Convolutional Neural Nets

Thang Vu

04.12.2025

Questionnaire

- Please go to Ilias
- Open the questionnaire ‚NN Basics'
- Note that the questionnaire is anonymous, meaning we only receive the final statistics and responses, not the identities of the individuals who submitted them.

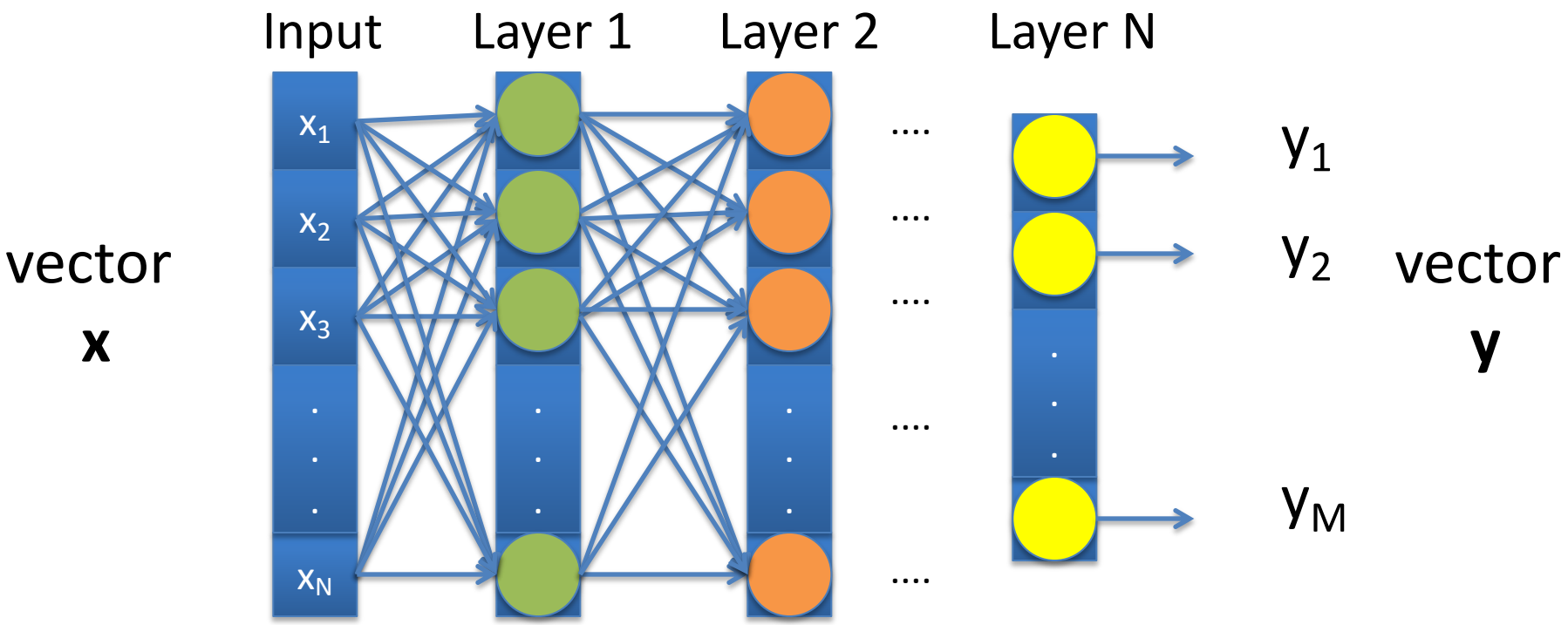
Outline

- Motivation
- Convolutional Neural Nets (CNN)
- Training
- Applications

Outline

- Motivation
- Convolutional Neural Nets (CNN)
- Training
- Applications

Feed-forward Neural Nets



Challenges

- Challenge 1: What if the input is a matrix?

0	1	0	0	1
2	3	3	3	4
4	0	1	0	1
3	4	2	3	3

Challenges

- Challenge 1: What if the input is a matrix?

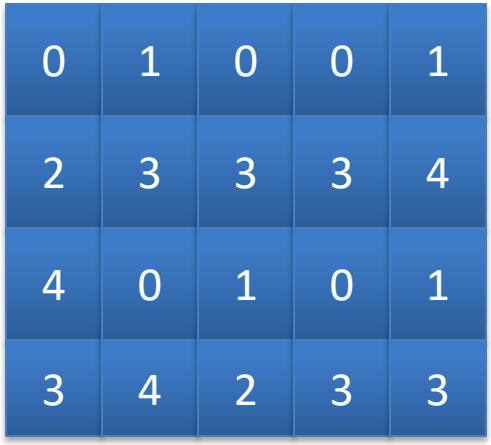
0	1	0	0	1
2	3	3	3	4
4	0	1	0	1
3	4	2	3	3

0
2
4
3
1
3
0
4

…

Challenges

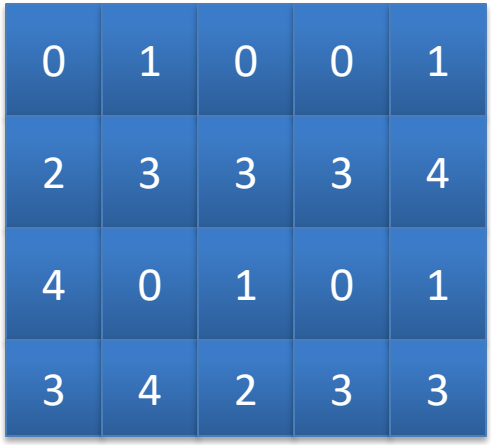
- Challenge 1: What if the input is a matrix?



- #parameters ↗

Challenges

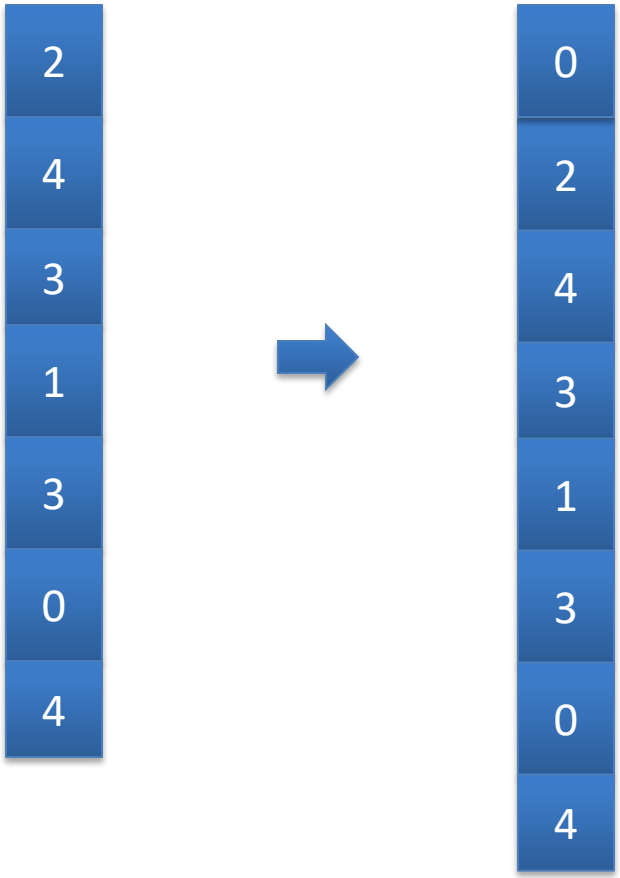
- Challenge 1: What if the input is a matrix?



- Don‘t want to blow up the number of parameters
  - If possible, reduce the number of parameters

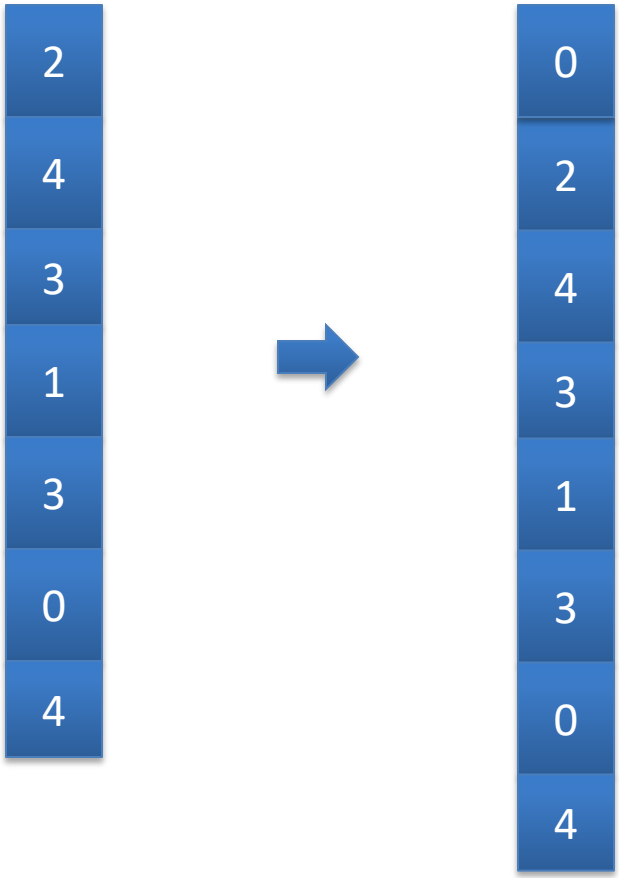
Challenges

- Challenge 2: What if the input is shifted?



Challenges

- Challenge 2: What if the input is shifted?

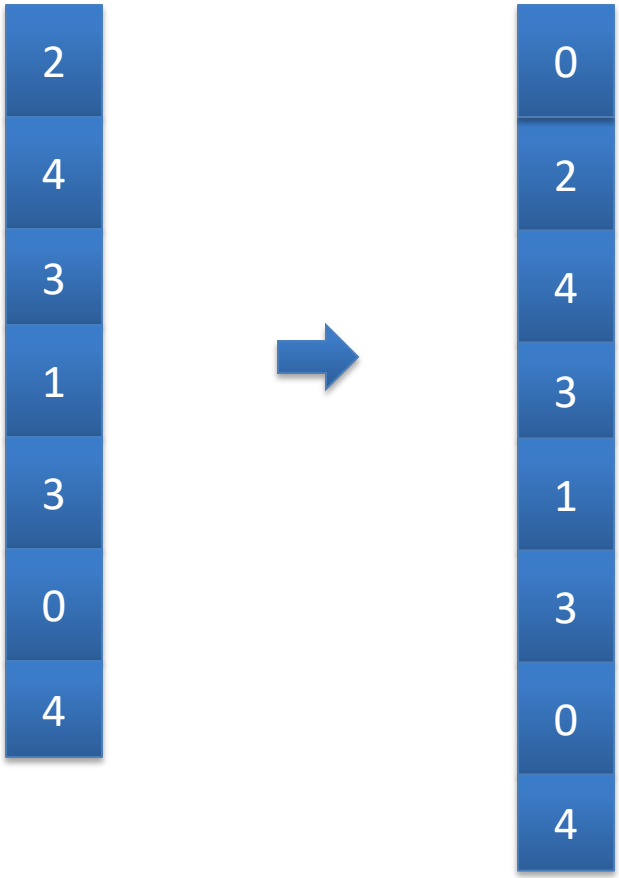


Real-world example 1:

- Language processing:
  - Input 1: „I am happy today“
  - Input 2: „Today I am happy“

Challenges

- Challenge 2: What if the input is shifted?

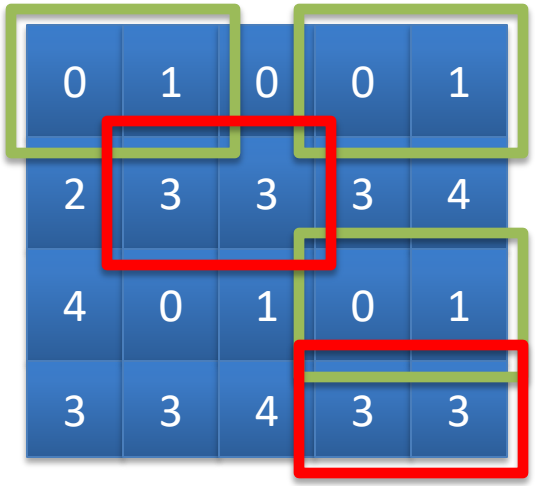


Real-world example 2:

- Computer vision:

Challenges

- Challenge 2: What if the input is shifted?



