AI, Machine Learning, Deep Learning

Nguyen Xuan Hoai, PhD
AI Academy Vietnam

Contents

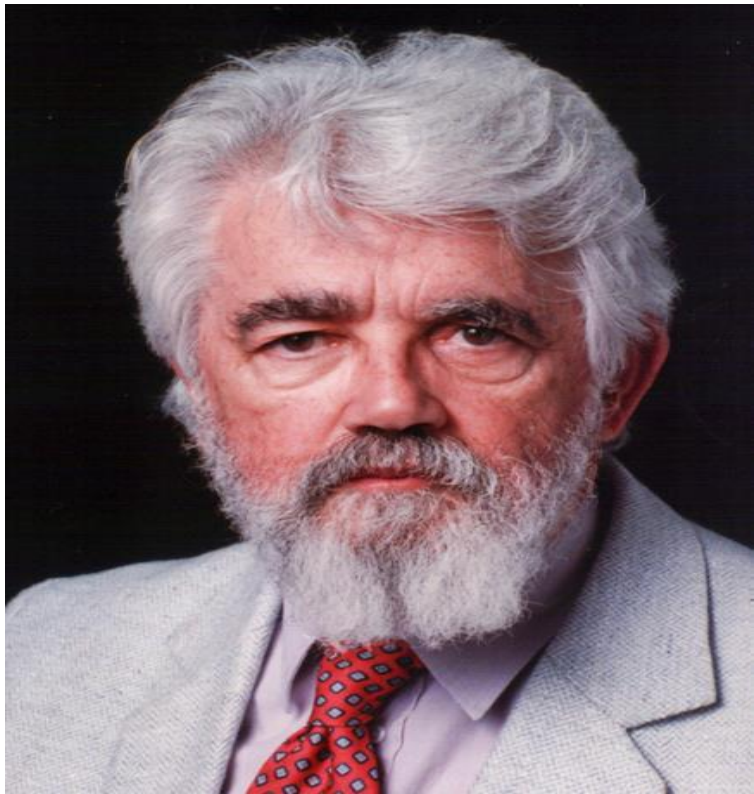
- Introduction to AI and machine learning
- Machine learning problems.
- Building machine learning models.
- Training and evaluation of machine learning models.
- Neural networks and deep learning.
- Machine learning libraries.

Introduction to Machine Learning

What is AI?

- John McCarthy coined the term “Artificial Intelligence” in 1955.
- Not easy to give a concise and unique definition of AI
- Two schools of though on AI :

Strong (general) AI vs. Weak (specific) AI.

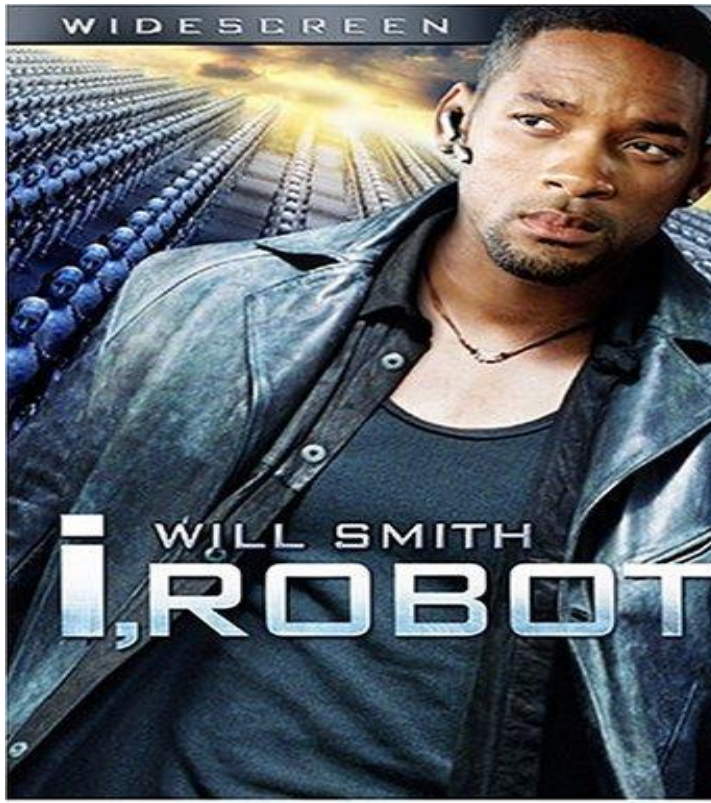


Strong AI

Strong AI: Robot and AI could become a new race (human-being, self-aware).

- Machine can think, reason, fully conscious.
- Expected by some AI researchers, Futurists,

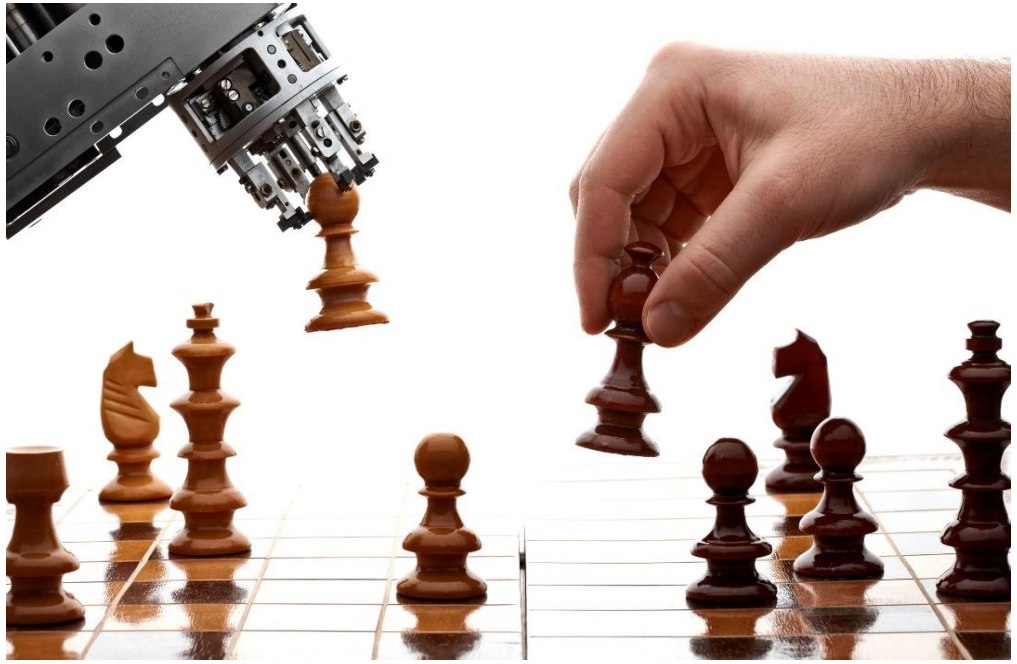
…. and Hollywood!

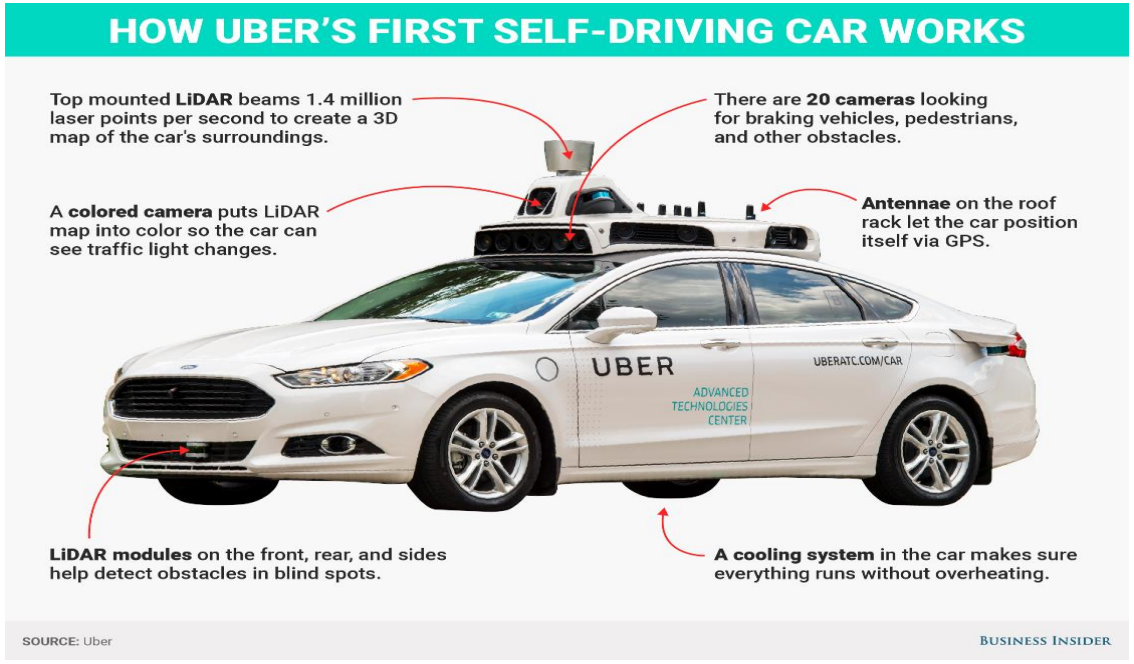


Weak AI

Weak AI: AI could mimic some intelligent behaviors of human.

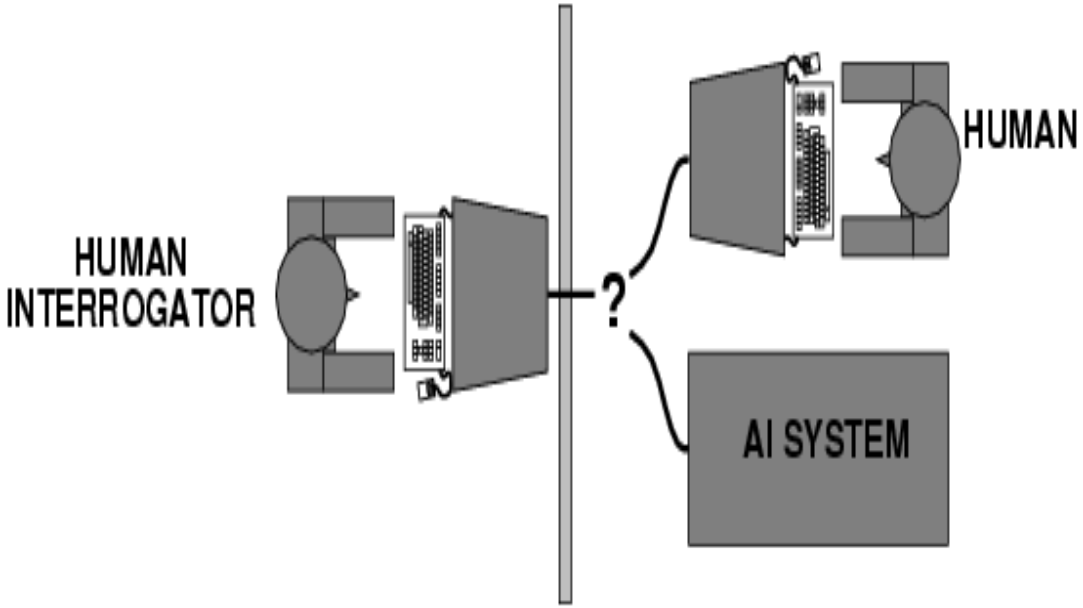
- Accepted by most AI researchers
- Often be confused with Strong AI.

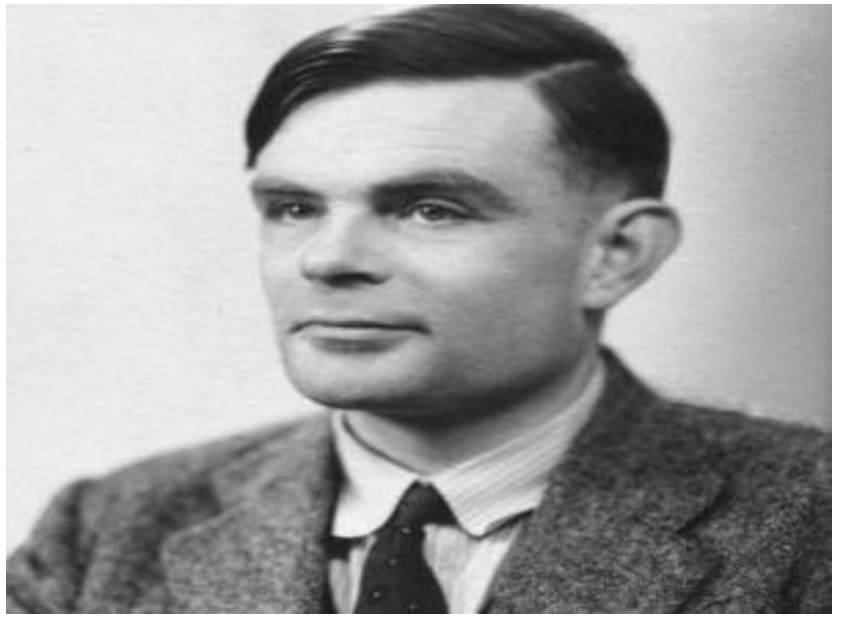




Turing Test

- A. M. Turing (1950) *Computing Machinery and Intelligence*, Mind 49: 433-460.

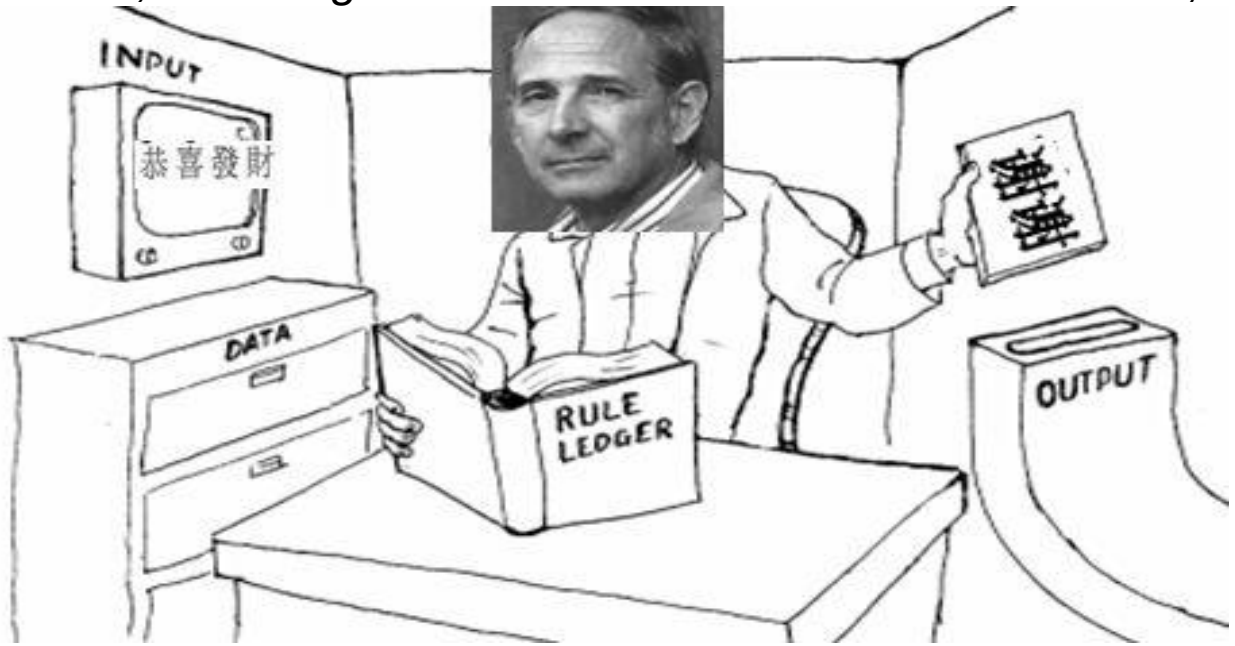




- Predicted by 2000, AI machine could have 30% chance to deceive human in 5 minutes.
- Influential and controversial for AI field in the next 50 years.
- Suggest components of intelligence: knowledge), reasoning, language understanding, learning, …

Chinese Room Argument

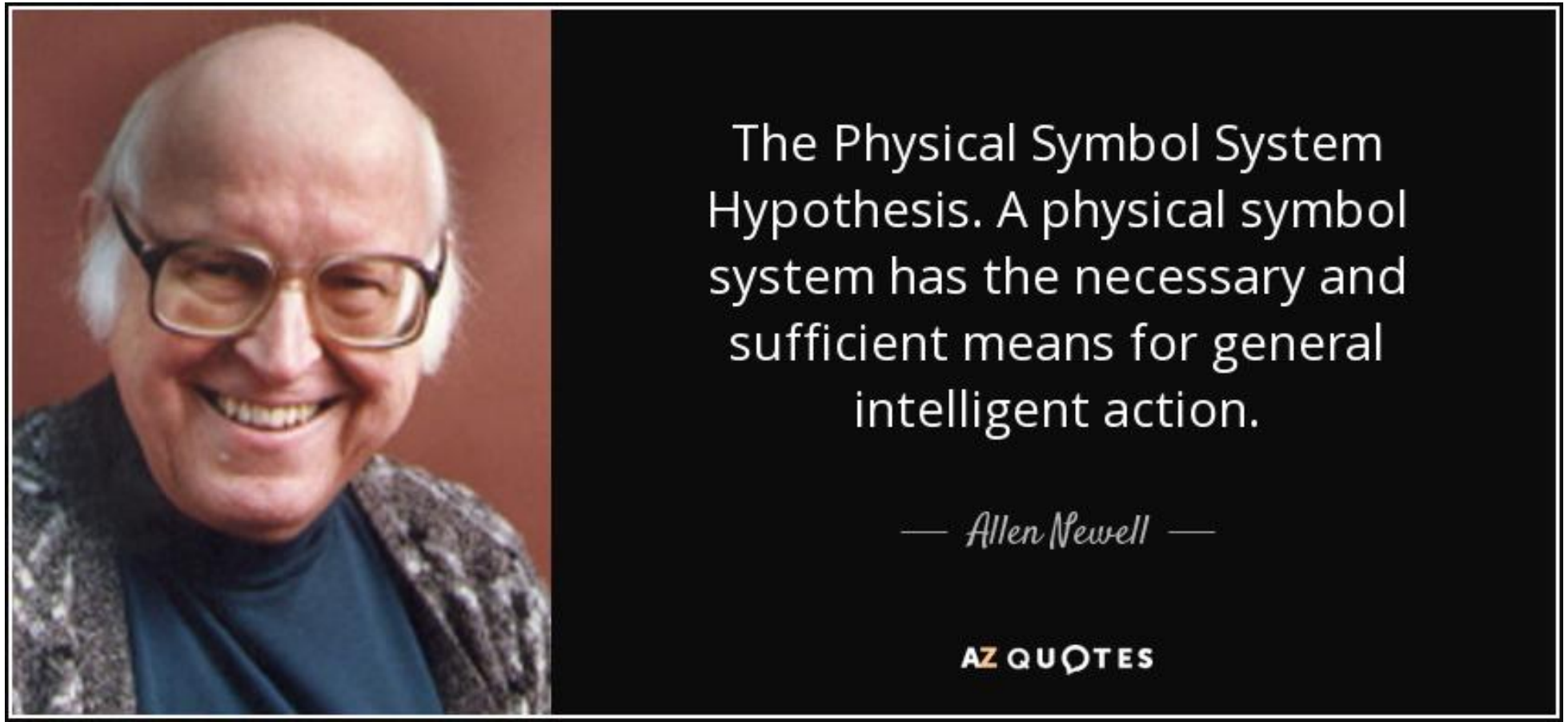
Searle, John. *Minds, Brains, and Programs*. Behavioral and Brain Sciences 3, 1980, pp. 417-424.



- AI programmed machines could not really have thinking or consciousness but only imitate human, even if they pass the Turing test.

