Jaime S. Cardoso
jaime.cardoso@inesctec.pt

INESC TEC and **Faculdade de Engenharia, Universidade do Porto**, Portugal

Introduction to Computer Vision and Machine Learning

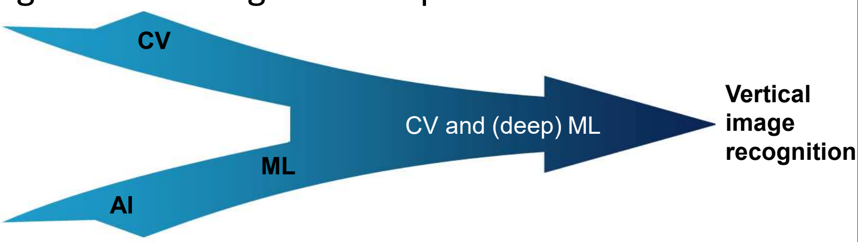
(and Deep Learning?!)

VISUM
July 06th, 2020, Porto, Portugal

1

Roadmap

- **A brief history of Computer Vision**
  - Convergence of Machine Learning and Signal Processing and Computer Vision



- **The main components in ML**
- **Deep learning and Vertical Image Recognition**

2

2

Face recognition

3

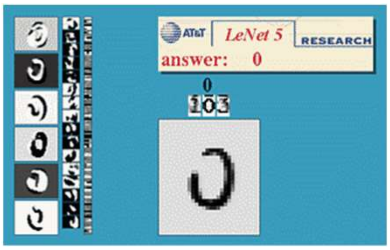
Autonomous cars

4

Optical character recognition (OCR)

Technology to convert scanned docs to text

- If you have a scanner, it probably came with OCR software



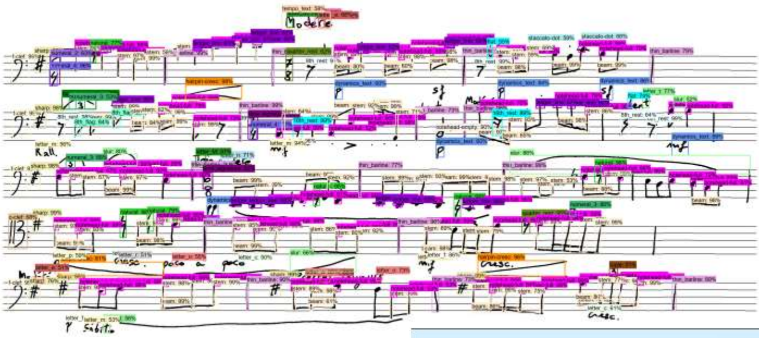
Digit recognition, AT&T labs
http://www.research.att.com/~yann/

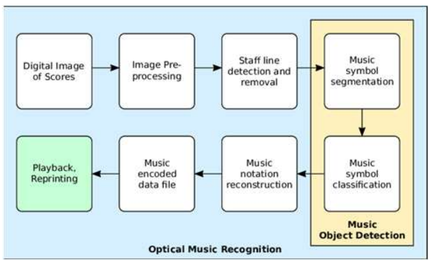
License plate readers
http://en.wikipedia.org/wiki/Automatic_number_plate_recognition

James Hays

5

Optical Music Recognition

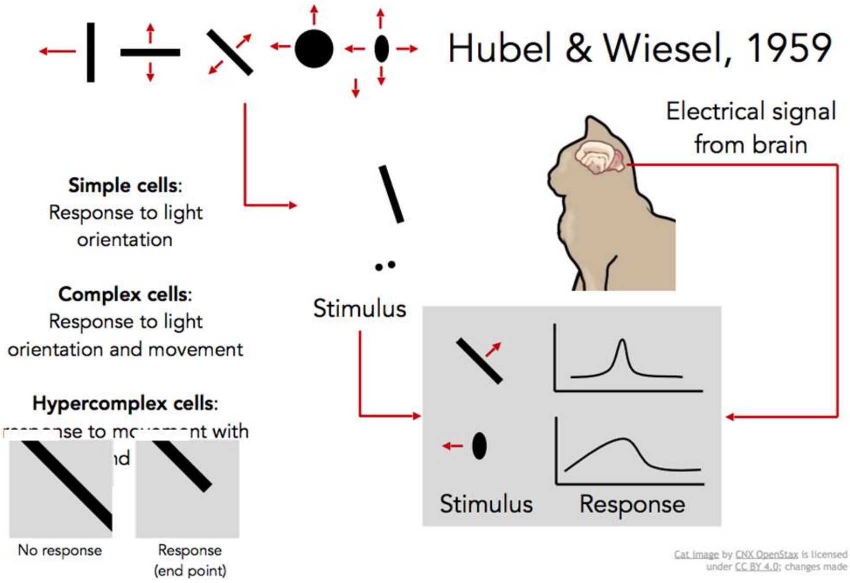




6

6

The early days of computer vision



Fei-Fei Li et al.

7

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

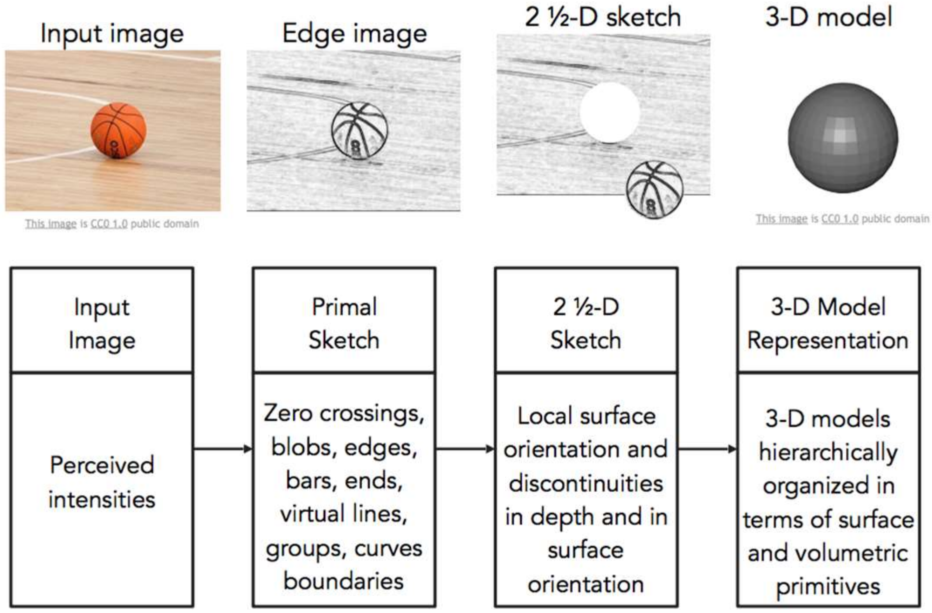
THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Fei-Fei Li et al.

8



Stages of Visual Representation, David Marr, 1970s

Fei-Fei Li et al.

9

Artificial Intelligence (AI)

- " [...automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning..." (Bellman, 1978)
- " The branch of computer science that is concerned with the automation of intelligent behaviour." (Luger and Stubblefield, 1993)
- "The ultimate goal of AI is to create technology that allows computational machines to function in a highly intelligent manner. (Li Deng 2018)

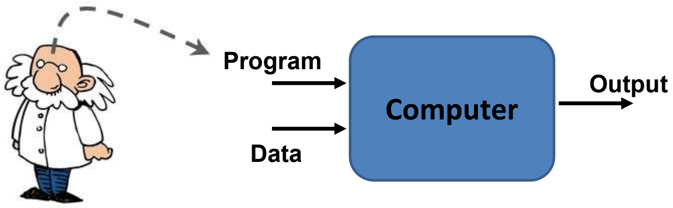
10

10

AI: three generations

1st wave of AI: **the sixties**

- emulates the decision-making process of a human expert



11

11

AI: three generations

1st wave of AI: **the sixties**

- Based on expert knowledge
  - "if-then-else"
- Effective in narrow-domain problems
- Focus on the head or most important parameters (identified in advance), leaving the "tail" parameters and cases untouched.

- Transparent and interpretable
- Difficulty in generalizing to new situations and domains
- Cannot handle uncertainty
- Lack the ability to learn algorithmically from data

12

12

History of ideas in CV (recognition)

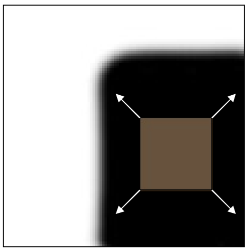
- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features

Svetlana Lazebnik

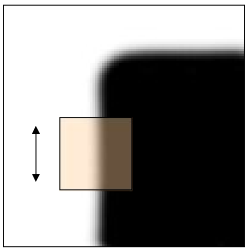
13

Corners

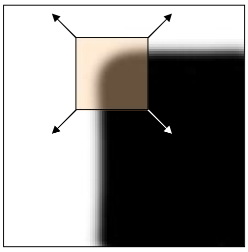
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"**flat**" region: no change in all directions



"**edge**": no change along the edge direction



"**corner**": significant change in all directions

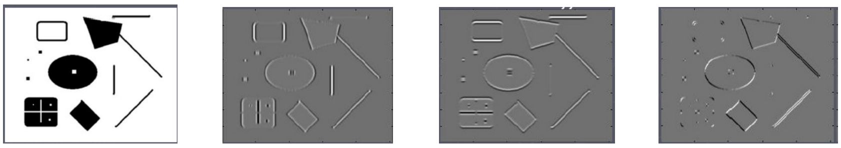
Alyosha Efros

14

Harris Detector Formulation

$$M = \sum w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

2 x 2 matrix of image derivatives (averaged in neighborhood of a point).

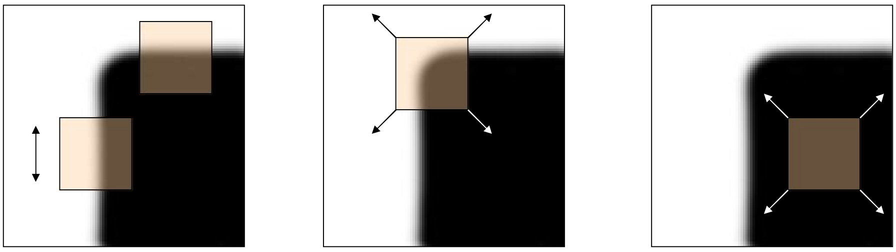


Notation: $I_x \Leftrightarrow \frac{\partial I}{\partial x}$ $\quad I_y \Leftrightarrow \frac{\partial I}{\partial y}$ $\quad I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$

James Hayes

15

Corner response function



"edge":
$\lambda_1 >> \lambda_2$
$\lambda_2 >> \lambda_1$

"corner":
λ_1 and λ_2 are large,
$\lambda_1 \sim \lambda_2$;

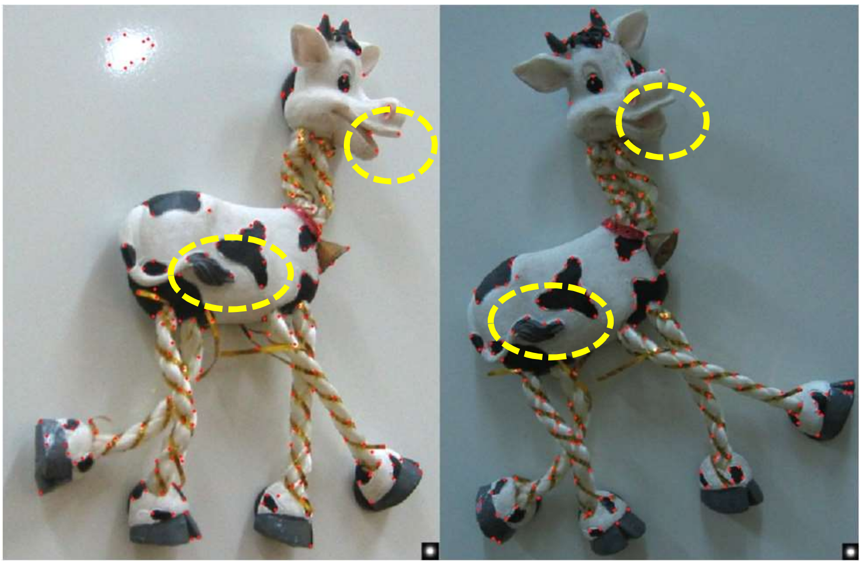
"flat" region
λ_1 and λ_2 are small;

$$C = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: constant (0.04 to 0.06)