- Wish to recognize patterns
  - Position independent

Convolutional neural nets

- Proposed by LeCun et al. 1990
  - Hand-written digit recognition with a back-propagation network
- One version of this network was proposed by Waibel in 1989
  - Phoneme recognition using time-delay neural networks
- Image recognition with Convolution Neural Nets has won all the competitions
- Also used in speech recognition and natural language processing

Convolutional neural nets

- The first CNN which won the ImageNet Large Scale Visuale Recognition Challenge (ILSVRC) 2012

Model	Top-1 (%)	Top-5 (%)
Sparse Coding	47.1	28.2
SIFT + FVs	45.7	25.7
CNN	37.5	17.0

Results on ILSVRC-2010 test data

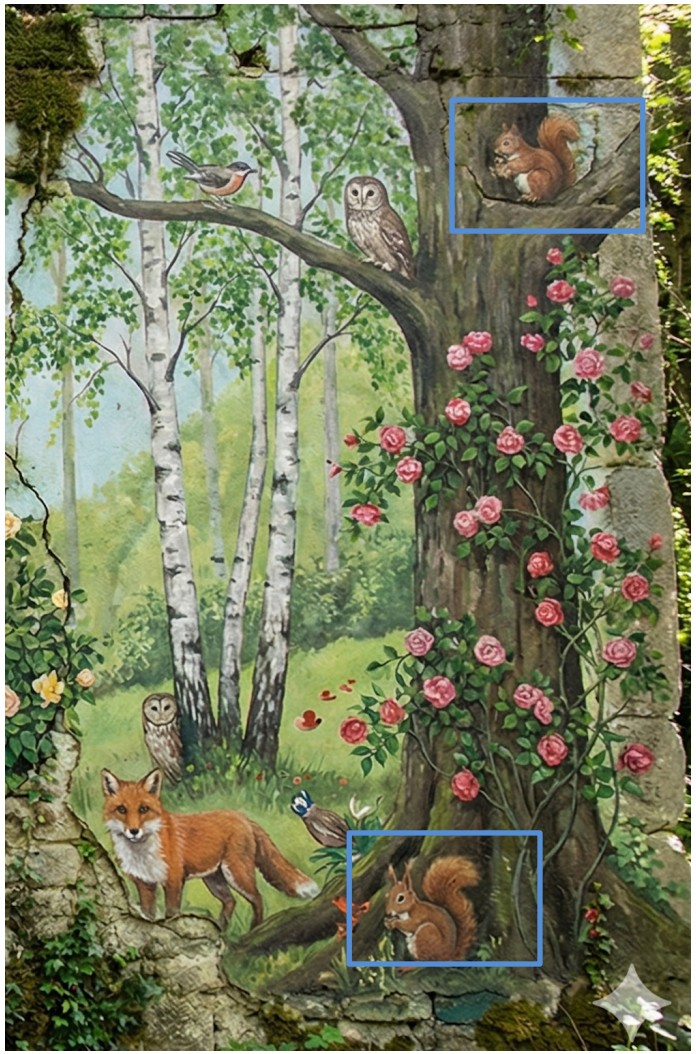
Model	Top-5 (%)
Sparse Coding	26.2
5 CNNs	16.4
7 CNNs	15.3

Results on ILSVRC-2012 test data

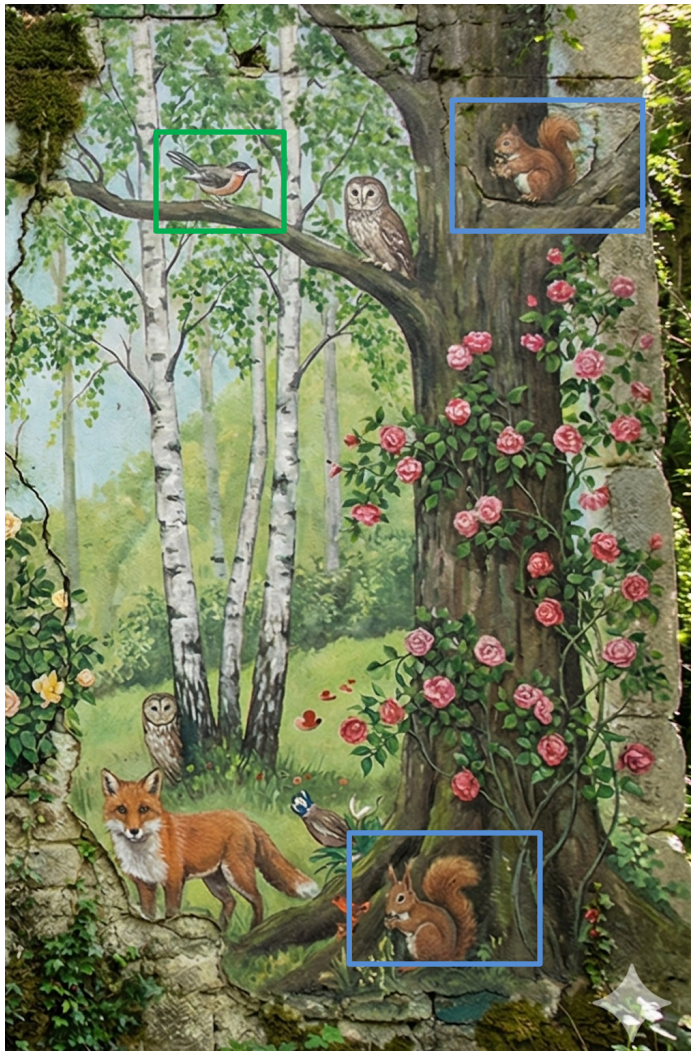
- *Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)*

