Introduction

Elena Trunz

September 11, 2023

Uni Bonn

Overview

Organization

Workload

- Course duration: three weeks
- 15 $\times$ lectures
- 15 $\times$ practical exercise (programming)

$\rightarrow$ 120 hours

Instructors

- Dr. Moritz Wolter, researcher HPCA Lab
- Dr. Elena Trunz, researcher Visual Computing Group

Course materials

- Lecture videos
- Lecture slides
- Exercise templates incl. TODOs and explanations
- Solutions to exercises

Applications of Machine Learning

Machine learning is everywhere

- Search
- Machine translation
- Virtual personal assistants
- Fraud detection
- Self driving cars
- Recomendations
- Photo editing

SIRI ALEXA GOOGLE CORTANA

Applications of machine learning

- Natural language processing
- Computer vision
- Information retrieval
- Medical diagnosis
- Speech recognition
- Many more...

ChatGPT

- Generative Pre-trained Transformer is used to answer questions posed in a chat-like manner

Can you please generate a short description of yourself that I could put in my latex beamer presentation?

Certainly! Here's a short description of me that you can use in your LaTeX Beamer presentation:

sql Copy code

```
\begin{frame}
  \frametitle{Meet ChatGPT}
  \begin{itemize}
    \item ChatGPT is an AI language model developed by OpenAI.
    \item Powered by GPT-3.5 architecture.
    \item A versatile tool for natural language understanding and generation
    \item A knowledge repository up to September 2021.
    \item Here to assist with a wide range of tasks and questions!
  \end{itemize}
\end{frame}
```

You can incorporate this slide into your Beamer presentation to introduce me to your audience.

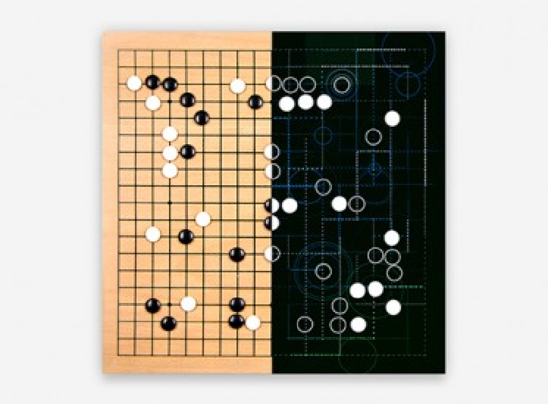
Meet ChatGPT

- ChatGPT is an AI language model developed by OpenAI.
- Powered by GPT-3.5 architecture.
- A versatile tool for natural language understanding and generation.
- A knowledge repository up to September 2021.
- Here to assist with a wide range of tasks and questions!

Stylization

Gatys et al. "A neural algorithm of artistic style", 2015. Photo in A. by Andreas Praefcke.

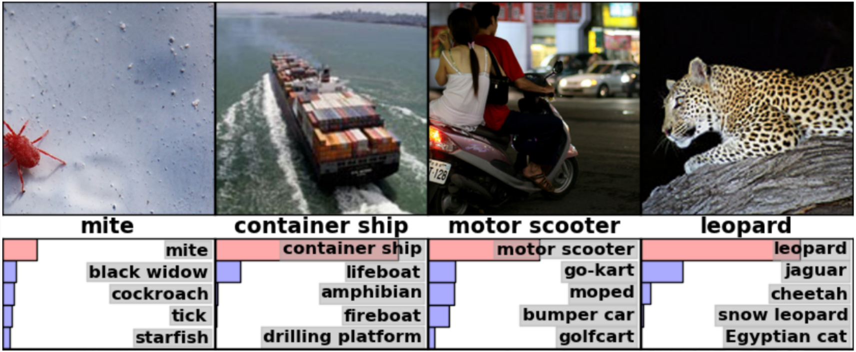
Go



Google's AI masters the game of Go a decade earlier than expected

Article by Will Knight.

