From
Basic Machine Learning Models
to
Advanced Kernel Learning

Scott Pesme
Post-doc Inria Grenoble

Michaël Arbel
Researcher Inria Grenoble

Julien Mairal
Researcher Inria Grenoble

We work on the "theoretical aspects" of machine learning

= Institut National de Recherche en Informatique et en Automatique

Course structure

Scott Pesme
Post-doc Inria Grenoble

Michaël Arbel
Chercheur Inria Grenoble

Julien Mairal
Chercheur Inria Grenoble

Basic Machine Learning Models

Advanced Kernel Learning

2 x (~1h15 lecture) + 15 minutes break

Final grade: 50% exam + 50% homework

What to expect?

- Understanding of basic ML methods:
    - *Linear regression, linear classification, unsupervised learning, neural networks*
    - *Optimisation methods (gradient descent)*
- A deeper dive into kernel methods

You will mostly need pre-requisites from linear algebra!

Introduction

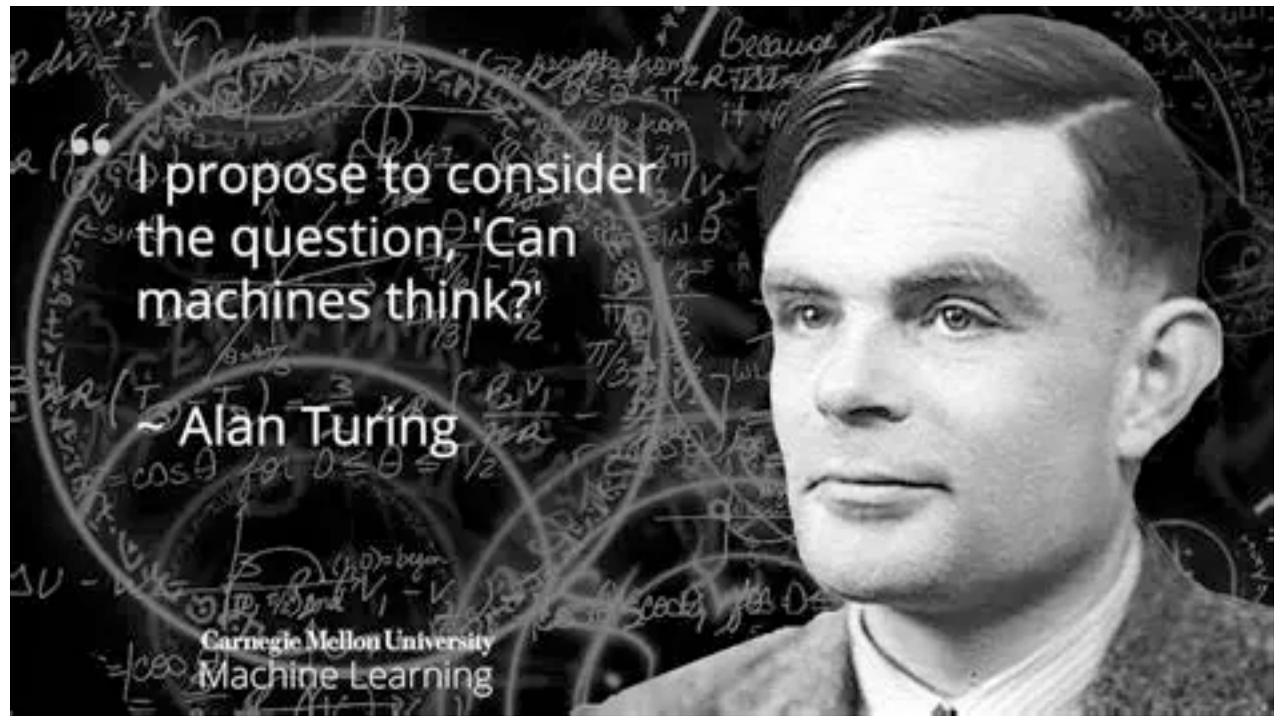
A bit of AI history:
the idea of thinking machines is not new!

https://www.youtube.com/watch?v=aygSMgK3BEM

What's going to happen to us if machines can think? Can they?

I don't really know, but you come back in 4 or 5 years I'll probably say "Sure they can think".

A bit of AI history



Dartmouth Conference
Coined the term "Artificial intelligence"

Deep learning breakthrough
AlexNet (2012), AlphaGo (2016)
LLMs

Origins

"Birth of AI"

1940s - 1950s

1956

1960s - 1970s

AI Winters

1990s-2000s

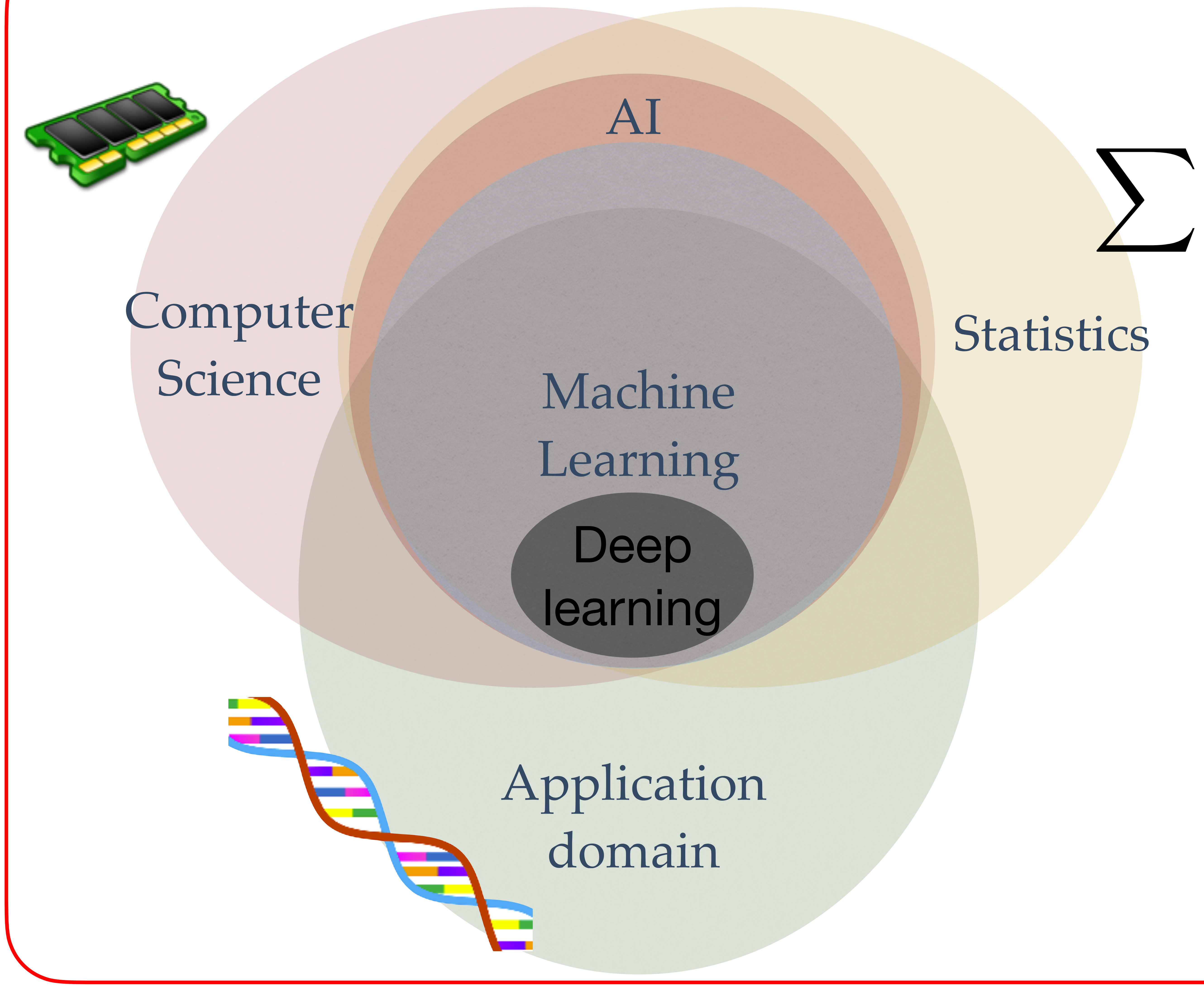
2010s - 2020s

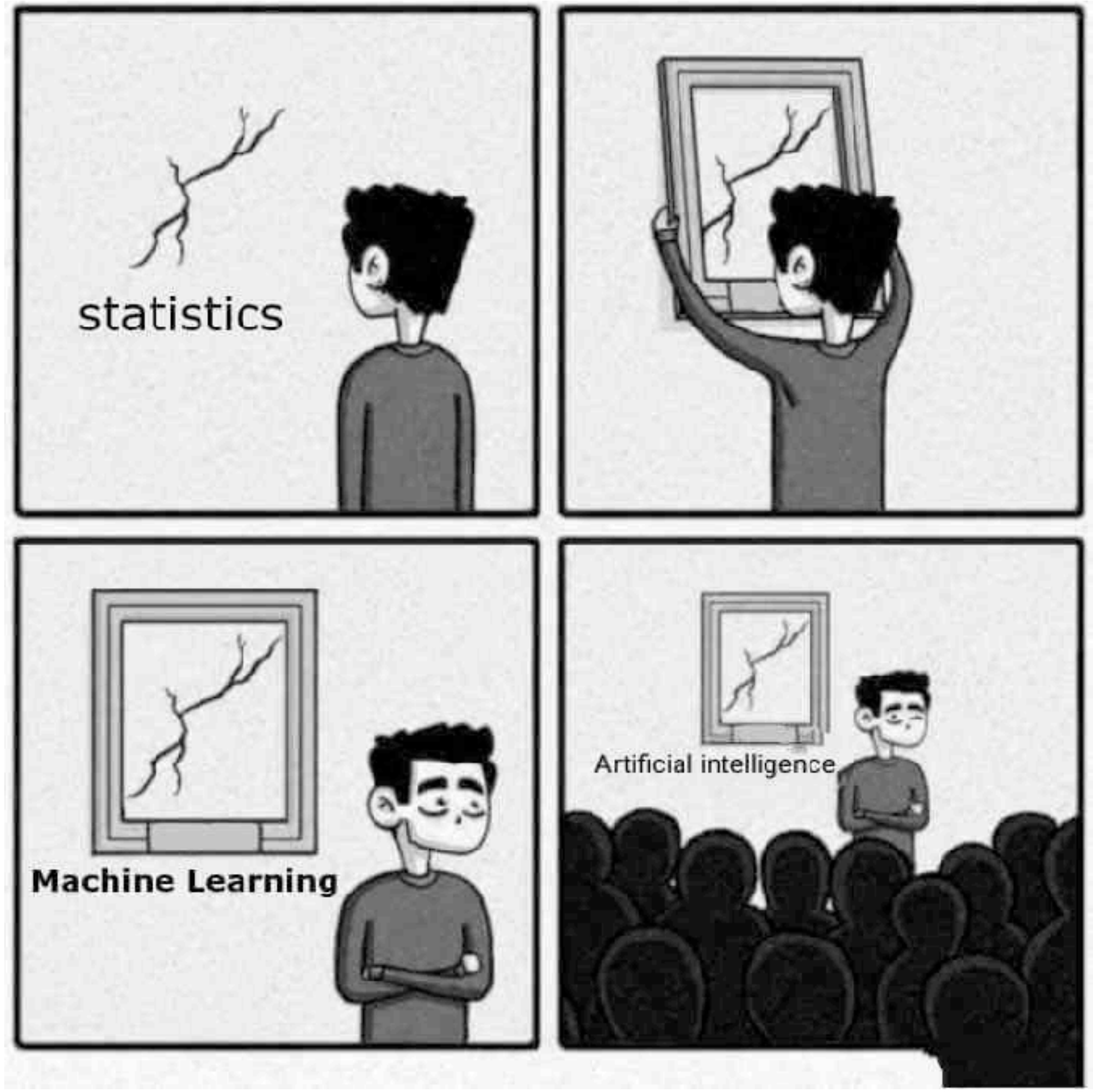
Theoretical foundations
Alan Turing: idea of "machine intelligence"
Turing test
Symbolic reasoning and neural networks

Expert systems

Practical breakthroughs
IBM's Deep Blue
SVM, Reinforcement learning
Search engines, recommendation systems, logistics

Some vocabulary





Back to the introduction

What is Machine Learning?

Field which develops
“machines” / “algorithms” / “functions”
that learn from data

How does it work?

