

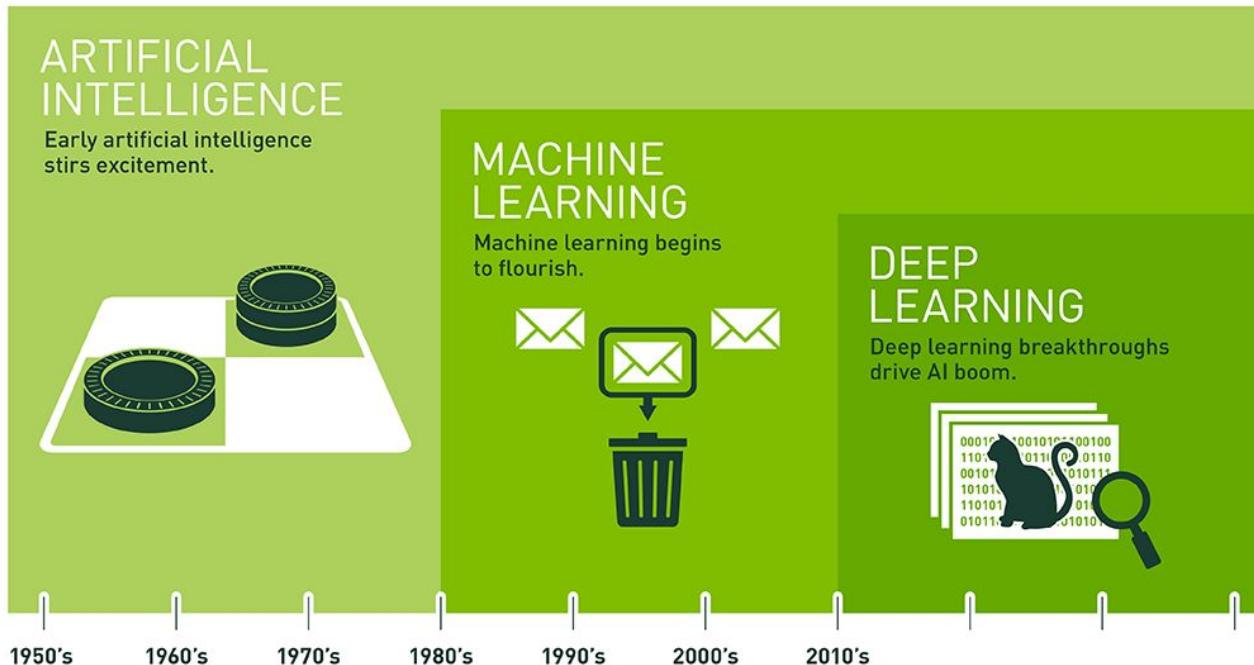
MACHINE LEARNING INTRODUCTION



Although AI has been around for decades, new advances have ignited a boom in practical applications through advancements in statistical techniques known as Deep Learning.

Deep learning powers self-driving cars, image/video recognition systems, natural language understanding, speech recognition/synthesis, life-saving advances in medicine and many more applications.

As an example, you are using DL-powered applications in Google, Facebook and Apple products that you use everyday.



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created larger disruptions.



DEMO ABOUT ▾ LOG IN

Lyrebird is back!

September 4th 2017

Lyrebird - Create a digital copy of your voice.   



THIS IS AN ARTIFICIAL VIDEO CREATED BY LYREBIRD.
IT DOES NOT CONVEY THE OPINION OF BARACK OBAMA.

Demos:

<https://soundcloud.com/user-535691776/dialog>
<https://www.youtube.com/watch?v=dkoi7sZvWiU>

DeepFakes

GAN

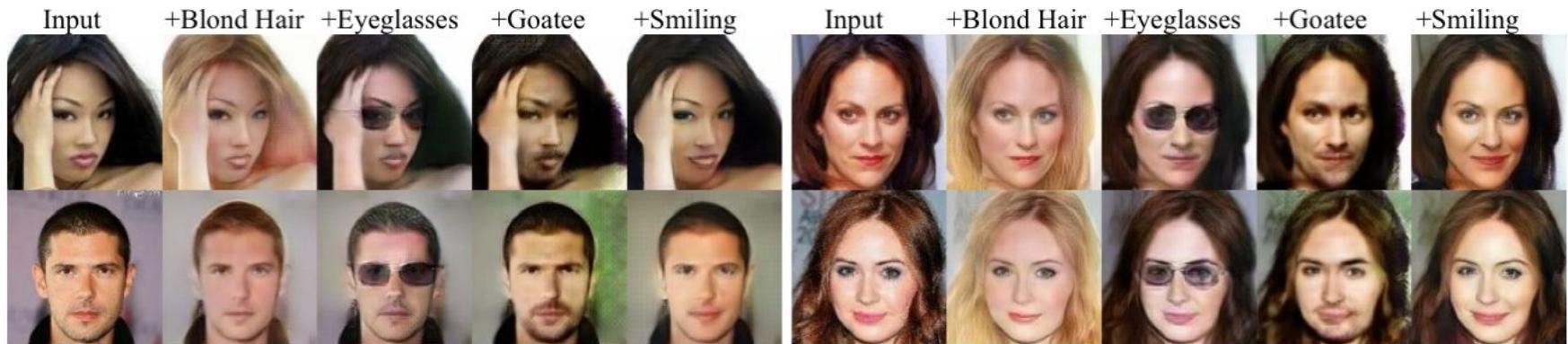
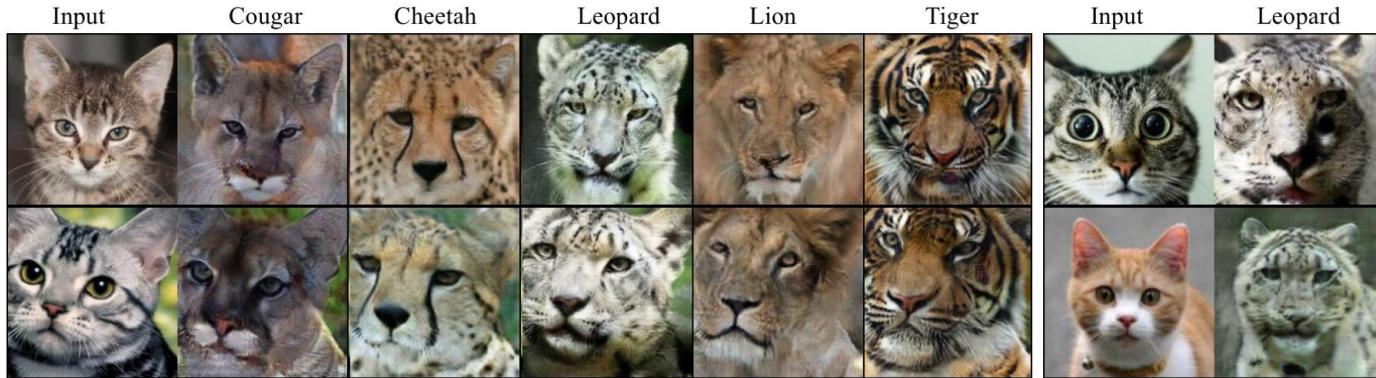


