REINFORCEMENT LEARNING

Introduction
Basic principles

Diego Klabjan
Professor, Industrial Engineering and Management Sciences

Northwestern | McCORMICK SCHOOL OF ENGINEERING

Credit

- Several slides adaption of the lecture material from
  - Sergey Levine from Berkeley
    - http://rail.eecs.berkeley.edu/deeprlcourse-fa17/index.html
  - David Silver from University College London
    - http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
- A few examples taken from Lex Fridman, MIT
  - https://selfdrivingcars.mit.edu
- Tremendous acknowledgment to Hindeke Tewodros
  - Prepared all of the slides
  - Former undergraduate student at Northwestern

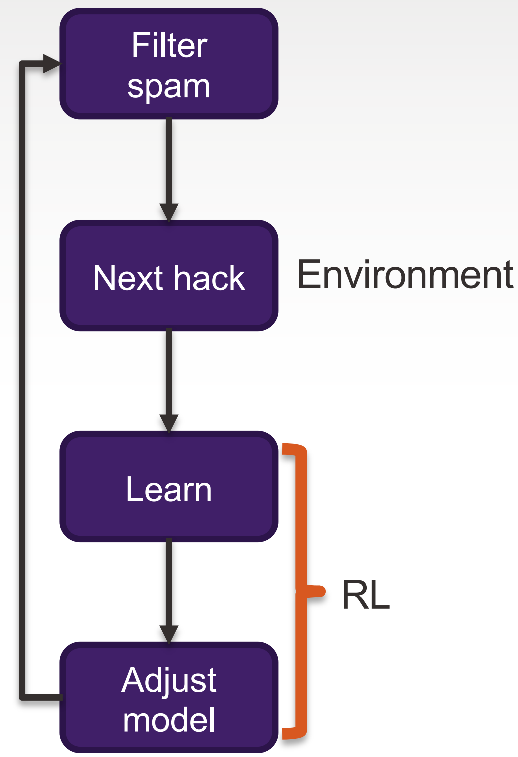
BASIC CONCEPTS

Supervised, Unsupervised

- Data consisting of feature vector and optional labels
- Unsupervised
  - No labels
  - Customer segmentation
    - Any clustering
- Supervised
  - Predict labels to unseen data
  - Pricing for logistic services
  - Predict demand for bikes
    - Bike-sharing program
- No feedback loop
  - Email spam filter counterattacked my spammers

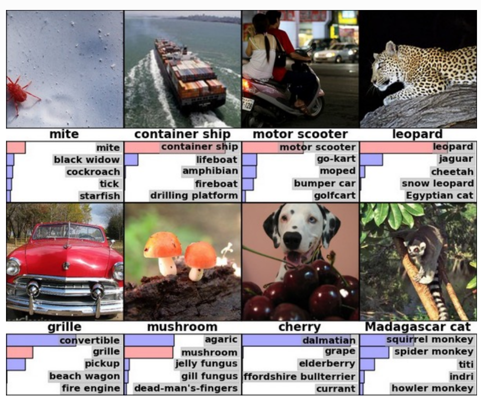
Reinforcement Learning

- Algorithm that learns on its own
- Spam filter
  - New hack
  - Algorithm learns it
  - Automatically adjust strategy
- Supervision/labels replaced by reward
  - Make adjustment and model reward
  - Adjustment
    - Purchase extra 10 units of stock due to cold weather
    - Reward is revenue for these units – procurement cost

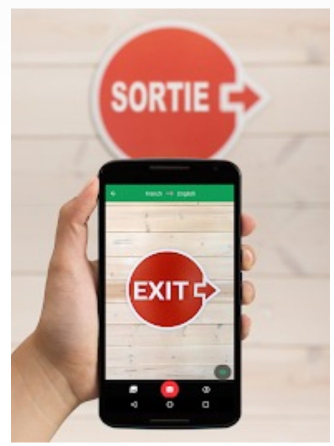


One-off Decision Making

- When system making a single isolated decision
  - Classification, regression
- When that decision does not affect future decisions



http://rocknrollnerd.github.io/ml/2015/05/27/leopard-sofa.html



https://play.google.com/store/apps/details?id=com.google.android.apps.translate&hl=en

Sequential Decision Making

- Limited supervision
  - You know **what** you want, but not **how** to get it
- Actions have consequences
  - Future reward unknown

Caveat: Which ones have been operationalized?

