Deep Learning

Session 1

Machine Learning Ingredients

Outline

1. About me
2. About the class
3. About the course
4. Machine Learning: Ingredients
5. Revisiting Linear Regression
6. Gradient Descent

About me

Rubén Zazo. **rzazo@faculty.ie.edu**

https://www.linkedin.com/in/ruben-zazo-78467b86/

M.Sc. Telecommunications Engineer.
Ph.D. Telecommunications and Computer Science

Thesis on Machine Learning: Speech Processing with Deep Neural Networks and LSTMs.

What this course is about

This course is fundamentally about **neural networks** as an algorithm/technique to accomplish machine learning.

We will explore:

- The fundamentals behind artificial neural networks.
- What do we call by deep learning or deep neural networks?
- Why deep neural networks are suitable to large scale problems & industry?
- Most famous and standards Deep Neural Nets.

What this course is about

- This course is focused on basics. This is not a very advanced neural nets course.
- … but if you want to jump into the void count me in.
- I want to put this clear cause

What this course is about

- … I want my five stars when you rate me after these lectures

=

What this course is about

In our way, we will also:

- Revisit machine learning fundamentals.
- Learn to use TensorFlow 2.0 & Keras to develop our own DNNs [Python]

What this course is NOT about

We will make use of, but this course is NOT a course of

- Mathematics.
- Coding.

What this course is NOT about

We will make use of, but this course is NOT a course of

- Mathematics.
- Coding.

.. But about the concepts and learn how to apply them.

Why this course is important

Why Deep Learning

Artificial Intelligence:

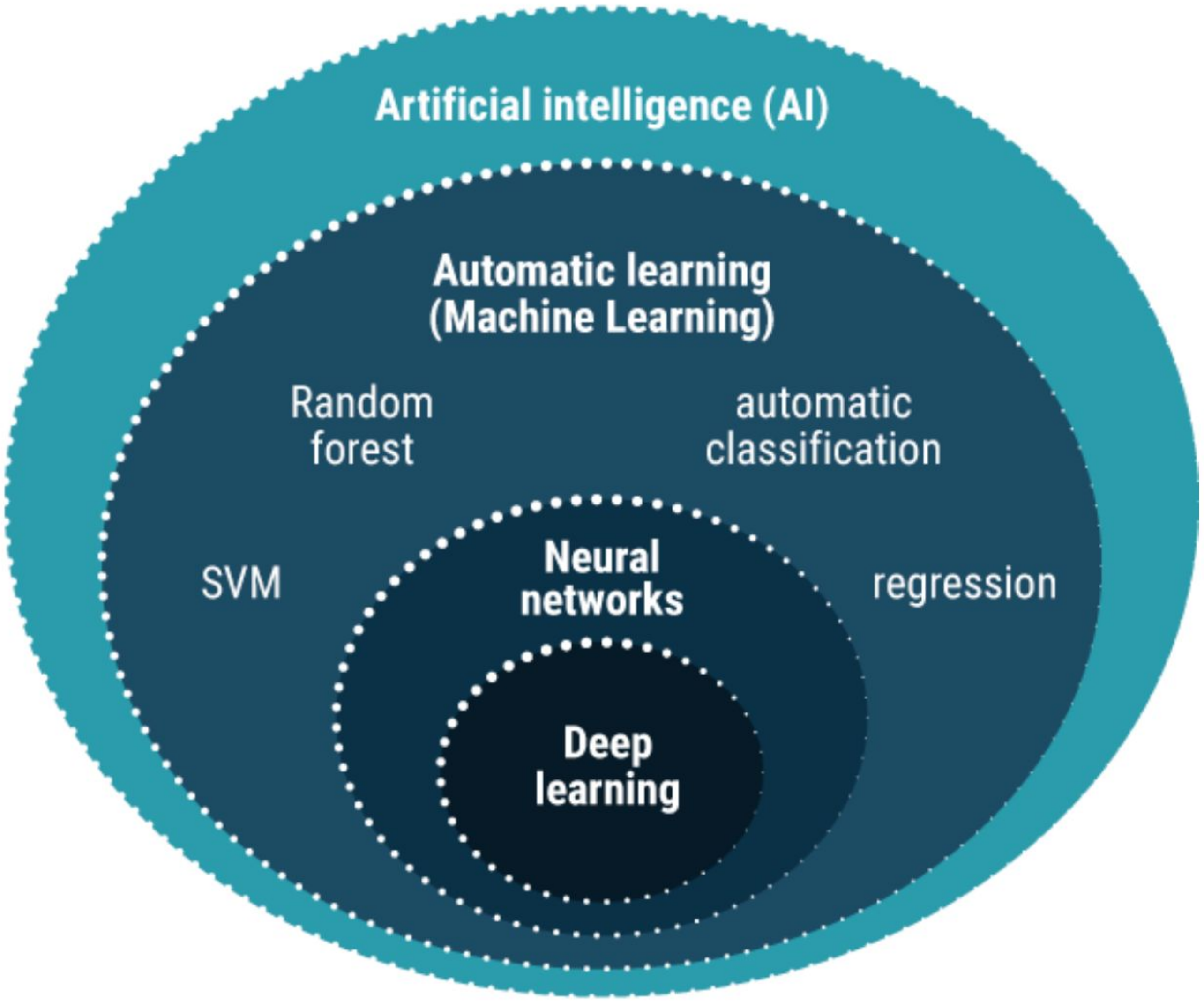
- Machines Behave as Human

Machine Learning:

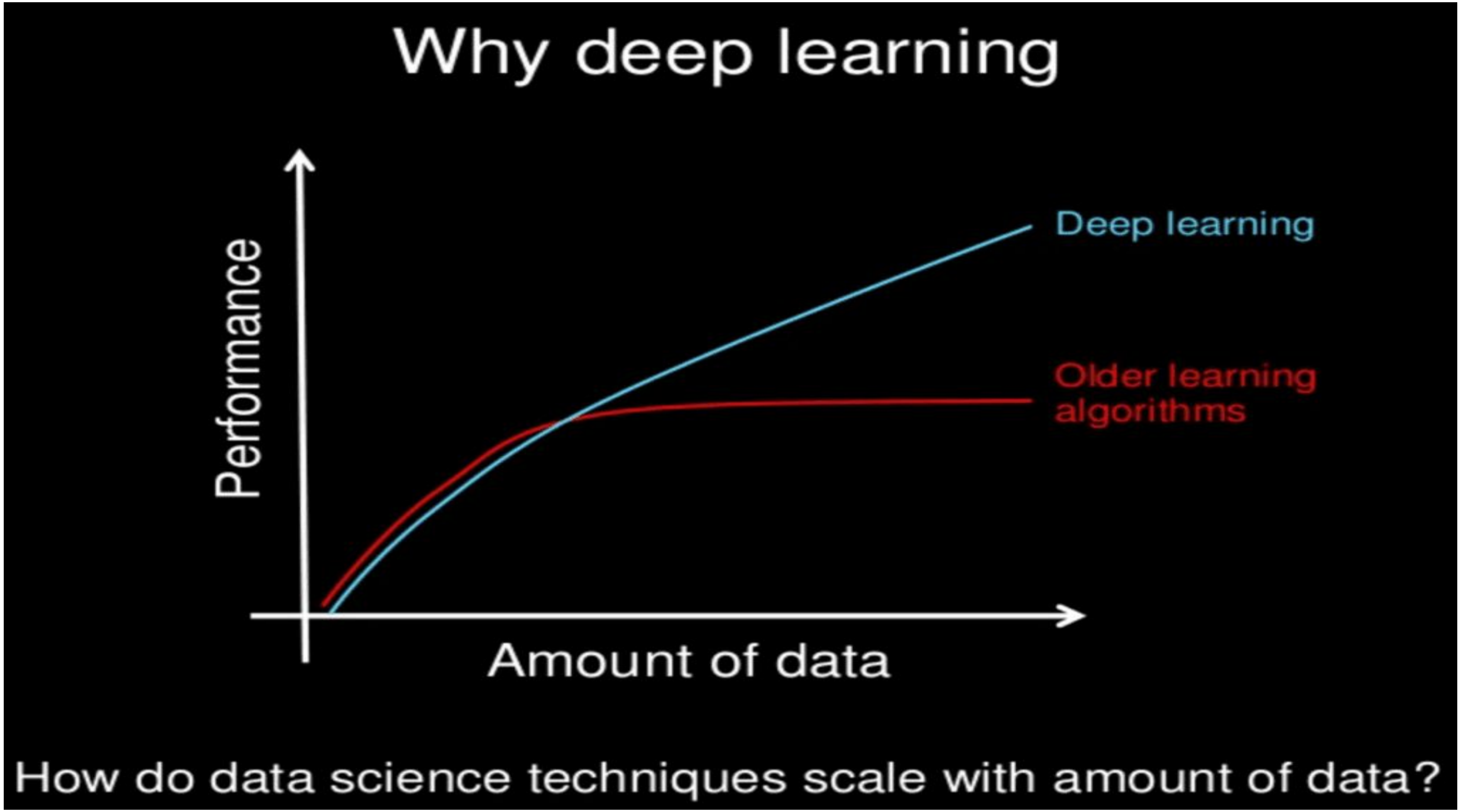
- Machines Able to Learn

Deep Learning:

- A ML Algorithm
- Mimics the Brain



Why this course is important

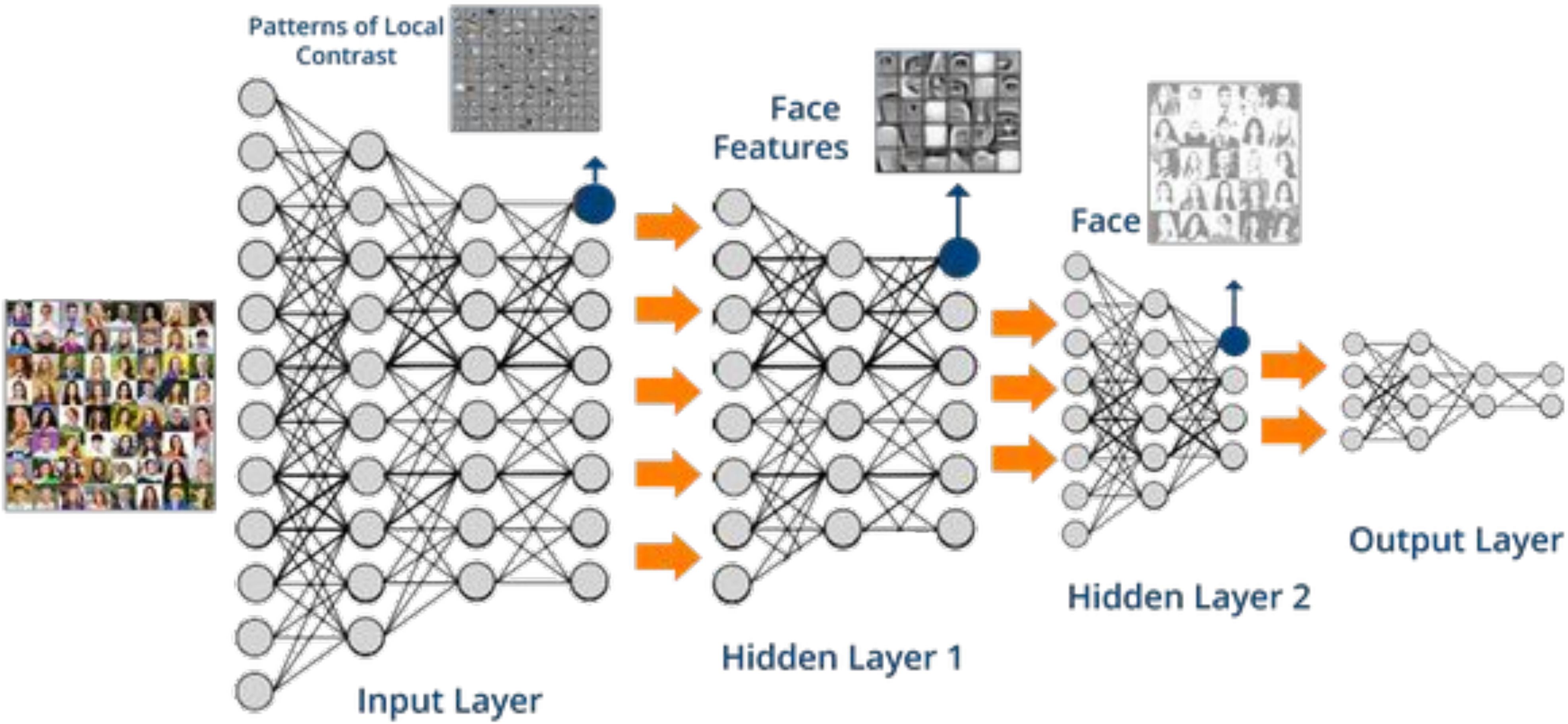


Why Deep Learning

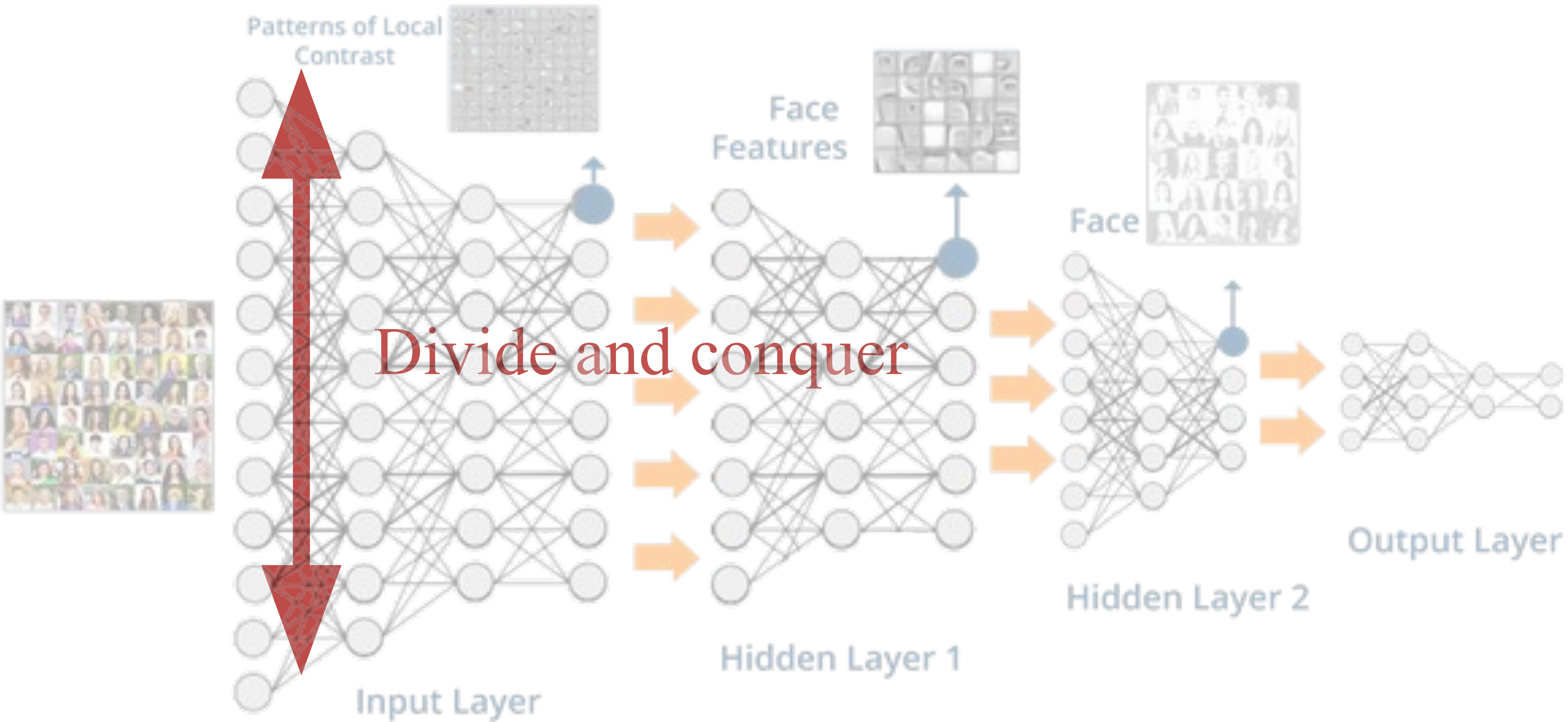
Neural nets as a better way to **represent complex signal** (extract useful patterns)

Retail	Marketing	Healthcare	Telco	Finance
• Demand forecasting • Supply chain optimization • Pricing optimization • Market segmentation and targeting • Recommendations	• Recommendation engines & targeting • Customer 360 • Click-stream analysis • Social media analysis • Ad optimization	• Predicting Patient Disease Risk • Diagnostics and Alerts • Fraud	• Customer churn • System log analysis • Anomaly detection • Preventative maintenance • Smart meter analysis	• Risk Analytics • Customer 360 • Fraud • Credit scoring

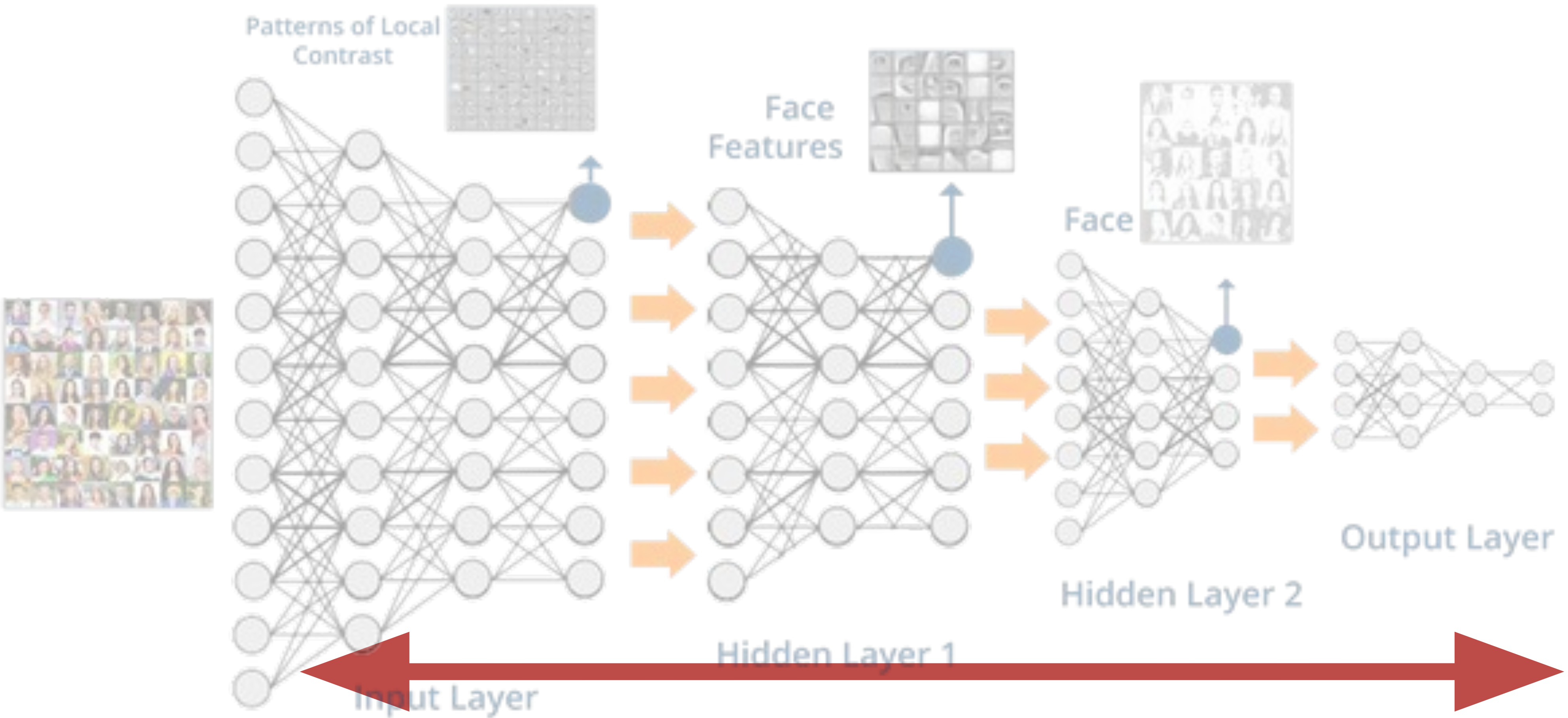
High Level: How do they learn?



High Level: How do they learn?

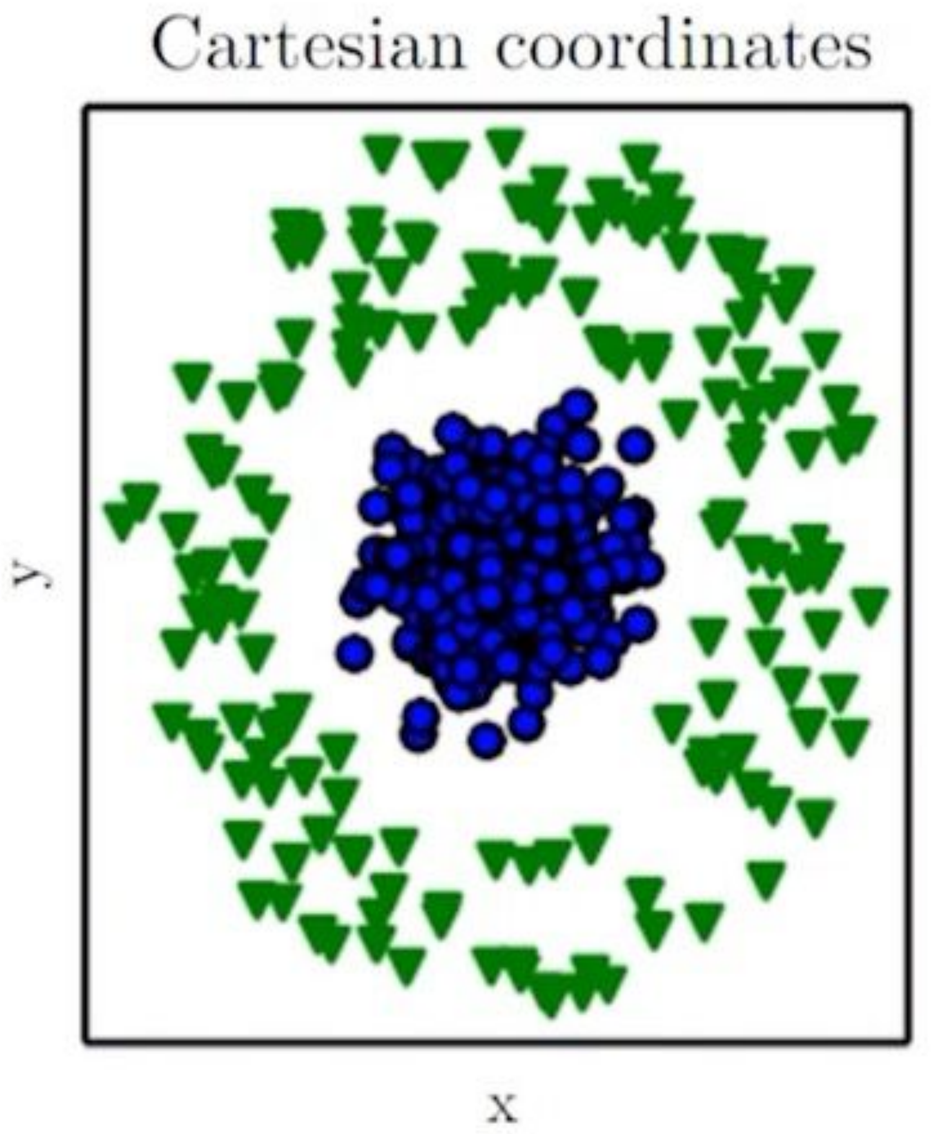


High Level: How do they learn?

