MACHINE LEARNING

Learning Opencv Orelly

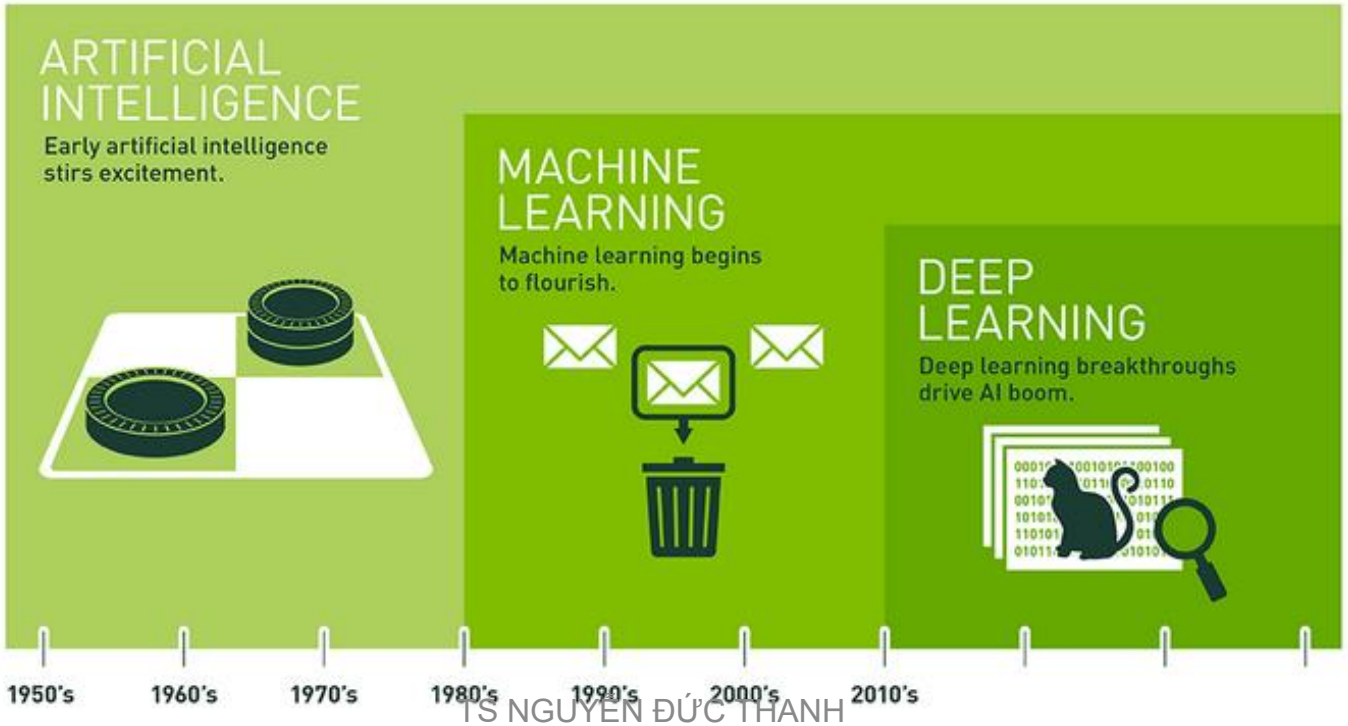
https://machinelearningcoban.com/

Machine Learning with Matlab

Deep Learning

KHÁI NIỆM MACHINE LEARNING

- Machine learning (máy học, ML) là tập hợp các thuật toán dùng để máy tính có khả năng nhận thức, từ data chuyển sang thông tin.
- ML là một phần của trí tuệ nhân tạo Artificial Intelligence AI.
- AI nghiên cứu tăng trí thông minh của máy tính như máy dịch ngôn ngữ, nhận dạng hình ảnh, tư duy…
- Những năm gần đây ML phát triển mạnh nhờ Deep Learning

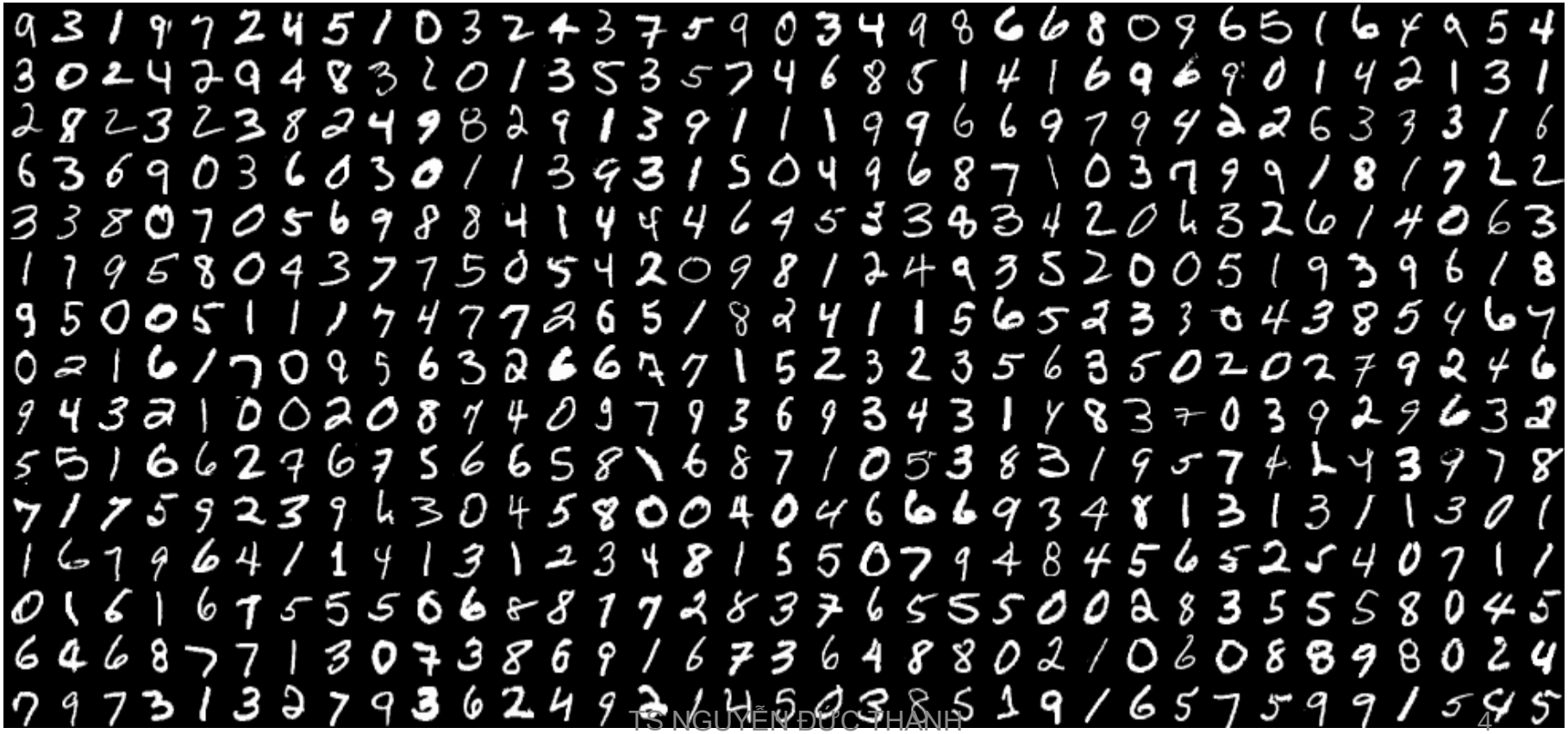


PHÂN LOẠI ML

- Theo phương thức học, các thuật toán Machine Learning thường được chia làm 4 nhóm: Supervised learning, Học có giám sát SL, Unsupervised learning Học không có giám sát UL, Semi-supervised learning Học có giám sát một phần SSL và Reinforcement learning Học củng cố RL.
- **Supervised learning** là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (*input, outcome*) đã biết từ trước. Cặp dữ liệu này còn được gọi là (*data, label*), tức (*dữ liệu, nhãn*). Nghĩa là khi huấn luyện ta cung cấp cho máy các dữ liệu cần phân loại và cho biết luôn kết quả, sau đó trong bước thực thi máy phải phân loại du7a5a trên kiến thức đã học.
- **Ví dụ** trong nhận dạng chữ viết tay, ta có ảnh của hàng nghìn chữ số được viết bởi nhiều người khác nhau. Chúng ta đưa các bức ảnh này vào trong một thuật toán và chỉ cho nó biết mỗi bức ảnh tương ứng với chữ số nào. Sau khi thuật toán tạo ra một mô hình, tức một hàm số mà đầu vào là một bức ảnh và đầu ra là một chữ số, khi nhận được một bức ảnh mới mà mô hình **chưa nhìn thấy bao giờ**, nó sẽ dự đoán bức ảnh đó chứa chữ số nào.

PHÂN LOẠI ML

- Ví dụ này khá giống với cách học của con người khi còn nhỏ. Ta đưa bảng chữ cái cho một đứa trẻ và chỉ cho chúng đây là chữ A, a đây là chữ B, b. Sau một vài lần được dạy thì trẻ có thể nhận biết được đâu là chữ A, a đâu là chữ B, b trong một cuốn sách mà chúng chưa nhìn thấy bao giờ.



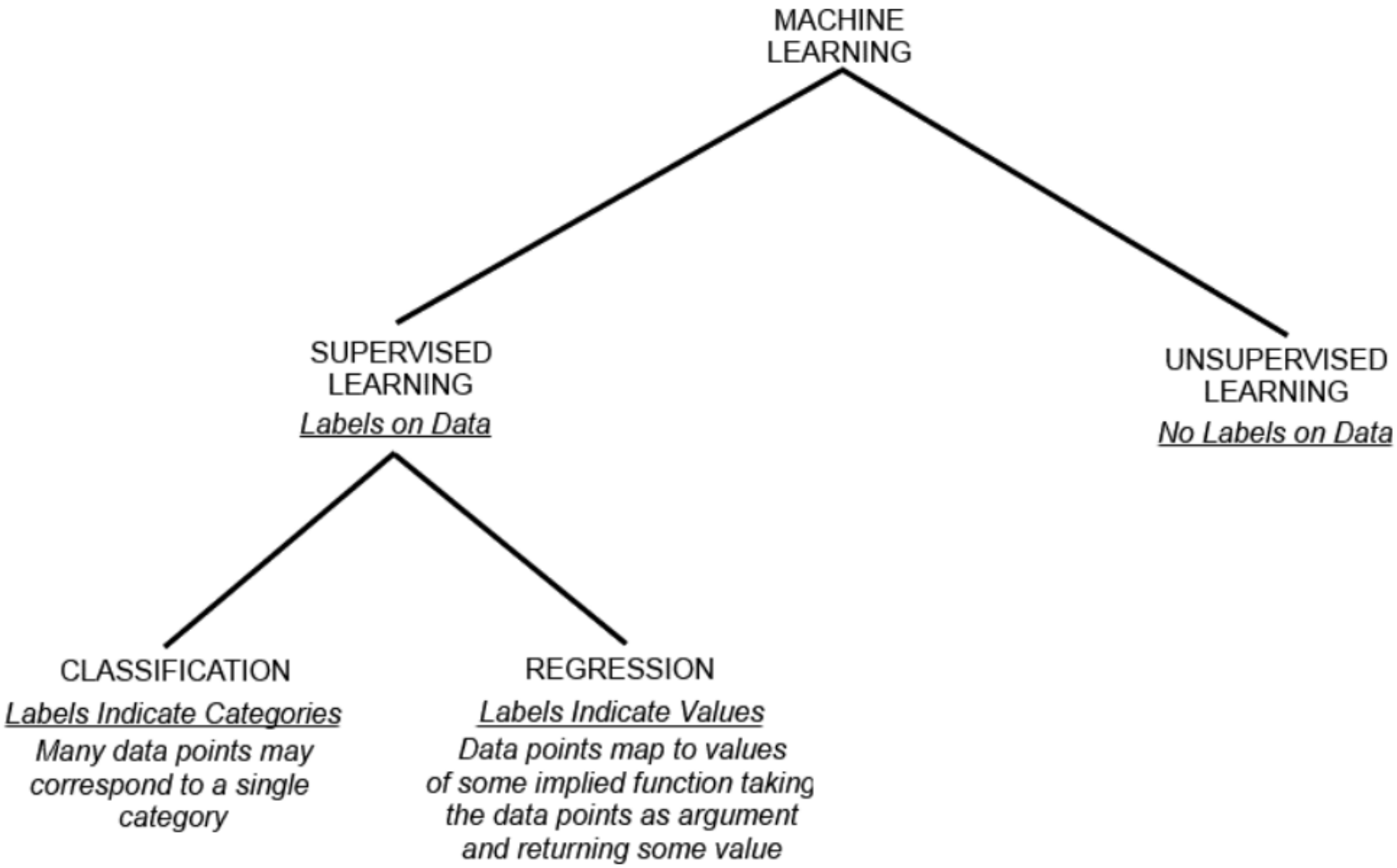
PHÂN LOẠI ML

- Ví dụ : Thuật toán dò tìm các khuôn mặt trong 1 bức ảnh dữ liệu học là hàng ngàn cặp (ảnh, mặt người) và (ảnh, không phải mặt người) được đưa vào.
- Ví dụ : Tiến lên một bước ta có thuật toán dò tìm các khuôn mặt thân quen trong 1 bức ảnh dữ liệu học là hàng ngàn cặp (ảnh, mặt người quen) và (ảnh, không phải mặt người quen) được đưa vào. Sau quá trình huấn luyện máy có thể tìm được những khuôn mặt thân quen trong những ảnh khác không nằm trong tập huấn luyện.
- Thuật toán SL còn được chia làm hai loại là Classification và Regression
- Bài toán được gọi là phân lớp nếu dữ liệu có thể gán nhãn là một cái tên, ví dụ nhận dạng chữ số, nhận dạng khuôn mặt người nào đó, nhận dạng đồ vật.
- Bài toán là regression khi ta gán cho nhãn một đại lượng số nào đó, ví dụ dự báo rating của một quảng cáo, lợi nhuận một sản phẩm, ước đoán tuổi một người nào đó

