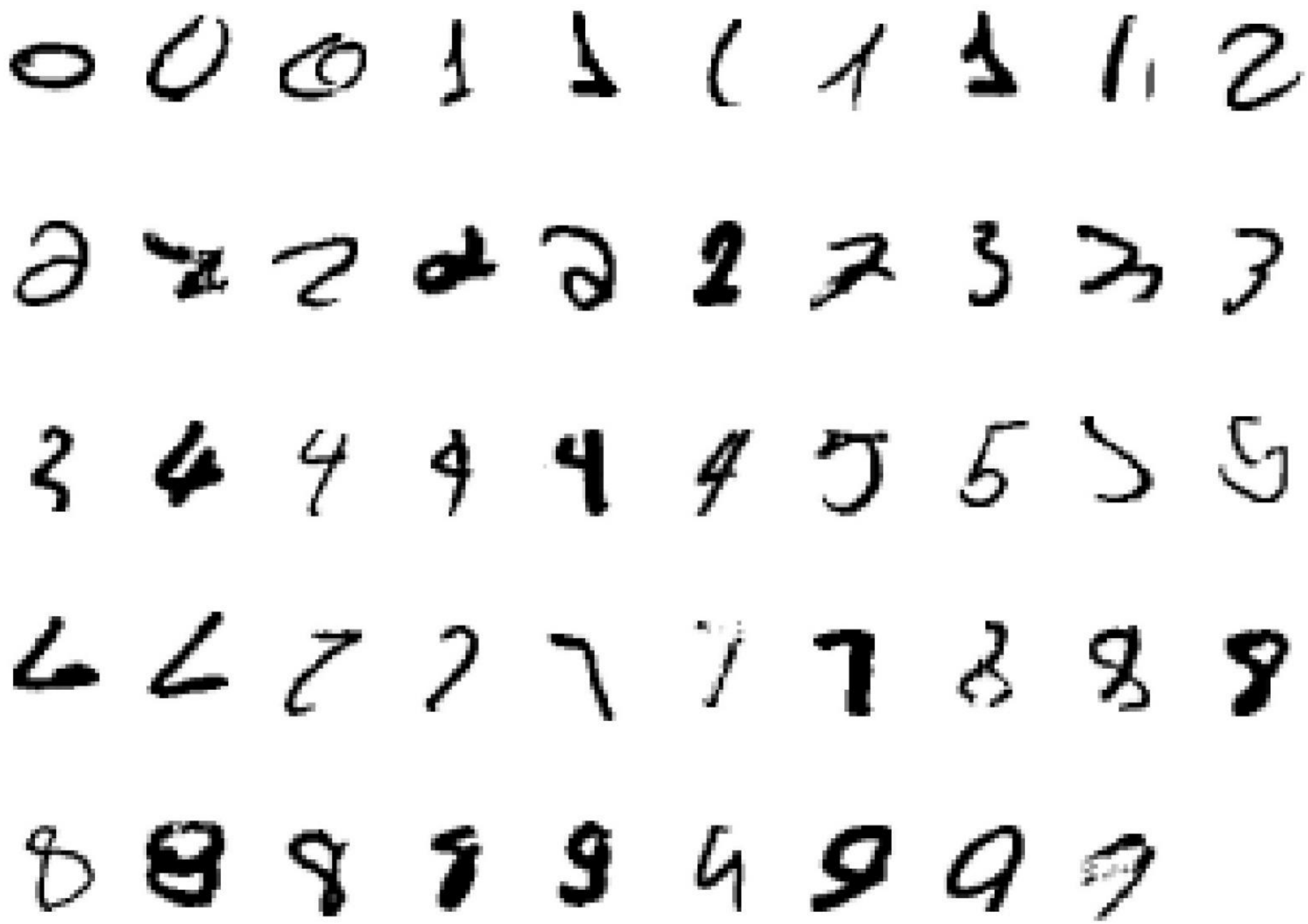


CNN success: LeNet

- handwritten digit recognition
- Yann LeCun, demo http://yann.lecun.com
- several hidden layers
- several convolutional filters
- pooling
- several other tricks