Nicholas Cage DeepFakes movie compilation



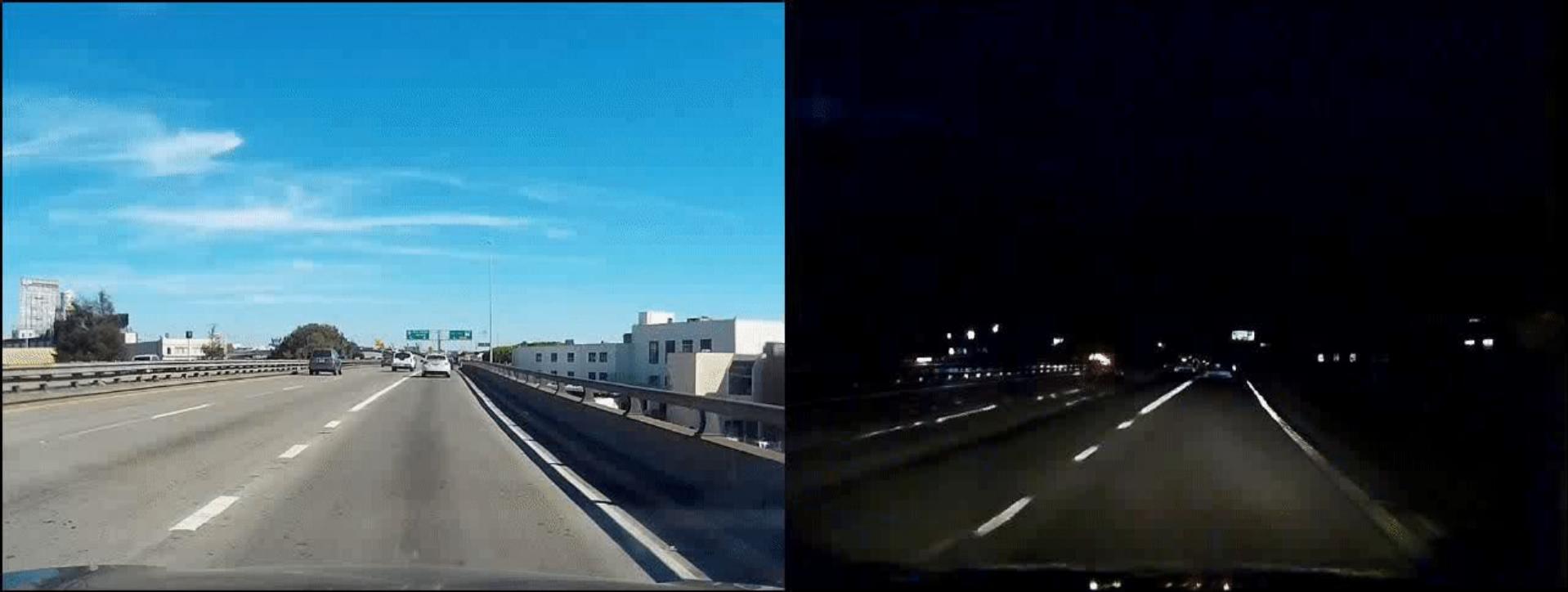
Nick Cage DeepFakes Movie Compilation

Demo: <https://youtu.be/BU9YAHigNx8?t=42s>



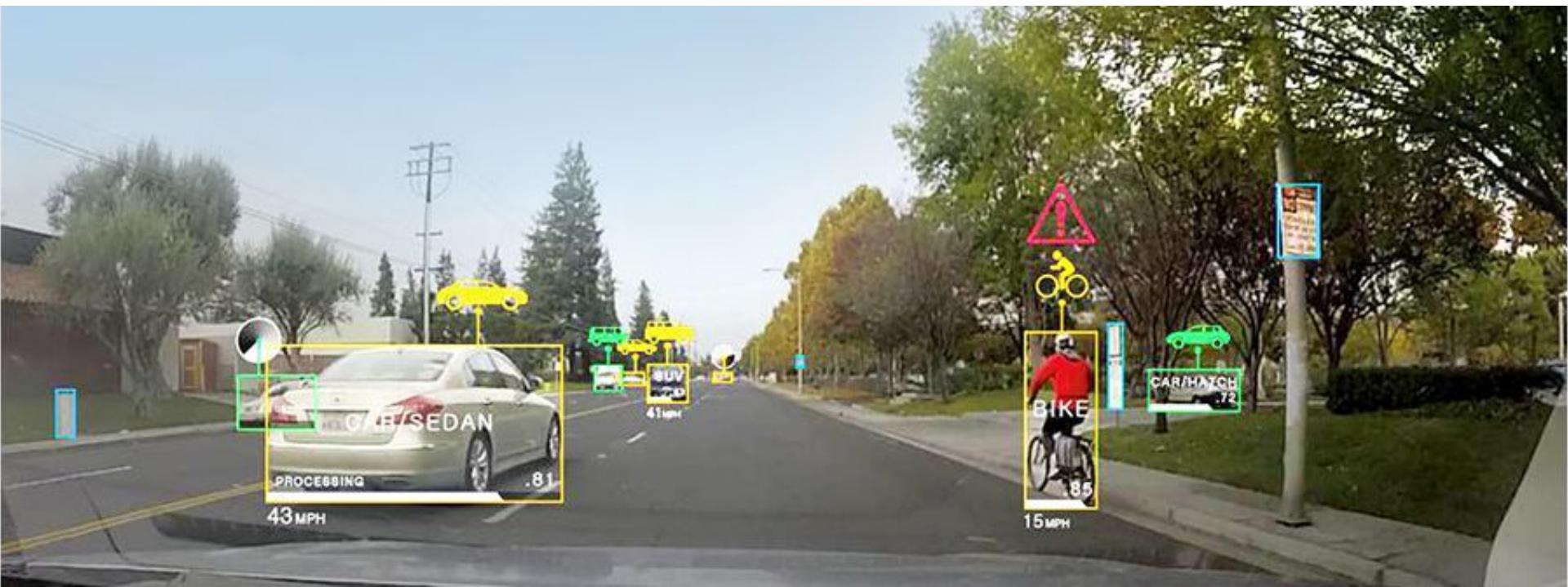


More info: <https://github.com/mingyuliutw/UNIT>



More info: <https://github.com/mingyuliutw/UNIT>

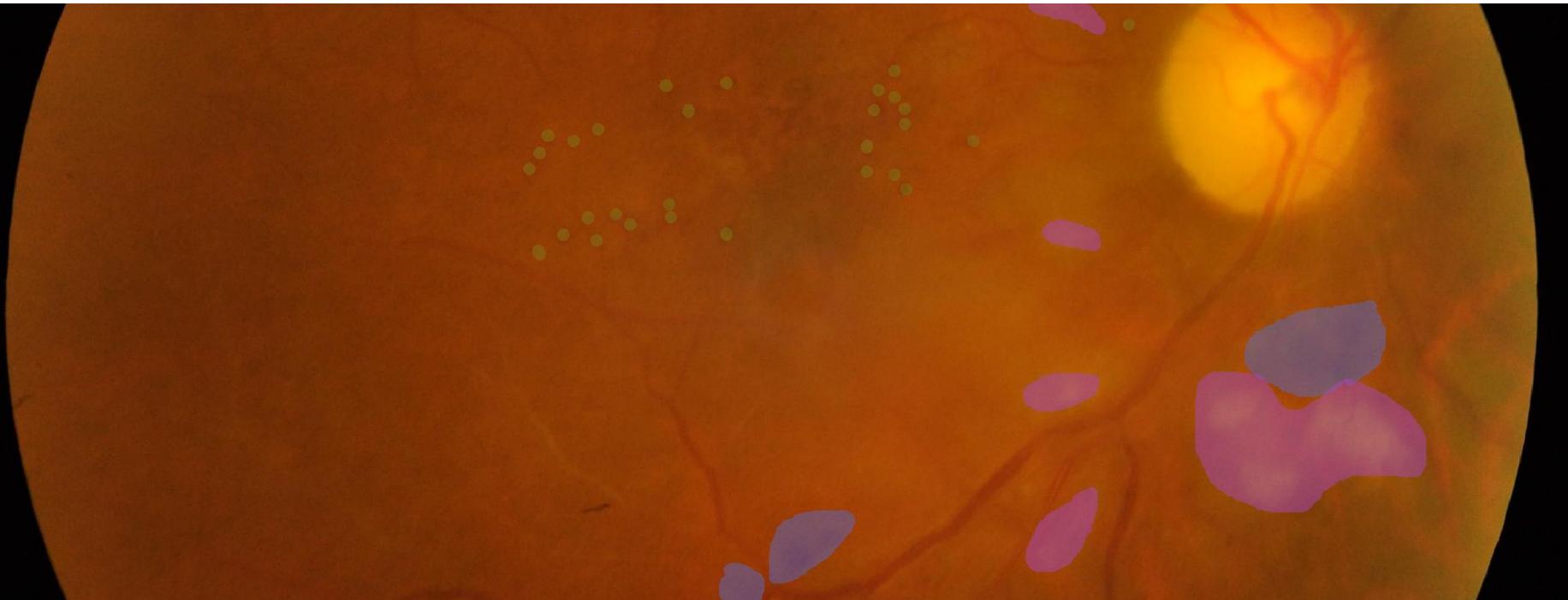
Self-driving cars



Text recognition



Medical applications



Google sunroof

1234 Bryant St, Palo Alto, CA 94301, USA X Q

✓ Analysis complete. Your roof has:

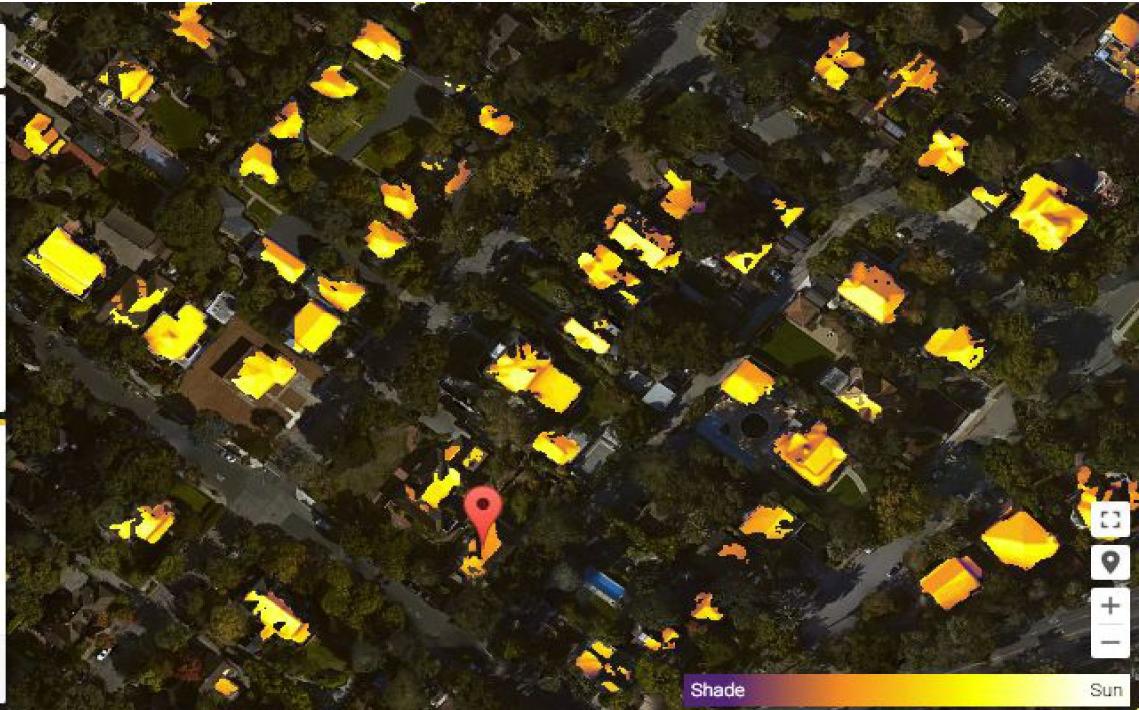
 1,658 hours of usable sunlight per year
Based on day-to-day analysis of weather patterns

 708 sq feet available for solar panels
Based on 3D modeling of your roof and nearby trees

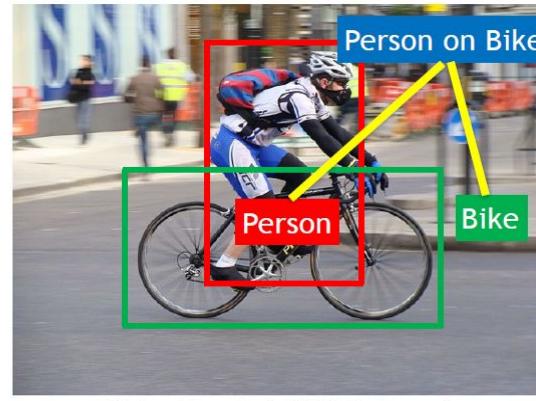
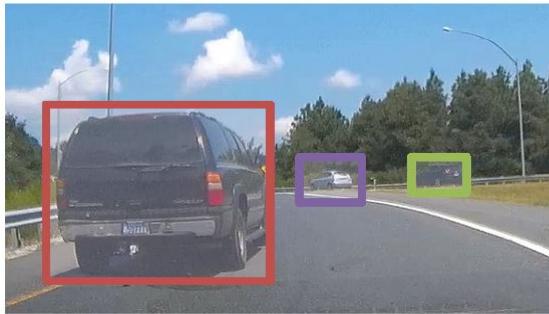
If your electric bill is at least \$175/month, leasing solar panels could reduce it.

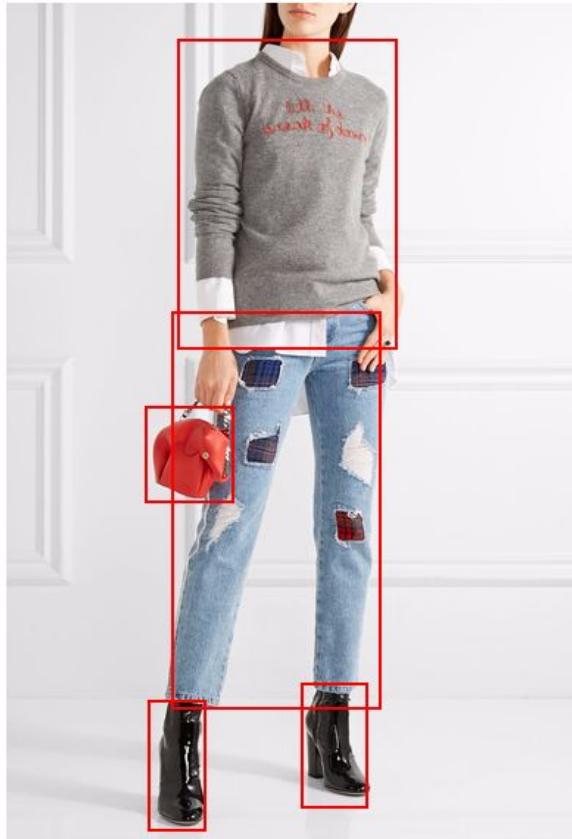
[FINE-TUNE ESTIMATE](#) [SEE SOLAR PROVIDERS](#)

Wrong roof? Drag the marker to the right one.



Object detection - Action classification - Image captioning



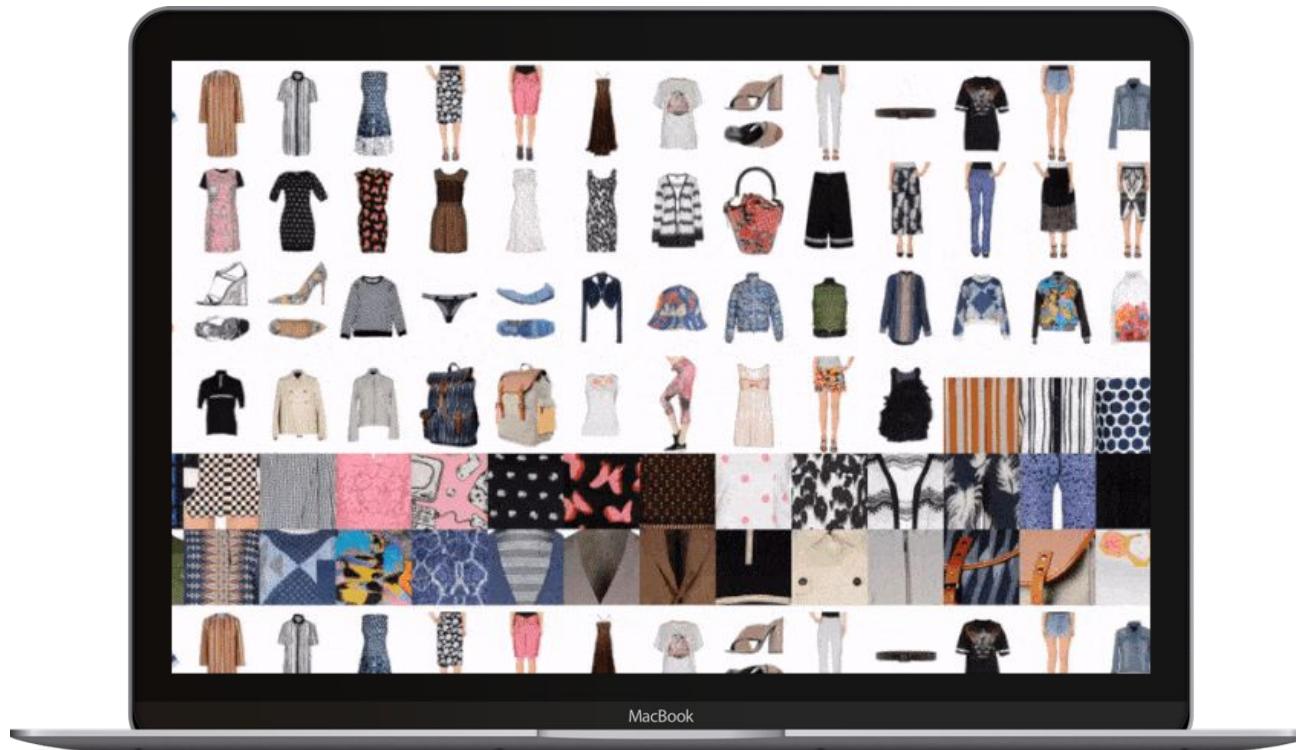


jeans
boots
boots
sweater
bag

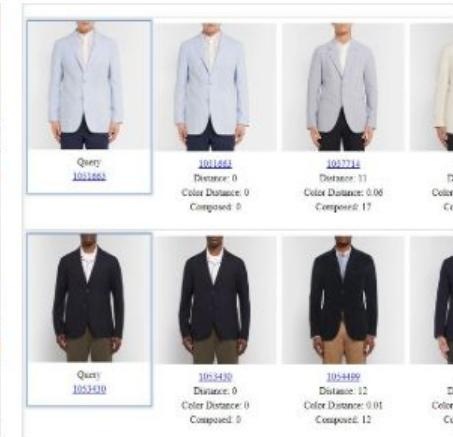
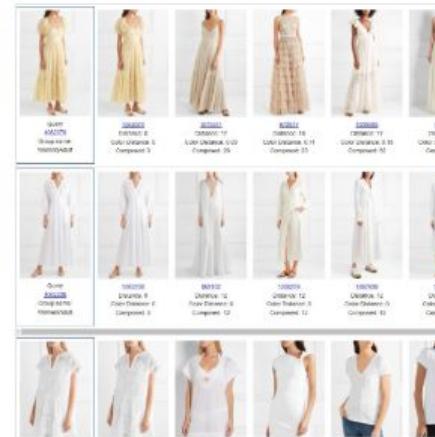