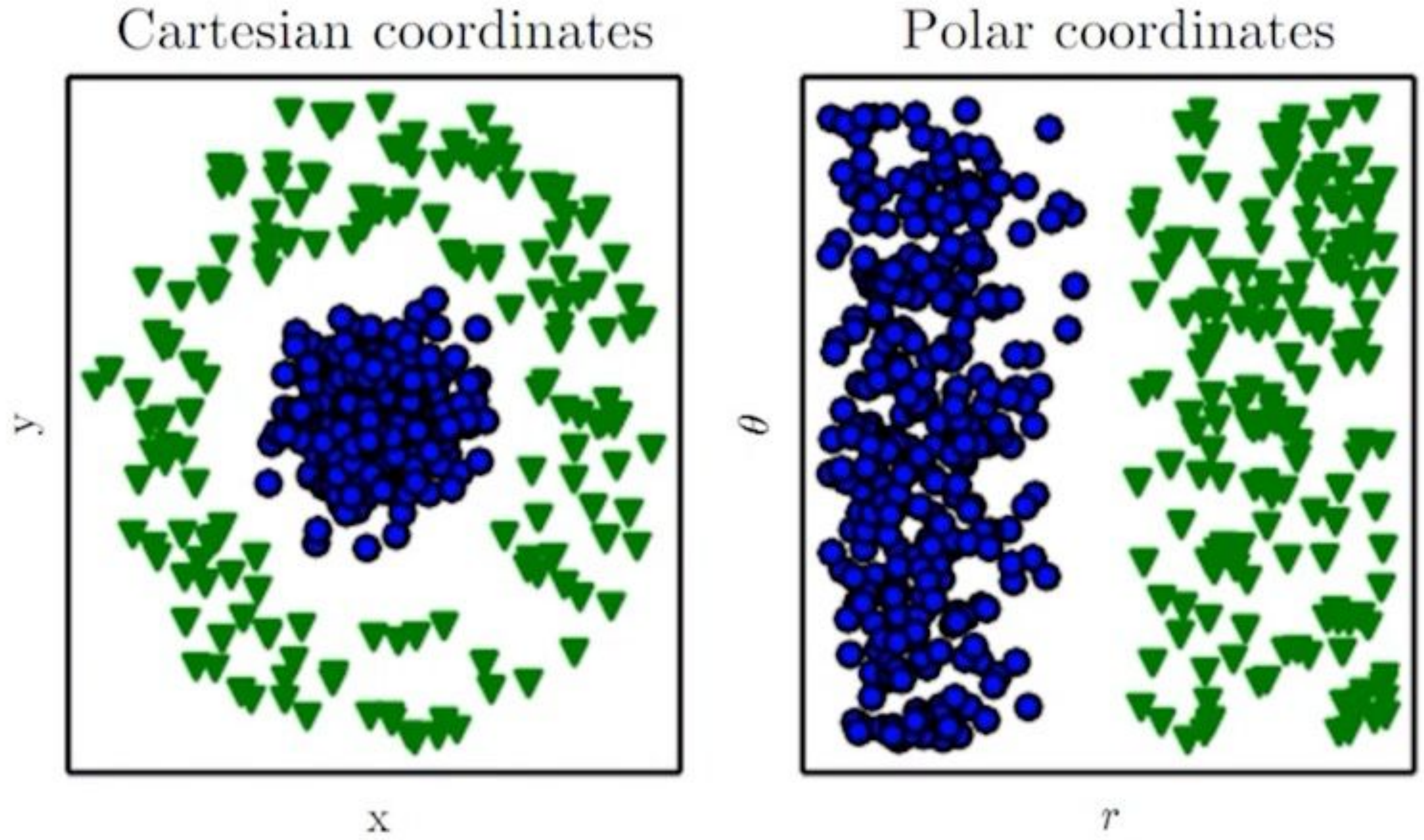


Higher layers, higher levels of abstraction

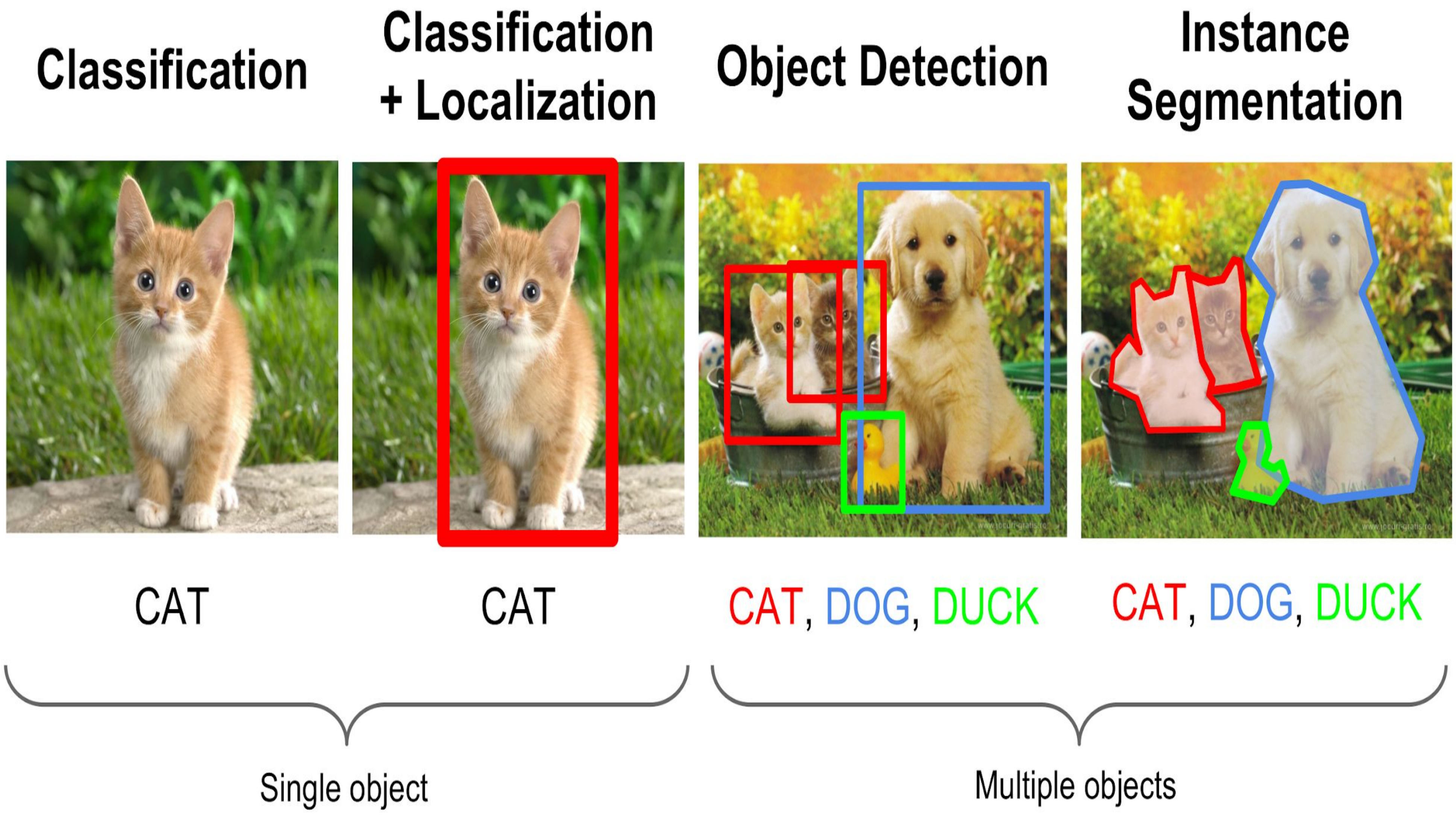
Yeah, but… how?

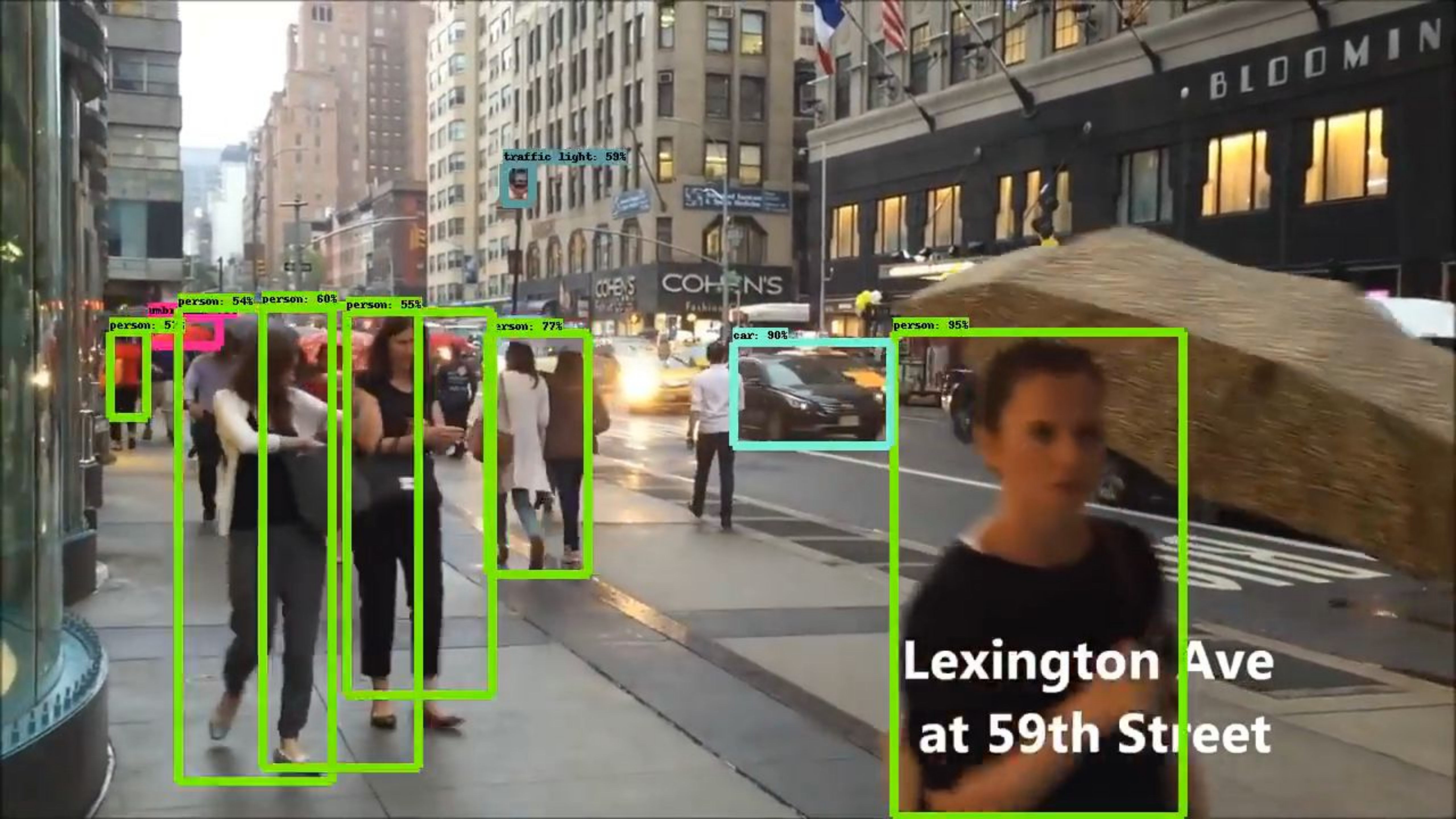


Yeah, but… how?



What this course is about




BLOOMIN
traffic light: 59%
COHEN'S
person: 54%
person: 60%
person: 55%
person: 77%
car: 90%
person: 95%
Lexington Ave
at 59th Street

Course Rules And Etiquette

- Respect.
- Do your own work.
- Assume the best of people.

Evaluation

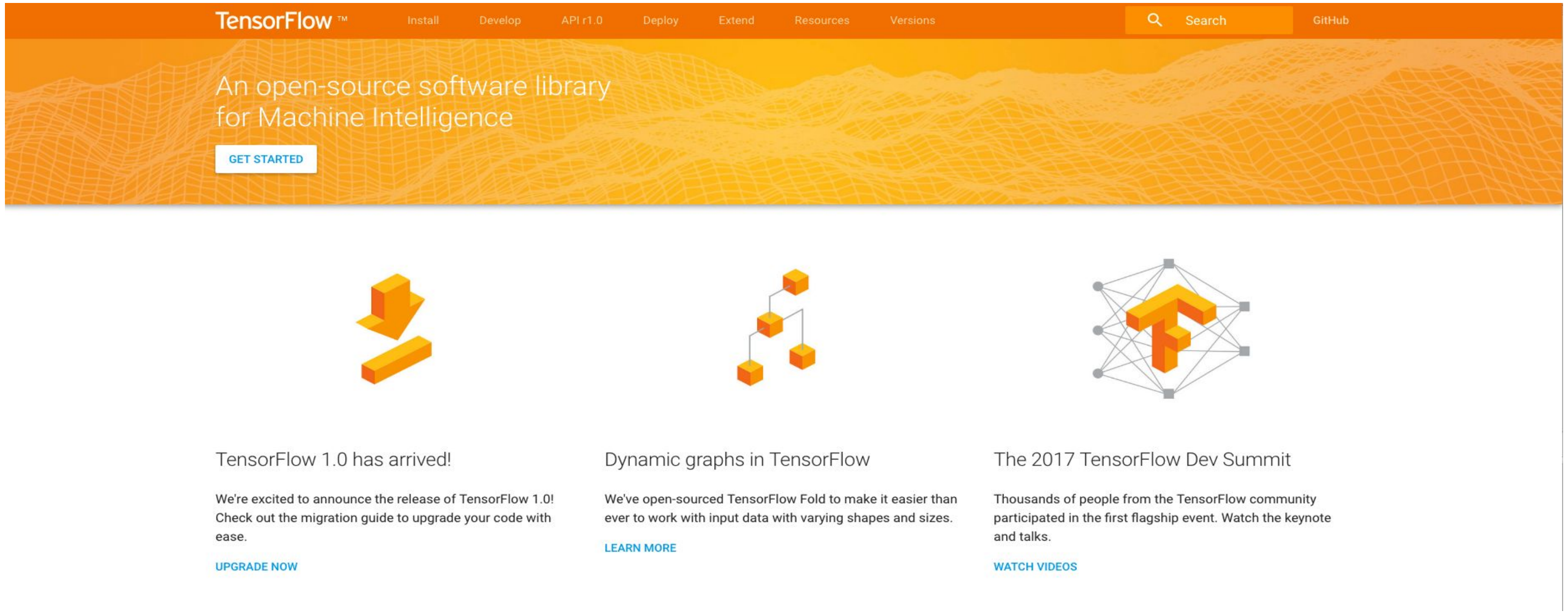
- Class Participation 20%
  - Forum, questions, optional exercises...
- Practice Workgroups 40%
  - Practices.
- Intermediate Exam 10%
- Final Exam 30%
  - Session 15. Test type.
  - You should get at least 3.5 out of 10, otherwise you will have an additional individual work.

Syllabus

SESSION #	SESSION NAME	TYPE	PRACTICES
1	Machine Learning Ingredients	F2F	
2	Logistic Regression & Support Vector Machines	F2F	
3	Artificial Neural Networks. Fundamentals	F2F	
4	Deep Learning. L-Layer ANN’s I	F2F	
5	Deep Learning. L-Layer ANN’s II	F2F	
6	Deep Learning. L-Layer ANN’s III	F2F	
7	Convolutional Neural Networks	F2F	
8	Convolutional Neural Networks II	F2F	P1 Announcement
9	Convolutional Neural Networks III	F2F	
10	LSTMs	F2F	
11	Practice Session: LSTMs	F2F	
12	Unsupervised Learning: Autoencoders / GANs	F2F	P2 Announcement
13	Practice Session: GANs	F2F	
14	Final Recap.	F2F	
15	Final Exam	F2F	

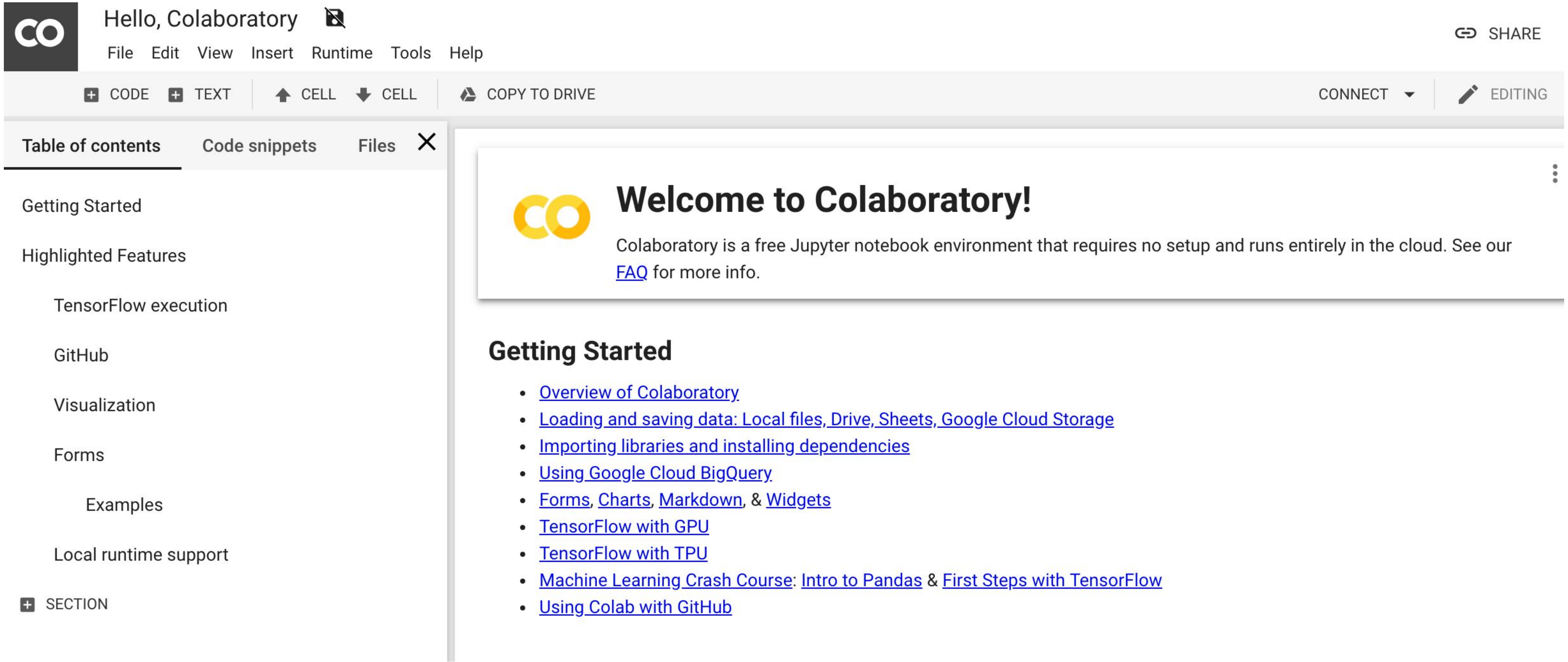
Practice Sessions

- Mark 1 Perceptron – 1960
- Torch – 2002
- CUDA – 2007
- Theano – 2008
- Caffe – 2014
- DistBelief – 2011
- TensorFlow 0.1 – 2015
- PyTorch 0.1 – 2017
- TensorFlow 1.0 – 2017
- PyTorch 1.0 – 2017
- TensorFlow 2.0 – 2019



Practice Sessions

- Google Colaboratory: https://colab.research.google.com/notebooks/welcome.ipynb



Main Goals of this class

Unbox linear regression: machine learning is not magic

What is machine learning?

