Imperial College London

May 6th, 2021

Convolutional Neural Networks

Olivier Dubrule

1

Imperial College London

Objectives of the Day

- Understand what Convolutional Neural Networks (CNNs) are
- Be at ease with calculations associated with the CNN parameters
- Get familiar with classical CNN structures on well-known examples

2

Imperial College London

Convolutional Neural Networks

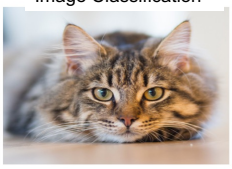
1. Convolutional Neural Networks
2. Pooling and Fully Connected Layers
3. Examples
4. Transfer Learning

3

Imperial College London

Computer Vision Topics

Image Classification



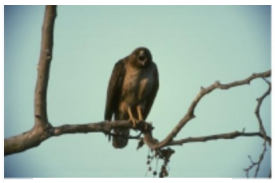
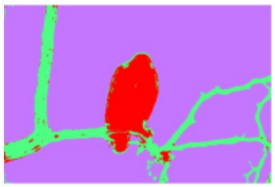
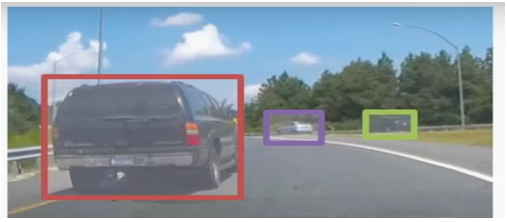


Image Segmentation





Object Localization (1 object) or Detection (several objects)

4

Imperial College London

MNIST

Extract from the MNIST dataset

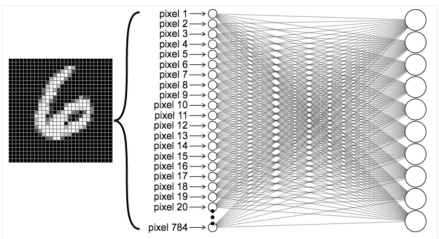
MNIST: Modified National Institute of Standards and Technology database

5

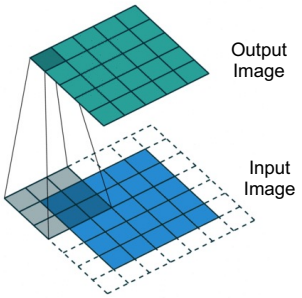
Imperial College London

What are Convolutional Neural Networks?

FeedForward NN Approach on MNIST:
Spatial Organization is Lost



A CNN will take into account the spatial organization of the dataset

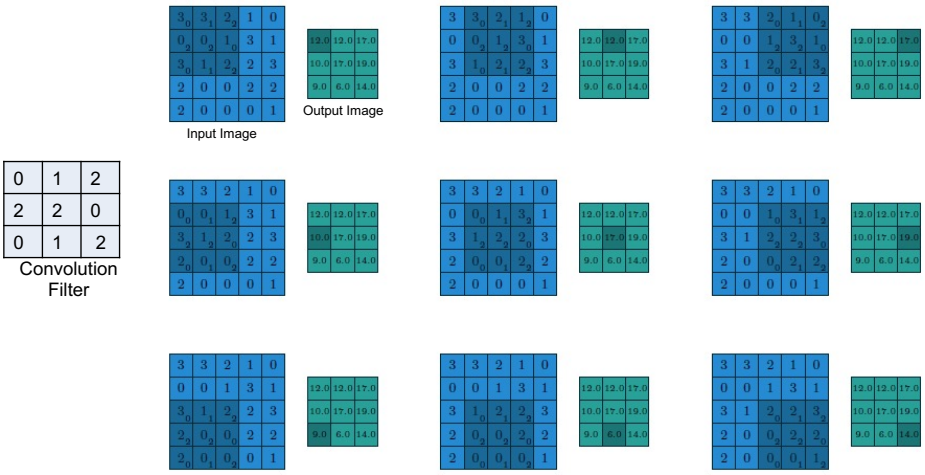


The grey 3x3 moving window is the convolutional filter

6

Imperial College London

What is the Convolution Operation?



From Dumoulin & Visin, 2018

7

Imperial College London

Exercise: Calculate the Output Image

Input Image

4	2	3	3
2	1	2	5
3	2	0	3
1	2	3	1

*

Convolution Filter (or Kernel)

2	-3	2
1	0	1
2	-3	3

=

Output Image

12	20
13	9

8

Imperial College London

Filters (or Kernels) for Edge Detection

Filter for Detecting Vertical Edges

Input Image

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

* *Convolution Operator*

Filter (or Kernel)

1	0	-1
1	0	-1
1	0	-1

=

Output Image

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

Inspired from Andrew Ng

9

Imperial College London

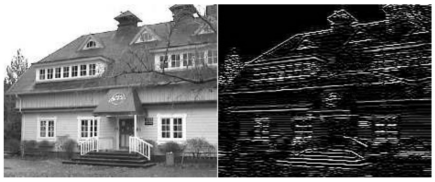
Which Filter should I use? (1)

Blur

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Horizontal Edges

-1	-1	-1
2	2	2
-1	-1	-1



https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

10

Imperial College London

Which Filter should I use? (2)