Getting the Intuition

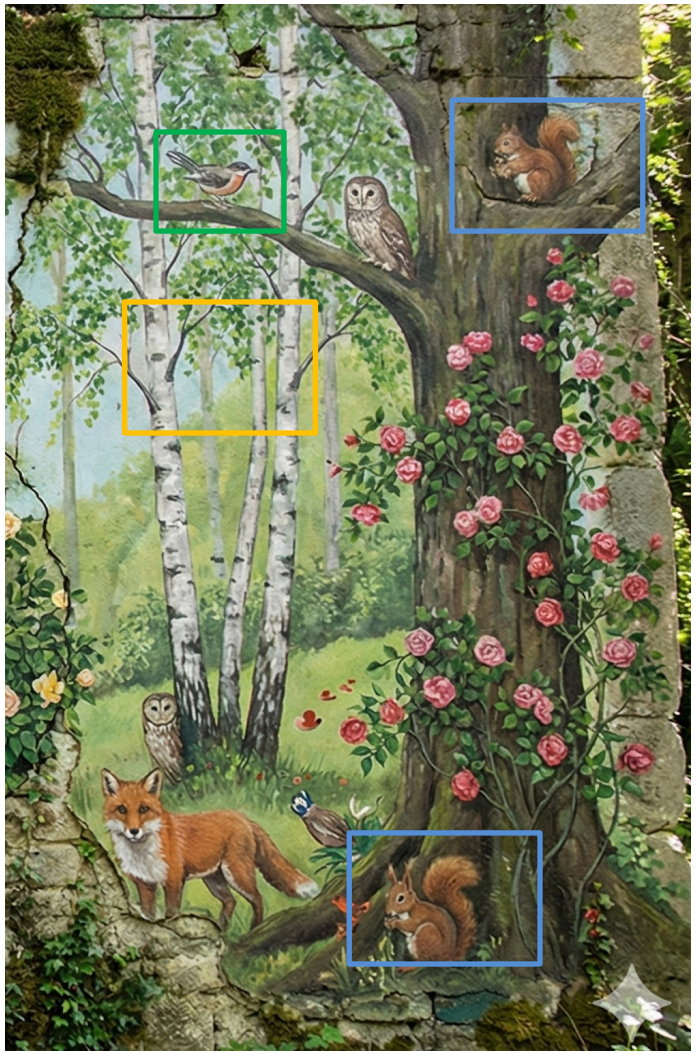
Getting the Intuition



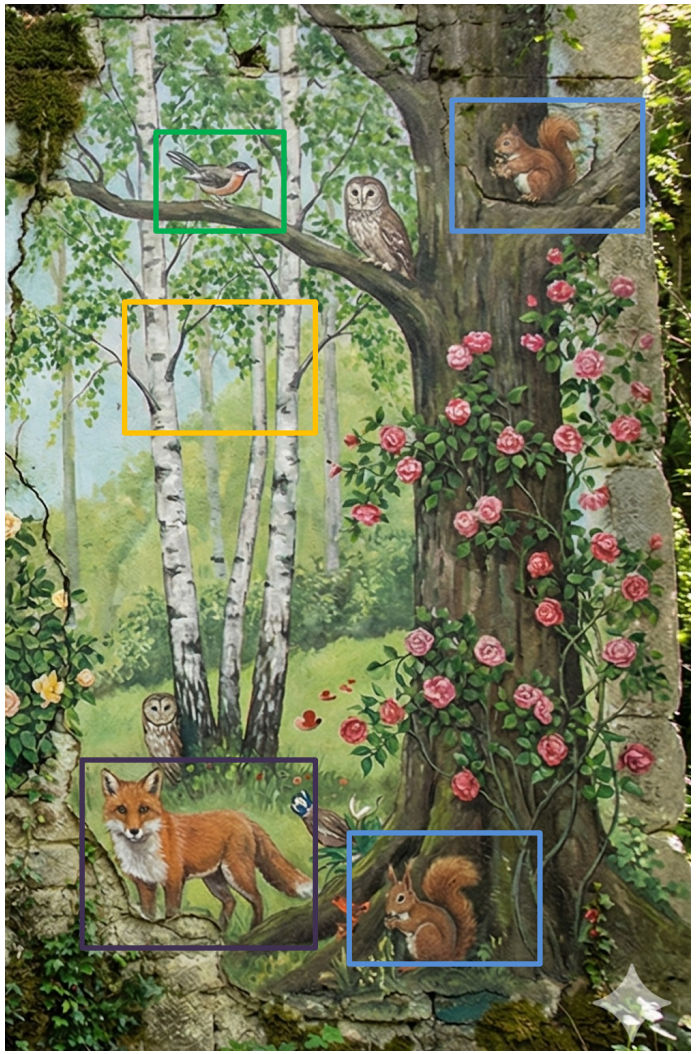
Getting the Intuition



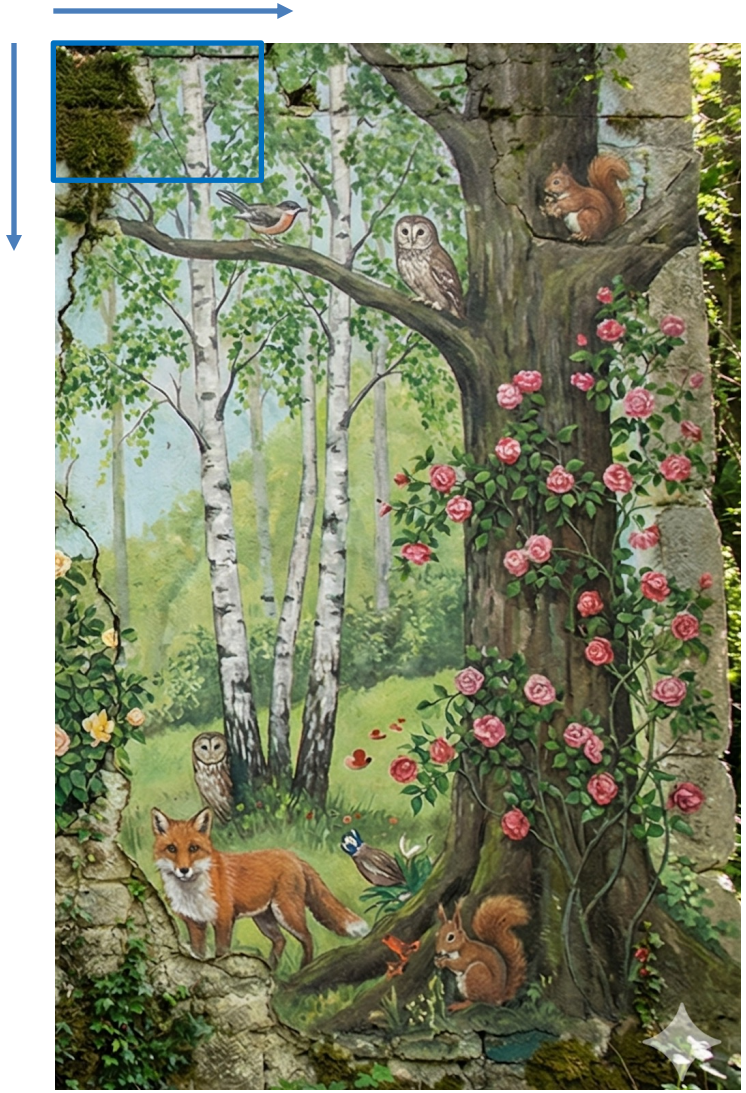
Getting the Intuition



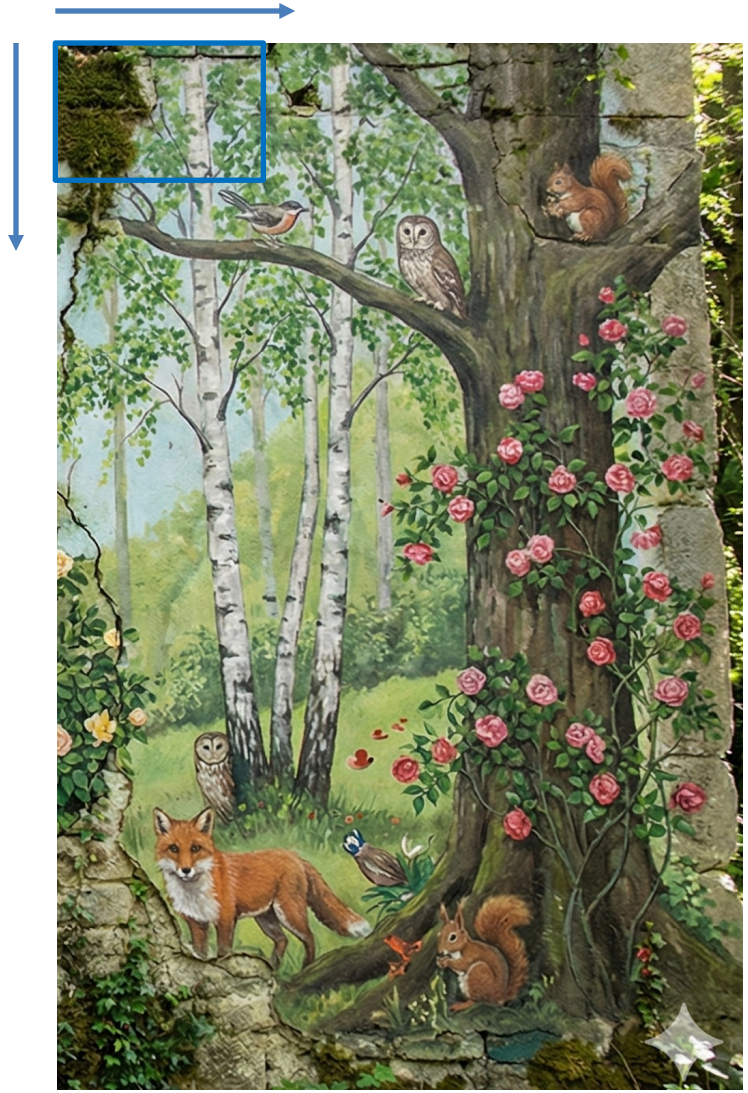
Getting the Intuition



Getting the Intuition



Getting the Intuition

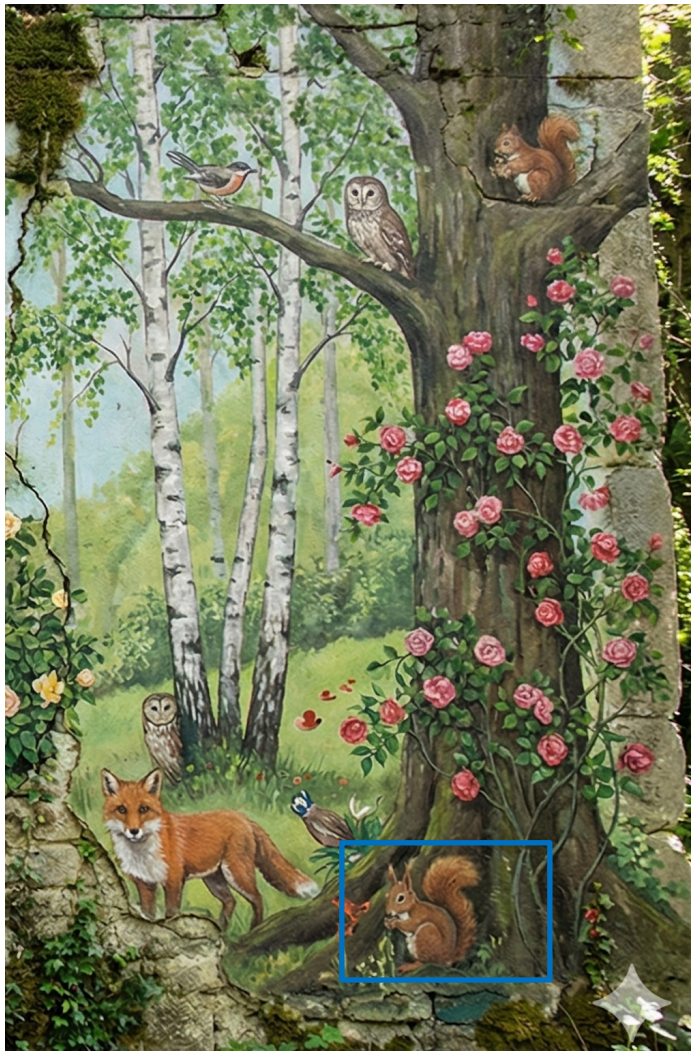


No squirrel found!

Getting the Intuition

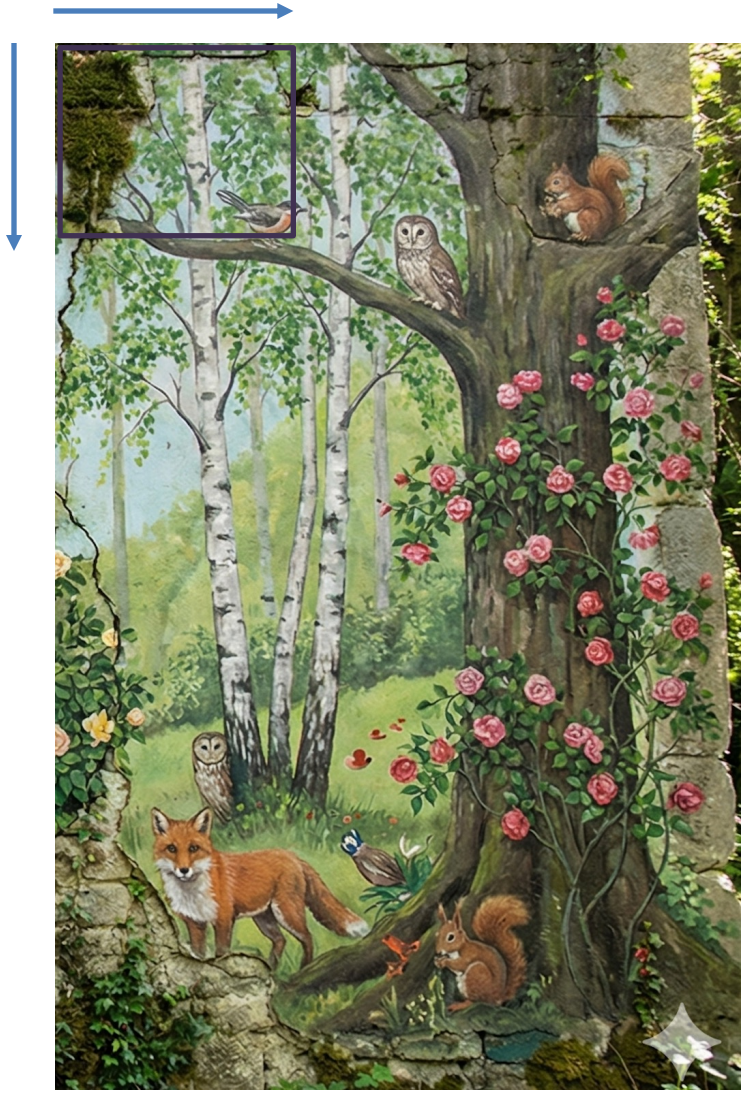
Squirrel found!

Getting the Intuition



Squirrel found!

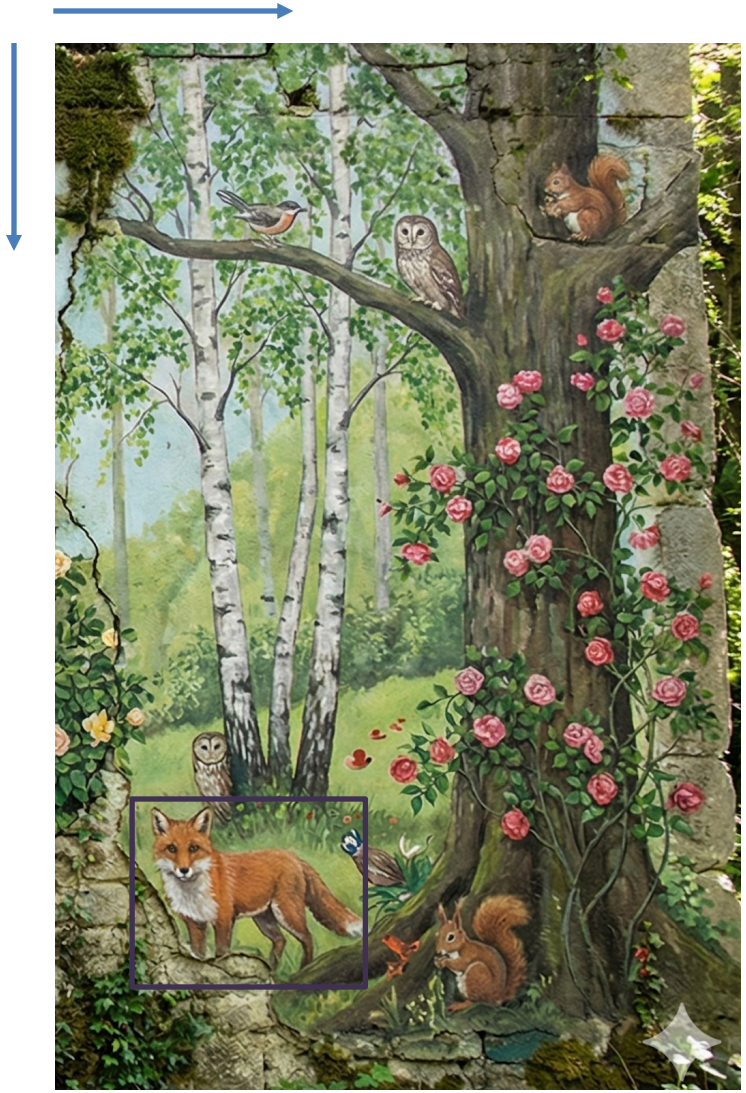
Getting the Intuition



Getting the Intuition

No fox found!

Getting the Intuition



Fox found!

Getting the Intuition

- After the scanning process, we find information:

[No squirrel found, no squirrel found, … **squirrel found**, no squirrel found, …., **squirrel found**, ….]

Filtering out irrelevant information

[squirrel found, squirrel found]

Getting the Intuition

- After the scanning process, we find information:

[No fox found, no squirrel found, … **fox found**, ….]