Field of study that gives computers the ability to learn without being explicitly programmed.
Arthur Samuel (1959).

What is machine learning?

Field of study that gives computers the ability to learn without being explicitly programmed.
Arthur Samuel (1959).

Instead of writing task-specific programs by hand, we build algorithms able to learn from existing cases (i.e. e-mail spam classifier algorithm).

What is machine learning?

Field of study that gives computers the ability to learn without being explicitly programmed.
Arthur Samuel (1959).

Instead of writing task-specific programs by hand, we build algorithms able to learn from existing cases (i.e. e-mail spam classifier algorithm).

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.
Tom Mitchell (1998).

What is machine learning?

A Machine Learning algorithm:

1. Learn: The machine learning algorithm is able to learn from examples (i.e. right labeled spam e-mails into spam or not; we know the ground truth) to take decisions.

What is machine learning?

A Machine Learning algorithm:

1. Learn: The machine learning algorithm is able to learn from examples (i.e. right labeled spam e-mails into spam or not; we know the ground truth) to take decisions.

2. Response to unseen examples: The learning process allows the algorithm to take a decision over new samples (i.e. to decide is a new e-mail is spam or not).

What is machine learning?

A Machine Learning algorithm:

1. Learn: The machine learning algorithm is able to learn from examples (i.e. right labeled spam e-mails into spam or not; we know the ground truth) to take decisions.

2. Response to unseen examples: The learning process allows the algorithm to take a decision over new samples (i.e. to decide is a new e-mail is spam or not).

3. Is adaptive: if the examples change, the algorithm must change too by re-training with these new data (i.e. new e-mails are in other language than those used for first training).

Ingredients of ML

Classification problem: Cats vs Dogs.

Ingredients of ML

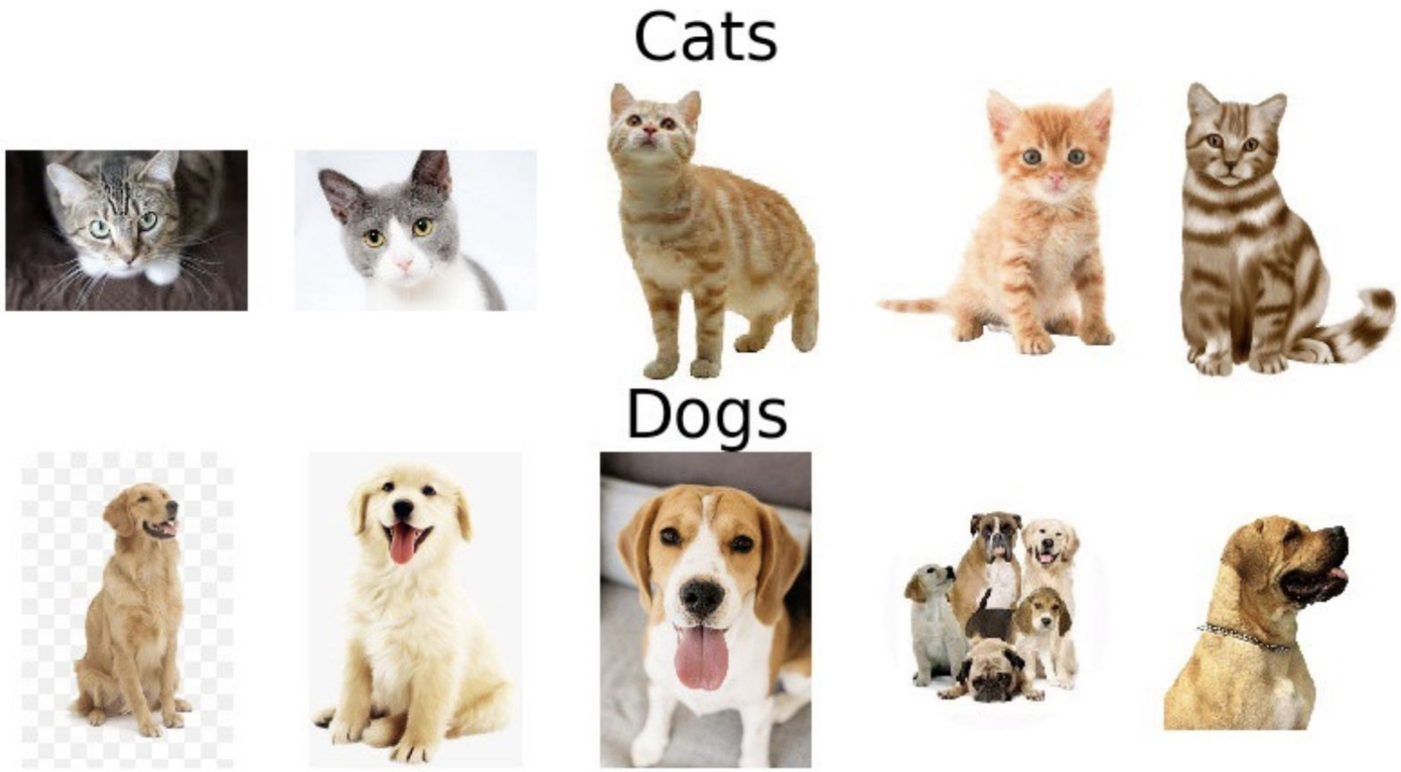
Classification problem: Cats vs Dogs.

1: Inputs and Outputs

Ingredients of ML

Classification problem: Cats vs Dogs.

1: Inputs and Outputs



Ingredients of ML

Classification problem: Cats vs Dogs.

2: Mapping function

(model)
(hypothesis)

X
input

y = F(X)
The mapping function

Y
output

1.0 DOG

0.0 CAT

Ingredients of ML

Classification problem: Cats vs Dogs.

2: Mapping function

(model)
(hypothesis)

X input	y = F(X) The mapping function	Y output
		0.4 DOG 0.6 CAT

Ingredients of ML

Classification problem: Cats vs Dogs.

1: Inputs and Outputs. x and y

2: Mapping Function. Something to get y from x.

3: Cost function. How well am I doing?

0.4 DOG

0.6 CAT

WRONG! this should be →

1.0 DOG → ERROR = 0.6

0.0 CAT

Ingredients of ML

Classification problem: Cats vs Dogs.

1: Inputs and Outputs. x and y

2: Mapping Function. Something to get y from x.

3: Cost function. How well am I doing?

4: The learning procedure. a.k.a: The Magic

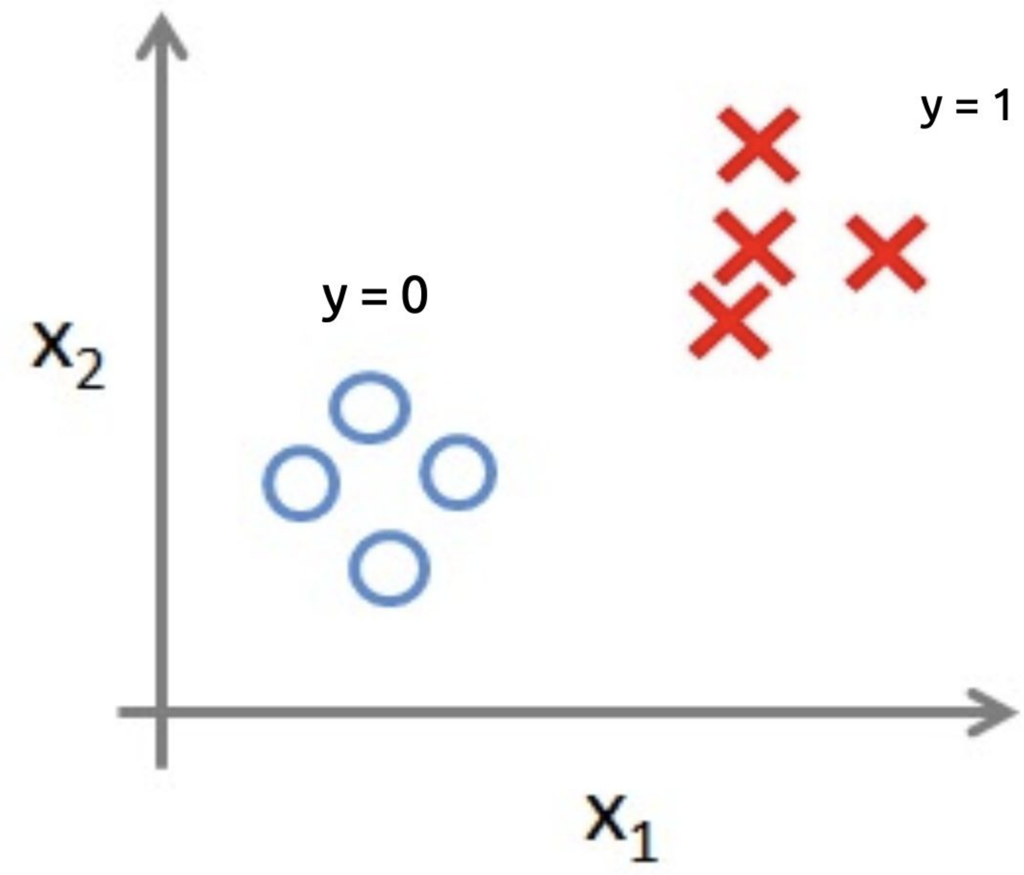
– Something that minimizes the cost function

