Neural networks and image classification

Andrey Stotskiy

3 October 2023

Outline

1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models
3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

Outline

1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models
3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

Computer vision (recap)

Vision task: understand, what is depicted on an image

Computer vision: developing a computer model for vision. CV is a part of Artifial Intelligence (AI)

Turing test for CV: answer any question about the image that can be aswered by a human

Outline

1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models
3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

Binary classification

- Does this image contain a pedestrian?
- Binary answer $y \in \{ \underset{\text{no}}{0}, \underset{\text{yes}}{1} \}$
- Alternatively, the estimated probability of the positive answer $p_{\text{yes}} \in [0;\ 1]$

Multiclass classification

- Which object is shown on this image?
- The set of *allowed* object classes is determined in advance!

- Integer answer $y \in \left\{ \underset{\text{cat}}{1}, \underset{\text{car}}{2}, \ldots, \underset{\text{rat}}{S} \right\}$
- Alternatively, a list of estimated probabilities

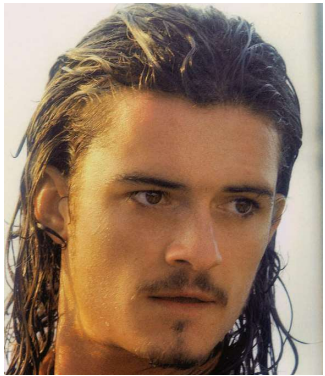
$$p_i \in [0;\ 1] \quad i \in 1, \ldots, S \quad \sum_{i=1}^{S} p_i = 1$$

Attribute recognition

Male
Asian
Bearded
Smiling

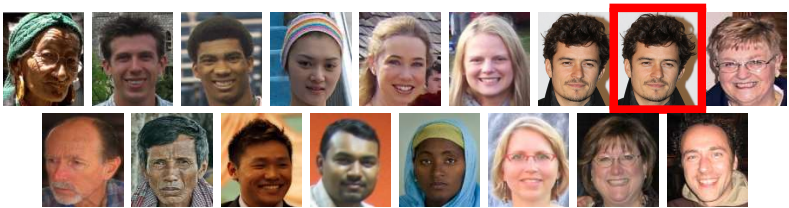
- *Attributes* are properties or characteristics that are commonly expressed by some object
- Human attributes may include race, sex, age, color of hair, current emotional state or the presence of wearable accessories such as masks, glasses and hats
- Attribute recognition can often be reduced to one or more classification tasks, for example:
  - *sex* → binary
  - *race* → multiclass
  - *age* → multiclass (over discrete age groups)

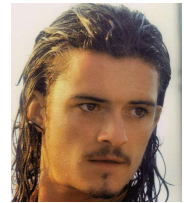
Verification



Are these two images of the same person?

Identification





- Given a database of images and a single "query" image, determine if the person in the "query" image is present in the database
  - "white" or allow list (face authentication for staff)
  - "black" or deny list (find criminals from surveillance)
- Identification can be reduced to N pairwise Verification tasks

Outline

1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models
3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

Metrics: Goodhart's law

When a measure becomes a target,
it ceases to be a good measure.

Metrics: Accuracy

- Accuracy - percentage of correctly classified samples

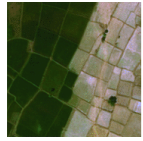
Dataset	CNN	Original	BP[23]	CBP[11]	KP	Others	
CUB [43]	VGG-16 [38]	73.1*	84.1	84.3	**86.2**	82.0	84.1
	ResNet-50 [15]	78.4	N/A	81.6	84.7	[18]	[16]
Stanford Car [19]	VGG-16	79.8*	91.3	91.2	**92.4**	**92.6**	82.7
	ResNet-50	84.7	N/A	88.6	91.1	[18]	[14]
Aircraft [27]	VGG-16	74.1*	84.1	84.1	**86.9**	80.7	
	ResNet-50	79.2	N/A	81.6	85.7	[14]	
Food-101 [4]	VGG-16	81.2	82.4	82.4	84.2	50.76	
	ResNet-50	82.1	N/A	83.2	**85.5**	[4]	

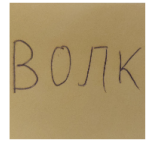
Table 2. Performance comparisons among all baselines, where KP is the proposed kernel pooling method with learned coefficients. Following the standard experimental setup, we use the input size of 448×448 for CUB, Stanford Car and Aircraft datasets except the original VGG-16 (marked by an asterisk *), which requires a fixed input size of 224×224. For Food-101, we use the input size of 224×224 for all the baselines.