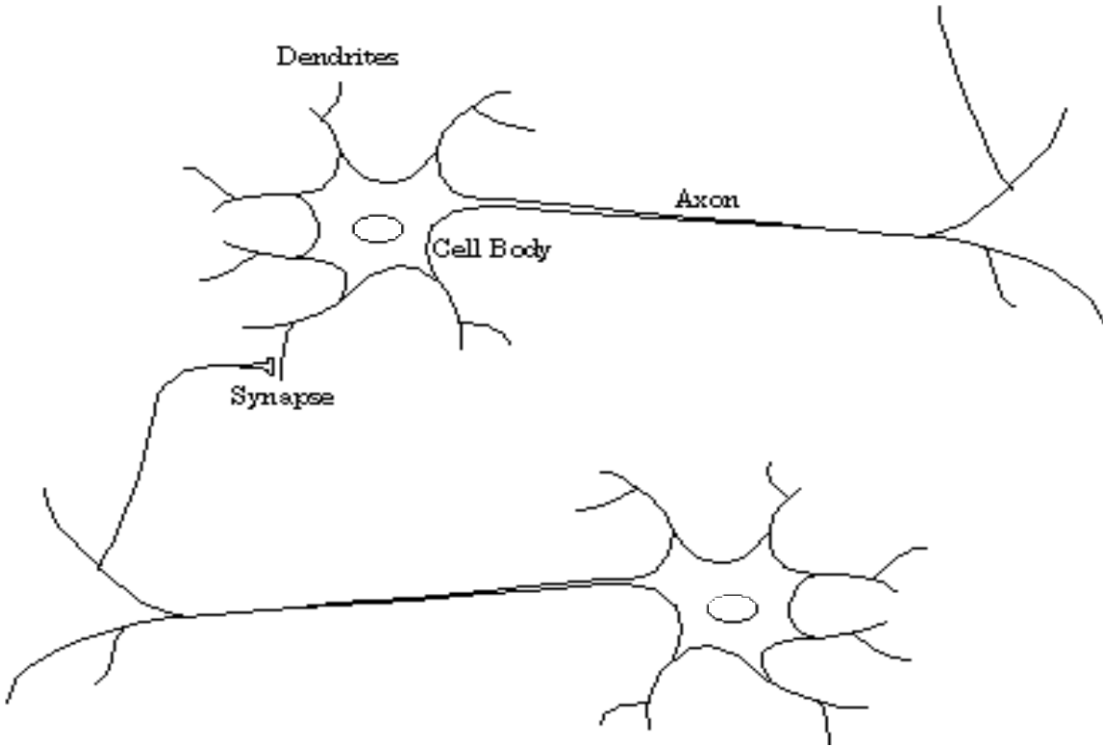
Paradigms for Building AI

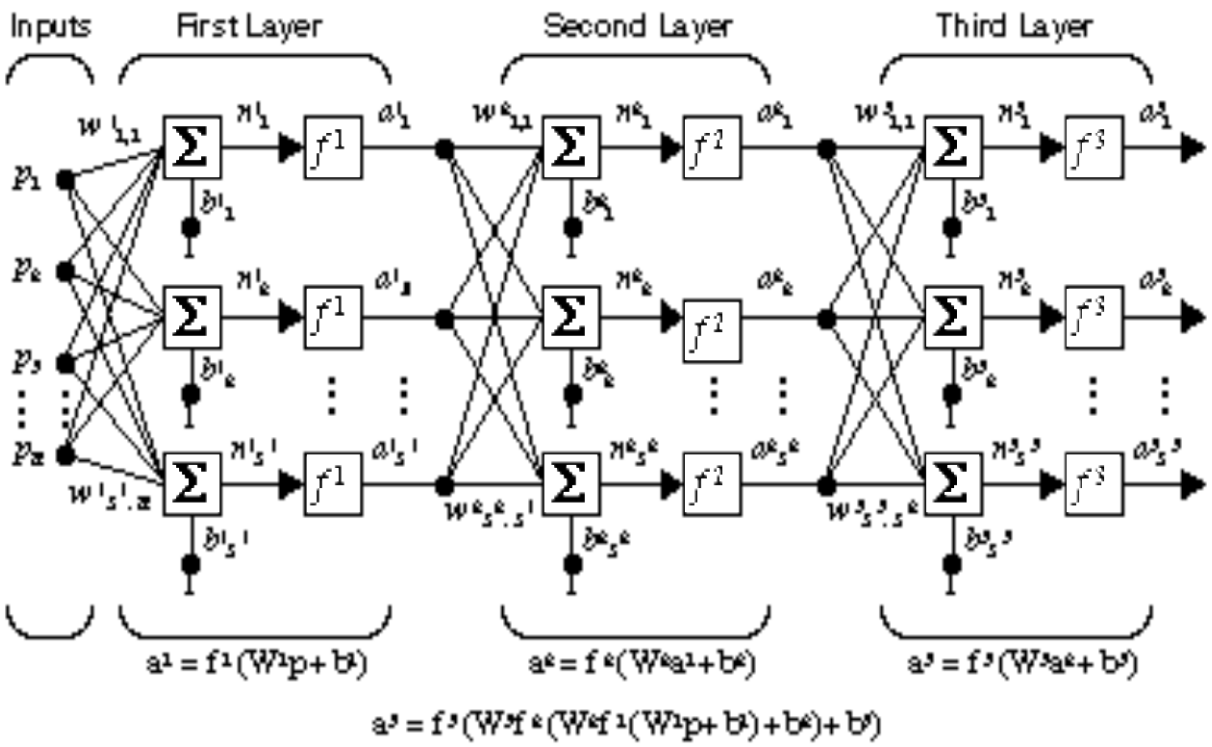
- Top-down: Cognitive science → Symbolism (Allen Newell and Herbert A. Simon) Symbolic representation, Symbolic Computation.
  - General Problem Solver, Planner, Knowledge Based Systems, Expert Systems ...



Paradigms for Building AI

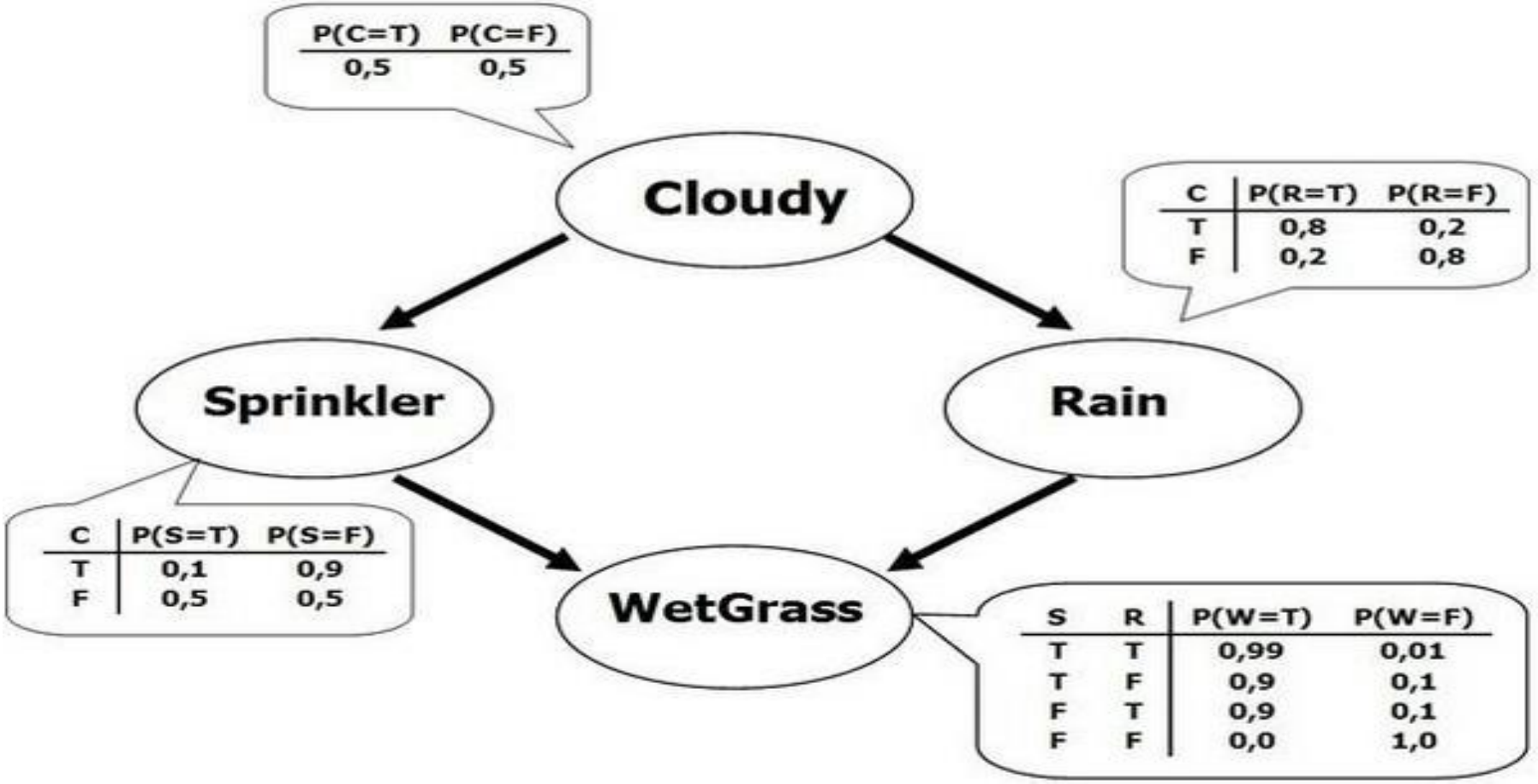
- Bottom-up: Neural and Brain Science → Connectionism (Sub-symbolic)
  - Neural Networks, Deep Learning





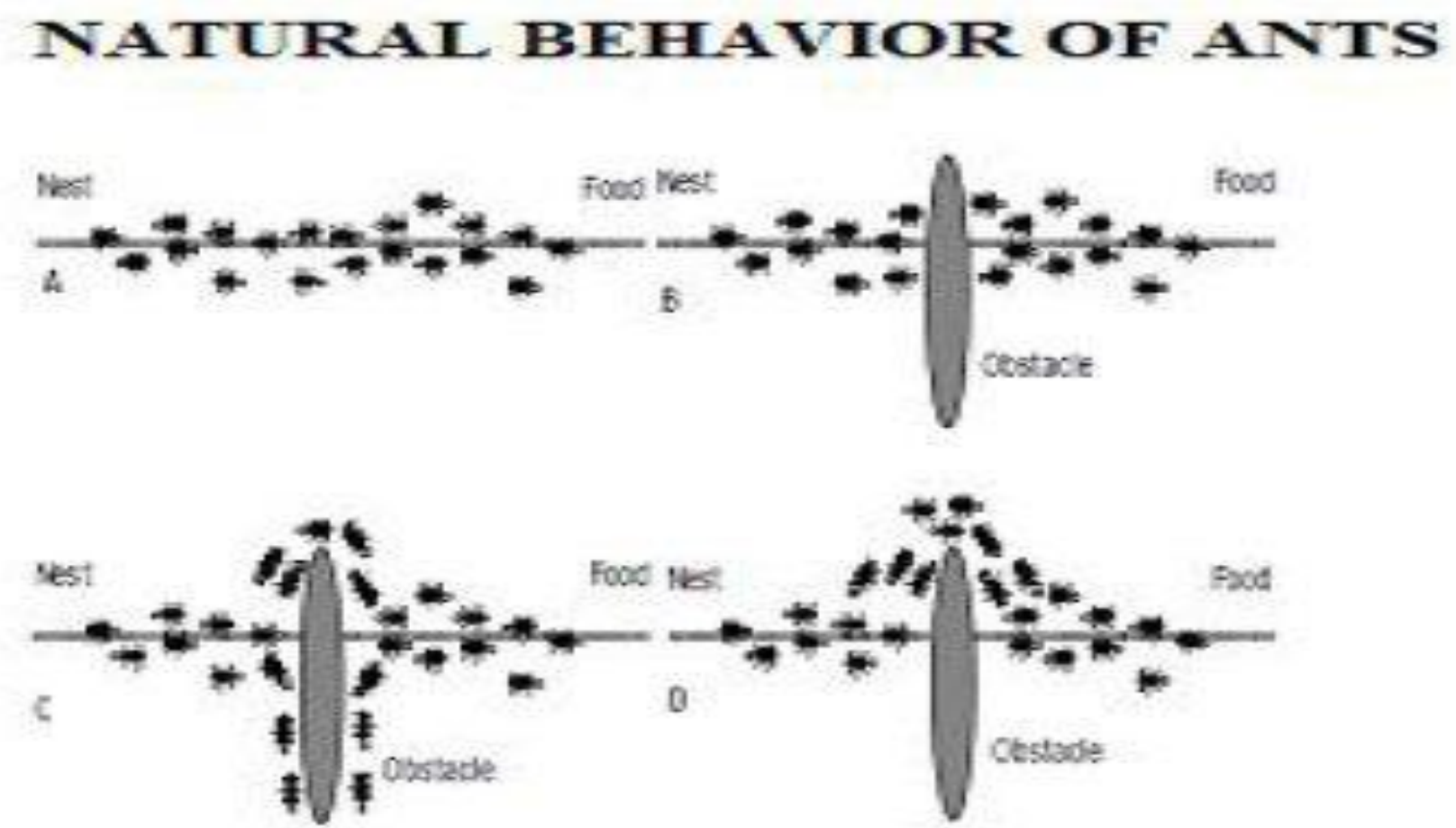
Paradigms for Building AI

- Uncertain AI: handle uncertainty information/ data (Probabilistic AI, Fuzzy Based AI)
  - Statistical techniques, Graphical Models, Fuzzy Logic, …



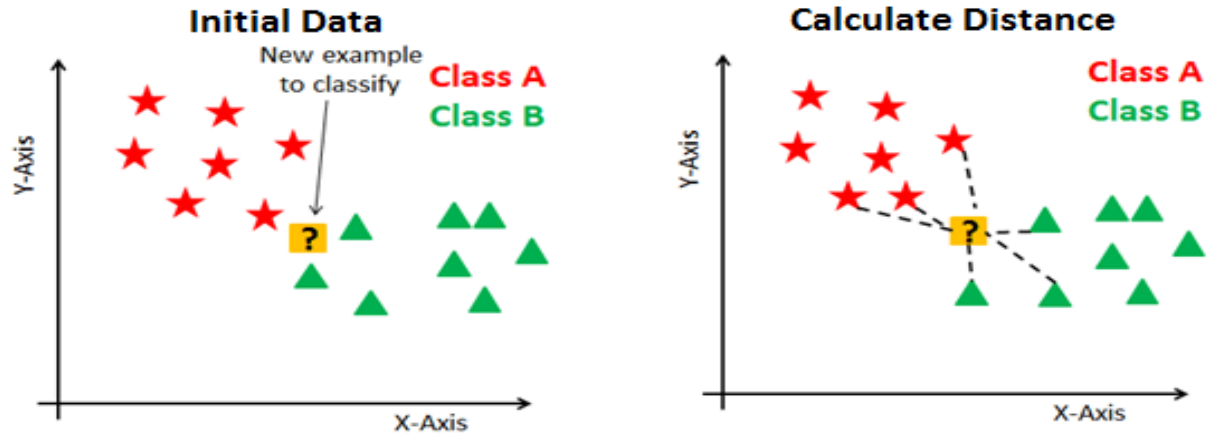
Paradigms for Building AI

- Simulation Based AI: Based on natural adaptation
  - Evolutionary Computation, Swarm Intelligence.



Paradigms for Building AI

- Similarity Based (lazy) AI: learning by imitation based on past knowledge and data.
  - Case-based reasoning, K-nearest neighbors, SVM, ….



Past AI Achievements

- MYCIN (1984, Standford).
- Proved a mathematical conjecture (Robbins conjecture) unsolved for decades.
- During the 1991 Gulf War, US forces deployed an AI logistics planning and scheduling program that involved up to 50,000 vehicles, cargo, and people
- Deep Blue defeated the reigning world chess champion Garry Kasparov in 1997.
- Gulf War 2 (2003), Artificial War.
- NASA's on-board autonomous planning program controlled the scheduling of operations for a spacecraft.
- Human identification through eyes detection and analysis at Heathrow airport using evolutionary computation technique.
- Washing machine generation using NeuroFuzzy Technology, Robotic Vacuum-Cleaner, …
- .....

AI Today

Weak AI could be used to assist/replace human in:

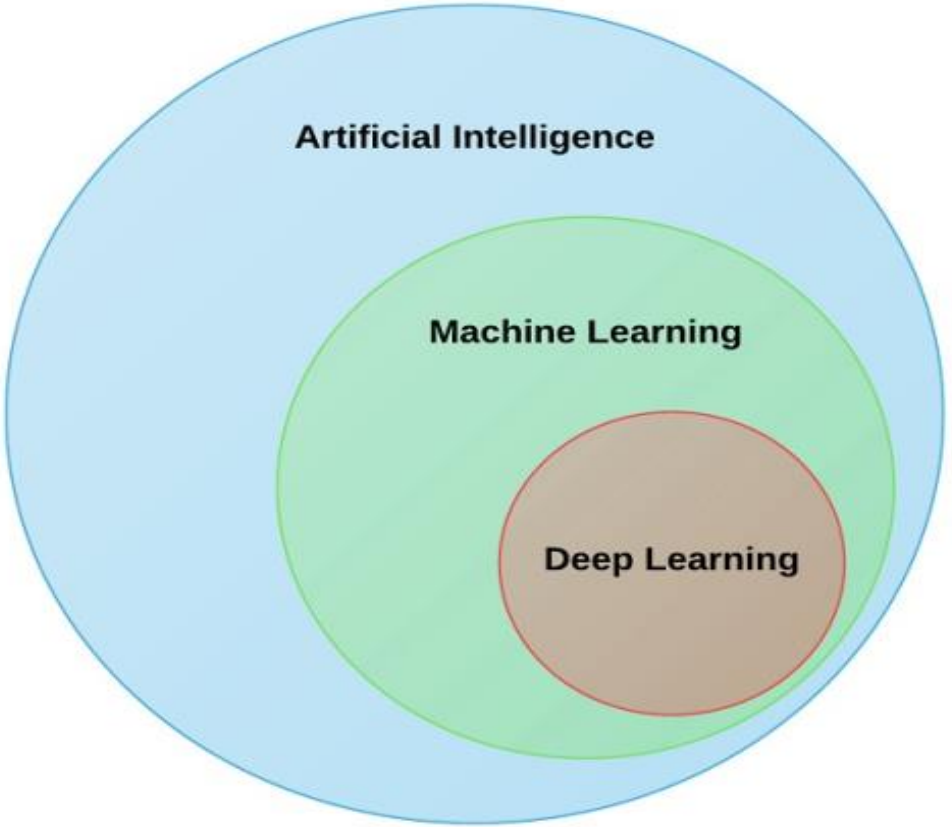
- Exacting information from big data:
  - Image, video processing (Computer vision).
  - Speech processing (Speech Recognition).
  - Natural Language Processing/ Understanding.
- Decision Making:
  - Prediction, forecasting.
  - Optimization.
  - Recommendation.

AI Today

- Many Deep Learning Based AI Systems have surpassed human capabilities in solving some certain tasks:
  - Computer Vision, Pattern Recognition, Speech Recognition, Machine Translation, Game Playing, ….
- AI have been applied in all spheres of life
  - AI grant projects (Google Brain, OpenAI, …)
- AI transformed many fields and big companies (Google Facebook, Amazon, …) và AI has become a new industry!

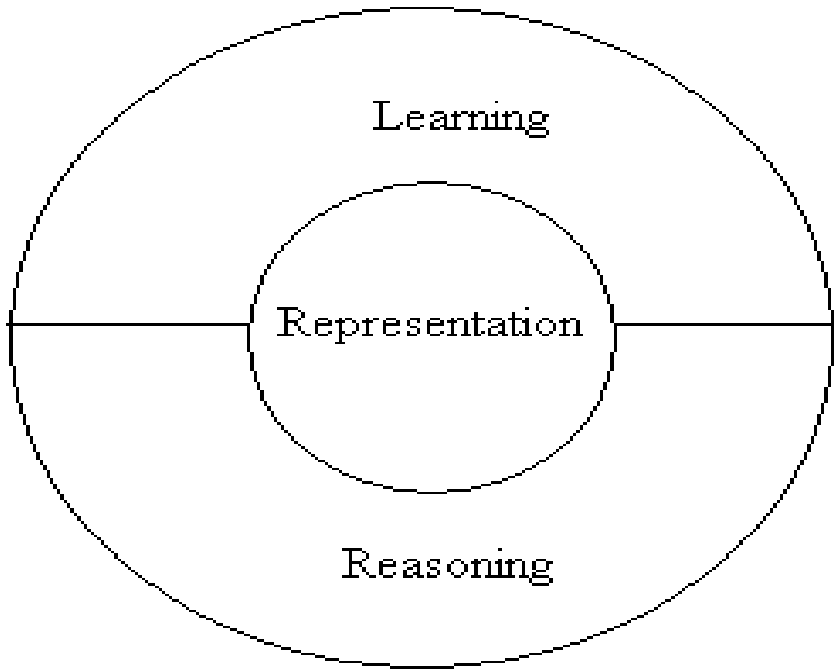
Subfields of AI

- Machine Learning.
- Data Science, Mining and Knowledge Discovery.
- Computer Vision.
- Natural Language Processing.
- Speech Recognition.
- Evolutionary and Natural Computation.
- Fuzzy Computation and Technologies.
- Artificial Life.
- Knowledge-Based Systems.
- Automated Reasoning.
- Logic and Constraint Programming.
- Intelligent Planning.
- .....

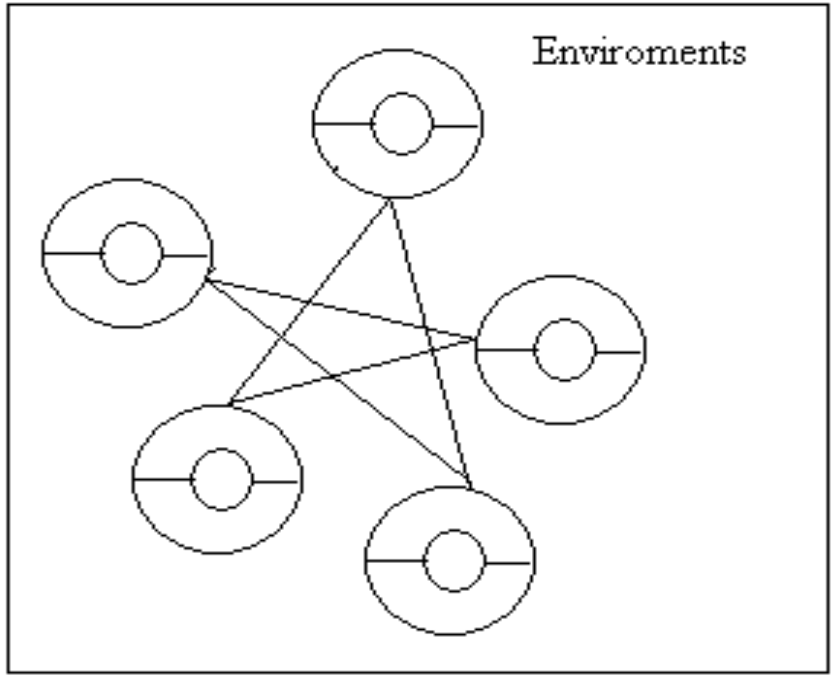


Core components of AI

- *Representation*.
- *Reasoning*.
- *Learning*.
- *Interaction*.



One Agent



Multi-Agents

What is machine learning?

Definition of machine learning (T. Mitchell): A computer program is said to learn from experience E for solving a task T, measured by performance measure P iff it could solve T better after using E (measured by P).

Example : - Learn to play chess

- Learn to detect malware
- Learning for autonomous vehicles.

Why “learn”?