bag
heels
sunglasses
pants
heels
top

Image similarity API



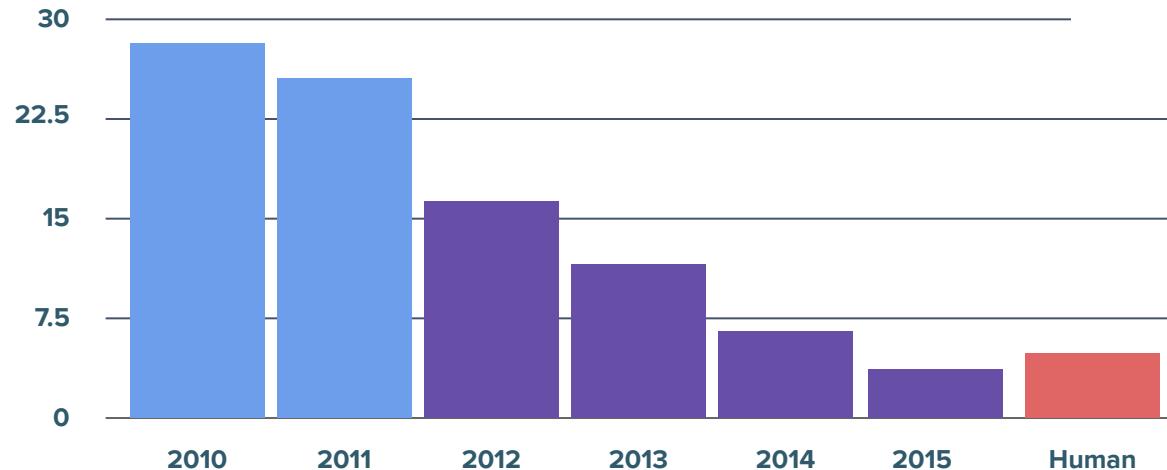
Yoda-similarity



Imagenet Challenge

Classification error %

1,000 object classes
1.5 million images



Better than human performance in image classification tasks

CHESS (1997)

World champion Kasparov vs Deep Blue (IBM)

Game complexity: 10^{50}

GO (2016)

World champion Lee Sedol vs AlphaGo (Google)

Game complexity: 10^{170}

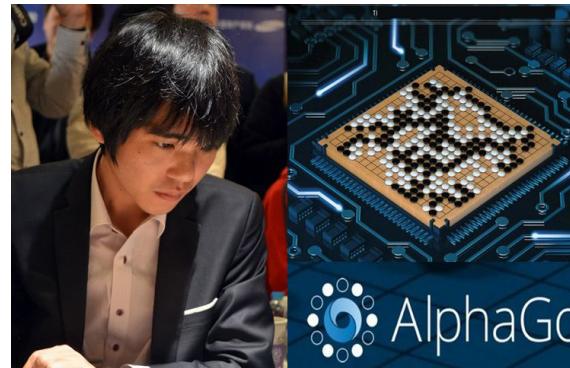
Where Deep Blue mainly relied on brute computational force to evaluate millions of positions, AlphaGo also relied on neural networks and reinforcement learning, which more closely resemble human decision-making

CHESS (2017)



AlphaZero vs StockFish
4-hour training time starting from scratch

GO (2017)



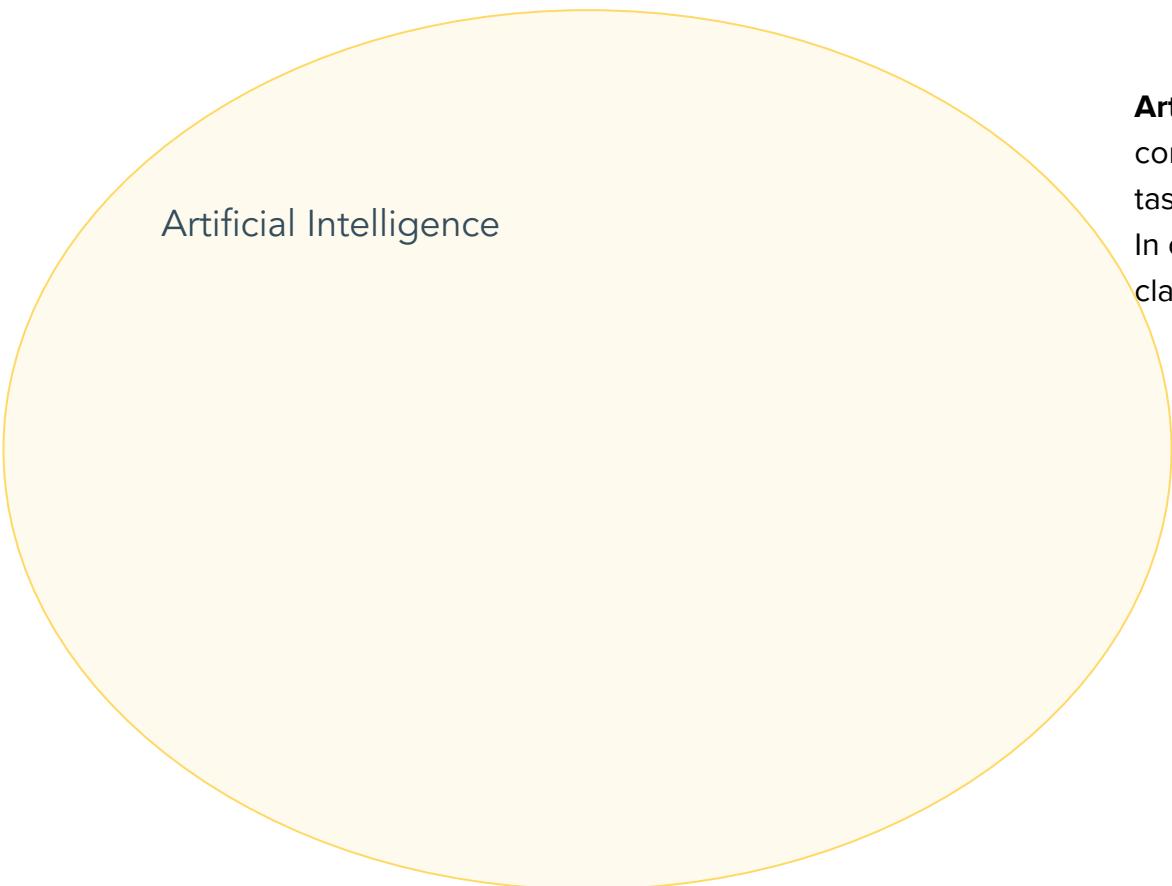
AlphaGo vs AlphaGo Zero (Google)
3-day training time starting from scratch

Based on reinforcement learning

AlphaZero searches fewer positions than its predecessor (80k per second vs 70M per second)
GM Magnus Carlsen: "I always wondered how it would be if a superior species landed on earth
and showed us how they play chess. I feel now I know"

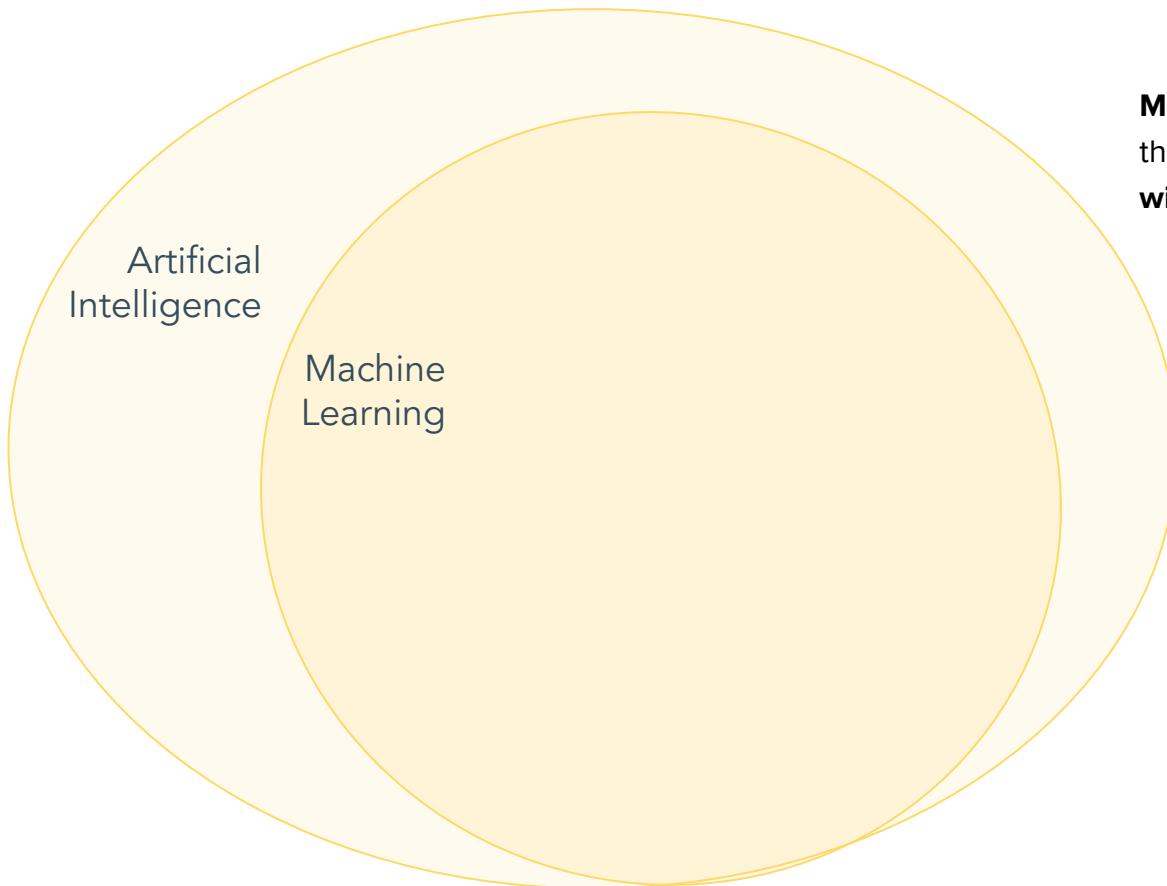


What is
Artificial
Intelligence?

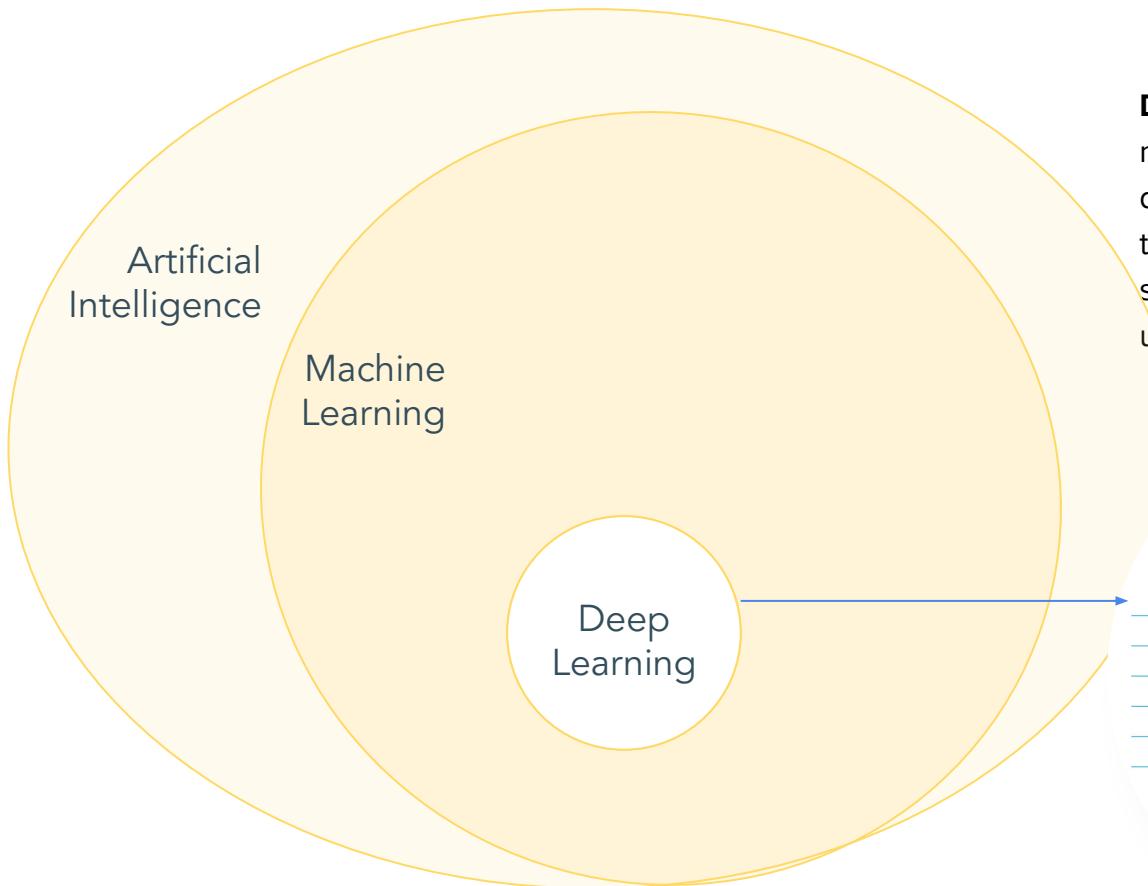


Artificial Intelligence

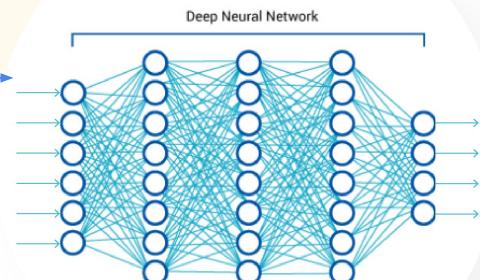
Artificial Intelligence (AI) is the broader concept of machines being able to carry out tasks in a way that we would consider “smart”. In other words, we are not able to write a classic computer program to perform the task.

