Seminar in Communication Networks

Learning, Reasoning and Control

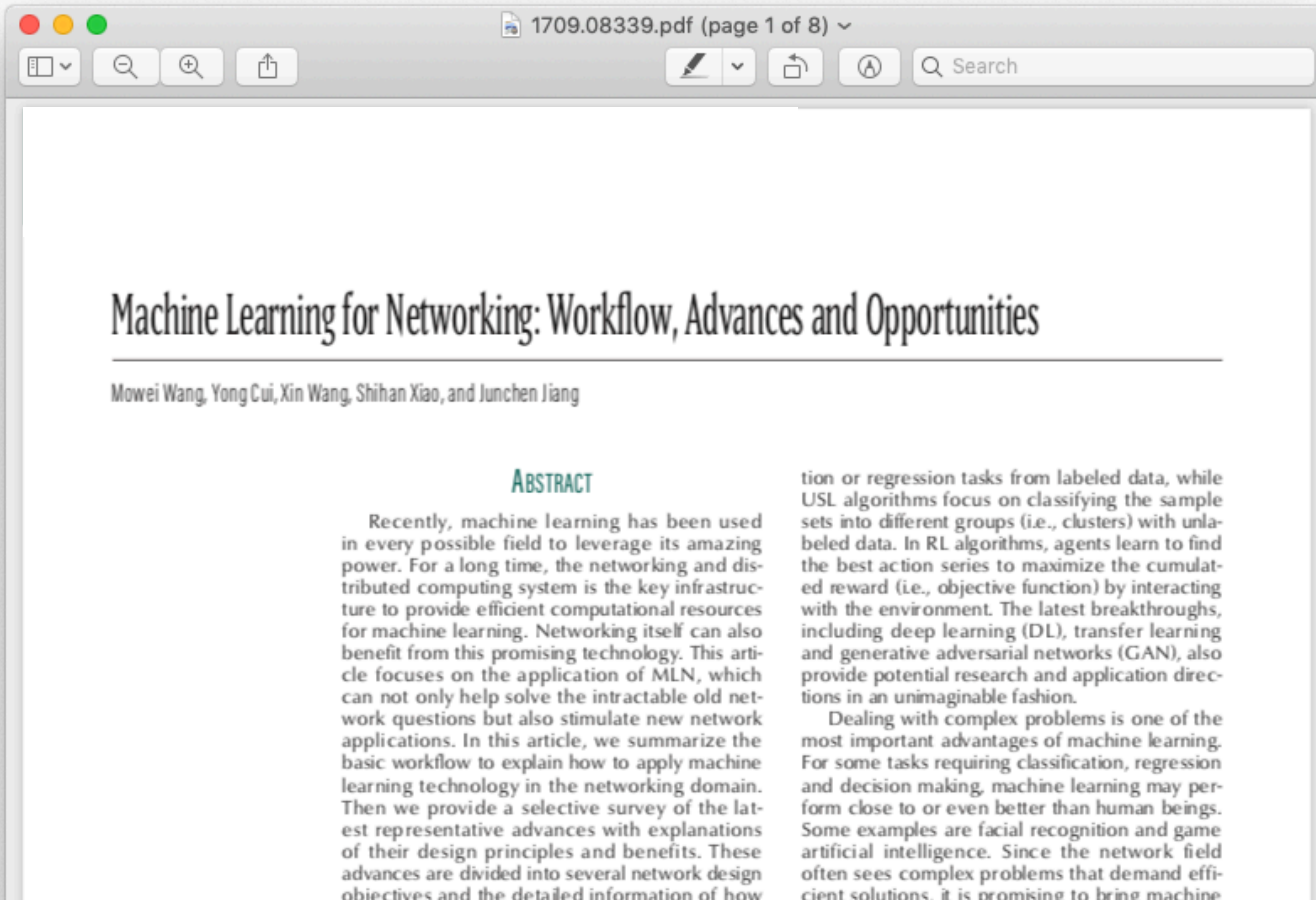
Prof. Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

25 September 2019

Machine Learning for Networking:
Workflow, Advances and Opportunities



Machine Learning for Networking: Workflow, Advances and Opportunities

Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang

Abstract

Recently, machine learning has been used in every possible field to leverage its amazing power. For a long time, the networking and distributed computing system is the key infrastructure to provide efficient computational resources for machine learning. Networking itself can also benefit from this promising technology. This article focuses on the application of MLN, which can not only help solve the intractable old network questions but also stimulate new network applications. In this article, we summarize the basic workflow to explain how to apply machine learning technology in the networking domain. Then we provide a selective survey of the latest representative advances with explanations of their design principles and benefits. These advances are divided into several network design objectives and the detailed information of how

tion or regression tasks from labeled data, while USL algorithms focus on classifying the sample sets into different groups (i.e., clusters) with unlabeled data. In RL algorithms, agents learn to find the best action series to maximize the cumulated reward (i.e., objective function) by interacting with the environment. The latest breakthroughs, including deep learning (DL), transfer learning and generative adversarial networks (GAN), also provide potential research and application directions in an unimaginable fashion.

Dealing with complex problems is one of the most important advantages of machine learning. For some tasks requiring classification, regression and decision making, machine learning may perform close to or even better than human beings. Some examples are facial recognition and game artificial intelligence. Since the network field often sees complex problems that demand efficient solutions, it is promising to bring machine

Overview

workflow	steps to use ML in networking from problem formulation to deployment
survey	summarize latest works ▪ design principles ▪ benefits
opportunities	solve intractable old questions stimulate new applications

Developing algorithms and systems to deal with complex problems is a challenging task

Networks are **hard to**

- manage
- optimize
- secure

and not necessarily improving

ML is good in that

Recently ML techniques have made breakthroughs in:

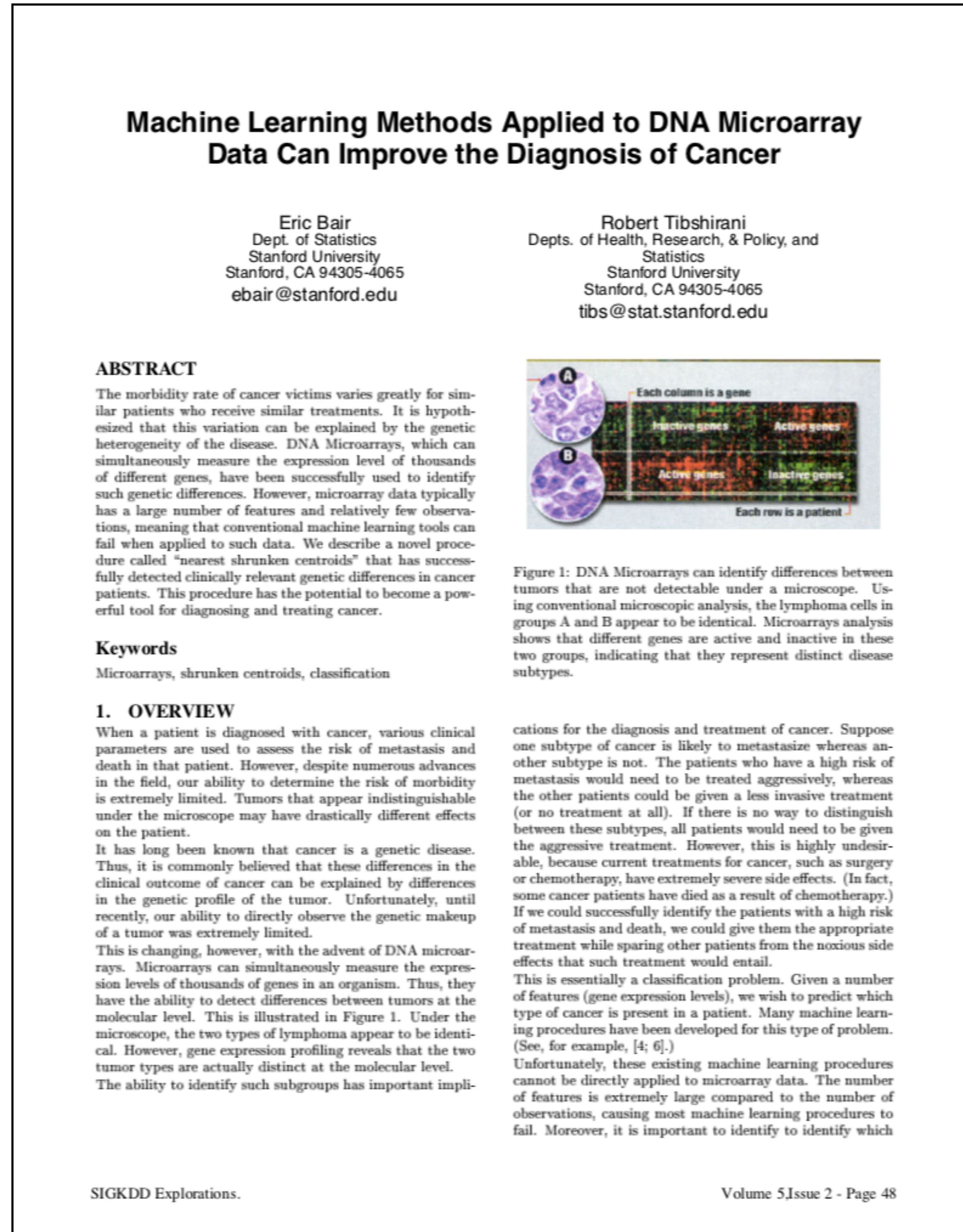
bioinformatics

speech recognition

computer vision

Recently ML techniques have made breakthroughs in:

- bioinformatics
- speech recognition
- computer vision



Machine Learning Methods Applied to DNA Microarray Data Can Improve the Diagnosis of Cancer

Eric Bair
Dept. of Statistics
Stanford University
Stanford, CA 94305-4065
ebair@stanford.edu

Robert Tibshirani
Depts. of Health, Research, & Policy, and Statistics
Stanford University
Stanford, CA 94305-4065
tibs@stat.stanford.edu

ABSTRACT

The morbidity rate of cancer victims varies greatly for similar patients who receive similar treatments. It is hypothesized that this variation can be explained by the genetic heterogeneity of the disease. DNA Microarrays, which can simultaneously measure the expression level of thousands of different genes, have been successfully used to identify such genetic differences. However, microarray data typically has a large number of features and relatively few observations, meaning that conventional machine learning tools can fail when applied to such data. We describe a novel procedure called "nearest shrunken centroids" that has successfully detected clinically relevant genetic differences in cancer patients. This procedure has the potential to become a powerful tool for diagnosing and treating cancer.

Keywords

Microarrays, shrunken centroids, classification

1. OVERVIEW

When a patient is diagnosed with cancer, various clinical parameters are used to assess the risk of metastasis and death in that patient. However, despite numerous advances in the field, our ability to determine the risk of morbidity is extremely limited. Tumors that appear indistinguishable under the microscope may have drastically different effects on the patient.

It has long been known that cancer is a genetic disease. Thus, it is commonly believed that these differences in the clinical outcome of cancer can be explained by differences in the genetic profile of the tumor. Unfortunately, until recently, our ability to directly observe the genetic makeup of a tumor was extremely limited.

This is changing, however, with the advent of DNA microarrays. Microarrays can simultaneously measure the expression levels of thousands of genes in an organism. Thus, they have the ability to detect differences between tumors at the molecular level. This is illustrated in Figure 1. Under the microscope, the two types of lymphoma appear to be identical. However, gene expression profiling reveals that the two tumor types are actually distinct at the molecular level.

The ability to identify such subgroups has important implications for the diagnosis and treatment of cancer. Suppose one subtype of cancer is likely to metastasize whereas another subtype is not. The patients who have a high risk of metastasis would need to be treated aggressively, whereas the other patients could be given a less invasive treatment (or no treatment at all). If there is no way to distinguish between these subtypes, all patients would need to be given the aggressive treatment. However, this is highly undesirable, because current treatments for cancer, such as surgery or chemotherapy, have extremely severe side effects. (In fact, some cancer patients have died as a result of chemotherapy.) If we could successfully identify the patients with a high risk of metastasis and death, we could give them the appropriate treatment while sparing other patients from the noxious side effects that such treatment would entail.

This is essentially a classification problem. Given a number of features (gene expression levels), we wish to predict which type of cancer is present in a patient. Many machine learning procedures have been developed for this type of problem. (See, for example, [4; 6].)

Unfortunately, these existing machine learning procedures cannot be directly applied to microarray data. The number of features is extremely large compared to the number of observations, causing most machine learning procedures to fail. Moreover, it is important to identify to identify which

Figure 1: DNA Microarrays can identify differences between tumors that are not detectable under a microscope. Using conventional microscopic analysis, the lymphoma cells in groups A and B appear to be identical. Microarrays analysis shows that different genes are active and inactive in these two groups, indicating that they represent distinct disease subtypes.

Recently ML techniques have made breakthroughs in:

- bioinformatics
- speech recognition
- computer vision

SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS

Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton

Department of Computer Science, University of Toronto

ABSTRACT

Recurrent neural networks (RNNs) are a powerful model for sequential data. End-to-end training methods such as Connectionist Temporal Classification make it possible to train RNNs for sequence labelling problems where the input-output alignment is unknown. The combination of these methods with the Long Short-term Memory RNN architecture has proved particularly fruitful, delivering state-of-the-art results in cursive handwriting recognition. However RNN performance in speech recognition has so far been disappointing, with better results returned by deep feedforward networks. This paper investigates *deep recurrent neural networks*, which combine the multiple levels of representation that have proved so effective in deep networks with the flexible use of long range context that empowers RNNs. When trained end-to-end with suitable regularisation, we find that deep Long Short-term Memory RNNs achieve a test set error of 17.7% on the TIMIT phoneme recognition benchmark, which to our knowledge is the best recorded score.

Index Terms— recurrent neural networks, deep neural networks, speech recognition

1. INTRODUCTION

Neural networks have a long history in speech recognition, usually in combination with hidden Markov models [1, 2]. They have gained attention in recent years with the dramatic improvements in acoustic modelling yielded by deep feedforward networks [3, 4]. Given that speech is an inherently dynamic process, it seems natural to consider recurrent neural networks (RNNs) as an alternative model. HMM-RNN systems [5] have also seen a recent revival [6, 7], but do not currently perform as well as deep networks.