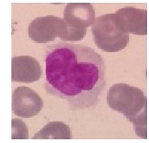
- Top-K Accuracy (aka Rank) - percentage of sample for which the correct class is within K most likely predicted classes (often K=5)

Data domains and modalities









Every computer vision algorithm is designed to operate on images sampled from some *statistical population*. This population is described by an empirical distribution over the set of all "valid" (for that algorithm) images.

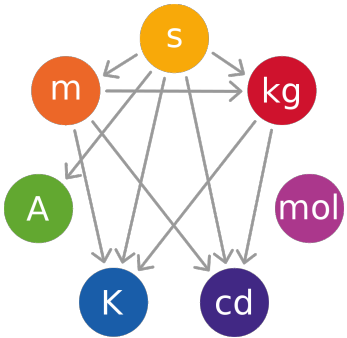
$$img \sim \mathrm{P}(\mathbb{I}) \qquad\qquad \mathbb{I} \subseteq \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times\mathrm{C}}$$

These algorithms work by exploiting the inherent properties and invariants of the *statistical population* they support.

Outline

1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models
3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

Datasets



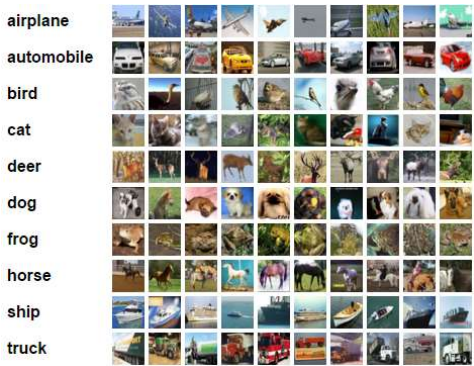
These leaderboards are used to track progress in Classification

Trend	Dataset	Best Model
	InDL	ConvNext
	N-ImageNet	Event Spike Tensor
	N-ImageNet (mini)	Event Imge
	ImageNet C-OOD (class-out-of-distribution)	ViT-L/32-384 with Max-logit
	Autoimmune Dataset	Swin Transformer Base (Patch 4 Window 12)

⋮

paperswithcode.com/task/classification-1

CIFAR-10 and CIFAR-100



Subset of the TinyImages collection

- 60000 images total
- CIFAR-10: 10 classes
  - 5000 training images per class
  - 1000 testing images per class
- CIFAR-100: 100 classes
  - 500 training images per class
  - 100 testing images per class

Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.
www.cs.toronto.edu/~kriz/cifar.html

ImageNet

Goal: create a dataset with at least 1000 images for each of the original 117 000 synsets/classes

~14 000 000 images (~1 000 000 images with bounding box annotations)
~22 000 non-empty classes (~**10 000 classes with at least 1000 examples**)

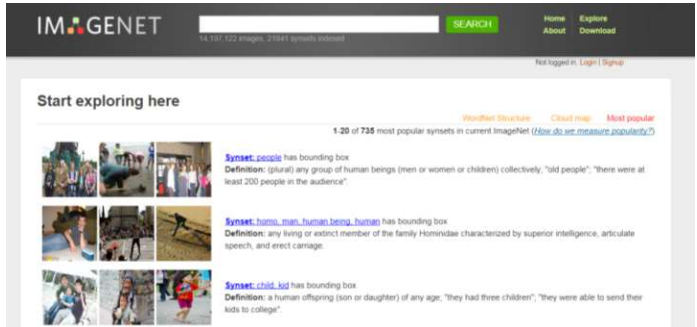


image-net.org

ImageNet: Annotation problems

OpenImages

Goal: create the largest **open** dataset of real-life photograps with diverse annotations

- ~9 000 000 images **licensed under CC BY 2.0**
- ~60 000 000 annotations for ~20 000 categories
- Various supplementary annotations are also available (for example, localized text descriptions)

