The Deep Learning Revolution

rethinking machine learning pipelines

Soumith Chintala

Facebook AI Research

7th April 2015

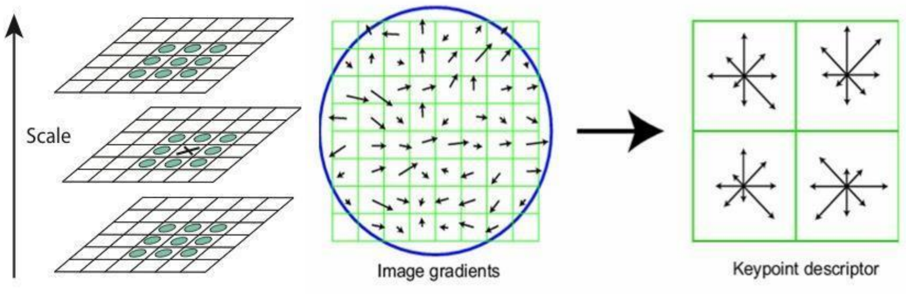
EmergingTech 2015

Getting your attention

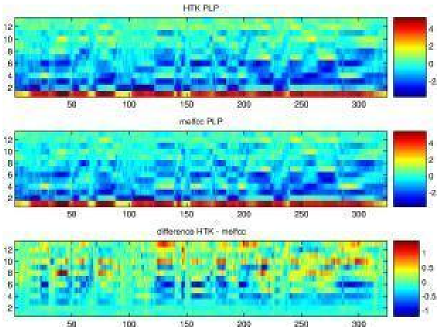
- Changed the landscape of:
    - Natural language processing
    - Speech recognition
    - Computer vision
    - Robotics
    - Modern statistical physics
    - Computational Biology
    - Digital assistants (Siri, Cortana, etc.)

Traditional Machine Learning

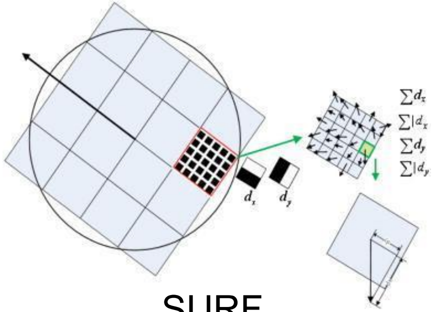
Hand-Crafted Features



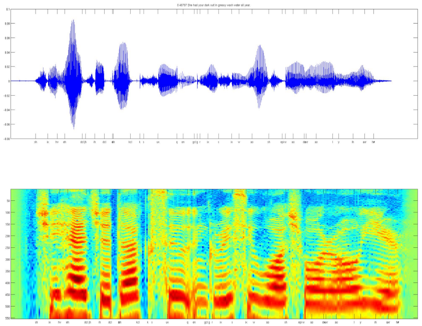
SIFT/HOG



MFCC



SURF



Spectrogram

Traditional ML

- Feature Engineering + [your favorite classifier]
- Steps:
  - Find a poor sod to think of good features
  - Find another unfortunate chap to extract those features from your data
  - Collect all the features and grind them through:
    - Logistic regressor
    - Decision trees, random forests, boosted
    - Support Vector Machines

Deep learning

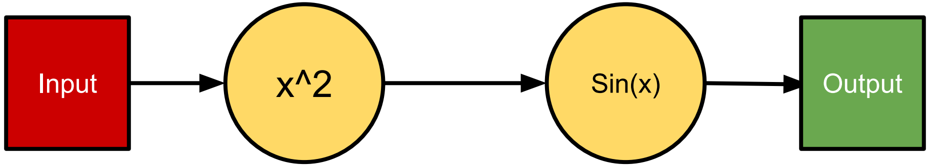
- End-to-end learning
  - No feature engineering
- Chained cascade of non-linear transforms
- General framework

How to win at life (5-step process)

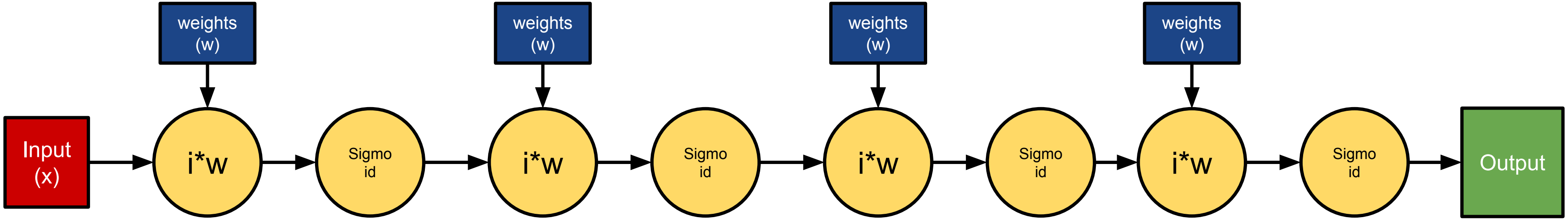
- Pick a problem
- Get as much data as you can
- Expand your dataset more
- Train several deep nets
- Ensemble
- Win!

Deep Neural Networks

What is a neural network?

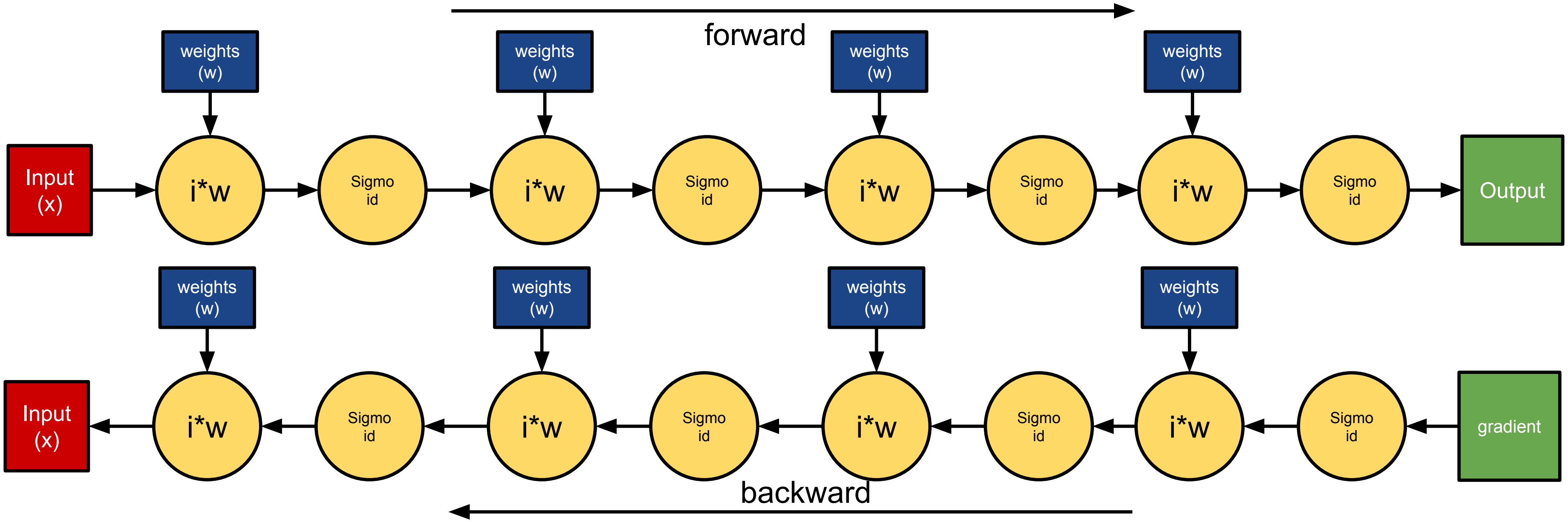


What is a deep neural network?



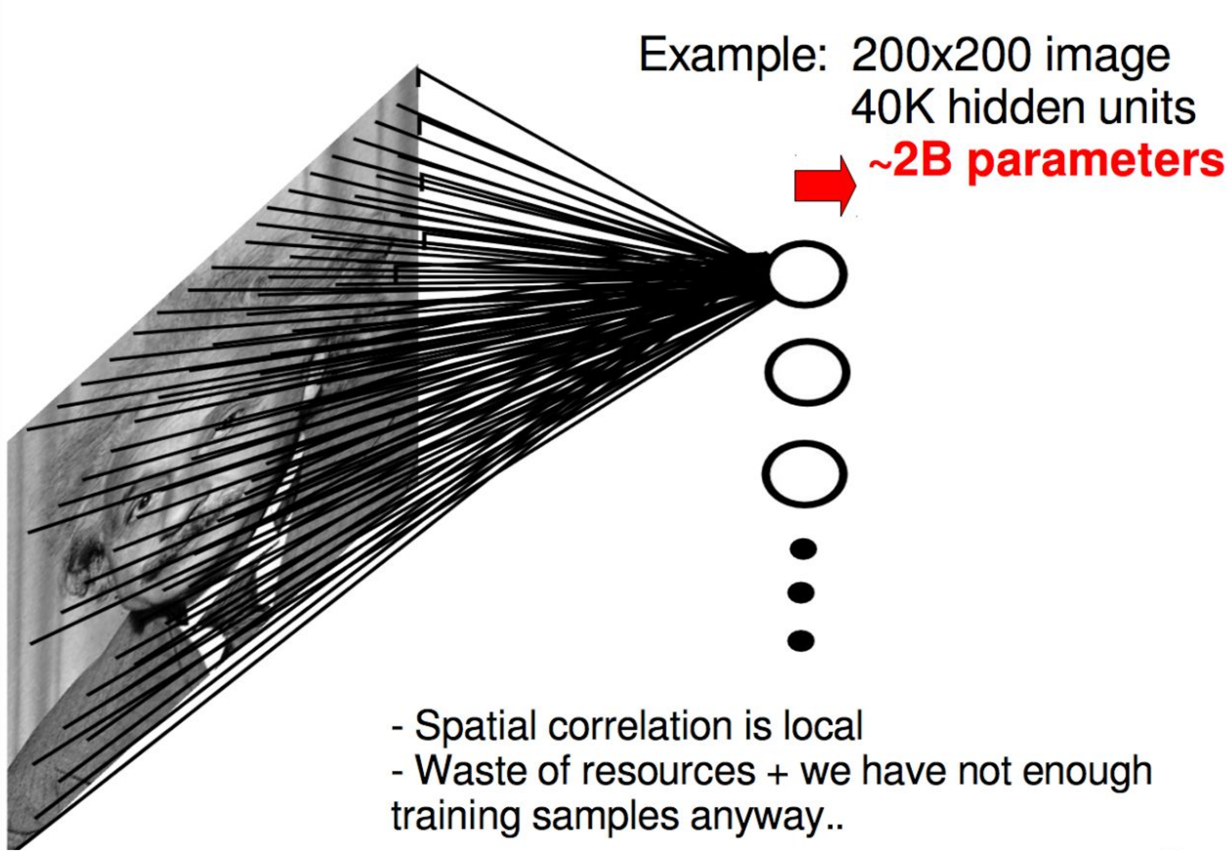
Neural networks vs computation graphs?