- By changing and adapting the mapping function

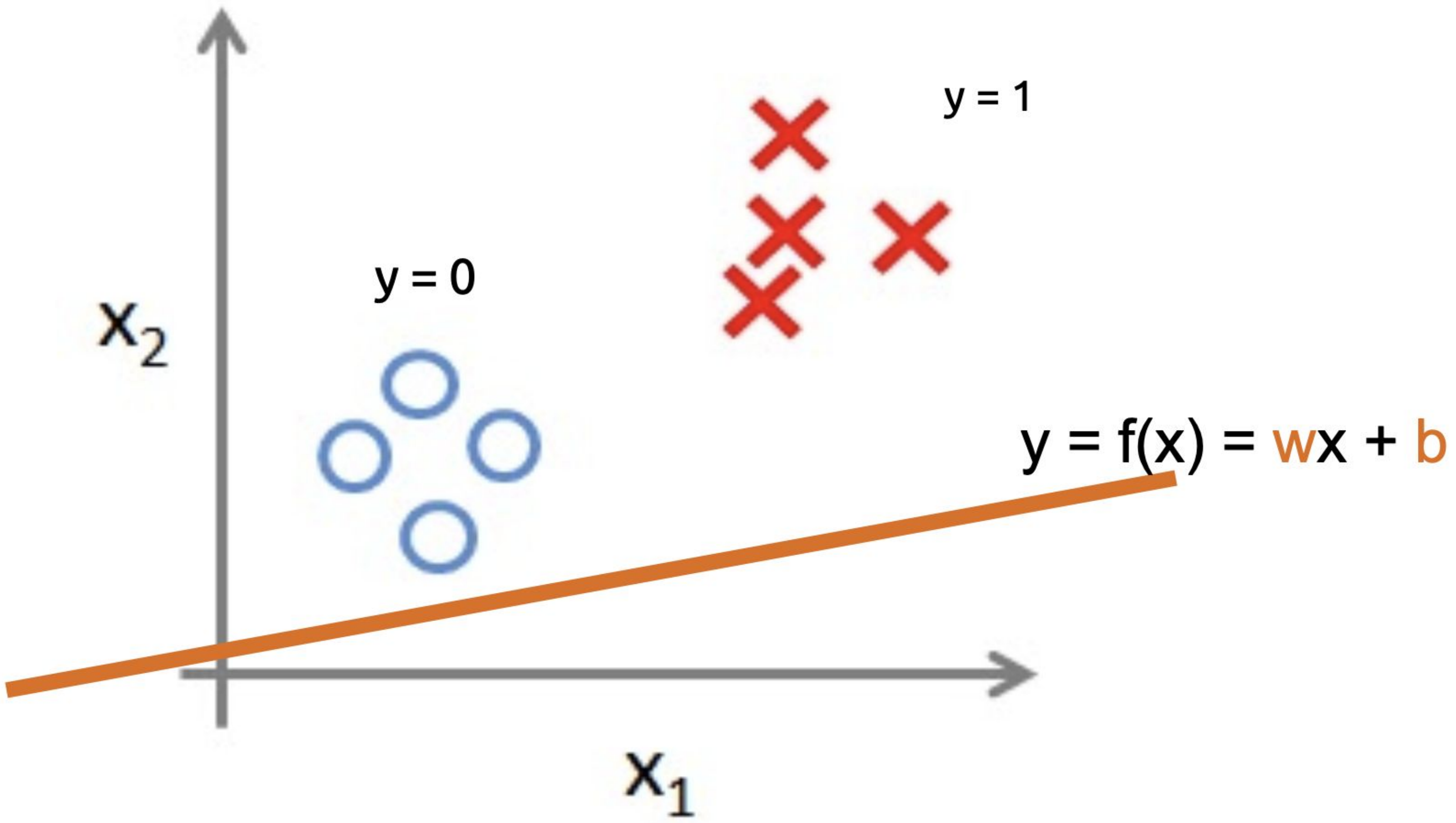
The Learning Process

Learning:
Minimize the error on the training data
adjusting the mapping function

The Learning Process

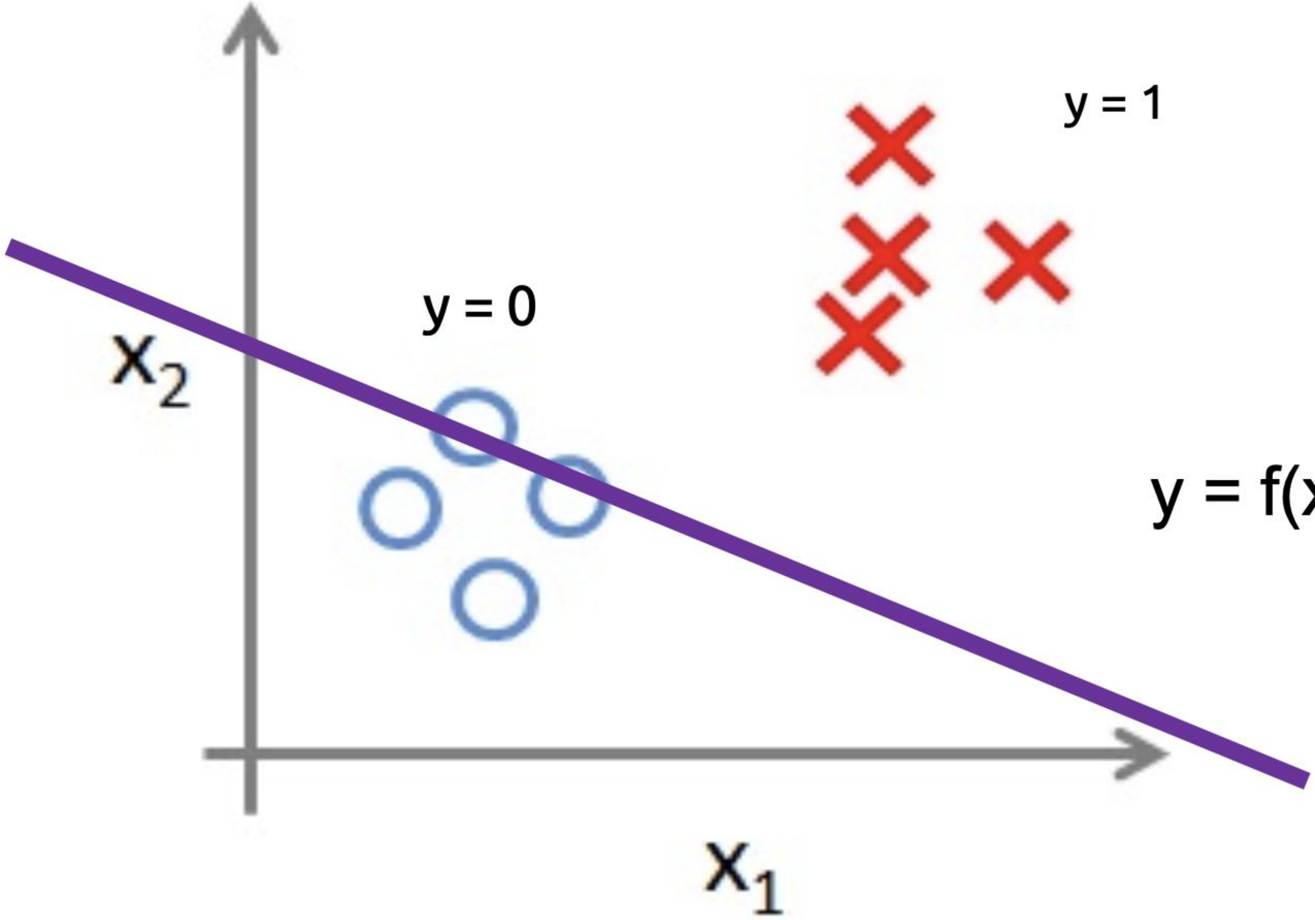


The Learning Process



COST/ERROR =
HIGH

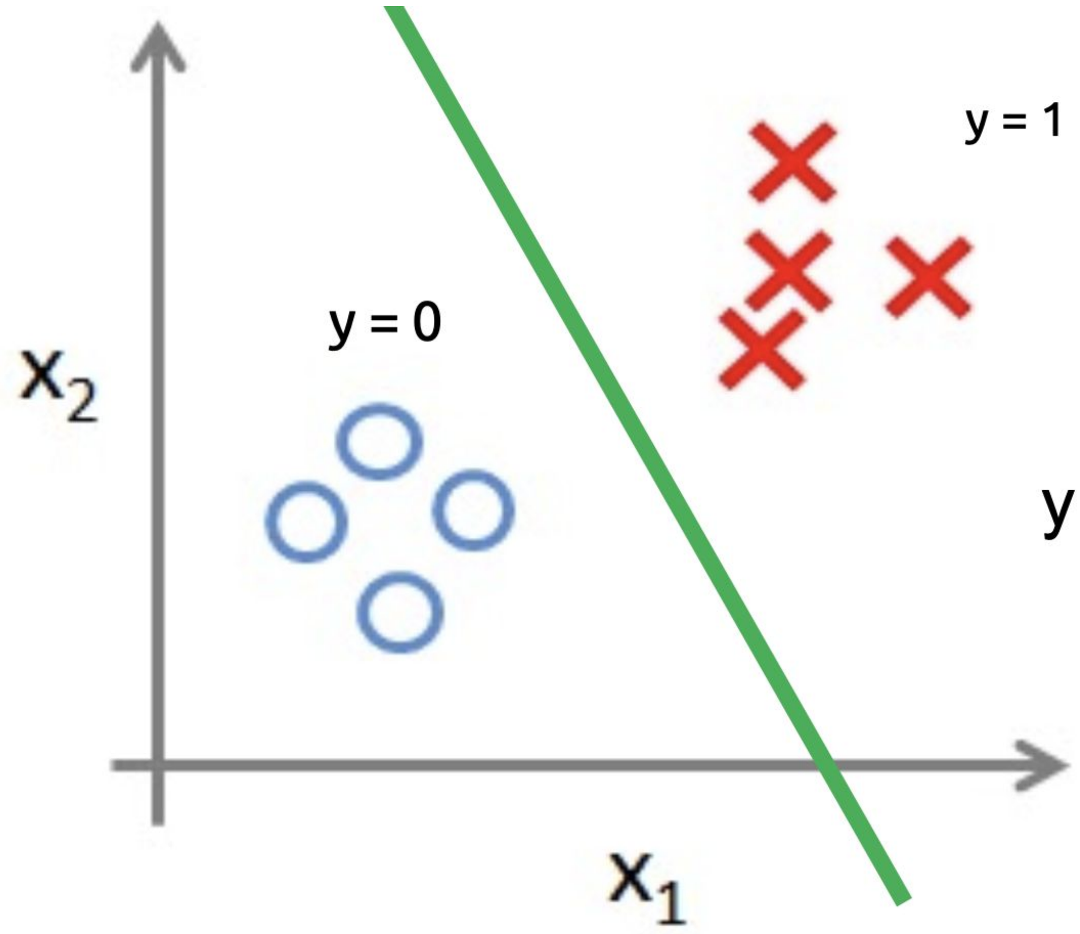
The Learning Process



$y = f(x) = wx + b$

COST/ERROR = MEDIUM

The Learning Process



$$y = f(x) = wx + b$$

COST/ERROR = LOW

The Learning Process

Learning:

Minimize the cost function (the error on the training data) by changing the trainable params and therefore adjusting the mapping function

Revisiting linear regression

Basic elements/ingredients on Machine Learning.

- Inputs & Outputs

$$x,\ y$$

Revisiting linear regression

Basic elements/ingredients on Machine Learning.

- Inputs & Outputs

$$x,\ y$$

- Mapping function (model/hypothesis)

$$h_w(x) = y = wx + b$$

Revisiting linear regression

Basic elements/ingredients on Machine Learning.

- Inputs & Outputs

$$x,\ y$$

- Mapping function (model/hypothesis)

$$h_w(x) = y = wx + b$$

- Loss/Cost function

$$J(w)$$

Revisiting linear regression

Basic elements/ingredients on Machine Learning.

- Inputs & Outputs

$$x, \; y$$

- Mapping function (model/hypothesis)