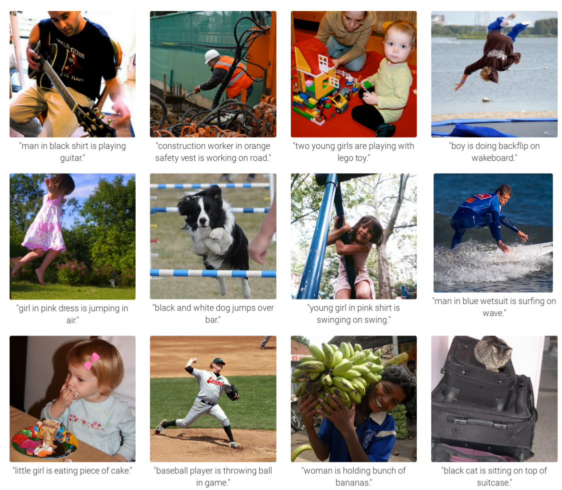
Image classification



Classification with Deep Convolutional Neural Networks

Krizhevsky et al. "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.

Image captioning



RNN for generating image descriptions

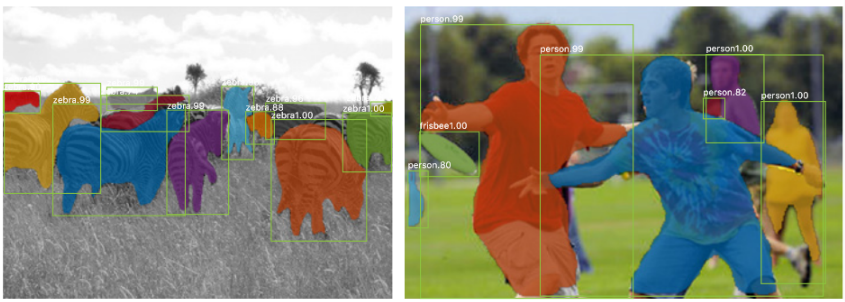
Karpythy and Fei-Fei "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015.

Visual Q&A

Visual question answering

Goyal et al. "Making the V in VQA Matter", CVPR 2017.

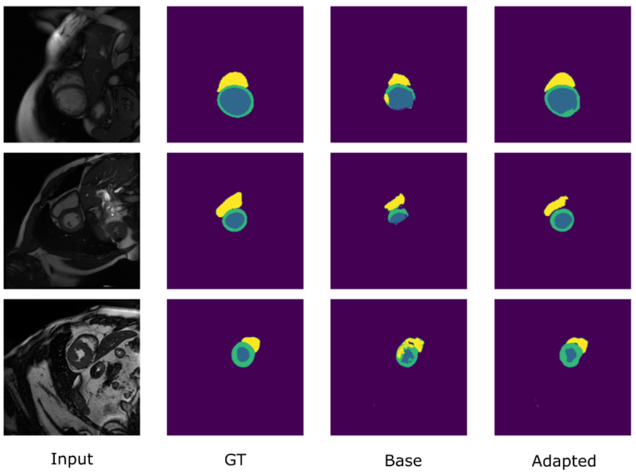
Object detection and image segmentation



Semantic segmentation of images

He et al. "Mask R-CNN", ICCV 2017.

Medical image segmentation



Segmentation of cardiac MR scans

Sheikh and Schultz "Unsupervised Domain Adaptation for Medical Image Segmentation via Self-Training of Early Features", MIDL 2022.

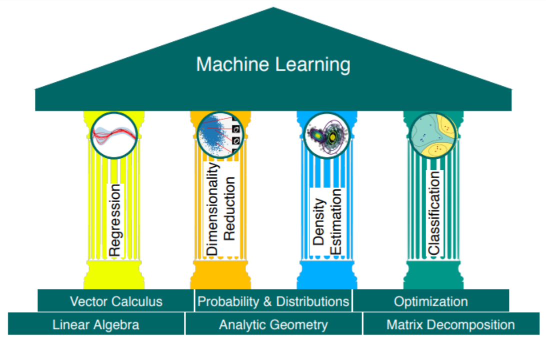
Synthesis

GANs can generate new faces

Karras et al. "A Style-Based Generator Architecture for Generative Adversarial Networks", CVPR 2019.

