Convolutional Networks

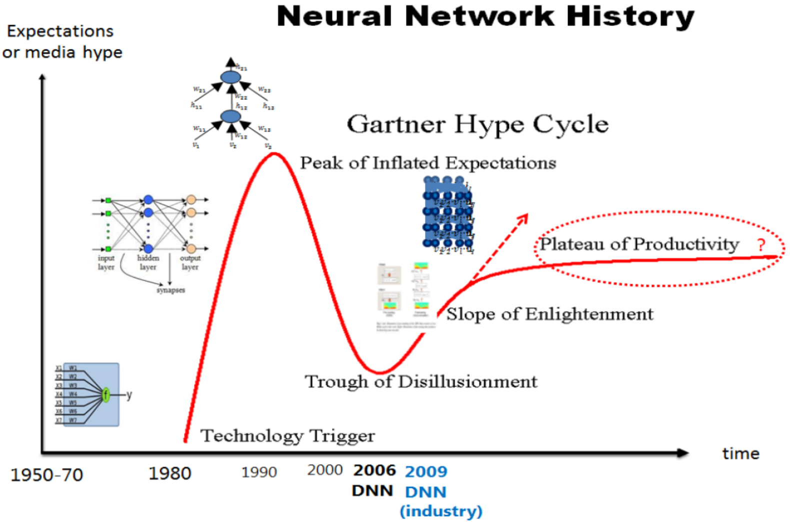
Yong Li

liyongforevercas@163.com
www.foreverlee.net

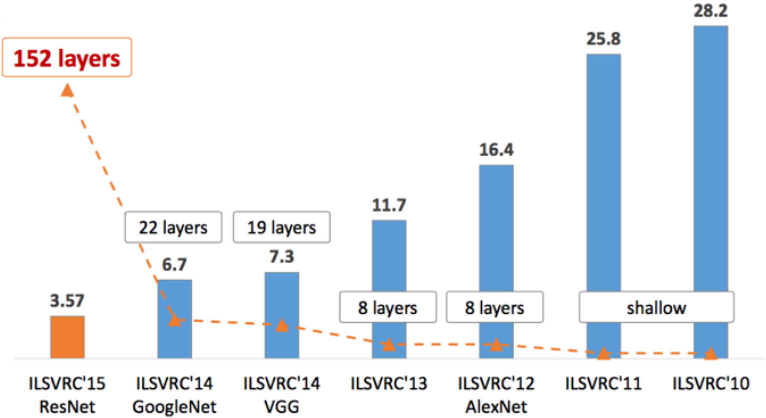
2017-04-09

Overview

Neural Network History



Convolutional Networks on ImageNet

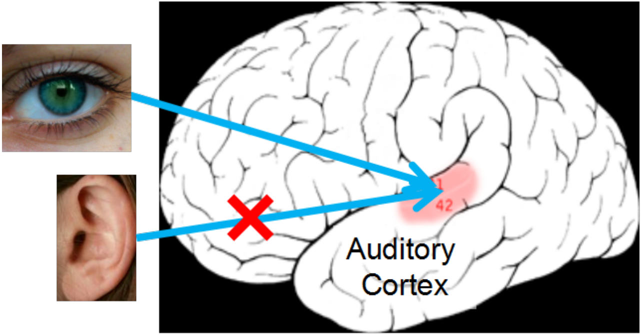


Cnn make 36.4% relative improvement in 2012.

Pic from: https://zhuanlan.zhihu.com/p/22094600

The Neuroscientific Basis for Convolutional Networks

- Layer structure.
- Sparse connection.



Data Types of Input for CNN

The data usually consists of several channels, each channel being the observation of a different quantity at some point in space or time.

- Audio waveform
- Fourier transform of audio
- CT scans
- Skeleton animation data
- Color image
- Color video data.

Convolution Operation

Convolution operation do operation as follows,

$$s(t) = \int x(a)w(t-a)da. \quad (1)$$

Convolution operation is typically denoted with an asterisk,

$$s(t) = (x * w)(t). \quad (2)$$

Example: laser senfor with noise $x(t)$ and a weighting function $w(a)$.