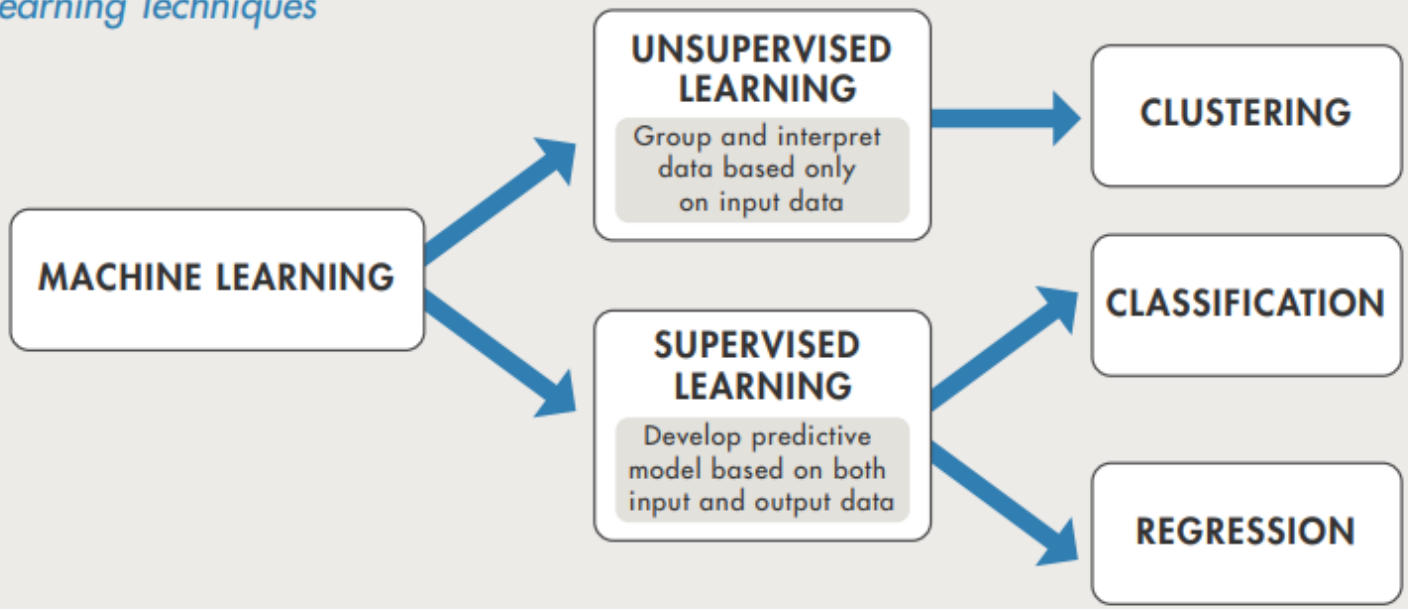
PHÂN LOẠI ML

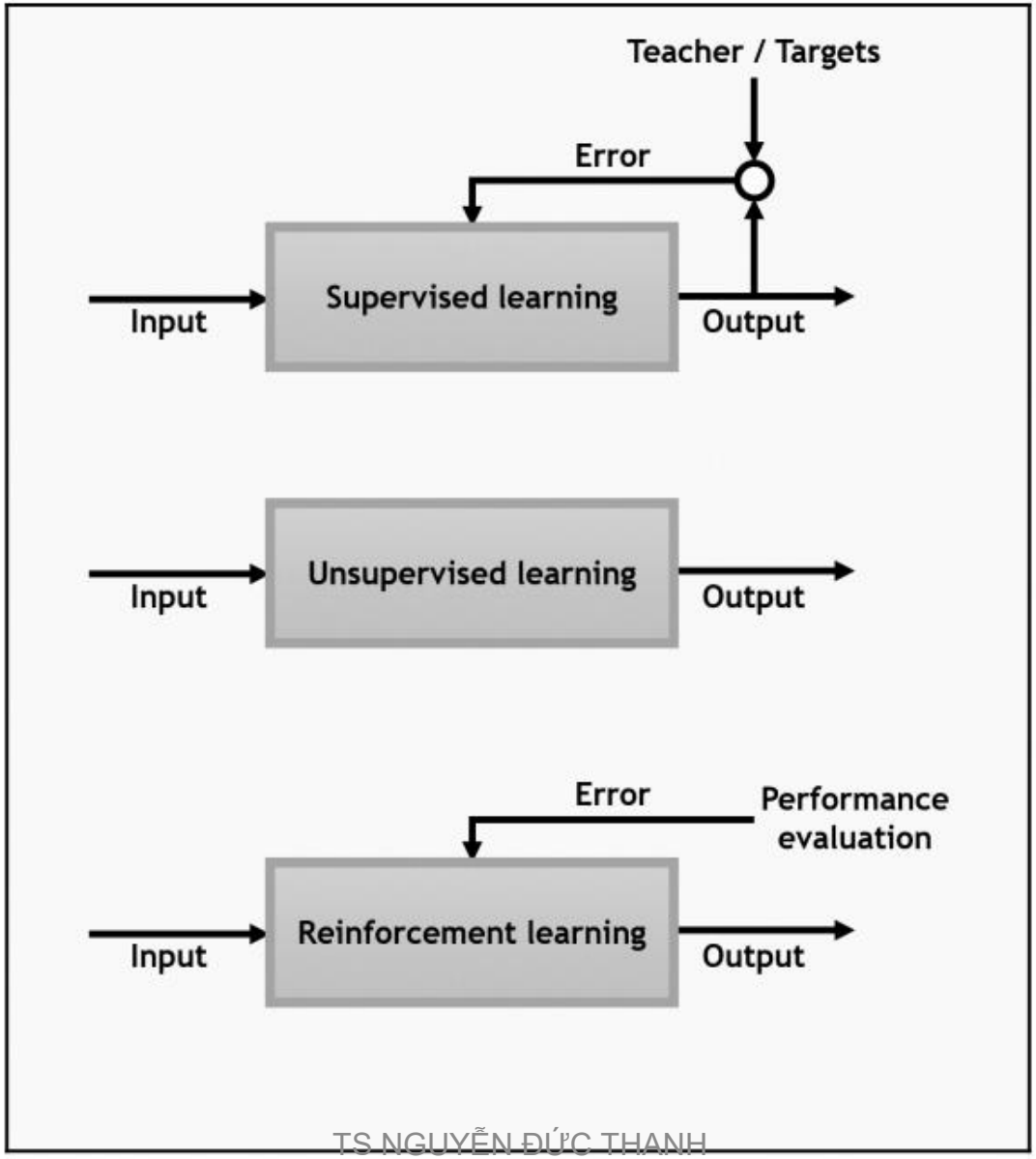
- **Học không giám sát** không cho biết nhãn mà chỉ có dữ liệu đầu vào. Thuật toán sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán. Ví dụ học phân biệt mặt chữ số mà không cho biết đó là chữ số gì.
- **Clustering** phân các dữ liệu gần giống nhau thành một nhóm. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.
- **Association**: là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước. Ví dụ: những khách hàng nam mua quần áo thường có xu hướng mua thêm đồng hồ hoặc thắt lưng; những khán giả xem phim Spider Man thường có xu hướng xem thêm phim Bat Man, dựa vào đó tạo ra một hệ thống gợi ý khách hàng (Recommendation System), thúc đẩy nhu cầu mua sắm.

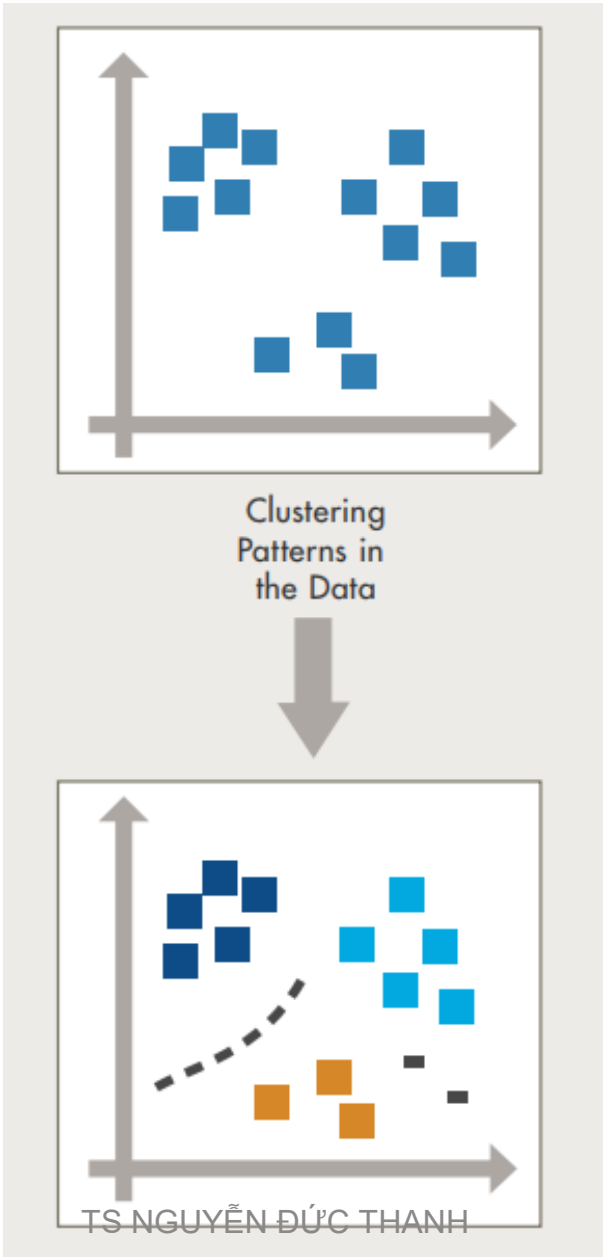
PHÂN LOẠI ML


MACHINE LEARNING
SUPERVISED LEARNING
Labels on Data
UNSUPERVISED LEARNING
No Labels on Data
CLASSIFICATION
Labels Indicate Categories
Many data points may correspond to a single category
REGRESSION
Labels Indicate Values
Data points map to values of some implied function taking the data points as argument and returning some value

Machine Learning Techniques




Teacher / Targets
Error
Supervised learning
Input
Output
Unsupervised learning
Input
Output
Error
Performance evaluation
Reinforcement learning
Input
Output


Clustering
Patterns in
the Data

PHÂN LOẠI ML

- **Semi-Supervised Learning** (Học bán giám sát); khi chúng ta có một lượng lớn dữ liệu nhưng chỉ một phần trong chúng được gán nhãn
- Một ví dụ điển hình là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh/văn bản khác chưa được gán nhãn được thu thập từ internet. Thực tế cho thấy rất nhiều các bài toán Machine Learning thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được (ảnh y học chẳng hạn). Ngược lại, dữ liệu chưa có nhãn có thể được thu thập với chi phí thấp từ internet.
- **Reinforcement Learning** (Học Củng Cố) là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance) dựa vào thưởng và phạt. Hiện tại, Reinforcement learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi (Game Theory), các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.

CÁC THUẬT TOÁN ML

Regression Algorithms

- Linear Regression
- Logistic Regression
- Stepwise Regression

Classification Algorithms

- Linear Classifier
- Support Vector Machine (SVM)
- Kernel SVM
- Sparse Representation-based classification (SRC)

Instance-based Algorithms

- k-Nearest Neighbor (kNN)
- Learning Vector Quantization (LVQ)

CÁC THUẬT TOÁN ML

Regularization Algorithms

- Ridge Regression
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Least-Angle Regression (LARS)

Bayesian Algorithms

- Naive Bayes
- Gaussian Naive Bayes

Clustering Algorithms

- k-Means clustering
- k-Medians
- Expectation Maximization (EM)