$$h_w(x) = y = wx + b$$

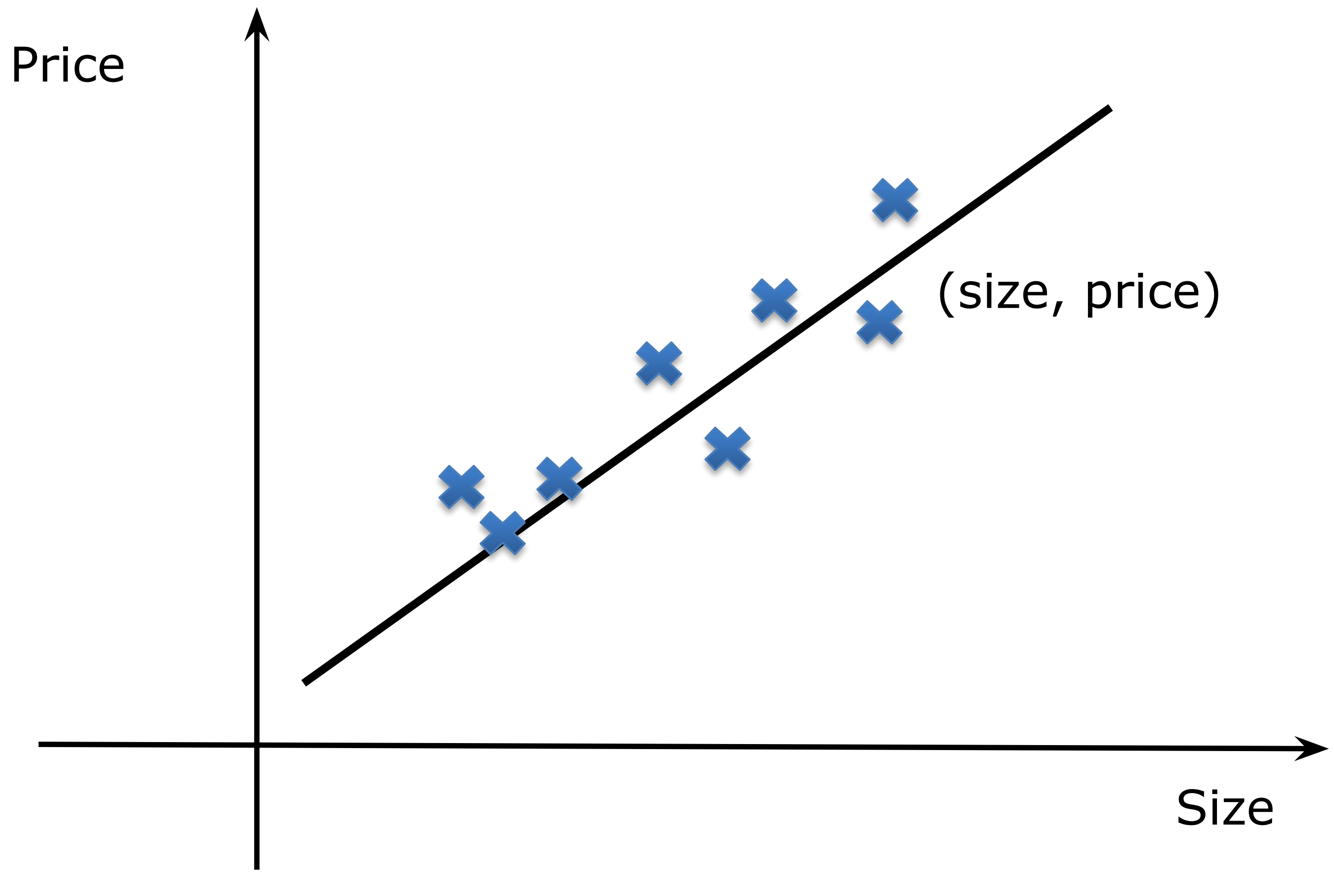
- Loss/Cost function

$$J(w)$$

- Learning procedure/algorithm

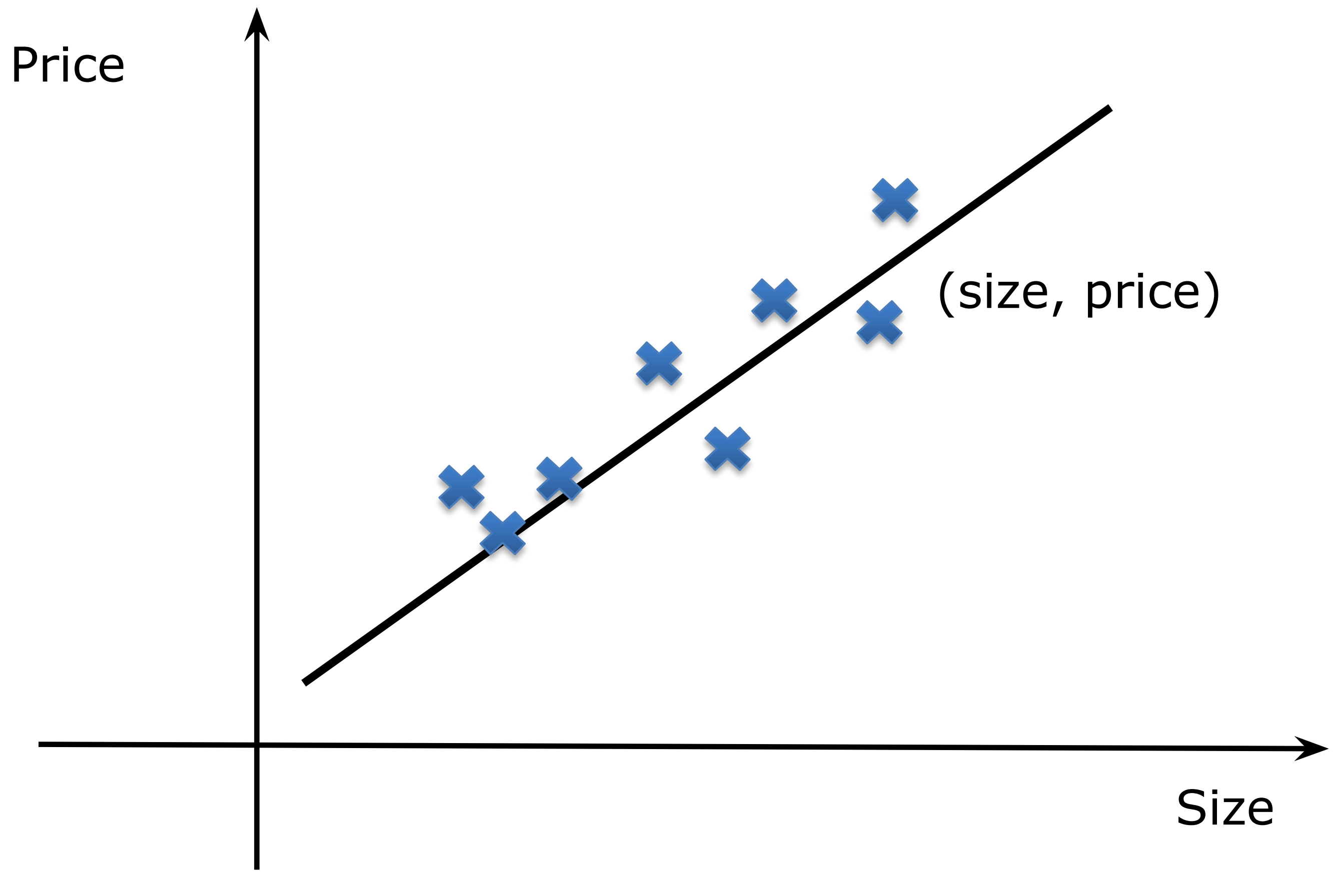
Revisiting linear regression

From some training data (size, price) we can make an algorithm that creates a model to fit the training data. So then, we can use that model to predict the price given new sizes.



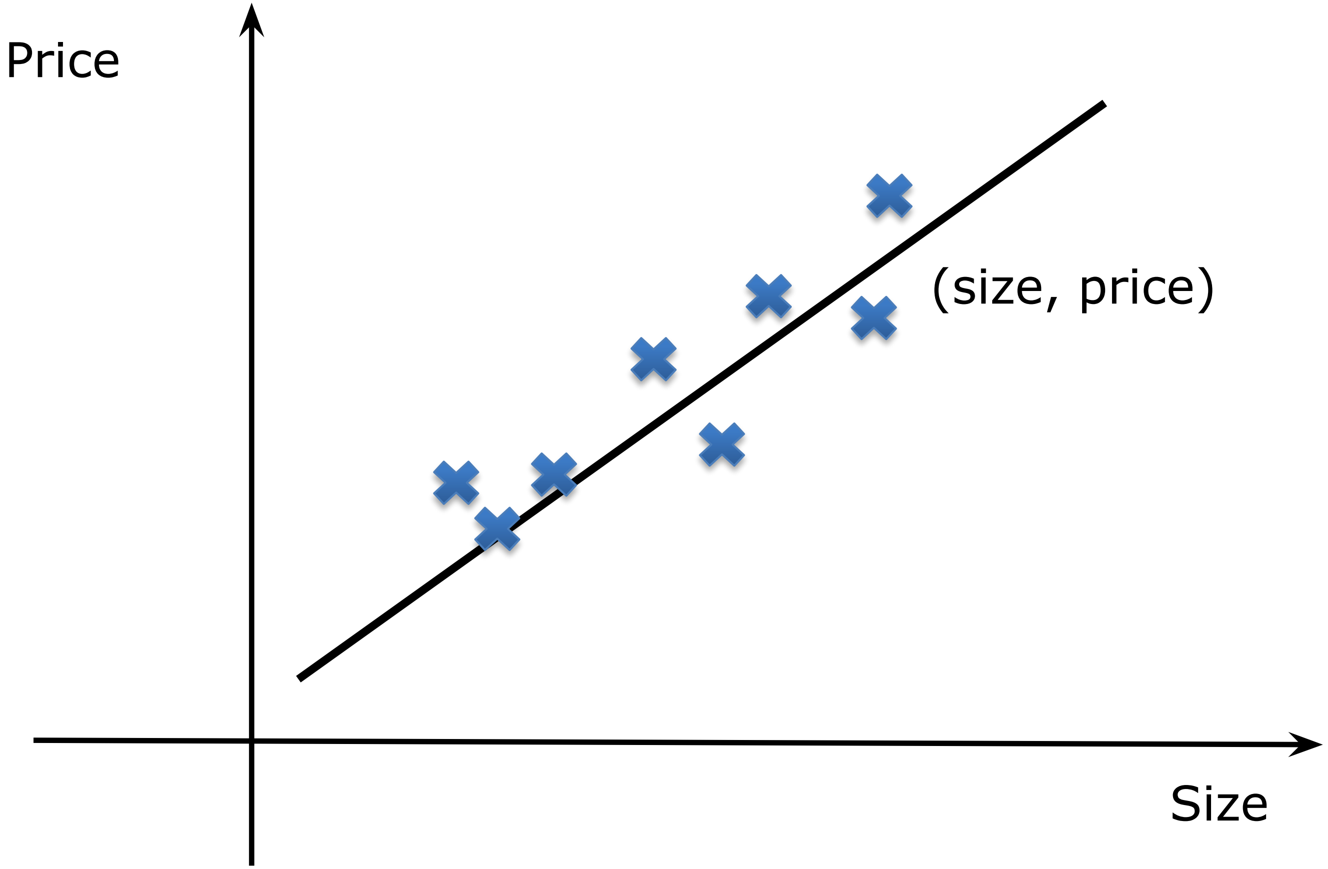
Revisiting linear regression

That means the algorithm learns from the training data



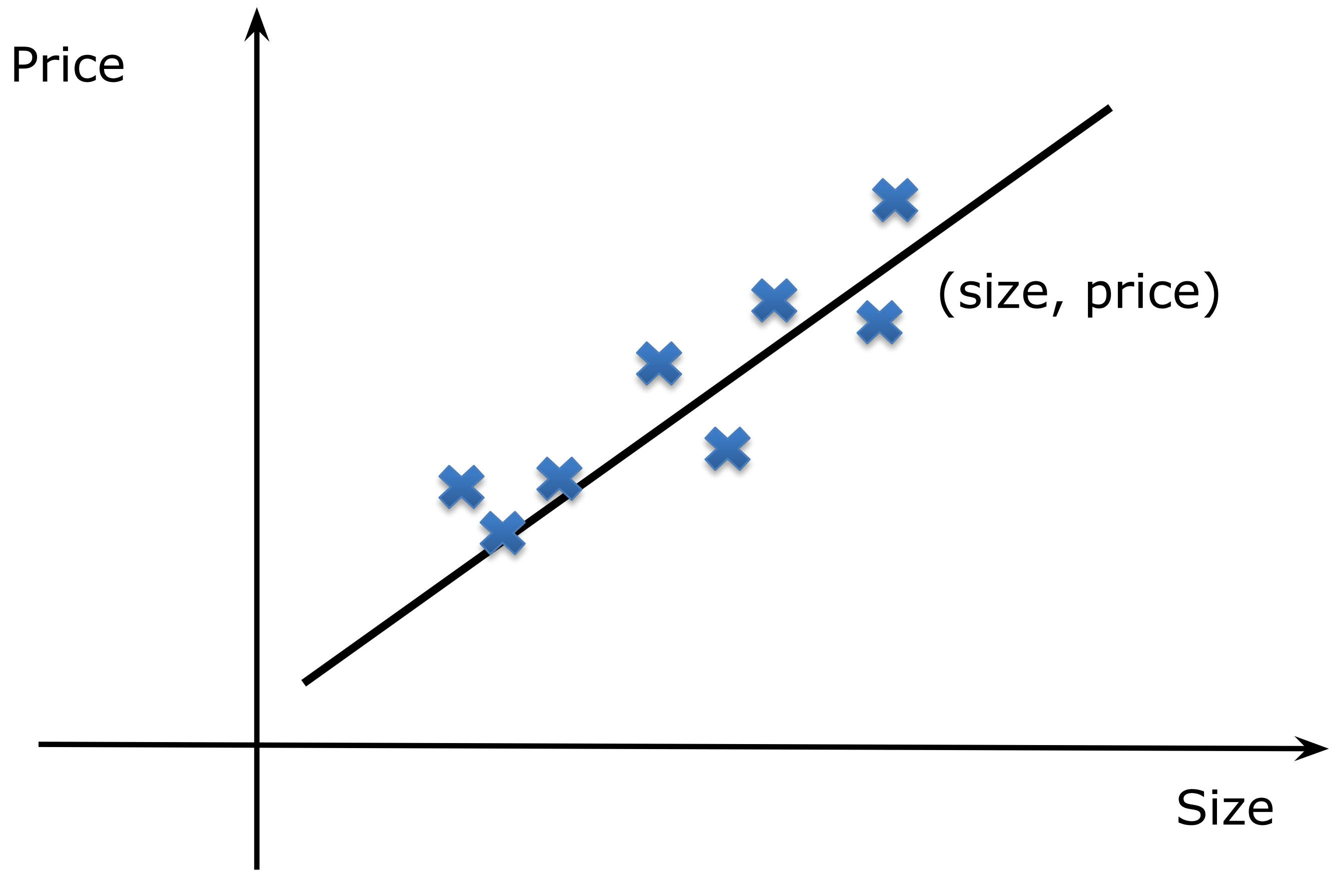
Revisiting linear regression

The algorithm needs the training data to learn. This training data
Is composed by the input and the actual output (ground truth). In this case,
real cases of the size and its associated price.



Revisiting linear regression

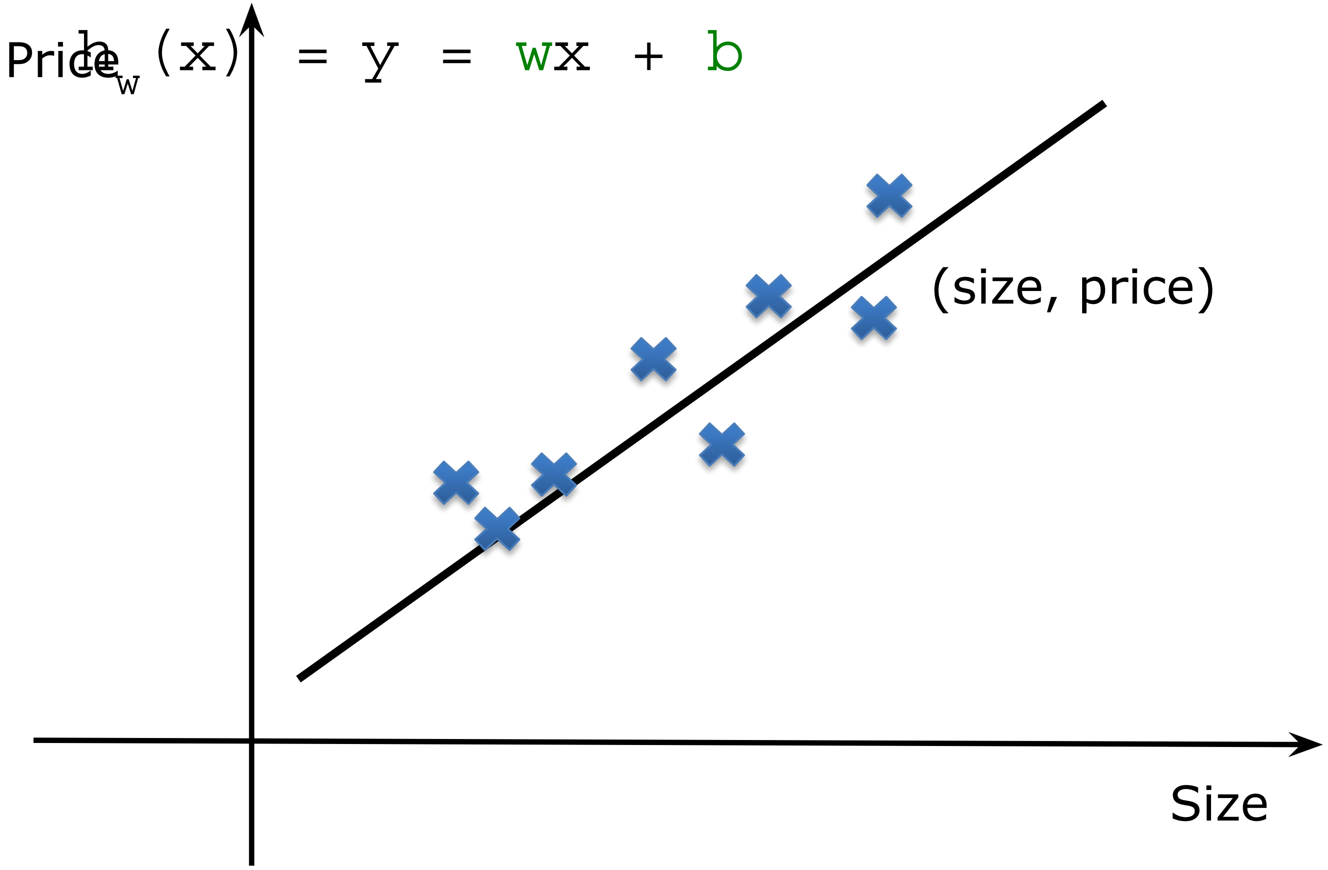
The process of learning consists of changing iteratively a given mathematical model to in every step fit better the training data – thus, minimizing the loss function



Revisiting linear regression

The algorithm modifies the model in every step by changing its parameters.
In this case the model is a straight line; therefore the model can be modified by changing the slope and the bias.