Training phase:

Neural network at school

Data

‘Donkey’
‘Chicken’

After training:

Proficient neural network

‘Lagopède Alpin’

It learns ‘on its own’ to extract the important features

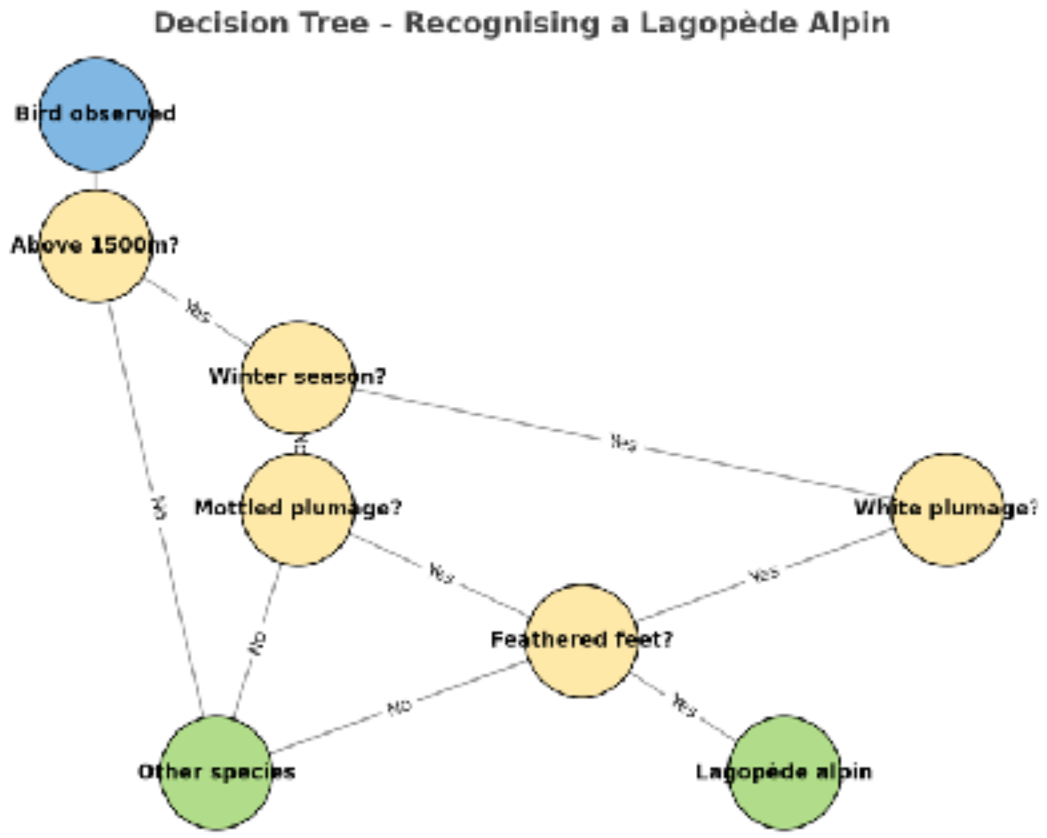
Machine learning vs expert systems

After training:

Proficient neural network

'Lagopède Alpin'

It learns 'on its own' to extract the important features



'Lagopède Alpin'

An "expert" gives a list of "if/then" rules

Pros and cons of two different "AI strategies"

Machine learning

Pros:

- No need to find the rules
- Adapts to data
- Works very well!!

Cons:

- Opacity (what the heck is going on??)
- Needs a "lot" of data

Expert systems

Pros:

- Transparent and understandable

Cons:

- Finding the rules is hard (impossible?)
- Doesn't work very well

Where is machine learning?

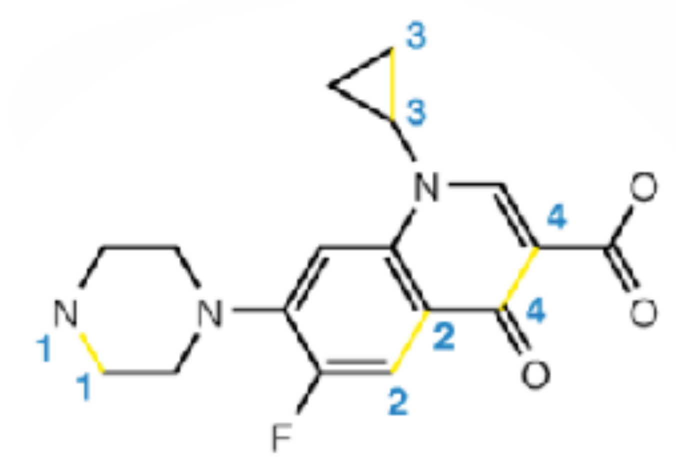
Text generation:

ChatGPT

MISTRAL AI_

Apple's photo app:

Drug discovery:



PlantNet:

Image generation:

But also:

Deep fakes

Scams

Mass surveillance

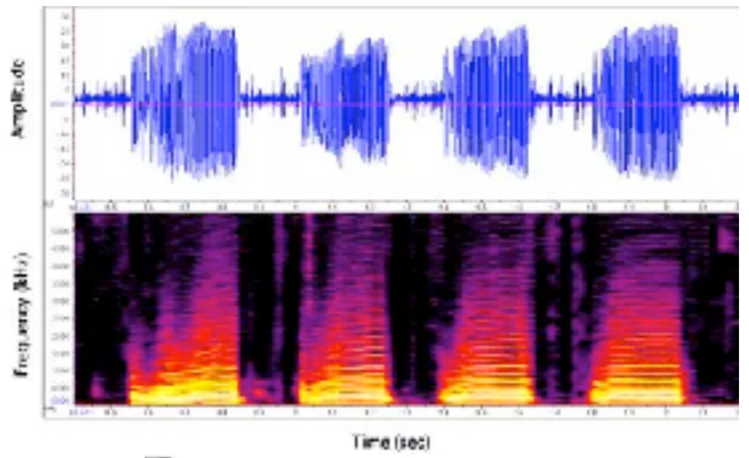
Learning a mapping / function from input to output

Input $x \in \mathbb{R}^d$

Text *What is the capital of France?*

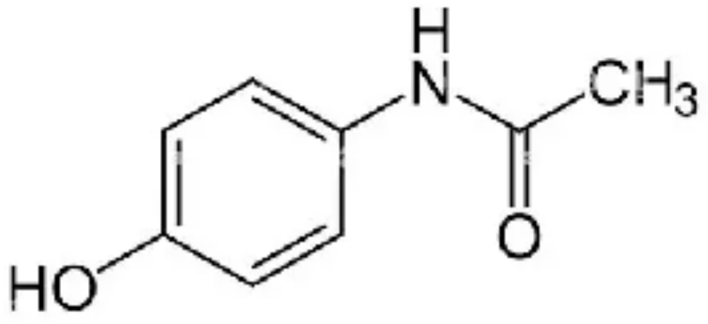
Image

Audio



Video

Graph



"Feature" vector (Height, age, income)

Trainable function

$$f_w(x) = \hat{y}$$

Output $y \in \mathbb{R}^p$

Text

Image

Audio

Video

Graph

"Feature" vector

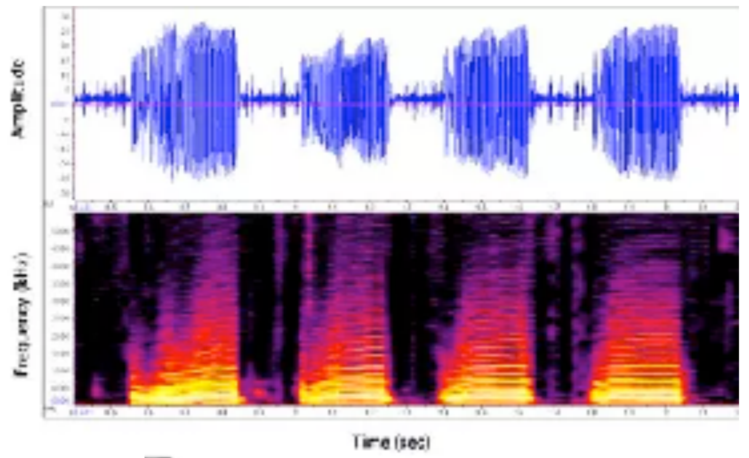
How is the input represented “inside the machine”?

Input

Text *What is the capital of France?*

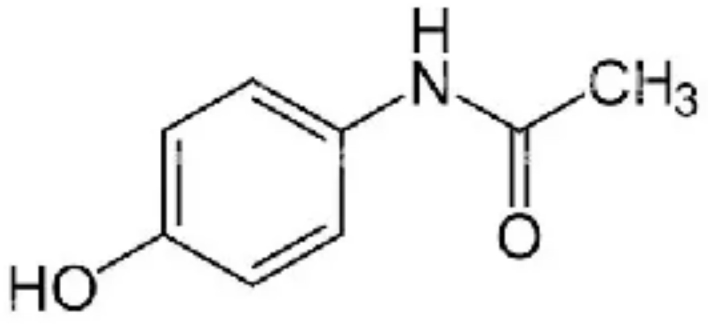
Image

Audio



Video

Graph

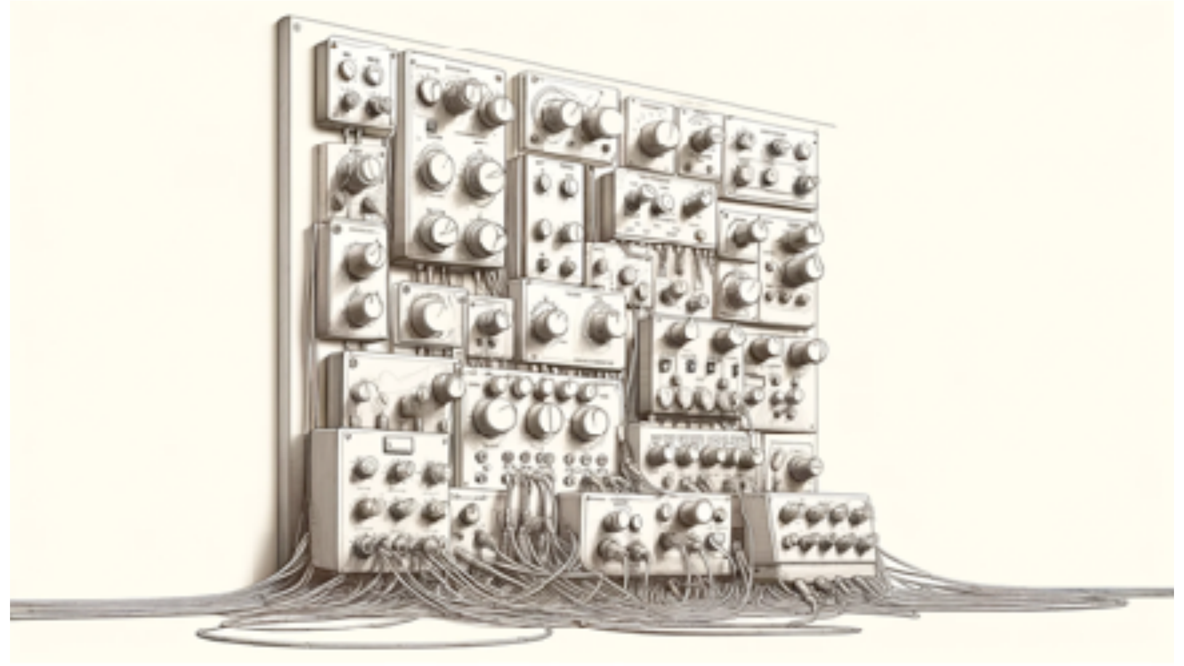


“Feature” vector (Height, age, income)

“Digital” representation

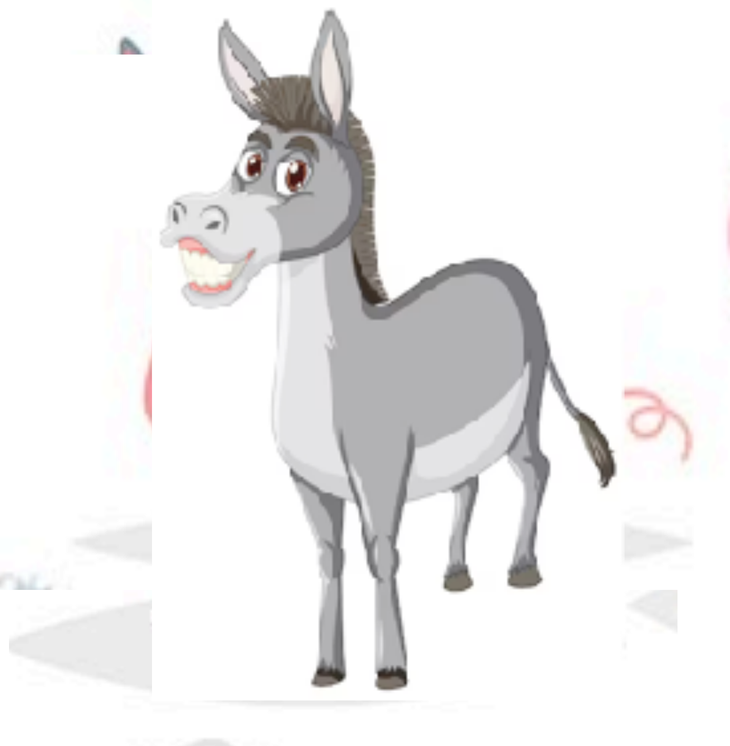
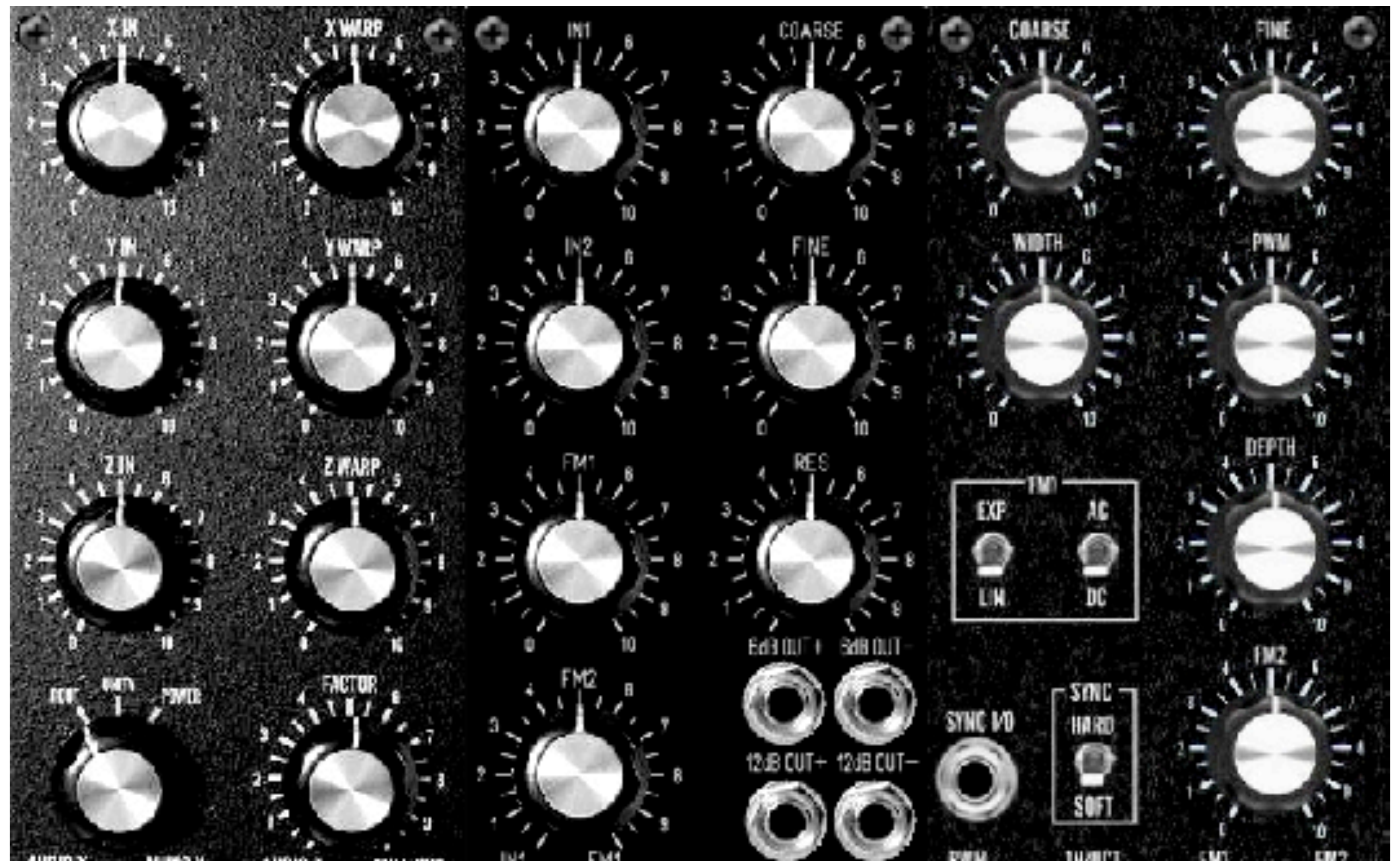
$$x = ???$$

How do we get a machine to learn?