storage.googleapis.com/openimages/web/index.html

Fine-graded classification

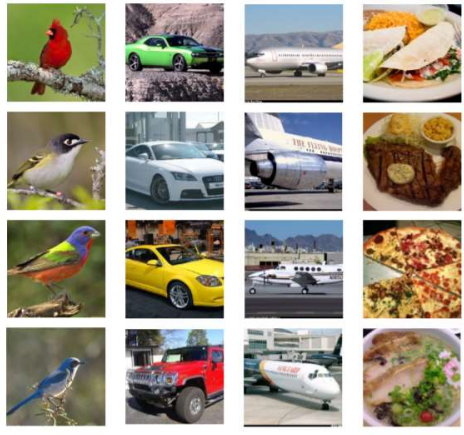
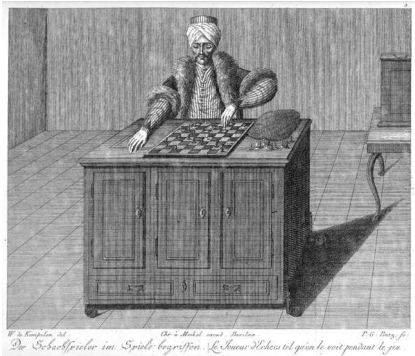


Figure 7. Images we used for visual recognition. From left to right, each column contains examples from CUB Bird [43], Stanford Car [19], Aircraft [27] and Food-101 [4].

Outline

1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models
3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

Mechanical Turk



"Mechanical Turk, Automaton Chess Player" was a robot created **in 1770** that could play chess (and even beat competent players). In 1820 it was revealed that the robot couldn't actually play chess by itself and that it was instead **controlled by a human sitting in a hidden compartment**.

Galaxy Zoo

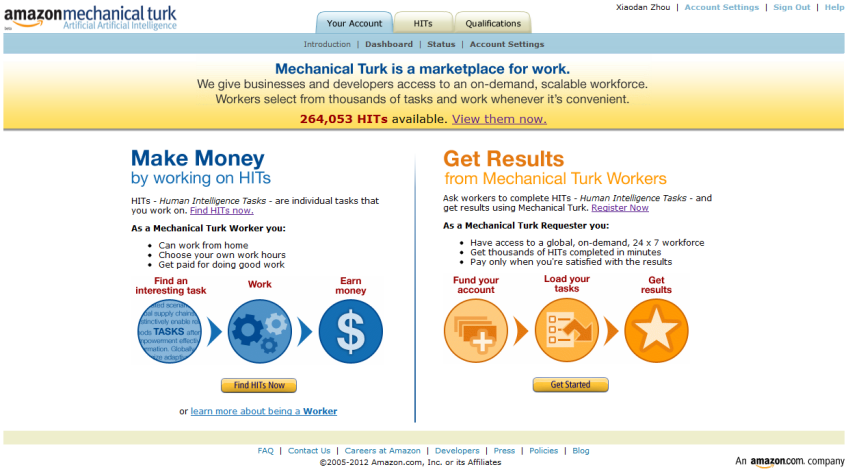
GALAXY ZOO galaxyzoo.org

- Classification of galaxy images
- The first large scale project of this kind
- More than 150 000 volunteers created over 60 000 000 annotations in a single year **for free (!)**

XKCD: 1897


TO COMPLETE YOUR REGISTRATION, PLEASE TELL US WHETHER OR NOT THIS IMAGE CONTAINS A STOP SIGN:
NO
YES
ANSWER QUICKLY—OUR SELF-DRIVING CAR IS ALMOST AT THE INTERSECTION.
SO MUCH OF "AI" IS JUST FIGURING OUT WAYS TO OFFLOAD WORK ONTO RANDOM STRANGERS.

Annotation as a service



Amazon Mechanical Turk

mturk.com

Outline

1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations

2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models

3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

Classical method pipeline

Image I

Extract image features

f = features(I)

Analyze image features

p = analysis(f)

Prediction p

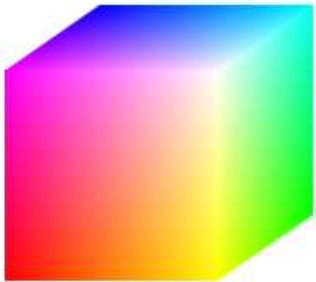
Outline

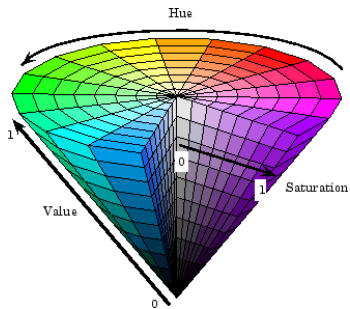
1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations

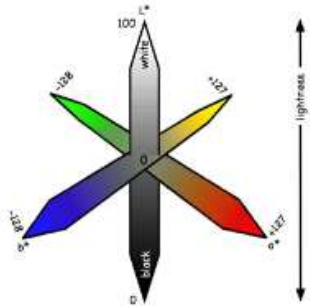
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models