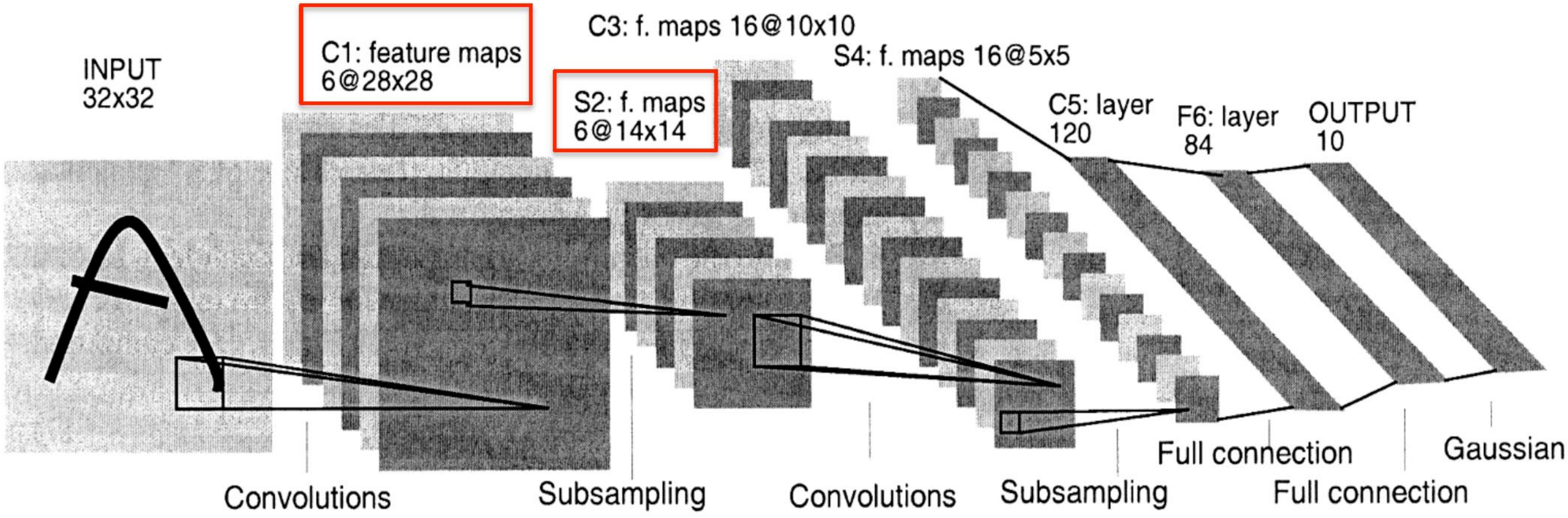
Hand-written Digit Recognition

- Input:



LeNet5 architecture

- handwritten digit recognition



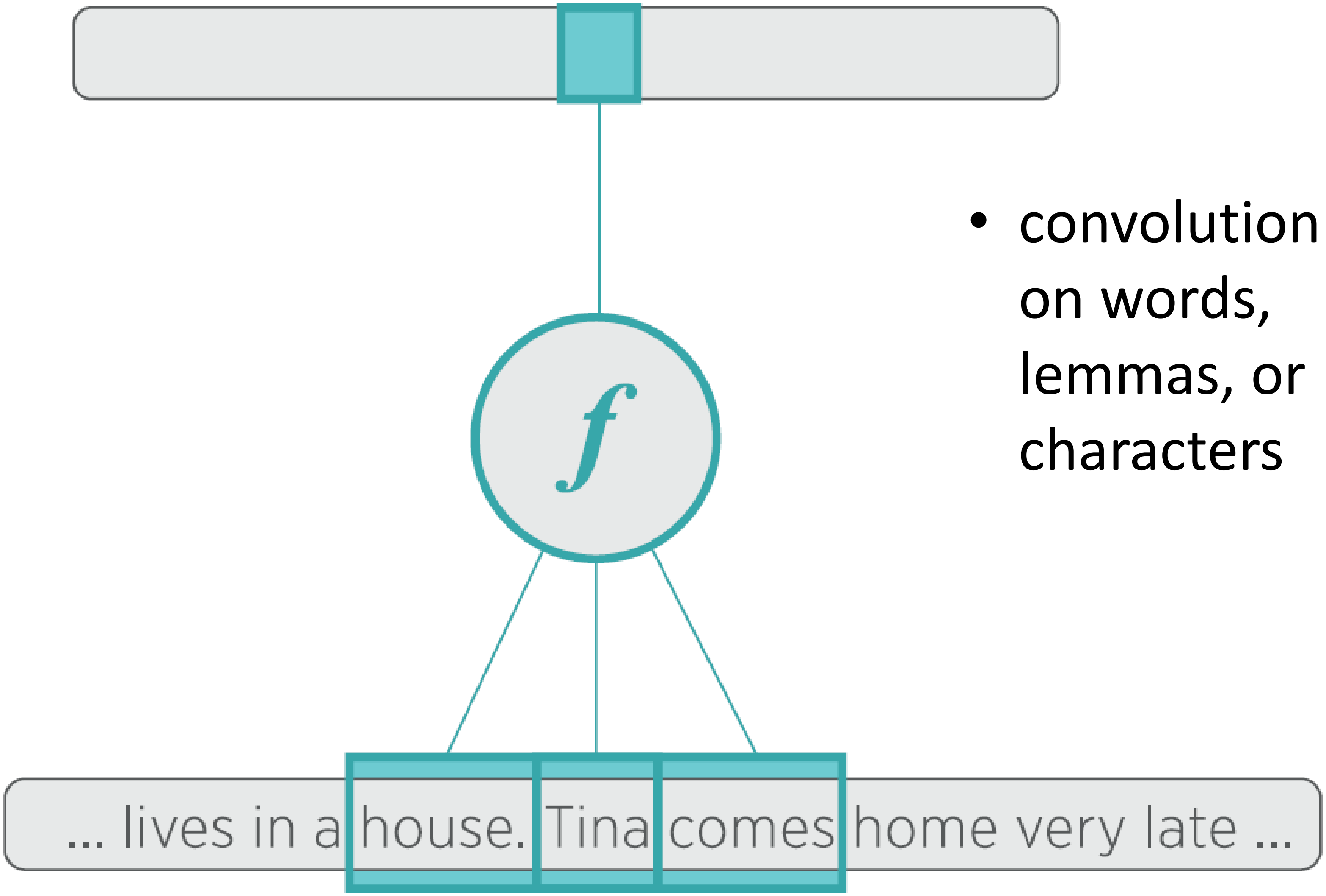
Errors of LeNet5

- 80 errors in 10,000 test cases

Benefits of CNNs

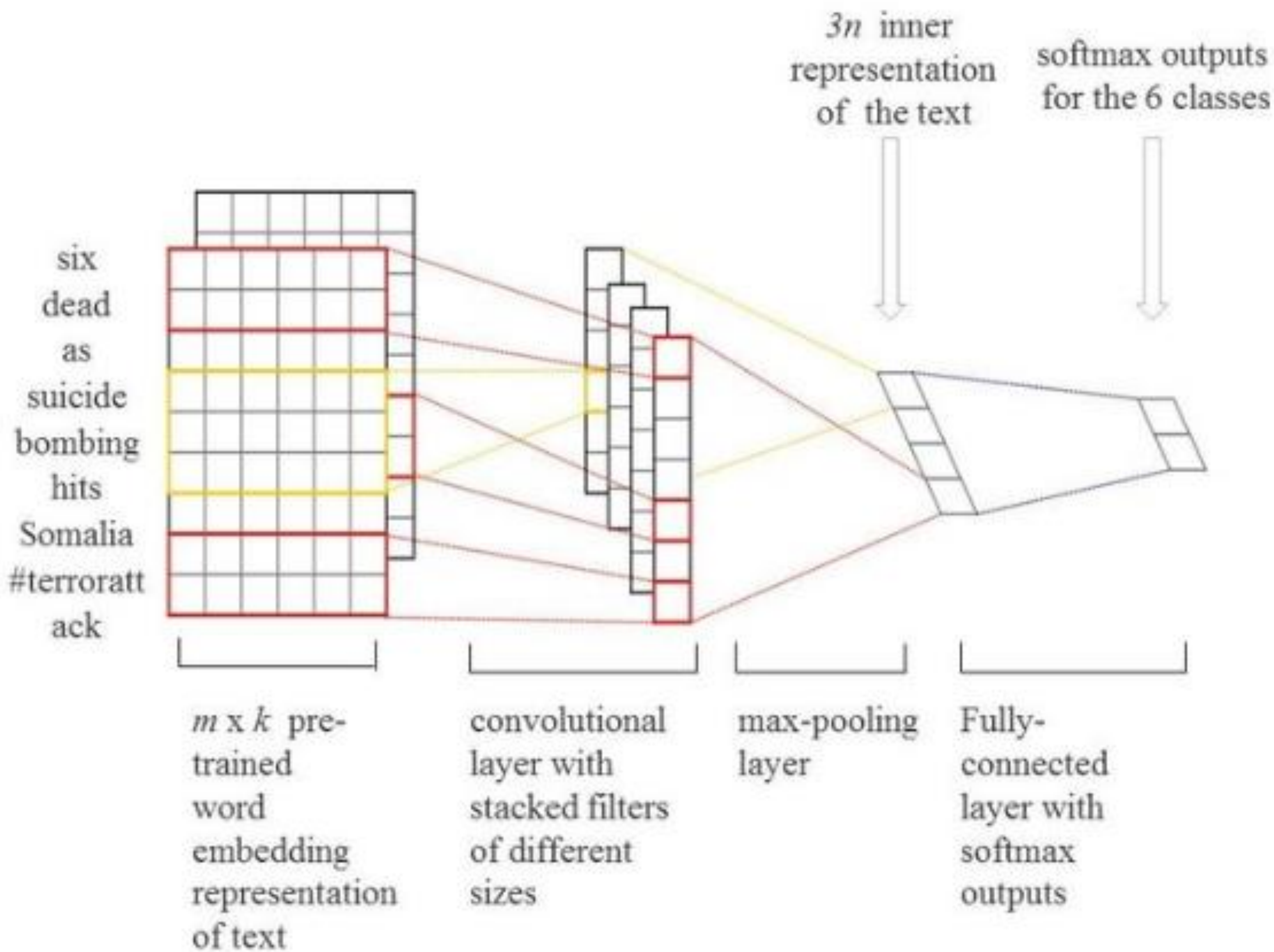
- The number of weights can be much less than for a fully connected layer.
- The small number of weights can use different parts of the image as training data. Thus, we have several orders of magnitude more data to train the fewer number of weights.
- We get translation invariance for free.
- Fewer parameters take less memory and thus all the computations can be carried out in memory in a GPU or across multiple processors.

