CS480/680: Introduction to Machine Learning

Lec 00: Introduction

Yaoliang Yu

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS
DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

May 6, 2024

Course Information

- Instructor: Yao-Liang Yu (yaoliang.yu@uwaterloo.ca)
- Office hours: MW 1-2 pm (Eastern) or by email appointment
- TA: Haoye Lu (h229lu), Yiwei Lu (y485lu), Saber Malekmohammadi (s3malekm) x 2, Argyris Mouzakis (amouzaki), Spencer Szabados (sszabado)
- Website: cs.uwaterloo.ca/~y328yu/mycourses/480
  slides, notes, assignments, policy, etc.
- Piazza: piazza.com/uwaterloo.ca/spring2024/cs480cs680
  announcements, questions, discussions, etc.
- Learn: learn.uwaterloo.ca/d2l/home/1021993
  assignments, solutions, grades, etc.

Prerequisites

- Basic linear algebra, calculus, probability, algorithm
  - CM339 / CS341 or SE 240; STAT 206 or 231 or 241
  - some relevant books on course website
- Coding

https://www.python.org/

https://julialang.org/

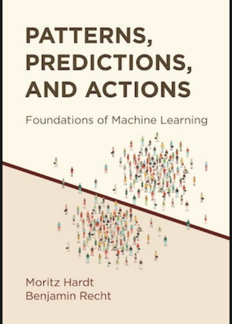
"Coding to programming is like typing to writing."

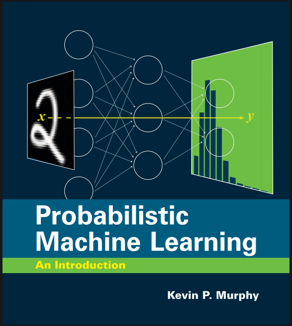
— *Leslie Lamport*

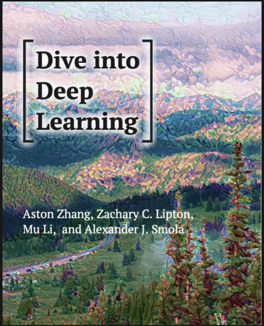
Textbooks

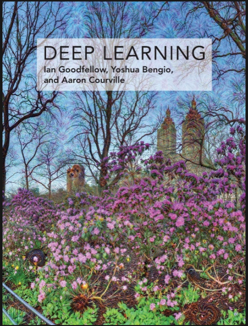
☞ links

- No required textbook
- Notes, slides, and code will be posted on the course website
- Some fine textbooks for the ambitious ones:







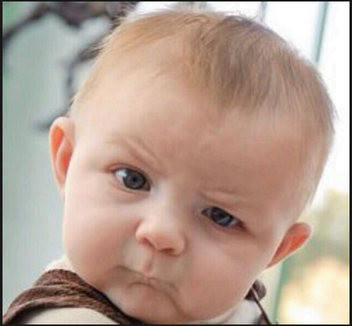


Workload

- Roughly 24 lectures, each lasting 90 minutes
- Expect 4 assignments, approx. 1 every 3 weeks
  - 18 points each; total: 72
- Kaggle competition: 14 (ranking) + 14 (4-page report)
  - CS680: upon approval can substitute with a course project
- Small, constant progress every week
- Submit on LEARN. Submit early and often
  - typeset using LaTeX is recommended

Policy

- Do your work **independently and individually**
  - discussion is fine, but no sharing of text or code
  - explicitly acknowledge any source that helps you
- Ignorance is no excuse
  - good online discussion, more on course website
- Serious offense will result in expulsion. . .
- NO late submissions!
  - except hospitalization, family urgency, . . . notify beforehand
  - one-time, two-day short-term absence for CS480: email Saber (s3malekm)
- Appeal within two weeks



Overview

A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

J. McCarthy, Dartmouth College
M. L. Minsky, Harvard University
N. Rochester, I.B.M. Corporation
C.E. Shannon, Bell Telephone Laboratories

August 31, 1955

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

fulltext

- Automatic Computers
- How Can a Computer be Programmed to Use a Language
- Neuron Nets
- Theory of the Size of a Calculation
- Self-Improvement
- Abstractions
- Randomness and Creativity

