EECS4404
Introduction to Machine Learning and Pattern Recognition
Lecture 1

Amir Ashouri

York University

September 9, 2019

Contents are largely based on the slides of Prof. Ruth Urner

Organization of the class

Basic Information

- Instructor: Amir Ashouri (aashouri@eecs.yorku.ca)
- Office Hours: Mondays (Time To be announced on the website)
- Website:
`https://wiki.eecs.yorku.ca/course_archive/2019-20/F/4404E`
- TA: Mahmoud Afifi (mafifi@eecs.yorku.ca)
Main point of contact for assignments

Lectures

Lecture times	Mon, Wed, 14:30 - 16:00
Place	CB 129, CC 211
First lecture	September 4
Reading week	October 12-18
Midterm	October 21
Last lecture	Monday, December 2
More info	https://registrar.yorku.ca/enrol/dates/fw19#2

Evaluation

EECS4404

- 30 % Assignments
- 30 % Midterm
- 40 % Final exam

EECS5327

- 15 % Assignments
- 25 % Midterm
- 35 % Final exam
- 25 % Project OR Paper presentation

Evaluation - Undergrads and Grads

Assignments:

- 3 Assignments
- Roughly due end of September/October/November
- Mix of theoretical and programming questions

Tests:

- In-class midterm (tentatively October 21)
- Final exam (December 5 - 20)

Evaluation Grads only
A mandatory project OR a presentation

Presentation

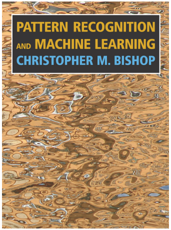
- Short presentation (10 minutes) on an ML research *(sub)topic*
- Send me an email until **Monday, October 7**, specifying your topic and some possible related papers
- Papers (3-4) to be selected/approved by **October 9** by the instructor (normally taken from a recent NeurIPS, CVPR, ICLR, and ICML conference)
- Students need to study those papers, survey them in 4 pages, and present the ideas in a presentation (12-15 slides).
  Survey due **Nov 25**, **presentations on Nov 28 and 30**

Project

- Send me an email until **Monday, October 7**, specifying your research area, project, and possibly ideas for course project
- We set a meeting on **Wednesday, October 9**, to discuss you project. Project delivery deadline is **Nov 29**

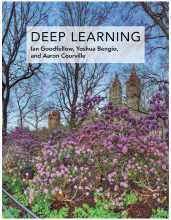
Pattern Recognition and Machine Learning (PRML)
by Christopher M.Bishop. Springer. (2006).

- Available on Amazon.
- Available in bookstore



Deep Learning
by Ian Goodfellow, Yoshua Bengio and Aaron Courville.
www.deeplearningbook.org. (2016)

- Available on Amazon.
- Full content available online: `https://www.deeplearningbook.org/contents/TOC.html`



Material

Lectures will include a mixture of high level motivation and explanations and low-level derivations

Material in readings and lectures will overlap but won't be identical. **You are responsible for both!**

Slides used in class will generally be posted to the course website the day within 24 hours after lecture.

In cases where lectures are delivered on the board, these notes may not be posted. **Don't skip lectures and plan to catch up by looking at the posted notes!**

Programming

Assignments to be in done in **Matlab** and **TensorFlow**:

- We use Python/TensorFlow for our Deep Learning assignment
- We use Matlab for the other two assignments

- Available on EECS lab machines and can be installed on your own computers

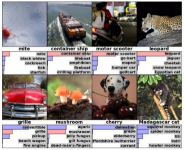
- If you are not familiar with it, the bonus is that you are going need to learn it!

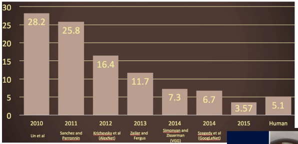
https://www.tensorflow.org/tutorials
https://www.mathworks.com/support/learn-with-matlab-tutorials.html

ImageNet Challenge

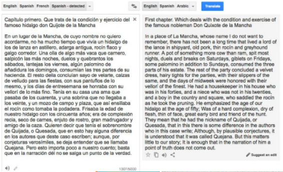
IMAGENET

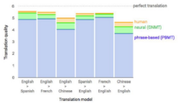
- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.





