Convolutional Neural Networks

- Image classification at the example of VGG net -

Lecturer: Michael Möller – michael.moeller@uni-siegen.de
Exercises: Hartmut Bauermeister – hartmut.bauermeister@uni-siegen.de

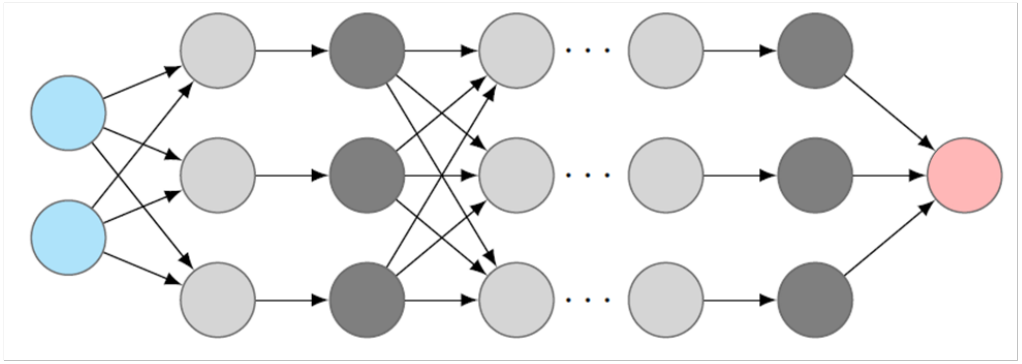
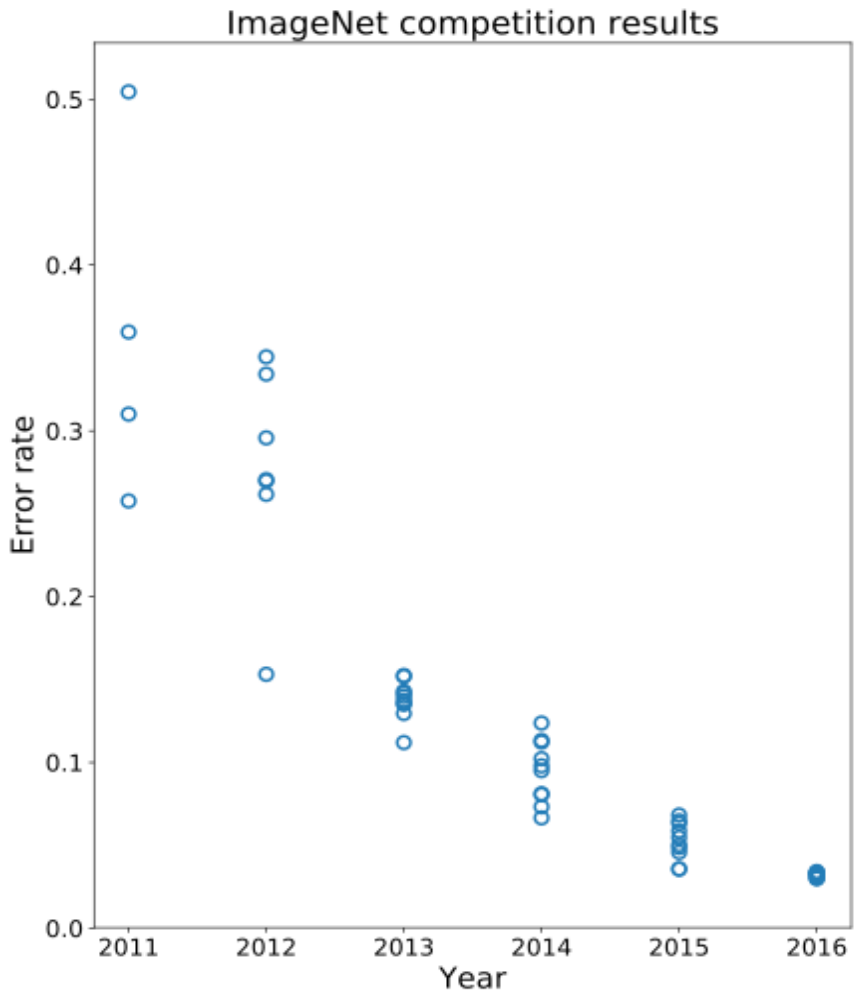


Image classification

for large scale image recognition" by



In 2017 29 out of the 38 groups got error rates below 5%. The anticipated next challenge will involve 3d data.