3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

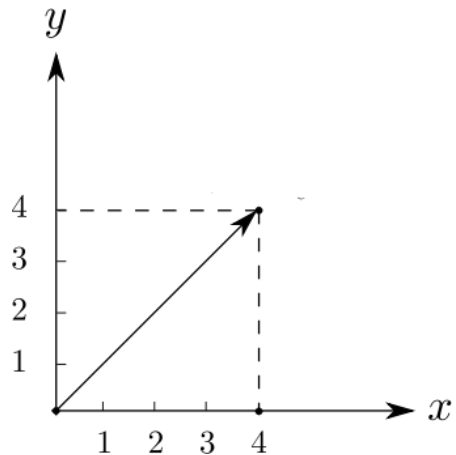
Features: color



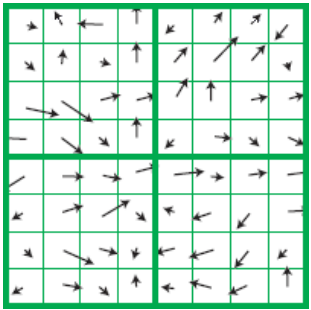


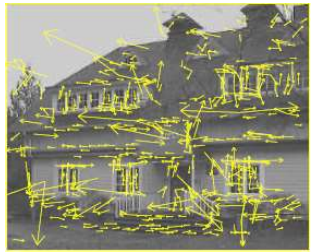


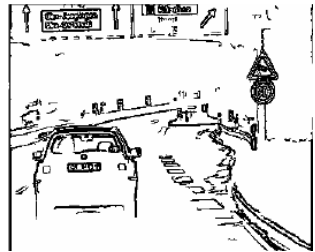
Features: position



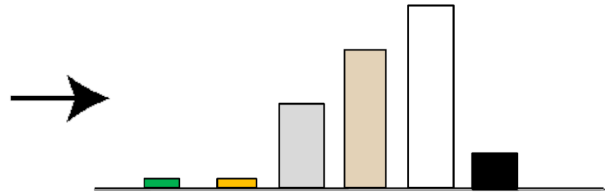
Features: gradients and edges

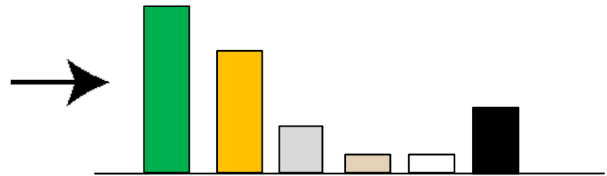




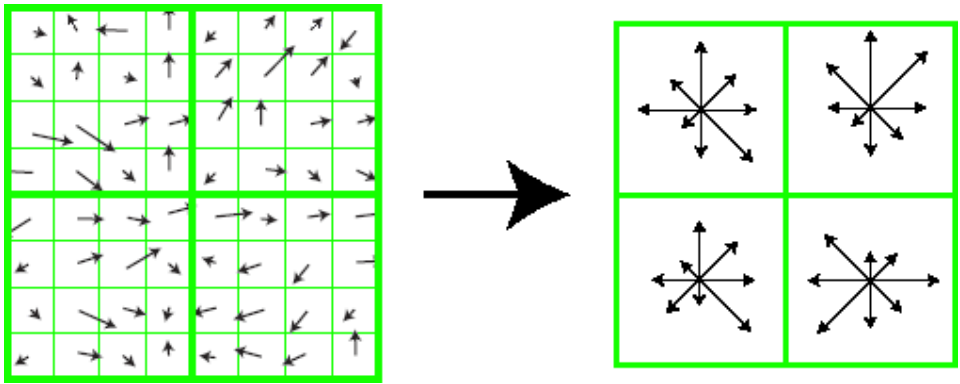


Features: statistics and histograms





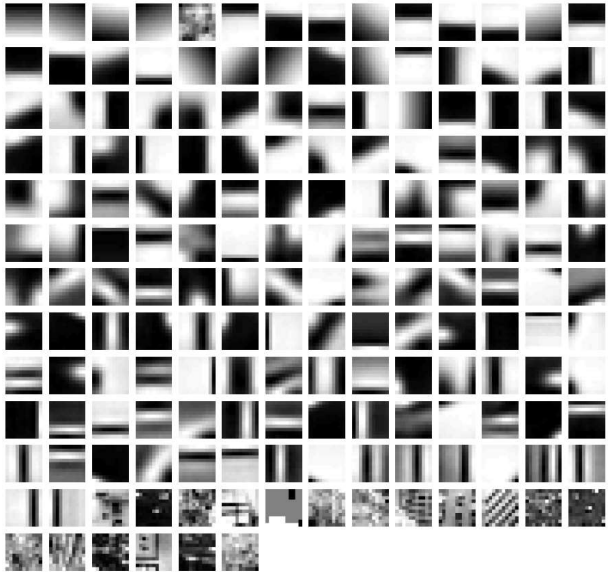
Features: statistics and histograms