Filtering out irrelevant information

[fox found]

Getting the Intuition

- After filtering, we merge relevant information

[squirrel found, squirrel found]

[fox found]

Getting the Intuition

- After filtering, we merge relevant information

[**squirrel found, squirrel found**]

[**fox found**]

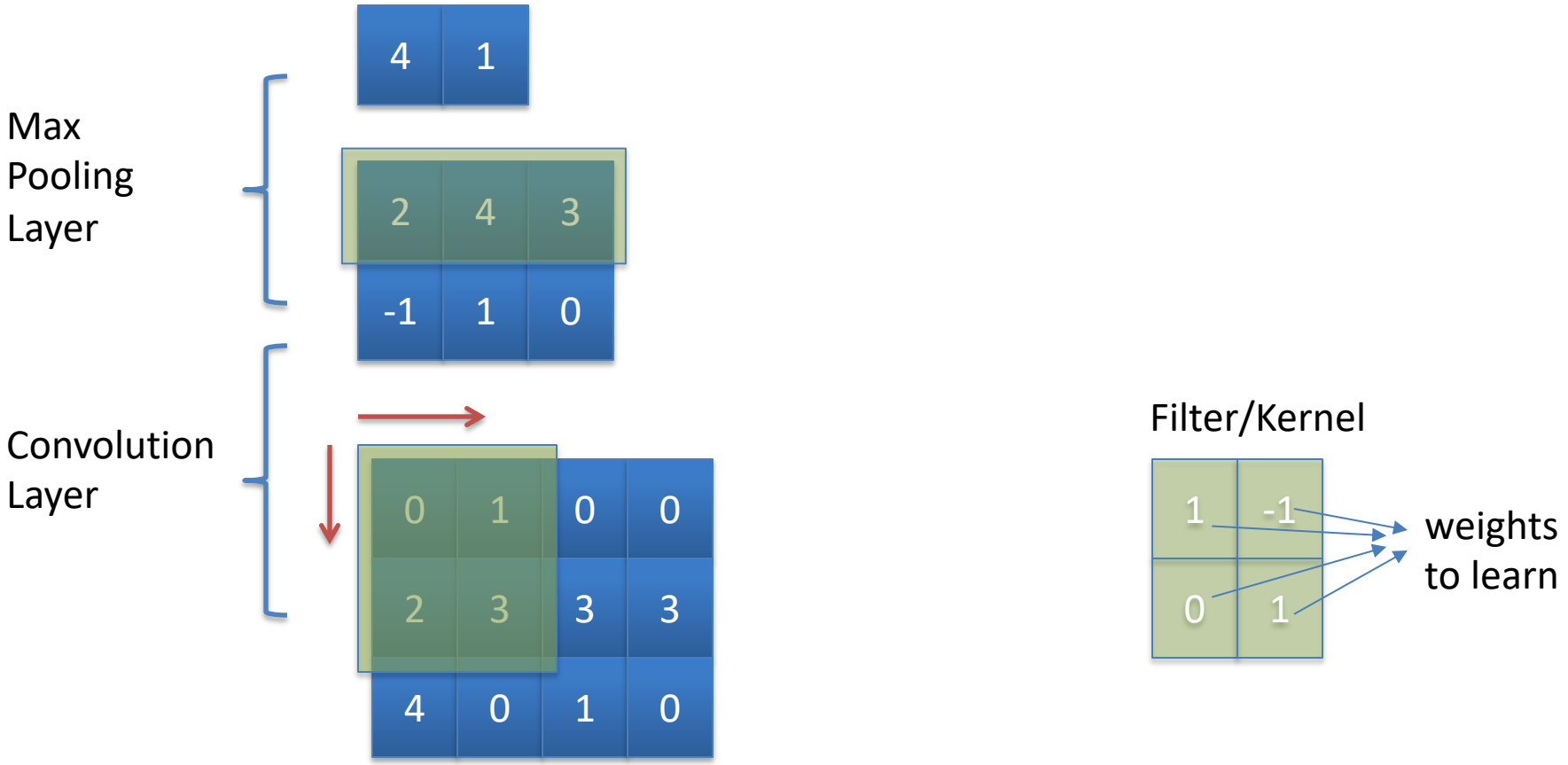
Decision

Nature scene

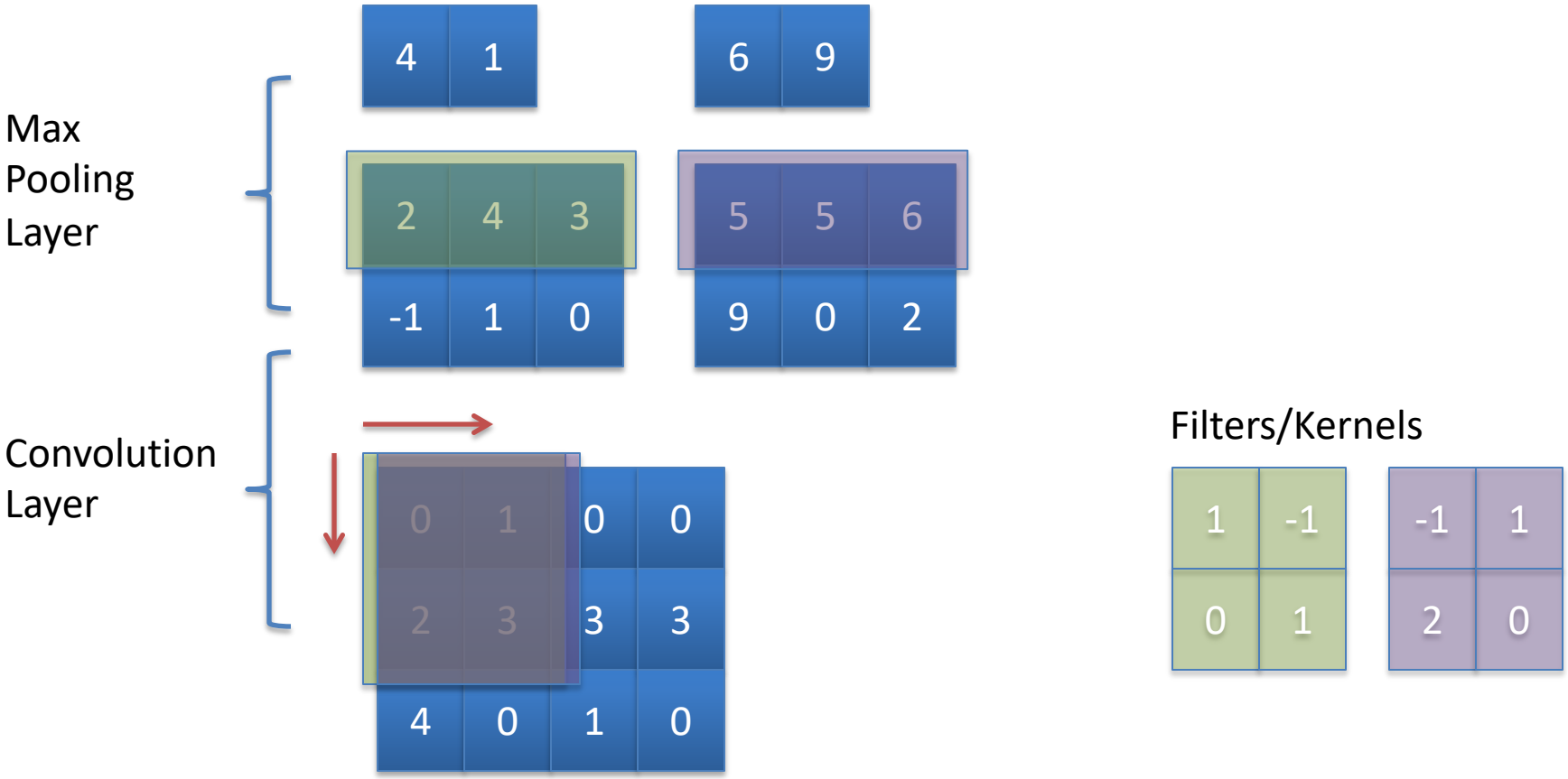
Outline

- Motivation
- Convolutional Neural Nets (CNN)
- Training
- Applications

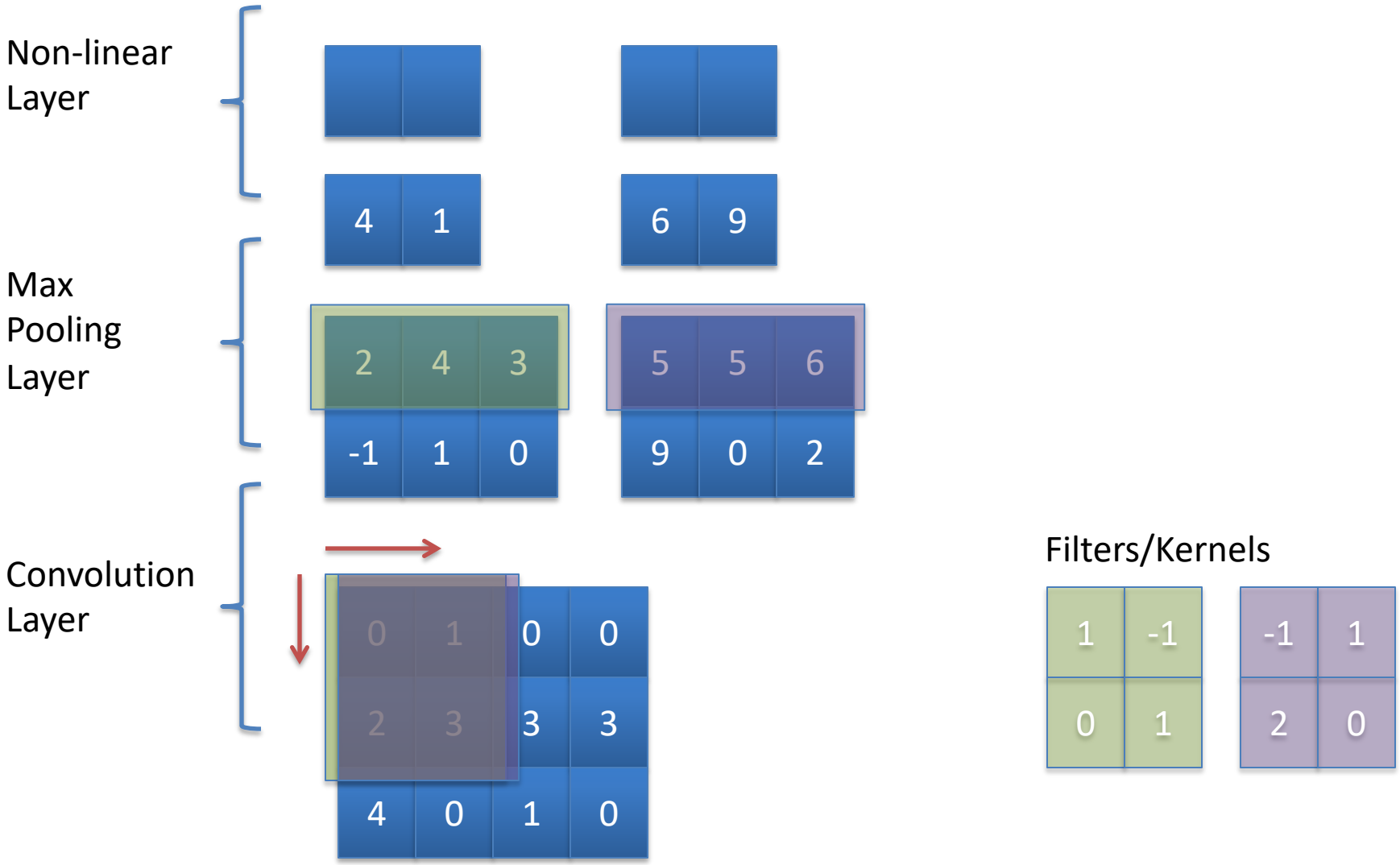
Convolutional Neural Nets



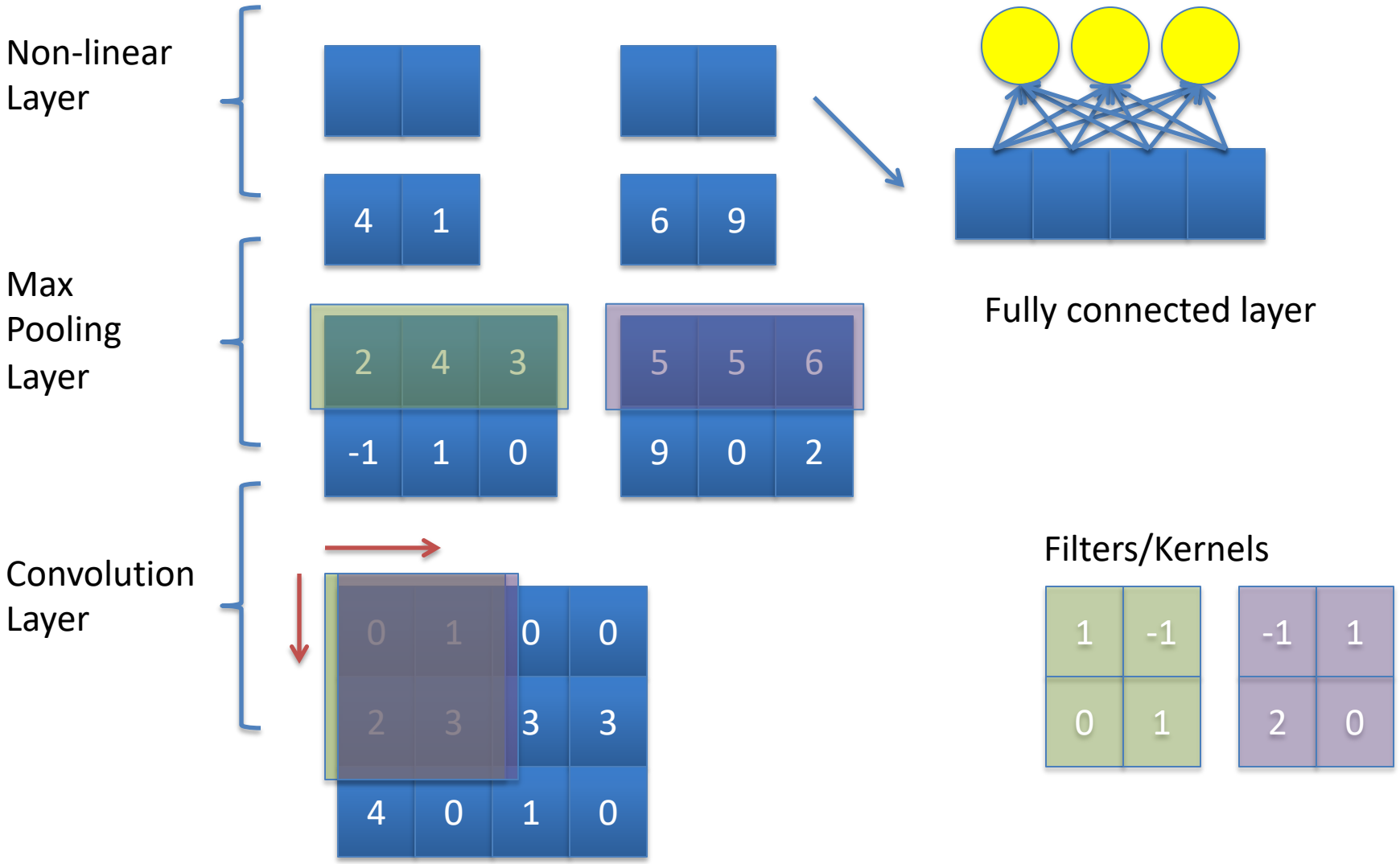
Convolutional Neural Nets



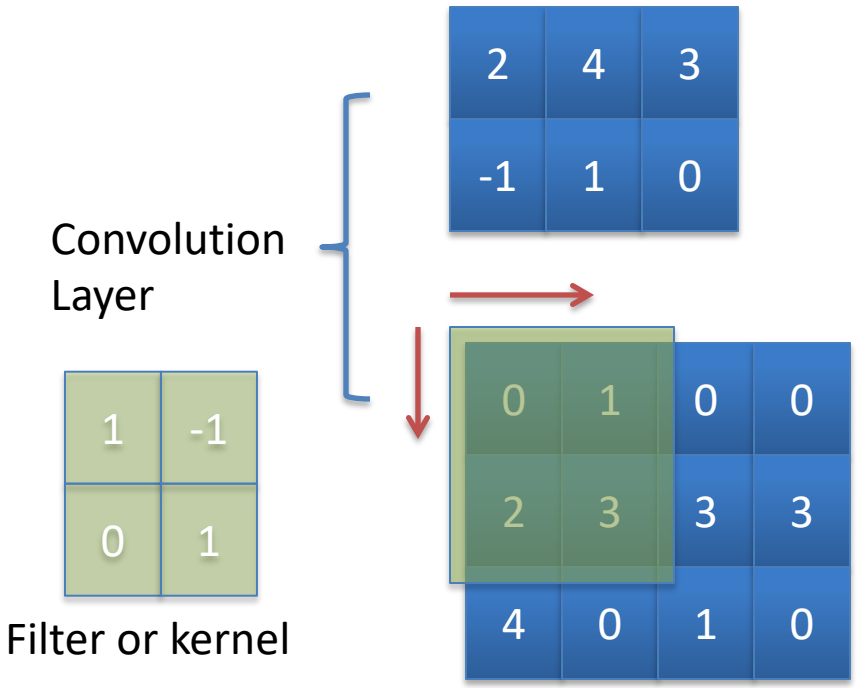
Convolutional Neural Nets



Convolutional Neural Nets



Convolution Layer

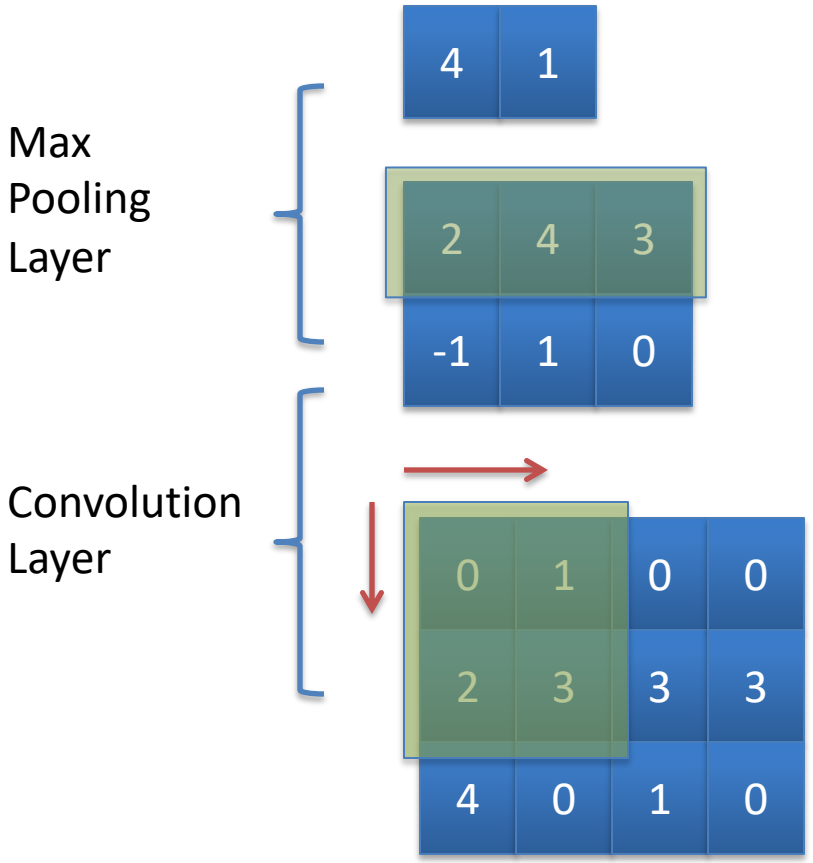


- Learn filters (each is a *mxn* window with weights)
- Shift this window through the input
- Shift called ‚stride' e.g. = 1
- Apply the convolution:

$$f^{l}_{hw} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} w^{l}_{ij} f^{l-1}_{h+i,w+j}$$

- Intuition:
  - Learn local features which are important for classification
  - Position independent

(Max) Pooling Layer



- Define a window size (*mxn*)
- Shift this window through the input
- Shift called ‚stride‘, e.g. 1
- Get the max value:

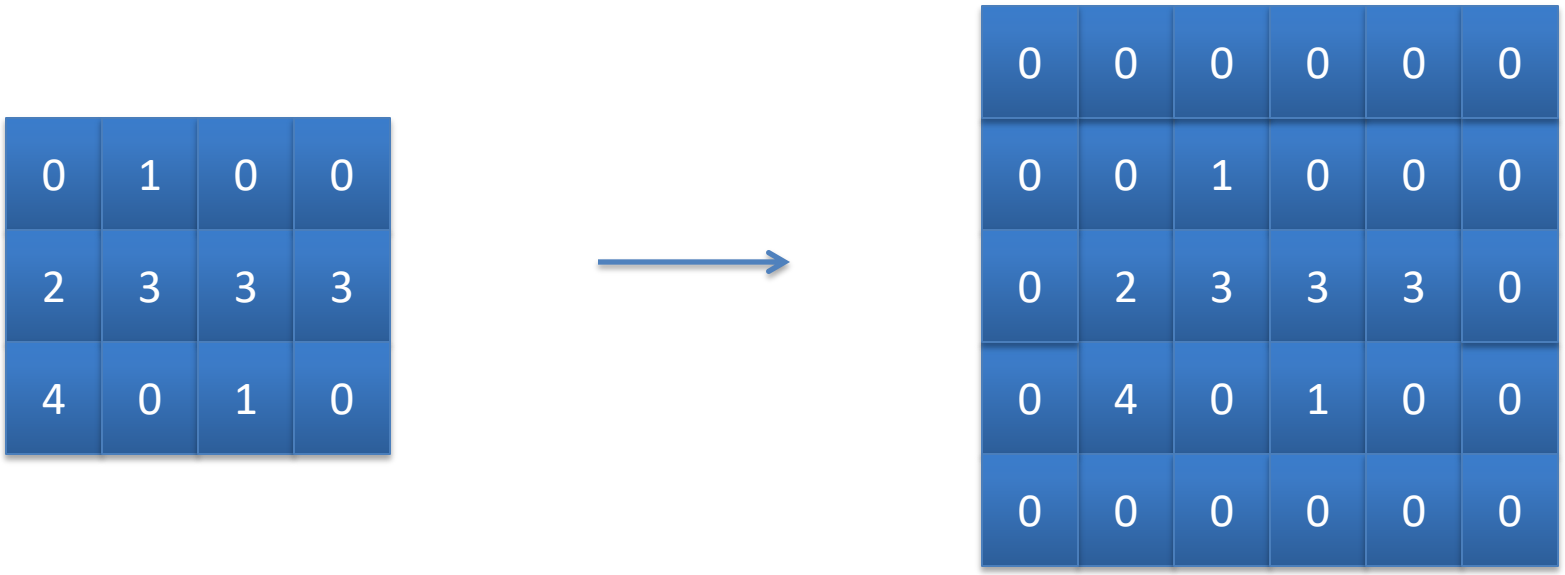
$$f_{hw}^{l+1} = \max_{0<i<m,0<j<n} \{ f_{h+i,w+j}^{l} \}$$

- Intuition:
  - The network should learn to pickup the most important information for further classification

CNN Hyper-parameters

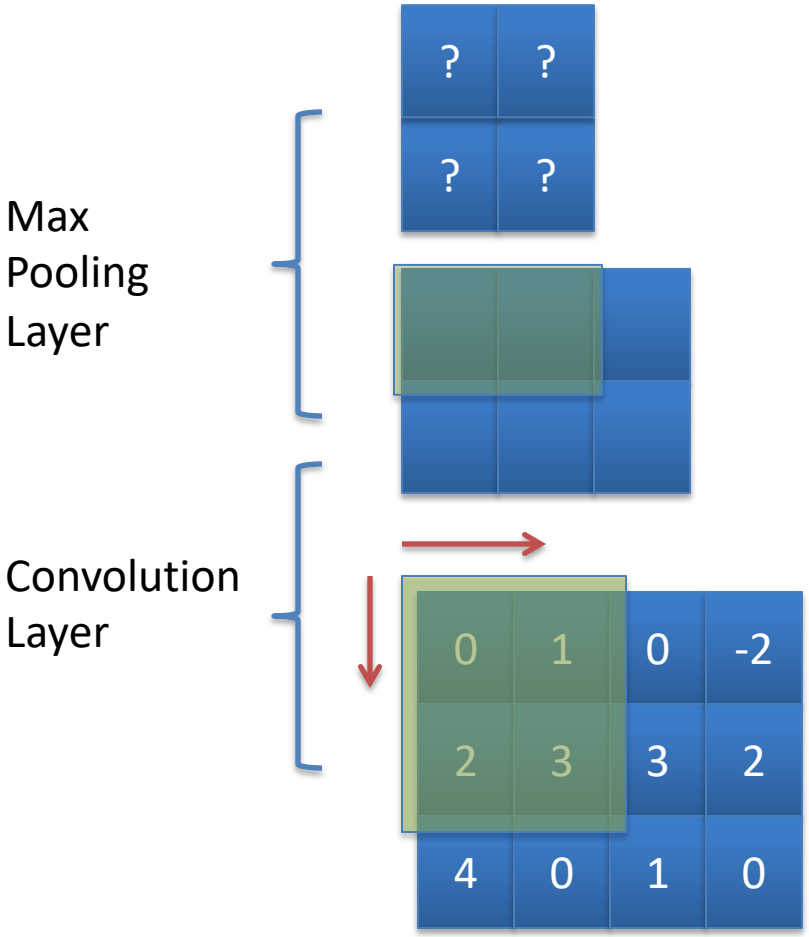
- Convolution Layer:
  - Number of filters
  - Filter size: height and width
  - Stride
- Max pooling layer
  - Pooling size: height and width
  - Stride

Zero Padding



- Motivation:
  - The filters are not well defined for data input which are near the borders → zero padding will help
  - Zero padding size is also one of the hyper-parameters