16

Harris corner detector



Darya Frolova, Denis Simakov

17

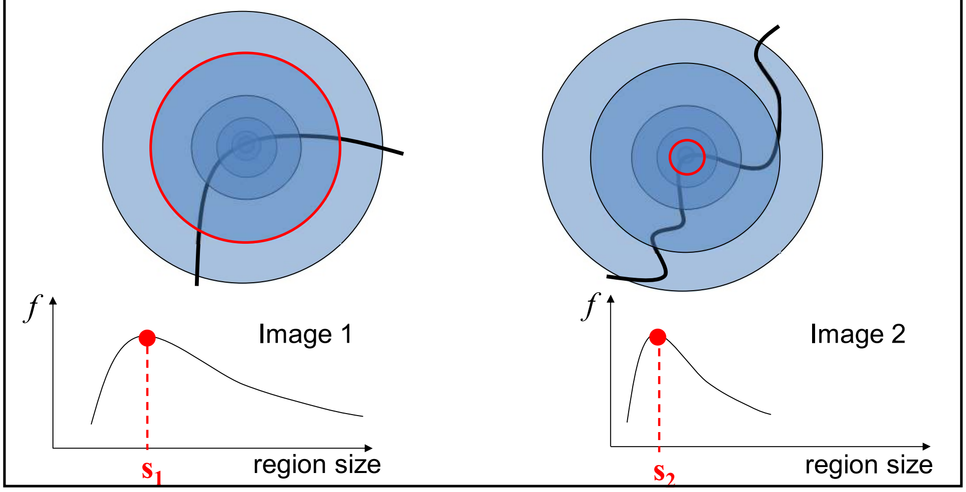
Scale invariant detection

- Consider regions (e.g. circles) of different sizes around a point
- Find regions of corresponding sizes that will look the same in both images?

18

Scale invariant detection

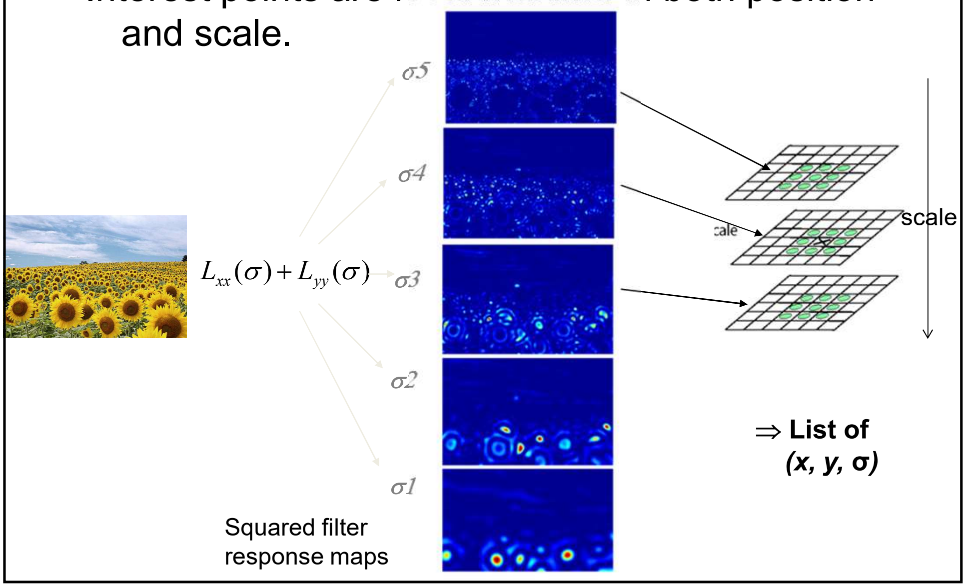
Intuition - Find scale that gives local maxima of some signature function *f* in both position and scale.



19

Scale invariant interest points

Interest points are local maxima in both position and scale.

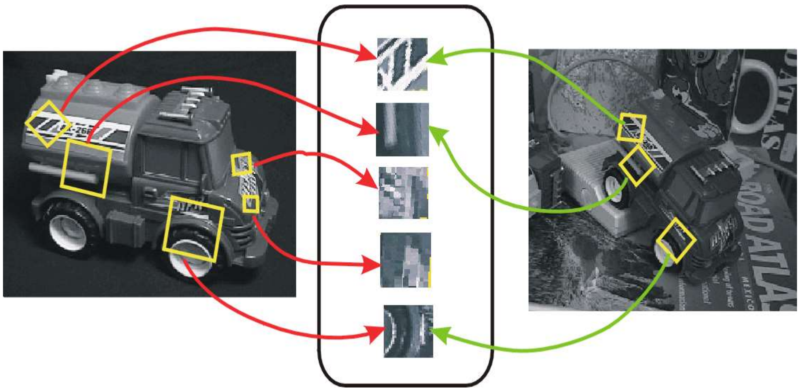


20

Invariant local features

Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …



Feature Descriptors

21

Local descriptors

- In each detected feature (point), a descriptor is then extracted
- Histogram-based descriptors
  - Based on the histogram of oriented gradient
  - SIFT, SURF, GLOH and HOG
- Compact descriptors
  - Based on binary strings obtained comparing pairs of image intensities
  - BRIEF, ORB, BRISK and FREAK

22

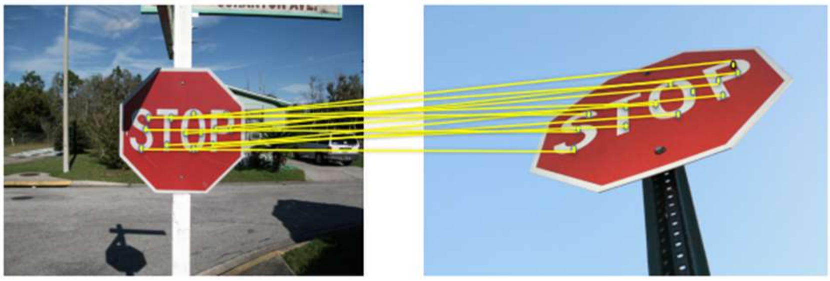


Image is public domain

Image is CC BY-SA 2.0

"SIFT" & Object Recognition, David Lowe, 1999

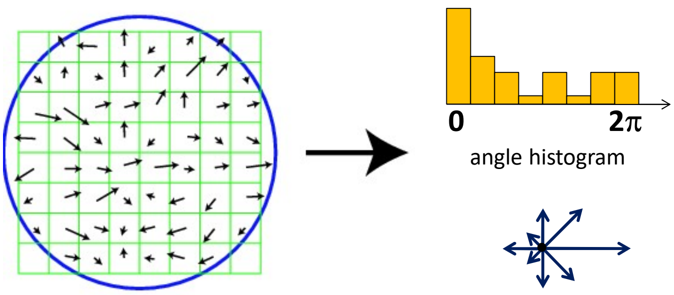
Fei-Fei Li et al.

23

SIFT descriptor

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations



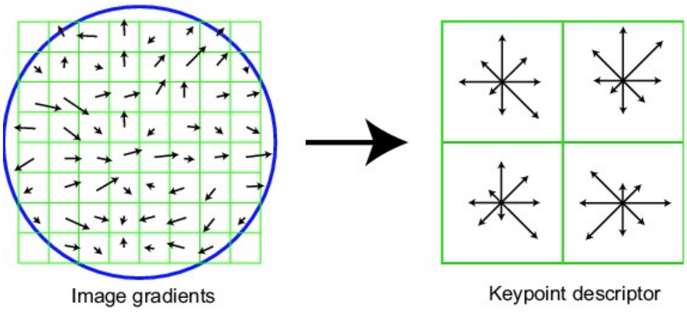
Distinctive image features from scale-invariant keypoints. David G. Lowe. *IJCV* 60 (2), pp. 91-110, 2004.

24

SIFT descriptor

Full version

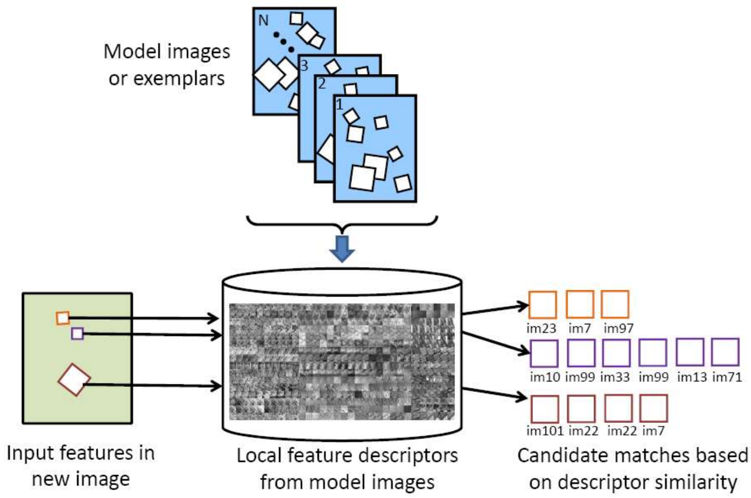
- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