- Machine to learn: allow it to solve problems based on data and experience instead of by algorithms.
- When learning is needed:
  - No prior human knowledge (Find ways on Mars),
  - Human could not explain how he solve the problem (voice recognition)
  - Solution is constantly needed (Network (re)routing)
  - For special tasks (with respect to each user)

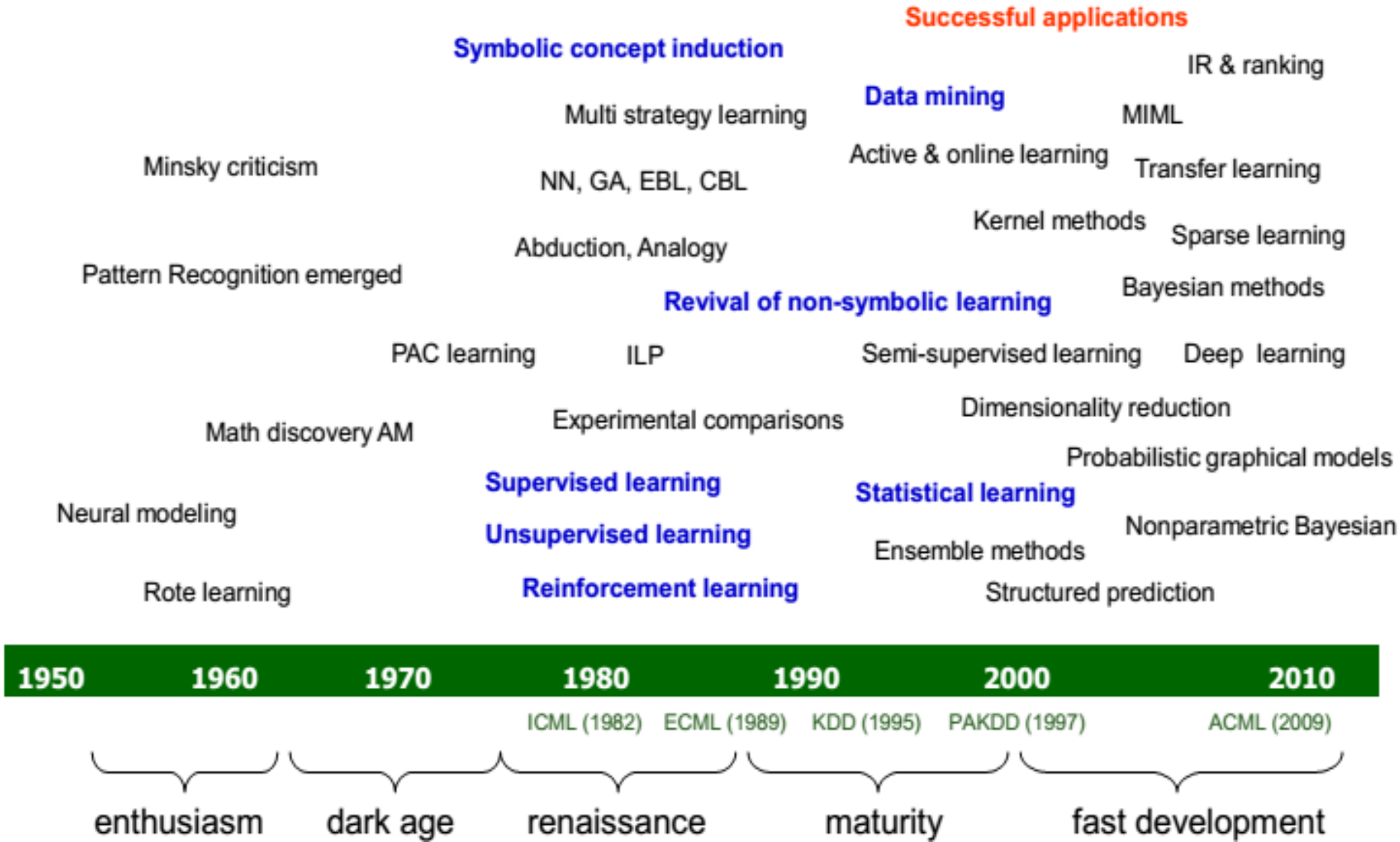
How to learn?

- Build models from data to solve problems, not by hardwired algorithms.
- Data is cheap, abundant, and available.
- Knowledge is expensive (e.g from doctors).

Example: Medical expert for reading MRI images

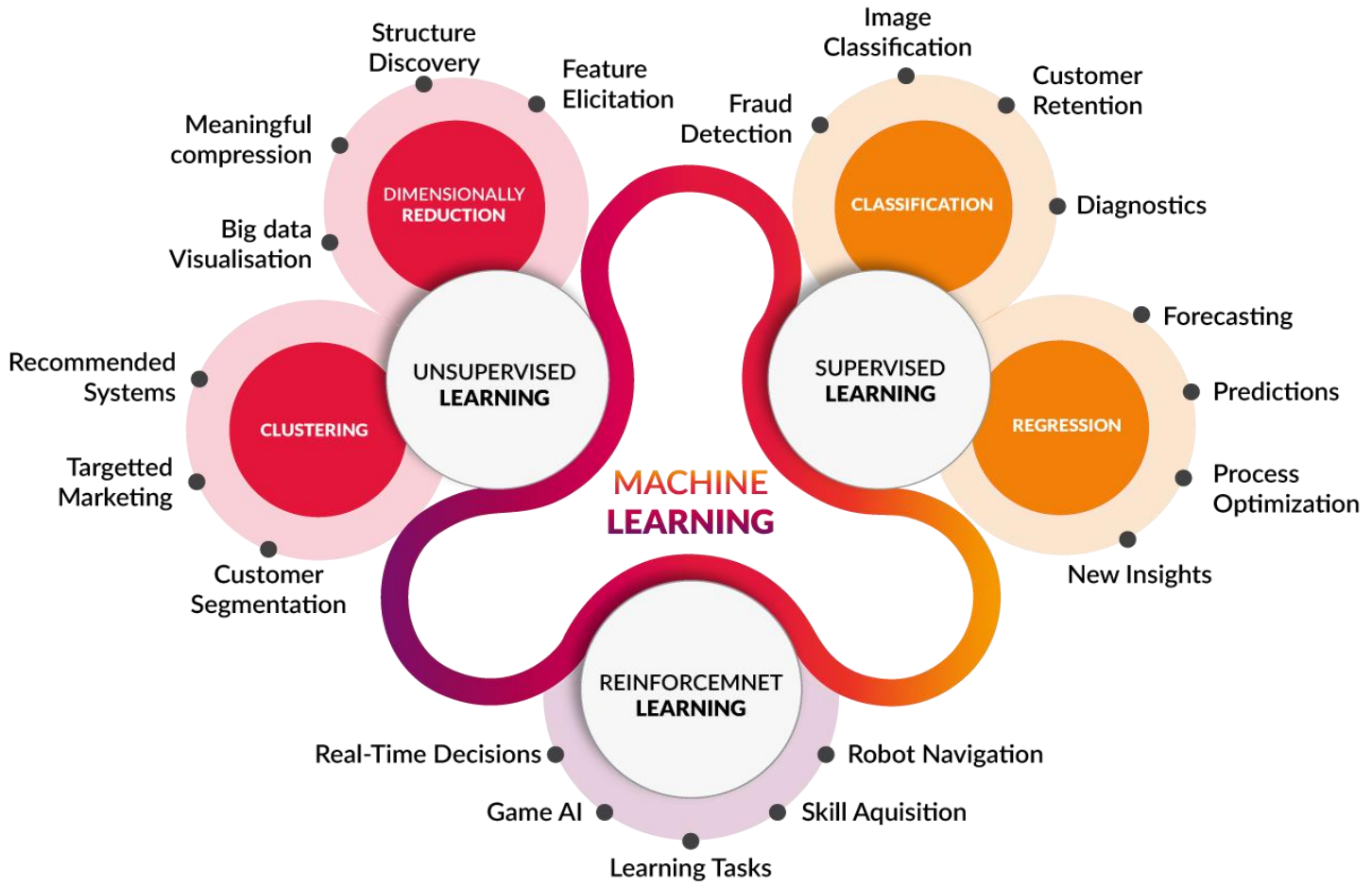
- Task: Build models from data to approximate the knowledge of the medical expert.

Timeline of ML(Pre-Deep Learning)

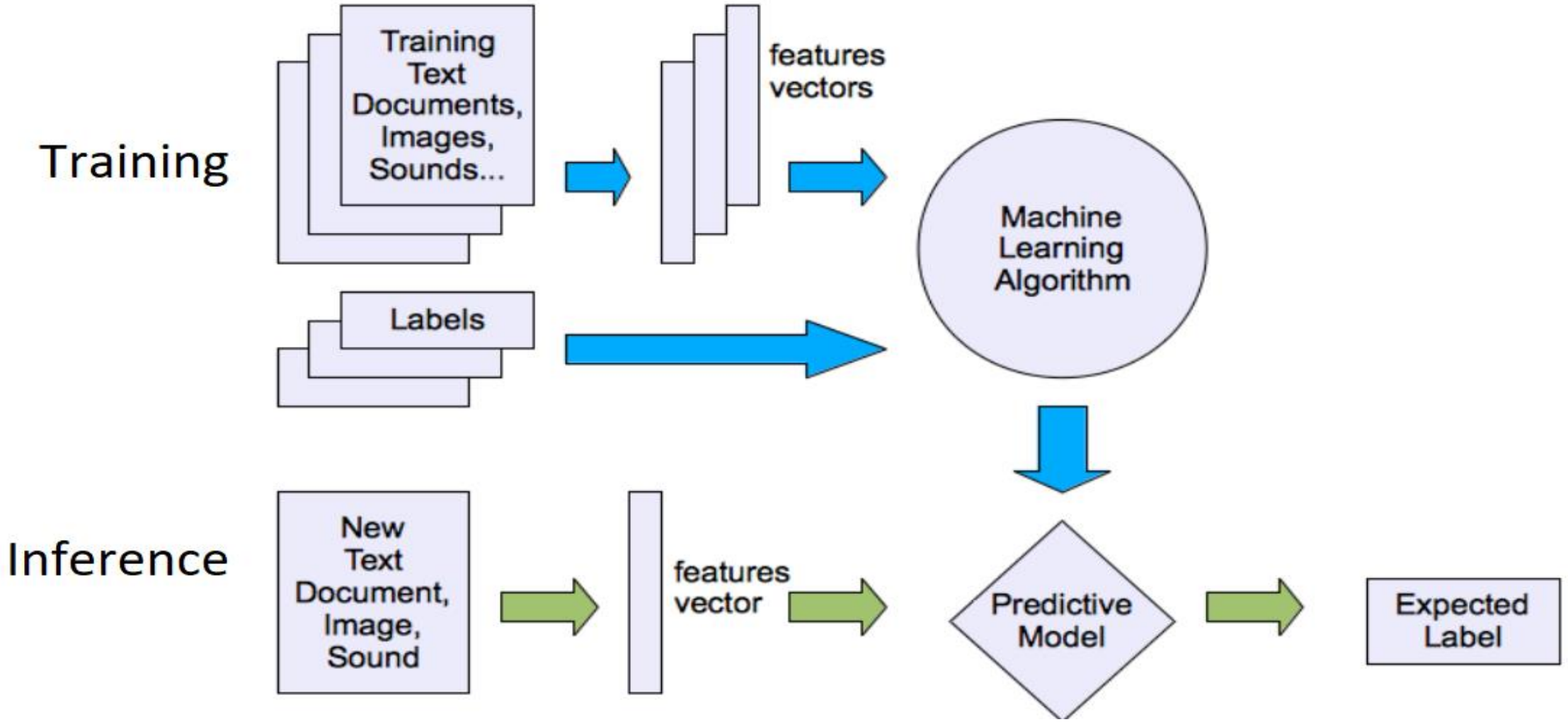


Machine learning problems

Machine Learning Problems

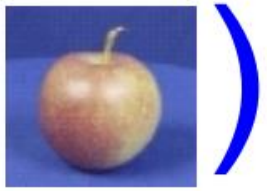


Supervised Learning



Supervised Learning

Task (T): Build a predictive function to give an output for each input.

f() = “apple”

f() = “tomato”

f() = “cow”

$$y = f(\mathbf{x})$$

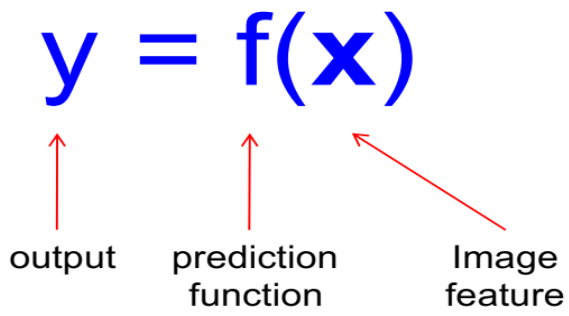
output — prediction function — Image feature

If y is discrete – Classification problem.

If y is continuous –Regression problem.

Supervised Learning

Data (E): Build a predictive function to give an output for each input.



Training data: (x_i, y_i, i=1,2, …, n) is the dataset training machine learning model. y_i is the i^{th} desired output (ground truth, labelled by human/expert).

Testing data: (x_j, y_j, j=1,2, …, m) is the dataset for testing/evaluating the performance (goodness) of the learned models.

Supervised Learning

Error Function: Measure the error of the machine learning model on training data.

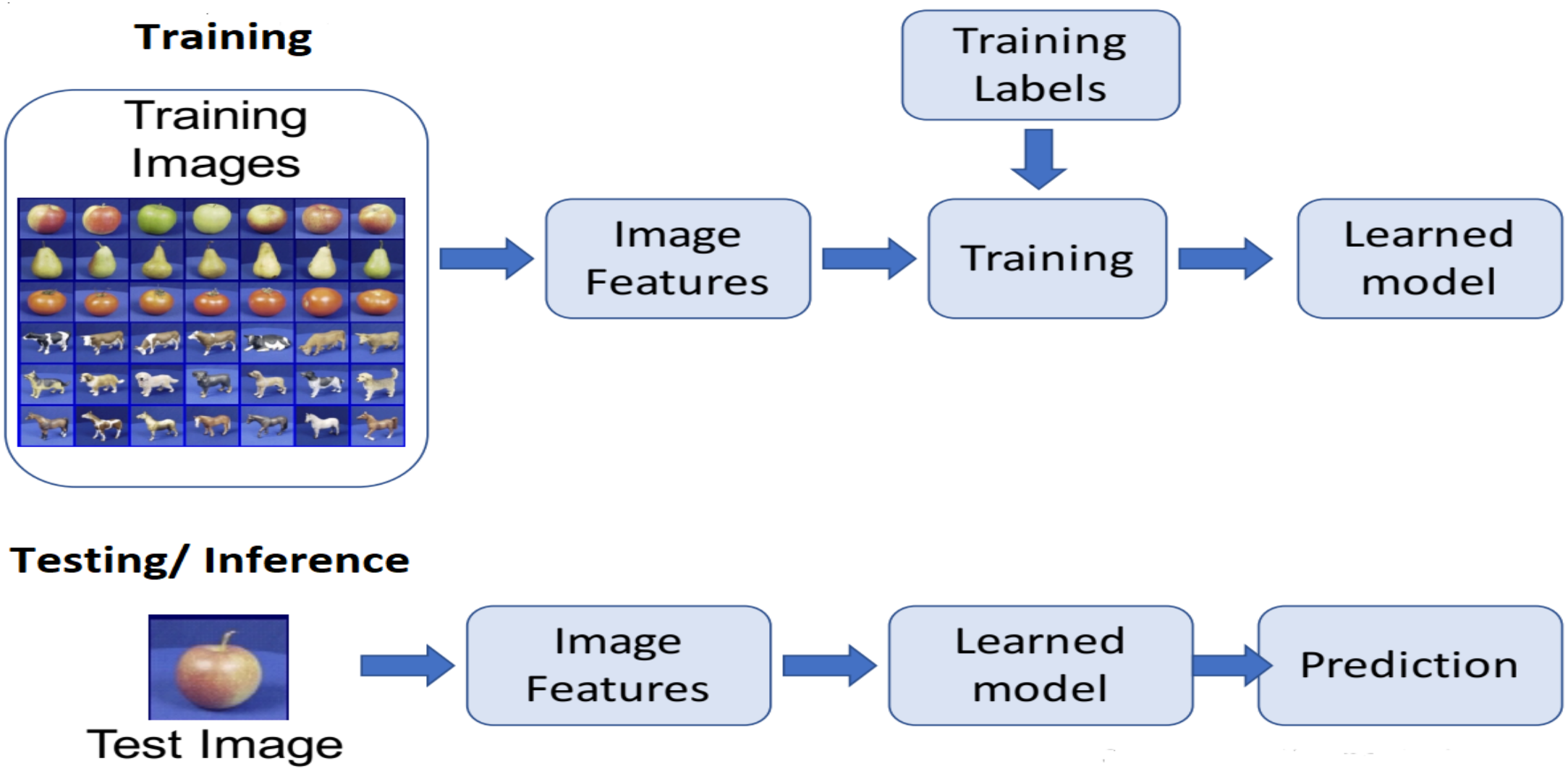
$$\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Loss Function (– P): Measure the loss of machine learning model on training data.

$$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} w_i^2$$

Training machine learning models: Minimize the loss function on training data.

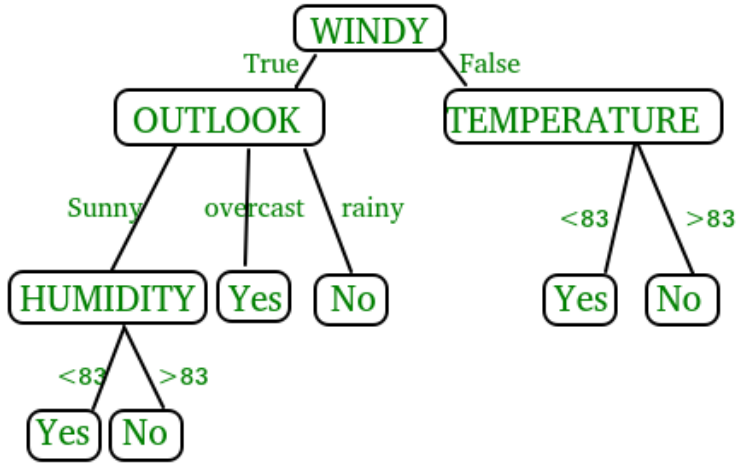
Supervised Learning



Supervised Learning

Examples for classification problems:

- Weather classification
- Malware detection for data center security.



Example for regression problems:

- Real estate pricing.
- Stock price forecasting
- Forecasting connection demand at BTSs.

Supervised Learning

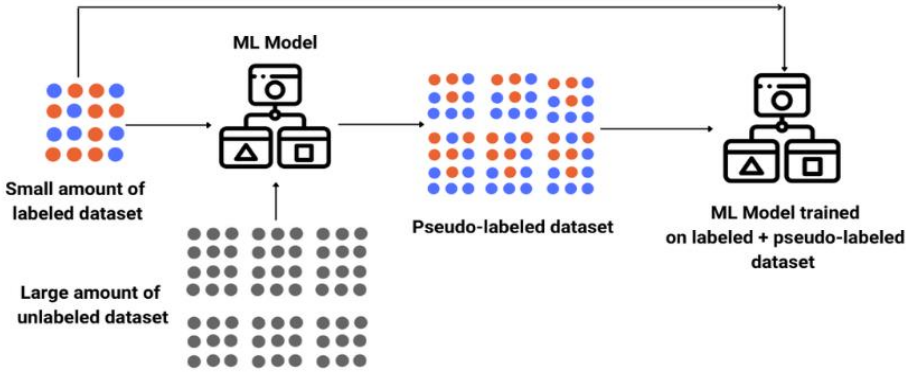
Techniques for supervised learning:

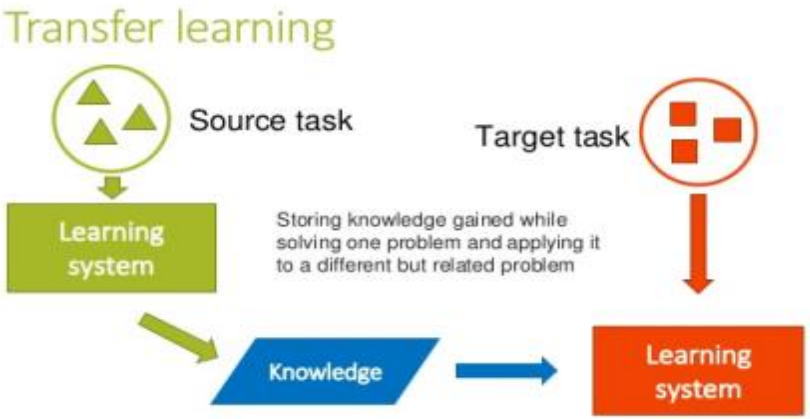
- Symbolic Machine Induction
  - Rule induction, inductive logic programming, decision tree, ….
- Statistical Machine Learning
  - Statistical regression models, Bayesian models, ….
- Connectionist Machine Learning
  - Feed forward neural networks, ….
- Fuzzy Machine Learning
  - Fuzzy systems, …
- Lazy Machine Learning
  - K-NN, SVM, ….

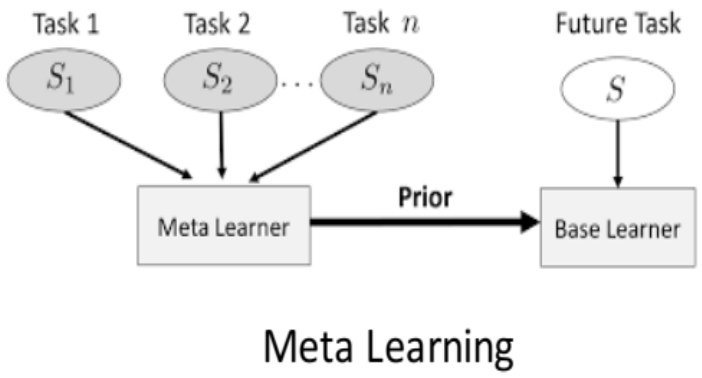
Weakly Supervised Learning

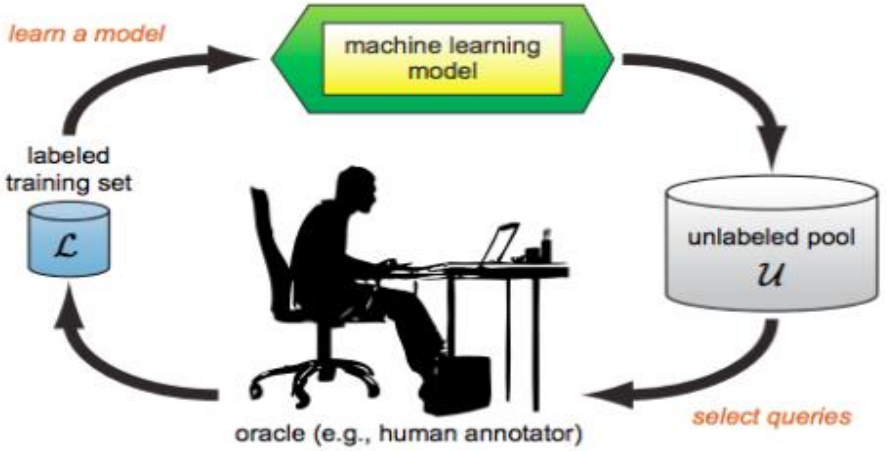
Data labelling can be very expensive. Weakly supervised learning is used to reduce the amounts of labelled data needed.

- Semi-supervised learning.
- Active Learning (Human-in-the-loop).
- Meta learning (zero shot, one-shot).
- Transfer learning.

