COMPUTATIONAL APPLICATIONS TO POLICY AND STRATEGY (**CAPS**)

Session 1 – Learning and Decision-Making

Leo Klenner, Henry Fung, Cory Combs

Outline

I. Admin
1. Definitions and Background of AI
2. Applications and Risks of AI
3. Humand Learning and Decision-Making
4. Types of Machine Learning
5. Learning and Optimality

Admin

- **Course goals:** connect domain expertise and AI, introduce conceptual tools useful for "AI generalists"
- **Course process:** focus on discussion and interactive exchange of perspectives
- **Course resources:** go to capsseminar.github.io/fall19 for lecture notes, slides, cases
- **Course policy:** attend all six sessions for transcript verification

1. Definition and Background of AI

What do we mean when we talk about AI?

1.1 Defining AI

	Human-based	**Ideal Rationality-based**
Reasoning-based	Systems that think like humans	Systems that think rationally
Behavior-based	Systems that act like humans	Systems that act rationally

1.2 Defining Rationality

- **Perfect rationality:**
  - the capacity to generate maximally successful behavior given the available information
- **Calculative rationality:**
  - the capacity to compute, *in principle*, the perfect rational decision given the initially available information
- **Bounded optimality:**
  - the capacity to generate maximally successful behavior given the available information and computational resources

1.3 Constraints in Rationality

Type	**Information**	**Computation**	**Time**	**Frequency**	**Desirability**
Perfect rationality	Yes	No	Yes	Rarely exists	High
Calculative rationality	Yes	Yes	No	Often exists	Low
Bounded optimality	Yes	Yes	Yes	Often exists	Depends on the bounds

1.4 Framing Bounded Optimality

- **Agent:**
  - the algorithm that makes decisions and is constrained by specifications like computational efficiency
- **Environment:**
  - where the agent carries out its decisions and is constrained by environmental features like imperfect information through fog-of-war

1.4 Rule-Based Systems

- **Rule-Based Systems:**
  - deterministic decision-making based on pre-configured rules
  - knowledge but no training process
  - evaluates branches of if-then statements
  - linear mapping of inputs to outputs

```
# Simple example of deterministic decision-making:
# whenever A is true, B is assigned the value 2

if condition A is True:
    then set variable B = 2
```

1.4 Machine Learning

- **Machine Learning:**

  - non-deterministic decision-making based on transformation of data
  - process of learning through training on data
  - evaluates complex optimization problems
  - non-linear mapping of inputs to outputs

```
# Simple example of non-deterministic decision-making:
# whenever A is true, B is assigned one of the possible even values between 0
# and n

if condition A is True:
    then set variable B = (choose random even number from range (0, n))
```

2. Applications and Risks of AI

What are applications and risks of AI?

