

Overview

- 1 Artificial Intelligence and Machine Learning
- 2 Machine Learning Capabilities
- 3 Taxonomy of Machine Learning Problems
- 4 Types of Machine Learning Systems
- 5 Urban Applications
- 6 Recent Research at PEER and UMD

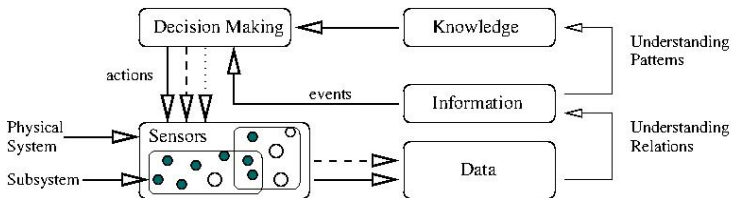
Quick Review

Man and Machine (Traditional View)

Man	Machine
<ul style="list-style-type: none">● Good at formulating solutions to problems.● Can work with incomplete data and information.● Creative.● Reasons logically, but very slow.● Performance is static.● Humans break the rules.	<ul style="list-style-type: none">● Manipulates 0s and 1s.● Very specific abilities.● Requires precise descriptions of problem solving procedures.● Dumb, but very fast.● Performance doubles every 18-24 months.● Machines will follow the rules.

Importance of Sensor Networks

Pathway from sensing and data collection to ... action ... improved performance.



Chain of dependency relationships:

1. improved performance \leftarrow actions
2. actions \leftarrow ability to identify events.
3. identify events \leftarrow data processing
4. data processing \leftarrow types and quality of data
5. types and quality of data \leftarrow sensor design and placement.

Pathway to System Efficiency

We need computational models that:

- Improve **situational awareness** – to understand what is actually happening in a building or city?
- Connect **sensor measurements** to short- and long-term **urban needs** (e.g., decisions on a bus stop; longer term urban planning).
- Capture the **spatial**, **temporal**, and **intensity** aspects of environmental phenomena (e.g., fires, flooding) and their **impact** on natural (e.g., air quality) and **man-made systems** (e.g., transportation networks, food chains).
- **Look ahead** and **forecast future states** of the system?

Artificial Intelligence and Machine Learning

Opportunity: Can use AI/ML to solve problems in completely new ways.

Artificial Intelligence (AI) and Machine Learning (ML)

Technical Implementation (2020, Google, Siemens, IBM)

- AI and ML will be **deeply embedded** in new **software and algorithms**.

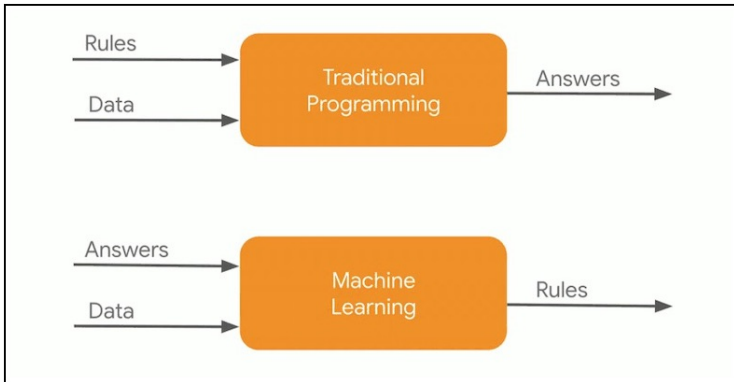
Artificial Intelligence:

- **Knowledge representation** and **reasoning** with ontologies and rules. Semantic graphs. Executable **event-based processing**.

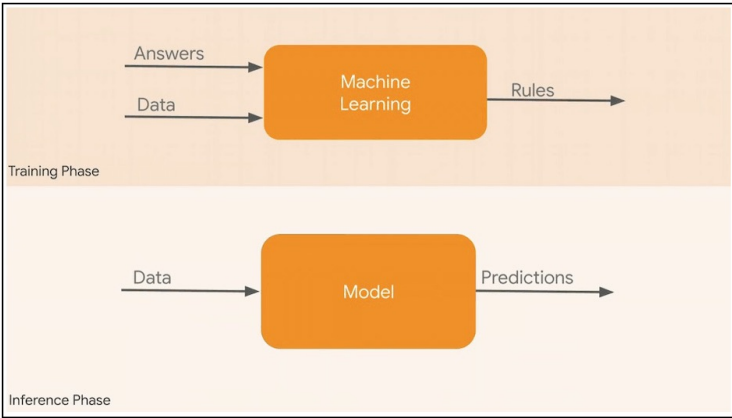
Machine Learning:

- Modern neural networks. Input-to-output prediction.
- Data mining.
- Identify **objects**, **events**, and **anomalies**.
- Learn structure and sequence. **Remember stuff**.