The neural network  looks rather like: 

Training:

Neural network at school

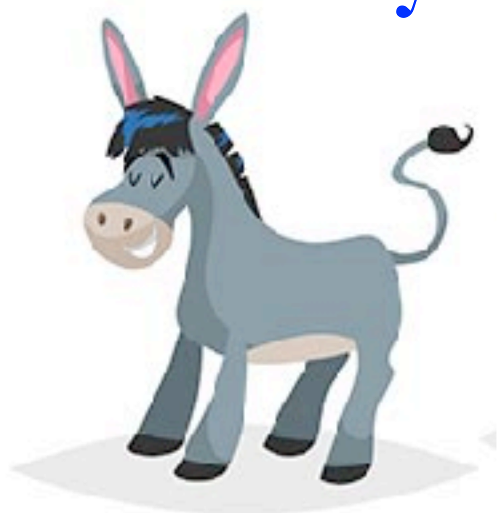
Data  

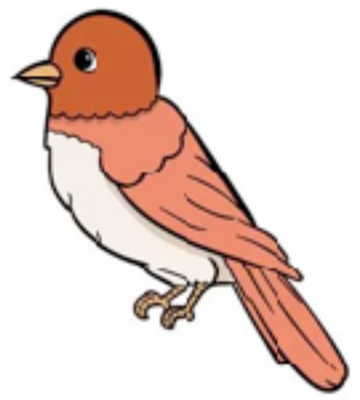
Training a neural network corresponds to turning knobs

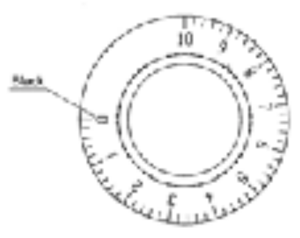
Same story, but with math

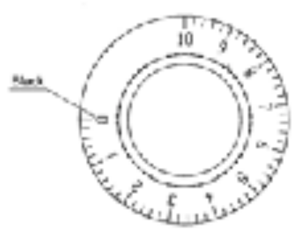
Two Possibilites

'Donkey' 'Bird'

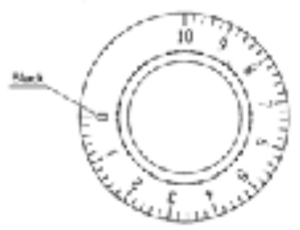








......



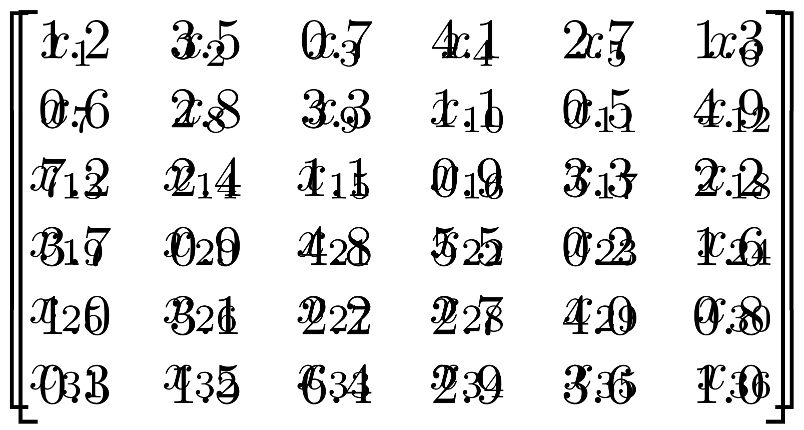
Knob 1 Knob 2 ...... Knob p

Machine Response

"Label": $+1$ -1

Magic synthesiser f_w

Pixel Intensity:



$x \in \mathbb{R}^d$

Rotation angle of the knobs

$$(3.2, \ -4.1, \ \ldots\ldots \ , \ 10.7)$$

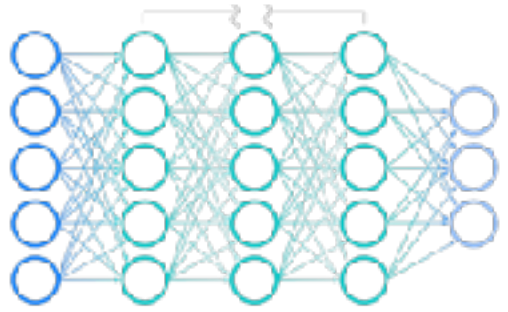
$$w = (w_1, w_2, \cdots, w_p) \in \mathbb{R}^p$$

$$\rightarrow f_w(x) \in \mathbb{R}$$

The magical synthetiser: neural networks

$$x \in \mathbb{R}^d \rightarrow f_w(x) = w_7 \max(0, w_1 x_1 + w_2 x_2 + w_3 x_3) + w_8 \max(0, w_4 x_4 + w_5 x_5 + w_6 x_6)$$

$$\rightarrow f_w(x) = W_L \phi(W_{L-1} \phi(\cdots \phi(W_1 x + b_1) \cdots) + b_{L-1}) + b_L.$$



The learning procedure, but with math

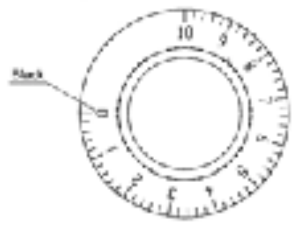
'Donkey'

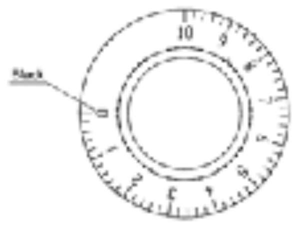
Label $+1$

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} & x_{16} & x_{17} & x_{18} \\ x_{19} & x_{20} & x_{21} & x_{22} & x_{23} & x_{24} \\ x_{25} & x_{26} & x_{27} & x_{28} & x_{29} & x_{30} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \end{bmatrix} \longrightarrow$$

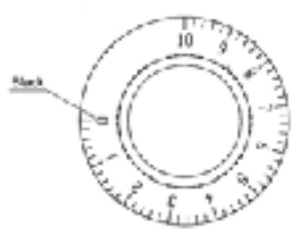
$x \in \mathbb{R}^d$

Magic synthetiser f_w





......



Knob 1 Knob 2 ...... Knob p

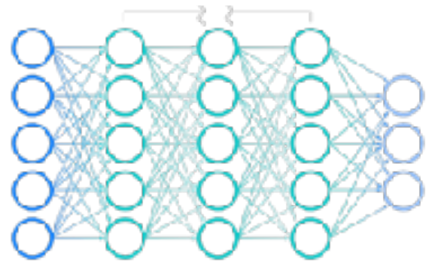
$w = (w_1, w_2, \cdots, w_p) \in \mathbb{R}^p$

$\longrightarrow$ Machine Response

$f_w(x) \in \mathbb{R}$

Magic synthesisers: neural networks

$$x \longrightarrow f_w(x) = w_7 \, \max(0, w_1x_1 + w_2x_2 + w_3x_3) + w_8 \, \max(0, w_4x_4 + w_5x_5 + w_6x_6)$$



How can we get it to learn?

Before training, it predicts $f_w(x)$ (-234.5), and we want it to predict "Donkey" $(+1)$

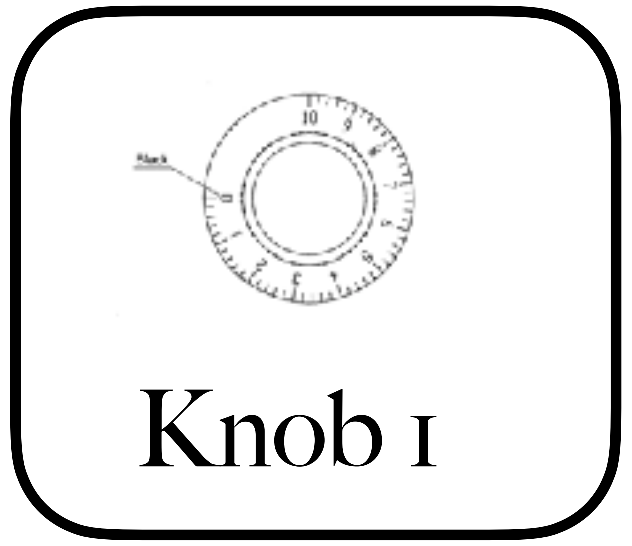
We can model it's mistake by $L(w) = (f_w(x) - 1)^2$

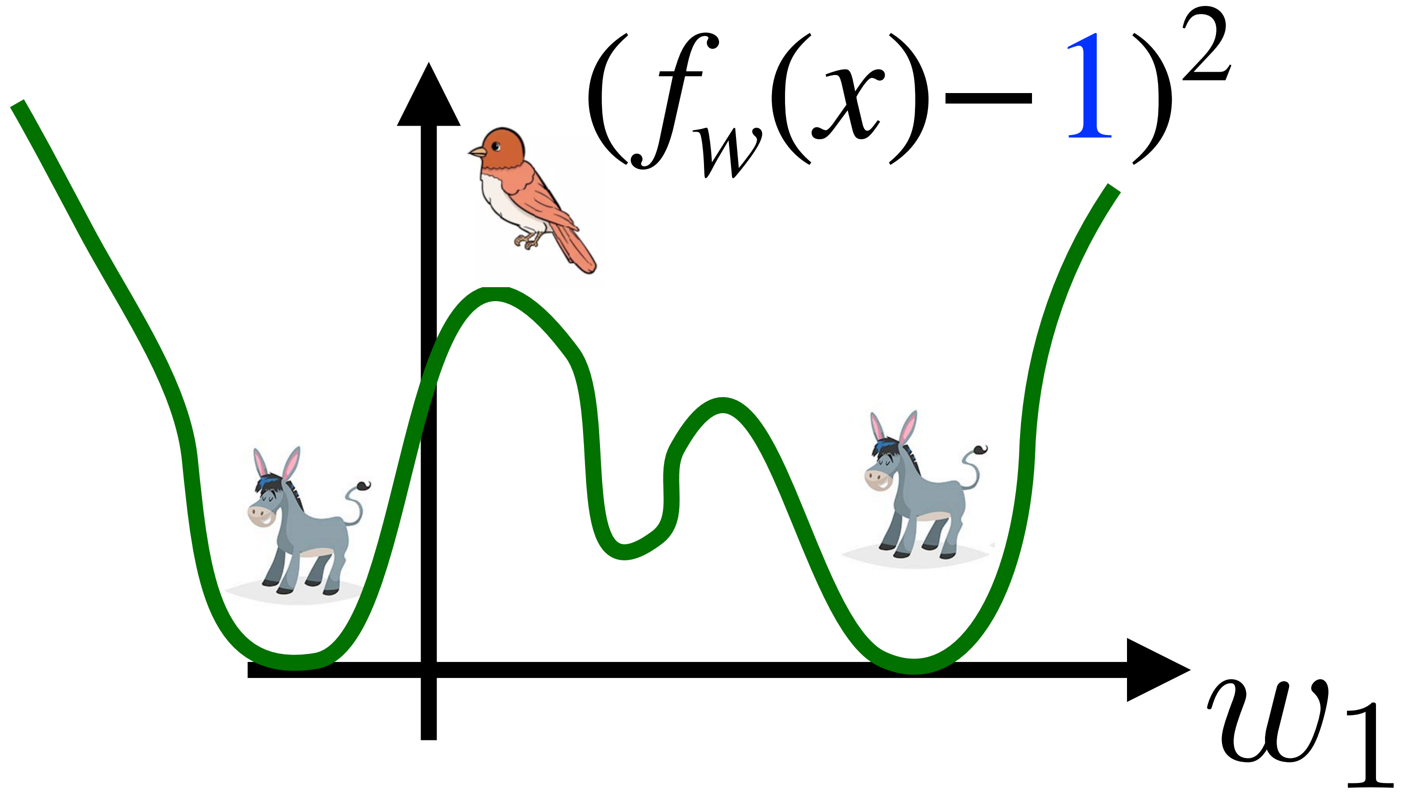
We want to minimise the mistake! $\min_{w \in \mathbb{R}^p} (f_w(x) - 1)^2$

It boils down to minimising a function!