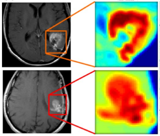
- Natural language processing

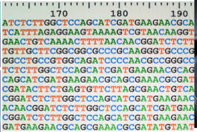


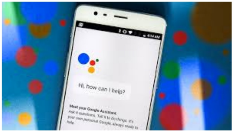


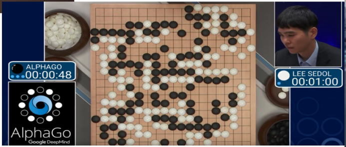
Google's Neural Machine Translation (Wu et al, 2016)

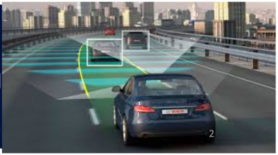
DIAGNOSTICS



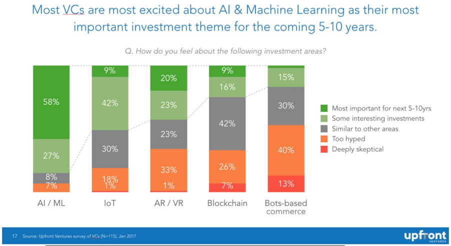


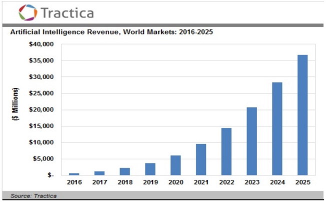


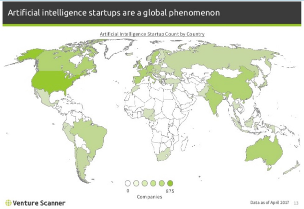


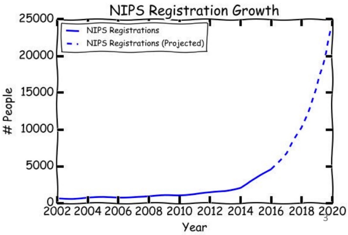


ML Growth


Most VCs are most excited about AI & Machine Learning as their most important investment theme for the coming 5-10 years.
Q. How do you feel about the following investment areas?
AI / ML: 58%, 27%, 8%, 7%
IoT: 9%, 42%, 30%, 18%, 1%
AR / VR: 20%, 23%, 23%, 33%, 1%
Blockchain: 9%, 16%, 42%, 26%, 7%
Bots-based commerce: 15%, 30%, 40%, 13%
Most important for next 5-10yrs
Some interesting investments
Similar to other areas
Too hyped
Deeply skeptical
17 Source: Upfront Ventures survey of VCs (N=115), Jan 2017
upfront

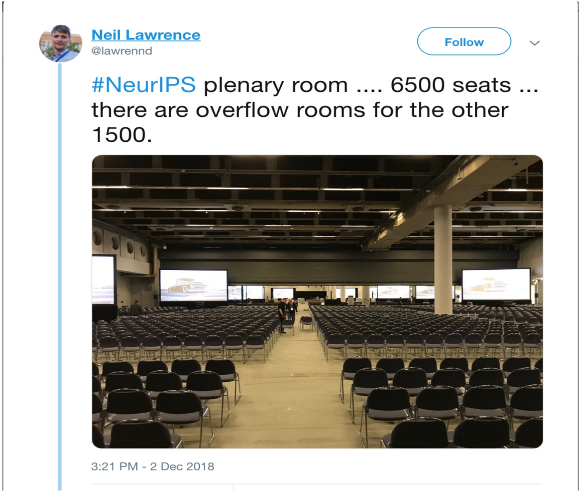

Tractica
Artificial Intelligence Revenue, World Markets: 2016-2025
($ Millions)
$40,000
$35,000
$30,000
$25,000
$20,000
$15,000
$10,000
$5,000
$-
2016 2017 2018 2019 2020 2021 2022 2023 2024 2025
Source: Tractica


Artificial intelligence startups are a global phenomenon
Artificial Intelligence Startup Count by Country
0
875
Companies
Venture Scanner
Data as of April 2017 13

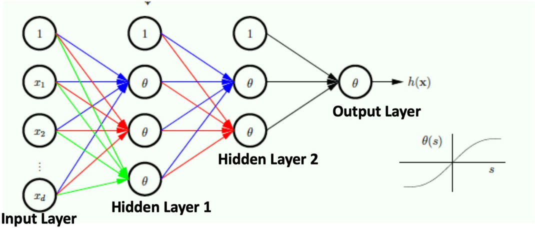

NIPS Registration Growth
NIPS Registrations
NIPS Registrations (Projected)
People
25000
20000
15000
10000
5000
0
2002 2004 2006 2008 2010 2012 2014 2016 2018 2020
Year

NIPS (NeurIPS) Attendance!