Traditional Programming vs AI-ML Workflow



Traditional Programming vs AI-ML Workflow



Sensor Networks and AI-ML Enabled Decision Making

Dependencies Among Systems in Built Environment:

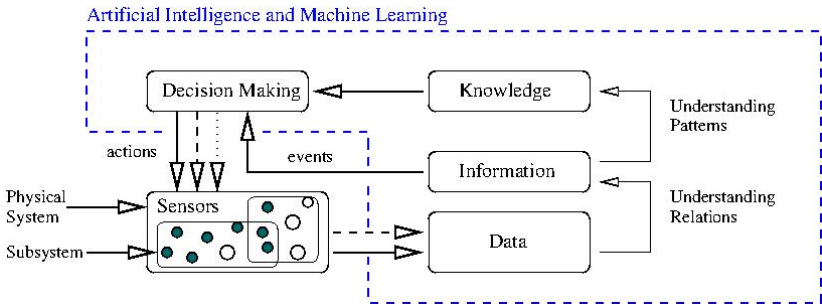


Pathway to Enhanced Situational Awareness/Decision Making:

- Gather and process sensed data.
- Mine data to understand relationships.
- Integrate predictions into decision making framework.

Sensor Networks and AI-ML Enabled Decision Making

Pathway from **sensing** and **data collection** to ... action ... improved performance, now **enabled** by **AI** and **ML** capabilities:



Software Support in Python and Java

Software Support in Python:

- Pandas (for tabular data analysis).
- TensorFlow (open source library for machine learning).
- Keras (neural network library).
- Jupyter Notebook (for web-based authoring of documents).
- Anaconda (packages to perform data science in Python/R).

Software Support in Java:

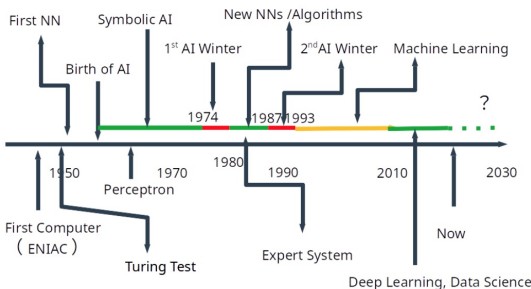
- Apache Jena (for knowledge representation and reasoning).
- Weka (for data mining).
- Deep Learning for Java (DL4J) (for machine learning).

Note: Jupyter → Julia, Python and R.

Machine Learning

Capabilities

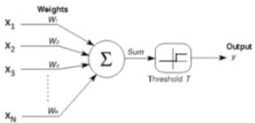
A Brief History



- 1943: First neural networks invented (McCulloch and Pitts)
- 1958-1969: Perceptrons (Rosenblatt, Minsky and Papert).
- 1980s-1990s: CNN, Back Propagation.
- 1990s-2010s: SVMs, decision trees and random forests.
- 2010s: Deep Neural Networks and deep learning.

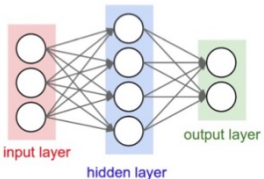
Machine Learning Capabilities (1980-1990)

Expressive Power of a Neural Network

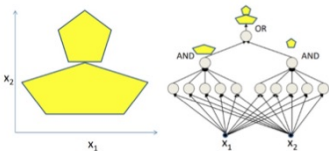
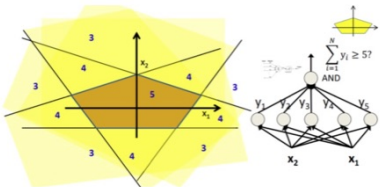


$$y = \begin{cases} 1 & \text{if } \sum_{i=1}^d w_i x_i \geq T \\ 0 & \text{else} \end{cases}$$

Neural Network with Single Hidden Layer



Approximation of Functions / Boolean Logic



Machine Learning Capabilities (1997-2014)

Learning Streams of Text

- Download complete works of Shakespeare (5.4 million characters)
- Train machine to remember text.
- Write new Shakespeare!

```

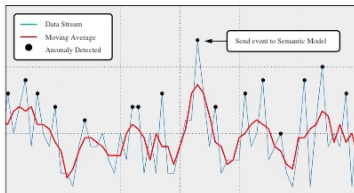
Sample after 1000 Iterations
-----
OTHERS. Allow up, who, Will a moutice,
Resson that boy Skents allien; comn iugius,
And encarsenover heth last;
TARTOR. When no; devires at though me beligo jody?
NANCHELOT. Trom juthar and bur itnot spock,
That as take hava wendisho

Sample after 2000 Iterations
-----
VIBOTARD. Walk this boy and door as am 'stone!
NTRURLEN. Being entainare af Saton.
Els that by much that, I mion's now, who make foil the Kid!!
CLOVE. Why which Ham'd?
QUICKLY. And stand quast of I; this Fi

Sample after 2500 Iterations
-----
FORD. Nay, You're, excoun: and now did yet.
PAROLLES. Take DOTRY
This who is begin Gnoban a bows; but yet which that have be,
Oll, thou stan, and me not ready withered gids
And he in the pleases or pardon us.
Mer. I pray you, how can, and tu

Sample after 3000 Iterations
-----
Rume, 'tigioning, dear?
My las, an hour, chork'd more in my grain,
I an thing forent innoion, madem! I thank I do speak you?
ARRESITKO. but his boines, giving to know; foward to
the distvemail.' The to you well know yea, my lowi.
SECOND CHMNROR. He needs, for the refored are;
  
```

Time Series Anomaly Detection

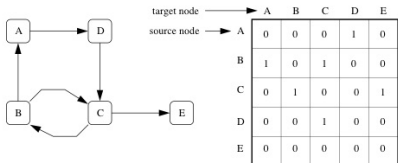


Time Series Prediction



Machine Learning Capabilities (2014-present)