(https://en.wikipedia.org/wiki/ImageNet)

VGG net - architectures

Network architectures the VGG-team studied

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Subtract the mean of each channel RBG, computed over the entire training data set.

Almost all convolutions are 3x3 with stride 1 and zero-padding 1, i.e. convolutions do not change the size of the input!

Maxpooling is performed in 2x2 windows with stride 2. It reduces the x- and y-dimension by a factor of 2.

The remaining structure is 7x7x512 and gets vectorized to be fed into the fully connected (FC) layers.

Hidden linear layers are followed by an activation function.

VGG net - architectures

The last layer in the VGG architecture was called *soft-max*. What is that?

Representation of a classification result: $y \in \mathbb{R}^c$, where c is the number of classes. Identification: $y = e_i \Leftrightarrow$ the object belongs to class i

A network prediction like $y = (2, -1, 5, 1, -3, 2, 9)^T$ would be difficult to interpret.

Therefore, one exploits a soft-max layer $sm : \mathbb{R}^c \to \mathbb{R}^c$ defined via

$$(sm(x))_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

The result satisfies $sm(x)_i \geq 0 \; \forall i, \quad \sum_{j=1}^{c} (sm(x))_j = 1$ and therefore admits an interpretation as probabilities!

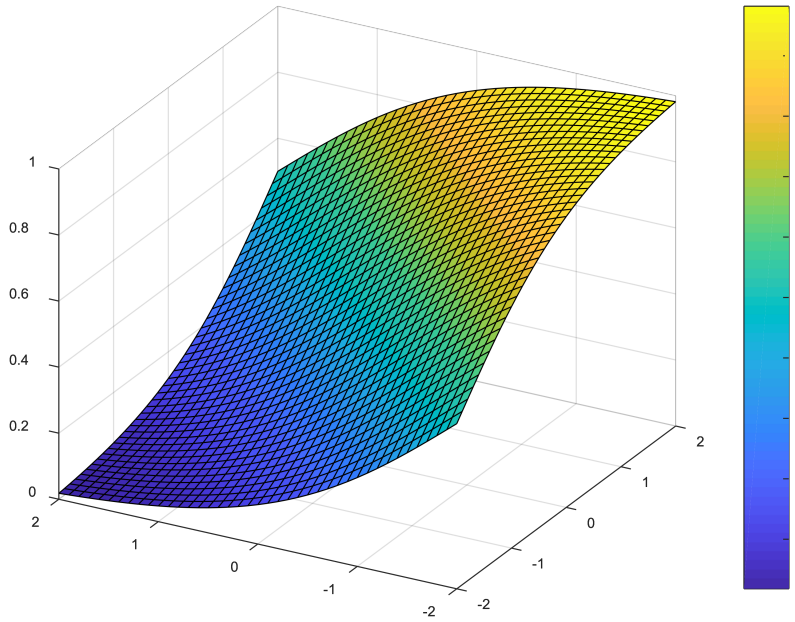
VGG net - architectures

The soft-max layer never yields unit normal vectors (why?).