Convolution Operation

Discrete convolution is as follows,

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a). \tag{3}$$

two-dimensional convolution is as follows,

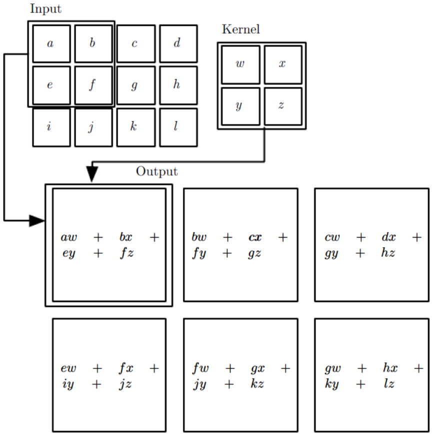
$$S(i,j) = (I * K)(i,j) = \sum_{m} \sum_{n} I(m,n)K(i-m, j-n). \tag{4}$$

Many neural network libraries implement a related function called the cross-correlation,

$$S(i,j) = (I * K)(i,j) = \sum_{m} \sum_{n} I(i+m, j+n)K(m,n). \tag{5}$$

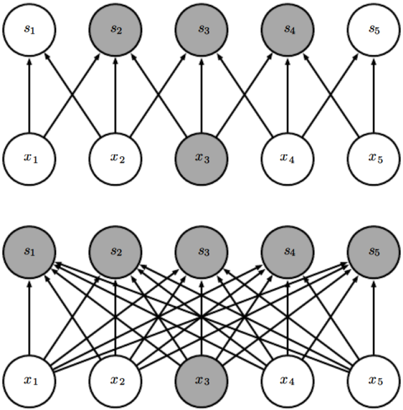
Motivation

Three important ideas: sparse interactions, parameter sharing equivariant and representations.



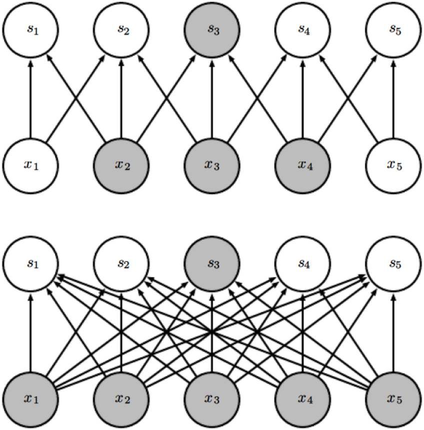
Sparse Connectivity

Viewed from below.

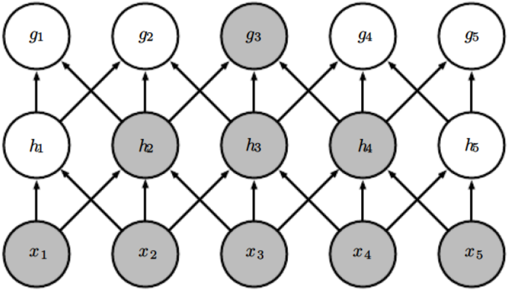


Sparse Connectivity

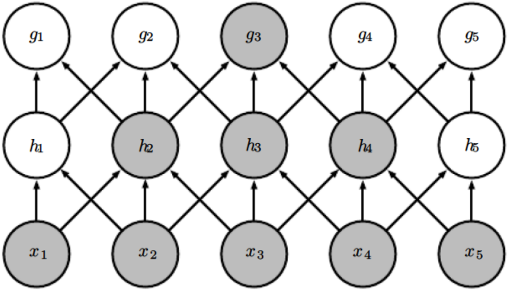
Viewed from above.



The receptive field



Parameter sharing



Equivariance to Translation

- A function is equivariant means that if the input changes, the output changes in the same way.
- Specifically, 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.

Efficiency of edge detection

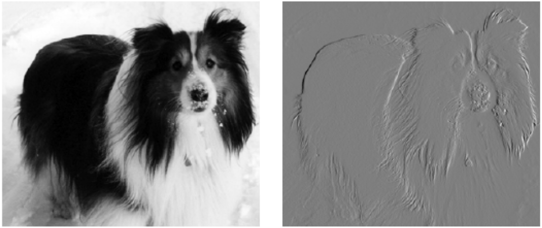


Image size: 320*280

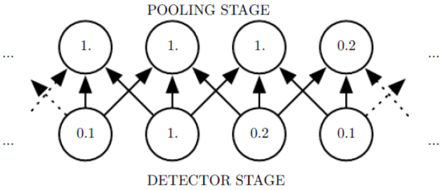
- For convolution: kernel [-1, 1], 319*280*3 operations (two multiplications and one addition per output pixel)
- Matrix multiplication would take 320*280*319*280

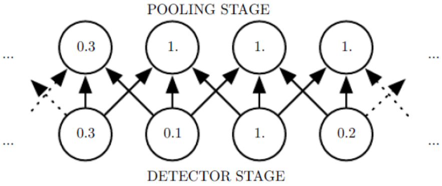
Pooling

Three stage: First, convolution; Second, nonlinear activation; Third pooling function.