Edges

-1	-1	-1
-1	8	-1
-1	-1	-1

Sobel Filters

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

Horizontal

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

Vertical

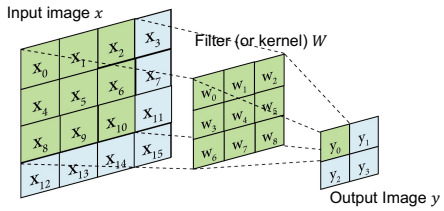
https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

11

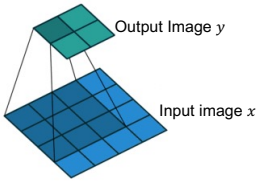
Imperial College London

Idea Behind Convolutional Neural Networks:

Parametrize the Filter and Optimize its Coefficients Based on the Objective!



$$W * x = y$$

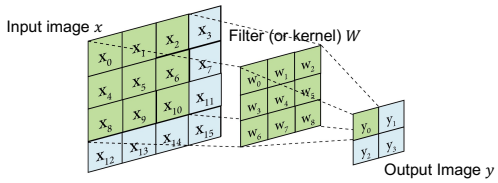


$$y_0 = w_0x_0+w_1x_1+w_2x_2+w_3x_4+w_4x_5+w_5x_6+w_6x_8+w_7x_9+w_8x_{10}$$

12

Imperial College London

Making Each Kernel Operation Non-Linear



$$y_0 = w_0x_0+w_1x_1+w_2x_2+w_3x_4+w_4x_5+w_5x_6+w_6x_8+w_7x_9+w_8x_{10}$$

A bias term can be added

$$y_0 = w_0x_0+w_1x_1+w_2x_2+w_3x_4+w_4x_5+w_5x_6+w_6x_8+w_7x_9+w_8x_{10} + \boldsymbol{b}$$

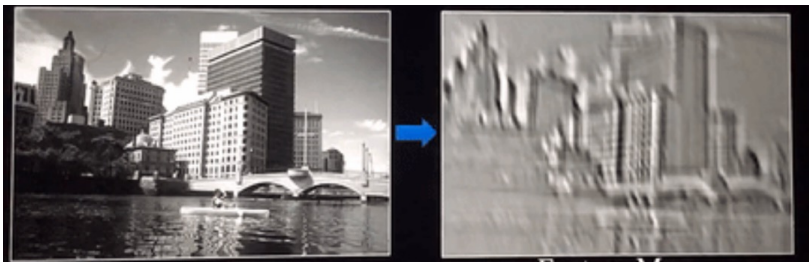
And a non-linear transformation is applied by an activation function g

$$y_0 = \boldsymbol{g}(w_0x_0+w_1x_1+w_2x_2+w_3x_4+w_4x_5+w_5x_6+w_6x_8+w_7x_9+w_8x_{10} + \boldsymbol{b})$$

13

Imperial College London

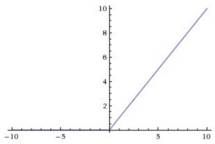
Linear Convolution Operation….



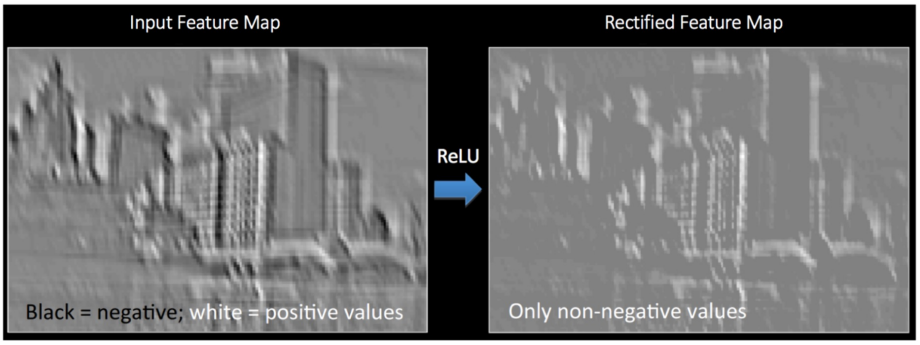
https://ujwlkarn.files.wordpress.com/2016/08/giphy.gif?w=480&zoom=2