CÁC THUẬT TOÁN ML

Artificial Neural Network Algorithms

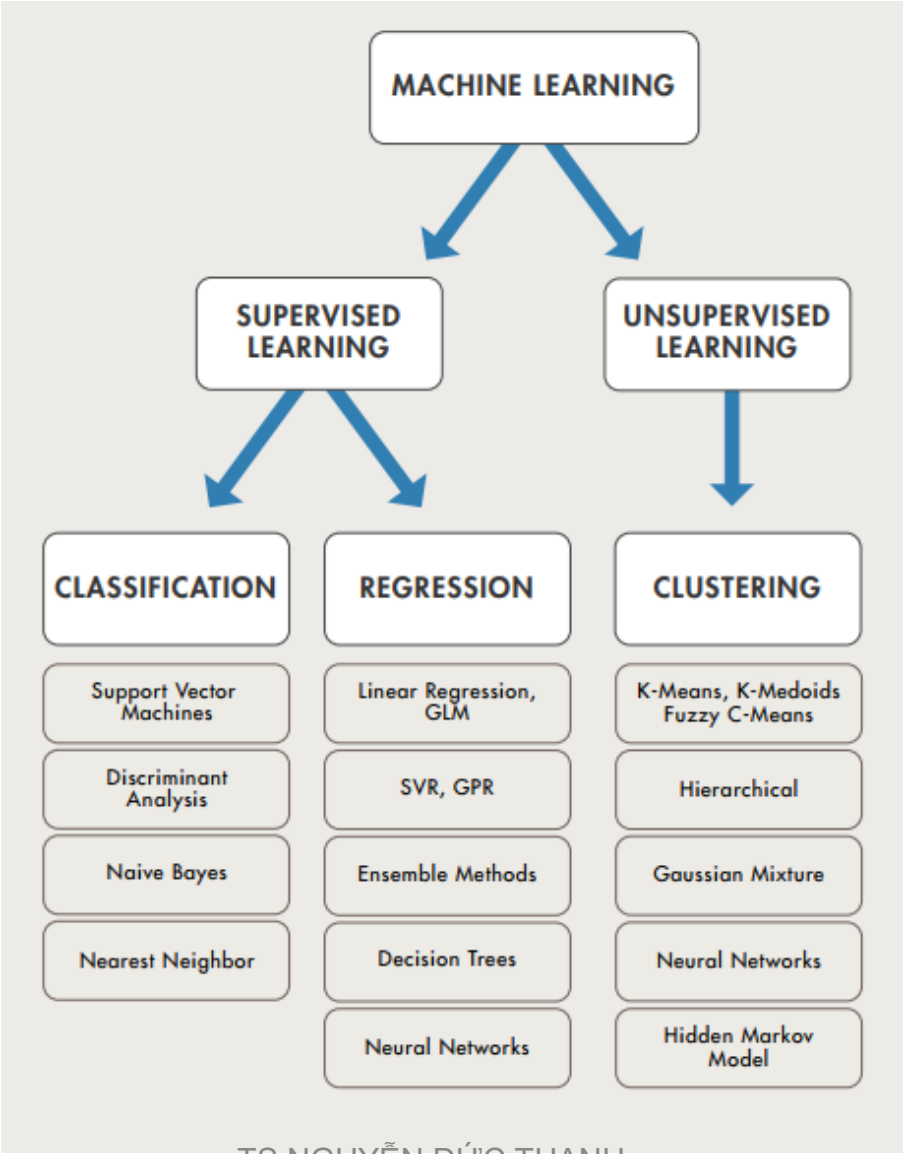
- Perceptron
- Softmax Regression
- Multi-layer Perceptron
- Back-Propagation
- Deep Learning

Dimensionality Reduction Algorithms

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

Ensemble Algorithms

- Boosting
- AdaBoost
- Random Forest


MACHINE LEARNING
SUPERVISED LEARNING
UNSUPERVISED LEARNING
CLASSIFICATION
REGRESSION
CLUSTERING
Support Vector Machines
Discriminant Analysis
Naive Bayes
Nearest Neighbor
Linear Regression, GLM
SVR, GPR
Ensemble Methods
Decision Trees
Neural Networks
K-Means, K-Medoids Fuzzy C-Means
Hierarchical
Gaussian Mixture
Neural Networks
Hidden Markov Model

- Khi nào dùng ML?
- Dùng ML khi cần giải bài toán phức tạp không thể miêu tả bằng công thức và dùng lượng dữ liệu lớn

Hand-written rules and equations are too complex—as in face recognition and speech recognition.

The rules of a task are constantly changing—as in fraud detection from transaction records.

The nature of the data keeps changing, and the program needs to adapt—as in automated trading, energy demand forecasting, and predicting shopping trends.

Creating Algorithms that Can Analyze Works of Art

Researchers at the Art and Artificial Intelligence Laboratory at Rutgers University wanted to see whether a computer algorithm could classify paintings by style, genre, and artist as easily as a human. They began by identifying visual features for classifying a painting's style. The algorithms they developed classified the styles of paintings in the database with 60% accuracy, outperforming typical non-expert humans.

The researchers hypothesized that visual features useful for style classification (a supervised learning problem) could also be used to determine artistic influences (an unsupervised problem).

They used classification algorithms trained on Google images to identify specific objects. They tested the algorithms on more than 1,700 paintings from 66 different artists working over a span of 550 years. The algorithm readily identified connected works, including the influence of Diego Velazquez's "Portrait of Pope Innocent X" on Francis Bacon's "Study After Velazquez's Portrait of Pope Innocent X."

Optimizing HVAC Energy Usage in Large Buildings

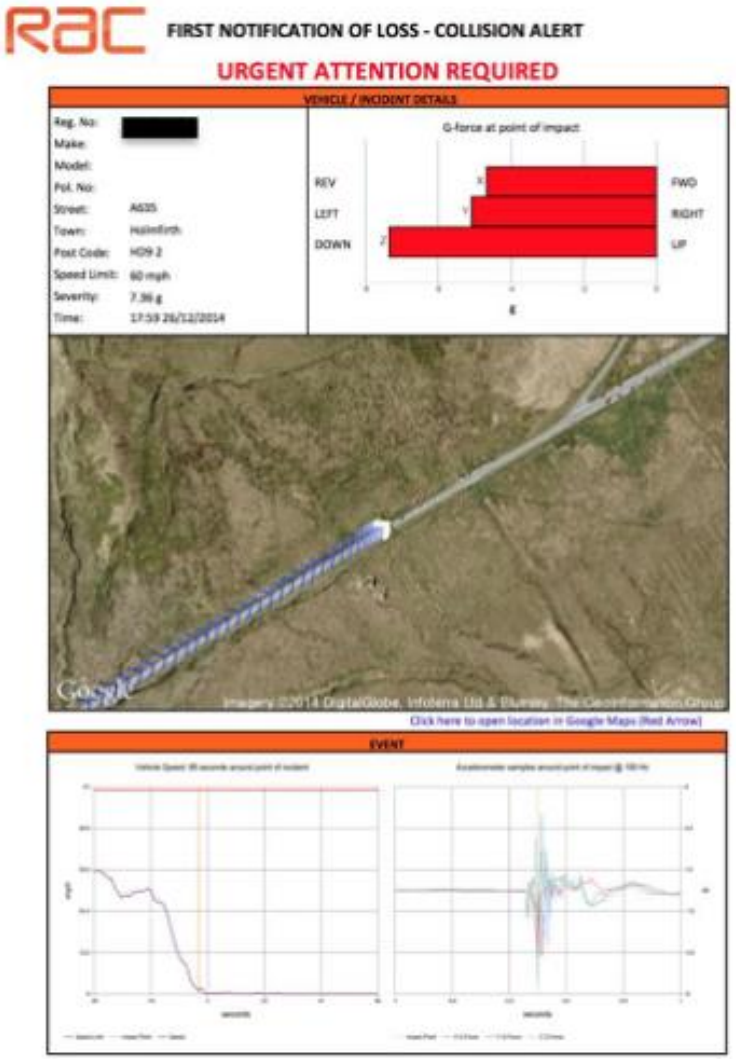
The heating, ventilation, and air-conditioning (HVAC) systems in office buildings, hospitals, and other large-scale commercial buildings are often inefficient because they do not take into account changing weather patterns, variable energy costs, or the building's thermal properties.

Building IQ's cloud-based software platform addresses this problem. The platform uses advanced algorithms and machine learning methods to continuously process gigabytes of information from power meters, thermometers, and HVAC pressure sensors, as well as weather and energy cost. In particular, machine learning is used to segment data and determine the relative contributions of gas, electric, steam, and solar power to heating and cooling processes. The building IQ platform reduces HVAC energy consumption in large-scale commercial buildings by 10% - 25% during normal operation.

Detecting Low-Speed Car Crashes

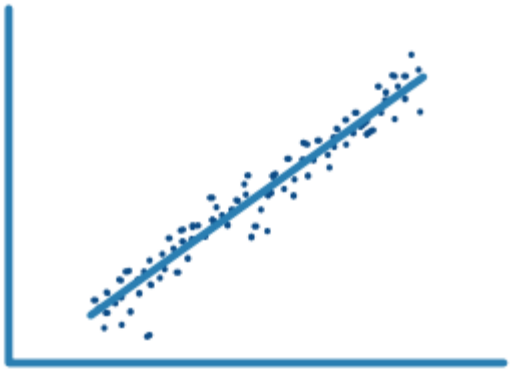
With more than 8 million members, the RAC is one of the UK's largest motoring organizations, providing roadside assistance, insurance, and other services to private and business motorists.

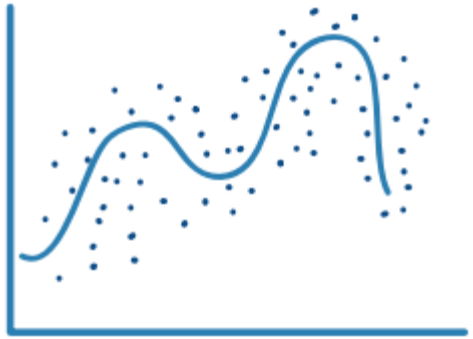
To enable rapid response to roadside incidents, reduce crashes, and mitigate insurance costs, the RAC developed an onboard crash sensing system that uses advanced machine learning algorithms to detect low-speed collisions and distinguish these events from more common driving events, such as driving over speed bumps or potholes. Independent tests showed the RAC system to be 92% accurate in detecting test crashes.



REGRESSION HỒI QUI

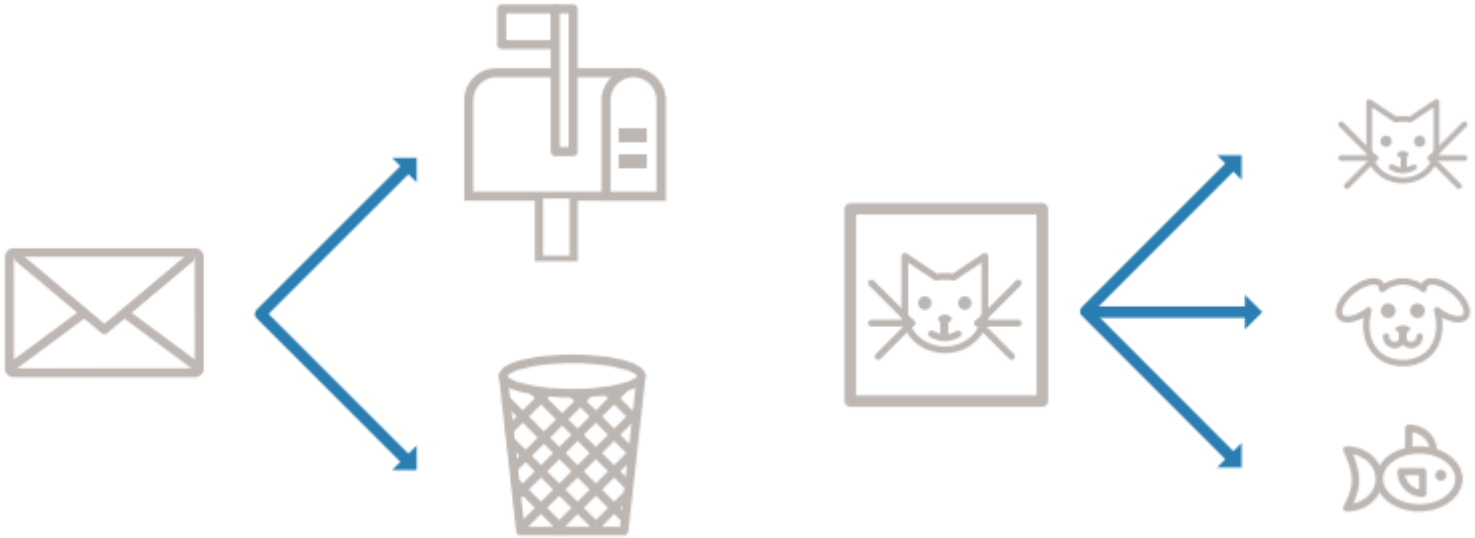
- Regression là quay về trạng thái trước ít phát triển hơn, đơn giản hơn so với thực tế.
- Linear regression là tìm một quan hệ tuyến tính giữa một đại lượng ra và nhiều đại lượng vào
- Nonlinear regression tìm quan hệ phi tuyến





CLASSIFICATION

- Phân lớp thành hai hay nhiểu lớp
- Phân email thành spam hay không spam
- Phân biệt một con vật, đồ vật…



BINARY CLASSIFICATION Logistic Regression

- Fits a model that can predict the probability of a binary response belonging to one class or the other. Because of its simplicity, logistic regression is commonly used as a starting point for binary classifcation problems.
- Best Used... when data can be clearly separated by a single linear boundary, as a baseline for evaluating more complex classifcation methods

CLASSIFICATION k Nearest Neighbor (kNN)

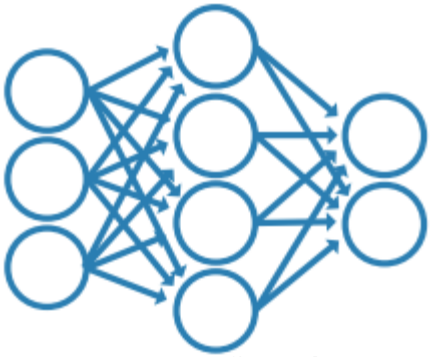
- kNN categorizes objects based on the classes of their nearest neighbors in the dataset. kNN predictions assume that objects near each other are similar. Distance metrics, such as Euclidean, city block, cosine, and Chebychev, are used to fnd the nearest neighbor.
- Best Used...
  - When you need a simple algorithm to establish benchmark learning rules
  - When memory usage of the trained model is a lesser concern
  - When prediction speed of the trained model is a lesser concern

Support Vector Machine (SVM)

- Classifes data by fnding the linear decision boundary (hyperplane) that separates all data points of one class from those of the other class. The best hyperplane for an SVM is the one with the largest margin between the two classes, when the data is linearly separable. If the data is not linearly separable, a loss function is used to penalize points on the wrong side of the hyperplane. SVMs sometimes use a kernel transform to transform nonlinearly separable data into higher dimensions where a linear decision boundary can be found.
- Best Used...
  - For data that has exactly two classes (you can also use it for multiclass classifcation with a technique called errorcorrecting output codes)
  - For high-dimensional, nonlinearly separable data
  - When you need a classifer that’s simple, easy to interpret, and accurate

