Neural Networks: Convolutional Neural Networks

Andrzej Kordecki

Neural Networks (ML.ANK385 and ML.EM05): Lecture 11
Division of Theory of Machines and Robots
Institute of Aeronautics and Applied Mechanics
Faculty of Power and Aeronautical Engineering
Warsaw University of Technology

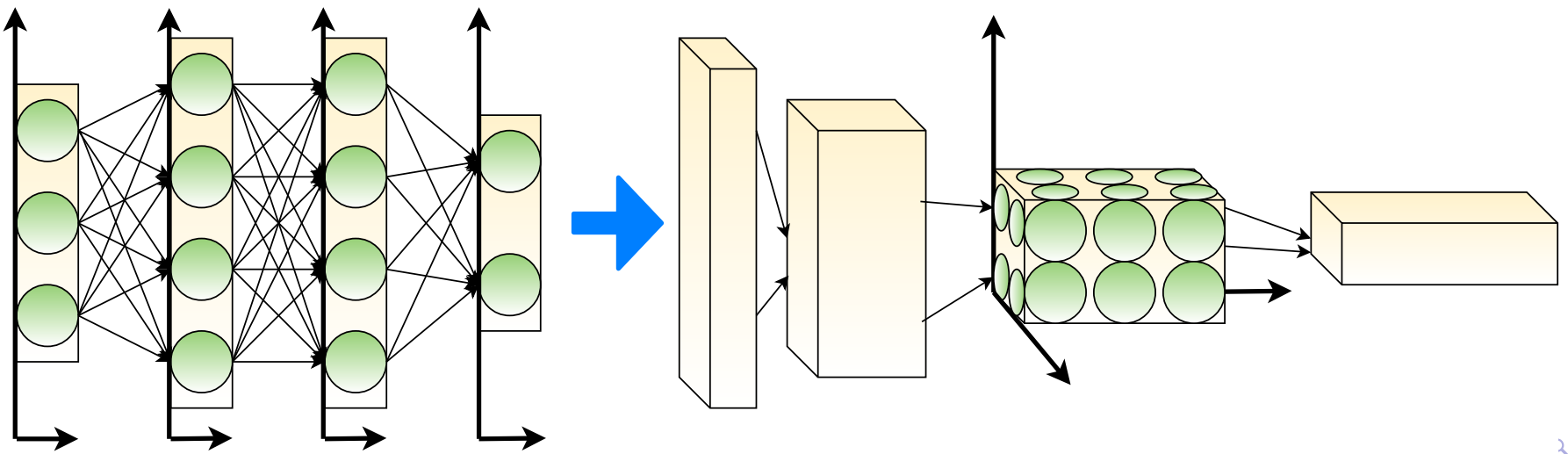
Table of Contents

Convolutional Neural Networks

Convolutional Neural Networks

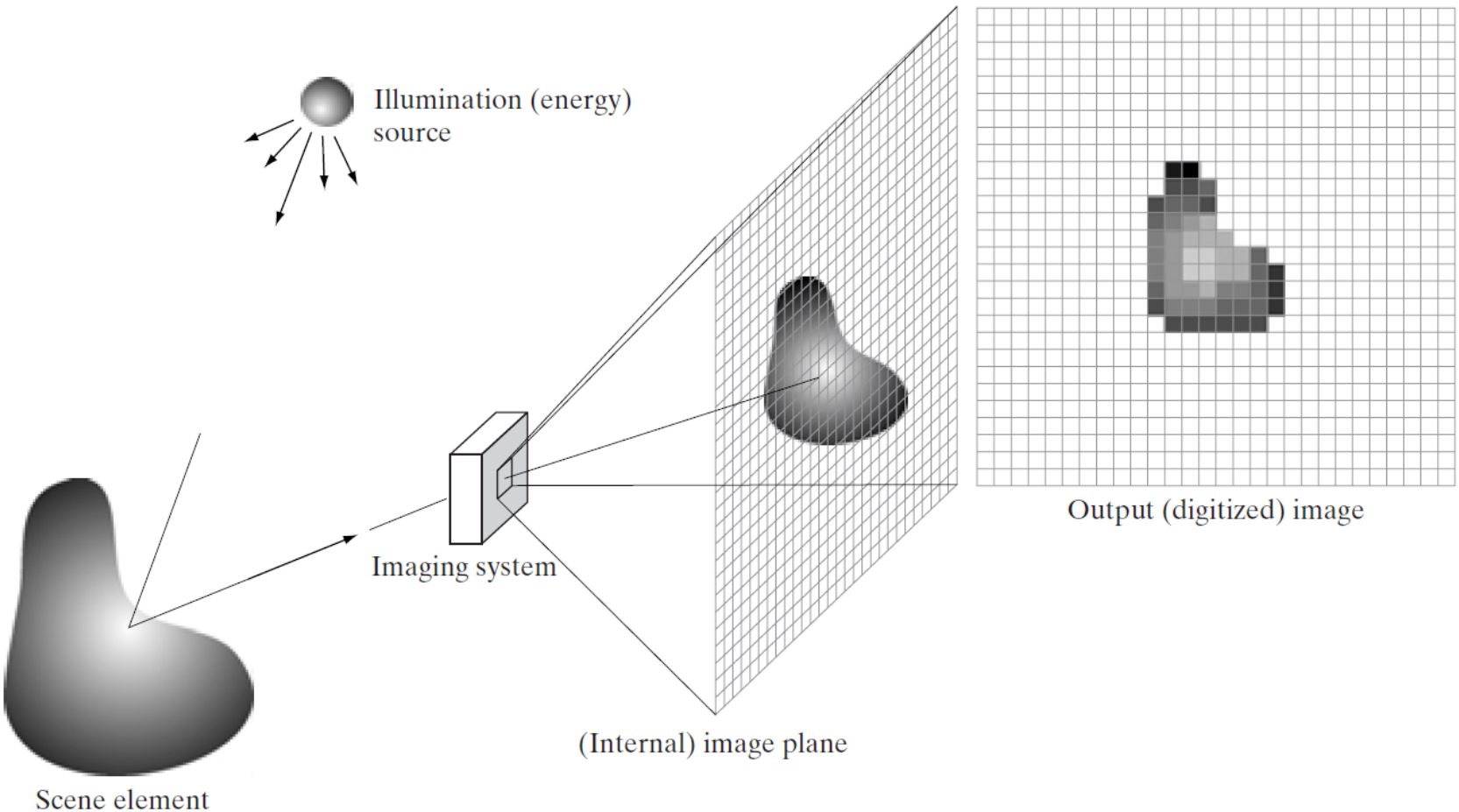
Convolutional networks convolutional (CNN, ConvNet or Conv) are a specialized kind of neural network for processing data that has a known, grid-like topology. Examples include time-series data, which can be thought of as a 1D grid taking samples at regular time intervals, and image data, which can be thought of as a 2D grid of pixels.

Regular neural network $\Rightarrow$ Convolutional neural network



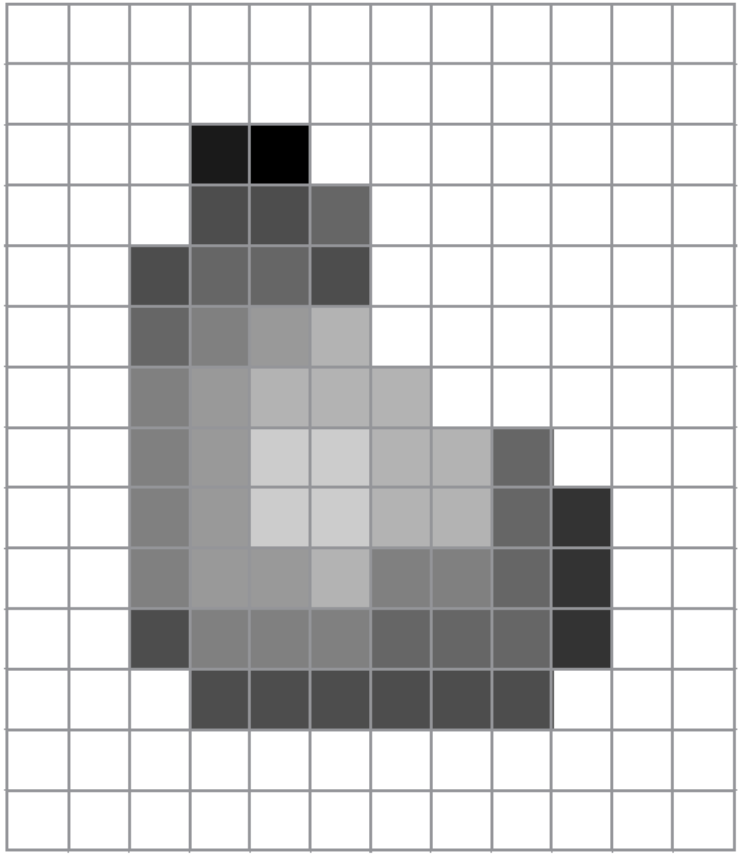
What is an image?

We will focused on image processing tasks.



What is an image?

A two-dimensional grid (matrix) of intensity values (sampling and quantization).



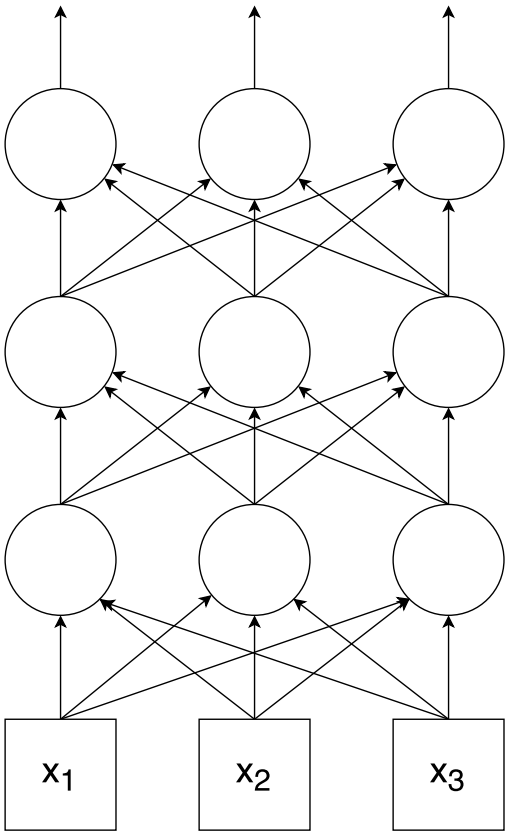
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

What is an image?

Neural networks in image processing:

- Network should exhibit invariance to translation, scaling and elastic deformations (depend on training set),
- Nearby pixels (local pixels) are more strongly correlated than distant ones,
- Information can be merged at later stages to get higher order features.