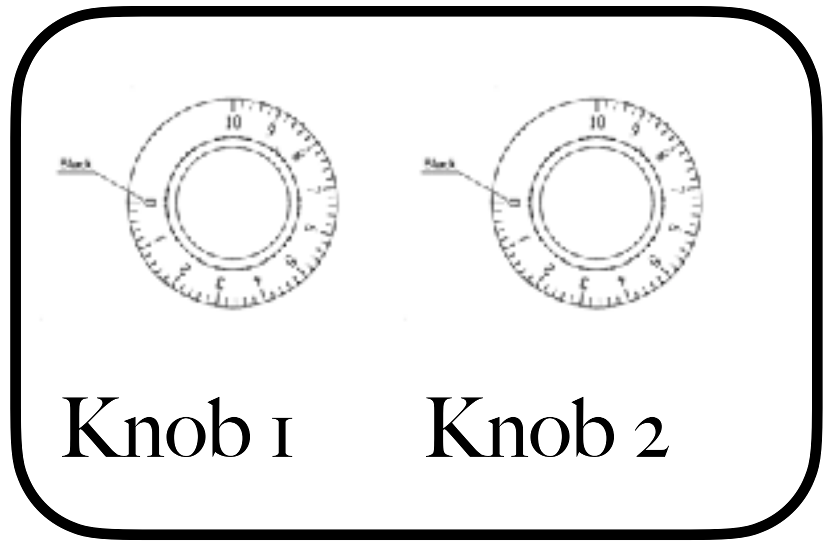
$$\min_{w \in \mathbb{R}^p} (f_w(x) - 1)^2$$

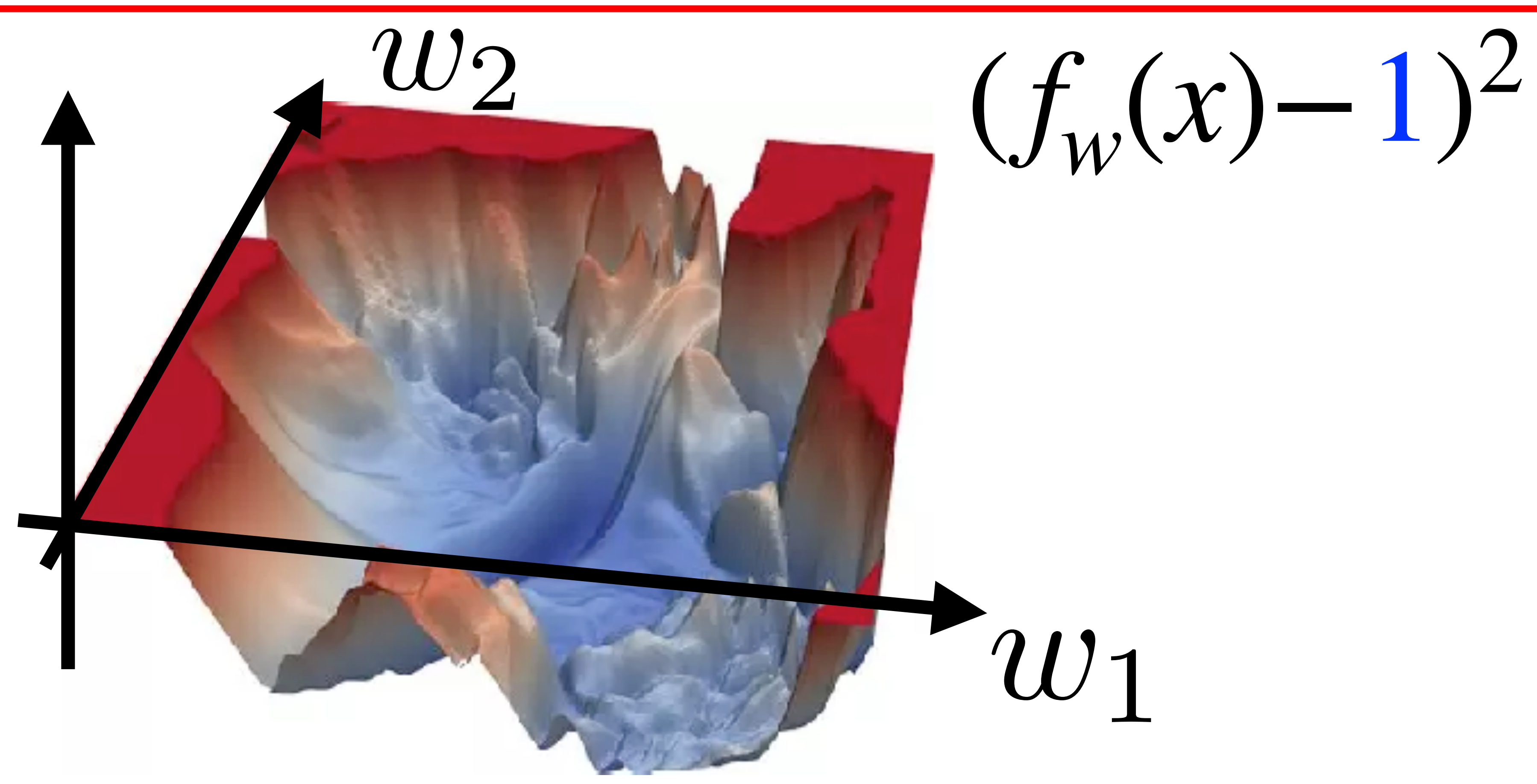
$$w = (w_1)$$

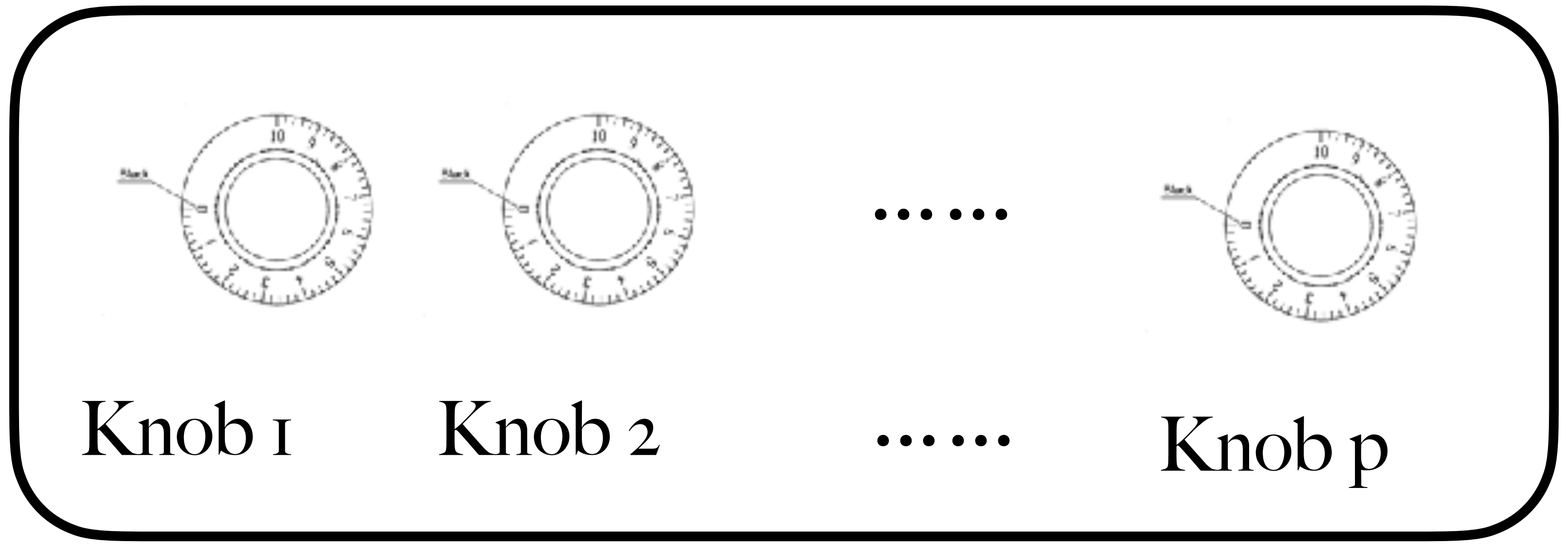


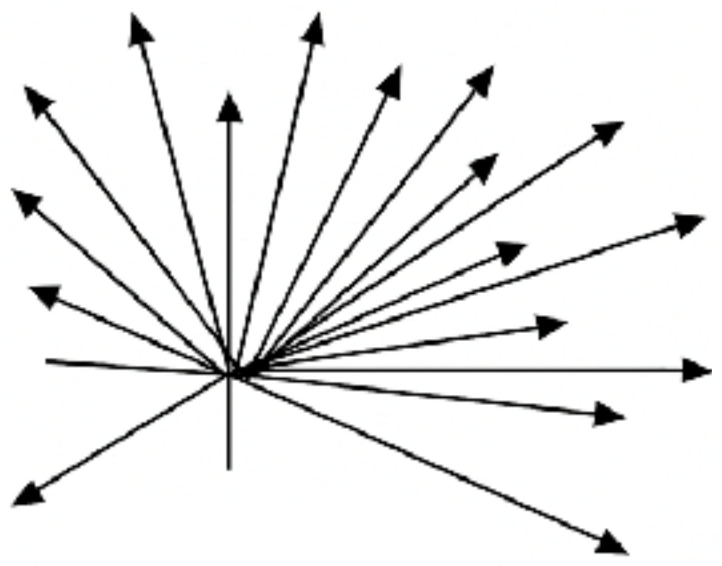


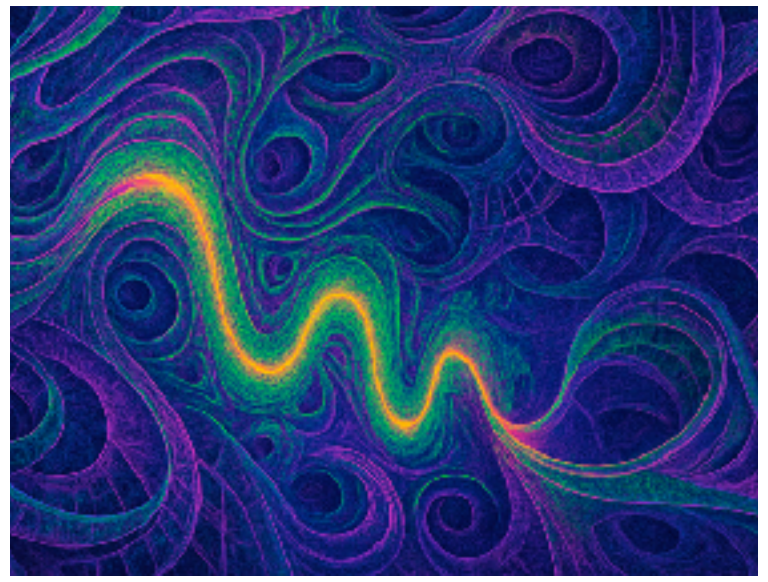
$$w = (w_1, w_2)$$





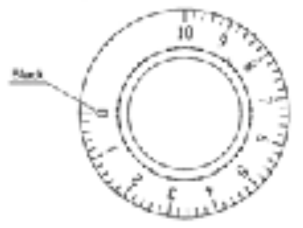






It boils down to minimising a function! $\min_{w \in \mathbb{R}^p} (f_w(x) - 1)^2$

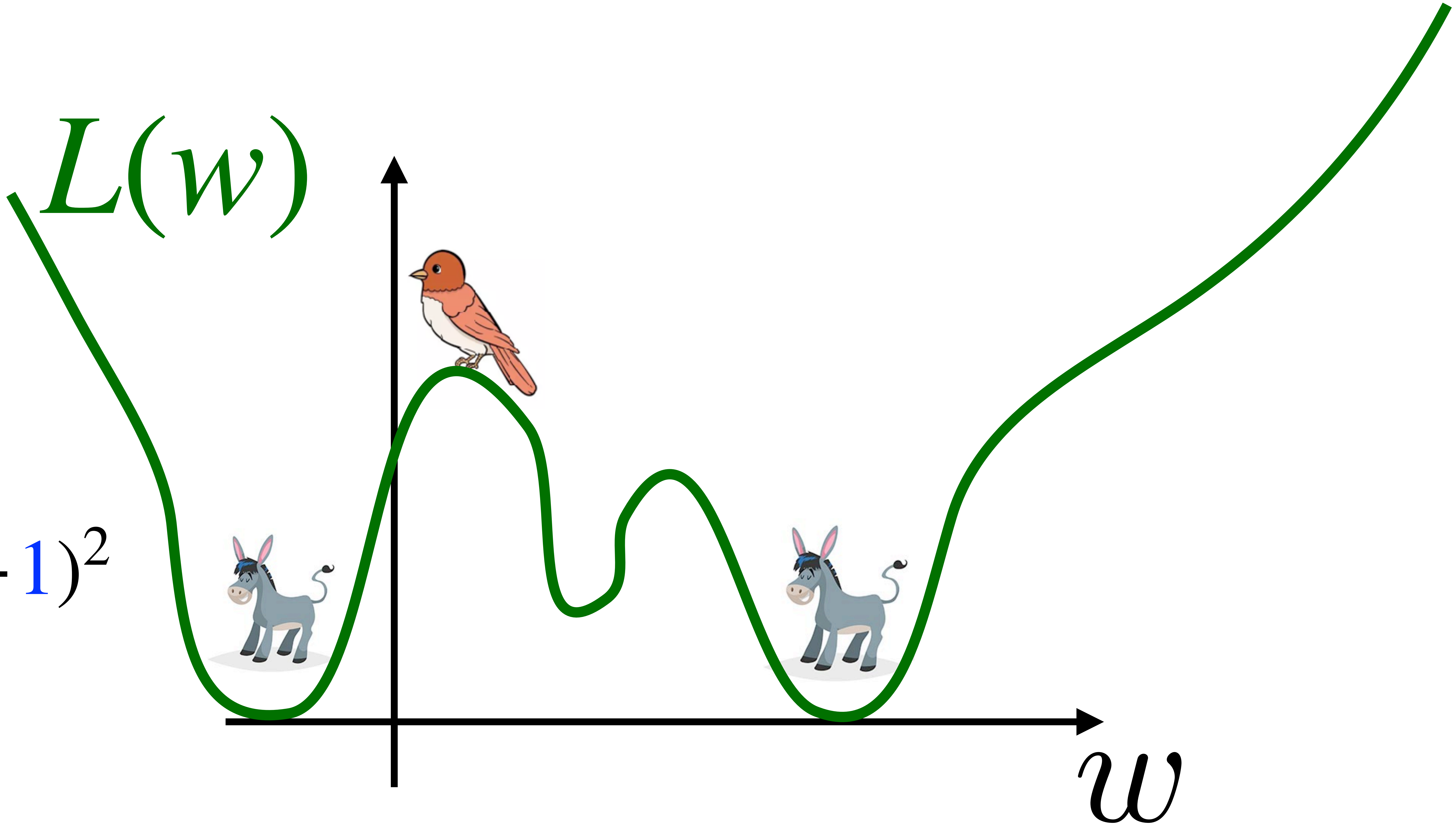
Back to 1D:



Knob 1

$$\min_{w \in \mathbb{R}^p} L(w)$$

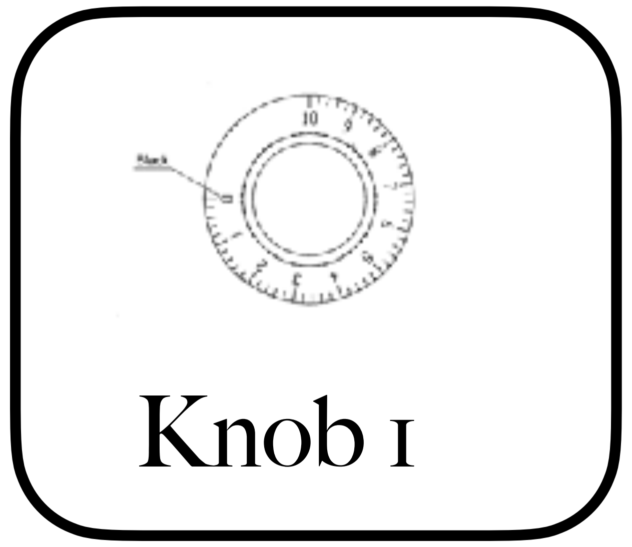
$$L(w) = (f_w(x) - 1)^2$$



It boils down to minimising a function!

