CONVOLUTIONAL NETWORKS

(MODULE 4)

Chakravarthy Bhagvati
School of Computer and Information Sciences
University of Hyderabad

14 JUNE 2018

OVERVIEW

- Introduction to *Convolution*
- Two applications of convolution
  - spatial image processing
  - linear systems
- Neural Network perspective
- Convolutional Networks
- Recent Deep Convolution Networks
- Variants
- Summary and Conclusion

CONVOLUTION

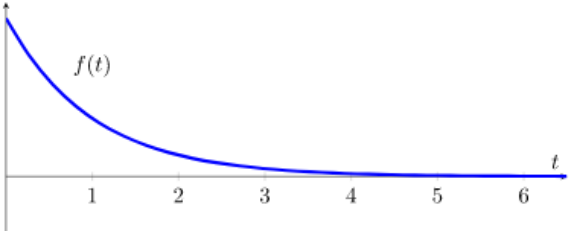
One of the *most important* operations in
image processing, pattern recognition and computer vision

- Operation defined on two functions producing another function which is normally seen as modification of one of the input functions
- Integral of the product of the two input functions after reversing one of the two and shifting it across the other
- Mathematically, convolution of $f(x, y)$ with $h(x, y)$ is

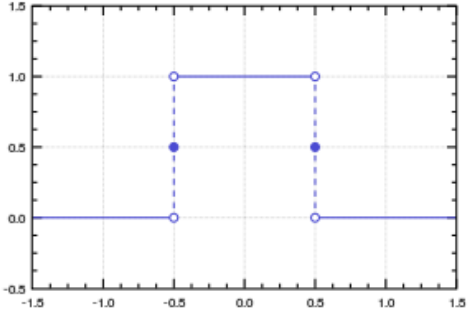
$$g(x, y) = \int_{-\infty}^{+\infty}\int f(x, y)h(\tau - x, \nu - y)\, d\tau\, d\nu$$

CONVOLUTION EXAMPLE

Input $f(x)$



Input $h(x)$



Convolution Output: $f(x) \star h(x)$

WHY CONVOLUTION

There are two main reasons for the popularity of convolution

- Forms the basis for *Spatial Image Processing*
- Forms the mathematical foundation for *Linear Systems*

Spatial Image Processing

- Modify pixel values based on the values of the neighbouring pixels
- Image function is $f(x, y)$; Neighbouring pixels and their contributions $h(x, y)$
- Output $g(x, y)$ is modified $f(x, y)$ given by

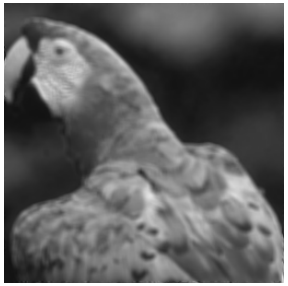
$$g(x, y) = f(x, y) \star h(x, y)$$

SPATIAL OPERATORS

$\star$

1	1	1
1	1	1
1	1	1

$=$



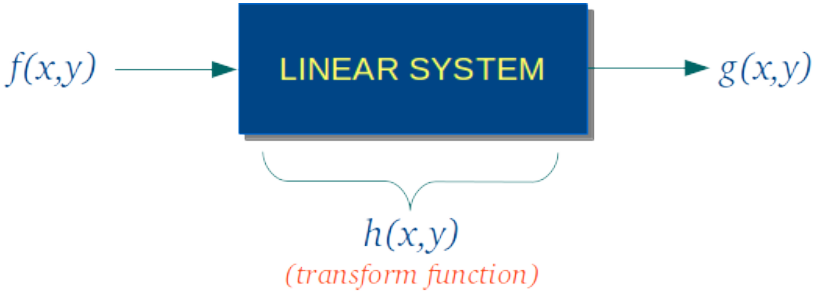
$\star$

-1	-2	-1
0	0	0
1	2	1

$=$

LINEAR SYSTEMS

- Block diagram of a linear system

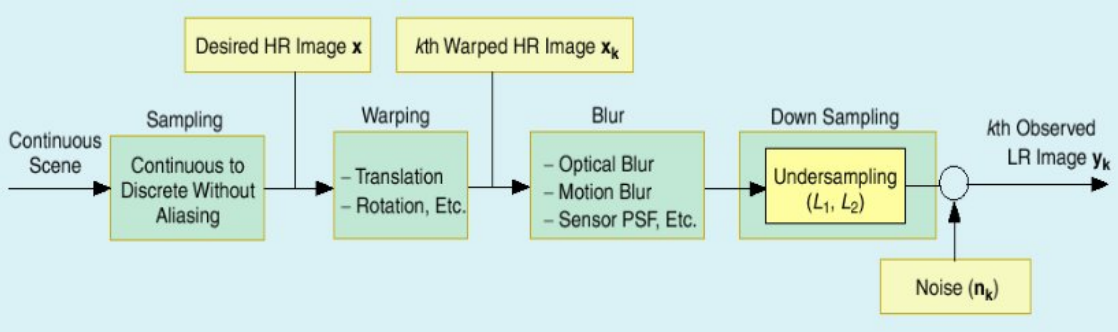


- Linear system model is given by

$$g(x, y) = f(x, y) \star h(x, y)$$

LINEAR SYSTEMS ...