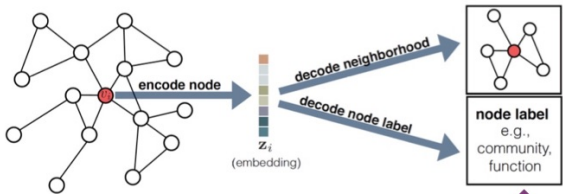
Traditional Approach to Graph Representation



Graph Analysis

- Connectivity / reachability analysis
- Cycle detection
- Traversal problems
- Shortest path problems
- Traceability problems (MBSE)
- Matching problems
- Topological sort problems
-

Graph Embedding Techniques



Graph Analytics

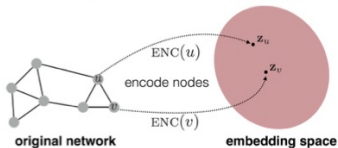
- Node Classification
- Node Clustering
- Anomaly Prediction
- Attribute Prediction
- Link Prediction
- Recommendation
- Etc ...

- Each **node** in the graph is **mapped** to a **low-dimensional space**.
- Goal is to **preserve local linkage structure** (not global structure).
- Each dimension corresponds to a **community** in the network.

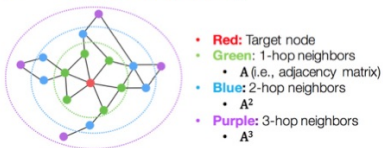
Captures semantics in domain application

Machine Learning Capabilities (2014-present)

Graph-to-Embedding Space Transformation



Graph Reachability



Goal: Design encoder so that similarity in embedding space is closely approximates similarity in original network.

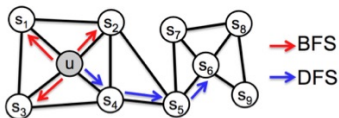
Graph Embedding Vector Design: node2vec, DeepWalk, ...

Node2vec: Combine two strategies:

BFS: Breadth First Search provides a local view of graph neighborhood.

DFS: Depth First Search provides a global view of the neighborhood.

Encoder is just a simple embedding vector lookup.



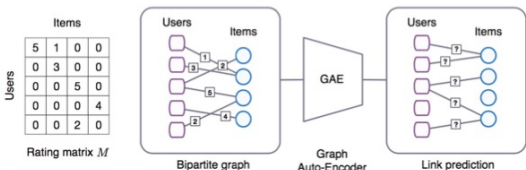
$N_{BFS}(u) = \{s_1, s_2, s_3\}$ Local microscopic view

$N_{DFS}(u) = \{s_4, s_5, s_6\}$ Global macroscopic view

Machine Learning Capabilities (2014-present)

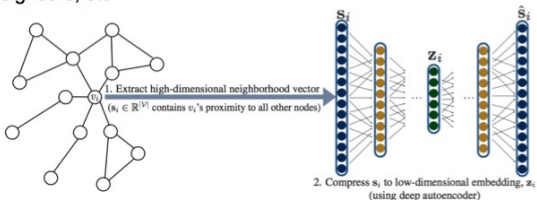
Graph Auto-encoder Link Prediction

Link prediction procedures can be used in new types of system validation / verification.



Deep Graph Auto-encoder Design

Requirements **traceability** needs **arbitrarily large levels of reachability** – first order neighbors, second-order neighbors, etc.

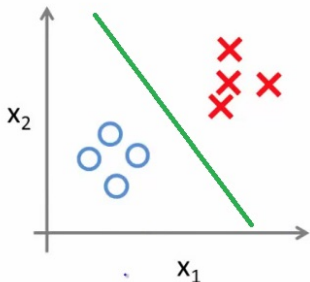


Classification of Machine Learning Problems

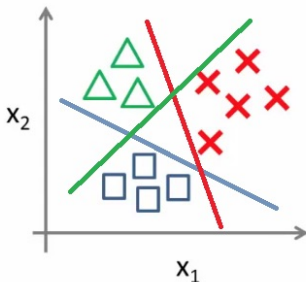
Binary and Multi-Class Classification

Task of **separating elements** of a set into **two (or more) groups** on the basis of a **classification rule** (e.g., shape, color, etc).

Binary classification:



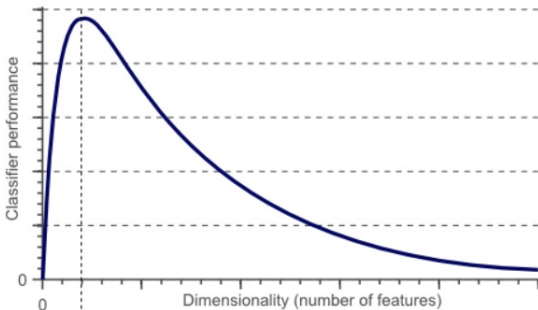
Multi-class classification:



Classification of Machine Learning Problems

Curse of Dimensionality

Machine learning problems are inherently statistical and involve high-dimensional data. **Increases** in the **problem dimensionality**, **decrease** the number of data points available for **classification** in each **dimension**.