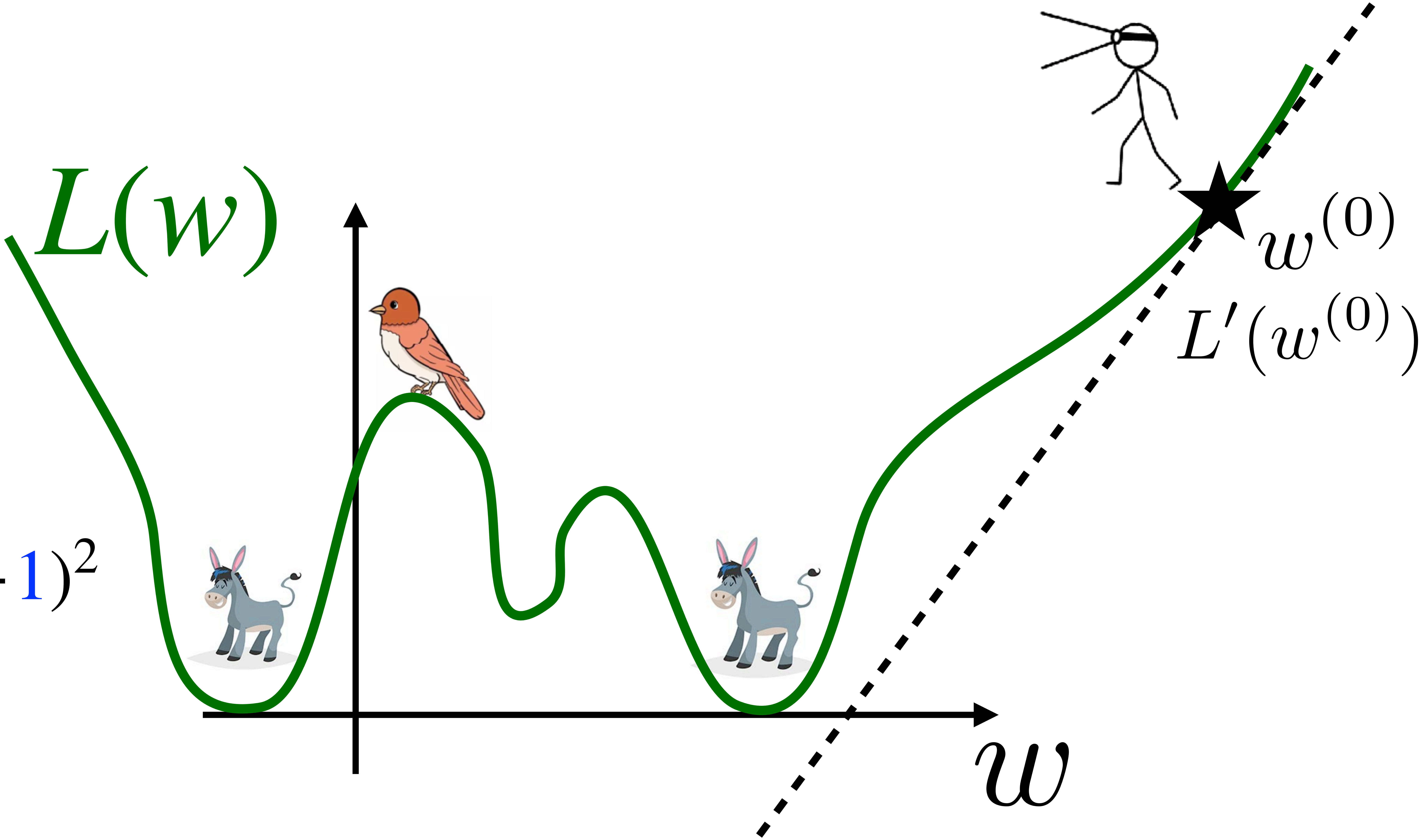
$$\min_{w \in \mathbb{R}^p} (f_w(x) - 1)^2$$

Back to 1D:



$$\min_{w \in \mathbb{R}^p} L(w)$$

$$L(w) = (f_w(x) - 1)^2$$

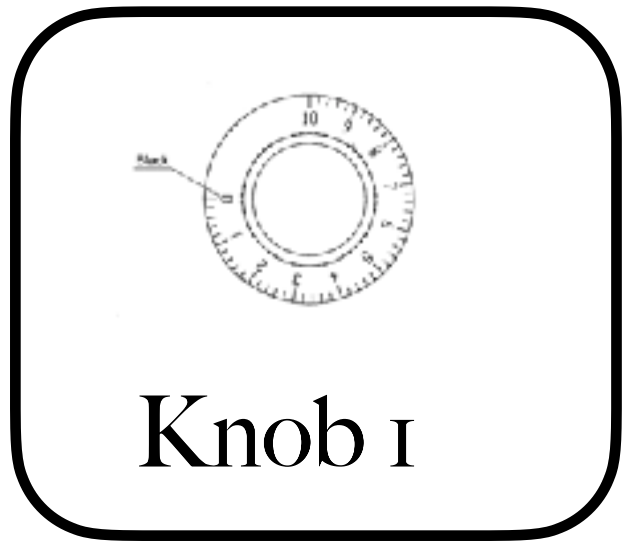


It boils down to minimising a function!

$$\min_{w \in \mathbb{R}^p} (f_w(x) - 1)^2$$

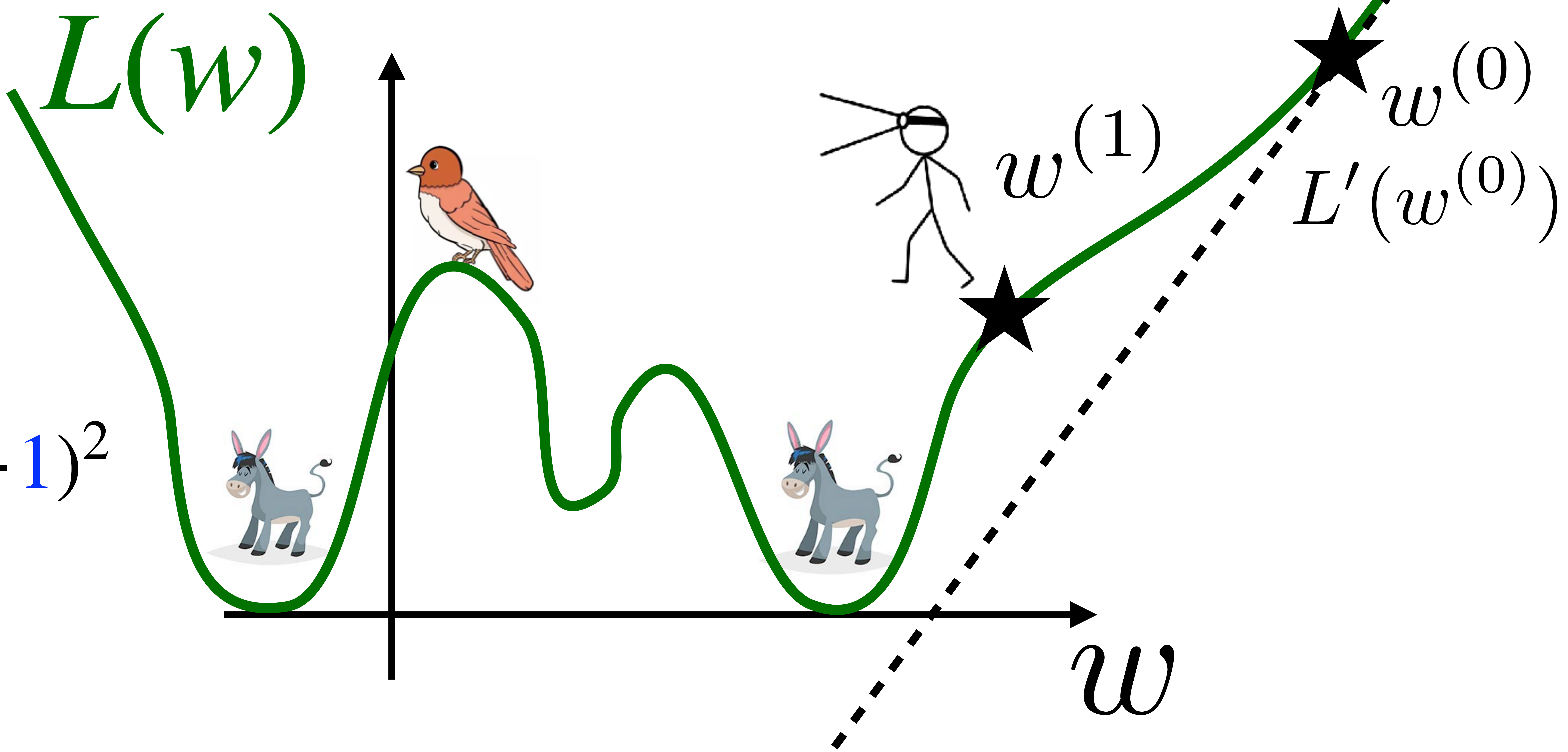
Back to 1D:

$$w^{(1)} = w^{(0)} - \gamma L'(w^{(0)})$$



$$\min_{w \in \mathbb{R}^p} L(w)$$

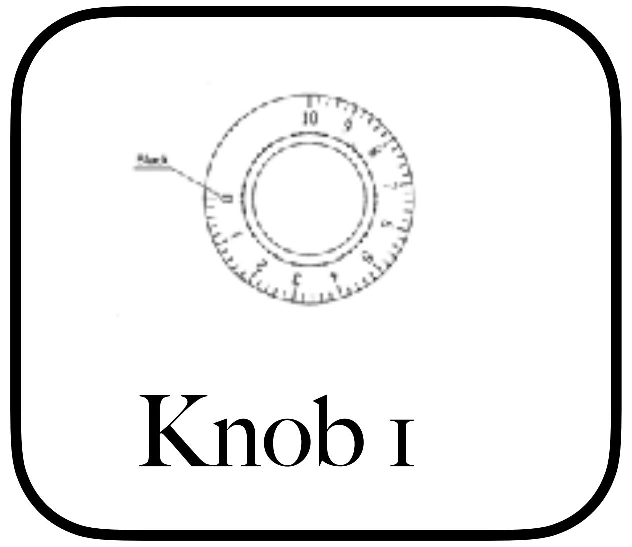
$$L(w) = (f_w(x) - 1)^2$$

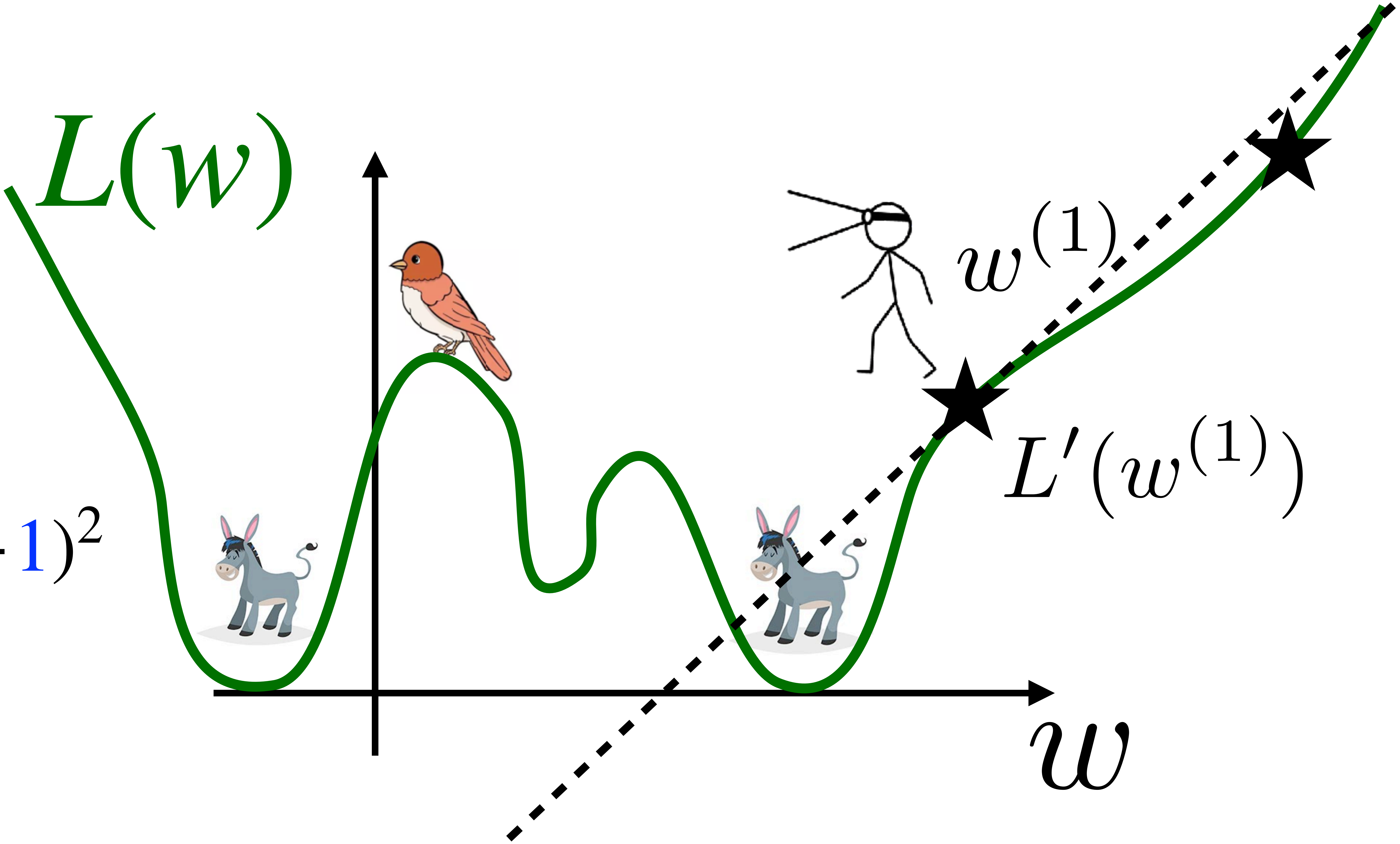


It boils down to minimising a function! $\min_{w \in \mathbb{R}^p} (f_w(x) - 1)^2$

