MACHINE LEARNING
COURSE INTRODUCTION AND STRUCTURE

Degree: 9 CFU | Academic Year 2025/26

Instructor: Giorgio Gambosi - Danilo Croce

WHY MACHINE LEARNING?

AN (ALMOST) INFORMAL INTRO

COMPUTER SCIENCE IS…

SOLVING PROBLEMS BY MEANS OF ALGORITHMS ENCODED AS PROGRAMS

- Problem analysis
- Mathematical modeling
- Thinking an algorithm
- Implementing it in a program
- Checking correctness and efficiency

THINKING AN ALGORITHM

A SEQUENCE OF SIMPLE STEPS SOLVING THE PROBLEM

How many ‘i’ occur in this text?

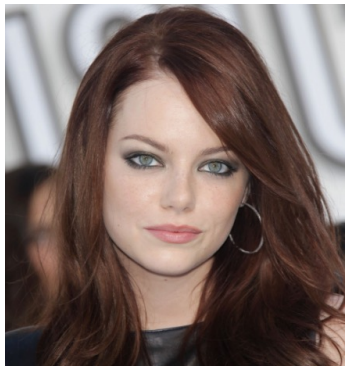
when mr. bilbo baggins of bag end announced that he would shortly be celebrating his eleventy-first birthday with a party of special magnificence, there was much talk and excitement in hobbiton. bilbo was very rich and very peculiar, and had been the wonder of the shire for sixty years, ever since his remarkable disappearance and unexpected return. the riches he had brought back from his travels had now become a local legend, and it was popularly believed, whatever the old folk might say, that the hill at bag end was full of tunnels stuffed with treasure. and if that was not enough for fame, there was also his prolonged vigour to marvel at. time wore on, but it seemed to have little effect on mr. baggins. at ninety he was much the same as at fifty. at ninety-nine they began to call him well-preserved, but unchanged would have been nearer the mark. there were some that shook their heads and thought this was too much of a good thing; it seemed unfair that anyone should possess (apparently) perpetual youth as well as (reputedly) inexhaustible wealth. ‘it will have to be paid for,' they said. ‘it isn't natural, and trouble will come of it!’ but so far trouble had not come; and as mr. baggins was generous with his money, most people were willing to forgive him his oddities and his good fortune. he remained on visiting terms with his relatives (except, of course, the sackville-bagginses), and he had many devoted admirers among the hobbits of poor and unimportant families.

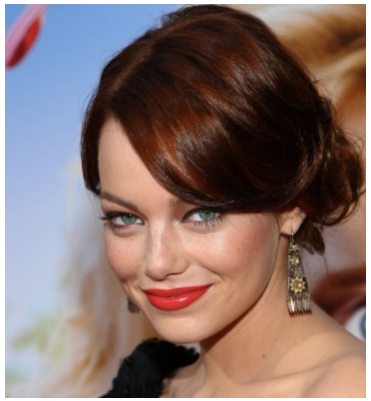
A DEFINITION OF "I"

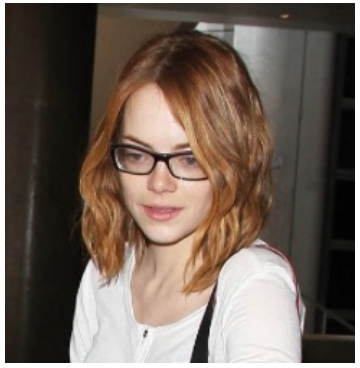
- We need a precise characterization of the letter "i" in order to identify its occurrences
- We assume only lower case letters, then the letter appear as a short vertical line with a dot above
- Also: if the text is binary encoded testo, for example in ASCII then the letter appear as a specific sequence of 7 bits, that is 1101001

BUT THINGS ARE NOT THAT SIMPLE…

How many times Emma Stone appear in this set of photos?







IN CASE YOU DO NOT KNOW HER ...

EMMA STONE

BUT HOW COULD WE CHARACTERIZE EMMA STONE?

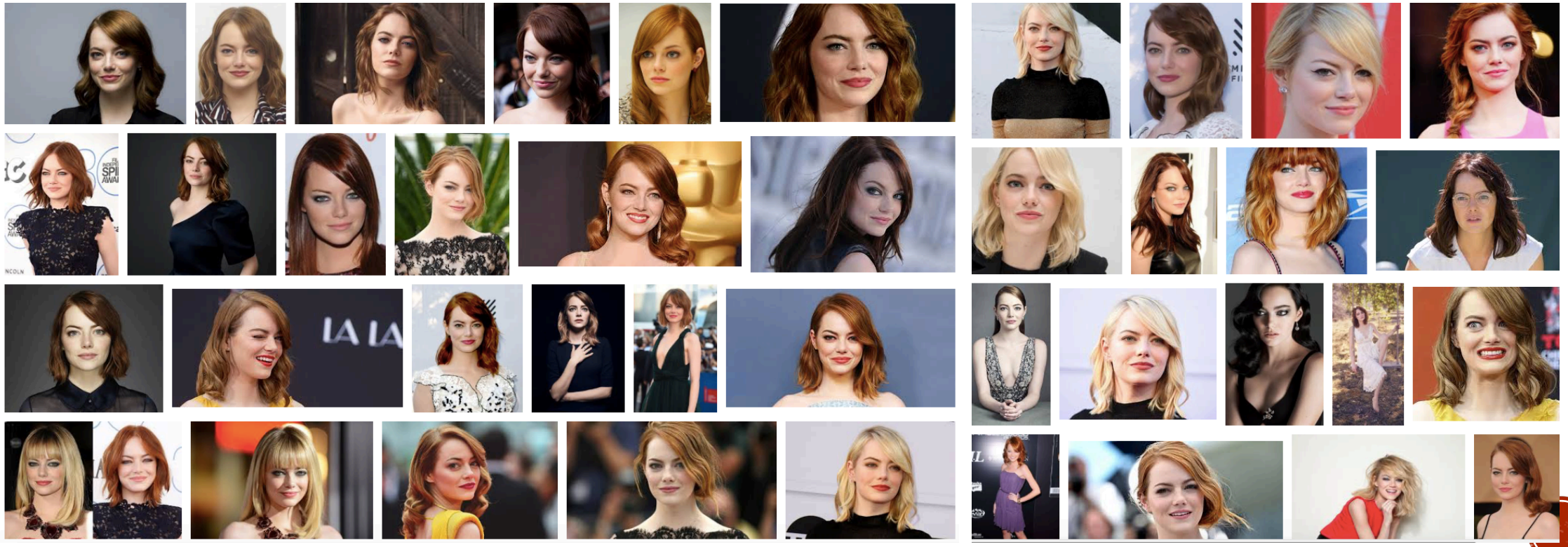
- Blonde?
- Green eyes?