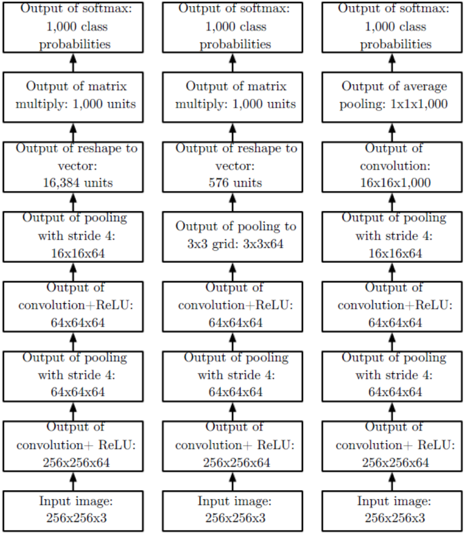
- A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs.
- Max pooling operation reports the maximum output within a rectangular neighborhood. pooling helps to make the representation become approximately
- invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount,

Pooling


POOLING STAGE
...
1.
1.
1.
0.2
...
...
0.1
1.
0.2
0.1
...
DETECTOR STAGE


POOLING STAGE
...
0.3
1.
1.
1.
...
...
0.3
0.1
1.
0.2
...
DETECTOR STAGE

Examples of architectures



Probability Analysis of Convolution and Pooling

- Weights for one hidden unit must be identical to the weights of its neighbor, but shifted in space.
- weights must be zero, except for in the small, spatially contiguous receptive field assigned to that hidden unit.

Convolution Function

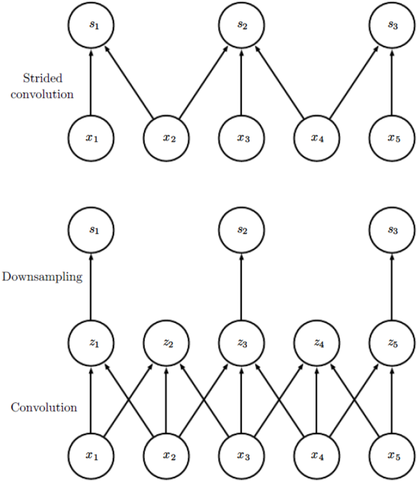
The convolution output is as follows,

$$Z_{i,j,k} = \sum_{l,m,n} V_{l,j+m-1,k+n-1} K_{i,l,m,n}, \tag{6}$$

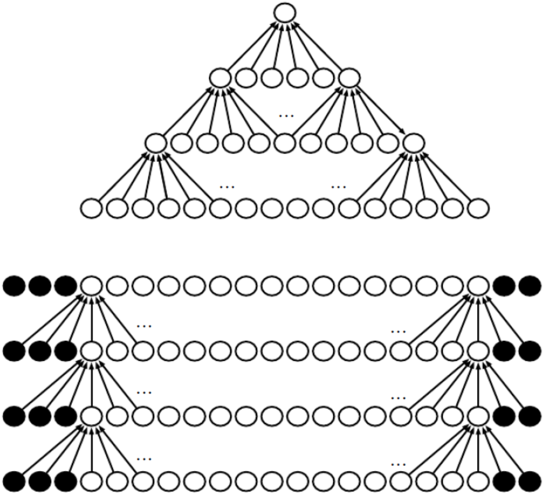
where i is the channel of the output. Downsampled convolution is as follows,

$$Z_{i,j,k} = c(K, V, s)_{i,j,k} = \sum_{l,m,n} [V_{l,(j-1)\times s+m,(k-1)\times s+n,} K_{i,l,m,n}]. \tag{7}$$