CLASSIFICATION Neural Network

- Inspired by the human brain, a neural network consists of highly connected networks of neurons that relate the inputs to the desired outputs. The network is trained by iteratively modifying the strengths of the connections so that given inputs map to the correct response.
- Best Used...
  - For modeling highly nonlinear systems
  - When data is available incrementally and you wish to constantly update the model
  - When there could be unexpected changes in your input data
  - When model interpretability is not a key concern

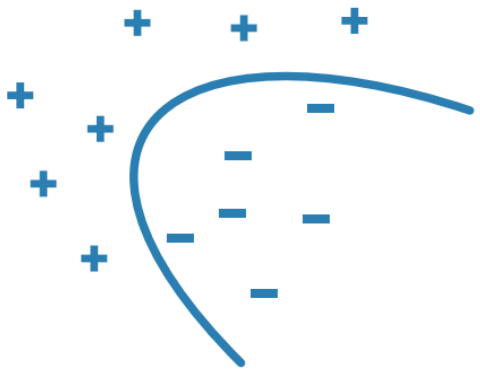


CLASSIFICATION Naïve Bayes

- A naive Bayes classifer assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It classifes new data based on the highest probability of its belonging to a particular class.
- Best Used...
  - For a small dataset containing many parameters
  - When you need a classifer that's easy to interpret
  - When the model will encounter scenarios that weren't in the training data, as is the case with many fnancial
  - and medical applications

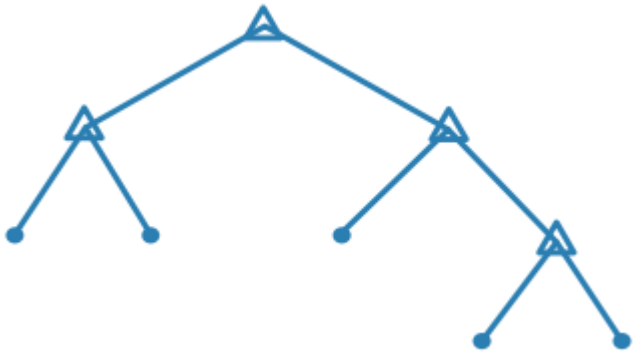
CLASSIFICATION **Discriminant Analysis**

- Discriminant analysis classifes data by fnding linear combinations of features. Discriminant analysis assumes that different classes generate data based on Gaussian distributions. Training a discriminant analysis model involves fnding the parameters for a Gaussian distribution for each class. The distribution parameters are used to calculate boundaries, which can be linear or quadratic functions. These boundaries are used to determine the class of new data.
- Best Used...
  - When you need a simple model that is easy to interpret
  - When memory usage during training is a concern
  - When you need a model that is fast to predict



CLASSIFICATION **Decision Tree**

- A decision tree lets you predict responses to data by following the decisions in the tree from the root (beginning) down to a leaf node. A tree consists of branching conditions where the value of a predictor is compared to a trained weight. The number of branches and the values of weights are determined in the training process. Additional modifcation, or pruning, may be used to simplify the model.
- Best Used...
  - When you need an algorithm that is easy to interpret and fast to fit
  - To minimize memory usage
  - When high predictive accuracy is not a requirement



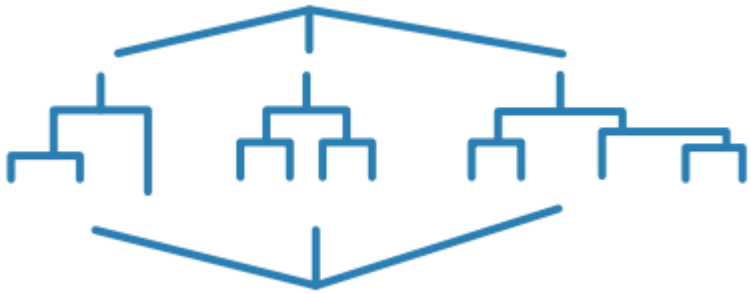
CLASSIFICATION Bagged and Boosted Decision Tree

In these ensemble methods, several “weaker” decision trees are combined into a “stronger” ensemble.

- A bagged decision tree consists of trees that are trained independently on data that is bootstrapped from the input data.
- Boosting involves creating a strong learner by iteratively adding “weak” learners and adjusting the weight of each weak learner to focus on misclassifed examples.

Best Used...

- When predictors are categorical (discrete) or behave nonlinearly
- When the time taken to train a model is less of a concern



CLUSTERING *k*-Means

- Partitions data into k number of mutually exclusive clusters.

How well a point fts into a cluster is determined by the distance from that point to the cluster's center.

Best Used...

- When the number of clusters is known
- For fast clustering of large data sets

CLUSTERING *k*-Medoids

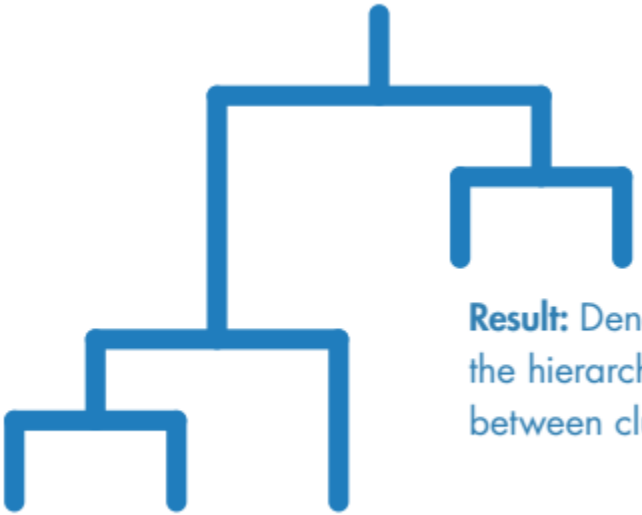
- Similar to k-means, but with the requirement that the cluster centers coincide with points in the data.
  Best Used...
  - When the number of clusters is known
  - For fast clustering of categorical data
  - To scale to large data sets

Hierarchical Clustering

Produces nested sets of clusters by analyzing similarities between pairs of points and grouping objects into a binary hierarchical tree.

Best Used...

- When you don’t know in advance how many clusters are in your data
- You want visualization to guide your selection



Result: Dendrogram showing the hierarchical relationship between clusters

Self-Organizing Map

Neural-network based clustering that transforms a dataset into a topology-preserving 2D map.

Best Used...

- To visualize high-dimensional data in 2D or 3D
- To deduce the dimensionality of data by preserving its topology (shape)

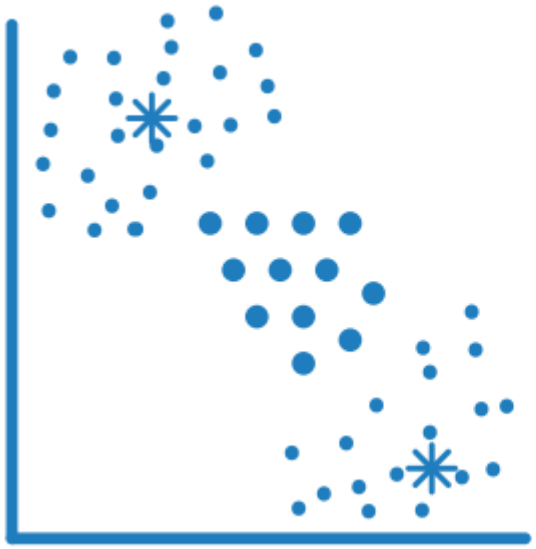
Result:
Lower-dimensional (typically 2D) representation

Fuzzy c-Means

Partition-based clustering when data points may belong to more than one cluster.

Best Used...

- When the number of clusters is known
- For pattern recognition
- When clusters overlap



Result: Cluster centers (similar to k-means) but with fuzziness so that points may belong to more than one cluster

Gaussian Mixture Model

Partition-based clustering where data points come from different multivariate normal distributions with certain probabilities.

Best Used...

- When a data point might belong to more than one cluster
- When clusters have different sizes and correlation structures within them

Result: A model of Gaussian distributions that give probabilities of a point being in a cluster

Dimensionality Reduction

Machine learning is an effective method for fnding patterns in big datasets. But bigger data brings added complexity.

As datasets get bigger, you frequently need to reduce the number of features, or dimensionality

Suppose you have electroencephalogram (EEG) data that captures

electrical activity of the brain, and you want to use this data to predict a future seizure. The data was captured using dozens of leads, each corresponding to a variable in your original dataset.

Each of these variables contains noise. To make your prediction

algorithm more robust, you use dimensionality reduction techniques to derive a smaller number of features. Because these features are calculated from multiple sensors, they will be less susceptible to noise in an individual sensor than would be the case if you used the raw data directly

Principal component analysis (PCA

Performs a linear transformation on the data so that most of the variance or information in your high-dimensional dataset is captured by the frst few principal components. The frst principal component will capture the most variance, followed by the second principal component, and so on.

In datasets with many variables, groups of variables often move

together. PCA takes advantage of this redundancy of information

by generating new variables via linear combinations of the original

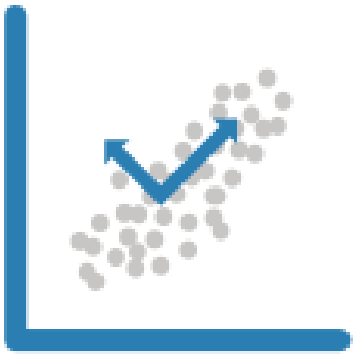
variables so that a small number of new variables

captures most of the information.

Each principal component is a linear combination

of the original variables. Because all the principal comp orthogonal to each other, there is no

redundant information.



Factor analysis

Identifes underlying correlations between variables in your dataset to provide a representation in terms of a smaller number of unobserved latent, or common, factors

Your dataset might contain measured variables that overlap,

meaning that they are dependent on one another. Factor

analysis lets you ft a model to multivariate data to estimate

this sort of interdependence.

In a factor analysis model, the measured variables

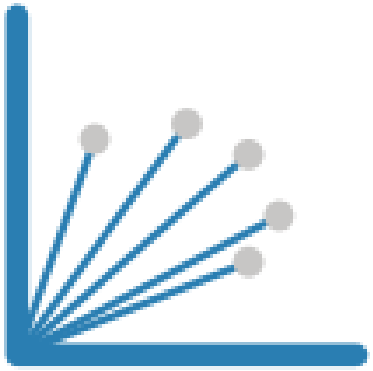
depend on a smaller number of unobserved (latent)

factors. Because each factor might affect several

variables, it is known as a common factor.

Each variable is assumed to be dependent

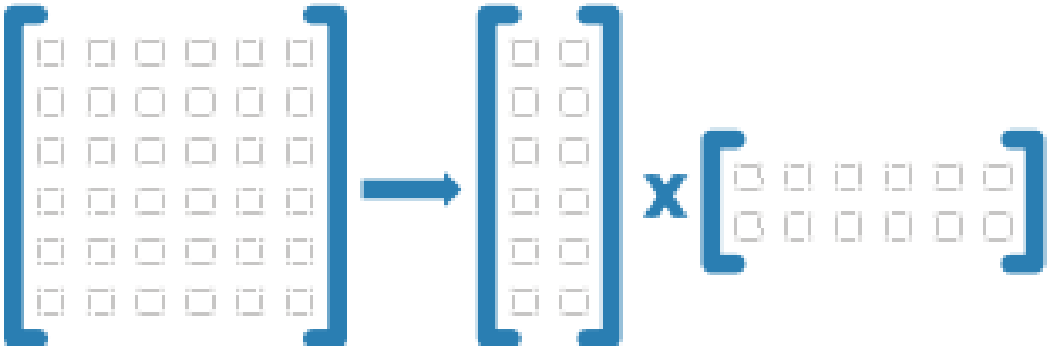
on a linear combination of the common factors.



Nonnegative matrix factorization

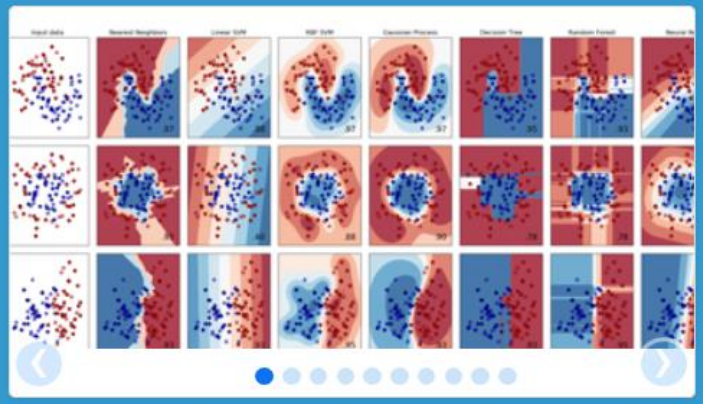
Used when model terms must represent nonnegative quantities, such as physical quantities.

This dimension reduction technique is based on a low-rank approximation of the feature space. In addition to reducing the number of features, it guarantees that the features are nonnegative, producing models that respect features such as the nonnegativity of physical quantities.



