Macro NN architecture

Machine Learning and Data Mining

Maxim Borisyak
National Research University Higher School of Economics (HSE)

November 7, 2017

Outline

Super inspirational quotes

Network architecture is more like an art.

Behind every is a poorly formulated science.

Network architecture

> *Neural Network Architecture plays crucial role in Deep Learning.*

Most of the non-trivial architectures:

- derived from common sense;
- explained by math;
- demonstrated on some real problems.

Usual disclaimer

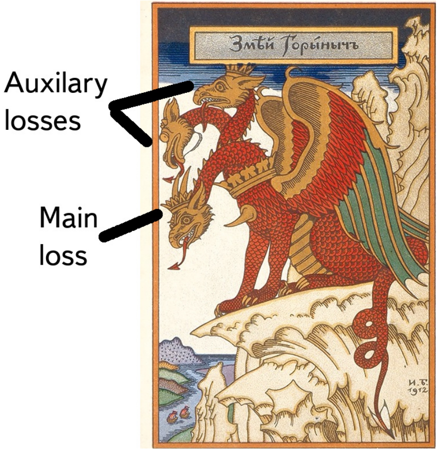
The following examples are not aimed to be cover major architecture tricks. Just some examples happened to be known by the author.

Auxilary losses

Auxilary problems

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \dots$$

- solving several objectives with one network:
  - brining more information about the solution;
- auxilary losses should share the same solution;

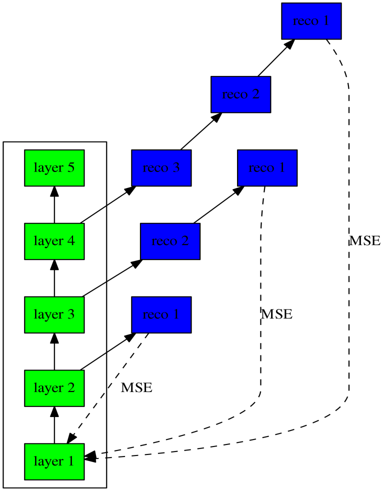


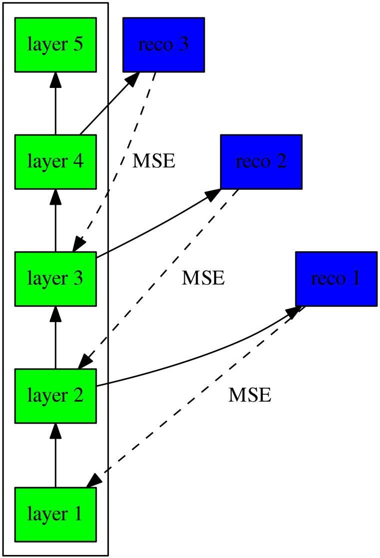
Auxilary problems: examples

Are the following auxilary problems reasonable:

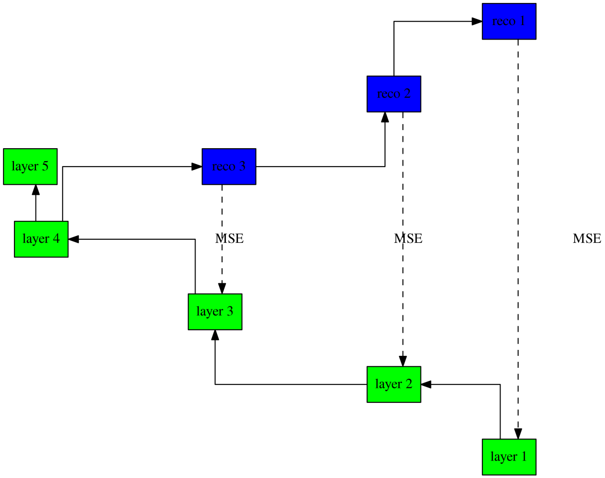
- even vs. odd digit for MNIST;
- reconstructing initial image for MNIST;
- producing countour of target objects for detection problems;
- predicting type of a street-sign for detection problem;
- predicting super-class for CIFAR-100;
- predicting faces properies (e.g. smile/anger/neutral, female/male etc) for dimensionality reduction?

Reconstruction regularisation





Reconstruction regularization

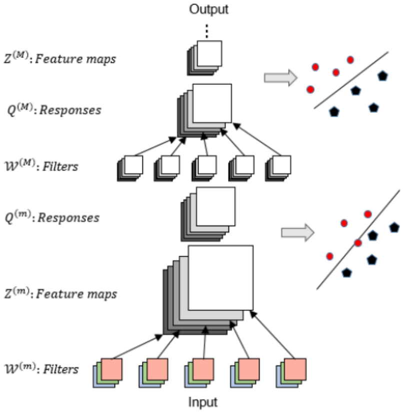


Reconstruction regularization

- unsupervised loss may be in conflict with classification loss;
    - reconstruction generally require higher network capacities;
    - discriminative features might be lost as unimportant for reconstruction;
- rarely used in practice.