- Neural networks are trained via back-propagation
- Every node has f(x) and df(x)/dx



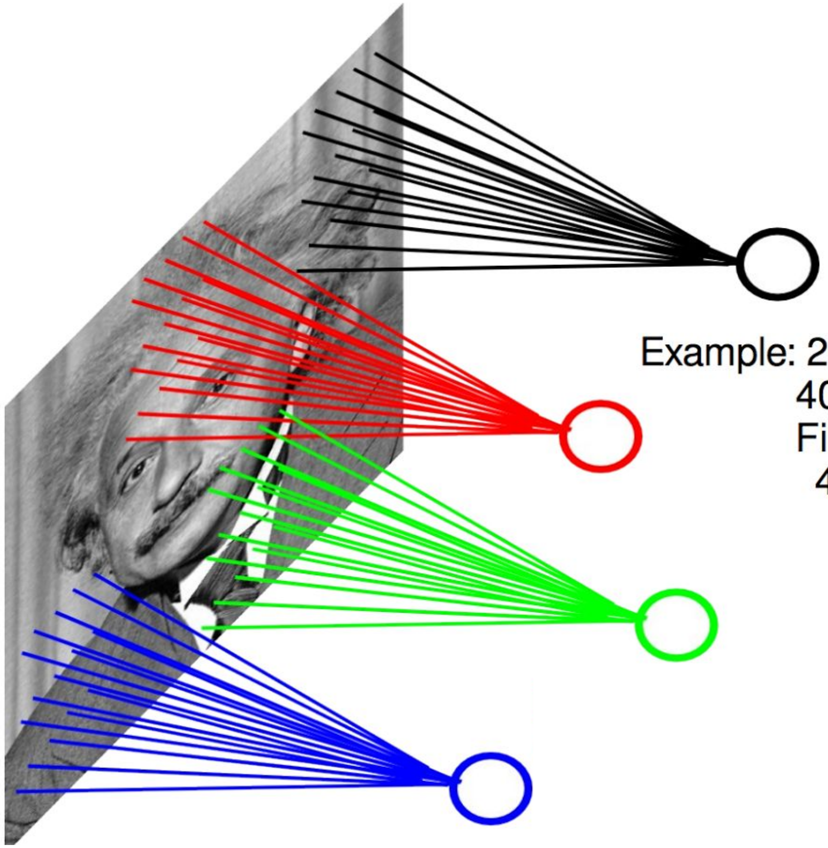
One weird trick: Convolutions

Fully-connected layers: issues



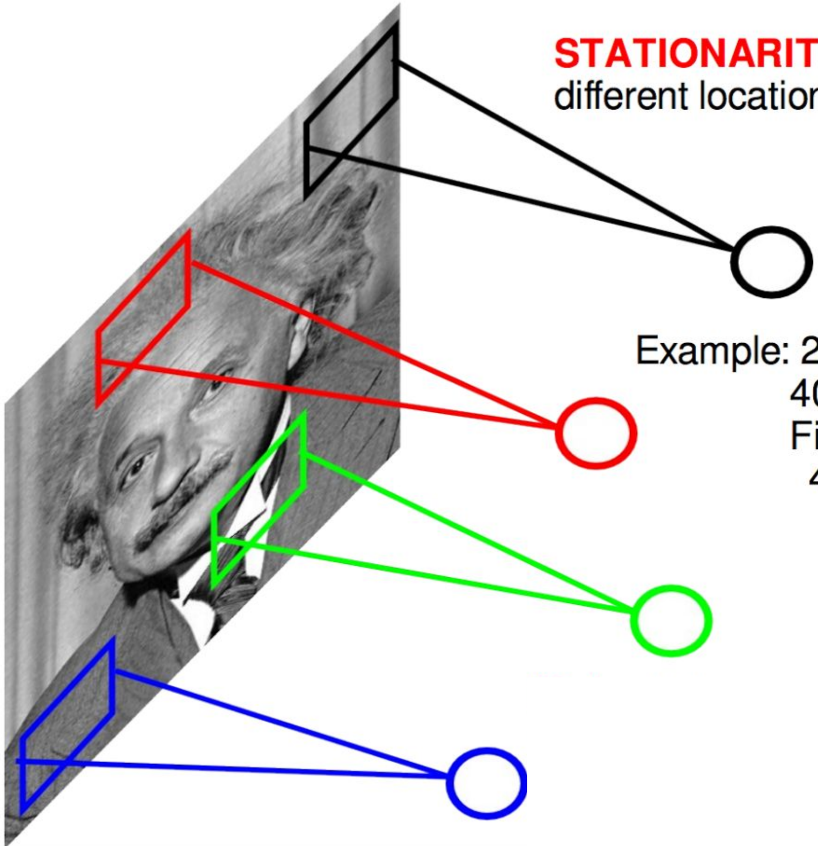
- Spatial correlation is local
- Waste of resources + we have not enough training samples anyway..

Locally connected layers



Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

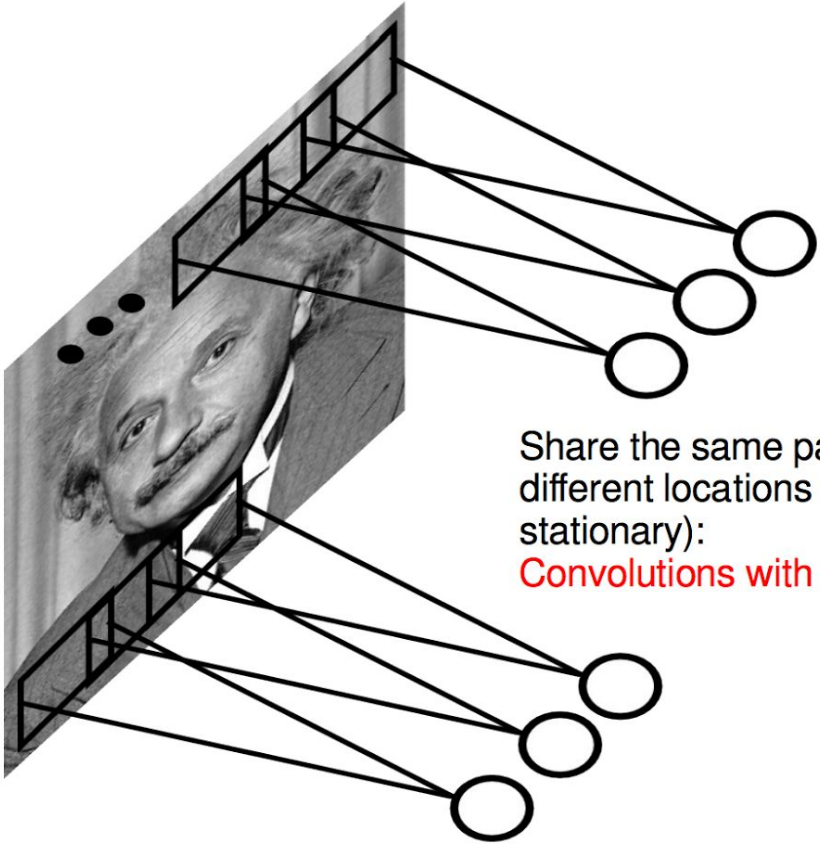
Convolutional layers



STATIONARITY? Statistics is similar at different locations

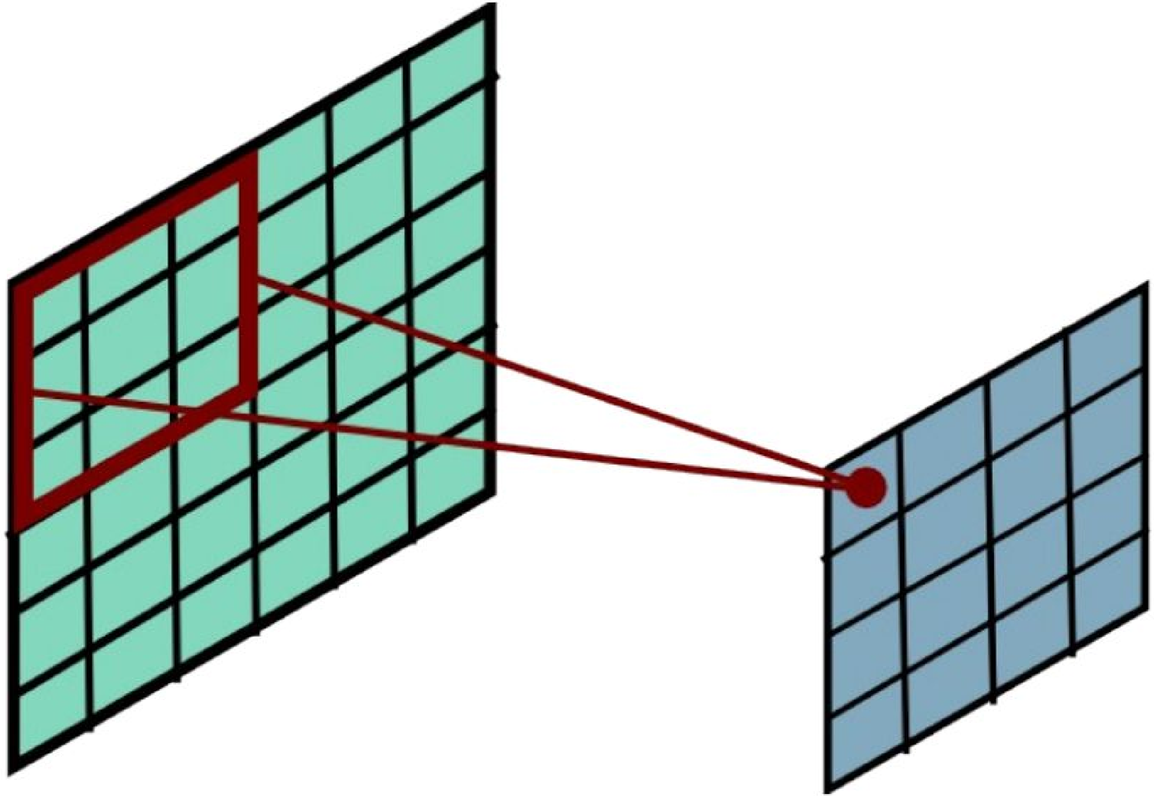
Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