14

Imperial College London



.....Followed by ReLU Activation

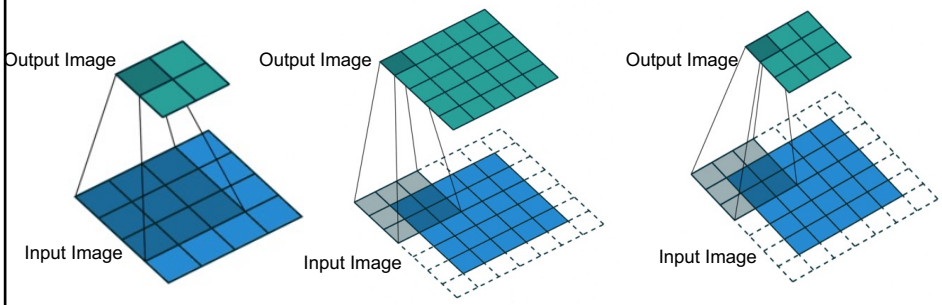


https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

15

Imperial College London

Padding and Stride



No padding, Stride 1 **Padding 1, Stride 1** **Padding 1, Stride 2**

16

Imperial College London

The Two-Dimensional Convolution Parameters

Parameters

Input Image Size: $n \times n$

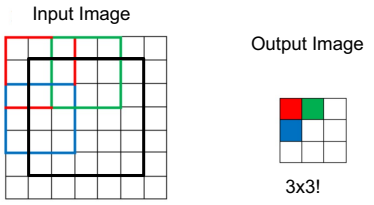
Filter (or Kernel) Size: f $\times f$

Padding: p

Stride : s

Output Image Size

$$\left(\frac{n+2p-f}{s}+1\right) \times \left(\frac{n+2p-f}{s}+1\right)$$



$n = 5\ p = 1\ f = 3\ s = 2$

17

Imperial College London

The Two-Dimensional Convolution Parameters

Parameters

Image Size: $n \times n$

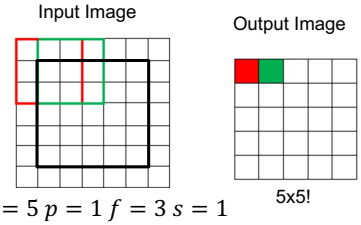
Filter (or Kernel) Size: f $\times f$

Padding: p

Stride : s

Output Image Size

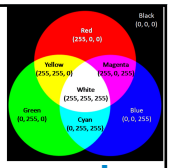
$$\left(\frac{n+2p-f}{s}+1\right) \times \left(\frac{n+2p-f}{s}+1\right)$$



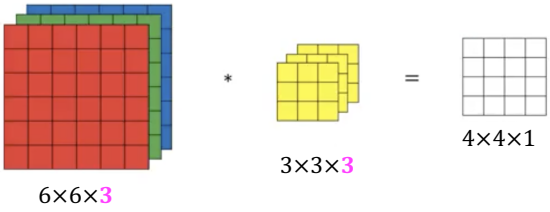
$n = 5\ p = 1\ f = 3\ s = 1$

18

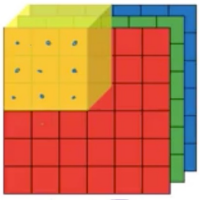
Imperial College London



If the Two-Dimensional image has 3 RGB channels…



27 operations each time (28 if bias term)



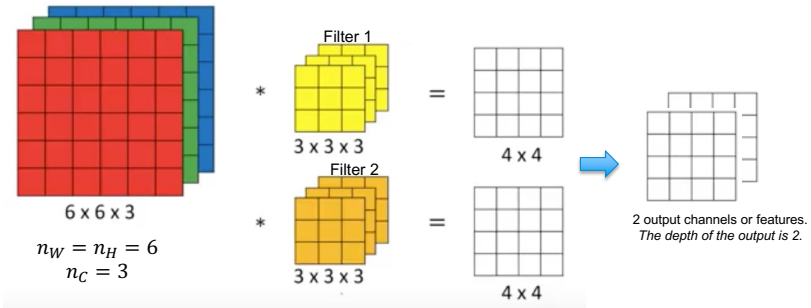
Each slice of the filter can detect different features in each of the 3 colour channels, for instance vertical edges in Red, horizontal edges in Green and vertical edges in Blue!

Andrew Ng https://www.youtube.com/watch?v=KTB_OFoAQcc&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&t=0s&index=7

19

Imperial College London

If Different Filters are Applied …



Exercise: *if my input Image is 50x50 with 10 channels, and if I apply 25 filters each of size 3x3x10 with a stride of 1, no padding, what is the size and number of channels of the output image?*

Answer: *size is 48x48. with 25 channels*

Andrew Ng https://www.youtube.com/watch?v=KTB_OFoAQcc&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&t=0s&index=7

20

Imperial College London

Number of Weights to Train for a Given Layer of the CNN

If I have 100 filters of size 4 x 4 x 3 (each with bias term) in layer l, what is the total number of weights (or degrees of freedom) to train for this layer?

$$\textbf{Number of filters} \times (\textbf{size of filter} + \textbf{1 bias term}) =$$

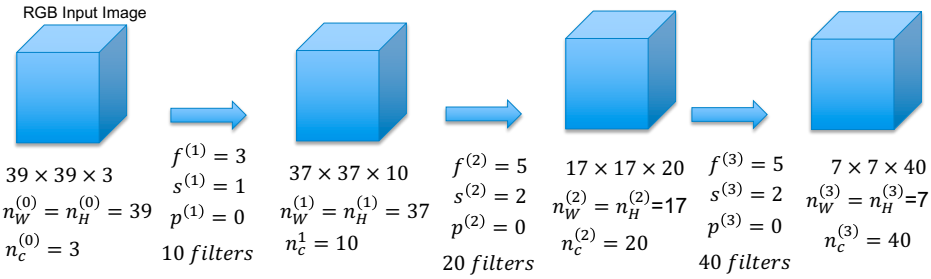
$$\mathbf{100} \times (\mathbf{4} \times \mathbf{4} \times \mathbf{3} + \mathbf{1}) = \ \mathbf{100} \times \mathbf{49} = \mathbf{4900}\ \textbf{weights}$$

Note that the number of parameters is independent of the size of the input image, which was not the case for feed-forward neural networks!