Deeply supervised networks

- try to solve original problem early;
- improved gradient flow (almost impossible to make it vanish);
- quite strong regularization effect;
- no unsupervised vs. supervised conflict.

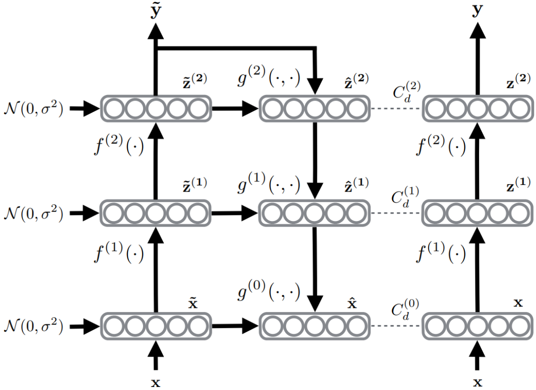


(a) DSN illustration

Ladder Networks

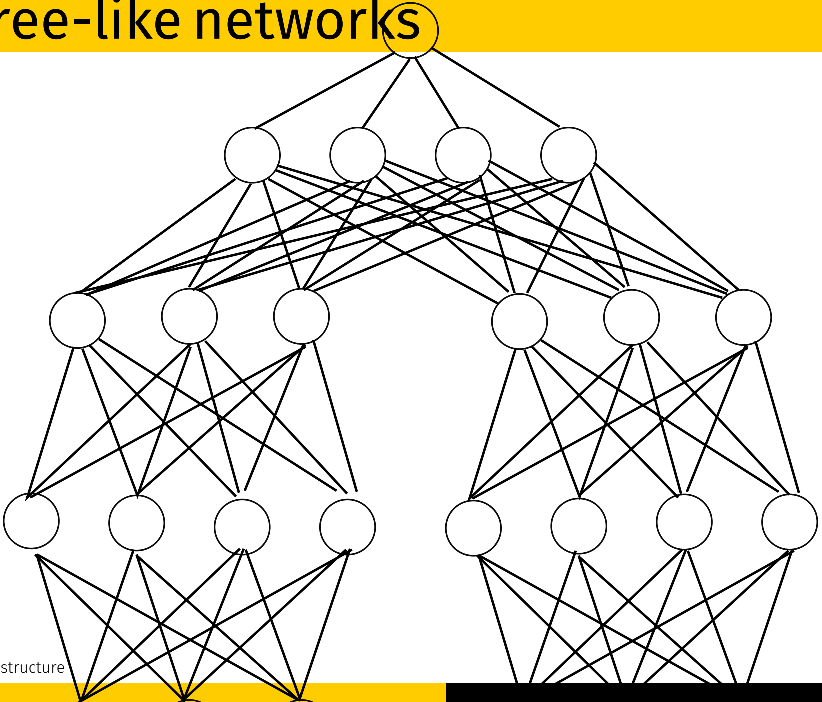
- replaces reconstruction with denoising:

$$\mathcal{L} = \|f(x+\varepsilon) - x\|^2 \to \min,\ \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