Instead of combining RNNs with HMMs, it is possible to train RNNs 'end-to-end' for speech recognition [8, 9, 10]. This approach exploits the larger state-space and richer dynamics of RNNs compared to HMMs, and avoids the problem of using potentially incorrect alignments as training targets. The combination of Long Short-term Memory [11], an RNN architecture with an improved memory, with end-to-end training has proved especially effective for cursive handwriting recognition [12, 13]. However it has so far made little impact on speech recognition.

RNNs are inherently deep in time, since their hidden state is a function of all previous hidden states. The question that inspired this paper was whether RNNs could also benefit from depth in space; that is from stacking multiple recurrent hidden layers on top of each other, just as feedforward layers are stacked in conventional deep networks. To answer this question we introduce *deep Long Short-term Memory* RNNs and assess their potential for speech recognition. We also present an enhancement to a recently introduced end-to-end learning method that jointly trains two separate RNNs as acoustic and linguistic models [10]. Sections 2 and 3 describe the network architectures and training methods, Section 4 provides experimental results and concluding remarks are given in Section 5.

2. RECURRENT NEURAL NETWORKS

Given an input sequence $\boldsymbol{x} = (x_1, \ldots, x_T)$, a standard recurrent neural network (RNN) computes the hidden vector sequence $\boldsymbol{h} = (h_1, \ldots, h_T)$ and output vector sequence $\boldsymbol{y} = (y_1, \ldots, y_T)$ by iterating the following equations from $t = 1$ to T:

$$h_t = \mathcal{H}\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

where the W terms denote weight matrices (e.g. W_{xh} is the input-hidden weight matrix), the b terms denote bias vectors (e.g. b_h is hidden bias vector) and $\mathcal{H}$ is the hidden layer function.

$\mathcal{H}$ is usually an elementwise application of a sigmoid function. However we have found that the Long Short-Term Memory (LSTM) architecture [11], which uses purpose-built *memory cells* to store information, is better at finding and exploiting long range context. Fig. 1 illustrates a single LSTM memory cell. For the version of LSTM used in this paper [14] $\mathcal{H}$ is implemented by the following composite function:

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \quad (3)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \quad (4)$$

$$c_t = f_tc_{t-1} + i_t\tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \quad (5)$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right) \quad (6)$$

$$h_t = o_t\tanh(c_t) \quad (7)$$

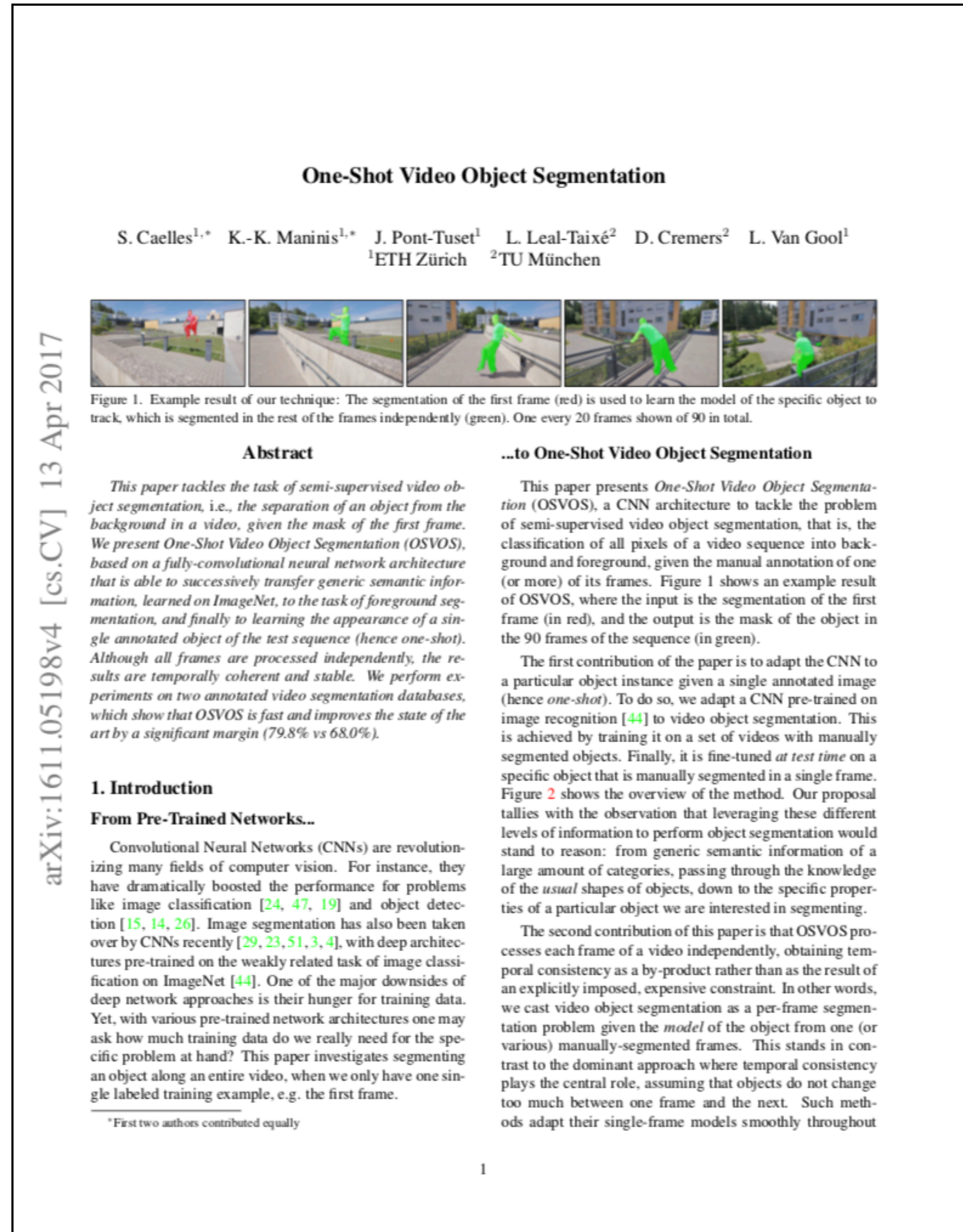
where σ is the logistic sigmoid function, and i, f, o and c are respectively the *input gate*, *forget gate*, *output gate* and

Recently ML techniques have made breakthroughs in:

bioinformatics

speech recognition

computer vision



One-Shot Video Object Segmentation

S. Caelles[1,*] K.-K. Maninis[1,*] J. Pont-Tuset[1] L. Leal-Taixé[2] D. Cremers[2] L. Van Gool[1]
[1]ETH Zürich [2]TU München

arXiv:1611.05198v4 [cs.CV] 13 Apr 2017

Figure 1. Example result of our technique: The segmentation of the first frame (red) is used to learn the model of the specific object to track, which is segmented in the rest of the frames independently (green). One every 20 frames shown of 90 in total.

Abstract

This paper tackles the task of semi-supervised video object segmentation, i.e., the separation of an object from the background in a video, given the mask of the first frame. We present One-Shot Video Object Segmentation (OSVOS), based on a fully-convolutional neural network architecture that is able to successively transfer generic semantic information, learned on ImageNet, to the task of foreground segmentation, and finally to learning the appearance of a single annotated object of the test sequence (hence one-shot). Although all frames are processed independently, the results are temporally coherent and stable. We perform experiments on two annotated video segmentation databases, which show that OSVOS is fast and improves the state of the art by a significant margin (79.8% vs 68.0%).

1. Introduction

From Pre-Trained Networks...

Convolutional Neural Networks (CNNs) are revolutionizing many fields of computer vision. For instance, they have dramatically boosted the performance for problems like image classification [24, 47, 19] and object detection [15, 14, 26]. Image segmentation has also been taken over by CNNs recently [29, 23, 51, 3, 4], with deep architectures pre-trained on the weakly related task of image classification on ImageNet [44]. One of the major downsides of deep network approaches is their hunger for training data. Yet, with various pre-trained network architectures one may ask how much training data do we really need for the specific problem at hand? This paper investigates segmenting an object along an entire video, when we only have one single labeled training example, e.g. the first frame.

*First two authors contributed equally

...to One-Shot Video Object Segmentation

This paper presents *One-Shot Video Object Segmentation* (OSVOS), a CNN architecture to tackle the problem of semi-supervised video object segmentation, that is, the classification of all pixels of a video sequence into background and foreground, given the manual annotation of one (or more) of its frames. Figure 1 shows an example result of OSVOS, where the input is the segmentation of the first frame (in red), and the output is the mask of the object in the 90 frames of the sequence (in green).

The first contribution of the paper is to adapt the CNN to a particular object instance given a single annotated image (hence *one-shot*). To do so, we adapt a CNN pre-trained on image recognition [44] to video object segmentation. This is achieved by training it on a set of videos with manually segmented objects. Finally, it is fine-tuned *at test time* on a specific object that is manually segmented in a single frame. Figure 2 shows the overview of the method. Our proposal tallies with the observation that leveraging these different levels of information to perform object segmentation would stand to reason: from generic semantic information of a large amount of categories, passing through the knowledge of the *usual* shapes of objects, down to the specific properties of a particular object we are interested in segmenting.

The second contribution of this paper is that OSVOS processes each frame of a video independently, obtaining temporal consistency as a by-product rather than as the result of an explicitly imposed, expensive constraint. In other words, we cast video object segmentation as a per-frame segmentation problem given the *model* of the object from one (or various) manually-segmented frames. This stands in contrast to the dominant approach where temporal consistency plays the central role, assuming that objects do not change too much between one frame and the next. Such methods adapt their single-frame models smoothly throughout

1

Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov

Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player

Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player
- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match

Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player
- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match
- **2017** AlphaMaster beats Ke Jie, the Go world champion

Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player
- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match
- **2017** AlphaMaster beats Ke Jie, the Go world champion
- **2018** AlphaZero crashes AlphaMaster 100-0 (self-play)

Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player
- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match
- **2017** AlphaMaster beats Ke Jie, the Go world champion
- **2018** AlphaZero crashes AlphaMaster 100-0 (self-play)
- **2018** AlphaZero in Chess, Go, Shogi

Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player
- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match
- **2017** AlphaMaster beats Ke Jie, the Go world champion
- **2018** AlphaZero crashes AlphaMaster 100-0 (self-play)
- **2018** AlphaZero in Chess, Go, Shogi
- DeepMind stops AlphaZero, concluding that it **beats at any "perfect information" game**

StarCraft is a deep, complicated war strategy game. Google's AlphaStar AI crushed it.

DeepMind has conquered chess and Go and moved on to complex real-time games. Now it's beating pro gamers 10-1.

By Kelsey Piper | Updated Jan 24, 2019, 7:04pm EST

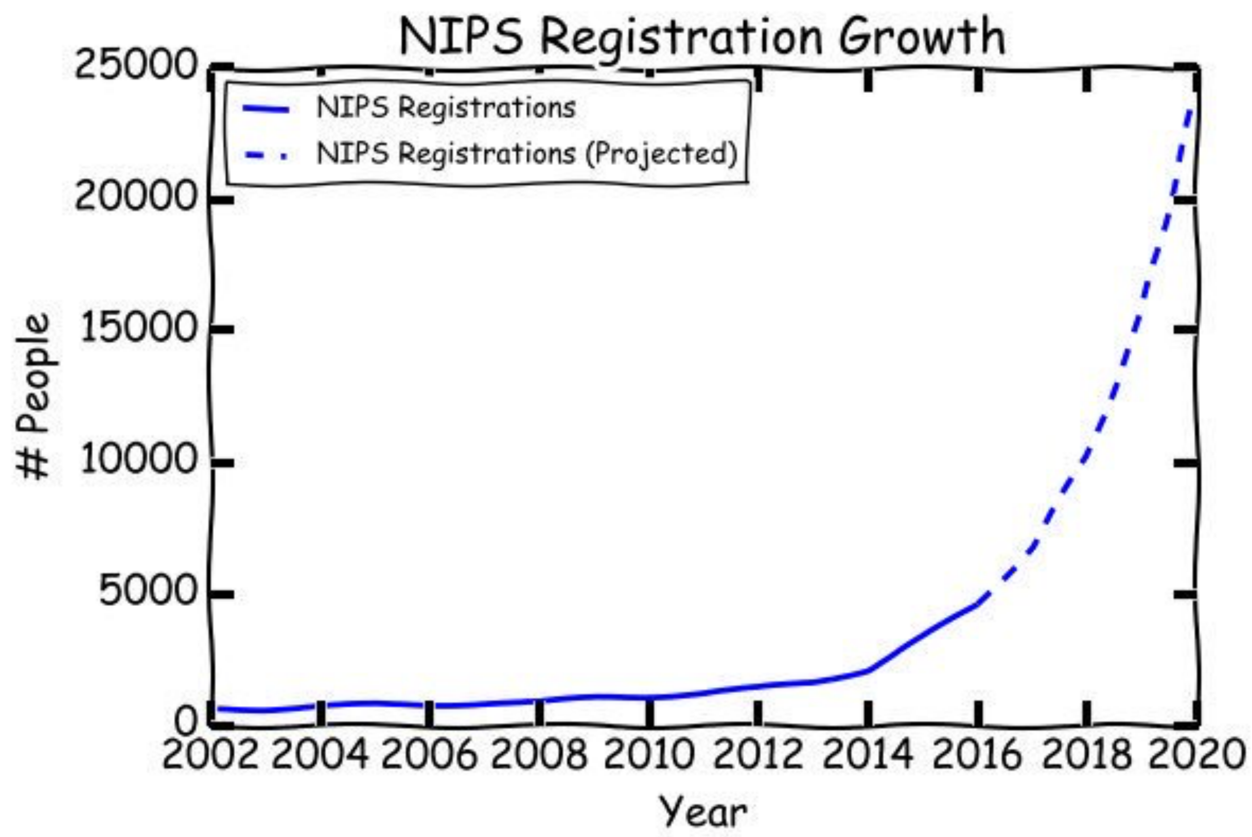
OpenAI Bot Crushes Dota 2 Champions And This is Just the Beginning

