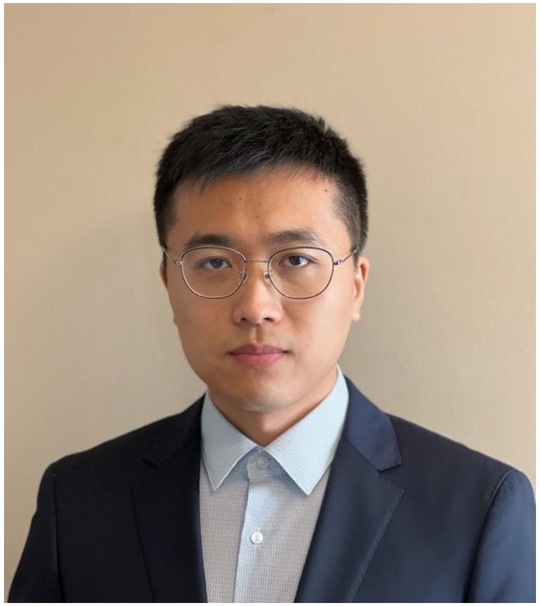
ITCS 6156/8156
Machine Learning

Instructor: Dr. Hongfei Xue
Email: hongfei.xue@charlotte.edu
Class Meeting: Tue & Thu, 4:00 PM – 5:15 PM, WWH 130

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Instructor



- Instructor: Dr. Hongfei Xue, Assistant Professor
- Email: hongfei.xue@charlotte.edu
- Office: Woodward Hall 205A
- Office Hours: Tue & Thu, 5:15 PM - 6:00 PM (i.e., after each class for 45 mins)

Teaching Assistant

- TA: Tonmoy Hasan, Ph.D. Student
- Email: thasan1@charlotte.edu
- Office Hours: Mon. 3:00 PM – 4:00 PM(Virtual), Thu. 3:00 PM – 4:00 PM (In-person, WWH 453)
- Zoom Link for Office Hours: Link

Policy

- **Email Policy:** To ensure efficient and prompt handling of your emails, please include the *course number* (ITCS 6156/8156) and your *800/801 number* in the email subject. This will facilitate quicker recognition and response to your inquiries.

- **Late Submission:** All assignments (Homework & Projects) are due on the specific day and time posted. You can submit an assignment up to 3 days late with a fixed daily penalty of 20% out of total points. The latest submission (3 days late) will receive at most 40% of max points even if it's all correct; 0 points if more than 3 days late. Excuses that you did not have enough time for an assignment will not be considered. (Suggestion: start your assignments early.)

- **Exam:** If you miss an exam because of sickness or similar reasons, visit a physician and obtain a note detailing the period during which you were medically incapable of taking the exam. Notify the instructor immediately via email if you are going to miss an exam, before the exam takes place unless medically impossible. See the instructor as soon as you return to class. If you miss an exam without a valid excuse, you will receive a zero grade for that exam. No make-up exam will be available without a valid excuse.

Academic Dishonesty Policy

- This course will operate with **a ZERO-TOLERANCE policy regarding cheating and other forms of academic dishonesty**. Students are expected to follow the **University Policy: 407, Code of Student Academic Integrity** for all class activities, homework, assignments, and exams. Any act of academic dishonesty will subject the student to penalty, including the high probability of failure of the course (i.e., assignment of a grade of "F") and formal reports to the Dean of Students Office.
  - Following all of the instructions given by the course instructor, TAs, and other test proctors.
  - The homework, quiz, and project assignments must be done ***individually***. *Importantly, under NO circumstances may students rely on the work of their peers, including but not limited to GitHub repositories, generative AI, and code submissions from previous academic terms.*
  - Use of reference materials in the library or online is allowed, provided that the homework explicitly cites the references used. Note that copying the solutions from online sources or the previous semester is still considered cheating even if you cite the sources.
  - All work for this course must be original for this course. Therefore, recycled papers, work submitted to other courses, and major assistance in the preparation of assignments without identifying and acknowledging such assistance are not acceptable.
  - *Students who do share their work with others are as responsible for academic dishonesty as the student receiving the material.* Students are not to show work to other students, in class or outside the class. Students are responsible for the security of their work and should ensure that printed copies are not left in accessible places, and that file/directory permissions are set to be unreadable to others.
  - Any student may withdraw their submission (homework, projects) at any time, no questions asked, BEFORE any AI violation is discovered.
  - We (the instructor, TAs, and other test proctors) reserve the right to make the ultimate determination regarding breaches of academic integrity policies.
  - You are not allowed to post course homework, exams, solutions, etc., on a public forum.

Course Information

- **Prerequisites:** Students are expected to meet the following prerequisites to ensure a smooth learning experience in this course:
  - Programming Backgrounds: Students are required to possess a basic understanding of programming concepts, including Data Structures and Algorithms. In addition, students are expected to be comfortable with programming in Python.
  - Mathematical Foundations: This course involves some mathematical concepts that form the backbone of various topics. Students should have a basic knowledge of Linear Algebra, Calculus, and Statistics.
- To augment the learning experience of the course, pertinent background material will be accessible on the course website throughout the course.

Course Information

- **Course Description:** Machine learning has a remarkable impact across diverse domains, including autonomous control, robotics, computer vision, natural language processing, speech recognition, health science, biology, data mining, etc. This course will introduce fundamental concepts, techniques, and algorithms underlying the theory and practice of machine learning. Major machine learning models and techniques we will cover include linear regression, logistic regression, support vector machines, k-nearest neighbors, decision trees, k-means clustering, principle component analysis, Bayesian learning methods, ensemble methods, reinforcement learning, neural networks, auto-encoders, convolution neural networks, graph neural networks, recurrent neural networks, attention mechanism, generative models, etc.

Course Information

- **Textbook:** There is no required textbook for this course. However, the following free texts are highly recommended:
  - Pattern Recognition and Machine Learning by Christopher Bishop. Springer, 2007.
  - The Elements of Statistical Learning: Data Mining, Inference, and Prediction by T. Hastie, R. Tibshirani, & J. H. Friedman. Springer Verlag, 2009 - 2017
  - Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville. MIT Press, 2016. The Atkins Library in UNCC continues its efforts to reduce textbook costs for students of eBook availability for UNC Charlotte courses. The library identifies and purchases unlimited online access to this book.
  - Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. Cambridge University Press, 2020.
  - Deep Learning: Foundations and Concepts by Chris Bishop.

Course Information

- **Course Website:** http://havocfixer.github.io/teaching/Fall24_ITCS6156_8156.html
- **Course Content Schedule:** (Refer to syllabus for details)
- **Grading Policy:**
  - 10 Quizzes: 10% (1% for each quiz)
  - 5 Homework: 25% (5% for each homework)
  - 1 Course Project: 30% (Paper Pre.+ Mid-term Rep. & Pre. + Final Term Rep. & Pre.)
  - Mid-term exam: 15%
  - Final-term exam: 20%
- **Grading Scale:**
  - Grade A: [88, 100]
  - Grade B: [74, 88)
  - Grade C: [60, 74)
  - Grade F: [0, 60)
- **Dates for Exams:**
  - Mid-term exam: Oct. 08 (Mon.), 4:00 PM – 5:15 PM, WWH 130
  - Final-term exam: TBD

Course Information

- **Bonus everywhere!**
- Different standards for 6/8 levels.
- **Quiz:** In-class quiz; open-book; multiple-choice questions.
- **Homework:** Coding and math assignments.
- **Course Project:**
  - For PhD students, it can be a topic related to your research (You need to discuss the project topic with me). You can also select a predefined project topic in the class. You should work on the course project **individually**.
  - For Master students: You will select a predefined project topic in class. You may form teams of up to 3 students.
- **Exams:** Close-book exam. Cheat sheet allowed.

Topics of This Class

How to Automate Solutions to Computational Problems

- Spam email classification:
  - Binary classification of emails: Spam vs. Ham (Legitimate message)

- Expert Systems approach (Rule-based)
  - A group of experts write rules determining whether an email is spam or not.
  - A programmer implement the rules into computer code

- Example rules:
  - Classify the email as spam in “Money” appears in the text.
  - What if the email is sent by your parents?