Convolutional layers

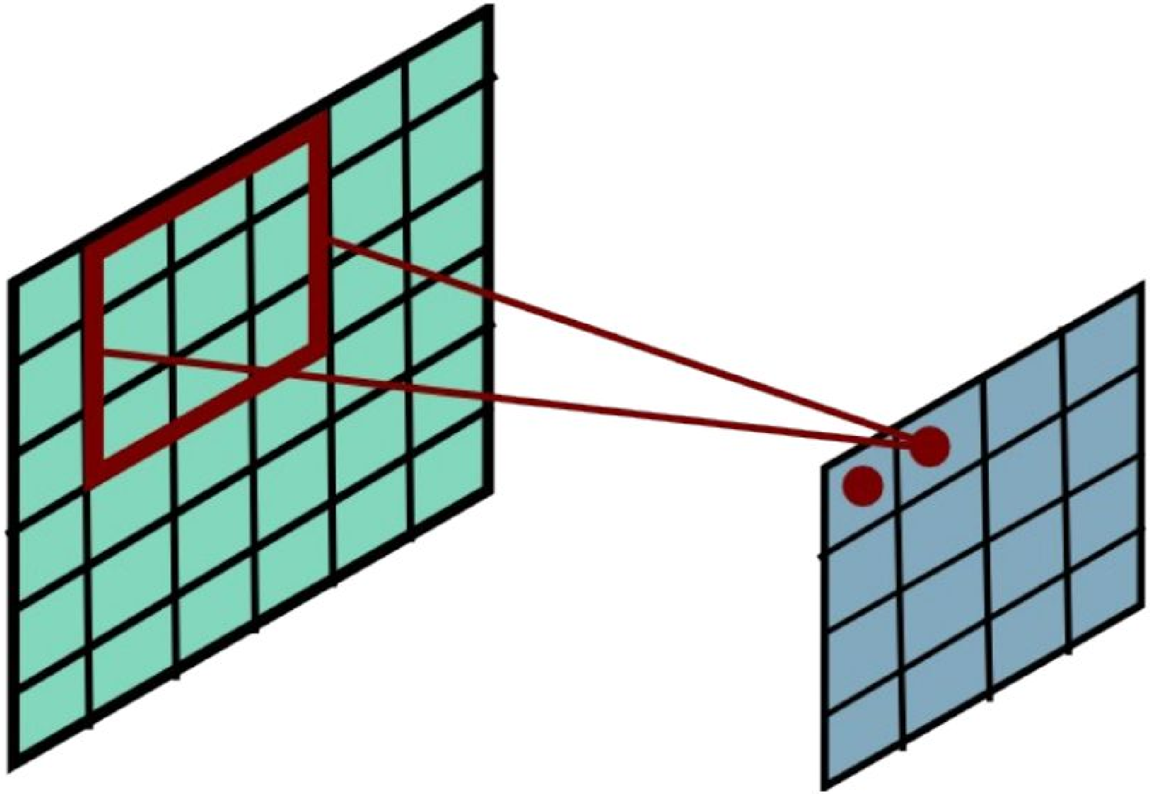


Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

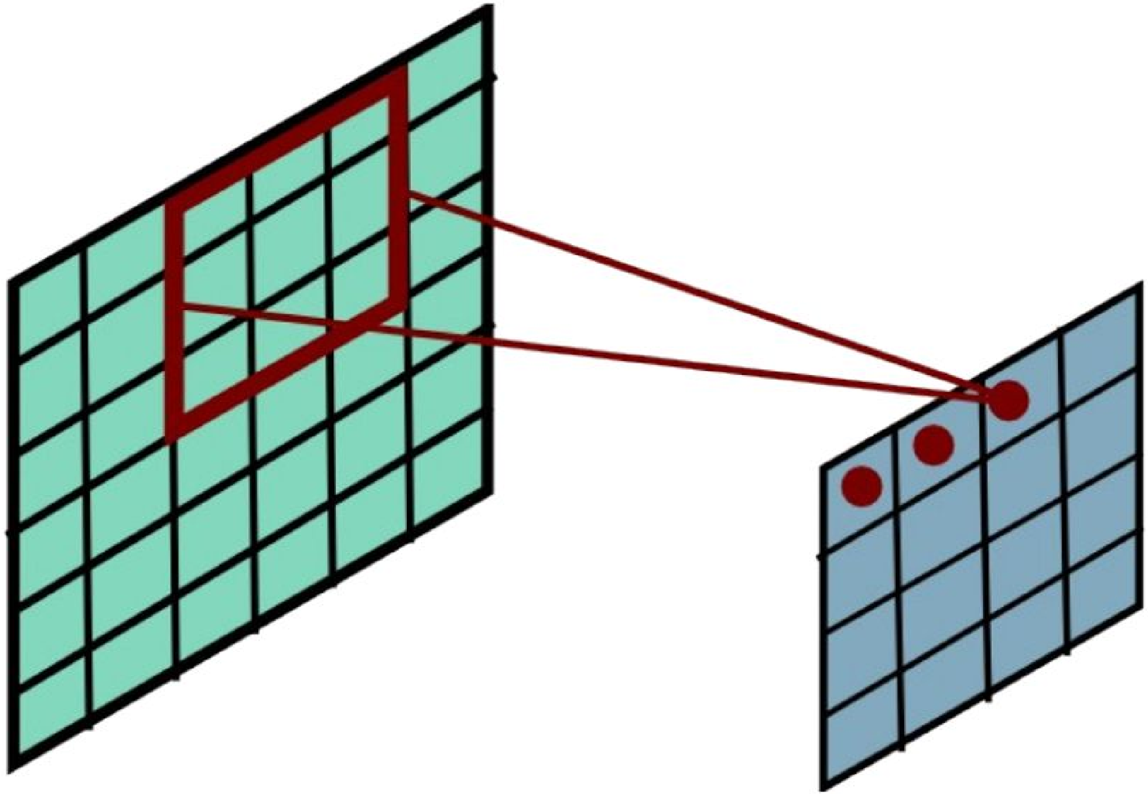
Convolution layer



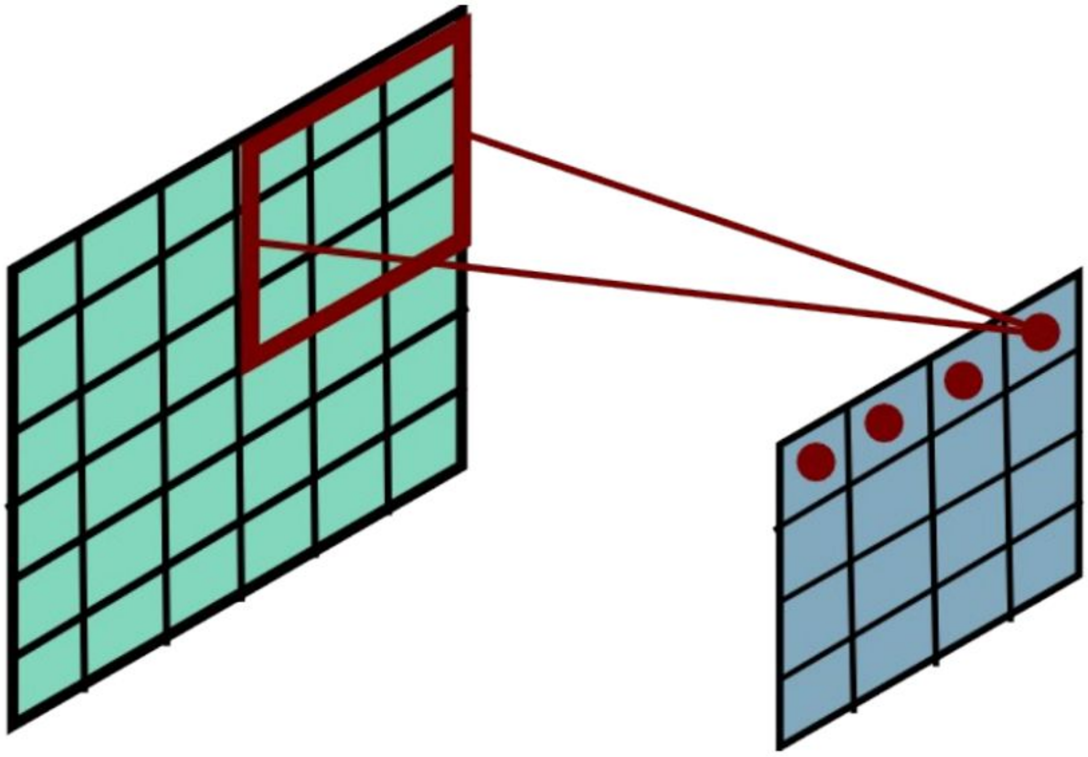
Convolution layer



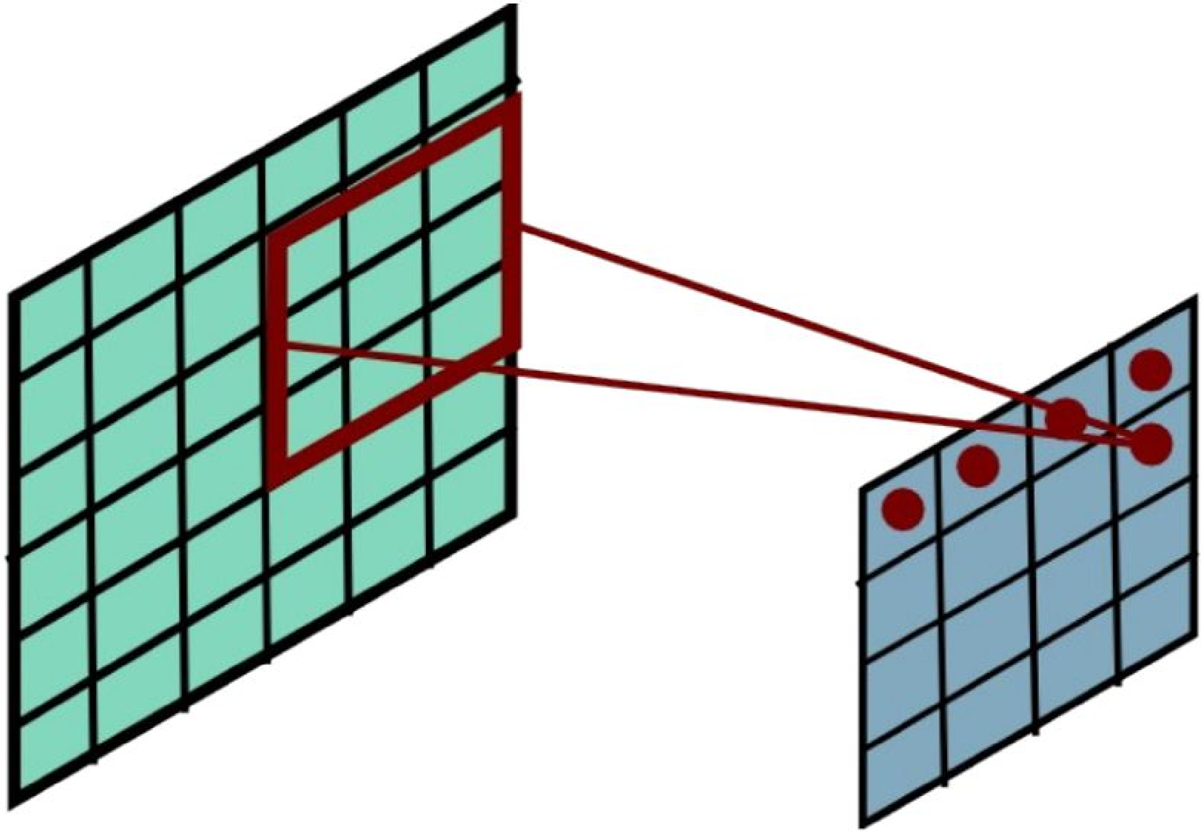
Convolution layer



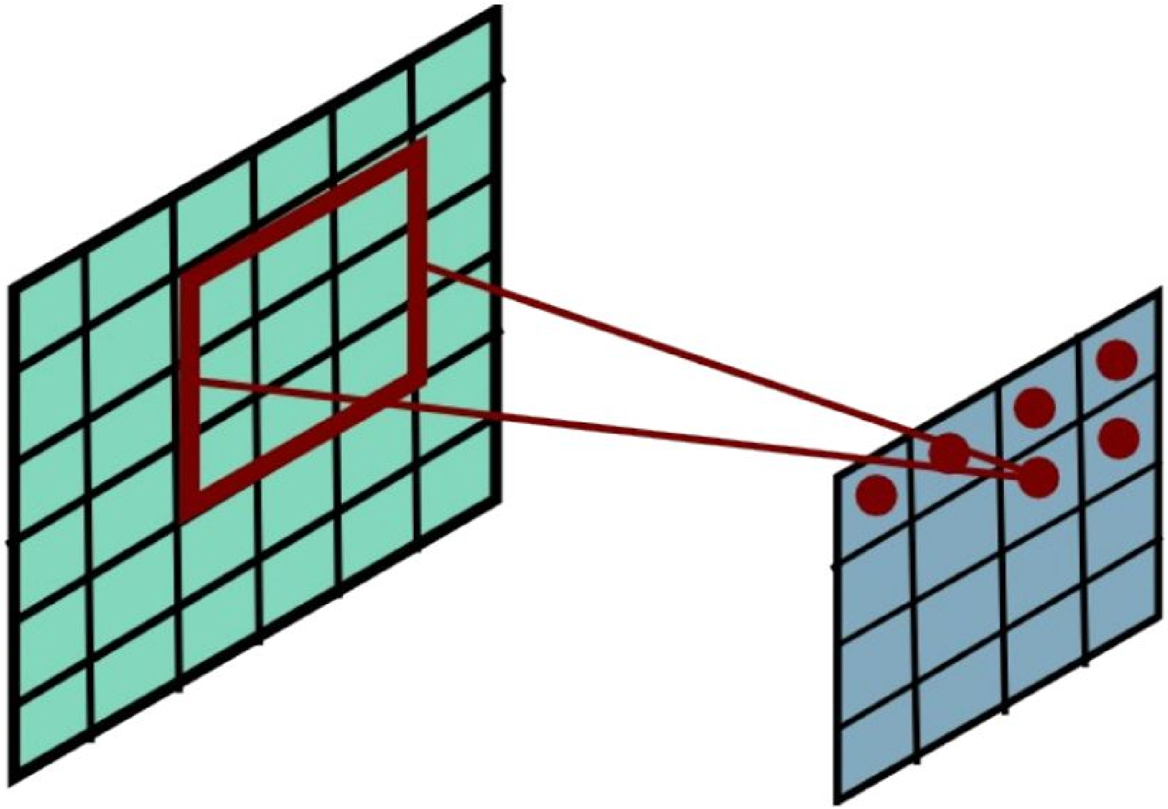
Convolution layer