MEASURES OF THE VALUE OF INFORMATION

By John McCarthy

Dartmouth College, Hanover, New Hampshire

Communicated by Claude Shannon, July 12, 1956

1. *Introduction.*—Our knowledge of a future event may take the form of a set of probabilities $p_1, \ldots, p_n$. For example, we might have probabilities of $^3/_8$, $^1/_8$, and $^1/_2$ for rain, snow, and clear as tomorrow's weather. In communication theory our interest is in the various events only as carriers of a coded message. For this purpose Shannon's[1] entropy $-\sum p_i \log p_i$ is the appropriate measure of our uncertainty, and a function $A \sum p_i \log p_i + B$ is a good measure of what it is worth to be given these probabilities. In our weather example we care which event occurs. Furthermore, we may be more interested in whether the sky is clear than in whether rain or snow occurs if the weather is bad. In this paper we show that any convex function of a set of probabilities may serve as a measure of the value of information and that two such functions are equivalent in an appropriate sense if and only if they differ by a linear function.

2. *The Forecaster and His Client.*—We get our quantitative measures of the value of information from a situation in which a client pays a forecaster for predictions of a future event according to the following rules:

(i) The forecaster gives the client probabilities $q_1, \ldots, q_n$ for the events, where $\sum q_i = 1$.

(ii) The client takes action on the basis of these probabilities, and one of the possible events occurs.

(iii) If the ith event occurs, the client pays the forecaster $f_i(q_1, \ldots, q_n)$, which is abbreviated $f_i(q)$.

(iv) We assume that neither the forecaster nor the client can influence the predicted event, although the forecaster can make experiments to help predict it, and the client gets an amount which depends on both the action he takes and on the event which occurs. In what follows, it is assumed that the forecaster and the client both wish to maximize the expected value of their incomes.

Assuming that to the forecaster the probabilities of the possible events are $p_1, \ldots, p_n$, his expectation is $\sum p_i f_i(q)$ if he tells the client the q's. A payoff rule is said to "keep the forecaster honest" if, regardless of the value of $p = (p_1, \ldots, p_n)$, the forecaster's expectation is maximized if and only if he puts $q = p$, i.e., $q_i = p_i$ for each i.

Theorem 1. *A payoff rule keeps the forecaster honest if and only if* $f_i(q) = (\partial/\partial q_i) f(q)$, *where* $f(q)$ *is a convex function of* q *which is homogeneous of the first degree. The expectation of an honest forecaster is then* $\sum p_i f_i(p) = f(p)$.

We omit the proof. The derivative has to be taken in a suitable generalized sense. $f(q)$ is called a "payoff function" if it satisfies the conditions of Theorem 1.

I. J. Good[2] considered the problem of paying the forecaster with the restriction that $f_i(q) = F(q_i)$, i.e., the payoff depends only on the probability assigned to the event which actually occurred. He showed that putting $F(x) = A \log x + B$ keeps the forecaster honest, and Gleason (unpublished) showed that this is the only $F(x)$ which does. The forecaster's expectation is then $A \sum p_i \log p_i + B$, i.e., he is paid a fixed fee *minus* the expected uncertainty about the event after his prediction.

3. *The Client's Expectation.*—Suppose that on the basis of the forecaster's prediction the client chooses the jth of the actions open to him and that his payoff if the ith event occurs is a_{ij}. His expectation will be $g(p) = \max \sum_i a_{ij} p_i$ if j is chosen optimally.

From the theory of convex functions we have

Theorem 2. *Any function* $g(p)$ *defined for* $p_1 \geq 0, \ldots, p_n \geq 0$ *which is convex and homogeneous of the first degree can be written in the form* $\max_j \sum a_{ij} p_i$. *Unless* $g(p)$ *is piecewise linear, there will have to be an infinite number of actions* j.

If we put $f(p) = g(p)$, the client is eliminated from the picture, since under this condition he turns all his gains over to the forecaster and is reimbursed for all his losses. This is not a satisfactory solution to the problem, so let us see what payoffs f are equivalent in their effect on the forecaster's efforts to get information.