IT SEEMS A DIFFICULT TASK

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

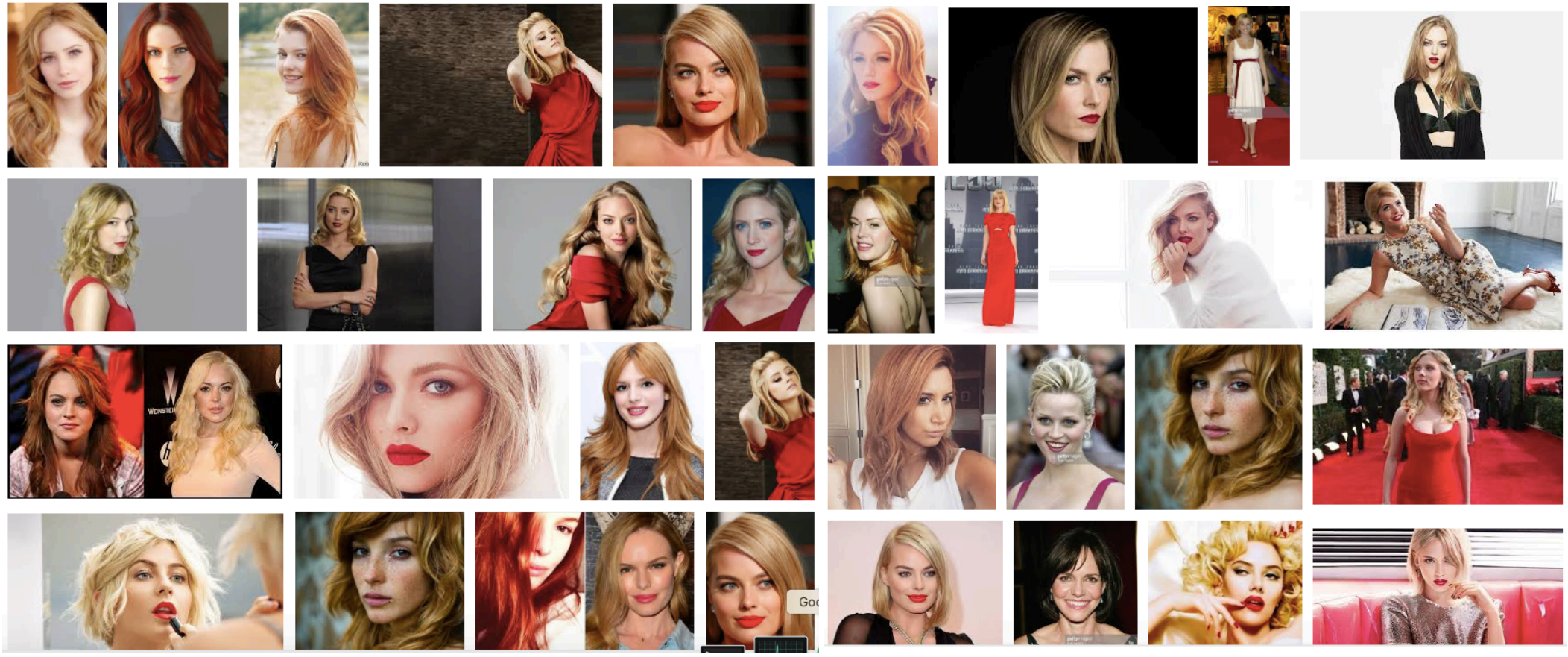
BUT, HOW CAN WE 'DEFINE' EMMA STONE?

BY EXAMPLES



BUT, HOW CAN WE 'DEFINE' EMMA STONE?

BY NEGATIVE EXAMPLES



INDUCTION

- We are not able to ‘explain’ how to identify Emma Stone and code it in an algorithm
- I can only exhibit many examples of Emma Stone (and of “not Emma Stone”)
- I need an algorithm which from all these examples ‘learns’ how to identify Emma Stone in such a way that…

Emma Stone?

YES

Emma Stone?

NO

WHAT CAN WE DO?

- We have to build a 'model' of Emma Stone and of 'not Emma Stone' starting from photos
- An image: 50×30 dots (pixel), each with 3 associated integer values (base colors) from 0 to 255
- An image: sequence of $50\times30\times3 = 4500$ values $x_1, x_2, \dots, x_{4500}$ each in $\{0, \dots, 255\}$
- What do we need? A function from $\{0, \dots, 255\}^{4500}$ to $\{0,1\}$ where 1='Emma Stone' and 0='not Emma Stone'

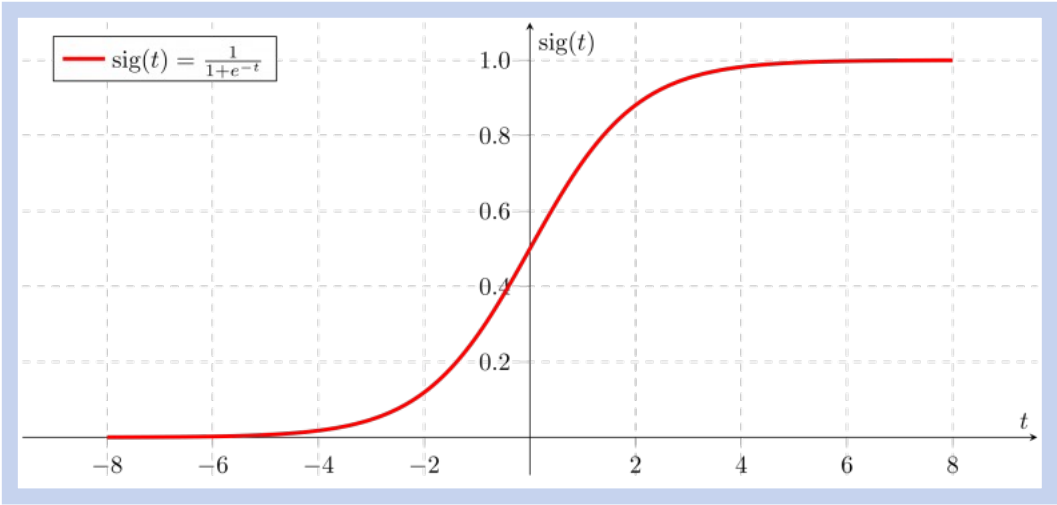
WHAT CAN WE DO?