How to Automate Solutions to Computational Problems

- Cons of **Expert Systems** approach (Rule-based)
  - **Cognitively demanding**: Difficult for humans to reason with many useful but imprecise features that are indicative (signals) of spam or not spam:

    - Words, phrases, images, meta-data, time series, …

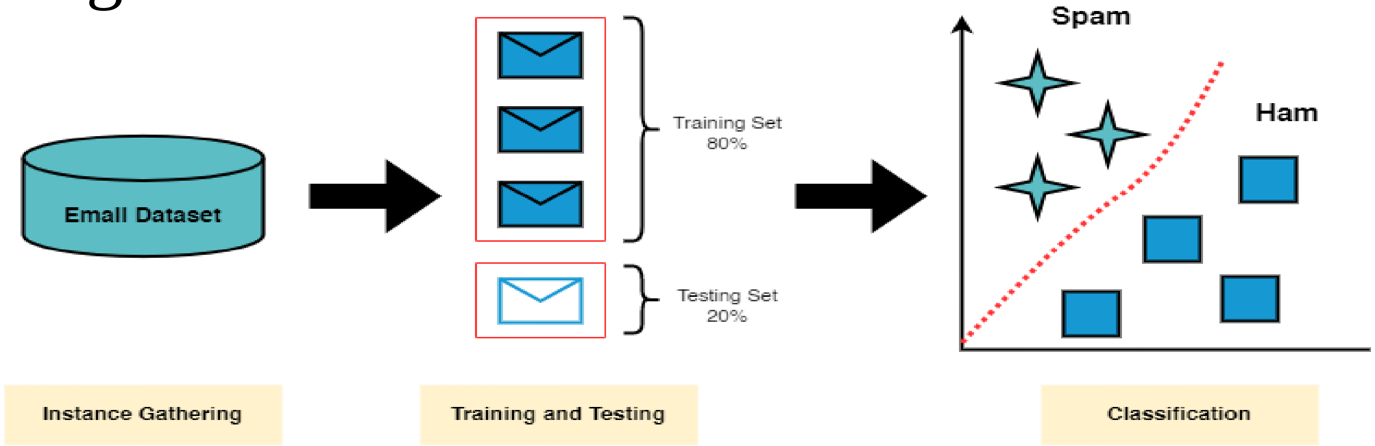
    - Need to combine a large number of signals, figure out their relative importance in determining spam vs. ham label.
  - **Brittle**: Always going to miss some usef features or patterns

    - Spam filtering is adversarial, new features need be added over time.

Expert (Rule-based) Systems

Why Machine Learning?

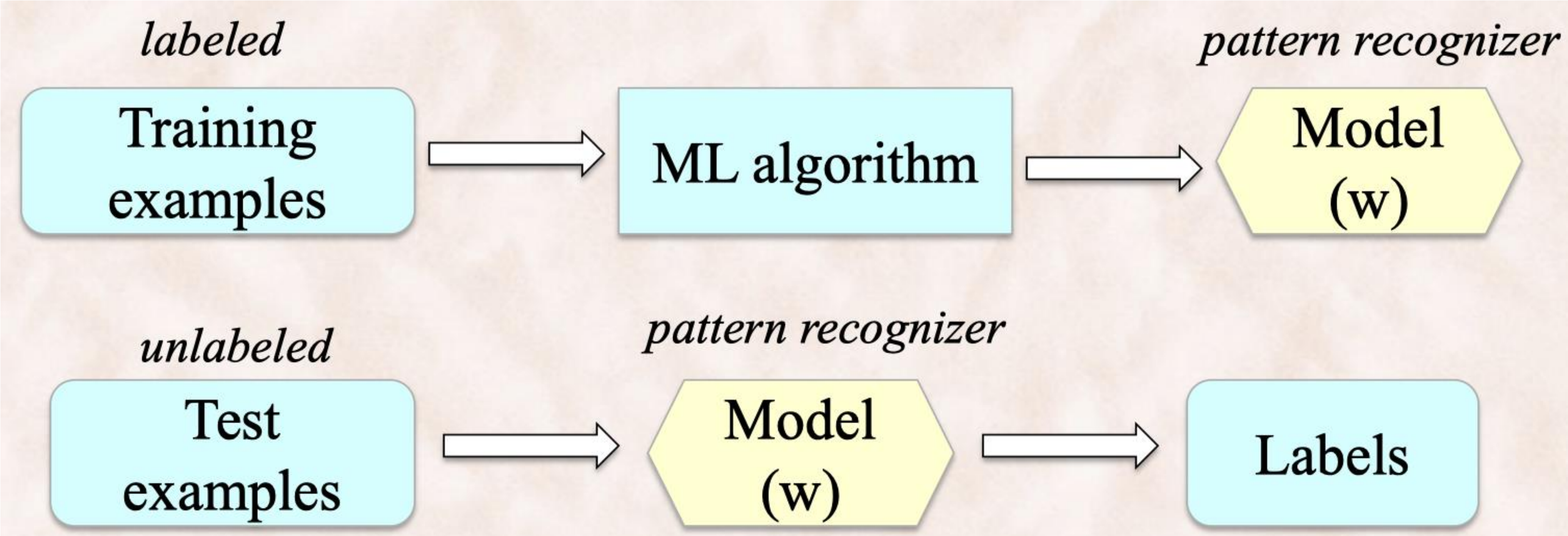
- Machine Learning algorithms can automatically learn the weights to combine features.



- A typical Machine Learning (ML) approach:
  1. Acquire a large enough dataset of labeled examples:
     - Each email is an instance, the label is spam (+1) vs. not spam (-1).
  2. Represent emails as feature vectors:
     - Each feature has a weight, the sign of the weighted sum of features should match the label.
       - Traditional ML: Engineer the features.
       - Deep ML: Learn the features
  3. Learn the weights so that the model (weighted combination of features) does well on labeled examples.

What's Machine Learning?

- **Machine Learning** is to construct computer programs that learn from experience to perform well on a given task.
  - **Supervised Learning**: discover patterns from labeled examples that enable predictions on (previously unseen) unlabeled examples.

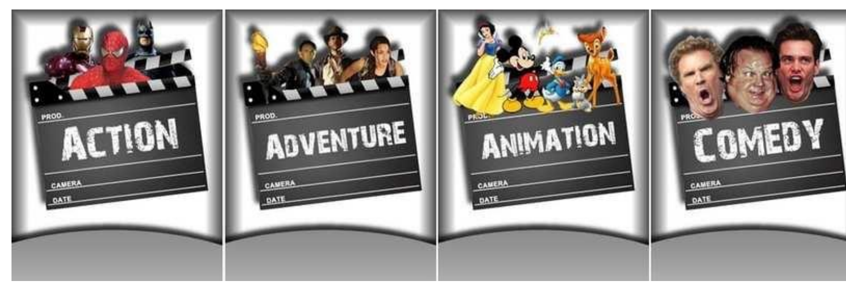


Machine Learning

- Function is everywhere!
  - Function → f ; Input instance → x; Output Targe → y

$f($  $)$ → 

$f($ 

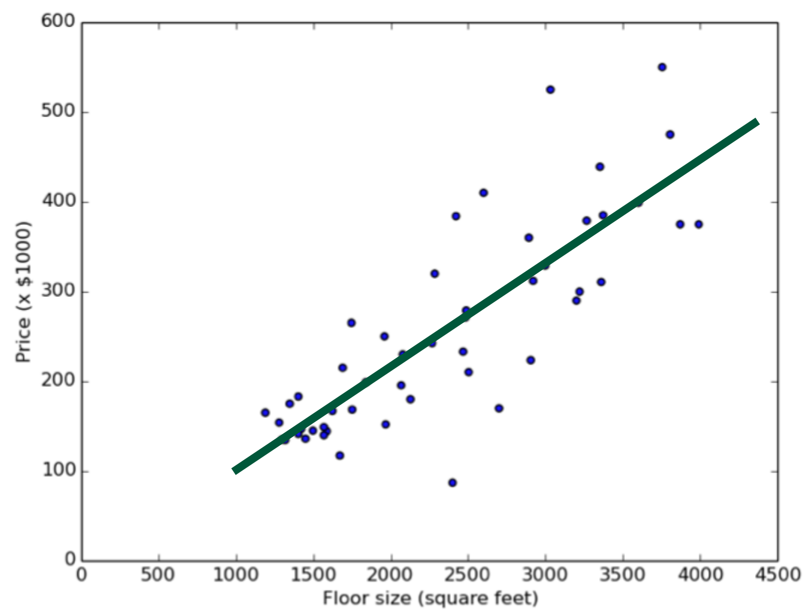
$)$ → 

- Machine Learning Task:
  - learn an (unknown) function $f: X$ → Y that maps input instances x $\in X$ to output targets $f(x) \in Y$.