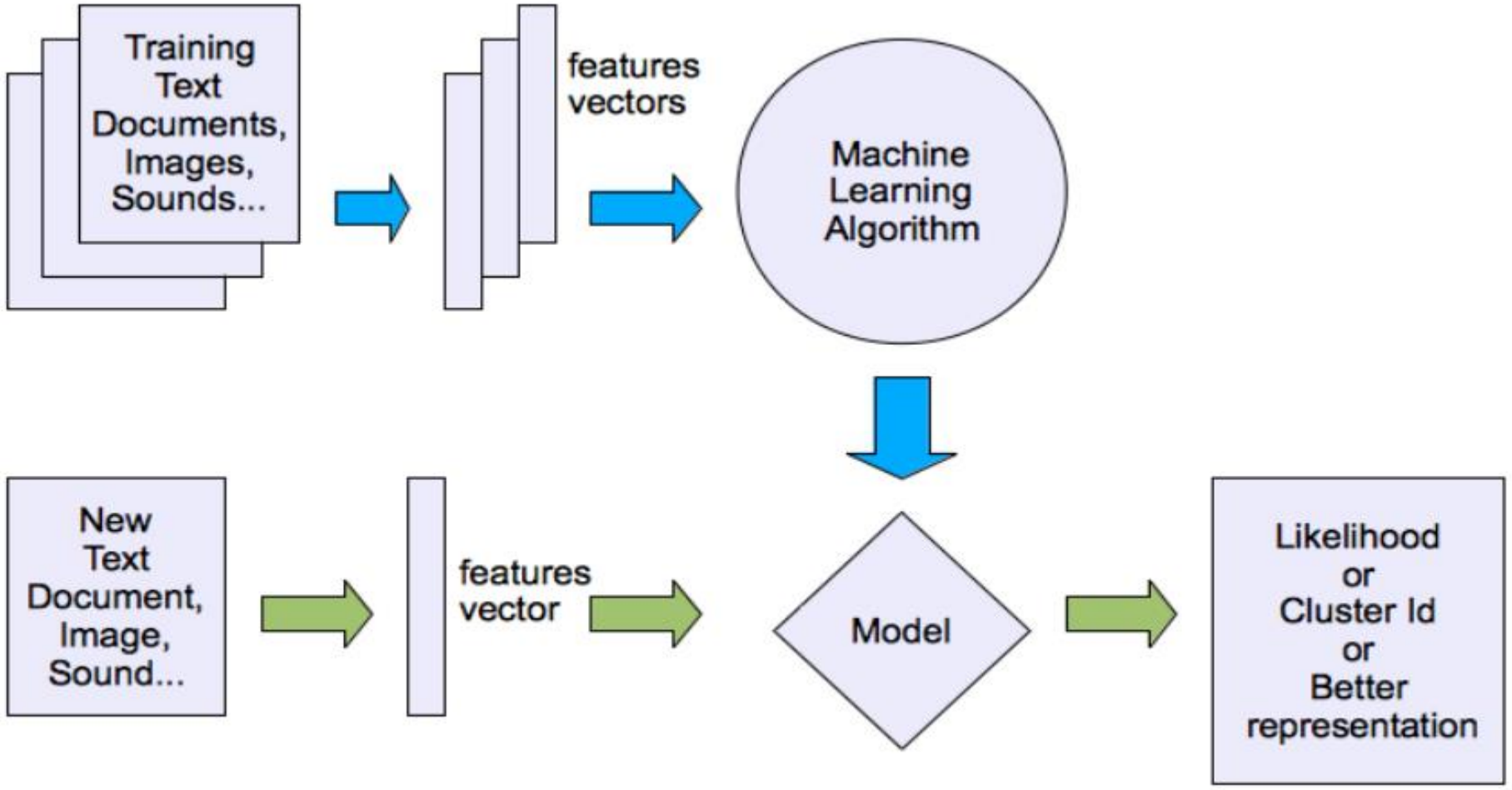








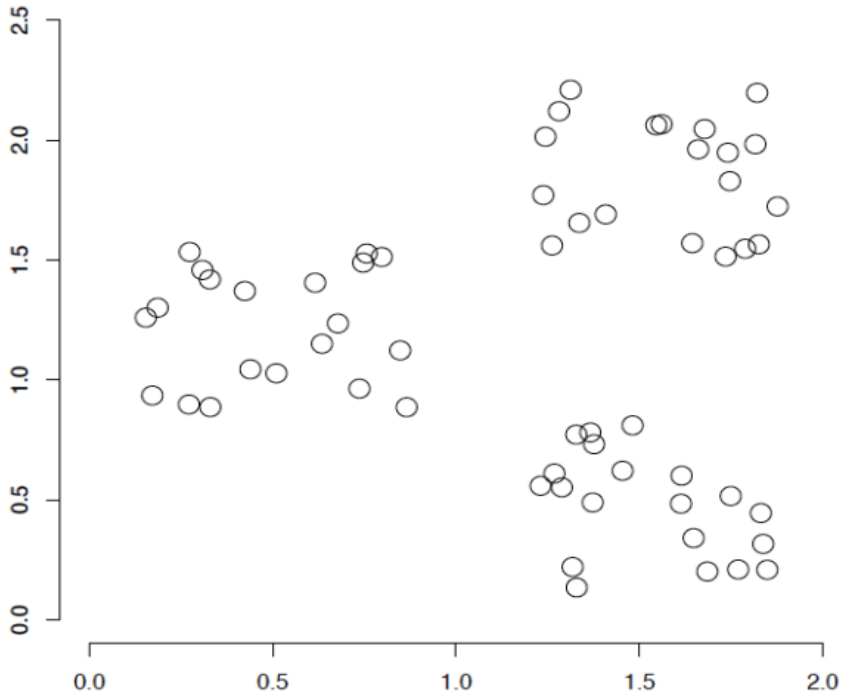
Unsupervised Learning

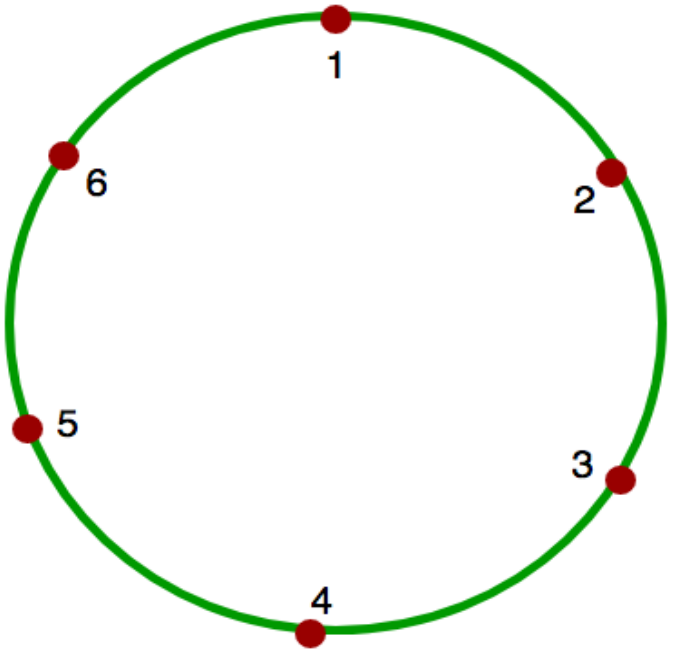


Unsupervised Learning

Clustering (T): Group data objects/ instances into cluster so that:

- Data instances within one cluster are “similar” (“closed”) to each other
- Data instances from different clusters are “dissimilar’ (“far”) to each other
- Clustering is rather subjective.





Circle with 6 equidistant points

Unsupervised Learning

Data (E): {x_i, I =1,2 4, …, n} unlabeled vector data.

“similarity”/”disimilarity”, “far”/”closed” are measured by similarity measures:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

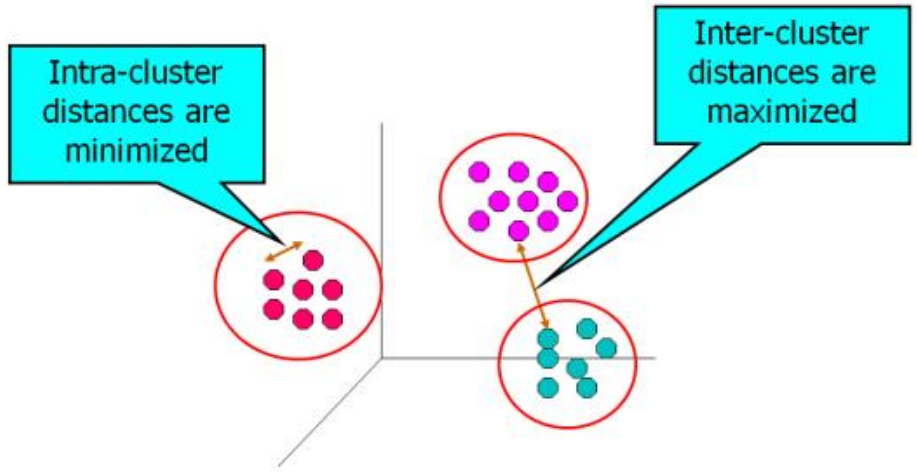
$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

There might be no prior information for the number of clusters!

Unsupervised Learning

Goodness measures for clustering (P):

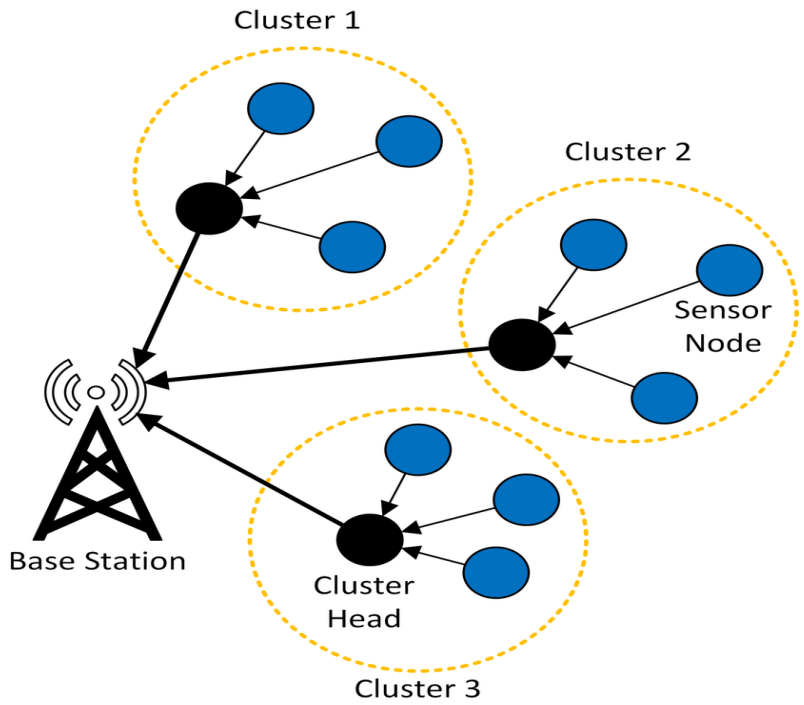
- Based on intra-similarity and extra-similarity measures of data instances.
- Based on ground-truth: need labeled data.
- Based on purity, entropy, ….



Unsupervised Learning

Examples of Clustering application problems:

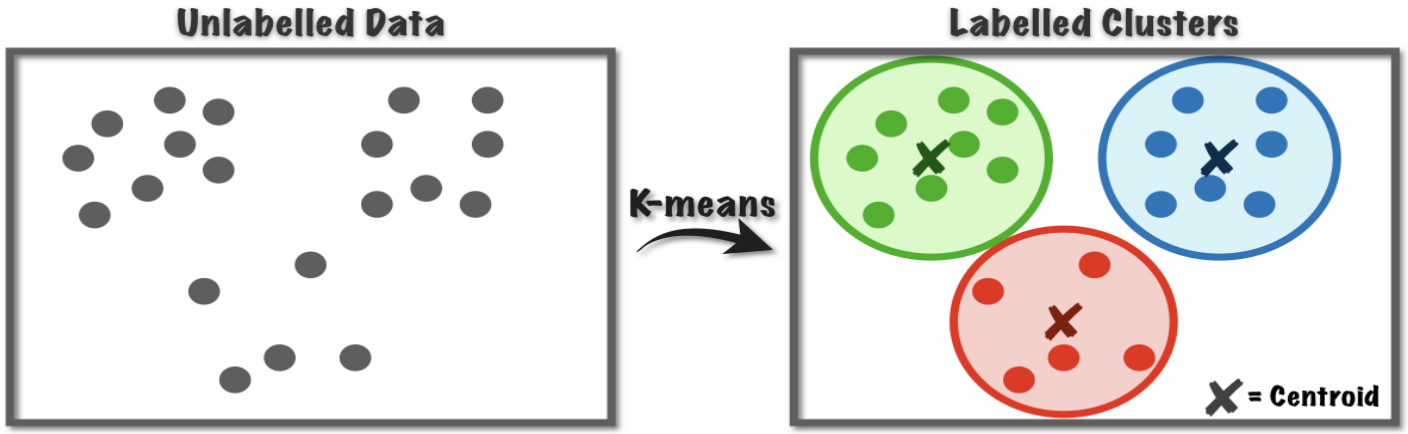
- Customer segmentation.
- Anomaly detection.
- Clustering access connections for location optimization of BTS.



Unsupervised Learning

Techniques for Clustering:

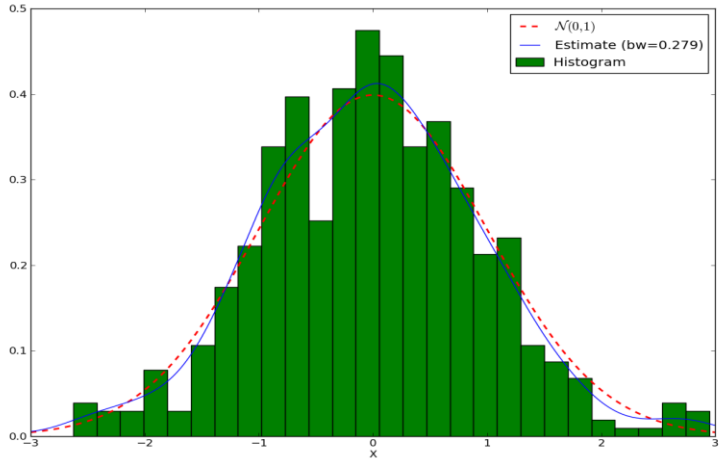
- Partitioning Algorithms (eg. K-means).
- Hierarchy algorithms.
- Density-based algorithms.
- Grid-based algorithms.
- Model-based algorithms.

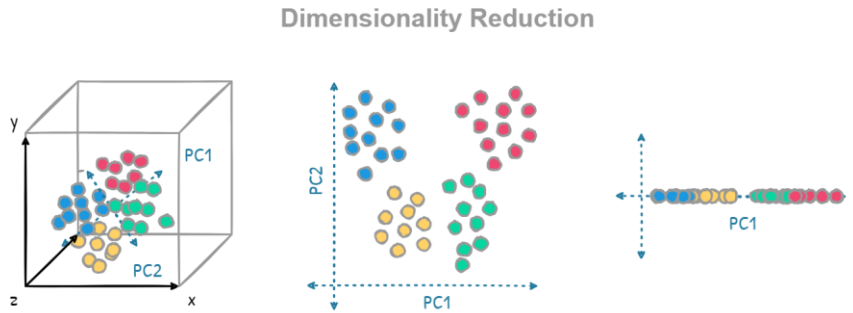


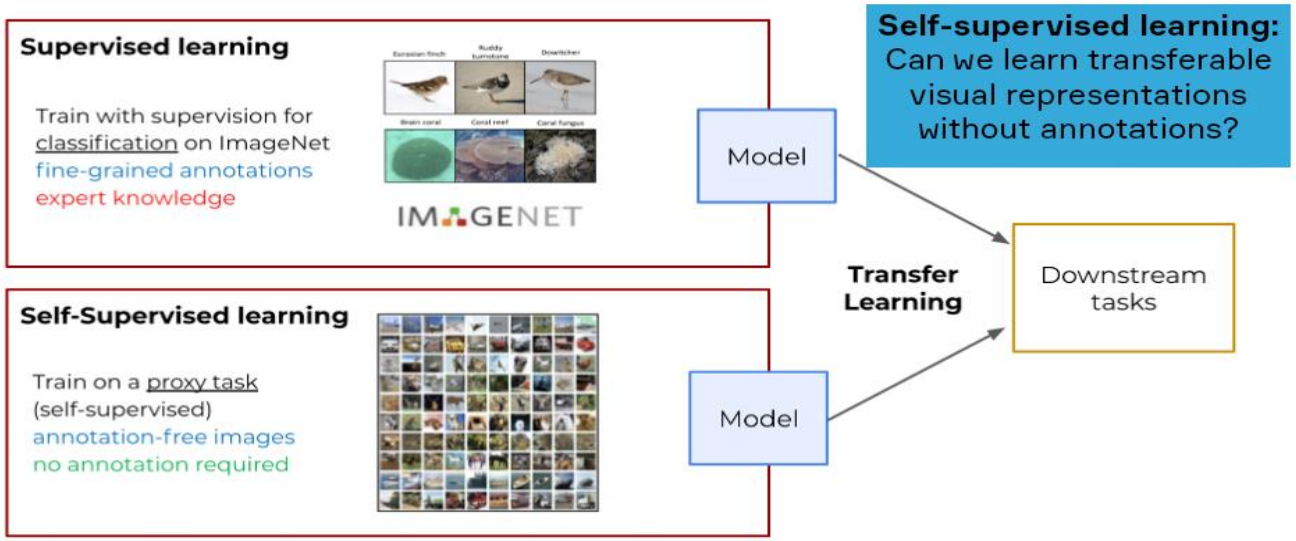
Unsupervised Learning

Other unsupervised learning problems:

- Density Estimation.
- Dimensionality Reduction
- Self-supervised learning



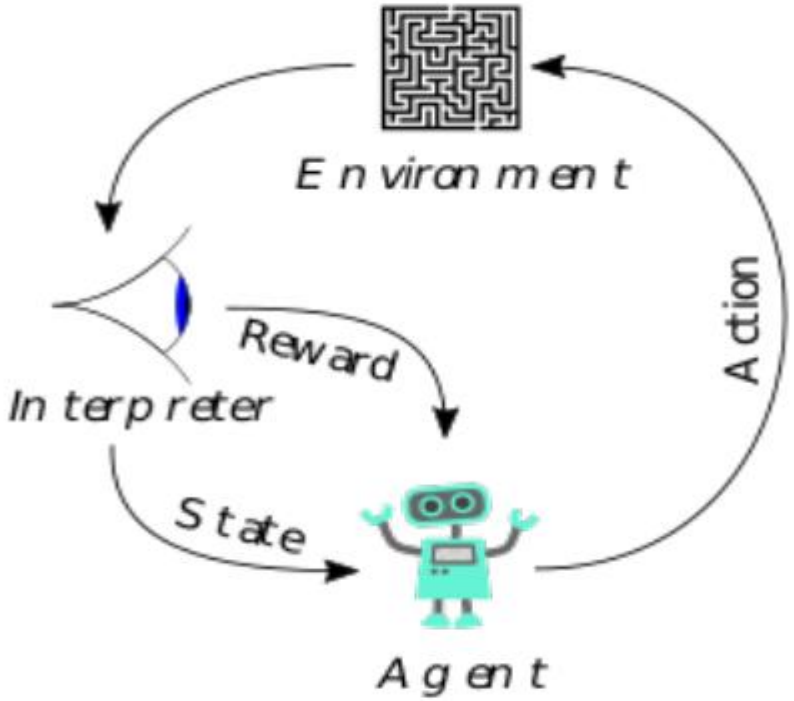




Reinforcement Learning

General AI Agent Problem (T):

- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at environment step



Task: Maximize the rewards that AI Agent could receive in the long run (e.g after time T and/or condition C is satisfied).

Reinforcement Learning

General AI Agent Problem (T): For each agent

- Policy function: To choose action at each state t of the agent.
- Value Function: To estimate the reward after each step of the agent.
- Model: Agent's representation of the environment.

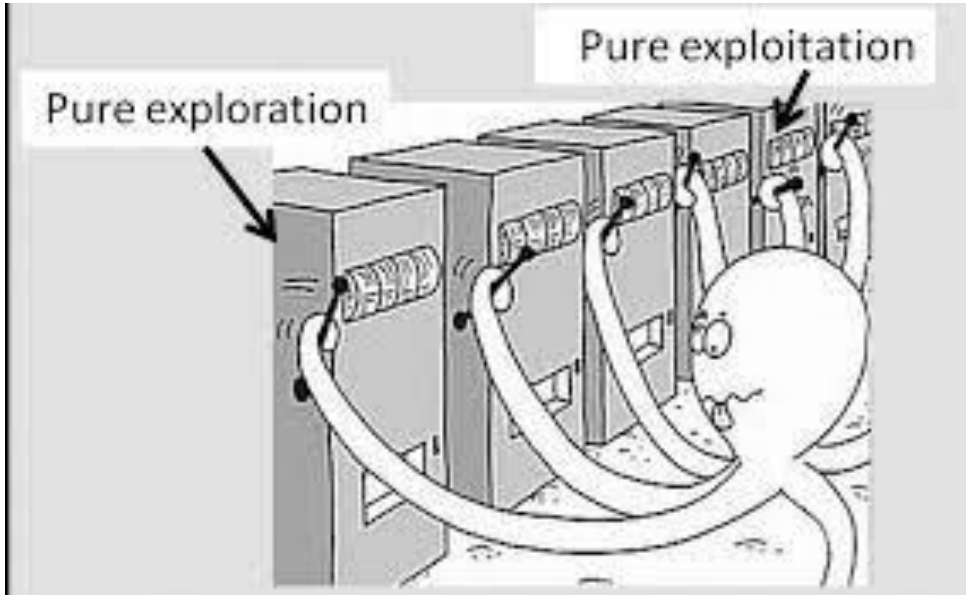
Data (E): Collected when the agent acts in the environment.

Task: Learn the policy function so as to maximize the reward for the agent in the long run (e.g after time T and/or condition C is satisfied).