$$w^{(1)} = w^{(0)} - \gamma L'(w^{(0)})$$

Back to 1D:





$$\min_{w \in \mathbb{R}^p} L(w)$$

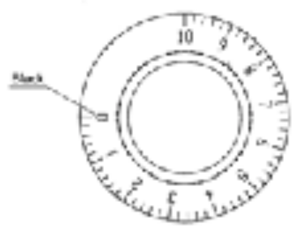
$$L(w) = (f_w(x) - 1)^2$$

It boils down to minimising a function!

$$\min_{w \in \mathbb{R}^p} (f_w(x) - 1)^2$$

Back to 1D:

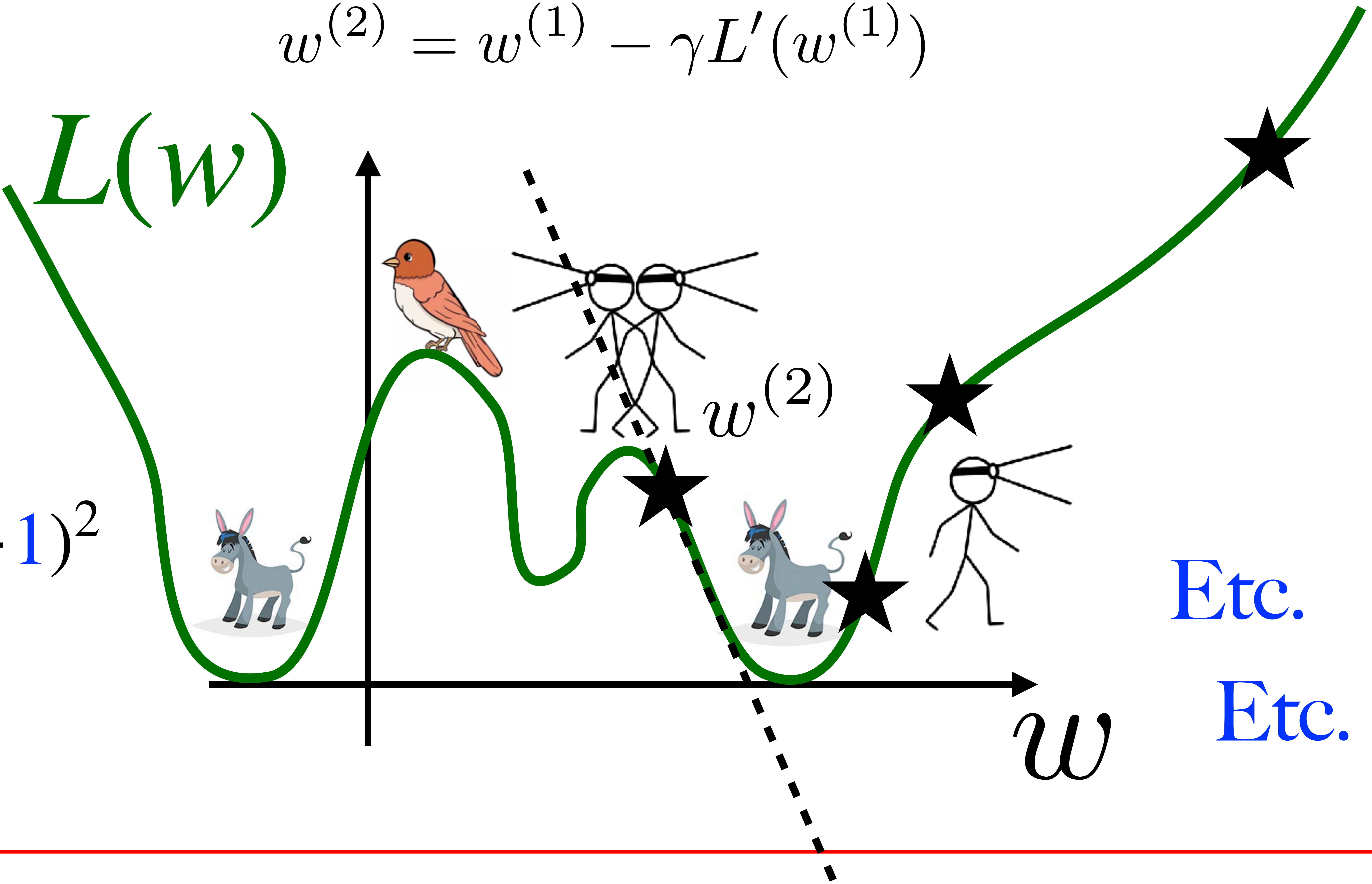
$$w^{(1)} = w^{(0)} - \gamma L'(w^{(0)})$$
$$w^{(2)} = w^{(1)} - \gamma L'(w^{(1)})$$



Knob 1

$$\min_{w \in \mathbb{R}^p} L(w)$$

$$L(w) = (f_w(x) - 1)^2$$



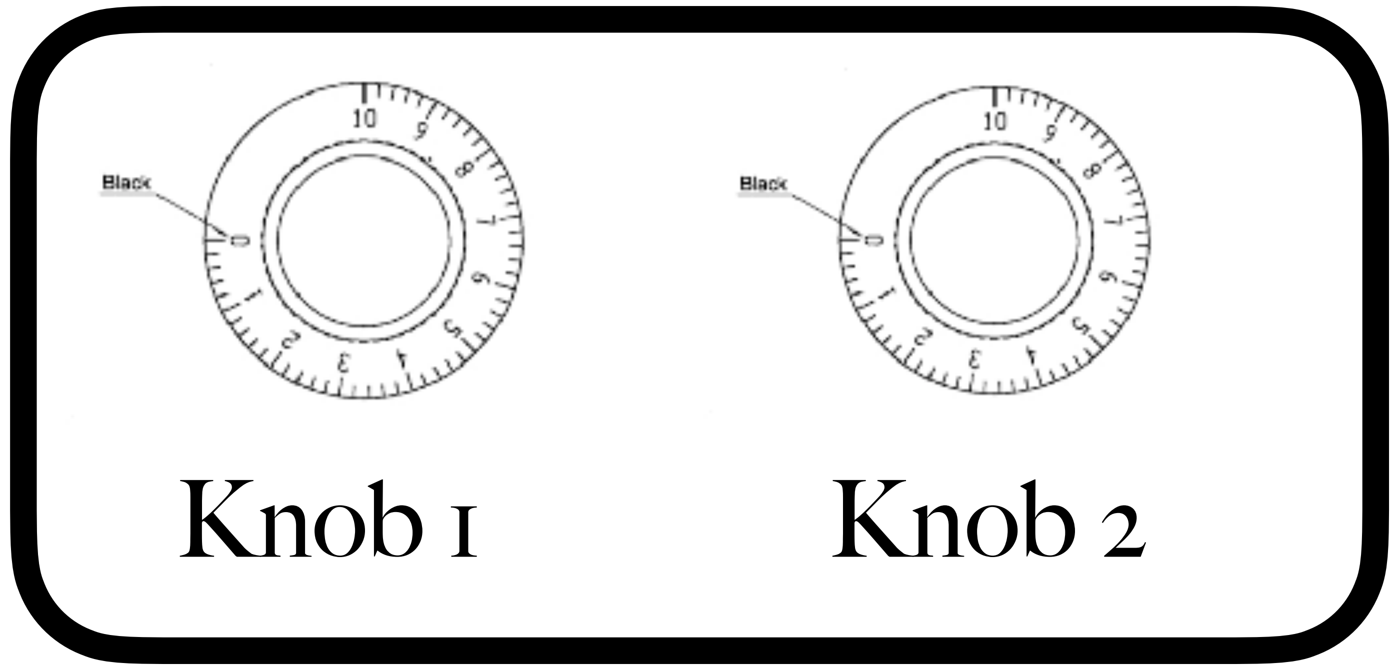
Etc.

Etc.

It boils down to minimising a function!

$$\min_{w \in \mathbb{R}^p} (f_w(x) - 1)^2$$

En 2D:

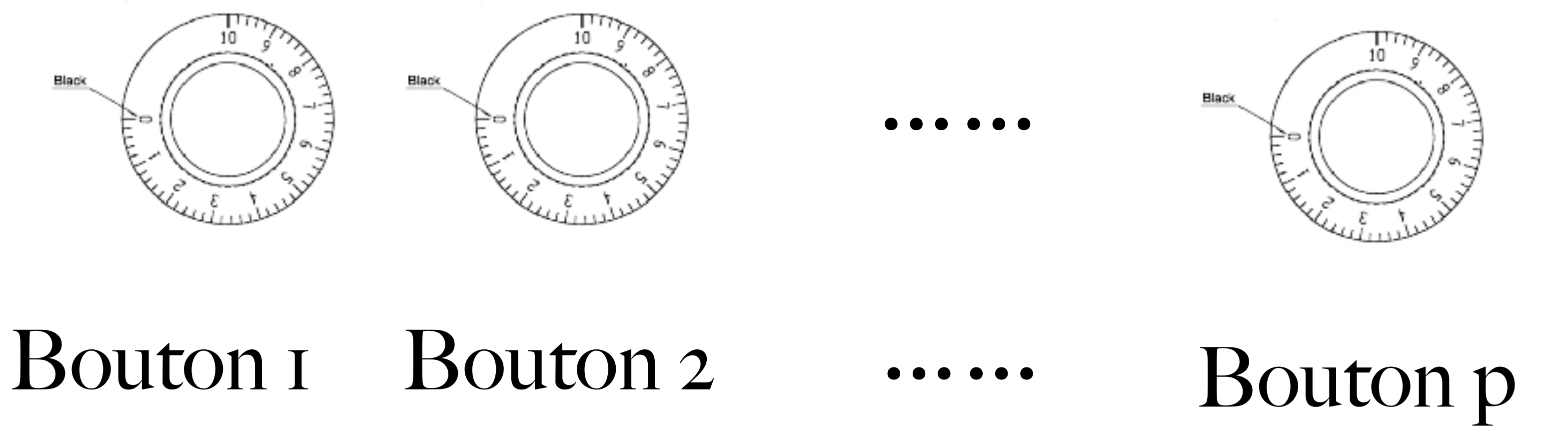


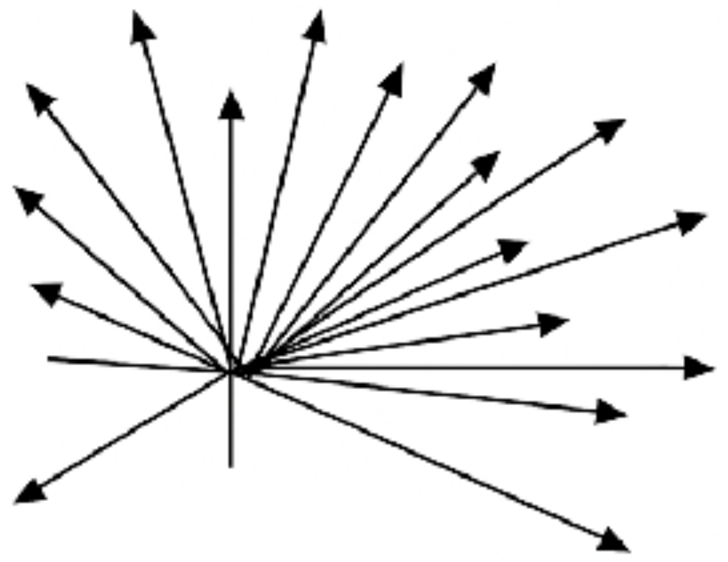
$$\min_{w \in \mathbb{R}^p} L(w)$$

$$L(w) = (f_w(x) - 1)^2$$

It boils down to minimising a function!

$$\min_{w \in \mathbb{R}^p} (f_w(x) - 1)^2$$





$$\min_{w \in \mathbb{R}^p} L(w)$$

$$L(w) = (f_w(x) - 1)^2$$

Gradient descent method:

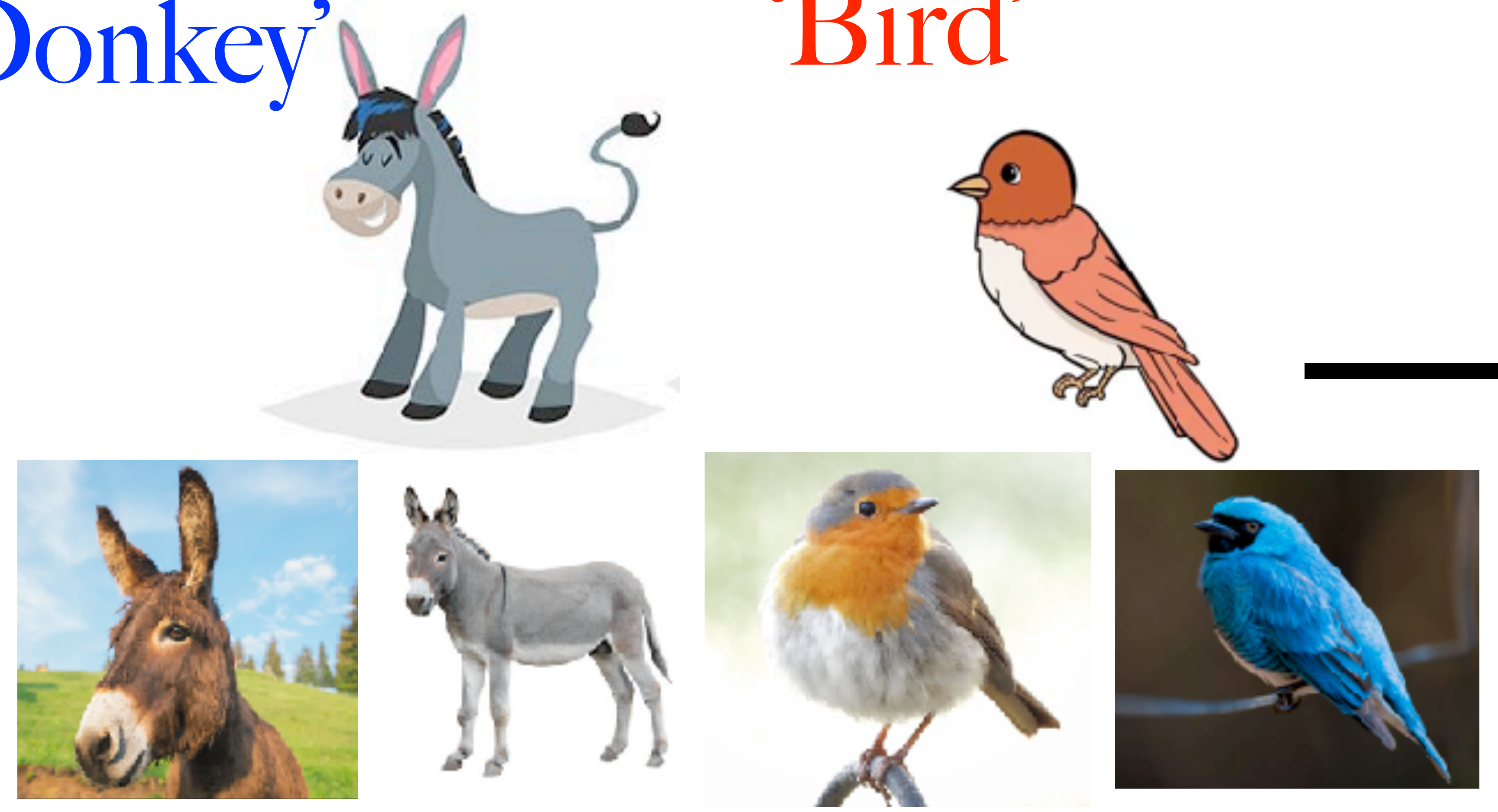
$$w^{(k+1)} = w^{(k)} - \gamma \nabla L(w^{(k)})$$

New knob Configuration = Old configuration + How much to turn them × In which direction

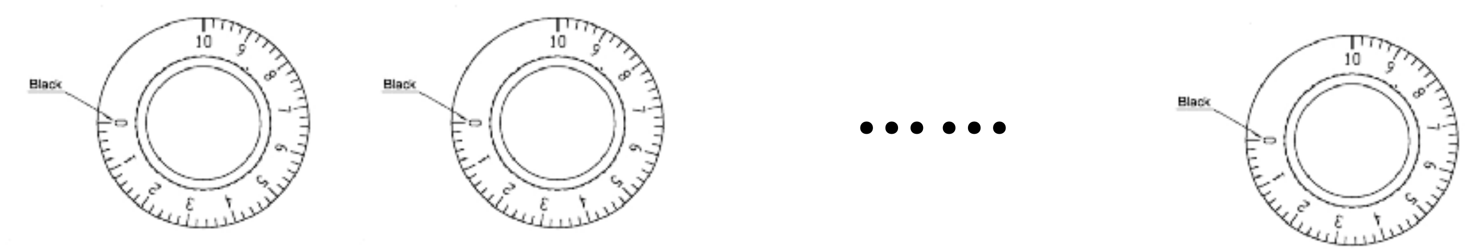
If "everything goes well": $L(w^{(k)}) \underset{k \to \infty}{\longrightarrow} 0$ $\quad f_{w^{(k)}}(x) \underset{k \to \infty}{\longrightarrow} 1$

But there is more than one image!

‘Donkey’ ‘Bird’



Magic synthetiser f_w



Knob 1 Knob 2 …… Knob p

$$w = (w_1, w_2, \cdots, w_p) \in \mathbb{R}^p$$

Machine’s Response

$$f_w(x) \in \mathbb{R}$$

Data:

$$(x_1, y_1), \cdots, (x_n, y_n)$$

$$\begin{bmatrix} 1.2 & 3.5 & 0.7 & 4.1 & 2.7 & 1.3 \\ 0.6 & 2.8 & 3.3 & 1.1 & 0.5 & 4.9 \\ 7.2 & 2.4 & 1.1 & 0.9 & 3.3 & 2.2 \\ 3.7 & 0.9 & 4.8 & 5.5 & 0.2 & 1.6 \\ 1.0 & 3.1 & 2.2 & 2.7 & 4.0 & 0.8 \\ 0.3 & 1.5 & 6.4 & 2.9 & 3.6 & 1.0 \end{bmatrix} \quad +1$$

$$\begin{bmatrix} 3.1 & 0.8 & 2.4 & 5.2 & 1.9 & 3.6 \\ 4.7 & 1.2 & 0.9 & 3.4 & 2.8 & 5.0 \\ 0.5 & 3.3 & 4.1 & 1.8 & 2.0 & 0.7 \\ 2.6 & 5.5 & 1.1 & 0.4 & 3.9 & 2.3 \\ 3.0 & 2.7 & 5.8 & 1.5 & 0.6 & 4.2 \\ 1.4 & 3.6 & 0.2 & 4.5 & 2.1 & 5.3 \end{bmatrix} \quad -1$$

Loss to minimise:

$$\min_w \sum_{i=1}^{n} (y_i - f_w(x_i))^2$$

“Supervised learning”

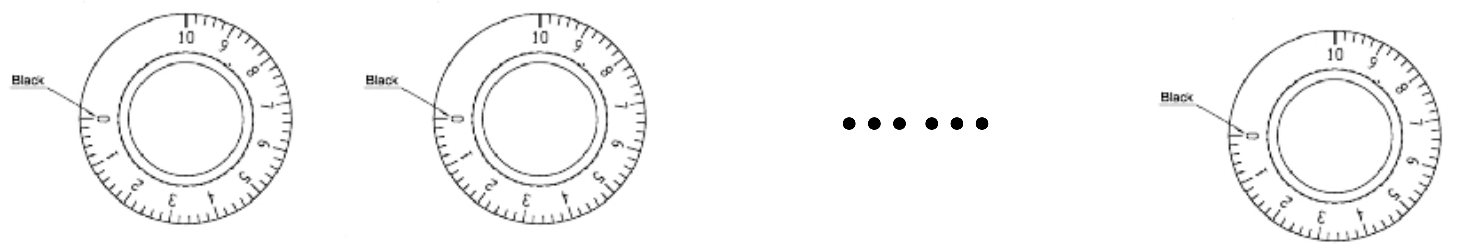
What about ChatGPT?

The donkey is an equid. As a herbivore, it frequently consumes fibrous plants.

Magic synthetiser f_w

The donkey is an [redacted]

$x = (48, 345, 432, 4)$



Knob 1 Knob 2 ...... Knob p

$w = (w_1, w_2, \cdots, w_p) \in \mathbb{R}^p$

The next word

$f_w(x) \in \{1, 2, \cdots, \#\text{tokens}\}$

Animal ?
Mammal ?
Equid ?

Data (sentences):

$$x_1, x_2, \cdots, x_n$$

Fonction à minimiser:

$$\min_w \sum_{i=1}^{n} (\text{mot d'après} - f_w(x_i))^2$$

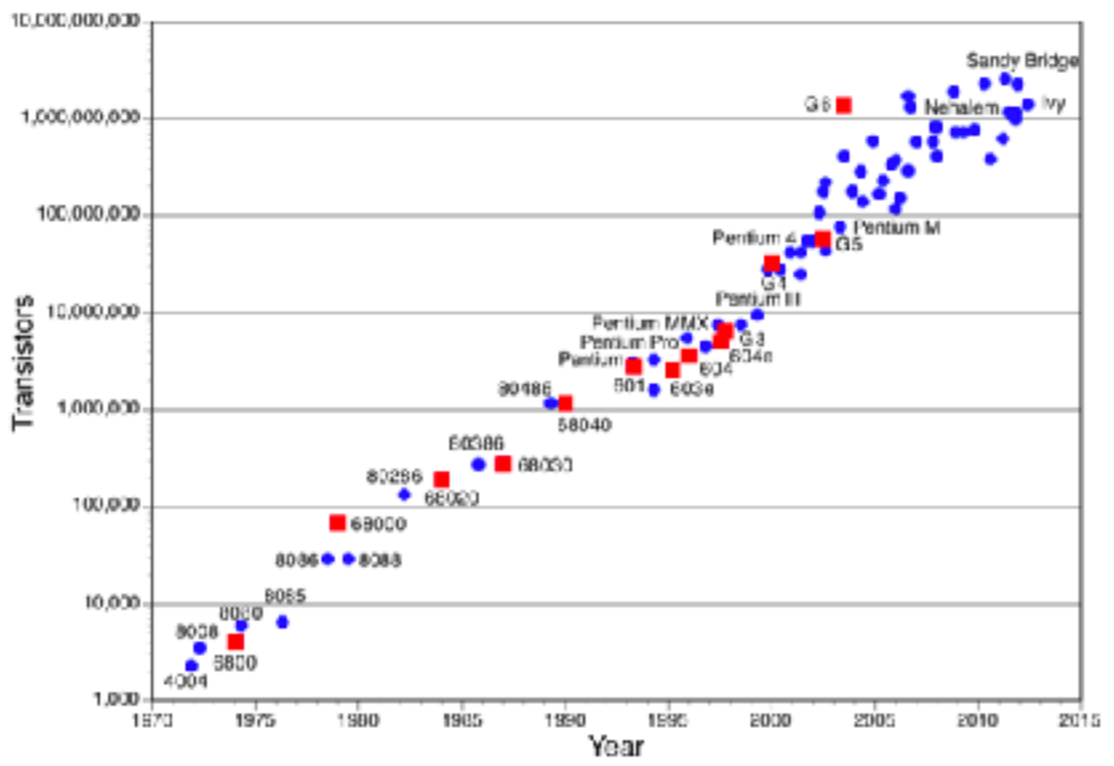
Apprentissage "semi-supervisé"

Taking a step back

Three ingredients behind the success of modern machine learning:

Data: internet (text et images)

1950s: 10^3 FLOPs
2024: 10^{25} FLOPs

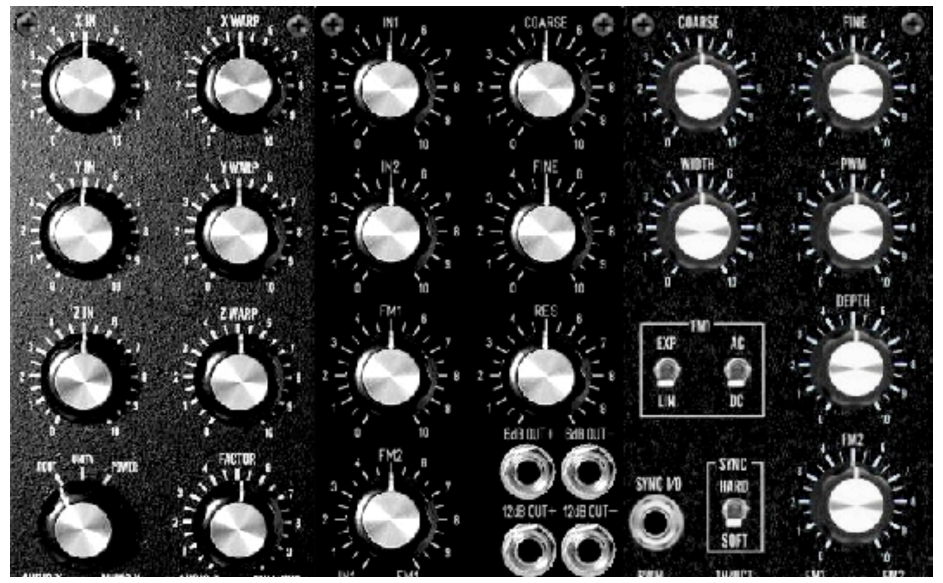


Compute power: graphics processing units (GPU)

Architectures et training techniques:

Neural networks, training algorithms, ...

Training Chat-GPT, some numbers

What's the capital of Switzerland? →  → Bern

Size of the synthesiser:

A trillion (10^{12}) trainable knobs ≈ Paris covered with knobs

Energy to turn the knobs: 10GWh energy consumption (€2M euros electricity bill)

To compare:

Bringing up a child to its twenties:
2000Kal / day —> 100Watts
20 MWh from 0 to 20 years old

therefore: Training ChatGPT ≈ 500 x 

0 —> 20

But not much compared to the ≈ 1GWh daily consumption due to queries!