25

Large-scale image search

Combining local features, indexing, and spatial constraints

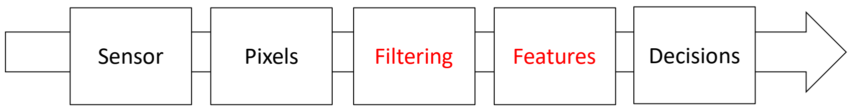


K. Grauman and B. Leibe

26

History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features



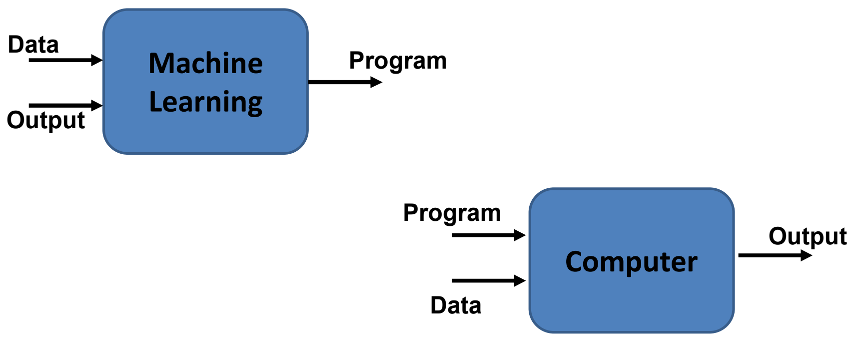
Svetlana Lazebnik

27

AI: three generations

2^{nd} wave of AI: **the eighties**

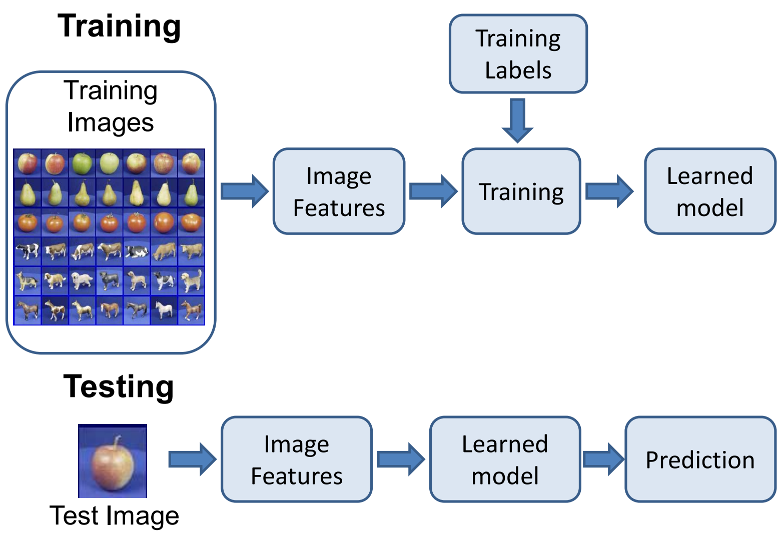
- Based on (shallow) machine learning



28

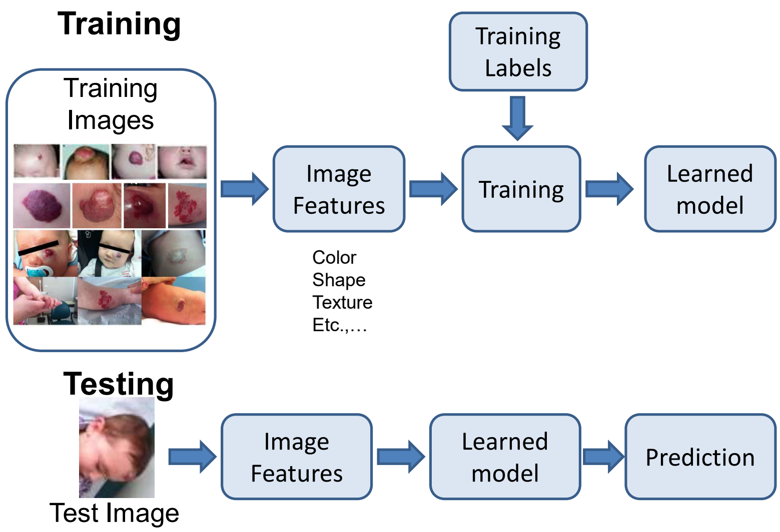
28

Classification in computer vision



29

Classification in computer vision



Features are designed by humans, requiring significant expertise

30

30

An example: decision process

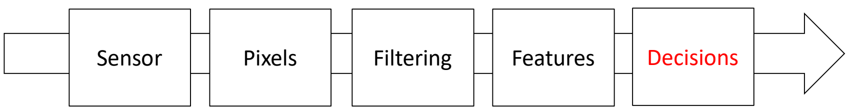
- What kind of information can distinguish one class from the other?
  - Length, width, weight, texture, etc.
- What can cause problems during sensing?
  - Lighting conditions, position, camera noise, etc.
- What are the steps in the process?
  - Capture image -> isolate object of interest -> take measurements -> make decision

31

31

An example: our system

- **Sensor**
  - The camera captures a 2D image
- **Preprocessing**
  - Adjustments for average intensity levels
  - Segmentation to separate object from background
- **Feature Extraction**
  - Assume a specialist told us that length and color help on the classification task.



32

32

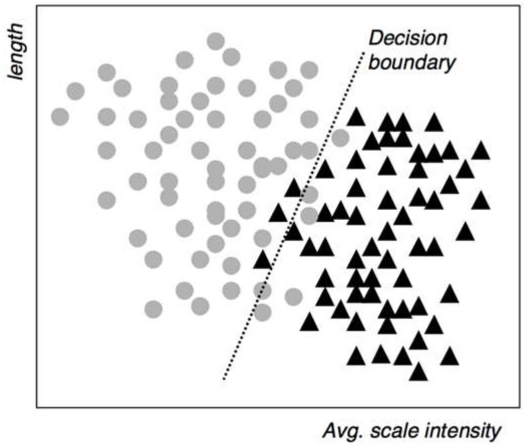
An example: multiple features

- We can use two features in our decision:
  - lightness: x_1
  - length: x_2
- Each lesion image is now represented as a point (feature vector)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

in a two-dimensional **feature space**.

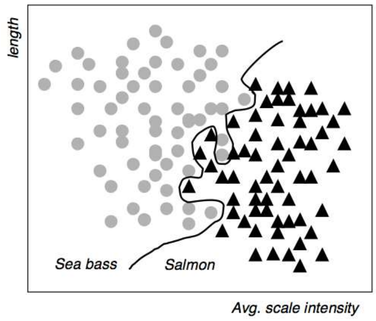
An example: multiple features



Scatter plot of lightness and length features for training samples. We can compute a **decision boundary** to divide the feature space into two regions with a classification rate of 95.7%.

An example: generalization

- **The issue of generalization**
  - The recognition rate of our linear classifier (95.7%) met the design specifications, but we still think we can improve the performance of the system
  - We then design a classifier that obtains an impressive classification rate of 99.9975% with the following decision boundary

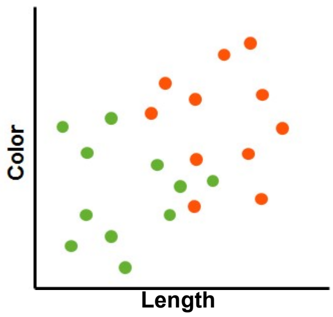


35

35

Data Driven Design

- When to use?
  - Difficult to reason about a generic rule that solves the problem
  - Easy to collect examples (with the solution)



36

36

Data Driven Design

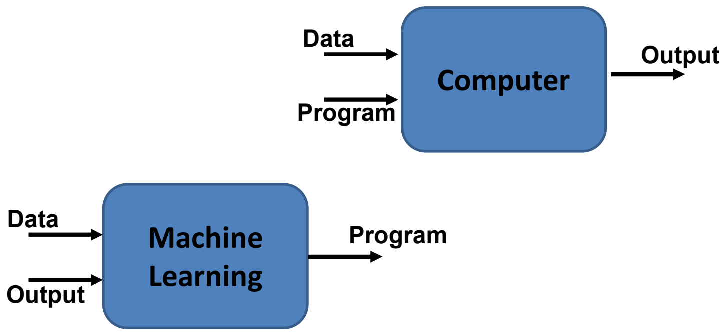
- There is little or no domain theory
- Thus the system will learn (i.e., generalize) from training data the general input-output function
  - Programming computers to use example data or past experience
- The system produces a program that implements a function that assigns the decision to any observation (and not just the input-output patterns of the training data)

37

37

What is Machine Learning?

- Automating the Automation



38

38

Data Driven Design

- A good learning program learns something about the data beyond the specific cases that have been presented to it
  - Indeed, it is trivial to just store and retrieve the cases that have been seen in the past
    - This does not address the problem of how to handle new cases, however
- Over-fitting a model to the data means that instead of general properties of the population we learn idiosyncracies (i.e., non-representative properties) of the sample.

39

39

DISTINCT LEARNING PROBLEMS

40

40

Taxonomy of the Learning Settings

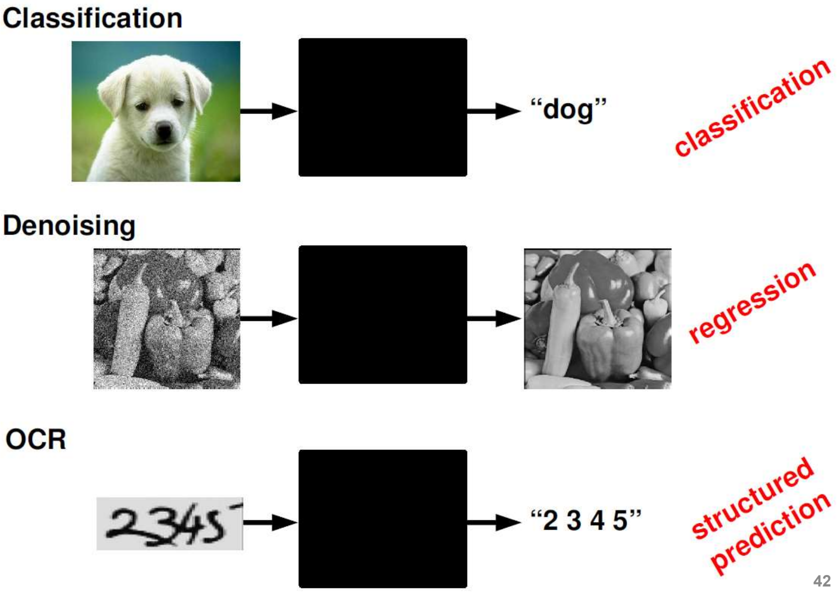
Goals and available data dictate the type of learning problem

- Supervised Learning
  - Classification
    - Binary
    - Multiclass
      - Nominal
      - Ordinal
  - Regression
  - Ranking
  - Counting
- Semi-supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- etc.

41

41

Supervised Learning: Examples



42

42

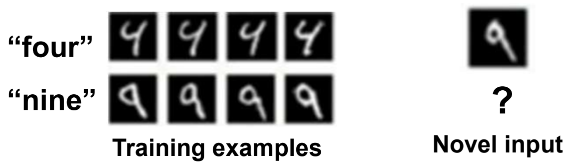
Regression

- Predicting house price
  - Output: price (a scalar)
  - Inputs: size, orientation, localization, distance to key services, etc.

- Given a collection of labelled examples (= houses with known price), come up with a function that will predict the price of new examples (houses).

Classification

- Given a collection of *labelled* examples, come up with a function that will predict the labels of new examples.



- How good is some function we come up with to do the classification?
- Depends on
  - Mistakes made
  - Cost associated with the mistakes

Classification/Regression

$$y = f(\mathbf{x})$$

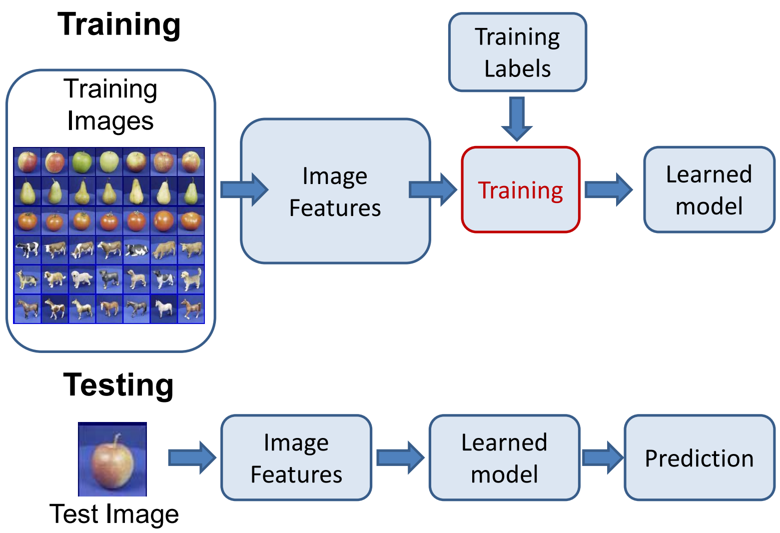
output   prediction function   feature vector

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* **x** and output the predicted value $y = f(\mathbf{x})$

45

45

Supervised Learning in computer vision



46

46

... but with common traits

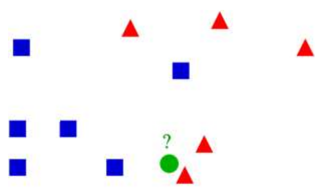
FOR THE SAME PROBLEM, DIFFERENT SOLUTIONS

47

47

k-Nearest neighbour classifier

- For a new point, find the k closest points from training data
- Labels of the k points "vote" to classify



k = 1

If the query lands here, the 1NN consist of 1 positive, so we classify it as positive.