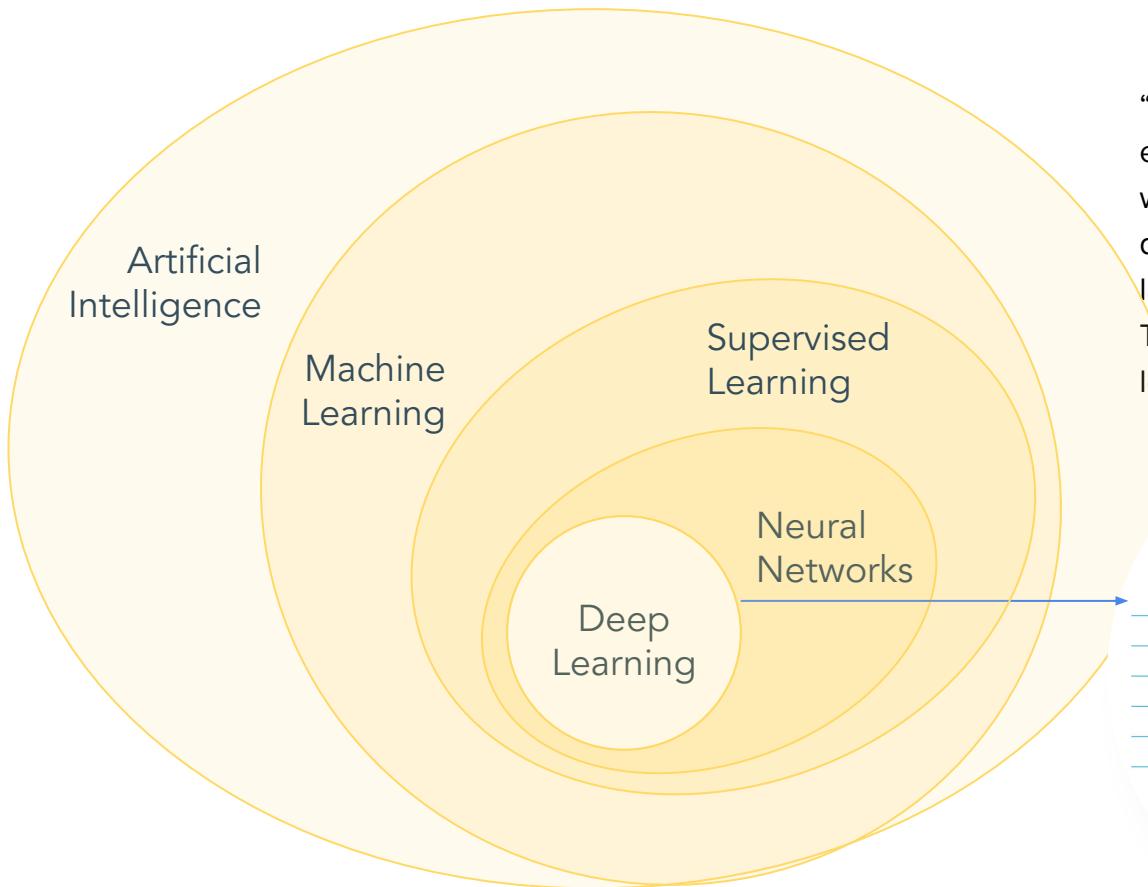


Machine learning is a broad set of techniques that enable computers to perform a task **without being explicitly programmed**.

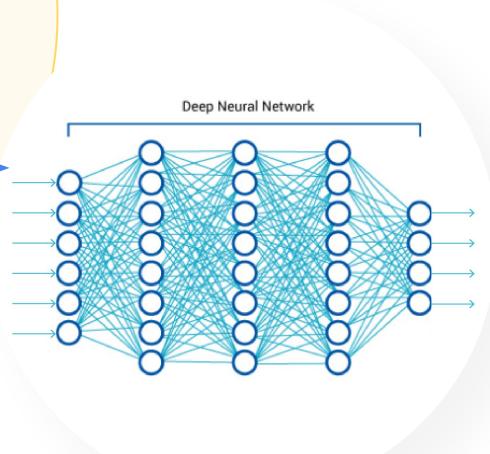


Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, partially supervised or unsupervised.





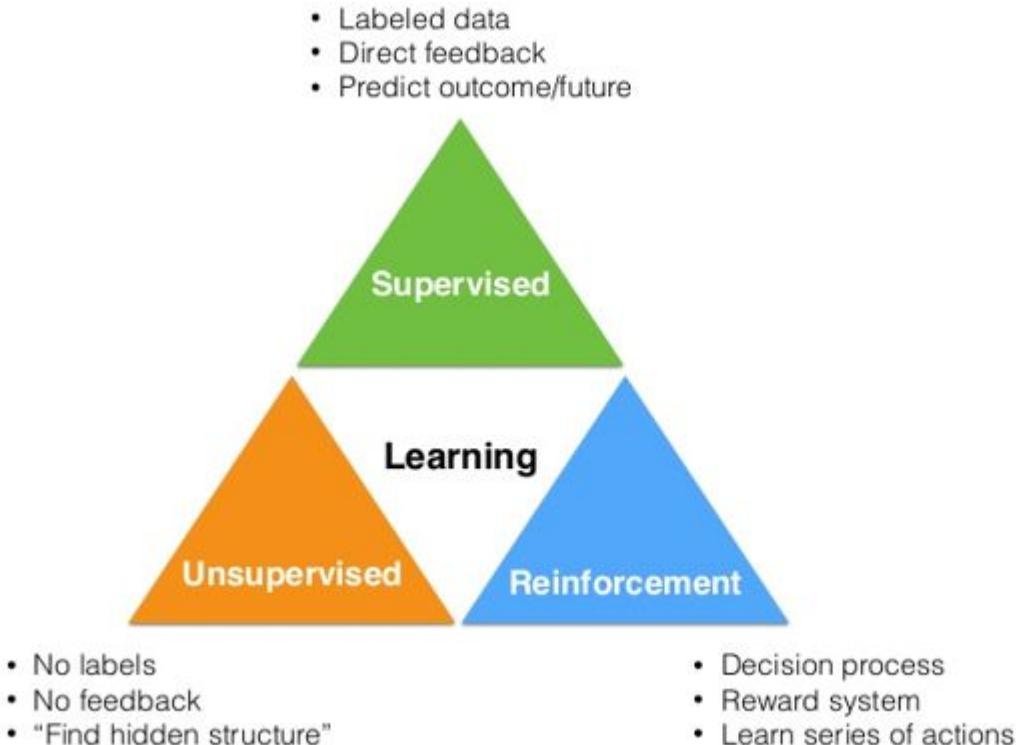
“Deep architectures are perhaps best exemplified by **multi-layer neural networks** with several hidden layers. In general terms, deep architectures are composed of multiple layers of parameterized non-linear modules. The parameters of every module are subject to learning.” [Yoshua Bengio and Yann LeCun]





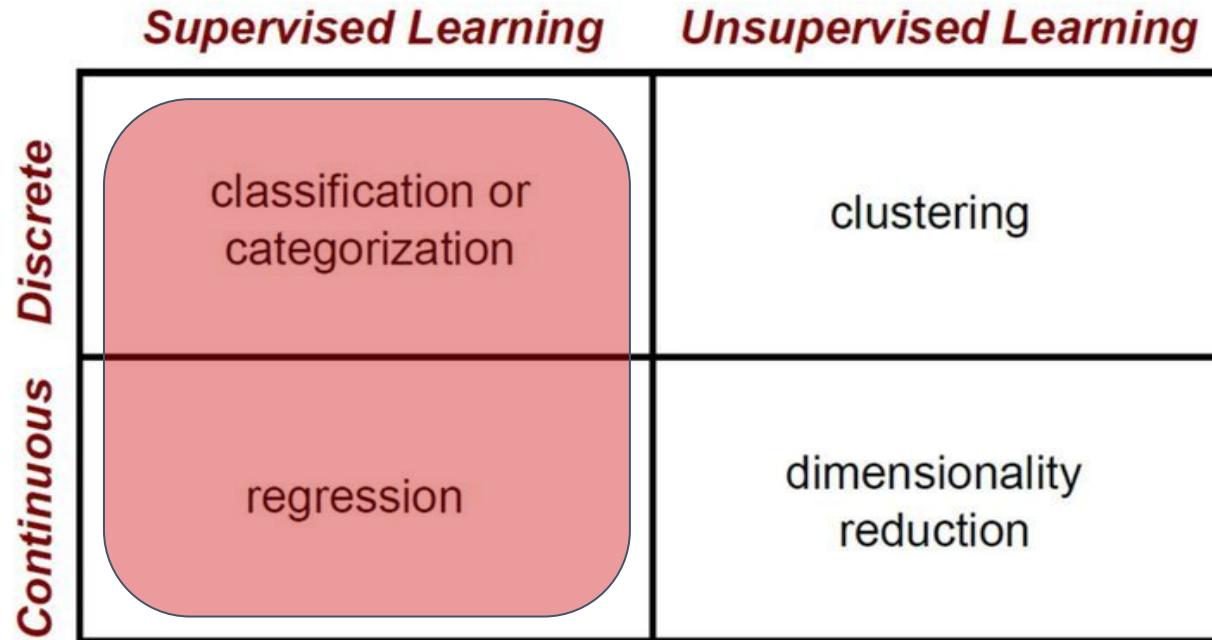
Basic concepts

Machine learning



	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>	classification or categorization	clustering
<i>Continuous</i>	regression	dimensionality reduction

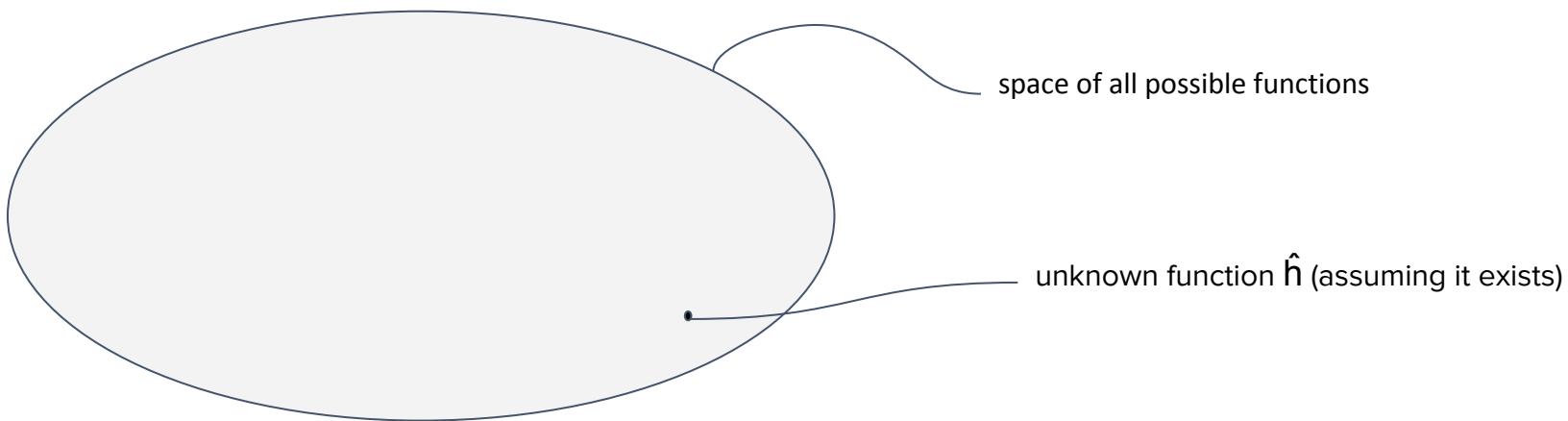
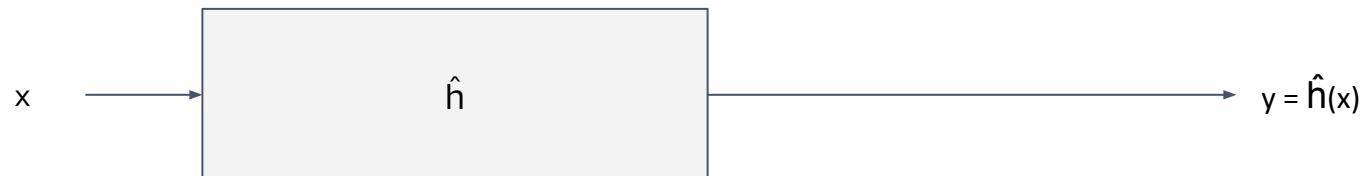
Training dataset with m samples: $(x^{(1)}, y^{(1)}), \dots (x^{(m)}, y^{(m)})$
Each $x^{(i)}, y^{(i)}$ can be multi-dimensional





Machine learning is a broad set of techniques that enable computers to perform a task **without being explicitly programmed**.