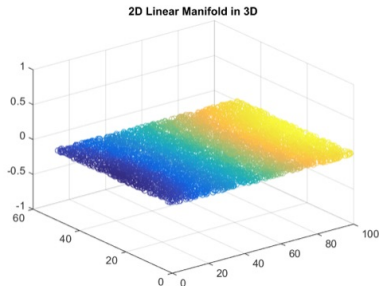
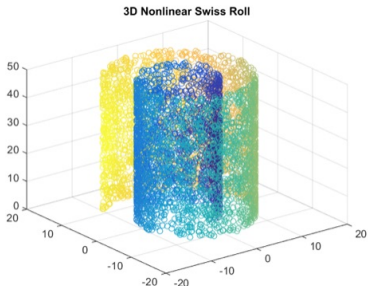


Classification of Machine Learning Problems

Dimensionality Reduction

Strategies of **dimensionality reduction** involve transformation of data to new (lower) dimension in such a way that some of the dimensions can be **discarded without a loss of information**.

Example: Projection of Swiss Roll data in 3D to 2D ...



Classification of Machine Learning Problems

Autoencoders

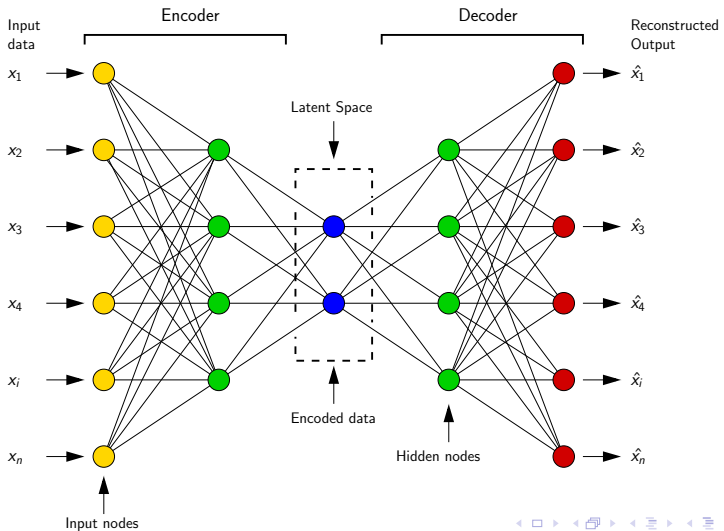
Autoencoder neural networks use unsupervised machine learning algorithms to: (1) find compressed representations of the input data (**encoder**), and (2) reconstruct the original data from the compressed data (**decoder**).

Applications:

- Dimensionality reduction.
- Image processing (compression and denoising).
- Feature extraction; anomaly detection.
- Image generation.
- Sequence-to-sequence translation.
- Recommendation systems.

Classification of Machine Learning Problems

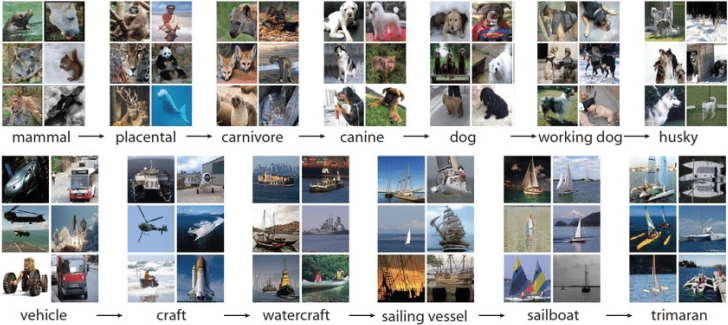
AutoEncoder (Encoder-Decoder-Reconstruction)



Classification of Machine Learning Problems

ImageNet and Deep Learning Capabilities:

- Identify objects in an image.
- 27 high-level categories; 21,800 sub-categories.



ImageNet and Deep Learning

Capabilities (2018):

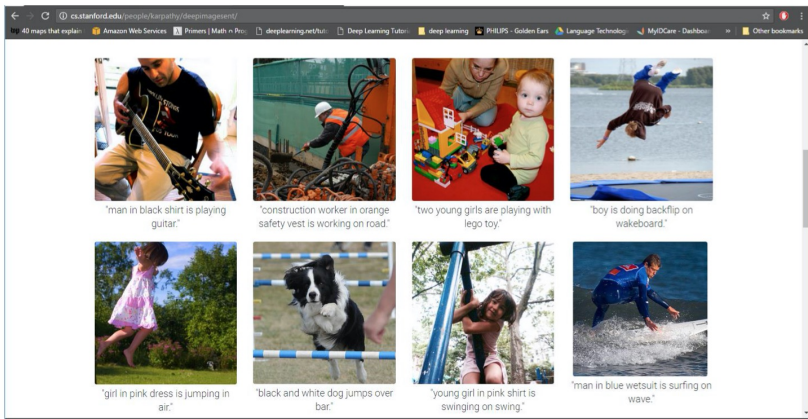
- Identify relationship among multiple objects in a image.

Example. Dog riding skateboard



ImageNet and Deep Learning

Captions generated by a neural network:



The screenshot shows a web browser window with the URL cs.stanford.edu/people/karpathy/deeplearning/. The browser's address bar and tabs are visible at the top. Below the browser window, eight images are displayed in a 2x4 grid, each with a caption generated by a neural network. The captions are:

- "man in black shirt is playing guitar."
- "construction worker in orange safety vest is working on road."
- "two young girls are playing with lego toy."
- "boy is doing backflip on wakeboard."
- "girl in pink dress is jumping in air."
- "black and white dog jumps over bar."
- "young girl in pink shirt is swinging on swing."
- "man in blue wetsuit is surfing on wave."

