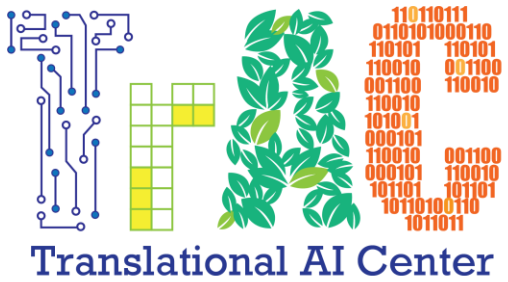
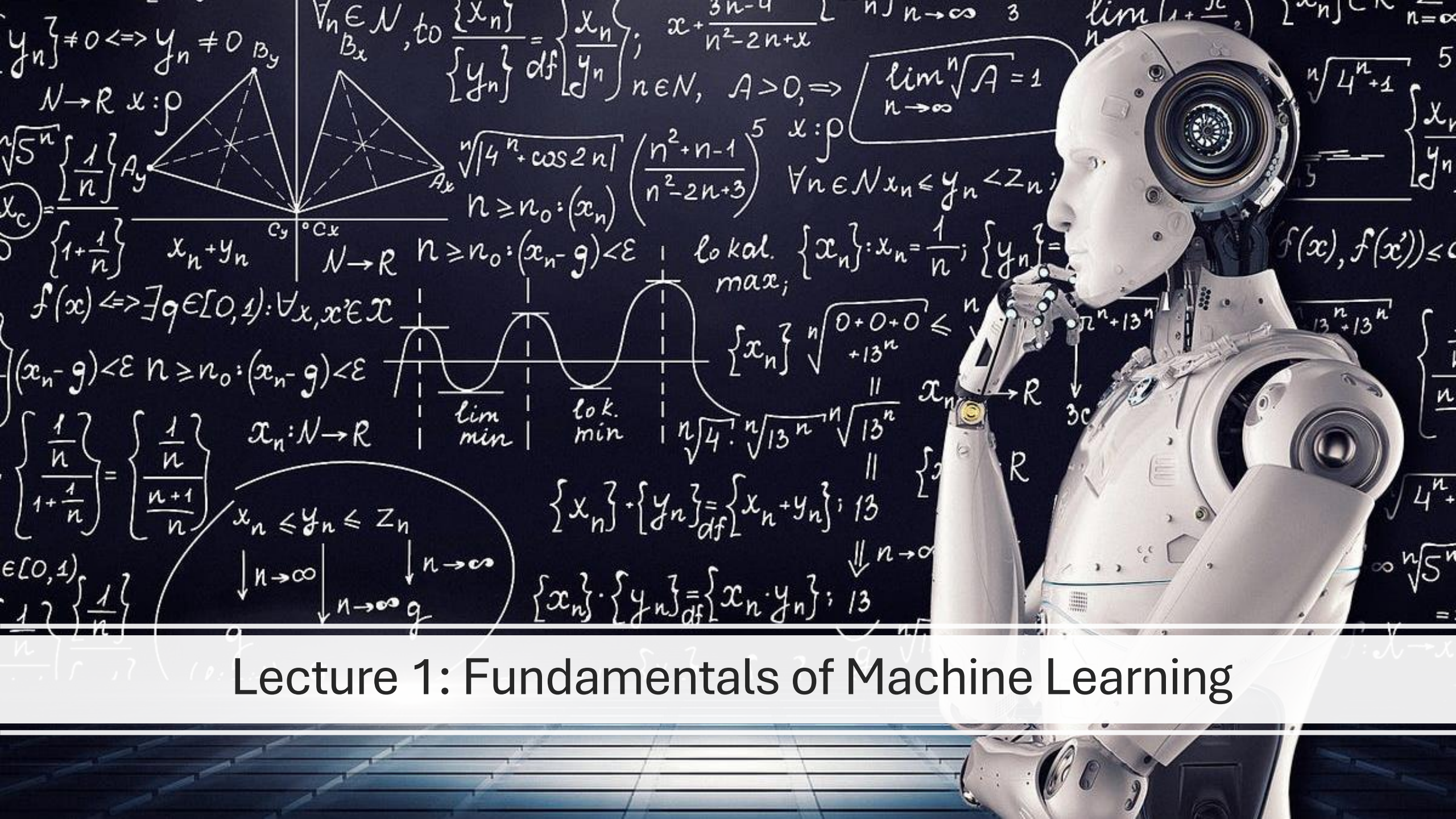


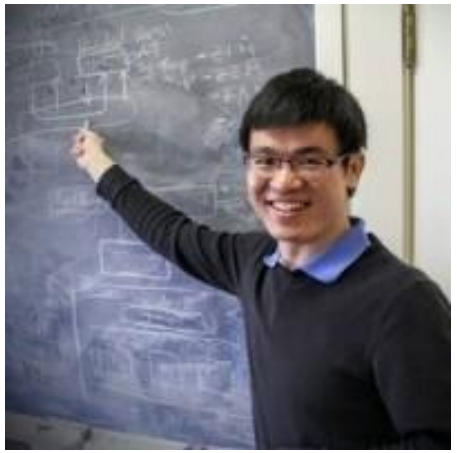

# Bootcamp 2024 – “Deep Dive into AI”



# Lecture 1: Fundamentals of Machine Learning

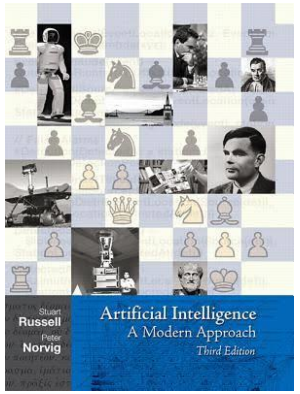
# Information

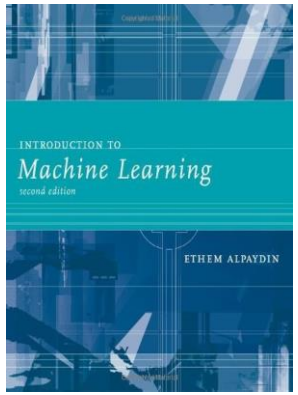
- Instructor: Zhanhong Jiang (zhjiang@iastate.edu), PhD
  - Data scientist at TrAC
  - Research areas: decentralized learning/optimization and reinforcement learning, deep learning applications to cyber physical systems
- Format
  - 3 hours split into 3 sessions
  - Each session (~1 h) is followed by 5 min break
  - Not too much coding in these sessions, but with small pieces of sample code for illustrations

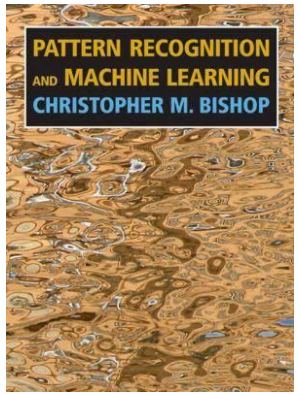


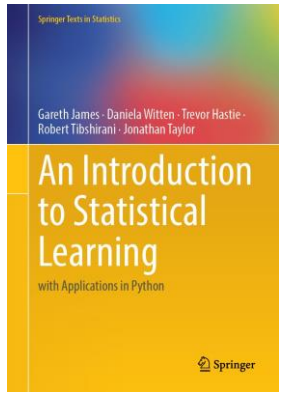
# Resources

- Artificial Intelligence: A Modern Approach (3rd Edition), Russell and Norvig. Prentice Hall, 2009
- Introduction to Machine Learning (2nd Edition), Alpaydin. MIT Press, 2010
- Pattern Recognition and Machine Learning. Christopher Bishop. Springer, 2006 (**available online**)
- An Introduction to Statistical Learning: with Applications in Python. Gareth James et al. Springer, 2023



# What we expect for you

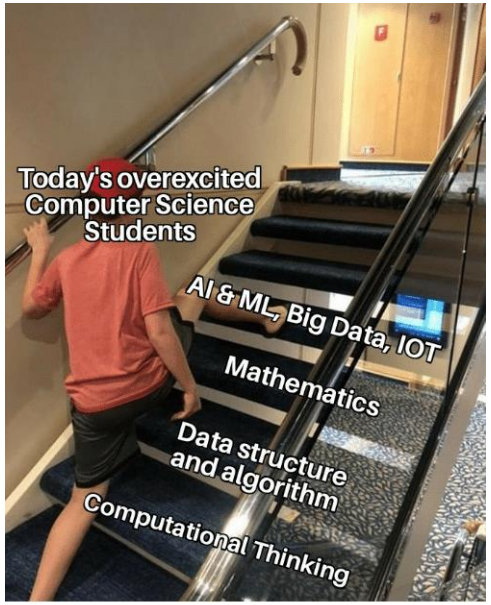
- You know basic concepts of machine learning
- You know a few basic algorithms/models of machine learning
- You initially know how to leverage machine learning to solve your own problems

# Outline

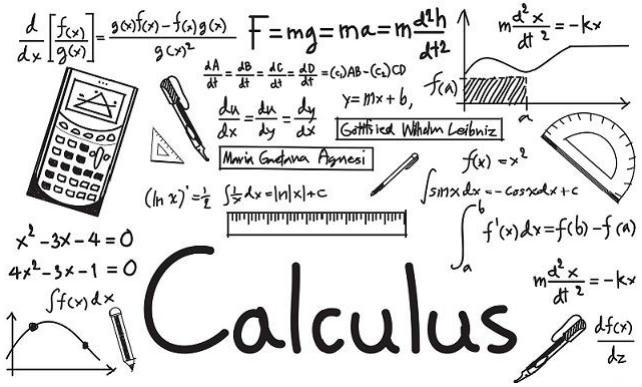
- Session 1: What is machine learning (ML)
- Session 2: Different types of ML
- Session 3: How to frame a learning problem

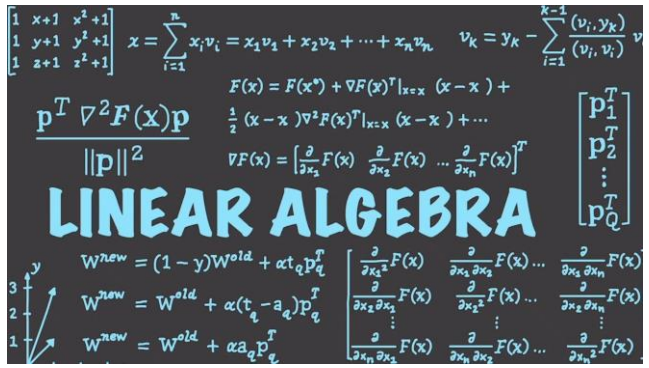
# Notes

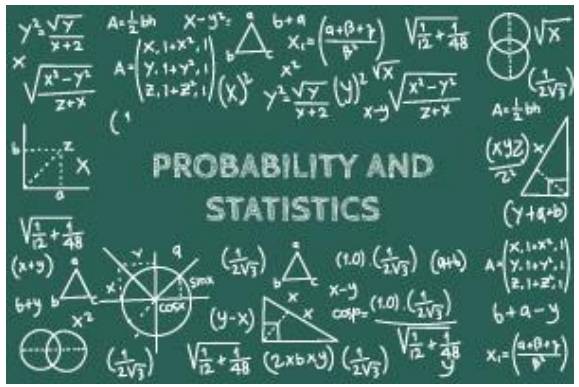
- We are not going to cover much math in all sessions
- Math is critical to understand AI/machine learning
  - Calculus
  - Linear algebra
  - Probability & Statistics
  - Optimization theory
  - …



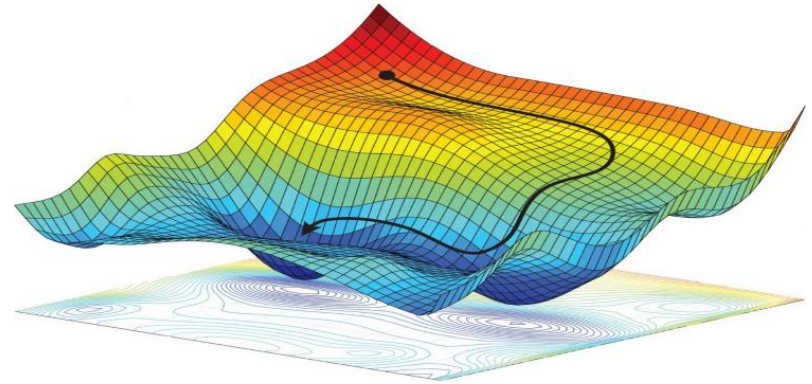
Reality is often disappointing




OPTIMIZATION



# Outline

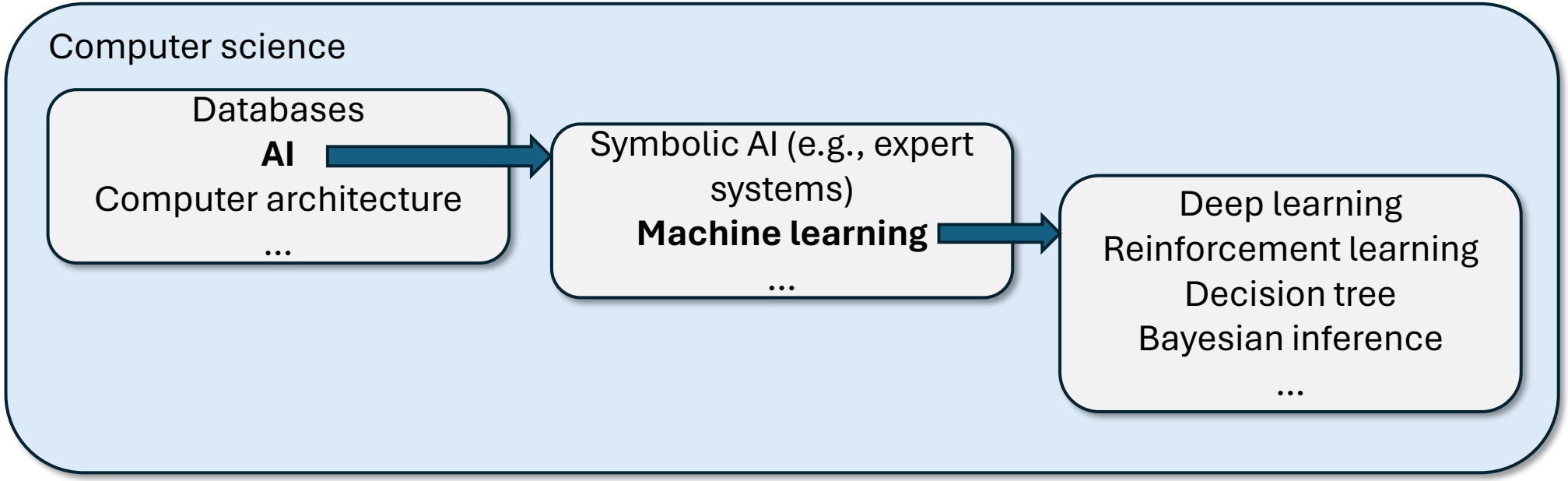
- Session 1: What is machine learning (ML)
- Session 2: Different types of ML
- Session 3: How to frame a learning problem