Linear Regression

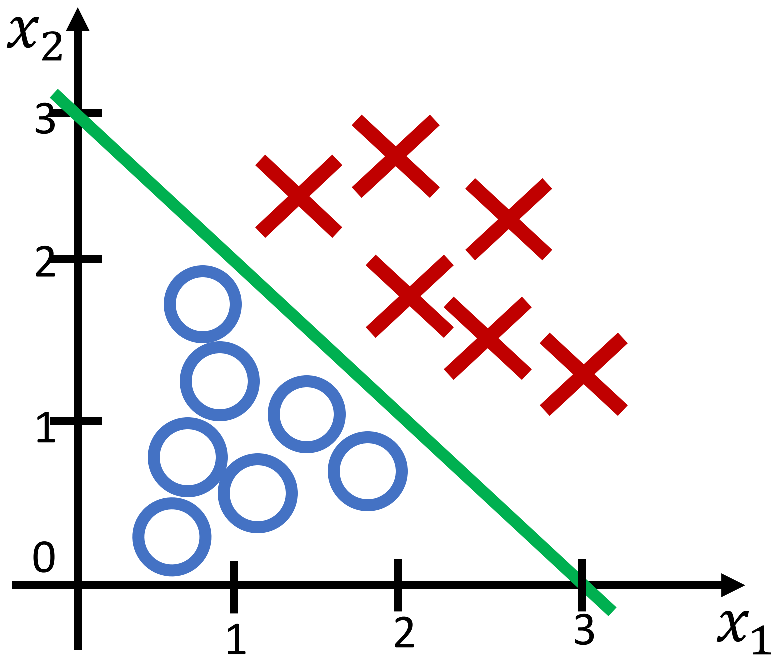
- Given the floor size in square feet, predict the selling price:
  - Input x: the floor size of the house
  - Output y: the selling price of the house
  - Need to learn a function h such that $h(x) \approx f(x)$

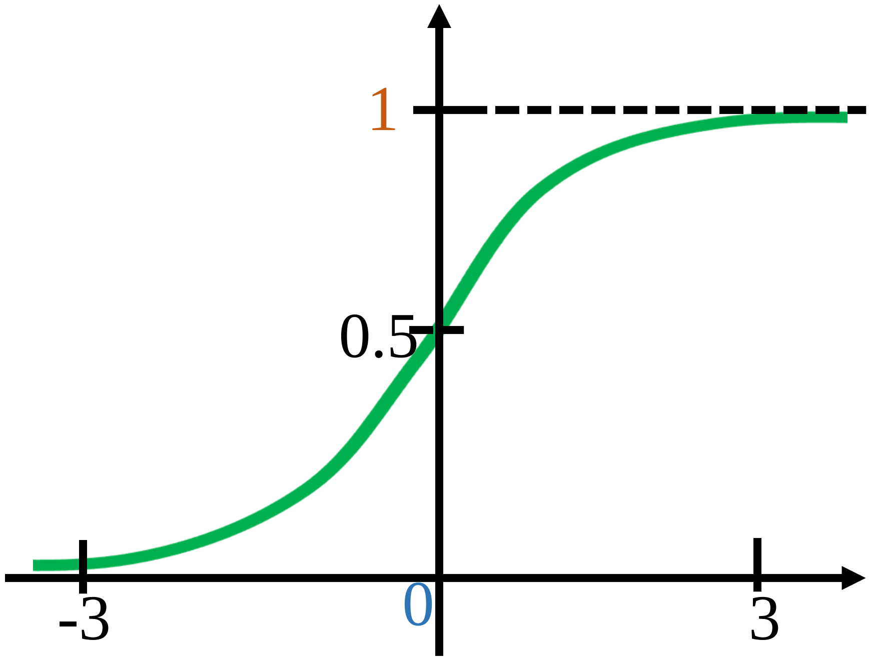
- Assumption: the prediction results are the linear combination of input attributes (features).



Logistic Regression

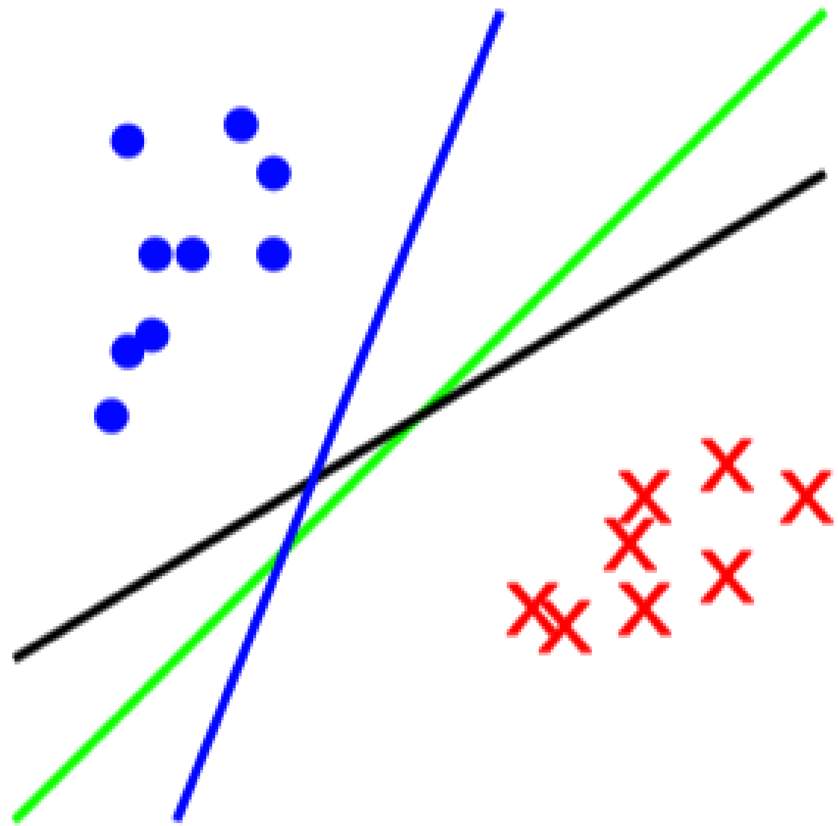
- Objective: To accurately determine the likelihood (probability) of each data instance being classified into a designated category.