PHẦN MỀM ML

- Matlab: Statistics and Machine Learning Toolbox, Computer Vision System Toolbox™, and Image Processing Toolbox™.
- Opencv: modules ml, dnn (deep neural network), objdetect
- Python: numpy, scikit_learn



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, … — Examples

Regression

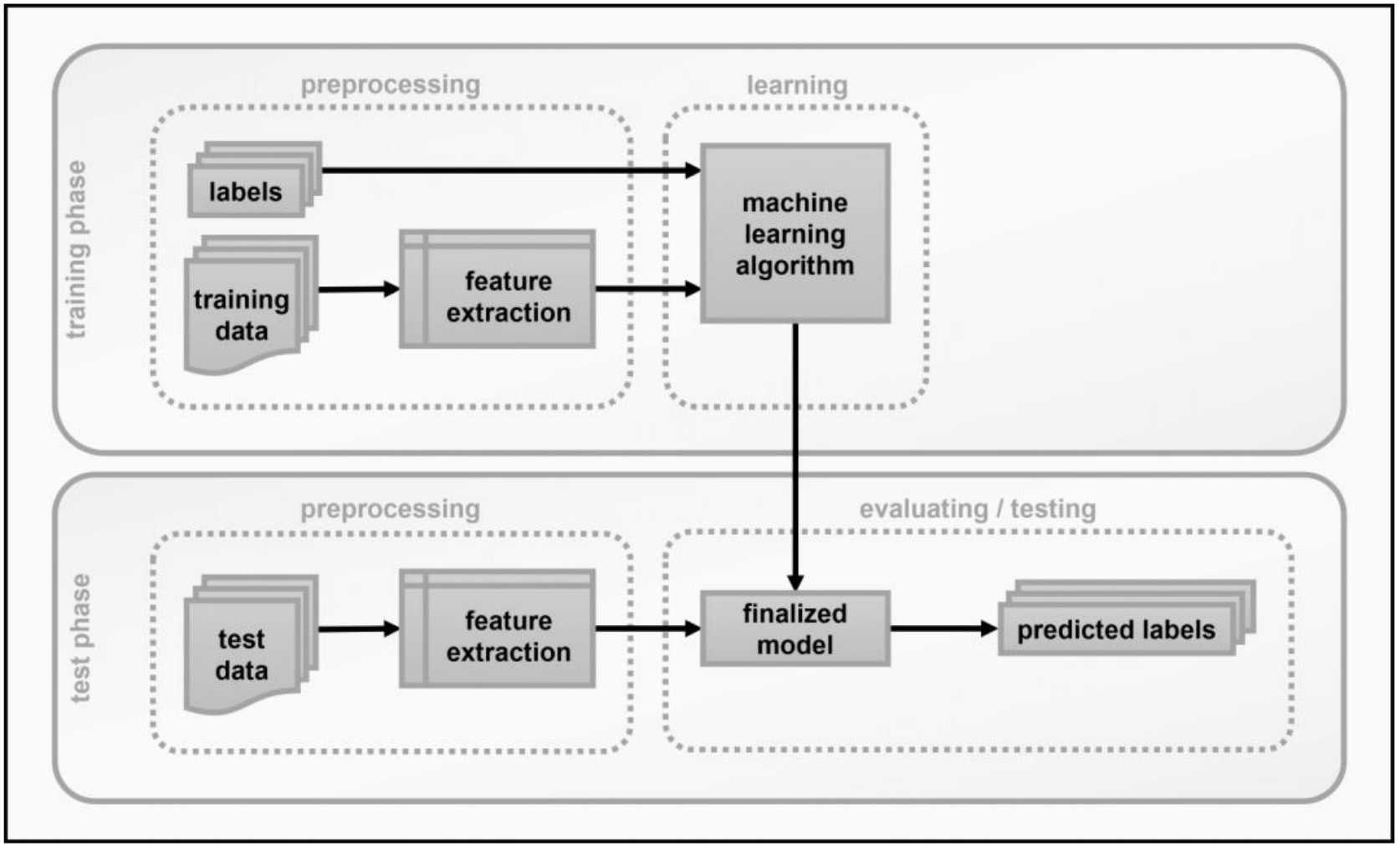
Predicting a continuous-valued attribute associated with an object.

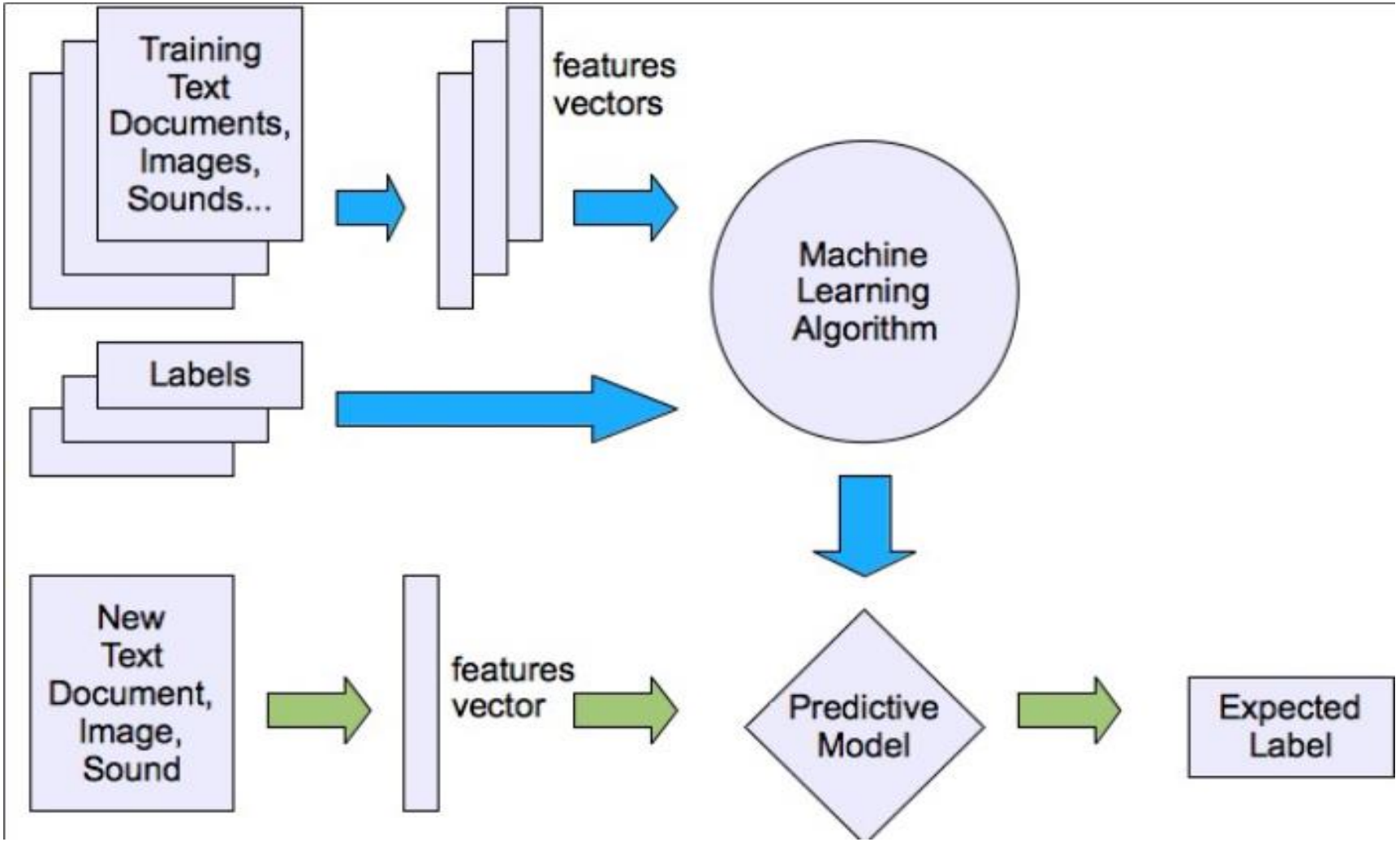
Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, … — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, … — Examples


preprocessing
learning
training phase
labels
training data
feature extraction
machine learning algorithm
preprocessing
evaluating / testing
test phase
test data
feature extraction
finalized model
predicted labels


Training
Text
Documents,
Images,
Sounds...
features
vectors
Machine
Learning
Algorithm
Labels
New
Text
Document,
Image,
Sound
features
vector
Predictive
Model
Expected
Label

ML AND COMPUTER VISION

As a form of artificial intelligence, machine learning enables computers to learn through experience: to make predictions about the future using collected data from the past. On top of that, computer vision is one of today's most exciting application fields of machine learning, with deep learning and convolutional neural networks driving innovative systems such as self-driving cars and Google's DeepMind.

CÁC THUẬT TOÁN ML TRONG OPENCV

Algorithm	Comment
Mahalanobis	A distance measure that accounts for the "stretchiness" of the data space by dividing out the covariance of the data. If the covariance is the identity matrix (identical variance), then this measure is identical to the Euclidean distance measure [Mahalanobis36].
K-means	An unsupervised clustering algorithm that represents a distribution of data using *K* centers, where *K* is chosen by the user. The difference between this algorithm and expectation maximization (described shortly) is that here the centers are not Gaussian and the resulting clusters look more like soap bubbles, since centers (in effect) compete to "own" the closest data points. These cluster regions are often used as sparse histogram bins to represent the data. Invented by Steinhaus [Steinhaus56], as used by Lloyd [Lloyd57].
Normal/Naïve Bayes classifier	A generative classifier in which features are assumed to be Gaussian distributed and statistically independent from one another, a strong assumption that is generally not true. For this reason, it's often called a "naïve Bayes" classifier. However, this method often works surprisingly well. Original mention [Maron61; Minsky61].
Decision trees	A discriminative classifier. The tree finds one data feature and a threshold at the current node that best divides the data into separate classes. The data is split and we recursively repeat the procedure down the left and right branches of the tree. Though not usually the top performer, it's often the first thing you should try because it is fast and has high functionality [Breiman84].
Expectation maximization (EM)	A generative unsupervised algorithm that is used for clustering. It will fit *N* multidimensional Gaussians to the data, where *N* is chosen by the user. This can be an effective way to represent a more complex distribution with only a few parameters (means and variances). Often used in segmentation. Compare with K-means listed previously [Dempster77].

CÁC THUẬT TOÁN ML TRONG OPENCV

Algorithm	Comment
Boosting	A discriminative group of classifiers. The overall classification decision is made from the combined weighted classification decisions of the group of classifiers. In training, we learn the group of classifiers one at a time. Each classifier in the group is a "weak" classifier (only just above chance performance). These weak classifiers are typically composed of single-variable decision trees called *stumps*. In training, the decision stump learns its classification decisions from the data and also learns a weight for its "vote" from its accuracy on the data. Between training each classifier one by one, the data points are reweighted so that more attention is paid to data points where errors were made. This process continues until the total error over the data set, arising from the combined weighted vote of the decision trees, falls below a set threshold. This algorithm is often effective when a large amount of training data is available [Freund97].
Random trees	A discriminative forest of many decision trees, each built down to a large or maximal splitting depth. During learning, each node of each tree is allowed to choose splitting variables only from a random subset of the data features. This helps ensure that each tree becomes a statistically independent decision maker. In run mode, each tree gets an unweighted vote. This algorithm is often very effective and can also perform regression by averaging the output numbers from each tree implemented [Ho95; Criminisi13; Breiman01].

CÁC THUẬT TOÁN ML TRONG OPENCV

K-nearest neighbors	The simplest possible discriminative classifier. Training data are simply stored with labels. Thereafter, a test data point is classified according to the majority vote of its *K* nearest other data points (in a Euclidean sense of nearness). This is probably the simplest thing you can do. It is often effective but it is slow and requires lots of memory [Fix51], but see the FLANN entry.
Fast Approximate Nearest Neighbors (FLANN)[a]	OpenCV includes a full fast approximate nearest neighbor library developed by Marius Muja [Muja09]. This allows fast approximations to nearest neighbor and *K* nearest neighbor matching.
Support vector machine (SVM)	A discriminative classifier that can also do regression. A distance function between any two data points in a higher-dimensional space is defined. (Projecting data into higher dimensions makes the data more likely to be linearly separable.) The algorithm learns separating hyperplanes that maximally separate the classes in the higher dimension. It tends to be among the best with limited data, losing out to boosting or random trees only when large data sets[b] are available [Vapnik95].
Face detector/cascade classifier	An object detection application based on a clever use of boosting. The OpenCV distribution comes with a trained frontal face detector that works remarkably well. You may train the algorithm on other objects with the software provided. You may also use other features or create your own features for this classifier. It works well for rigid objects and characteristic views. After its inventors, this classifier is also commonly known as a "Viola-Jones Classifier" [Viola04].

CÁC THUẬT TOÁN ML TRONG OPENCV

Waldboost	A derivative of the cascade method of Viola (see the preceding entry), Waldboost is an object detector that is very fast and often outperforms the traditional cascade classifier for a variety of tasks [Sochman05]. It is in *.../opencv_contrib/modules*.
Latent SVM	The Latent SVM method uses a parts-based model to identify composite objects based on recognizing the individual components of the object and learning a model of how those components should expect to be found relative to one another [Felzenszwalb10].
Bag of Words	The Bag of Words method generalizes techniques heavily used in document classification to visual image classification. This method is powerful because it can be used to identify not only individual objects, but often scenes and environments as well.