Course Contents

Outline



1st part: Mathematical foundations

2nd part: Foundations of machine learning

3th part: Deep learning

Image was taken from [DFO20]

1st part: Mathematical foundations

Day 01: Introduction

Day 02: Calculus – Optimization

Day 03: Linear algebra – Matrix decomposition

Day 04: Statistics – Probability theory

2nd part: Foundations of machine learning

Day 05: Machine learning basics

- Exercise: k-nearest neighbors for classification and regression

Day 06: Support vector machines

- Exercise: SVMs for classification and time-series prediction

Day 07: Decision trees and random forests

- Exercise: Performance of DT vs RF

Day 08: Clustering and density estimation

- Exercise: Clustering and compression using k-means

Day 09: Dimensionality reduction

- Exercise: Principal Component Analysis

3th part: Deep learning

Day 10: Introduction to neural networks

Exercise: Image classification

Day 11: Convolutional neural networks

Exercise: Image classification with CNN

Day 12: Initialization, Optimization and Regularization

Exercise: Decoding brain waves

Day 13: Segmentation

Exercise: Medical image segmentation

Day 14: Explainable machine learning

Exercise: Explainable machine learning and deepfake analysis

Day 15: Sequence processing

Exercise: Track 1 – Recurrent text generation

Exercise: Track 2 – Pre-trained transformer text generation

Literature

- M. Deisenroth, A. Faisal and C. Ong *Mathematics for Machine Learning.* Cambridge University Press, 2020 [link]
- I. Goodfellow, Y. Bengio and A. Courville *Deep Learning.* MIT Press, 2016 [link]
- M. Mohri A. Rostamizadeh and A. Talwalkar *Foundations of Machine Learning.* MIT Press, 2018 [link]
- C. Bishop *Pattern Recognition and Machine Learning.* Springer, 2006
- T. Hastie and R. Tibshirani. *The Elements of Statistical Learning.* Springer, 2009

Some Machine Learning Basics

What is machine learning?

Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed.

$\rightarrow$ Making predictions based on data

What do we mean by learning?

"A computer program is said to learn from *experience* E with respect to some class of *tasks* T and *performance measure* P, if its performance at tasks in T, as measured by P, improves with experience E." (Mitchell, 1997[1])

[1] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill, New York.

Experience E

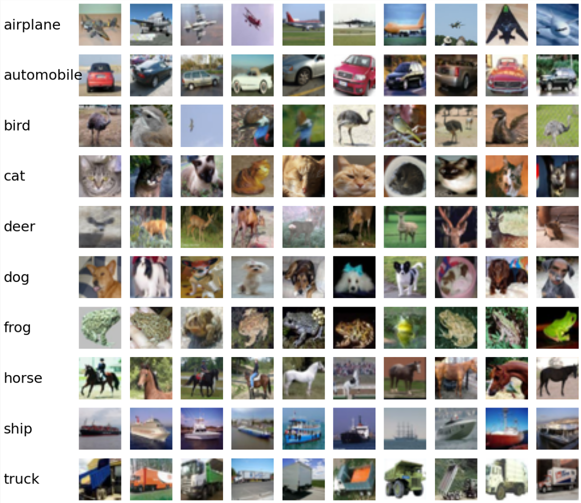
The experience E is the information that the algorithm can use during learning:

- *dataset*
- aka (training) data

Dataset contains *examples* (*data points*): collection of *features* that have been quantitatively measured from some object or event

- m-dimensional data point: $\mathbf{x} \in \mathbb{R}^m$
- each entry x_i is another feature

Example dataset: CIFAR10



10 classes

50,000 training images

10,000 testing images

Alex Krizhevsky "Learning Multiple Layers of Features from Tiny Images", 2009.

Supervised vs. unsupervised learning

Supervised learning:

- Each data point $\mathbf{x}$ comes with an associated *label* or *target* value $\mathbf{y}$
- The algorithm is learned by trying to match the targets, or predict $\mathbf{y}$ from $\mathbf{x}$

Unsupervised learning:

- Only data points $\mathbf{x}$, no *labels*
- The goal is to uncover the inherent structure within the data

Machine learning tasks

Tasks specify how a machine learning system should process *examples*. Common tasks include

- **classification**: determine category of input i.e. $f : \mathbb{R}^m \to \{1, ..., K\}$
- **regression**: predict numerical value(s) for some input, i.e. $f : \mathbb{R}^m \to \mathbb{R}^p$
- **density estimation**: learn probability density (if $\mathbf{x}$ is continuous) or probability mass (if $\mathbf{x}$ is discrete) function $p_{\text{model}} : \mathbb{R}^m \to \mathbb{R}$ on the space that the examples were drawn from
- **dimensionality reduction**: find a compact, lower-dimensional representation of high-dimensional data $\mathbf{x} \in \mathbb{R}^D$, which is often easier to analyze than the original data

Performance measure P

To evaluate a machine learning algorithm, we must design a quantitative measure of its performance.