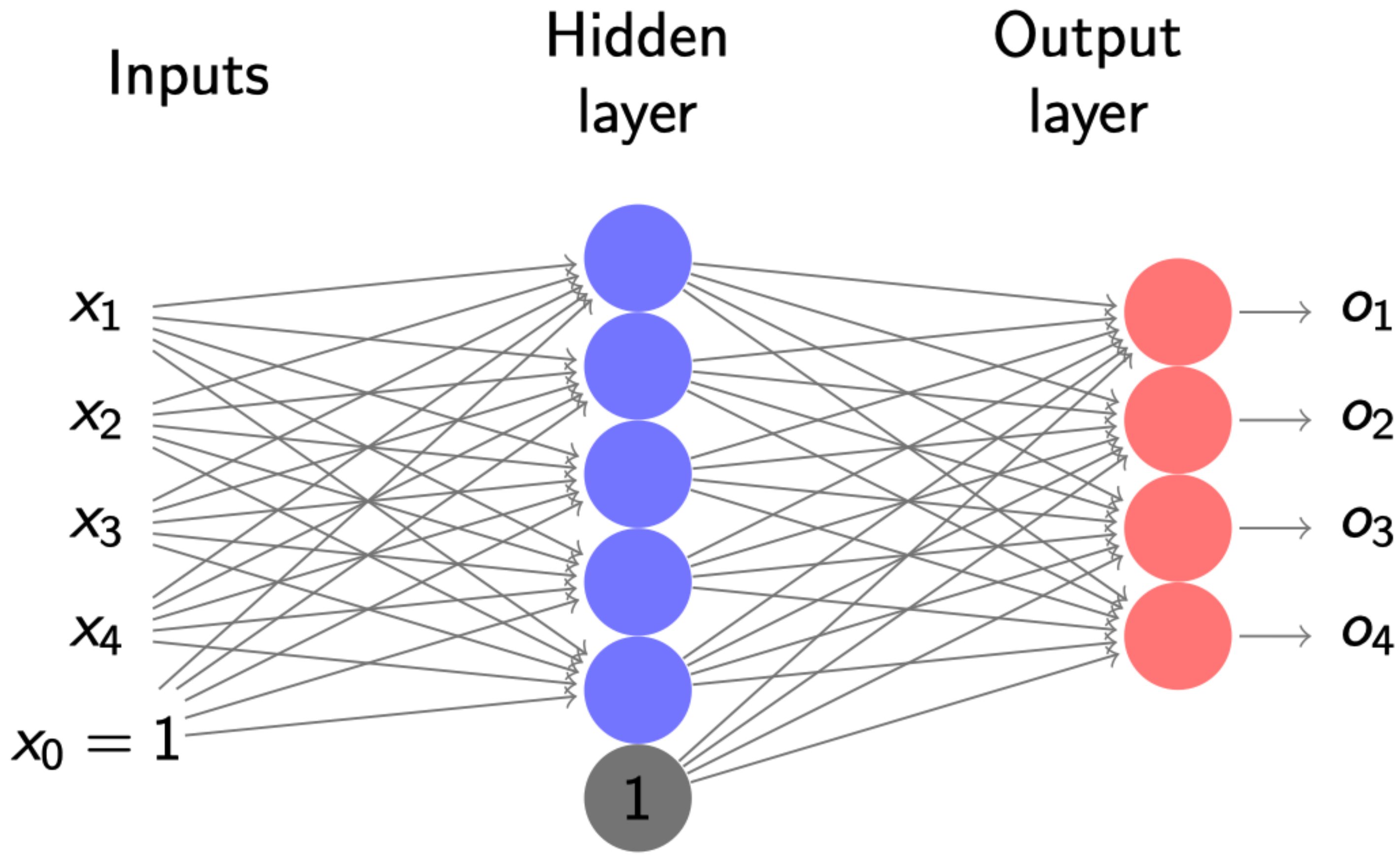
Support Vector Machine

- Objective: to find the optimal hyperplane that maximally separates classes of data points in a feature space.



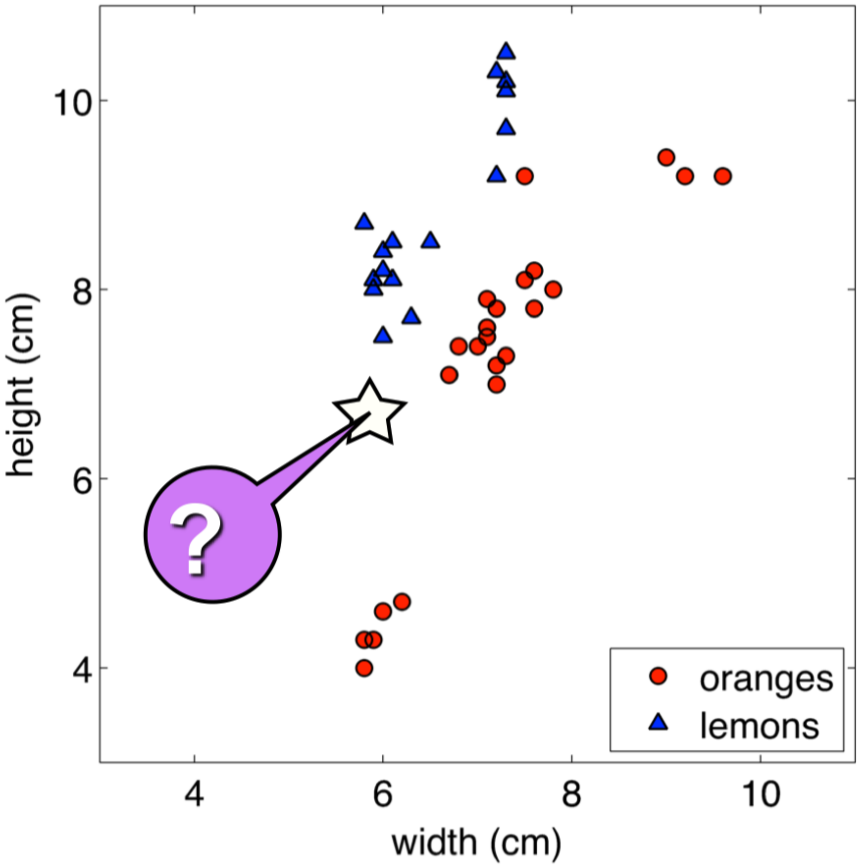
Neural Networks

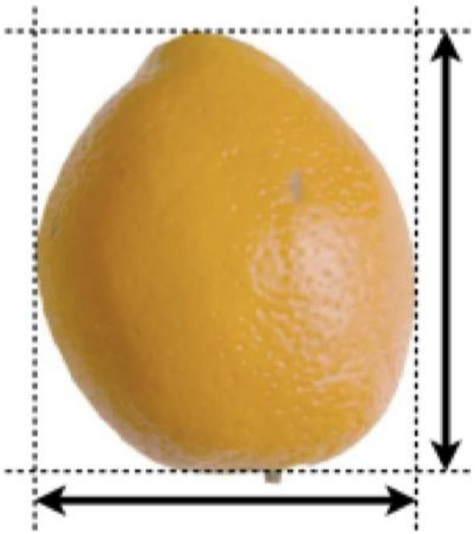
- Assumption: complex patterns and relationships in data can be approximated and learned through a series of interconnected layers of simple computational units (neurons).



k-Nearest Neighbor

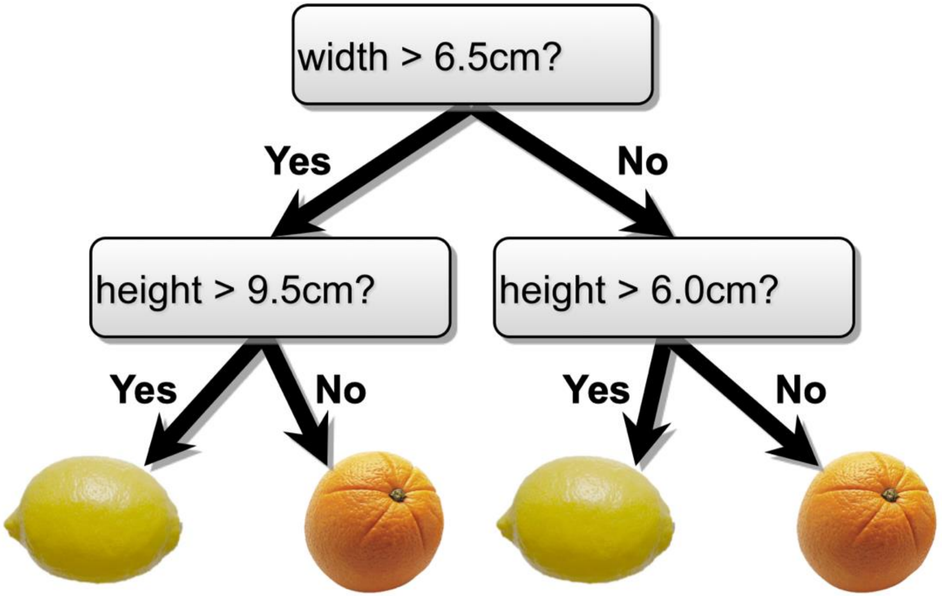
- Assumption: similar data points are likely to belong to the same class, with proximity measured by a distance metric in the feature space.