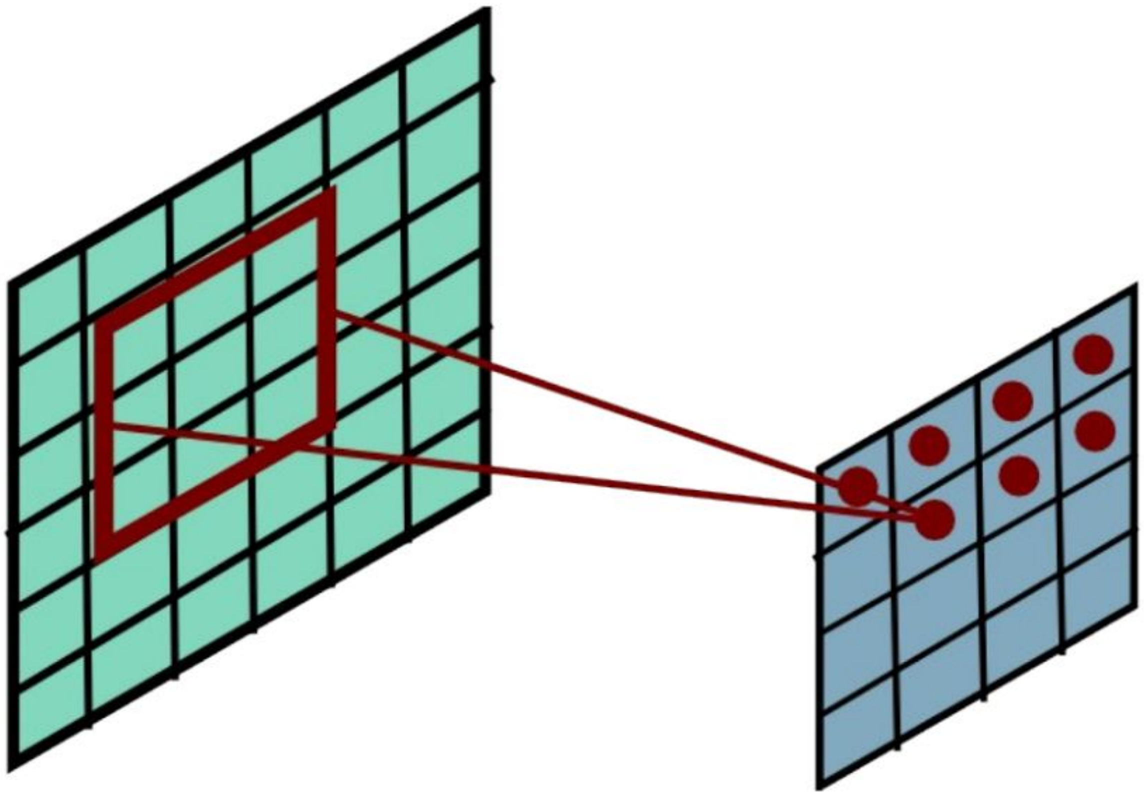
Convolution layer



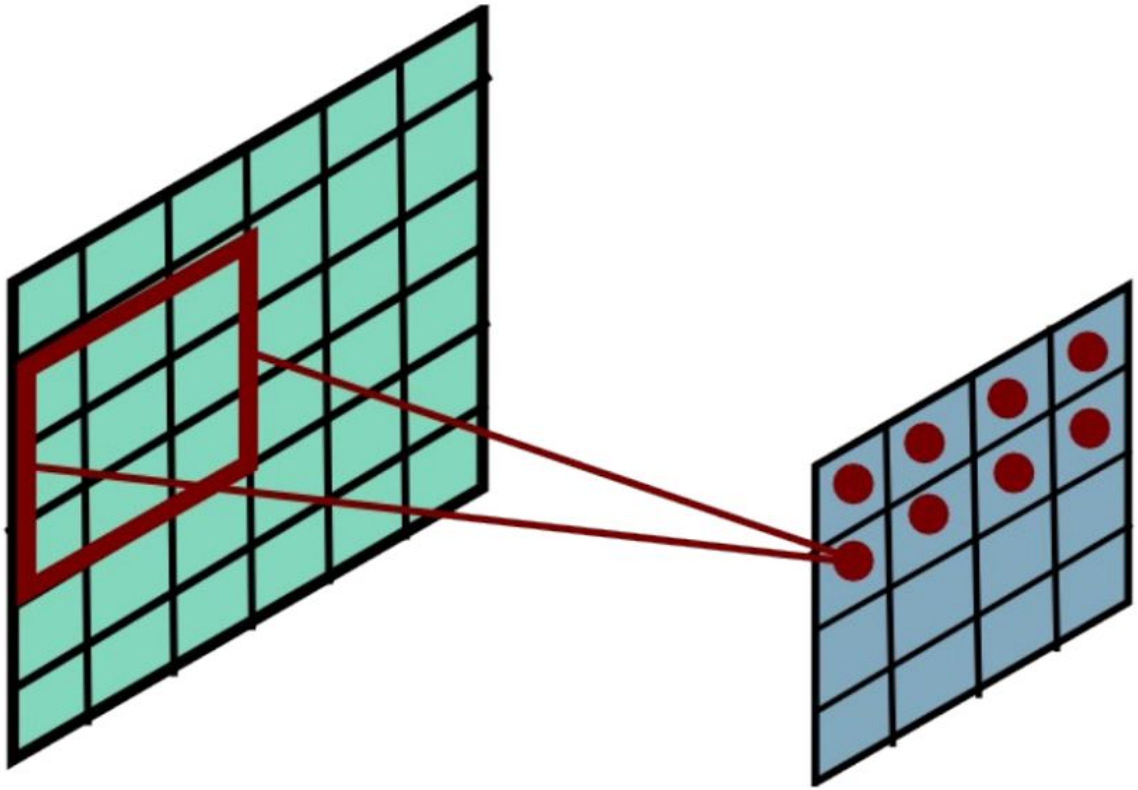
Convolution layer



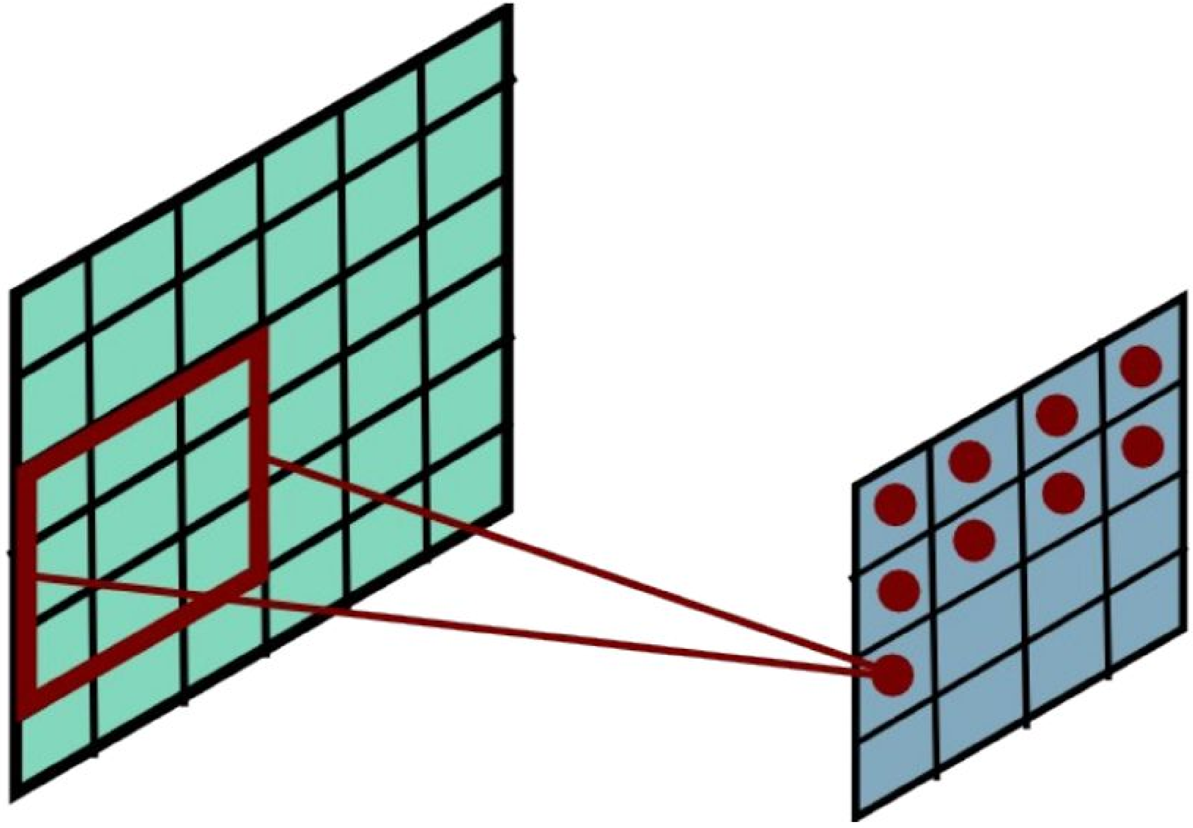
Convolution layer



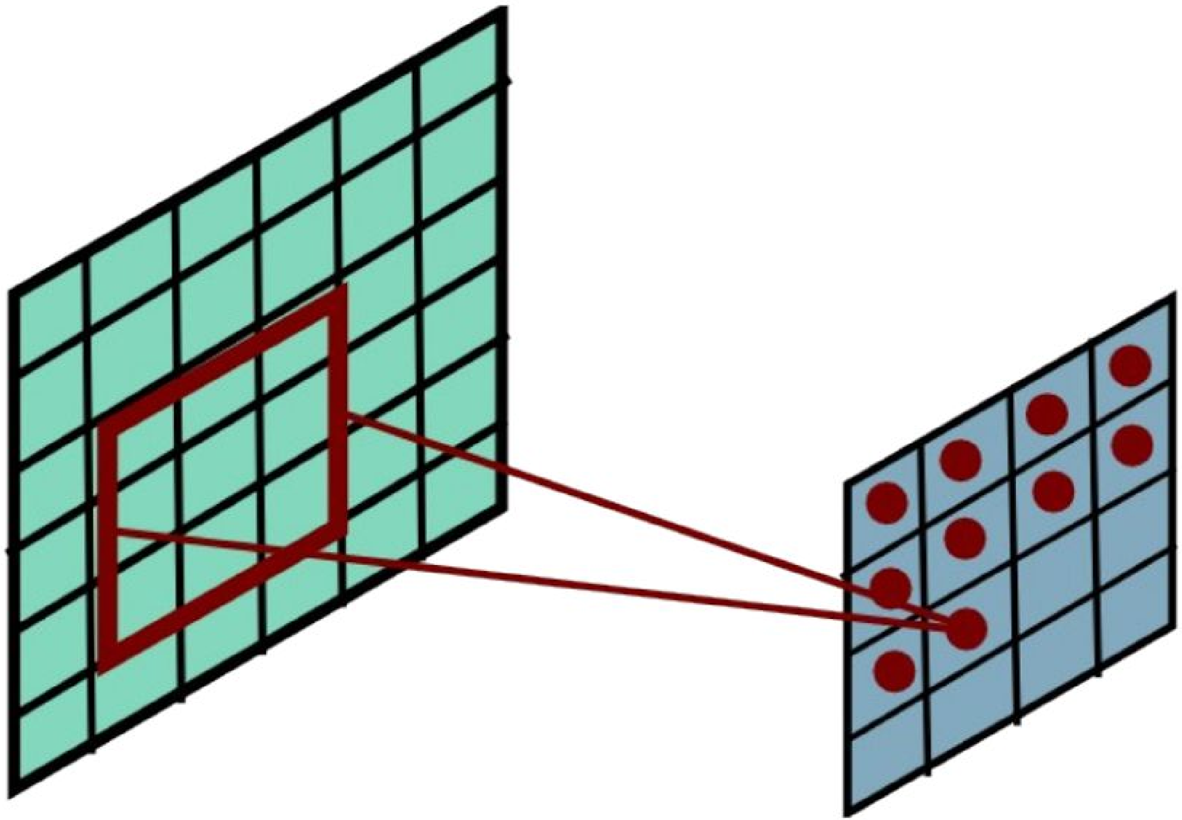
Convolution layer



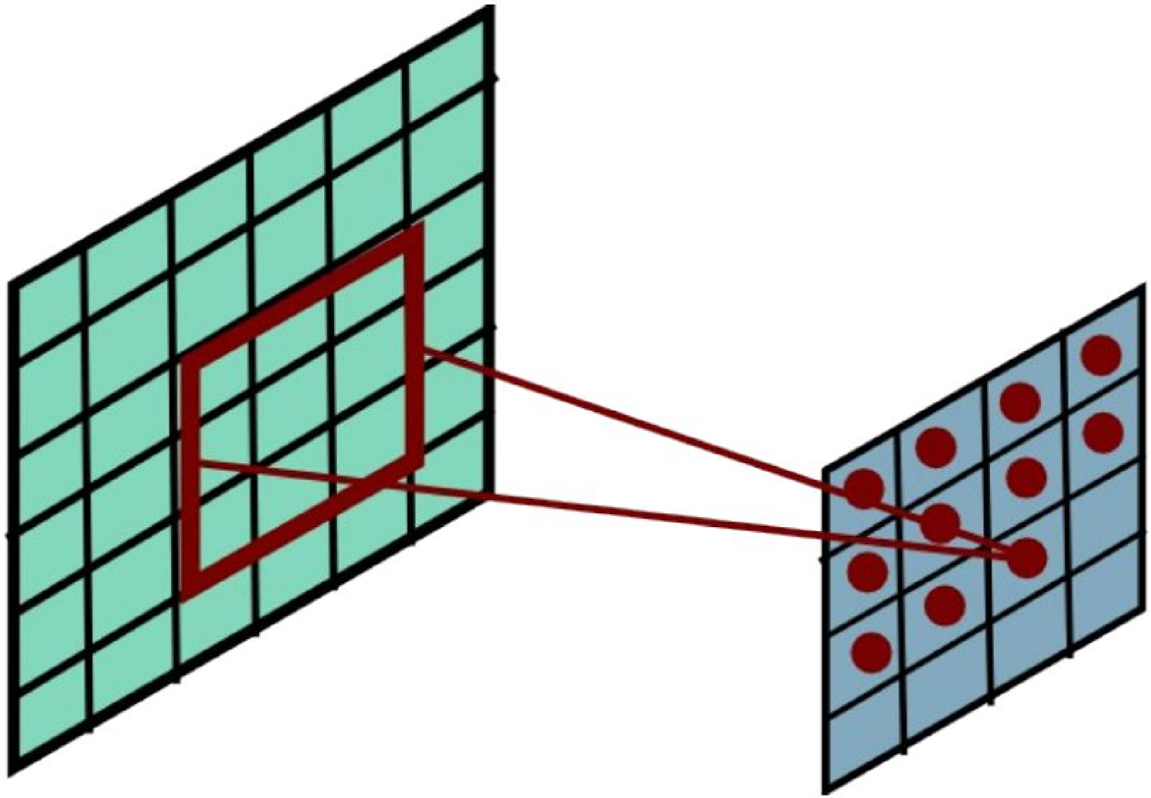
Convolution layer



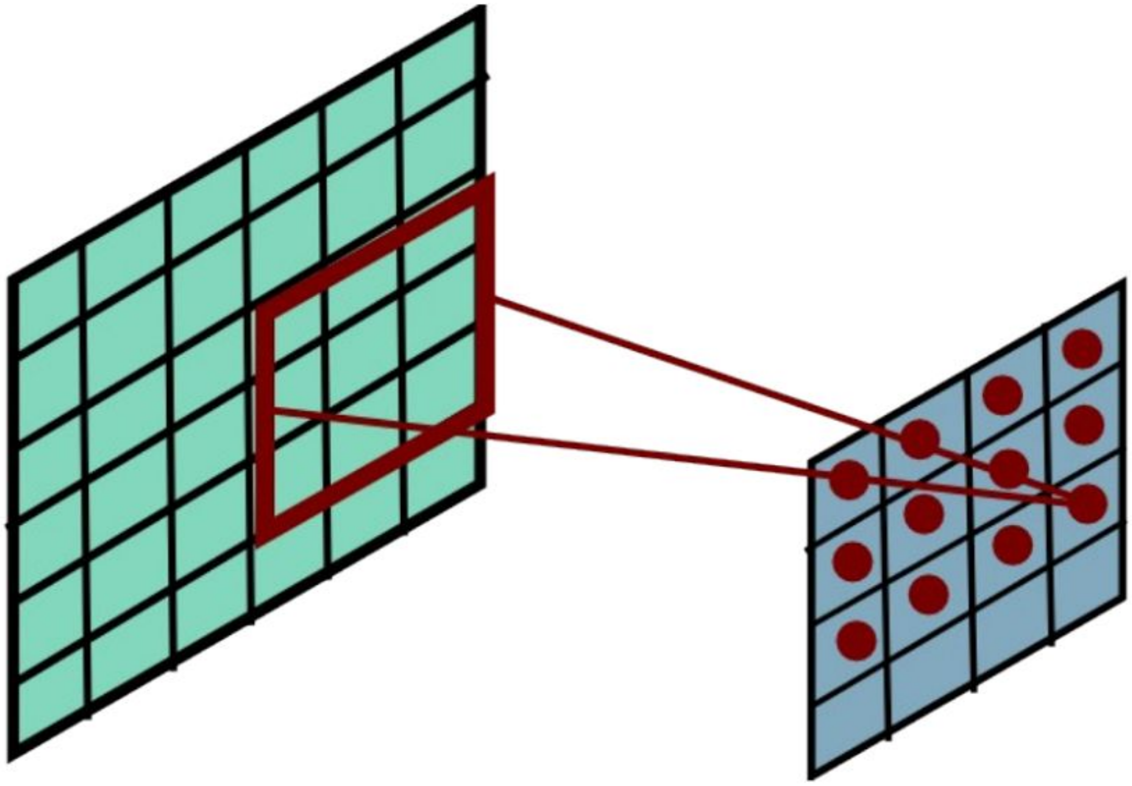
Convolution layer



