Outline of the Course

1. The Learning Problem *(April 3)*
2. Is Learning Feasible? *(April 5)*
3. The Linear Model I *(April 10)*
4. Error and Noise *(April 12)*
5. Training versus Testing *(April 17)*
6. Theory of Generalization *(April 19)*
7. The VC Dimension *(April 24)*
8. Bias-Variance Tradeoff *(April 26)*
9. The Linear Model II *(May 1)*
10. Neural Networks *(May 3)*
11. Overfitting *(May 8)*
12. Regularization *(May 10)*
13. Validation *(May 15)*
14. Support Vector Machines *(May 17)*
15. Kernel Methods *(May 22)*
16. Radial Basis Functions *(May 24)*
17. Three Learning Principles *(May 29)*
18. Epilogue *(May 31)*

- theory; mathematical
- technique; practical
- analysis; conceptual

Learning From Data

Yaser S. Abu-Mostafa
California Institute of Technology

Lecture 1: The Learning Problem

Sponsored by Caltech's Provost Office, E&AS Division, and IST • Tuesday, April 3, 2012

The learning problem - Outline

- Example of machine learning
- Components of Learning
- A simple model
- Types of learning
- Puzzle

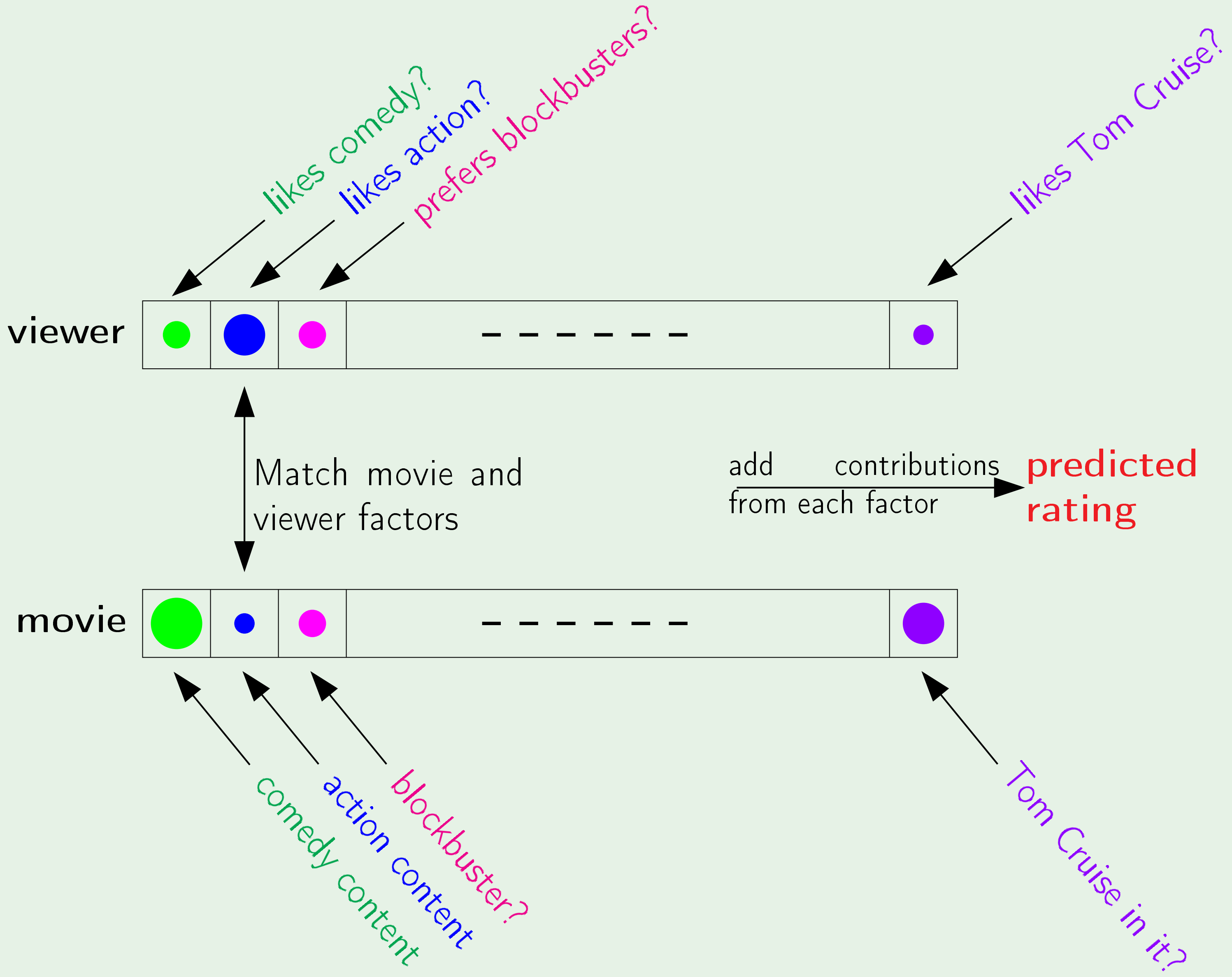
Example: Predicting how a viewer will rate a movie