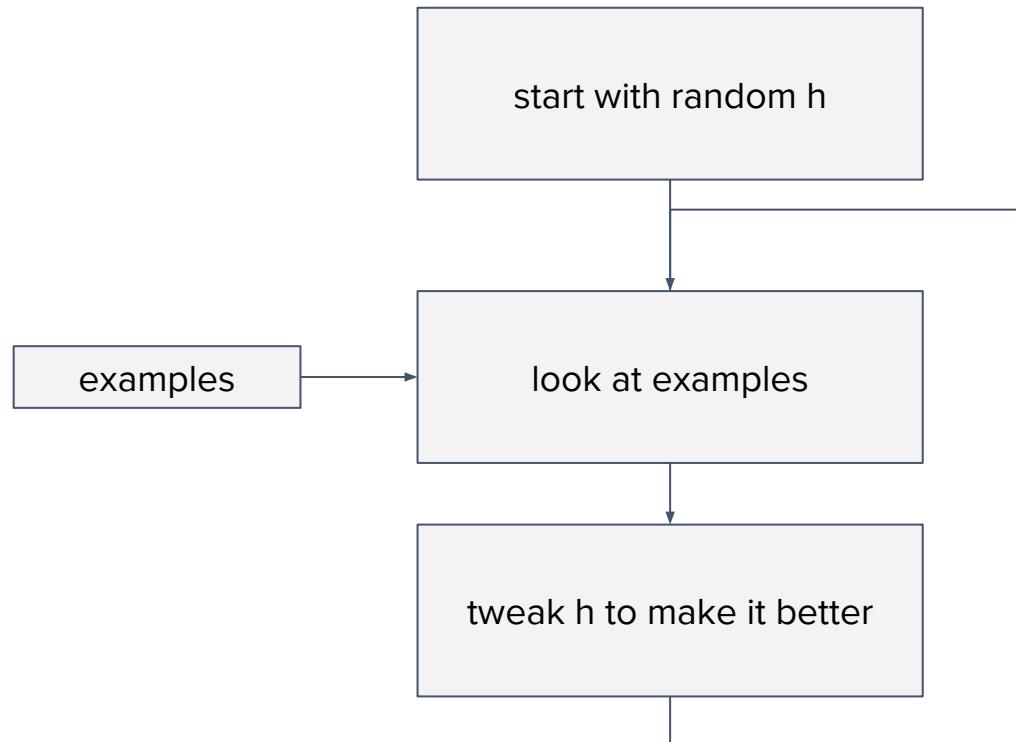
\hat{h} = unknown function that implements the algorithm





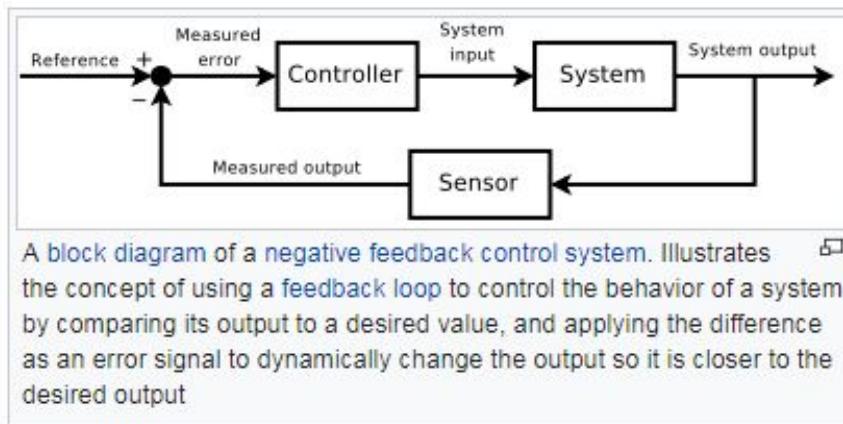
h = hypothesis function

How to find a good candidate for h ?



Control theory: harnessing the power of feedback loops.

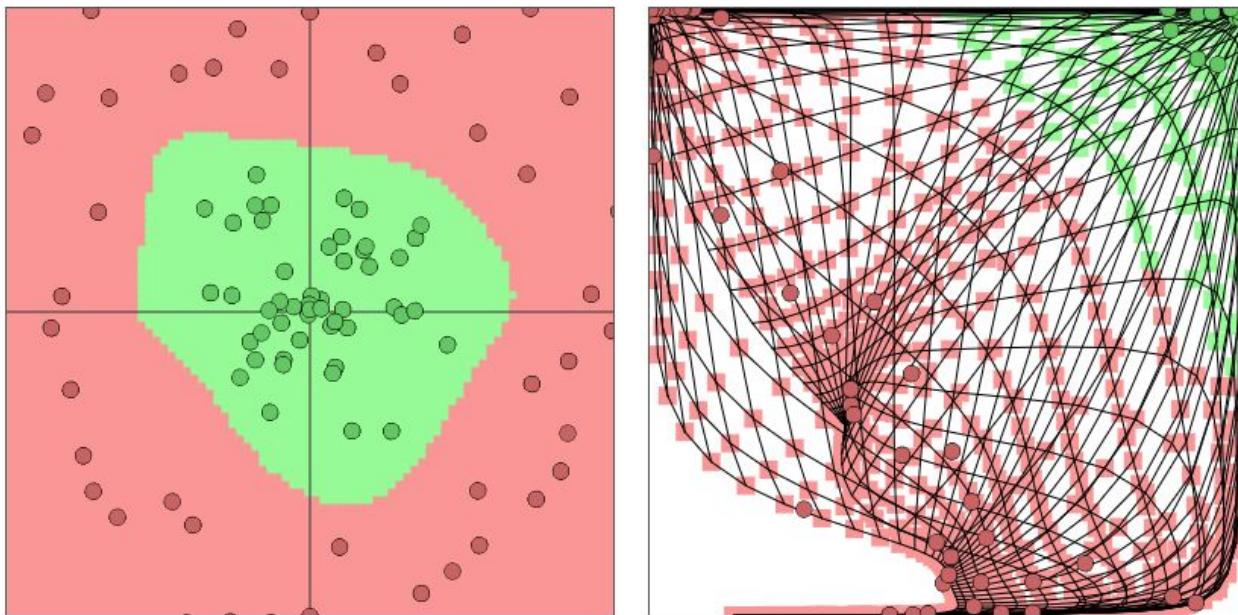
Wide applicability in many fields: physics, biology, climate science, mechanical engineering, electronic engineering, software engineering, social sciences, economics, etc.

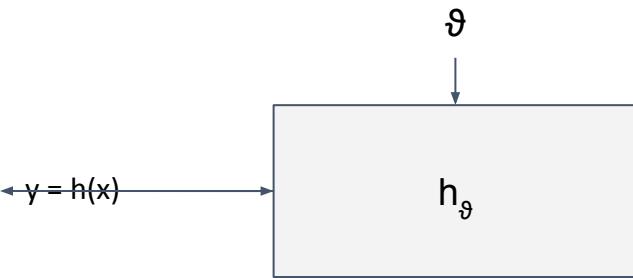


Let's see a learning process in action..

Demo: <http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html>

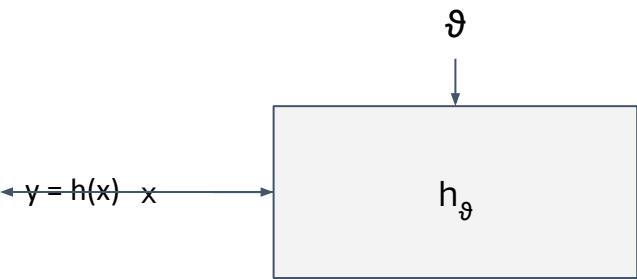
Note: bi-dimensional input $x^{(i)}$, binary output $y^{(i)} \in \{0, 1\}$





h = hypothesis function

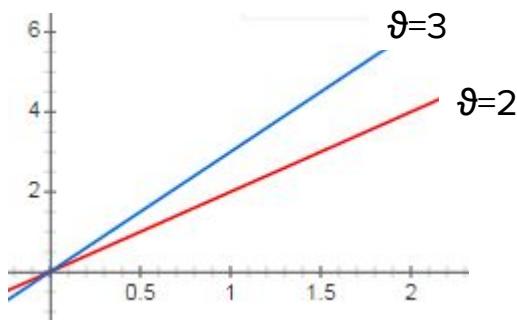
- instead of choosing a specific function for h , let's make it depend on one or more **parameters** θ , so that h_θ represents a **SET of functions**
- by changing the parameters, the candidate h changes.



h = hypothesis function

- instead of choosing a specific function for h , let's make it depend on one or more **parameters θ** , so that h_θ represents a **SET of functions**
- by changing the parameters, the candidate h changes.

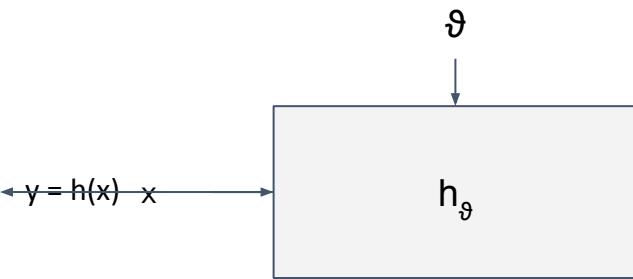
Example: $h(x) = 2x \longrightarrow h_\theta(x) = \theta x$



$$\theta=2 \longrightarrow h(x) = 2x$$

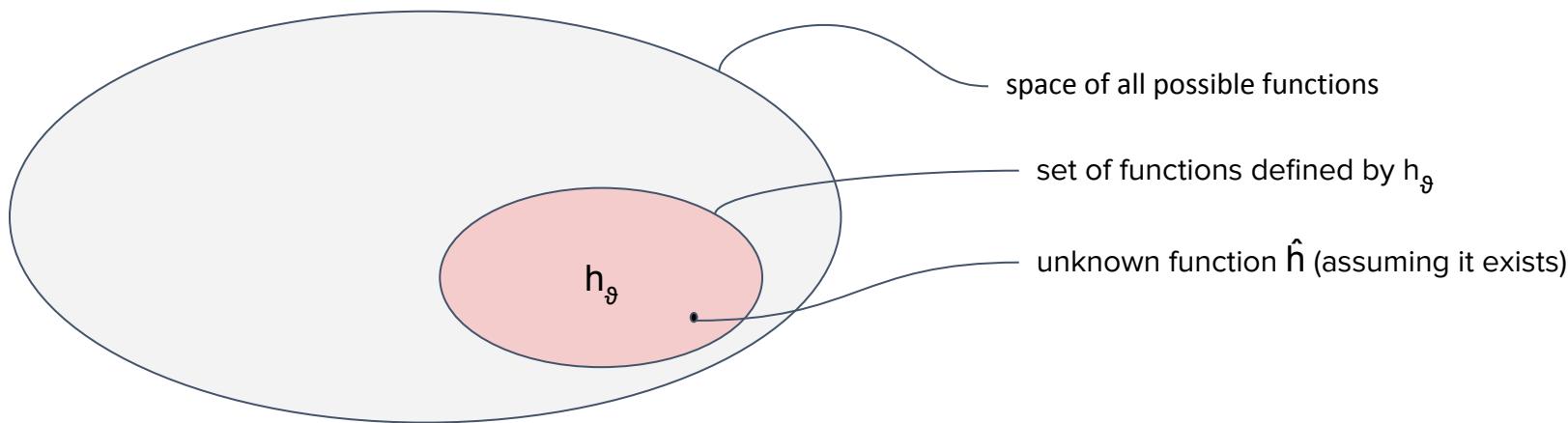
$$\theta=3 \longrightarrow h(x) = 3x$$

....



h = hypothesis function

- instead of choosing a specific function for h , let's make it depend on one or more **parameters** ϑ ,
- so that h_θ represents a **SET of functions**
- by changing the parameters, the candidate h changes.