- We look for a function from $\{0, \dots, 255\}^{4500}$ to $\{0,1\}$
- Given any photo $X = [3,6,243, \dots, 167]$ (4500 values) we interpret $f(X)$ as the probability that it shows Emma Stone
- We classify the photo as showing Emma Stone if $f(X)$ is greater than a threshold value (for example $f(X) > 0.5$)

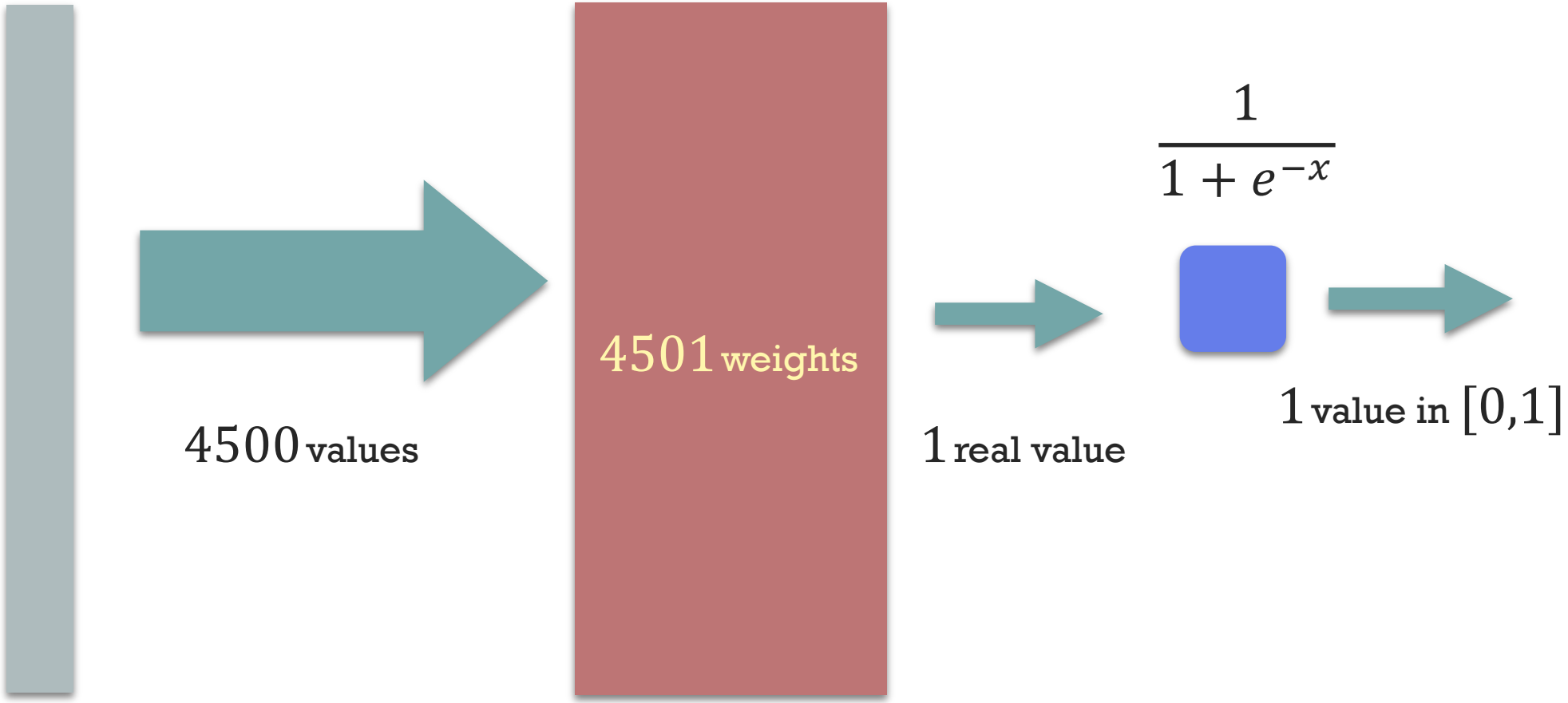
WHAT CAN WE DO?

We do not look among all possible functions, only among the ones with a given structure. For example:

- The 4500 values are combined by summing them, each multiplied by an associated weight
- Result: a single numerical value
- The resulting value is transformed into a value in $[0,1]$



HOW TO MAKE A PREDICTION



Representation of information

4500 values in $\{0, \dots, 255\}$

f

Overall: 4501 parameters

THE BEST POSSIBLE FUNCTION

Each of the possible functions:

- Is characterized by the values of the 4501 weights.
- When applied to the set of examples, results in a certain number of errors

Choose the function with minimum number of errors

Also, choose the values of the 4501 weights whose corresponding function has minimum number of errors

AND IF THERE ARE MORE POSSIBLE CASES?

IDENTIFYING DIGITS

WHAT CAN WE DO?

- We first define a "model" of the digits
- An image: $28x28$ dots (pixel), each with an associated integer value from 0 (white) to 255 (black)
- An image: sequence of $28x28=784$ integer values $x_1, x_2, x_3, \ldots, x_{784}$ each one in $[0,255]$
- What are we looking for? A function from $[0,255]^{784}$ to $\{0,1,2,3,4,5,6,7,8,9\}$ making as few errors as possible on the given set of examples

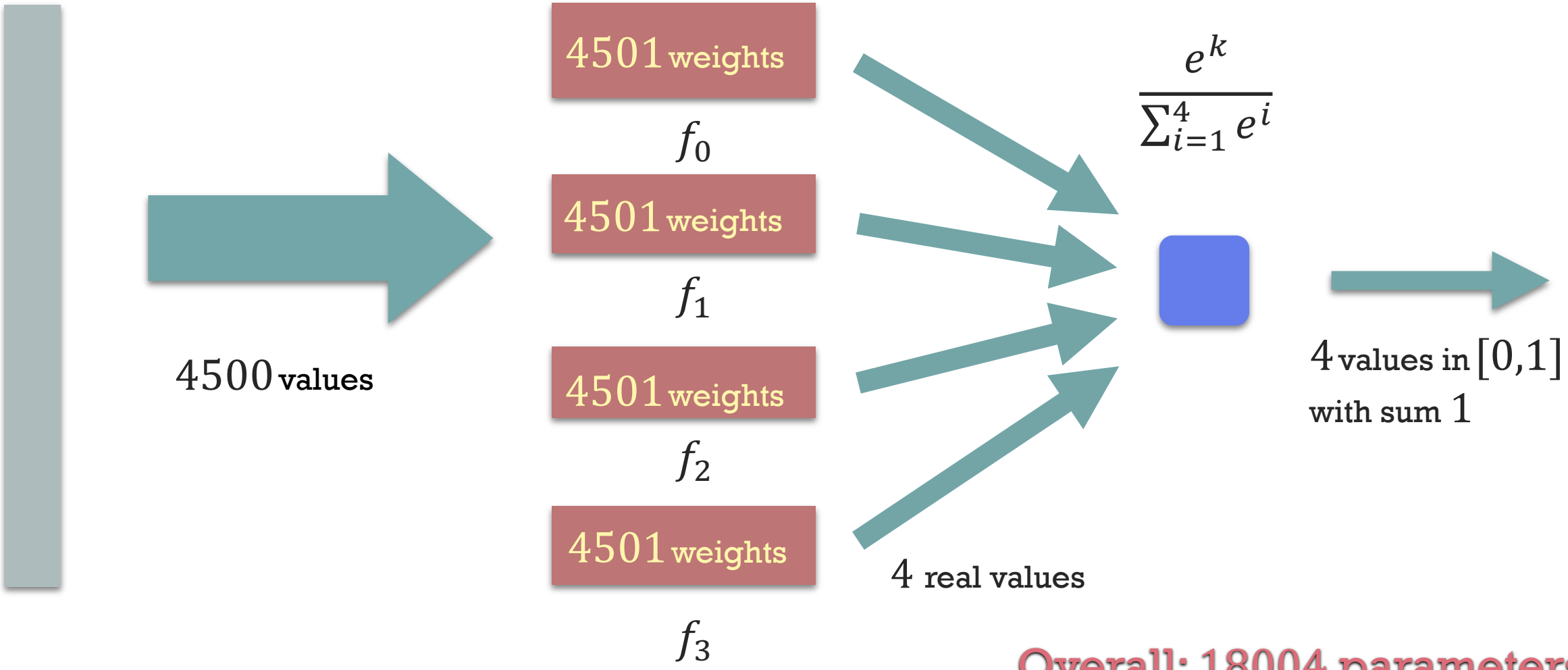
WHAT CAN WE DO?

