Introduction to Deep Learning

LECTURE 3

Elad Hoffer
Nir Ailon
Ran El-Yaniv

Technion Israel Institute Of Technology,
Computer-Science department

Overview

1. Background
2. Convolution layer
3. Pooling layers

3.1 Pooling

4. Convolutional networks
5. Case studies

Background

Background

- Convolutional networks are rooted in ideas from "NeoCognitron" (Fukoshima 80') and inspired from Hubel's and Wiesel's work on visual primary cortex (59')
- Their first modern form appeared in Lecun's 98' work to recognize handwritten digits (LeNet).
- They reached their current status as visual recognition de-facto standard, after beating traditional computer-vision techniques in 2012 ImageNet competition by huge margin (AlexNet).

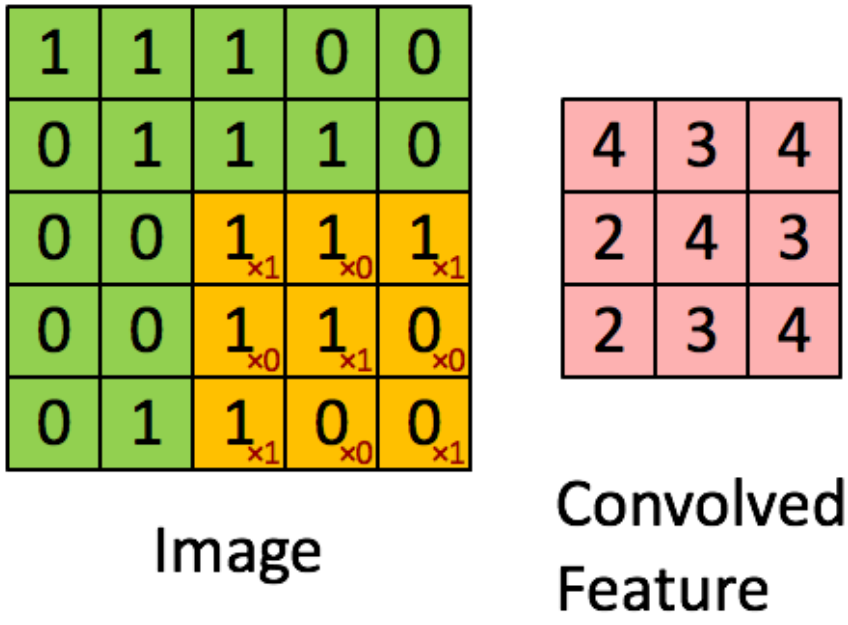
Convolution layer

Spatial Convolution

- Natural images have the property that many patches share statistical properties, and can often be described with a fairly small set of the same features
- This suggests that the features that we learn at one part of the image can also be applied to other parts of the image, and we can use the same features at all locations.
- This is a widely used property in computer-vision and image-processing related tasks.

Spatial Convolution

For example, having learned features over small (say 5×5) patches sampled from the image, we can then convolve them with the larger image, thus obtaining a different feature activation value at each location in the image.



Convention and terminology

- $2d$ grids that represent image space are called *feature maps*. Each value in a feature map represent information about a specific location in the input image.
- The parameters learned for each convolution layer are also ordered as a $2d$ map - usually called “kernels” or “filters”
- Convolutions can have *strides* - moving the filters with a fixed step and *padded* - adding zero values to avoid spatial size decrease
- Bias values are added per-map (number of bias elements is the number of output feature maps)

Forward function

It is most natural to think of this as correlation between two cubes:

- Our input is $X \in \mathbb{R}^{N \times W \times H}$ where N is the number of feature maps, each of spatial size $W \times H$.
- We wish to correlate them with $W \in \mathbb{R}^{M \times N \times K \times K}$ - M different filters of size $N \times K \times K$
- Output is $Y \in \mathbb{R}^{M \times (W-K+1) \times (H-K+1)}$ (no stride, no padding)

$$y_{m,i,j} = \sum_{n=0}^{N-1} \sum_{k=0}^{K-1} \sum_{l=0}^{K-1} w_{m,k,l} \cdot x_{n,(i+k),(j+l)} + b_m$$

It can also be expressed as sum of M 2d-convolutions

$$Y_m = \sum_{n=0}^{N-1} W_m * X_n + b_m$$

Properties

Convolutional layers can be seen as a specific configuration of a fully-connected layer. The assumptions we made about images are embedded as stationary entities allow us to use:

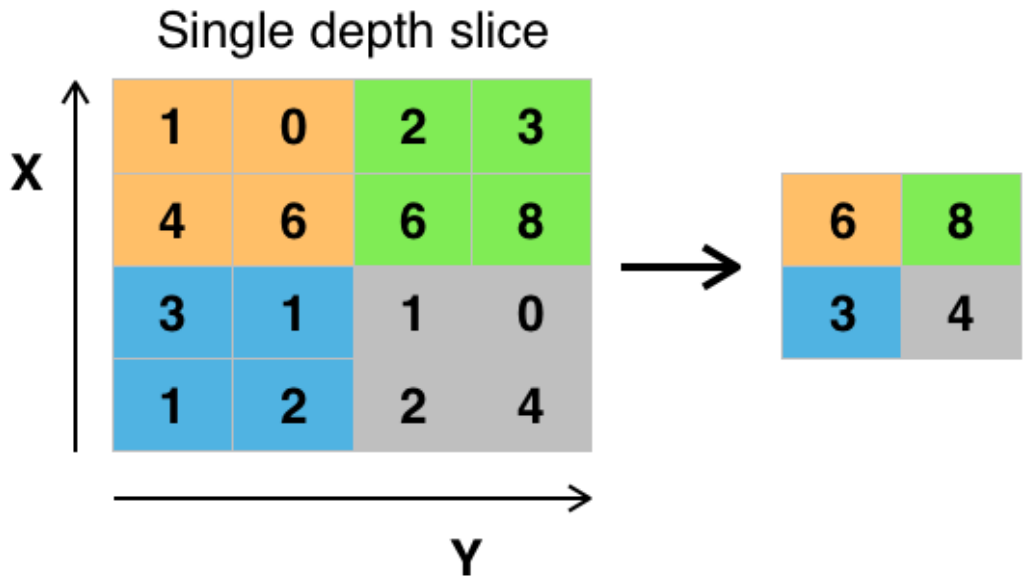
- Local connectivity - units in subsequent layer are connected only to a small subset in previous layer.
- Shared weights - weights used to extract information are shared across all spatial locations (this is not always the case - e.g faces).

Both these changes allow a dramatic reduction in the number of trainable parameters.

Pooling layers

Pooling

To describe a large image, one natural approach is to aggregate statistics of features at various locations. For example, one could compute the mean (or max) value of a particular feature over a region of the image.



Pooling

Pooling operations serve two main purposes:

- Dimensionality reduction - pooling is usually done with stride bigger than 1, effectively reducing the spatial size of the maps and allowing easier processing.
- Local invariance to translation/transformation - by pooling a spatial area to single point, we allow small invariance to location and deformations. This also reduces the ability to overfit on specific images (generalizes from patches).

Pooling functions

Most effective pooling methods are

- Max Pooling - taking the maximum element from the pooled area.
- Average Pooling - taking the average of the pooled area.

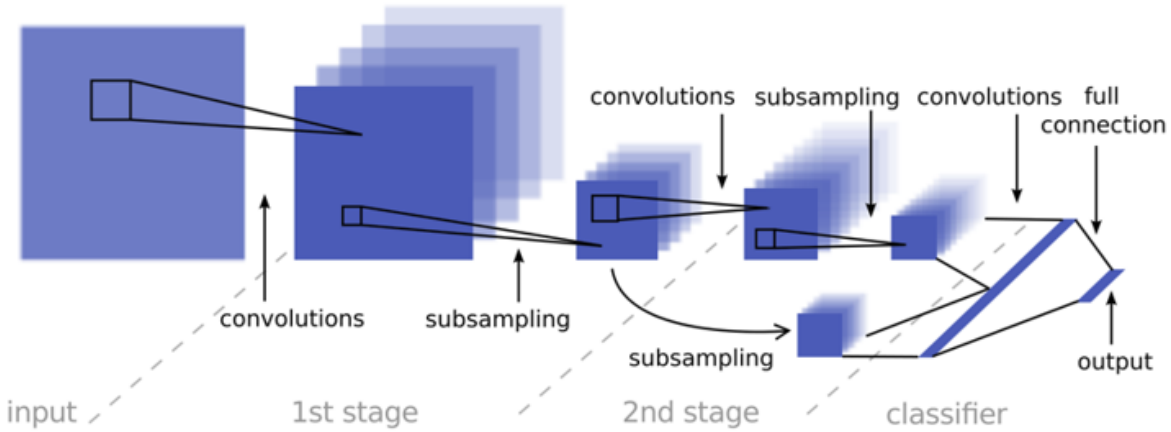
Less popular, bet useful in some cases:

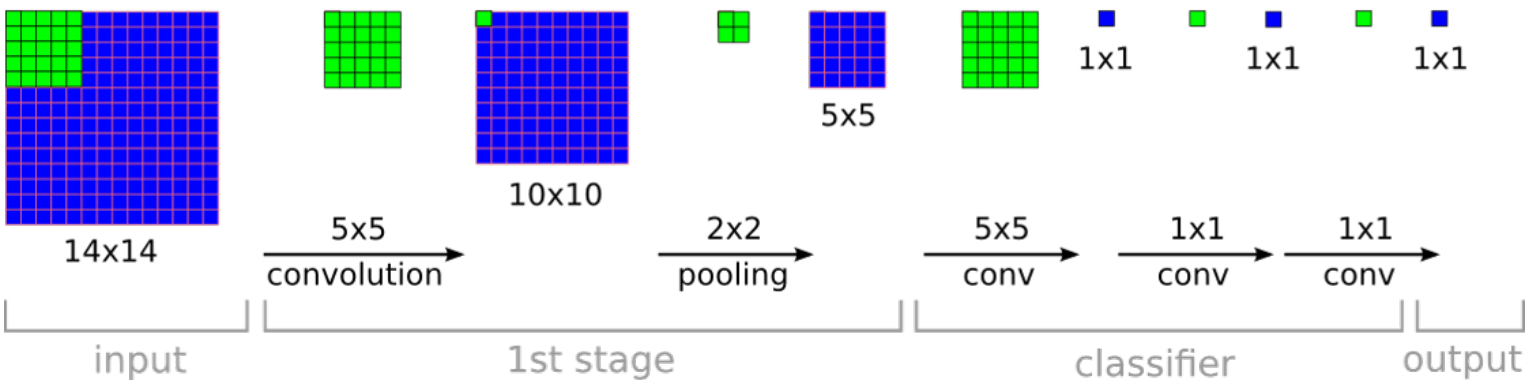
- Stochastic pooling - taking a random value.
- Fractional pooling - using a random pool size (2-3).
- L_p Pooling - taking the L_p norm of the pooled area.