4. *The Forecaster's Experiments.*—Assume that the forecaster has a priori probabilities $r_1, \ldots, r_n$ for the events, that he has a choice of m experimental procedures with expected costs to him of $c_1, \ldots, c_m$, and that the conditional probability of the kth outcome of the hth experiment given that the ith event will occur is s_{khi}. The experiment chosen by the forecaster will depend on the c's, the s's, and the r's and on the payoff function chosen by the client. We call two payoff rules equivalent if, for any set of c's, s's, and r's, they lead to the same choice of experiment by the forecaster.

Theorem 3. $f(q)$ *and* $f^*(q)$ *are equivalent if and only if* $f(q) = f^*(q) + \sum a_i q_i$, *i.e., if the two payoff functions differ by a linear function of the* q*'s.*

The proof is omitted. If f and f^* are equivalent, then $f_i(q) = f_i^*(q) + a_i$, so that the payoff rules differ by an amount which depends only on the event which occurs and not on the forecaster's prediction. The forecaster's and client's interests will be identical if we put $f(q) = g(q) + \sum a_i q_i$. The a_i's are subject to negotiation between the client and the forecaster, and they determine both a base level of payment and also a betting relation between the client and forecaster. If f is normalized so that $f(1, 0, \ldots, 0) = f(0, 1, \ldots, 0) = \ldots$, the payment for a precise correct prediction is independent of the event predicted.

5. *Conclusion.*—The foregoing analysis shows that any convex function of a set of probabilities will, under appropriate circumstances, be a measure of the value of the information contained in a set of probabilities in the sense that it is an appropriate payment to a forecaster who furnishes the probabilities.

The intuitive content of the convexity restriction is that it is always a good idea to look at the outcome of an experiment if it is free. For suppose that the experiment has two outcomes, A and A^*, which would give one probabilities p and p^* for the event in question. Let t be the probability that A is the outcome. If we decide not to look, our expectation is $f(tp + (1 - t)p^*)$, while if we decide to look, our expectation is $tf(p) + (1 - t)f(p^*)$.

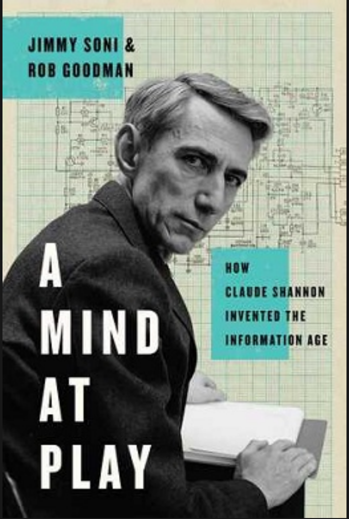
Finally, we remark that there are yet more general ways of paying the forecaster. For example, the client may agree to pay a certain fraction α of the costs of experimentation. Then the payoff function can be scaled down by a factor α with the identity of interests still preserved. We hope to treat these matters on another occasion.

[1] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication* (Urbana: University of Illinois Press, 1949).

[2] I. J. Good, "Rational Decisions," *J. Roy. Stat. Soc., B*, Vol. 14, No. 1, 1952.

Claude Shannon (1916–2001)

- Documentary
- Oral history
- Claude E. Shannon: Founder of Information Theory
- A Chess-Playing Machine
- Claude E. Shannon: Unicyclist, juggler and father of information theory
- Interchange between Kolmogorov and Shannon, recounted by Vitushkin, page 20...



A. G. Vitushkin. "On Hilbert's thirteenth problem and related questions". *Russian Mathematical Surveys*, vol. 59, no. 1 (2004), p. 11.

What is Machine Learning (ML)?

"Machine learning is 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 class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." — *Tom Mitchell (1998)*

Some Studies in Machine Learning Using the Game of Checkers

Abstract: Two machine-learning procedures have been investigated in some detail using the game of checkers. Enough work has been done to verify the fact that a computer can be programmed so that it will learn to play a better game of checkers than can be played by the person who wrote the program. Furthermore, it can learn to do this in a remarkably short period of time (8 or 10 hours of machine-playing time) when given only the rules of the game, a sense of direction, and a redundant and incomplete list of parameters which are thought to have something to do with the game, but whose correct signs and relative weights are unknown and unspecified. The principles of machine learning verified by these experiments are, of course, applicable to many other situations.