Text seen by ChatGPT:

Trained on a 10 trillion (10^{13}) words ≈ 10 million copies of War and Peace ≈ 100 000 years to read (no sleeping)

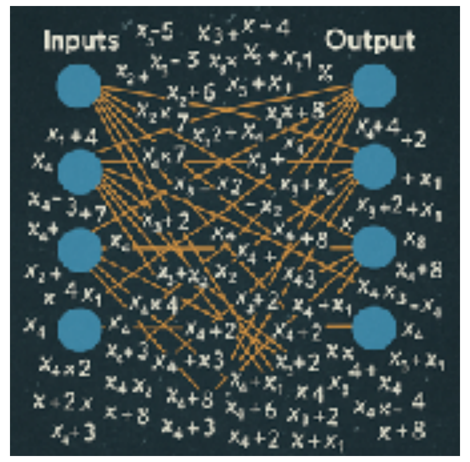
Why don't we understand?

$\longrightarrow$

$\longrightarrow$

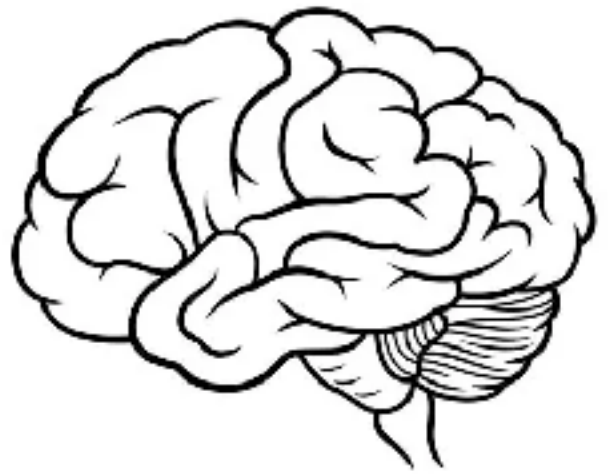
'Lagopède Alpin'

$$f_w(x) = W_L\phi(W_{L-1}\phi(\cdots\phi(W_1x+b_1)\cdots)+b_{L-1})+b_L.$$



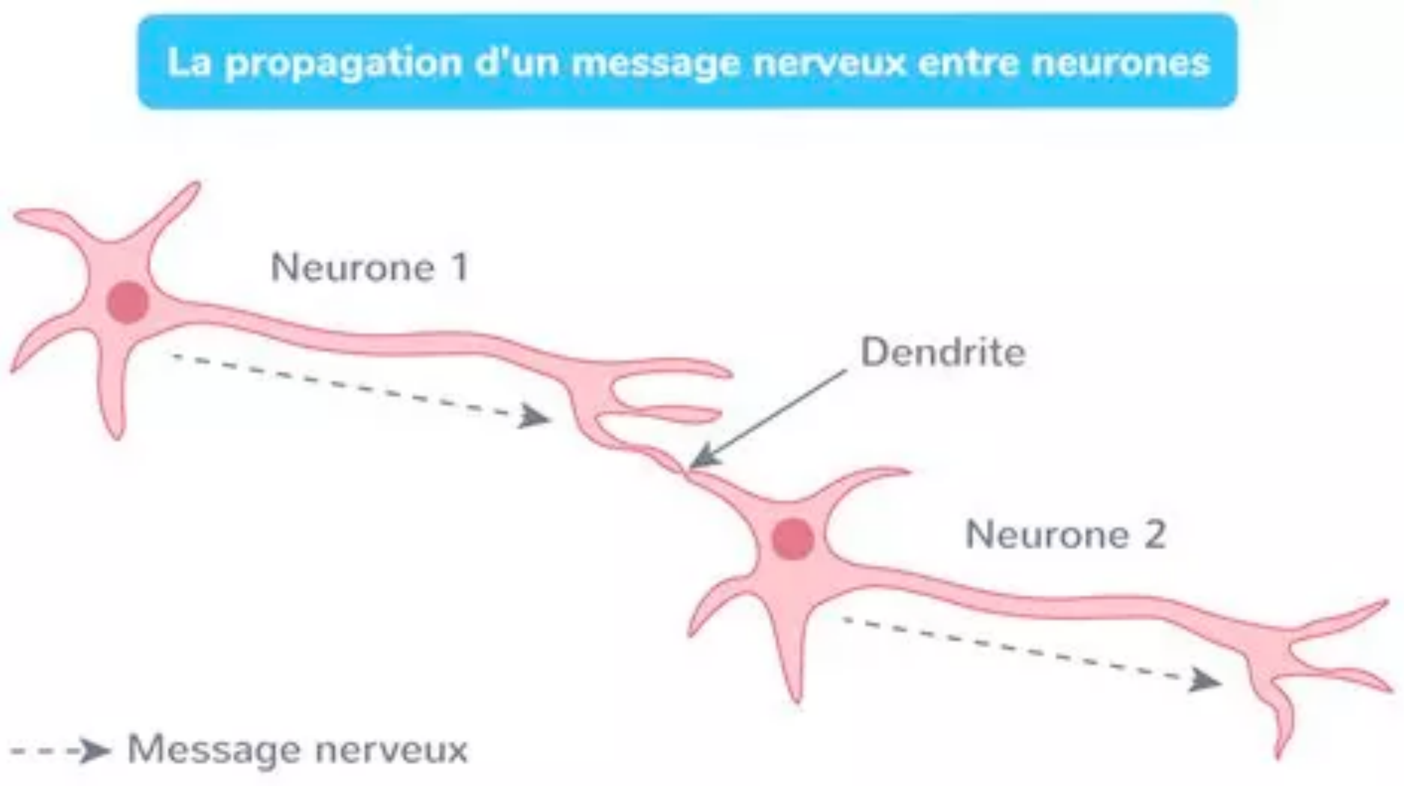
Just because we know exactly what happens at the 'local scale' doesn't mean we understand the behaviour at the macroscopic level.

$\longrightarrow$



$\longrightarrow$

'Lagopède Alpin'



What are the issues?

 →  → ‘Lagopède Alpin’

It works brilliantly, but we don’t understand why!

How does it make a decision? With which criteria?
When does it make a mistake? Can we fool it?
Is it biased? Can we extract sensible data?

<u>Issues:</u> security, ethics, reliability, sustainability…

Recent successes challenge many of our conceptions: Machines now navigate territories reserved to conscious beings

Does ChatGPT *understand* things?

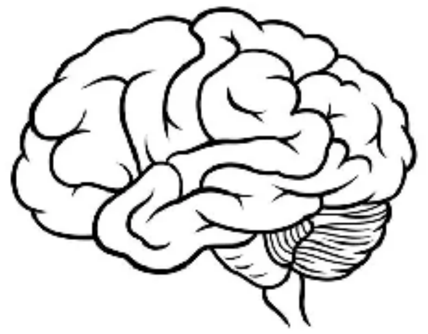
Is it *intelligent*?

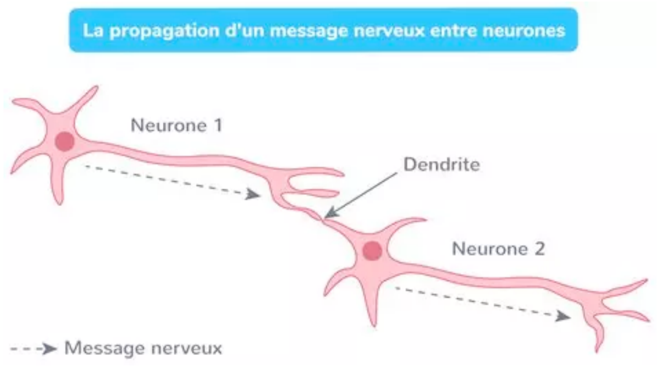
Is it *creative*?

Could it be *conscious?*

How do we even define these words?

What about us, humans:





Is consciousness / intelligence only computational?

Is creativeness about interpolation or extrapolation?

A better understanding of these seemingly 'intelligent' machines will shed light on many aspects of our own.

So what can we do?

As we don’t properly understand them, Current neural network can be depicted as::

Chat-GPT:

Hard to follow what is going on

What theoreticians look at:

It already has similar properties (it flies!), and much simpler to analyse and understand.

“Nothing is more practical than a good theory”

Practical
deep learning

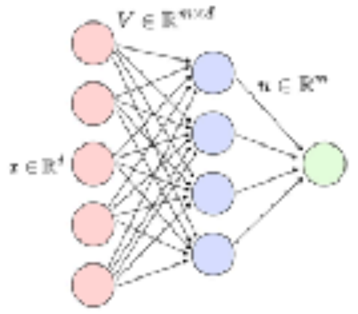
Transformers

$$\mathrm{Atten}(h)(t) = h(t) + \sum_{i=1}^{n} W_o^i \sum_{s=1}^{T} \sigma[(W_Q^i h(t))^\top W_K^i h(s)]\, W_V^i h(s)$$
$$\mathrm{FF}(h)(t) = f(h)(t).$$

Intuition
through
experiments

Theoretical
deep learning

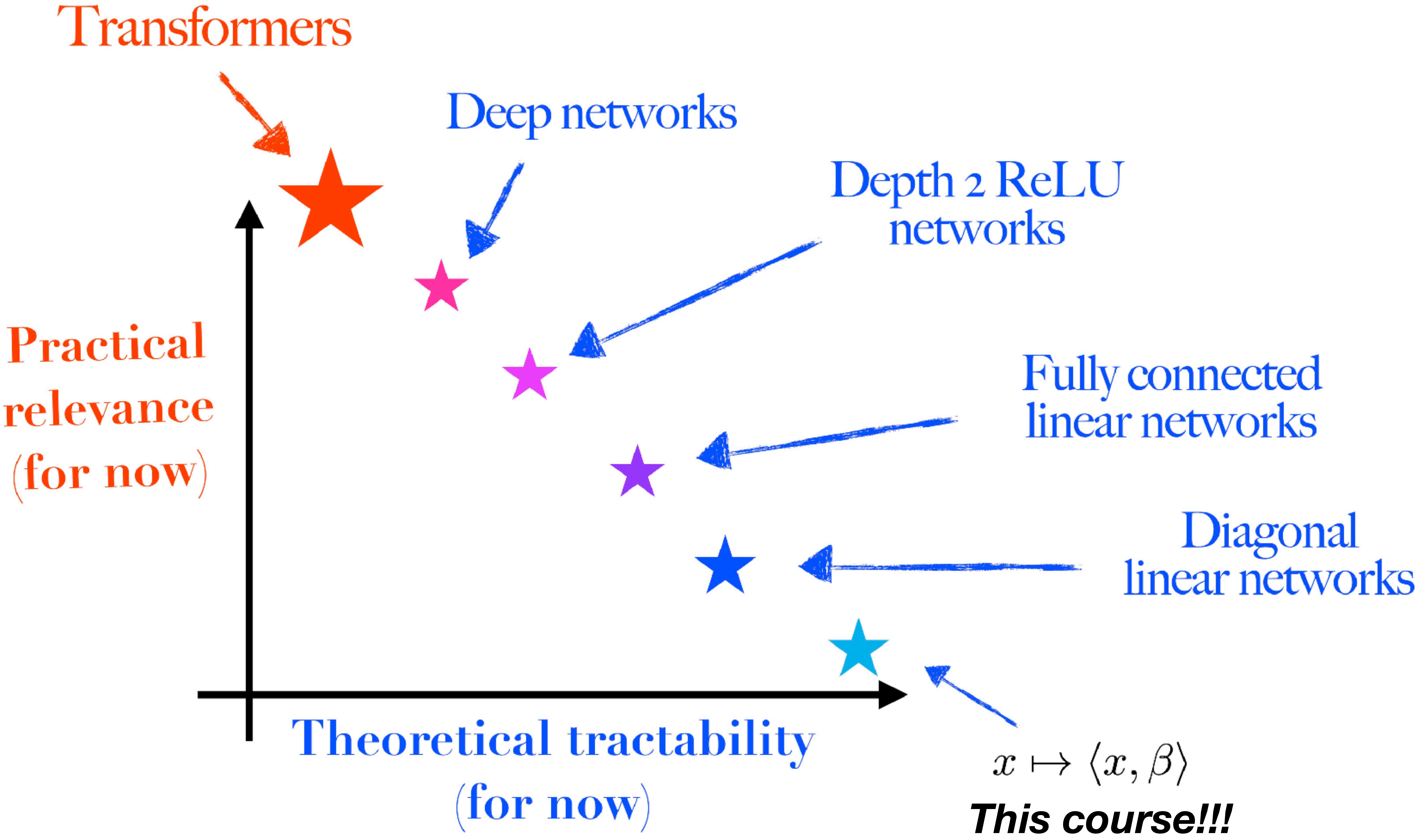
Simplified
network



Understanding
and
Improvements

Future of
deep learning (??)

Architecture of
the future


Transformers
Deep networks
Depth 2 ReLU networks
Fully connected linear networks
Diagonal linear networks
Practical relevance (for now)
Theoretical tractability (for now)
$x \mapsto \langle x, \beta \rangle$
This course!!!

Up next: let’s do some math!

What you know / don't know yet:

1. What is the gradient of $w \in \mathbb{R}^d \mapsto \langle w, x \rangle$

2. Let $A \in R^{d \times d}$, what is $w^\top A w$ equal to? a) $\sum_{i,j=1}^{d} A_{ij} w_i w_j$ b) $\sum_{i=1}^{d} A_{ij} w_i w_j$ c) $\|Aw\|^2$

3. Let $A \in R^{d \times d}$, what is the gradient of $w \in \mathbb{R}^d \mapsto w^\top A w$?

4. Which of the following are convex functions?

a) $w \mapsto \|w\|^2$ b) $w \mapsto w^\top A w$ c) $w \mapsto \exp(w)$ d) $w \mapsto \ln(w)$

5. Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable. Which minimal condition guarantees that $w^\star$ is a global minimiser?

a) $\nabla f(w^\star) = 0$ b) $\nabla^2 f(w^\star) \succeq 0$ c) $\nabla f(w^\star) = 0$ and $\nabla^2 f(w^\star) \succeq 0$