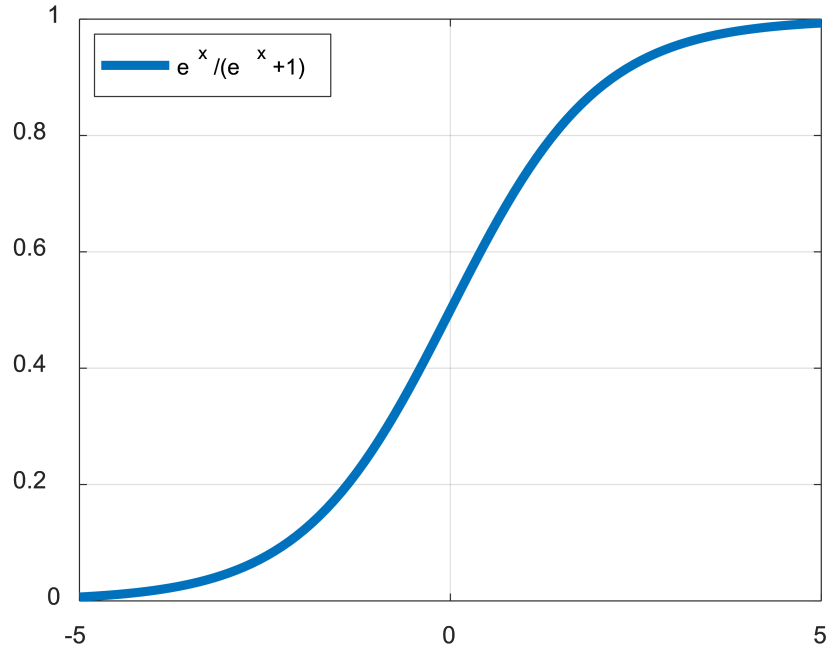
What is the derivative of the soft-max layer?

Visualizations

x_1-component in 2d



x_1-component for fixed x_2=0

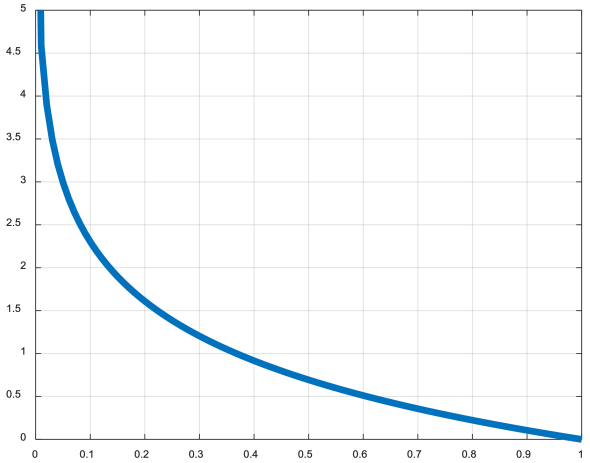


VGG net - architectures

What loss function does one use in image classification?

Most common: Cross entropy loss

$$\begin{aligned}\mathcal{L}(z, y) &= -\sum_i y_i \log(z_i), \\ &= -\log(z_i) \quad \text{if } y = e_i\end{aligned}$$