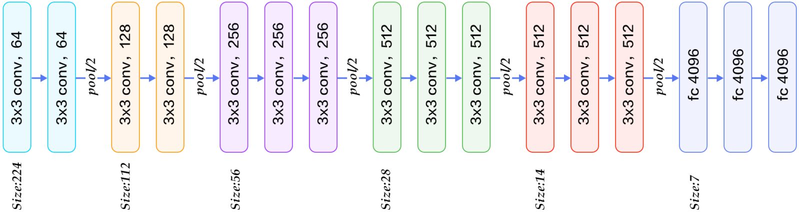


Network structure

Tree-like networks



VGG

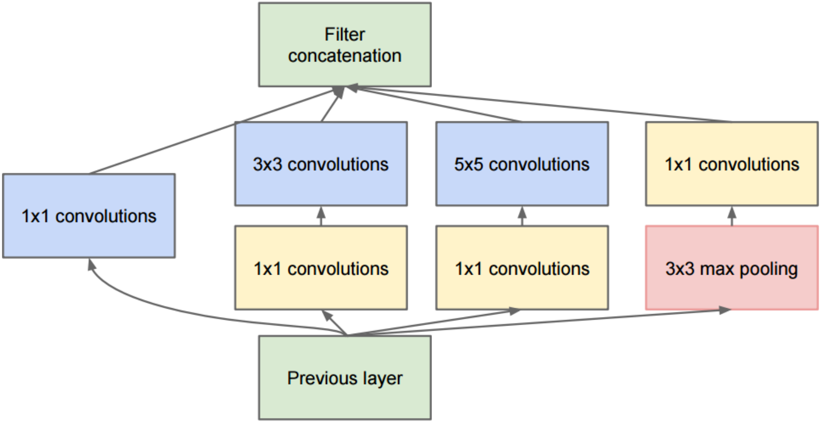


3x3 conv，64
3x3 conv，64
pool/2
3x3 conv，128
3x3 conv，128
pool/2
3x3 conv，256
3x3 conv，256
3x3 conv，256
pool/2
3x3 conv，512
3x3 conv，512
3x3 conv，512
pool/2
3x3 conv，512
3x3 conv，512
3x3 conv，512
pool/2
fc 4096
fc 4096
fc 4096
Size:224
Size:112
Size:56
Size:28
Size:14
Size:7

Inception

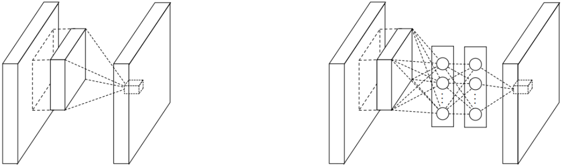
- blue blocks: conv;
- red blocks: pool;
- green blocks: concat;
- yellow blocks: softmax.

Inception block



(b) Inception module with dimension reductions

NIN: conv on steriods



(a) Linear convolution layer (b) Mlpconv layer

Figure 1: Comparison of linear convolution layer and mlpconv layer. The linear convolution layer includes a linear filter while the mlpconv layer includes a micro network (we choose the multilayer perceptron in this paper). Both layers map the local receptive field to a confidence value of the latent concept.

ResNet

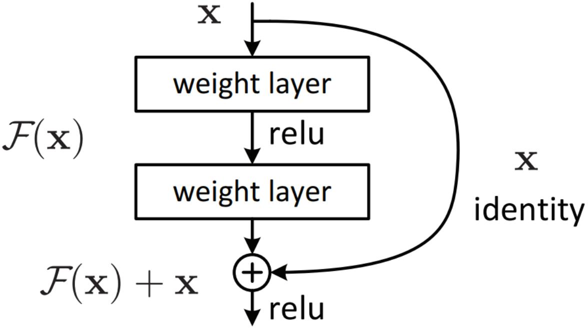
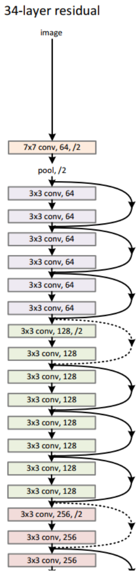


Figure 2. Residual learning: a building block.

ResNet



34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256

ResNet

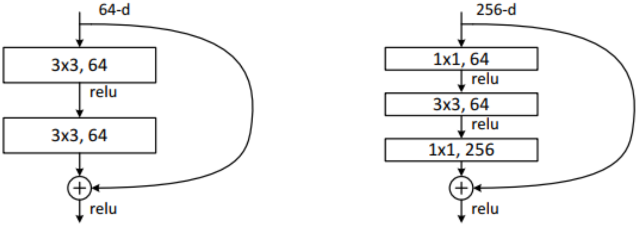


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

Highway networks

Feed-forward networks:

$$y = H(x, W_H)$$

Residual connection:

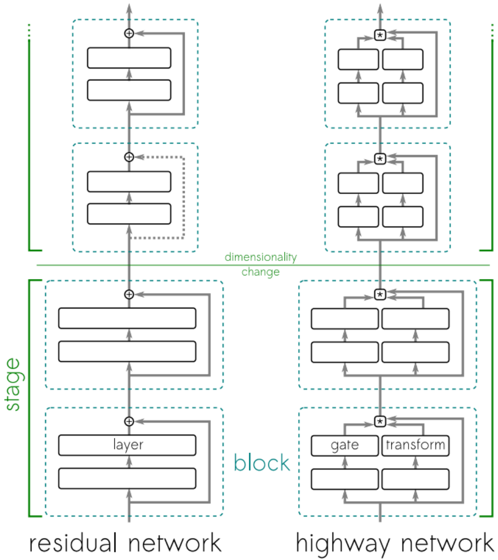
$$y = H(x, W_H) + x$$

Highway connection:

$$y = T(x, W_T)H(x, W_H) + C(x, W_C)x;$$

- x, y - input, output;
- $H(x, W_H)$ - some transformation, e.g. convolution;
- $T(x, W_T), C(x, W_C) \in [0, 1]$ - gates (*transform* and *carry*).