21

Imperial College London

Our first CNN Example: Recognize Whether Image of Cat or Dog



Reshape

$7 \times 7 \times 40 = 1960$

Full Convolution (ie single neuron)

Probability of being a certain class

How Many Weights (with bias terms)?

=10x28+20x251+40x501+1961=
280+5020+20040+1961=27301

Inspired from Andre Ng

22

Imperial College London

Counting the Degrees of Freedom (or Parameters) for a layer l

Assume layer l has $n_c^{(l)}$filters of size $f^{(l)}$, a padding $p^{(l)}$ and a stride $s^{(l)}$

Assume the input image has size $n_H^{(l-1)} \times n_W^{(l-1)} \times n_c^{(l-1)}$
and the output image has size $n_H^{(l)} \times n_W^{(l)} \times n_c^{(l)}$

The size of each filter is $f^{(l)} \times f^{(l)} \times n_c^{(l-1)}$and the depth or number of output features is the number of filters $n_c^{(l)}$

And we can generalize the formula $n_H^{(l)} = \frac{n_H^{(l-1)} + 2p^{(l)} - f^{(l)}}{s^{(l)}} + 1$ and $n_W^{(l)} = \frac{n_W^{(l-1)} + 2p^{(l)} - f^{(l)}}{s^{(l)}} + 1$

And the total number of degrees of freedom (or parameters) of layer l is:

$$\textbf{Number of filters} \times (\textbf{size of filter} + \textbf{1 bias term}) = n_c^{(l)} \times \left(f^{(l)} \times f^{(l)} \times n_c^{(l-1)} + 1\right)$$

Independent of the size of the input and output images!

23

Imperial College London

Convolutional Neural Networks

1. Convolutional Neural Networks
2. Pooling and Fully Connected Layers
3. Examples
4. Transfer Learning

24

Imperial College London

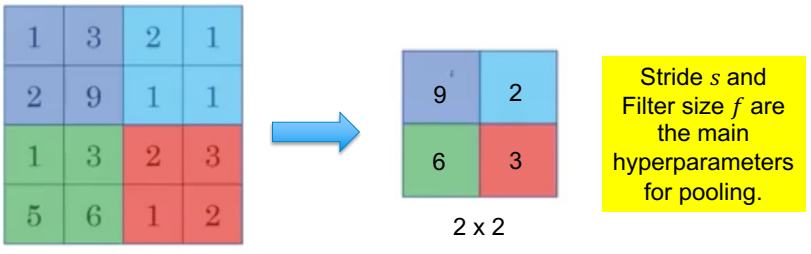
Three Types of Layers in a Convolutional Network

- Convolutional Layers (CONV)
- Fully Connected Layers (FC)
- Pooling Layers (POOL)

25

Imperial College London

Example of Max Pooling Layer (for Downsampling)



4 x 4

2 x 2

Stride s and Filter size f are the main hyperparameters for pooling.

Here $s = 2$ and $f = 2$ Quite often $f = s$

Pooling makes the input representations (feature dimension) smaller for the next layer, Max Pooling is often used because it may be interesting to keep the high values for the activation of the next layer as they may characterize some important features. Pooling reduces the number of parameters and computations in the network, therefore controlling overfitting.

26

Imperial College London

We already know Fully Connected Layers!

Transform for example 32x32x3 image into 10x1 vector.

1. Reshape image into 1D vector x of dimension 32x32x3= 3072
2. Apply full convolution through matrix W

x is 3072x1

Wx

W (or θ) is 10x3072 if no bias term

g non-linear activation function

$g(Wx)$

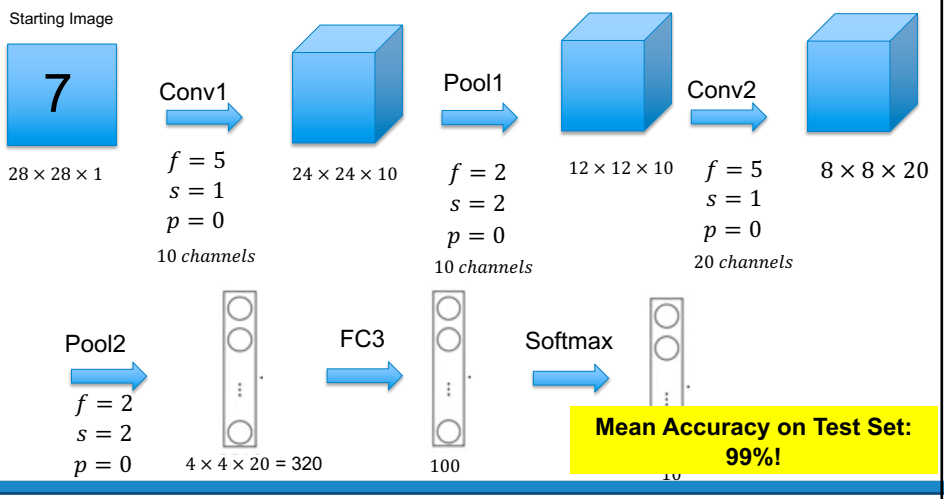
Wx is 10x1

Fully connected layers often have many weights as they connect every neuron in input layer to every neuron in output layer. Contrary to convolutional layers, their number of parameters depends on size of input an size of output image or vector.