Machine Learning at Scale

Object-recognition module:

- 24 million nodes.
- 140 million parameters.
- 15 billion connections.

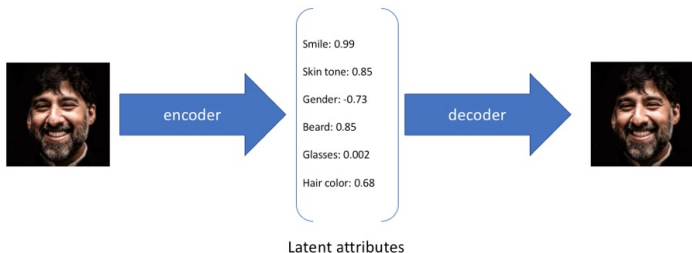
Source: Fei Fei Li, TEDTalk, YouTube 2015.

Classification of Machine Learning Problems

Standard Autoencoders vs. Variational Autoencoders:

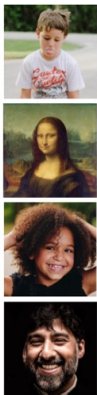
- A **standard autoencoder** outputs a **single value** for each **encoding dimension**.
- **Variational autoencoders** provide a **probability distribution** for each latent attribute.

Example: Single value representations for latent attributes:

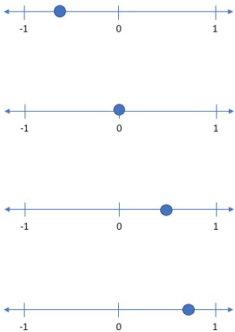


Classification of Machine Learning Problems

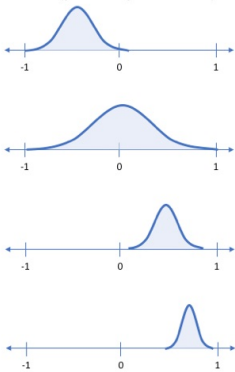
Discrete Value and Probability Distribution: Representations for smile latent attribute:



Smile (discrete value)



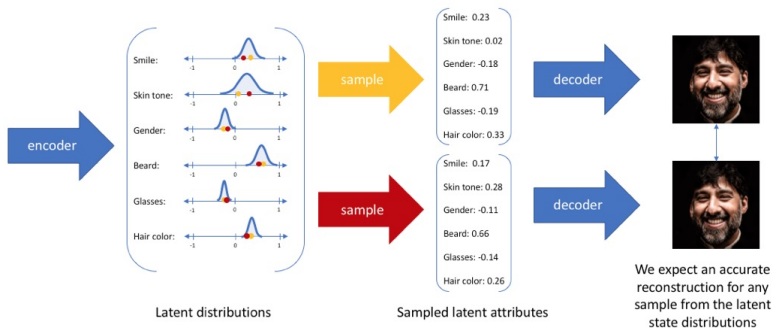
Smile (probability distribution)



vs.

Classification of Machine Learning Problems

Image Reconstruction: sampled from latent distributions ...



Source: Jordan J., Variational Autoencoders, Data Science, March 2018.

Classification of Machine Learning Problems

Transformers

Transformer models (2017) are neural networks that use a technique called **self-attention** to take into account the **context** of **elements in a sequence**, not just the elements themselves.

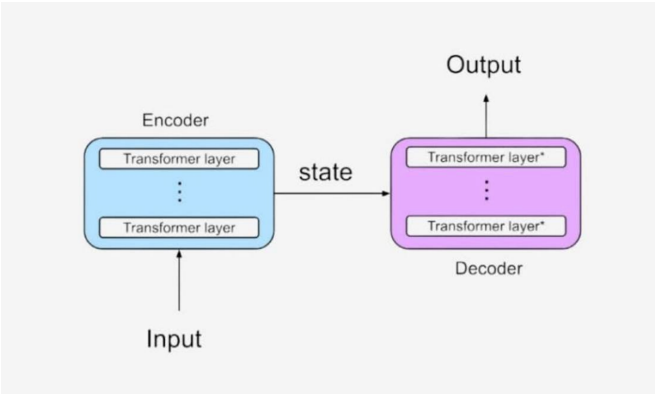
Benefits:

This ability makes transformers ideal for:

- Analyzing sentences/paragraphs of text, and
- Translating languages and understanding complex sentences.

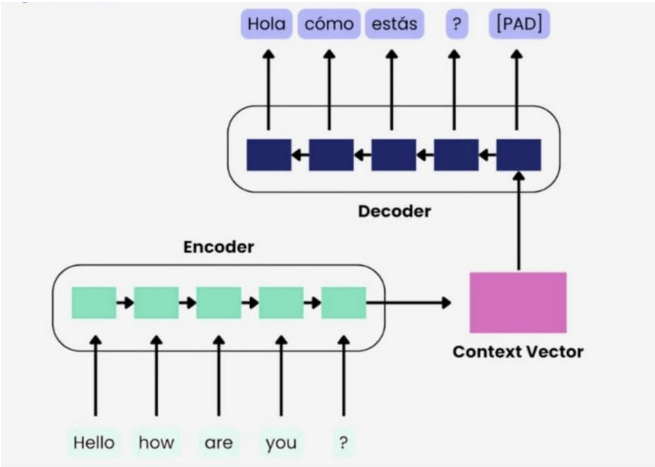
Classification of Machine Learning Problems

Transformer Model:



Classification of Machine Learning Problems

Transformer Neural Networks:



Machine Learning Systems

Types of Machine Learning Systems:

- Supervised machine learning.
- Unsupervised machine learning.
- Semi-supervised machine learning.
- Reinforcement machine learning.