Notorious applications

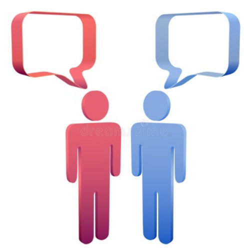
robotics

http://blogs.wiscanada.com/content/will-social-services-be-changed-by-artificial-intelligence-and-robotics/

autonomous driving

https://www.aspeninstitute.org/tag/autonomous-vehicles/

language & dialogue (structured prediction)



https://www.dreamstime.com/stock-photo-people-talk-3d-social-media-speech-bubbles-image16446490

operations management

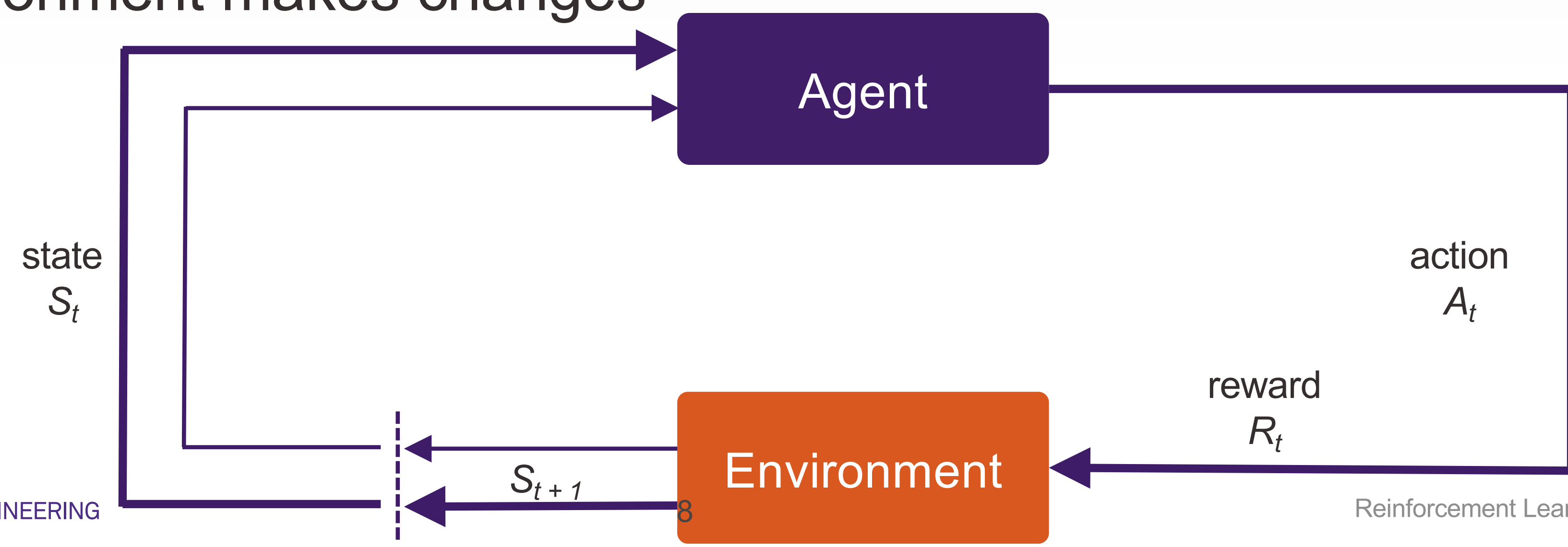
https://www.mainline.com/services/assess/

finance

https://www.123rf.com/photo_13129528_business-background-with-finance-graphs-pen-and-calculator.html