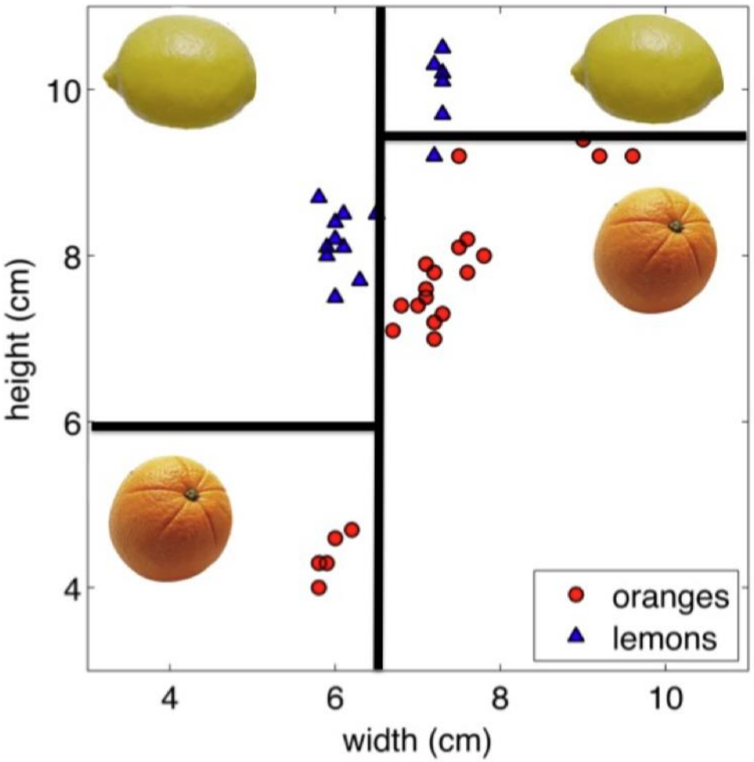




Decisions Trees

- Assumption: complex decisions can be modeled through a series of simple, hierarchical decision rules based on the input features.





Generative Models for Classification

How about this approach: build a model of "how data for a class looks like"

- **Generative** classifiers try to model $p(\mathbf{x}|y)$
- Classification via Bayes rule (thus also called Bayes classifiers)

(prob. of features given label)

(class label)

$p(\mathbf{x}|y) \longleftarrow y$

e.g:
- height
- weight
- color

elefant

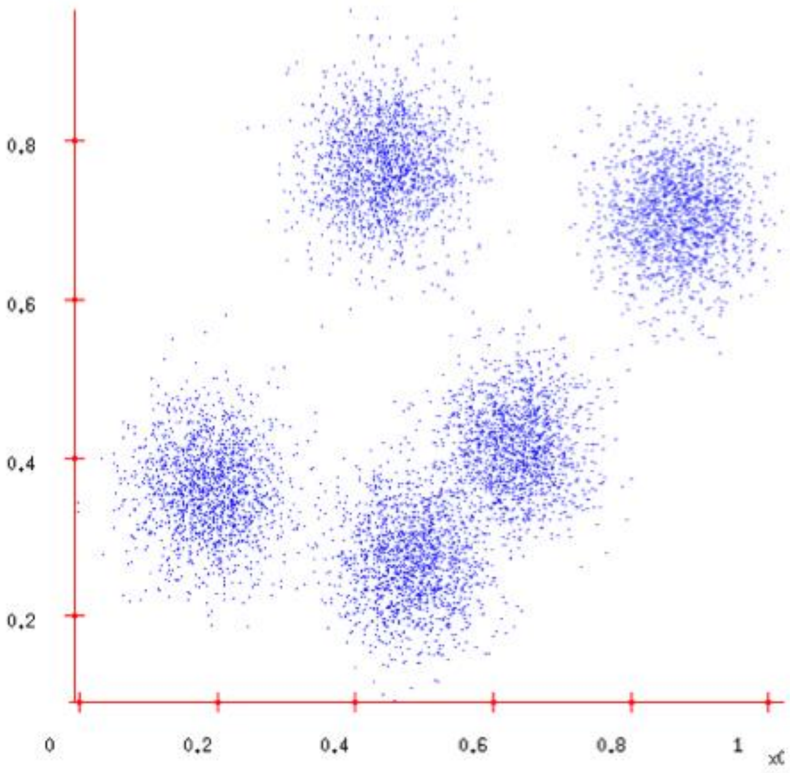
bear

Tries to model:
- How does data look like for a class?

Classification (How?)

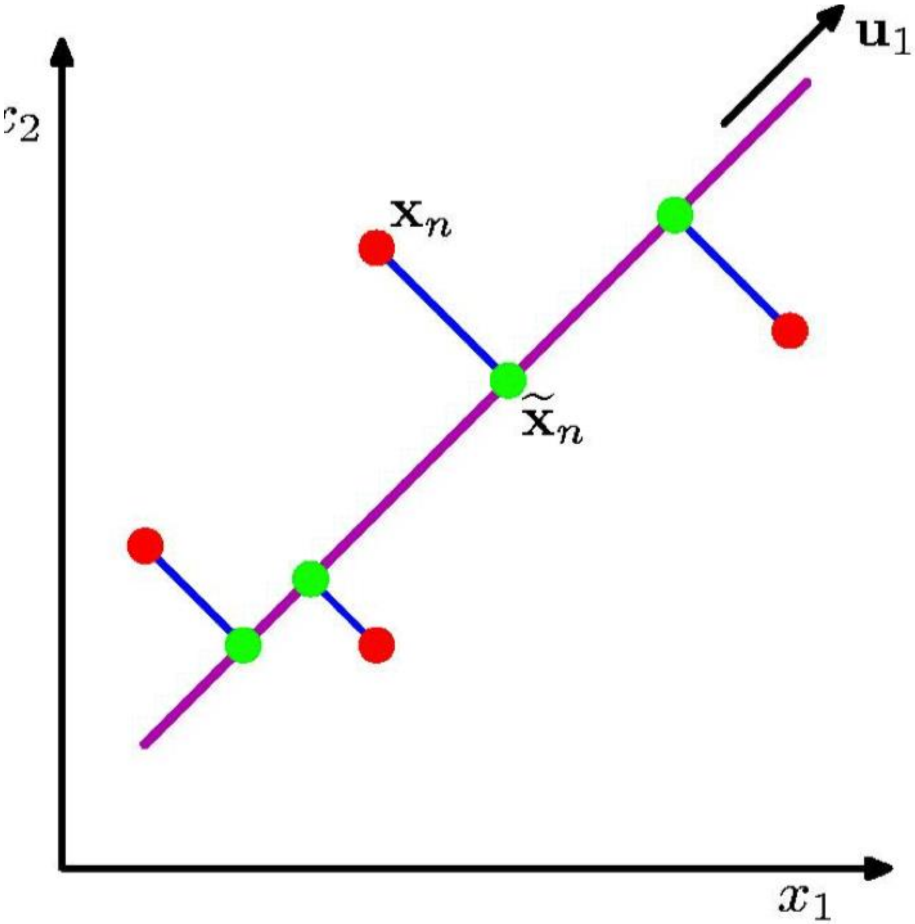
Clustering

- Objective: partition a dataset into k distinct clusters without supervision.



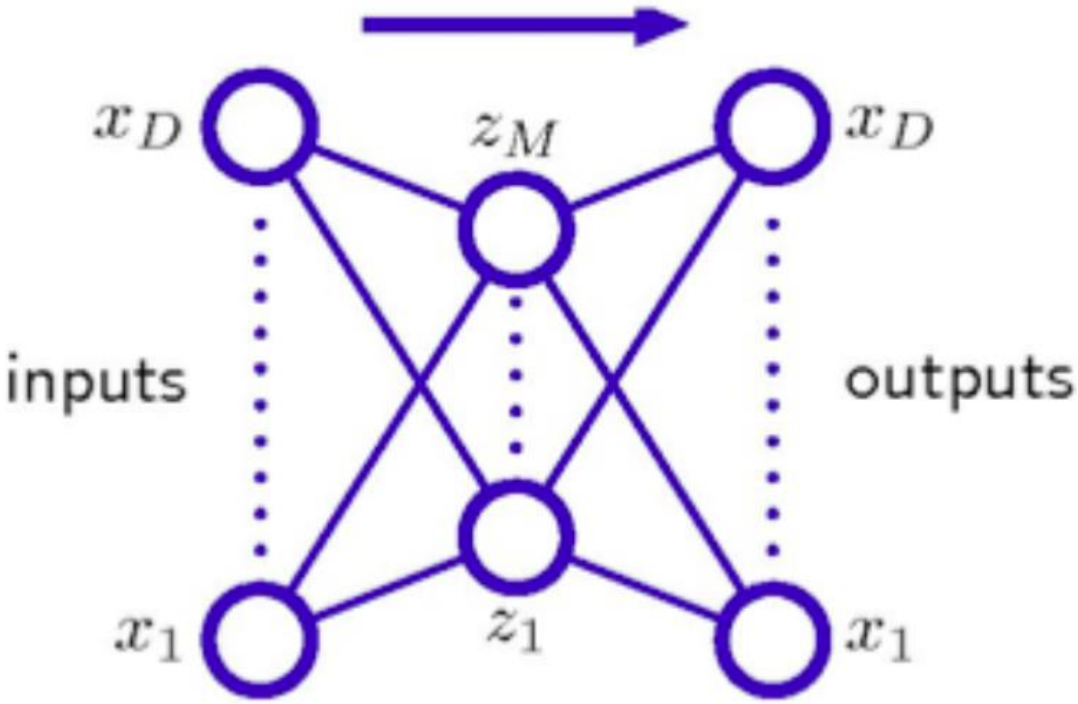
Principal Component Analysis

- Objective: to reduce the dimensionality of a dataset by transforming it into a new set of variables (principal components).