Deep Lerning

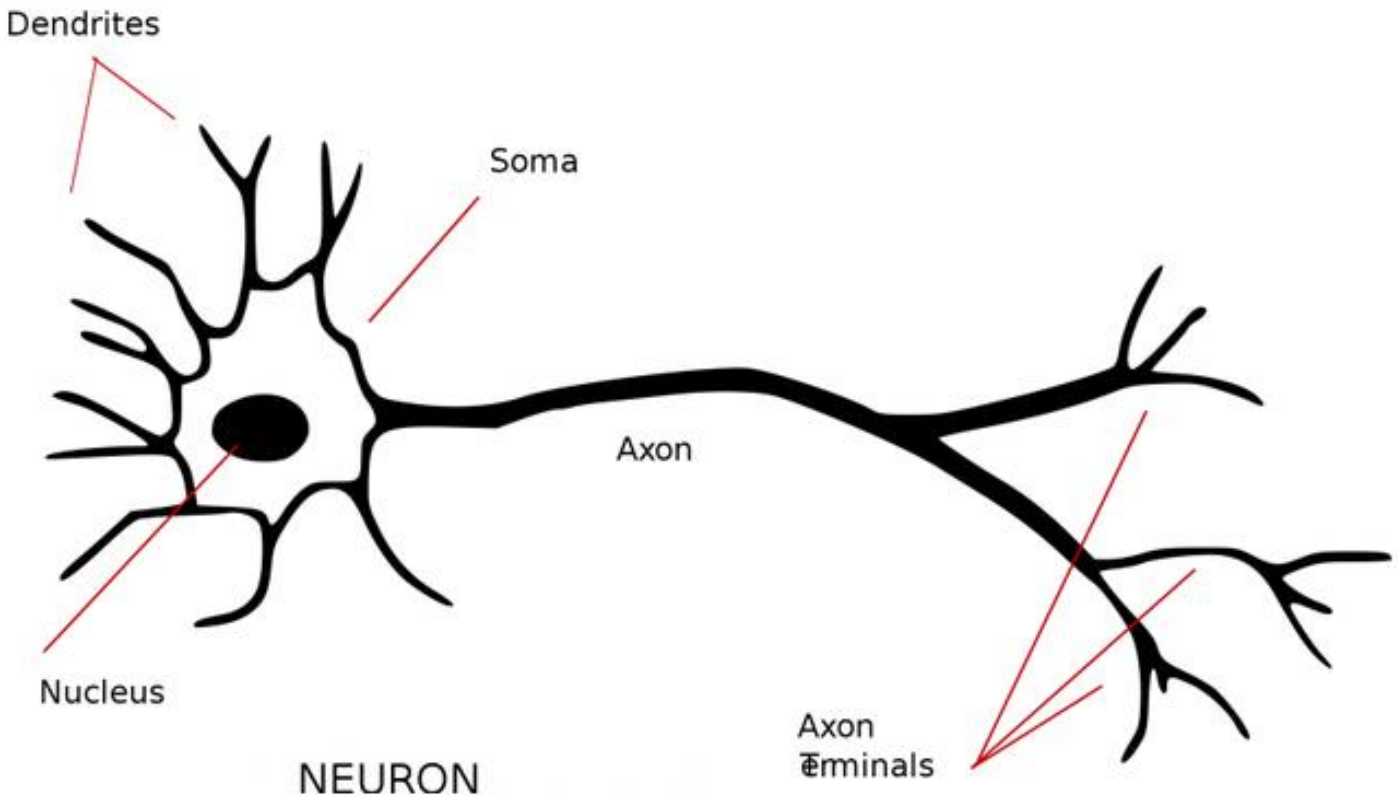
MACHINE LEARNING TRONG THỊ GIÁC MÁY TÍNH

- Thường dùng để phân lớp, phát giác vật.
- Cần huấn luyện với tập dữ liệu lớn
- Ví dụ phát giác người trong ảnh, cần chuẩn bị tập ảnh có người và không có người, người có thể xuất hiện ở nhiều góc độ, tư thế, gần xa, bị che, độ chiếu sáng thay đổi, hình nền thay đổi.
- Ta phải gán nhãn cho ảnh tùy theo muốn nhận dạng cái gì.
- Dùng các thuật toán để trích xuất đặc trưng cần thiết dùng cho huấn luyện
- Các đặc trưng tạo thành vector gọi là feature vector
- Cần giảm kích thước feature vector để huấn luyện nhanh dùng các phương pháp như PCA

- Tập dữ liệu mẫu bao gồm training dataset, validation dataset và test dataset
- Huấn luyện là xây dựng bộ phân lớp classifier và điều chỉnh thông số bộ phân lớp sao cho sai số ít nhất, tức là cực tiểu hàm tổn thất.

PERCEPTRON

- Mô hình đơn giản của một neuron trong não người. Nhà sinh lý học Frank Rosenblatt đề xuất năm 1955 để nghiên cứu hoạt động não bộ.



PERCEPTRON

- Mô hình phân 2 lớp
- Có N dữ liệu, mỗi dữ liệu d chiều, X=[x1,x2,…,xN]∈ Rd×N là ma trận chứa các điểm dữ liệu mà mỗi cột xi∈Rd×1 là một điểm dữ liệu trong không gian d chiều.
- Thêm các nhãn tương ứng với từng điểm dữ liệu được lưu trong một vector hàng y=[y1,y2,…,yN] ∈R1×N, với yi=1 nếu xi thuộc class 1 (xanh) và yi=−1 nếu xi thuộc class 2 (đỏ).
- Hàm phân lớp $$f_{\mathbf{w}}(\mathbf{x}) = w_1 x_1 + \cdots + w_d x_d + w_0 = \mathbf{w}^T \bar{\mathbf{x}} = 0$$
- $\bar{\mathbf{x}}$ là điểm dữ liệu mở rộng bằng cách thêm phần tử x0=1 lên trước vector x , sau đây thay $\bar{\mathbf{x}}$ bằng x

PERCEPTRON

- Tìm các trọng số wi để cực tiểu hàm tổn thất
- Gỉa sử dữ liệu 2 chiều f(x)>0 nếu thuộc lớp 1 và <0 nếu thuộc lớp 0
- Nếu yi trái dấu f(xi) thì phân lớp sai, hàm tổn thất do phân lớp sai là

$$J(\mathbf{w}) = \sum_{\mathbf{x}_i \in \mathcal{M}} (-y_i \mathbf{w}^T \mathbf{x_i})$$

- M là tập hợp các dữ liệu phân lớp sai
- Cực tiểu hàm J để tìm qui luật thay đổi trọng số

$$J(\mathbf{w}; \mathbf{x}_i; y_i) = -y_i \mathbf{w}^T \mathbf{x}_i$$

$$\nabla_{\mathbf{w}} J(\mathbf{w}; \mathbf{x}_i; y_i) = -y_i \mathbf{x}_i$$

$$\mathbf{w} = \mathbf{w} + \eta y_i \mathbf{x}_i$$ η >0 gọi là vận tốc học learning rate

PERCEPTRON

1/Chọn ngẫu nhiên một vector hệ số w với các phần tử gần 0.

2/Duyệt ngẫu nhiên qua từng điểm dữ liệu xi:

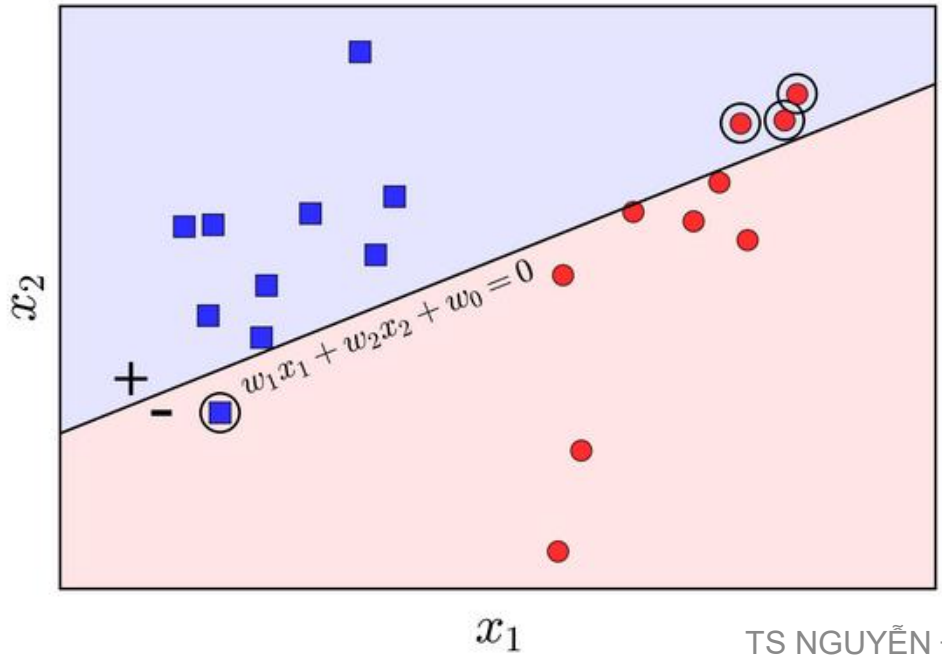
Nếu xi được phân lớp đúng, tức $sgn(w^T xi)=yi$

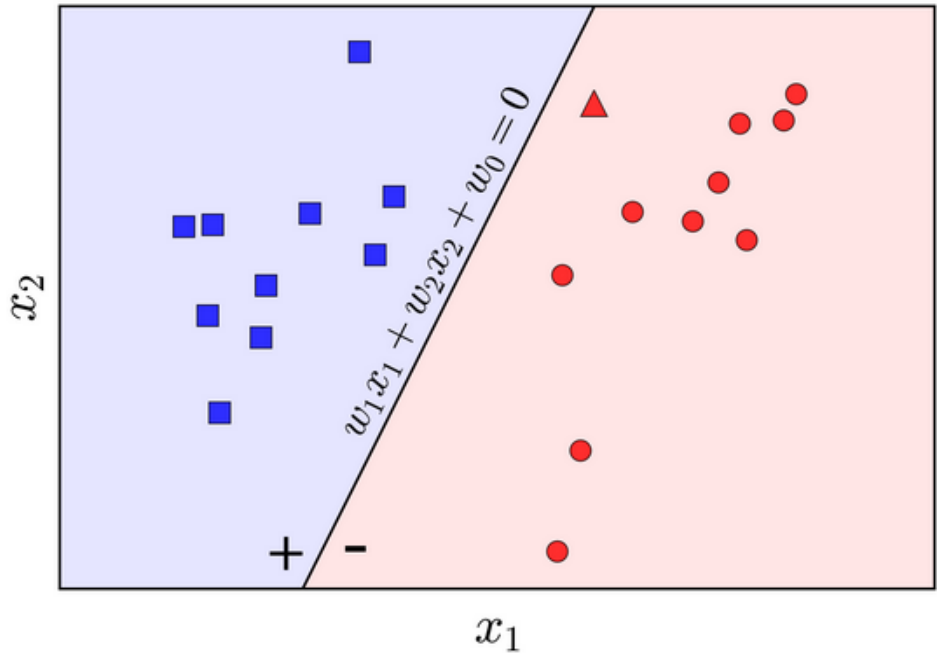
, chúng ta không cần làm gì.

Nếu xi bị phân lớp sai, cập nhật w theo công thức: $w=w+\eta yixi$

3/Kiểm tra xem có bao nhiêu điểm bị phân lớp sai. Nếu không còn điểm nào, dừng thuật toán. Nếu còn, quay lại bước 2.

https://github.com/tiepvupsu/tiepvupsu.github.io/blob/master/assets/pla/perceptron.py





HUẤN LUYỆN PERCEPTRON PYTHON

```
#perceptron learning
#adapted from Vu Huu Tiep https://machinelearningcoban.com/
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(2)
means = [[2, 2], [4, 2]]
cov = [[.3, .2], [.2, .3]]
N = 10
X0 = np.random.multivariate_normal(means[0], cov, N).T
X1 = np.random.multivariate_normal(means[1], cov, N).T
X = np.concatenate((X0, X1), axis = 1)
```

HUẤN LUYỆN PERCEPTRON PYTHON

```
#label
y = np.concatenate((np.ones((1, N)), -1*np.ones((1, N))), axis = 1)
#Xbar
X = np.concatenate((np.ones((1, 2*N)), X), axis = 0)
#calculate output
def h(w, x):
    return np.sign(np.dot(w.T, x))
#stop
def has_converged(X, y, w):
     return np.array_equal(h(w, X), y)
#True if h(w, X) == y else False
```

HUẤN LUYỆN PERCEPTRON PYTHON

```
#training function
def perceptron(X, y, w_init):
    w = [w_init]
    N = X.shape[1]
    while True:
        # mix data
        mix_id = np.random.permutation(N)
        for i in range(N):
            xi = X[:, mix_id[i]].reshape(3, 1)
            yi = y[0, mix_id[i]]
```

HUẤN LUYỆN PERCEPTRON PYTHON

```
 if h(w[-1], xi)[0] != yi:
                w_new = w[-1] + yi*xi #w[-1]:last element of w
                w.append(w_new)
          if has_converged(X, y, w[-1]):
             break
    return (w)
d = X.shape[0]
#init w
w_init = np.random.randn(d, 1)
#training
w = perceptron(X, y, w_init)
l=len(w)
print('iteration',l-1)
```

HUẤN LUYỆN PERCEPTRON PYTHON

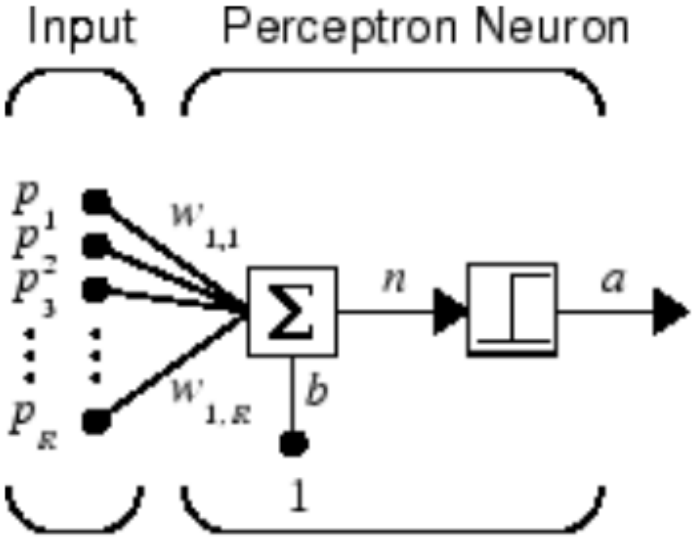
```
b1=w[l-1]
print('Weight Vector %5.2f,%5.2f,%5.2f' %(b1[0],b1[1],b1[2]))
def draw_line(w):
    w0, w1, w2 = w[0], w[1], w[2]
    if w2 != 0:
        x11, x12 = -100, 100
        return plt.plot([x11, x12], [-(w1*x11 + w0)/w2, -(w1*x12 + w0)/w2],
'k')
    else:
        x10 = -w0/w1
        return plt.plot([x10, x10], [-100, 100], 'k')
# Visualization
def show():
    fig, ax = plt.subplots(figsize=(5, 5))
    ani = plt.cla()
```