Introduction

The studies reported here have been concerned with the programming of a digital computer to behave in a way which, if done by human beings or animals, would be described as involving the process of learning. While this is not the place to dwell on the importance of machine-learning procedures, or to discourse on the philosophical aspects,[1] there is obviously a very large amount of work, now done by people, which is quite trivial in its demands on the intellect but does, nevertheless, involve some learning. We have at our command computers with adequate data-handling ability and with sufficient computational speed to make use of machine-learning techniques, but our knowledge of the basic principles of these techniques is still rudimentary. Lacking such knowledge, it is necessary to specify methods of problem solution in minute and exact detail, a time-consuming and costly procedure. Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort.

• *General methods of approach*

At the outset it might be well to distinguish sharply between two general approaches to the problem of machine learning. One method, which might be called the *Neural-Net Approach,* deals with the possibility of inducing learned behavior into a randomly connected switching net (or its simulation on a digital computer) as a result of a reward-and-punishment routine. A second, and much more efficient approach, is to produce the equivalent of a highly organized network which has been designed to learn only certain specific things. The first method should lead to the development of general-purpose learning machines. A comparison between the size of the switching nets that can be reasonably constructed or simulated at the present time and the size of the neural nets used by animals, suggests that we have a long way to go before we obtain practical devices.[2] The second procedure requires reprogramming for each new application, but it is capable of realization at the present time. The experiments to be described here were based on this second approach.

• *Choice of problem*

For some years the writer has devoted his spare time to the subject of machine learning and has concentrated on the development of learning procedures as applied to games.[3] A game provides a convenient vehicle for such study as contrasted with a problem taken from life, since many of the complications of detail are removed. Checkers, rather than chess,[4-7] was chosen because the simplicity of its rules permits greater emphasis to be placed on learning techniques. Regardless of the relative merits of the two games as intellectual pastimes, it is fair to state that checkers contains all of the basic characteristics of an intellectual activity in which heuristic procedures and learning processes can play a major role and in which these processes can be evaluated.

Some of these characteristics might well be enumerated. They are:

(1) The activity must not be deterministic in the practical sense. There exists no known algorithm which will guarantee a win or a draw in checkers, and the complete

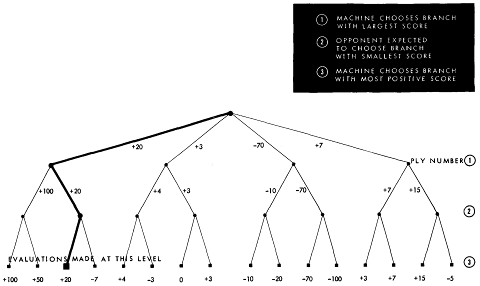


Figure 2 **Simplified diagram showing how the evaluations are backed-up through the "tree" of possible moves to arrive at the best next move. The evaluation process starts at ③.**

rates as being better than the book move and the number it rates as being poorer. The sides are then reversed and the process is repeated. At the end of a book game a correlation coefficient is computed, relating the machine's indicated moves to those moves adjudged best by the checker masters.[16]

It should be noted that the emphasis throughout all of these studies has been on learning techniques. The temptation to improve the machine's game by giving it standard openings or other man-generated knowledge of playing techniques has been consistently resisted. Even when book games are played, no weight is given to the fact that the moves as listed are presumably the best possible moves under the circumstances.

For demonstration purposes, and also as a means of avoiding lost machine time while an opponent is thinking, it is sometimes convenient to play several simultaneous games against different opponents. With the program in its present form the most convenient number for this purpose has been found to be six, although eight have been played on a number of occasions.

Games may be started with any initial configuration for the board position so that the program may be tested on end games, checker puzzles, et cetera. For nonstandard starting conditions, the program lists the initial piece arrangement. From time to time, and at the end of each game, the program also tabulates various bits of statistical information which assist in the evaluation of playing performance.