Limitations of Neural Networks



All neurons are connected to all neurons of the previous layer, as well as all neurons of the next layer.

$$y_k^{(N)} = f(\sum_j y_j^{(N-1)} w_{j,k}^{(N)})$$
$$= f(\sum_j f(\sum_i y_j^{(N-2)} w_{i,j}^{(N-1)}) w_{j,k}^{(N)})$$
$$= \ldots =$$
$$f(\sum_j f(\sum \sum \ldots \sum x_i w_{i,j}^{(1)}) w^{(1)} \ldots w_{j,k}^{(N)})$$

Limitations of Neural Networks

Limitations of Neural Network:

- Need substantial number of training samples (some network architectures need less data, but still quiet a lot),
- Slow learning (convergence take time),
- Inadequate parameter selection techniques that lead to poor minimum.

The architecture of a CNN is designed to take advantage of the 2D structure of an input image

What is Convolution?

The convolution operation is typically denoted with an asterisk:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{+\infty} x(a)w(t-a)$$

where: $s(t)$ - feature map, $x(t)$ - input, w - kernel (filter), t - time. For a two-dimensional image I as our input, we also want to use a two-dimensional kernel K:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n).$$

What is Convolution?

Input matrix

a	b	c
d	e	f
g	h	i
j	k	l

* Kernel

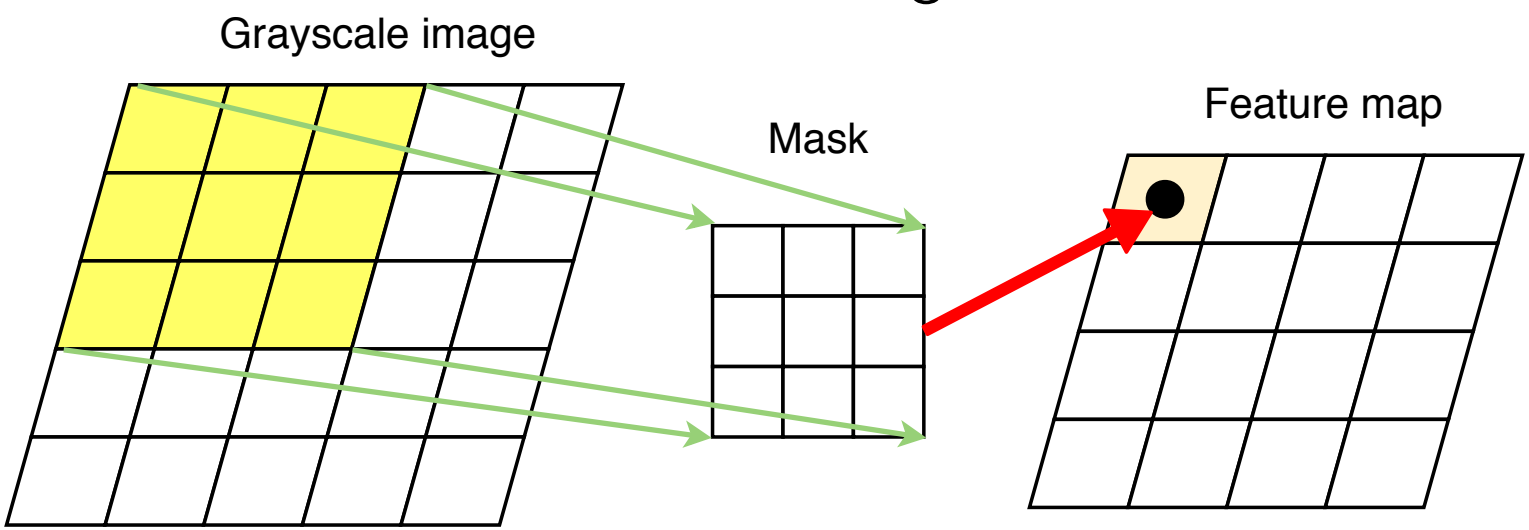
w_1	w_2
w_3	w_4

Origin

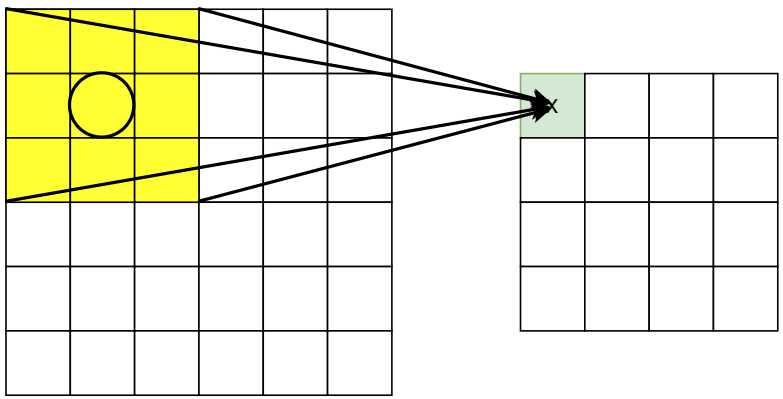
Output

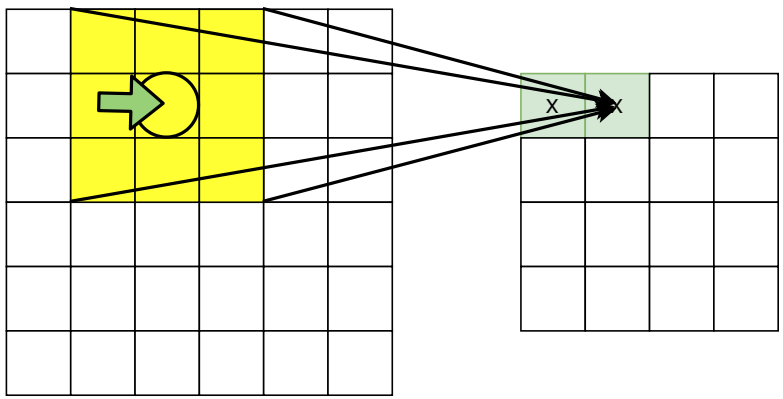
$w_1a+w_2b+w_3d+w_4e$	$w_1b+w_2c+w_3e+w_4f$
$w_1d+w_2e+w_3g+w_4h$	$w_1e+w_2f+w_3h+w_4i$
$w_1g+w_2h+w_3j+w_4k$	$w_1h+w_2i+w_3k+w_4l$

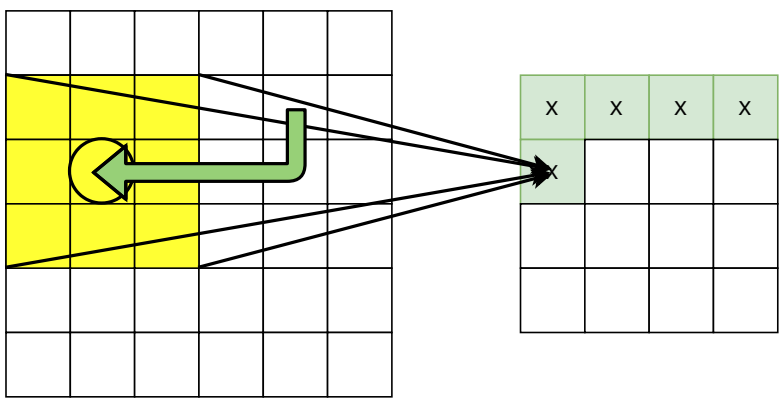
In case of images

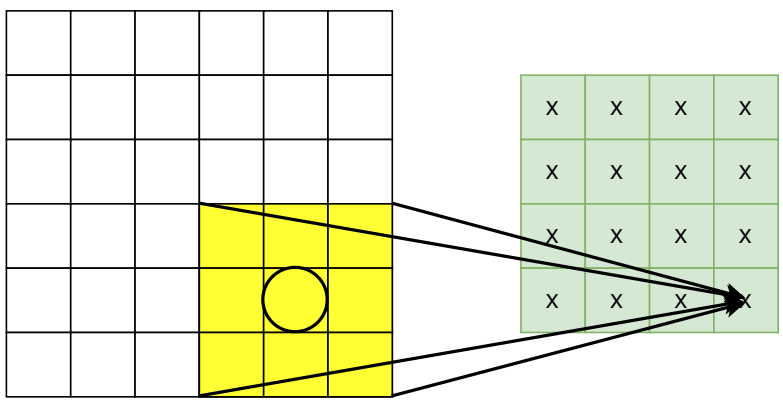


What is Convolution?









Example

What does this convolution kernel do?

0.01	0.08	0.01
0.08	0.62	0.08
0.01	0.08	0.01

Gaussian filter

Example

What does this convolution kernel do?

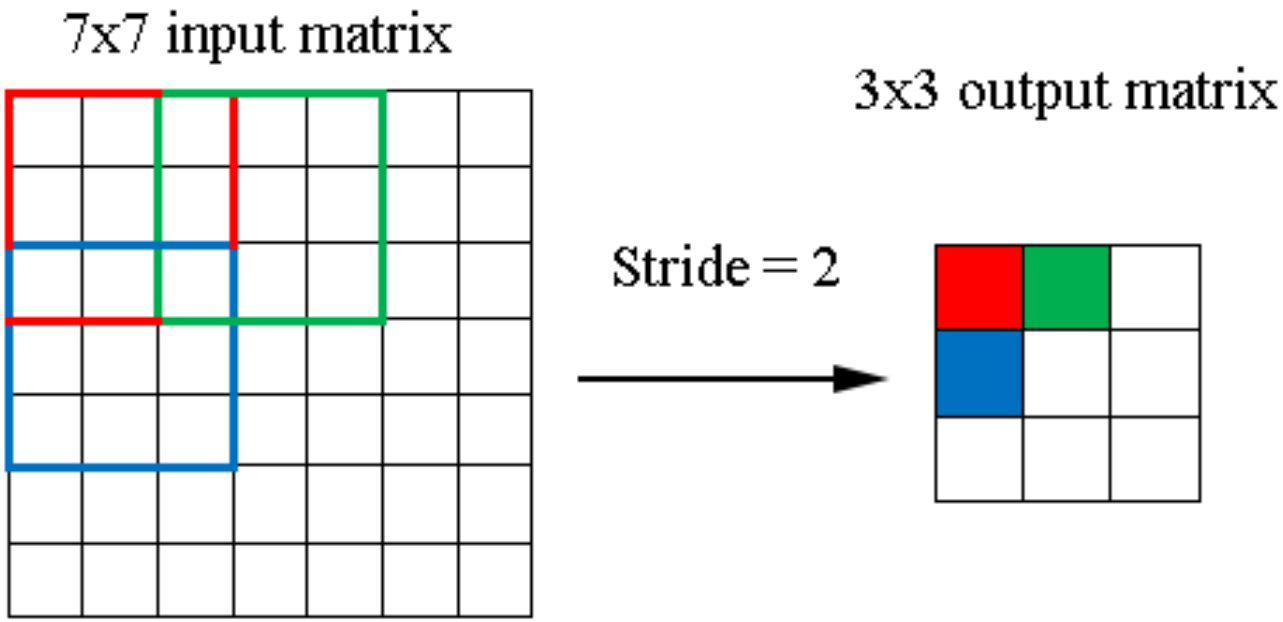
1	2	1
0	0	0
-1	-2	-1

Sobel filter

Stride and Padding

Stride controls how the filter convolves around the input matrix:

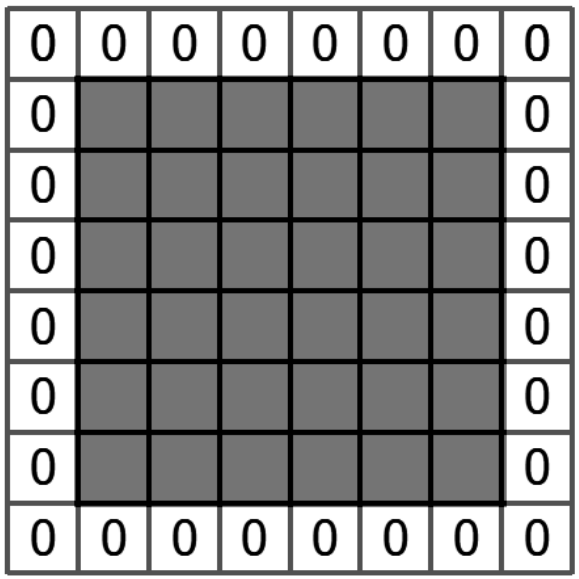
- The amount by which the filter shifts is the stride.
- When the stride is 1 then we move the filters one pixel at a time and if stride is 2 then the filters jump 2 pixels.



Stride and Padding

What happens when you apply 5×5 filter to a 32×32 input image? The output image would have size 28 x 28.

The spatial dimensions will decrease as long as we keep applying convolutions. To prevent this, we can add zeros around the border - we perform zero padding. Example of padding equal 1:



Convolutional Neural Networks

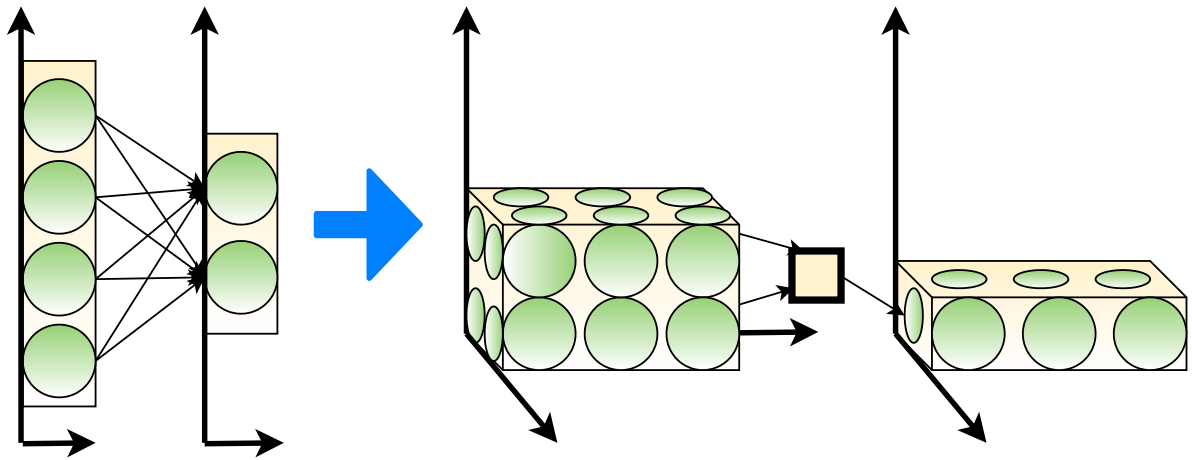
Definition:

Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

Convolution is a specialized kind of linear operation on two functions of a real-valued argument.

Convolutional Neural Networks

Convolution layer (CONV):



Unlike a regular Neural Network, the layers of a CNN have neurons arranged in 3 dimensions:

- width,
- height ,
- depth - number of filters.

Convolutional Neural Networks

CNN are very similar to ordinary Neural Networks:

- Made of neurons that have weights and biases,
- Each neuron receives some inputs and performs a dot product,
- Still have a loss function,
- Most of the rules developed for learning of regular Neural Networks still apply.

But, the use of convolution method makes them unique.

Convolution Improvements

The three improvements of convolution over regular NN:

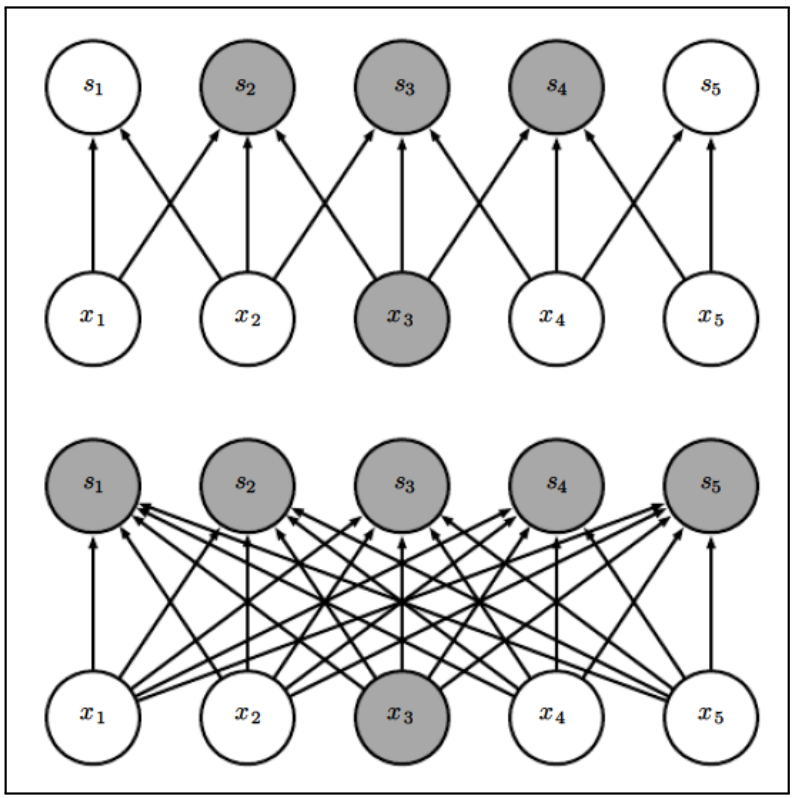
- sparse interactions,
- parameter sharing,
- equivariant representation.

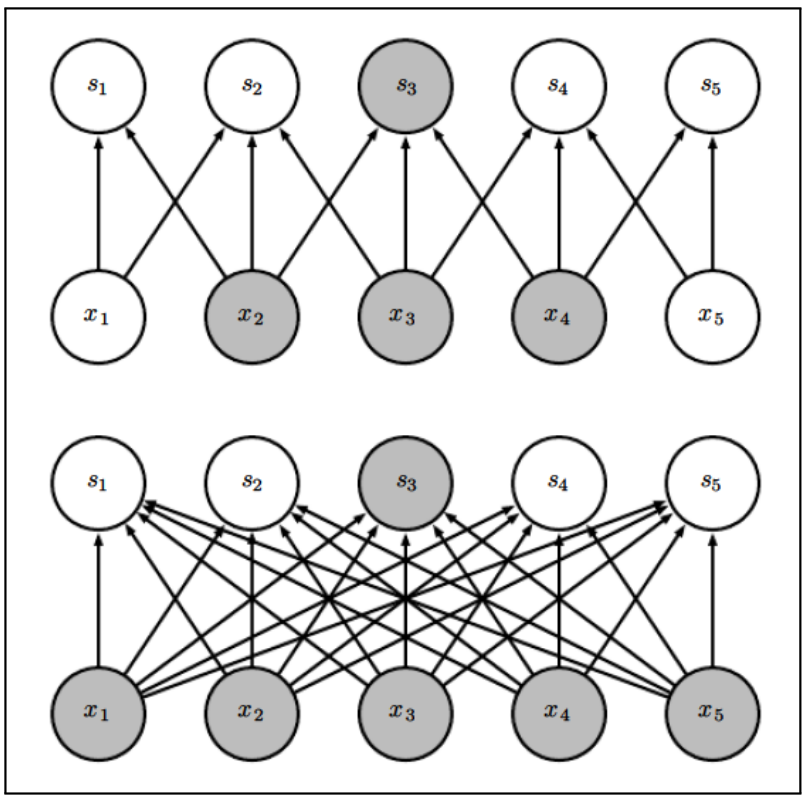
Sparse interactions

Sparse interactions:

- Number of parameters (weights) is defined by kernel size, which is smaller than the input size. We use less parameters, which improves statistical efficiency.
- If there are m inputs and n outputs, then matrix multiplication requires $m \times n$ parameters like in case of regular network. If we limit the number of connections each output may have to k like in case of CNN, then the sparsely connected approach requires only $k \times n$ parameters. It is possible to obtain good performance with $k << m$.