HUẤN LUYỆN PERCEPTRON PYTHON

```
    #points
    ani = plt.plot(X0[0, :], X0[1, :], 'b^', markersize = 8, alpha =
    .8)
    ani = plt.plot(X1[0, :], X1[1, :], 'ro', markersize = 8, alpha =
    .8)
    ani = plt.axis([0 , 6, -2, 4])
    ani = draw_line(w[l-1])
    # hide axis
    cur_axes = plt.gca()
    cur_axes.axes.get_xaxis().set_ticks([])
    cur_axes.axes.get_yaxis().set_ticks([])
    label = 'Perceptron Learning: iter %d/%d' %(l-1, l-1)
    ax.set_xlabel(label)
    plt.show()
show()
```

Classification with a 2-Input Perceptron MATLAB

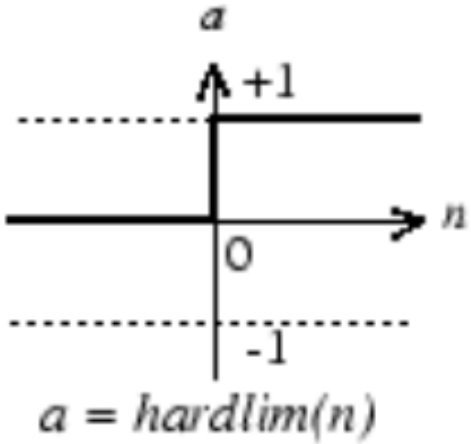
- A 2-input hard limit neuron is trained to classify 5 input vectors into two categories.
- Each of the five column vectors in X defines a 2-element input vectors and a row vector T defines the vector's target categories. We can plot these vectors with PLOTPV.
- The perceptron must properly classify the 5 input vectors in X into the two categories defined by T. Perceptrons have HARDLIM neurons. These neurons are capable of separating an input space with a straight line into two categories (0 and 1).
- PERCEPTRON creates a new neural network with a single neuron. The network is then configured to the data, so we can examine its initial weight and bias values. (Normally the configuration step can be skipped as it is automatically done by ADAPT or TRAIN.)



$a = \mathbf{hardlim}(\mathbf{W}\mathbf{p} + b)$

Where

R = number of elements in input vector



$a = hardlim(n)$

Classification with a 2-Input Perceptron MATLAB

- The input vectors are replotted with the neuron's initial attempt at classification.
- Here the input and target data are converted to sequential data (cell array where each column indicates a timestep) and copied three times to form the series XX and TT.
- ADAPT updates the network for each timestep in the series and returns a new network object that performs as a better classifier.
- Now SIM is used to classify any other input vector, like [0.7; 1.2]. A plot of this new point with the original training set shows how the network performs. To distinguish it from the training set, color it red.
- .The perceptron correctly classified our new point (in red) as category "zero" (represented by a circle) and not a "one" (represented by a plus).

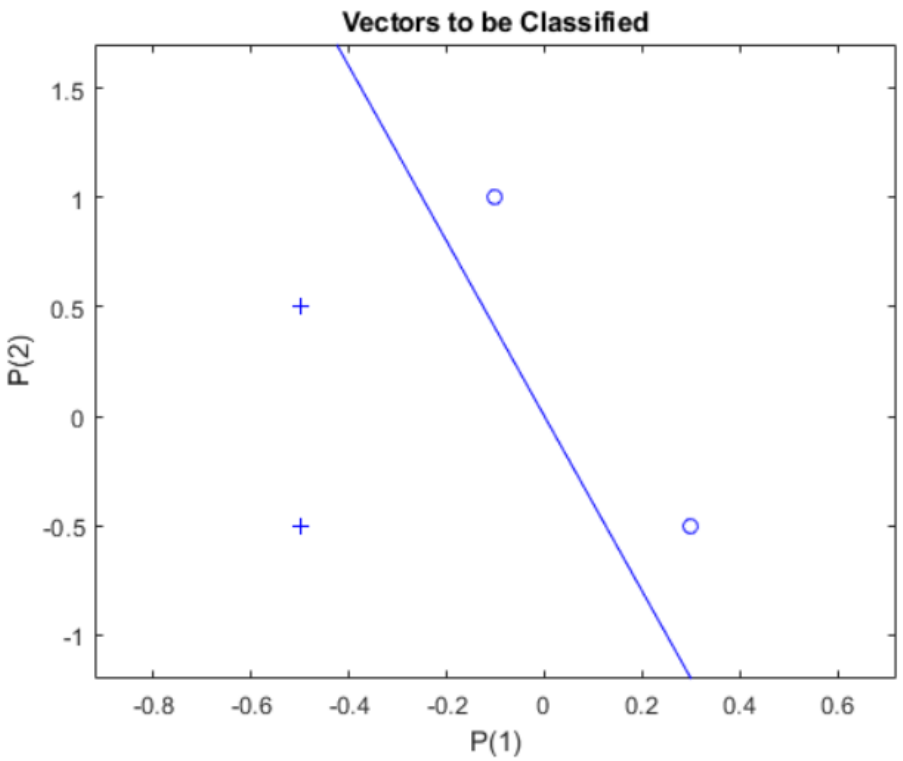
Classification with a 2-Input Perceptron MATLAB

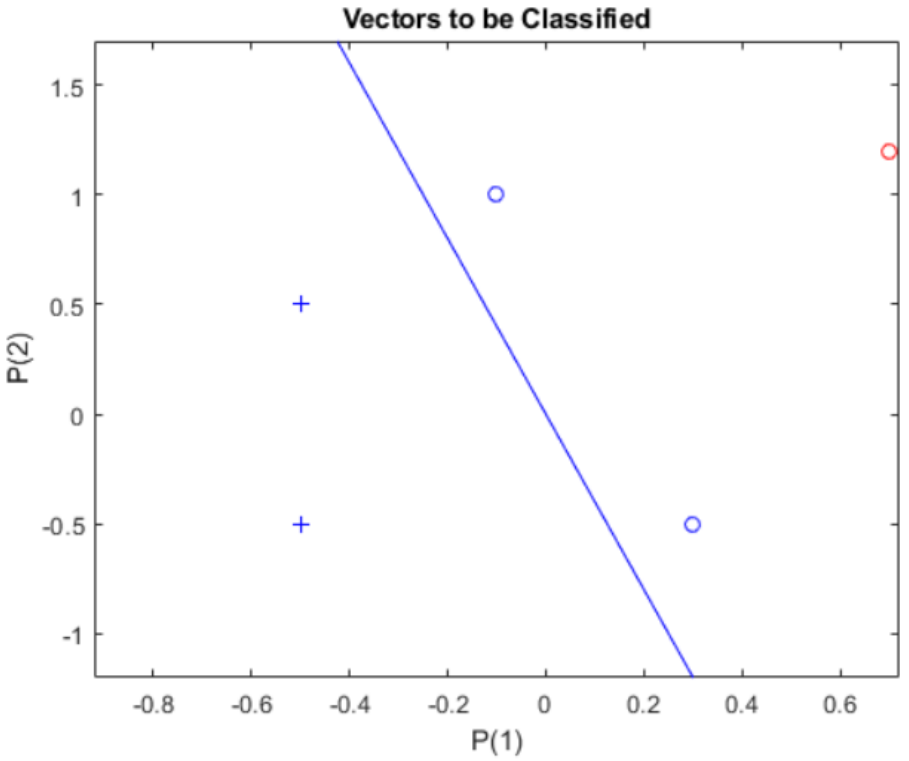
```
X = [ -0.5 -0.5 +0.3 -0.1;  ...
      -0.5 +0.5 -0.5 +1.0];
T = [1 1 0 0];
plotpv(X,T); %plot data and target
net = perceptron;
net = configure(net,X,T);
plotpv(X,T);
plotpc(net.IW{1},net.b{1}); %plot separate line
)%Convert to Sequential Series and copy 3 three times
XX = repmat(con2seq(X),1,3);
TT = repmat(con2seq(T),1,3);
```

Classification with a 2-Input Perceptron MATLAB

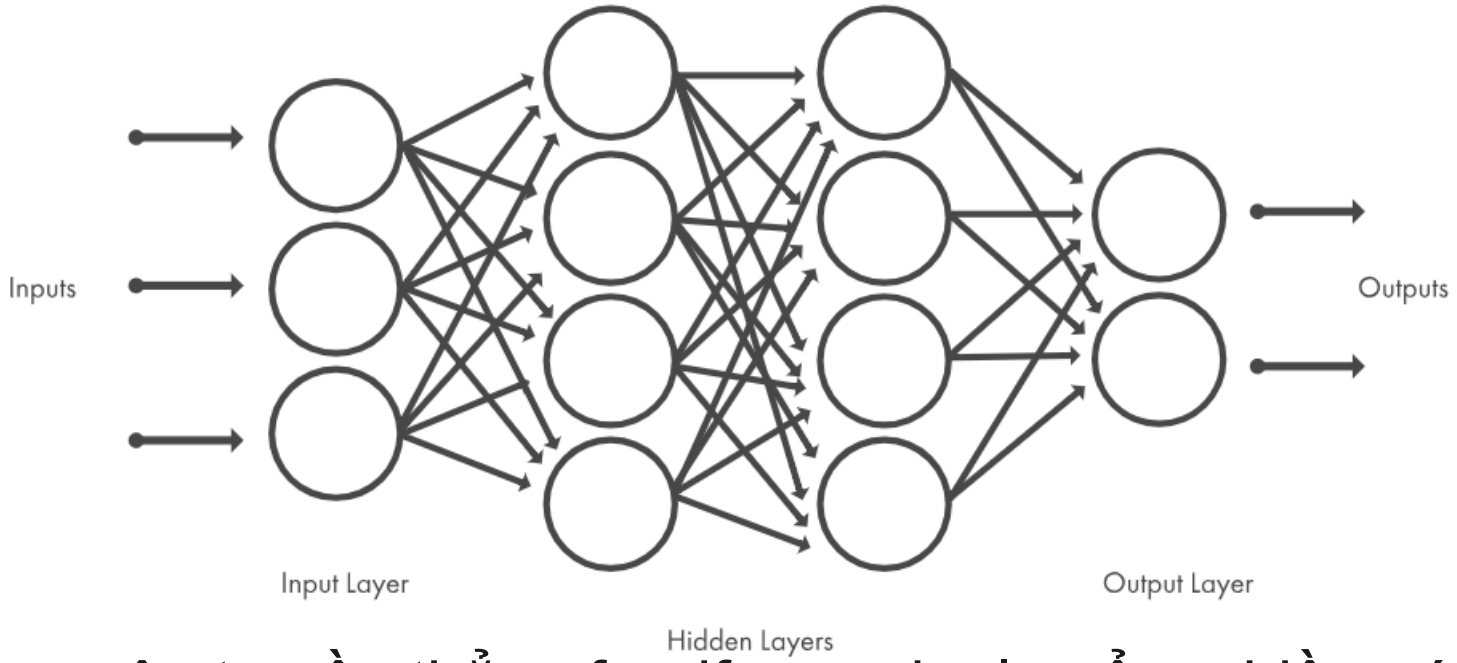
```
%Training
net = adapt(net,XX,TT);
plotpc(net.IW{1},net.b{1}); plot line
%Try new data
x = [0.7; 1.2];
y = net(x);
plotpv(x,y);
point = findobj(gca,'type','line');
point.Color = 'red';
hold on;
plotpv(X,T);
plotpc(net.IW{1},net.b{1});
hold off;
```

Classification with a 2-Input Perceptron MATLAB





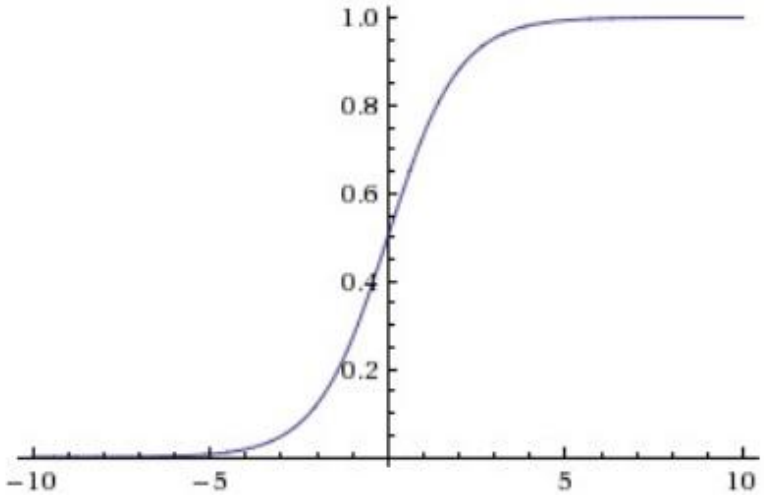
NEURAL NETWORK MLP
MULTI LAYER PERCEPTRON