1d convolution for text



- convolution on words, lemmas, or characters

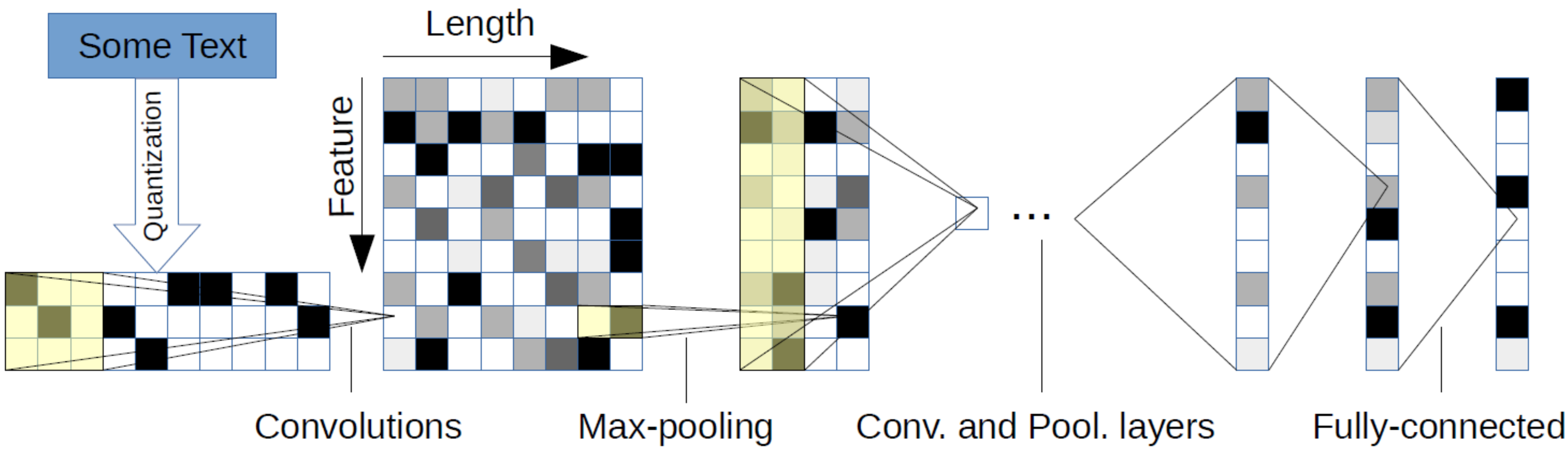
1d CNN architecture



2d convolution on characters

- text classification
- Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." Advances in Neural Information Processing Systems. 2015.
- convolution, max-pooling
- ReLU activation h(x) = max(0, x)
- backpropagation with momentum 0.9
- minibatch =128, starting step 0.01 is halved every 3 epoch
- character quantization, alphabet of 70 characters
  abcdefghijklmnopqrstuvwxyz0123456789
  -,;.!?:’’’/\|_@#$%ˆ&*˜‘+-=<>()[]{}
- one-hot encoding of characters