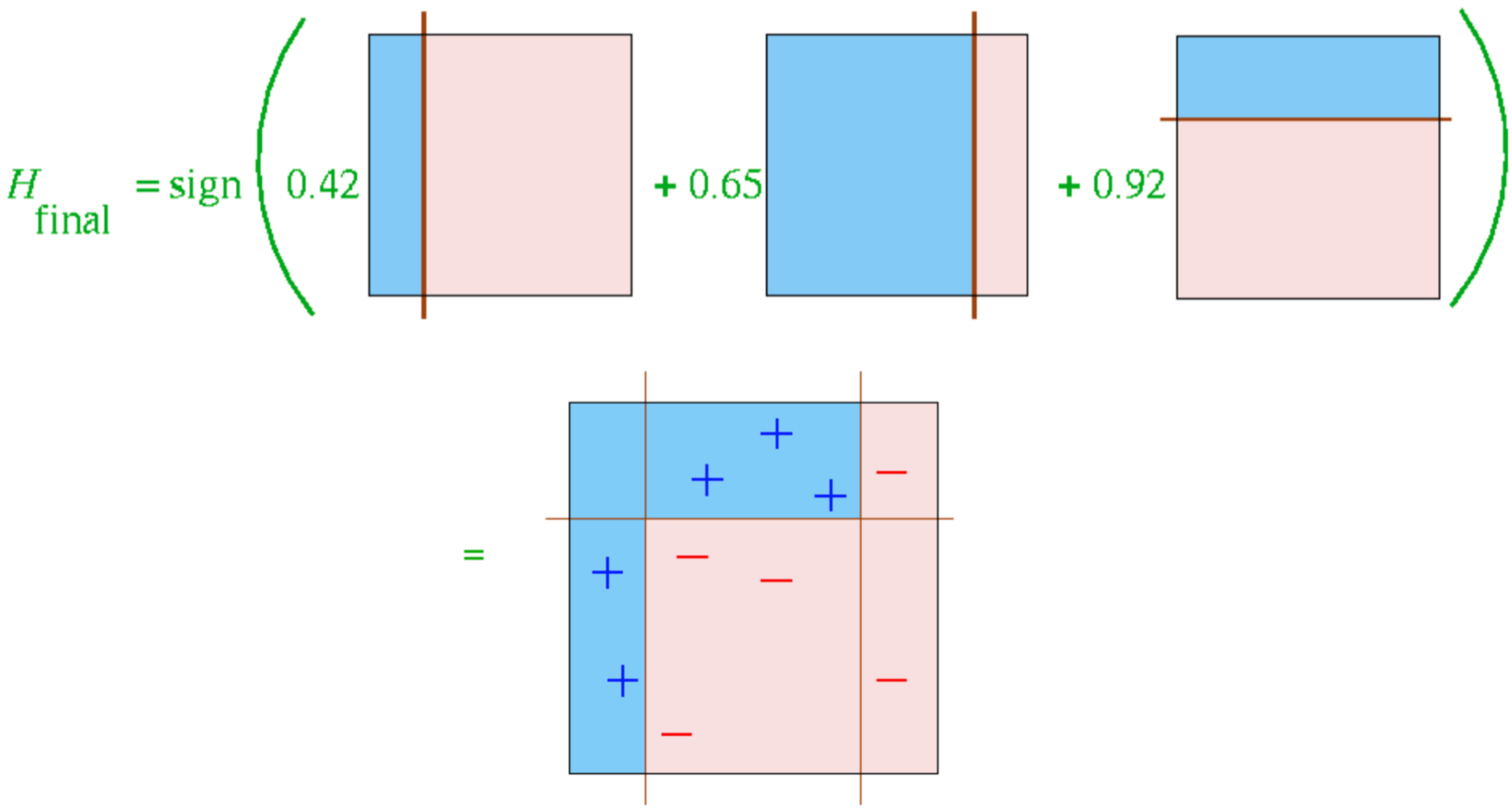
Autoencoders

- Objective: to learn a compressed, low-dimensional representation of input data by encoding it into a latent space and then reconstructing the original input.



Ensemble Methods

- Assumption: Combining multiple diverse models or algorithms can yield more accurate and robust predictions than any single model alone.



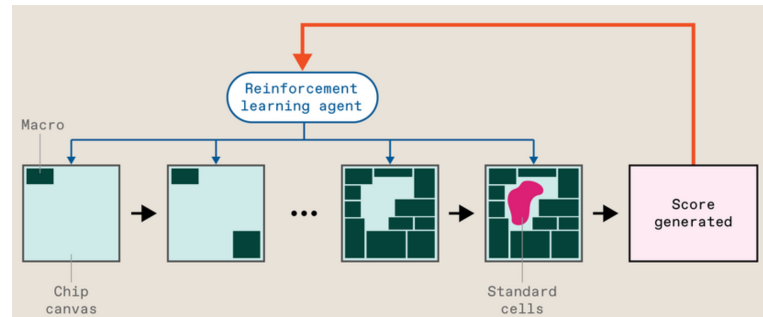
Netflix Prize 2007

- The Netflix Prize was an open competition for the best collaborative filtering algorithm to predict user ratings for films, based on previous ratings without any other information about the users or films.
- Rewarded $50,000 in 2007.
- Original progress prize winner (BellKor) was ensemble of 107 models!
  - *"Our experience is that most efforts should be concentrated in deriving substantially different approaches, rather than refining a simple technique."*
  - *"We strongly believe that the success of an ensemble approach depends on the ability of its various predictors to expose different complementing aspects of the data. Experience shows that this is very different than optimizing the accuracy of each individual predictor."*

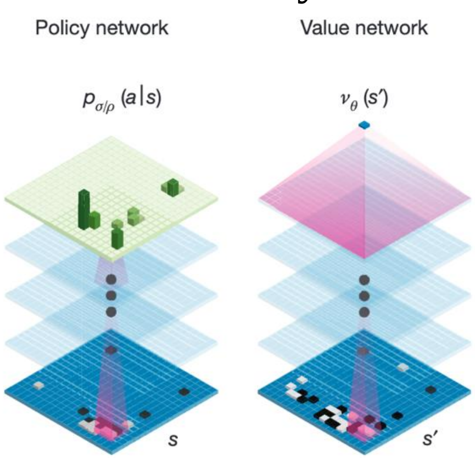
Reinforcement Learning

- Assumption: An agent can learn optimal behavior or policies by interacting with an environment, receiving feedback in the form of rewards or punishments for its actions.