Reinforcement Learning

Reinforcement learning application problems:

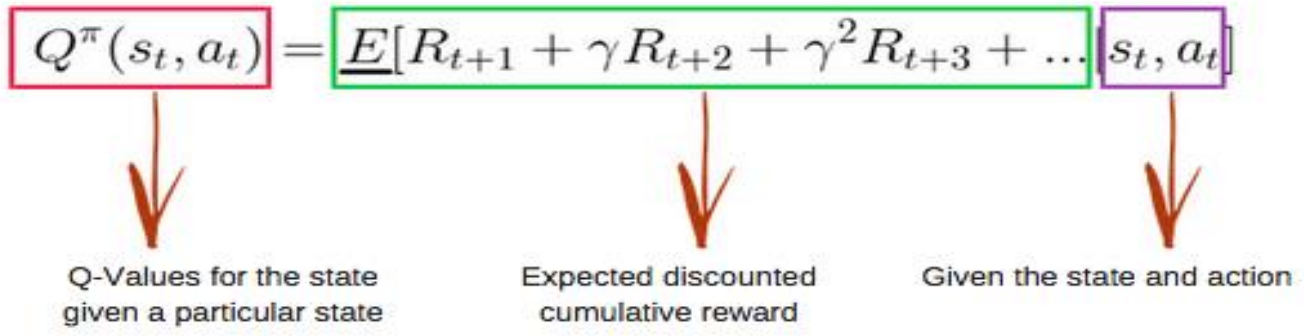
- Multi-armed bandits (exploration vs. exploitation)
- Robot control.
- Learn to play chess.
- Learn optimal strategies for resource allocation in data centers



Reinforcement Learning

Techniques for reinforcement learning:

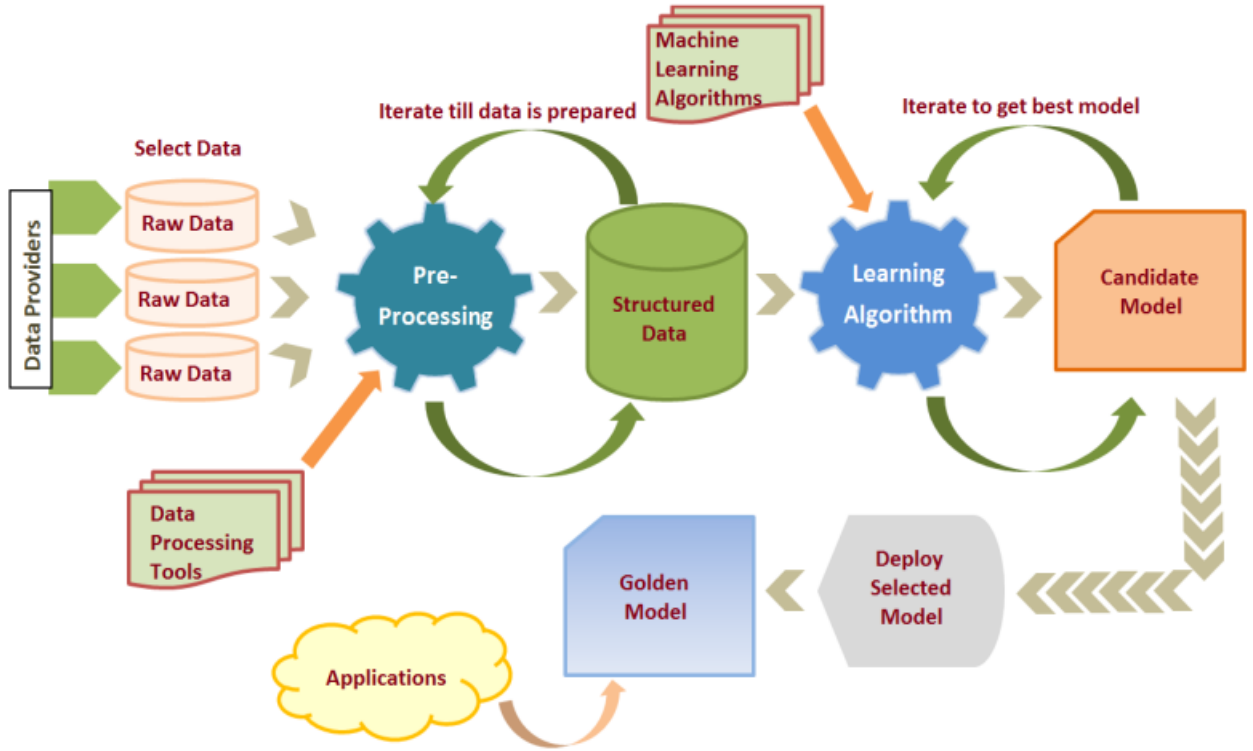
- Value-based, policy-based, model-based.
- Q-learning, Temporal Difference Learning, Deep RL.



Machine Learning Pipeline

Machine Learning Pipeline

- Data Collection
- Data Preparation
- Model Selection
- Model Training
- Evaluation
- Deployment



Data Collection

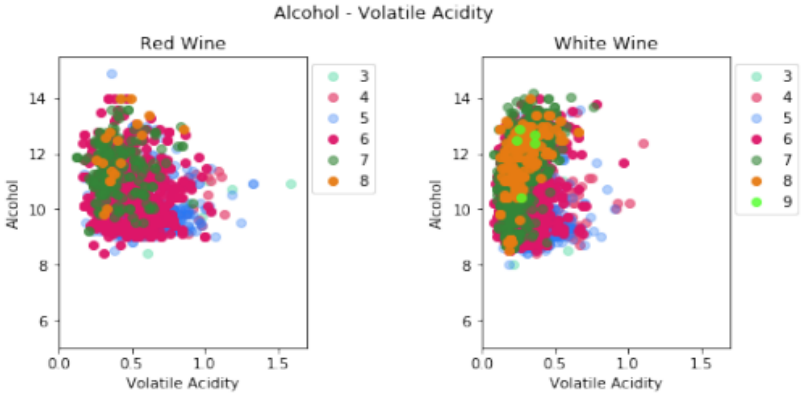
- How large the data is: numbers of items or stories
- Source of data: environment, system stories, human data
- Available of data source and time
- Complexity: types of data, relational data, distributed data
- Frequency of data collection
- Which form of data analysis

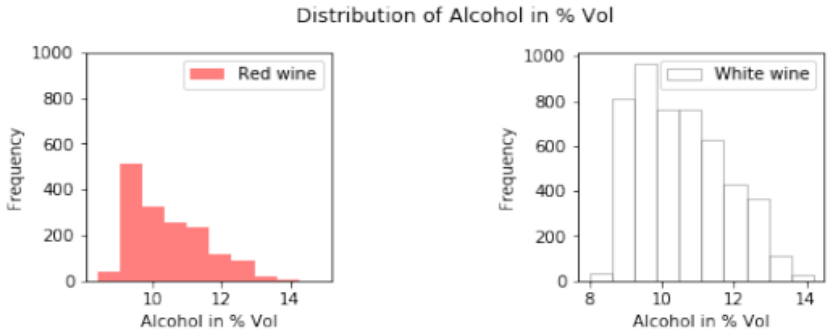
Data Preparation

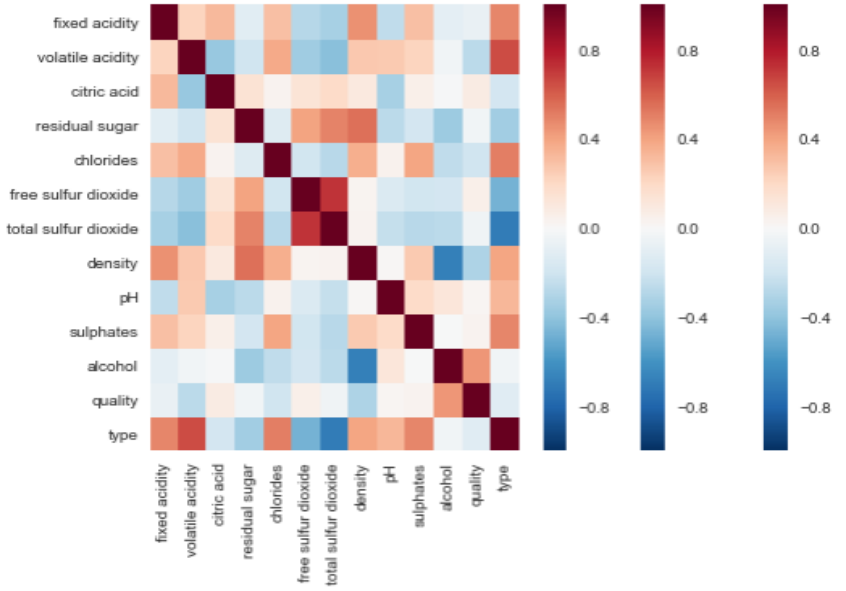
- Why data preparation needed
  - Data need to be formatted for a given toolbox
  - Data need to be made adequate for a given data mining method
- Major Tasks in Data Preparation
  - Data exploration
  - Data cleansing
  - Data Transformation (feature engineering, feature extraction)
  - Balancing
  - Feature selection

Data Exploration

- What features relevant with target
- Correlation among features
- Correlation between features of difference classes

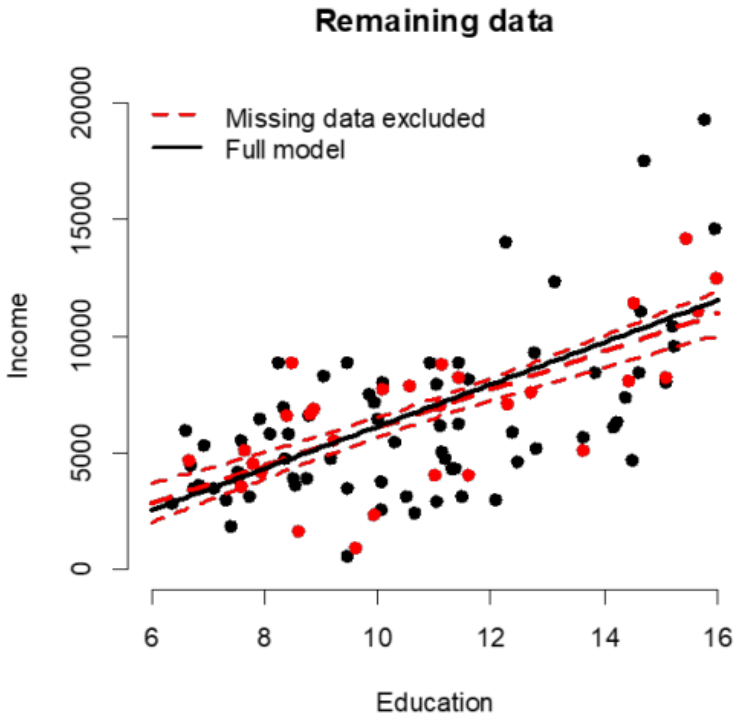






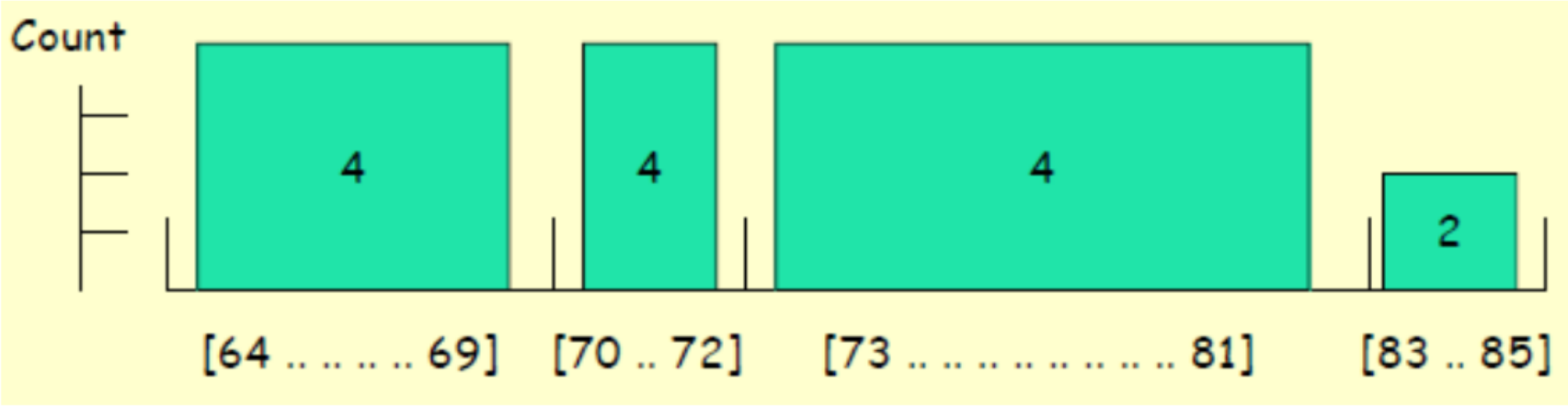
Data Cleansing – Missing Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred.
- Missing values may carry some information content



Data Transformation - Discretization

- Divide the range of a continuous attribute into intervals
  - Some methods require discrete values, e.g. most versions of Naïve Bayes, Decision Tree, Rule Based
  - Reduce data size by discretization
  - Prepare for further analysis
  - Discretization is very useful for generating a summary of data



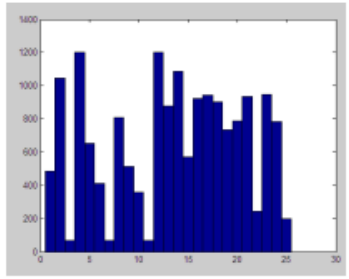
Data Transformation – Normalization

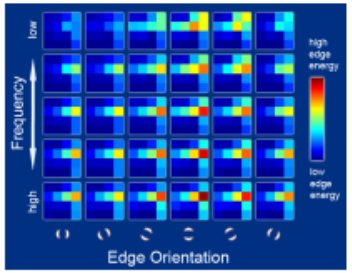
- Normalization
  - Different scales of attributes/features reduce accuracy of distance-based models
    - Age: 0-99, Salary: 1 million to 1 billion VND
  - Normalization to equalize scale of features, normally from 0 to 1
  - Some simple methods:
    - min-max normalization
    - z-score normalization
    - normalization by decimal scaling

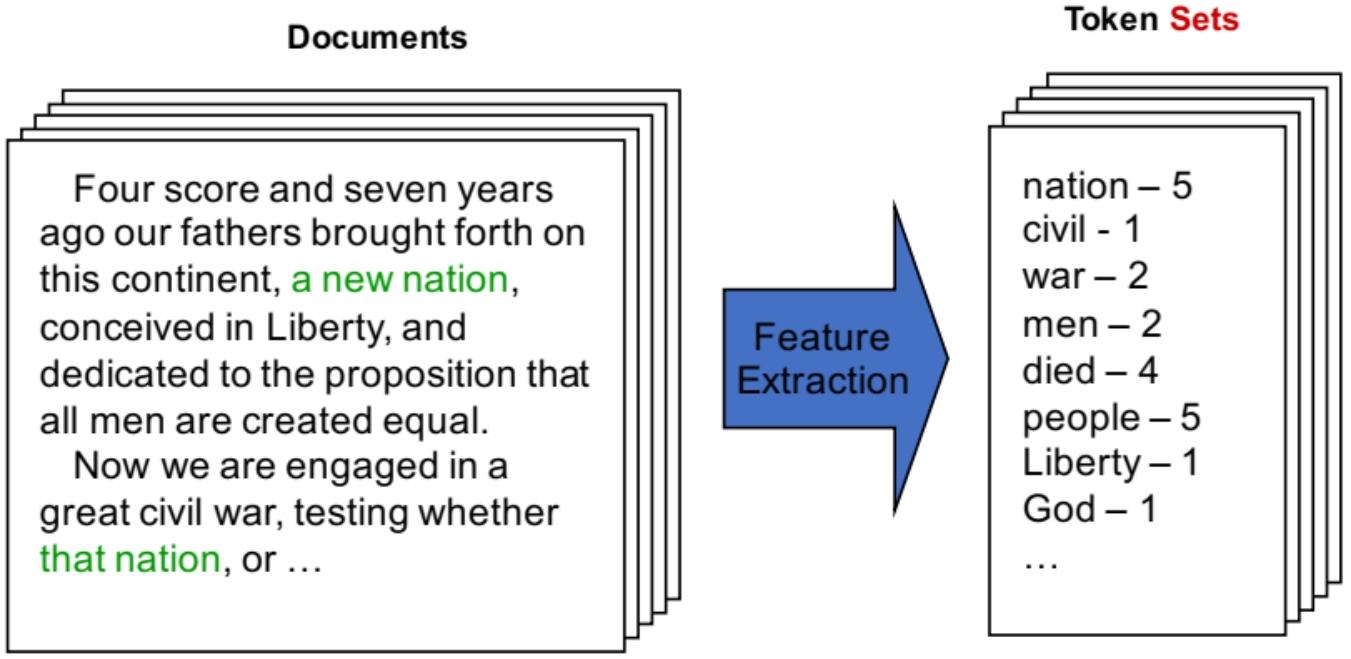
$$v' = \frac{v - \min_v}{\max_v - \min_v}(new\_max_v - new\_min_v) + new\_min_v$$

Feature Engineering

- Feature engineering:
  - Raw pixels to histograms, SHIFT, GIST descriptors,
  - Text: BoW, TF-IDF
  - Categorial: coding, one-hot
  - Graph topology
- Feature extraction
  - PCA, LDA, Autoencoder…
- Feature learning (representation learning: very interesting with deep learning methods)

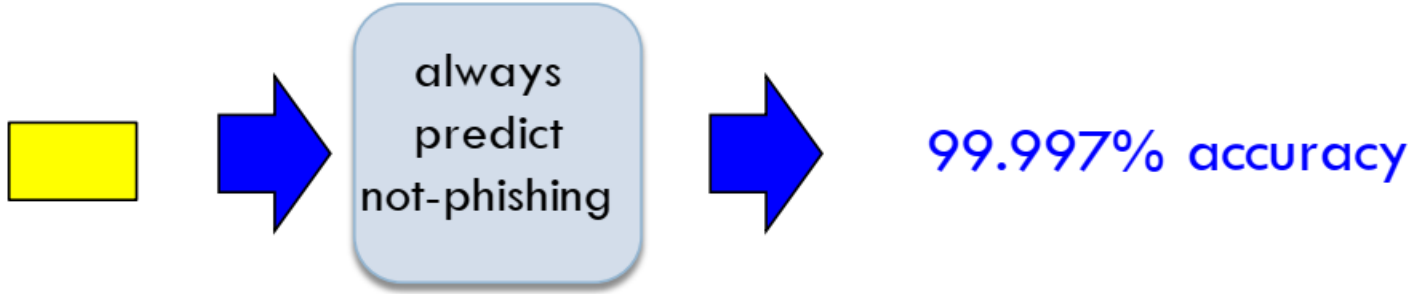


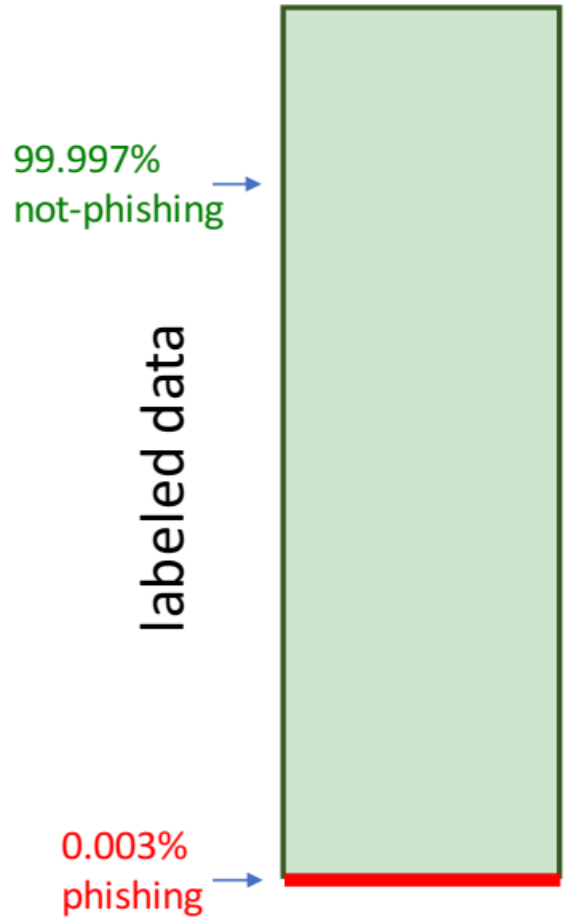




Handling Imbalanced Data

- Imbanlanced data: number of one class is far bigger than the rest
- Phishing email detection
  - Every 1 million email, there are 30 phishing emails





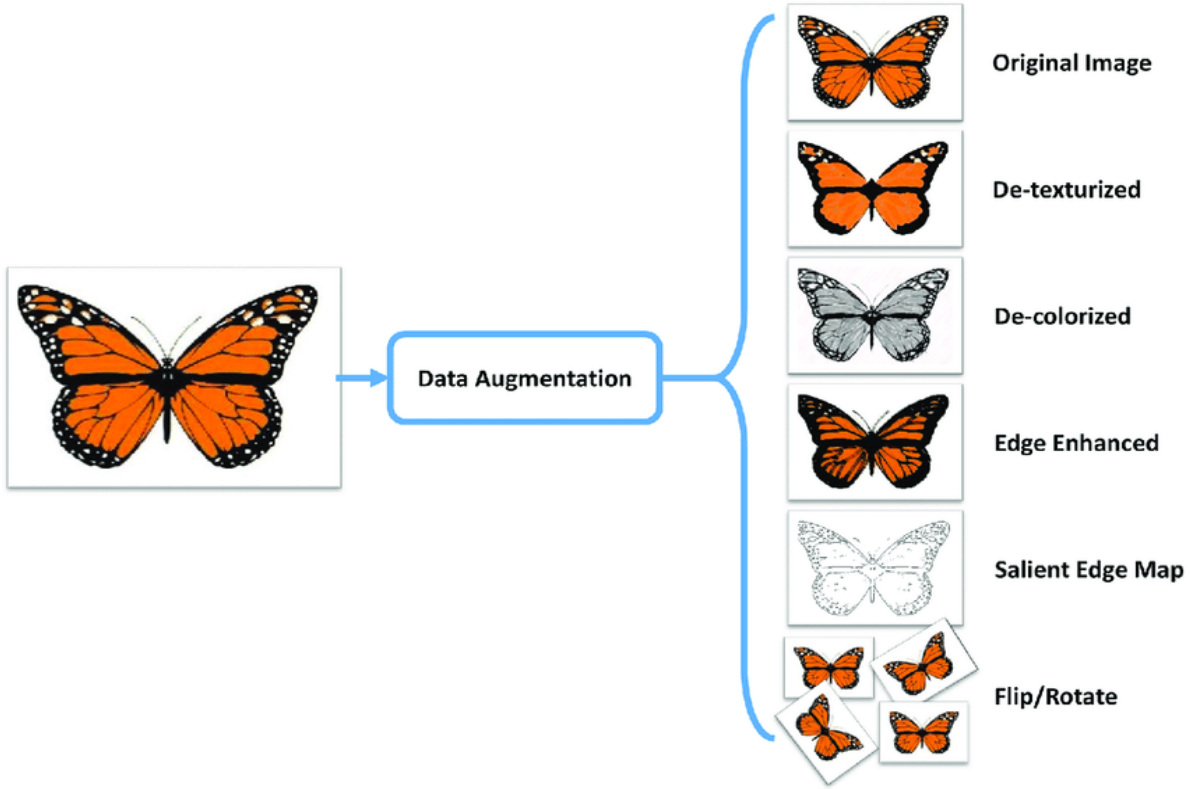
Data Augmentation

Data Augmentation for:

- Handling the lack of training data (eg. Imbalanced).
- Helping machine learning models be more robust (against noise).