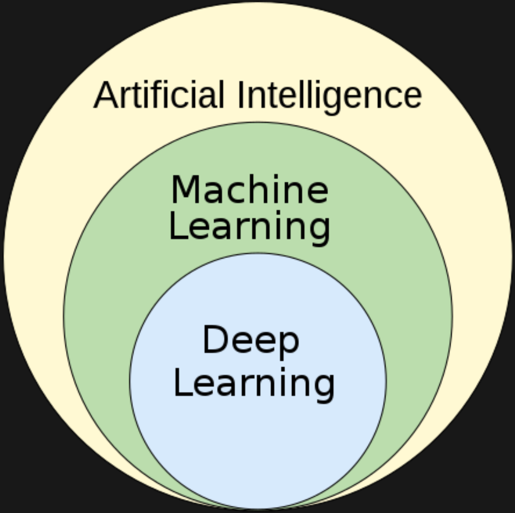
Numerous other features have also been added to make the program convenient to operate (for details see Appendix A), but these have no direct bearing on the problem of learning, to which we will now turn our attention.

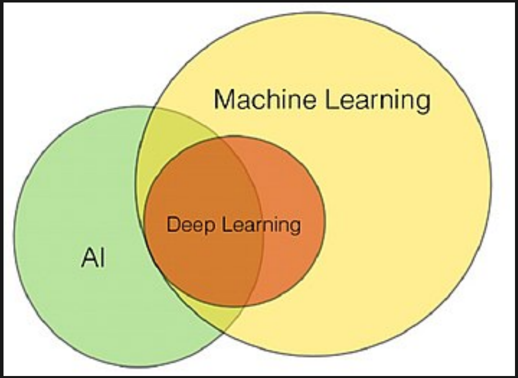
Rote learning and its variants

Perhaps the most elementary type of learning worth discussing would be a form of rote learning in which the program simply saved all of the board positions encountered during play, together with their computed scores. Reference could then be made to this memory record and a certain amount of computing time might be saved. This can hardly be called a very advanced form of learning; nevertheless, if the program then utilizes the saved time to compute further in depth it will improve with time.

Fortunately, the ability to store board information at a ply of 0 and to look up boards at a larger ply provides the possibility of looking much farther in advance than might otherwise be possible. To understand this, consider a very simple case where the look-ahead is always terminated at a fixed ply, say 3. Assume further that the program saves only the board positions encountered during the actual play with their associated backed-up

State of Affairs



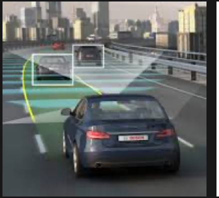
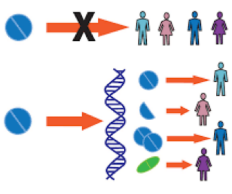