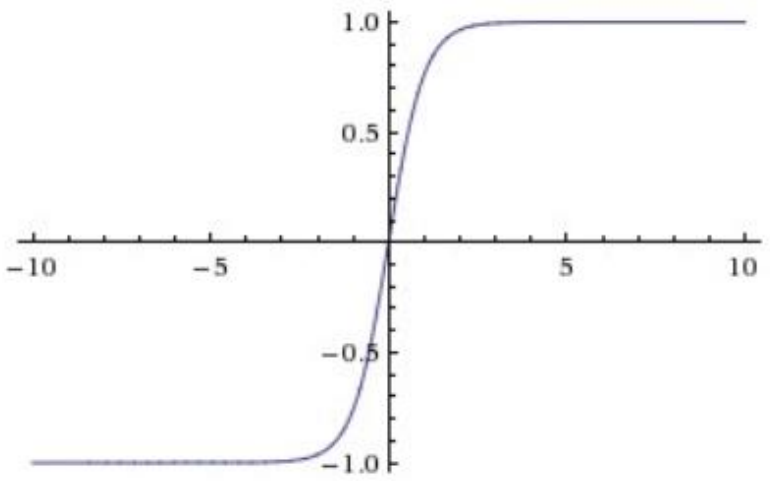


Mạng nơrôn truyền thẳng feedforward mlp gồm nhiều nút nơrôn kết nối với nhau qua các trọng số và chia thành nhiều lớp: lớp vào, các lớp ẩn, lớp ra. Mỗi nút có hàm kích hoạt vi phân được

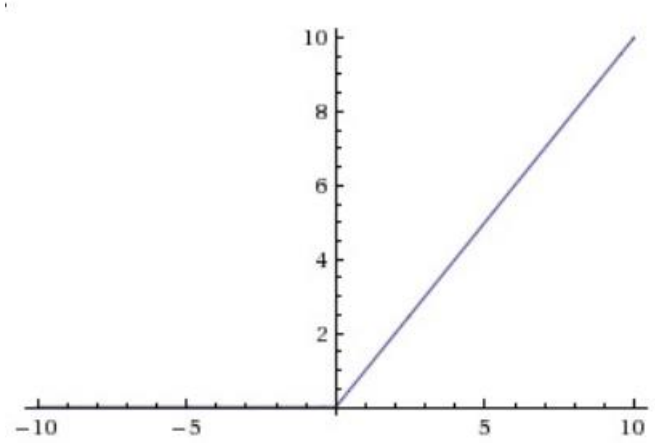
MLP sử dụng trong phân nhiều lớp hay hồi qui phi tuyến

Thuật toán huấn luyện MLP là Lan Truyền Ngược



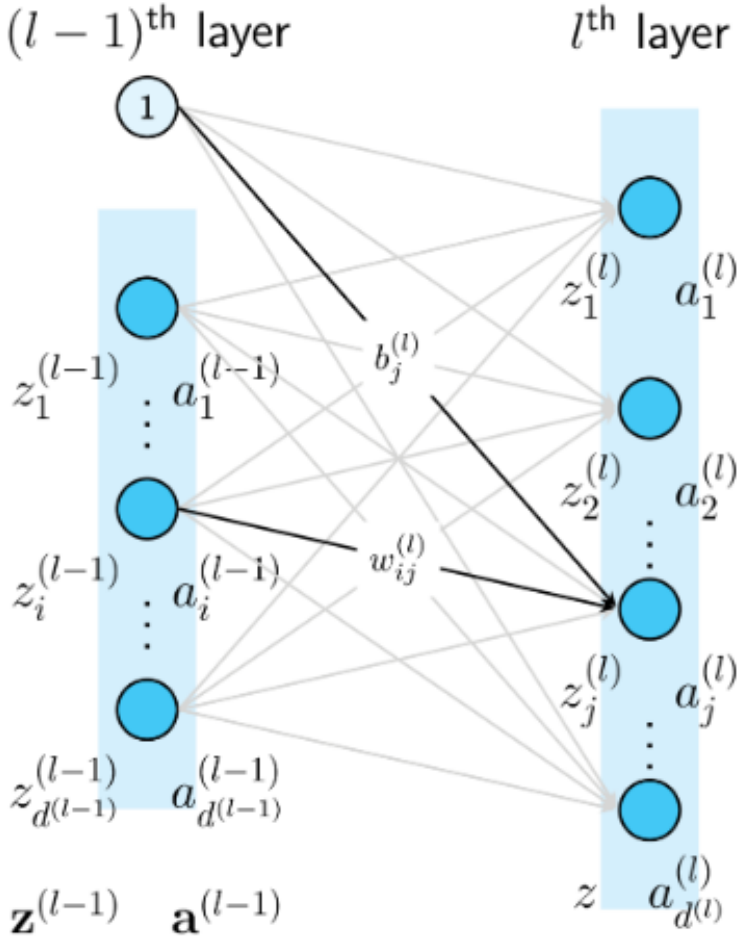


Hàm kích hoạt sigmoid và tanh



Hàm ReLU
rectified linear unit

NEURAL NETWORK MLP
MULTI LAYER PERCEPTRON



$$\mathbf{W}^{(l)} \in \mathbb{R}^{d^{(l-1)} \times d^{(l)}}$$

$$\mathbf{b}^{(l)} \in \mathbb{R}^{d^{(l)} \times 1}$$

$$z_j^{(l)} = \mathbf{w}_j^{(l)T} \mathbf{a}^{(l-1)} + b_j^{(l)}$$

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)T} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$$

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)})$$

THUẬT TOÁN LAN TRUYỀN NGƯỢC BACKPROPAGATION

- **Bước thuận**: Với ma trận trọng số và bias, ngõ vào x ta tính lần lượt các ngõ ra mỗi lớp cho đến lớp ra là yhat

$$\mathbf{a}^{(0)} = \mathbf{x}$$
$$z_i^{(l)} = \mathbf{w}_i^{(l)T}\mathbf{a}^{(l-1)} + b_i^{(l)}$$
$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)T}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}, \quad l = 1, 2, \ldots, L$$
$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)}), \quad l = 1, 2, \ldots, L$$
$$\hat{\mathbf{y}} = \mathbf{a}^{(L)}$$

- Tính hàm tổn thất

$$J(\mathbf{W}, \mathbf{b}, \mathbf{X}, \mathbf{Y}) = \frac{1}{N}\sum_{n=1}^{N} ||\mathbf{y}_n - \hat{\mathbf{y}}_n||_2^2$$
$$= \frac{1}{N}\sum_{n=1}^{N} ||\mathbf{y}_n - \mathbf{a}_n^{(L)}||_2^2$$

THUẬT TOÁN LAN TRUYỀN NGƯỢC BACKPROPAGATION

- **Bước ngược**: tìm đạo hàm của hàm tổn thất theo **W** và **b** sao cho cực tiểu J, cập nhật **W** và **b** theo phương pháp gradient descent , tính từ sau ra trước

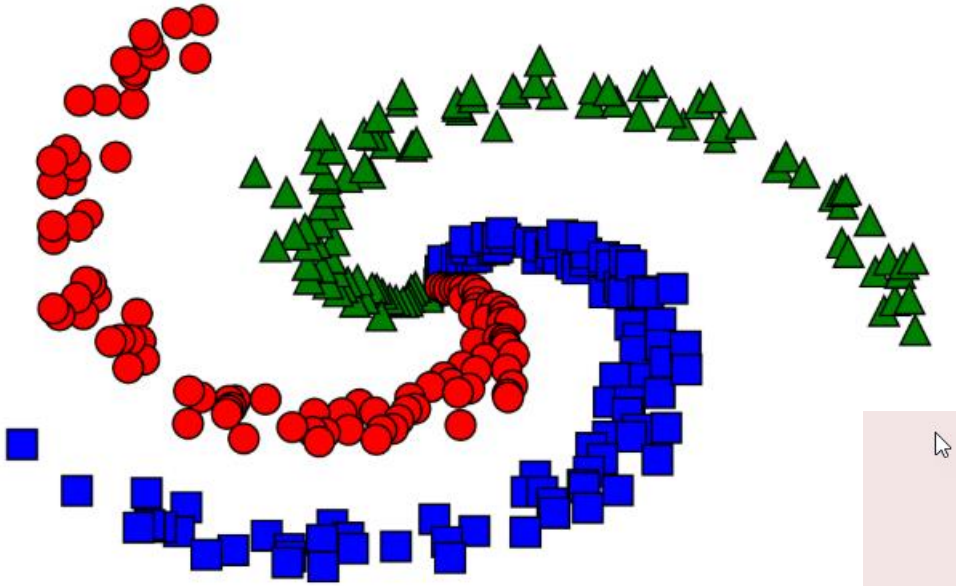
$$\theta = \theta - \eta \nabla_\theta J(\theta)$$

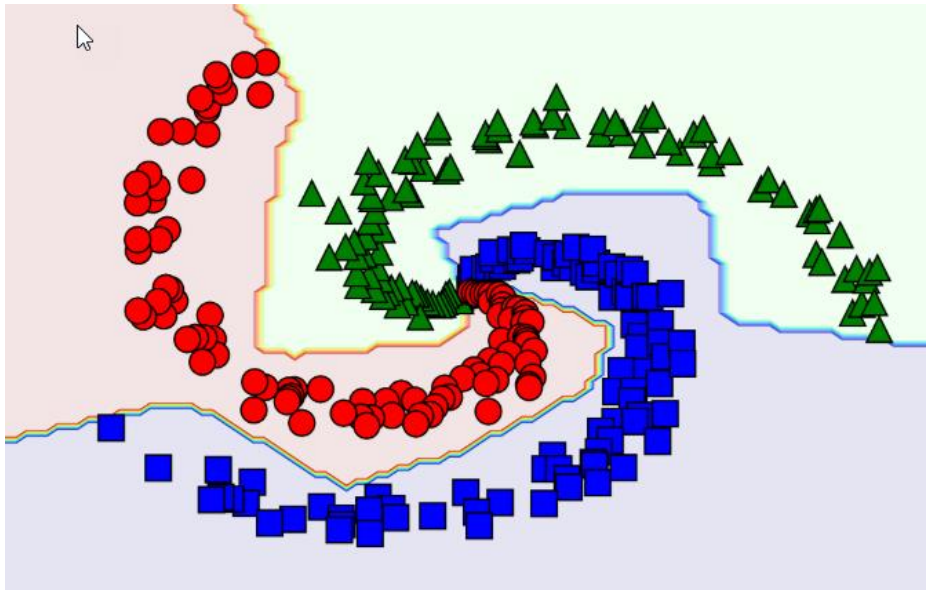
$$\frac{\partial J}{\partial w_{ij}^{(l)}} = \frac{\partial J}{\partial z_j^{(l)}} \cdot \frac{\partial z_j^{(l)}}{\partial w_{ij}^{(l)}} = e_j^{(l)} a_i^{(l-1)}$$

$$\frac{\partial J}{\partial b_j^{(l)}} = e_j^{(l)}$$

$$e_j^{(l)} = \frac{\partial J}{\partial z_j^{(l)}} = \frac{\partial J}{\partial a_j^{(l)}} \cdot \frac{\partial a_j^{(l)}}{\partial z_j^{(l)}} = \left(\sum_{k=1}^{d^{(l+1)}} \frac{\partial J}{\partial z_k^{(l+1)}} \cdot \frac{\partial z_k^{(l+1)}}{\partial a_j^{(l)}} \right) f'(z_j^{(l)}) = \left(\sum_{k=1}^{d^{(l+1)}} e_k^{(l+1)} w_{jk}^{(l+1)} \right) f'(z_j^{(l)}) = \left(\mathbf{w}_{j:}^{(l+1)} \mathbf{e}^{(l+1)} \right) f'(z_j^{(l)})$$

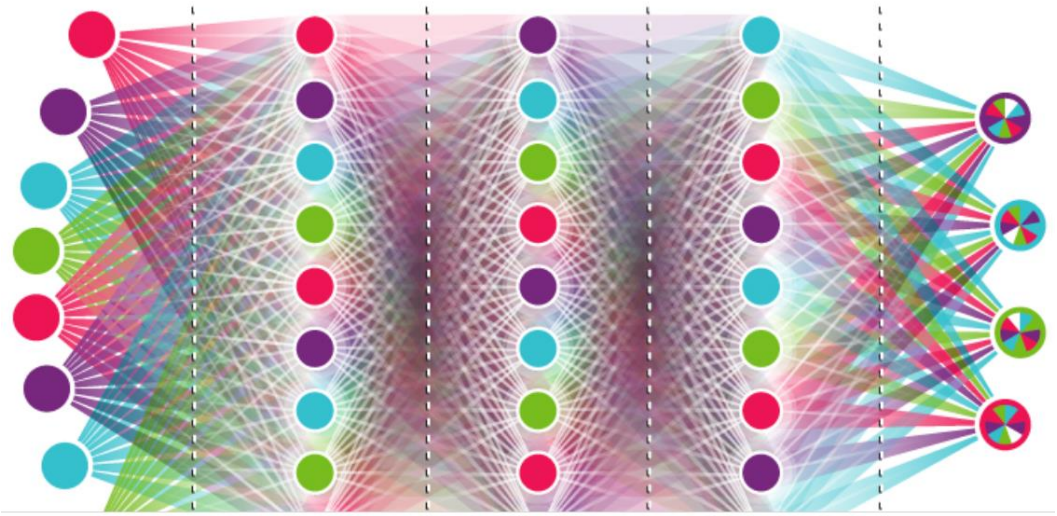
NEURAL NETWORK PYTHON





DEEP NEURAL NETWORK DEEP LEARNING

- Mạng feed forward NN không mô phỏng đầy đủ não bộ con người nên kém thông minh
- Deep neural network có cấu trúc phức tạp hơn CNN convolutional NN, RNN Recurrent NN
- Deep Learning đã đạt được nhiều thành tưu trong nhận dạng hình ảnh xử lý tiếng nói, dịch ngôn ngữ nói, trí tuê nhân tạo



HỒI QUI TUYẾN TÍNH

- Cho tập dữ liệu (xi,yi)

K MEANS CLUSTERING