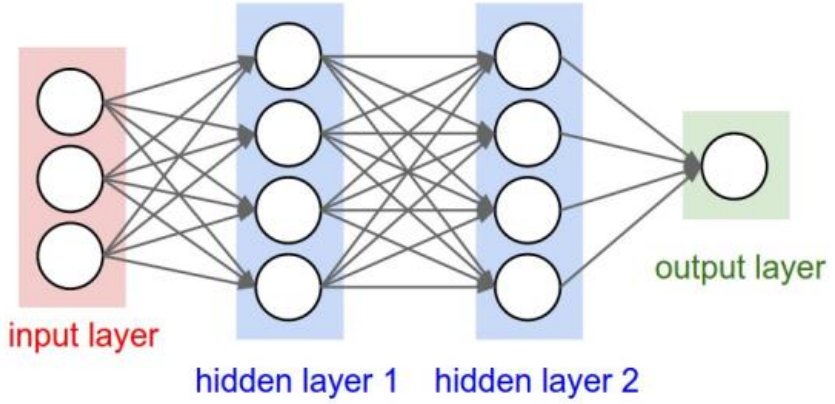
Methods:

- Data transformation.
- Noise injection.
- Data generation (eg GANs)



Model Selection

- Hyper parameter
  - Regularization strength
    - Naïve Bayes, linear and logistic regression
  - SVM
    - Kernel trick/feature extraction
  - Decision trees
    - Depth and leaves
  - Boosting
    - Number of rounds
  - Neural network
    - Step size/learning rate
    - Mini-batch size…

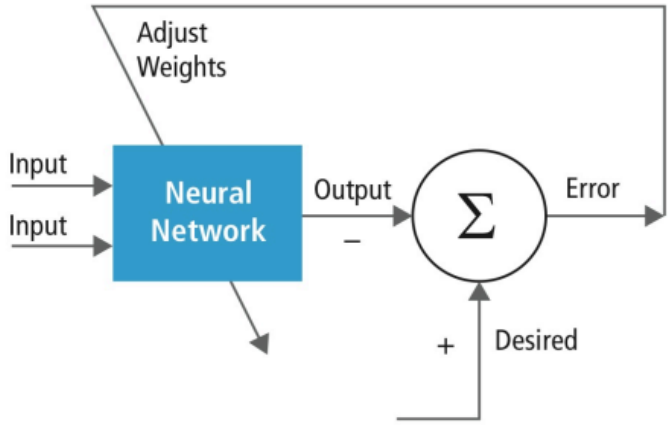


Model Training

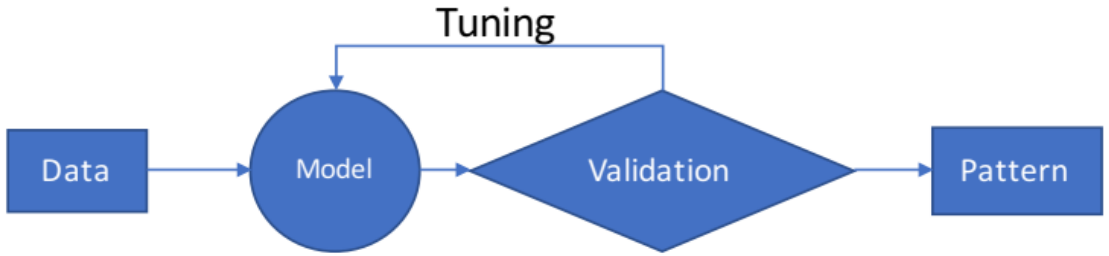
- Tuning parameter to find the best model that approximate distribution of data

$$y = f(x)$$

output — prediction function — Input data

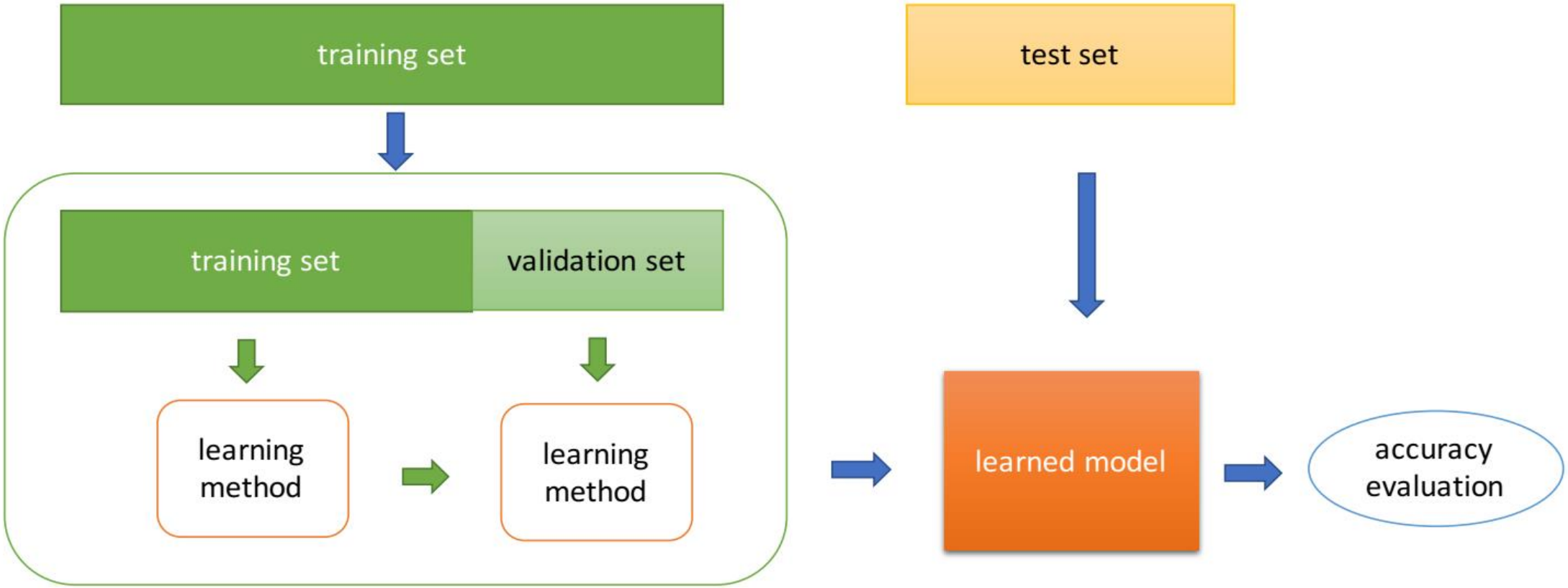


Supervised Learning

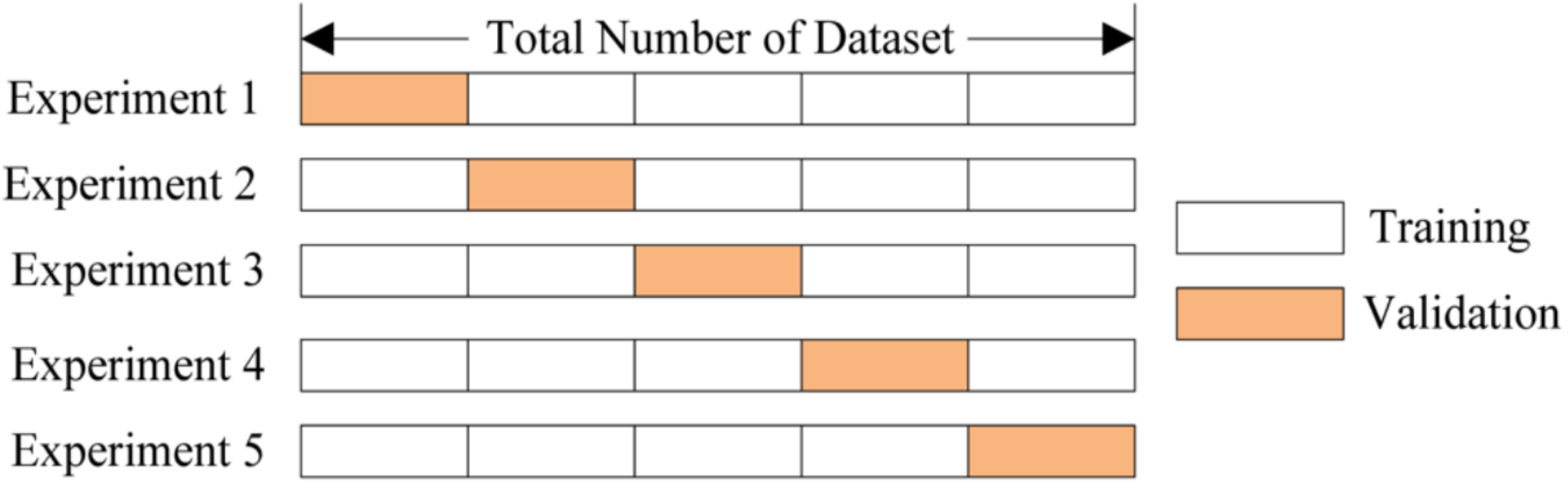


Unsupervised Learning

Datasets for Training and Evaluation

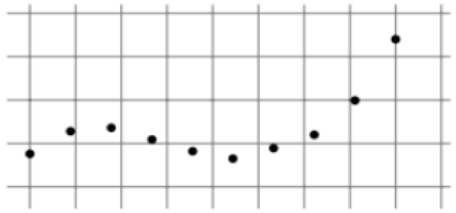


Cross Validation



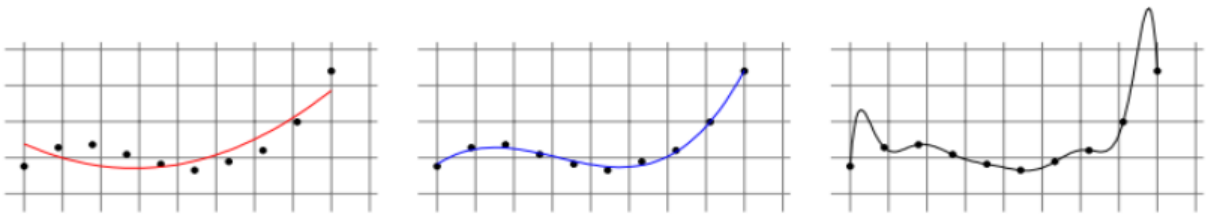
Learning Generalization

- Overfitting

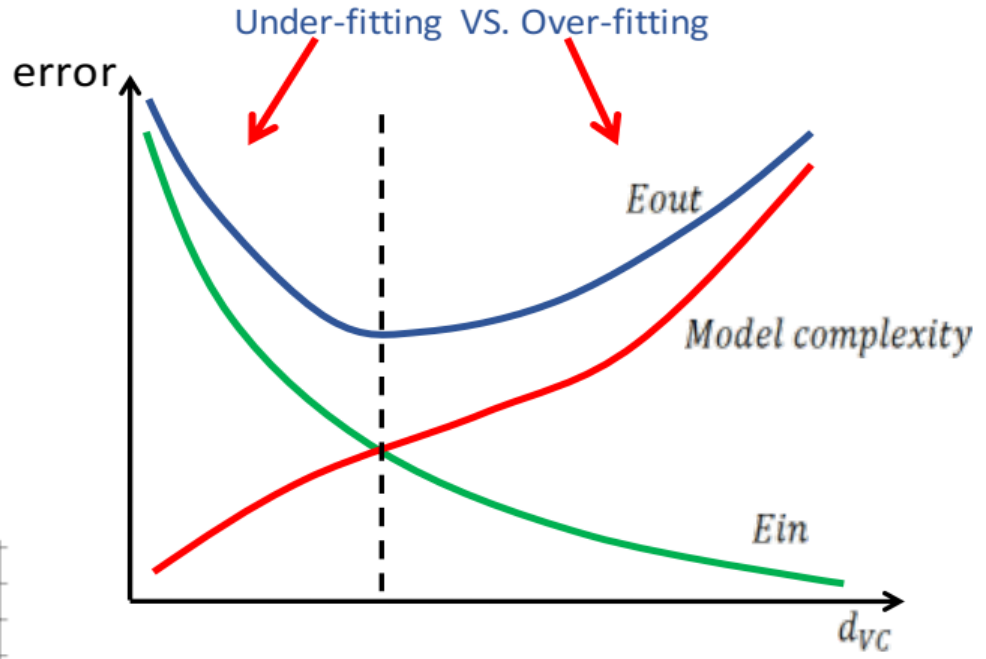


Training data

degree 2 degree 3 degree 10

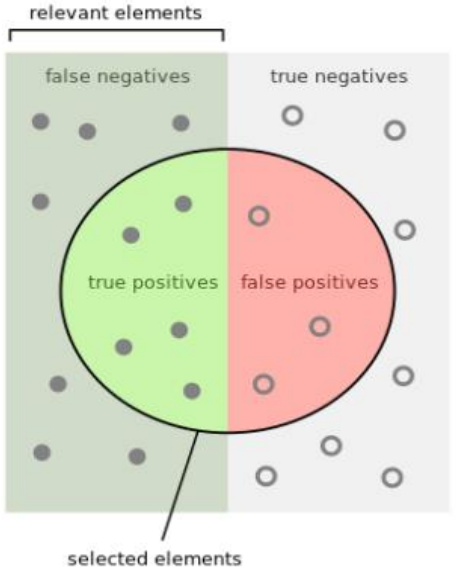


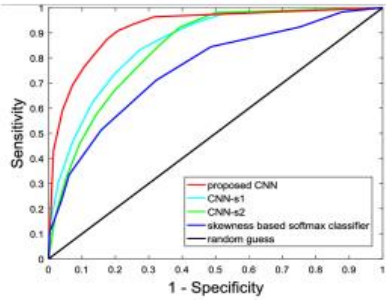
3 model with different number of parameters

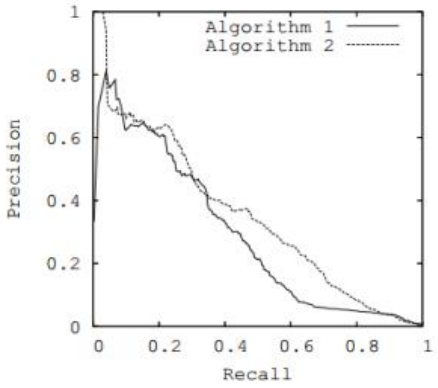


Model Evaluation: Classification

Accuracy	How many number of all emails including phishing is correctly classify
Precision	Number of correctly predicted phishing email/Number of predicted as phishing
Recall	Number of correctly predicted phishing email/ Number of actual phishing email
f1-score	Mean of precision and recall
ROC - Receiver operating characteristic, PRC - Precision Recall Curve, AUC – Area under the curve of ROC	

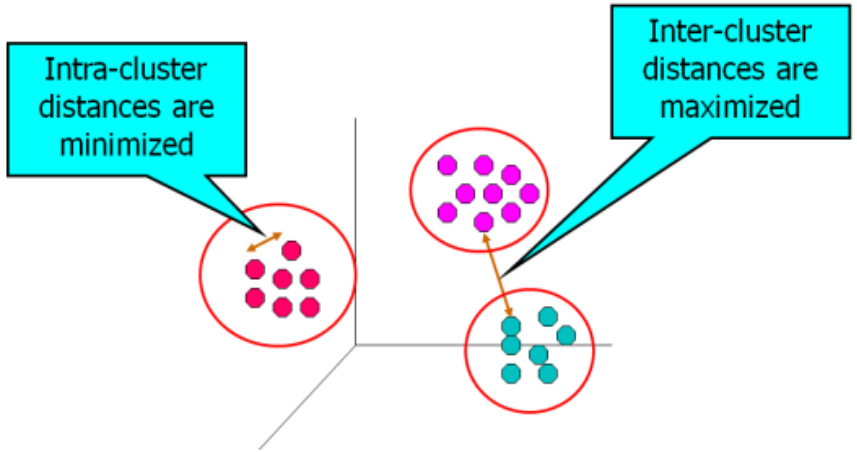






Model Evaluation: Clustering

- Intra-cluster cohesion (tính nhỏ gọn)
  - Cohesion: how the data near the centroid of the clusters
  - SSE (Sum of squared error) is usually used



- Inter-cluster cohesion (tính giãn cách)
  - Measure the space among clusters
  - Usually as the sum of distance among centroids of clusters

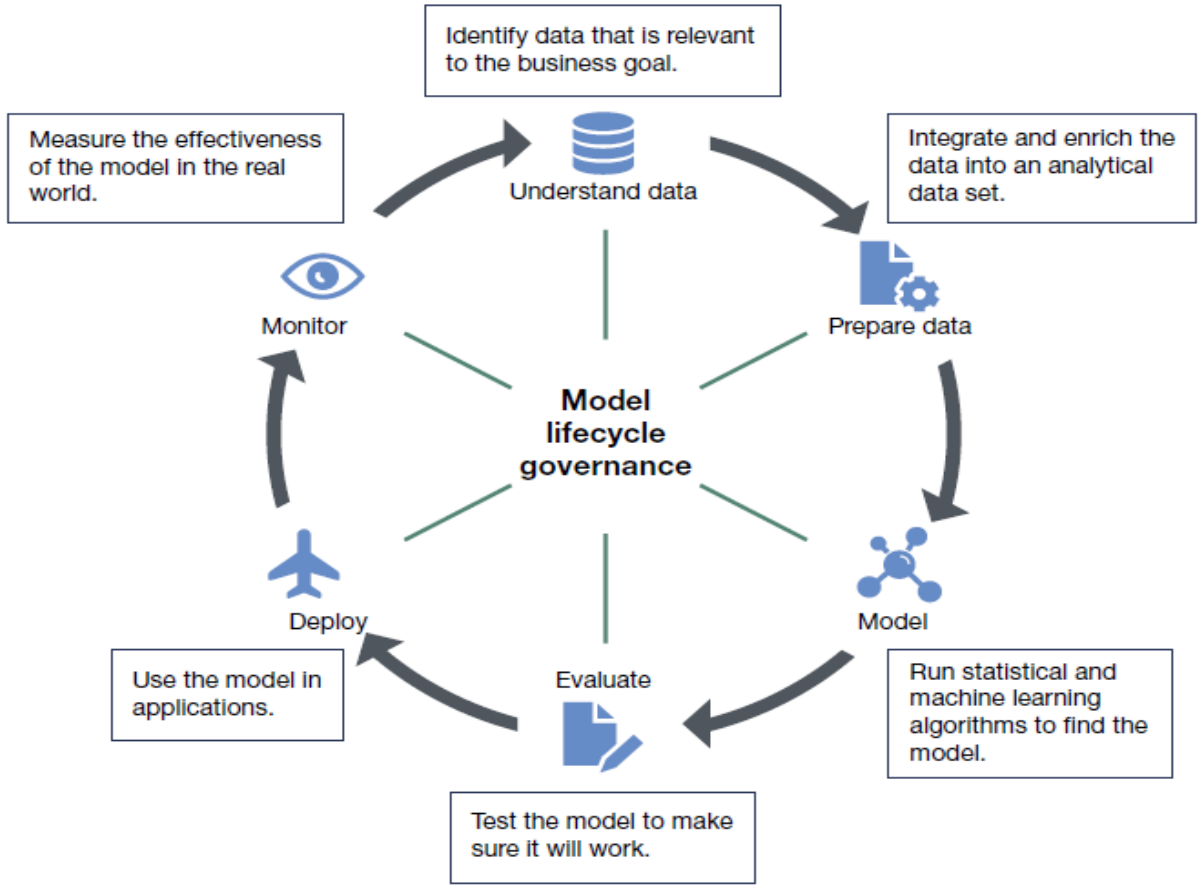
Machine Learning Engineering

ModelOps: Governance of models through all model life cycle.

Objective: Model quality (model fit).

