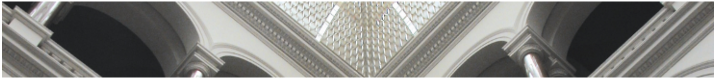
Convolutional Neural Networks

Philipp Seegerer
Technische Universität Berlin - Machine Learning Group
Beginners Workshop Machine Learning 2019



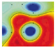
Technische Universität Berlin

Communicated by Dana Ballard

Backpropagation Applied to Handwritten Zip Code Recognition

Y. LeCun
B. Boser
J. S. Denker
D. Henderson
R. E. Howard
W. Hubbard
L. D. Jackel
AT&T Bell Laboratories Holmdel, NJ 07733 USA

The ability of learning networks to generalize can be greatly enhanced by providing constraints from the task domain. This paper demonstrates how such constraints can be integrated into a backpropagation network through the architecture of the network. This approach has been successfully applied to the recognition of handwritten zip code digits provided by the U.S. Postal Service. A single network learns the entire recognition operation, going from the normalized image of the character to the final classification.



Agenda

Convolution

Convolutional Neural Networks

Backpropagation

Why CNNs?

Application

Example

Summary

Agenda

Convolution

Convolutional Neural Networks

Backpropagation

Why CNNs?

Application

Example

Summary

Convolution

Convolution:

$$(f * g)(t) = \int f(s)g(t-s)ds$$

Discrete convolution:

$$[f * g]_t = \sum_{s=-\infty}^{\infty} f_s \cdot g_{t-s}$$

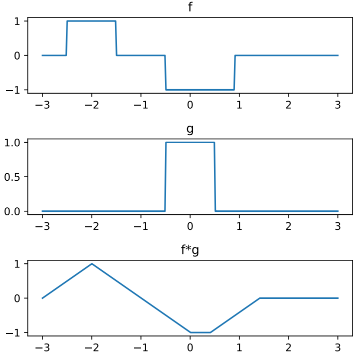


Figure: Example of 1D convolution.

Examples

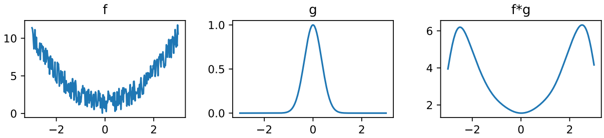


Figure: Smoothing with a Gaussian kernel.

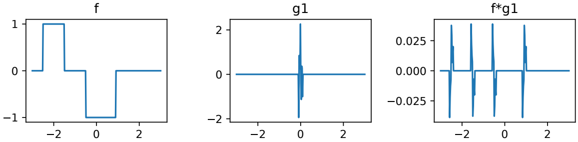
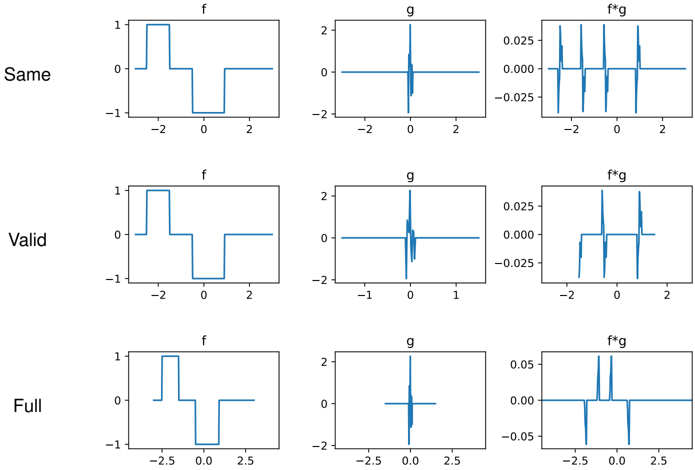


Figure: Differentiation.

Border Mode



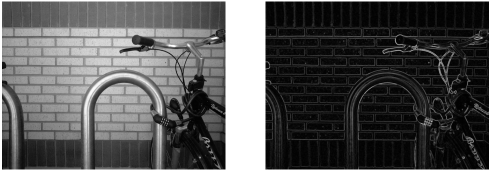
2D Convolution

Let x be a 2D tensor, w a filter kernel and y the filtered version of x:

$$y = x * w \tag{1}$$

$$y_{ij} = (x * w)_{ij} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} x_{i-m,j-n} \cdot w_{m,n} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} x_{i+m,j+n} \cdot w_{-m,-n} \ , \tag{2}$$

where (k_1, k_2) is the size of the kernel.