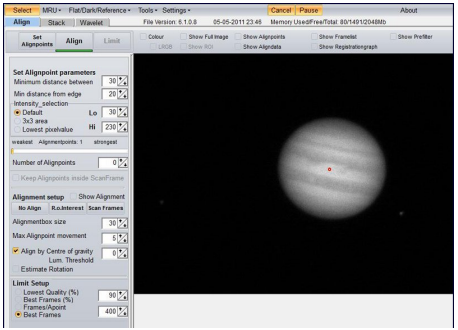
- Linear systems are used in *super-resolution*: generating high-resolution image from a set of low-resolution images. The model used is

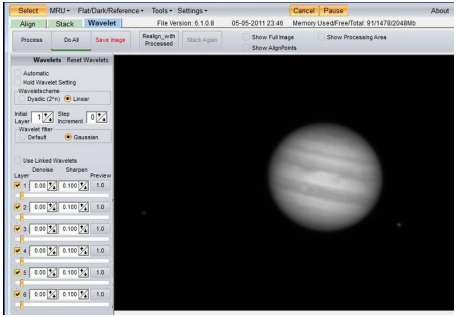


$$y_k = D \star B_k \star M_k \star x + \eta_k \qquad \text{for } 1 \leq k \leq p$$

M_k is a warp matrix, B_k is the blur matrix, D is a sub-sampling matrix and η_k is the noise matrix

SUPER-RESOLUTION





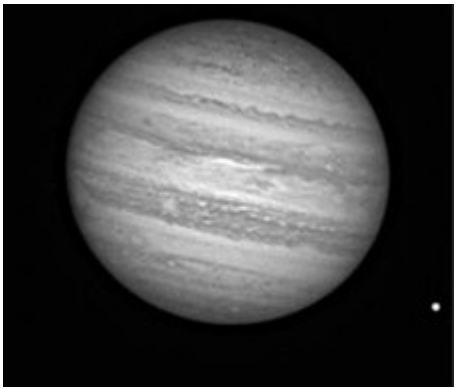
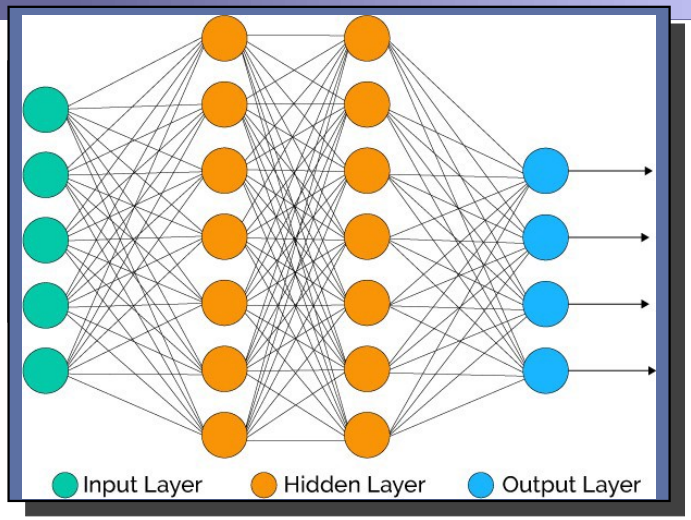


IMAGE RESTORATION

- Linear systems are used in *Image Restoration*
  - Image is $g(x, y)$, distorting function is $h(x, y)$ and the original *clean (unknown)* image is $f(x, y)$
  - Restoration is done by removing the effects of convolution by $h(x, y)$
- Image acquisition is almost always modelled as a linear system in which the *ideal* world-scene is transformed into a digital image after convolution with the transform function of the camera or any image input device
  - the device transfer function is called *Point Spread Function* (PSF)

NEURAL NETWORK

- Consists of neurons in layers
- Fully interconnected layers
- No lateral connections within layers
- One input and one output layer



Problems

- No 2D structure of images
- Too many parameters to learn
- No local features (convolution)

CONVOLUTION LAYER!