Convolutional networks

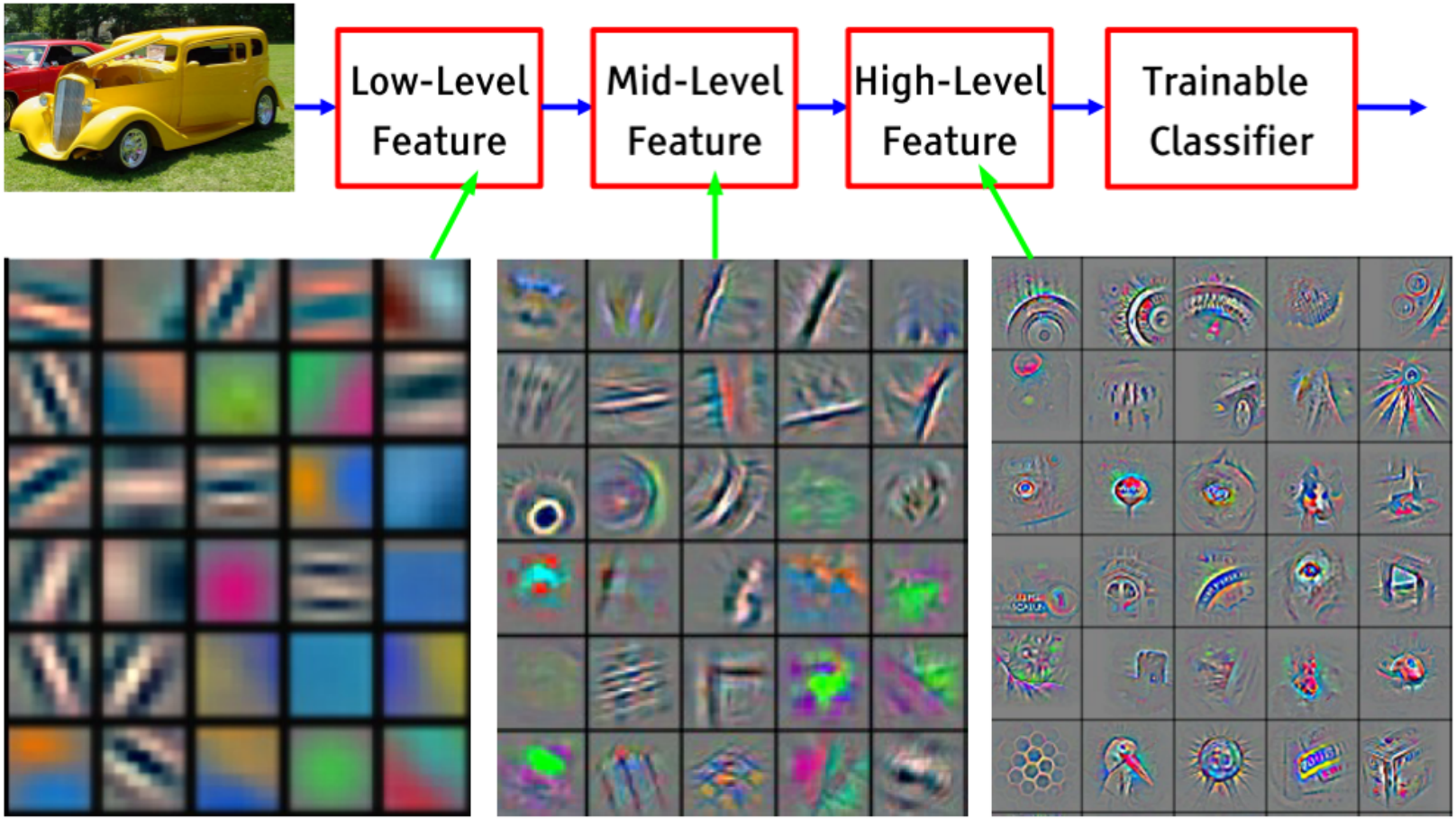
ConvNets

A convolutional network (ConvNet/CNN), is composed of multiple layers of convolutions, pooling and non-linearities.





What do convolutional networks learn?



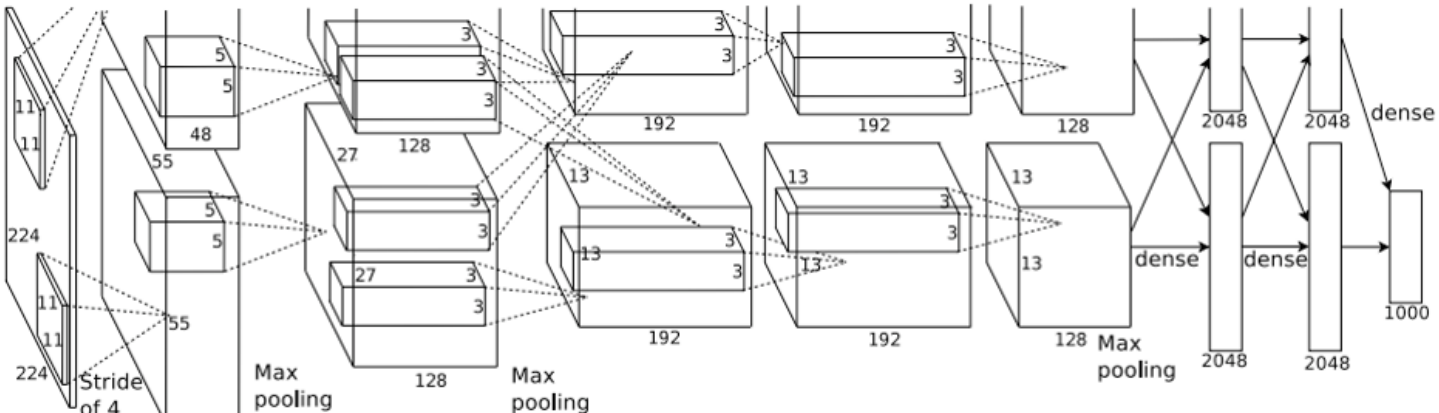
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Case studies

AlexNet

The first work that popularized Convolutional Networks in Computer Vision was the "AlexNet", developed by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton.