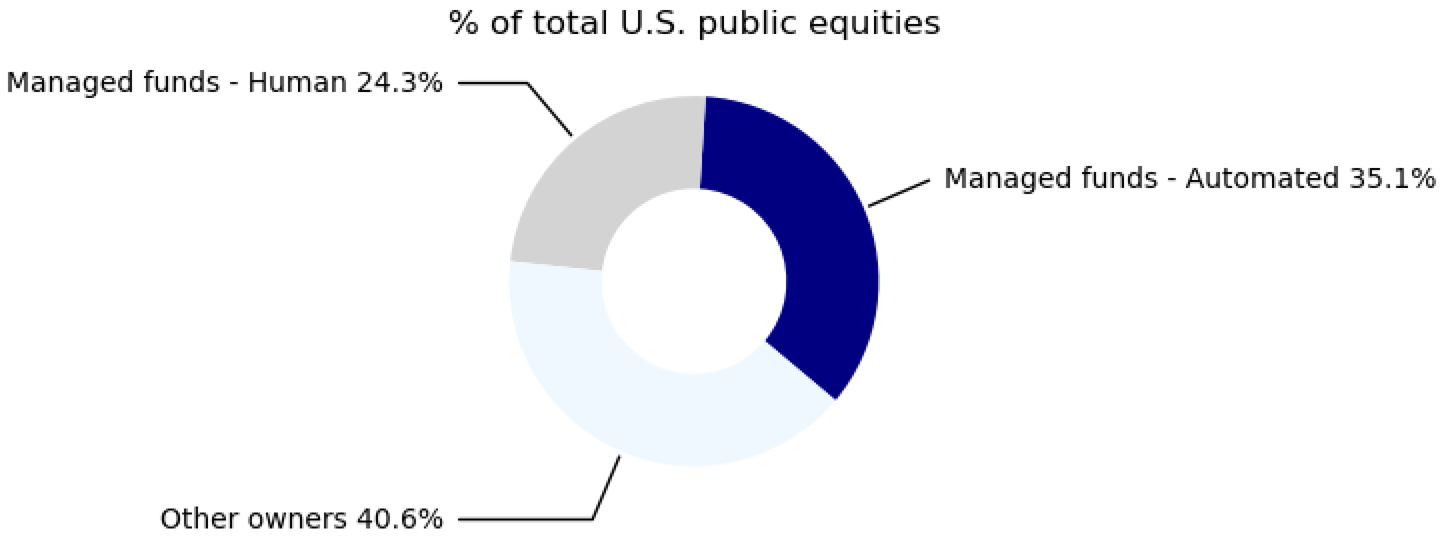
2.1 Timeline of AI in Financial Markets

- **1970s:** rules-based algorithms start to execute trades
- **1975:** creation of the first index fund, using rules to track the components of financial markets indecies
- **1990s:** creation of exchange-traded funds, which use rules to automate specific investment strategies, and of quantiative funds, which drive the use of advanced algorithms in financial markets
- **2010s:** increasing focus on machine learning within quantitative funds for analysis and strategy creation

2.2 Types of Agency in Financial Markets

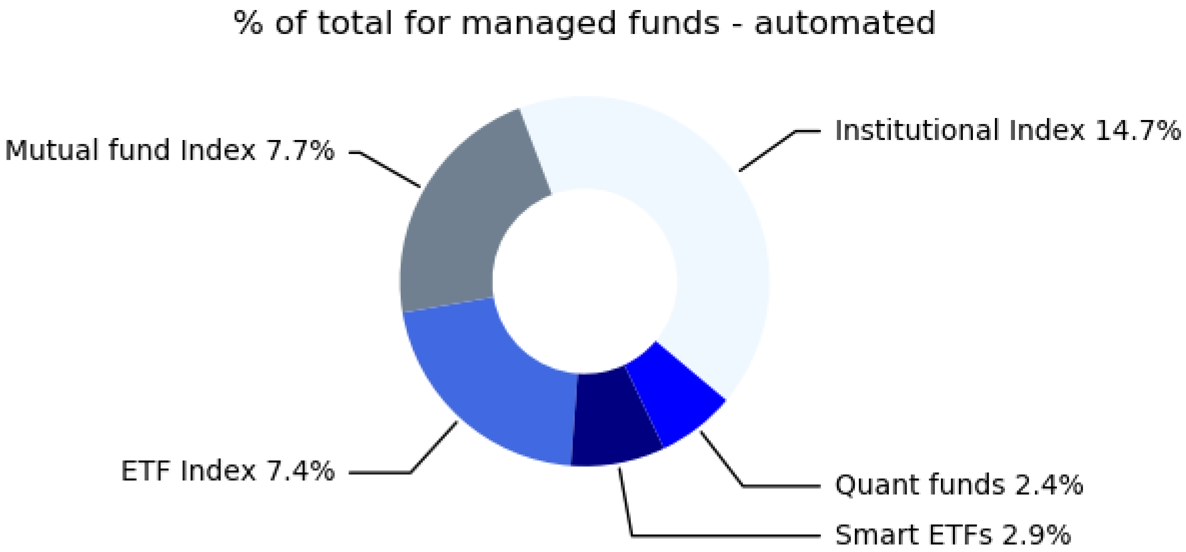
Type	Primary function	Primary product
Human	Create strategies	Mutual funds
Rules-based	Execute trades, or mimic human strategies and execute trades	Index funds
Human plus algorithm	Algorithms perform data analysis, humans select trades	Quant funds
Machine learning	Create and execute strategies	Quant funds

