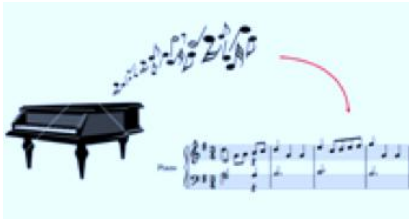Welcome to
The Art of Machine Learning
(ECE 208/408, TEE 408)

Zhiyao Duan
Professor of ECE and CS
University of Rochester

Audio Information Research (AIR) Lab

Machine Understanding of Sounds



MUSIC INFORMATION RETRIEVAL

- Music transcription, alignment
- Source separation
- Generation
- Interactive performance



SPEECH PROCESSING

- Separation and enhancement
- Verification and anti-spoofing
- Emotion analysis
- Diarization
- Text-to-speech
- Voice conversion



ENVIRONMENTAL SOUND UNDERSTANDING

- Sound search by vocal imitation
- Sound event detection
- Source localization
- HRTF modeling
- Smart acoustics



AUDIO-VISUAL PROCESSING

- Talking face generation
- Music performance analysis and generation
- Audio-visual source separation

What is Machine Learning?

- "... the field of study that gives computers the ability to learn without being explicitly programmed."

---- Arthur Samuel, 1959

- "A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

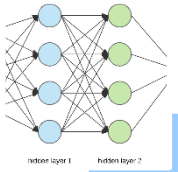
---- Tom Mitchell, 1997

Three Key Elements



Data

- "Experiences"
- Measurements
- Observations
- Annotations
- …



Model

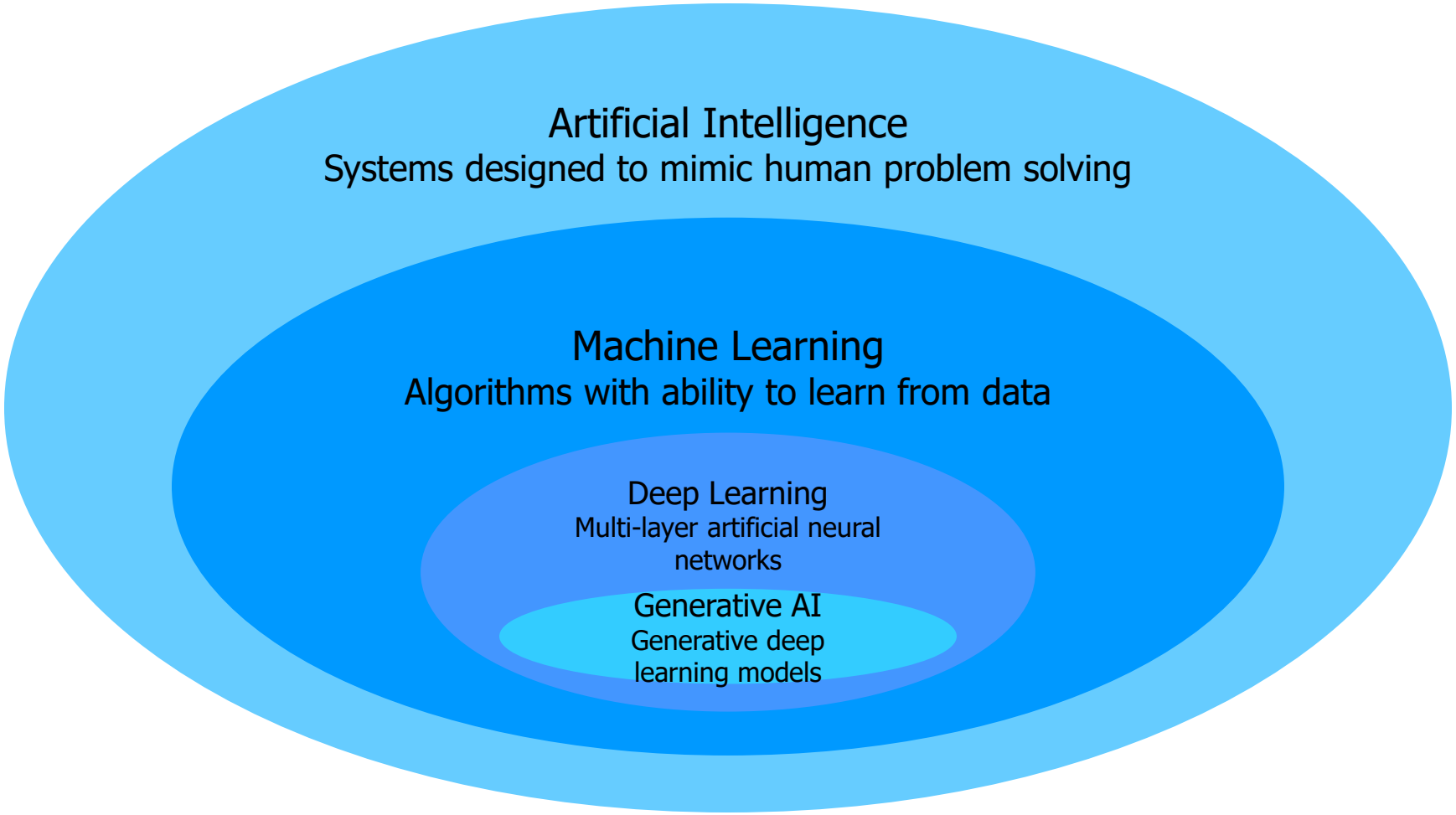
- Mathematical
- Procedural
- Architectural
- …