Supervised Machine Learning

Supervised Machine Learning

Learning algorithms are trained with **labeled data** and adjust the model parameters to **minimize** the **discrepancy** between the **computed output** and **desired output**.

Data(x,y):

- x is data, y is the label.

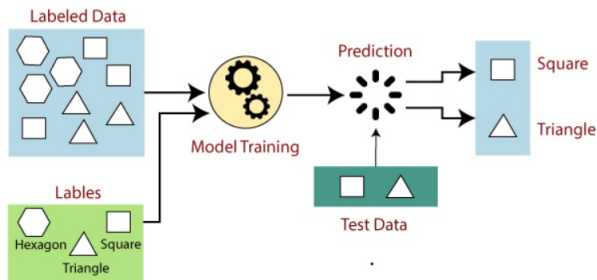
Goal:

- Learn **function** to map $x \rightarrow y$.

Common Algorithms: Regression, classification, naive bayes, object detection, neural networks, random forests, convolution neural networks.

Supervised Machine Learning

Supervised Machine Learning Process and Testing



Challenges:

- Data preparation and pre-processing; avoid unlikely and incomplete data.
- Identifying the right features to train the machine on.

Supervised Machine Learning

Advantages

- Can predict output based on previous experiences.
- Can have an exact idea about the classes of objects.
- Very useful in real-world applications such as fraud detection.

Disadvantages

- Not suitable for solution of complex tasks.
- Domain of expertise is very narrow – cannot predict correct output if test data is different from training dataset.
- Training requires prior knowledge of the classes of objects.
- Manual labeling of a large data set can be very time consuming.

Unsupervised Machine Learning

Unsupervised Machine Learning

Learning algorithms examine the structure of **unlabeled data**, and divide it into **groups** having the **closest features**.

Data(x):

- x is data, no labels.

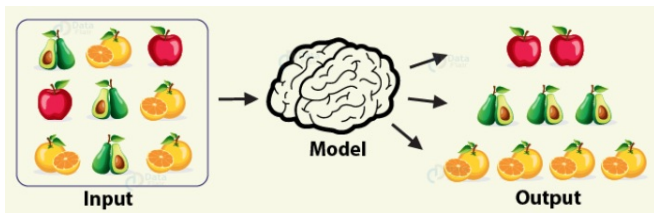
Goal:

- Learn **hidden** or **underlying** structure (or patterns in) of the data.

Common Algorithms: K-means clustering, feature or dimensionality reduction.

Unsupervised Machine Learning

Unsupervised Machine Learning Process



Abilities and Challenges

- No supervision needed.
- Unsupervised learning is closer to human cognitive function – it deduces patterns from a wide variety of application and learns over time.

Unsupervised Machine Learning

Advantages

- Ability for a machine to tackle problems that humans might find insurmountable.
- Ideal for exploring raw and unknown data – training data does not need to be labelled.

Disadvantages

- Lower accuracy of results because the input data is now known and not labeled by people in advance.
- User needs to spend time interpreting and labeling classes/groups which follow classification.

Semi-Supervised Learning

Semi-Supervised Learning

Semi-supervised learning is an approach to machine learning where algorithms use **large amounts** of **unlabeled data** to augment **small amounts** of **labeled data** to **improve predictive accuracy**.

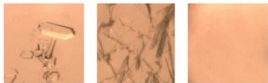
Semi-Supervised Learning in Humans

Concept learning in Children:

- Let $x = \text{animal}$, $y = \text{concept}$ (e.g., cat).
- Parent points to animal and says: cat!
- Children subsequently observe animals by themselves and incrementally refine understanding.

Semi-Supervised Learning

Unlabeled and Labeled Data



0 1 2 3 4 5 6 7 8 9
8 9 0 1 2 3 4 5 6 7



Unlabeled data, X_i

Cheap and abundant !



Human expert/
Special equipment/
Experiment

"Crystal" "Needle" "Empty"

"0" "1" "2" ...

"Sports"
"News"
"Science"
...

Labeled data, Y_i

Expensive and scarce !

Semi-Supervised Learning

Algorithms

- Self-training, generative models, co-training.
- Graph-based algorithms.
- Semi-supervised support vector machines.

Applications

- Speech recognition and analysis.
- Spam detection and filtering.
- Video surveillance.
- 2D and 2D structure prediction.

Semi-Supervised Learning

Advantages

- Provides the benefits of both unsupervised and supervised learning while avoiding the challenge of finding large amounts of labeled data.

Disadvantages

- Cannot provide significant benefits over supervised learning unless one is **absolutely sure** that an assumption holds on the relationship between labels and the unlabeled data distribution.

Mathematically, we need:

$$p(x, y) = p(y)p(x|y), \quad (1)$$

where $p(x|y)$ is an identifiable mixture model.

Reinforcement Learning

Reinforcement Learning

Reinforcement learning algorithms use trial-and-error procedures to determine which action can provide the greatest reward.

Data: state-action pairs.