NIPS (NeurIPS)
2018 - Montreal



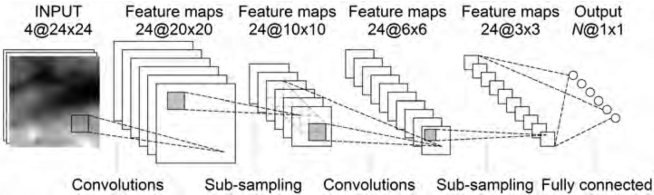
Neural Networks - Architecture



$$h_i^{(1)} = \theta\left(\sum_{j=0}^{d} w_{ji}^{(1)} \cdot x_j\right) \qquad h_i^{(2)} = \theta\left(\sum_{j=0}^{d} w_{ji}^{(2)} \cdot h_j^{(1)}\right) \qquad y = sign\left(\sum_{j=0}^{d} w_{ji}^{(3)} \cdot h_j^{(2)}\right)$$

Output `i' (Hidden Layer 1) Weights:Layer 1 Inputs Output `i' (Hidden Layer 2) Weights: Layer 2 Output

Convolutional Neural Networks



- Image statistics are translation invariant (objects and viewpoint translates)
- Expect low-level features to be local (e.g. edge detector)
- Expect high-level features learned to be coarser

Tensor flow Assignments

- **Python based** ML Library released by Google in 2015
- Automatic Training for Neural Networks
- GPU Support (Not Required for Assignments in this courses)

- Installation through Anaconda Environment is Recommended (See Installation Guide on Course Webpage)

- Tons of Resources!
  - Tensorflow.Org Tutorials
  - CS231n Stanford Tutorial (http://cs231n.stanford.edu/)
  - See Course Webpage for a simple tutorial (Updated, Use Chrome Browser)

Tensor flow Example (https://www.tensorflow.org)

```
import tensorflow as tf
x = tf.placeholder(tf.float32, [None, 784])
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
y = tf.nn.softmax(tf.matmul(x, W) + b)
```

Initialize Computational Graph

```
cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))

train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
```

Loss Function and Optimizer

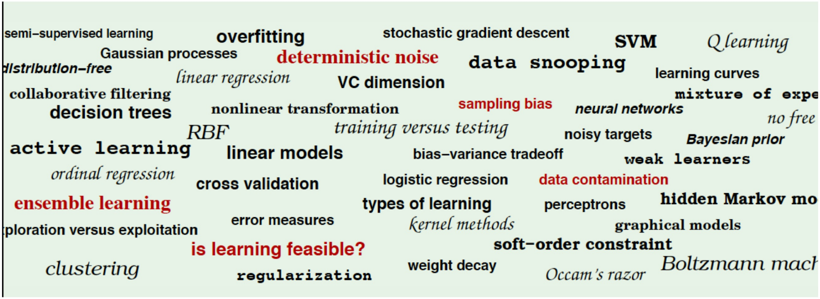
```
sess = tf.InteractiveSession()

tf.global_variables_initializer().run()

for _ in range(1000):
  batch_xs, batch_ys = mnist.train.next_batch(100)
  sess.run(train_step, feed_dict={x: batch_xs, y_: batch_ys})I
```

Training Routing

Bag of ML Jargons



5

Machine Learning examples

Machine Learning - examples

Self driving cars

Community detection

Fraud detection

Species preservation

Recommender systems

Computational Biology

Logistics

Consumer behavior analysis

Face recognition

Medical diagnosis

Speech recognition

Computer vision

Stock market prediction

Spam filters

Automated translation

Character recognition

Machine Learning

Machine Learning – Why do we need it?

Machine Learning – Why do we need it?

Some tasks are too complex to be implemented directly:

- Self driving cars
- Speech recognition
- Complex rules for classification tasks on high dimensional data
  - Fraud detection
  - Document classification

Machine Learning – Why do we need it?

Some tasks are too complex to be implemented directly:

- Self driving cars
- Speech recognition
- Complex rules for classification tasks on high dimensional data
  - Fraud detection
  - Document classification

$\rightarrow$ Learn a program based on data!

What is machine learning?

What is machine learning?