# AI and machine learning

- Artificial intelligence (AI) is a branch of computer science that uses techniques and algorithms to mimic human intelligence
- Machine learning (ML) is one of several AI techniques for sophisticated tasks

# AI and machine learning

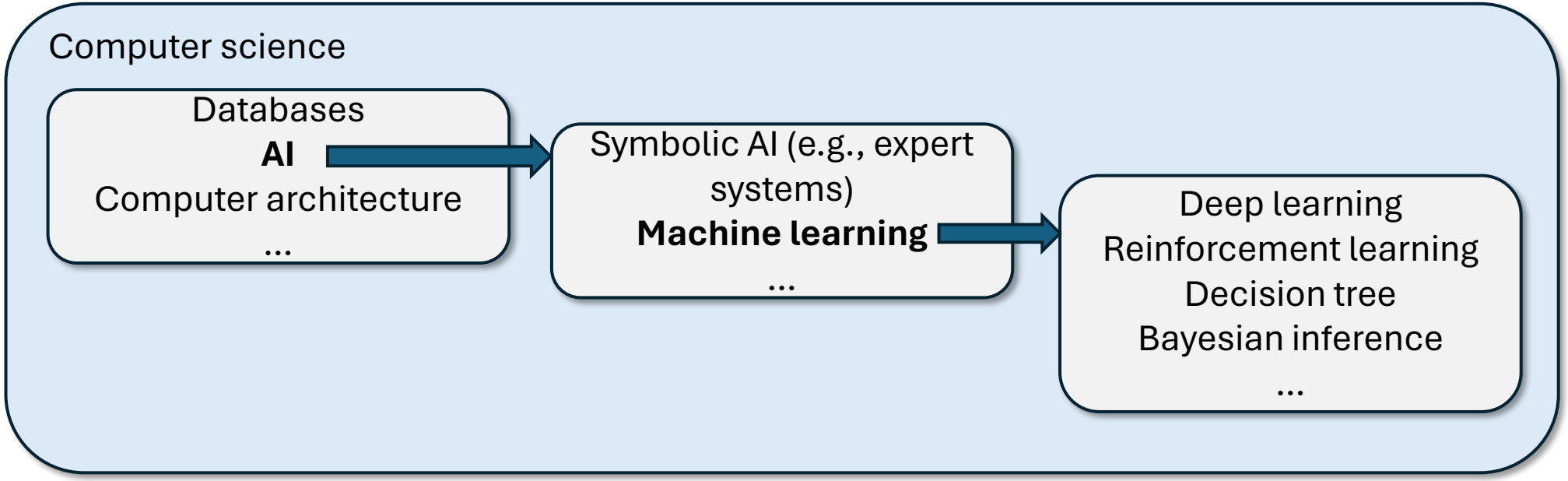
- Artificial intelligence (AI) is a branch of computer science that uses techniques and algorithms to mimic human intelligence
- Machine learning (ML) is one of several AI techniques for sophisticated tasks



# AI and machine learning

- Artificial intelligence (AI) is a branch of computer science that use
- Mac
  soph

**When mentioning AI nowadays, mostly it is ML**



# What is machine learning

Machines are taking over!

Traditionally, machines are hardware, while machine learning is software

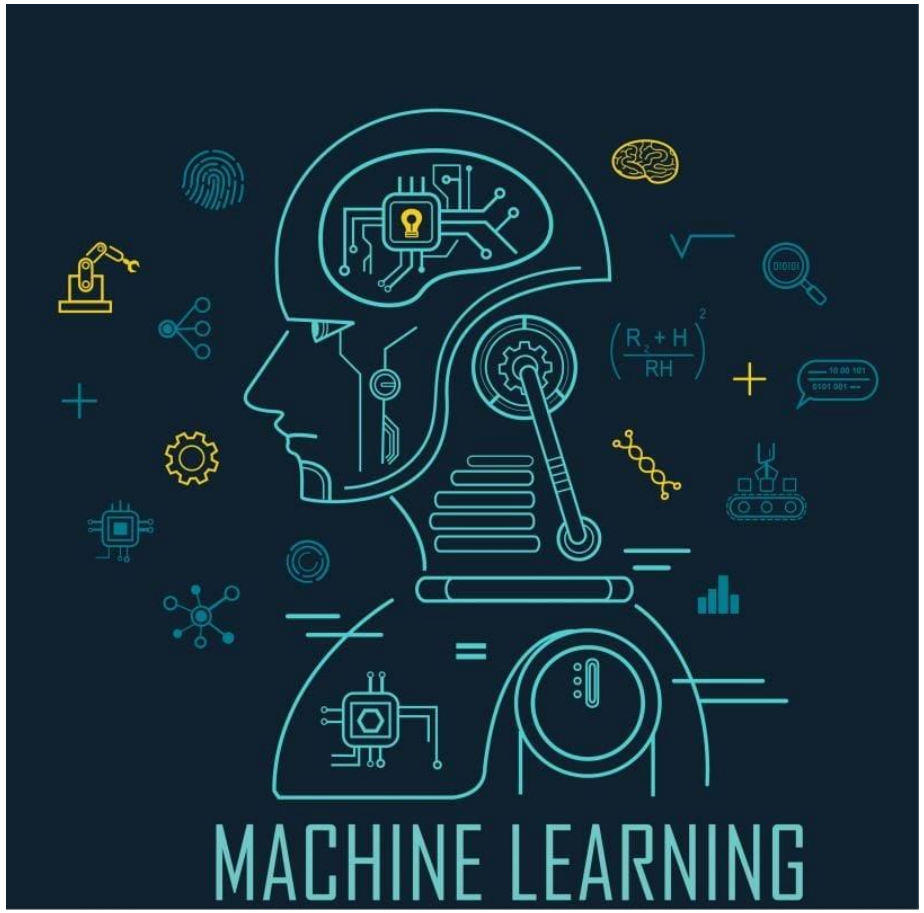



Image credit: Built In

# What is machine learning

"Learning is any process by which a system improves performance from experience." – **Herbert Simon**

Herbert Simon

Definition by **Tom Mitchell** (1997):

Machine learning is the study of algorithms that

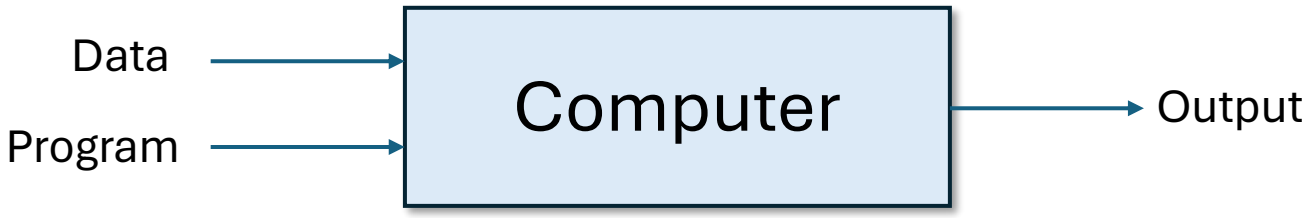
- improves their performance $P$
- at some task $T$
- with experience $E$

Example: $T$ (playing checkers game), $E$ (the experience of playing thousands of games), $P$ (the fraction of games it wins against human opponents)

Tom Mitchell

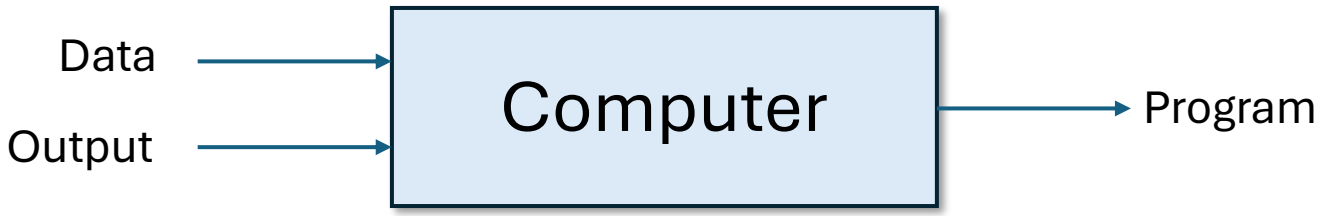
# What is machine learning

## Traditional AI techniques




**Static:** hard-coded set of steps and scenarios
**Rule based:** expert knowledge
**No generalization:** handling special cases is difficult

## Machine Learning

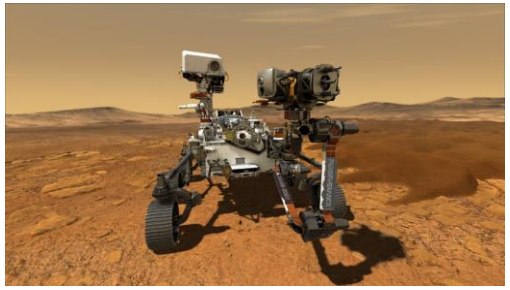


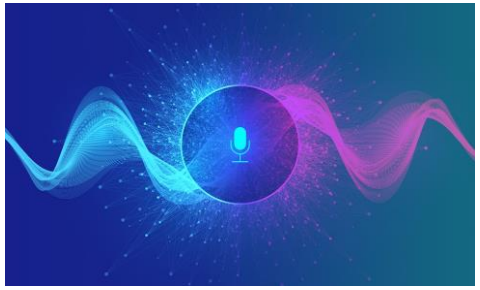

**Dynamic:** evolves with data, finds new patterns
**Data-driven:** discovers knowledge
**Generalization:** adapts to new situations and special cases

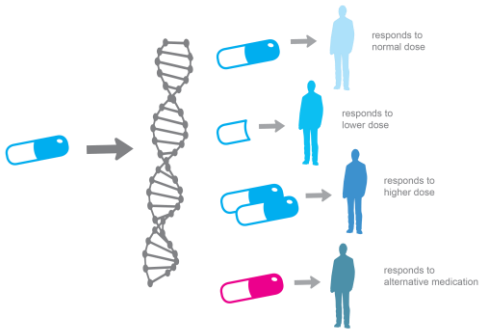
# When to use machine learning

ML is used when:

- When human expertise does not exist (navigating on Mars)
- Humans can’t explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amount of data (genomics)
- …



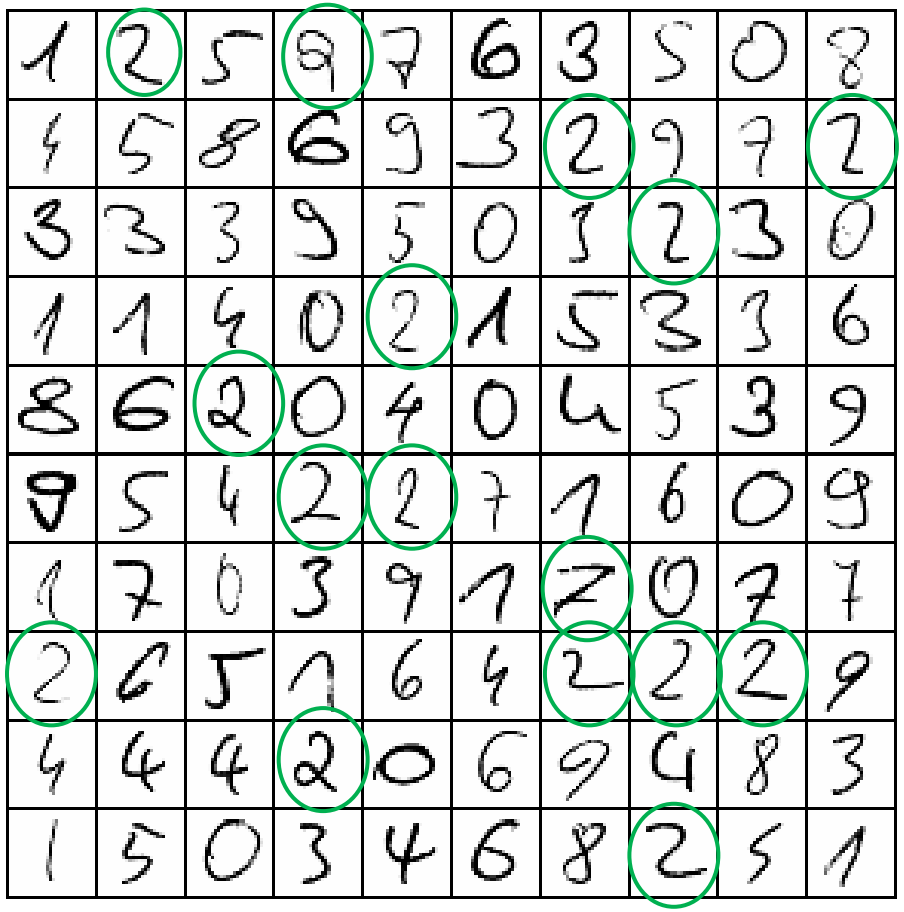





ML is not always useful: no need to use it for calculating payroll

# A classic example

- Handwritten digit recognition
- It is hard to say what makes a 2



# More examples

- Recognizing patterns
  - Facial identifies or expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns
  - Images or motion sequences
- Detecting anomalies
  - Unusual credit card transactions
  - Unusual sensor readings in a nuclear power plant
- Prediction
  - Future stock prices or weather forecast

# State-of-the-art applications of Machine Learning

# Autonomous cars

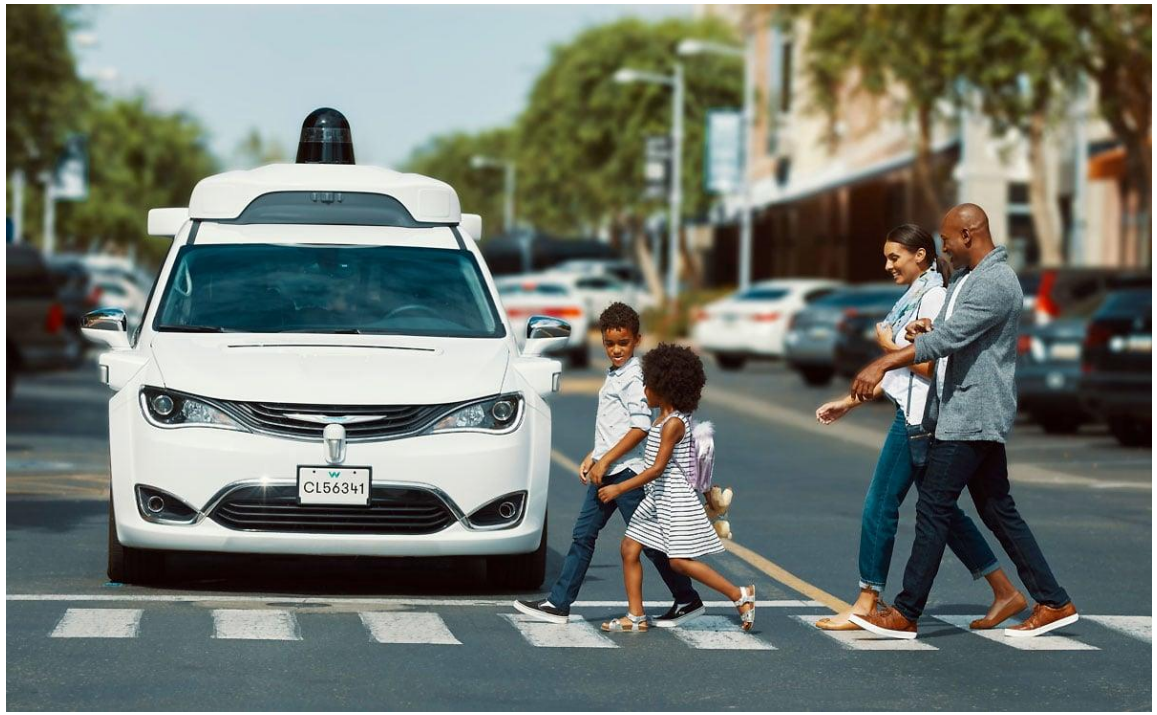



Image credit: Automotive News

Image credit: BM

ML algorithms help significantly in land detection, object and human identification, etc.