(a) Original image.

(b) After Sobel filtering.

Figure: Sobel filtering as example of 2D convolution. *Images courtesy of Davidwkennedy, CC BY-SA 3.0 licence,* *https://commons.wikimedia.org/w/index.php?curid=8904806*

Agenda

Convolution

Convolutional Neural Networks

Backpropagation

Why CNNs?

Application

Example

Summary

Convolutional Layer

Let z be the preactivation in a convolutional layer (assume only 1 output channel):

$$z_{ij} = (x * w)_{ij} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \sum_{c=1}^{C} x_{i+m,j+n,c} \cdot w_{-m,-n,c} + b_{ij} \quad , \tag{3}$$

where C is the number of channels of x and w.

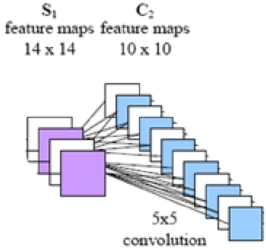


Image taken and adapted from the website of **Parallel Architecture Research Eindhoven**

Weight Sharing

Express convolution as matrix multiplication (as in fully connected layer in MLP)

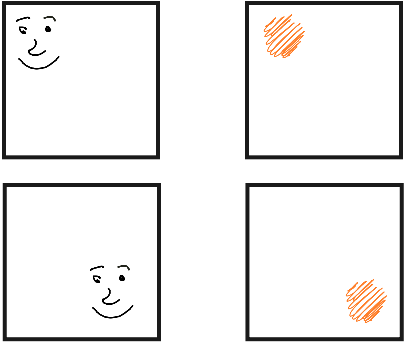
$$y = v * x = W^T x$$

$$W^\top = \left(\begin{array}{cccccc|cccccc}
v_{11}^1 & v_{11}^2 & v_{11}^3 & 0 & 0 & 0 & v_{12}^1 & v_{12}^2 & v_{12}^3 & 0 & 0 & 0 \\
0 & v_{11}^1 & v_{11}^2 & v_{11}^3 & 0 & 0 & 0 & v_{12}^1 & v_{12}^2 & v_{12}^3 & 0 & 0 \\
0 & 0 & v_{11}^1 & v_{11}^2 & v_{11}^3 & 0 & 0 & 0 & v_{12}^1 & v_{12}^2 & v_{12}^3 & 0 \\
0 & 0 & 0 & v_{11}^1 & v_{11}^2 & v_{11}^3 & 0 & 0 & 0 & v_{12}^1 & v_{12}^2 & v_{12}^3 \\
\hline
v_{21}^1 & v_{21}^2 & v_{21}^3 & 0 & 0 & 0 & v_{22}^1 & v_{22}^2 & v_{22}^3 & 0 & 0 & 0 \\
0 & v_{21}^1 & v_{21}^2 & v_{21}^3 & 0 & 0 & 0 & v_{22}^1 & v_{22}^2 & v_{22}^3 & 0 & 0 \\
0 & 0 & v_{21}^1 & v_{21}^2 & v_{21}^3 & 0 & 0 & 0 & v_{22}^1 & v_{22}^2 & v_{22}^3 & 0 \\
0 & 0 & 0 & v_{21}^1 & v_{21}^2 & v_{21}^3 & 0 & 0 & 0 & v_{22}^1 & v_{22}^2 & v_{22}^3 \\
\hline
v_{31}^1 & v_{31}^2 & v_{31}^3 & 0 & 0 & 0 & v_{32}^1 & v_{32}^2 & v_{32}^3 & 0 & 0 & 0 \\
0 & v_{31}^1 & v_{31}^2 & v_{31}^3 & 0 & 0 & 0 & v_{32}^1 & v_{32}^2 & v_{32}^3 & 0 & 0 \\
0 & 0 & v_{31}^1 & v_{31}^2 & v_{31}^3 & 0 & 0 & 0 & v_{32}^1 & v_{32}^2 & v_{32}^3 & 0 \\
0 & 0 & 0 & v_{31}^1 & v_{31}^2 & v_{31}^3 & 0 & 0 & 0 & v_{32}^1 & v_{32}^2 & v_{32}^3
\end{array}\right)$$

W has 144 entries, but only 18 effective parameters v_{ij}^t.