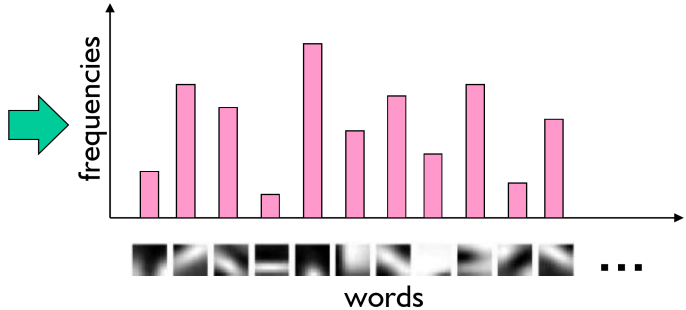
HoG: Histogram of oriented Gradients

Features: patterns and bags of words

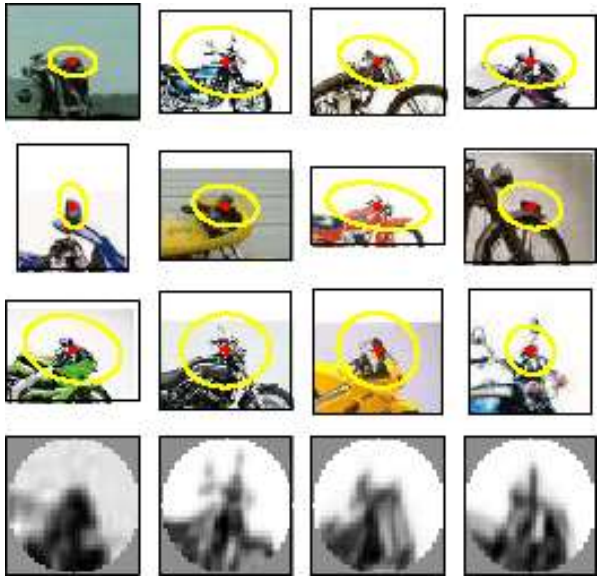
Features: patterns and bags of words



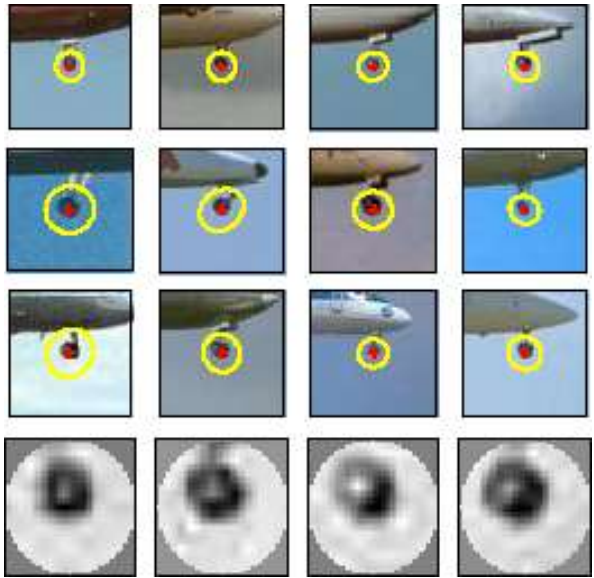
Features: patterns and bags of words



Features: patterns and bags of words



Features: patterns and bags of words



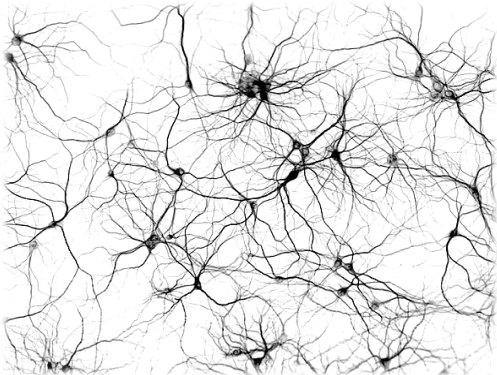
Outline

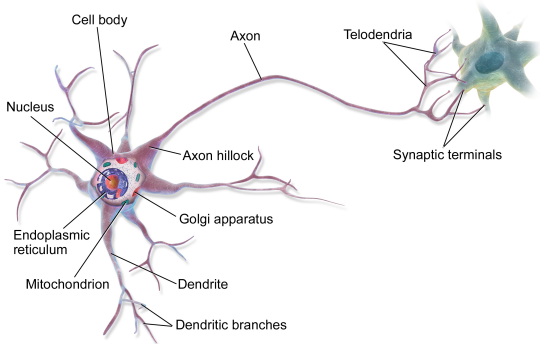
1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations

2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models

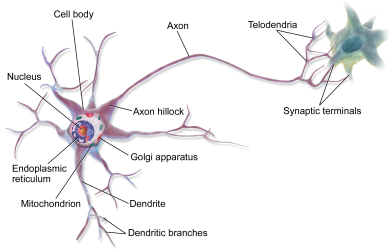
3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

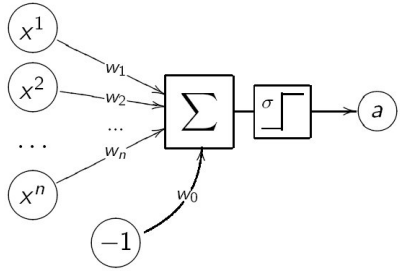
Biological neurons

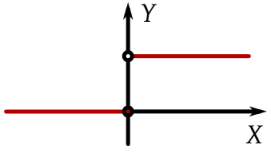




McCulloch-Pitts neuron model



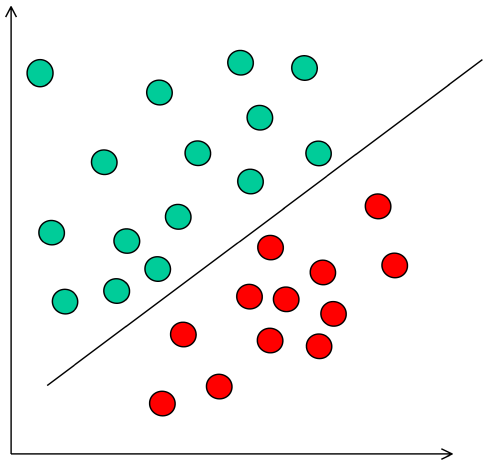




$$a\left(x, w\right) = \sigma\left(\sum_{i=1}^{n} w_i x_i - w_0\right)$$

Warren S. McCulloch; Walter Pitts (1943)

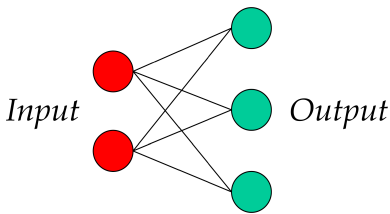
Neurons as linear classifiers



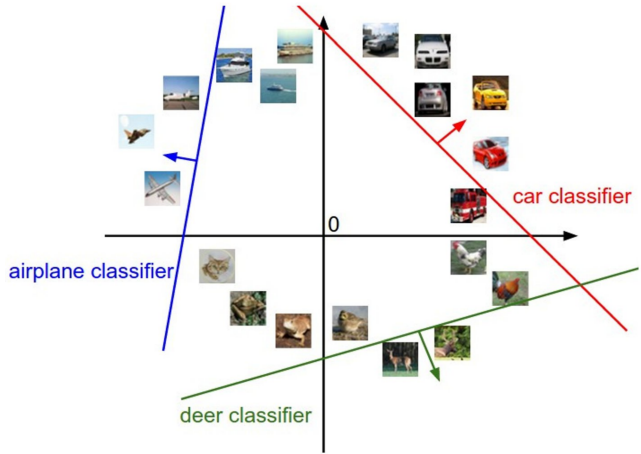
$$\sigma\left(\sum_{i=1}^{n} w_i x_i - w_0\right)$$

Optimal parameters w_i can be found using classical iterative methods.