Algorithm

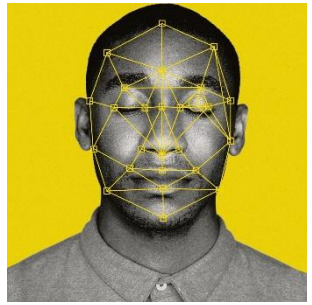
- Random search
- Gradient descent
- Newton's method
- Genetic algorithm
- …

Machine Learning in Context



Many Applications

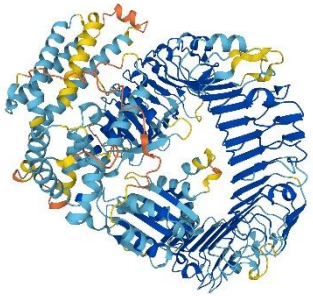




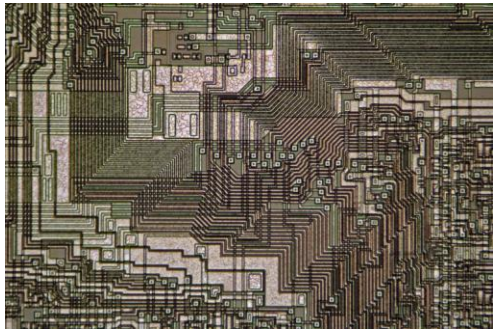


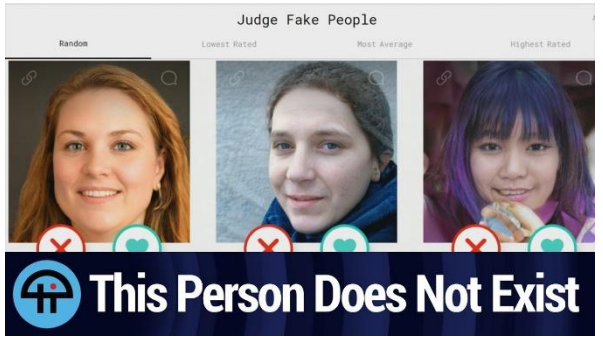


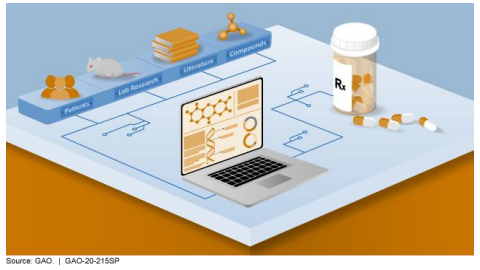


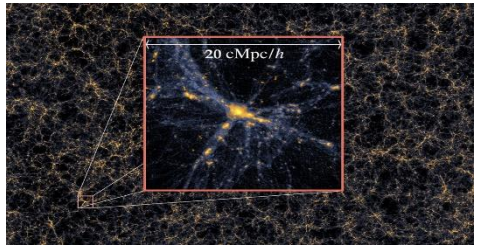






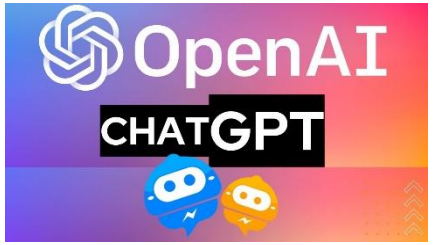














Course Topics

- Fundamental concepts of machine learning
  - Training, validation, testing
  - Overfitting, underfitting, cross validation
  - Bias/variance tradeoff, regularization, generalization
  - Supervised, semi-supervised, unsupervised, reinforcement learning
- Various machine learning models
  - Nearest neighbors, decision trees, linear models, generalized linear models, support vector machines, multi-layer perceptron, convolutional neural networks, recurrent neural networks, transformer, GAN, diffusion model, K-means, principal component analysis, autoencoders, dimensionality reduction, Gaussian mixture models, Q learning, etc.
- Applications of machine learning in engineering problems
  - E.g., Circuit design, salary prediction, maternal health risk assessment, lung ultrasound image classification, music generation