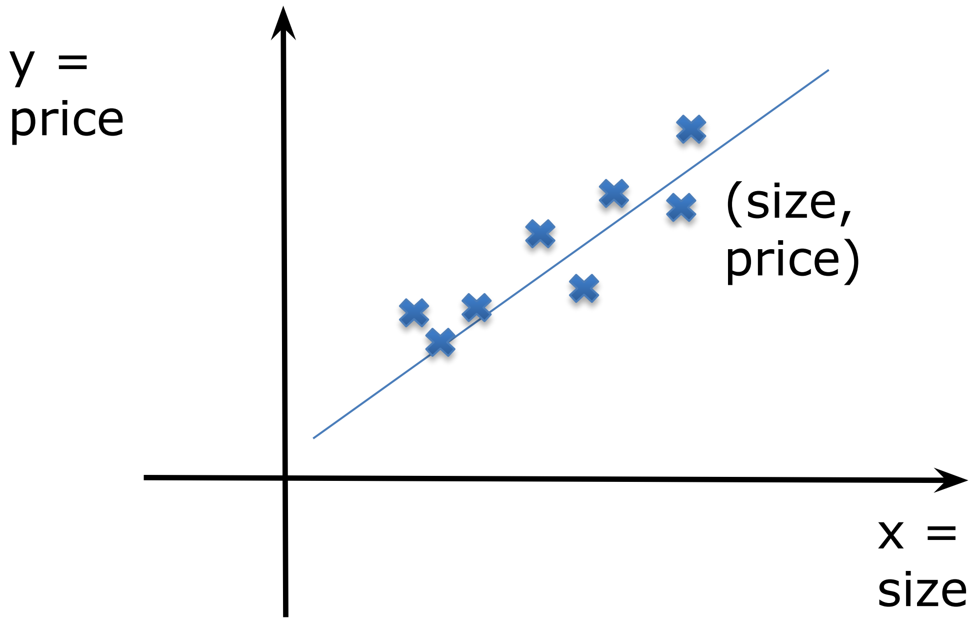
$$h_w(x) = y = wx + b$$



Revisiting linear regression

The algorithm modifies the model in every step by changing its parameters.
In this case the model is a straight line; therefore the model can be modified by changing the slope and the bias.

$$h_w(x) = y = wx + b$$



Who is Who

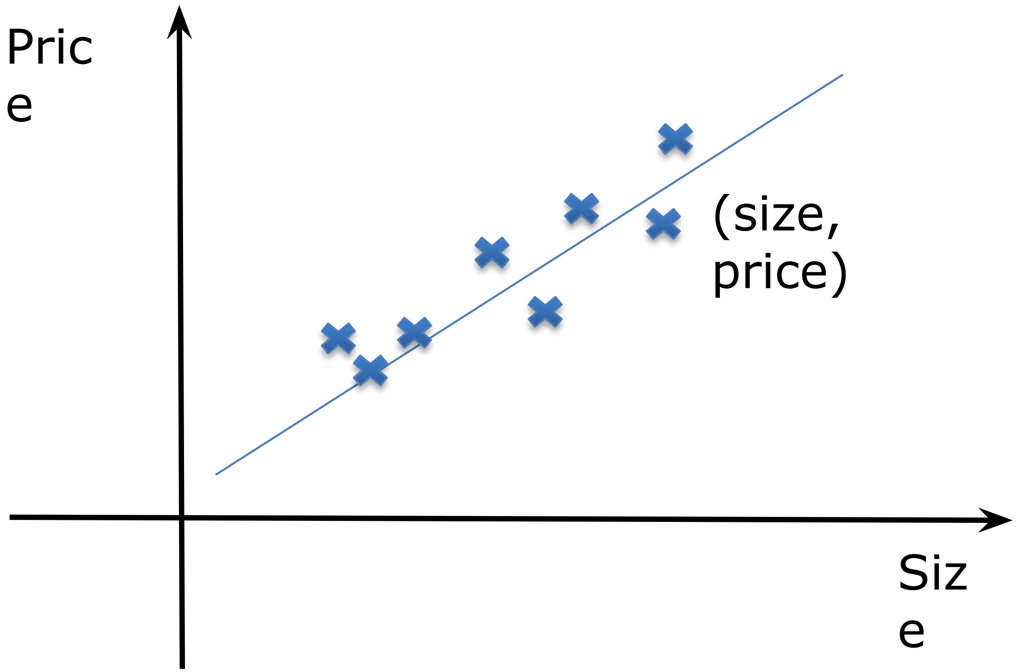
The model: straight line equation $y = w_1x + w_0$
The parameters: slope (w_1) and bias (w_0)
The output: the price (y)
The input: the size (x)

The algorithm **learns w_1 and w_0** (the parameters) to fit the training data (x, y)

Revisiting linear regression

The mapping function in linear regression (univariate)

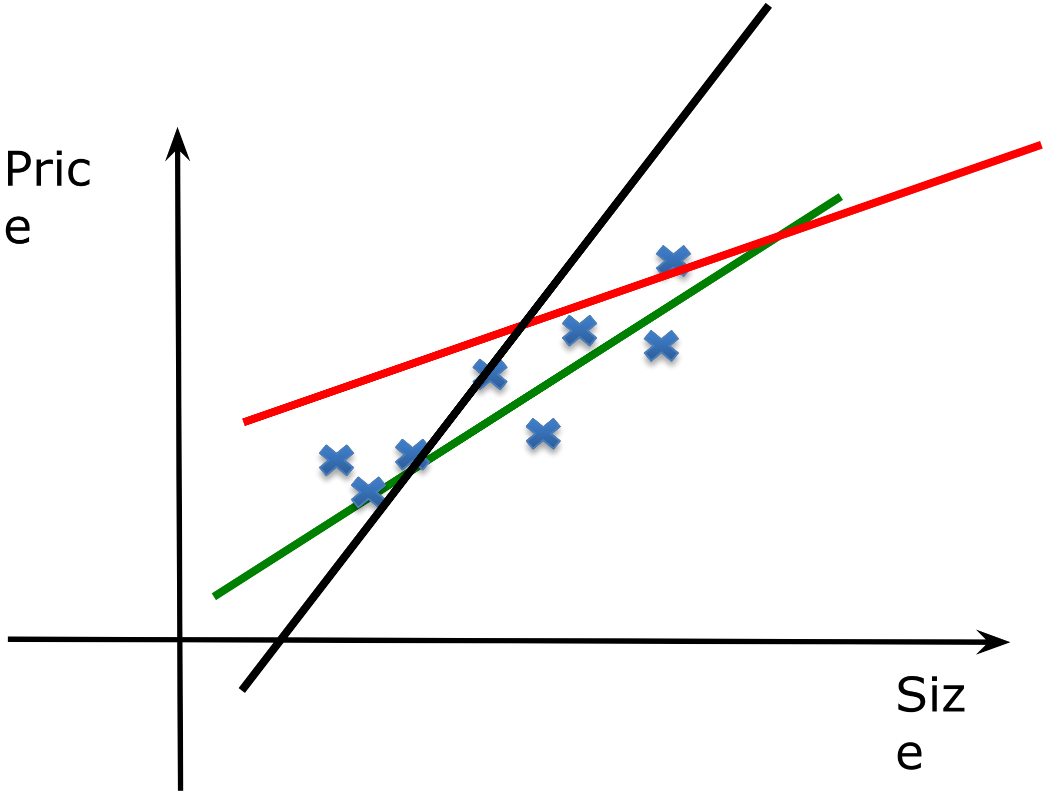
$$y = h_w(x) = wx + b$$



Revisiting linear regression

The problem then is to estimate the parameters of the mapping function to reach the most accurate output

$$y = h_w(x) = wx + b$$



Revisiting linear regression

Idea: Minimize the error between the estimated output and the actual output in the training set

$$\min_{w} \; J(w) = (h_{w}(x) - y)$$

Revisiting linear regression

Idea: Minimize the error between the estimated output and the actual output in the training set

$$\min_w \; J(w) = (h_w(x) - y)$$

The error: $(h_w(x) - y)$ is the difference between what our algorithm estimates and the real output.

And again, what our algorithm estimates depends on the parameters (w)

Revisiting linear regression

Idea: Minimize the error between the estimated output and the actual output in the training set

$$\min_w \ J(w) = (h_w(x) - y)$$

For all the training set (m samples) … just take the average

$$\min_w \ J(w) = 1/m \ \Sigma_m \ (h_w(x)^{(i)} - y^{(i)})$$

Revisiting linear regression

Idea: Minimize the error between the estimated output and the actual output in the training set

$$\min_w \quad J(w) = (h_w(x) - y)$$

For all the training set (m samples) …

$$\min_w \quad J(w) = 1/m \ \Sigma \ (h_w(x)^{(i)} - y^{(i)})$$

Wait, if we are going to minimize this function, let's make it quadratic

$$\min_w \quad J(w) = 1/2m \ \Sigma \ (h_w(x)^{(i)} - y^{(i)})^{2}$$

Revisiting linear regression

Why making it quadratic?

1. Overshooting or undershooting are both errors

$$(5 - 3)^2 = (1 - 3)^2$$

2. Larger errors are penalized

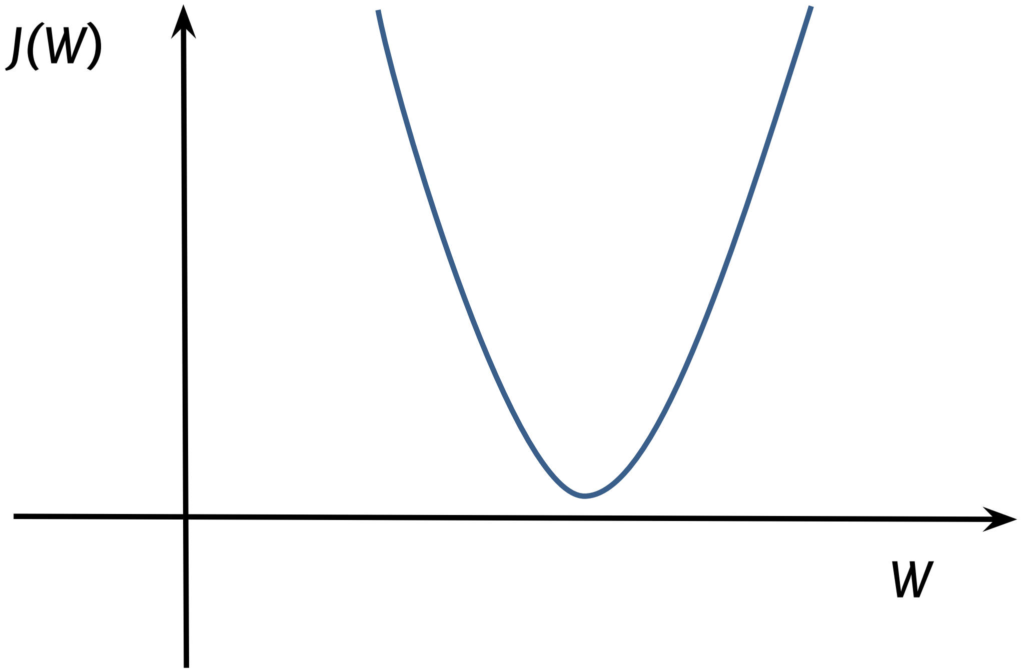
$$(10 - 3) > (5 - 3)$$
$$(10 - 3)^2 >> (5 - 3)\hat{}^2$$

3. Make the Loss function convex

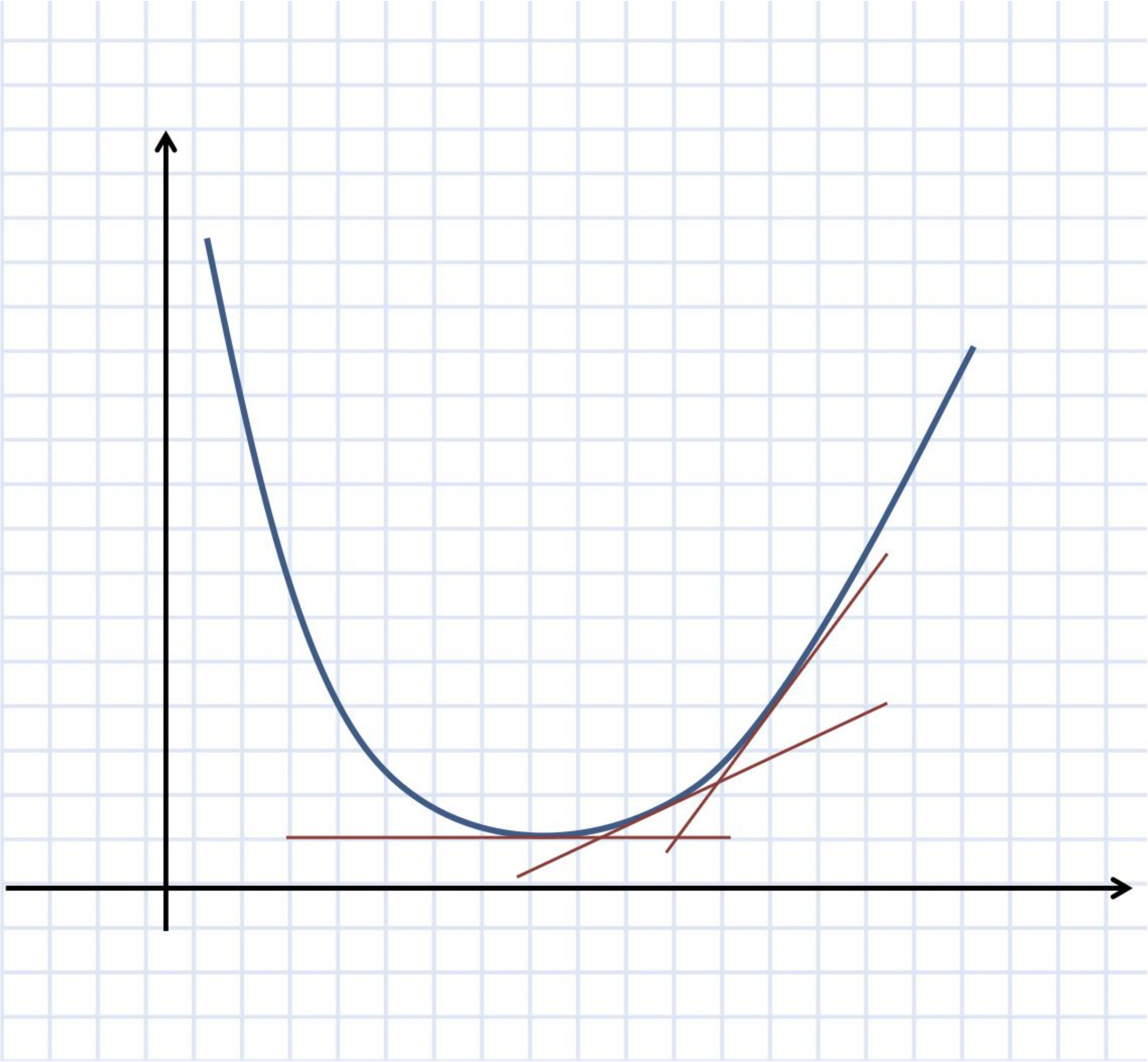
Revisiting linear regression

$$\min_w \quad J(\Theta) = 1/2m \ \Sigma \ (h_w(x)^{(i)} - y^{(i)})^2$$

$J(w)$ is a function of W while the mapping function is a function of the input x for a fixed w