Sparse interactions



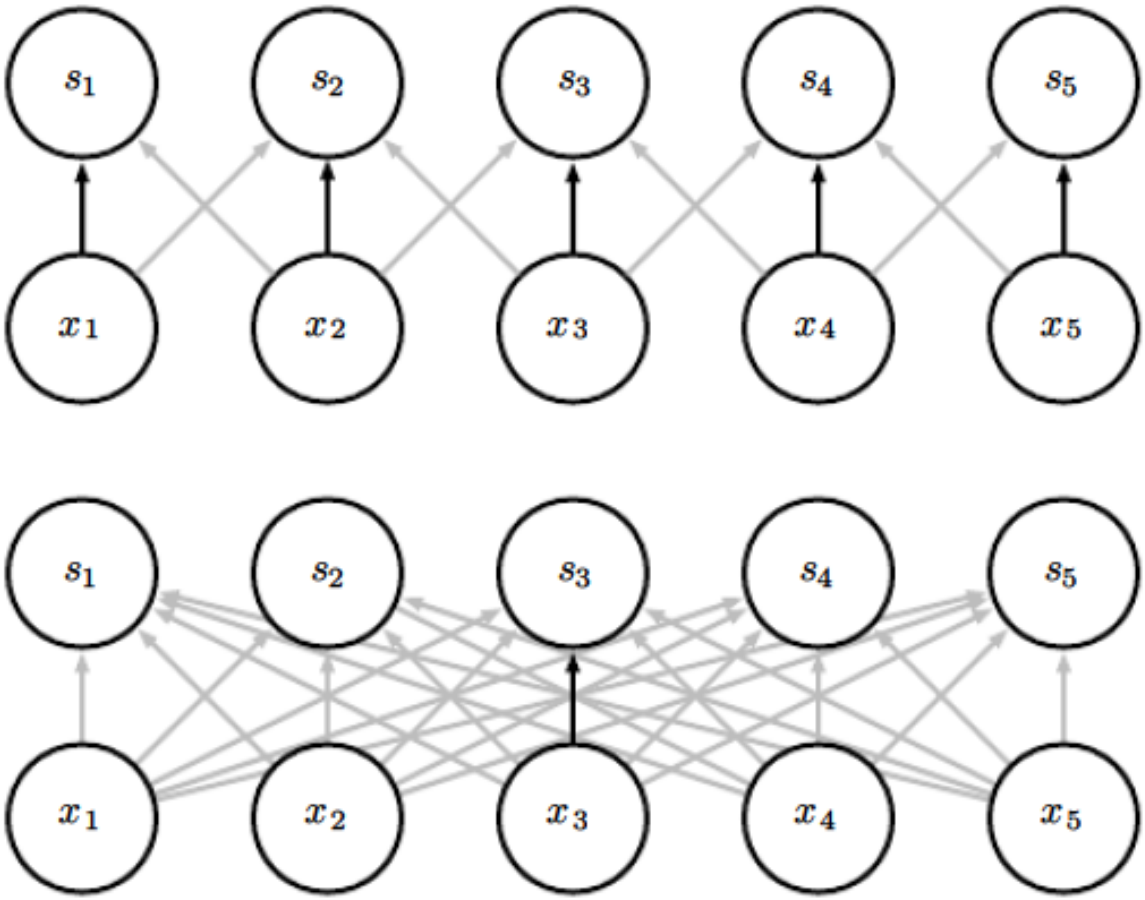


Parameter sharing

Parameter sharing:

- The same parameter is used in more than one function in a model. In a CNN, each member of the kernel parameters are used at every position of the input rather than learning a separate set of parameters for every location (we learn only one set).
- The runtime of forward propagation of CNN is $O(k \times n)$ and its further reduce the storage requirements of the model to k parameters. Convolution makes NN more efficient.

Parameter sharing



Equivariant representation

- The equivariant means that if the input changes, the output changes in the same way. We can used in simplification of mathematical operation.
- A function $f(x)$ is equivariant to a function g if:

$$f(g(x)) = g(f(x))$$

In the case of convolution, if we let g be any function that translates the input (i.e., shifts it) then the convolution function is equivariant to g.

Convolution summary

- **Question**: How many parameters regular neural network need for a very small 32x32x3 (32 wide, 32 high, 3 color channels) image?
  Answer: A fully-connected regular neural network in first hidden layer would have 32*32*3 = 3072 weights. This amount still seems manageable.
- **Question**: How many parameters regular neural network need for a small 200x200x3 image?
  Answer: It would have 200*200*3 = 120,000 weights. Clearly, this full connectivity have huge number of parameters that would quickly lead to overfitting.

CNN Architecture

CNN Architecture

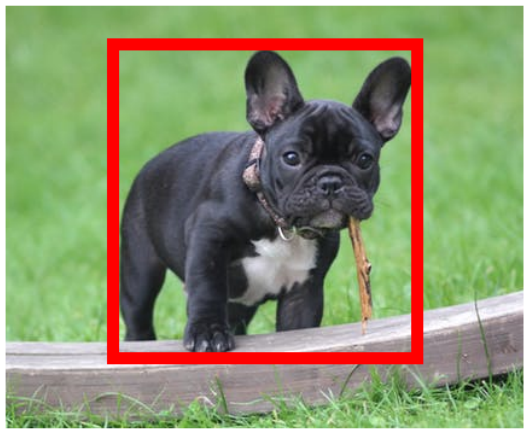
Image classification is one of the CNN typical tasks.

Classification

Dog

Single Object

Localization
+ classification

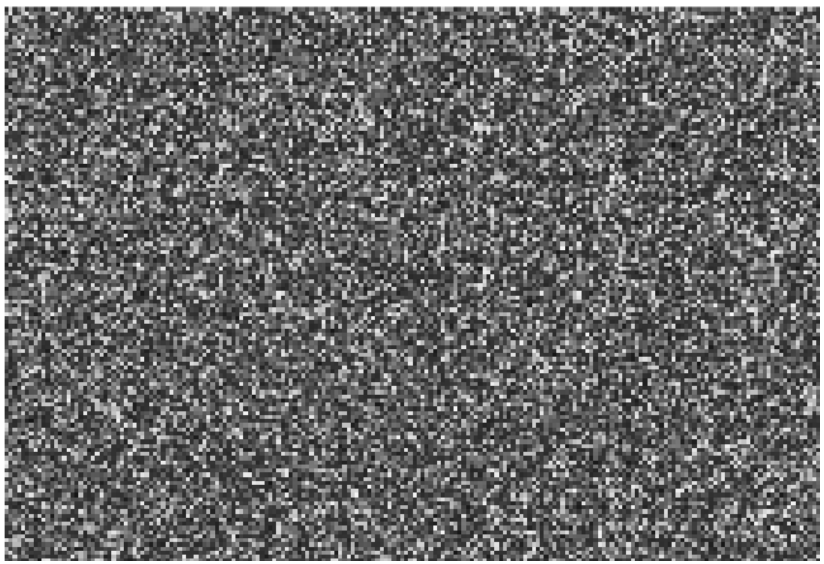


Dog

Single Object

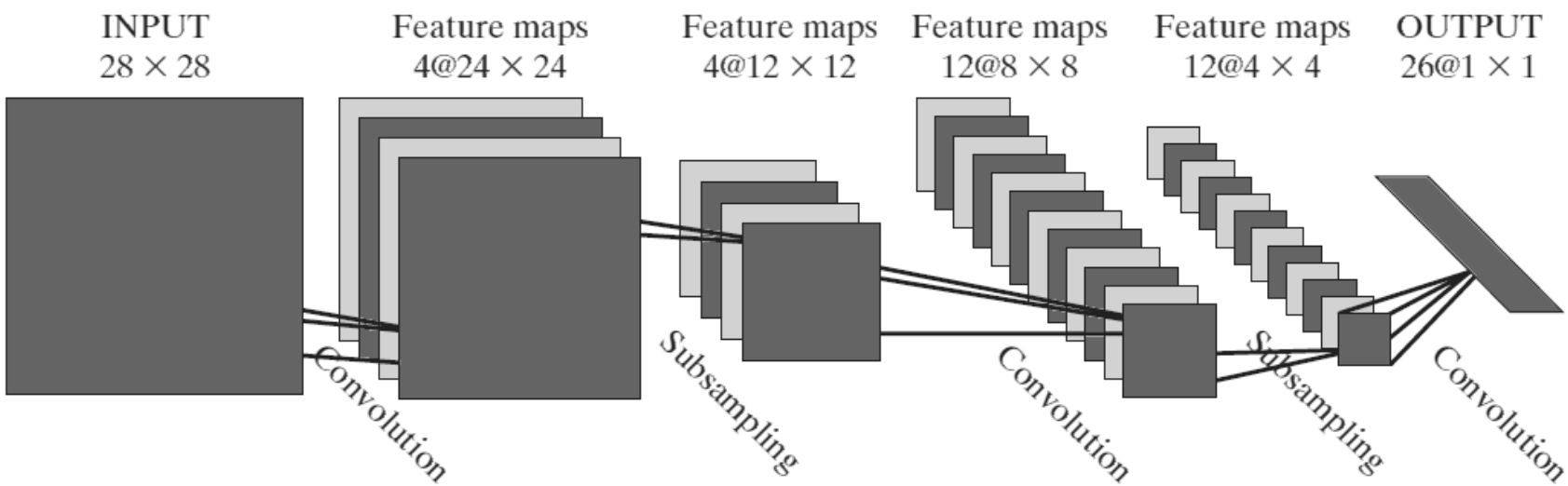
Convolutional Neural Networks

Can we use MLP for image classification? Is it possible to analysis image without use of local pixel neighborhood?



CNN Architecture

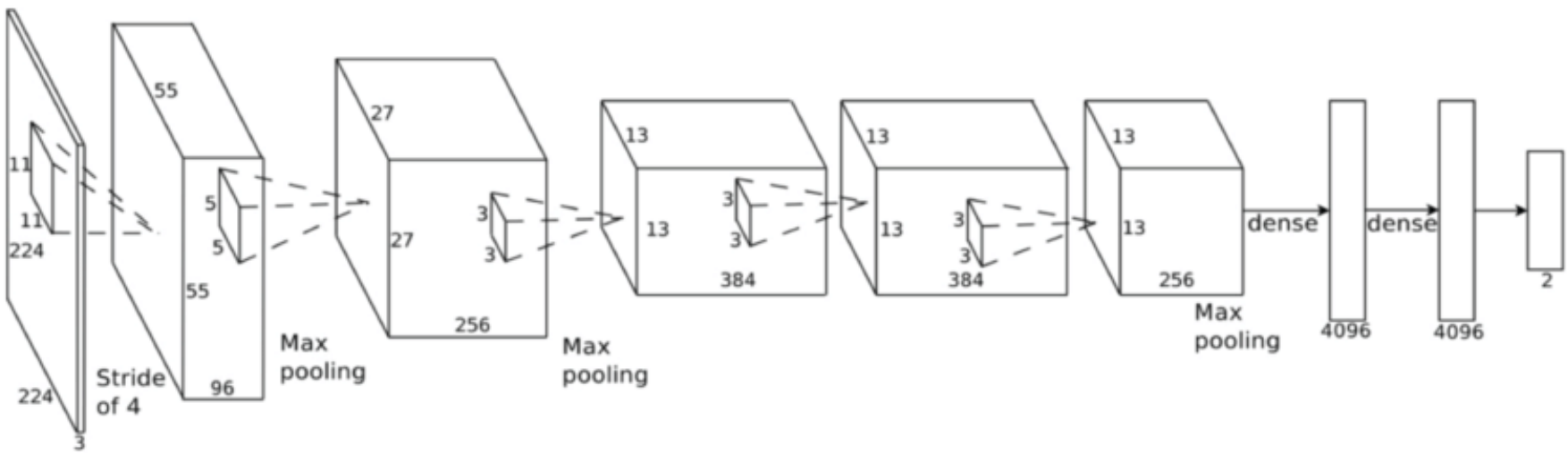
A convolutional network was designed specifically to recognize two-dimensional shapes **in images** with a high degree of invariance to translation, scaling, skewing, and others.
Classification CNN LeNet-5 architecture:



Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86(11): 2278–2324, 1998.