Course Objectives

- Good understanding of fundamental concepts and various models and applications of machine learning
- Build intimate connections between theory and practice

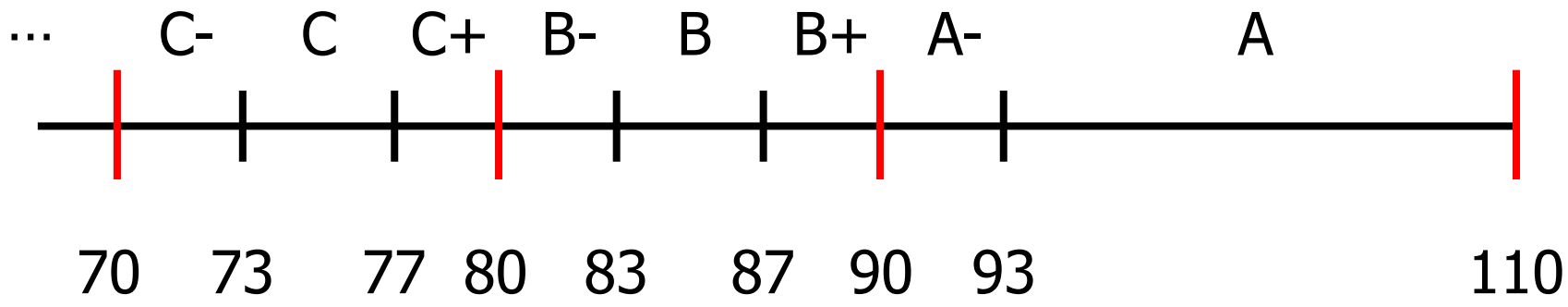
- Gain experience in doing small-scale research projects
- Improve capabilities of problem solving, team-working, presentation, etc.

Assignments

- Total (110 points)
  - Homework (80 points)
    - Programming using Python
  - Final project (30 points)
    - Proposal (5 points)
    - Project update (5 points)
    - Final report (10 points)
    - Presentation/demo (10 points)
  - Attendance (-6 points to 0 points)

Grading

- For students enrolled in 408



- Students enrolled in 208 will get 10 points boost
- No extra credit
- No curve

Important Policies

- Do your own work
  - For HWs
    - You cannot copy code or writeups from other students or external resources. You cannot let others copy your code or writeup either.
    - If the assignment allows you to write your code on top of external code, you need to state clearly which part is written by you in your submission.
    - You cannot use AI tools like ChaptGPT in any component of the work.
  - For the final project
    - You are allowed to use external software packages and code in the implementation of your project, but you need to state clearly which part is your work and cite other work appropriately.
    - When writing the report, you cannot copy phrases with more than 5 consecutive words from other papers.
    - When paraphrasing other work, cite the work appropriately.
    - You are allowed to use AI tools in coding and polishing your final report and presentation, but you cannot use them to draft your final report or presentation. You need to state clearly in your final report how you used AI tools.

Important Policies

- Late homework: 20% of full credit deduction / day
- Extension is possible for two kinds of reasons:
  - 1) illness and family emergency: typical extension is 3 days
  - 2) school-related travels: extension can be as long as the travel length
- Please email me before the submission deadline if extension is needed

Important Policies

- Attendance
  - In-class discussions are very helpful for understanding the course content
  - There will be 12 random checks of attendance
  - Each miss will result in 0.5 points deduction

Prerequisites

- General programming
  - E.g., ECE 114
- Linear algebra
  - E.g., MATH 165

- Preferred but not required
  - Probability and statistics, e.g., ECE 270

Textbooks

- **LWLS** - Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, Thomas B. Schön, Machine Learning: A First Course for Engineers and Scientists, Cambridge University Press, 2022. <PDF>
- **WBK** - Jeremy Watt, Reza Borhani, Aggelos K. Katsaggelos, Machine Learning Refined: Foundations, Algorithms, and Applications (2nd Edition), Cambridge University Press, 2020. <Companion Website>
- **Mitchell** - Tom M. Mitchell, Machine Learning, McGraw-Hill Education, 1997. <PDF>
- **GBC** - Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press. <PDF>
- **BB** - Christopher M. Bishop and Hugh Bishop, Deep Learning: Foundations and Concepts, Springer, <Companion Website>
- **Géron** - Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition. <Online Book> accessible with UR email.

Tips for Studying This Course

- This is a challenging course!

- Come to lectures and be active
  - Helps you grasp the main ideas quickly
- Devote enough time after class
  - Reading + implementation
  - Expect 10 hours home study time each week
- Start doing homework early
  - Discuss with others, TAs, and me
  - Discuss on Blackboard
  - Submit homework on time!