- AlexNet was submitted to the ImageNet ILSVRC challenge in 2012 and significantly outperformed the second runner-up (top 5 error of 16% vs 26% error).
- The Network was deeper, bigger, than previous networks - $60M$ parameters, 8 trainable layers.

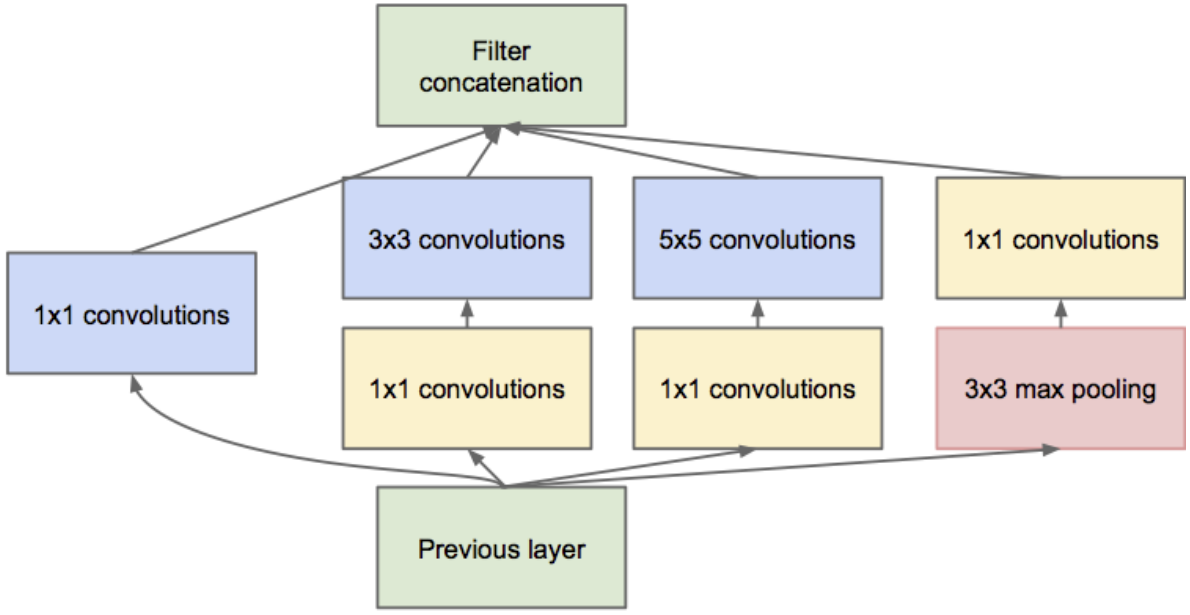


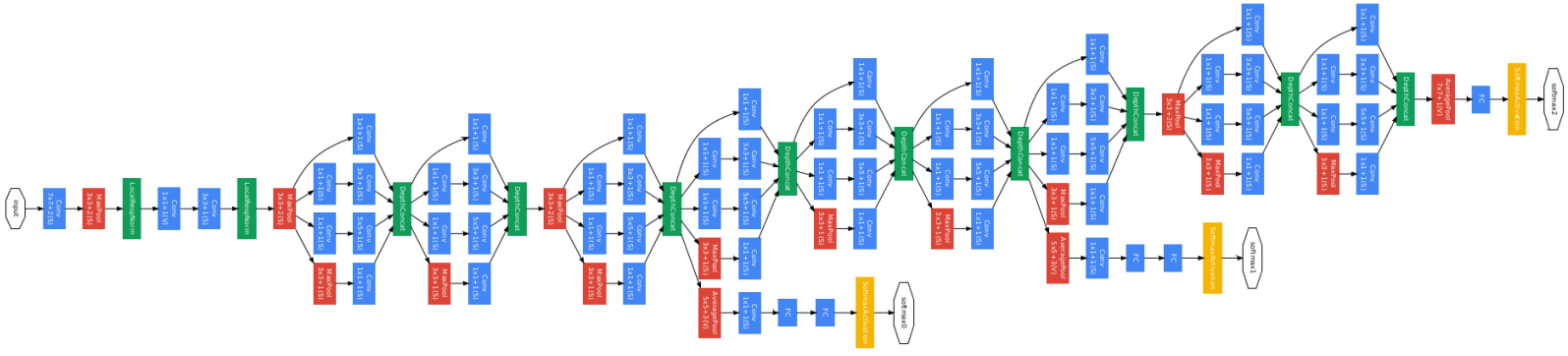
GoogLeNet

The ILSVRC 2014 winner was a Convolutional Network from Szegedy et al. from Google called *GoogLeNet*. It featured some new ideas