CNN Architecture

AlexNet is the winner of the ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2012.

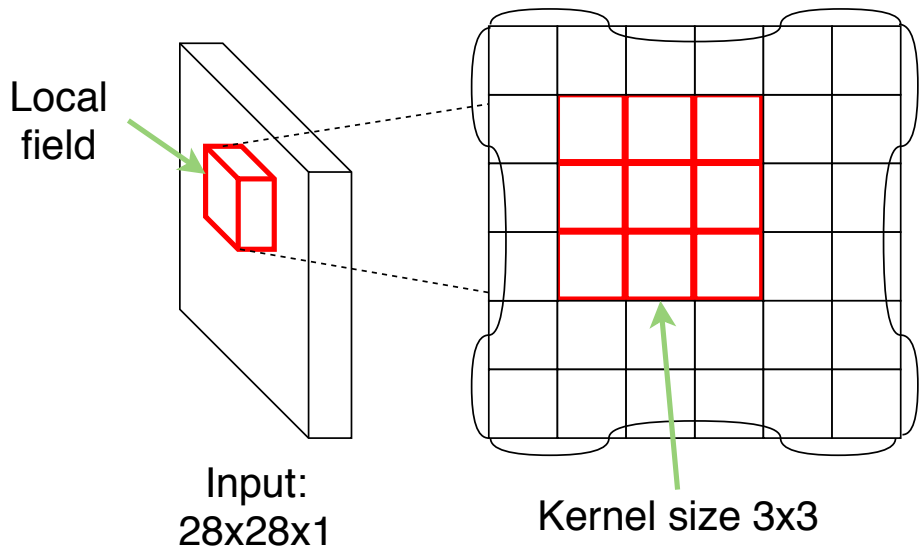


A Krizhevsky, I Sutskever, GE Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems, 1097-1105, 2012

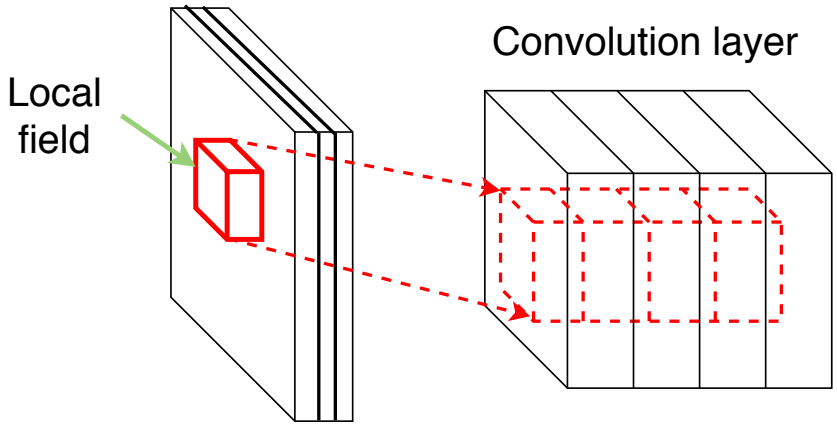
CNN Architecture

Features of the network:

- Feature extraction. Each neuron takes its inputs from a local receptive field (kernel) in the previous layer to extract local data features. The exact location of extracted feature is less important, so long as its position relative to other features is approximately preserved.



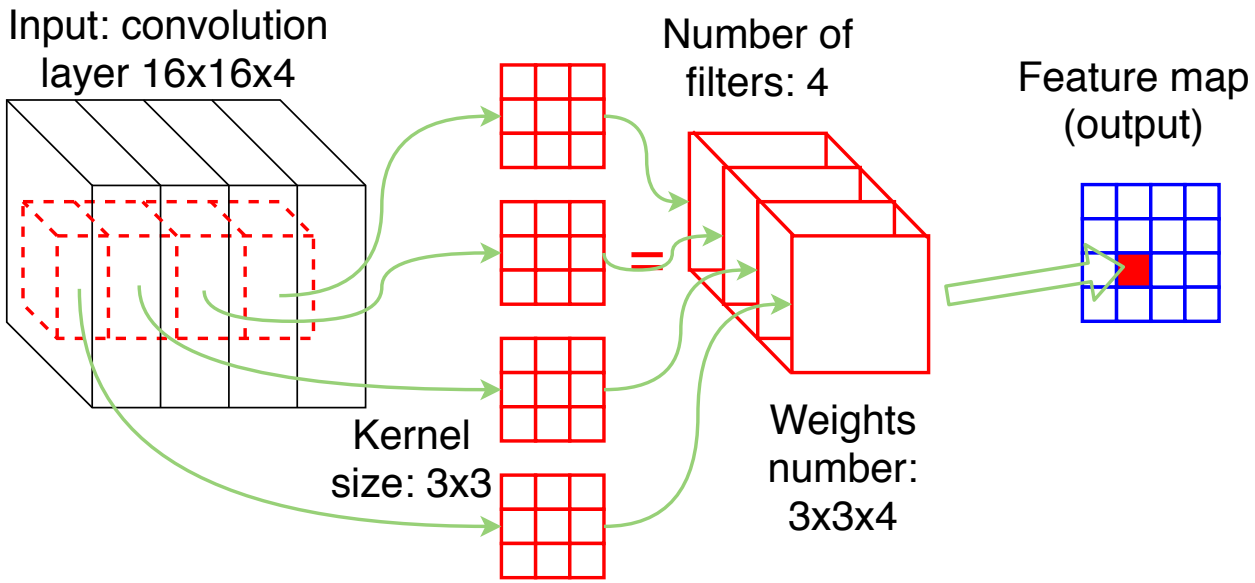
CNN Architecture



- Each neuron in the convolutional layer is connected only to a local region in the input volume spatially through the full depth of the input layer. There are multiple neurons along the depth, all using the same region in the input.
- The structure of neurons remain the same as in MLP. They still compute a dot product of their weights with the input followed by a non-linear function, but their connectivity is now restricted to be local spatially.

CNN Architecture

The neurons in a CONV layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully-connected manner.

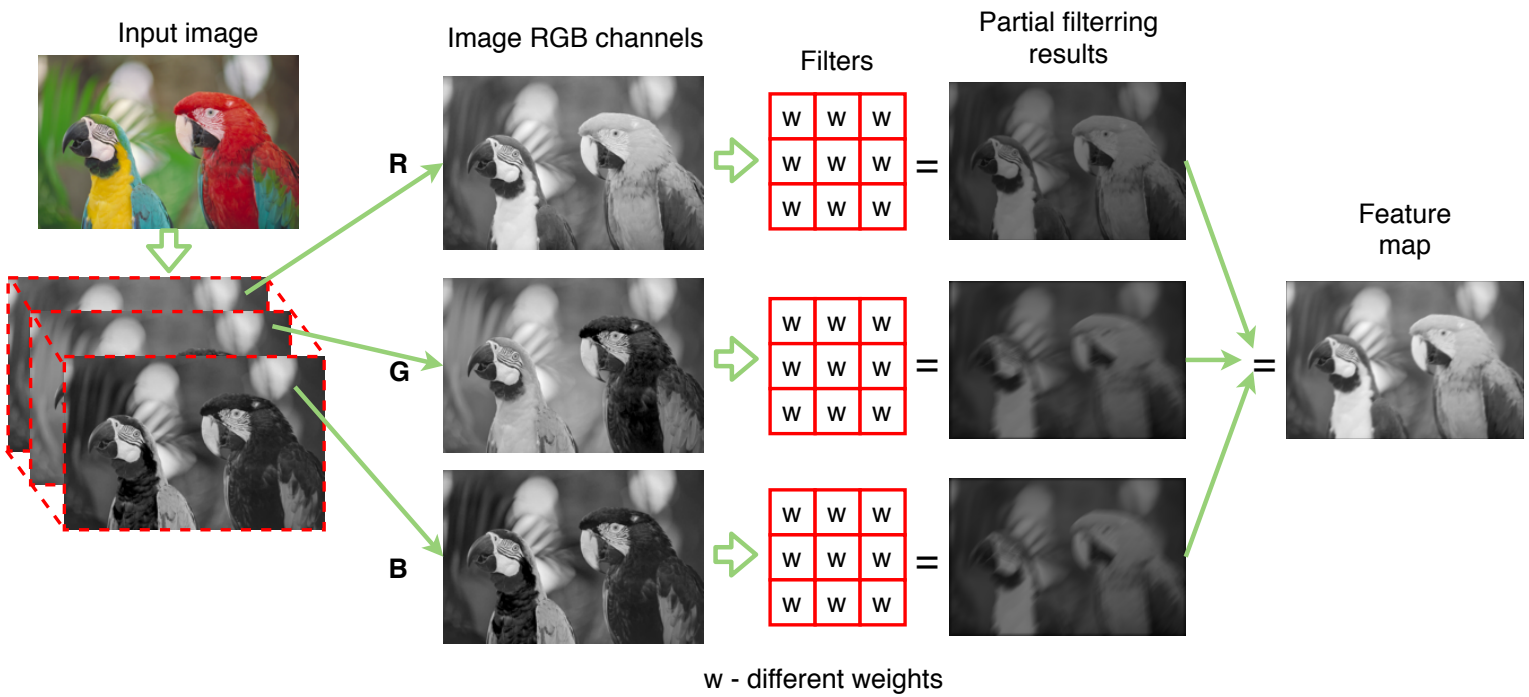


For each layer we need separate set of filters. Example: If present layer have 8 filters and previous layer have 16 than we need to find parameters of $8 * 16 = 128$ of 3×3 filters.

CNN Architecture

Features of the network:

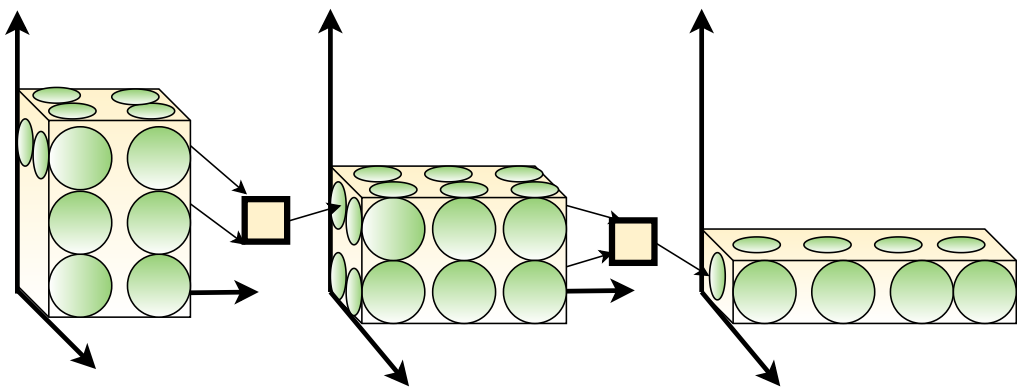
- Feature mapping. Each computational layer of the network is composed of multiple feature maps, with each feature map being in the form of a plane within which the neurons are constrained to share the same set of weights.