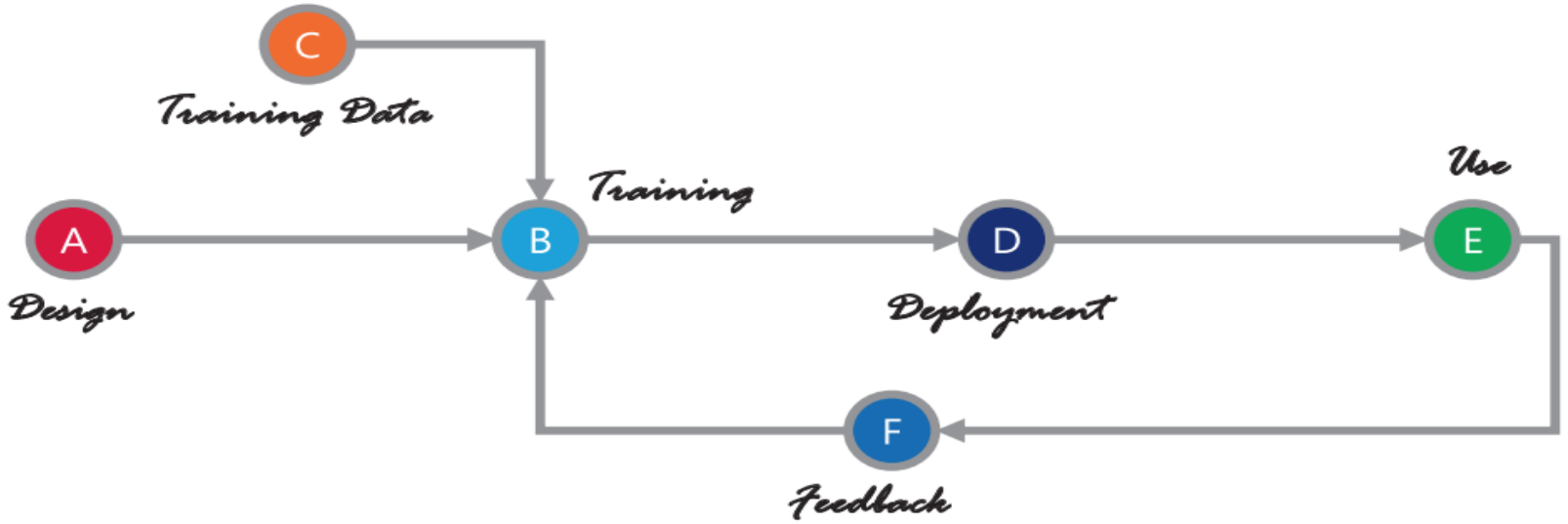
Subset - MLOps:

- Model development.
- Model store and versioning.
- Model testing/ evaluating.
- Model deployment (CI/CD).
- Model roll back.
- E.g: Kubeflow

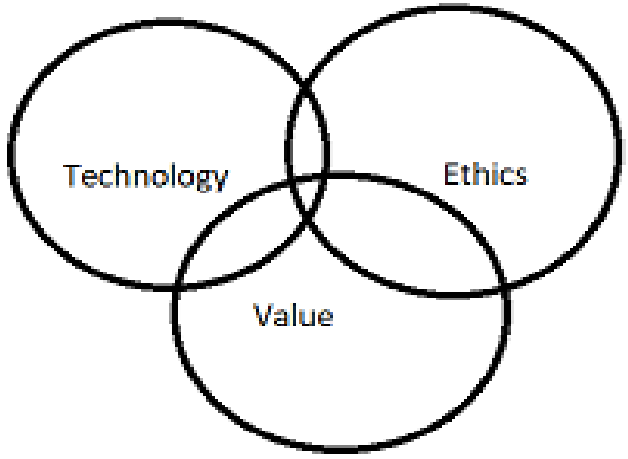


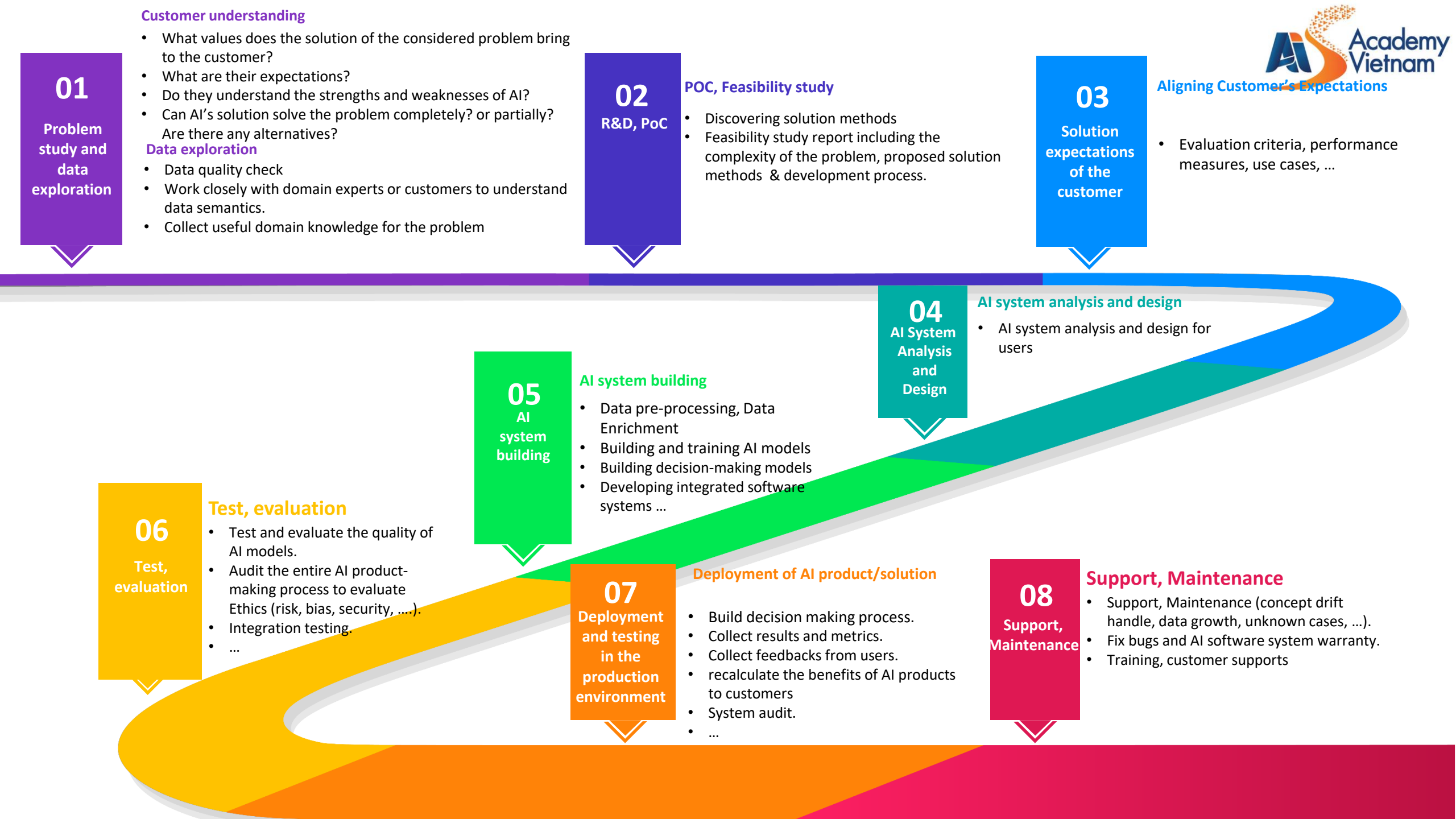
AI/Machine Learning Project

Full AI Development Process



Three Pillars for AI Solutions




01
Problem study and data exploration
Customer understanding
What values does the solution of the considered problem bring to the customer?
What are their expectations?
Do they understand the strengths and weaknesses of AI?
Can AI's solution solve the problem completely? or partially? Are there any alternatives?
Data exploration
Data quality check
Work closely with domain experts or customers to understand data semantics.
Collect useful domain knowledge for the problem
02
R&D, PoC
POC, Feasibility study
Discovering solution methods
Feasibility study report including the complexity of the problem, proposed solution methods & development process.
03
Solution expectations of the customer
Aligning Customer's Expectations
Evaluation criteria, performance measures, use cases, …
04
AI System Analysis and Design
AI system analysis and design
AI system analysis and design for users
05
AI system building
AI system building
Data pre-processing, Data Enrichment
Building and training AI models
Building decision-making models
Developing integrated software systems …
06
Test, evaluation
Test, evaluation
Test and evaluate the quality of AI models.
Audit the entire AI product-making process to evaluate Ethics (risk, bias, security, ….).
Integration testing.
…
07
Deployment and testing in the production environment
Deployment of AI product/solution
Build decision making process.
Collect results and metrics.
Collect feedbacks from users.
recalculate the benefits of AI products to customers
System audit.
…
08
Support, Maintenance
Support, Maintenance
Support, Maintenance (concept drift handle, data growth, unknown cases, …).
Fix bugs and AI software system warranty.
Training, customer supports

Neural Networks and Deep Learning

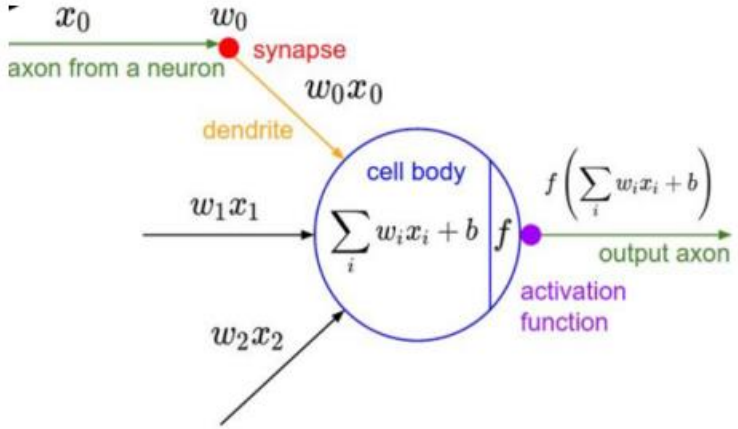
First Building Block – Perceptron

Linear classification

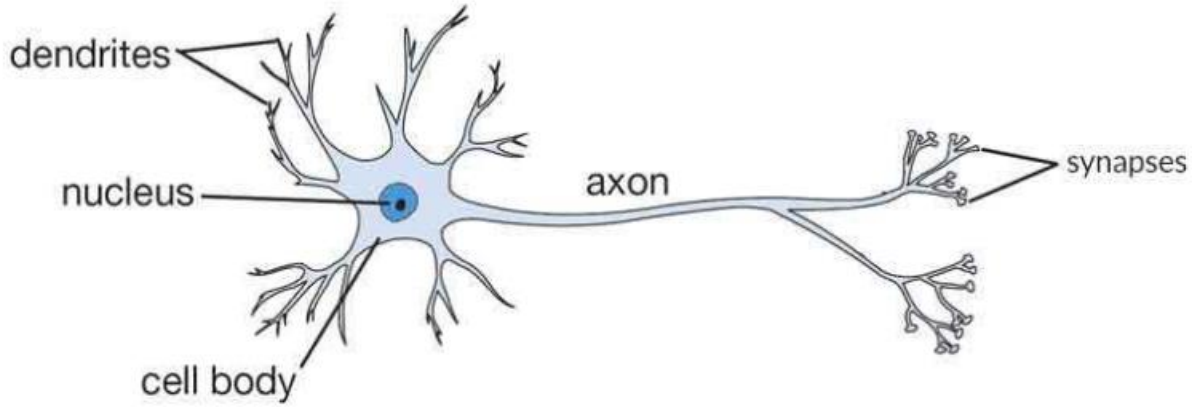
$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$$

Output = sign(Wx+b)

Perceptron

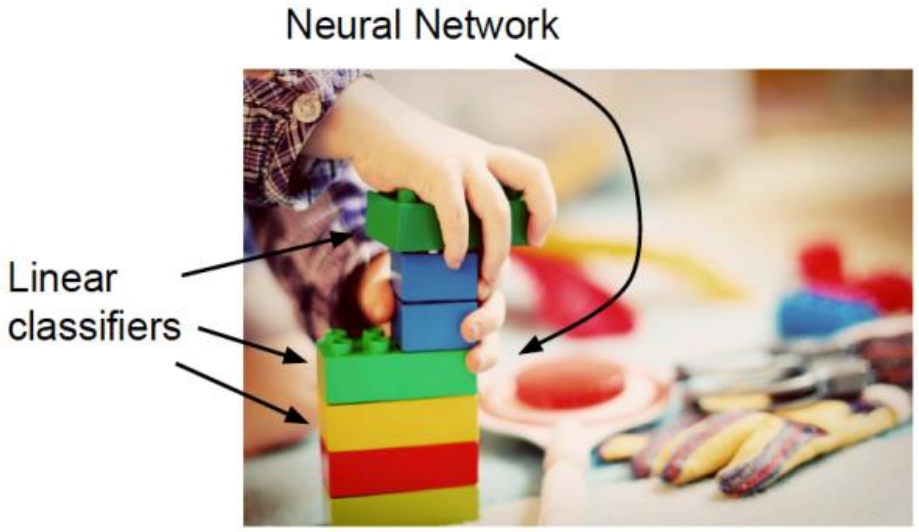


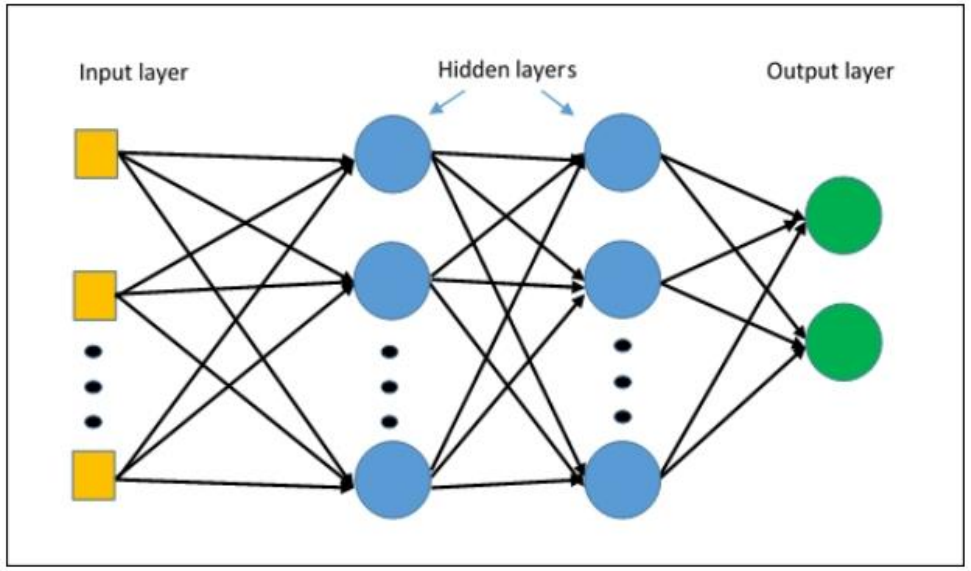
Biological Neuron



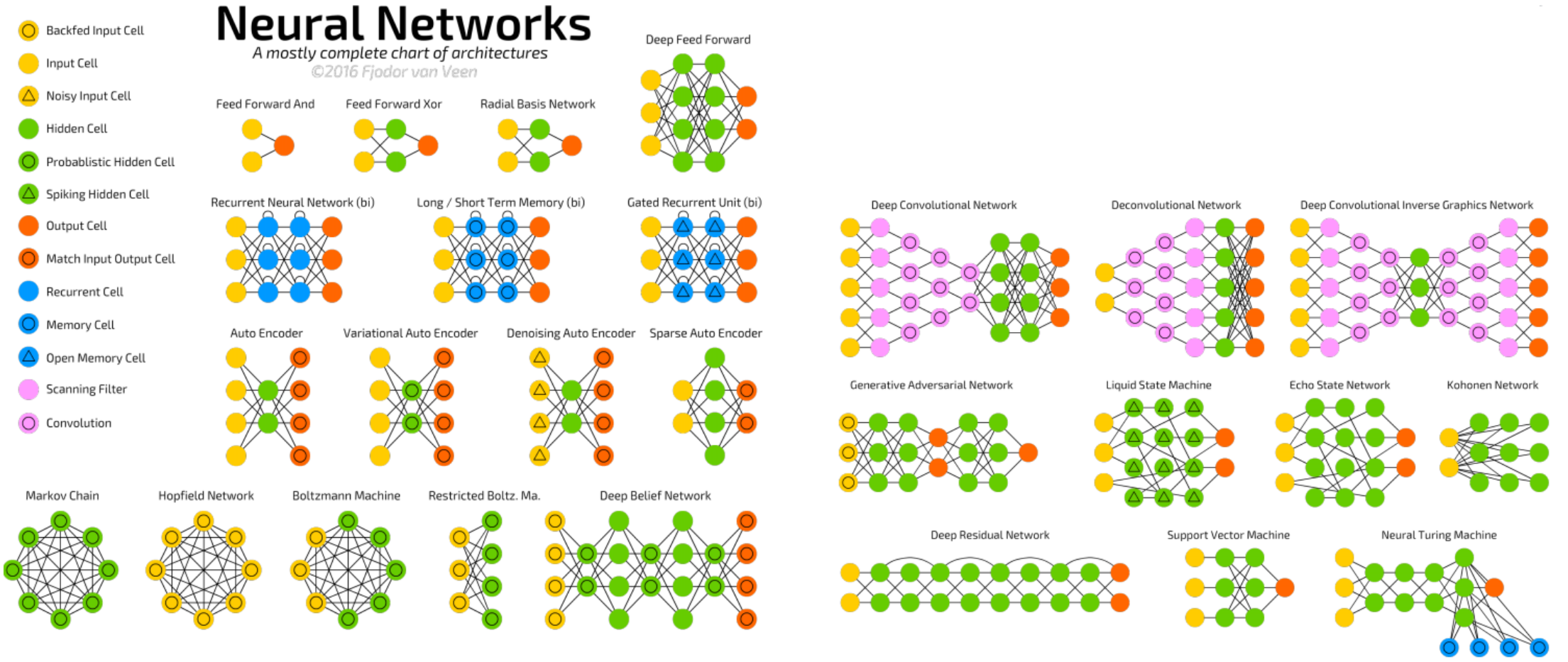
Neural Networks – Multi-layer Perceptron

- Multi-layers Perceptron
  - Layers: Bricks with many types
  - Network design: Construction and connection



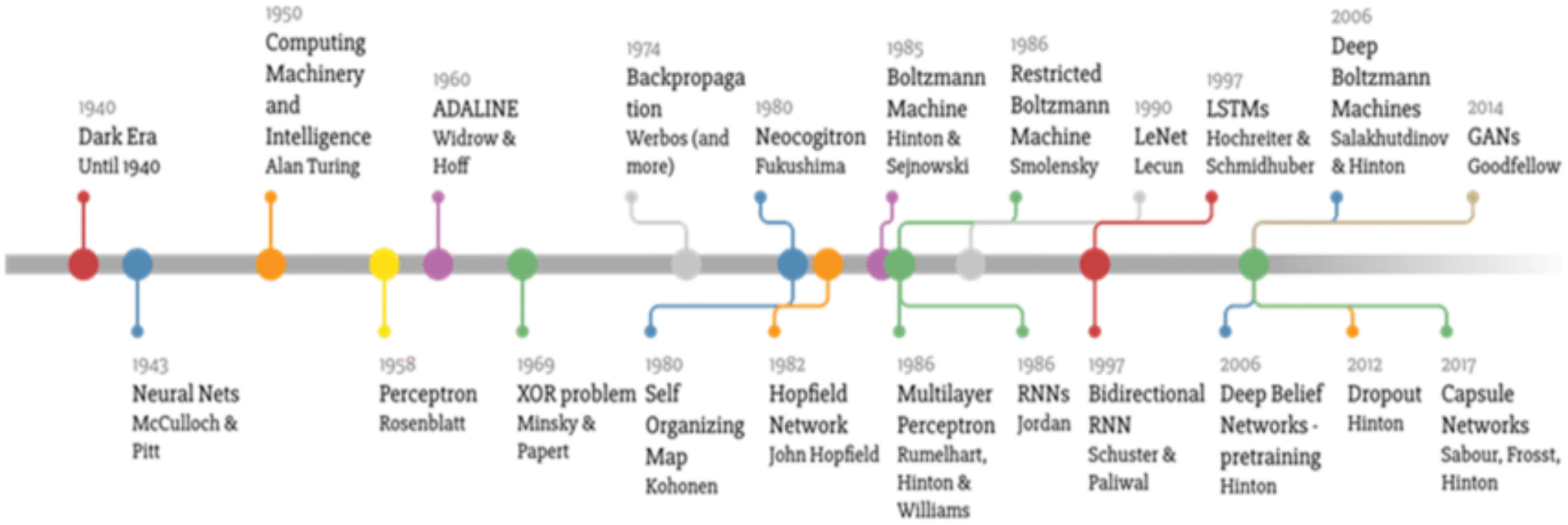


Neural Networks – Architecture



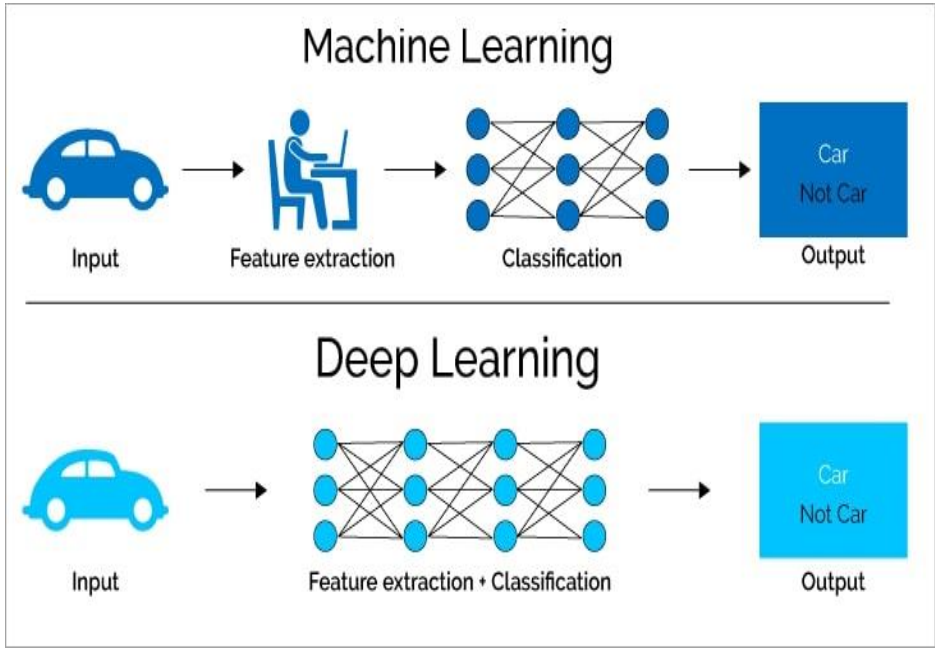
Deep Learning

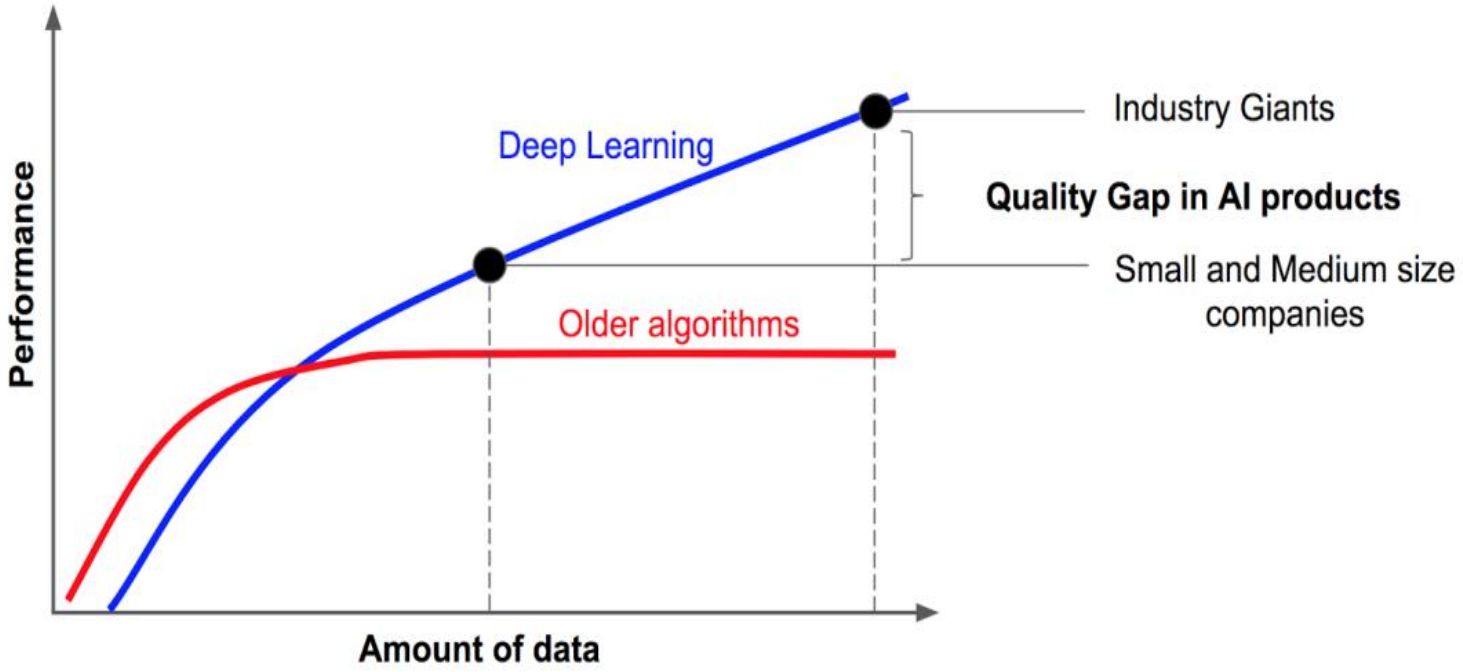
Neural Networks with many layers.