Blue = negative
Red = positive

48

48

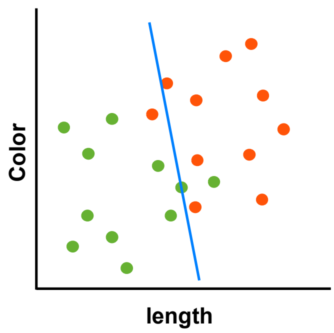
kNN as a classifier

- **Advantages**:
  - Simple to implement
  - Flexible to feature / distance choices
  - Naturally handles multi-class cases
  - Can do well in practice with enough representative data
- **Disadvantages:**
  - Large search problem to find nearest neighbors → Highly susceptible to the **curse of dimensionality**
  - Storage of data
  - Must have a meaningful distance function

49

49

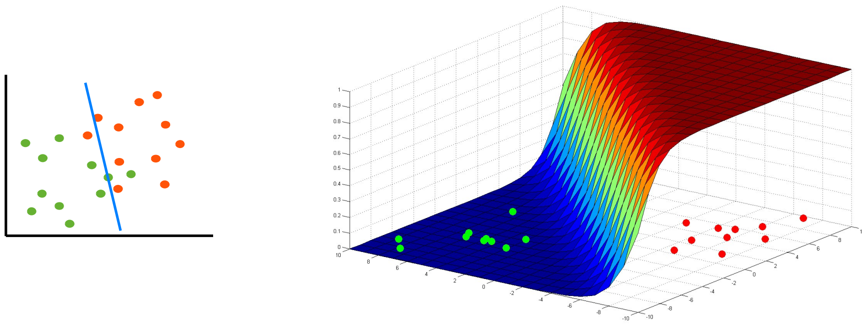
Design of a Classifier



50

50

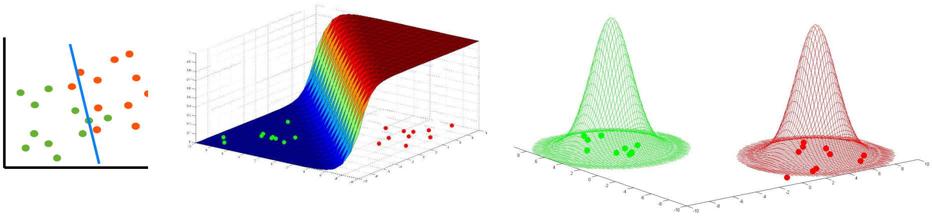
Design of a Classifier



51

51

Design of a Classifier



52

52

DIFFERENT SOLUTIONS BUT WITH COMMON INGREDIENTS

53

53

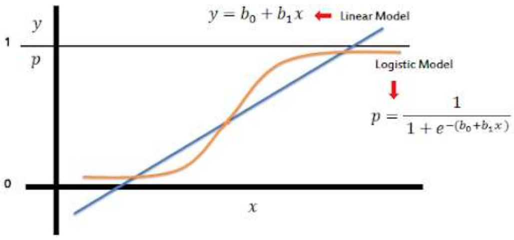
Common steps

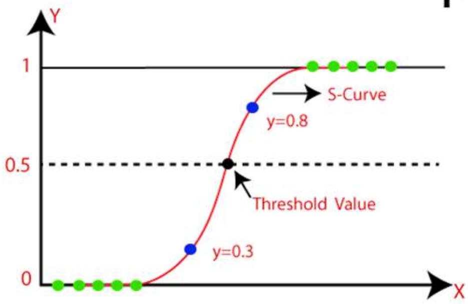
- The learning of a model from the data entails:
  - **Model representation**
  - **Evaluation**
  - **Optimization**

54

54

Logistic Regression



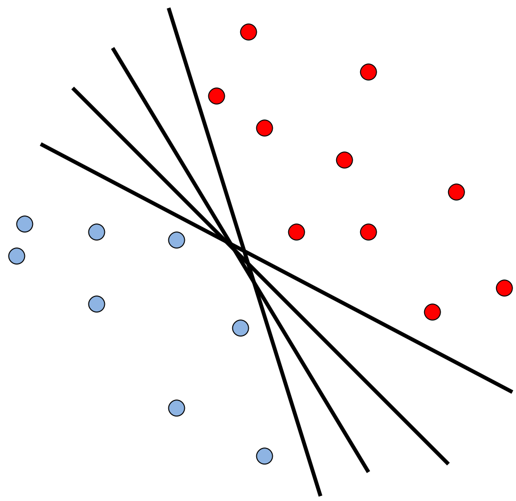


55

55

Linear classifiers

Find linear function to separate positive and negative examples



$\mathbf{x}_i$ positive : $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

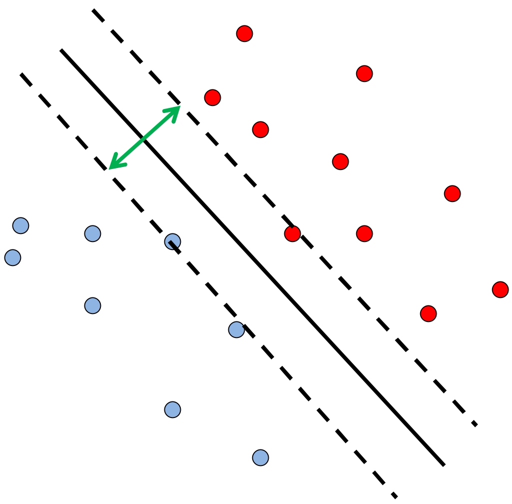
$\mathbf{x}_i$ negative : $\quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$

Which line
is best?

56

56

Support Vector Machines



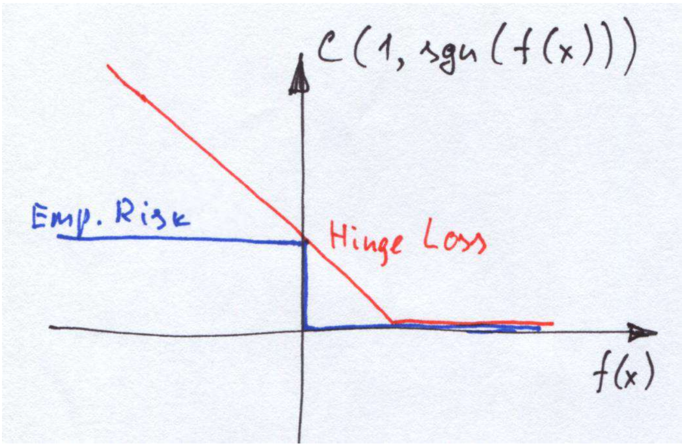
Classifier based on *optimal separating line (for 2D case)*

Maximize the ***margin*** between the positive and negative training examples

57

57

- Hinge Loss

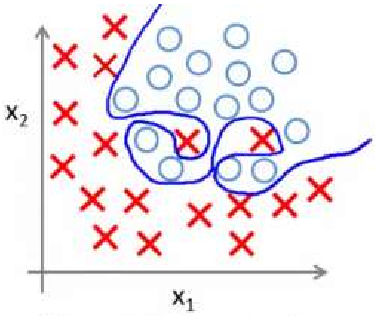


58

58

General Setting

- Hyper parameters / user defined parameters



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots$$

59

59

Hyper parameters / user defined parameters

AVOIDING OVERFITTING AND DATA MEMORIZATION

60

60

Regularization

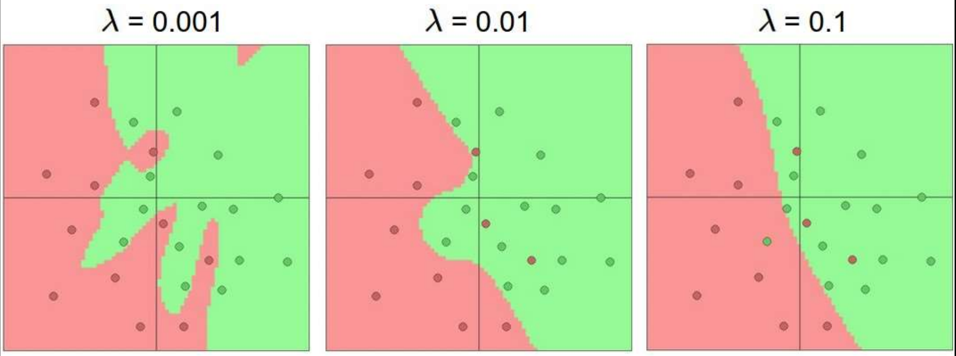
- To build a machine learning algorithm we specify model family, a cost function and optimization procedure
- Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error
  - There are many regularization strategies
- Regularization works by trading increased bias for reduced variance. An effective regularizer is one that makes a profitable trade, reducing variance significantly while not overly increasing the bias.

61

61

Regularization

- Evaluation
  - Minimize (error in data) + λ (model complexity)

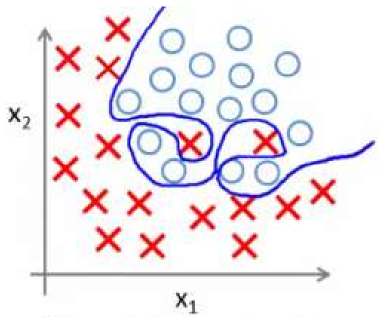


62

62

Regularized classifier

- Hyper parameters / user defined parameters



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots$$

63

63

Parameter Norm Penalties

- Penalize complexity in the loss function
  - Model complexity
  - Weight Decay

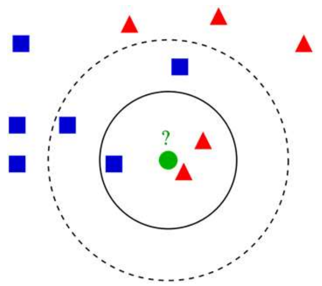
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2,$$

64

64

k-Nearest neighbour classifier

- For a new point, find the k closest points from training data
- Labels of the k points "vote" to classify



k = 5

If the query lands here, the 5 NN consist of 3 negatives and 2 positives, so we classify it as negative.

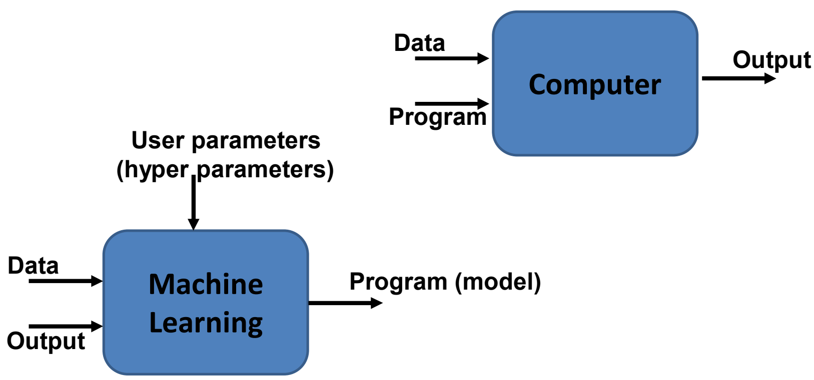
Blue = negative
Red = positive

65

65

What is Machine Learning?

- Automating the Automation



66

66

THERE ARE SO MANY OPTION TO DESIGN A CLASSIFIER…

A FAIR JUDGEMENT OF YOUR ALGORITHM

67

67

Model assessment, selection

- How to Compare Models?
- How can we select the right complexity model ?

68

68

Training - general strategy

- We try to simulate the real-world scenario.
- Test data is our future data. It should not be used in any design option of the classifier.
- Validation set can be our test set - we use it to select our model.
- The whole aim is to estimate the models' true error on the sample data we have.