27

Imperial College London

MNIST CNN Example (similar to LeNet-5 Architecture)



Mean Accuracy on Test Set: 99%!

28

http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf

28

Imperial College London

MNIST: CNN Layers (Bias Terms are used)

	Size of input image	Number of input channels	f	p	s	Size of output image	Number of output channels or filters	Number of output neurons	Size of Filter + 1	Number of Parameters
Conv1	28	1	5	0	1		10			
Pool1			2	0	2		10			
Conv2			5	0	1		20			
Pool2			2	0	2		20			
	Size of input							Number of output neurons		
FC1								100		
Softmax								10		
							Total Neurons		Total Parameters	

Exercise

Imperial College London

MNIST: CNN Layers (Bias Terms are used)

	Size of input image	Number of input channels	f	p	s	Size of output image	Number of output channels or filters	Number of output neurons	Size of Filter + 1	Number of Parameters
Conv1	28	1	5	0	1	24	10	5760	26	260
Pool1	24	10	2	0	2	12	10	1440		
Conv2	12	10	5	0	1	8	20	1280	251	5020
Pool2	8	20	2	0	2	4	20	320		
	Size of input							Number of output neurons		
FC1	320							100	321	32100
Softmax	100							10	101	1010
							Total Neurons	8910	Total Parameters	38390

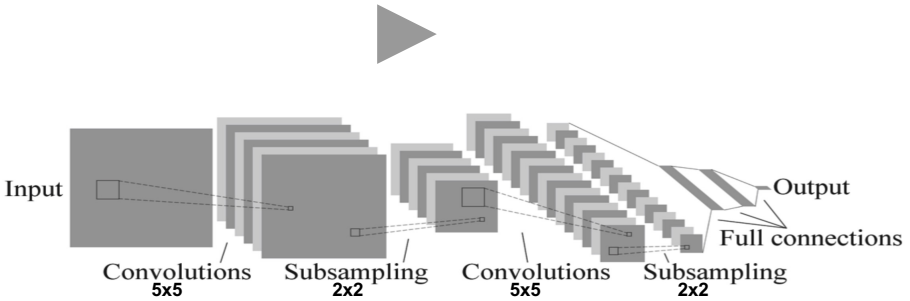
Exercise Correction

Imperial College London

MNIST: An Interactive CNN Example similar to LeNet5

From Paper:
An Interactive Node-Link Visualization of Convolutional Neural Networks, by A. Harley

http://scs.ryerson.ca/~aharley/vis/conv/flat.html



31

Imperial College London

Structure of Harley‘s Network

From Paper:
An Interactive Node-Link Visualization of Convolutional Neural Networks, by A. Harley

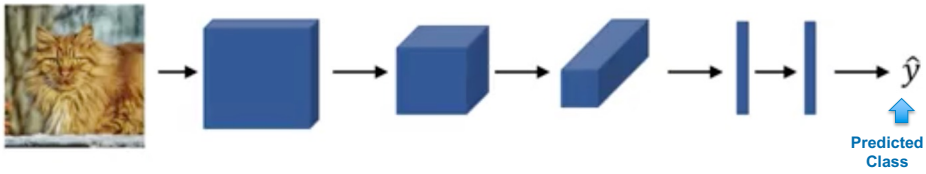
	Size of input image	Number of input channels	f	p	s	Size of output image	output channels or filters	Number of output neurons	Size of Filter + 1	Number of Parameters
Conv1	32	1	5	0	1	28	6	4704	26	156
MaxPool	28	6	2	0	2	14	6	1176		
Conv2	14	6	5	0	1	10	16	1600	151	2416
MaxPool	10	16	2	0	2	5	16	400		
	Size of input							Number of output neurons		
FC1	400							120	401	48120
FC2	120							100	121	12100
Softmax	100							10	101	1010
							Total Neurons	8110	Total Parameters	63802

32

Imperial College London

CNN Parameters are optimized the usual way

Example of cat image identification. Training set $(x^{(1)}, y^{(1)}) \ldots (x^{(m)}, y^{(m)})$



Loss $Function$ $\mathrm{J} = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}_i, y_i)$ (Typically $\mathcal{L}(\hat{y}_i, y_i)$ is cross-entropy for image i)

The value of m depends whether batch, minibatch or stochastic gradient descent is used

33

33

Imperial College London

Convolutional Neural Networks

1. Convolutional Neural Networks
2. Pooling and Fully Connected Layers
3. Examples
4. Transfer Learning

34

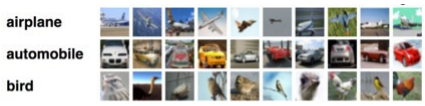
Imperial College London

Some of the Best-Known Datasets

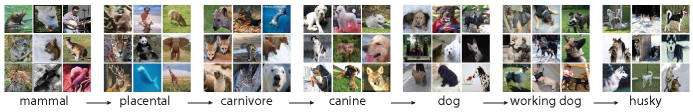
Very deep neural networks work best when trained on very large datasets!