# Autonomous cars

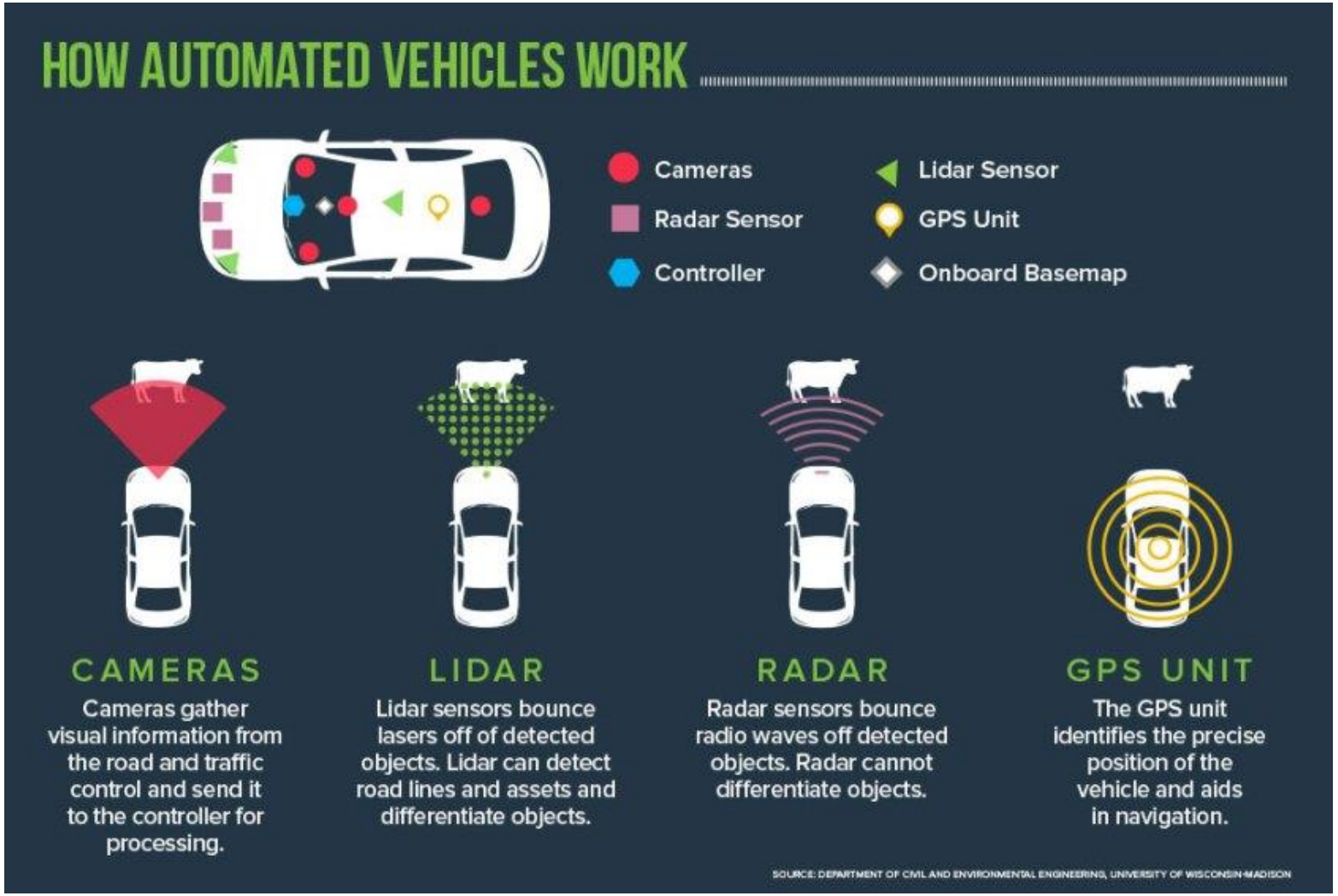



Image credit: GT

# Deep Learning in the headlines

BUSINESS NEWS

MIT Technology Review

## Is Google Cornering the Market on Deep Learning?

A cutting-edge corner of science is being wooed by Silicon Valley, to the dismay of some academics.

By Antonio Regalado on January 29, 2014

How much are a dozen deep-learning researchers worth? Apparently, more than $400 million.

This week, Google reportedly paid that much to acquire DeepMind Technologies, a startup based in


BloombergBusinessweek
Technology

Acquisitions

## The Race to Buy the Human Brains Behind Deep Learning Machines

By Ashlee Vance | January 27, 2014

intelligence projects. "DeepMind is bona fide in terms of its research capabilities and depth," says Peter Lee, who heads Microsoft Research.

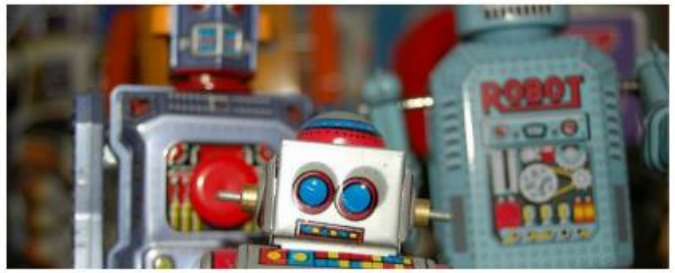
According to Lee, Microsoft, Facebook (FB), and Google find themselves in a battle for deep learning talent. Microsoft has gone from four full-time deep learning experts to 70 in the past three years. "We would have more if the talent was there to