Performance measure is task-spesific:

- Classification:
  - *Accuracy*: proportion of examples for which the model produces the correct output
  - *Error rate*: proportion of examples for which the model produces an incorrect output
- Density estimation:
  - Average log-probability the model assigns to examples

Generalization

We want to evaluate performance of our system on data that it has *not yet seen*.

This estimates how well the system will *generalize*, in comparison to performance on already seen data used during the learning process.

- Unseen data makes up our *test set* or test data
- Already seen data is the *training set* or training data

During training we want to reduce the *training error* – this is simply an optimization problem.

What distinguishes learning from optimization is the interest in also reducing the *test error* (*generalization error*).

Image Classification

Classification task

Given:

- Input examples
- Corresponding labels

Find: Classifier that predicts a class from the input.

Example: given an x-ray image, predict if the person is sick.

Image classification

$\longrightarrow$ cat

Semantic gap

What we see

```
[[206  84 145   1  24  61 124  70 186 103  82 114 255 203 169]
 [147 250 234  12 245  90  72 193  21 100 181 247 161 240 110]
 [ 91  68  20 176 107 176 193 254 245 109 186 232 184 246 145]
 [  6  72 199 103  85  64   7 189 254  75  52 137 190 223 240]
 [ 64  98 182  71 255  36 212  19 185  60  66 148 105 177 198]
 [ 32 214  18  81  66 134 229 238  34  43  33  59  67 229  88]
 [ 75 231  24  73 167 252 151  14  38 166 137 150  71  29   4]
 [150 121  24 175 255  69 104 105 240 182 104 194  58 137  82]
 [233  74 152 138  83 142  58 150  24 110 109  75 224 236 221]
 [113  70 194 104  37 181 117  97 240  43 197 109 221 119 142]
 [ 68  51 170 173  25  68 154 152  46 106  72  54 115  89  46]
 [222 101 166 211 102 238 165  54  51 252   4  94 128 161   0]
 [ 75 180 254 150 144  53 197 152  66 108  57  32 245 190  93]
 [ 11  32 193 251 225 199 147 105  52  91  32 207  47  80 112]
 [115  20 114  74 212  57 134  92 195 193 167 177 241 131 153]
 [ 49 117 252  34 102 116 222  83 205 119  38 194 253 164  71]
 [ 59 159 178  52   5 148 225   1 136 184  75  22  43  39 107]
 [142   8  38 104 184 249 186 133 190 229 197 119 144 105 235]
 [ 51 157  72  11 215 159  79  11 129  59  23 125  21   7 207]
 [ 18 223  10  32 153 137   6 201  16   7 155 136   9  26 197]
 [184  63 124 159 173  99  83   5  36  36 186 105  38  21  34]
 [189 106  15  64 146 215  87  17  68 148 130  99 202 210 252]
 [149 144 228 159  65  63 164 179  78 199  92 135  91 231 242]
 [141  96 169  31  18  59 243 190  12 154 227  43 128  98 133]
 [205  80 149 171 116  87  62 231 142 193  95  56 239 145  67]
 [ 31 106 144 102 235 215 157 225 169 196  64 219 123  50  58]
 [180  44 126  18  94  64  35 234 196  95 107  87  95  97   7]
 [ 84  84 247 104 188 106  99 140  58  40 171  92 111  77 135]
 [220 163 200 177 156 226  95  21 152 176   2   1  96  90  12]
 [ 67 101 141  37 163 159 212 212  48 220  65 116 188  89  34]]
```

What the computer sees

- An image is just a grid of numbers between 0 and 255
- E.g. this RGB image: $300 \times 400 \times 3$

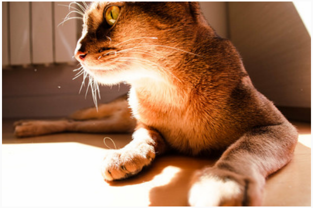
Challenges: deformation

This, this, this and this images are CC0 1.0 public domain.