Proposed by Yann LeCun in late 1980's

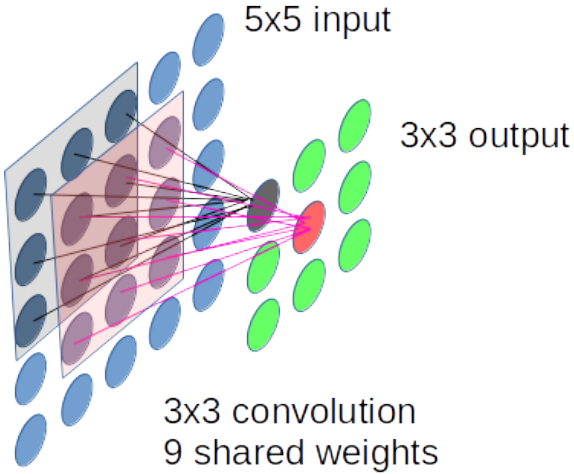
The basic ideas of *Convolution Neural Network* (CNN)

- Remove the fully-connected nature of neural networks
- Implement convolution operation in a neural network layer

Key features of Convolution Neural Network

- Local connections
- 2D structure
- Weight-sharing
- Sliding window

CONVOLUTION LAYER



Note how 2D structure, local weights and weight sharing are used

0	0	0	0	0	0
0	105	102	100	97	96
0	103	99	103	101	102
0	101	98	104	102	100
0	99	101	106	104	99
0	104	104	104	100	98

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

320				

CNN

So, what's the big deal about
convolution layers as against
standard implementation of convolution?

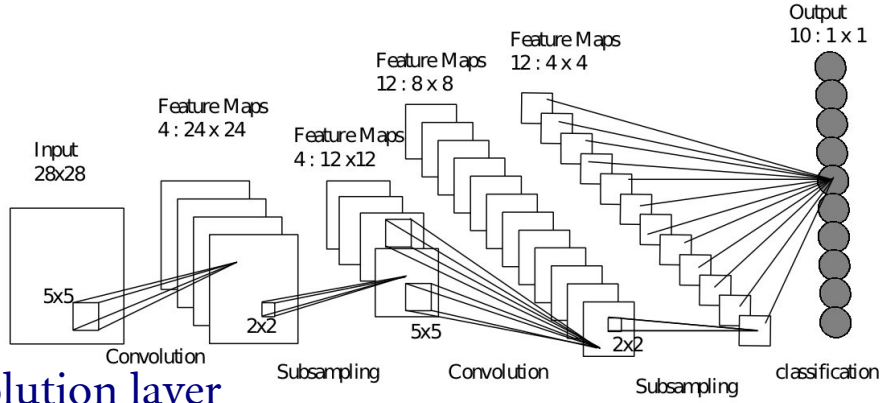
- In convolution, we give the mask/filter coefficients – in effect, we are designing the *features*
- In a convolution neural network, the coefficients are *learned*, that is, the features are not given by us but learned by the network
- It is a *fundamentally different* approach!

CONVOLUTION NEURAL NETWORKS

A Convolutional Neural Network (CNN) consists of

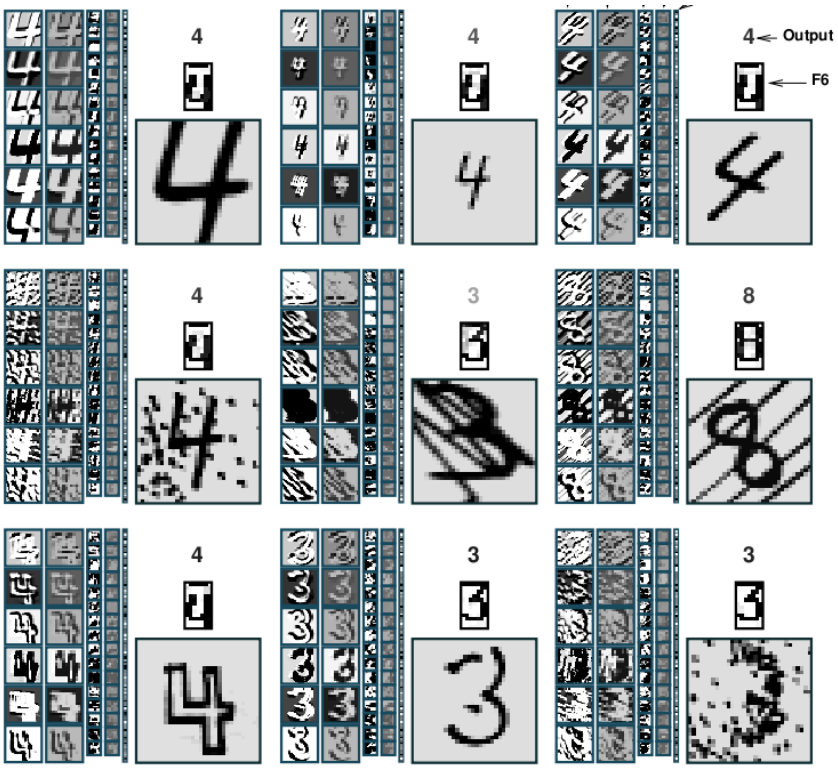
- convolution layers
  - each node has inputs from nodes located in a small *kernel size* neighbourhood
  - every node shares weight assignment of all nodes
  - output is a *feature map*
- sub-sampling layers
  - each node is connected to 2×2 area in a feature map
  - each node performs averaging of the input
- fully connected layers
  - function as a pattern classifying network
  - usually the last layer before the output layer