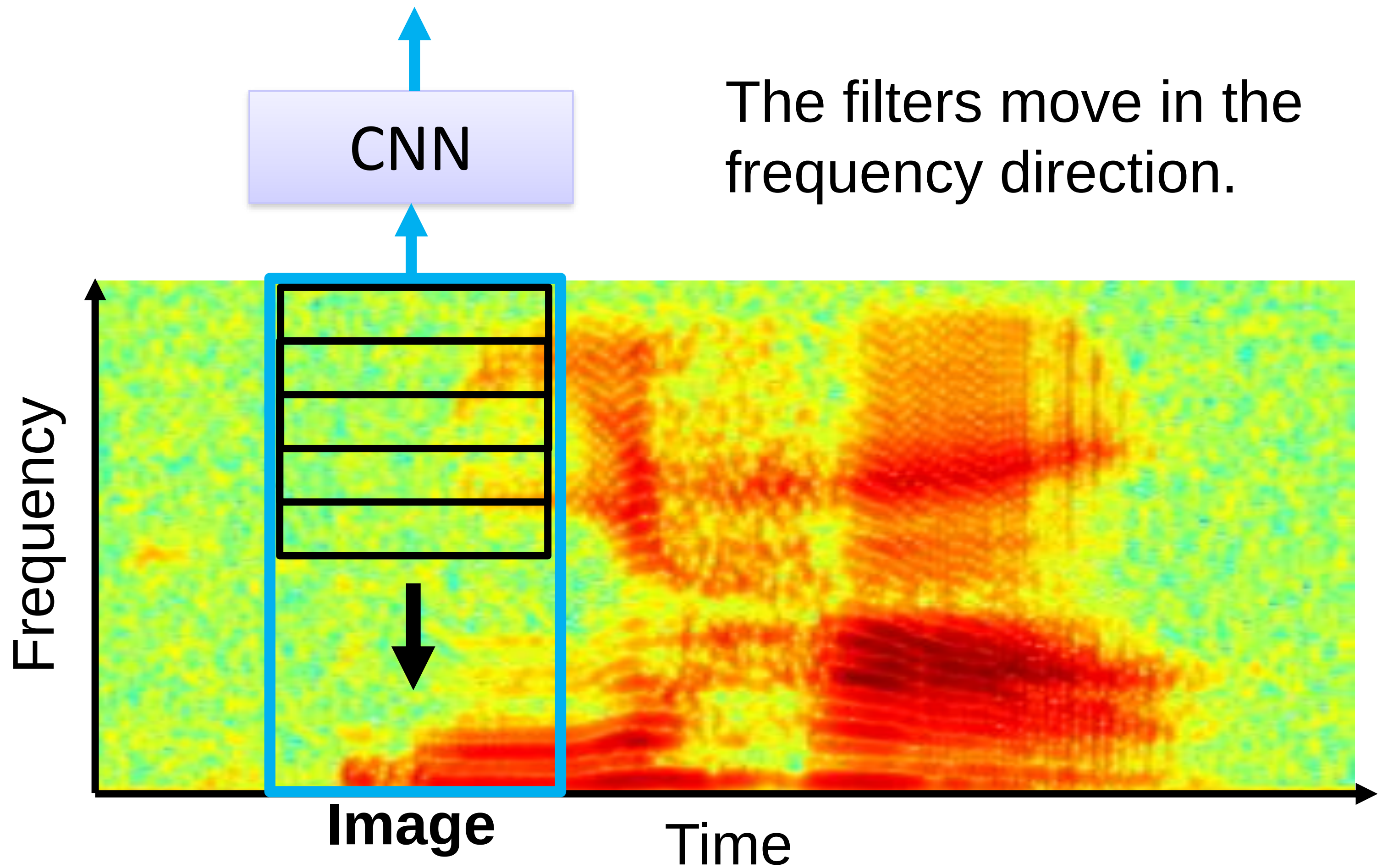
2d text convolution network architecture