Multiple classes



Perceptron



Outline

1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations

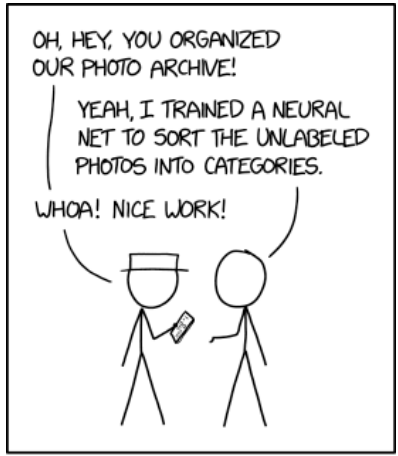
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models

3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

Outline

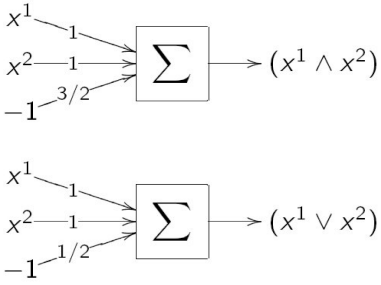
1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models
3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

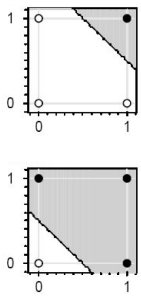
XKCD: 2173



ENGINEERING TIP:
WHEN YOU DO A TASK BY HAND, YOU CAN TECHNICALLY SAY YOU TRAINED A NEURAL NET TO DO IT.

Perceptron: representable functions



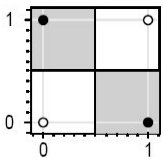


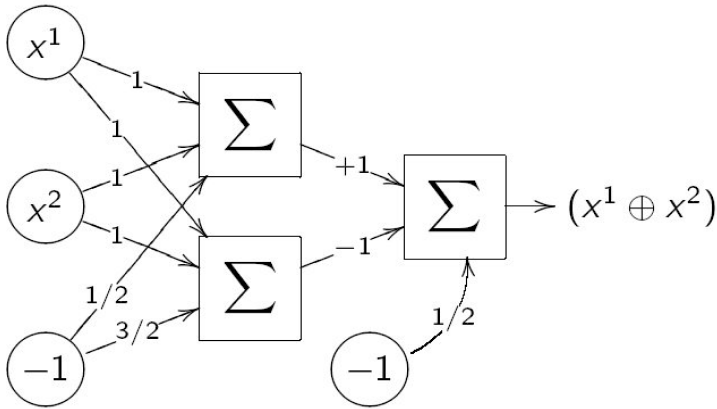
$$x^1 \wedge x^2 = \sigma\left(x^1 + x^2 - \frac{3}{2}\right)$$