69

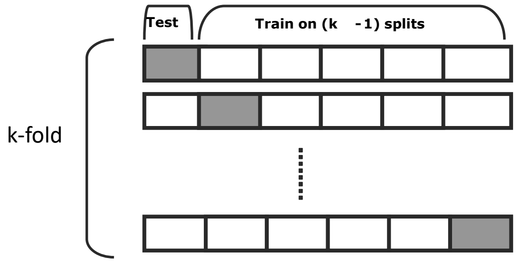
Hold out / test set method

- It is simple, however
  - We waste some portion of the data
  - If we do not have much data, we may be lucky or unlucky with our test data

- With **cross-validation** we reuse the data

70

K-fold cross validation



In 3 fold cross validation, there are 3 runs.
In 5 fold cross validation, there are 5 runs.
In 10 fold cross validation, there are 10 runs.

the error is averaged over all runs

71

71

Evaluation

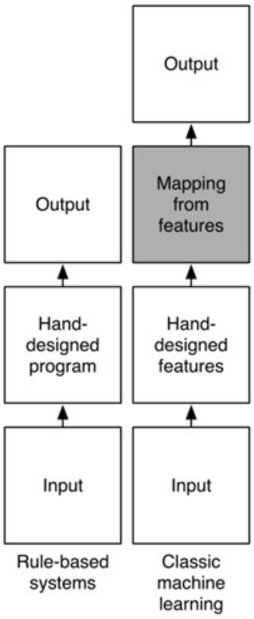
- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

72

72

BEYOND THE CLASSICS

Classical Computer Vision with Machine Learning



AI: three generations

3rd (current) wave of AI: **since ~2008**

- Based on deep machine learning
- Features automatically learnt from data

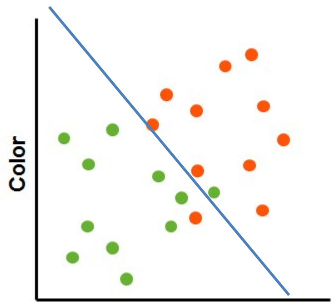
Performance comparable to human experts in narrow domains

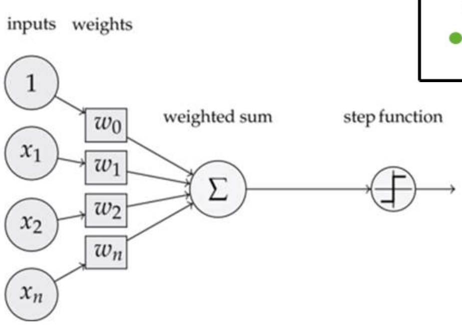
75

75

Do we need deep learning?

- For simple (linear) classification problems?



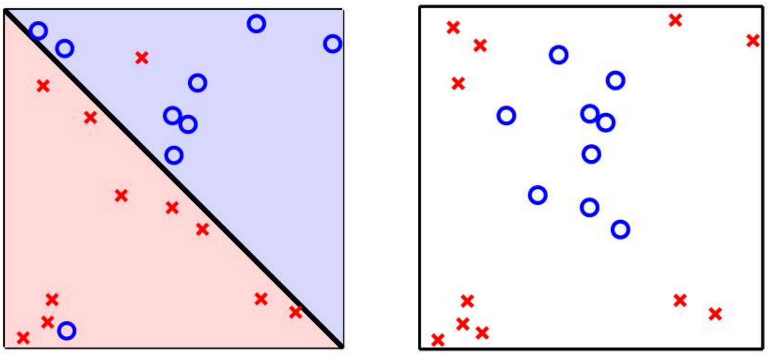


76

76

Do we need deep learning?

- The Linear Model has its Limits

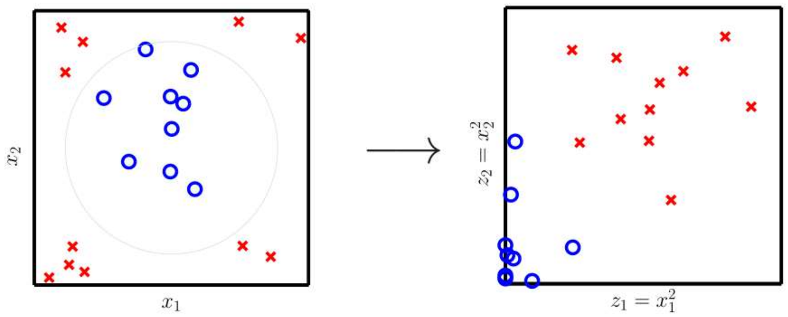


(a) Linear with outliers (b) Essentially nonlinear

To address (b) we need something more than linear.

Do we need deep learning?

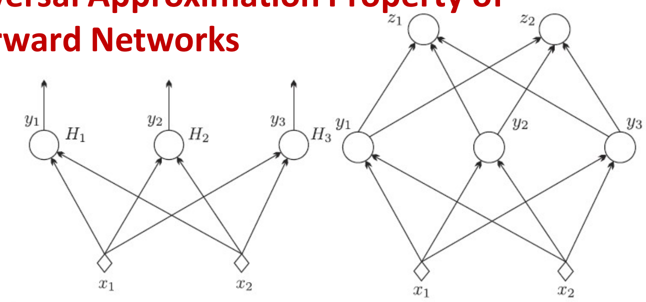
- Change Your Features Using a Transform

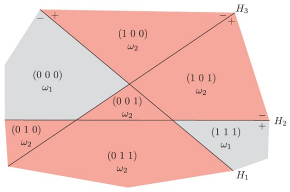


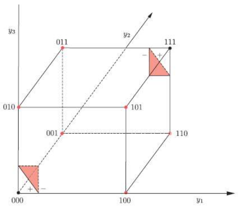
$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \longrightarrow \mathbf{z} = \Phi(\mathbf{x}) = \begin{bmatrix} 1 \\ x_1^2 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} 1 \\ \Phi_1(\mathbf{x}) \\ \Phi_2(\mathbf{x}) \end{bmatrix}$$

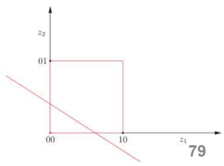
Do we need deep learning?

- For complex binary classification problems?
  - No: **Universal Approximation Property of Feed-forward Networks**





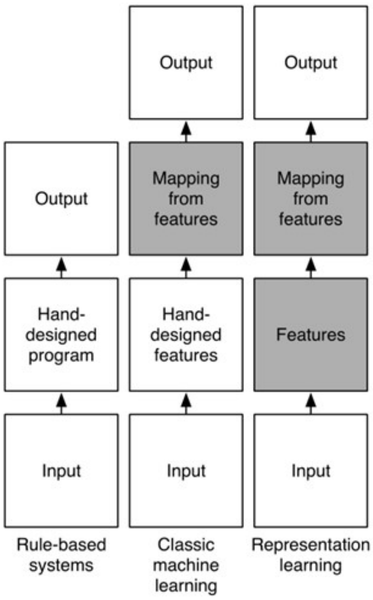




79

79

Representation Learning

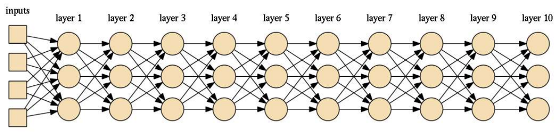


80

80

Do we need deep learning?

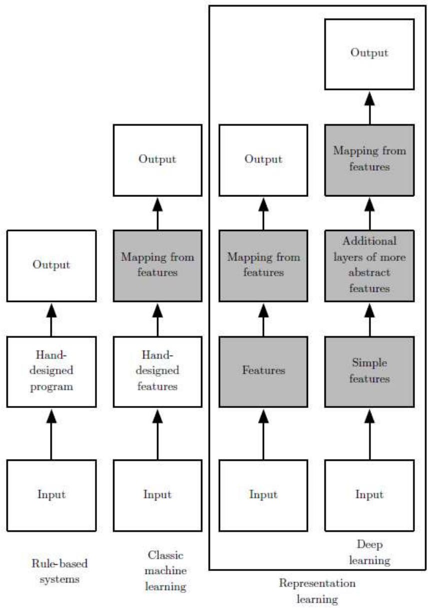
- For complex binary classification problems?
  Yes
  - In any learning task, we have to be concerned with what is feasibly "learnable" in a given representation.
  - Using networks with more layers, one can obtain more **compact representations** of the input-output relation.
    - We say that a network is **compact** if it consists of relatively few free parameters (few computational elements) to be learned/tuned during the training phase.
    - For a given number of training points, we expect compact representations to result in better generalization performance.
  - For complex tasks, where more complex concepts have to be learned, for example, recognition of a scene in a video recording, language and speech recognition, the underlying functional dependence is of a very complex nature so that we are unable to express it analytically in a simple way.



81

81

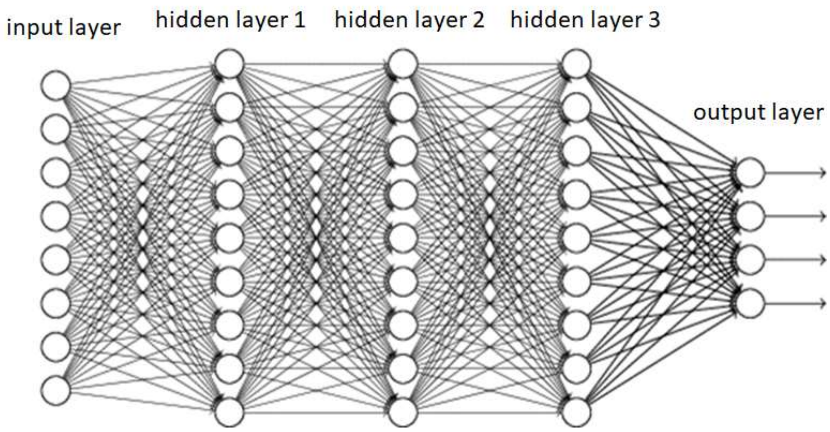
Representation Learning



82

82

Deep multilayer perceptron



83

83

Questions

- Question: What does a hidden unit do?
- Answer: It can be thought of as a classifier or feature detector.
- Question: How many layers? How many hidden units?
- Answer: Cross-validation or hyper-parameter search methods are the answer. In general, the wider and the deeper the network the more complicated the mapping.
- Question: How do I set the weight matrices?
- Answer: Weight matrices and biases are learned. First, we need to define a measure of quality of the current mapping. Then, we need to define a procedure to adjust the parameters.