Course Information

- Course website: all materials
  - https://hajim.rochester.edu/ece/sites/zduan/teaching/ece408/index.html
- Blackboard: announcement, discussions, assignment submission, grade tracking
- Instructor office hour: 2-3 PM on Tuesdays in CSB 720; additional time by appointments
- TA office hours: Jianan Wu, 1:30-3:30 PM on Fridays in CSB 401A

Ready? Let's Go!



Let's get to know each other first!

Let's design a spam filter

- Engineering approach
- Step 1: look at some normal and spam email examples
- Step 2: design some features and rules for detecting spam
- Step 3: Evaluate these rules on new emails



(https://www.securitymetrics.com/blog/7-ways-recognize-phishing-email)

Machine Learning Approach

- Step 1: Collect normal and spam emails as training data
- Step 2: Extract features that might be useful
- Step 3: Train a classifier using the training data
- Step 4: Evaluate the classifier on unseen emails

- Hand-crafted rules are replaced by trained machine learning models
  - We need: data, model, and (training) algorithm

- In deep learning, hand-crafted features are often also replaced by automatically learned features, and this is called feature learning.

Data Types

- Numerical data – quantitative
  - Continuous: e.g., weight, voltage
  - Discrete: e.g., guitar fret, #people in a picture

- Categorical data – qualitative
  - Nominal: e.g., speech communication channel: megaphone, land phone, cellphone, walkie talkie, satellite, and VoIP
  - Ordered: e.g., sunny, partly cloudy, cloudy, flurries, snow, blizzard
  - Categorical data is often mapped to discrete numbers, but note that distances do not make sense

Data Examples

- An object (or data sample, example) often has multiple dimensions

Weight (g)	Color	Shape	Taste (1-5)	Calories	Water (%)	Label
200	red	round	4	100	85	apple
140	orange	round	5	66	86	orange
120	yellow	long	5	105	75	banana
150	green	round	1	70	90	apple
110	green	long	2	100	73	banana
200	orange	round	3	85	90	orange

- Notation for a data example: $(\boldsymbol{x}, y)$, where $\boldsymbol{x}$ is called the feature vector, and y is called its label
- Matrix notation for a set of samples: $(\boldsymbol{X}, \boldsymbol{y})$

Machine Learning Paradigms

- Supervised learning
  - Given examples $(\boldsymbol{X}, \boldsymbol{y})$, learn $f\colon \boldsymbol{x} \mapsto y$
- Unsupervised learning
  - Given examples $\boldsymbol{X}$, discover structures of data
- Semi-supervised learning
  - Given examples $(\boldsymbol{X}^l, \boldsymbol{y}^l)$ and $\boldsymbol{X}^u$, learn $f\colon \boldsymbol{x} \mapsto y$
- Reinforcement learning
  - Given sequences of (state, action, immediate reward): $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{r})$
  - Learn optimal behavior $f\colon \boldsymbol{s} \mapsto \boldsymbol{a}$ that is good in the long run

Supervised Learning

- Given examples $(\boldsymbol{X}, \boldsymbol{y})$, learn f: $\boldsymbol{x} \mapsto y$

- Classification
  - Label y is categorical or discrete
  - E.g., classifying fruits from their measurements
- Regression
  - Label y is continuous
  - E.g., predicting longevity from health status, social-economic status, etc.

- Dataset split
  - Training: used to learn the mapping f
  - Validation: used to tune hyperparameters of f
  - Test: used to evaluate performance of f on unseen data

Unsupervised Learning

- Given examples $\boldsymbol{X}$, discover structures of data

- Density estimation: how is data distributed in the data space?
  - E.g., finding out the distribution of SAT scores in college applications
- Clustering: which data examples form a cluster?
  - E.g., sorting out types of insects
- Dimensionality reduction: find the lower dimensional subspace or manifold where the data resides
  - E.g., reducing a 4K image (8.3M pixels) to a 100-d vector for scene classification
- Data generation: sample from data distribution
  - E.g., https://thispersondoesnotexist.com/

Semi-Supervised Learning

- Given examples $(\boldsymbol{X}^l, \boldsymbol{Y}^l)$ and $\boldsymbol{X}^u$, learn $f\colon \boldsymbol{x} \mapsto \boldsymbol{y}$

- Same goal as supervised learning, but some training data do not have labels
  - E.g., classifying topics of text documents
  - E.g., people tagging in iPhone photos

- Why/how do the unlabeled data help?
  - Data have structures (e.g., clusters)
  - Data examples in the same cluster tend to have the same label
  - Classification boundaries tend to go through low-density regions

Reinforcement Learning

- Given sequences of (state, action, immediate reward): $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{r})$
- Learn optimal behavior $f: \boldsymbol{s} \mapsto \boldsymbol{a}$ that is good in the long run

- Key property: rewards for an action are delayed and ambiguous
- E.g., chess, Atari, self-driving cars