10% improvement = 1 million dollar prize **by Netflix**

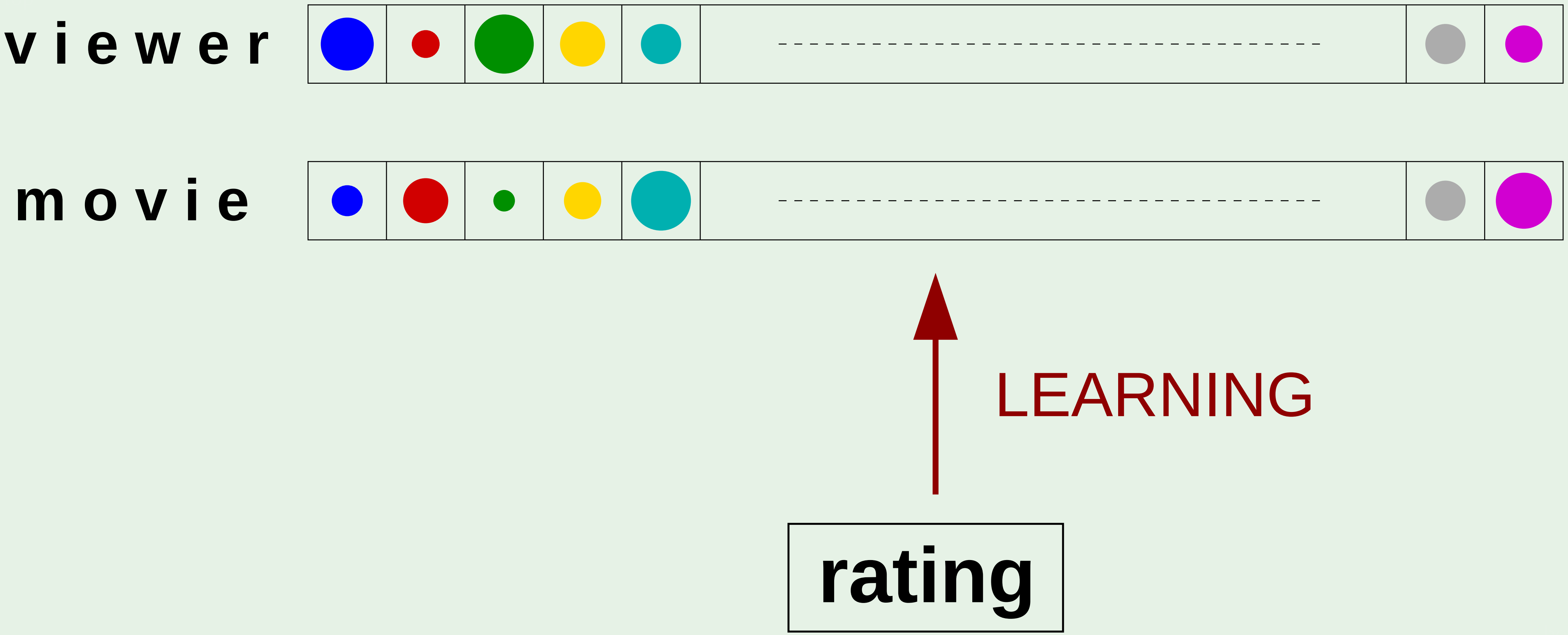
The essence of machine learning:

- A pattern exists.
- We cannot pin it down mathematically.
- We have data on it.

Movie rating - a solution



The learning approach



Components of learning

Metaphor: Credit approval

Applicant information:

age	23 years
gender	male
annual salary	$30,000
years in residence	1 year
years in job	1 year
current debt	$15,000
...	...

Approve credit?