2.3 Assessing the Impact of Algorithms on Financial Markets



% of U.S. Public Equity Asset Holdings worth USD 31 trillion

2.4 Assessing the Impact of Algorithms on Financial Markets



% of U.S. Public Equity Assets in different automated funds

2.5 Overview of Risks of AI

- **Machine Learning-Specific Risks**
  - overconfidence and mistrust
  - lack of data oversight
  - lack of continuous adaptation
  - lack of transparency and explainability

- **Ecosystem-Specific Risks**
  - complexity and speed

2.5.1 Overconfidence and Mistrust

- **Overconfidence:**
  - results discovered by a machine learner might spurious (based on artificial correlation), which can take time to uncover or might not happen at all
- **Mistrust:**
  - even if the results are proven to be non-spurious humans might have pervasive mistrust in algorithmic decision-making, which is why explainability and transparency are key

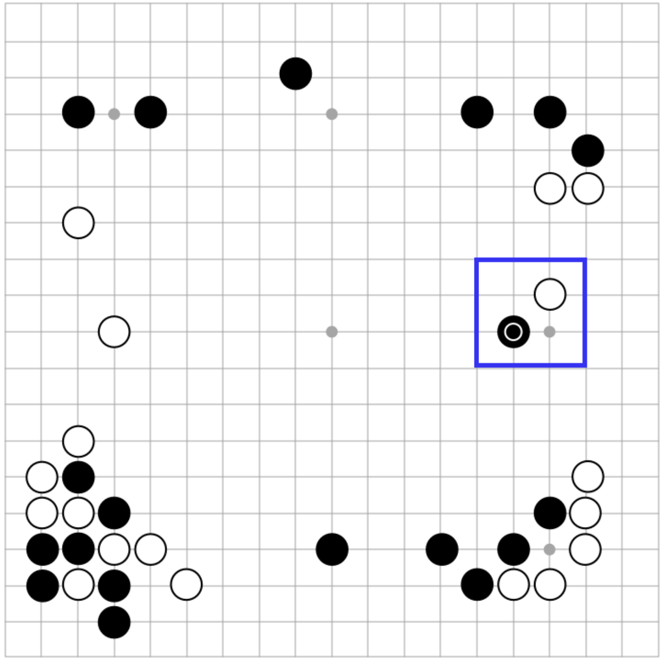
2.5.2 Lack of Data-Oversight

- **Dataset is big but not big enough:**
  - machine learners require extremely large training datasets that might exceed available levels, leading to suboptimal performance
- **Dataset is too big for an algorithm:**
  - if the training dataset is too large for an algorithm, it might overfit and learn simple behavior that doesn't generalize to test data
- **Garbage in, garbage out:**
  - if the dataset includes bias or is otherwise impaired, this will directly affect the algorithm's performance
- **Adversarial examples:**
  - datapoints intened to distort the algorithm's performance might be included in the data for malicious purpose

2.5.3 Lack of Continuous Adaptation

- **Static train-deploy patterns:**
  - <u>once trained, an algorithms stops learning</u>, which can prove challenging for deployments in environments that constantly change over time

2.5.4 Lack of Transparency and Explainability



> That's a very surprising move. I thought that was a mistake. I thought it was click miss. Exactly, if we were in online Go, we would call it a 'clicko'.

> Yeah, it's a very strange move. Something like this [changes black's position on the board] would be a more normal move and then this [moves white's position on the board] is how white would respond.

> Lee has left the room. He left the room after this move. Just to recover from this move. It's a very surprising move. I don't know whether it's a good or a bad move at this point.