When combined with a soft-max

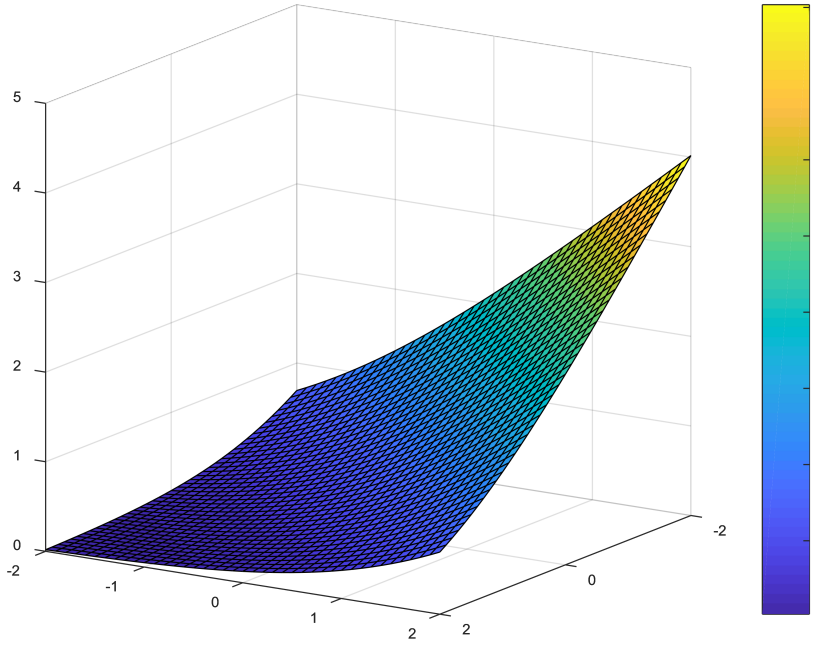
$$\mathcal{L}(sm(x), y) = -\log\left(\frac{e^{x_i}}{\sum_j e^{x_j}}\right), \quad \text{if } y = e_i$$

$$= -x_i + \log\left(\sum_j e^{x_j}\right),$$

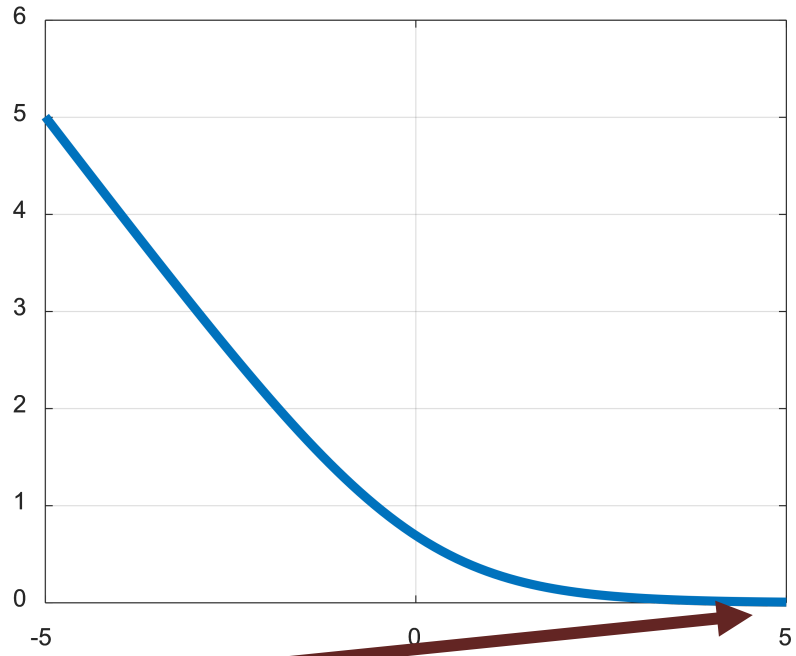
VGG net - architectures

Visualization of $\mathcal{L}(sm(x), y) = -\log\left(\frac{e^{x_i}}{\sum_j e^{x_j}}\right), \quad \text{if } y = e_i$

in 2d



For fixed x_2=0



Careful! This function keeps decreasing! Without things like quadratic regularization the optimization problem you solve during training might not have a minimizer!

VGG net - architectures

Returning to the network architectures the VGG-team studied

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Interesting questions:

- Where do most of these parameters come from?
- Is the architecture applicable to images of arbitrary size?

Following some discussions from the paper:

- Previous works have utilized fewer, but larger filters. What are possible advantages of more smaller layers?
- Important term for such a discussion: What is the *receptive field* of the neural network?

VGG net - training

The architecture is trained using SGD with momentum.

Minibatch size: 256 Momentum size: 0.9 Weight decay: $5 \cdot 10^{-4}$

Initial learning rate: 10^{-2}, decreased by a factor of 10 whenever the validation loss stopped improving. After this has been done 3 times, the iteration was stopped (after ~74 epochs).