https://en.wikipedia.org/wiki/Machine_learning

Machine Learning is Everywhere

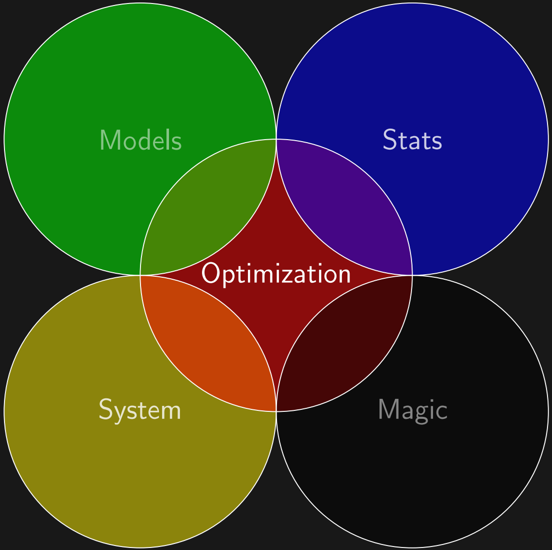
- Everyone uses ML everyday

- Lots of cool applications

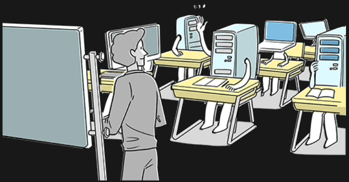
- Excellent for job-hunting

A Bit of Everything



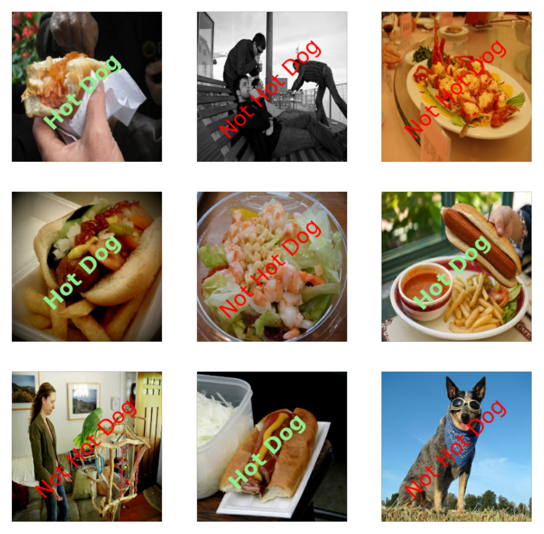
Learning Categories

- Supervised learning: teacher provides labels (answers)
  - classification: binary, multiclass, structured
  - regression: real-valued, multi-output, functional
  - ranking: pointwise, pairwise, listwise
- Unsupervised learning: go explore the world!
  - clustering – representation – visualization
- Reinforcement learning: teacher provides incentives
  - control – pricing – games
- Semi-supervised / self-supervised / active learning / etc.

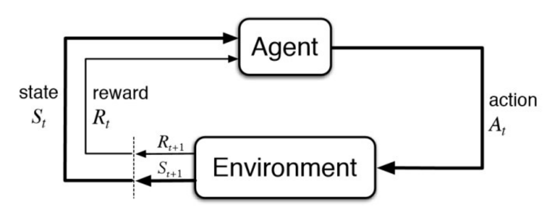


Supervised Learning ☞ hotdog app

example results



Reinforcement Learning



- Not in this course 😥, but see CS 486/686/885

D. Silver et al. "Mastering the game of Go with deep neural networks and tree search". *Nature*, vol. 529, no. 7587 (2016), pp. 484–489.

Unsupervised Learning

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

D. P. Kingma and P. Dhariwal. "Glow: Generative flow with invertible 1x1 convolutions". In: *Advances in Neural Information Processing Systems*. 2018, A. Radford et al. "Language models are unsupervised multitask learners". 2019.

Generative Adversarial Networks

$$\min_{\theta} \max_{\varphi} \hat{\mathbb{E}} \log S_{\varphi}(\mathbf{x}) + \hat{\mathbb{E}} \log(1 - S_{\varphi} \circ T_{\theta}(\mathbf{z}))$$

I. Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. 2014.


Parti


TheJasper
Whisperer.
com

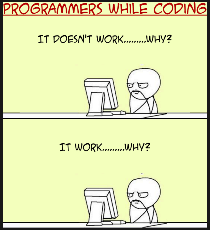

Parti

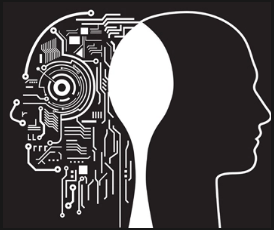
Focus of ML Research

- Representation: how to encode the raw data?
- Generalization: how well can we do on unseen data?
- Interpretation: how to explain the findings?
- Complexity: how much time and space?
- Efficiency: how many samples?
- Privacy: how to respect data privacy?
- Robustness: how to degrade gracefully under (malicious) error?
- Fairness: how to enforce algorithmic equity?
- Applications

What You Will Achieve

- Formulate ML problems and recognize pros and cons
- Understand and implement foundational ML models
- Develop and apply ML for new problems on real datasets
- Beware of potential ethical and safety issues of ML on society