AlphaGo vs Lee Sedol, Match 2, Move 37

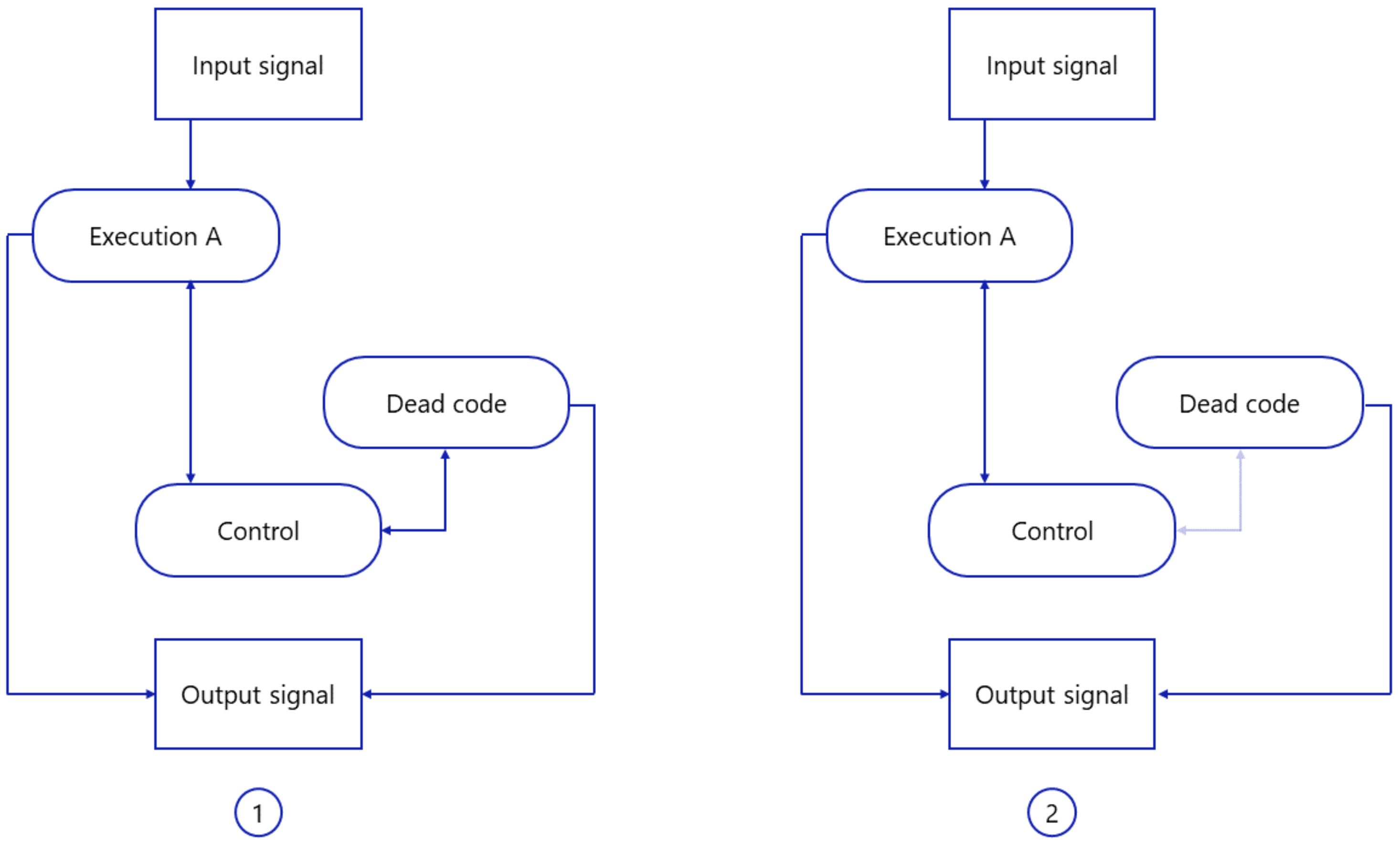
2.5.5 Complexity and Speed

- **Complexity:**
  - industrial software development processes are complex, iterative and involve multiple stakeholders often with heterogenous information and knowledge
- **Speed:**
  - Software gets built to, among other things, speed up decision-making processes, leaving humans often with limited response time to correct errors in real-time