CNN Architecture

Features of the network:

- Subsampling. Each convolutional layer is followed by a computational layer that performs local averaging and subsampling, whereby the resolution of the feature map is reduced. This operation has the effect of reducing the sensitivity of the output to shifts, rotations and other distortions.



Convolutional Layer

Convolutional Layer will compute the output of neurons that are connected to local regions in the input:

- During the forward pass, we slide each filter across the width and height of the input volume and compute dot products between the weights of the filter and the input at any position.
- The network can learn filters that activate when they see some type of visual feature, e.g.: an edge detection in some orientation or a blobs of color.

Convolutional Layer

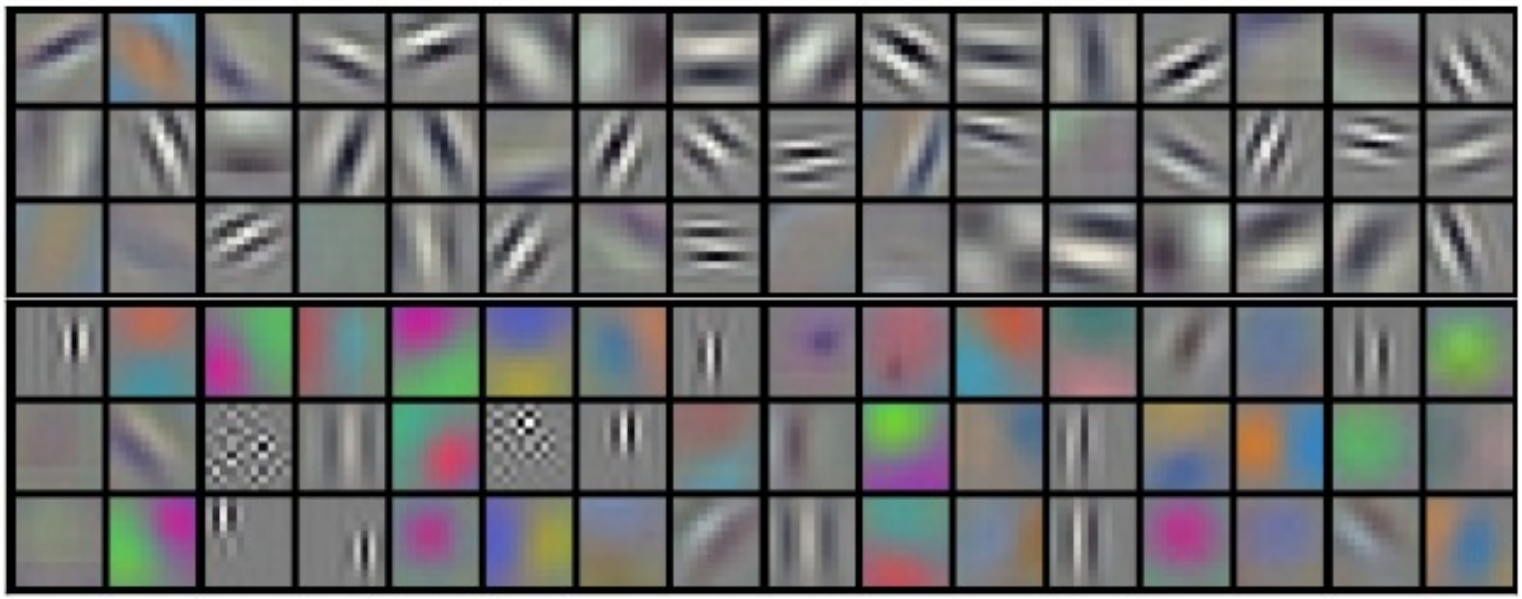
The depth of the output volume is a hyperparameter:

- It corresponds to the number of filters we would like to use. Each filter learns to look for something different feature in the input volume,
- The first layer takes as input the raw image, the size of kernel in regulate the local the image search area.
- polling, stride and zero-padding regulate the size of CONV layer.

The depth of convolutional Layer increase with each added layers.

Convolutional Layer

Parameter sharing: If one feature is useful to compute at some spatial position, then it should also be useful to compute at a different position.



CNN Layers

We can use additional types of layers to build CNN:

- Pooling Layer (POOL),
- Rectified Linear Units layer (RELU),
- Fully-Connected Layer (FC).
- Generalization layers like: Droopout layer or Batch normalization layer,
- Others, among important: transpose convolution (deconvolution) layer.

The CNN also have input layer, which hold the raw pixel values of the image.

CNN Layers

Note that some layers contain parameters and other don't:

- the CONV/FC layers perform transformations which depend on activations functions and parameters (weights).
- the RELU/POOL layers will implement a fixed function.

Training:

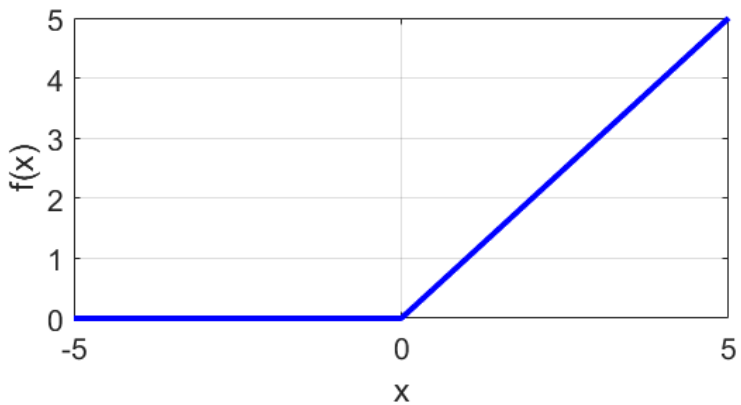
- The parameters in the CONV/FC layers will be trained with gradient methods,
- Other layers have implementation according to Authors instructions.

All weights in all layers of a CNN are training to extract features of input data.

ReLU Layer

Rectified Linear Units (ReLU) layer will apply an element wise activation function:

$$f(x) = max(0, x)$$



It usually applied after each CONV layer:

- The purpose of this layer is to introduce nonlinearity.
- The use of ReLU layers allows to train a lot faster without making a significant difference to the accuracy.

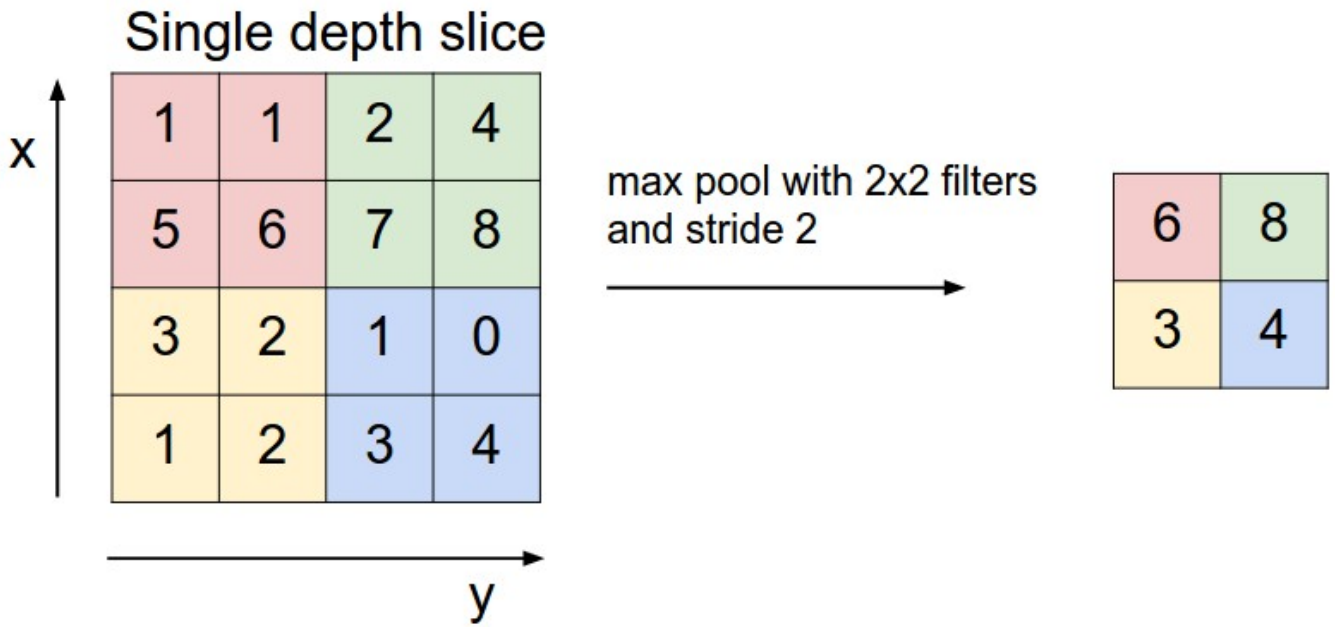
Pooling Layer

Pooling layer (downsampling layer)