After 2017 – Transformers, Generative Deep Learning, ….

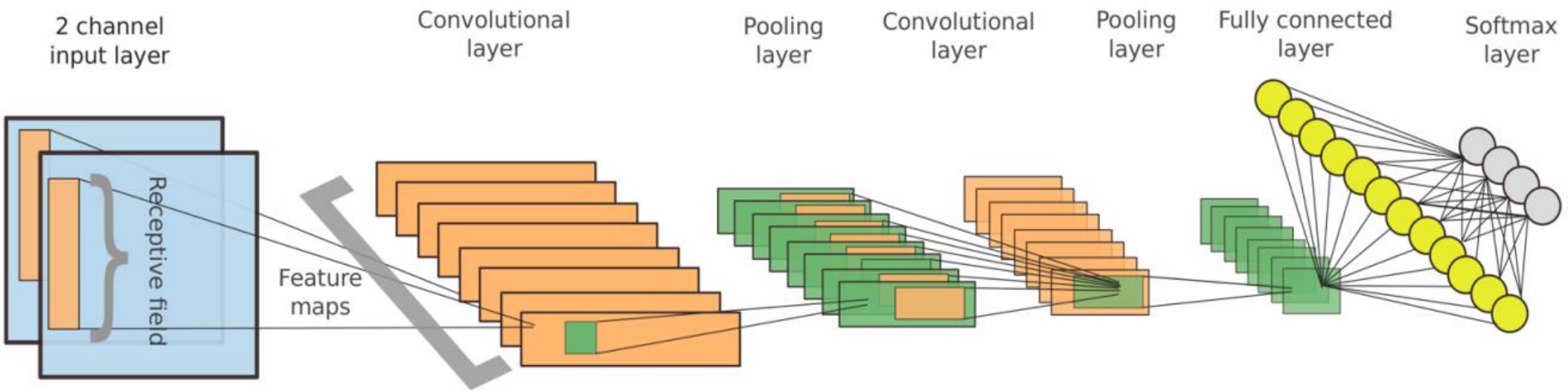
Deep Learning

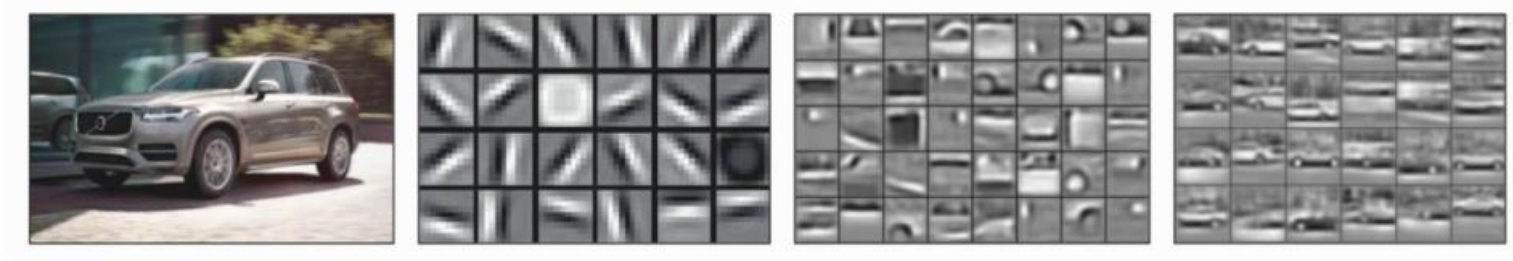




Vanilla DL Architectures

Convolutional Neural Networks (CNN):

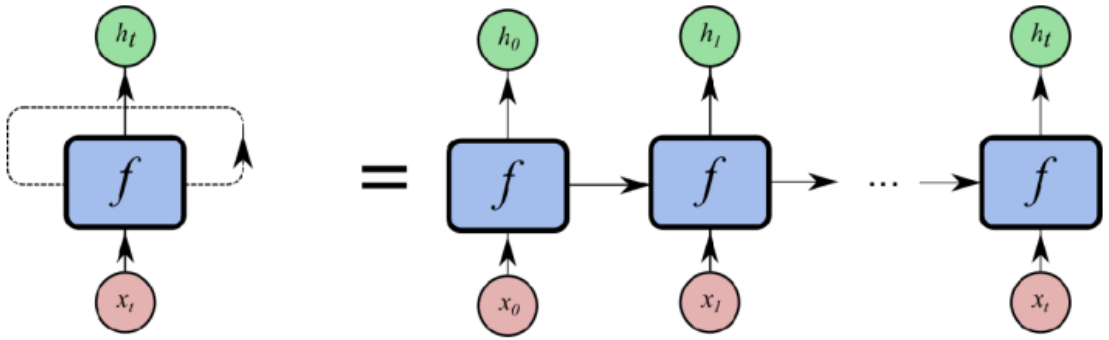




Vanilla DL Architectures

Recurrent Neural Networks (CNN):

- Usually want to predict a vector at some time steps
- Process a sequence of vector x by applying a recurrence formula at every step



$$h_t = f_W(h_{t-1}, x_t)$$

new state

some function with parameters W

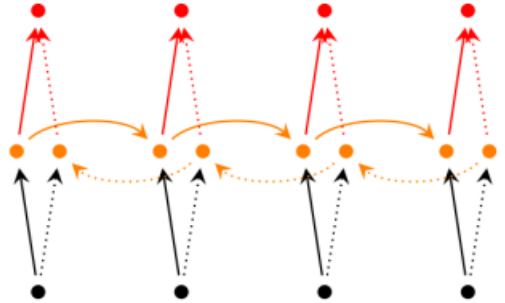
old state

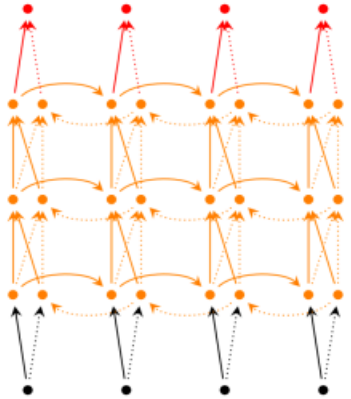
input vector at some time step

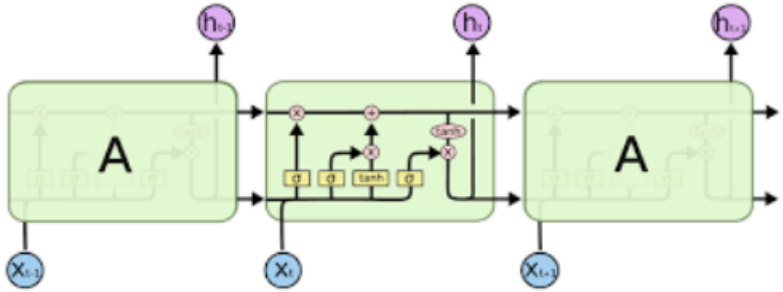
Vanilla DL Architectures

Recurrent Neural Networks (CNN):

- Bidirectional RNNs:
  - Output at time t depend on.
    - the previous elements in the sequer
    - future elements.
- Deep (Bidirectional) RNNs:
  - Similar to Bidirectional RNNs.
  - Have multiple layers per time step.
- Long short-term memory (next slides)

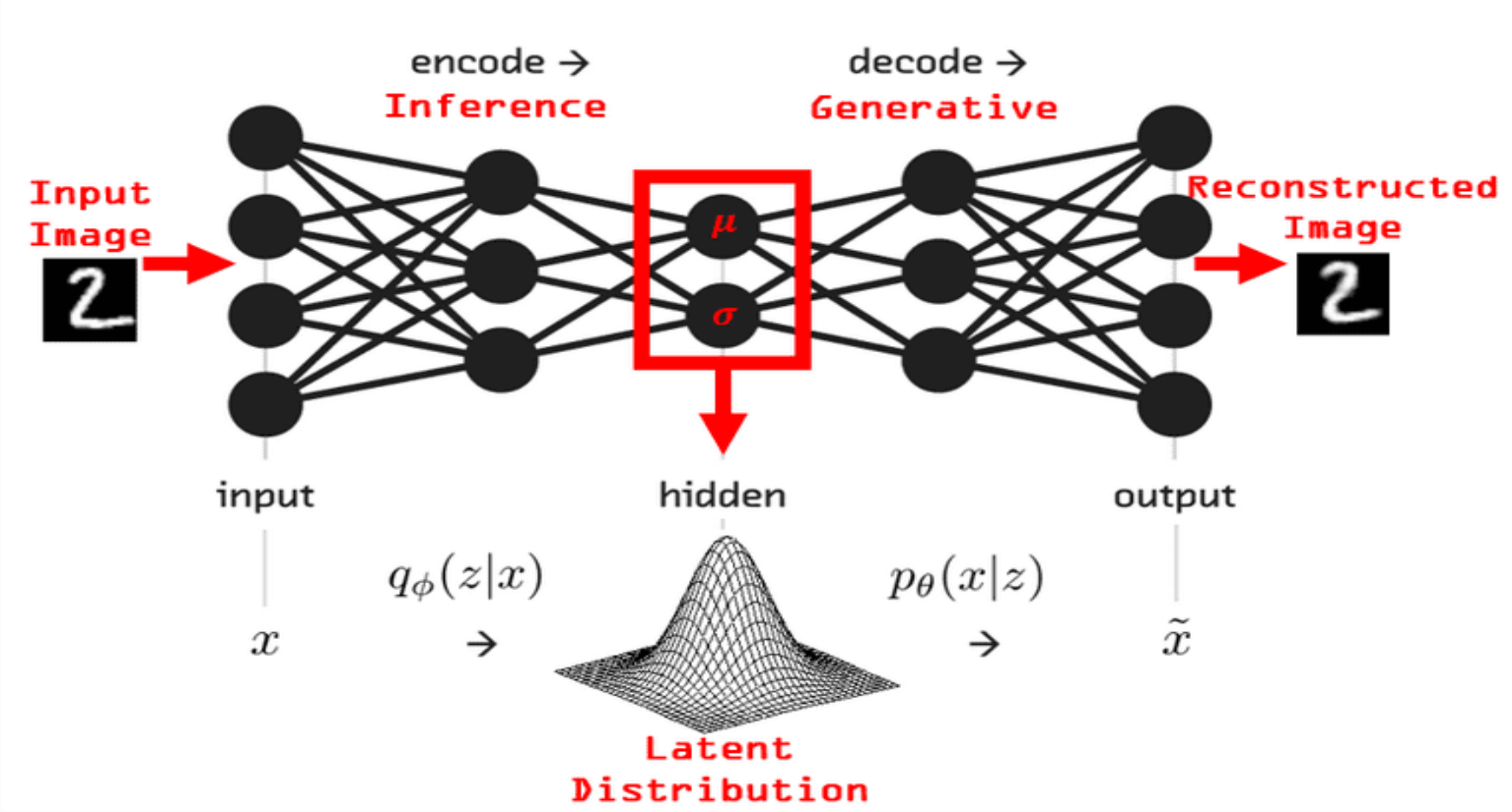






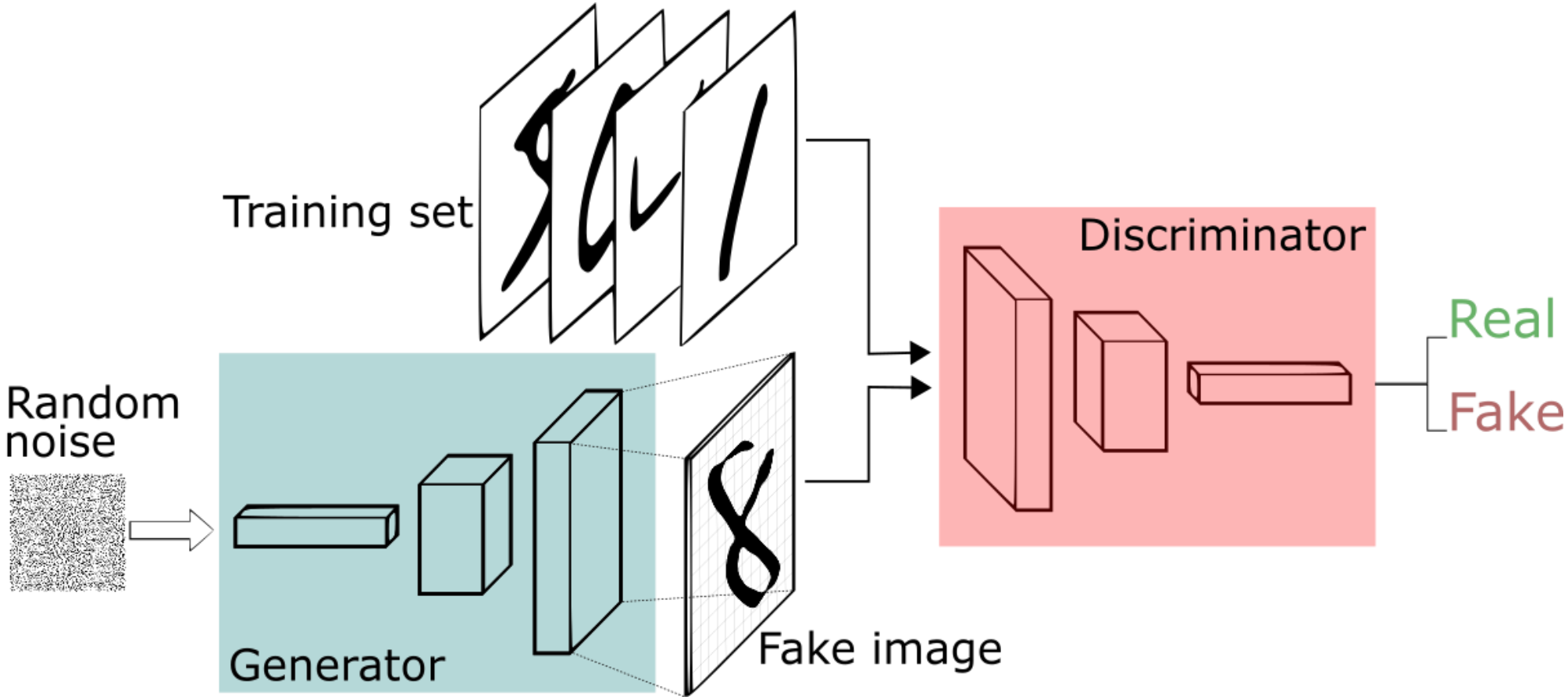
Vanilla DL Architectures

Autoencoders:



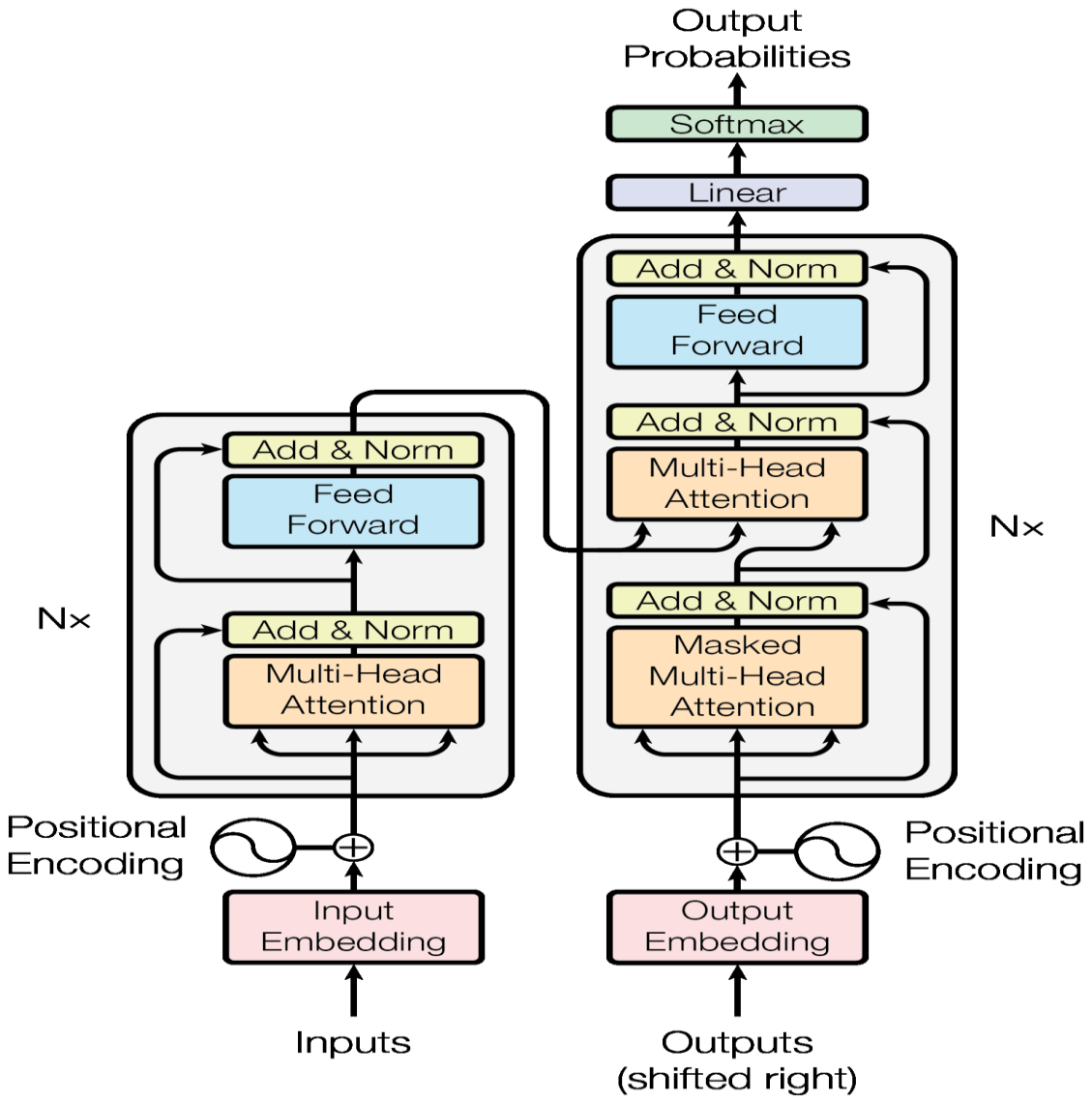
Vanilla DL Architectures

Generative Adversarial Networks (GANs):



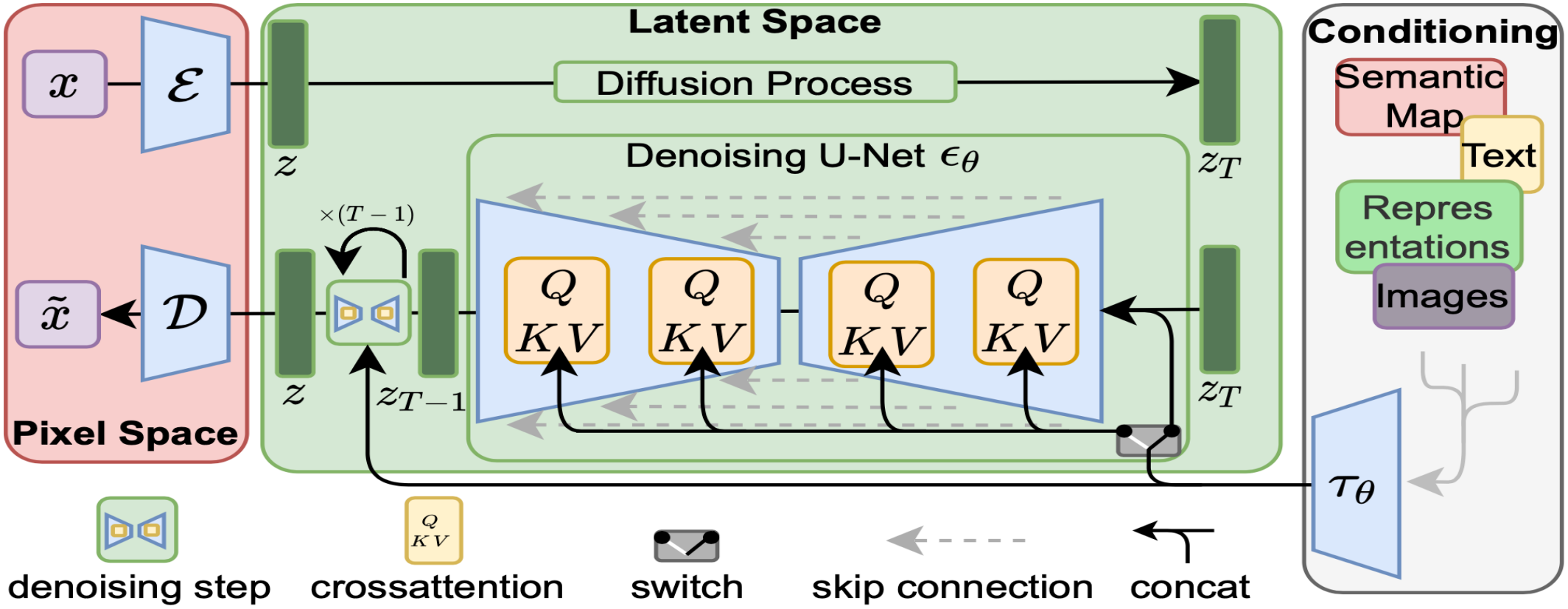
Vanilla DL Architectures

Transformer Networks:



Vanilla DL Architectures

Diffusion Models:



Machine Learning Toolboxes

Popular Machine Learning Toolboxes

- Minimum Requirements

- Some for industry and large scale projects

Popular Machine Learning Courses

- Coursera ML courses (https://www.coursera.org/specializations/machine-learning).
- Udemy ML courses (https://www.udemy.com/topic/machine-learning/).
- Machine Learning Courses from Top-tier Universities (Standford, UC, Canergie Mellon, …).
- Forum – Machine Learning Cơ Bản.
- AI Academy Vn's AI Engineer Training Program.

Thank you!