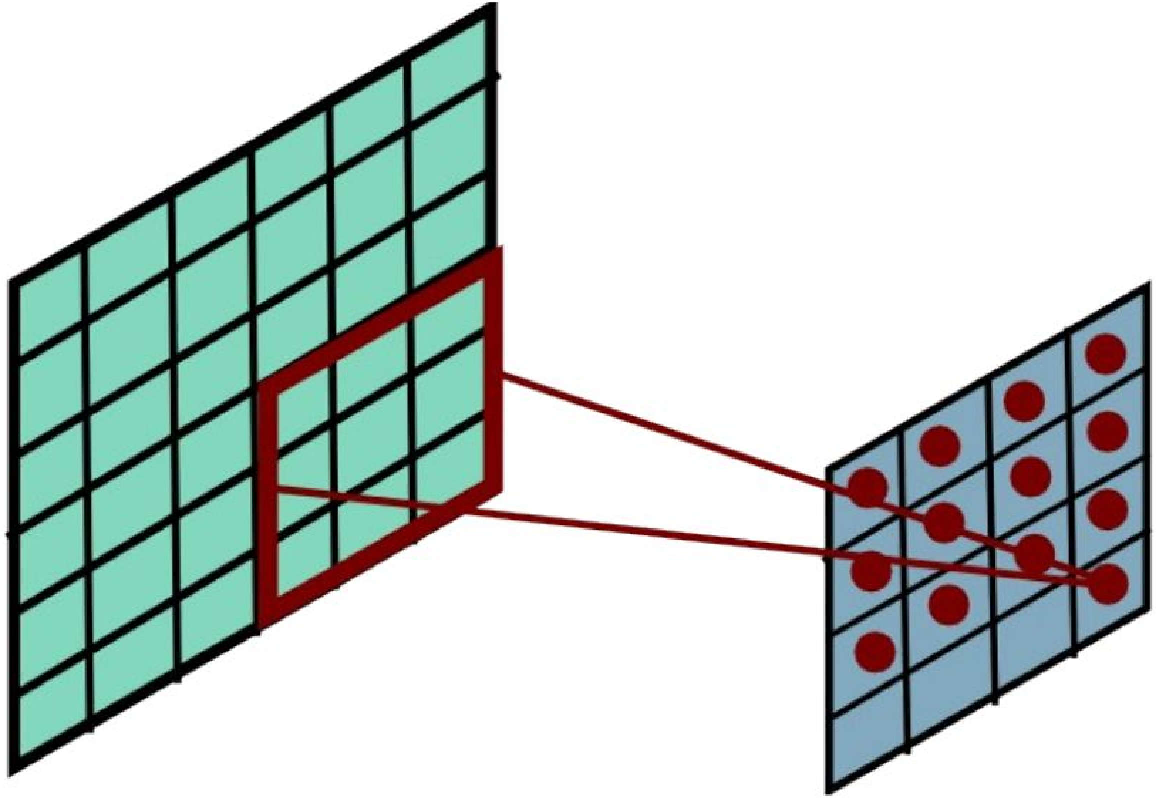
Convolution layer



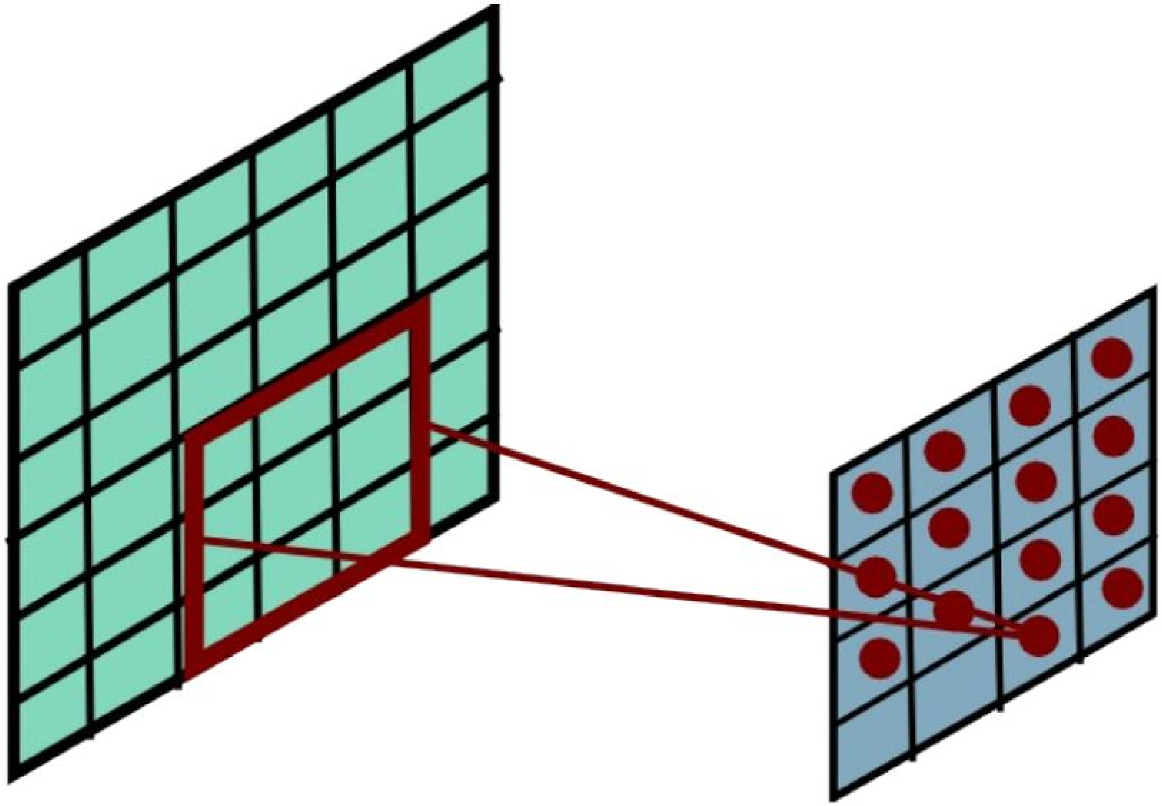
Convolution layer



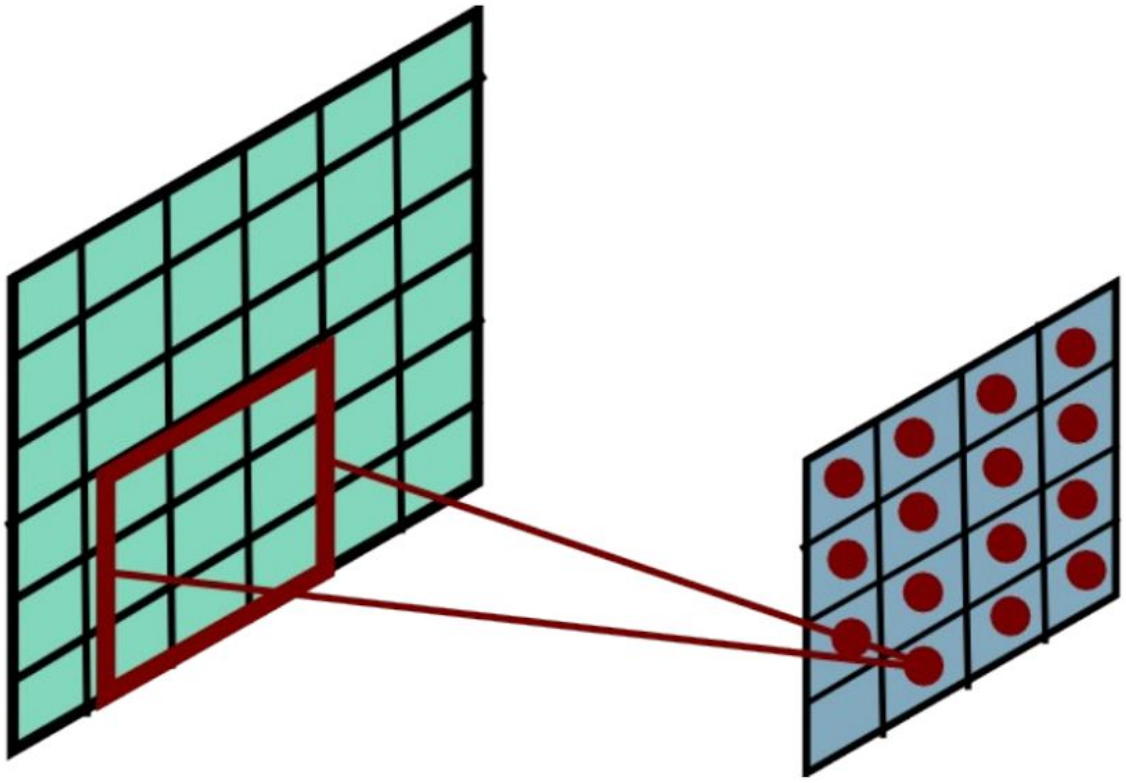
Convolution layer



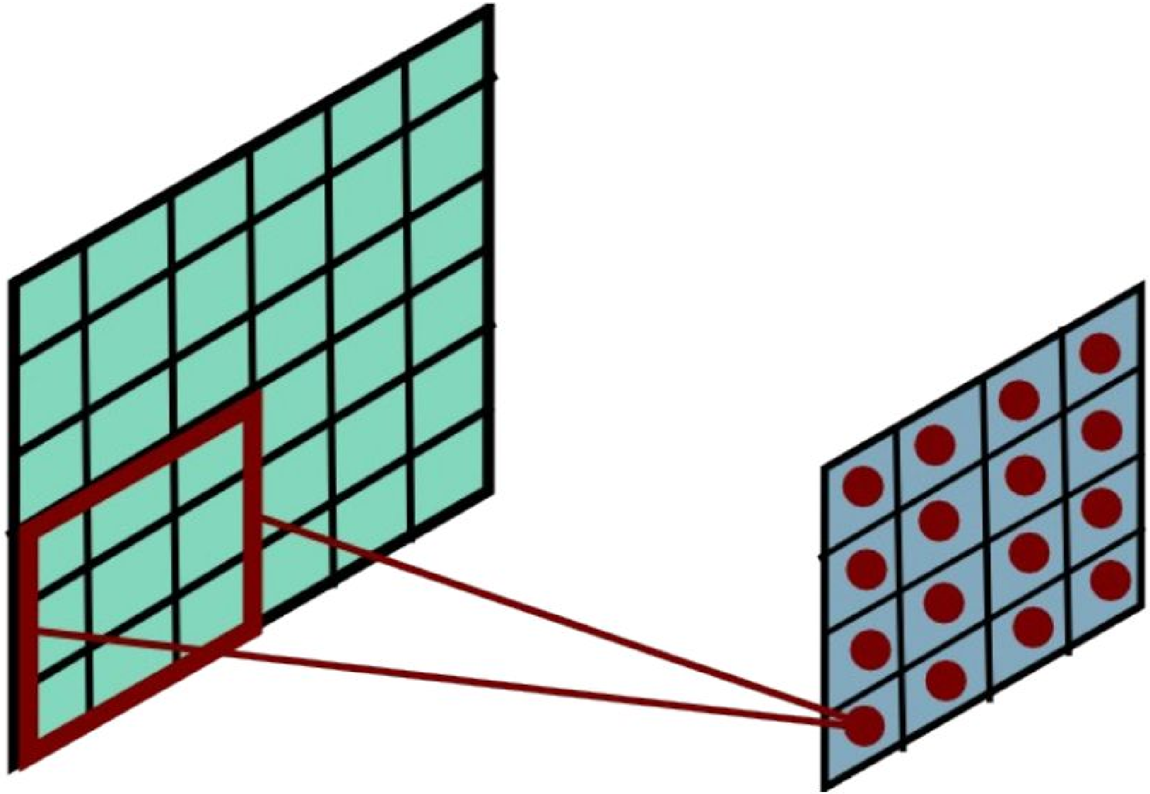
Convolution layer



Convolution layer

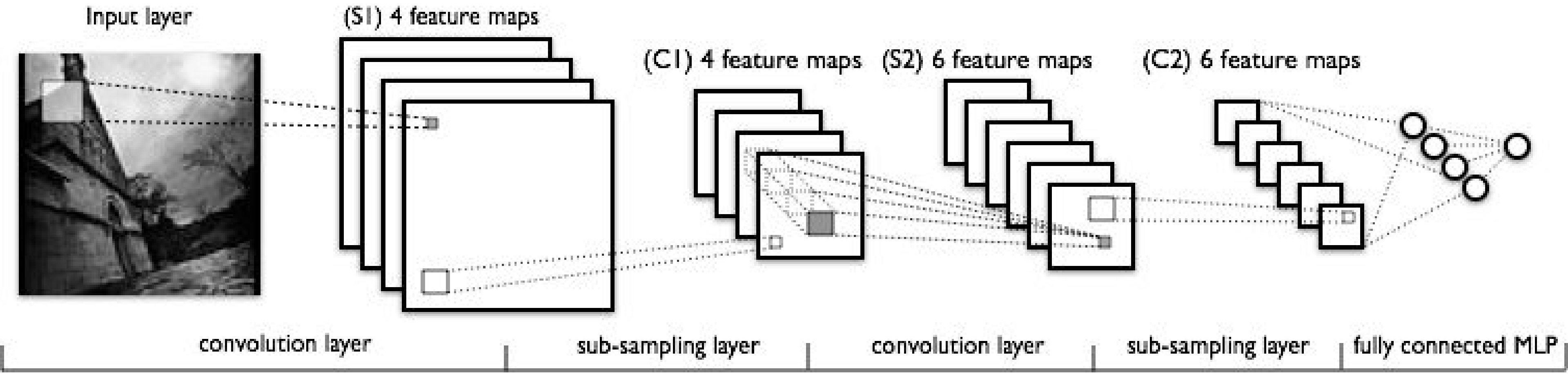


Convolution layer



Convolutional neural networks

- on the very small scale every piece of image can be processed the same way



ImageNet classification results 2012

1M training images, 1K categories, top-5 error

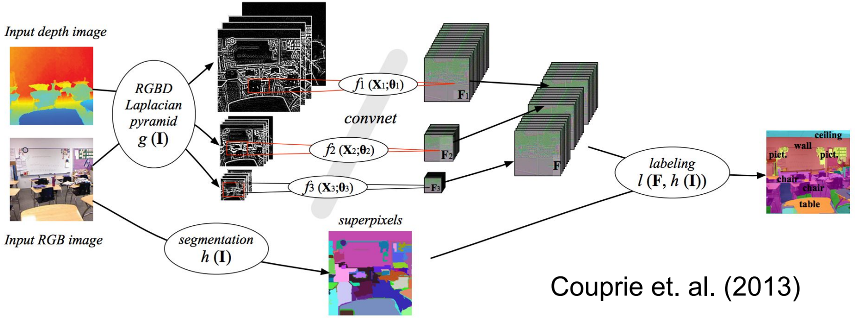
Human performance	~3-5%
Deep-learning models	~15%
Non-deep learning models ISI, Japan Oxford, England INRIA, France University of Amsterdam, etc.	~26%