Example



- Compute the output of the network after applying the convolution and max pooling layer
- b = 0
  stride = (1,1)

-1	1
2	-2

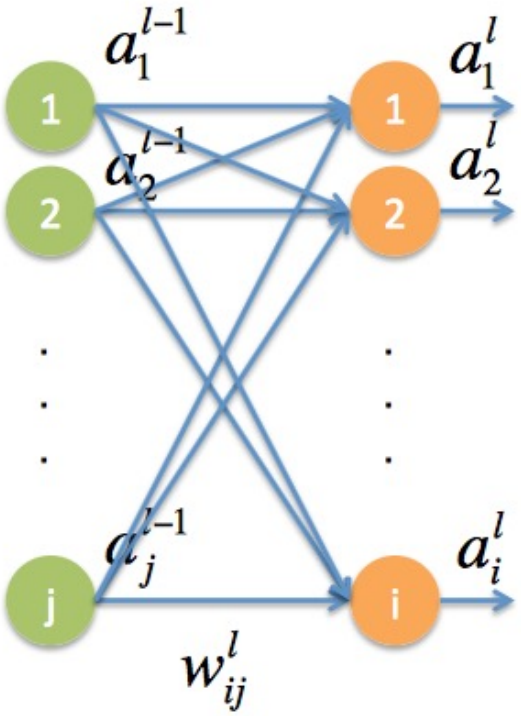
Outline

- Motivation
- Convolutional Neural Nets (CNN)
- Training
- Applications

CNN Training with Gradient Descent

- Almost the same as the training for MLP
- Backprogation can be applied
- New things are:
  - Gradient for the convolution layer
  - Gradient for the max pooling layer

Recap: Backpropagation



$$\frac{\partial C}{\partial w_{ij}^l} = \frac{\partial z_i^l}{\partial w_{ij}^l} \frac{\partial C}{\partial z_i^l}$$

$$\begin{cases} a_j^{l-1} & l > 1 \\ x_j & l = 1 \end{cases}$$

$$\delta_i^l$$

Forward pass

$$z^l = W^l a^{l-1} + b^l$$

$$a^l = \sigma(z^l)$$

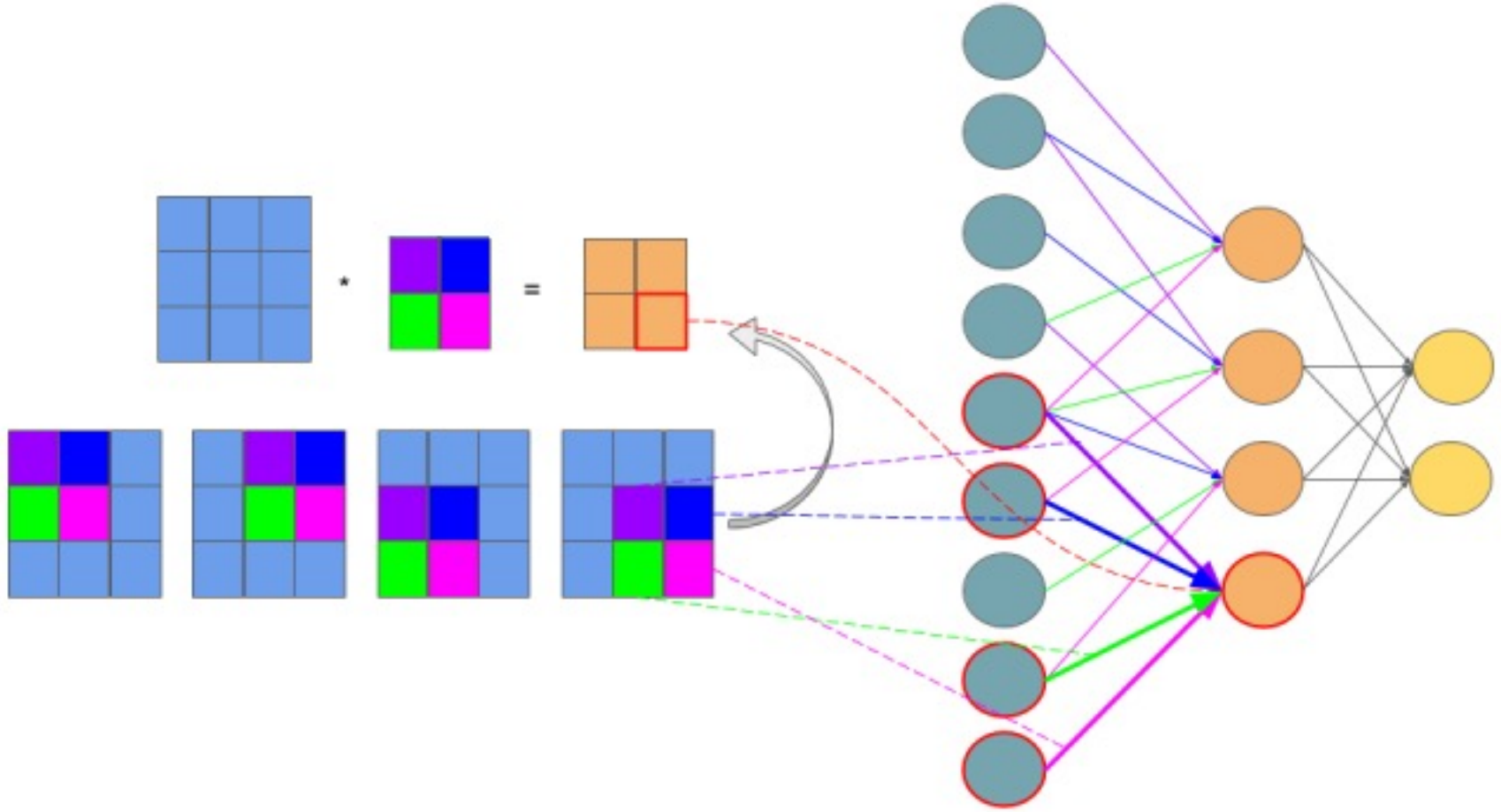
Backward pass

$$\delta^L = \sigma'(z^L)\nabla C(y)$$

$$\delta^l = \sigma'(z^l)(W^{l+1})^T \delta^{l+1}$$

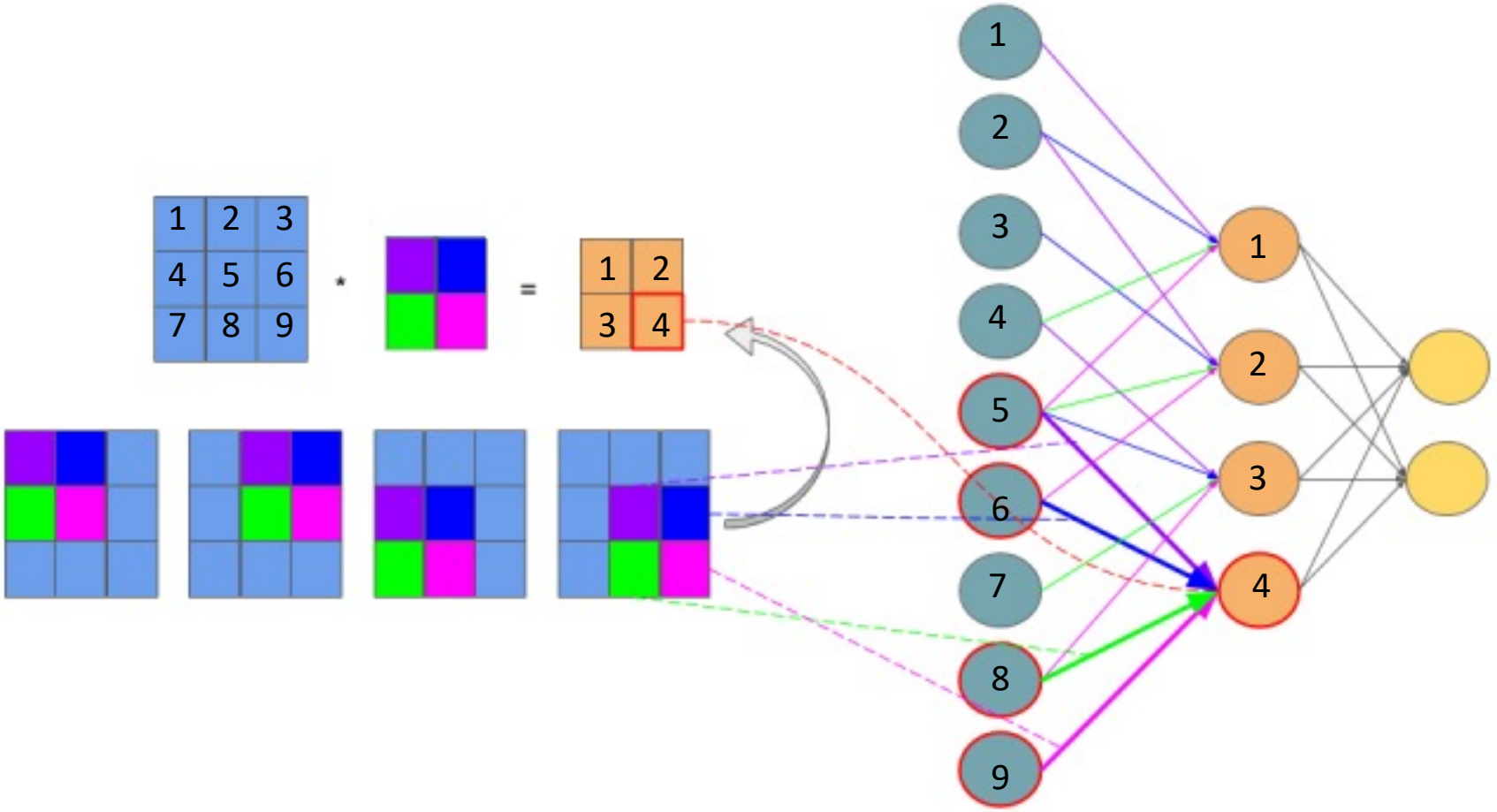
.....

Alternative CNN representation



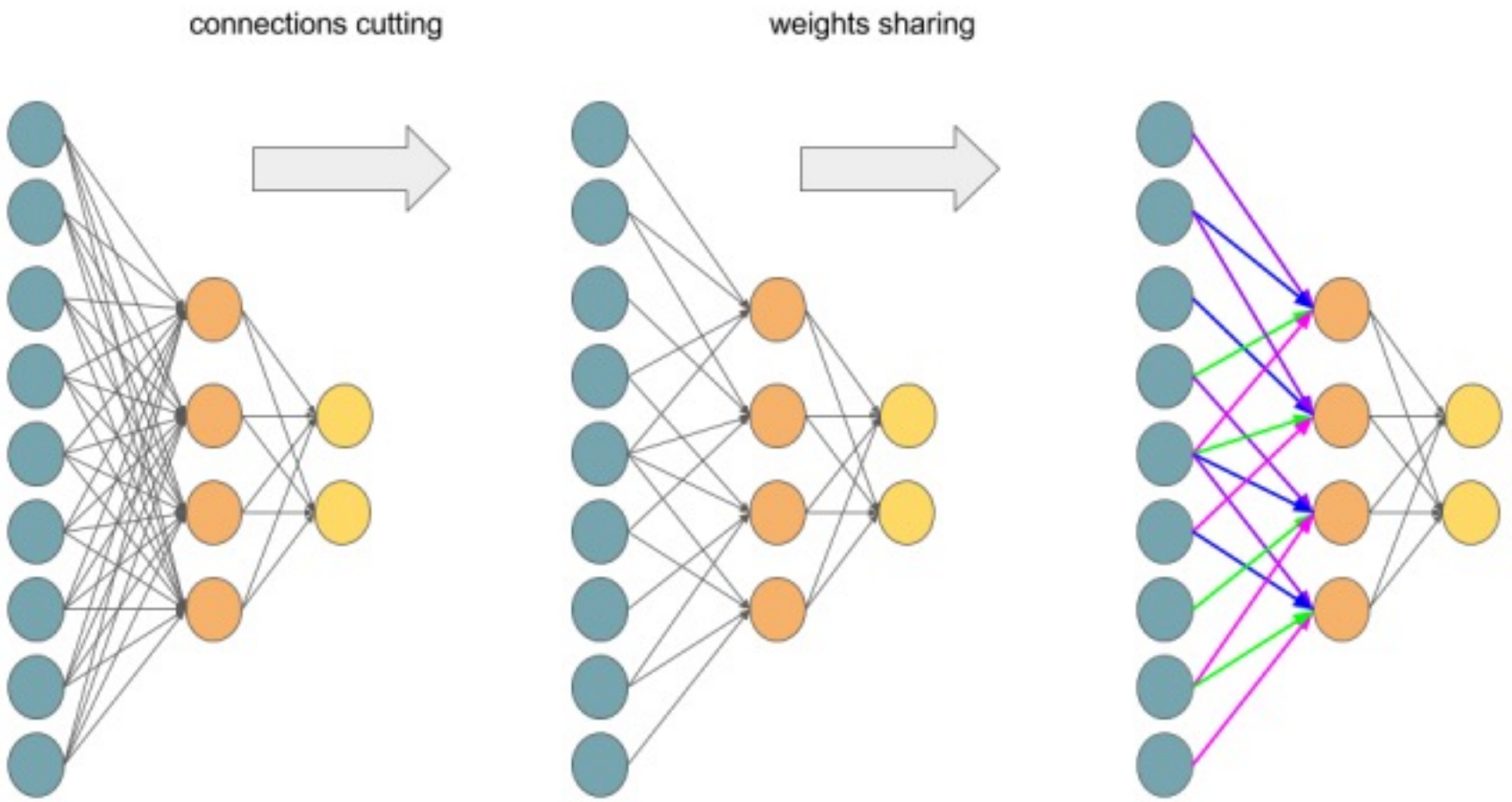
https://grzegorzgwardys.wordpress.com/2016/04/22/8/
(slightly adapted)