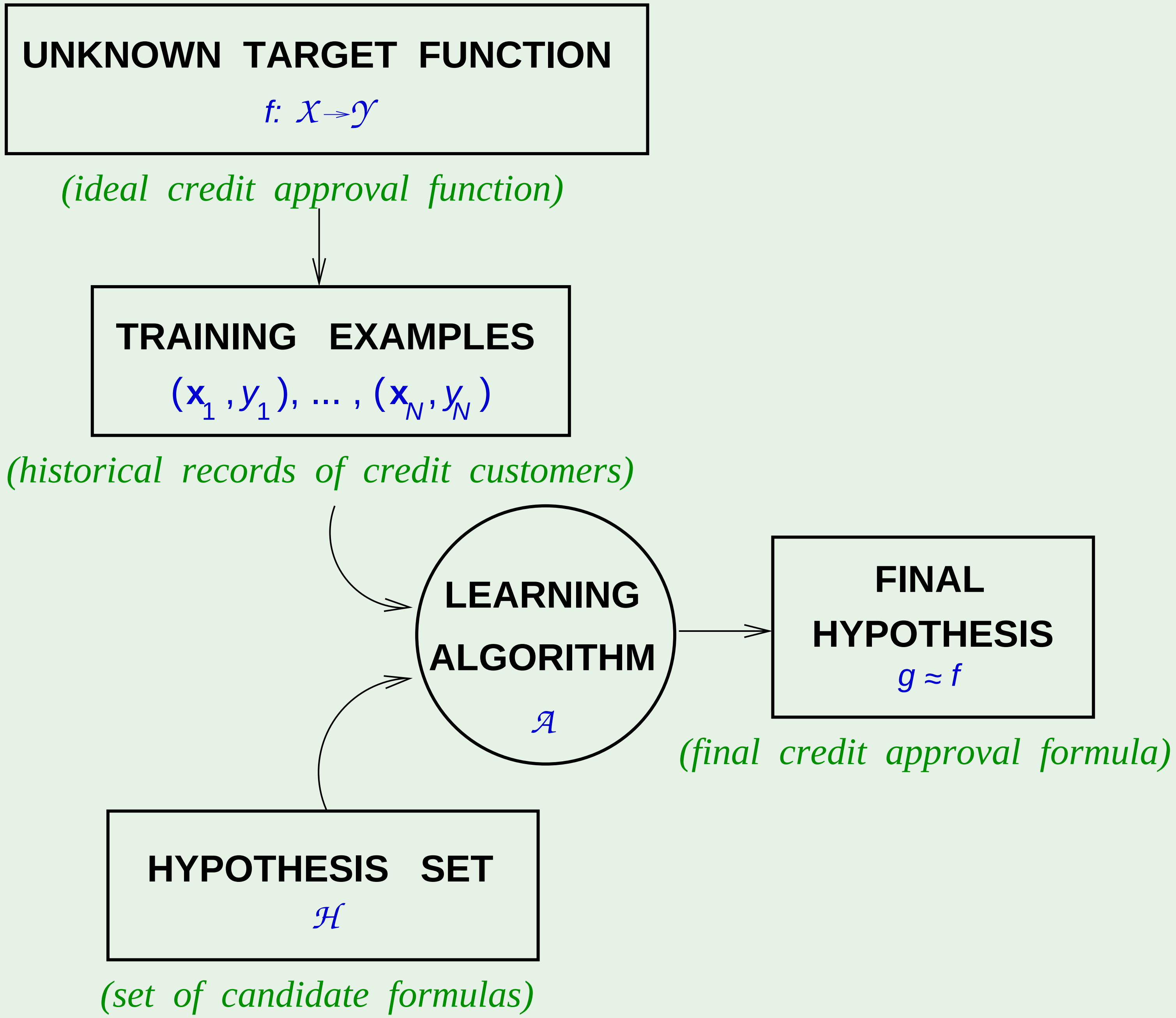
Components of learning

Formalization:

- Input: $\mathbf{x}$ *(customer application)*
- Output: y *(good/bad customer?)*
- Target function: $f : \mathcal{X} \to \mathcal{Y}$ *(ideal credit approval formula)* **f is true but unknown**
- Data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$ *(historical records)* **$y_i = f(x_i)$** **x_i, y_i are given**

$\downarrow \quad \downarrow \quad \downarrow$

- Hypothesis: $g : \mathcal{X} \to \mathcal{Y}$ *(formula to be used)* **g approximates f**


UNKNOWN TARGET FUNCTION
$f: \mathcal{X} \to \mathcal{Y}$
(ideal credit approval function)
TRAINING EXAMPLES
$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
(historical records of credit customers)
LEARNING ALGORITHM
$\mathcal{A}$
FINAL HYPOTHESIS
$g \approx f$
(final credit approval formula)
HYPOTHESIS SET
$\mathcal{H}$
(set of candidate formulas)

Solution components

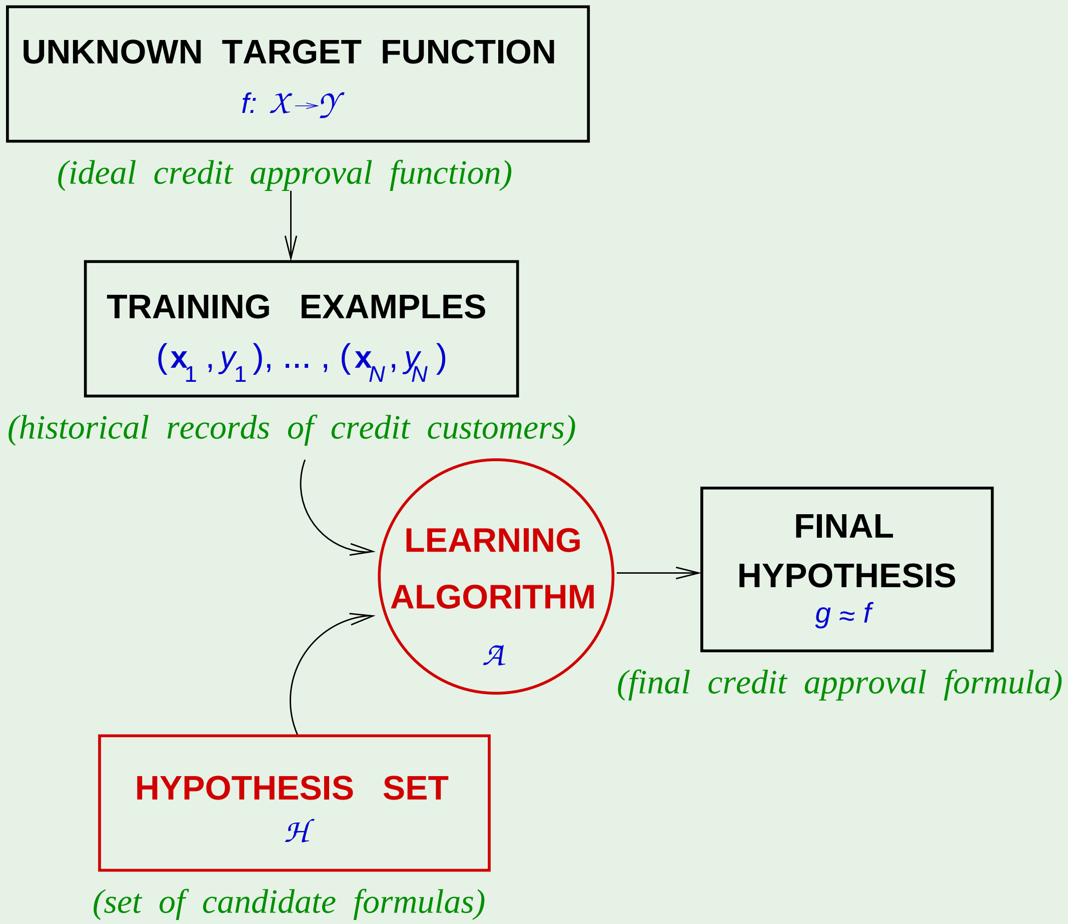
The 2 solution components of the learning problem:

- The Hypothesis Set

  $\mathcal{H} = \{h\} \qquad g \in \mathcal{H}$

- The Learning Algorithm

Together, they are referred to as the *learning model*.



A simple hypothesis set - the 'perceptron'

For input $\mathbf{x} = (x_1, \cdots, x_d)$ 'attributes of a customer'

age	23 years
gender	male
annual salary	\$30,000
years in residence	1 year
years in job	1 year
current debt	\$15,000
$\cdots$	$\cdots$

6 attributes => 6 w_i

x: Input Data Point

b: Bias

w: Weight

y: Output

Approve credit if $\sum_{i=1}^{d} w_i x_i >$ threshold, **-b**

Deny credit if $\sum_{i=1}^{d} w_i x_i <$ threshold. **-b**

This linear formula $h \in \mathcal{H}$ can be written as

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - \text{threshold}\right) \quad \textbf{+b}$$

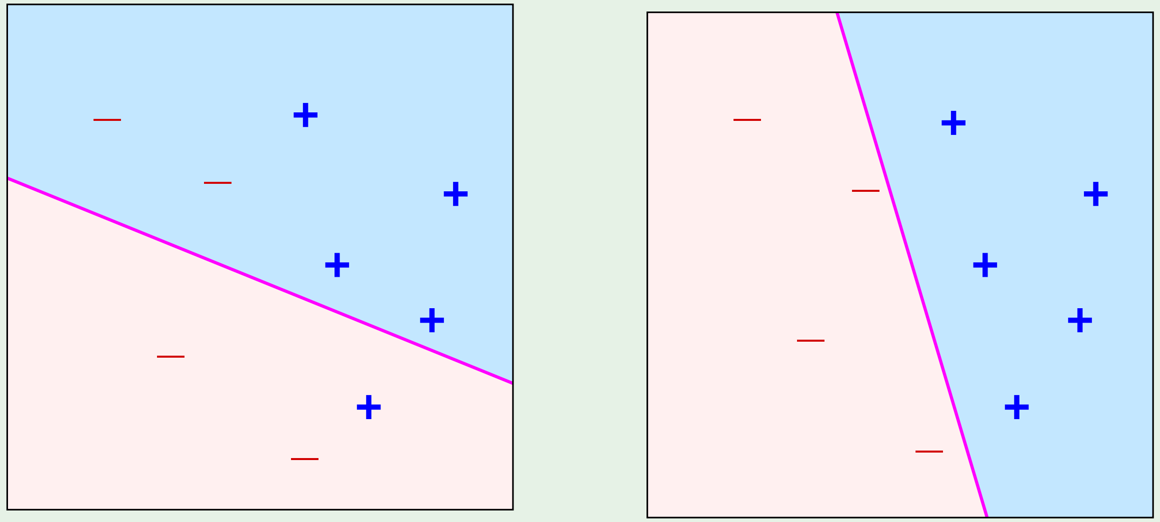
y = h(x) = sign($w^T x$ + b) => approve if h(x) > 0 or $w^T x$ + b > 0

Tensors: Scalars in R^1, Vectors in R^n, Matrices in R^{mxn}, Tensors in ...

y = h(x) = sign($w^T x$), b = 0, +: y = h(x) > 0

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i \; x_i\right) + \quad w_0 \quad \right)$$

axes? space?