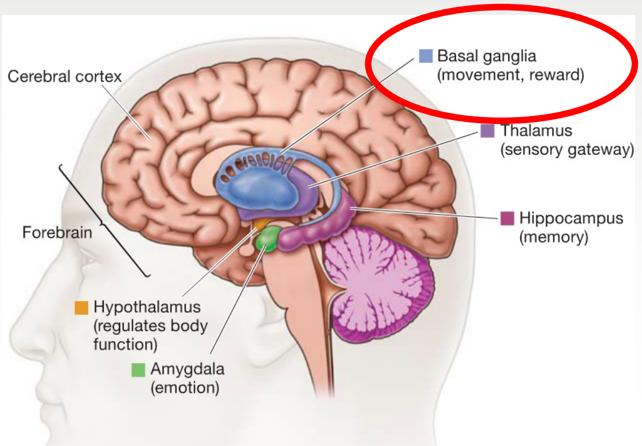
Setting and Model

- State
- Action
- Get reward
- Environment makes changes



Reward

http://funkypotato.com/pac-man/



https://www.pinterest.com/pin/563654328148648894/

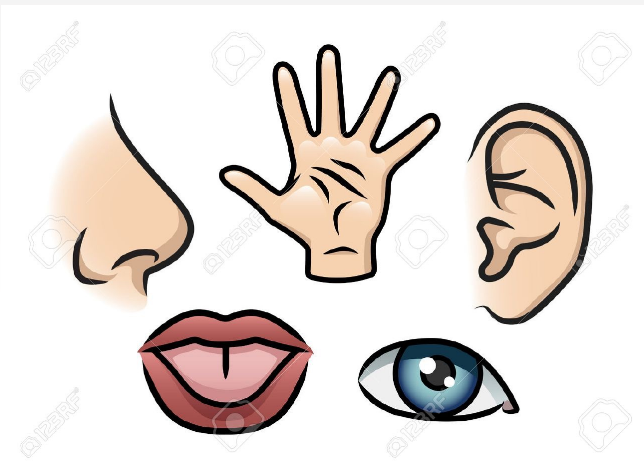
https://steemit.com/science/@coinbizpro/who-are-you-a-predator-or-a-victim

Imitation Learning

https://www.motorauthority.com/news/1031363_volvo-launches-research-project-to-map-driver-behavior

From Input to Output

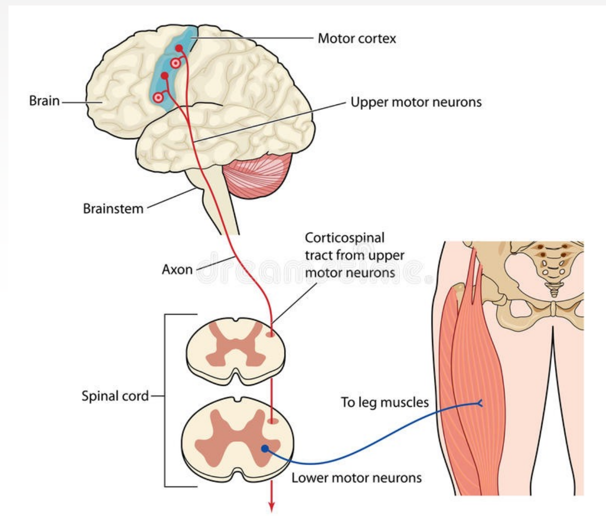
Sensory inputs



https://www.123rf.com/photo_26573894_stock-vector-a-cartoon-illustration-depicting-the-5-senses-smell-touch-hearing-taste-and-sight.html

DNN to encode state

Output: complex actions



https://www.dreamstime.com/stock-illustration-motor-nerves-leg-to-motor-cortex-originating-muscles-traveling-via-spinal-cord-brain-created-adobe-image61090743

DNN model actions

Limitations Today

- Humans can learn incredibly quickly
  - Deep RL methods are usually slow
- Humans can reuse past knowledge
  - Transfer learning in deep RL is an open problem
  - AlphaGo cannot play Atari
    - Even less optimize inventory
- Reward challenging
  - Read Super Intelligence by Nick Bolstrom

Inventory Management

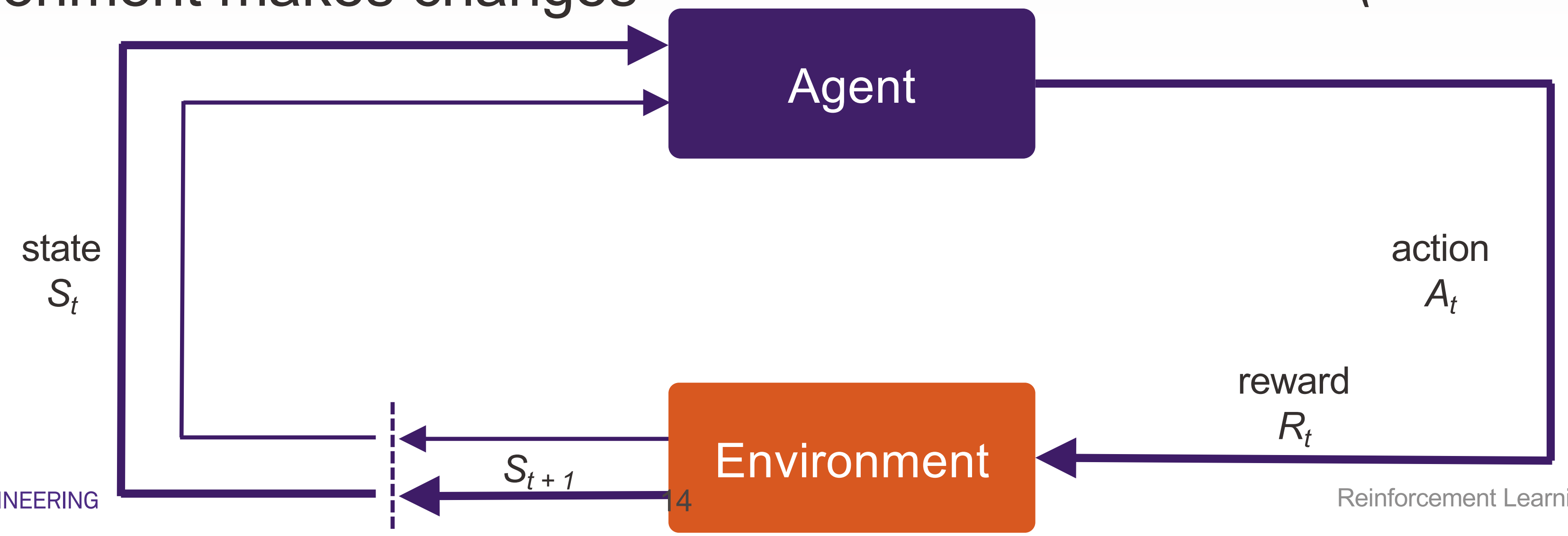
- Retail store with SKU's
  - Brick-and-mortal or e-commerce
- Have 10 pairs of Nike Air Jordan
- Should we order additional boxes?
  - Forecast to sell 5 tomorrow
  - Options
    - Do not order
    - Order 5 units
    - Order 50 units