From Ranzato CVPR tutorial

84

84

Explaining the current success of DNN

- Larger datasets have reduced the degree to which statistical generalization is a challenge for neural networks.
- Neural networks have become much larger, due to more powerful computers, and better software infrastructure.
- Algorithmic improvements
  - replacement of mean squared error with the cross-entropy family of loss functions
  - replacement of sigmoid hidden units with piecewise linear hidden units, such as rectified linear units

85

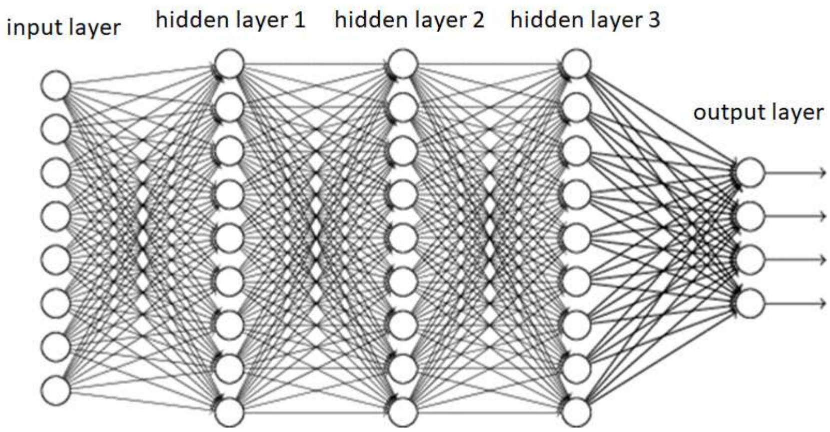
85

A BEGINNER'S GUIDE TO UNDERSTANDING CONVOLUTIONAL NEURAL NETWORKS

86

86

Deep multilayer perceptron

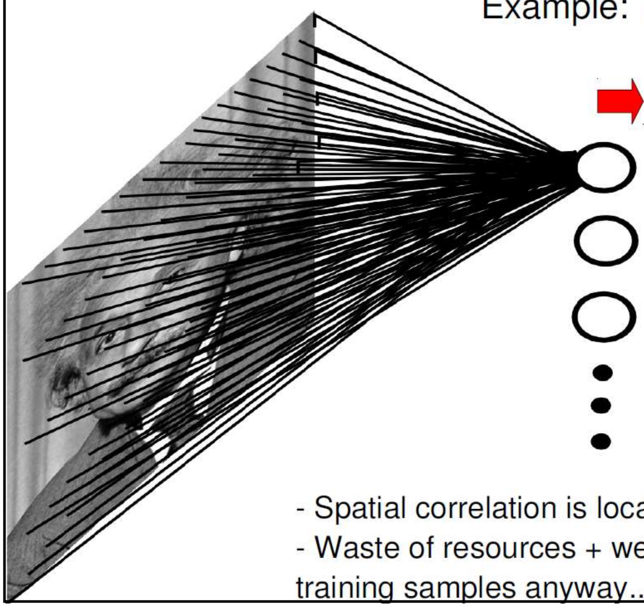


87

87

MLP for images

Example: 200x200 image
40K hidden units
~2B parameters!!!



- Spatial correlation is local
- Waste of resources + we have not enough training samples anyway..

88

88

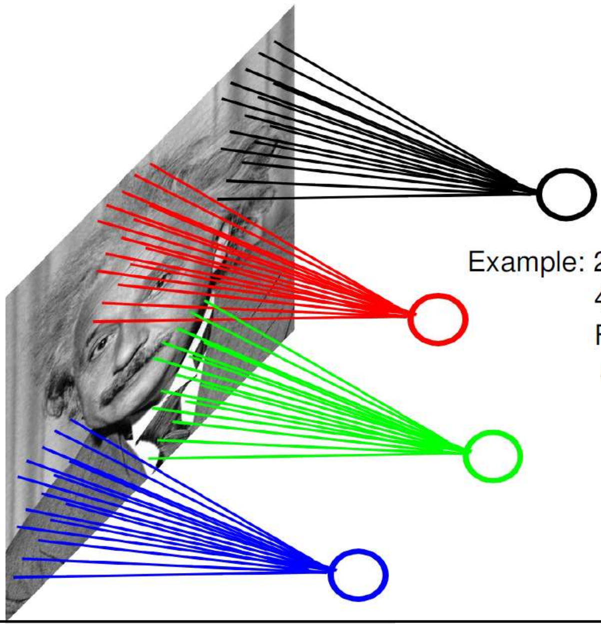
Convolutional Neural Network

- Scale up neural networks to process very large images /video sequences
  - Sparse connections
  - Parameter sharing
- Automatically generalize across spatial translations of inputs
- Applicable to any input that is laid out on a grid (1D, 2D, 3D, …)

89

89

Locally Connected Layer

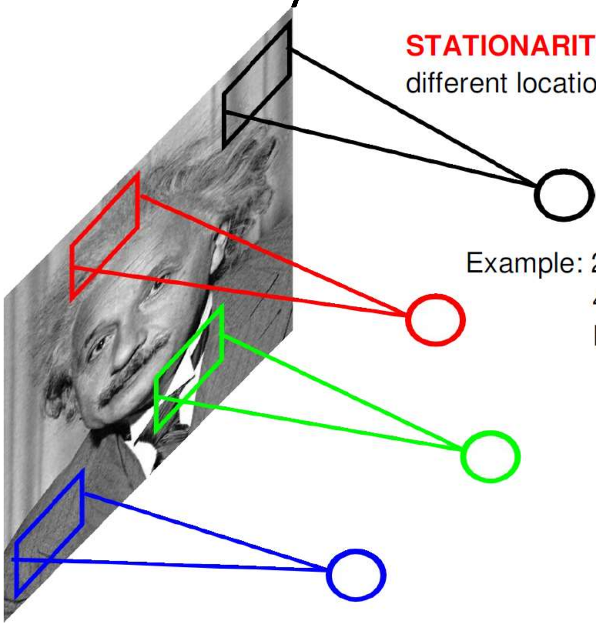


Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

90

90

Locally Connected Layer



STATIONARITY? Statistics is similar at different locations

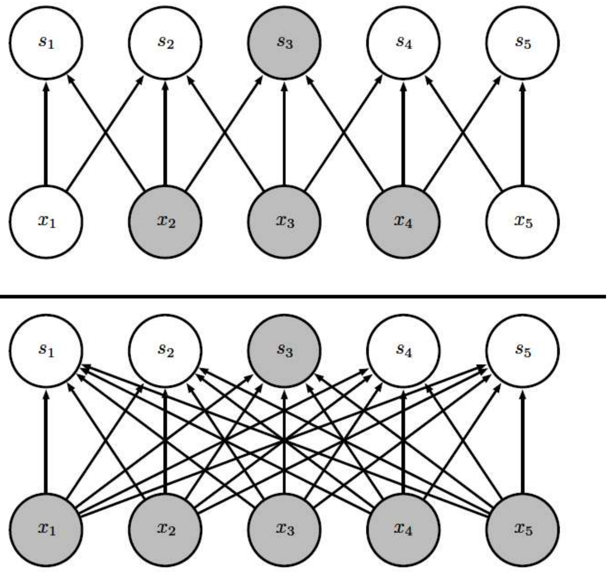
Example: 200x200 image
40K hidden units
Filter size: 10x10

91

91

Sparse Connectivity

Sparse connections due to small convolution kernel

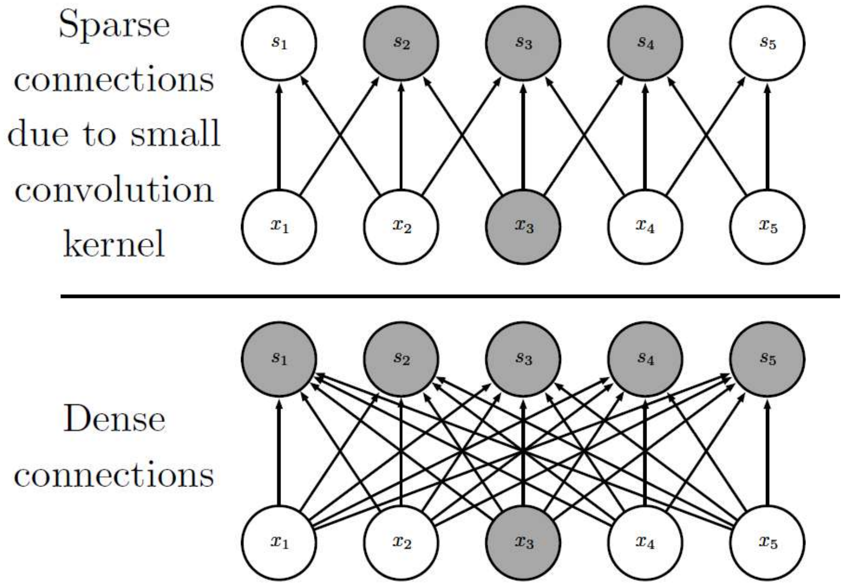


Dense connections

92

92

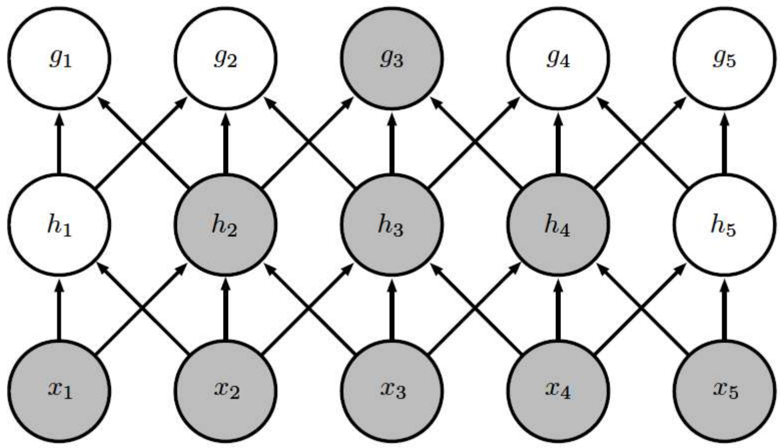
Sparse Connectivity



93

93

Growing Receptive Fields

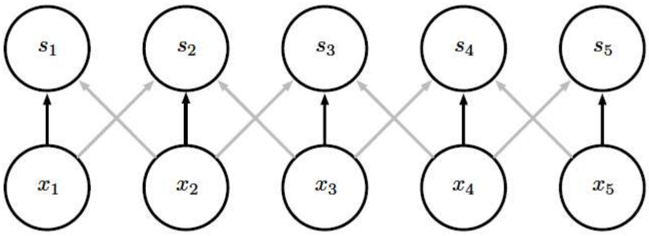


94

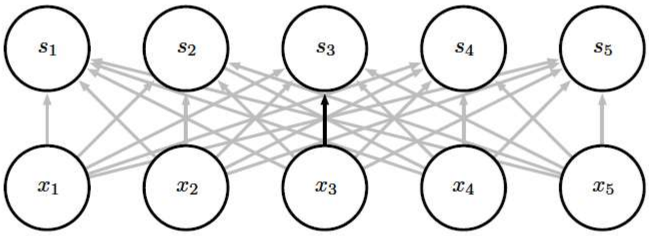
94

Parameter Sharing

Convolution shares the same parameters across all spatial locations



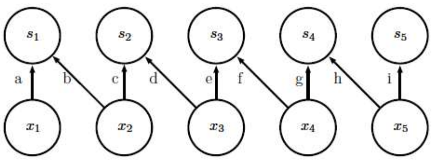
Traditional matrix multiplication does not share any parameters