- **MNIST: Handwritten digits, 60k Training Images, 10k Test Images (see also Fashion MNIST)**
- **CIFAR-10 / CIFAR-100: 50k Training Images, 10k Test Images of 10 (CIFAR-10) or 100 (CIFAR-100) classes**

  Color Images are 32x32, Task: Classification https://www.cs.toronto.edu/~kriz/cifar.html



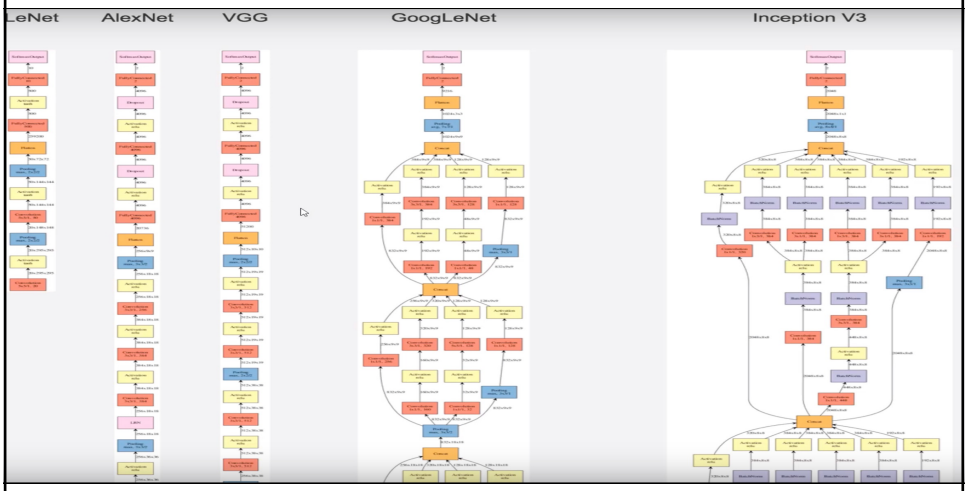
- **Imagenet: > 15 Million Images in 20,000 classes! https://en.wikipedia.org/wiki/ImageNet**



35

Imperial College London

Some of the best-known CNNs

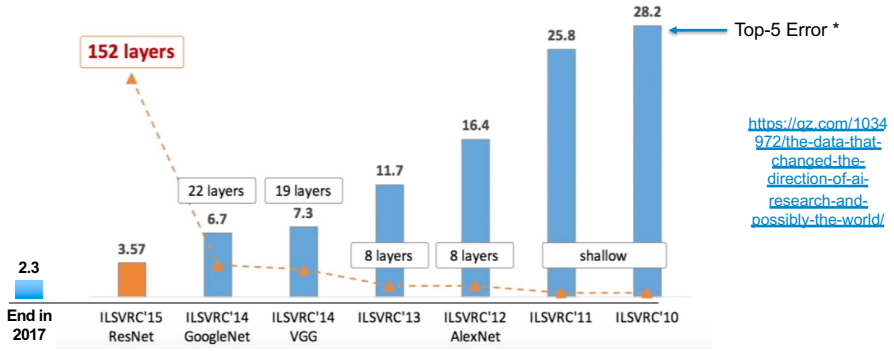


36

Imperial College London

The Image Large-Scale Visual Recognition Challenge

From 2010 to 2017, ImageNet ran an annual competition (ILSVRC: Image Large Scale Visual Recognition Challenge) in visual recognition where participants are presented with the challenge of classifying 1.2 million images belonging to 1000 different classes from Imagenet dataset.



https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/

* The ***Top-5 error*** *is the percentage of the times that the classifier did not include the correct class among its* ***top 5*** *guesses*

https://www.youtube.com/watch?v=DAOcjicFr1Y

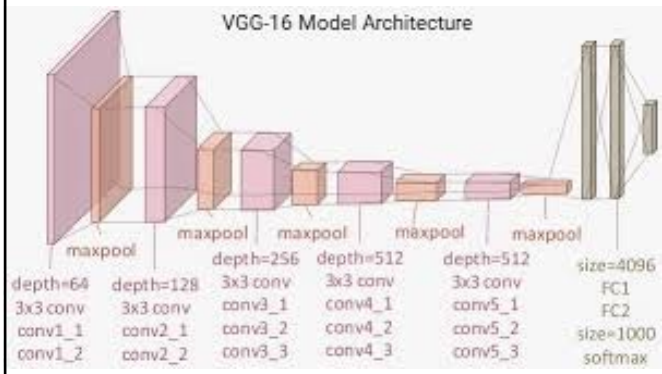
Imperial College London

Some of the best-known CNNs

- **LeNet-5** (Le Cun et al, 1998) for Digit Recognition: Architecture is CONV-POOL-CONV-POOL-FC-FC
- **AlexNet** (ILSRVC 2012): Very similar to Le-Net-5, with just more layers. About 60 MM parameters for 8 layers. First CNN-based winner of ILSRVC!
  Other characteristics:
  First use of ReLU, heavy data augmentation, dropout, SGD Momentum 0.9, Learning Rate 0.01, reduced by 10 after a number of iterations, L2 weight decay 0.00004, Ensemble averaging of 7 CNNs.