We search 10 functions $f_0, f_1, ..., f_9$ from $[0,255]^{784}$ to $[0,1]$ with the property that the sum of the 10 values they return when applied to the same argument is equal to 1.

For any image x=[0,3,78,167,…,35,2] (784 values)

- We interpret $f_0(x)$ as the probability that the digit in x is a 0, $f_1(x)$ as the probability that it is a 1, and so on
- We predict that the digit in the image is the one with highest computed probability

HOW TO MAKE A PREDICTION?



Representation of the information
(4500 integers in $\{0, \ldots, 255\}$)

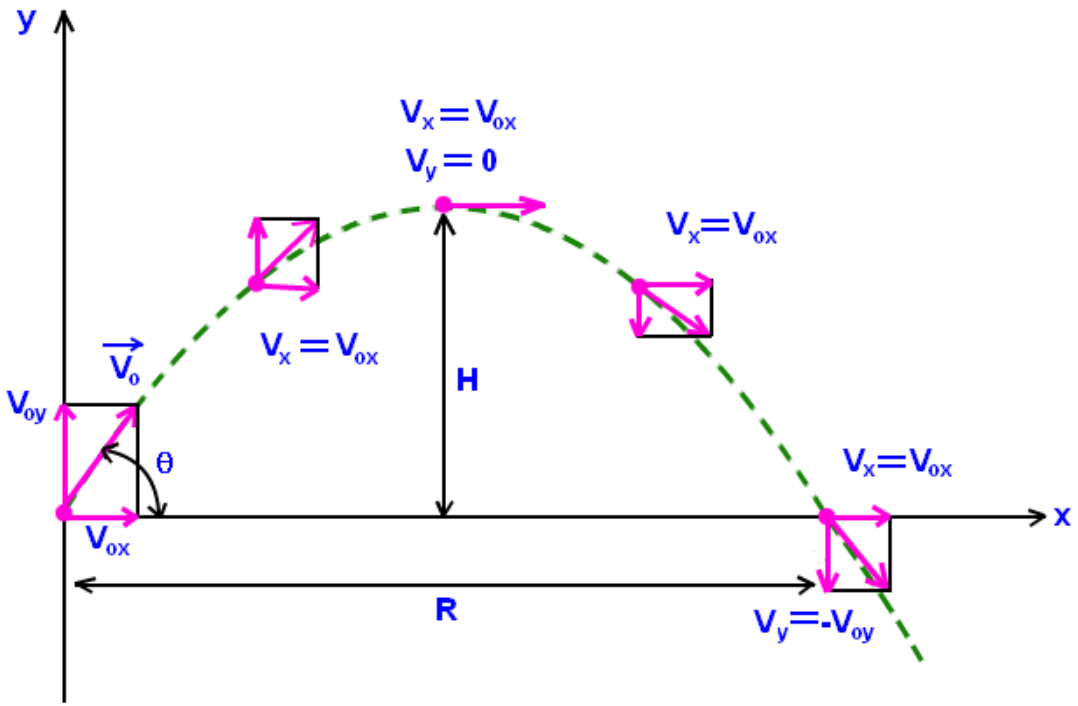
Overall: 18004 parameters

WHY MACHINE LEARNING?

SLIGHTLY MORE FORMALLY

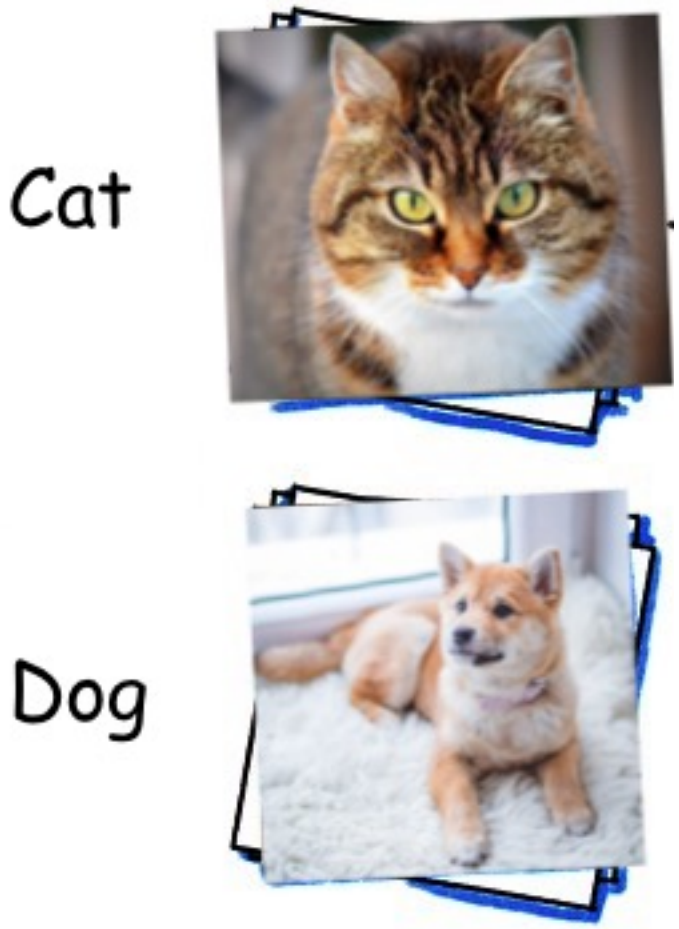
TYPES OF PROBLEMS: CLOSED VS OPEN

- **Closed problems (well-defined):**
  - The rules and solution methods are known
  - Solvable with traditional programming or equations
- **Examples:**
  - *Calculating the trajectory of a projectile (physics formulas)*
  - *Count the number of i-s in a text*
  - *Sorting a list of numbers*
  - *Solving a system of linear equations*



TYPES OF PROBLEMS: CLOSED VS OPEN

- **Open problems (ill-defined):**
  - No explicit rules, models or equations available
  - The "definition" is fuzzy or based on perception
- **Examples:**
  - *Recognizing a cat in a photo*
  - *Translating a sentence from one language to another*
  - *Predicting the outcome of a medical treatment*
  - *Detecting spam emails*



- **What do these have in common?**

  They **cannot** be solved by formulas or handcrafted rules alone

  They *require learning patterns from data.*

FROM RULE-BASED TO DATA-DRIVEN SOLUTIONS

- **Traditional programming:**
  - Programmer defines all rules and logic
  - Works well when the process is fully understood and formalizable
- **Rule-based systems** quickly become brittle or unmanageable in complex, evolving domains:
  - *Spammers invent new tricks every day.*
  - *Tumors can have endless visual variations.*
  - *Human language is ambiguous, context-dependent*.