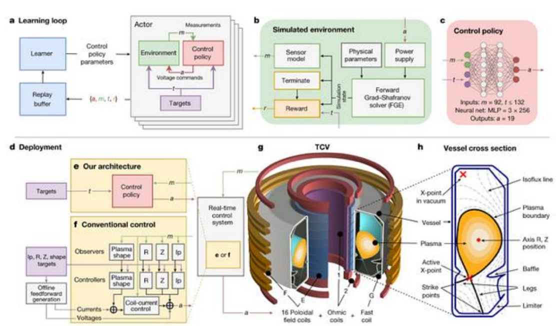
Play Dota



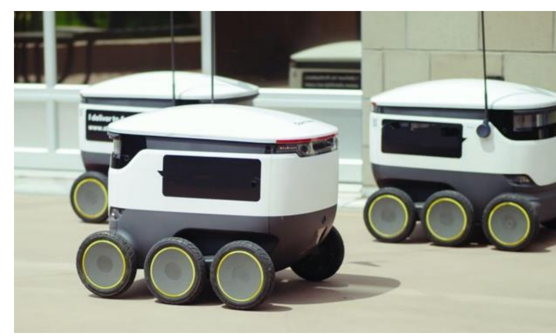
Chip Design



AlphaGo



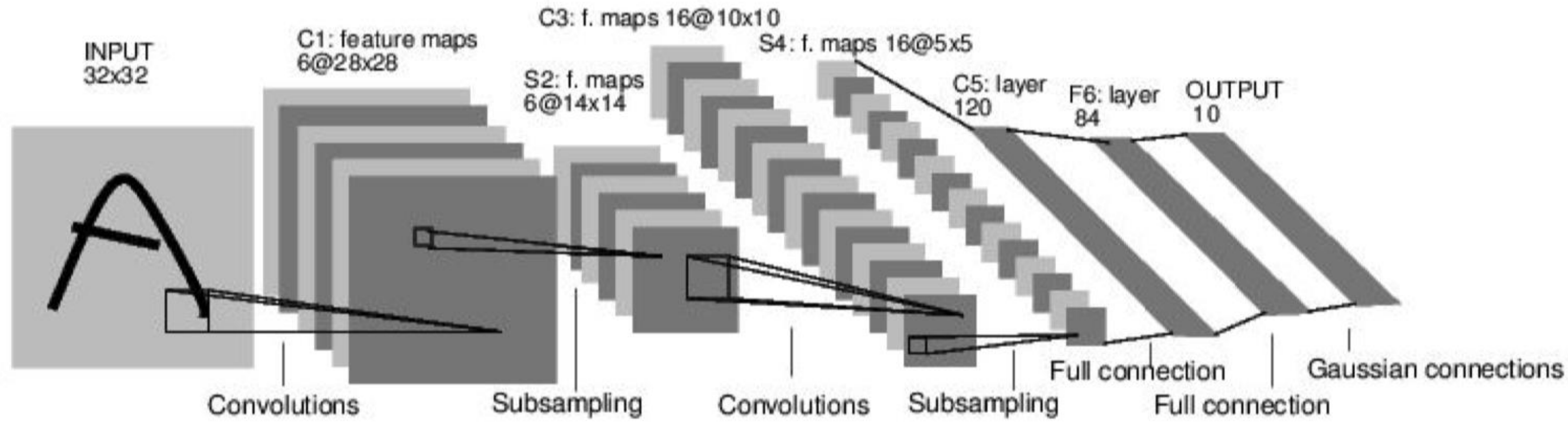
Nuclear Fusion



AWS Deep Racer

Convolutional Neural Networks

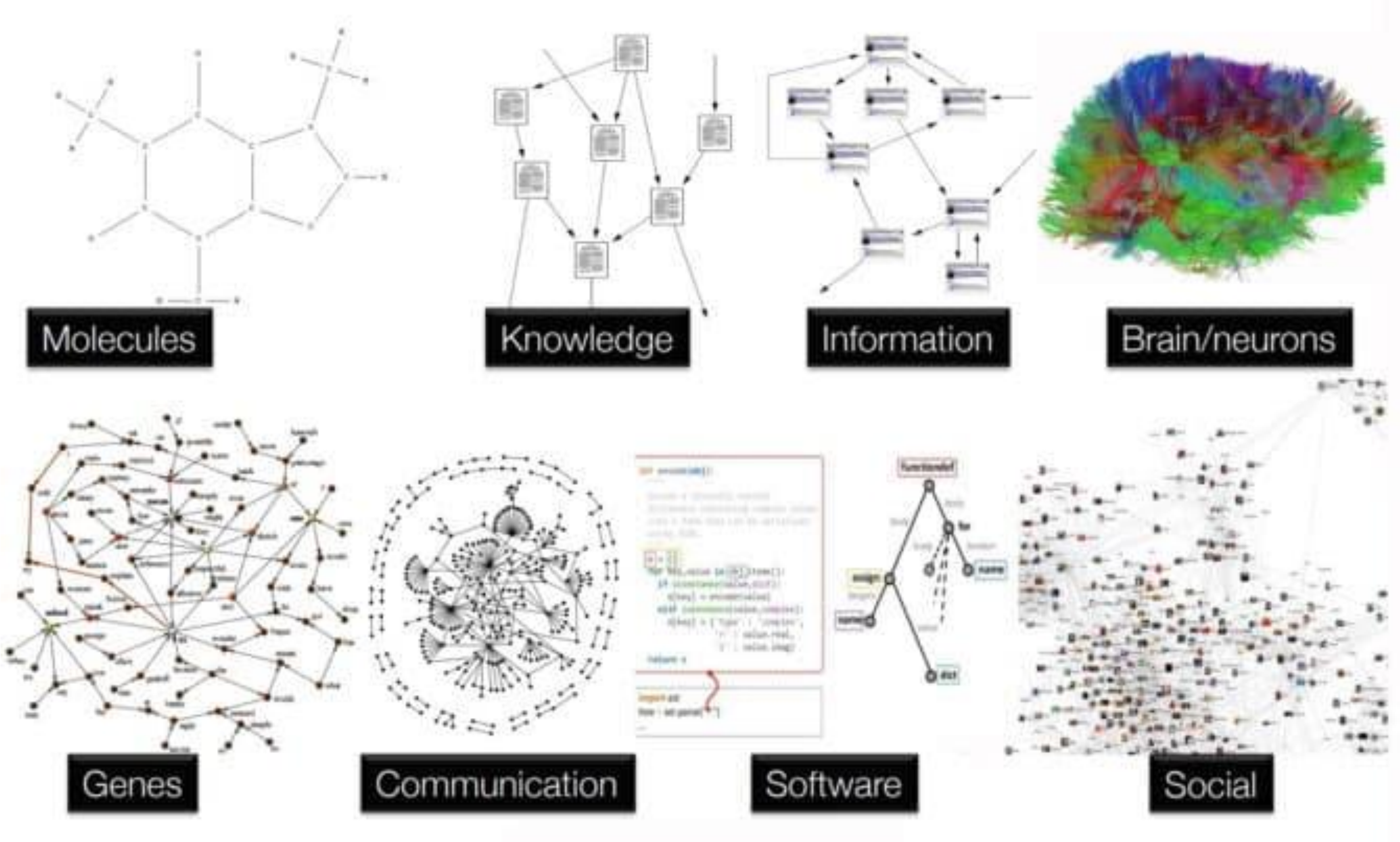
- Assumption: Input data, particularly images, can be effectively processed and analyzed using spatial hierarchies of patterns through localized and parameter-efficient convolutions.



Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e. architecture is [CONV-POOL-CONV-POOL-CONV-FC]

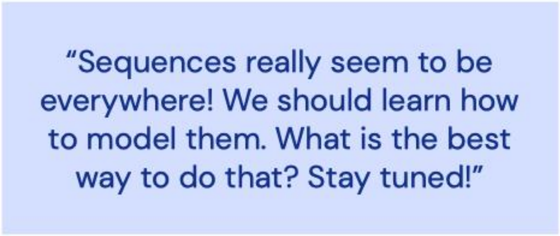
Graph Neural Networks

- Assumption: Data represented as graphs can be effectively analyzed by capturing dependencies and relationships between nodes through message passing or aggregation mechanisms.

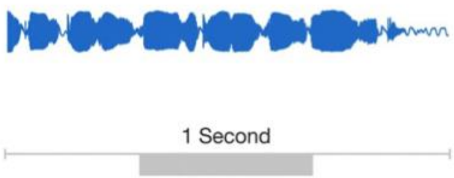


Recurrent Neural Networks

- Assumption: Sequential data has an inherent order, and understanding each element requires knowledge from its previous elements in the sequence.