ImageNet classification results 2015

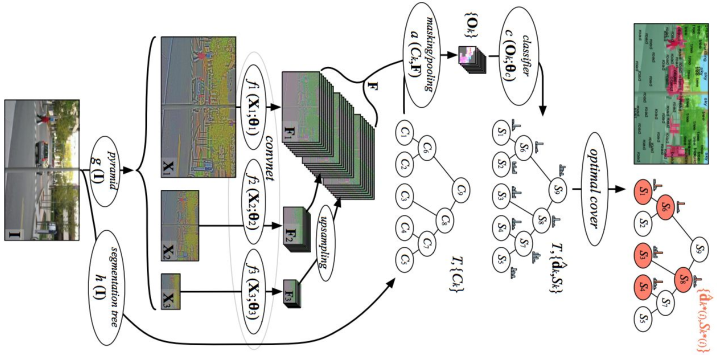
1M training images, 1K categories, top-5 error

Human performance	~3-5%
Deep-learning models	~4.5%
Non-deep learning models ISI, Japan Oxford, England INRIA, France University of Amsterdam, etc.	~26%

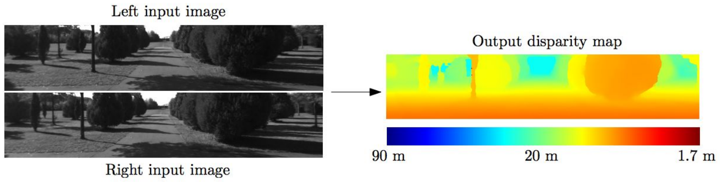
Segmentation and Stereo

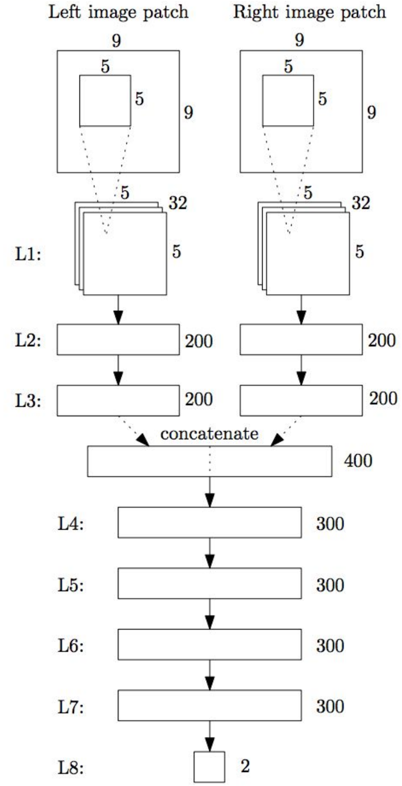


Couprie et. al. (2013)



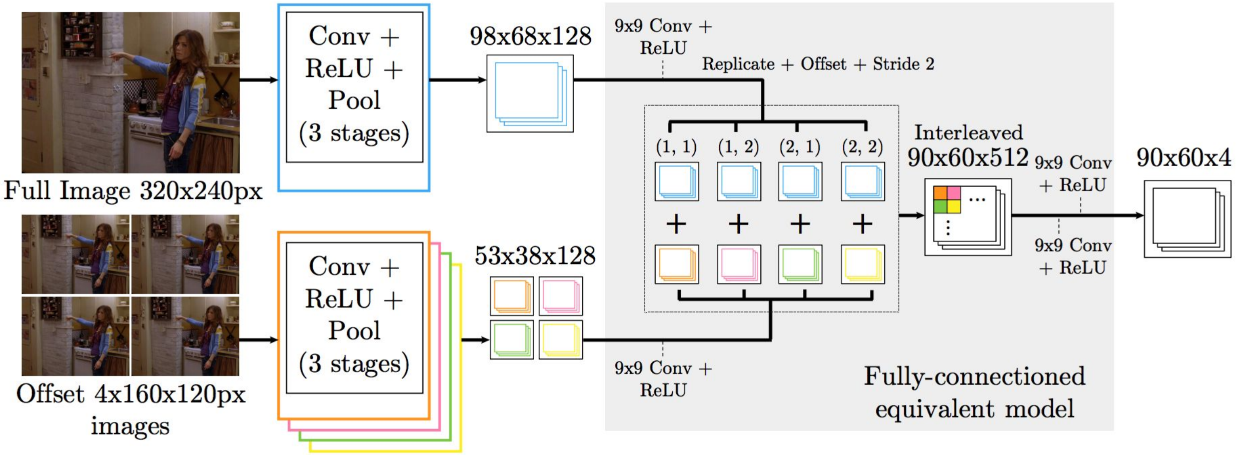
Farabet et. al. (2012)

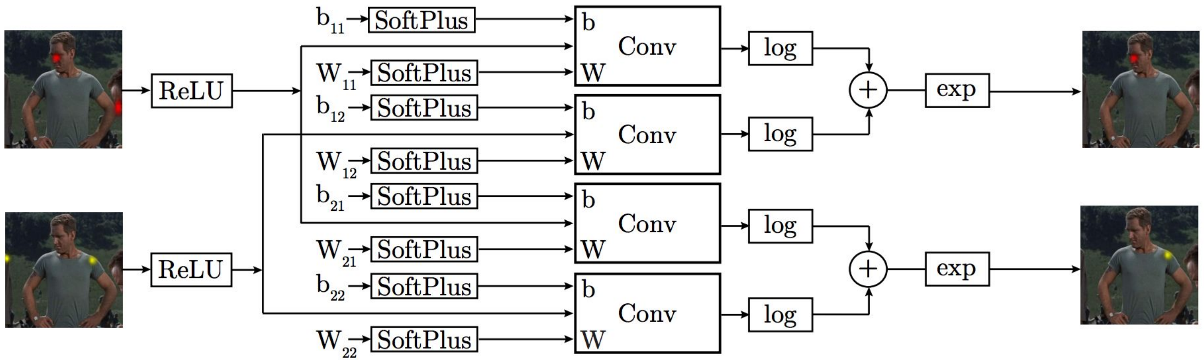


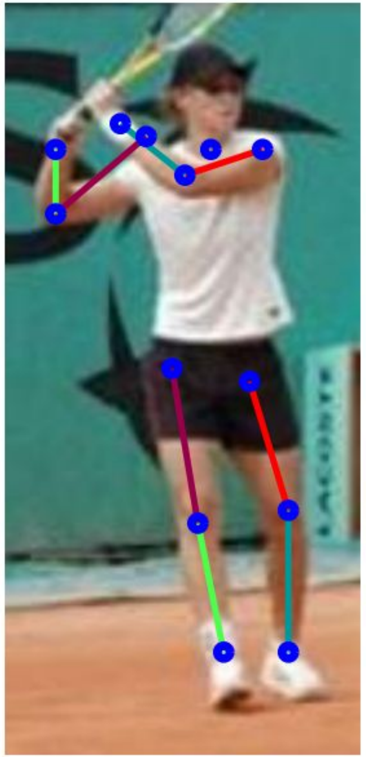


LeCun (2014)

ConvNets + Graphical Model (Tompson et. al. 2014)

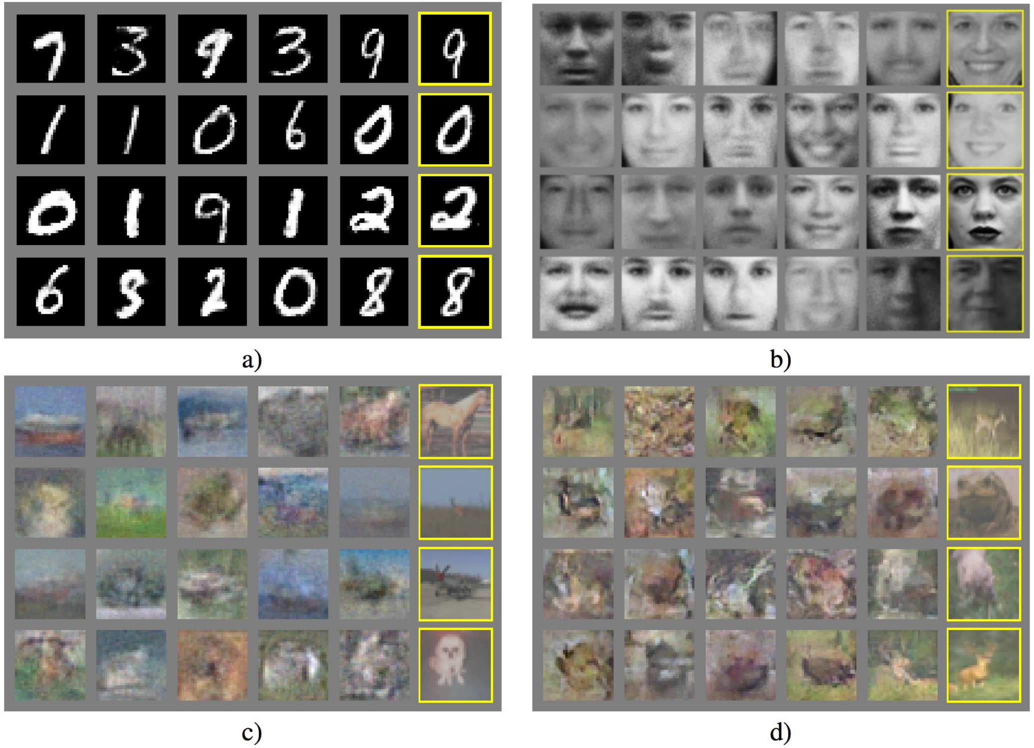




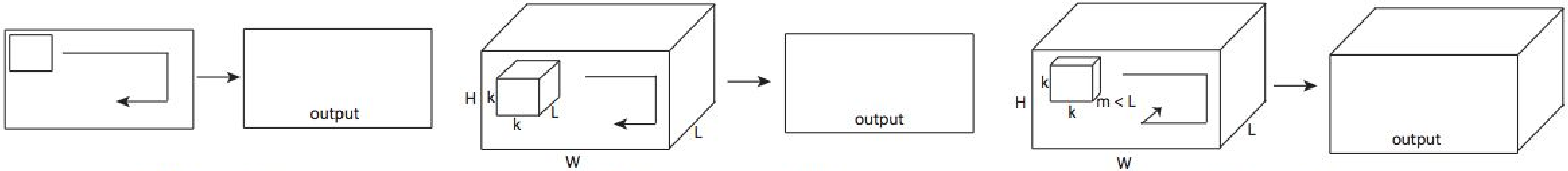


Generative Adversarial Nets

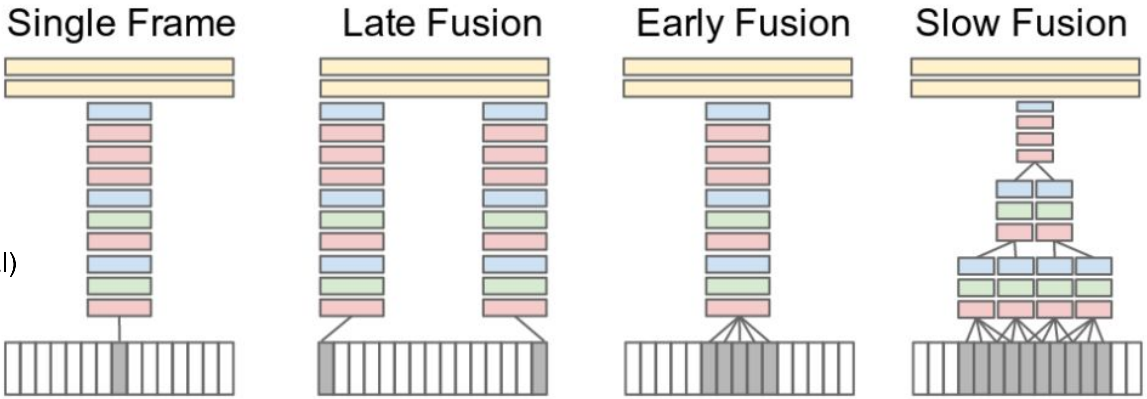
Goodfellow et. al. (2014)