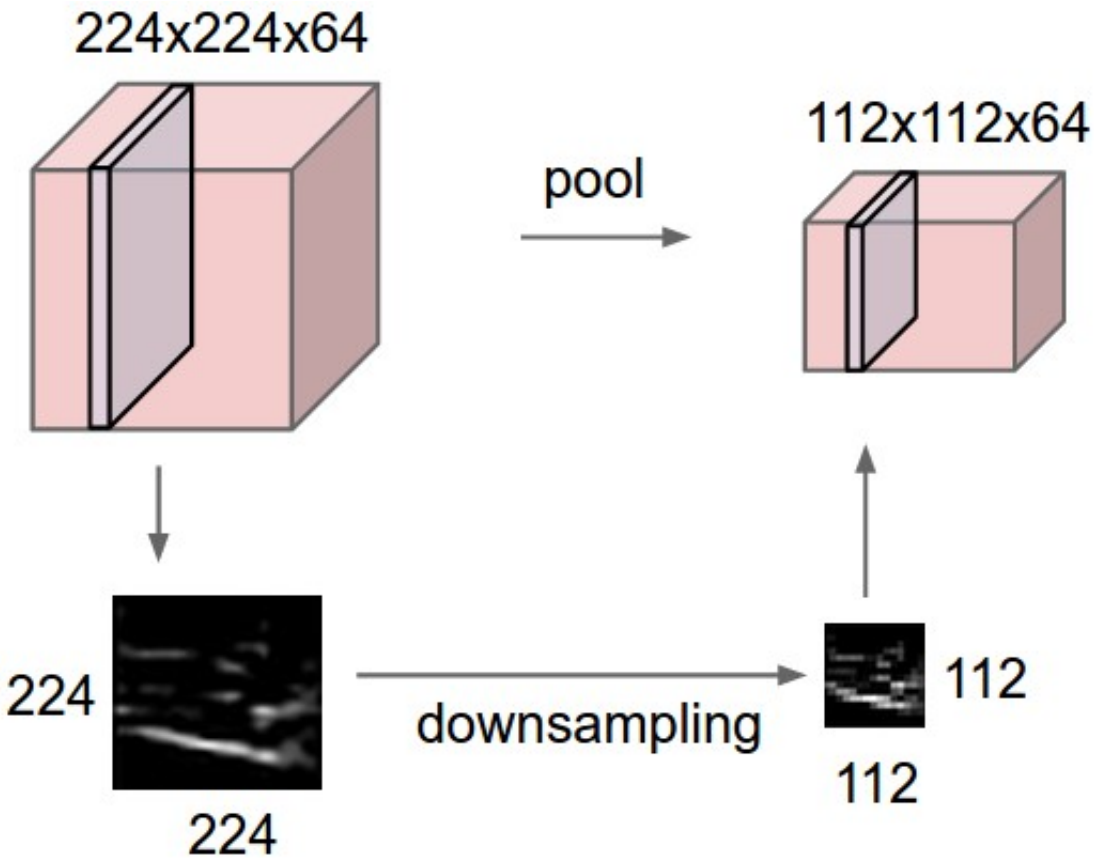
- Progressively reduce the spatial size of the network layer to reduce the amount of parameters and computation in the network, and hence to also control overfitting.
- Operates independently on every depth slice of the input:
  - max polling - use max operation (work better in practice),
  - average polling - use average operation.

Pooling Layer

The most common form is a pooling layer with filters of size 2×2 applied with a stride of 2 downsamples every depth slice in the input by 2 along both width and height, discarding 75% of the activations. Example of max polling:

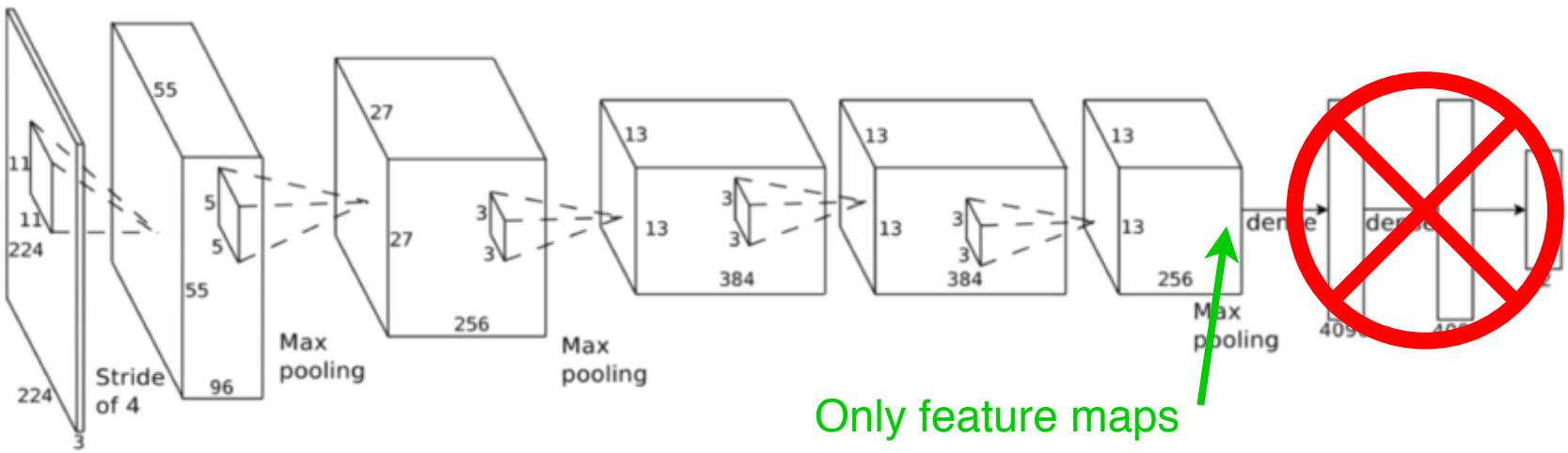


Pooling Layer



Feature maps

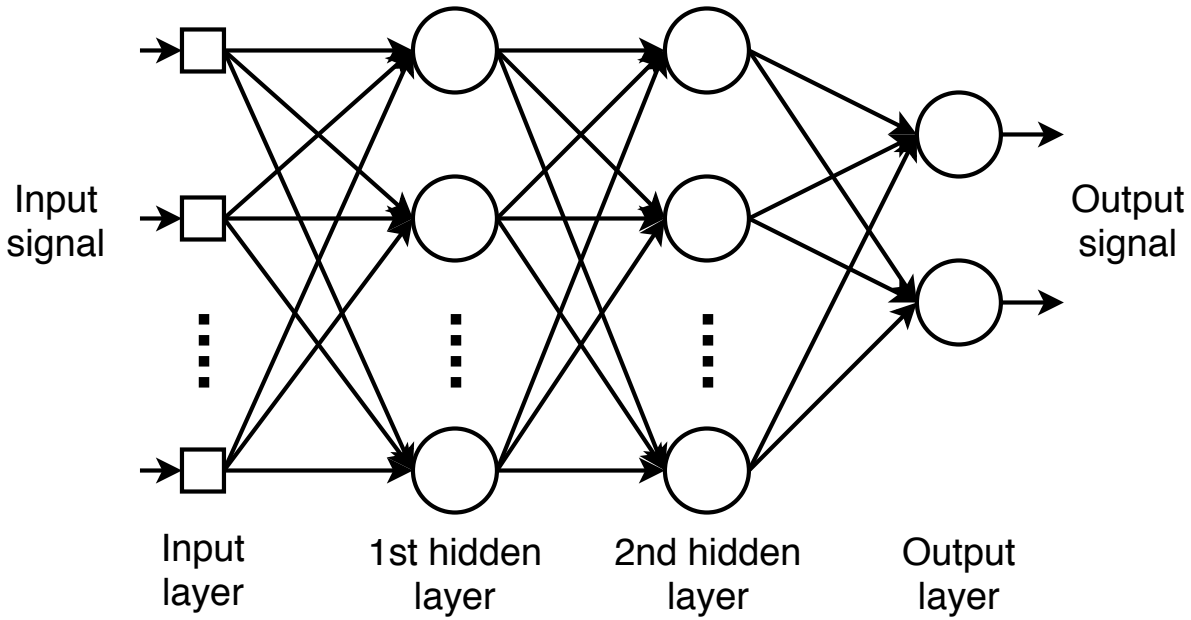
The base part of the CNN task is to extract feature maps:



The second part of the CNN task is classification.

Fully-connected Layer

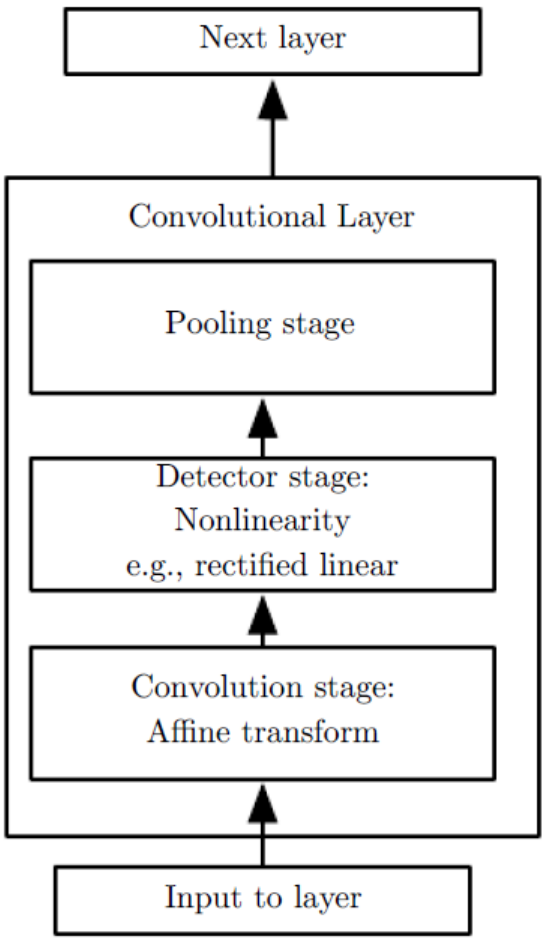
Neurons in a fully connected layer have full connections to all activations in the previous layer.



CNN Layers

A common CNN block of layers performs 3 tasks:

- In the first stage, the layer performs several convolutions in parallel to produce a set of linear activations.
- In the second stage, each linear activation is run through a nonlinear activation function, such as the ReLu function.
- In the third stage, we use a pooling function to modify the output of the layer further.



CNN training

The overall training process of CNN:

1. We initialize all parameters (weights) with random values,
2. Take a training image as input from image dataset.
3. Forward propagation step finds the output probabilities for each class (going through Conv, ReLU, pooling and the FC layer).
4. Calculate the total error at the output layer according to loss function,
5. Backpropagation step calculate the gradients of the error with respect to all weights in the network to update all weights and parameter values.
6. Repeat steps 2-5 with new images until stop criterion conditions are satisfied.