WIRED GEAR SCIENCE ENTERTAINMENT BUSINESS SECURITY DESIG

INNOVATION INSIGHTS | community content | featured

## Deep Learning's Role in the Age of Robots

BY JULIAN GREEN, JETPAC 05.02.14 2:56 PM




# Face recognition

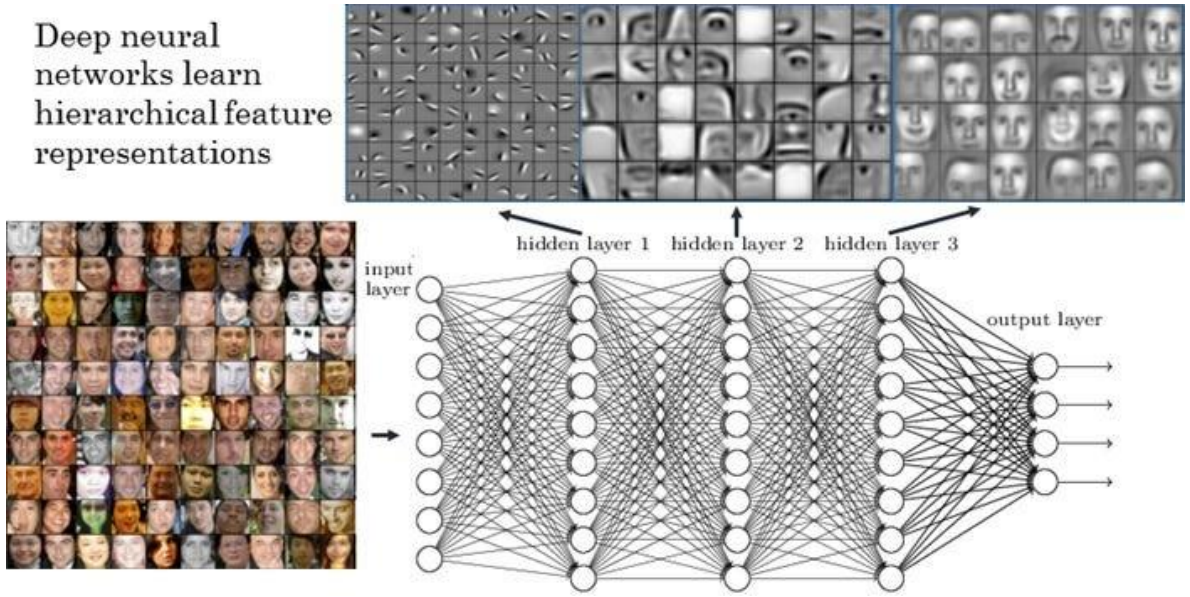



Image credit: Medium

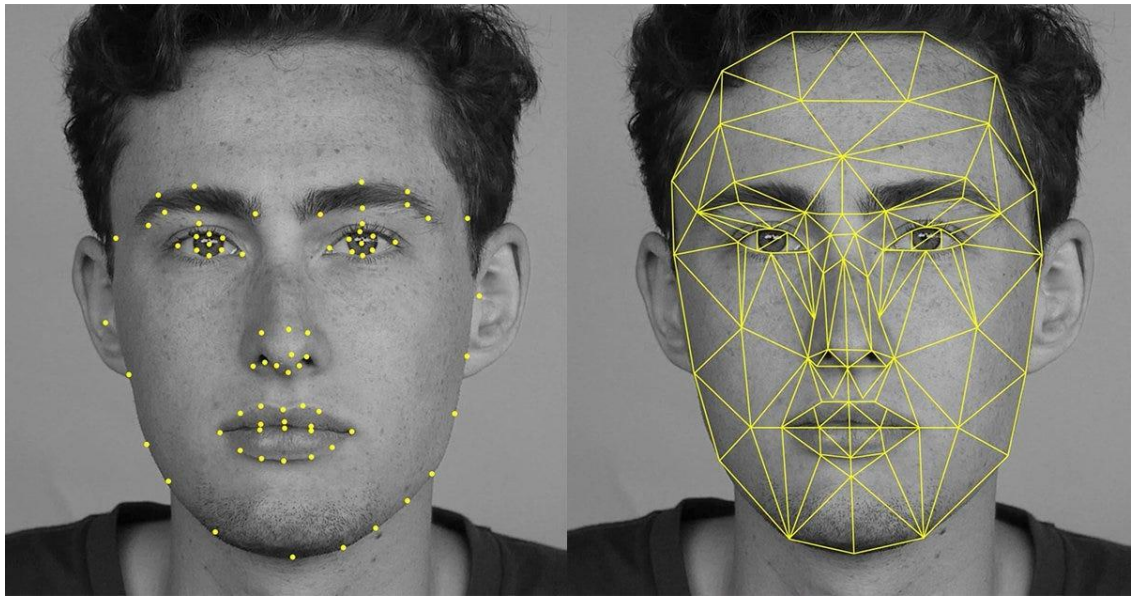


Image credit: Medium

# Multi-object detection

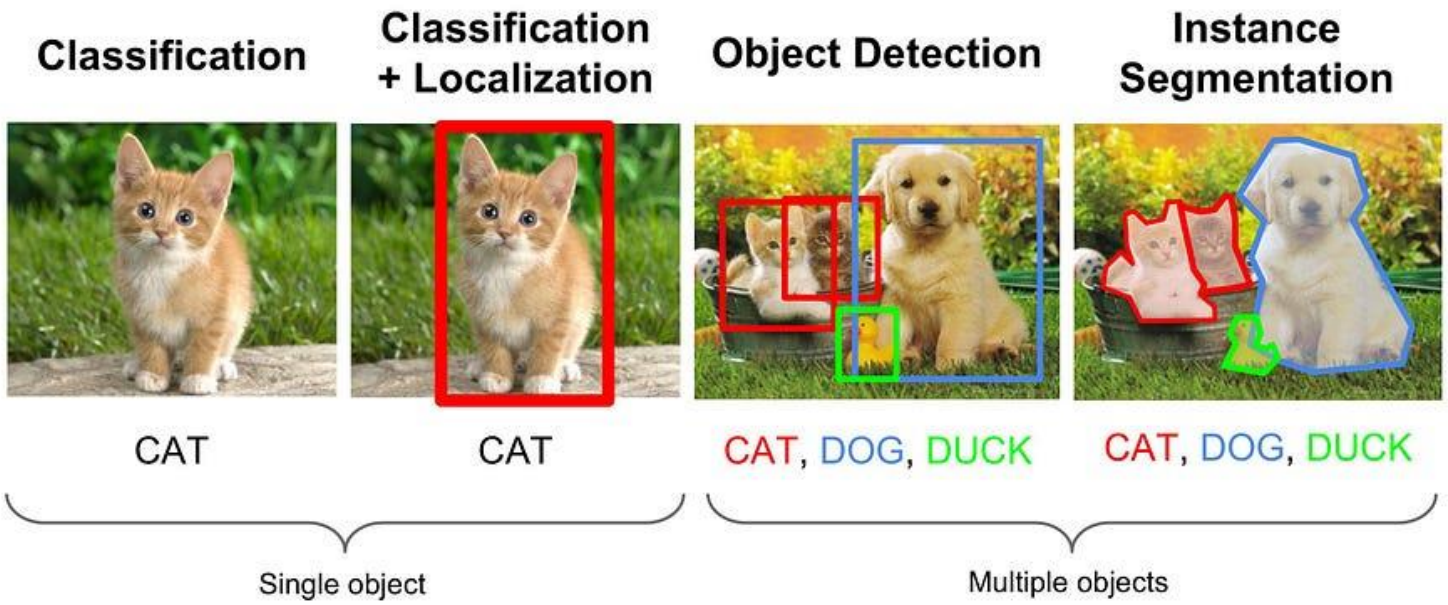



Image credit: Medium

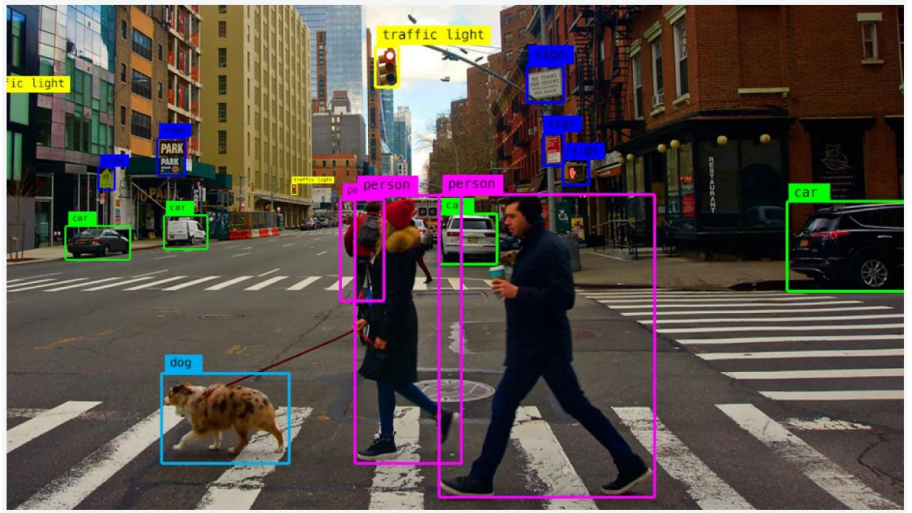



Image credit: Augmented AI

# Recommendation systems

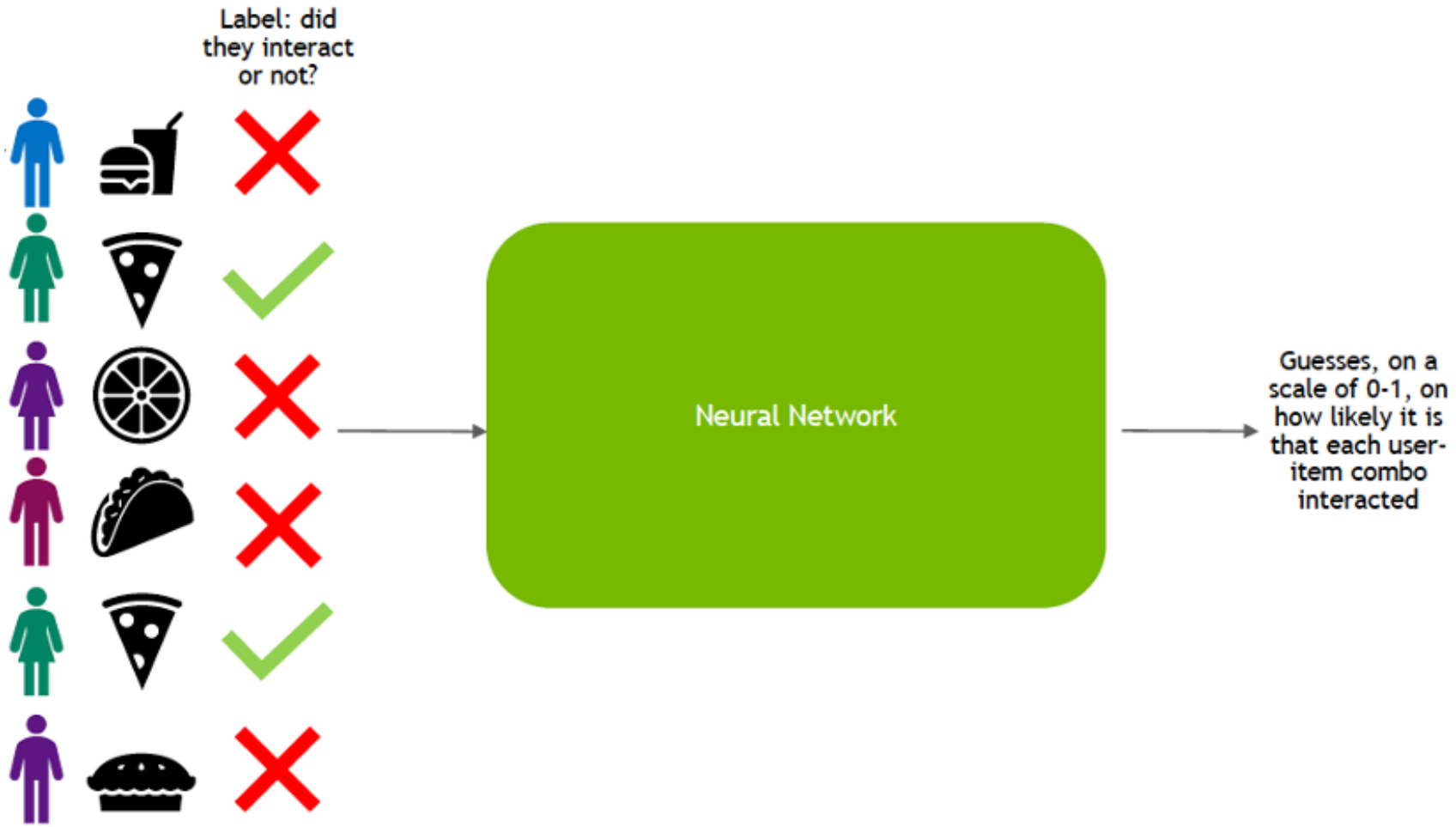



Image credit: Nvidia

# Speech recognition

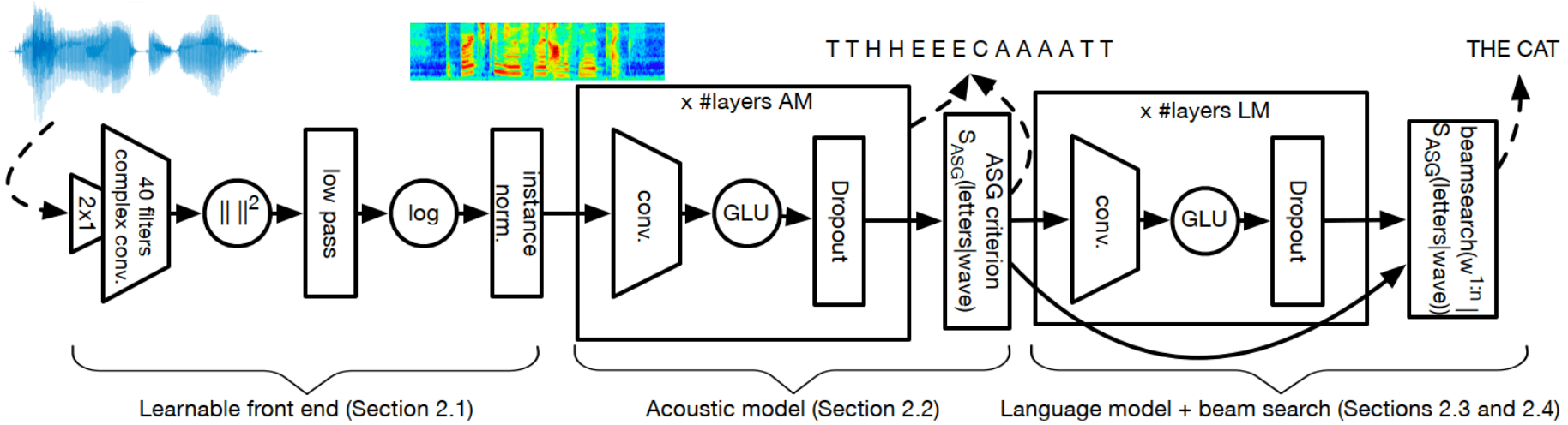



Image credit: AI summer

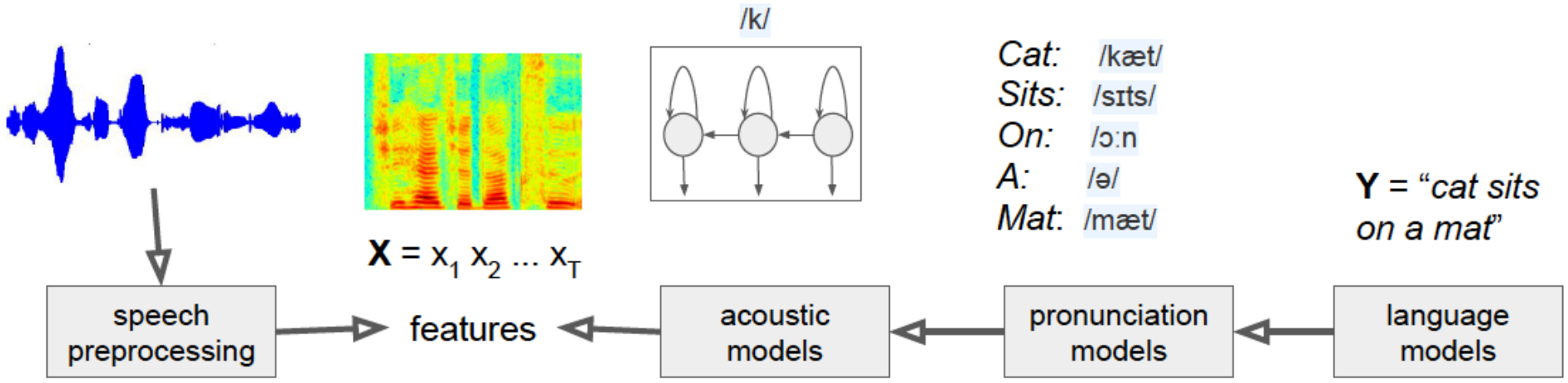
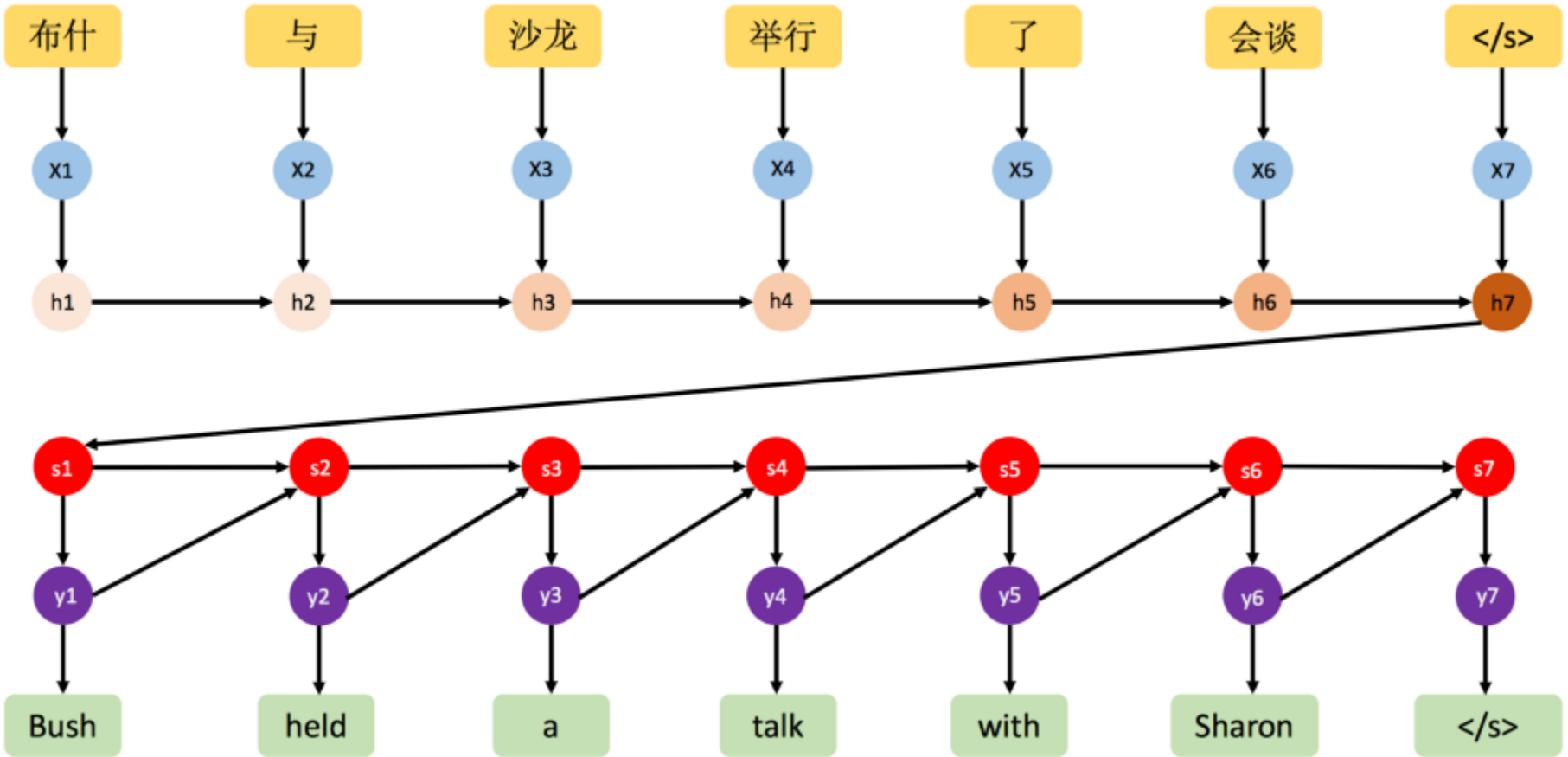



Image credit: Cosmec, Inc.

# Neural machine translation




(Sutskever et al., 2014)