Imperial College London

Some of the best-known CNNs: VGG

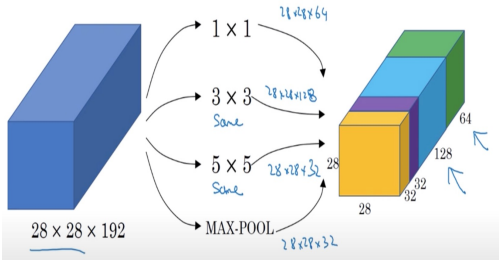


VGG *(co-winner of ILSRVC 2014): More layers (16 to 19) but small filters for convolution (3x3, Pad 1, Stride 1, in stacked layers) and pooling (2x2, stride 2) . More non-linearities , with less parameters in CONV layers. VGG16 has 138 MM parameters, mainly due to last Fully-Connected layers. VGG19 only slightly better than VGG16. Ensembling also used.*

Imperial College London

Some of the best-known CNNs: Inception Module

- **GoogLeNet/Inception** (co-winner of ILSRVC 2014): More layers (22), based on "Inception" module ("network within a network"), no Fully-Connected layer, only 5 Million parameters! General Philosophy based on basic Inception Module ("Network within a Network")

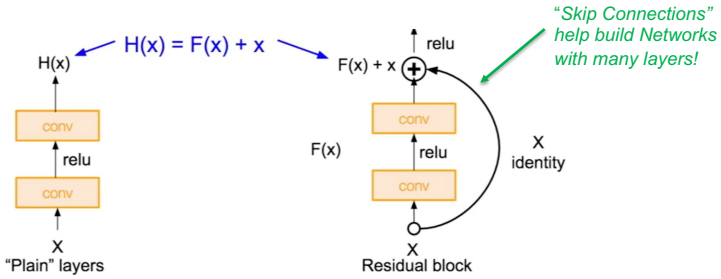


Example from Andrew Ng's C4W2L06

Imperial College London

Some of the best-known CNNs: ResNet

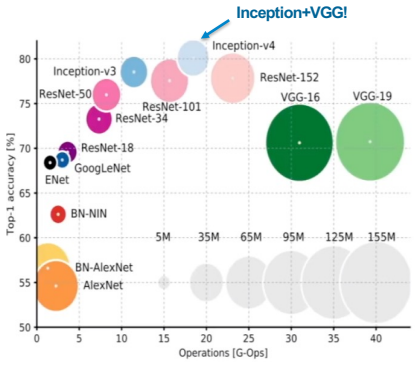
- **Resnet (winner of ILSRVC 2015):** Very deep network (152 layers) using residual connections. No FC layers. Did better than human performance!



41

Imperial College London

Comparing Accuracy, Speed and Memory Usage



- **Download open model implementations pre-trained on large datasets:**

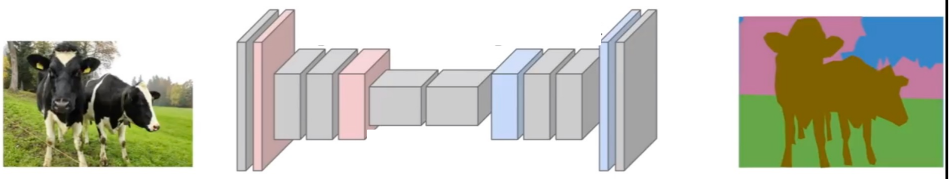
  Pytorch: Torchvision https://pytorch.org/docs/stable/torchvision/models.htm

https://www.youtube.com/watch?v=DAOcjicFr1Y

42

Imperial College London

Semantic Segmentation with Down- and Up-Sampling



Input:
3 x H x W

Predictions:
H x W

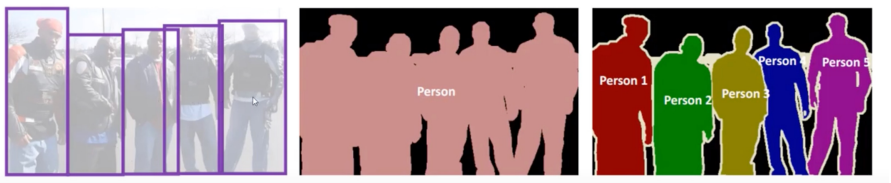
Semantic Segmentation associates a class label to each pixel of the image (rather than a single class label for the whole image, as for standard supervised classification)

https://www.youtube.com/watch?v=azM57JuQpQI

43

Imperial College London

Segmentation Model: U-Net (Ronneberger et al, 2015)