Alternative CNN representation



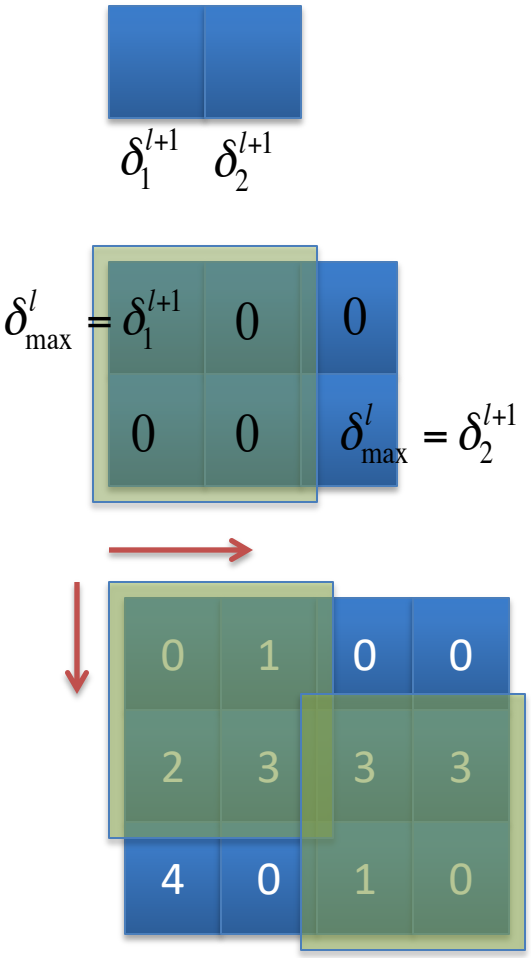
https://grzegorzgwardys.wordpress.com/2016/04/22/8/
(slightly adapted)

Alternative CNN representation



https://grzegorzgwardys.wordpress.com/2016/04/22/8/

Gradient Computation for CNN



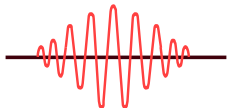
- Gradient of the convolution is a sum over all gradients of the shared parameters
- Gradient of the max pooling layer depends on the indices of the largest value
  - Need to store the indices of the largest value

Outline

- Motivation
- Convolutional Neural Nets (CNN)
- Training
- Applications

CNN for Speech Emotion Recognition [Neumann, Vu, 2017]

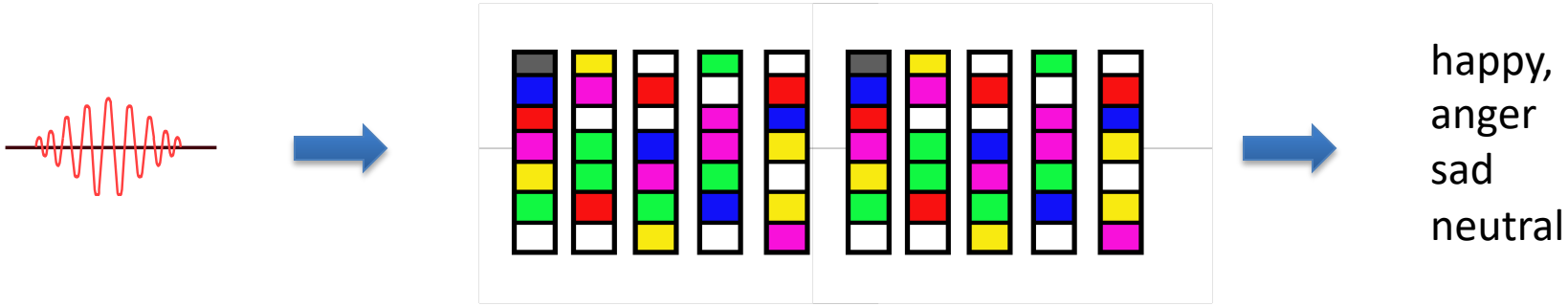
- Extract emotion from speech



Which emotion? (happy, anger, sad or neutral)

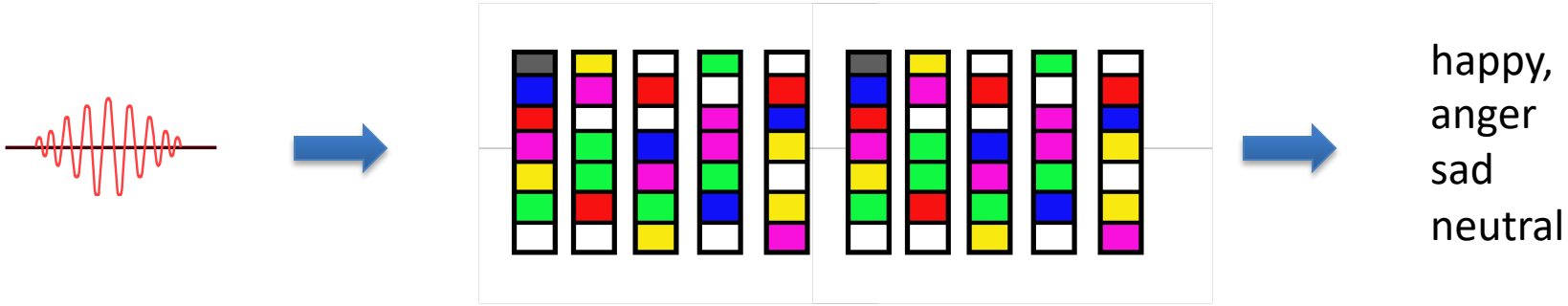
CNN for Speech Emotion Recognition [Neumann, Vu, 2017]

- Extract emotion from speech



CNN for Speech Emotion Recognition [Neumann, Vu, 2017]

- Extract emotion from speech



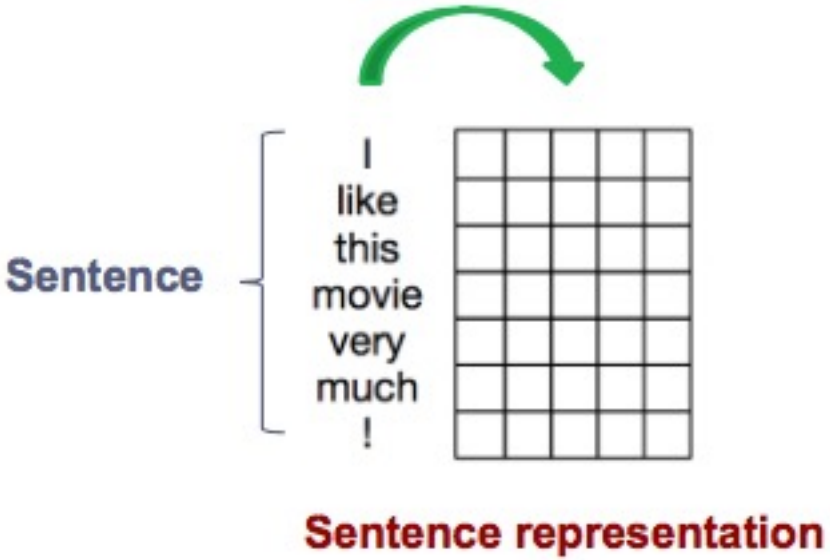
Features (dim.)	Average	Min	Max
logMel (26)	**61.71**	60.40	**62.66**
MFCC (13)	61.31	**60.85**	61.35
eGeMAPS (25)	60.25	59.41	60.94
Prosody (7)	56.34	55.82	57.57

CNN for Natural Language Processing

- Mostly used for sentence modelling
  - Kalchbrenner et al 2014
  - Kim, 2014
  - Etc.
- If the input is a sentence, where can we get the image for CNN?

CNN for Natural Language Processing

- If the input is a sentence, where can we get the image for CNN?
  - Word representations (word vectors / word embeddings): a word can be represented as a vector



Word → Vector

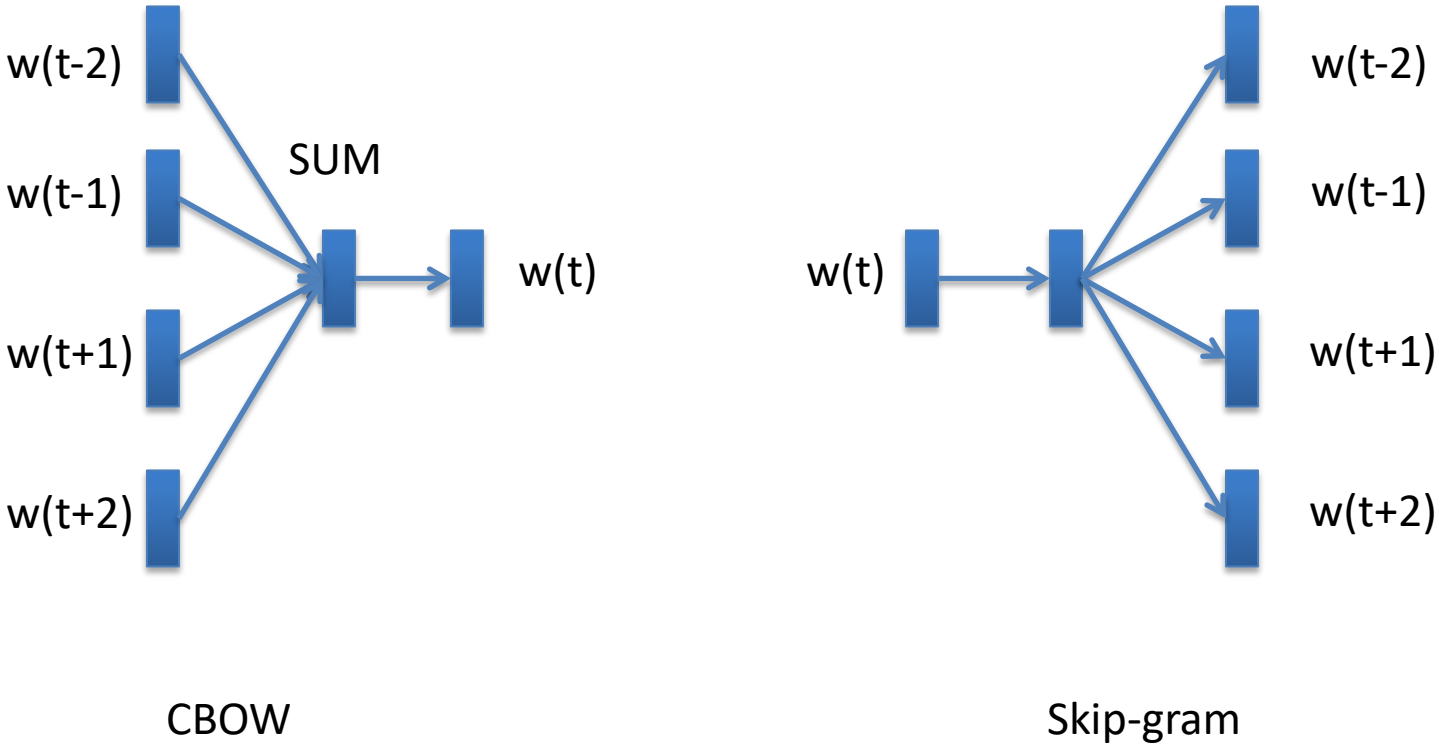
- 1 hot vector (1-of-N)
  - Define a lexicon = [cool, bad, great, very, not]
  - ‚cool‘ = [1, 0, 0, 0, 0]
  - ‚great‘ = [0, 0, 1, 0, 0]
  - ‚not‘ = [0, 0, 0, 0, 1]
  - ‚cooool‘ = [0, 0, 0, 0, 0]
- Properties:
  - Word vector has a dimension of the lexicon size
  - Each dimension corresponds to a word in the lexicon