THE WORKFLOW 1



Data Preprocessing

Select Algorithm &
Framework

Train & Tune Your Model

Integrate & Deploy

THE WORKFLOW 2



Select & Prepare Training Data

Choose & Optimize Your ML Model

Setup & Manage Environment For Training

Train & Tune Model (Trial & Error)

Deploy Your ML Model To Production

Scale & Manage Production Environment

How to choose the parameterized form of h ?

How to choose parameter values?

LINEAR REGRESSION

LET'S CONSIDER A FUNCTION y that can be calculated from a linear combination of the input variables (x)

When there is a single input variable (x), the method is referred to as simple linear regression.

When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression

KEY TERMINOLOGY

Labels

A **label** is the thing we're predicting the y variable in simple linear regression. The label could be the future price of wheat, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.

Features

A **feature** is an input variable the x variable in simple linear regression. A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features, specified as:

x_1, x_2, \dots, x_N

Examples/Instance

An **example** is a particular instance of data, \mathbf{x} . (We put \mathbf{x} in boldface to indicate that it is a vector.)

LINEAR REGRESSION

$$\mathcal{Y} = \Theta(\mathcal{X})$$

Or in extended format

$$y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_n X_n$$

Or in vectorized form:

$$\hat{y} = h_{\theta}(x) = \theta \cdot x$$

LINEAR REGRESSION: we have a closed form eq

$$MSE(X, h_{\theta}) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$



$MSE(\theta)$ to simplify

Normal equation :

$$\hat{\theta} = (X^T X)^{-1} X^T y \leftarrow$$

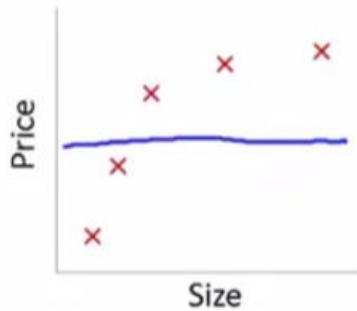
Solution To

find $\hat{\theta}$ that minimize the cost function

Linear regression with regularization

Model: $h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

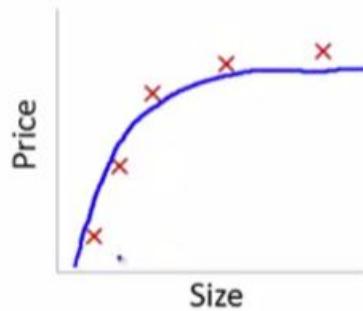
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^m \theta_j^2$$



Large λ

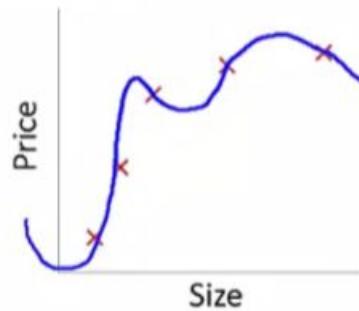
→ High bias (underfit)

→ $\lambda = 10000$. $\theta_1 \approx 0, \theta_2 \approx 0, \dots$
 $h_{\theta}(x) \approx \theta_0$



Intermediate λ

"Just right"



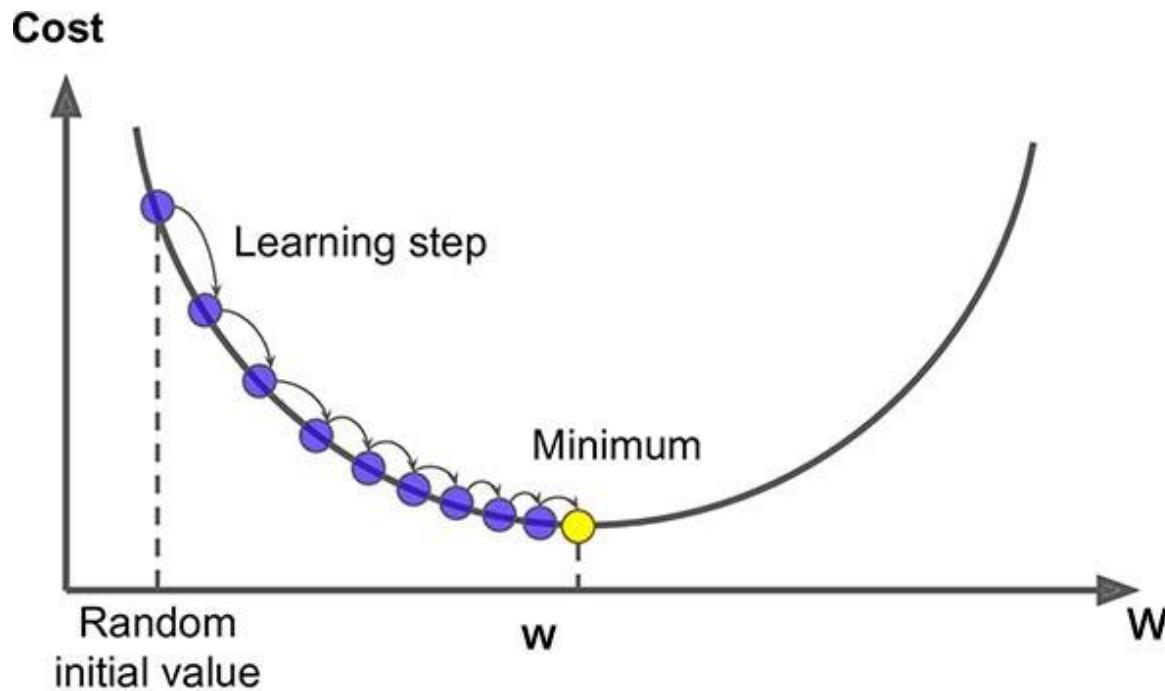
Small λ

High variance (overfit)

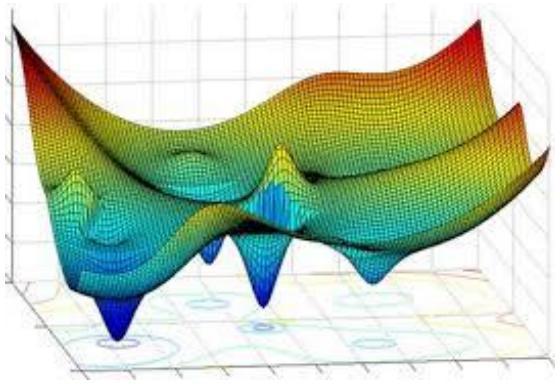
→ $\lambda = 0$

LET'S MOVE TO JUPYTER FOR THE DEMO

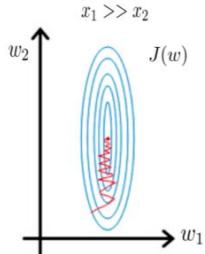
GRADIENT DESCENT



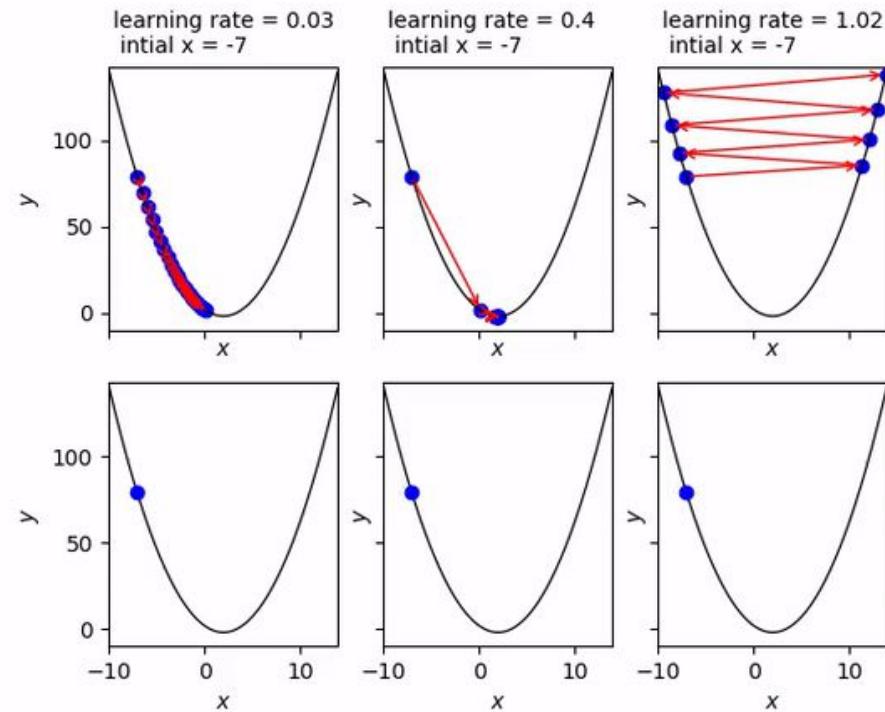
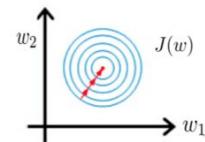
GRADIENT DESCENT fine tuning



Gradient descent
without scaling



Gradient descent
after scaling variables



GRADIENT DESCENT IMPROVED

SGD:

Note that in each iteration (also called update), only the gradient evaluated at a single point

$X(i)$

instead of evaluating at the set of all samples.

The key difference compared to standard (Batch) Gradient Descent is that only one piece of data from the dataset is used to calculate the step, and the piece of data is picked randomly at each step.

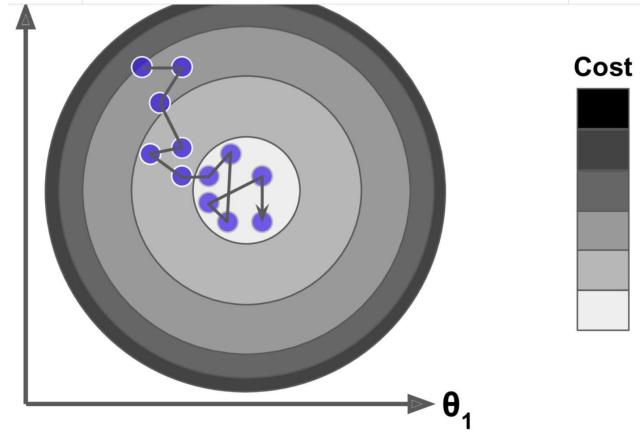


Figure 4-9. Stochastic Gradient Descent

知乎 @孙铭泽

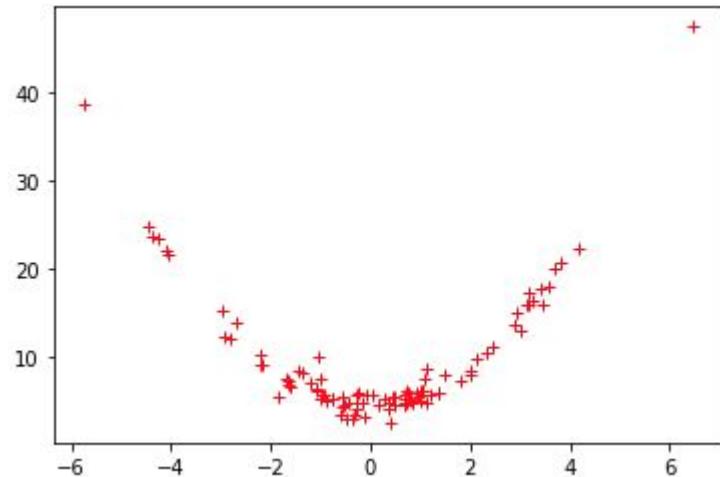
- Choose an initial vector of parameters w and learning rate η .
- Repeat until an approximate minimum is obtained:
 - Randomly shuffle examples in the training set.
 - For $i = 1, 2, \dots, n$, do:
 - $w := w - \eta \nabla Q_i(w)$.

POLYNOMIAL REGRESSION

What if my data are clearly not linear?

There's a linear model that can fit non-linear Data.