First explanation:

- Development of algorithms which allow a computer to “learn” specific tasks from training examples.
- Learning means that the computer should not just memorize the seen examples, but predict well on previously unseen instances
- Ideally, the computer should use the examples to extract a general “rule” how the specific task has to be performed correctly.

Representation of data

Data is represented as vectors of numerical features and a label, e.g. Disease or Not-Disease.

Patient	Age	Temperature	Blood-pressure	...	Disease
P1	45	37.2	80/120	...	Yes
P2	26	36.3	100/150	...	No
P3	74	36.7	70/130	...	Yes
...					
Pm	68	38.5	70/140	...	No

Representation of data

Data is represented as vectors of numerical features and a label, e.g. Disease or Not-Disease.

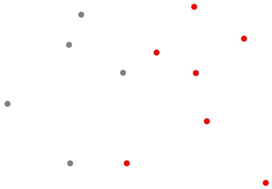
Patient	Age	Temperature	Blood-pressure	...	Disease
P1	45	37.2	80/120	...	Yes
P2	26	36.3	100/150	...	No
P3	74	36.7	70/130	...	Yes
...					
Pm	68	38.5	70/140	...	No

Goal: Predict the label of a new feature vector:

Patient	Age	Temperature	Blood-pressure	...	Disease
P	31	37.6	80/100	...	?

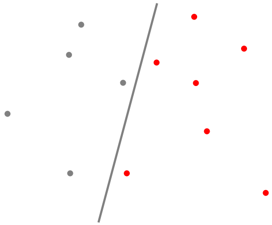
Training:

- Derive some rule based on data



Training:

- Derive some rule based on data

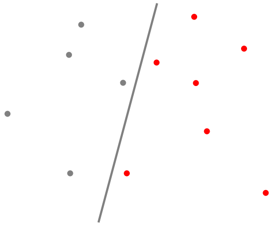


Training:

- Derive some rule based on data

Testing:

- The goal is that the learned predictor does well on new, unseen examples

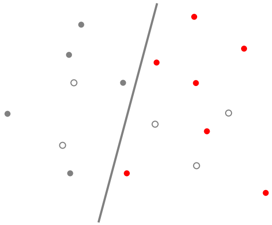


Training:

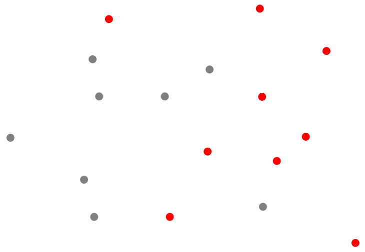
- Derive some rule based on data

Testing:

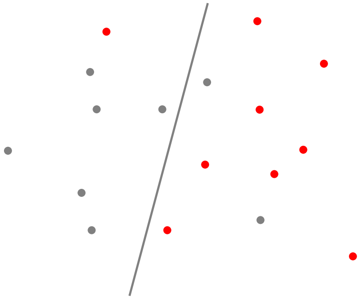
- The goal is that the learned predictor does well on new, unseen examples



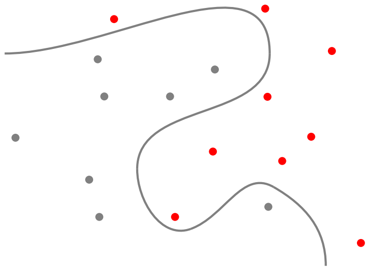
Finding a predictor based on data

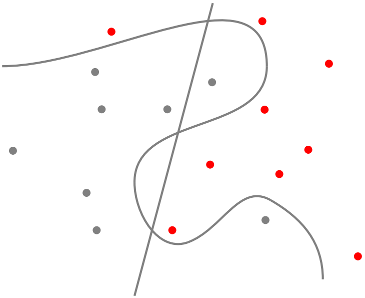


Finding a predictor based on data

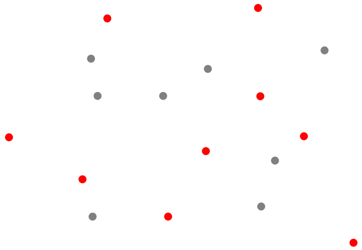


Finding a predictor based on data



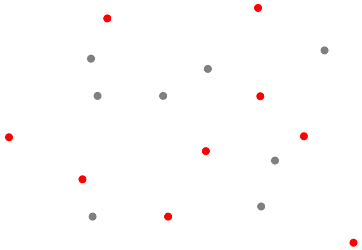


Which predictor is better?