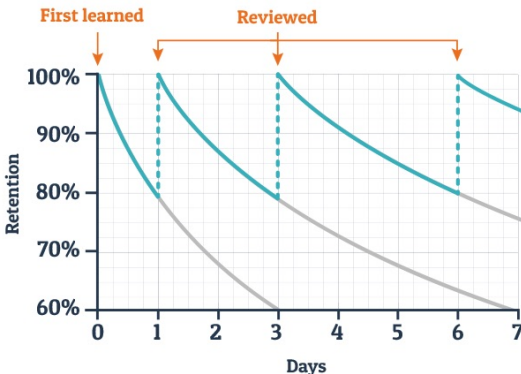
Goal: Maximize future rewards over many time steps.

Examples: Taking actions to enhance survival/performance in gaming, robotics, optimization of operations for industrial machinery.

Reinforcement Learning

Using Reinforcement to Improve Memory Retention

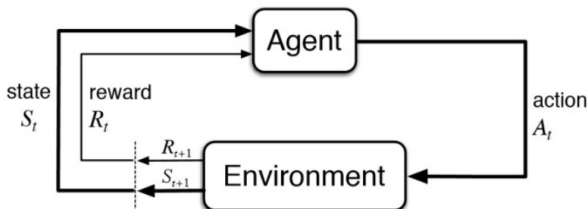
Typical Forgetting Curve for Newly Learned Information



Reinforcement Learning

Reinforcement Learning Process

In technical terms, reinforcement learning is a process in which a **software agent** makes **observations** and **takes actions** within an **environment**, and in return, it **receives rewards**.



The **main objective** is to **maximize long-term rewards**.

Reinforcement Learning

Definitions:

- **Environment:** Physical world in which the agent is operating.
- **State:** Current situation of the agent.
- **Reward:** Feedback from the environment.
- **Policy:** Method of map agent's state to actions.
- **Value:** Future reward that an agent would receive by taking an action in a particular state.

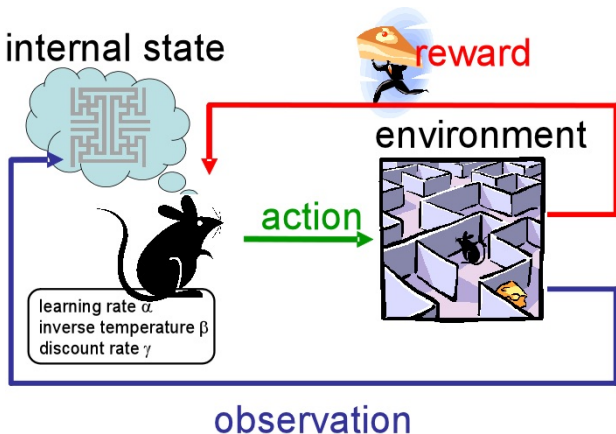
Note:

- These conditions may not always be present in real-world applications.

More Details: See Technical Tutorial on RL by Pieter Abbeel and John Schulman at UC Berkeley.

Reinforcement Learning

Simple Example: Mouse Searches Maze to find Cheese



Reinforcement Learning

Advantages

- Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.
- Errors can be corrected during the learning process.
- Learning process is very similar to humans, but it can often outperform humans ...

Disadvantages

- Not suitable for solving simple problems.
- Reinforcement learning requires lots of data and computation.
- Assumes incorrectly that the World follows a Markovian model, described in terms of sequences of possible events in which the probability of each event depends only on the current state.

Urban Applications

Opportunities for Machine Learning

Machine Learning Opportunities:

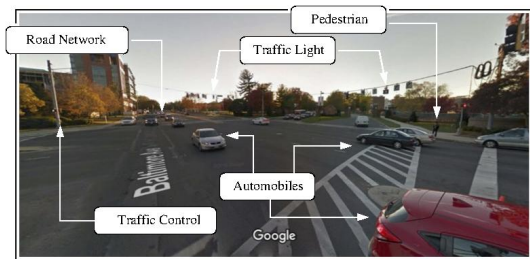
- Predicting system response and performance.
- Interpreting data and formulating models to predict component and subsystem-level properties.
- Information retrieval from images and text.
- Recognizing patterns in streams of sensed data.

Economic Considerations:

- **Urban infrastructure** is **permanent/semi-permanent** and **very expensive** to build and maintain.
- Prioritize **improvements** to **efficiency** by identifying and **removing bottlenecks** in performance.
- Use AI-ML for **design of actions** that **enhance** behavior/**system performance**.

Small Scale: Traffic Intersection at UMD

Goal. How to traverse a traffic intersection safely and without causing an accident?

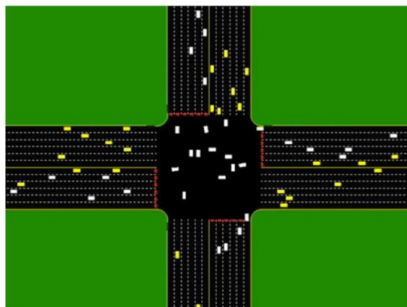


Required Capability. Observe, evaluate, reason, take actions.

Challenges. Multiple types of data, event-driven behavior, dynamic, time critical. Too much traffic congestion.

Self-Driving Cars

Goal. Improve performance by removing bottlenecks → no human driver; no traffic lights.



Remark: 95% of the requirements are for the system software.

Source: ISR visitor from GM Research.