Challenges: occlusion



Challenges: illumination

Challenges: background clutter

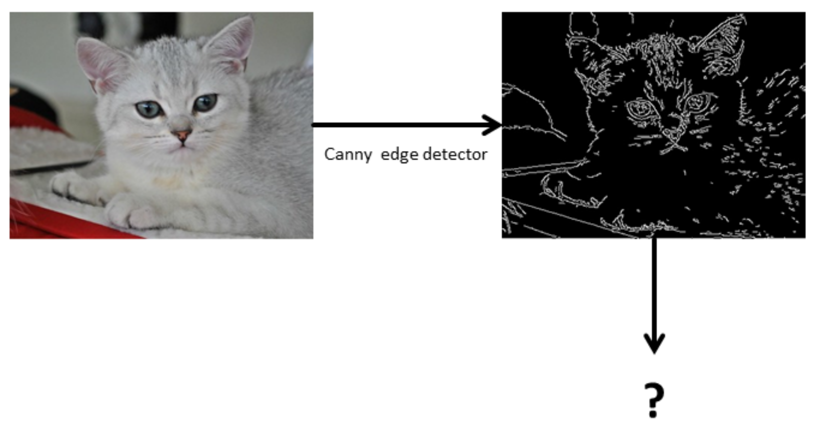
Challenges: intraclass variation

A rule-based image classifier

```
def classify_image():
    # do something
    return class_label
```

- There are many rule-based algorithms for e.g. sorting a list of numbers
- No obvious way to hard-code the algorithm for recognizing a cat or other classes

Attempts have been made



Data-driven classification

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

```
def train(train_images, train_labels):
    # machine learning
    return model

def predict(model, test_images):
    # use model to predict labels
    return test_labels
```

Nearest Neighbor Classification

First classifier: Nearest Neighbor

```
def train(train_images, train_labels):
    # memorize all data and labels
    return model

def predict(model, test_images):
    # predict label of most similar
    # training example
    return test_labels
```

What is similarity?

- The real meaning of similarity is a philosophical question
- We will take a more pragmatic approach

Definition: Let U be the universe of possible objects. The *distance* (*dissimilarity, metric*) on U is a function $d : U \times U \to \mathbb{R}$ that satisfies certain conditions (axioms).

Intuition behind axioms of a distance function

$$d(a, b) = 0 \Leftrightarrow a = b \qquad \textit{Identity of indiscernibles}$$

Otherwise there are objects that differ from each other, but we can't distinguish them.

$$d(a, b) = d(b, a) \qquad \textit{Symmetry}$$

Otherwise we could claim "Mary looks like Jenny, but Jenny bears no resemblance to Mary."

$$d(a, c) \leq d(a, b) + d(b, c) \qquad \textit{Triangle inequality}$$

Otherwise we could claim "Mary looks like Jenny and Sarah, yet Jenny bears no resemblance to Sarah."

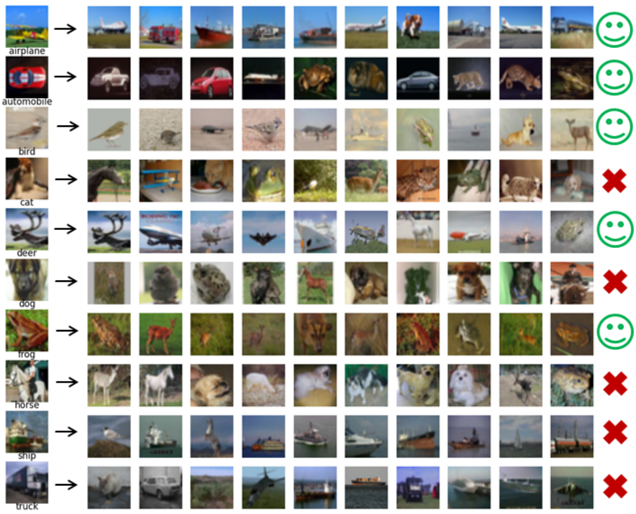
L1 (Manhattan) distance

One common similarity measure is the L1 distance:

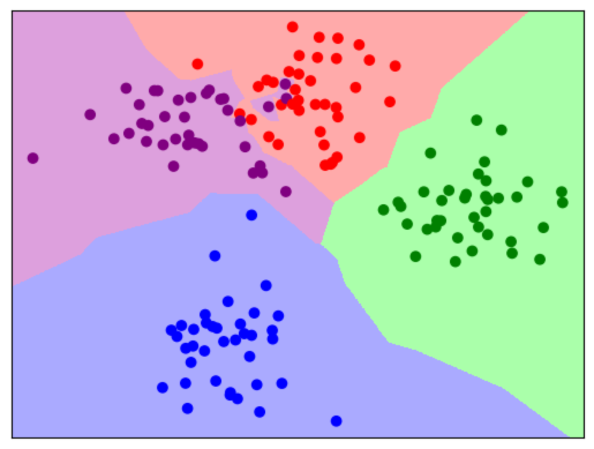
$$d_1(a, b) = \sum_{i=1}^{m} |a_i - b_i|$$