LeCun Network — LeNet-5



- Layer 1 is a convolution layer
  - kernel size 5×5
  - four feature maps each of size 24×24
- Layer 2 is a sub-sampling layer — reduces feature maps to 12×12
- Layer 3 is a convolution layer — twelve feature maps of size 8×8
- Layer 4 is sub-sampling layer — size reduction to 4×4
- Layer 5 is fully connected to 10 outputs

DIGIT RECOGNITION



FACE DETECTION

TRAINING A CNN

- A sample is presented to the network, its output is compared with the desired output and error is used for updating the weights (*same as any regular ANN*)
- *Calculating the output of a single node j*

$$X_j = f(\sum_{i \in M_j} w_{ij} X_i + B_j)$$

- *Calculating error*

$$E = \frac{1}{2} \sum_{i=1}^{N} (T^i - X^i)^2$$

- *Calculating gradient*

$$\Delta w = -\eta \frac{\partial E}{\partial w}$$

- *Updating the weights*

$$w = w + \Delta w$$

Back Propagation Algorithm

CNN AS AN EDGE DETECTOR

- *Architecture*: input layer connected to convolution layer; output is feature map
- *Convolution Layer*: kernel size 3×3
- The CNN is trained by giving an image as input and its corresponding Sobel edge detected image as target output
- Initial weights: -0.25 to $+0.25$ and Learning Rate: 0.0001

Initial Weights

$$\begin{matrix} 0.029512 & 0.051186 & -0.104802 \\ -0.199802 & -0.014951 & 0.111171 \\ 0.133135 & -0.010364 & 0.113527 \end{matrix}$$

Final Weights

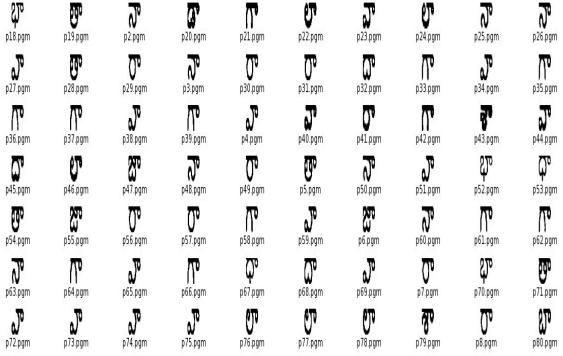
$$\begin{matrix} -1.049056 & 0.075150 & 0.941462 \\ -1.954816 & 0.074961 & 1.887997 \\ -1.027599 & 0.016904 & 1.039174 \end{matrix}$$

(after 40 epochs)

PATTERN CLASSIFICATION

We again designed a simple experiment: can a CNN separate images with horizontal edges from those without?

- *Architecture*: one convolution layer (3×3), a single feature map and a fully connected layer to a single output
- Training is done with the samples shown



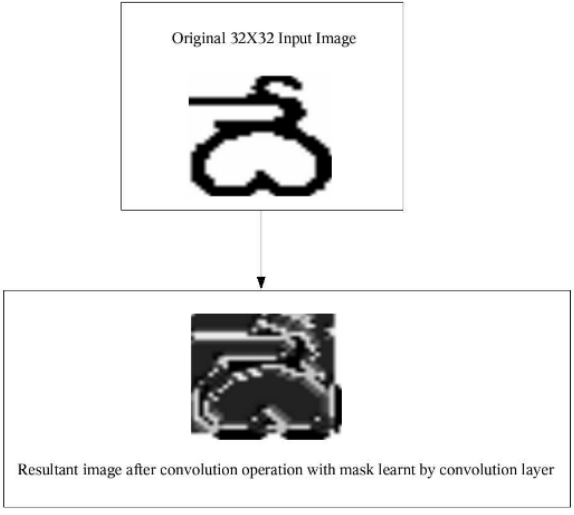
Positive Samples

Negative Samples

MORE DETAILS

- *Performance:* 82% accuracy

What did the cnn use as a feature?