What predictor would you now suggest?

Finding a predictor based on data



What predictor would you now suggest?

What if I told you every label is generated by a random coin flip?

Lesson 1:
Machine Learning can only work if "there is something we can learn".

- Output Y "has something to do" with input X "Similar inputs" lead to "similar outputs"
- There is a "simple relationship" or "simple rule" to generate the output for a given input

These assumptions are rarely made explicit, but something along this line has to be satisfied, otherwise ML is doomed.

Lesson 2:
We need to have an idea what we are looking for.

This is referred to as the **inductive bias** or **prior knowledge**.

Learning is impossible without inductive bias/prior knowledge.

Lessons learned

Lesson 2:
We need to have an idea what we are looking for.

This is referred to as the **inductive bias** or **prior knowledge**.

Learning is impossible without inductive bias/prior knowledge.

Let's try to get some intuition for what this means.

Predicting tastiness of Papayas

Task:
Use machine learning to generate a program that predicts whether a **papaya** is **tasty** or **not tasty**.

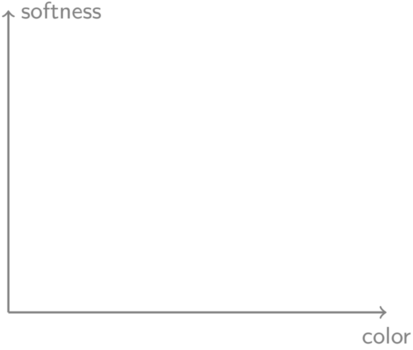
Step 1:
Choose a feature representation
Decide how to represent a papaya to a computer; that is, choose features to measure

Predicting tastiness of Papayas

Step 1:
Choose a feature representation
Decide how to represent a papaya to a computer; that is, choose features to measure

Prior knowledge:
Color and *Softness* can tell us whether a papaya is tasty

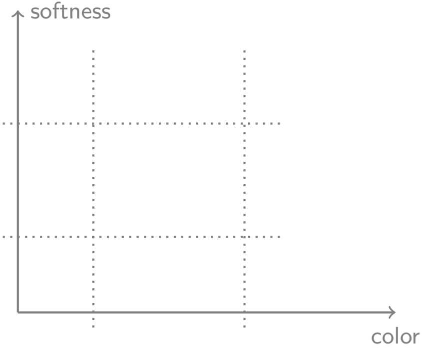
Step 2:
Choose a class of predictors
It is very important that this is done before looking at the actual data!



Predicting tastiness of Papayas

Step 2:
Choose a class of predictors
It is very important that this is done before looking at the actual data!

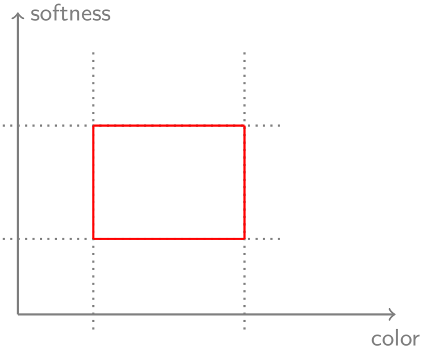
Prior knowledge:
There is a range in *color* and a range in *softness* where papayas are tasty



Step 2:
Choose a class of predictors
It is very important that this is done before looking at the actual data!

Prior knowledge:
There is a range in *color* and a range in *softness* where papayas are tasty

A rectangle in the two dimensional feature space may be a suitable predictor.



Predicting tastiness of Papayas

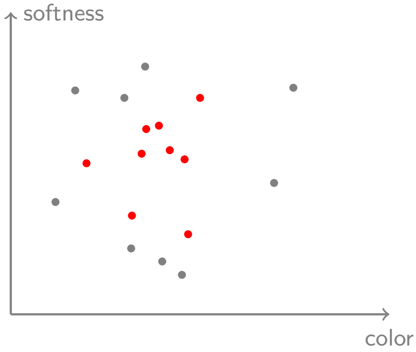
Step 3: Collect data
For every papaya you come across measure:

- feature 1: color
- feature 2: softness
- label: tasty or not tasty

Predicting tastiness of Papayas

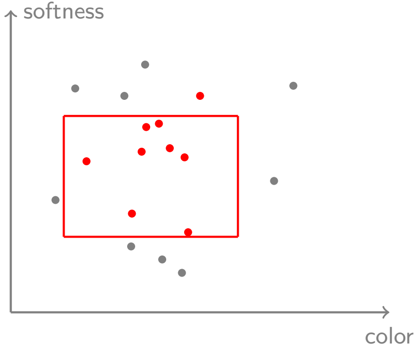
Step 3: Collect data
For every papaya you come across measure:

- feature 1: color
- feature 2: softness
- label: tasty or not tasty



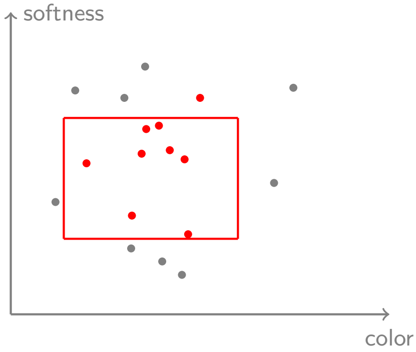
Step 4: Fit a model
Find a rectangle that fits your data best

Step 4: Fit a model
Find a rectangle that fits your data best



Step 4: Fit a model
Find a rectangle that fits your data best

Step 5: Predict
Use this rectangle to predict for future Papayas whether they are tasty or not.



Summary

Step 1 Choose feature representation
Feature space $X \subseteq \mathbb{R}^d$
Label space $Y = \{-1, 1\}$

Step 2 Choose class of predictors
Class H of hypotheses $h : X \to Y$

Step 3 Collect data
$(x_1, y_1), \ldots (x_n, y_n)$

Step 4 Fit a model
Choose a predictor h from the class that has minimal empirical error
This is called *Empirical Risk Minimization*

Step 5 Predict
$y = h(x)$

Inductive bias in animal learning

Any "system" that learns has an inductive bias.

Experiments on learning in rats:

Rats get two choices of water.
One choice makes them sick, the other one doesn't.

Any "system" that learns has an inductive bias.

Experiments on learning in rats:

Rats get two choices of water.
One choice makes them sick, the other one doesn't.

Experiment 1:

- Two types of water taste differently (neutral and sugar).

Inductive bias in animal learning

Any "system" that learns has an inductive bias.

Experiments on learning in rats:

Rats get two choices of water.
One choice makes them sick, the other one doesn't.

Experiment 1:

- Two types of water taste differently (neutral and sugar).
- Rats learn very fast not to drink the water that makes them sick.

Experiment 2:

- Same water, but one type of water is presented together with "audio-visual stimuli" (certain sounds and light conditions), while the other type of water is presented without these accompanying audio-visual stimuli.

Experiment 2:

- Same water, but one type of water is presented together with "audio-visual stimuli" (certain sounds and light conditions), while the other type of water is presented without these accompanying audio-visual stimuli.
- In this setting, rats did NOT learn to avoid the water that makes them sick.

Experiment 2:

- Same water, but one type of water is presented together with "audio-visual stimuli" (certain sounds and light conditions), while the other type of water is presented without these accompanying audio-visual stimuli.
- In this setting, rats did NOT learn to avoid the water that makes them sick.
- Apparently, they cannot make a connection between "sound of the food" and "sickness".

Inductive bias in animal learning

Would it be useful for rats to become "super-rats"?

The super-rats would be able to adapt their eating behavior also to sound and light.

Would such super-rats learn better to stay healthy?

No... The super-rats would pay attention to too many details for each situation, where they eat.

They would **not be able to generalize from their experiences to a useful prediction rule**.

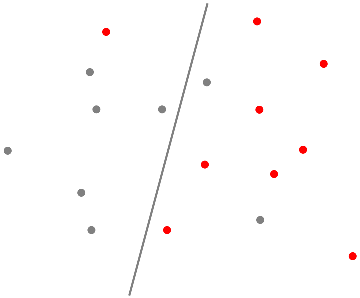
Inductive bias in animal learning

No... The super-rats would pay attention to too many details for each situation, where they eat.

They would **not be able to generalize from their experiences to a useful prediction rule**.

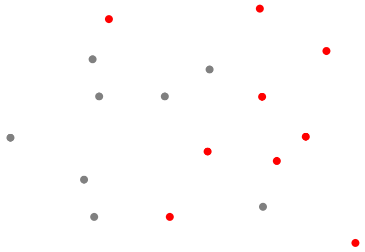
This phenomenon is called overfitting.