Viewpoint Invariance

- Use same feature detector at different positions
- Equivariant activations on translation: feature map changes in the same way as the input



Subsampling by Local Pooling

- Summarize neighboring neurons in feature map to a single value
- Usually non-overlapping regions $\longrightarrow$ stride
- Reduce resolution of feature map
- Translational invariance
- Examples for pooling functions: **maximum**, sum, average

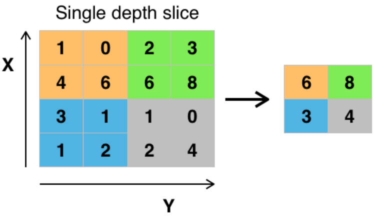


Figure: Max-pooling. *Image courtesy of Aphex34, CC BY-SA 4.0 licence,* *https://commons.wikimedia.org/w/index.php?curid=45673581*

Convolutional Neural Network

- A convolutional neural network alternates several stages of
  - Convolutions (convolve the image with different filters)
  - Pooling (pool features at neighboring locations)
- For classification, a fully connected layer followed by a softmax is added as last layer
- Implementation: `batch_size` × `num_channels` × `height` × `width`

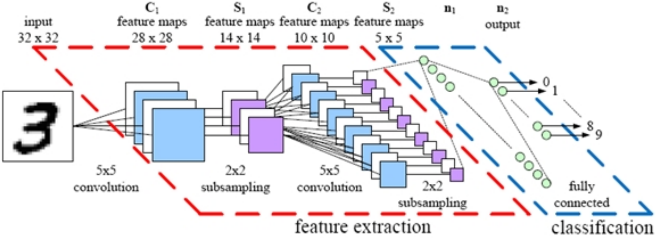


Figure: Prototypical CNN for classification. *Image taken from* ***Parallel Architecture Research Eindhoven***

Convolutional Neural Network (cont.)

Softmax:

- Transform “logits” to probabilities
- Multidimensional extension of logistic sigmoid

$$\xi(\vec{y^L})_i = \frac{e^{y_i^L}}{\sum_{\forall j} e^{y_j^L}} \quad ,$$

where $\vec{y^L}$ is the output of the neural network (each element corresponds to one class)

Cross-entropy loss:

- A.k.a “negative log-likelihood loss”
- Used for classification
- One-hot encoding $\vec{t}$ of target

$$E(\vec{y^L}, \vec{t}) = -\sum_i t_i \cdot \log(y_i^L)$$

Agenda

Convolution

Convolutional Neural Networks

Backpropagation

Why CNNs?

Application

Example

Summary

Forward and Backward Pass

Forward Pass

$$z^l_{ij} = (x^l * w^l)_{ij} = \sum_{m=0}^{k^l_1-1} \sum_{n=0}^{k^l_2-1} \sum_{c=1}^{C^l} x^l_{i+m,j+n,c} \cdot w^l_{-m,-n,c} + b^l_{ij} \tag{4}$$

Backward Pass Let $\delta^l = \frac{\partial E}{\partial z^l_{ij}}$ denote the error w. r. t. to cost function E:

$$\delta^l = w^l * \delta^{l+1} \tag{5}$$

Agenda

Convolution

Convolutional Neural Networks

Backpropagation

Why CNNs?

Application

Example

Summary

Translational Invariance

Meaning of an input is unchanged with respect to a particular transformation of it.

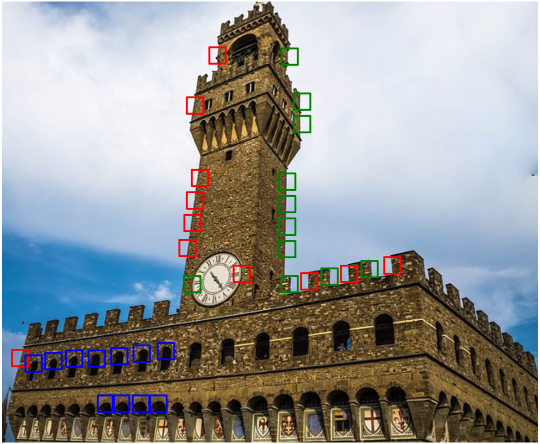
lighthouse

lighthouse

lighthouse

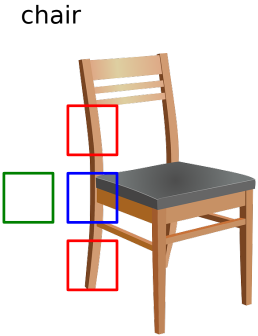
Homogeneity

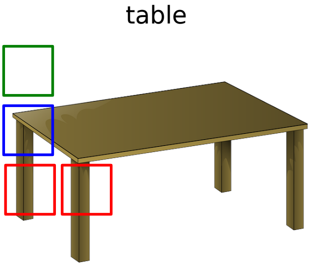
Same feature detectors are needed for different subsets of input dimensions (e.g. image patches).