ConvNets for Video

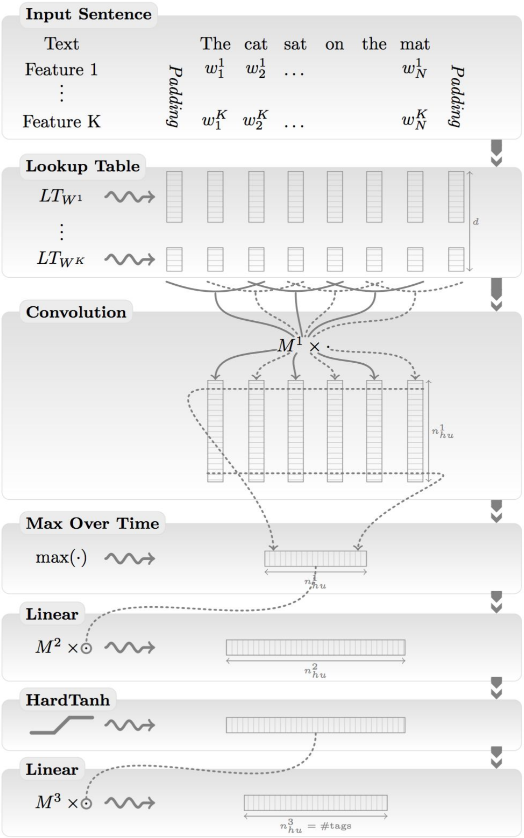


C3D (Tran et. al.) (a) 2D convolution (b) 2D convolution on multiple frames (c) 3D convolution

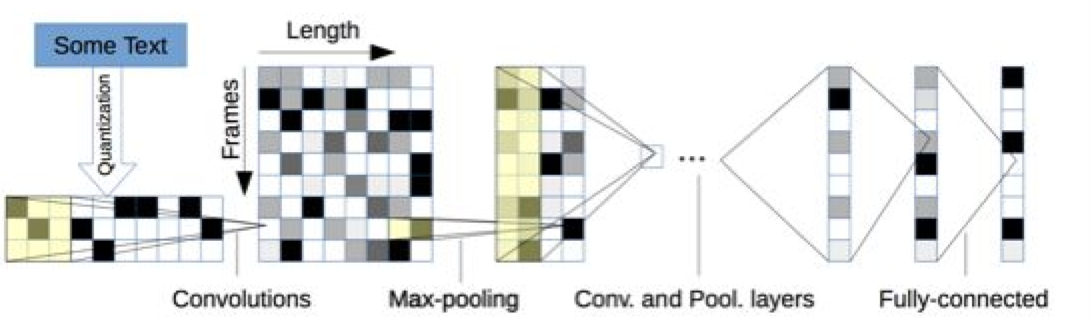


DeepVideo (Karpathy et. al)

ConvNets for NLP



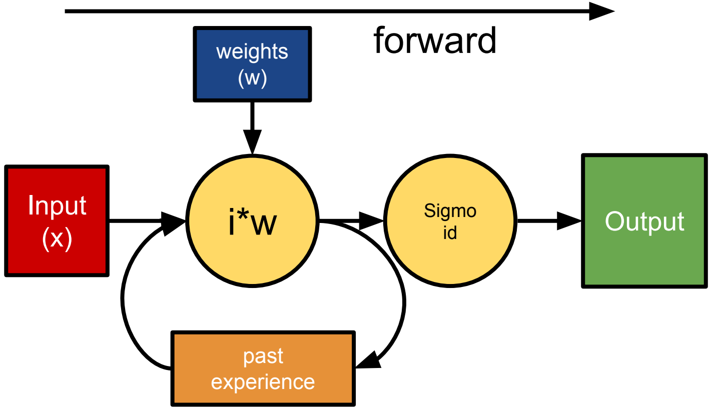
Collobert et. al.

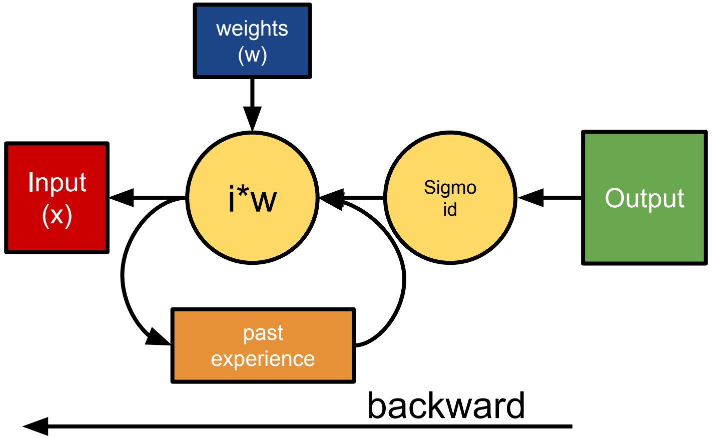


Zhang et. al.

Weird trick #2: Recurrent Nets

Recurrent Networks

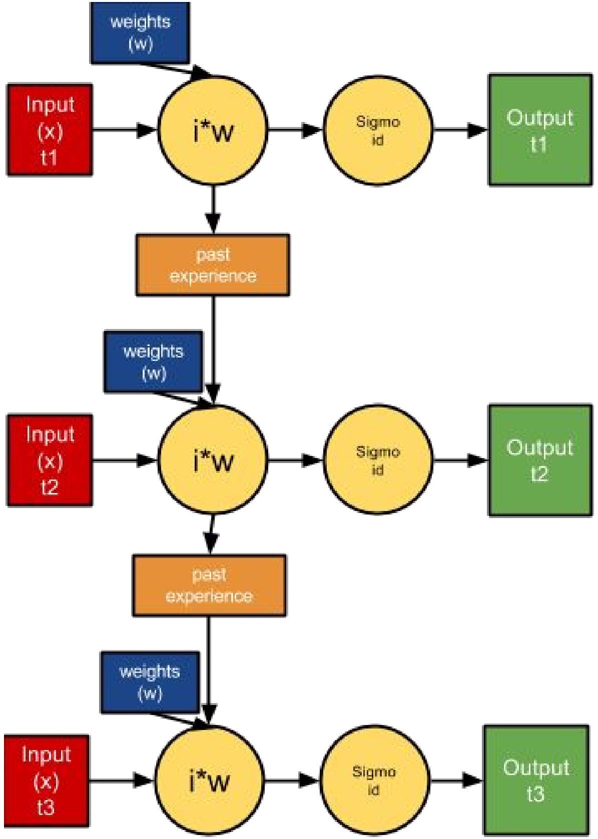




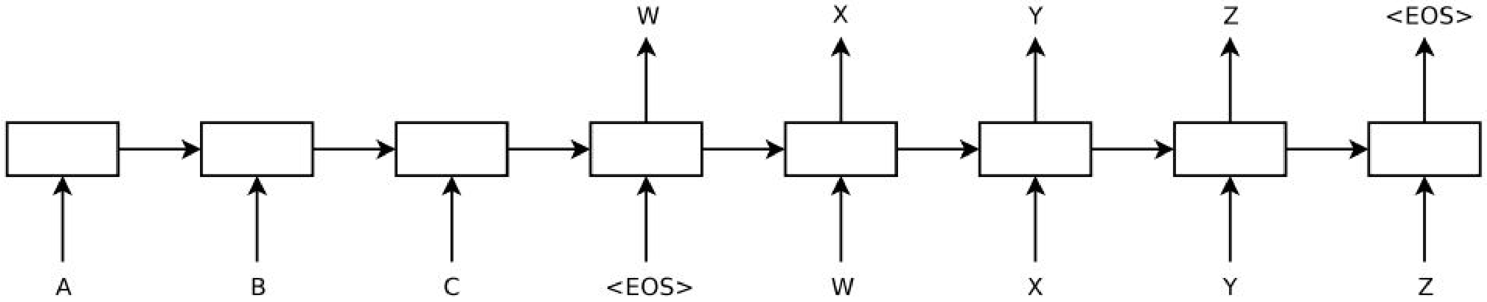
???

Acyclic computational graphs

Unfolding in time

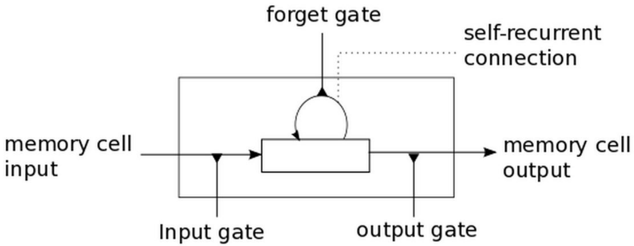


RNN-LSTMs



Sutskever et. al. (2014)

- Machine Translation
- Language Modeling
- Learning to execute (Python programs)



Basic LSTM unit (figure from deeplearning.net)

Examples

Input:

```
i=8827
c=(i-5347)
print((c+8704) if 2641<8500 else
    5308)
```

Target: 12184.

Input:

```
j=8584
for x in range(8):
  j+=920
b=(1500+j)
print((b+7567))
```

Target: 25011.

Sequence of character on the input and on the output.

Recurrent Nets for Q&A

Bilbo travelled to the cave. Gollum dropped the ring there. Bilbo took the ring.
Bilbo went back to the Shire. Bilbo left the ring there. Frodo got the ring.
Frodo journeyed to Mount-Doom. Frodo dropped the ring there. Sauron died.
Frodo went back to the Shire. Bilbo travelled to the Grey-havens. The End.
Where is the ring? A: Mount-Doom
Where is Bilbo now? A: Grey-havens
Where is Frodo now? A: Shire

Weston et. al. 2014
Facebook AI Research

Implementation and Engineering

Implementation: FLOP Eaters

- Convolutions are expensive
- Sequential Processing
- Matrix multiplies are expensive

Implementation: FLOP Eaters

- Convolutions are expensive
- Sequential Processing
- Matrix multiplies are expensive

GPUs

Training

- Days to weeks
- Engineering feat
- Multiple machines, multiple GPUs

Frameworks

- Torch
- Caffe
- Theano, Keras, Lasagne
- Common Features
  - syntax to define graph
  - every node in graph has derivative
  - train via backpropagation

```
features = nn.Sequential()
features:add(cudnn.SpatialConvolution(3, 96, 11, 11, 4, 4))
features:add(cudnn.ReLU(true))
features:add(cudnn.SpatialMaxPooling(2, 2, 2, 2))
features:add(cudnn.SpatialConvolution(96, 256, 5, 5, 1, 1))
features:add(cudnn.ReLU(true))
features:add(cudnn.SpatialMaxPooling(2, 2, 2, 2))

classifier = nn.Sequential()
classifier:add(nn.View(1024*5*5))
classifier:add(nn.Dropout(0.5))
classifier:add(nn.Linear(1024*5*5, 3072))
classifier:add(nn.Threshold(0, 1e-6))
classifier:add(nn.Linear(4096, nClasses))
classifier:add(nn.LogSoftMax())
model = nn.Sequential():add(features):add(classifier)
```

Trends

- Deeper nets
- Smaller convolutions
- RNN + LSTM
- Multiple GPUs + Multiple Machines
- Neural Machines
- Other kinds of memory units
- Better weight initialization
- Meta-problems

Challenges

- Scaling up for big data (videos, social networks etc.)
- Discrete optimization
- Memory that works

Questions