# Weather prediction

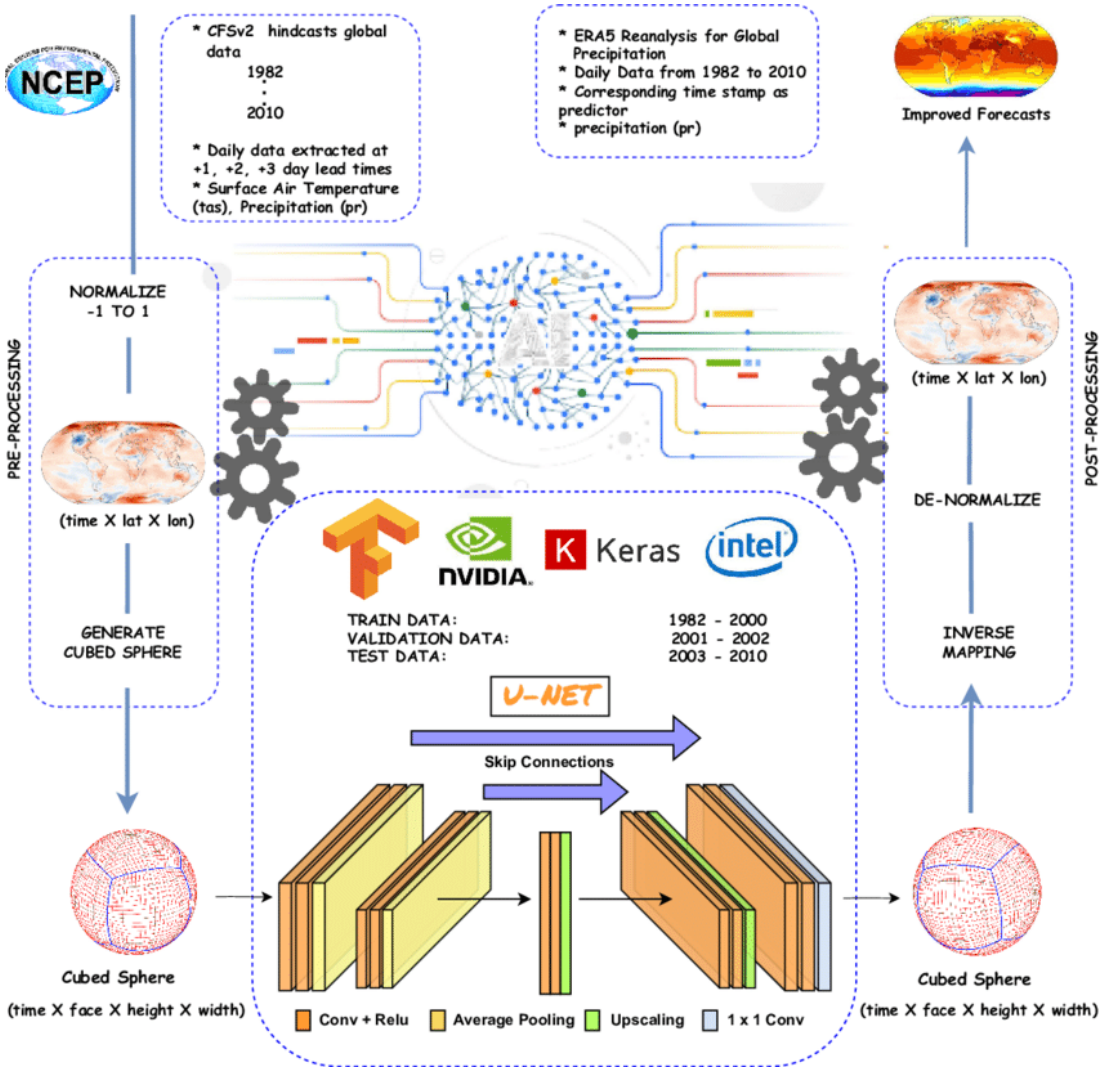



Image credit: Singh et al., 2022

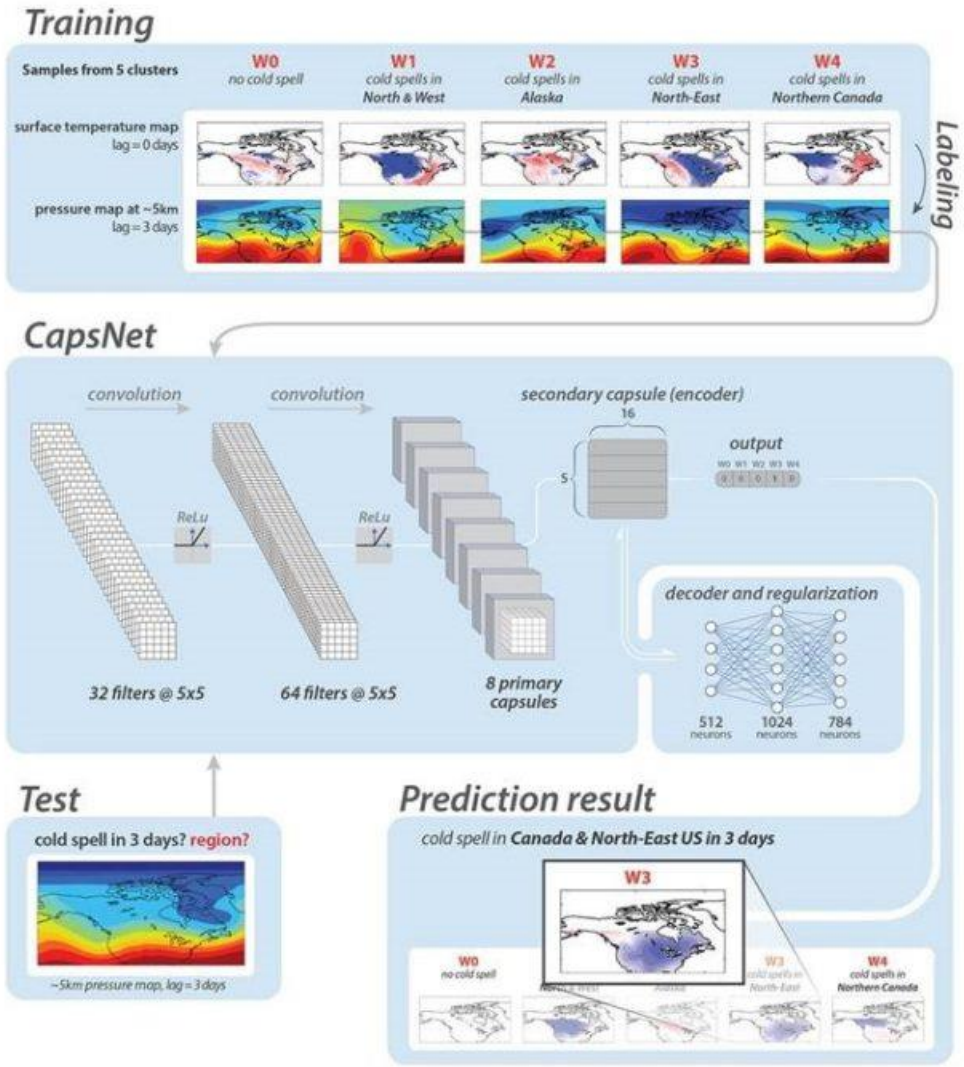



Image credit: Nvidia

# Biology

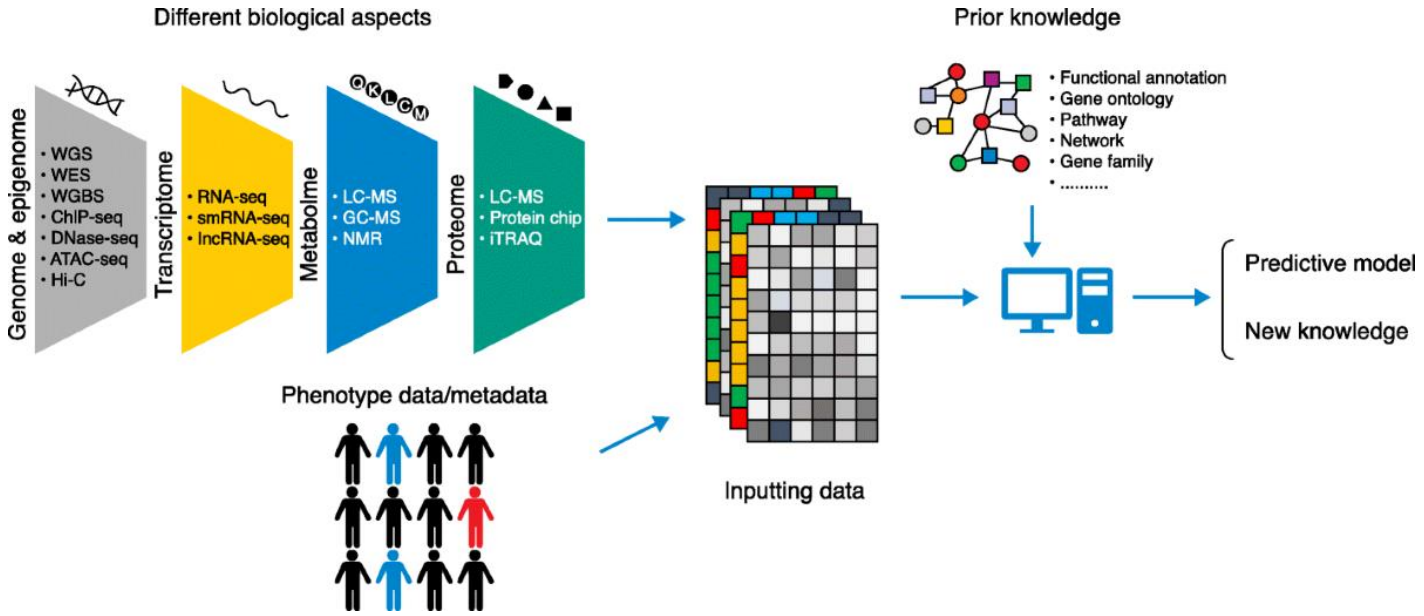



Image credit: BMC

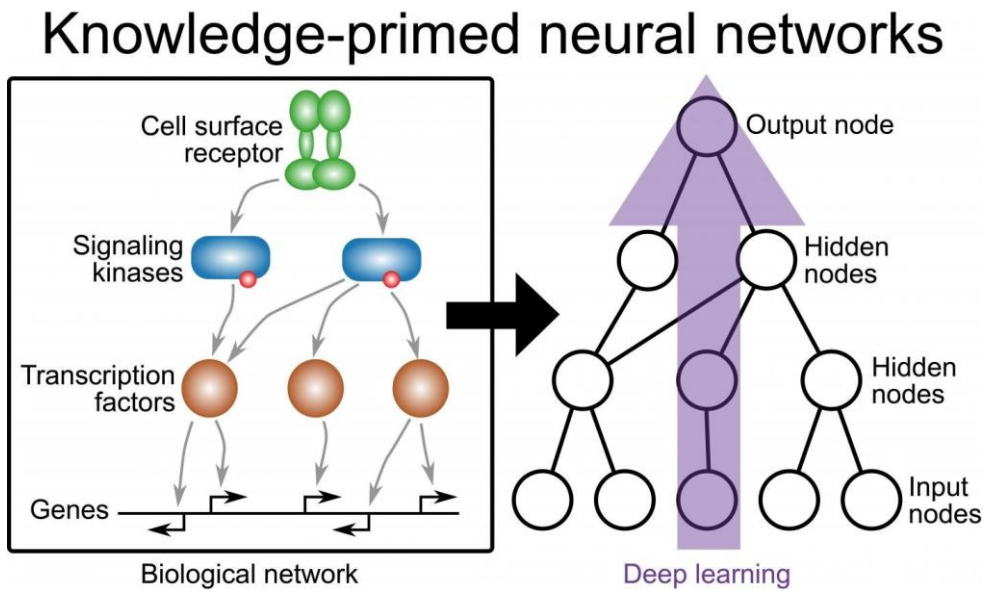



Image credit: phys org

# Engineering design

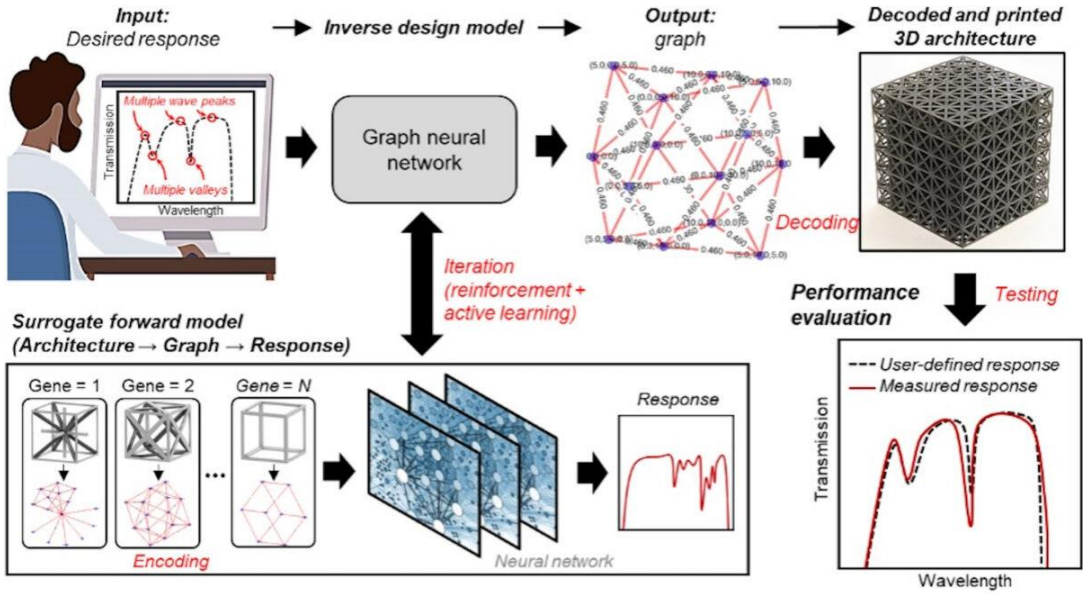



Image credit: Rayne Zheng

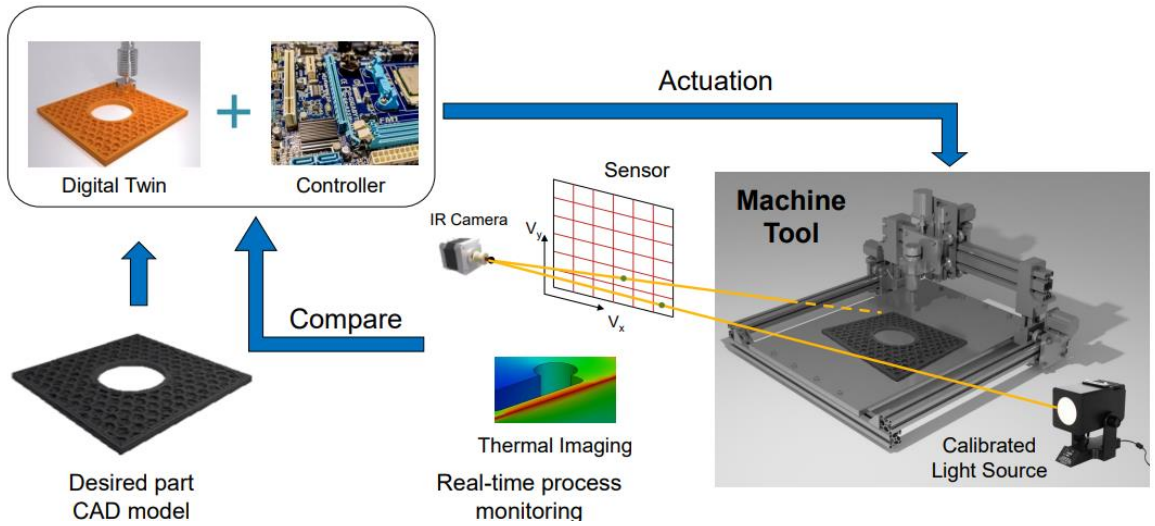



Image credit: Aditya Balu et al., 2022

# Finance

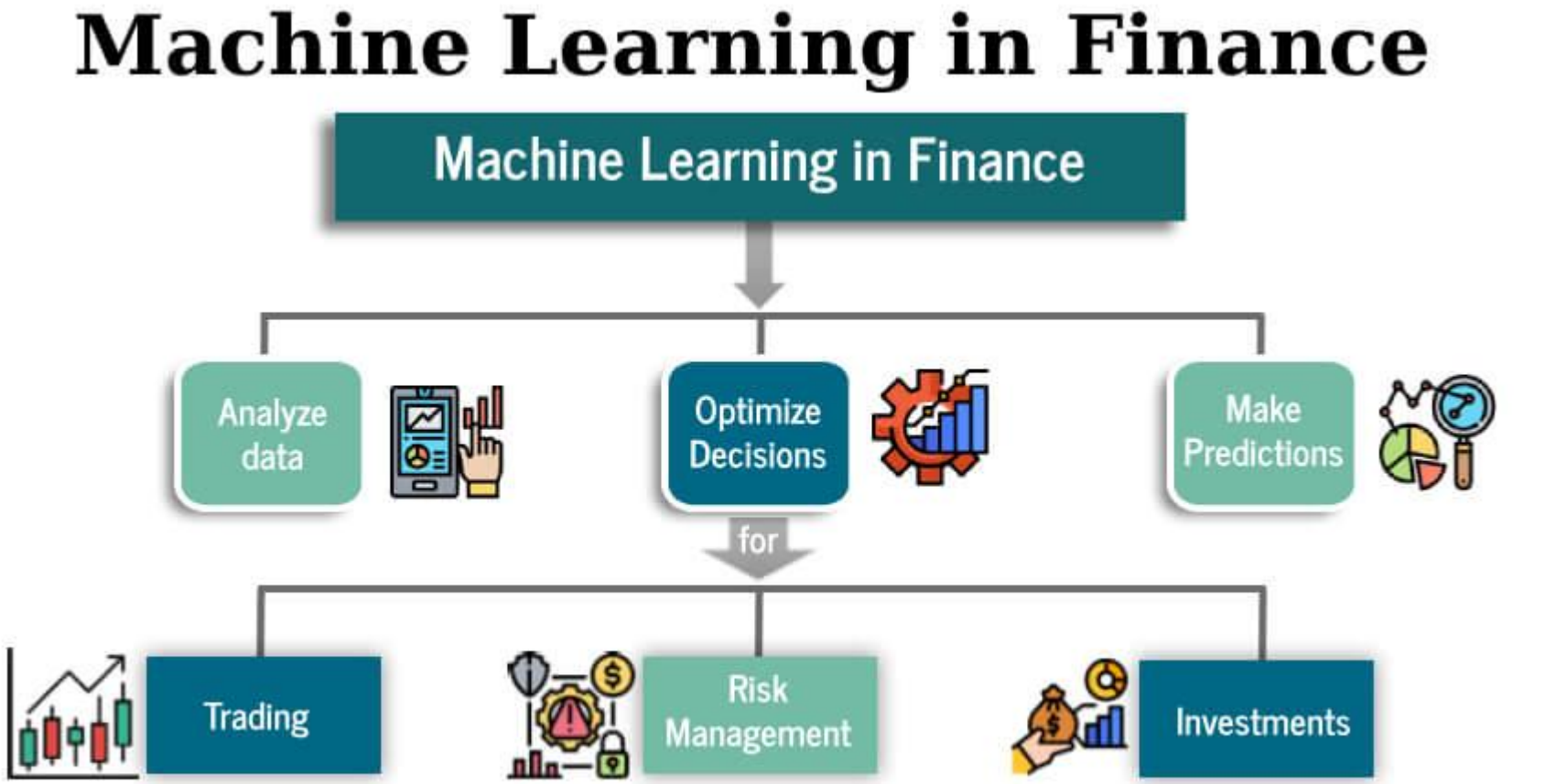



Image credit: EDUCBA

# Outline

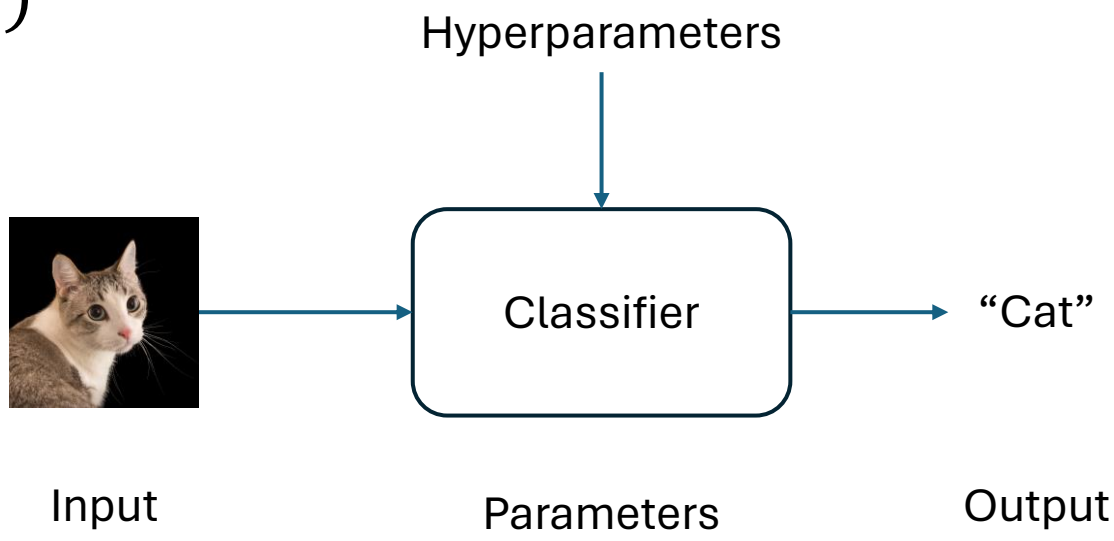
- Session 1: What is machine learning (ML)
- Session 2: Different types of ML
- Session 3: How to frame a learning problem

# Keep this in mind

- *Almost all machine learning algorithms can be cast as optimization problems*
- George Box: *“All models are wrong, but some are useful!”*
- ML now is still ***empirically*** driven

# Some terminologies

- Features/attributes/variables $(X/x)$
  - Input
- Labels/targets/classes $(Y/y)$
  - Output
- Parameters
  - Trainable or learnable
- Hyperparameters
  - Key constants



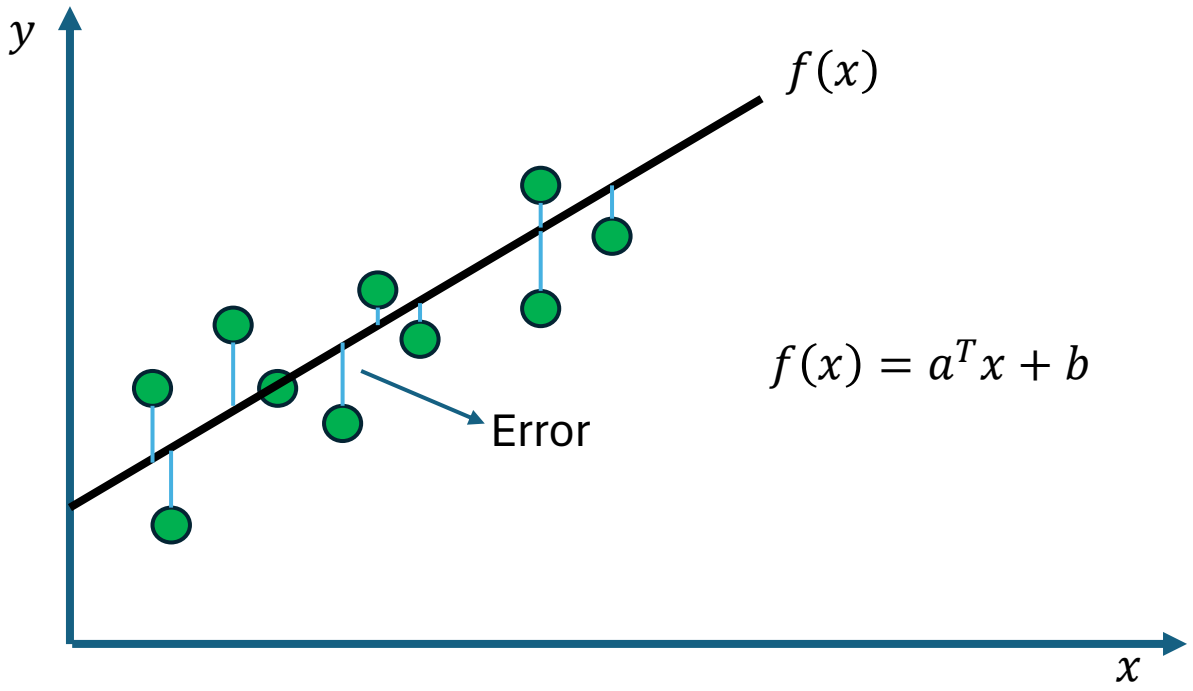
# Types of machine learning

- **Supervised (inductive) learning**
  - Training data and desired outputs (labels)
- **Unsupervised learning**
  - Training data only without labels
- **Semi-supervised learning**
  - Training data and a few labels