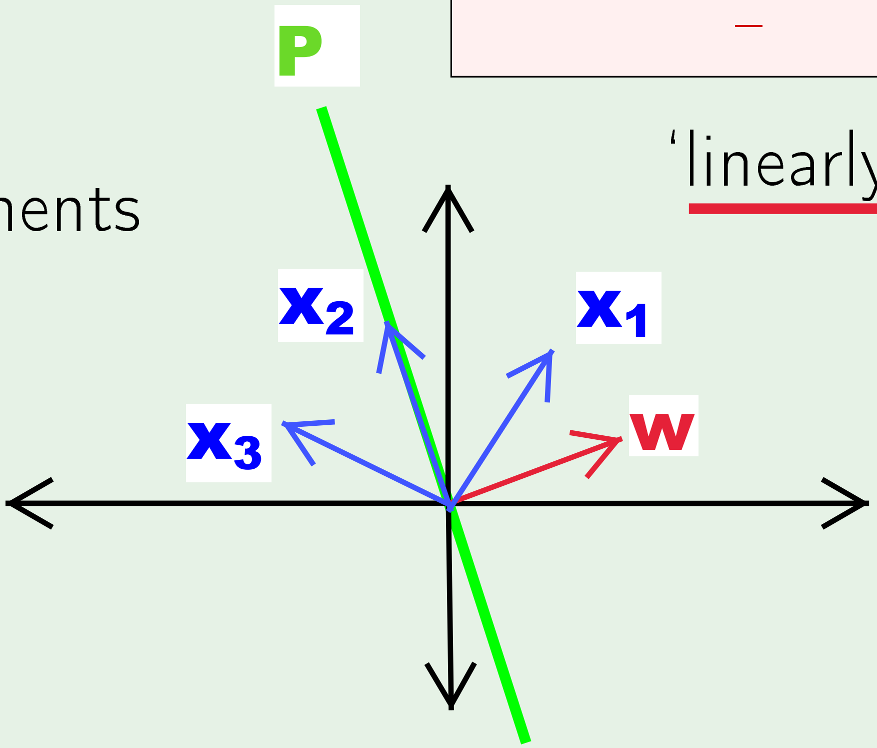
'linearly separable' data

Introduce an artificial coordinate $x_0 = 1$:

$$h(\mathbf{x}) = \text{sign}\left(\sum_{i=0}^{d} w_i \; x_i\right)$$

In vector form, the perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x})$$



$w^T x_1 > 0$

$w^T x_2 = 0$

$w^T x_3 < 0$

A simple learning algorithm - PLA

The perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\intercal \mathbf{x})$$

Given the training set:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$$

pick a misclassified point:

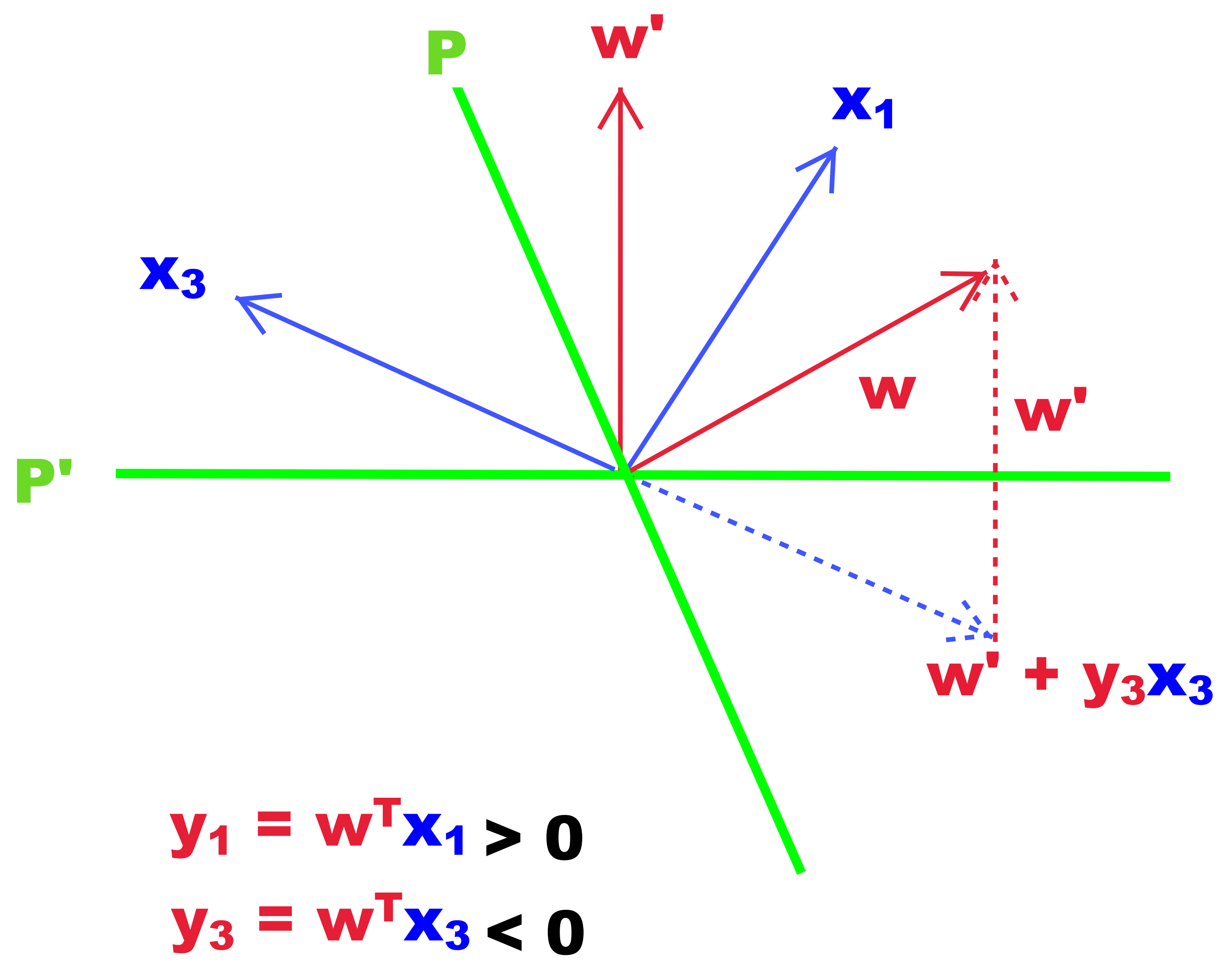
$$\text{sign}(\mathbf{w}^\intercal \mathbf{x}_n) \neq y_n$$