Finding a predictor based on data

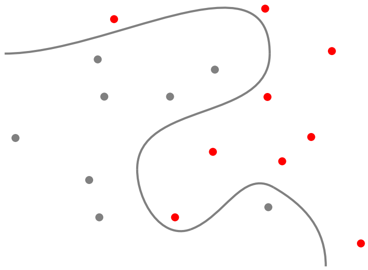


Which predictor is better?

Finding a predictor based on data



Which predictor is better?

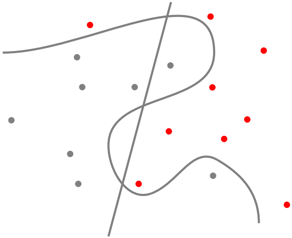


Which predictor is better?

Which predictor is better?

High level principle:

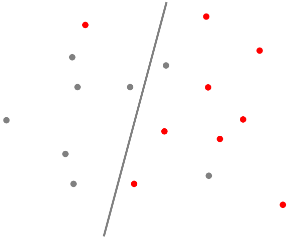
- Choose a simple predictor that does (reasonably) well on the data



Which predictor is better?

High level principle:

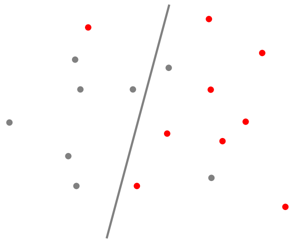
- Choose a simple predictor that does (reasonably) well on the data



Which predictor is better?

High level principle:

- Choose a simple predictor that does (reasonably) well on the data



$\rightarrow$ It can be shown that (a formal version of) this principle can yield guarantees that our predictions generalize from training data points to future data!

Machine Learning as Inductive Inference

WHO KNOWS WHAT INDUCTION AND DEDUCTION MEAN?

Deductive inference is the process of reasoning from one or more general statements (premises) to reach a logically certain conclusion.

Example:

- Premise 1: every person in this room is a student.
- Premise 2: every student is older than 10 years.
- Conclusion: every person in this room is older than 10 years.

If the premises are correct, then all conclusions are correct as well.

Nice in theory. For example, mathematics is based on this principle. But no natural way to deal with uncertainty regarding the premises.

Inductive inference: reasoning that constructs or evaluates general propositions that are derived from specific examples.

Example:

- We throw lots of things, very often.
- In all our experiments, the things fell down and not up.
- So we conclude that likely, things always fall down.

However: we can never be sure; our conclusion can be wrong!

Humans do inductive reasoning all the time: draw uncertain conclusions from our relatively limited experiences.

Example:
• You come 10 minutes late to every lecture I give.
• The first 7 times I don't complain.
• You conclude that I don't care and it won't have any consequences.

Humans do inductive reasoning all the time: draw uncertain conclusions from our relatively limited experiences.

Example:
• You come 10 minutes late to every lecture I give.
• The first 7 times I don't complain.
• You conclude that I don't care and it won't have any consequences.

BUT you cannot be sure ...

Second, more abstract description of what machine learning is:

Machine learning aims at automating the process of inductive inference.

Very important observation:

- Being able to predict is often easier than understanding the underlying mechanism.
- In particular, it is often not necessary to understand the underlying mechanism for being able to make good predictions!

Very important observation:

- Being able to predict is often easier than understanding the underlying mechanism.
- In particular, it is often not necessary to understand the underlying mechanism for being able to make good predictions!

Example:

- We observe that the sun rises every day.
- So we predict that the sun is also going to rise the next day.
- To make this prediction, we don't need to understand why this is the case.

In most applications of machine learning: focus is not so much on building models ("understanding the underlying process") but more on predicting.

ML in this course

Machine Learning

Generative Models

PCA

Supervised Learning

Decision trees

Unsupervised Learning

Dimensionality Reduction

Deep Learning

Transfer Learning

Support Vector Machines

Regression

Causality

Reinforcement Learning

Clustering

Classification

Active Learning

Plan for this course

1. **Supervised Learning: linear methods and Bayesian Models**
   - for **regression**
   - for **classification**
2. **Kernel methods: SVMs**
3. **Model selection, cross validation**
4. **Neural Networks**
   - **Convolutional Neural Networks**
5. **Unsupervised Learning**
   - **Clustering**
   - **Dimensionality reduction, PCA**