- **Reinforcement learning**
  - Rewards from sequence of actions
- More advanced learning is out of scope for now


An excellent machine learning package

# Supervised learning: regression

- Given a dataset $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict $y$ (real-valued) given $x$




$$f(x) = a^T x + b$$

sklearn.linear_model

```
>>> import numpy as np
>>> from sklearn.linear_model import LinearRegression
>>> X = np.array([[1, 1], [1, 2], [2, 2], [2, 3]])
>>> # y = 1 * x_0 + 2 * x_1 + 3
>>> y = np.dot(X, np.array([1, 2])) + 3
>>> reg = LinearRegression().fit(X, y)
>>> reg.score(X, y)
1.0
>>> reg.coef_
array([1., 2.])
>>> reg.intercept_
3.0...
>>> reg.predict(np.array([[3, 5]]))
array([16.])
```

*What we show here is linear, while it can be highly nonlinear*

# Supervised learning: regression

- Popular ML techniques: **linear regression**, **ridge regression**, support vector regressor, **neural networks, LASSO**, decision tree, **random forest**, polynomial regression, **XGBoost**, etc.
- All are built in Scikit-learn already

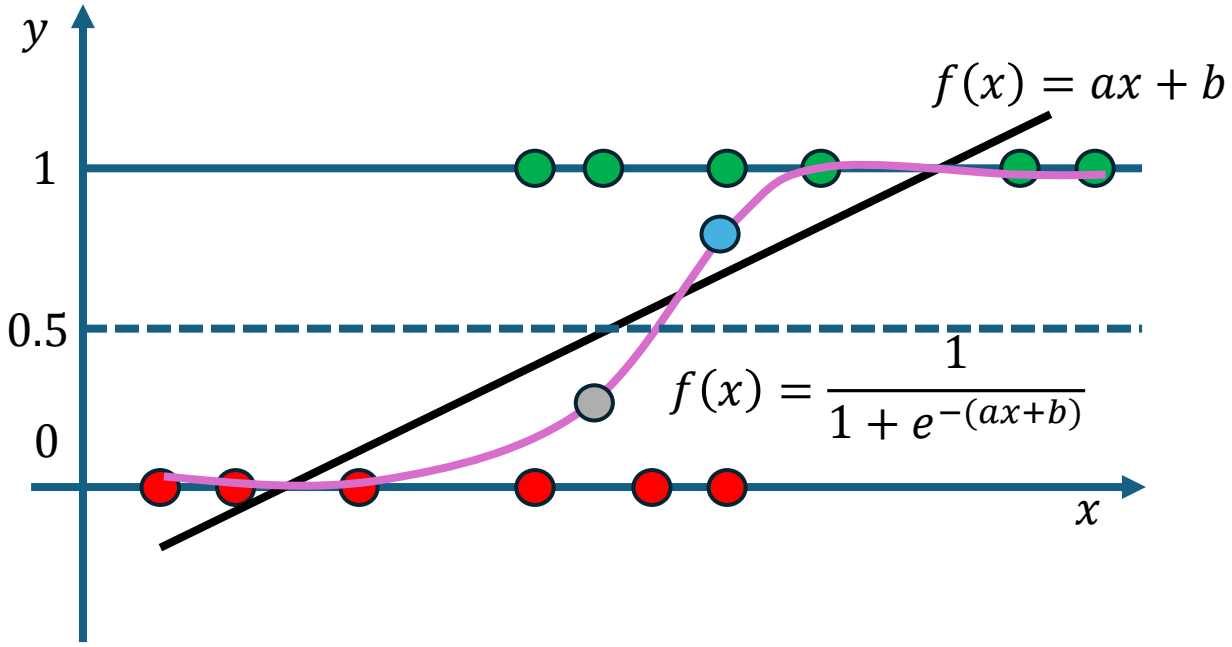
Energy and utility (image credit: WSN)

Stoch market analysis (image credit: Simplilearn)


Quality and process control (image credit: Leeway Hertz)

# Supervised learning: classification

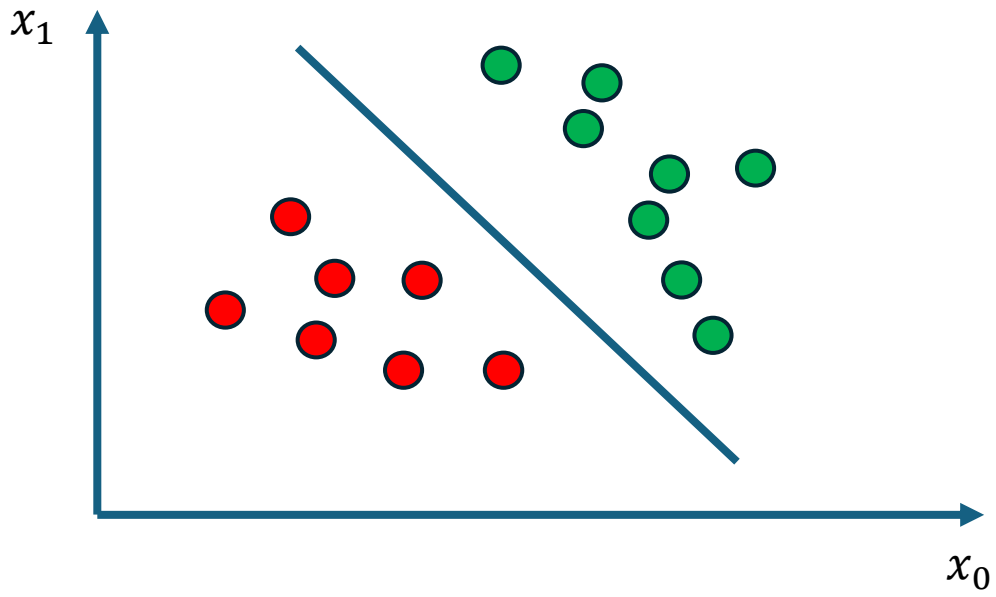
- Given a dataset $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict $y$ (categorical) given $x$