and update the weight vector:

$$\boxed{\mathbf{w} \leftarrow \mathbf{w} + y_n \mathbf{x}_n}$$

Algorithm

Learn what?



Iterations of PLA

- One iteration of the PLA:

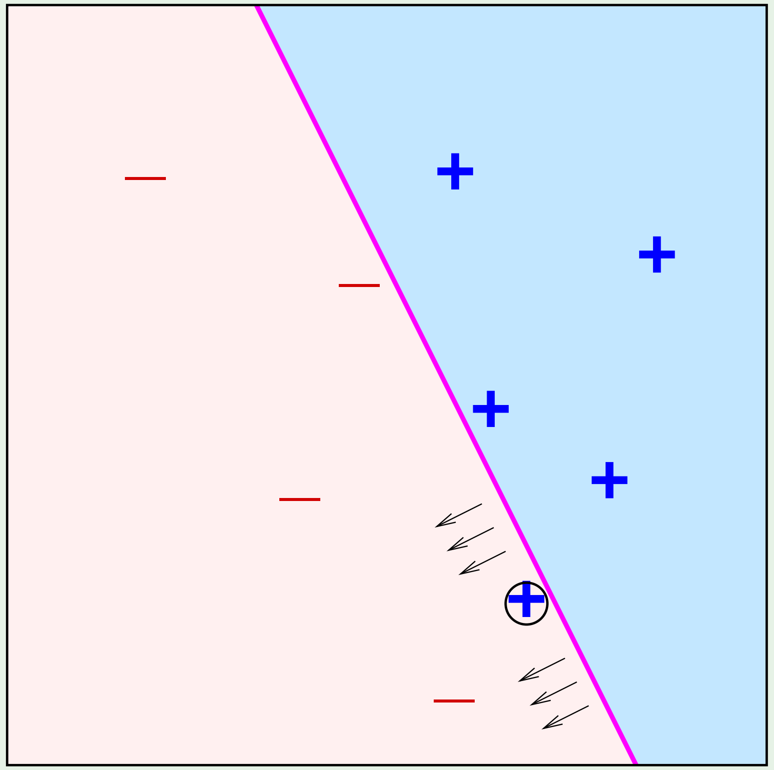
$$\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$$

where $(\mathbf{x}, y)$ is a misclassified training point.

- At iteration $t = 1, 2, 3, \cdots$, pick a misclassified point from

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$$

and run a PLA iteration on it.



- That's it!

How many Ws in real life?
68 billion

How many Ws in this figure?
How many w_i in each W?
Can you arbitrarily draw Ws on this figure?

The learning problem - Outline

- Example of machine learning
- Components of learning
- A simple model
- Types of learning
- Puzzle