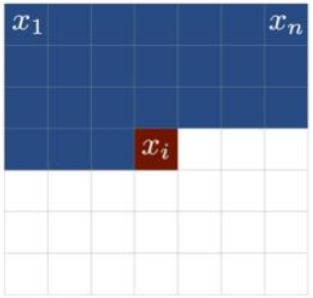


Words, letters

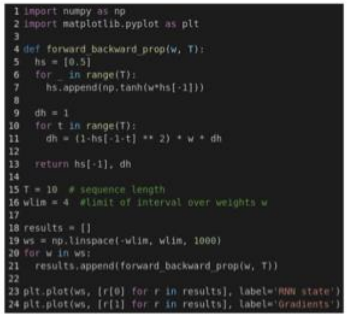


Speech

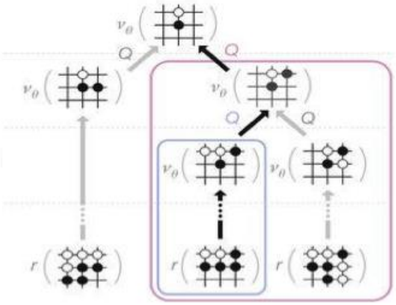
Videos



Images



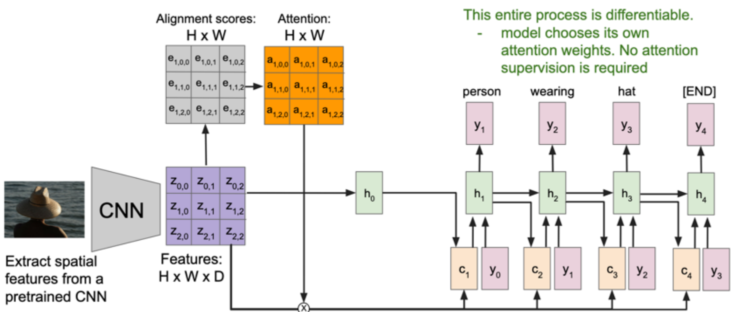
Programs



Decision making

Attention & Transformers

- Assumption: It's beneficial to selectively focus on certain parts of the input while processing others, thereby dynamically prioritizing information based on its relevance.

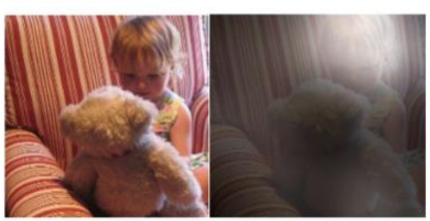


Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

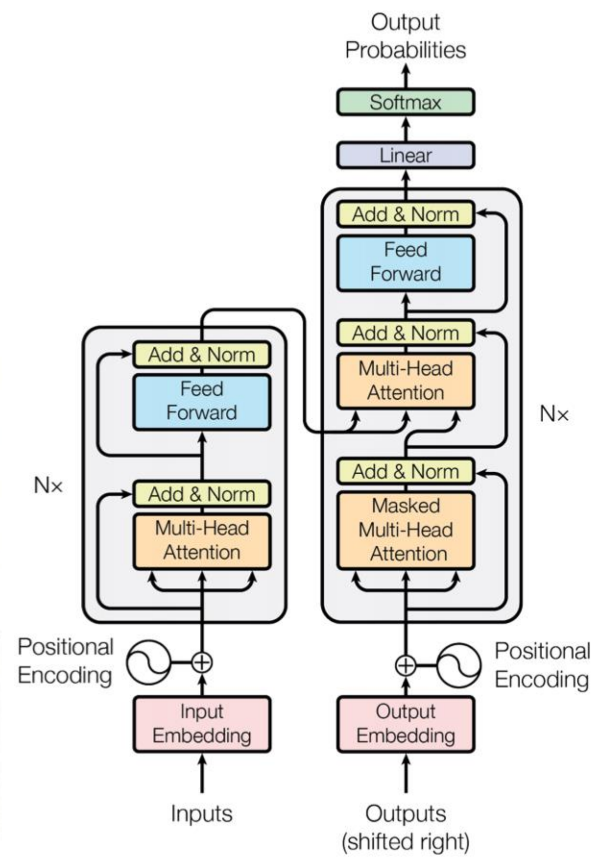
A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.

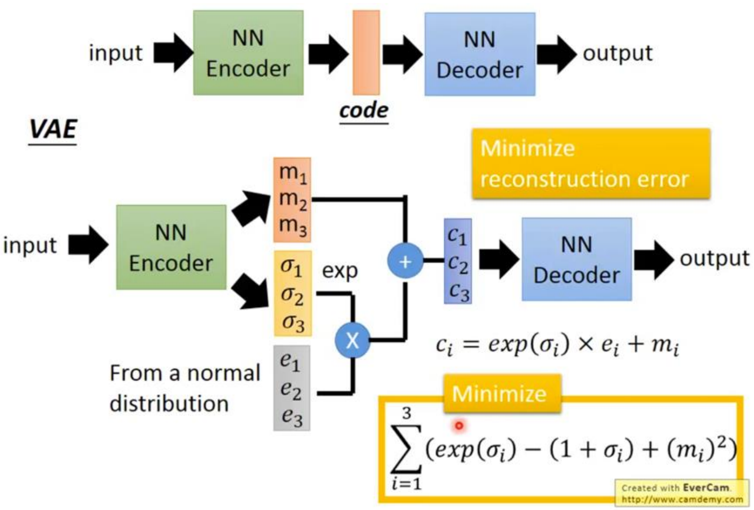
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

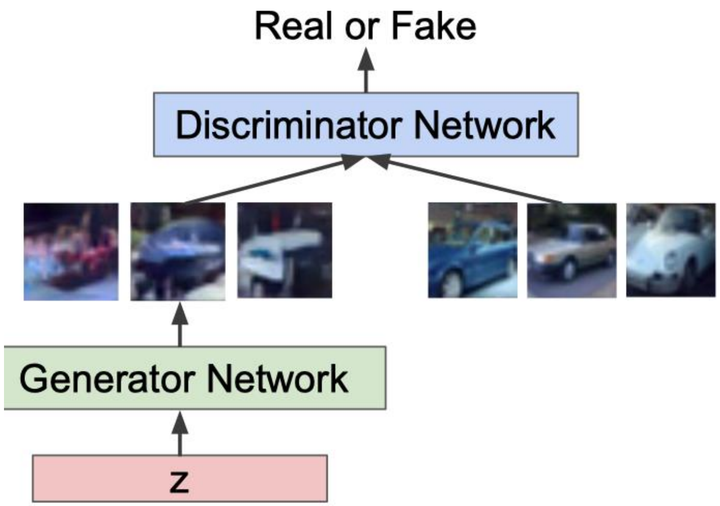


Deep Generative Models

- Variational Autoencoders:



- Generative Adversarial Networks:



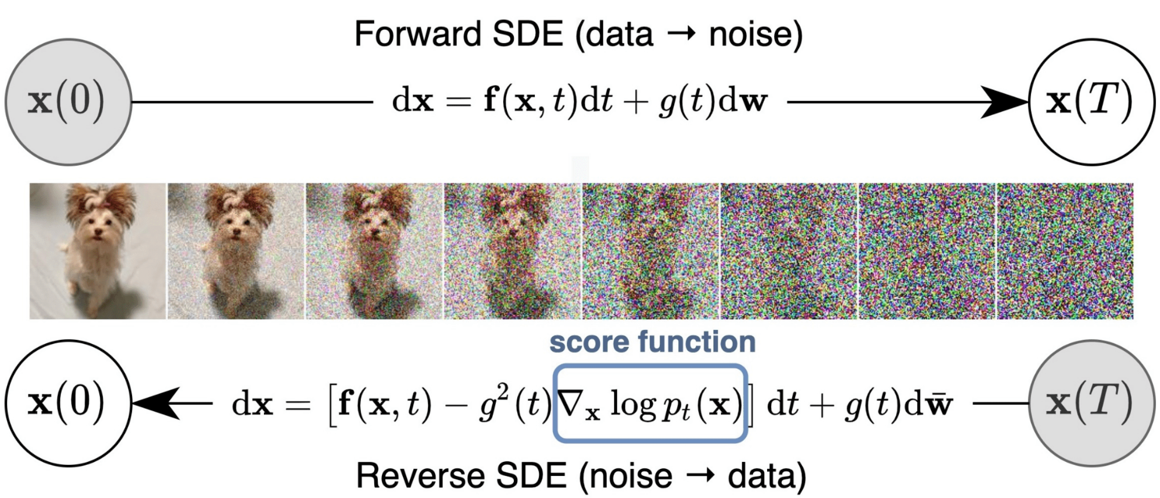
Deep Generative Models

- Diffusion Models:

A transparent sculpture of a duck made out of glass.

A raccoon wearing cowboy hat and black leather jacket is behind the backyard window. Rain droplets on the window.

A bucket bag made of blue suede. The bag is decorated with intricate golden paisley patterns. The handle of the bag is made of rubies and pearls.



Questions?