$$x^1 \vee x^2 = \sigma\left(x^1 + x^2 - \frac{1}{2}\right)$$

$$\neg x = \sigma\left(-x + \frac{1}{2}\right)$$

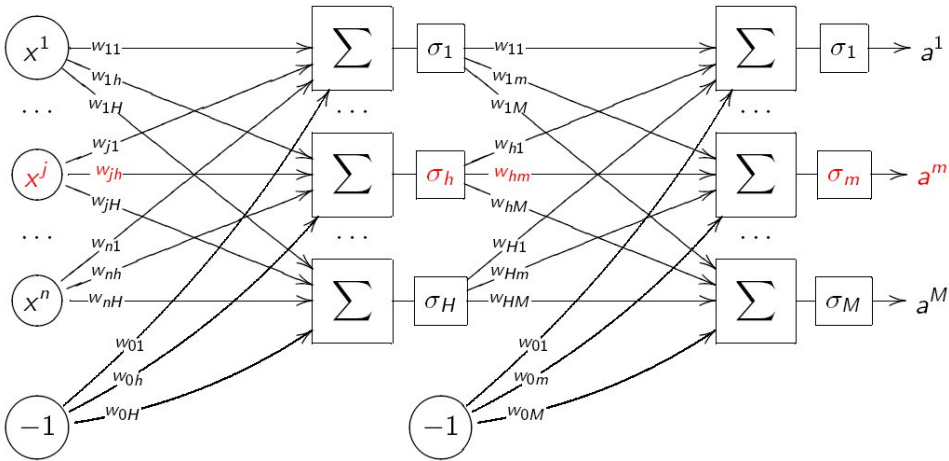
Perceptron: representable functions



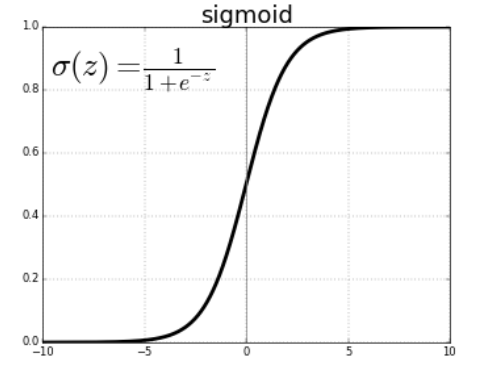


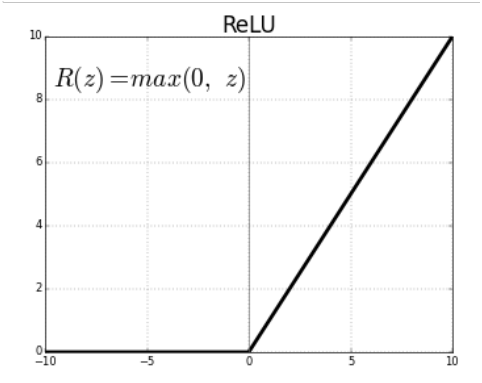
$x^1 \oplus x^2$ is **not** representable by a single perceptron.
but it **is** representable by two!

Multilayer perceptron



Activation functions





Universal approximation theorems

- Kolmogorov–Arnold representation theorem (1957)
- George Cybenko - arbitrary width, sigmoid activation function (1989)
- Kurt Hornik - arbitrary width, almost any activation function (1991)
- Gustaf Gripenberg - arbitrary depth, relu activation function (2003)
- Patrick Kidger - arbitrary depth, almost any activation function (2020)

Outline

1. Image classification task
   1.1. Classification and related tasks
   1.2. Method comparison and evaluation
   1.3. Datasets and benchmarks
   1.4. Labels and annotations
2. Classical computer vision methods (overview)
   2.1. Hand-crafted features
   2.2. Classical machine learning models
3. Modern computer vision methods (overview)
   3.1. Neural networks
   3.2. End-to-end trainig

Modern computer vision pipeline

Image I

Extract image features

f = features(I)

Analyze image features

p = analysis(f)

Prediction p

Image I

Neural Network

Prediction p

XKCD: 1838


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Loss function

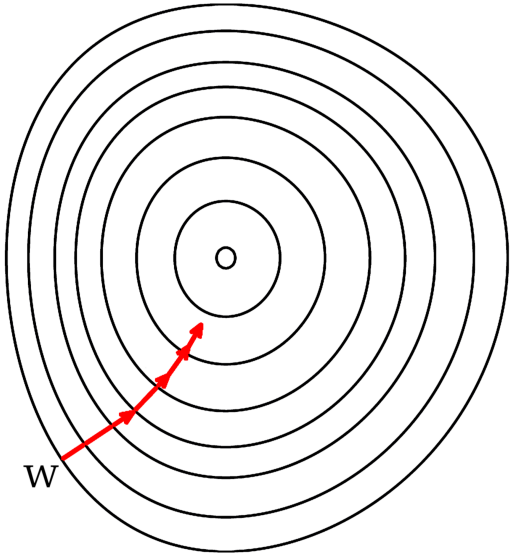
$$p_i^{\mathrm{pr}} = \frac{e^{y_i}}{\sum_{j=1}^{N} e^{y_j}}$$

Softmax activation function for the last layer

$$L\left(p^{\mathrm{pr}}, p^{\mathrm{gt}}\right) = -\sum_{i=1}^{N} p_i^{\mathrm{gt}} \cdot \log\left(p_i^{\mathrm{pr}}\right)$$

Categorical cross-entropy

Gradient descent



Minibatch stochastic gradient descent

Algorithm 2 Minibatch Stochastic Gradient Descent Training

Input: Function $f(\mathbf{x};\theta)$ parameterized with parameters θ.
Input: Training set of inputs $\mathbf{x_1},\ldots,\mathbf{x_n}$ and outputs $\mathbf{y_1},\ldots,\mathbf{y_n}$.
Input: Loss function L.
while stopping criteria not met **do**
    Sample a minibatch of m examples $\{(\mathbf{x_1},\mathbf{y_1}),\ldots,(\mathbf{x_m},\mathbf{y_m})\}$
    $\hat{\mathbf{g}} \leftarrow 0$
    for $i = 1$ to m **do**
        Compute the loss $L(f(\mathbf{x_i};\theta),\mathbf{y_i})$
        $\hat{\mathbf{g}} \leftarrow \hat{\mathbf{g}} +$ gradients of $\frac{1}{m}L(f(\mathbf{x_i};\theta),\mathbf{y_i})$ w.r.t θ
    $\theta \leftarrow \theta + \eta_k\hat{\mathbf{g}}$
return θ

Backwards gradient propagation

Will be discussed in more detail during the seminar