- blocks of 1014 characters (i.e. 1014 x 70)
- 6 convolutional layers with filter lengths 7, 7, 3, 3, 3, 3
- stride = 1
- between fully connected layers: dropout with p=0.5
- good results on very large datasets $>10^6$
- for smaller datasets bag of n-grams is very competitive

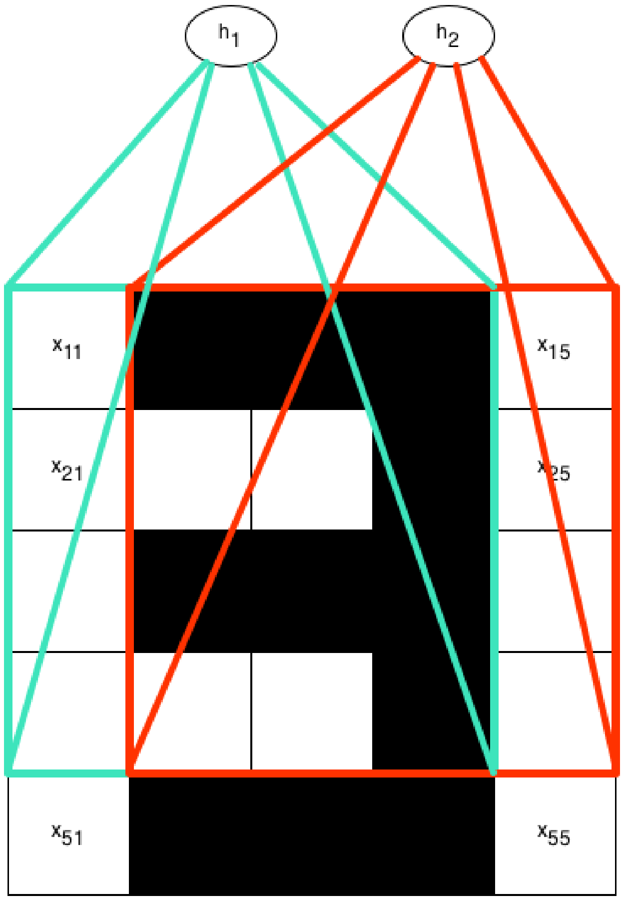
CNN in speech recognition

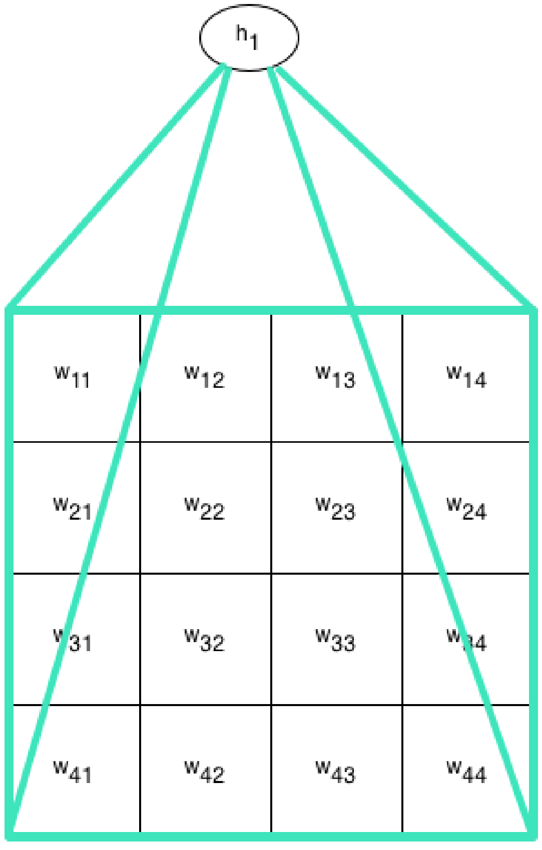
The filters move in the frequency direction.



Spectrogram

Example: what the following CNN returns





$w_{11} = 1$	$w_{12} = 1$	$w_{13} = 1$	$w_{14} = 0$
$w_{21} = 0$	$w_{22} = 0$	$w_{23} = 1$	$w_{24} = 0$
$w_{31} = 1$	$w_{32} = 1$	$w_{33} = 1$	$w_{34} = 0$
$w_{41} = 0$	$w_{42} = 0$	$w_{43} = 1$	$w_{44} = 0$