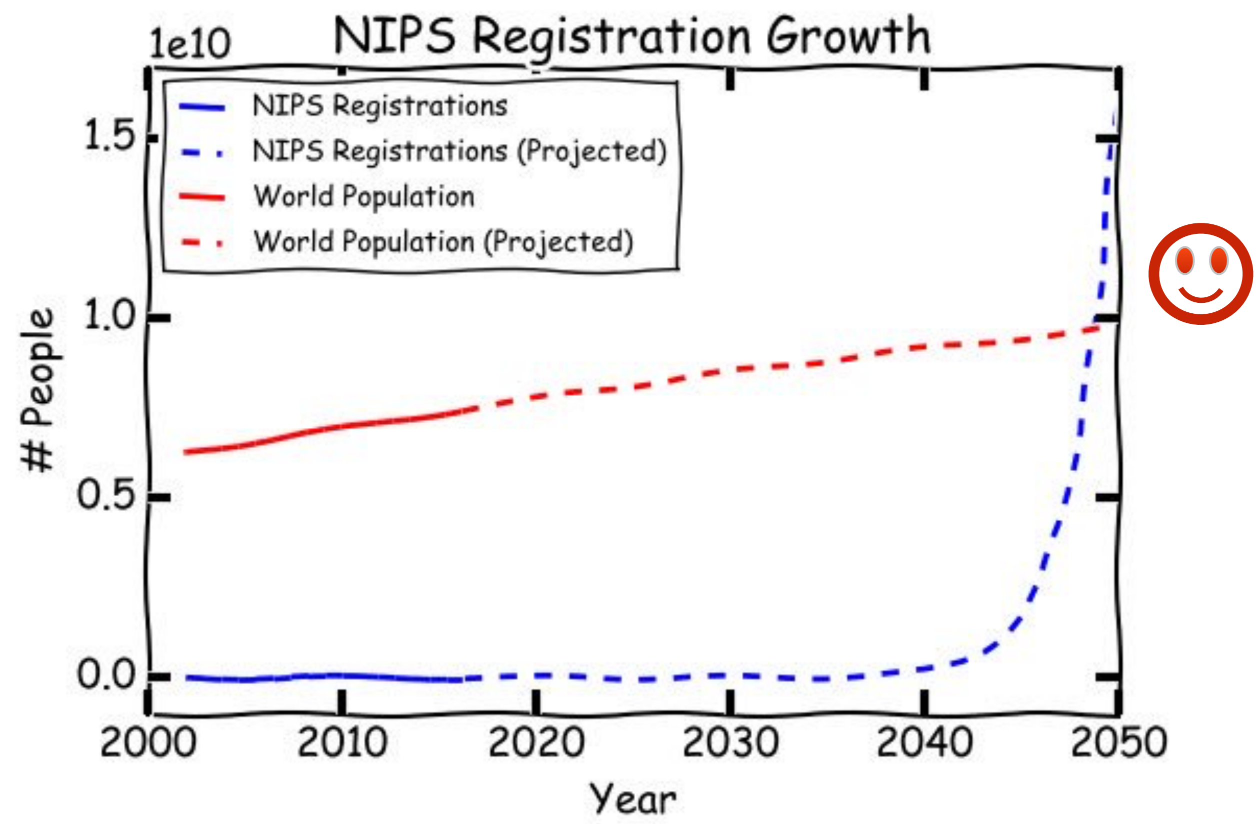
Machines, like humans, learn best when they're beaten

“AI is capable of vastly more than anyone on earth can even imagine
And its rate of improvement is exponential…”

— Elon Musk, CEO OpenAI

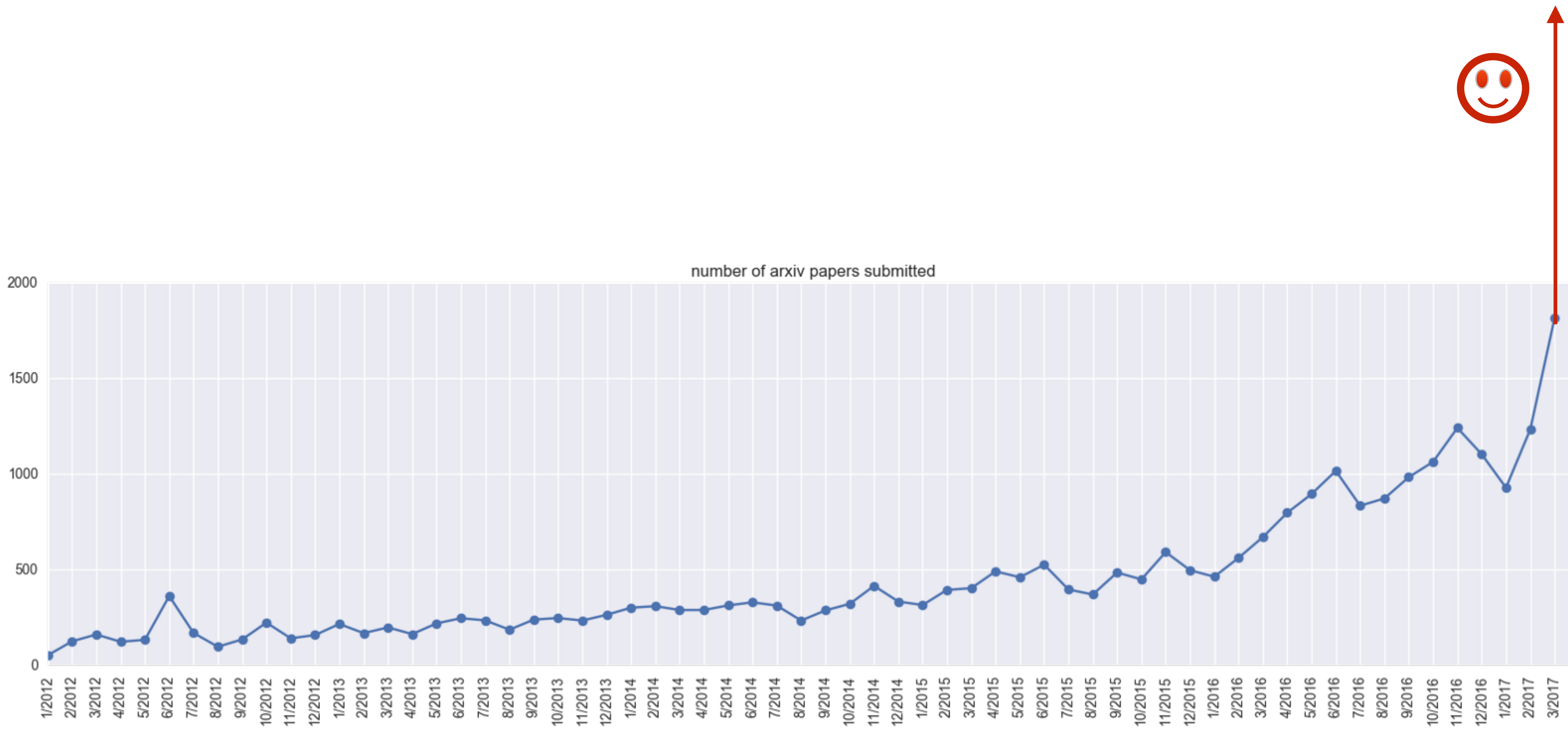
NIPS Attendance





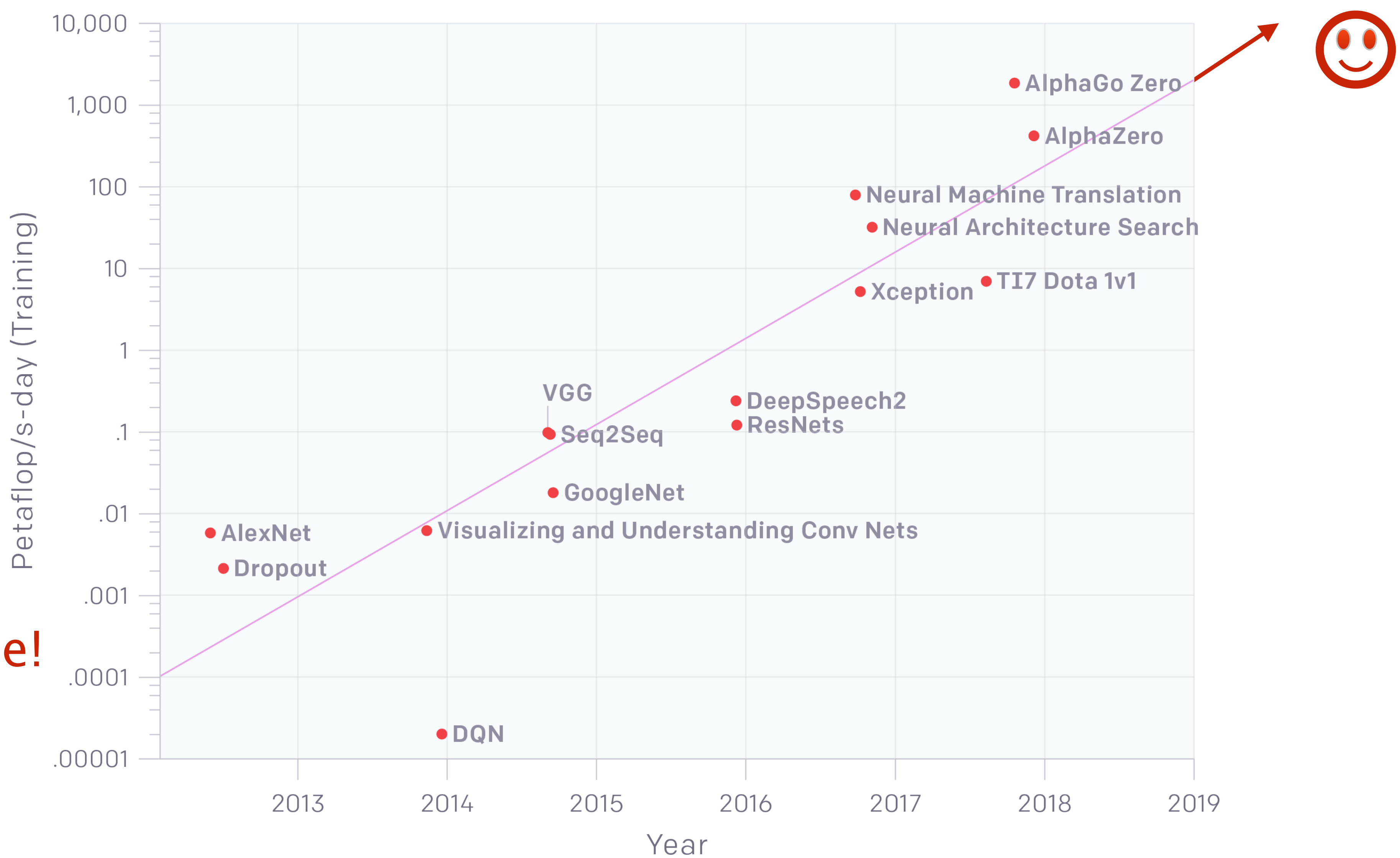
Source: @ML_Hipster

AI papers on arxiv



Source: Andrej Karpathy

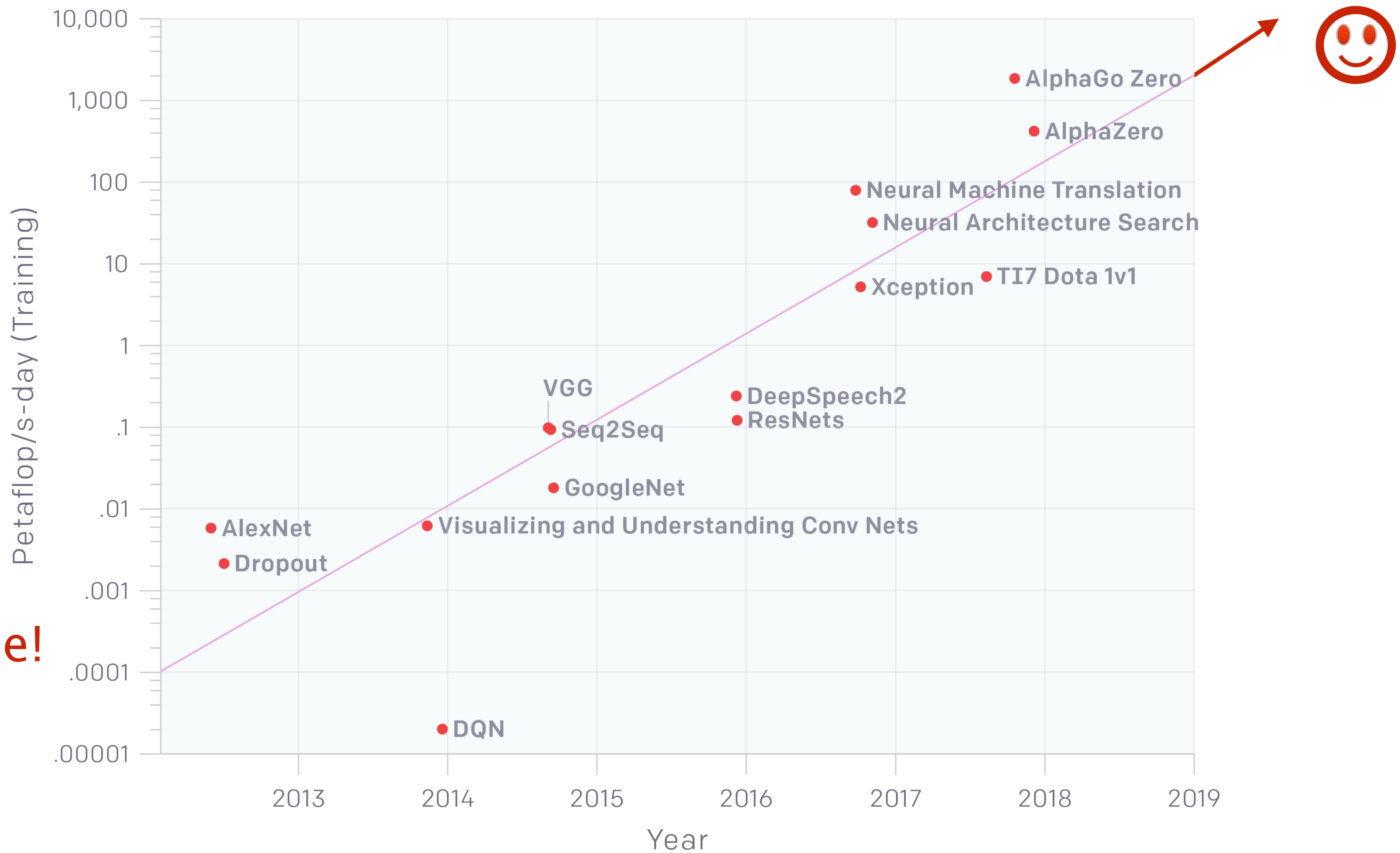
AlexNet to AlphaGo Zero:
A 300.000x Increase in Compute



Log scale!

Source: OpenAI

AlexNet to AlphaGo Zero:
A 300.000x Increase in Compute



Log scale!