*Class can be binary or multiple*

# Supervised learning: classification

- $x$ can be multi-dimensional
  - Each dimension signifies an attribute

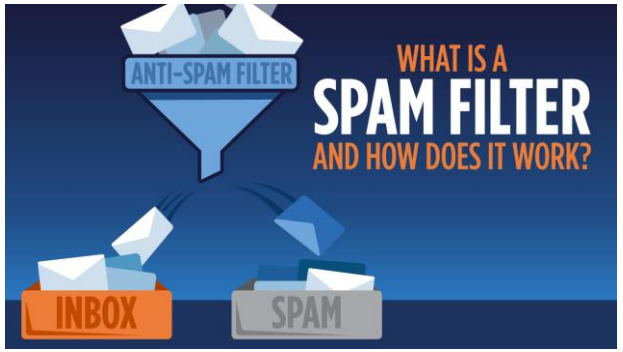



`sklearn.linear_model`

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.linear_model import LogisticRegression
>>> X, y = load_iris(return_X_y=True)
>>> clf = LogisticRegression(random_state=0).fit(X, y)
>>> clf.predict(X[:2, :])
array([0, 0])
>>> clf.predict_proba(X[:2, :])
array([[9.8...e-01, 1.8...e-02, 1.4...e-08],
       [9.7...e-01, 2.8...e-02, ...e-08]])
>>> clf.score(X, y)
0.97...
```

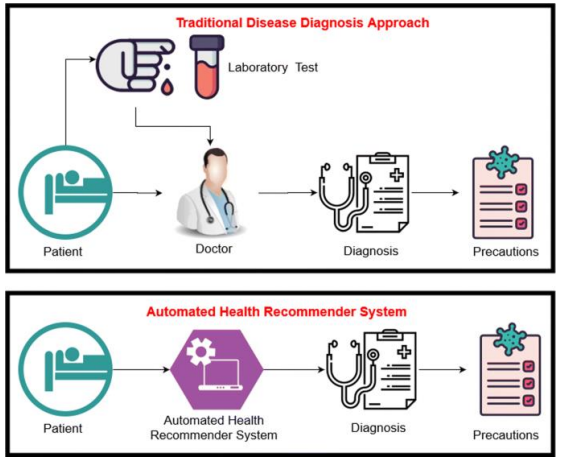
# Supervised learning: classification

- Popular ML techniques: **logistic regression**, **support vector machine**, K-nearest neighbors, decision tree, **neural networks**, naïve Bayes, **random forest**, linear discriminant analysis, etc.




Email spam filter (image credit: Socketlabs)

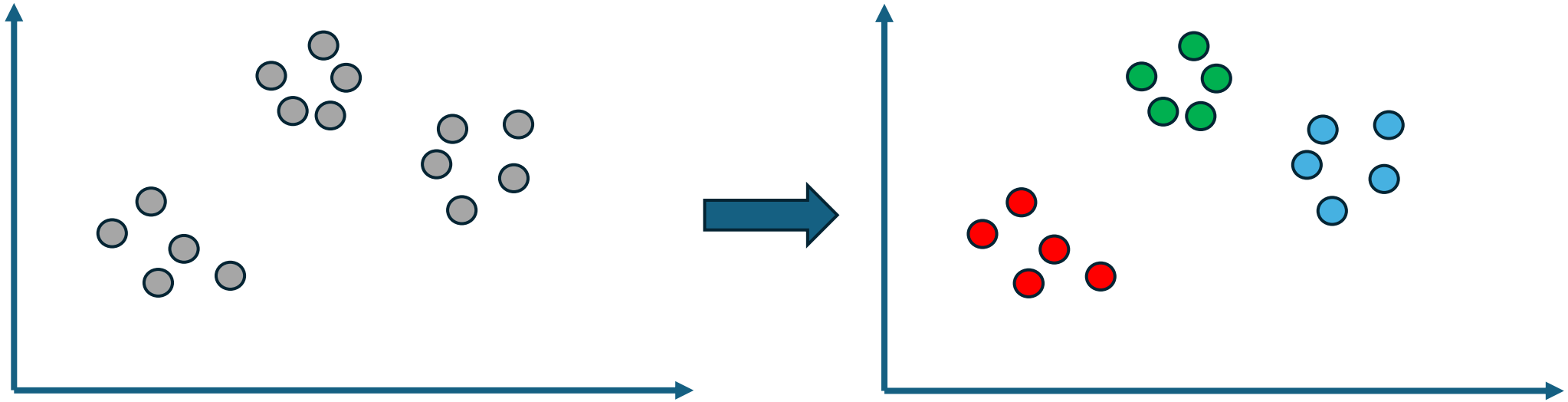
Anomaly detection (image credit: Cloudera)




Disease diagnosis (image credit: Multimedia Tools and Applications)

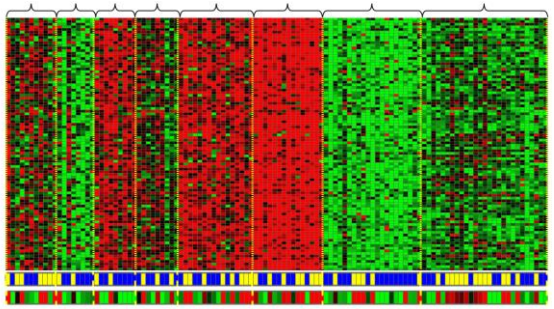
# Unsupervised learning: clustering

- Given $x_1, x_2, \ldots, x_n$ without any labels
- Output hidden structure behind the $x$'s (e.g., clustering)



# Unsupervised learning: clustering

- Popular ML techniques: **K-means**, density-based spatial clustering of applications with noise, **Gaussian mixture model**, balance iterative reducing and clustering using hierarchies, affinity propagation, mean-shift, etc.



Genes grouping (image credit: Daphne Koller)

Market segmentation(image credit: Andrew Ng)

Social network analysis (image credit: Xinyue Tan)

`sklearn.cluster`