Object Detection | Semantic Segmentation | **Instance Segmentation**

https://www.youtube.com/watch?v=azM57JuQpQI

44

Imperial College London

U-Net presented by Ronneberger himself

U-Net: Convolutional Networks for Biomedical Image Segmentation

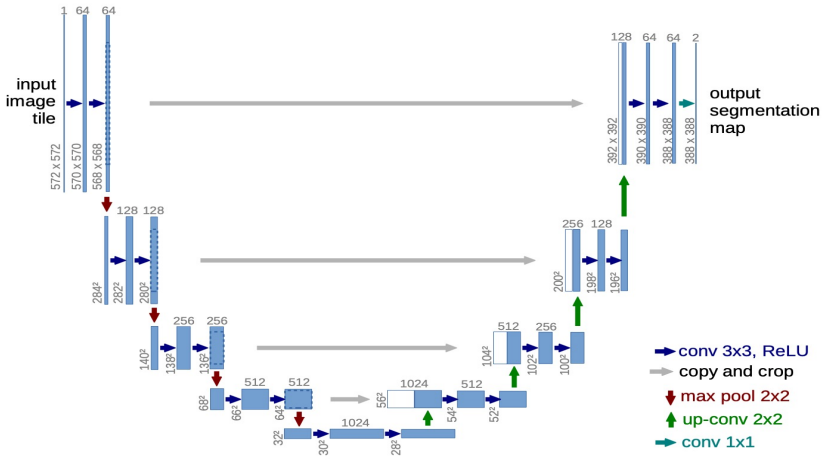
Olaf Ronneberger, Philipp Fischer, and Thomas Brox

Computer Science Department and BIOSS Centre for Biological Signalling Studies,
University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de,
WWW home page: http://lmb.informatik.uni-freiburg.de/

https://www.youtube.com/watch?v=81AvQQnpG4Q

Imperial College London

Segmentation Model: U-Net (Ronneberger et al, 2015)



Imperial College London

U-Net Loss Function for one Image (Ronneberger et al, 2015)

If we wish to segment the input image into K classes, we have K output activation maps $a_k(x)$, where x is the pixel and k is the class

Each map $a_k(x)$ is transformed into a probability $p_k(x)$, by Softmax:

$$p_k(x) = \frac{e^{a_k(x)}}{\sum_{k'=1}^{K} e^{a_{k'}(x)}}$$

If $l(x)$ is the actual class of the pixel x in the training image, the loss function is the cross-entropy:

$$J = \sum_{x \in Image} w(x) \log p_l(x)$$

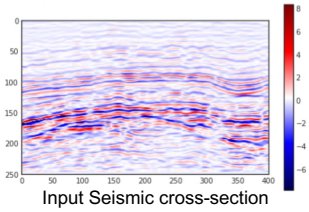
where $w(x)$ is a weighing function to give more weights to some pixels.

https://www.youtube.com/watch?v=azM57JuQpQI

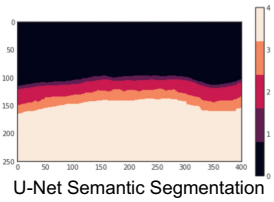
47

Imperial College London

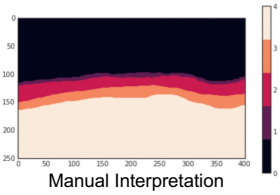
Applying U-Net to Seismic Interpretation



Input Seismic cross-section



U-Net Semantic Segmentation



Manual Interpretation

48

Imperial College London

Convolutional Neural Networks

1. Convolutional Neural Networks
2. Pooling and Fully Connected Layers
3. Examples
4. Transfer Learning

49

Imperial College London

Transfer Learning (1)

The most well-known CNN designs are available on-line and have been successfully trained on very large number of images (1,000,000s).

In many applications we often work from a relatively small number of images.

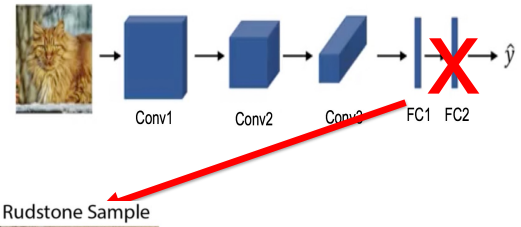
Why not start from an existing trained CNN sharing an objective of a similar nature (such as Classification) and tailor it to our application?

50

Imperial College London

Example of Transfer Learning Approach

Take existing trained network such as Inception v3 model, trained on ImageNet dataset for differentiating between 1,000 different classes of images.



Rudstone Sample

Re-train last layer of the network on images of interest, such as pictures of carbonate cores .

From Sharinia Kanagandran and Cedric John, Imperial College

51

Imperial College London

Conclusion

- Convolutional Neural Networks mostly used on 2-D images or 3-D Volumes for image classification, image segmentation, object localization/detection.
- Convolutional Neural networks are the basis for many successful Deep Learning Applications.
- AlexNet, VGG, GoogleNet/Inception and Resnet winners of ILSVRCs
- Transfer Learning: many complex but successful CNN architectures can be transferred from one application to another, by re-training the last layer.

Look at https://poloclub.github.io/cnn-explainer/

52