Basic premise of learning

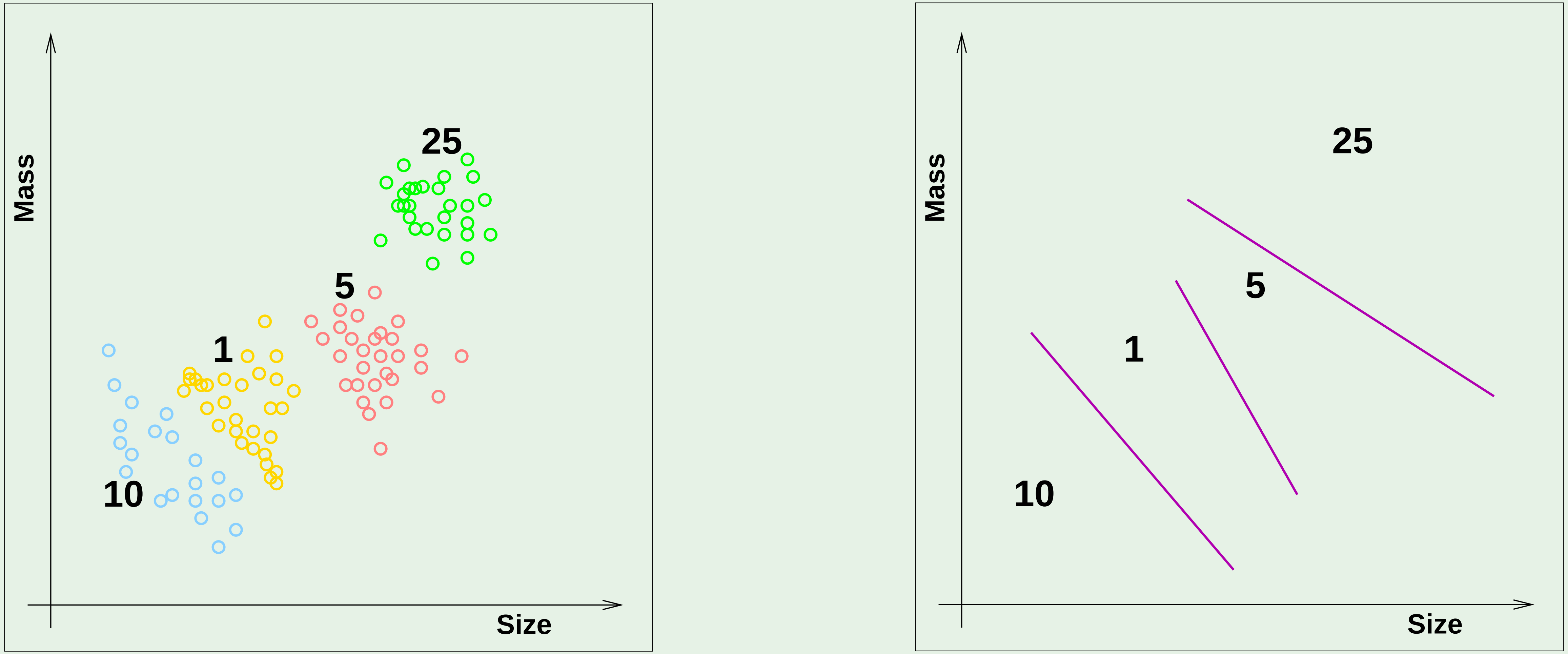
"using a set of observations to uncover an underlying process"

broad premise $\Longrightarrow$ many variations

- **Supervised Learning**
- **Unsupervised Learning**
- **Reinforcement Learning**