Algorithmic innovation, much more data, and funding

Source: OpenAI

Machine learning refresher

ML aims to construct algorithms and models
that can learn to make decisions directly from data

supervised

classification or regression

- labeled data

unsupervised

clustering samples into groups

- unlabeled data

reinforcement

agent learns the best action series

- to maximize a cumulative reward
- interacting with the environment

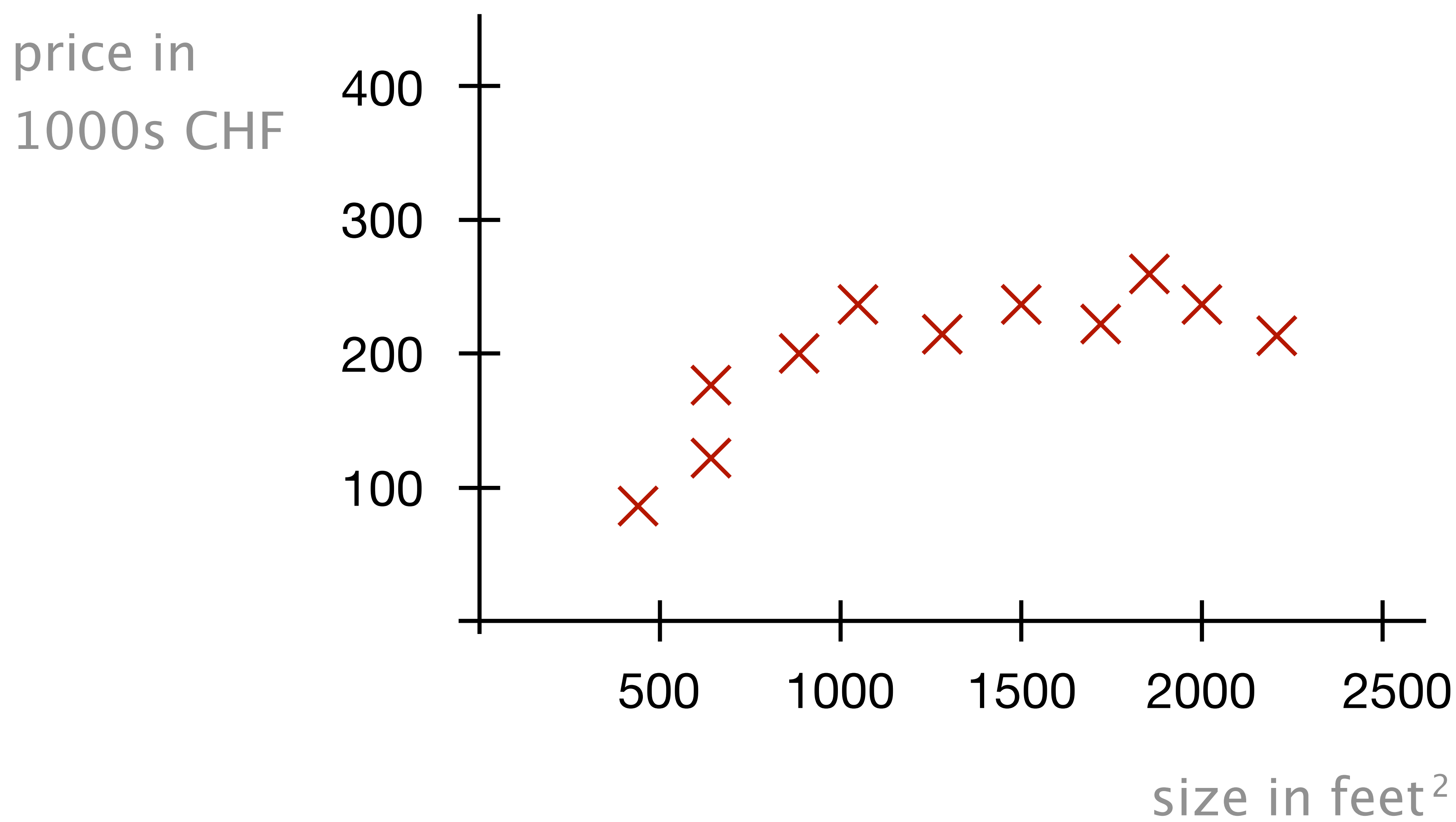
Machine learning refresher

ML aims to construct algorithms and models
that can learn to make decisions directly from data

supervised	classification or regression - labeled data
unsupervised	clustering samples into groups - unlabeled data
reinforcement	agent learns the best action series - to maximize a cumulative reward - interacting with the environment

Supervised learning

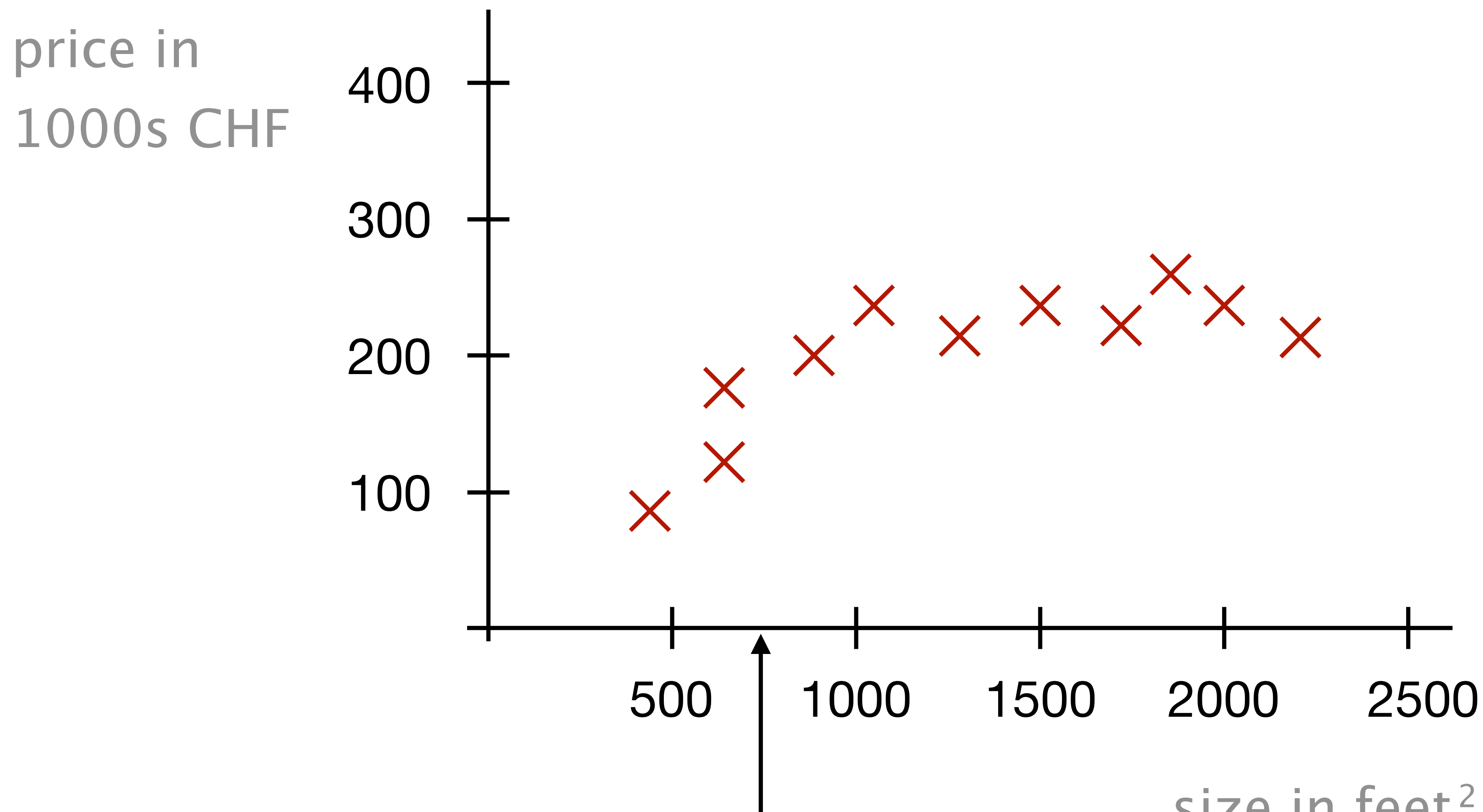
Housing price prediction



Examples from Machine Learning, Andrew Ng

Supervised learning

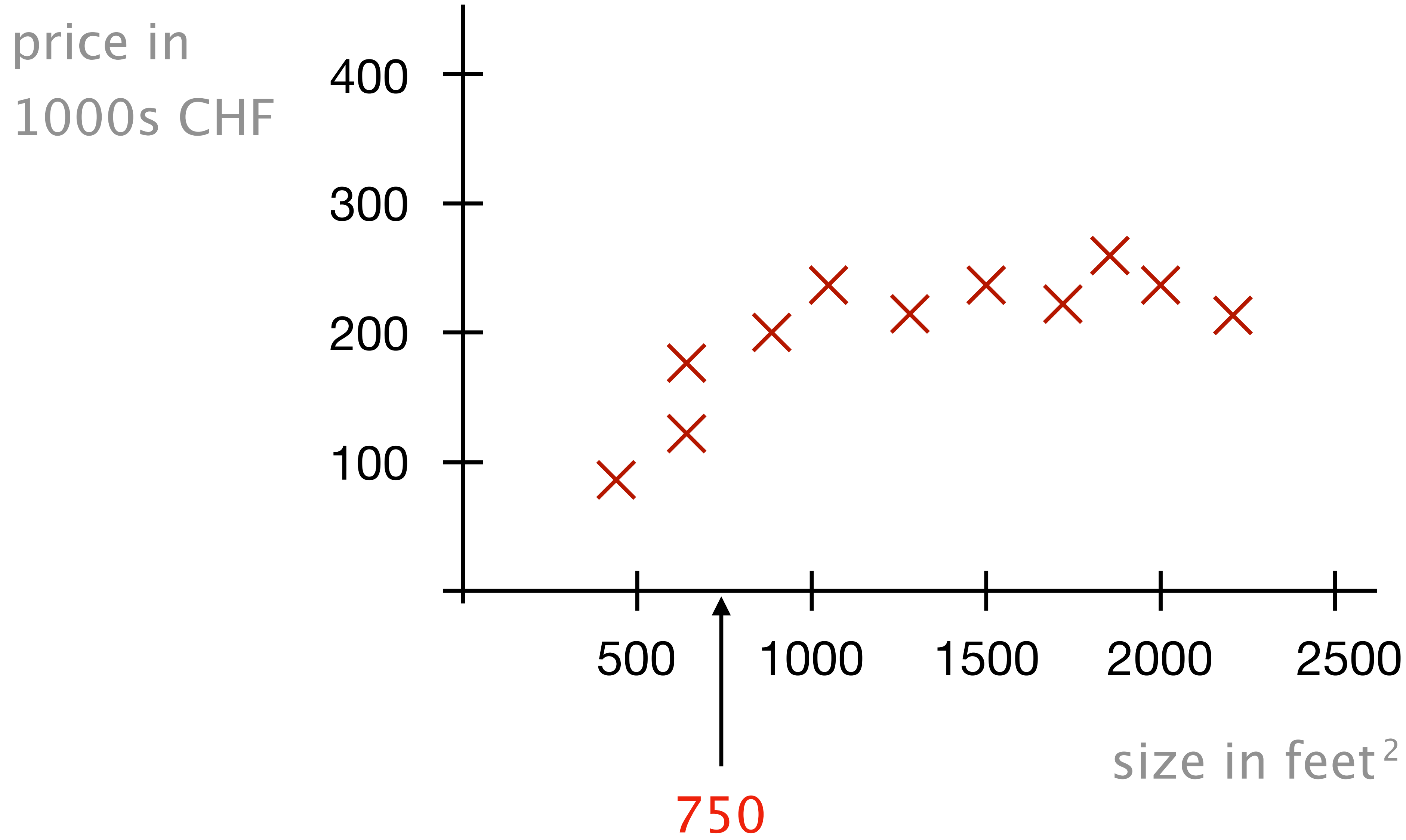
Housing price prediction



How much can I get for a 750 feet² house?