FROM RULE-BASED TO DATA-DRIVEN SOLUTIONS (2)

- **Machine Learning** adapts to scenarios by learning from examples and feedback.
  - Learns rules/patterns **from data**
  - Essential when explicit rules are *too complex, numerous,* or *unknown*
  - Can approximate solutions to "open" problems
- **The Power of Generalization**
  - ML models **extract structure** from past examples and **generalize** to new, unseen cases.
  - Not just "memorizing" the training data, but discovering *underlying usually implict regularities*.
  - **Example:** A model trained to ***recognize positive or negative sentiment in text*** can correctly classify messages that use new expressions it has never seen before.
    - For instance, even if it never encountered the word *"amazing"* when observing annotated material, it can infer that *"This movie was absolutely amazing!"* is positive, based on patterns learned from other similar words and contexts.

FROM PROBLEMS TO DATA-DRIVEN «SOLUTIONS»

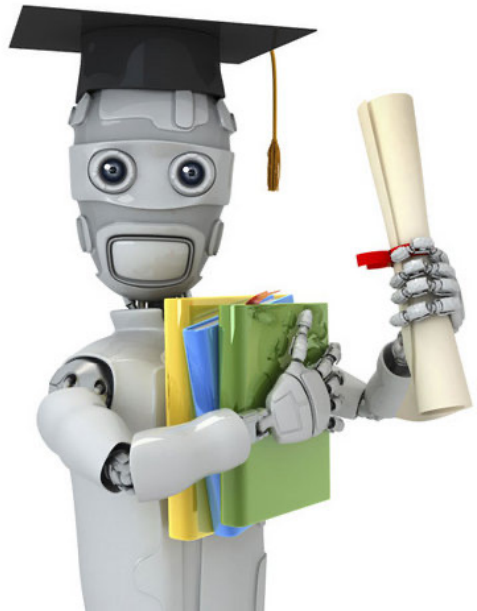
WHAT DOES IT MEAN FOR A PROGRAM TO LEARN?

"*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**.*" *(Tom M. Mitchell, 1997)*

Why start here?

- It highlights three pillars of any learning problem:
  - **The task** (**T**): *What* is to be learned or predicted?
    - e.g., classifying images, translating sentences)
  - **The experience** (**E**): *Where* does learning come from?
    - e.g., data, examples, feedback
  - **The performance measure** (**P**): *How* do we evaluate success?
    - e.g., accuracy, error rate

So…



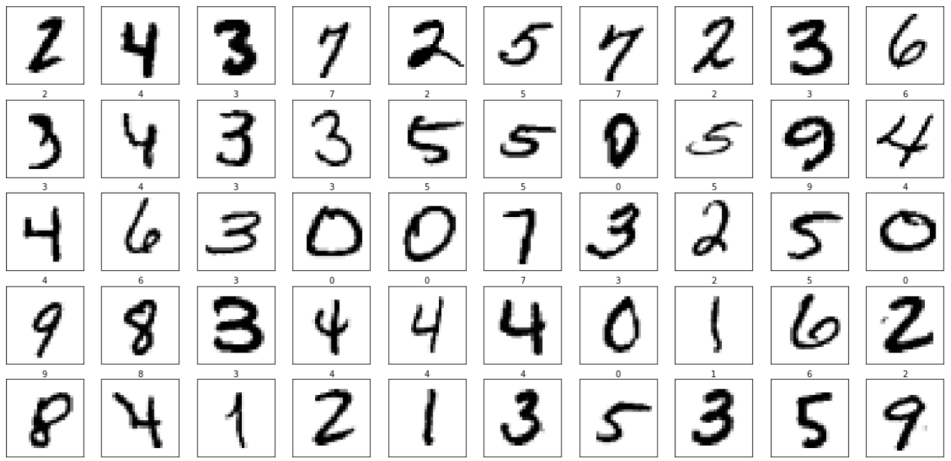
IT IS MORE THAN JUST CHOOSING AN ALGORITHM (ON GITHUB)…

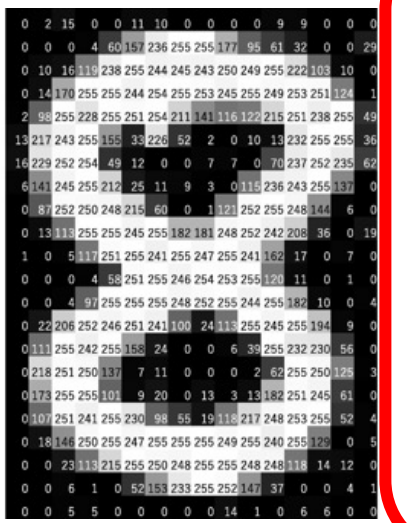
- We move from a real-world question-often ill-defined and without a formula-to a sequence of logical steps that transform observations into reliable predictions.
- Along this journey, we face crucial decisions:
  - What data best **describes our experience about the problem/task**?
  - How do we **represent information** so a computer can "understand" it?
  - Which **family of models** is best suited to capture the relevant patterns-and how do we compare alternatives?
  - How do we "train" a model to **extract knowledge** from the experience (examples)?
  - How do **we measure success**, and-most importantly-ensure our solution works on new, unseen situations?