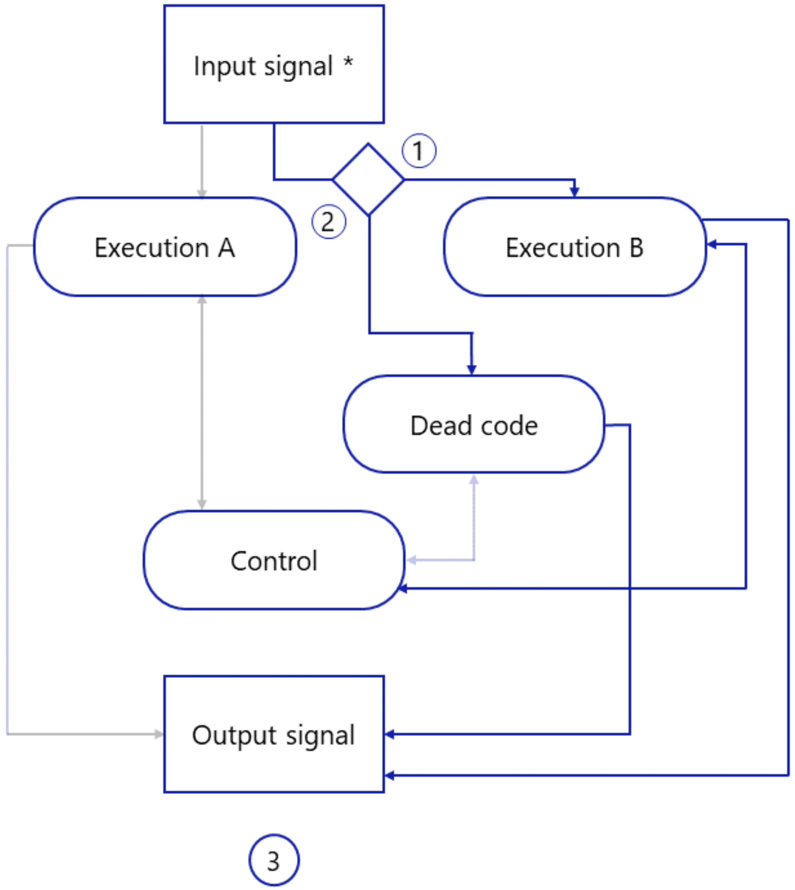
2.5.6 The Fall of Knight Capital Group

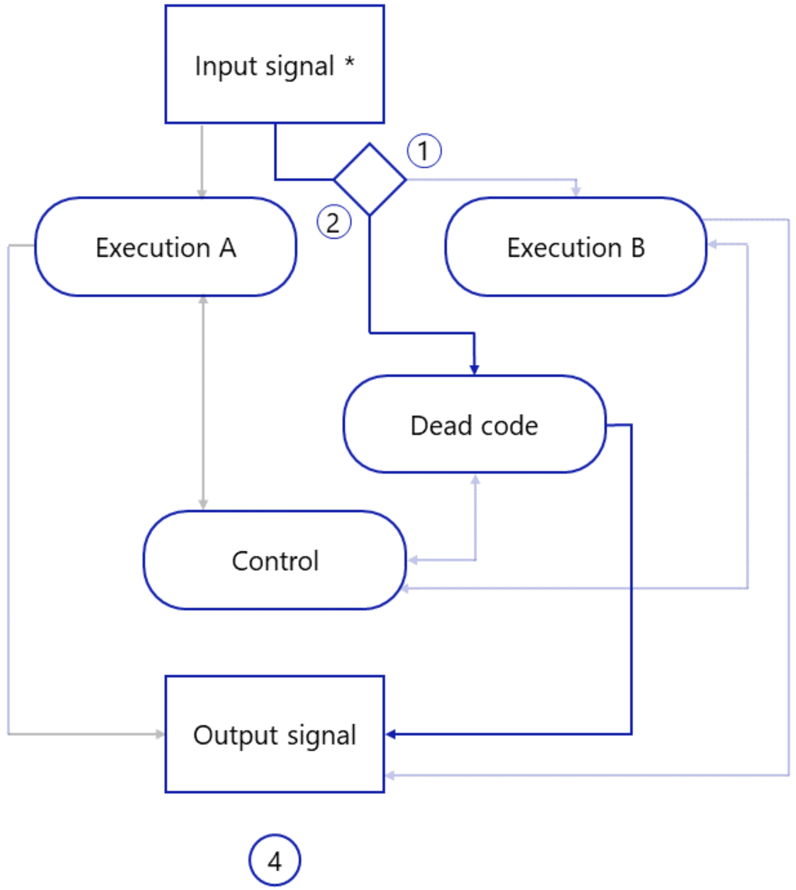
Initial drop in the stock price of Knight Capital Group following a trading disruption on August 1, 2012

2.5.7 Changes 1 and 2 to Knight's Trading Software



2.5.7 Changes 3 and 4 to Knight's Trading Software





3. Human Learning and Decision-Making

Case: Learning in a Counterinsurgency Team

3.1 Lines of Inquiry

- Types of different learning
- Rule-following and application to new environment
- Training and exception handling
- Strategies for improvementand adaptation during deployment
- Differences between Amerine and Nutsch

3.2 Excerpts from Amerine and Nutsch

> You get out of these courses and sometimes you have instructors that take what they teach very seriously and other times you don't.

> I found was that every major lesson I have learned throughout my career, whether it was in the Q Course [Army Special Forces Qualification Course] or Ranger School, I mean, everything that I was taught in the school house, I applied over there.

> All the major muscle movements during the campaign we really had been taught.

> But the sergeants and I, coming back as we're talking about this, we did the things you do in training. Each day we would do lessons learned, an internal AAR [After Action Report], whether it was five minutes or fifteen minutes, sit down and go 'Damn, what nearly killed us today? How do we make sure that doesn't happen again? You know, how do we survive the next hour? And how do we win?'

> I would have to say, even in our mission, we were the students.

> They couldn't read a map but they could describe to you passionately 'It's this village, don't you understand? It's this village right over here. It's this guy, he's the one we're after.'