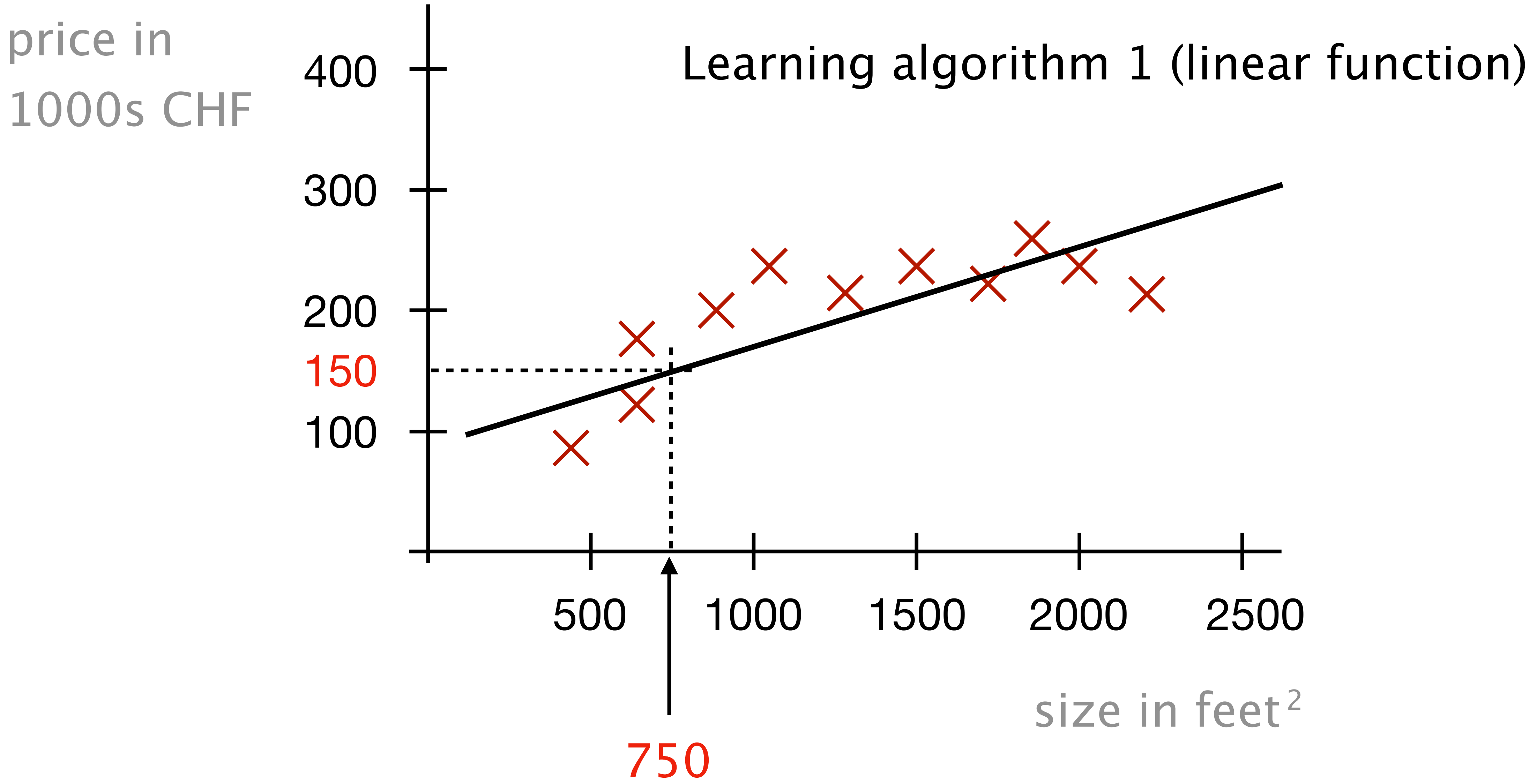
Supervised learning

Housing price prediction



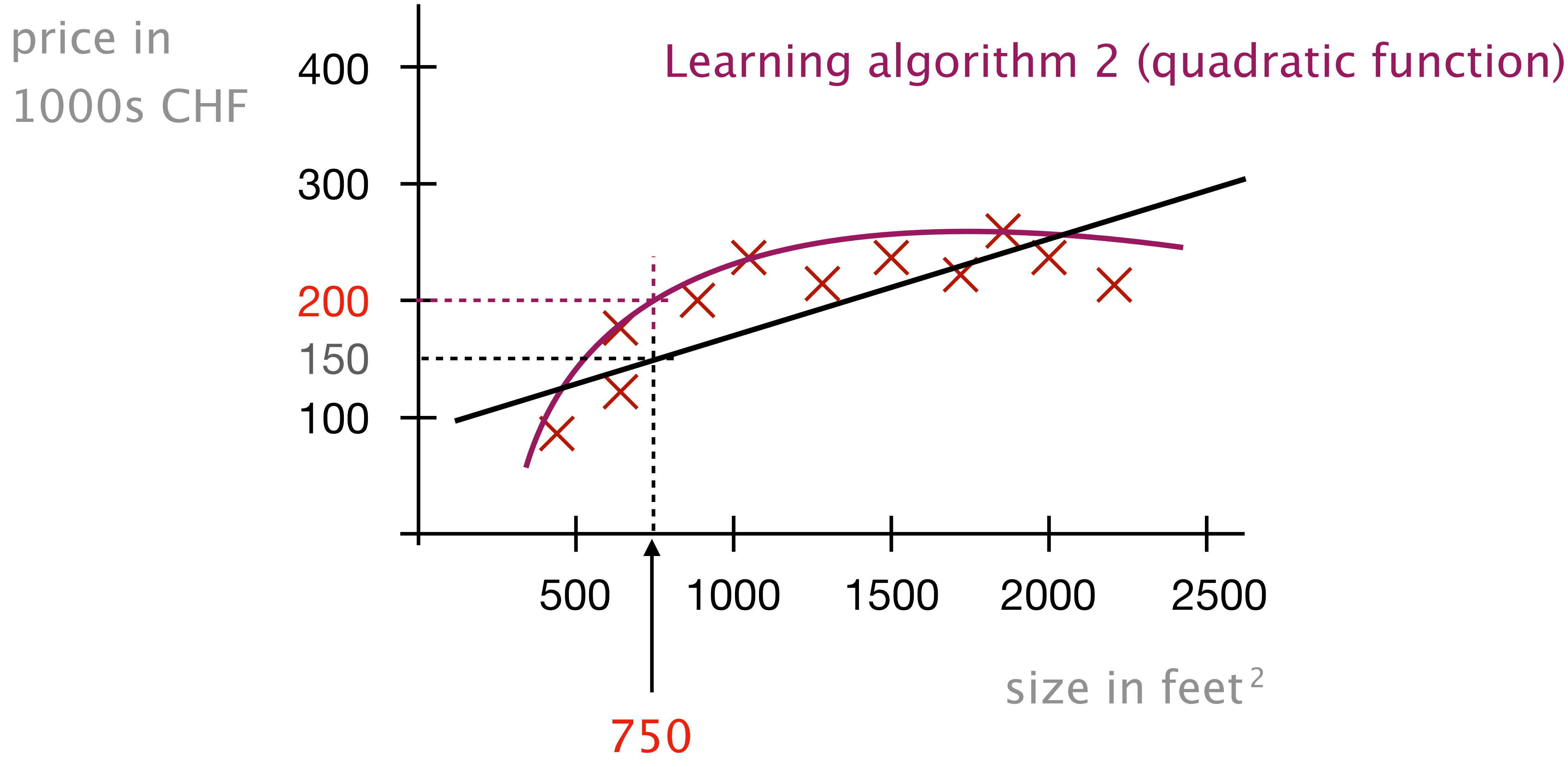
Supervised learning

Housing price prediction



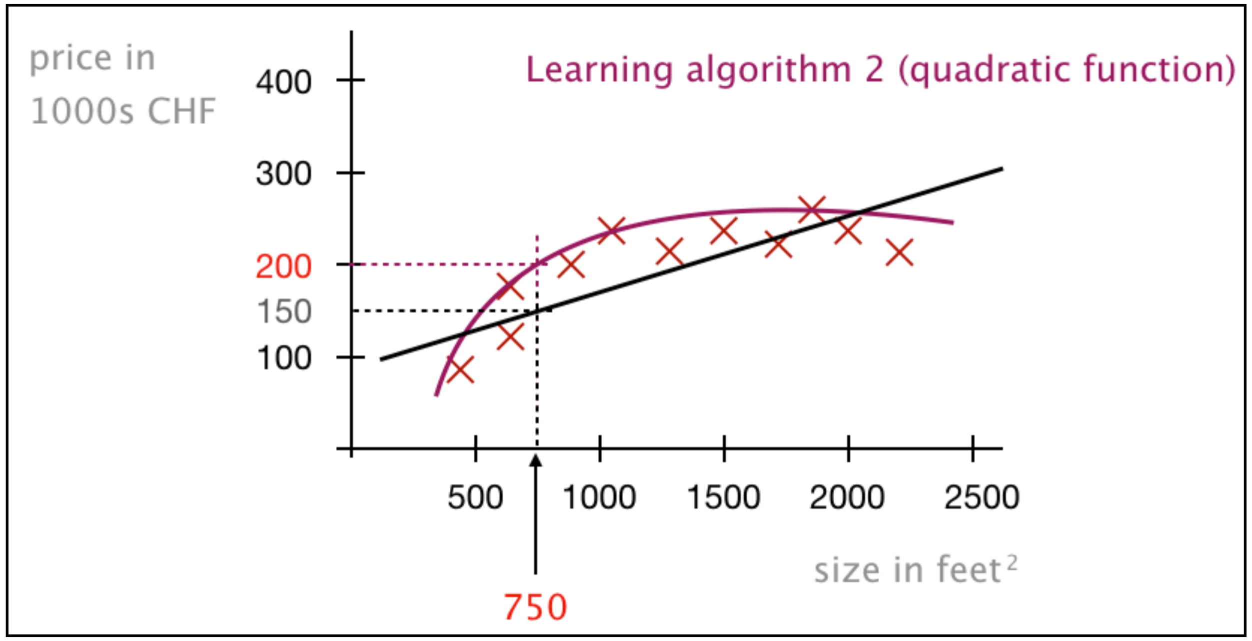
Supervised learning

Housing price prediction



Supervised learning

Housing price prediction

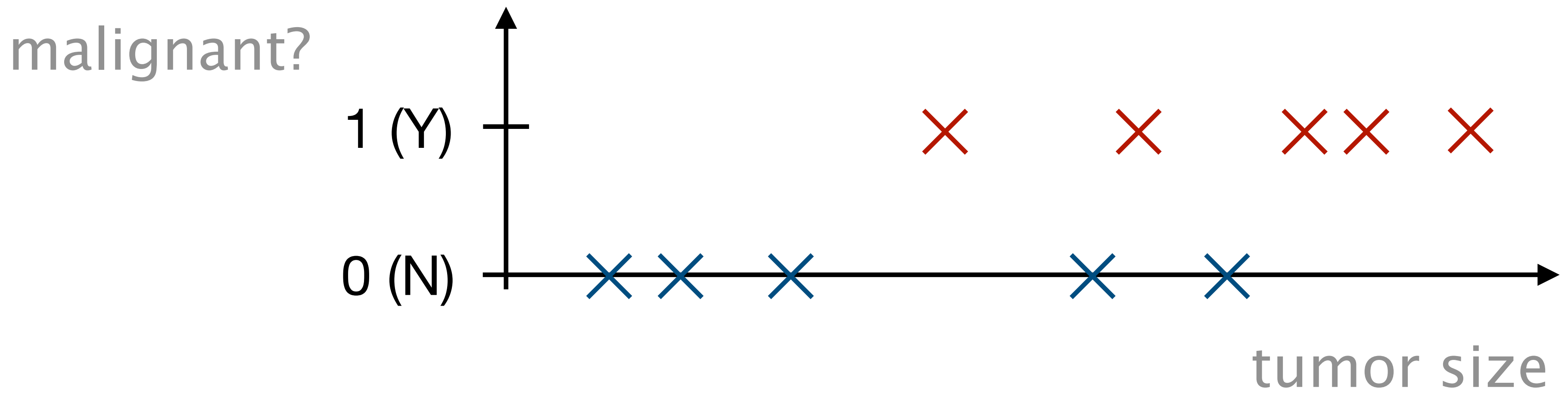


supervised — ‘right answers’ (labels) given for the training dataset

regression — predict continuous valued output (price)

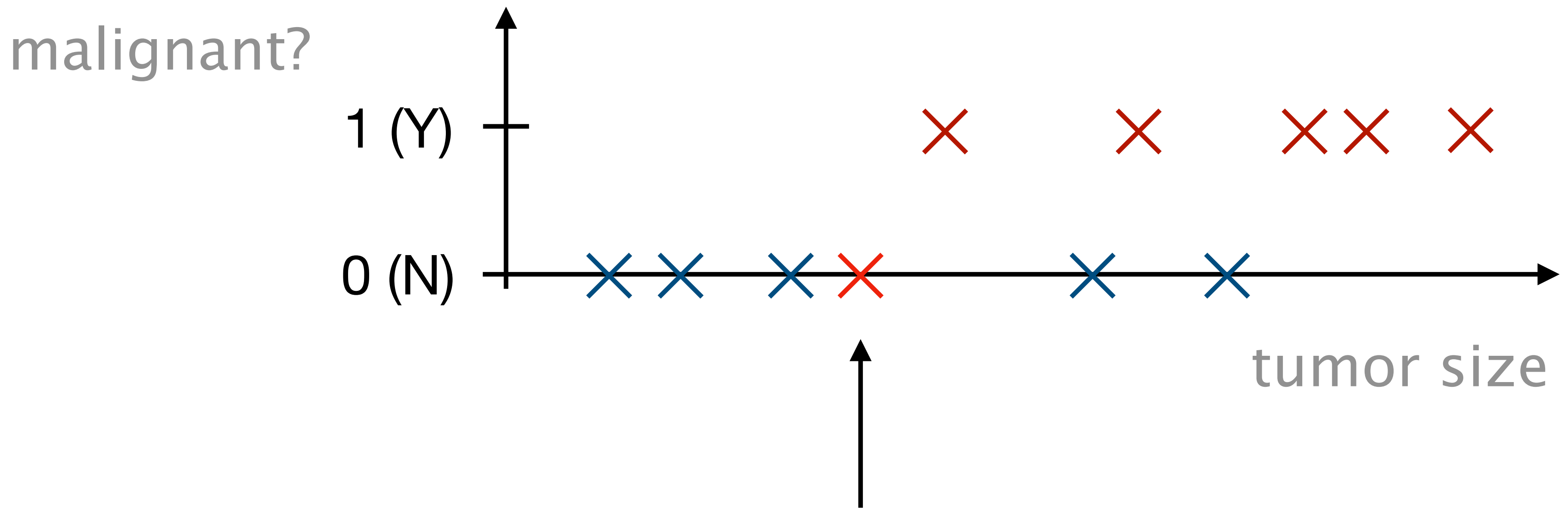
Supervised learning

Breast cancer (malignant, benign)



Supervised learning

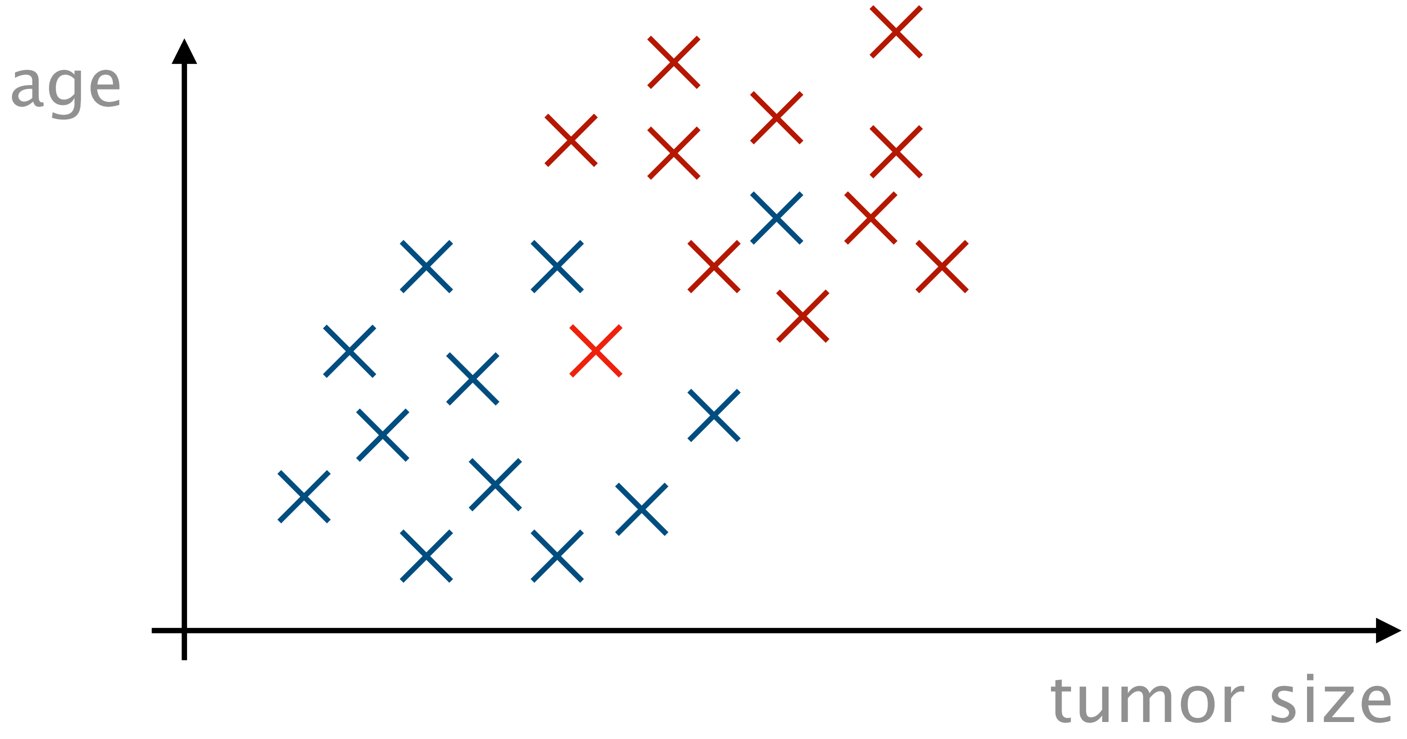
Breast cancer (malignant, benign)



Will this tumor be malignant or benign?