- Inception Module - concatenation of several convolution sizes
- 1×1 convolutions + Average Pooling instead of Fully Connected layers (both introduced by Lin in "Network-in-network" paper from 2013).
- This dramatically reduced the number of parameters in the network (5M, compared to AlexNet with 60M).
- Google later improved this network by introducing batch-normalization + different inception configuration (Inception v2-v4)

GoogLeNet


Filter concatenation
3x3 convolutions
5x5 convolutions
1x1 convolutions
1x1 convolutions
1x1 convolutions
1x1 convolutions
3x3 max pooling
Previous layer



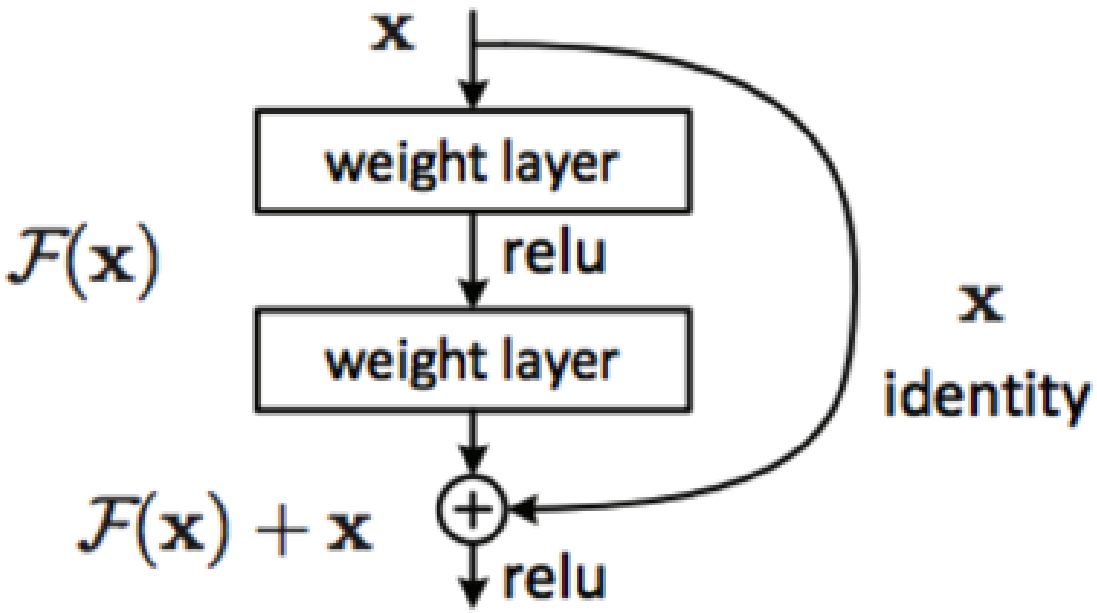
VGG Network

The runner-up in ILSVRC 2014 was the network from Karen Simonyan and Andrew Zisserman that became known as the VGGNet. Its main contribution was in showing that the depth of the network is a critical component for good performance.

- Their best network contains 16 trainable layers
- Features an extremely homogeneous architecture that only performs 3x3 convolutions and 2x2 pooling from the beginning to the end.
- Despite its slightly weaker classification performance, the VGG ConvNet features outperform those of GoogLeNet in multiple transfer learning tasks.
- Expensive to evaluate and uses a lot more memory and parameters (140M).

Residual networks

- Residual Network developed by Kaiming He et al. was the winner of ILSVRC 2015. It features a very deep architecture (152 layers) with special skip connections and heavy use of batch normalization. The architecture is also missing fully connected layers at the end of the network.



Trends in designing CNNs

Some noticeable trends in recent CNN architectures:

- Smaller convolution kernels - 3×3 is very dominant.
- Deeper - using smaller kernels is rectified by stacking more layers - effectively increasing the receptive field of the network.
- Less pooling - going deeper means we do not want to reduce spatial size too quickly. Modern network have small number (if any) of pooling layers.