- **A well-designed workflow is not just a technical necessity:**
  - **it is the bridge between human intuition, domain knowledge, and the automated discovery of patterns.**

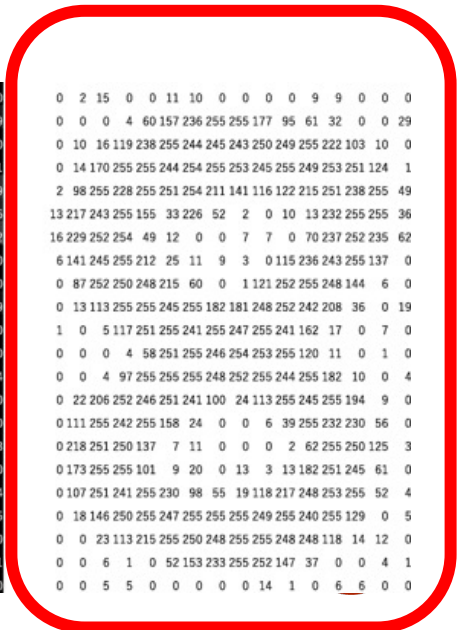
OBSERVING THE WORLD: BUILDING A DATASET

- **Digit recognition** means automatically classifying images as one of the digits from 0 to 9.
  - Start by collecting **examples from reality**: gather images, each labeled with the correct digit (0–9).
- Each example becomes a pair: **(input, output)**
  - *Input*: the image (a 28x28 pixel grid)
  - *Output*: the digit label





This is a 784-dimensional vector!

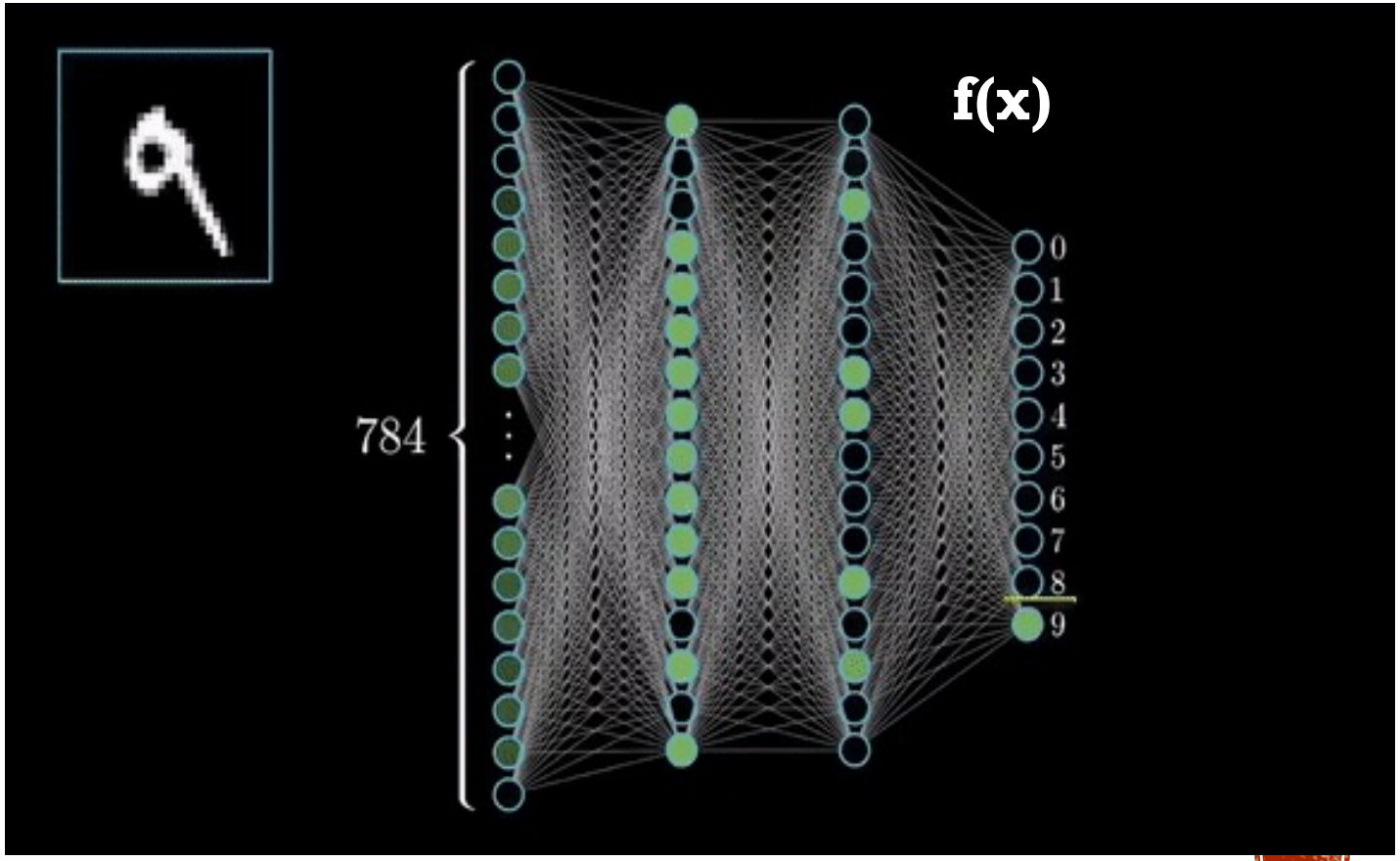


FROM EXAMPLES TO FUNCTIONS: FORMALIZING THE TASK

- We want to find a function **f(x)** that maps an input (image) to the correct output (digit).
- The challenge:
  - There is no explicit formula for "digit-ness."
  - The function must **separate examples** of different classes based on features.

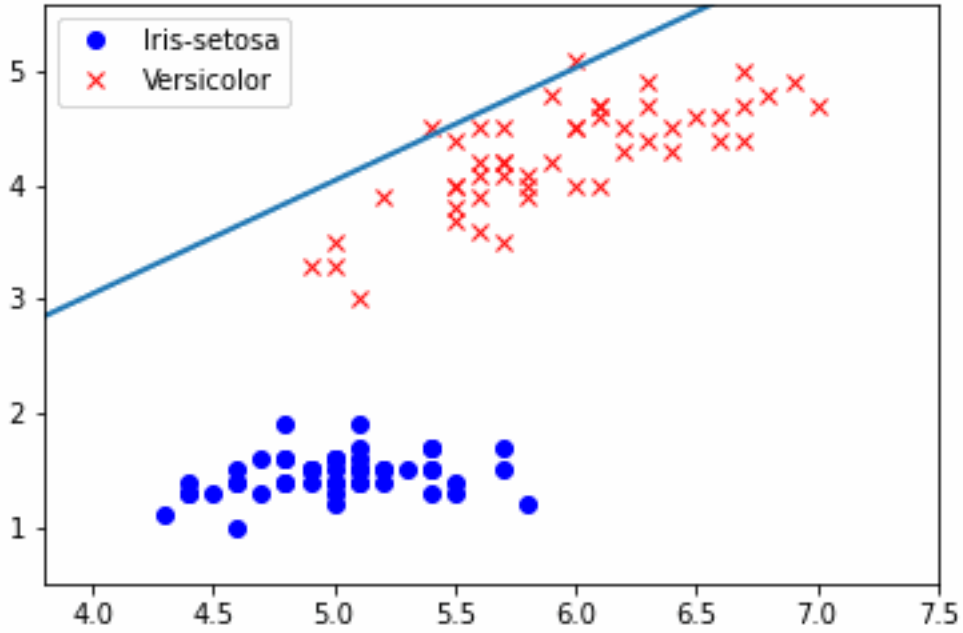
REPRESENTATION: HOW DO WE ENCODE THE PROBLEM?