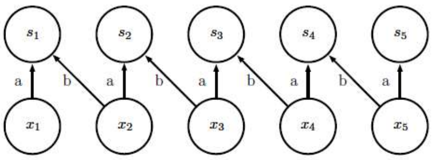
95

95

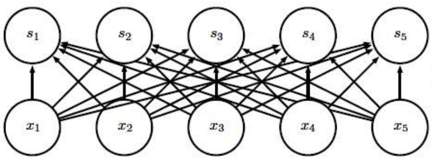
Kinds of Connectivity



Local connection: like convolution, but no sharing



Convolution

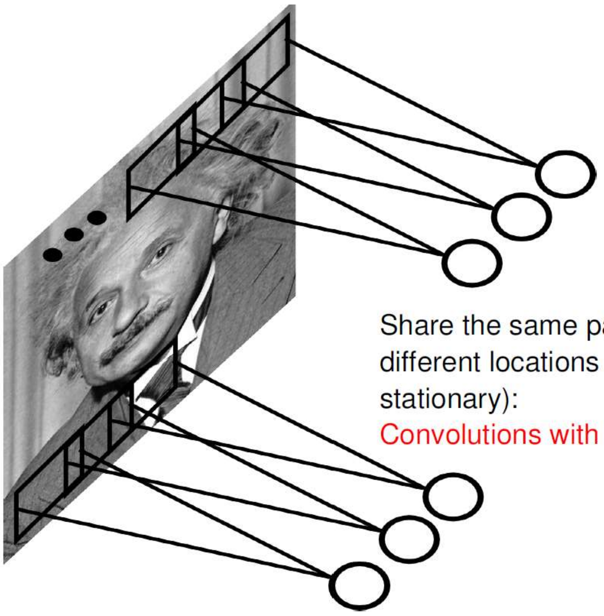


Fully connected

96

96

Convolutional Layer



Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

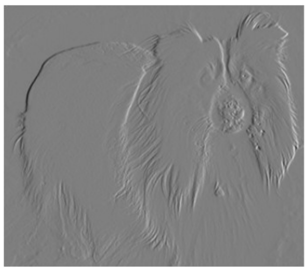
97

97

Convolutional Layer

1	0	-1

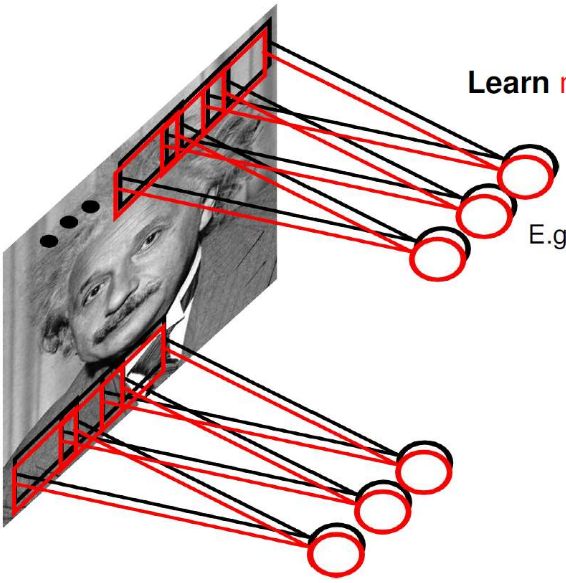
kernel



98

98

Convolutional Layer



Learn multiple filters.

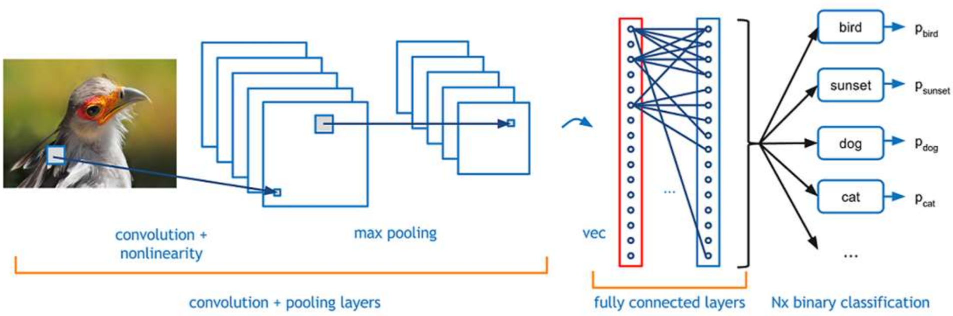
E.g.: 200x200 image
100 Filters
Filter size: 10x10
10K parameters

99

99

Convolutional Neural Networks

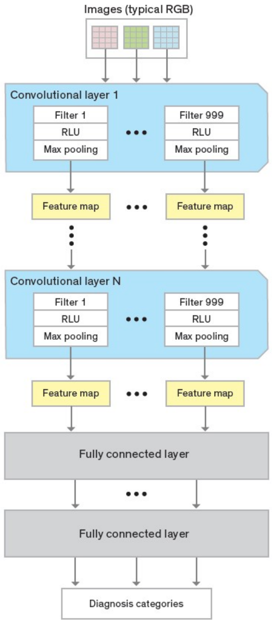
- Architecture overview
  - Convolutional Layer
  - Pooling Layer
  - Fully-Connected Layer
  - Normalization Layer



100

100

Inside a Convolutional Neural Network

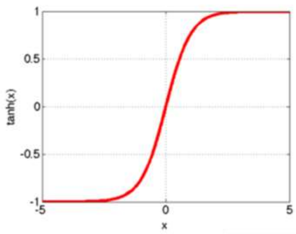


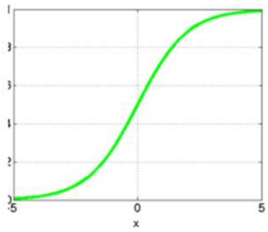
101

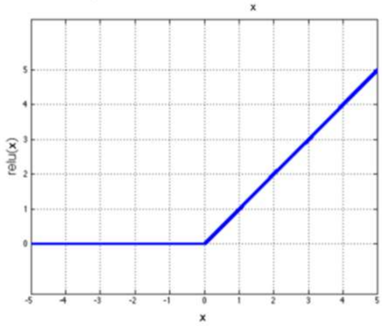
101

Activation Function

- Non-linearity
  - Per-element (independent)
  - Tanh
  - Sigmoid: 1/(1+exp(-x))
  - Rectified linear
    - Simplifies backprop
    - Makes learning faster
    - Avoids saturation issues
    - Preferred option



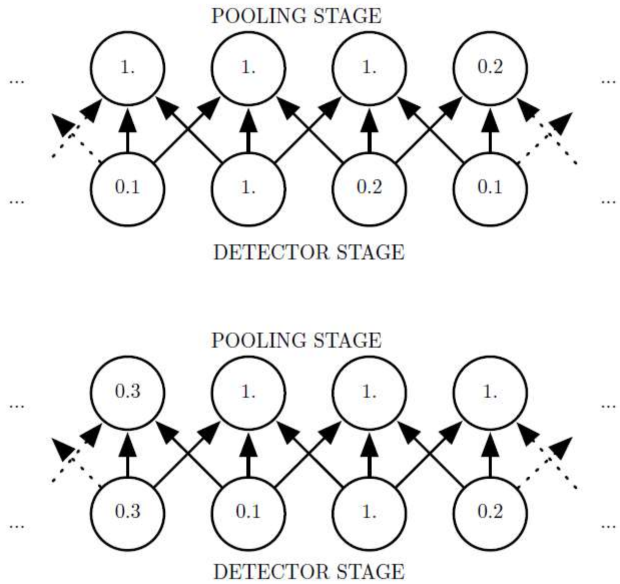




102

102

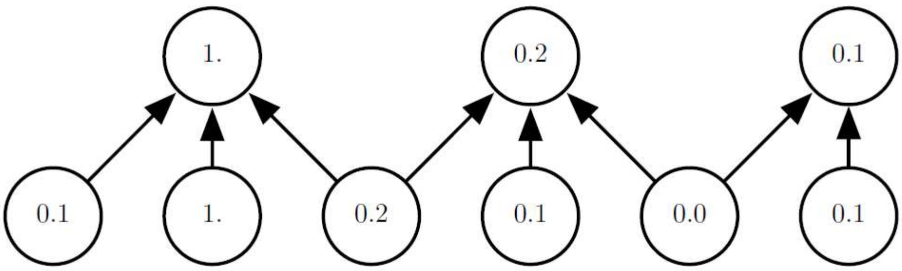
Max Pooling and Invariance to Translation



103

103

Pooling with Downsampling



104

104

Q&A

- Question: How many feature maps? What's the size of the filters?
- Answer: Usually, there are more output feature maps than input feature maps. Convolutional layers can increase the number of hidden units by big factors (and are expensive to compute). The size of the filters has to match the size/scale of the patterns we want to detect (task dependent).
- Question: How should I set the size of the pools?
- Answer: It depends on how much “invariant” or robust to distortions we want the representation to be. It is best to pool slowly (via a few stacks of conv-pooling layers).

105

105

REGULARIZATION

106

106

Regularization

- To build a machine learning algorithm we specify model family, a cost function and optimization procedure
- Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error
  - There are many regularization strategies
- Regularization works by trading increased bias for reduced variance. An effective regularizer is one that makes a profitable trade, reducing variance significantly while not overly increasing the bias.

107

Parameter Norm Penalties

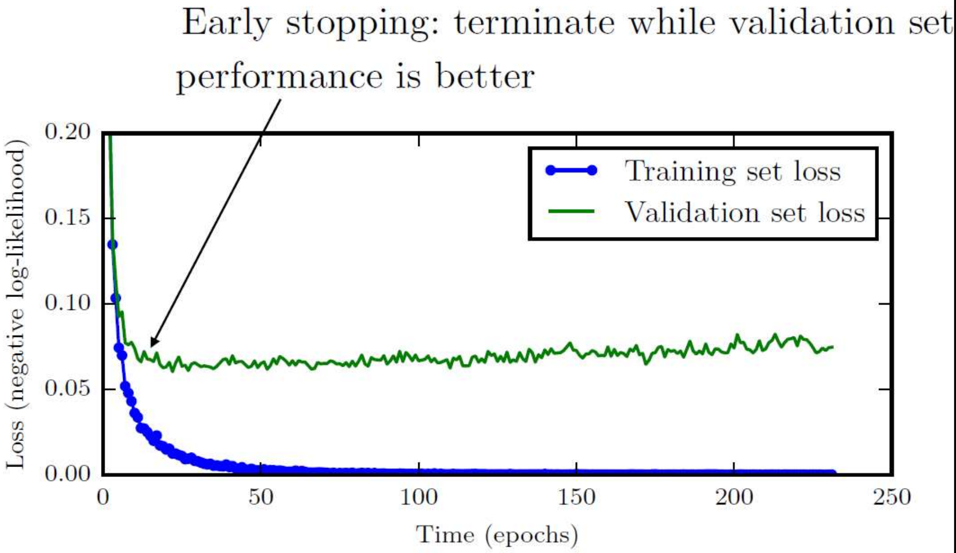
- Penalize complexity in the loss function
  - Model complexity
  - Weight Decay

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2,$$

- Penalize feature map complexity

108

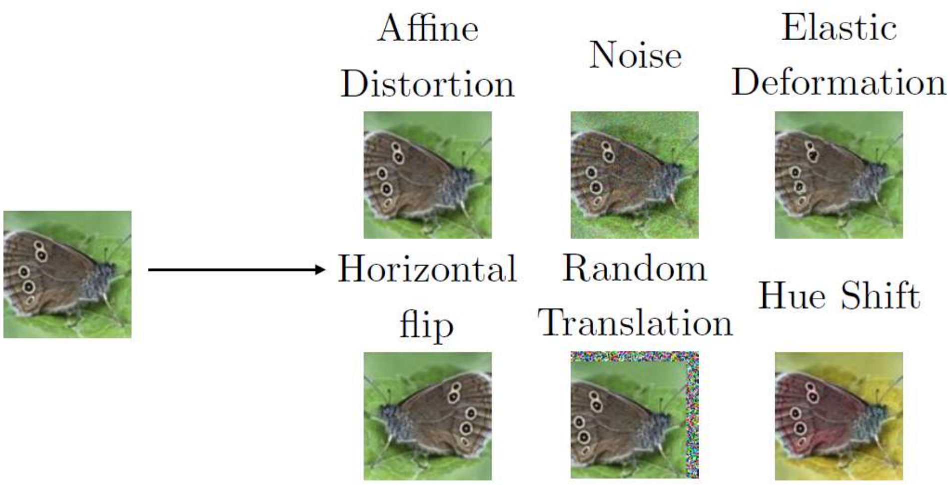
Early stopping



109

109

Dataset Augmentation



110

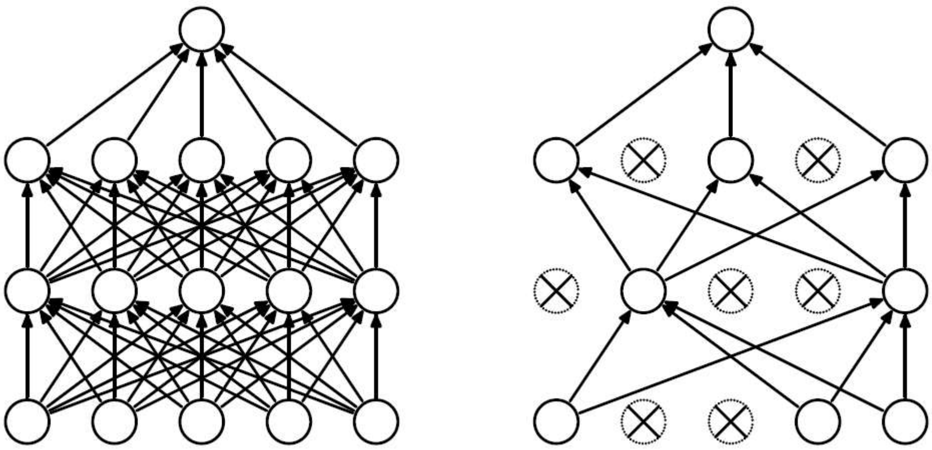
110

Noise Robustness

- Another way that noise has been used in the service of regularizing models is by adding it to the weights
  – Pushes the model to flat (loss) regions