Highway networks



Squeeze net

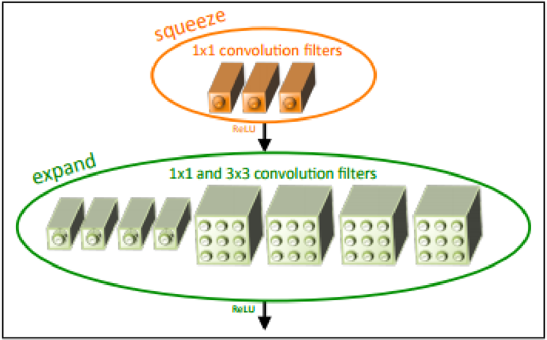
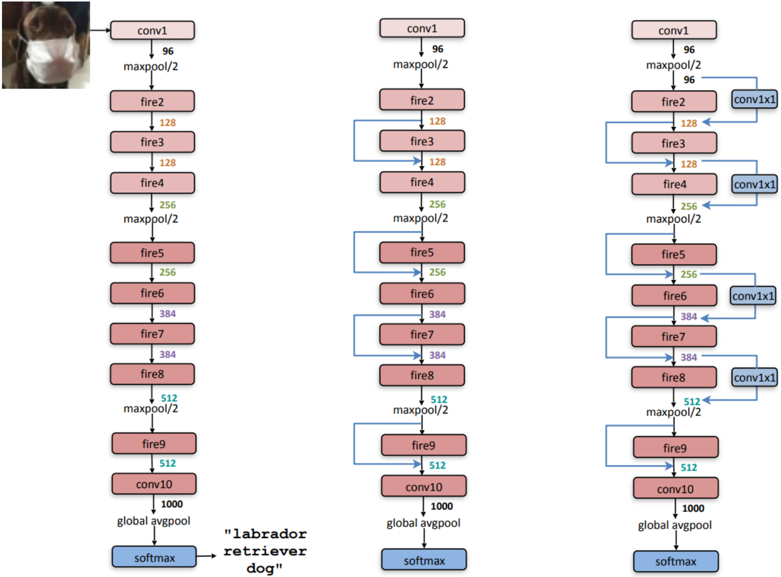
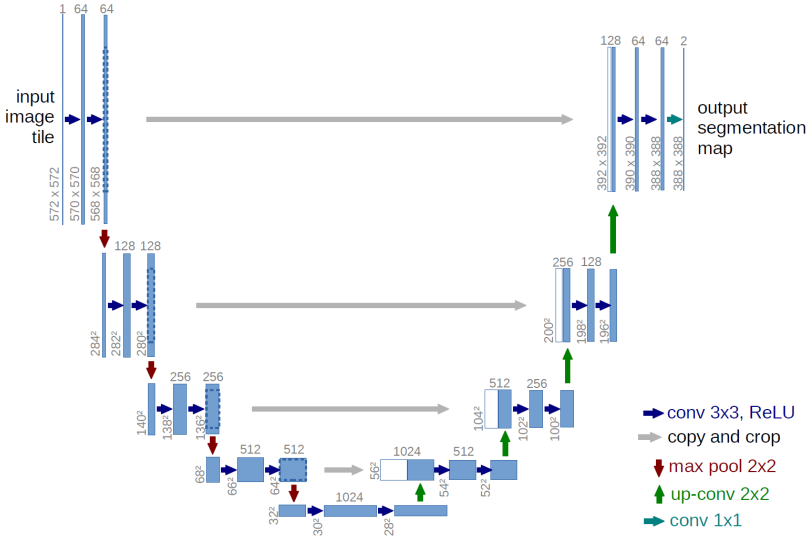


Figure 1: Microarchitectural view: Organization of convolution filters in the **Fire module**. In this example, $s_{1x1} = 3$, $e_{1x1} = 4$, and $e_{3x3} = 4$. We illustrate the convolution filters but not the activations.

Squeeze net


conv1
96
maxpool/2
fire2
128
fire3
128
fire4
256
maxpool/2
fire5
256
fire6
384
fire7
384
fire8
512
maxpool/2
fire9
512
conv10
1000
global avgpool
softmax
"labrador retriever dog"
conv1x1

U-net


1
64
64
input
image
tile
572 x 572
570 x 570
568 x 568
128
128
284²
282²
280²
256
256
140²
138²
136²
512
512
68²
66²
64²
1024
32²
30²
28²
1024
512
56²
54²
52²
512
256
104²
102²
100²
256
128
200²
198²
196²
128
64
64
2
392 x 392
390 x 390
388 x 388
388 x 388
output
segmentation
map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

Exercise

Suggest an architecture for a face recognition security system:

- system should be able to grant access to any person with sufficient rights.

Describe:

- data required;
- function of the neural network (classification, regression, clusterisation);
- architecture of the network;
- training procedure.