Let's look at the example in JPYNB



$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \varepsilon.$$

MAIN ISSUES OF ALL THE ALGORITHMS

- BIAS
- OVERFITTING
- BIAS / VARIANCE TRADE OFF

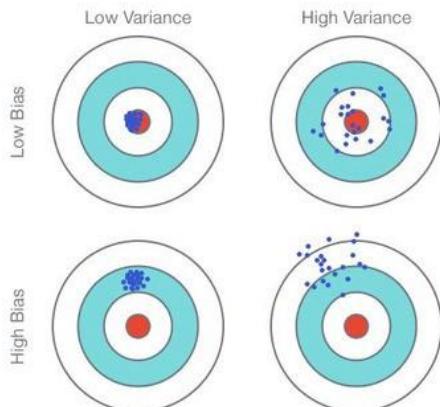
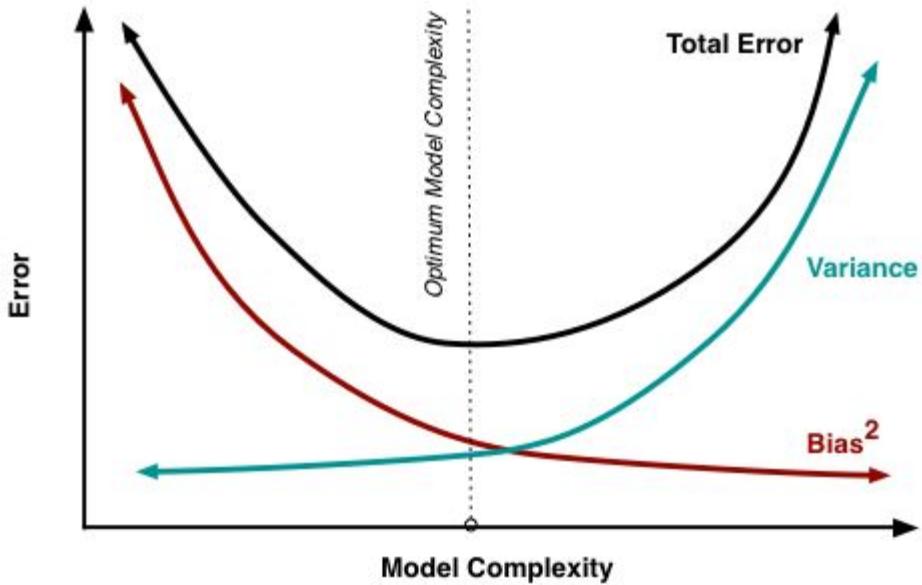


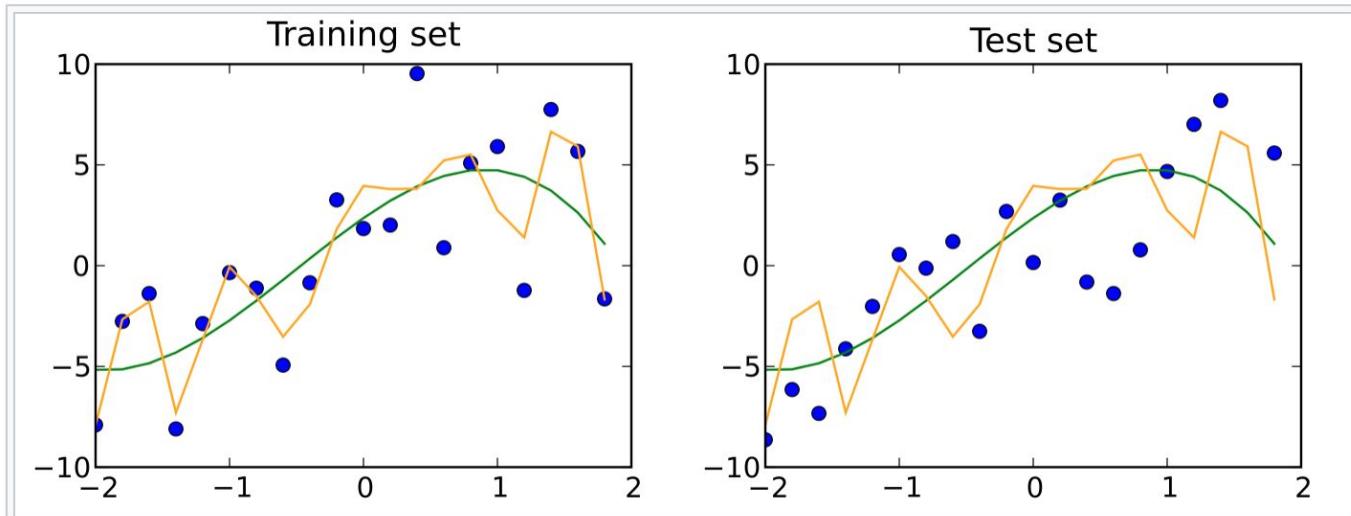
Fig. 1: Graphical Illustration of bias-variance trade-off , Source: Scott Fortmann-Roe., Understanding Bias-Variance Trade-off



Test dataset [edit]

A test dataset is a [dataset](#) that is [independent](#) of the training dataset, but that follows the same [probability distribution](#) as the training dataset. If a model fit to the training dataset also fits the test dataset well, minimal [overfitting](#) has taken place (see figure below). A better fitting of the training dataset as opposed to the test dataset usually points to overfitting.

A test set is therefore a set of examples used only to assess the performance (i.e. generalization) of a fully specified classifier.^{[7][8]}



A training set (left) and a test set (right) from the same statistical population are shown as blue points. Two predictive models are fit to the training data. Both fitted models are plotted with both the training and test sets. In the training set, the MSE of the fit shown in orange is 4 whereas the MSE for the fit shown in green is 9. In the test set, the MSE for the fit shown in orange is 15 and the MSE for the fit shown in green is 13. The orange curve severely overfits the training data, since its MSE increases by almost a factor of four when comparing the test set to the training set. The green curve overfits the training data much less, as its MSE increases by less than a factor of 2.

REGULARIZED LINEAR MODELS

A simple way to look at it is to think of introducing a parameter that measures complexity and a function that will increase our cost function every time we increase the complexity of the model itself.

That means finding a solution that should minimise the cost function (as usual) BUT BOOSTING the simpler solutions.

Ridge and Lasso are regularized linear models, that follow this approach, introducing hyperparameters to keep control of complexity.

LOGISTIC REGRESSION

Is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis.

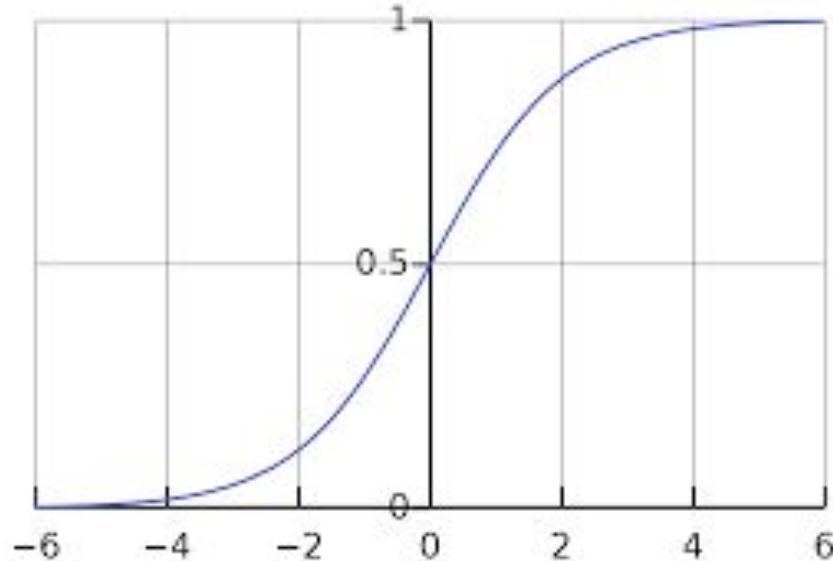
Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

$$y = \frac{1}{1 + e^{-z}}$$

$$z = \alpha_0 + \sum_{i=1}^N \alpha_i x_i$$

LOGISTIC REGRESSION

- It is a statistical approach
- It uses a sigmoid function
- It can take categorical variables
- Output will be between 0 and 1
- Used to solve classification problems



$$y = \frac{1}{1 + e^{-z}}$$

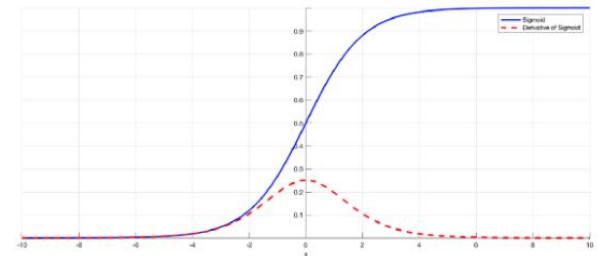
$$z = \alpha_0 + \sum_{i=1}^N \alpha_i x_i$$

LOGISTIC REGRESSION MATH

- No closed form solution
- But it's a convex function with one minimum
- But we can use Gradient Descent !

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

Derivative of the Sigmoid Function

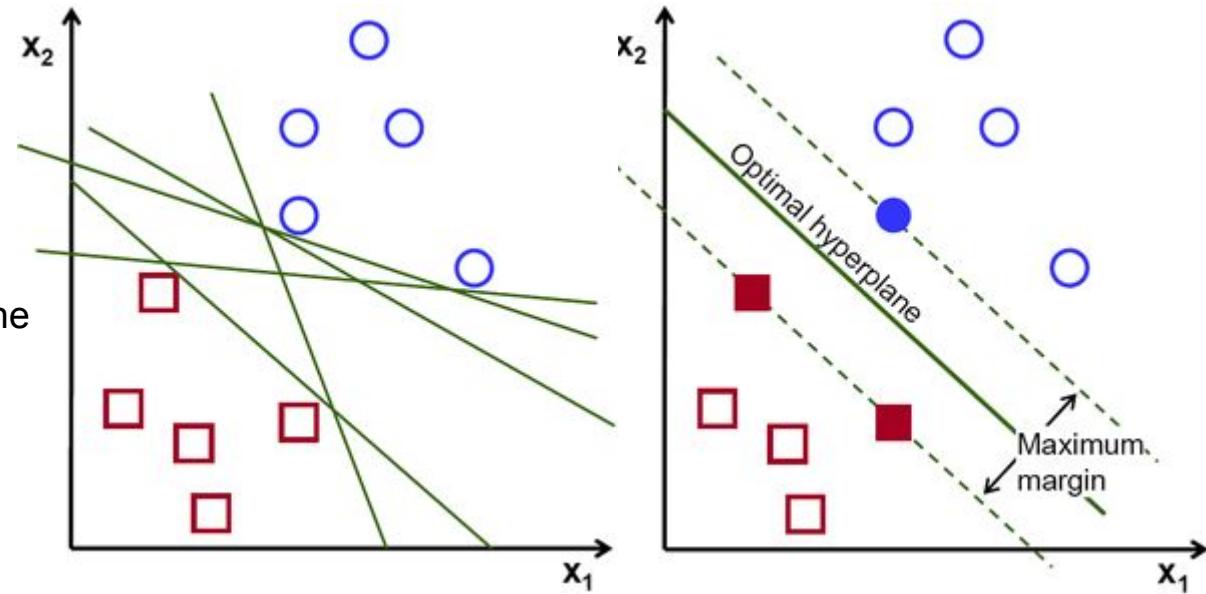


SUPPORT VECTOR MACHINES

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen.

Our objective is to find a plane that has the **maximum margin**, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.



HYPER PARAMETERS OPTIMIZATION WITH GridSearch

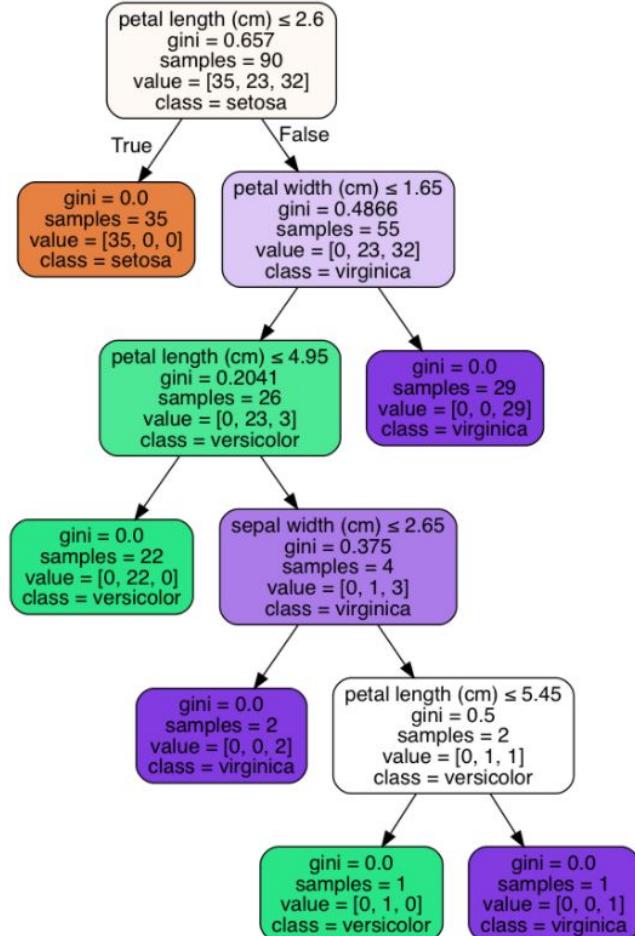
Grid search is essentially an optimization algorithm which lets you select the best parameters for your optimization problem from a list of parameter options that you provide, hence automating the 'trial-and-error' method. Although it can be applied to many optimization problems, it is most popularly known for its use in machine learning to obtain the parameters at which the model gives the best accuracy.

DECISION TREES

Versatile ML algorithms for both classification and regression.

Capable of fitting complex datasets. They are easy to understand, look at the graph on the right.

The objective is to subdivide the set with rules and try to move each class in a specific node and keep the classes separated as much as possible.



DECISION TREES

Computationally: prediction is very fast, since it's based on simple comparisons. Generally quick also to train, unless we're talking about huge datasets.

The algorithm used to train the Decision Trees in Scikit-learn is called **CART**.