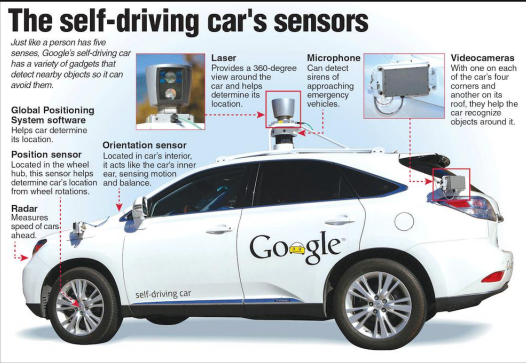
Remark: Tesla will produce self-driving cars by 2016.

Source: Elon Musk.

Stop signs and traffic lights are replaced by mechanisms for vehicle-to-vehicle communication (Adapted from <http://citylab.com>).

Google Self-Driving Car

Essentially: A network of sensors and computers on wheels.



Today: Modern automobiles → 100 million lines of code.

Tomorrow: Self-driving automobiles → 300 million lines of code.

Navigating a Busy Traffic Intersection

Step-by-Step Procedure:

How the car operates

- 1 Any object the vehicle's sensors spot is interpreted by software to determine if it's a pedestrian, cyclist, vehicle or something else.
- 2 Using what it's learned from previous driving, the software makes predictions about what objects will do next.
- 3 The software analyzes the information to decide whether it is safe to accelerate, turn or hit the brakes.

Source: Google
Graphic: Tribune News Service



How the car sees the world

This computerized image is what Google researchers monitoring sensor data see as they ride in the vehicle.

- Other vehicle
- Pedestrian
- Cyclist
- Objects that warrant caution
- A crosswalk, indicating the car needs to stop
- A traffic signal, warning of upcoming railroad tracks
- Path where Google's car intends to go

- Identify various kinds of objects (e.g., vehicles, crosswalk).
- Predict what objects will do next.
- Conduct safety assessment.
- Take action.

Google DeepMind (2018-2020)

Teach Self-Driving Cars to Navigate a City without a Map



Test Cities: London, Paris, New York.

ML Research at PEER

PEER Hub ImageNet (2018): Classification of Structural Engineering images:

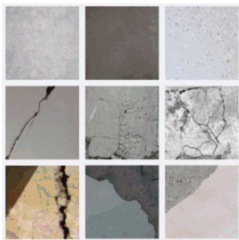


Fig 1a Pixel Level Samples



Fig 1b Object level Samples



Fig 1c Structure Level Samples

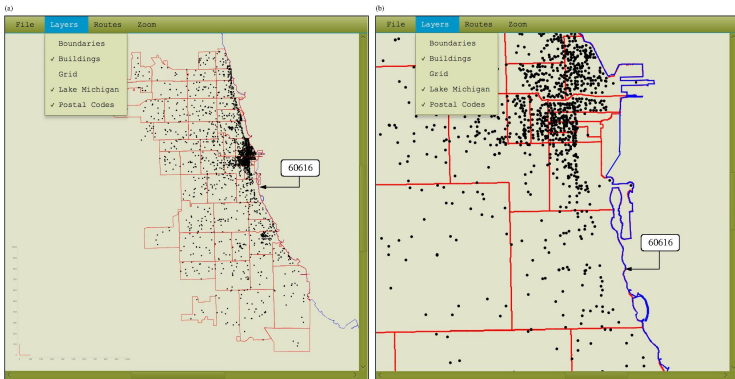
Source:

<https://apps.peer.berkeley.edu/phichallenge/detection-tasks/>

Preliminary Research at UMD

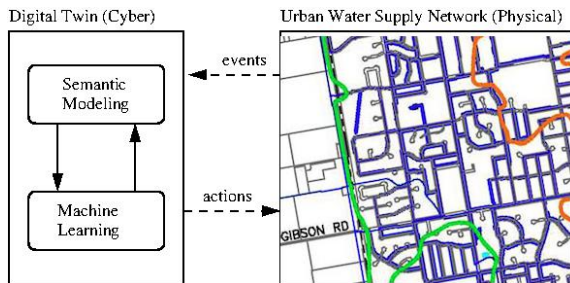
Large Scale: Management of City Operations

Case Study A (2019): Mine publically available data to understand Energy Consumption in 2,500 Buildings in Chicago.



Large Scale: Management of City Operations

Case Study B (2020): Can we teach a machine to understand the structure and behavior of water supply networks?



Reference: Coelho M., et al., Teaching Machines to Understand Urban Networks, ICONS 2020.

References

- Austin M.A., Delgoshaei P., Coelho M. and Heidarinejad M. , Architecting Smart City Digital Twins: Combined Semantic Model and Machine Learning Approach, Journal of Management in Engineering, ASCE, Volume 36, Issue 4, July, 2020.
- Coelho M., and Austin M.A. , Teaching Machines to Understand Urban Networks, The Fifteenth International Conference on Systems (ICONS 2020), Lisbon, Portugal, February 23-27, 2020, pp. 37-42.
- Bhiksha R., Introduction to Neural Networks, Lisbon Machine Learning School, June, 2018.
- Jordan J., Variational Autoencoders, Data Science, March 2018.
- Lu T., Fundamental Limitations of Semi-Supervised Learning, MS Thesis in Mathematics in Computer Science, University of Waterloo, Canada, 2009.
- Van Engelen J.E., and Hoos H.H., A Survey on Semi-Supervised Learning, Machine Learning, Vol. 109, 2020, pp. 373-440.