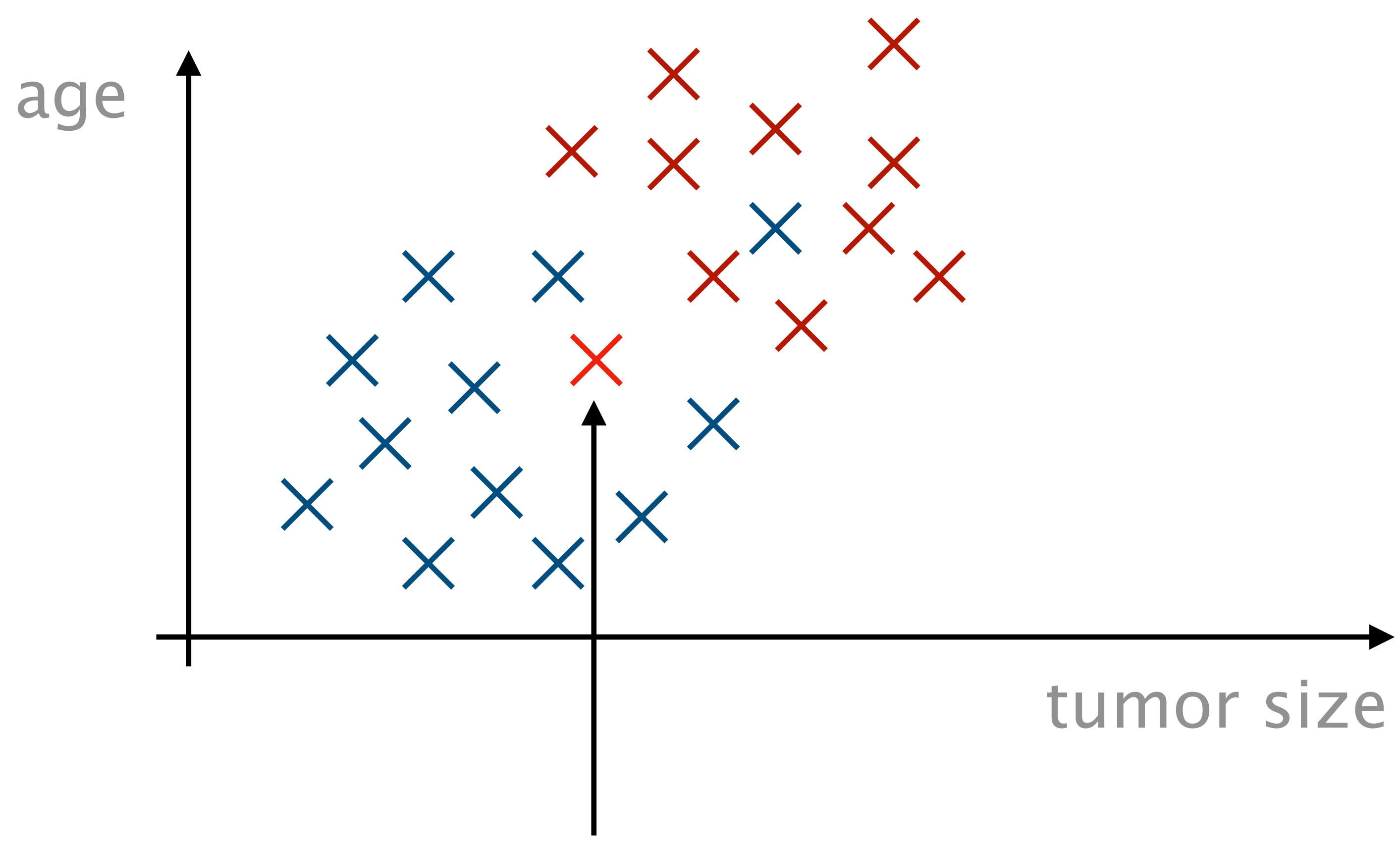
Supervised learning

Breast cancer (malignant, benign)



Supervised learning

Breast cancer (malignant, benign)

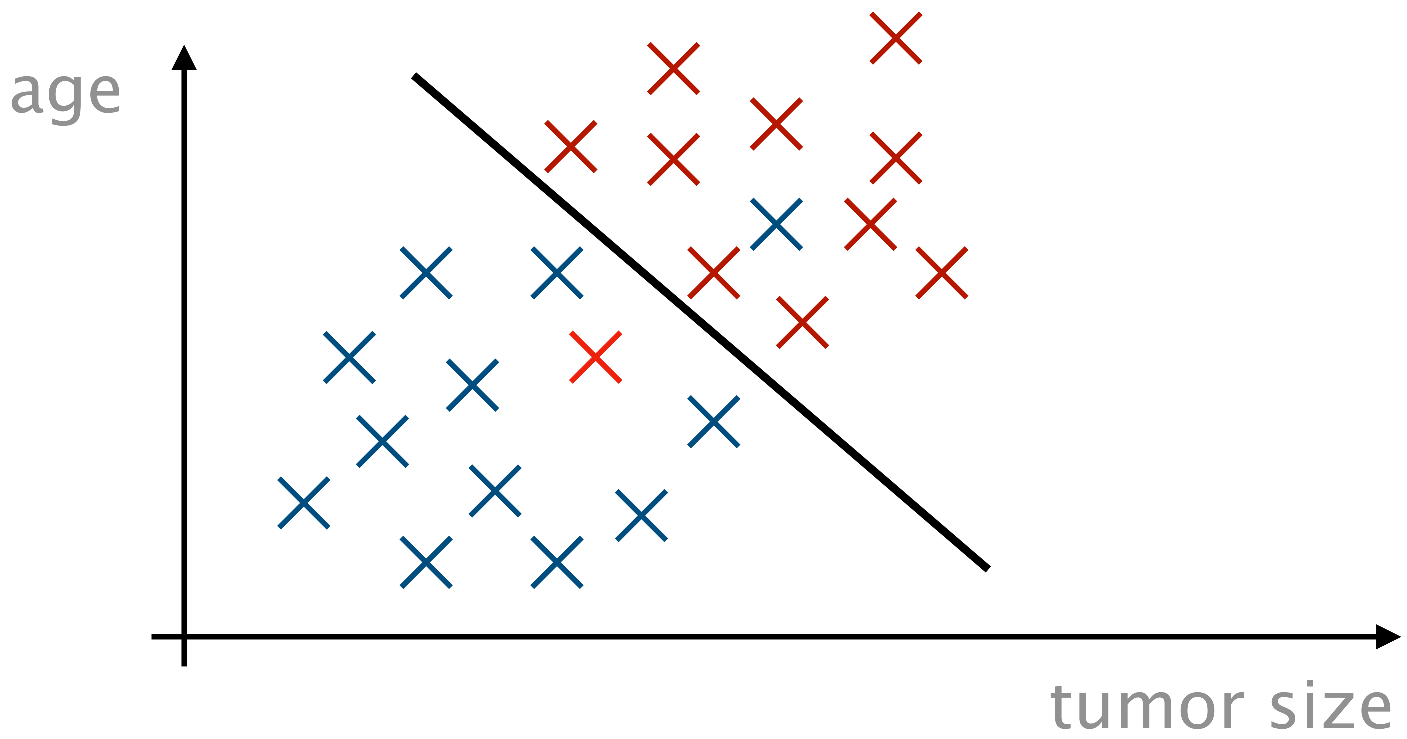


Will this tumor be malignant or benign?

Supervised learning

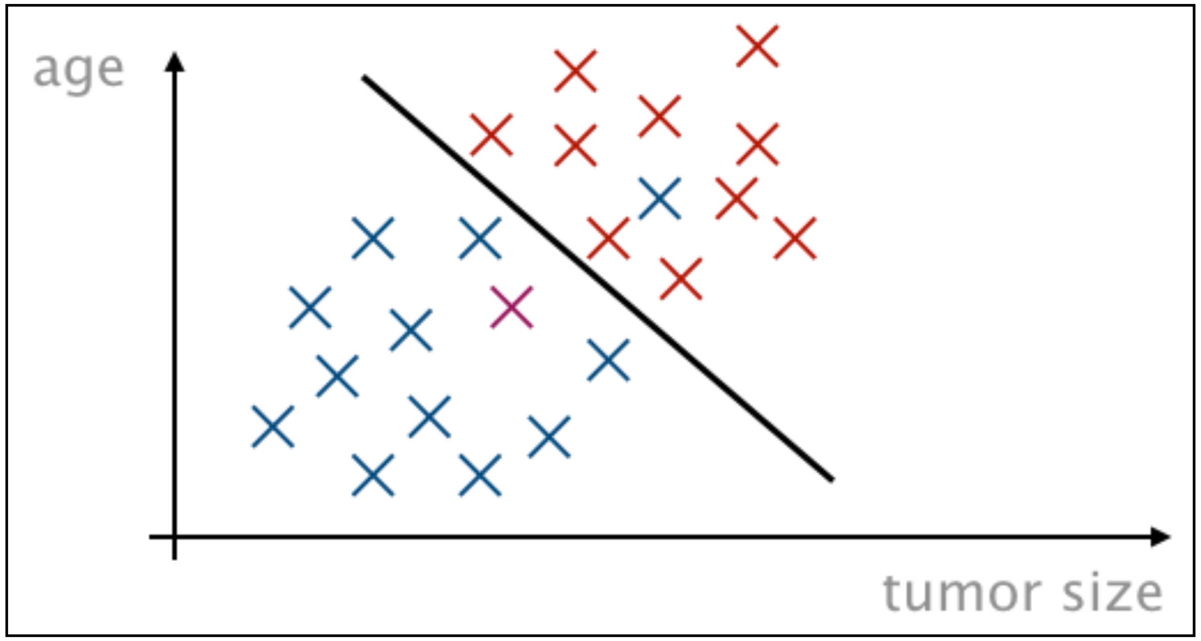
Breast cancer (malignant, benign)

Learning algorithm 1 (linear function)



Supervised learning

Breast cancer (malignant, benign)



supervised — 'right answers' (labels) given for the training dataset

classification — the output is a discrete value; large number of features (SVM)

Machine learning refresher

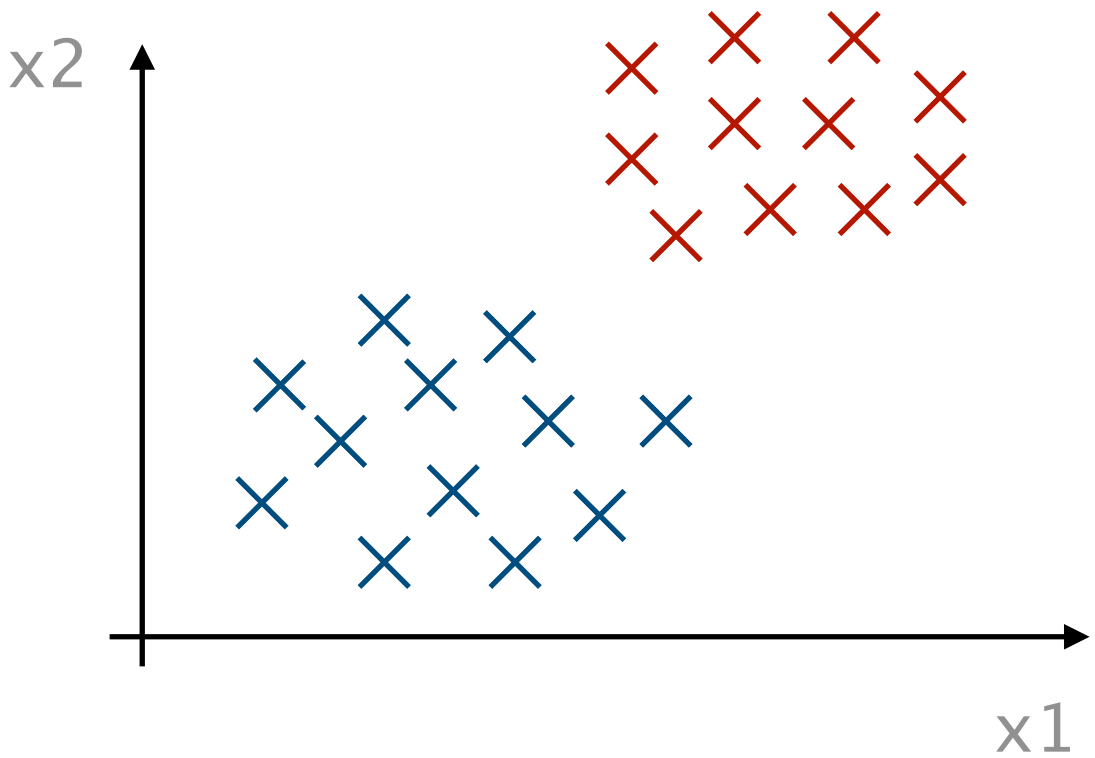
ML aims to construct algorithms and models
that can learn to make decisions directly from data