Depth

Meaning of an input is determined by how particular subparts of the input are interrelated, and not by the presence or absence of the subparts themselves.





Agenda

Convolution

Convolutional Neural Networks

Backpropagation

Why CNNs?

Application

Example

Summary

Handwritten Digit Recognition

– LeCun et al. 1998

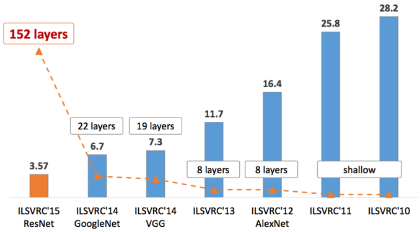
– Handwritten digits from MNIST

Figure: Examples from the MNIST database. *Image courtesy of Josef Steppan, CC BY-SA 4.0 licence,* *https://commons.wikimedia.org/w/index.php?curid=64810040*

Image Classification

ImageNet (12 millions of 224 $\times$ 224 images, 1000 classes):

(a) Examples from ImageNet



(b) ILSVRC. *Image from* `https://medium.com/`

Image Classification

ImageNet (12 millions of 224 $\times$ 224 images, 1000 classes):

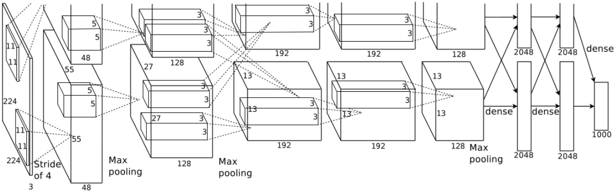


Figure: CNN architecture by Krizhevsky, Sutskever, and Hinton 2012.

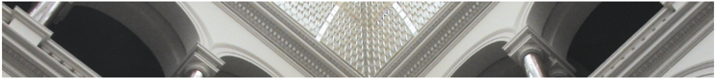
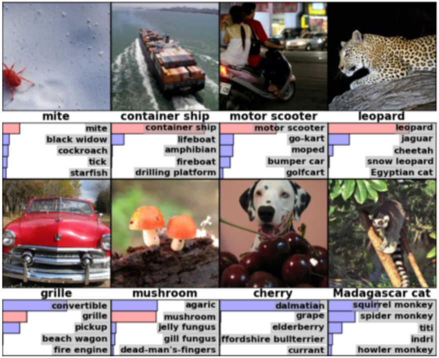
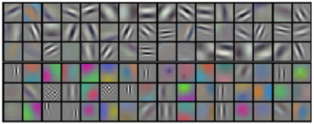


Image Classification

AlexNet Predictions



SuperVision conv. kernels



ILSVRC2012 Results

Team name	Error (5 guesses)
SuperVision	0.15315
SuperVision	0.16422
ISI	0.26172
ISI	0.26602
ISI	0.26646
ISI	0.26952
OXFORD_VGG	0.26979
XRCE/INRIA	0.27058
OXFORD_VGG	0.27079
OXFORD_VGG	0.27302
...	...

Images taken from Krizhevsky, Sutskever, and Hinton 2012. List of results taken from ILSVRC 2012.

.

Agenda

Convolution

Convolutional Neural Networks

Backpropagation

Why CNNs?

Application

Example

Summary

Example

Jupyter Notebook

Agenda

Convolution

Convolutional Neural Networks

Backpropagation

Why CNNs?

Application

Example

Summary

Summary

- Convolutional layers
  - reduce number of parameters (weight sharing)
  - translational invariance
  - model locally correlated structure
- CNNs are suitable for e. g. :
  - 1D: text, time series, DNA sequences
  - 2D: images
  - 3D: videos, molecules, volumetric images (CT, MRI)
  - ...

References

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: **Proceedings of the IEEE conference on computer vision and pattern recognition**, pp. 770–778.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: **Advances in neural information processing systems**, pp. 1097–1105.

LeCun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: **Proceedings of the IEEE** 86.11, pp. 2278–2324.

Szegedy, Christian et al. (2015). "Going deeper with convolutions". In: CVPR.

Technische Universität Berlin

Thank you!