3.3 Core Themes and Differences

Core themes:

- Application of rules vs adaptation
- Learning from a supervising instructor vs learning from interaction with the environment
- Learning through guidance vs learning through trial-and-error
- Transfer of knowledge across domains vs learning how to learn

Differences between Amerine and Nutsch:

- Amerine emphasizes the value of having knowledge that generalizes across environments
- Nutsch emphasizes the value of learning how to learn in training, so that even without knowledge about a new environment, fast adaptation is possible
- Amerine emphasizes *instructor-led* learning
- Nutsch emphasizes *process-driven* learning

4. Types of Machine Learning

- **Supervised learning**
- **Unsupervised learning**
- **Reinforcement learning**
- **Meta- and transfer learning**

4.1 Supervised learning

Supervised learning is learning from labeled data, where the labels are provided to the algorithm by a human and the algorithm learns to apply these labels to new data, e.g. for prediction

4.2 Unsupervised learning

Unsupervised learning is learning from data that does not have labels, for the purpose of identifying features that can be used to group or cluster the data, e.g. for pattern detection

4.3 Unsupervised learning

Reinforcement learning is about learning through trial and error from interaction with an environment what the best sequence of actions is to achieve a specific goal, e.g. for autonomous control

4.4 Meta- and transfer learning

Meta- and transfer learning are at the forefront of current AI research and concern how algorithms can <u>learn *how* to learn</u> and <u>quickly adapt previously learned behavior to new environments</u>