```
>>> from sklearn.cluster import KMeans
>>> import numpy as np
>>> X = np.array([[1, 2], [1, 4], [1, 0],
...               [10, 2], [10, 4], [10, 0]])
>>> kmeans = KMeans(n_clusters=2, random_state=0, n_init="auto").fit(X)
>>> kmeans.labels_
array([1, 1, 1, 0, 0, 0], dtype=int32)
>>> kmeans.predict([[0, 0], [12, 3]])
array([1, 0], dtype=int32)
>>> kmeans.cluster_centers_
array([[10.,  2.],
       [ 1.,  2.]])
```

# Semi-supervise learning

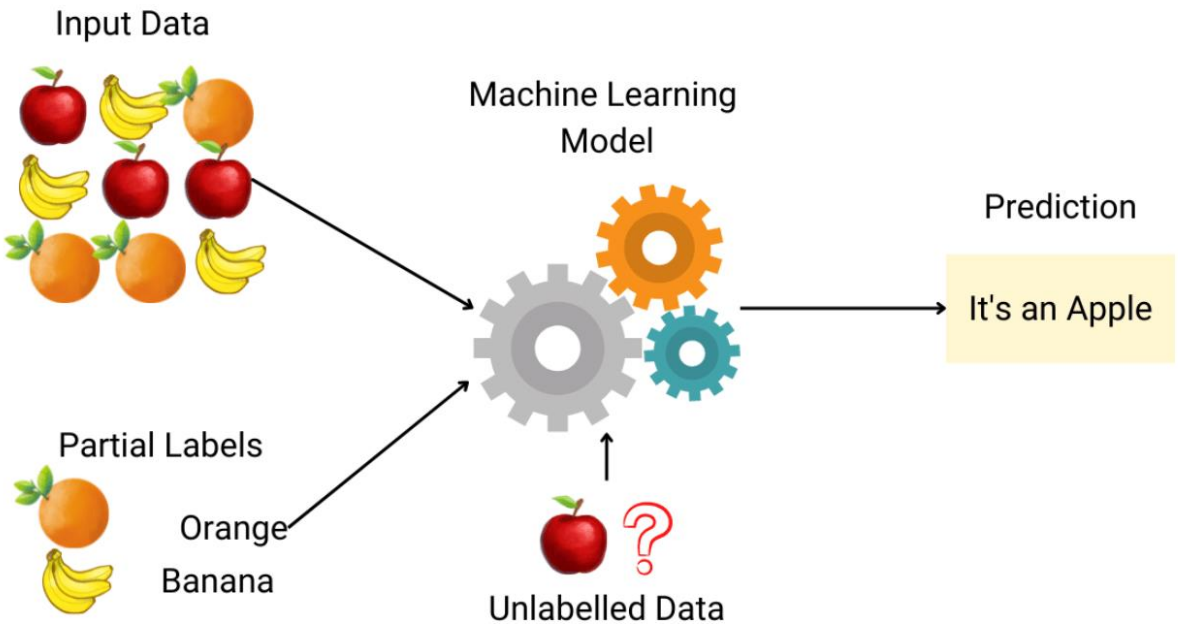
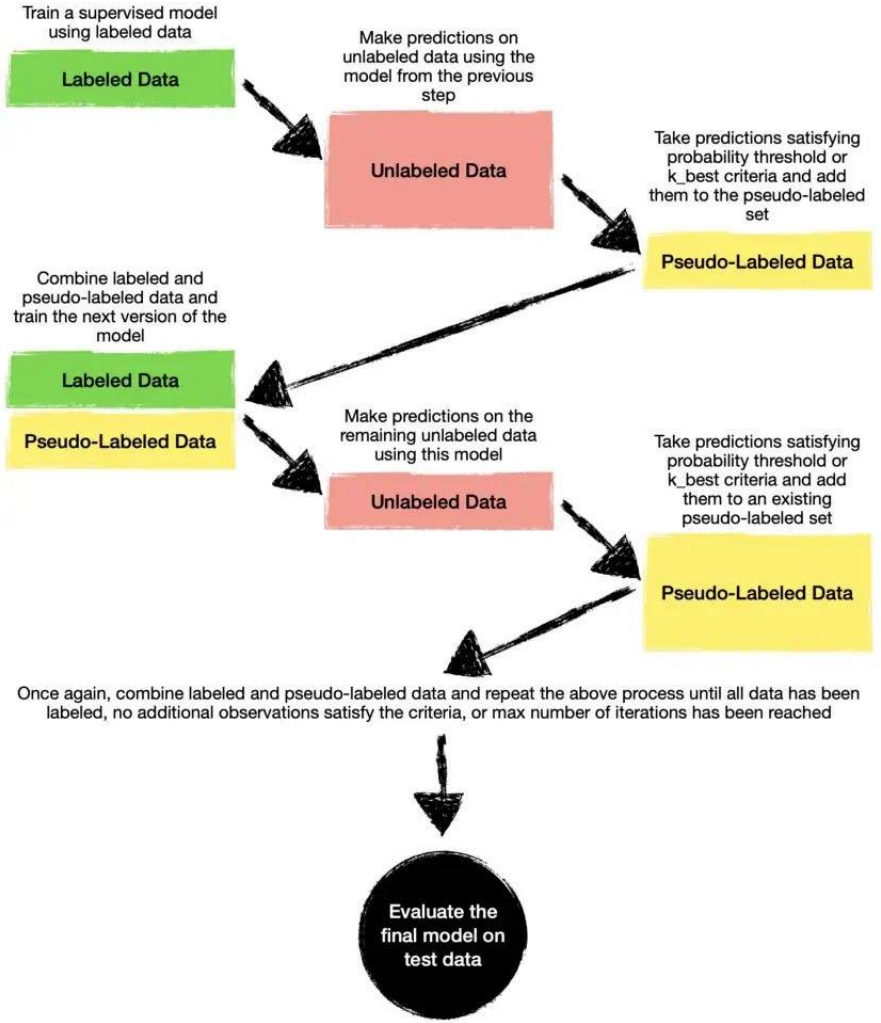



Image credit: medium




Self-training (image credit: Google)

`sklearn.semi_supervised`

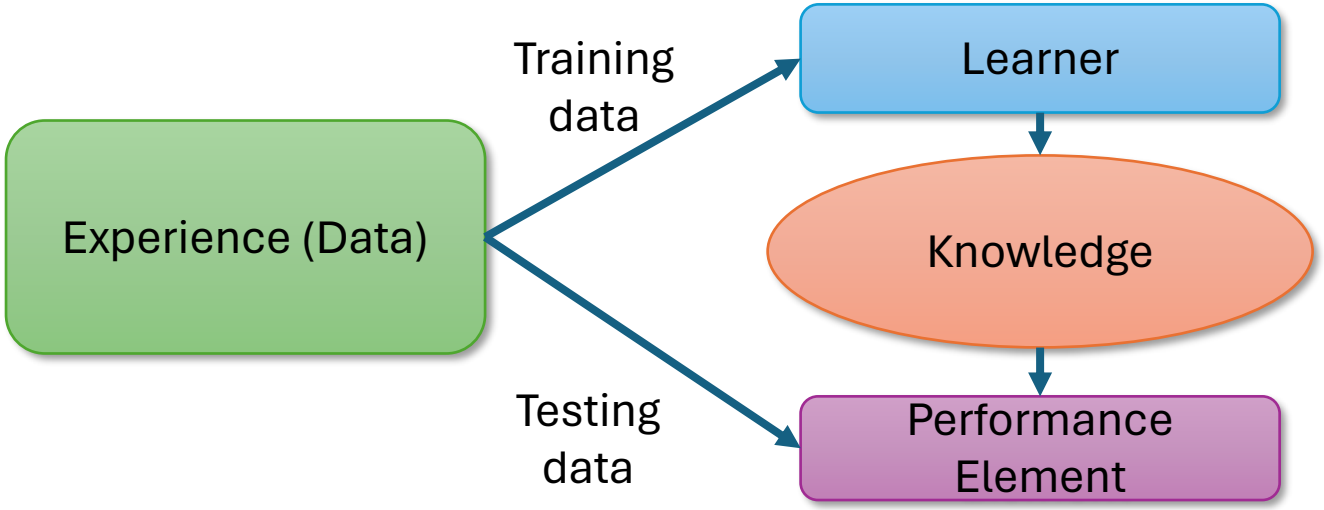
*Co-training: improved version of self-training*

# Outline

- Session 1: What is machine learning (ML)
- Session 2: Different types of ML
- Session 3: How to frame a learning problem

# Learning system

- Choose the experience (data)
- Choose what is to be learned (the unknown relationship or target function)
- Choose how to represent the relationship (model)
- Choose a learning algorithm to infer the function from experience (ML)



# Data

- Data is the key
- In most cases, a large amount of time is devoted to processing data
- Real-world data is dirty
  - Resampling
  - Imputation
  - Feature/attribute selection
  - Scale issue
  - …
- *Feature engineering*
- Data analytics session tomorrow
- Typically, domain knowledge helps here

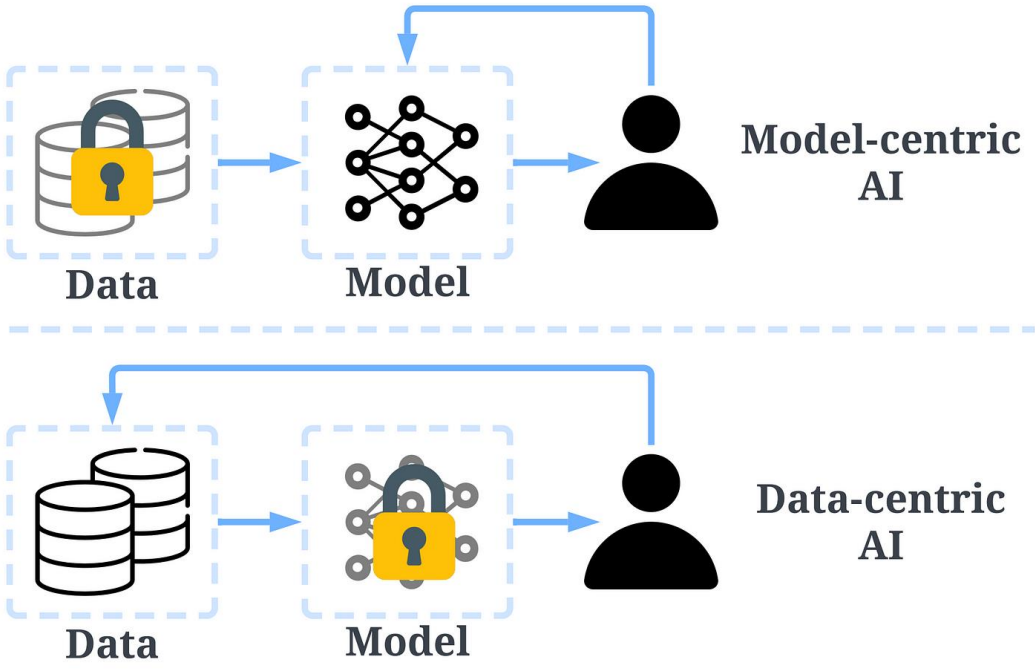
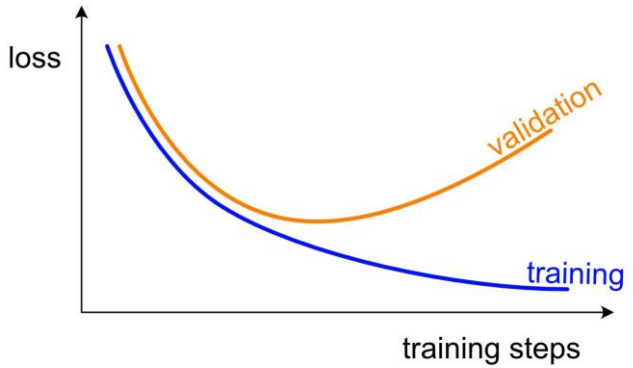
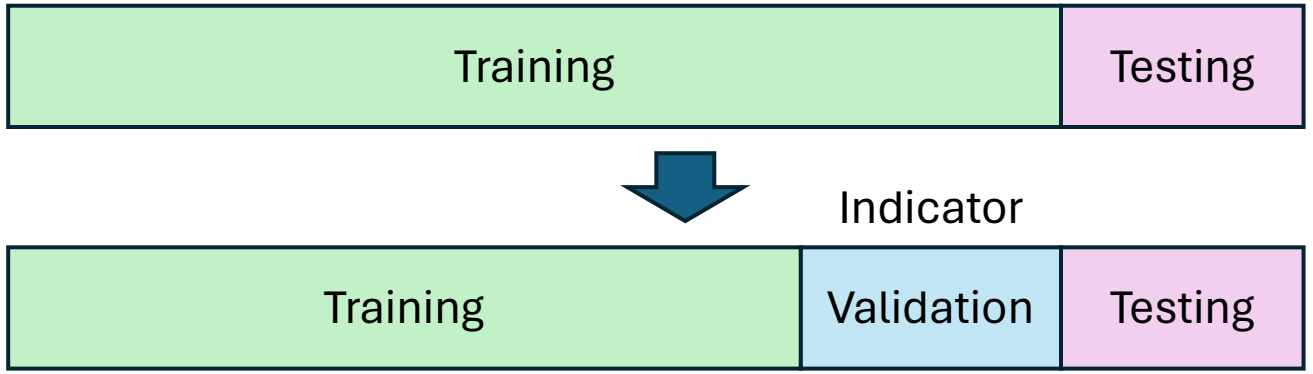



Image credit: Medium

# Data distribution




- For learning a model, data is split




*Data split ratio*

- Popular assumption (though problematic in real-world problems): training and testing data are sampled independent and identically distributed (IID)
- If distributions between training and testing are different, other advanced learning methods are required, e.g., *transfer learning*

# Function approximation

- Numeric functions
  - Linear regression
  - Neural networks
  - Support vector machine
- Symbolic functions
  - Decision trees
  - Rule-based method
- Instance-based functions
  - K-nearest neighbors
  - Case-based

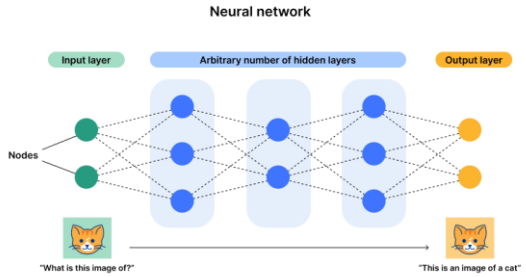



Image credit: Cloudflare

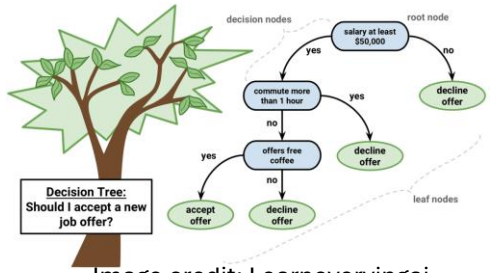



Image credit: Learneveryingai

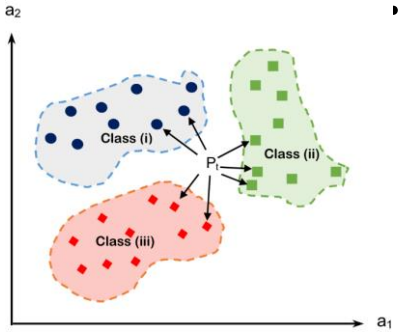



Image credit: Discover Applied Sciences

- Probabilistic graphical models
  - Naïve Bayes
  - Markov networks
  - Bayesian networks
  - Hidden Markov models (HMM)
- …

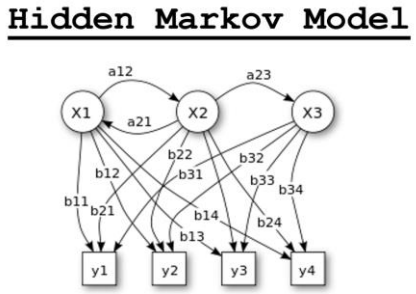



Image credit: Medium

# Learning ≈ Looking for a function

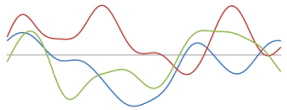
- Image recognition

$f($  $)$ = panda

- Speech recognition

$f($  $)$ = “Hello World”

- Timeseries classification

$f($  $)$ = Faulty

- Dialogue system

$f(\text{"What is } 1+1?\text{"}) =$ “2”

# Search/optimization algorithms

- Gradient descent
  - Perceptron
  - **Backpropagation**
- Dynamic programming
  - HMM
- Divide and Conquer
  - Decision tree induction
  - Rule learning
- Evolutionary computation
  - Genetic algorithms
  - Particle swarm optimization

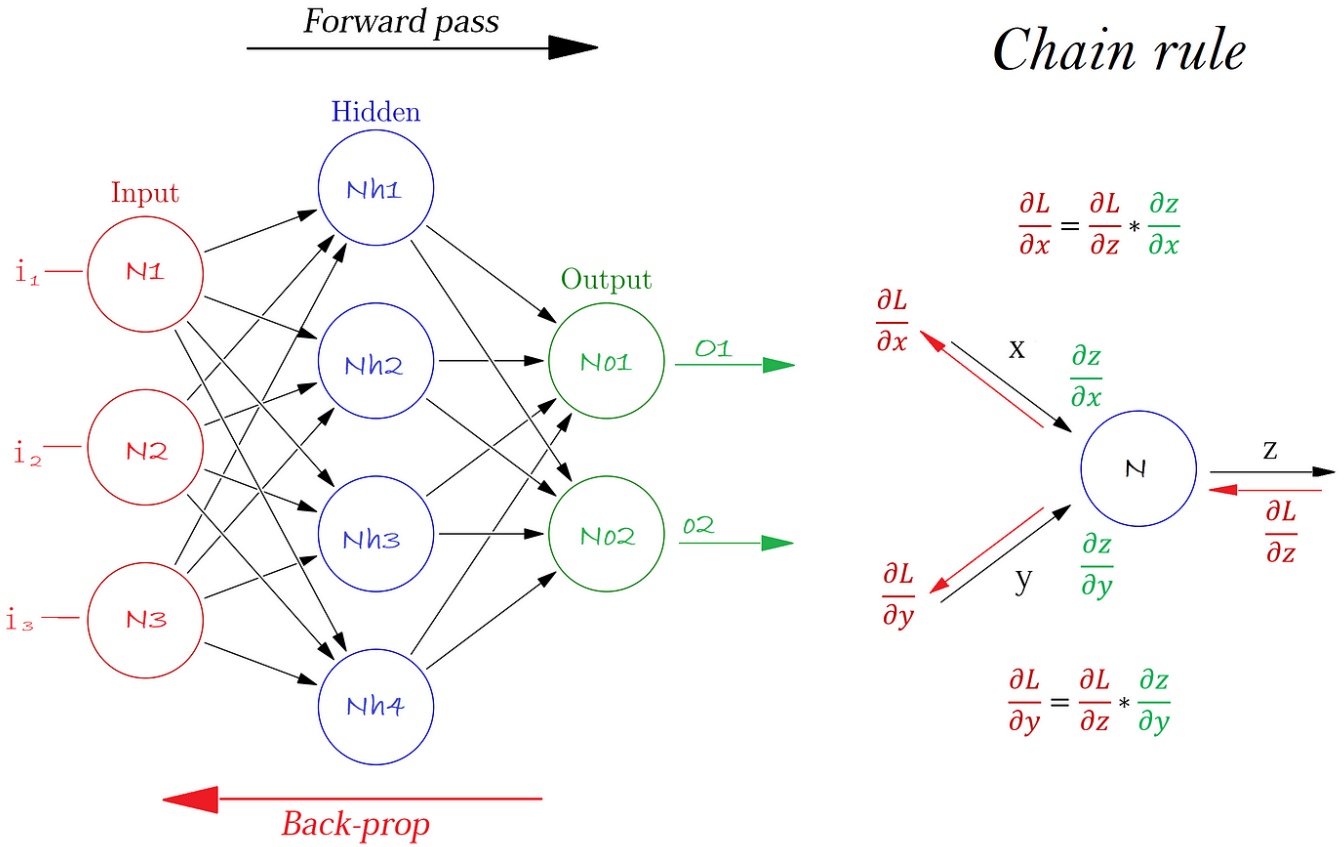



Image credit: Medium

# Evaluation metrics

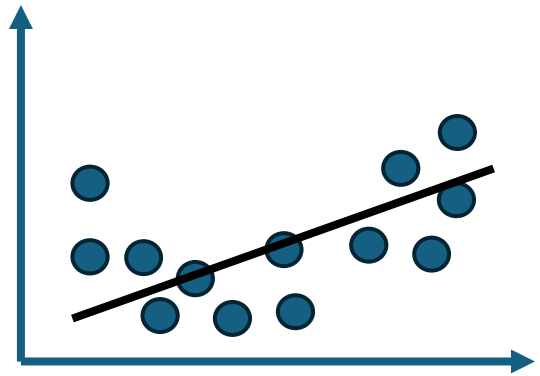
- Classification
  - Accuracy
  - Precision and recall
  - F1 score
  - …
- Regression
  - Mean square error
  - Mean absolute error
  - Mean absolute percentage error
  - R2 score
  - …
- Clustering
  - Mutual information
  - Homogeneity
  - Pair confusion matrix
  - …

`sklearn.metrics`

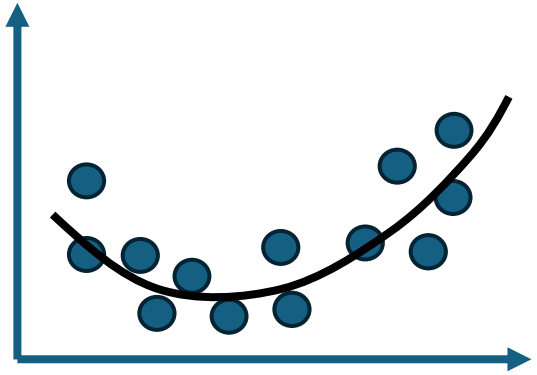
# Machine learning in practice

1. Formulate problem: including understanding domain knowledge, priors, and goals
2. Feature engineering: data integration, selection, cleaning, pre-processing, etc.
3. Learn model: select a proper model and use data to train, with hyperparameter tuning (discussed in the Deep Learning session)
4. Interpret results: evaluate the model with metrics and results with interpretability
5. Consolidation: deploy model and communicate discovered knowledge (you need analytics and visualization)

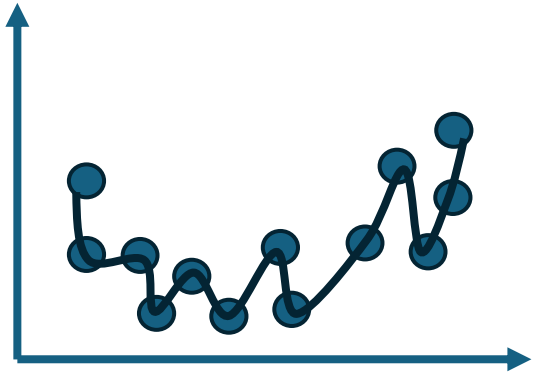
# Fitting problem



Underfitting



Good fit/robust



Overfitting

| Techniques to fight underfitting and overfitting | |
|---|---|
| Underfitting | Overfitting |
| More complex model | Simpler model |
| Less regularization | More regularization |
| Larger number of features | Smaller number of features |
| More data cannot help | More data can help |

# Learning from learning

- Learning involves direct or indirect experience to approximate a chosen target function
- Function approximation requires a search through a space of hypotheses for one that best fits a set of training data
- Data is key to most learning problems instead of models
- Diverse learning methods use different hypothesis spaces and/or employ different search techniques

# Two simple problems

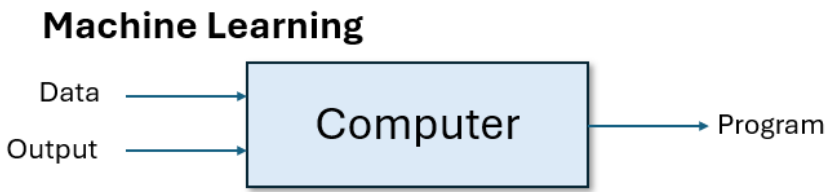
- Classification
  - Logistic regression
  - Iris flower dataset
- Regression
  - Linear regression
  - Diabetes dataset
- https://colab.research.google.com/drive/1JxLirVyrr_yVwNjwmhBQrrKKUIdDIGPz?usp=sharing
- We use Google Colab for demo; all packages have already been installed if using Colab
- More datasets: UCI ML Repo (https://archive.ics.uci.edu/datasets)
- More examples in Scikit-Learn: https://scikit-learn.org/stable/auto_examples/index.html

# Summary

What is machine learning

Different types of learning

How to frame a learning problem




- Supervised
- Unsupervised
- Semi-supervised
- Reinforcement