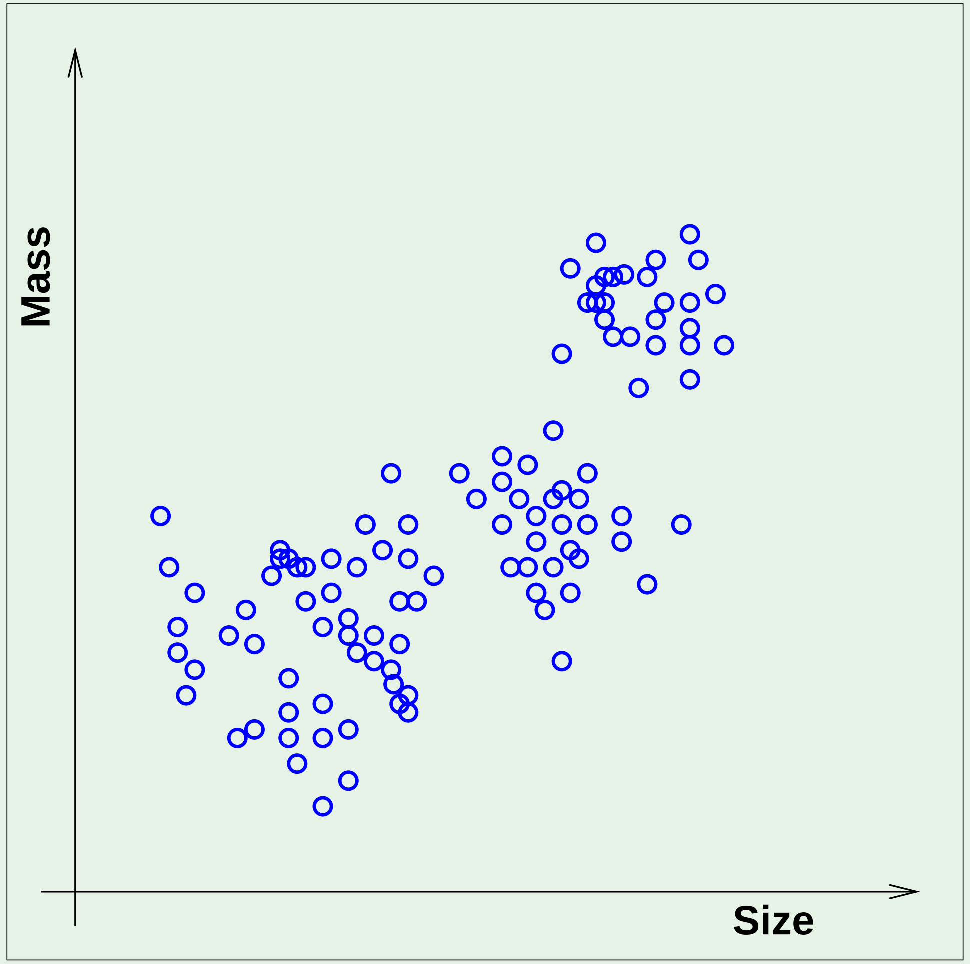
Supervised learning

Example from vending machines – **coin recognition**



Unsupervised learning

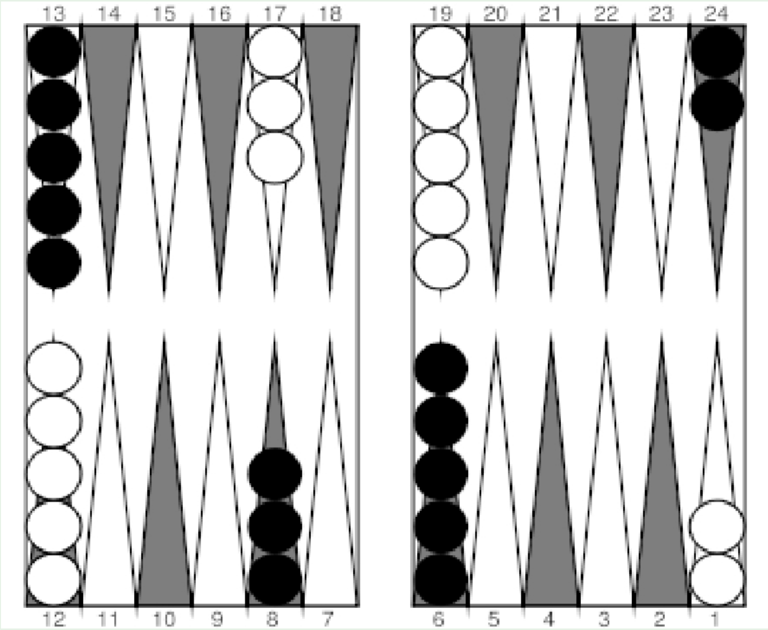
Instead of (input,correct output), we get (input, ?)



Reinforcement learning

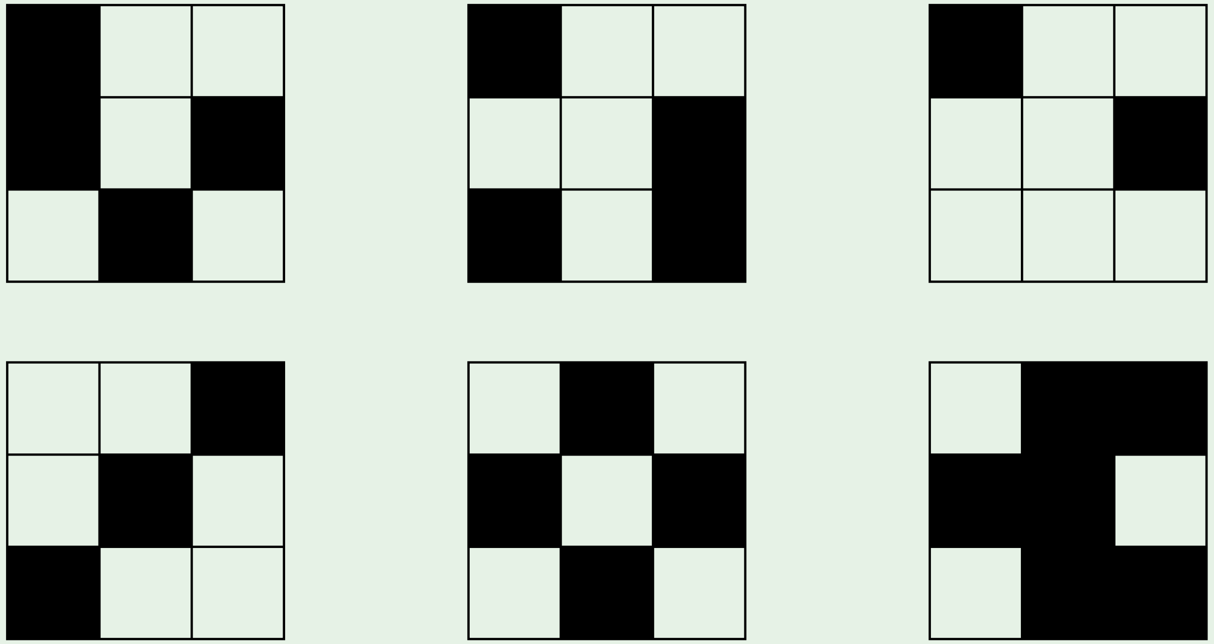
Instead of (**input**,**correct output**),

we get (**input**,*some* **output**,**grade for this output**)



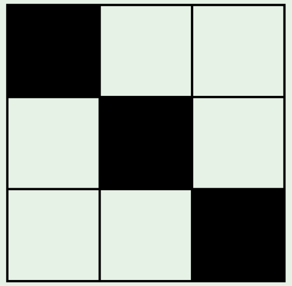
The world champion was a neural network!

A Learning puzzle



$f = -1$

$f = +1$



$f = ?$