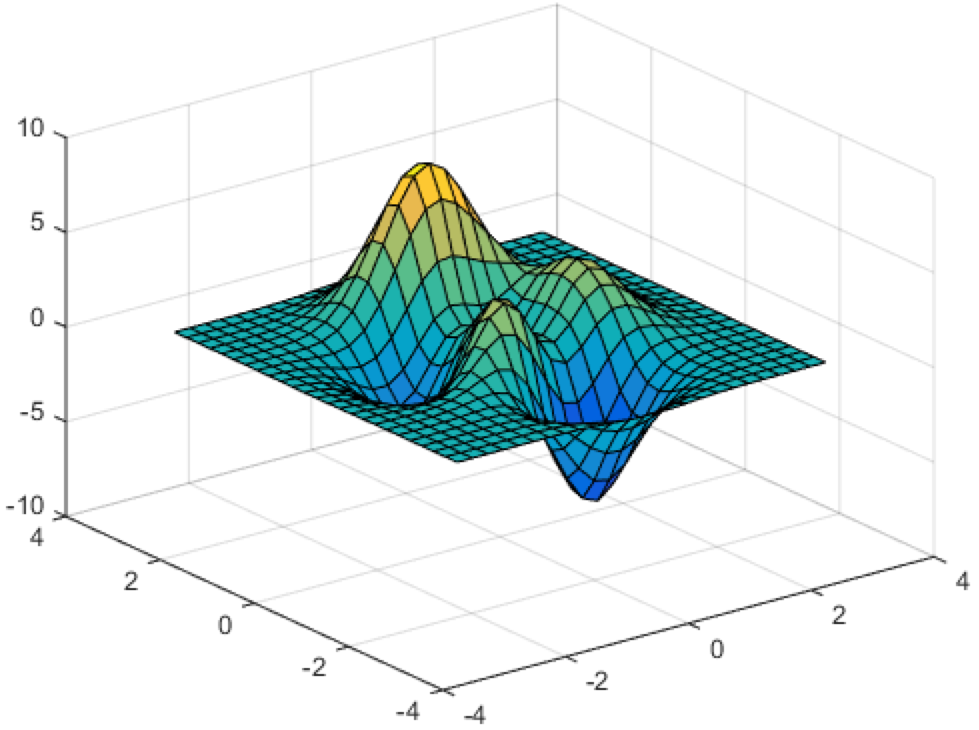
5. Learning and Optimality

Can we define learning as optimization?

5.1 Learning as Minimization and Maximization

- Minimize the rate of errors
- Maximize the rate of success

5.2 Optimization as Searching a Decision-Space



5.3 Two Types of Constraints

- Constraints *whether* the algorithm can solve the problem

- Constraints *how* the algorithm can solve the problem

- Making the “right” decision requires a <u>consistent definition of what constitutes an error or success</u> for a selected task. Both types of <u>contraints can influence these definitions</u>, in different manners.

5.4 Levels of Target Specification

- Constraints *whether* the algorithm can solve the problem

- Constraints *how* the algorithm can solve the problem

- Making the “right” decision requires a <u>consistent definition of what constitutes an error or success</u> for a selected task. Both types of <u>contraints can influence these definitions</u>, in different manners.

5.4 Levels of Target Specification

- **Ideal specification** describes the “wishes” or intentions of the system designers, which correspond to the hypothetical description of an ideal system

- **Design specification** describes the “blueprint” of the system, corresponding to the specifications that the designers actually use to build the system

- **Revealed specification** describes the actual “behaviour” exhibited by the system, i.e. what actually happens

5.5 Applied Target Specification

- CoastRunners case

- Real result: https://miro.medium.com/max/478/0*UoBrrtrY2rx2SvXr

5.6 What is a "Good" Decision?

- Not all optimized decisions are "good" but all "good" decisions are optimized
- "Good" is a joint product of policy, ethical, and technical considerations