SUPERVISED LEARNING

Nearest Neighbour: simple but not robust

Decision Trees: discover rules but too heavily dependent on features; restricted ability to generalise

Neural Networks: powerful but require large training sets; overfitting data

Support Vector Machines: simple, good general-purpose classifier; good generalisation but is a linear classifier

Evolutionary Algorithms: biologically inspired; success in various domains; generalisation not easy to understand

Learning proved very successful over the last few decades

Train a classifier by giving a large number of inputs and corresponding class labels: Supervised Learning

WHERE ARE WE STUCK?

Conventional Pattern Recognition

- Handcrafted features (usually *low-level*)
- Classifier has to do the hard work
- Makes mistakes which are not very human
- Often hard to predict performance

Same for Machine Learning too!

Syntactic Pattern Recognition

- Hand-crafted features (usually *high-level*)
- Classifier *can be* easy
- Feature extractor has to do the hard work

(and actually couldn't!!)

Common Problems

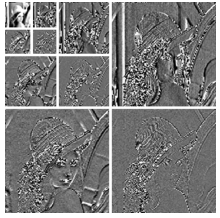
Insufficient labelled samples
Lack of computing power

REPRESENTATION

- Characterise *objects* (high-level concepts) in terms of *features* (low-level)
- Features should be directly computable from the input
  - Edges, Lines and Corners
  - Textures, Areas, Aspect Ratios
  - Chain codes, Histograms
- Popular features: *HOG, SIFT, SURF, …*
- *Unsupervised Learning (Clustering)* to use good features







CNN'S AND FEATURE MAPS

- The output of a convolution layer is called a *Feature Map*
  - it is the same as the convolved output $g(x, y)$
- But CNN's use *multiple* feature maps (LeNet-5 has 4, 4 and 12 feature maps at different layers)
- Multiple feature maps allow several convolutions to be performed in parallel to give different outputs which can be combined in subsequent layers
  - e.g., smoothing and edge detectors may be learnt and combined for a wavelet effect

CNN's provide ability to learn and do away with handcrafting features

Let's begin to summarise …

IMAGES AND NEURAL NETWORKS

- Neural networks have been shown to learn, complex, high-dimensional, non-linear mappings in Pattern Recognition applications
- There are some problems too, especially when used with images
  - Input dimensions are large and therefore number of interconnections and weights is large — training complexity
  - Fully connected networks — no translation or local invariances
  - Fully connected networks — input can be in any order, but images have strong 2D correlations and local structures

CONVOLUTION NEURAL NETWORKS

LeCun popularized *Convolutional Neural Networks* to make Neural Networks specially suited for processing images

- Local receptive fields
  - each node connected only to a limited set of nodes
  - not fully connected
- Shared weights
  - provides translation invariances
  - reduces number of weights for training
- Spatial sub-sampling
  - robustness against local distortions

...AND FINALLY!

- CNN's give the ability to learn features
- Moreover, these features could be *interpreted* - thanks to convolution
  - remember that ANN 'features' are only weights which we find hard to interpret
- CNN's directly and indirectly paved the way to the Deep Learning revolution

THANK YOU

OVERVIEW

- Developments in CNNs
- Some Famous CNNs
- Variants of CNNs

CNN DEVELOPMENTS

2009: Deep RNNs win three handwriting competitions at ICDAR 2009

- non a-priori linguistic knowledge
- simultaneous segmentation and recognition

High performance in detecting human actions by a 3D CNN

2010: Robustness through Dropout and distortions

Ensemble of Max-Pooling CNNs finally achieve 99.8% on MNIST

2012: Breakthrough year with several contests won by CNNs

VARIANTS OF CNN

- Transfer Learning
- Unsupervised Learning vs. Supervised Learning
- Better feature extraction demonstrated by yielding better results than SIFT on many vision tasks
- Noise in deeper layers to force abstract image representations
- Newer optimizations such as RMSProp, AdaGrad, AdaDelta, …

BEYOND CONVOLUTION

- Spiking neurons inspired by neuroscience
  - lower energy consumption
- Reinforcement learning
- Adversarial learning with GANs
- Learning parameter distributions
- Evolution to generate DNAs

FAMOUS CNNs & DNNs

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201