supervised — classification or regression
- labeled data

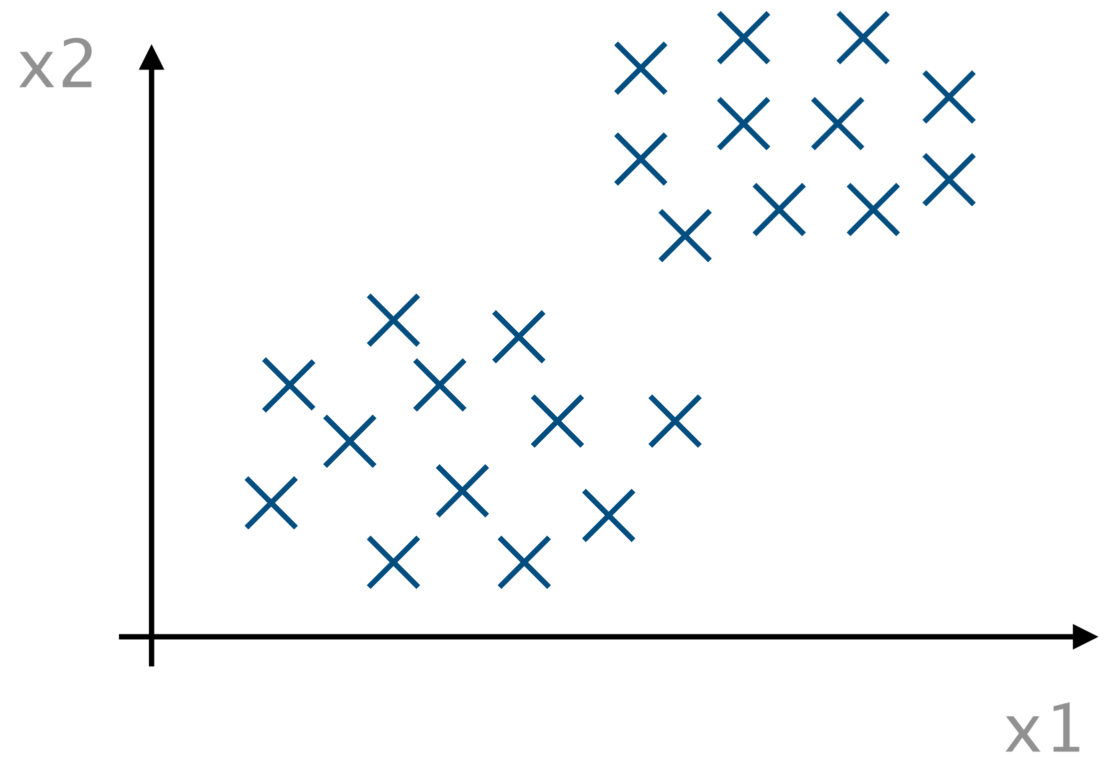
unsupervised — clustering samples into groups
- unlabeled data

reinforcement — agent learns the best action series
- to maximize a cumulative reward
- interacting with the environment

Unsupervised learning



supervised learning

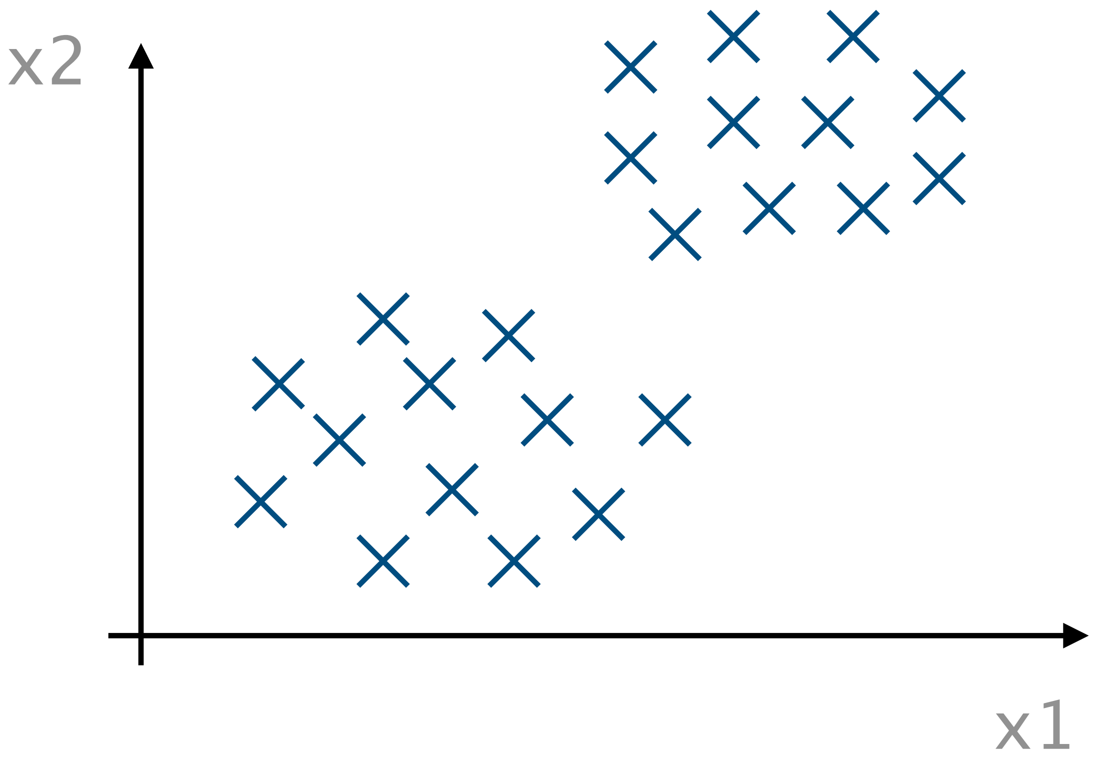


unsupervised learning

‘right answers’ (labels) given for the training dataset

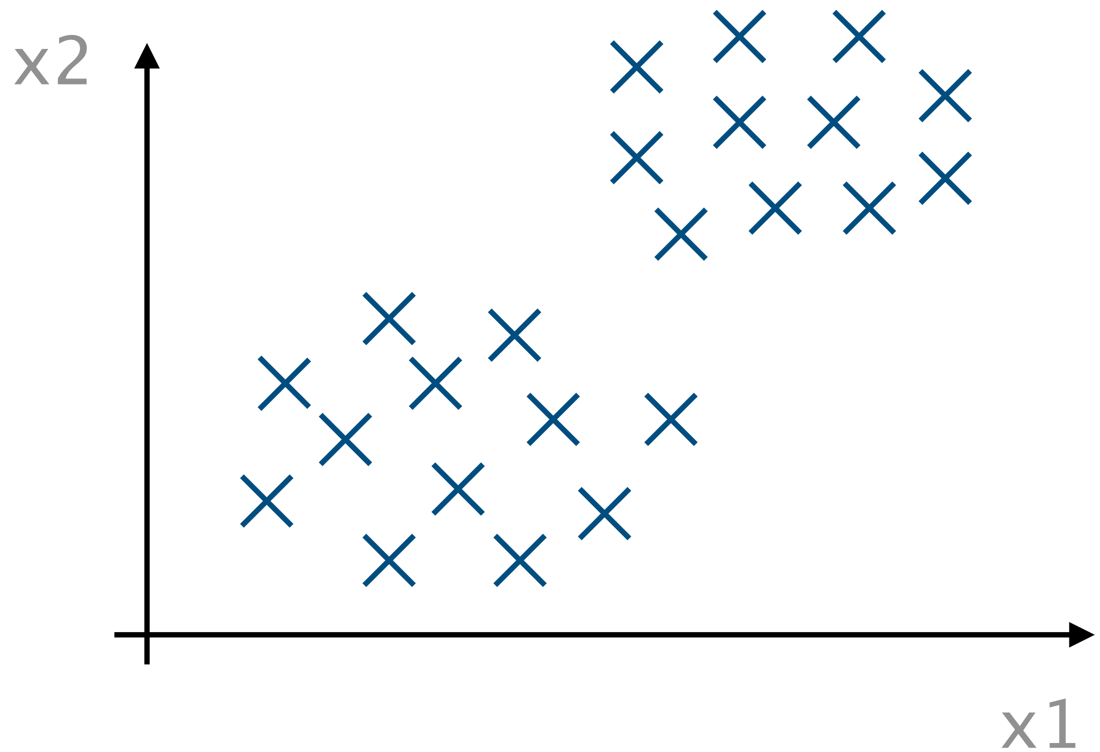
data is not labeled

Unsupervised learning



unsupervised learning

Unsupervised learning

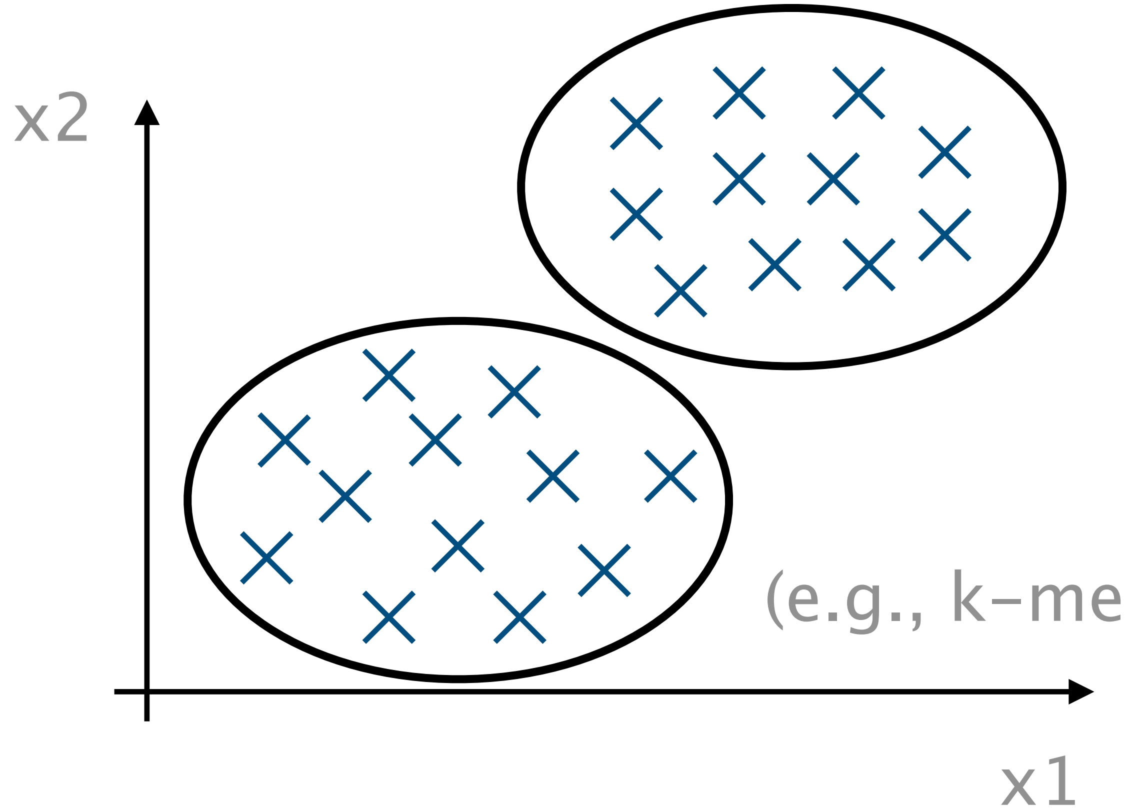


unsupervised learning

Here is a dataset. Can you find some structure?

Unsupervised learning

Learning algorithm (clustering)



(e.g., k-means, SVD)

unsupervised learning

Here is a dataset. Can you find some structure?

Machine learning refresher

ML aims to construct algorithms and models
that can learn to make decisions directly from data

supervised	classification or regression - labeled data
unsupervised	clustering samples into groups - unlabeled data
reinforcement	agent learns the best action series - to maximize a cumulative reward - interacting with the environment

Reinforcement learning (Agent and Environment)

At each step **t**, the agent:

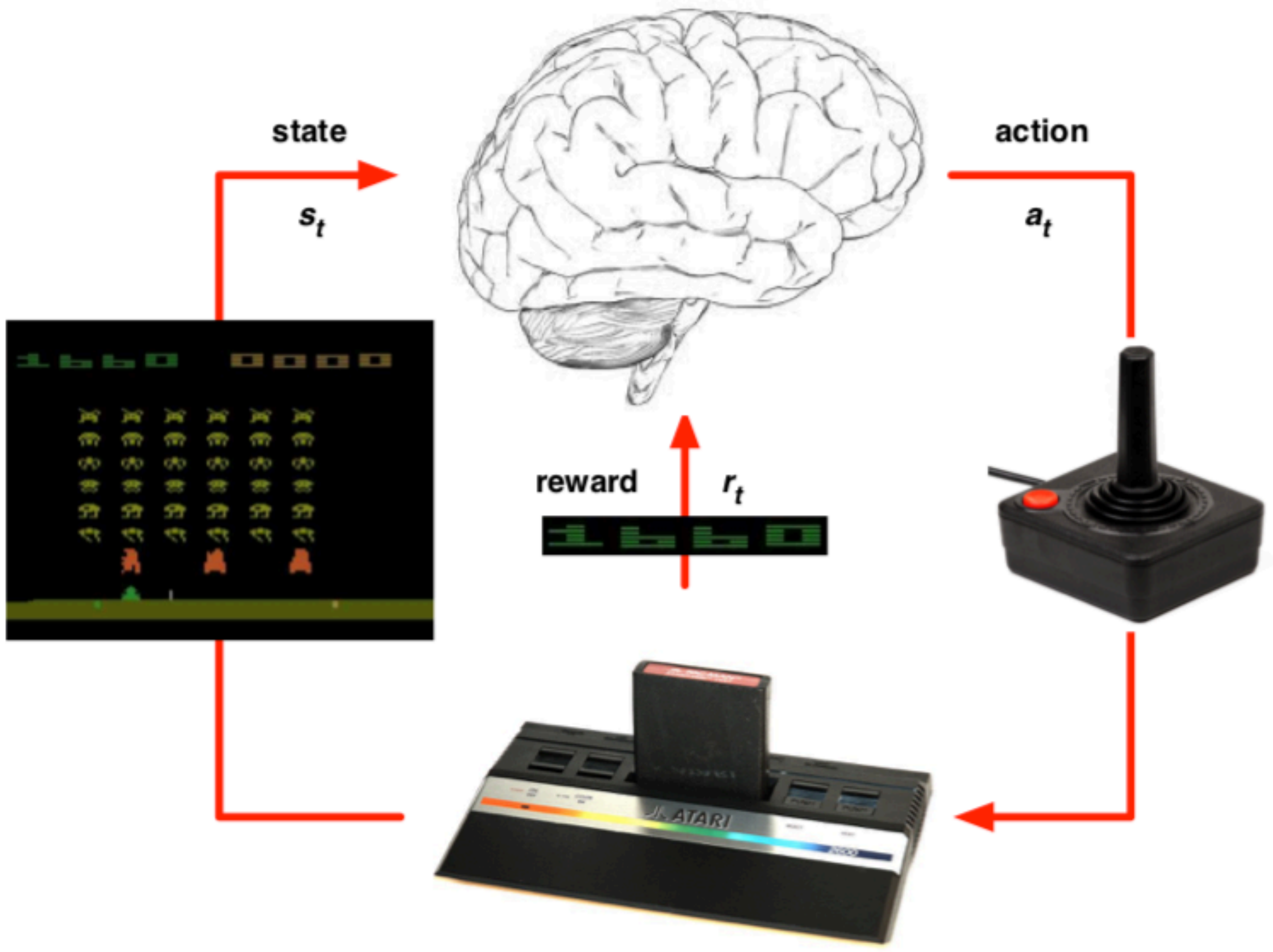
- observes state st
- receives reward rt
- executes action at