Gradient Descent



1. Pick a random w (w^0)

2. Repeat until convergence {

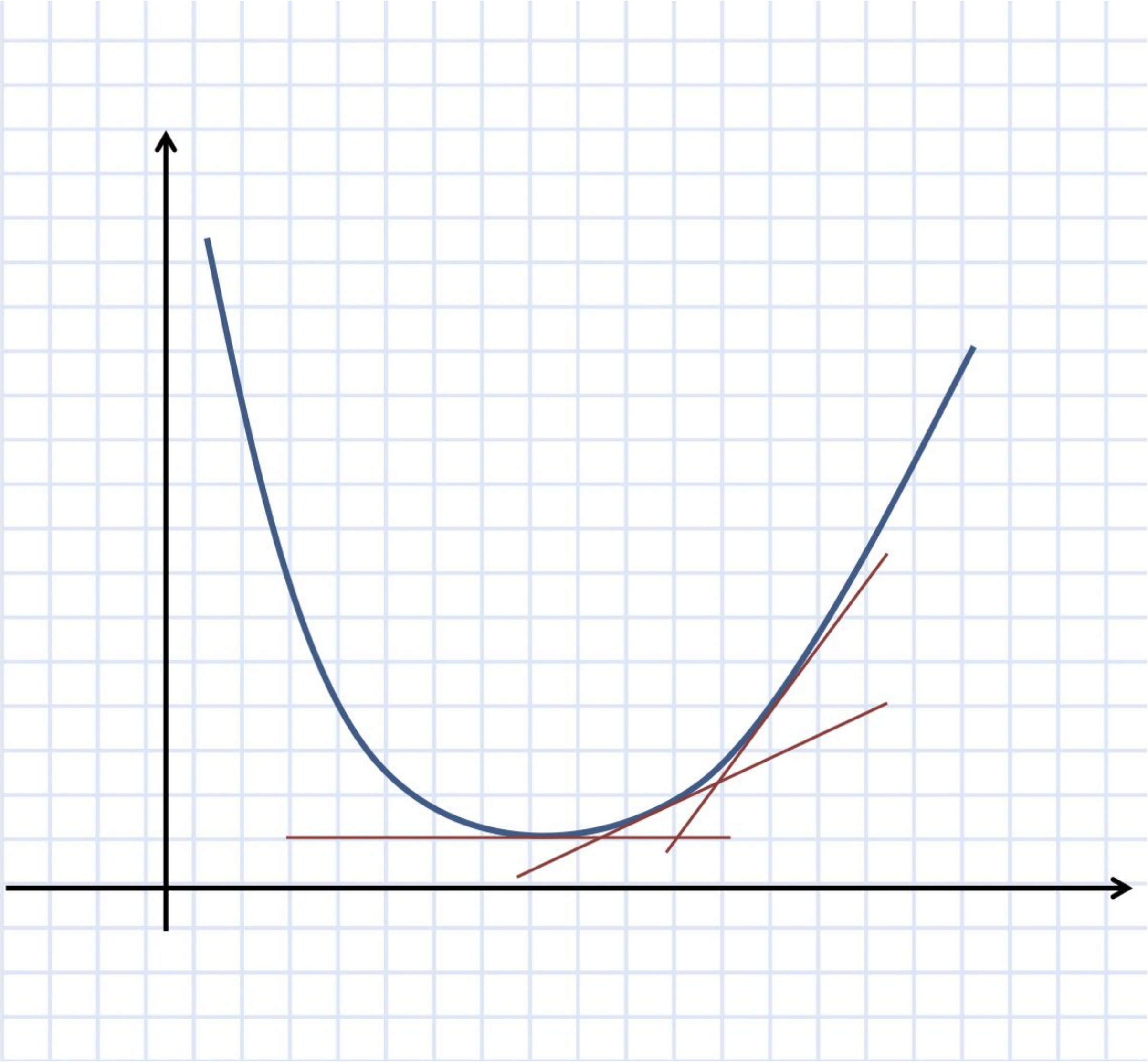
   $w^{i+1} = w^i - \alpha dL(w)/d(w)_{[w^i]}$

}

Gradient Descent

Gradient Descent is an iterative algorithm used as learning procedure to minimize cost functions in a fast and feasible way

Gradient Descent: What is that alpha?



1. Pick a random w (w^0)

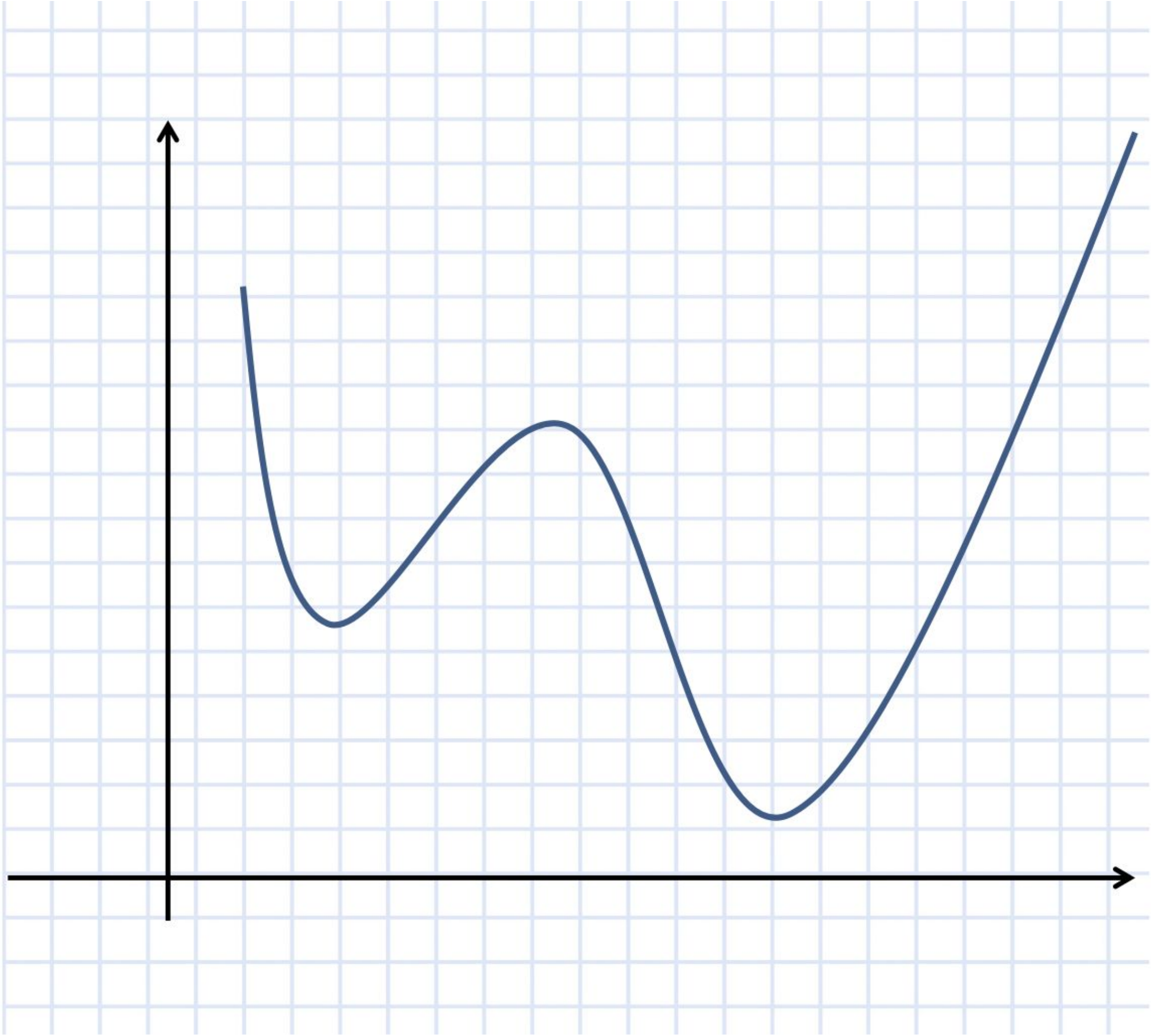
2. Repeat until convergence {

$$w^{i+1} = w^i - \alpha dL(w)/d(w)_{[w^i]}$$

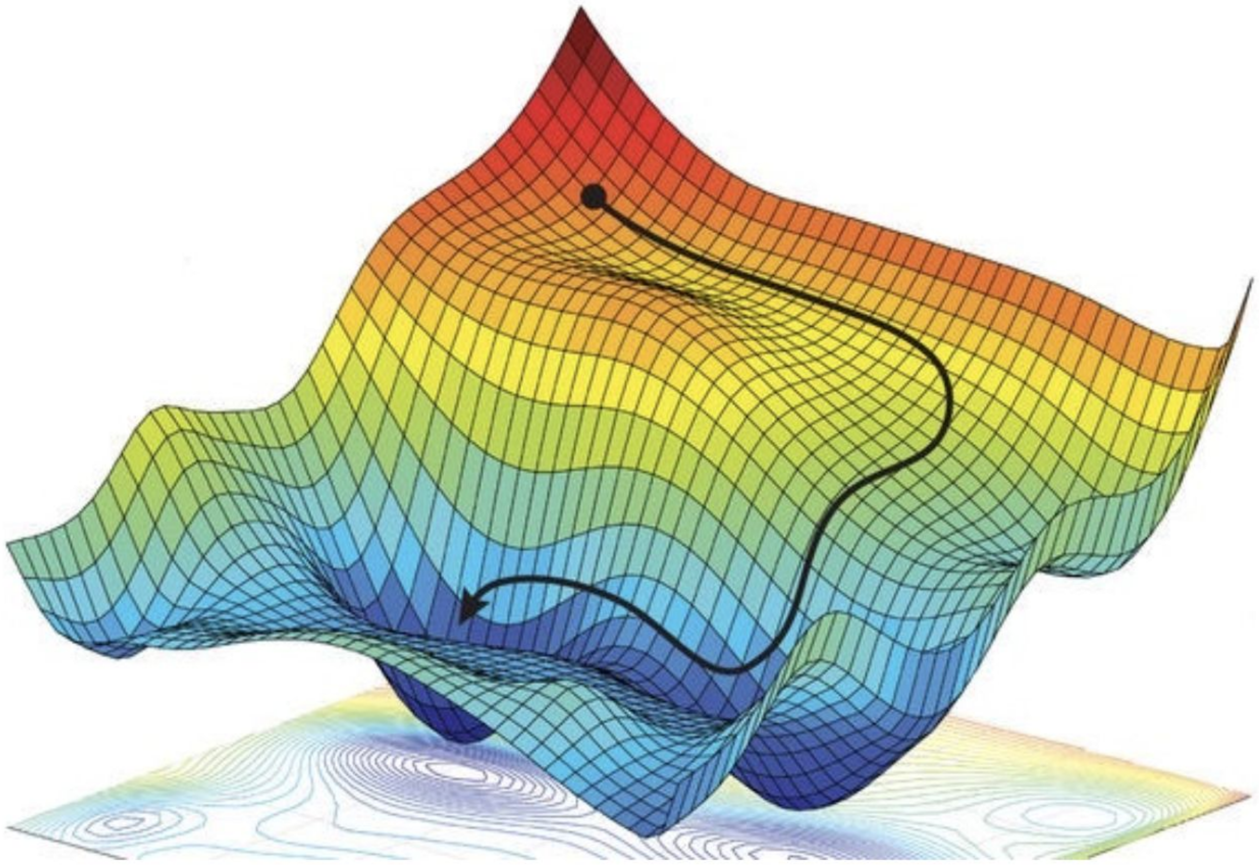
}

The learning rate role

The limits of Gradient Descent



Multivariate Gradient Descent



Revisiting linear regression! Easy!

Try. Basic elements/ingredients on house pricing Linear Regression problem?

- Inputs & Outputs
- Mapping function
- Loss function
- Learning process name

Revisiting linear regression! Easy!

Solution. Basic elements/ingredients on house pricing Linear Regression problem?

- Inputs & Outputs

  Input: x (the size); output y: the price

- Mapping function

$$h_w(x) = y = w_1 x + w_0$$

- Loss function

$$J(w) = 1/2m \ \Sigma \ (h_w(x)^{(i)} - y^{(i)})^2$$

- Learning process name

  Iterative Gradient Descent

Type of learning tasks?

Depending on the feedback available to learning:

1. Supervised learning: Learning is conducted through example inputs associated with their desired/right outputs. That is, training samples correctly labeled.

2. Unsupervised learning: There are no labels associated to the input examples. The task itself is to find patterns within the data or learning a better representation of them for the given task.

3. Reinforcement Learning: Learning feedback comes through rewards and punishments within a dynamic environment where a given goal is pursuit.

Type of learning tasks?

According to the type of the desired output:

1. Classification: Outputs are one or several classes to which the inputs are associated. Task consists of deciding which class correspond to the new unseen inputs.
2. Regression: Outputs are real continuous numbers rather than discrete values as in classification tasks.
3. Clustering: Set of inputs belongs to a given group, but unlike classification, those groups are not known beforehand.
4. Dimensionality reduction: Inputs are mapped into a lower dimensionality space, best suited from a given task