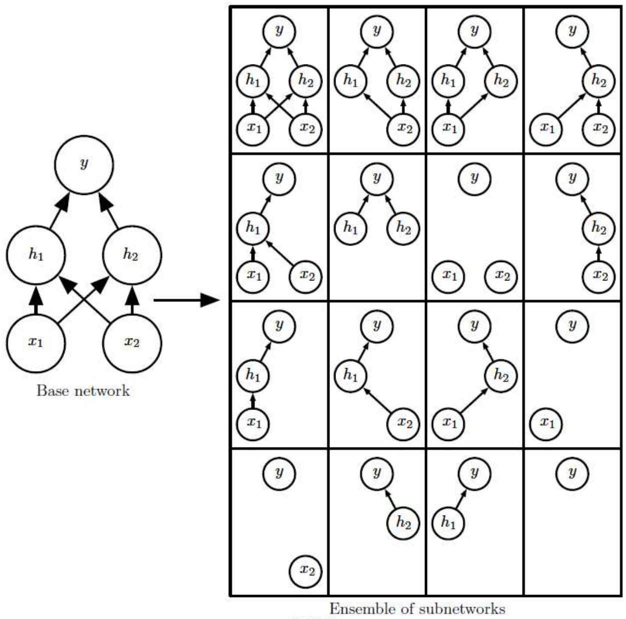
- Injecting Noise at the Output Targets
  – label smoothing

Dropout



(a) Standard Neural Net (b) After applying dropout.

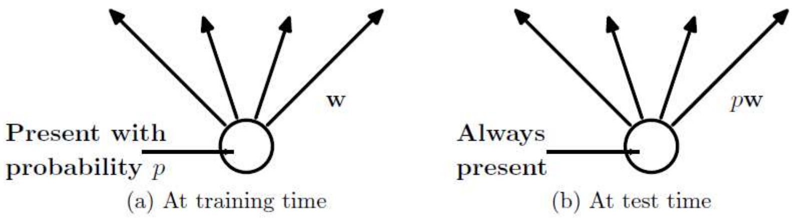
Dropout



(Goodfellow 2016) 113

113

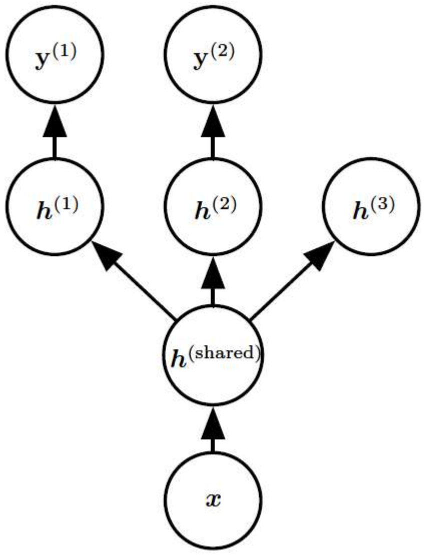
Dropout



114

114

Multi-Task Learning

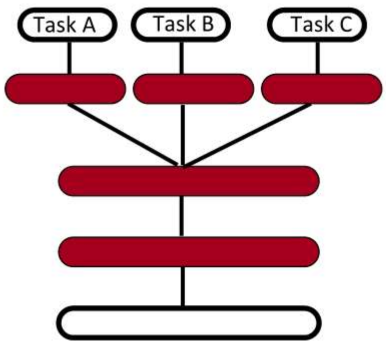


115

115

Multi-Task Learning

- Generalizing better to new tasks (tens of thousands!) is crucial to approach AI
- Example: speech recognition, sharing across multiple languages
- Deep architectures learn good intermediate representations that can be shared across tasks

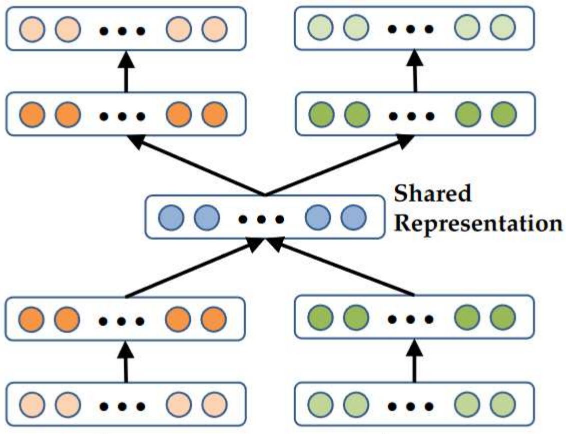


E.g. dictionary, with intermediate concepts re-used across many definitions

116

116

Multimodal Deep Learning



117

117

Others

- Sparse Representations=Sparse Feature Maps
- Sparse Models
- Add noise (additive noise, multiplicative noise)
  - To data
  - To model
- Ensembles
- Adversarial Examples
  - Training on adversarial examples is mostly intended to improve security, but can sometimes provide generic regularization

118

118

Transfer Learning

- In practice, very few people train an entire Convolutional Network from scratch (with random initialization). Instead, it is common to pretrain a ConvNet on a very large dataset
  - ConvNet as fixed feature extractor
  - Fine-tuning the ConvNet

119

119

Transfer Learning

- In practice, very few people train an entire Convolutional Network from scratch
  - Example: Face Recognition

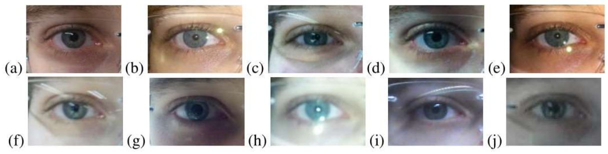
  - Use VGG Deep Net (trained with millions of faces – but for a different task) to extract a feature vector by removing the output layer
  - Use euclidean distance (or variations) for matching
    - Improves over state of the art approaches in several settings.

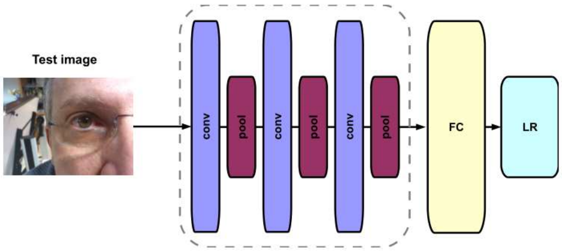
120

120

Transfer Learning

- Cross Sensor Adaptation for periocular recognition





121

121

References

- Richard Szeliski, *Computer Vision: Algorithms and Applications*, 2010 - http://szeliski.org/Book/
- David Forsyth and Jean Ponce, *Computer Vision: A Modern Approach* 2nd Edition, 2012
- Simon J. D. Prince, *Computer Vision: Models, Learning, and Inference*, 2012
- Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing* 3rd Edition, 2007
- Richard Hartley and Andrew Zisserman, *Multiple View Geometry* 2nd Edition, 2004
- Ian Goodfellow , Yoshua Bengio, Aaron Courville and Francis Bach, *Deep Learning*, 2016

122

References

- Fei-Fei Li et al. (Stanford University) - CS 131 Computer Vision: Foundations and Applications
  - http://vision.stanford.edu/teaching/cs131_fall1617/index.html
- James Tompkin et al. (Brown University) - CSCI 1430: Introduction to Computer Vision
  - https://cs.brown.edu/courses/csci1430/
- Kristen Grauman et al. (University of Texas at Austin) - CS 376: Computer Vision
  - http://vision.cs.utexas.edu/376-spring2018/
- Rob Fergus et al. (New York University) - CSCI-GA.2271-001: Computer Vision
  - https://cs.nyu.edu/~fergus/teaching/vision/index.html

123

References

- Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
- Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, John Wiley & Sons, 2001
- Thomas Mitchell, Machine Learning, McGraw-Hill, 1997.
- P. Domingos, "A few useful things to know about machine learning," CACM, 2012
- Andrew Moore, Support Vector Machines Tutorial, http://www.autonlab.org/tutorials/svm.html

124

124

References

- Selim Aksoy, Introduction to Pattern Recognition, Part I, http://retina.cs.bilkent.edu.tr/papers/patrec_tutorial1.pdf
- Ricardo Gutierrez-Osuna, Introduction to Pattern Recognition, http://research.cs.tamu.edu/prism/lectures/pr/pr_l1.pdf
- Pedro Domingos, Machine Learning, http://courses.cs.washington.edu/courses/cse446/14wi/
- Kristen Grauman, Discriminative classifiers for image recognition, http://www.cs.utexas.edu/~grauman/courses/spring2011/slides/lecture22_classifiers.pdf
- Victor Lavrenko and Nigel Goddard, Introductory Applied Machine Learning, http://www.inf.ed.ac.uk/teaching/courses/iaml/

125

125

References

- Recognizing and Learning Object Categories http://people.csail.mit.edu/torralba/shortCourseRLOC/index.html
- Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes, (K. Murphy, A. Torralba, W. Freeman), NIPS 2003
- Max-Margin Markov Networks , (B. taskar, C. Guestrin, D. Koller), NIPS 2004
- Large Margin Methods for Structured and Interdependent Output Variables, (I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun), JMLR, vol 6, 2005
- Learning Spatial Context: Using Stuff to Find Things, (G. heitz, D. Koller), ECCV 2008, http://ai.stanford.edu/~gaheitz/Research/TAS/
- An Empirical Study of Context in Object Detection, (S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, M. Hebert), CVPR 2009 http://www.cs.cmu.edu/~santosh/projects/context.html
- Generative Models for Visual Objects and Object Recognition via Bayesian Inference, L. Fei-Fei, 2006
- Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities, (B. Yao, L. Fei-Fei), CVPR 2010 http://videolectures.net/cvpr2010_fei_fei_mmco/
- No Hype, All Hallelujah: Structured Models in Computer Vision, (S. Nowozin), NIPS 2010

126

126

References

- Graphical Models for Time Series, (D. Barker, A. T. Cemgil), IEEE Signal Processing Magazine, vol 27, 2010
- Dynamic Graphical Models, (J. Bilmes), IEEE Signal Processing Magazine, vol 27, 2010
- A Martingale Framework for Detecting Changes in Data Streams by Testing Exchangeability, (S. Ho, H. Wechsler), TPAMI 2010
- Introduction to Statistical Relational Learning, (L. Getoor, B. Taskar), The MIT Press 2007
- Combining Video and Sequential Statistical Relational Techniques to Monitor Card Games, (L. Antanas, B. Gutmann, I. Thon, K. Kersting, L. De Raedt), ICML 2010
- Relational Learning for Collective Classification of Entities in Images, (A. Chechetka, D. Dash, M. Philipose), AAAI 2010
- Grouplet: A Structured Image Representation for Recognizing Human and Object Interactions, (B. Yao, L. Fei-Fei), CVPR 2010

Thank You for Your Attention!

127