The environment:

- receives action at
- emits state st
- emits reward rt

RL in a nutshell:

- the agent selects actions to maximize future reward

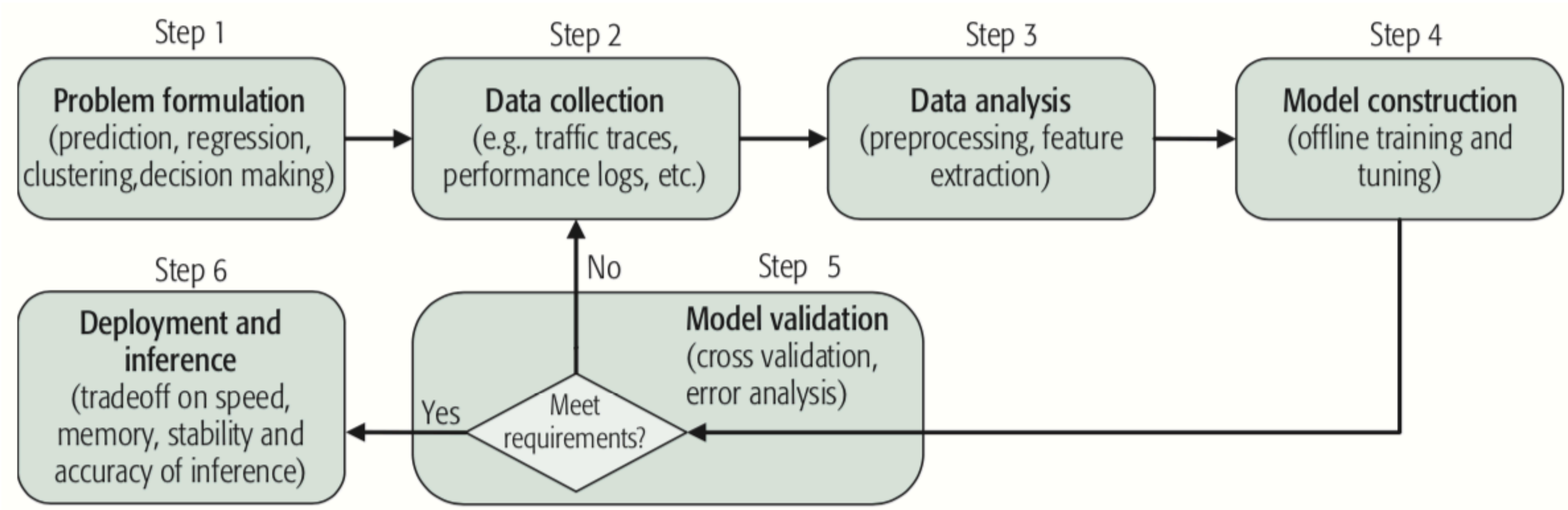


Example from DRL, David Silver (Google DeepMind)

Why is ML a good match?

ML is good in:	Which is useful for:
classification and prediction	intrusion detection performance prediction
decision making	network scheduling parameter adaptation
interaction with complex environments	load changing patterns network state

Workflow


Step 1
Problem formulation
(prediction, regression, clustering,decision making)
Step 2
Data collection
(e.g., traffic traces, performance logs, etc.)
Step 3
Data analysis
(preprocessing, feature extraction)
Step 4
Model construction
(offline training and tuning)
No
Step 5
Model validation
(cross validation, error analysis)
Meet requirements?
Yes
Step 6
Deployment and inference
(tradeoff on speed, memory, stability and accuracy of inference)

Step 1. Problem formulation

define and classify

formulate and abstract the problem

- classification problem
- clustering problem
- decision making

to help us decide:

- the best learning model
- type of data required
- amount of data required

Training is time consuming,
so better make a good decision

Step 2. Data collection

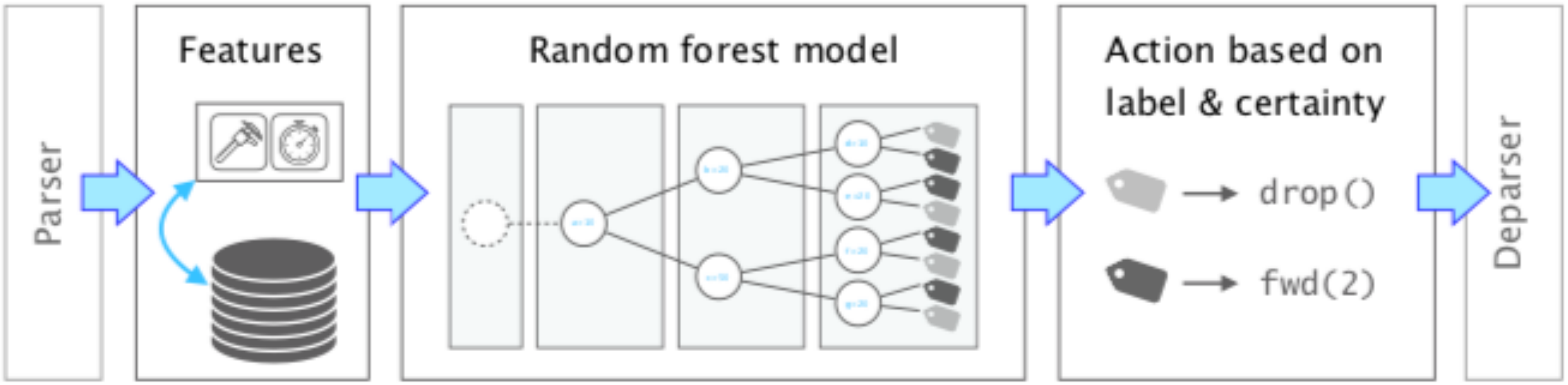
Gather a large amount of representative network data without bias

purpose	training and evaluation
type	traffic traces, logs network, application level
support	labels, feature extraction

Step 3. Data analysis

Pre-process and clean raw data

feature engineering	which parameters impact the most on the target performance



Example from pForest, Coralie Busse-Grawitz et. al.

Step 4. Model construction and validation

model selection

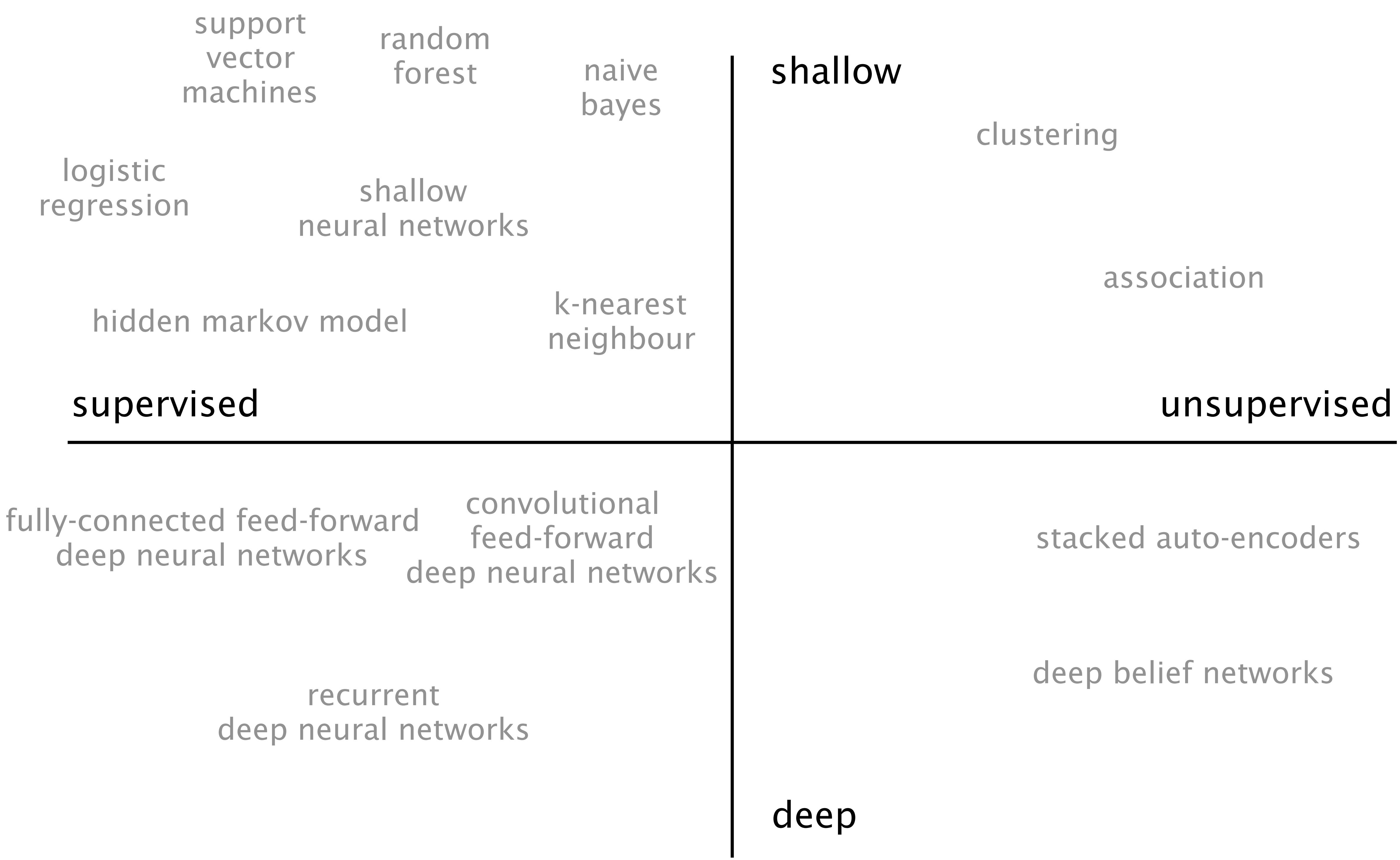
- problem category
- size of the dataset
- characteristics of scenario

tune parameters — training and tuning

adapt model

- lack of theoretical guidelines

Step 4. Model construction and validation



Step 5. Model validation and deployment

model validation

test accuracy (over/under fitting)

- increase data volume
- reduce model complexity

analyze wrong samples to find errors

iterate multiple times!

deployment

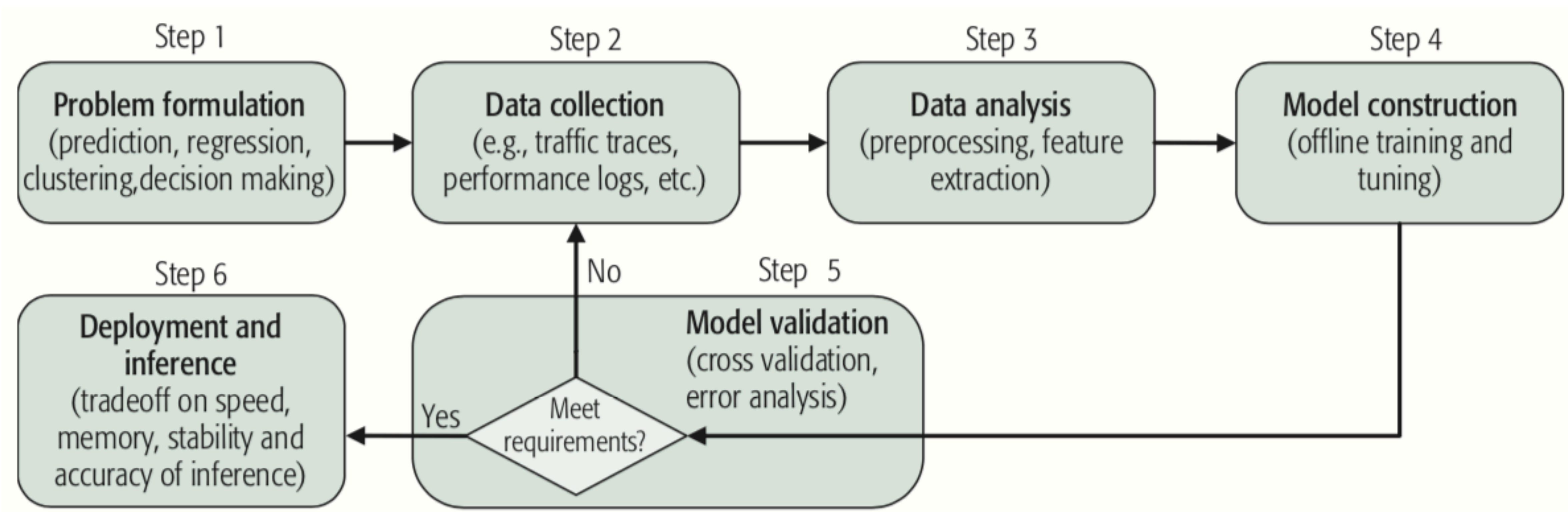
check resource constraints

accuracy vs. overhead

consider fault tolerance

Example 1. Next-generation firewall

We want to build an ML-based system to
detect and mitigate DDoS attacks



Example 2. Next-generation routing

We want to build an ML-based system to predict failures and reroute traffic proactively