$$\left| \overset{\text{image a}}{\begin{array}{|c|c|c|c|}\hline 4 & 56 & 13 & 21 \\ \hline 55 & 78 & 7 & 32 \\ \hline 96 & 10 & 33 & 12 \\ \hline 87 & 14 & 2 & 43 \\ \hline \end{array}} - \overset{\text{image b}}{\begin{array}{|c|c|c|c|}\hline 56 & 3 & 6 & 15 \\ \hline 71 & 42 & 8 & 11 \\ \hline 30 & 98 & 77 & 12 \\ \hline 40 & 82 & 15 & 22 \\ \hline \end{array}} \right| = \overset{\text{pixel-wise absolute value differences}}{\begin{array}{|c|c|c|c|}\hline 52 & 53 & 7 & 6 \\ \hline 16 & 36 & 1 & 21 \\ \hline 66 & 88 & 44 & 0 \\ \hline 47 & 68 & 13 & 21 \\ \hline \end{array}} \xrightarrow{\text{sum}} 539$$

NN results on CIFAR10

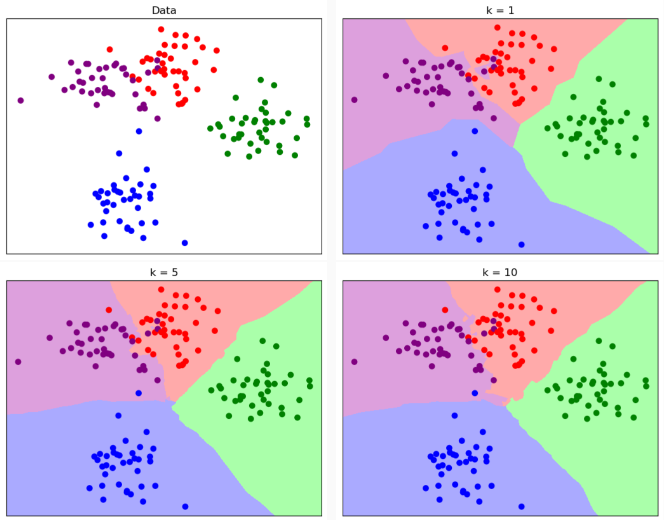


Decision regions of NN

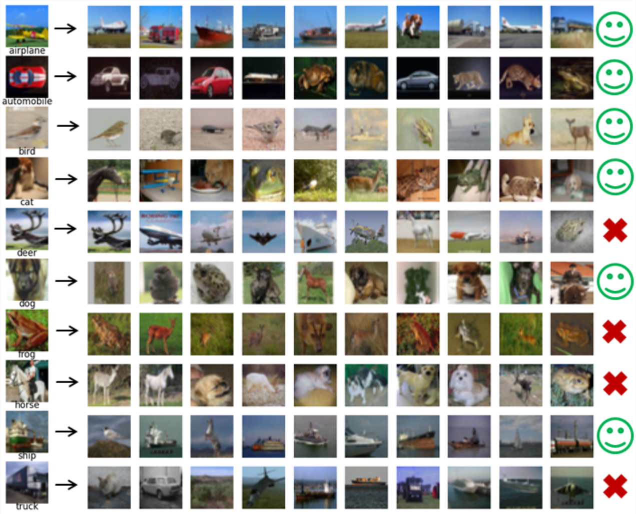


k-Nearest Neighbors classifier

Instead of copying label from nearest neighbor, take *majority vote* from k closest points:



10-NN results on CIFAR10



L2 (Euclidean) distance

Another common similarity measure is the *Euclidean distance*.

The Euclidean distance between two m-dimensional points is the length of a line segment between these points

$$d_2(a, b) = \sqrt{\sum_{i=1}^{m} (a_i - b_i)^2}.$$

Choosing k and distance metric

What is the best distance?

What is the best value of k?

Very problem-dependent!

Try them all out and see what works best

References

[DFO20] Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. https://mml-book.com. Cambridge University Press, 2020. DOI: 10.1017/9781108679930.