- **Input representation:**
  - Each image is turned into a vector of 784 pixel values.
  - Sometimes, additional features can be engineered (symmetry, curvature, etc.)
- **Output representation:**
  - Class labels (0, 1, …, 9), often encoded as "one-hot" vectors for neural networks.



LEARNING: FINDING THE BEST PREDICTIVE FUNCTION

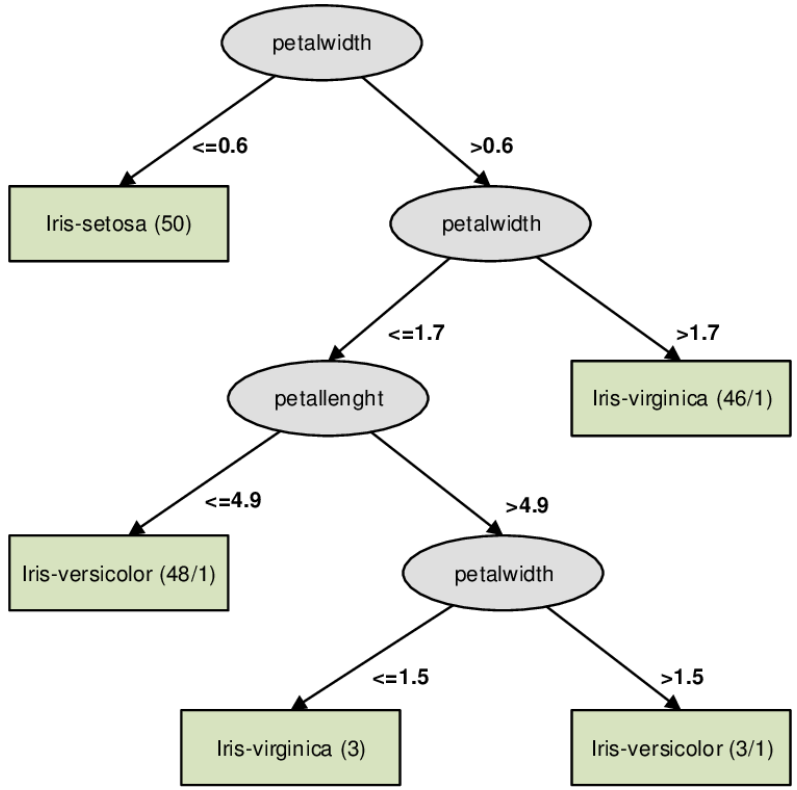
- Instead of designing **f(x)** by hand, we let the computer **learn** its parameters from examples.
- **Model choice:**
  - Could be a linear model, a tree, or a deep neural network.
- **Learning algorithm:**
  - Adjusts the function parameters to minimize errors on the *training examples*.
  - The process is *data-driven*: the more varied and informative the examples, the better the learning.



f(x) is a linear function
y = ax + b
learned to separate two classes,
where **a** and **b** are the parameters
to be learned from data

LEARNING: FINDING THE BEST PREDICTIVE FUNCTION

- Instead of designing **f(x)** by hand, we let the computer **learn** its parameters from examples.
- **Model choice:**
  - Could be a linear model, a tree, or a deep neural network.
- **Learning algorithm:**
  - Adjusts the function parameters to minimize errors on the *training examples*.
  - The process is *data-driven*: the more varied and informative the examples, the better the learning.

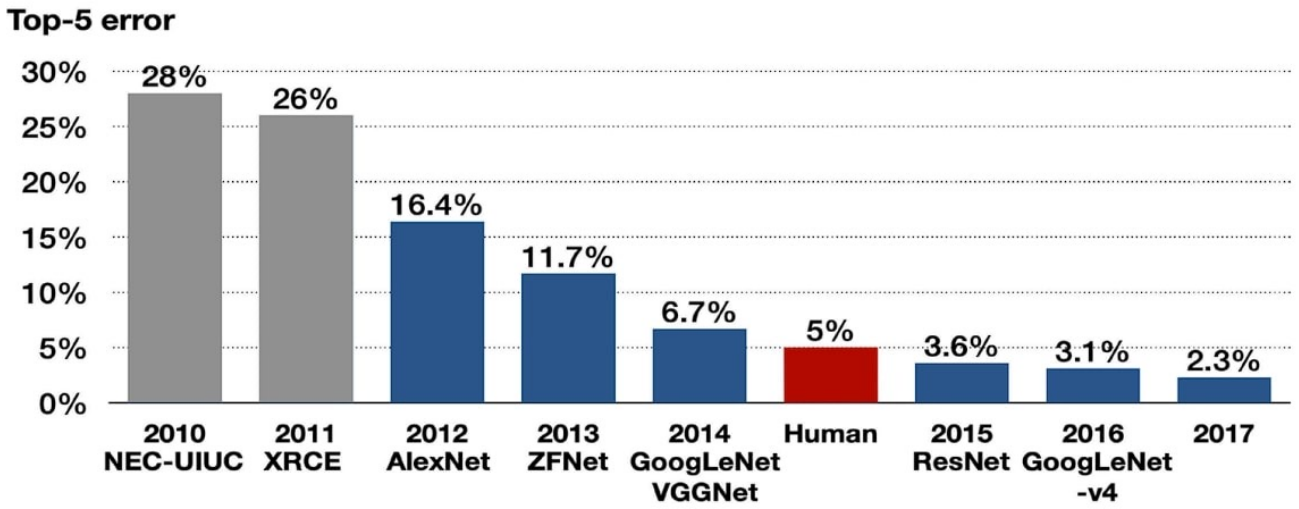


GENERALIZATION: THE TRUE TEST OF LEARNING

- The ultimate goal:
  - **Accurate predictions on new, unseen data.**
- After training, we **test the model** on **examples never used before**.
- This "out-of-sample" evaluation is unique:
  - In ML, success is measured not (*only*) by proof, but (*also*) by predictive performance on data never seen before.
- Only a model that ***generalizes*** well can be considered truly successful.

THE IMPORTANCE OF EVALUATION IN MACHINE LEARNING

- **Systematic evaluation** enables the whole field to compare models fairly, track progress over time, and push the boundaries of what is possible.
- *ImageNet* is a benchmark that transformed computer vision:
  - For years, classical approaches stagnated with high error rates.
  - The introduction of **AlexNet (2012)**-the first deep neural network to compete-reduced error dramatically, setting off the "deep learning revolution."



WHY STUDY MACHINE LEARNING, AND WHY THIS COURSE?