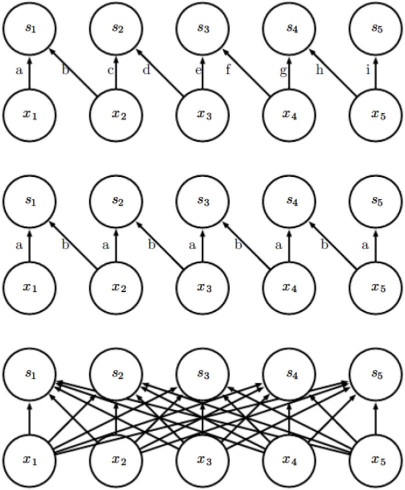
Convolution with a stride



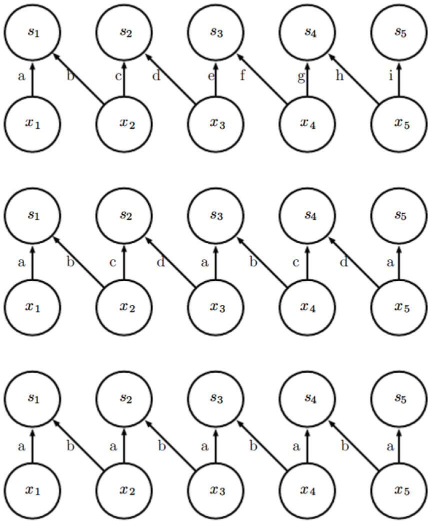
Padding



Local Connections, Convolution and Full Connections



Tiled Convolution



To train the network, we need to compute the derivatives with respect to the weights in the kernel. To do so, we can use a function

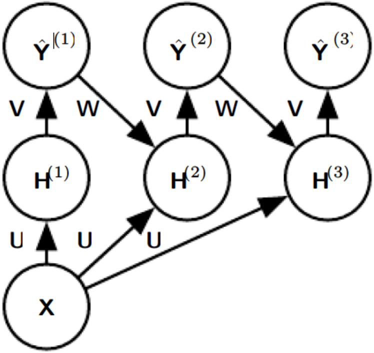
$$g(\mathbf{G}, \mathbf{V}, s)_{i,j,k,l} = \frac{\partial}{\partial K_{i,j,k,l}} J(\mathbf{V}, \mathbf{K}) = \sum_{m,n} G_{i,m,n} V_{j,(m-1)\times s+k,(n-1)\times s+l}. \tag{9.11}$$

If this layer is not the bottom layer of the network, we will need to compute the gradient with respect to $\mathbf{V}$ in order to back-propagate the error farther down. To do so, we can use a function

$$h(\mathbf{K}, \mathbf{G}, s)_{i,j,k} = \frac{\partial}{\partial V_{i,j,k}} J(\mathbf{V}, \mathbf{K}) \tag{9.12}$$

$$= \sum_{\substack{l,m \\ \text{s.t.} \\ (l-1)\times s+m=j}} \sum_{\substack{n,p \\ \text{s.t.} \\ (n-1)\times s+p=k}} \sum_q K_{q,i,m,p} G_{q,l,n}. \tag{9.13}$$

Structured Outputs



Efficient Convolution Algorithms

- Convolution is equivalent to converting both the input and the kernel to the frequency domain using a Fourier transform, performing point-wise multiplication of the two signals, and converting back to the time domain using an inverse Fourier transform.
- Accelation by matrix factorization.

Random or Unsupervised Features

Three basic strategies for obtaining convolution without supervised learning

- One is to simply initialize them randomly.
- Another is to design them by hand, s.t. detect edges.
- One can learn the kernels with an unsupervised criterion.

It may provide an inexpensive way to choose the architecture of a convolutional network.

Conclusion

References

Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016)
Deep Learning
MIT Press Chap9. P321-362

All the figures come from the textbook.

Conclusion

1. Neural Network History
2. The Neuroscientific Basis for Convolutional Networks
3. Convolutional Networks
   - Data Types
   - The Convolution Operation
   - Important Ideas
   - Pooling
   - Probability Analysis of Convolution and Pooling
   - Variants of the Basic Convolution Function
   - Structured Outputs
   - Efficient Convolution Algorithms
   - Random or Unsupervised Features

Wechat Official Accounts: machinelearning
Wechat Admin ID: ml-admin

liyongforevercas@163.com
www.foreverlee.net