Classification and Regression Tree

Impurity Criterion

Gini Index

$$I_G = 1 - \sum_{j=1}^c p_j^2$$

p_j : proportion of the samples that belongs to class c for a particular node

Entropy

$$I_H = - \sum_{j=1}^c p_j \log_2(p_j)$$

p_j : proportion of the samples that belongs to class c for a particular node.

*This is the definition of entropy for all non-empty classes ($p \neq 0$). The entropy is 0 if all samples at a node belong to the same class.

DECISION TREES

Computationally: prediction is very fast, since it's based on simple comparisons. Generally quick also to train, unless we're talking about huge datasets.

The algorithm used to train the Decision Trees in Scikit-learn is called **CART**.

Classification and Regression Tree

Main hyperparameters: max_depth, min_samples_split, min_samples_leaf...etc.

Impurity Criterion

Gini Index

$$I_G = 1 - \sum_{j=1}^c p_j^2$$

p_j : proportion of the samples that belongs to class c for a particular node

Entropy

$$I_H = - \sum_{j=1}^c p_j \log_2(p_j)$$

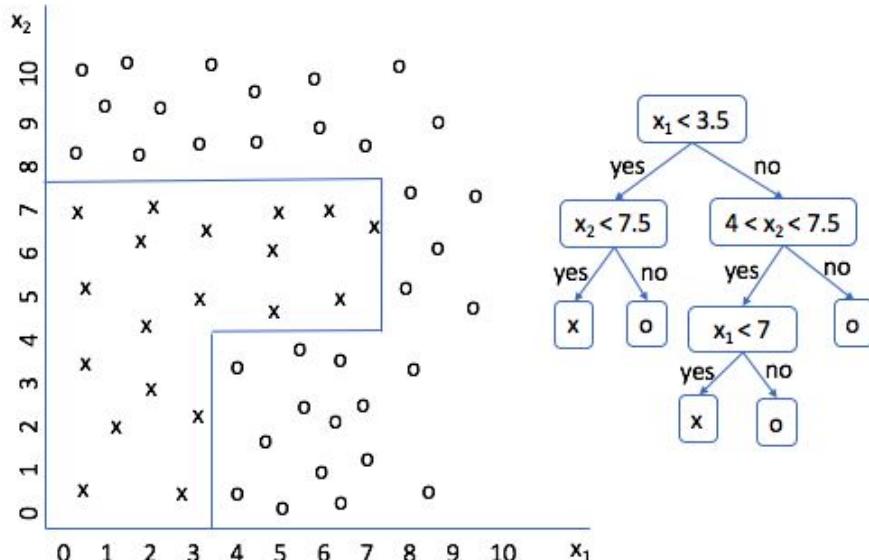
p_j : proportion of the samples that belongs to class c for a particular node.

*This is the definition of entropy for all non-empty classes ($p \neq 0$). The entropy is 0 if all samples at a node belong to the same class.

DECISION TREES

Interesting facts about decision trees:

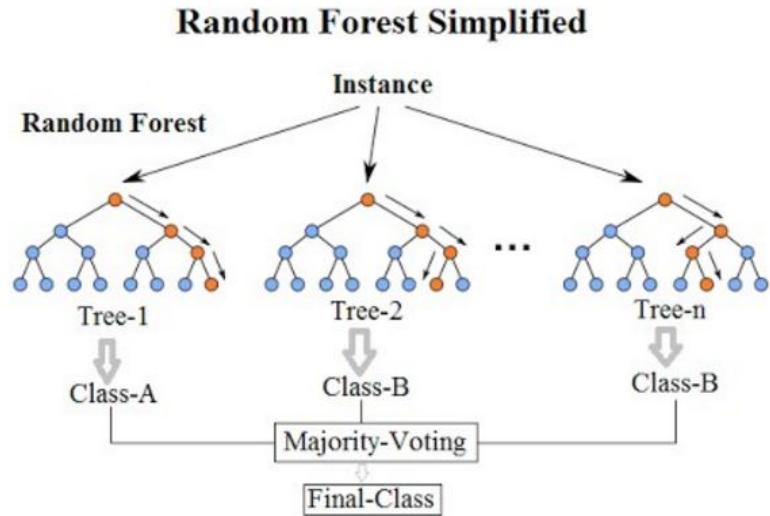
- non parametric models
- white box Machine Learning
- low tuning of hyperparameters
- no need to normalize data
- tend to overfit
- high variance \rightarrow unstable



RANDOM FOREST

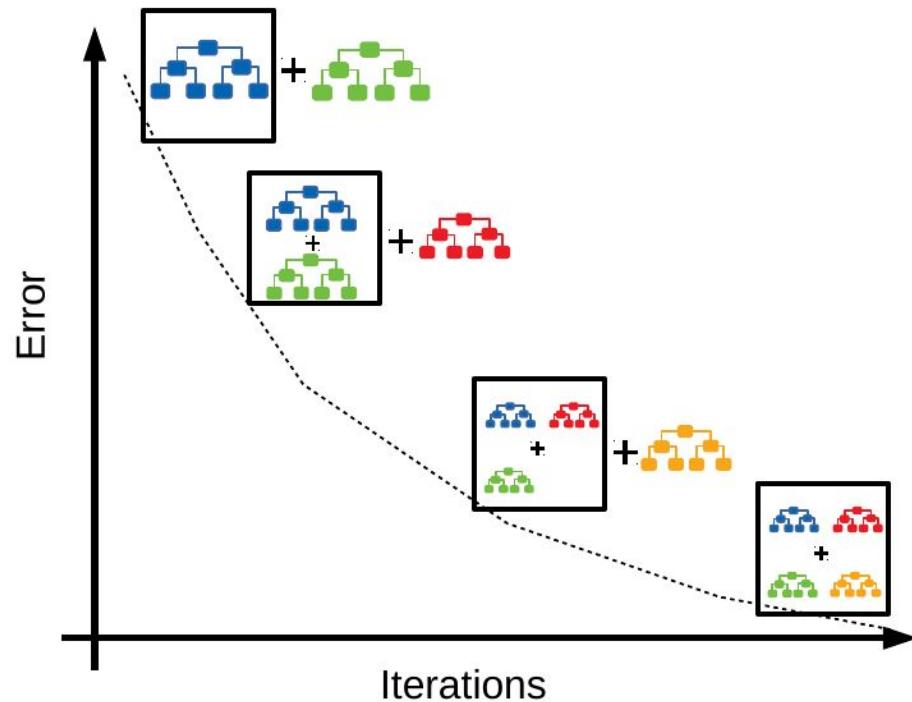
Random Forest is the first example of
ENSEMBLE LEARNING

- Train many decision trees, each one on a random sample data
- Combine their output:
 - Regression: **AVERAGE**
 - Classification: **MODE**



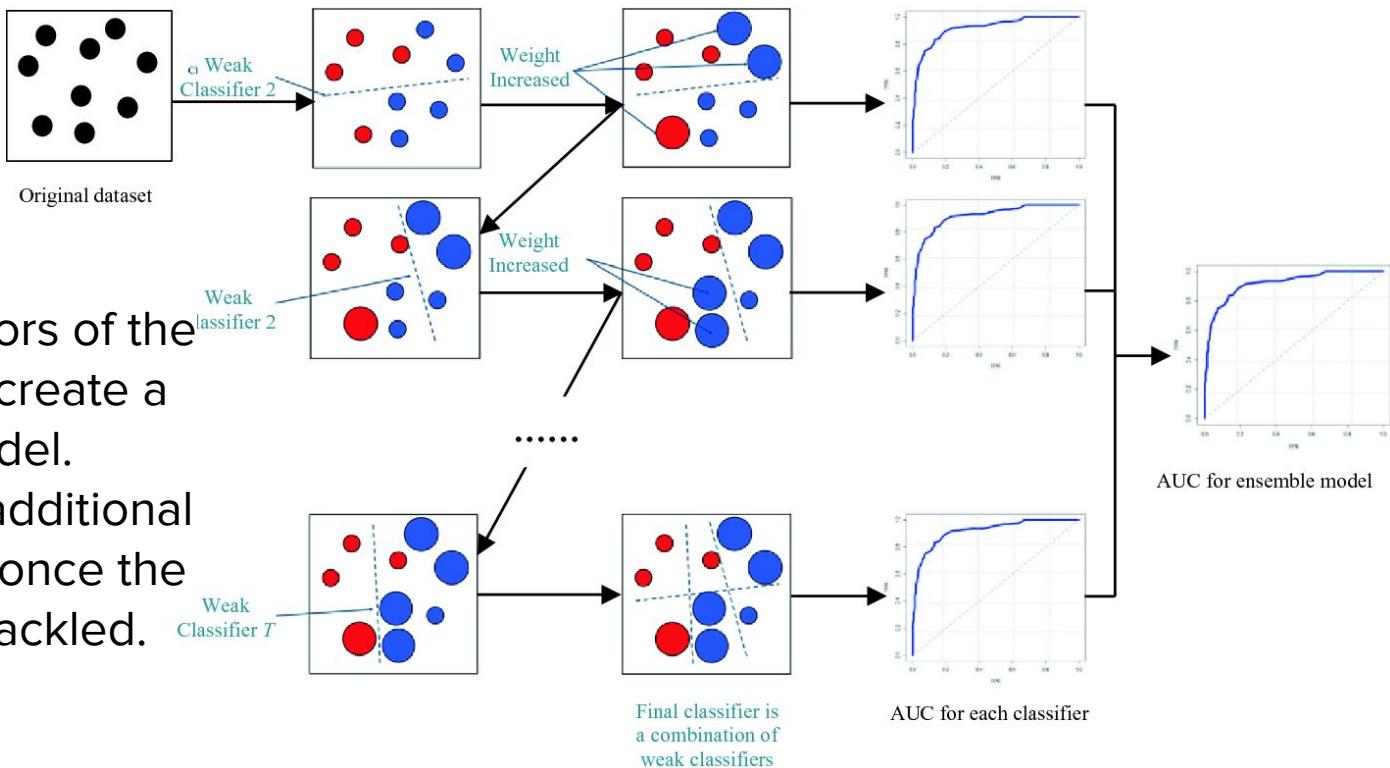
GRADIENT BOOSTING

Is a form of ENSEMBLE LEARNING and uses multiple weak learners to create a strong learner



GRADIENT BOOSTING

Working on the errors of the previous model to create a better decision model.
The weight of the additional trees are modified once the errors are getting tackled.



UNSUPERVISED MODELS: CLUSTERING

Unsupervised Learning:

No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

WHY UNSUPERVISED LEARNING

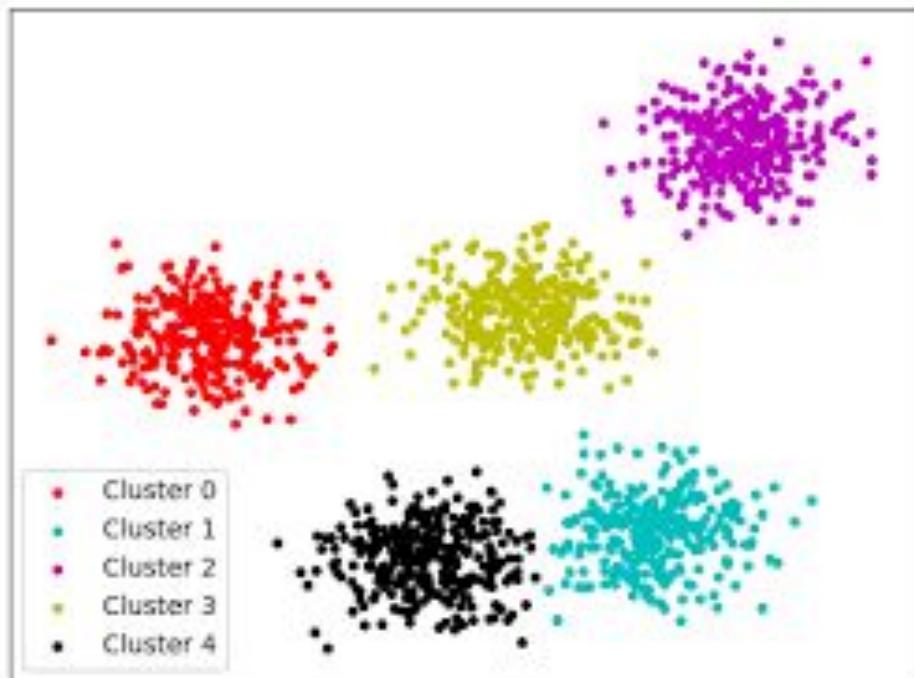
Why Unsupervised Learning is needed despite of these issues?

- Annotating large datasets is very costly and hence we can label only a few examples manually.
Example: Speech Recognition
- There may be cases where we don't know how many/what classes the data is divided into.
Example: Data Mining
- We may want to use clustering to gain some insight into the structure of the data before designing a classifier.

CLUSTERING

Uses:

- customer segmentation
- data analysis
- dimensionality reduction technique
- for anomaly detection
- for search engines
- to segment an image
- for semi-supervised learning



K MEANS ALGORITHMS