Transfer learning

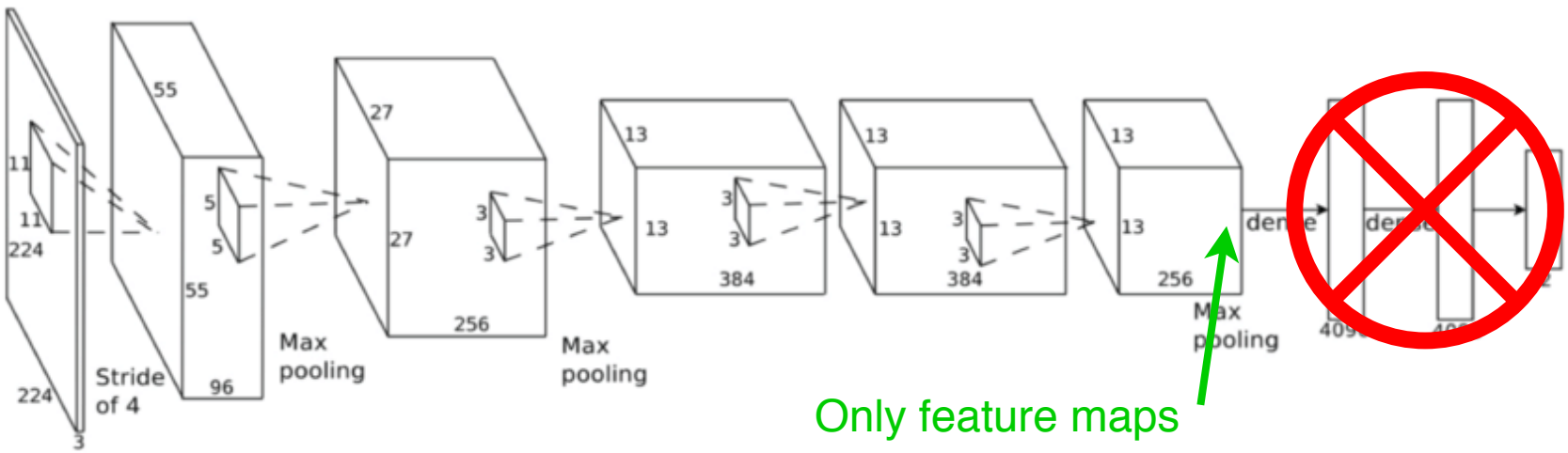
The data is critical part of creating the network, we can use a transfer learning to lessen the data demands:

- Transfer learning is the process of taking a pre-trained model, where the weights and parameters of a network that has been trained on a large dataset by somebody else and "fine-tuning" the model with your own dataset.
- The idea is that this pre-trained model will act as a feature extractor. In the simplest use of the network, wean remove the last layer of the network and replace it with your own classifier (training only classifier layers).

Example: the pre-trained model of ImageNet, which was trained on dataset that contains 14 million images with over 1,000 classes.

Encoder

The base part of the CNN task is to extract feature maps and classification part is replaced.



Transfer learning

Transfer learning of classification CNN in practices:

- new different classes,
- new data set labeled for new classification.

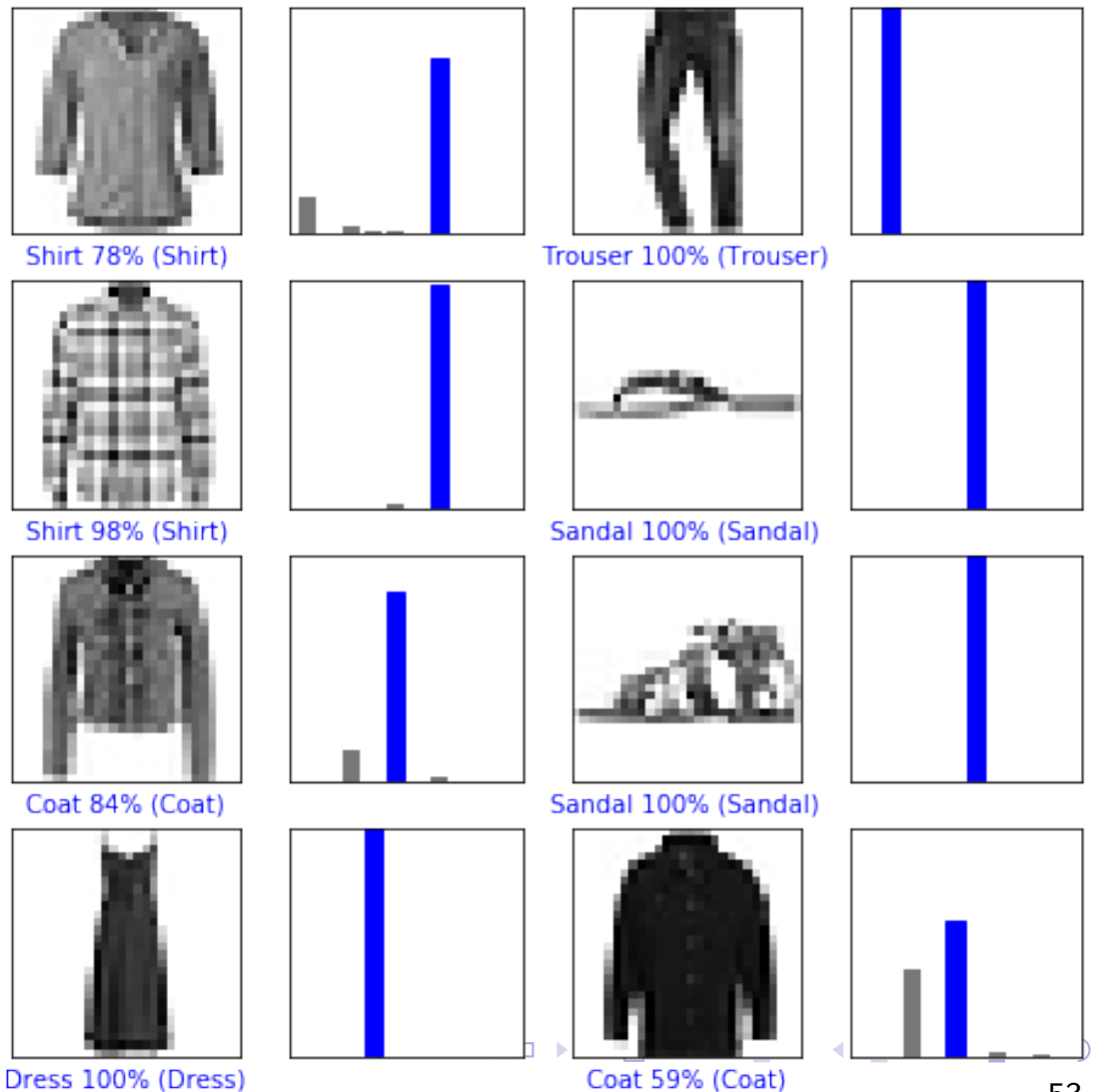
Fine-tune of CNN on a new set of classes:

1. Create chosen architecture of CNN and download pre-trained weights (like Inception-V3),
2. Remove output layer (or layers) responsibly for classification ,
3. Add new output layer for desired classes (with any needed additional layers like maxpooling, droopout with additional fully-connected layer),
4. Train only weights of output layer with new data,
5. Train whole network with new data.

Popular CNN architectures

Object classification:

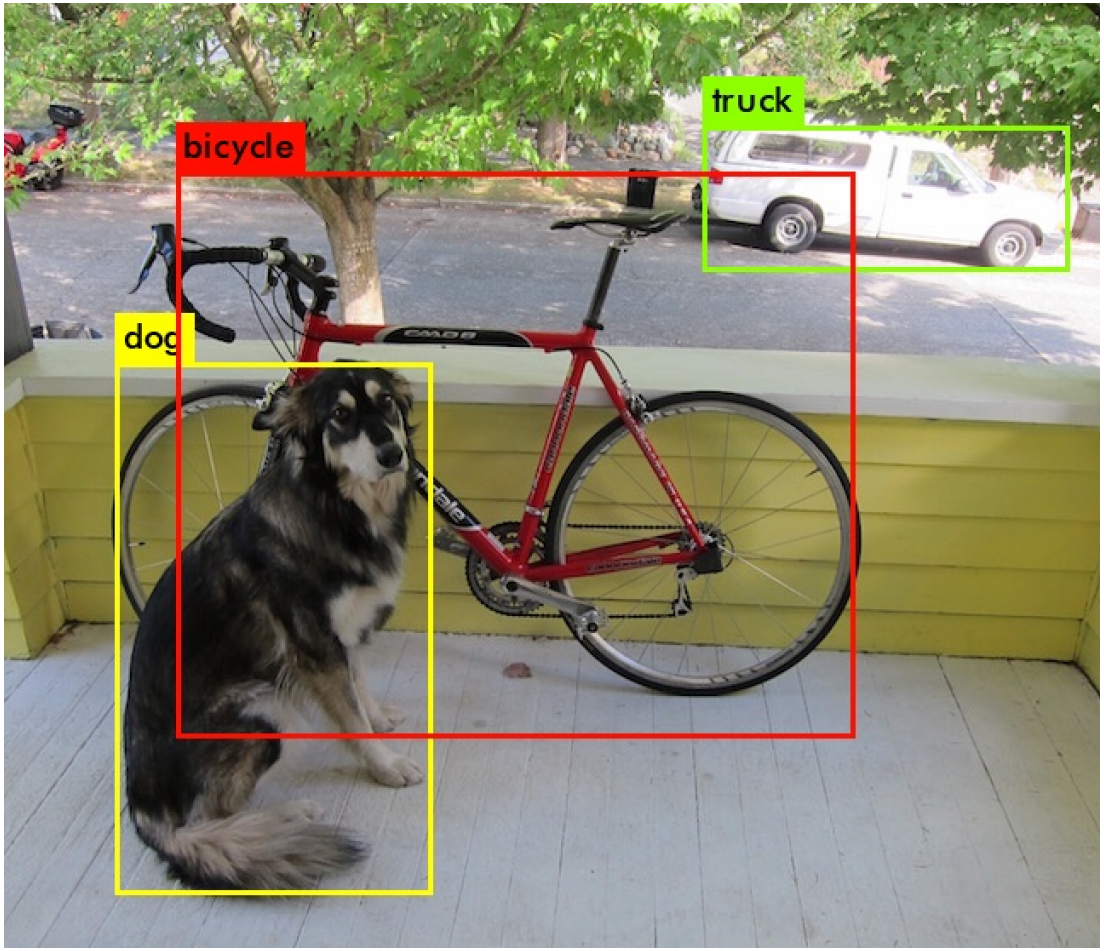
- AlexNet ,
- VGG16,
- ResNet50,
- Inception-v3,
- SqueezeNet,
- MobileNet-v2.



Popular CNN architectures

Object localization:

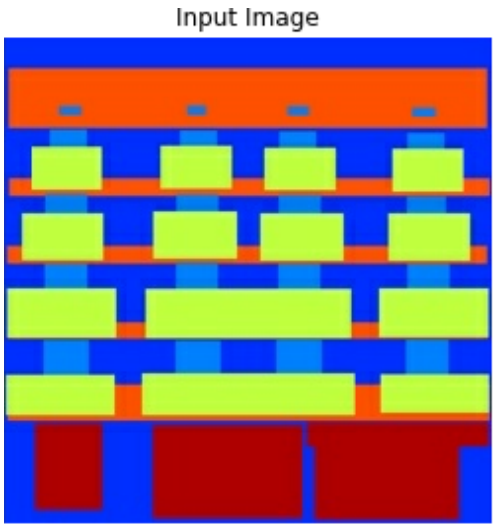
- Faster R-CNN,
- YOLO-V4,
- SSD,
- RetinaNet.



Popular CNN architectures

Pixel-to-pixel operation:

- U-NET,
- SegNet,
- GAN.



Questions