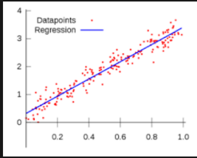


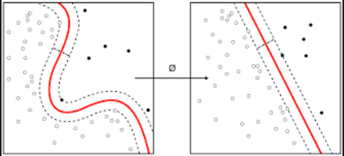


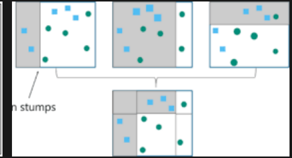
	Date	Topic	Slides	Notes	Supplementary	Assignments
00	May 06, 2024	Introduction	pdf		opt, stat	
01	May 08, 2024	Perceptron	pdf	pdf		
02	May 13, 2024	Linear Regression	pdf	pdf		pdf, tex (?)
03	May 15, 2024	Logistic Regression	pdf	pdf		
04	May 21, 2024	Hard-margin SVM	pdf	pdf		
05	May 22, 2024	Soft-margin SVM	pdf			
06	May 27, 2024	Reproducing Kernels	pdf			
07	May 29, 2024	Fully Connected NNs	pdf	pdf		pdf, tex (?)
08	Jun 03, 2024	Convolutional NNs	pdf			
09	Jun 05, 2024	Graph NNs	pdf	pdf		
10	Jun 10, 2024	Attention	pdf	pdf		
11	Jun 12, 2024	State-space	pdf	pdf		
12	Jun 17, 2024	Decision Trees	pdf			pdf, tex (?)
13	Jun 19, 2024	Boosting	pdf	pdf		
14	Jun 24, 2024	GANs	pdf	pdf		
15	Jun 26, 2024	Flows	pdf			
16	Jul 03, 2024	VAEs	pdf			
17	Jul 08, 2024	Optimal Transport	pdf			pdf, tex (?)
18	Jul 10, 2024	Hidden Markov Models	pdf			
19	Jul 15, 2024	Calibration	pdf			
20	Jul 17, 2024	Fairness	pdf			
21	Jul 22, 2024	Robustness	pdf			
22	Jul 24, 2024	Contrastive Learning	pdf			
23	Jul 29, 2024	Diffusion	pdf			

Classic

	Date	Topic	Slides	Notes	Supplementary	Assignments
00	May 06, 2024	Introduction	pdf		opt, stat	
01	May 08, 2024	Perceptron	pdf	pdf		
02	May 13, 2024	Linear Regression	pdf	pdf		pdf, tex (?)
03	May 15, 2024	Logistic Regression	pdf	pdf		
04	May 21, 2024	Hard-margin SVM	pdf	pdf		
05	May 22, 2024	Soft-margin SVM	pdf			
06	May 27, 2024	Reproducing Kernels	pdf			
12	Jun 17, 2024	Decision Trees	pdf			pdf, tex (?)
13	Jun 19, 2024	Boosting	pdf	pdf		

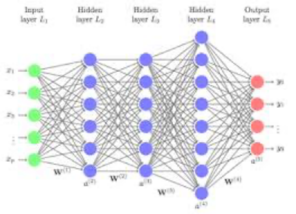


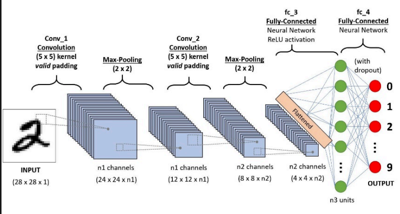


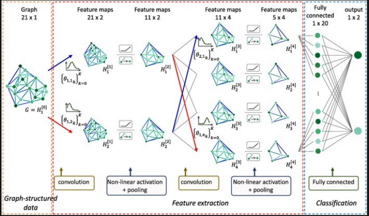


Neural Nets

07	May 29, 2024	Fully Connected NNs	pdf	pdf		pdf, tex (?)
08	Jun 03, 2024	Convolutional NNs	pdf			
09	Jun 05, 2024	Graph NNs	pdf	pdf		
10	Jun 10, 2024	Attention	pdf	pdf		
11	Jun 12, 2024	State-space	pdf	pdf		

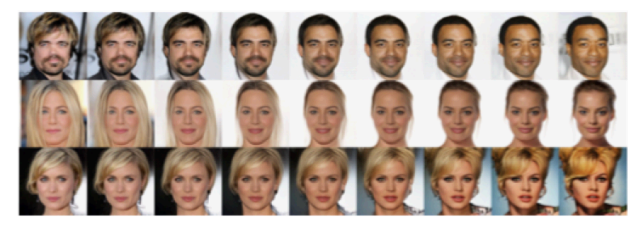






Generative Models

14	Jun 24, 2024	GANs	pdf	pdf		
15	Jun 26, 2024	Flows	pdf			
16	Jul 03, 2024	VAEs	pdf			
17	Jul 08, 2024	Optimal Transport	pdf			pdf, tex (?)
18	Jul 10, 2024	Hidden Markov Models	pdf			



Nascent

19	Jul 15, 2024	Calibration	pdf			
20	Jul 17, 2024	Fairness	pdf			
21	Jul 22, 2024	Robustness	pdf			
22	Jul 24, 2024	Contrastive Learning	pdf			
23	Jul 29, 2024	Diffusion	pdf