Word → Vector

- 1 hot vector (1-of-N) with word hashing
- The idea is to have a lexicon with n-gram characters
  - Lexicon = [‚am_‘, ... ‚coo‘, ‚col‘, ...‘ool‘, ‚ooo‘, ...]
  - ‚cool‘ = [0, ..., 1, 0, ..., 1, ..., 0, 0, ...]
  - ‚coool‘ = [0, ..., 1, 0, ..., 1, ..., 0, 1, ...]
- It captures some relations between words
- It‘s quite useful
  - if there are a lot of OOV words
  - for morphological rich languages

Word Embeddings

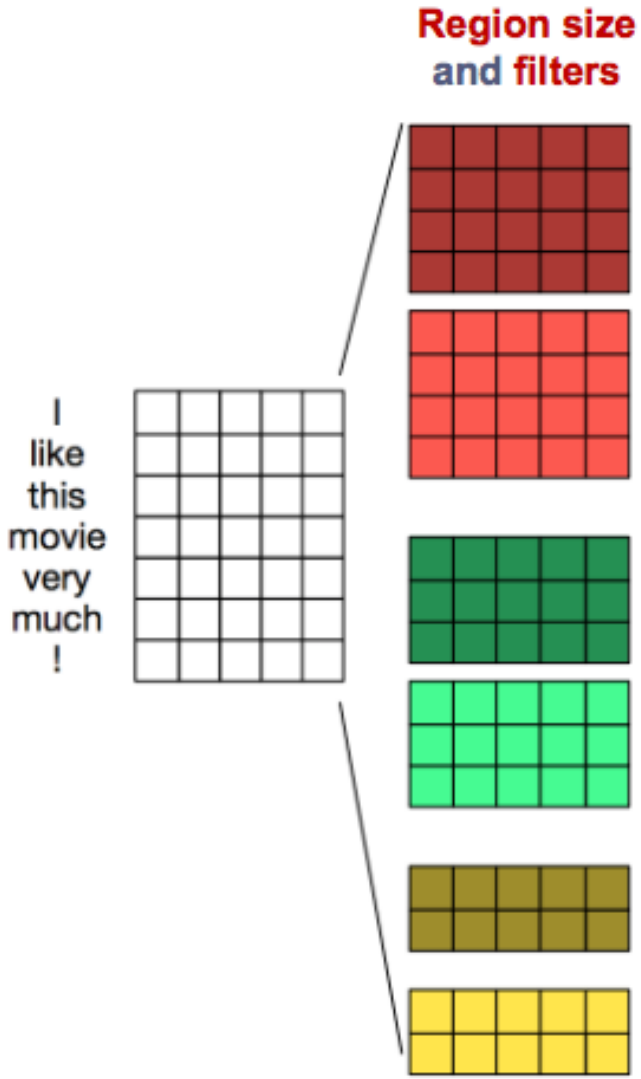
- Another way was proposed by T. Mikolov 2013
- https://code.google.com/p/word2vec/



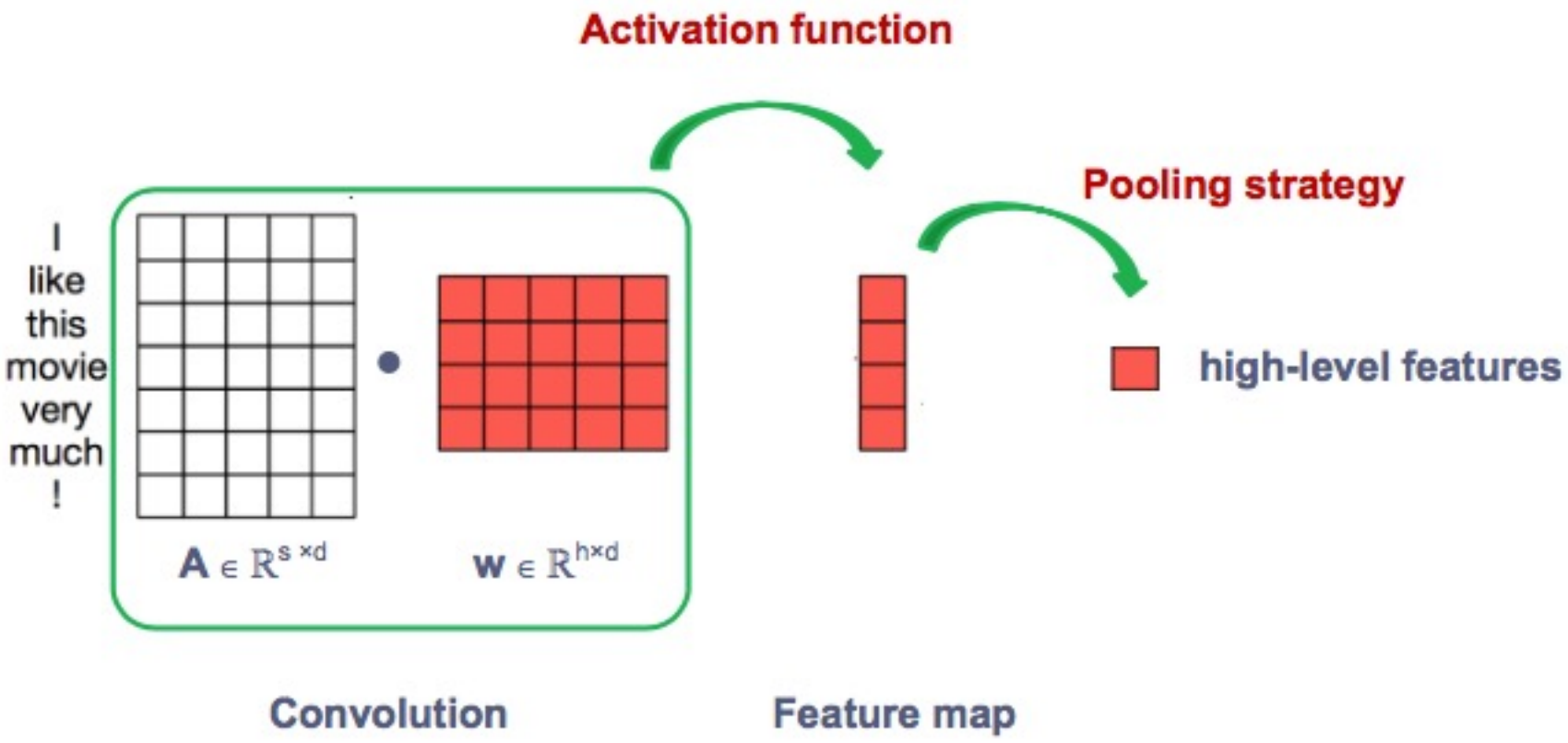
Word Embeddings

- Linear relations between words:
  - v(king) – v(queen) ≈ v(man) – v(woman)
  - v(japan) – v(tokio) ≈ v(germany) – v(berlin)
  - v(italy) – v(wine) ≈ v(germany) – v(?)
  - v(japan) – v(sushi) ≈ v(germany) – v(?)
- It can be used to solve analogy tasks
  - Rome : Italy ≈ Berlin : ?
  - Compute v(Berlin) – v(Rome) + v(Italy)
  - Find the closest vector

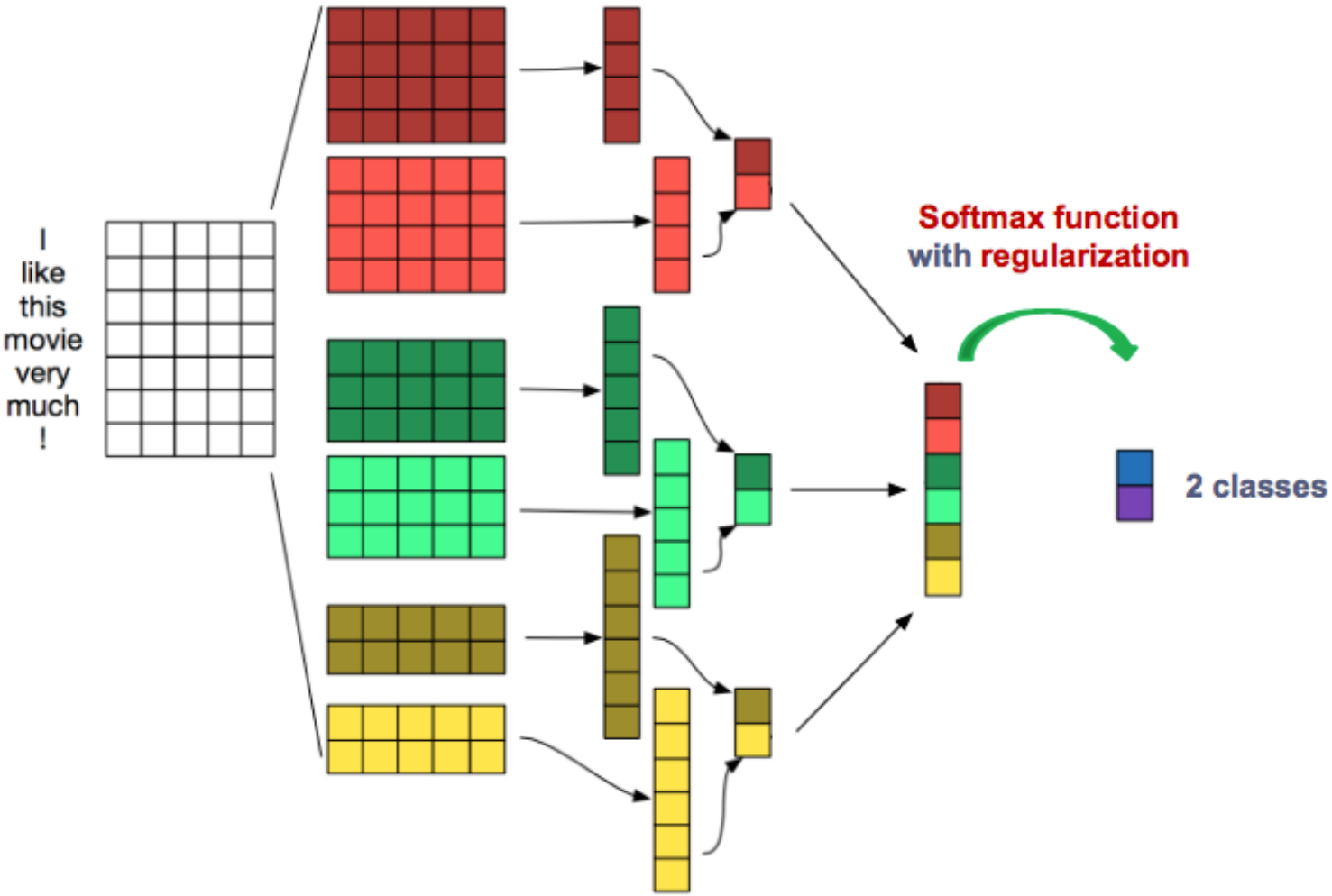
CNN for Natural Language Processing



CNN for Natural Language Processing



CNN for Natural Language Processing

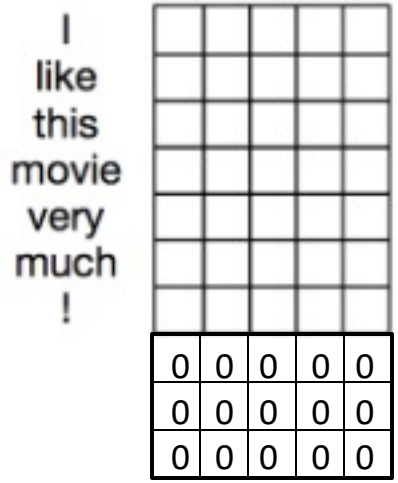


CNN for Natural Language Processing

- What if the sentences have different length?

CNN for Natural Language Processing

- What if the sentences have different length?
  - Define a max sentence length
  - Pad the sentence with zero vectos or cut the sentence to the max sentence length
  - E.g. with a max sentence length of 10
    - ‚I like this movie very much!‘ can be represented as follows



Thanks for listening!