Initialization from the paper: First train model A, use for initializing and training B, use for initializing and training C, ..., ... and training E.
Author's remark: A suitable initialization (as we discussed) also works!

"To obtain the fixed-size 224×224 ConvNet input images, they were randomly cropped from rescaled training images (one crop per image per SGD iteration). To further augment the training set, the crops underwent random horizontal flipping and random RGB color shift."

VGG net - testing

The testing procedure is also fine-tuned:

Rescale + create flipped image

size Q – at least 224

Feed into the network with FC layers “converted” to large 7x7 and 1x1 convolutional layers

Average the results (spatially as well as over the two images)

Additional possibilities: multiple crops, and multiple scales!

VGG net - testing

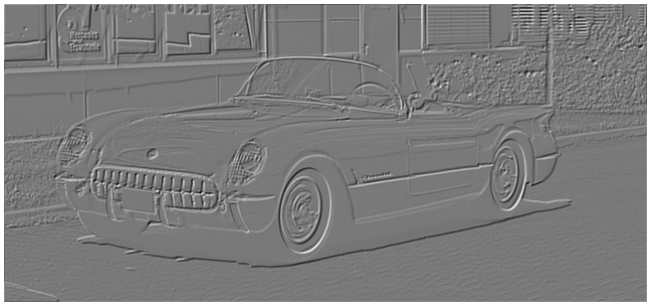
Evaluation in terms of top 5-validation error in %

Architecture E, conversion of fc into convs, testing at a scale of 384 pixels:	8%
Architecture E, testing at multiple scales:	7.5%
Architecture E, testing for multiple crops of size 224 for different scales:	7.4%
Architecture E, testing for multiple crops, and the full images (dense)	7.1%
Averaging the results of Architectures D and E, testing both for multiple crops, and the full images (dense) at different scales	6.8%
Winning Network of the ImageNet challenge 2014: GoogleLeNet (Ensemble of 7 networks)	6.7%
Winning Network of the ImageNet challenge 2017:	2.3%

VGG net - visualization

From this input

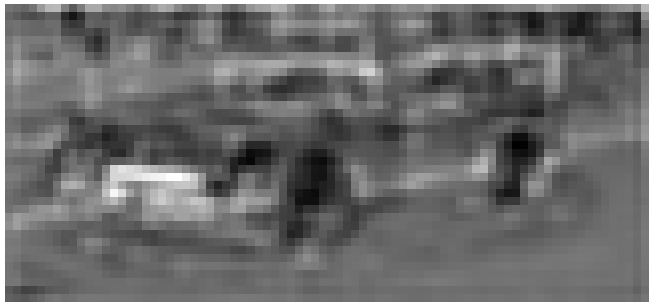
going



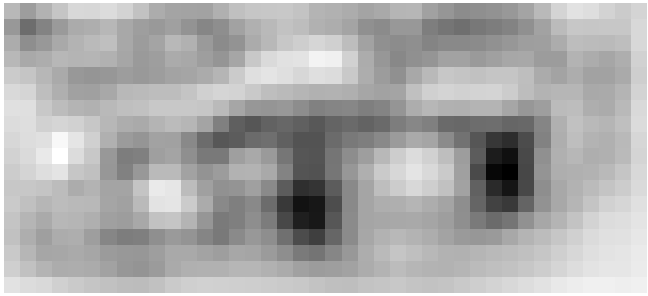
deeper

into

the



network



VGG net – final words

The VGG paper additionally contains an extension to object localization (via a bounding box). It utilizes the same network architecture as classification architecture D (16 Conv-layers).

It predict a bounding box described via center coordinates, width and height, and uses a Euclidean loss to train it. We can discuss some aspects in the lecture.

Conclusion of the VGG-Team: The basic idea of the network architecture does not really differ from the ones proposed by LeCun et al. 1989, but is much, much deeper.