- **Machine Learning** powers today's AI revolution-driving innovation in healthcare, finance, industry, cybersecurity, and beyond.
  - The ability to transform raw data into actionable insights is now essential across science, technology, and business.
  - ML bridges *mathematical rigor* and *real-world impact*, requiring both **solid foundations** and **hands-on experience**.
- **This course provides:**
  - **Essential skills** for modern research and industry:
  - Problem formalization, data representation, and model development
  - Model selection, comparison, and validation
  - Critical evaluation of uncertainty and generalization
- **Designed for students who want to:**
  - Become data scientists, AI engineers, or researchers
  - Build the next generation of intelligent systems ;-)

COURSE STRUCTURE & LEARNING GOALS

LEARNING OBJECTIVES & OUTCOMES

- **Main learning goals:**
  - Acquire a solid **theoretical understanding** of machine learning principles and mathematical foundations.
  - Master the most **important algorithms** for supervised and unsupervised learning.
  - Gain **practical experience** with real-world datasets and problem-solving.
  - Develop **critical assessment skills** for evaluating and selecting ML models.

- **Expected outcomes (Dublin descriptors):**
  - **Knowledge & Understanding:** Understand key ML concepts, algorithms (e.g., Decision Trees, SVM, Neural Networks), and their mathematical basis.
  - **Applying Knowledge:** Design, implement, and validate ML models for complex data analysis tasks.
  - **Making Judgements:** Critically compare models, evaluate performance, and justify methodological choices.
  - **Communication Skills:** Clearly present solutions, results, and implications of ML projects.
  - **Lifelong Learning:** Ability to autonomously update and expand knowledge in this rapidly evolving field.

PREREQUISITES

- **Mathematical background:**
  - Basic knowledge of **linear algebra** (matrices, vectors, eigenvalues)
  - **Calculus** (derivatives, optimization)
  - **Probability and statistics** (random variables, distributions, expectations)
- **Programming skills:**
  - Proficiency in at least one programming language (**Python** recommended)
  - Basic understanding of data structures and algorithms
- **Motivation and learning attitude:**
  - Willingness to engage in **project work** and collaborative problem-solving
  - Openness to both **theoretical study** and **practical experimentation**

COURSE STRUCTURE

- **Course workload:**
  - **Total credits:** 9 CFU (~72 hours including lectures & labs)
- **Delivery:**
  - Interactive lectures (theory & methods)
  - Hands-on laboratory sessions (Python, real datasets)
  - Group project and case studies
- **Teaching approach:**
  - Student interaction and discussion
  - Problem-based learning
  - Emphasis on both mathematical foundations and application

MAIN MODULES & TOPICS

1. **Foundations**
   - Conceptual framework,
   - Types of approaches to learning
   - Introduction to statistical learning theory
   - Probabilistic learning, bayesian approaches
2. **Supervised Learning**
   - Loss functions, risk minimization
   - Generative/discriminative models, logistic regression
   - Evaluation of ML systems (model optimizations)
   - Regression (linear, nonlinear), regularization
   - Linear classifiers: Fisher LDA, perceptron, Naive Bayes
   - SVMs, kernel methods
   - Multilayer perceptron
   - Decision trees
   - Ensemble models: bagging, boosting
   - Non parametric models for regression and classification
3. **Unsupervised Learning**
   - Clustering: k-means, mixture models
   - Dimensionality reduction: PCA, probabilistic PCA, factor analysis
4. **Introduction to Deep Learning**
   - Introduction to CNNs, RNNs,
   - The idea of Attention: Transformers

COURSE TIMELINE & MODULES OVERVIEW

- **Course start:** October 6, 2025
- **Duration:** 14 weeks (October – January)
- **Lab and project activities** are integrated throughout the semester.
- **Continuous interaction:** Theory, coding exercises, and discussions in each module.

TEACHING METHODS

- **Interactive lectures:** Clear exposition of theory, focus on intuition and mathematical rigor.
- **Hands-on labs:** Guided coding sessions (Python, scikit-learn, PyTorch) on real datasets.
- **Group projects:** Real-world problems, teamwork, presentation of results.
- **Case studies:** Applications in healthcare, finance, natural language, image recognition.

ASSESSMENT & RESOURCES

ASSESSMENT & EVALUATION

- **Assessment components & weighting:**
  - **Group project:** Application to a real dataset; written report (Jupyter notebook) and oral presentation
  - **Oral exam:** Discussion of theory and problem-solving
- **Evaluation criteria:**
  - Depth of **theoretical understanding**, ability to **implement and critically analyze ML models**, project quality, communication, and teamwork.
- **Exam schedule:**
  - **3 sessions per year:** January, June, September
  - **2 exam dates per session** (for a total of 6 opportunities per academic year)

EXAMPLE PROJECT / CASE STUDY

- **Practical group project:** Apply the entire machine learning pipeline to a real-world dataset.
  - **Examples:**
    - *Medical diagnosis from patient data*
    - *Credit scoring or fraud detection in financial datasets*
    - *Image classification*
- **Project workflow:**
  - **Problem definition & data exploration**
  - **Preprocessing & feature engineering**
  - **Model selection and training**
  - **Evaluation & model comparison**
  - **Interpretation and communication of results**
- **Skills developed:**
  - Data analysis and critical thinking
  - Implementation of supervised/unsupervised models
  - Collaborative research and scientific presentation

BIBLIOGRAPHY & LEARNING RESOURCES

- **Main textbooks:**
  - **Christopher M. Bishop**, *Pattern Recognition and Machine Learning*, Springer, 2006
  - **David Barber**, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, 2012
  - **Trevor Hastie, Robert Tibshirani, Jerome Friedman**, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2nd Edition, 2009
  - **Ian Goodfellow, Yoshua Bengio, Aaron Courville**, *Deep Learning*, MIT Press, 2016
- **Supplementary materials:**
  - Lecture slides and original notes provided by the instructor
  - **Jupyter notebooks** with practical exercises and lab code
  - Selected research papers and datasets (UCI, Kaggle, open data)
  - Online tutorials and videos for flipped classroom modules
- **Resources for further study:**
  - Scientific journals, preprints, and conference proceedings
  - Tools: scikit-learn, PyTorch, Colab

CLOSING & TAKEAWAYS

- **Course highlights:**
  - Comprehensive coverage of fundamental and advanced machine learning methods
  - Integration of theory, hands-on practice, and ethical considerations
  - Exposure to real-world data and state-of-the-art tools
  - Development of both critical and creative problem-solving abilities
- **Benefits for students:**
  - Preparation for academic research, PhD programs, or professional roles in data science, AI, and beyond
  - Ability to approach novel ML problems with methodological rigor and innovation
  - Communication and teamwork skills valued in both academia and industry
- **Strategic alignment:**
  - Course content and methods are consistent with leading international programs
  - Designed to support the reputation and objectives of the University of Rome "Tor Vergata" in the field of Computer Science

WEB SITE & TEAMS

- https://tvml.github.io/ml2526
- **MACHINE_LEARNING_2025-2026 class. Link from the web site**

- Questions?