www.niketalk.com

Setting and Model

- State
- Action
- Get reward
- Environment makes changes

- *Inventory level*
- *How much to order*
- *Reward of revenue - cost*
- *Random demand (adversary)*



Inventory Management

- State = inventory units at 8 am
- Action = how many to order of each SKU
- Reward = selling price of units – cost of procurement
- Environment = stochastic demand

$$S_{t+1} = S_t + a_t - D_t$$

$$R(S_t, a_t) = p_t E[D_t] - c_t a_t - U_t 1_{a_t > 0}$$

- What if demand exceeds inventory plus order?
- What if orders have lead time?

APPLICATIONS

Portfolio Management

- Cash and list of securities to invest
  - Can change position every morning
- State
  - Number of positions in each security and cash
- Action
  - How much to buy/sell of each security
- Reward
  - Profit and loss
  - Reality much more complex

$$reward = utility(wealth)$$

- Environment
  - Interest rates on cash
  - Liquidity of securities

Tic-tac-toe

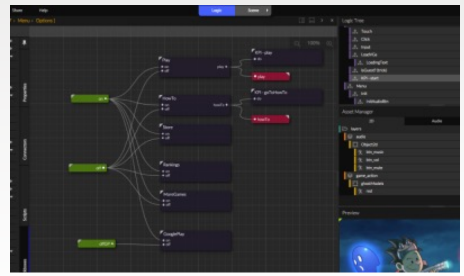
- State = positions on board
- Action = next move
- Reward
  - Zero if not end
  - 1 if win
  - -1 if we lose
- Environment
  - Move of opponent

	x	
x	o	
	o	

Playing Video Games

https://www.wired.com/2010/03/red-steel-2-review/

https://elearningindustry.com/directory/elearning-software/wimi5



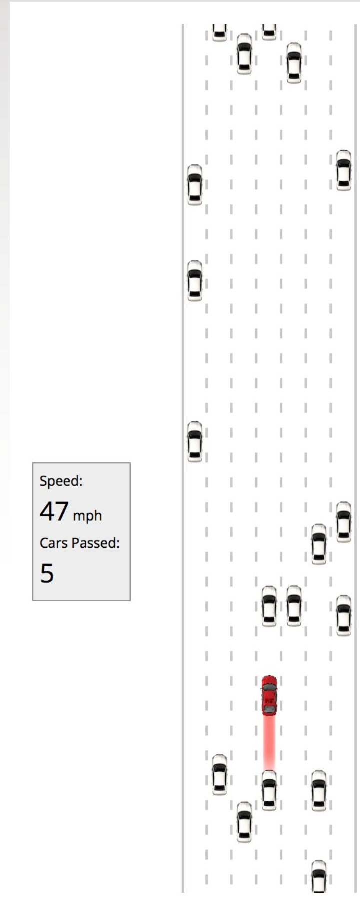
video game AI pipeline

game API → extract relevant features → state machine for behavior → planner → low-level bot control → controls

https://medium.com/@Dougly/closures-with-swift-58b274d849ad

https://www.theverge.com/2014/12/16/7382735/the-best-video-games-of-2014 https://selfdrivingcars.mit.edu/deeptraffic/

- Discrete time (1 second)
- State
  - Location of all cars
    - Deep learning encoding
    - Object recognition
  - Your own location
- Action
  - Position of your car next second
  - In cars translate to control
- Reward
  - Cost of crash, maintain speed, etc
- Environment
  - Obstacles on road, pedestrians, etc



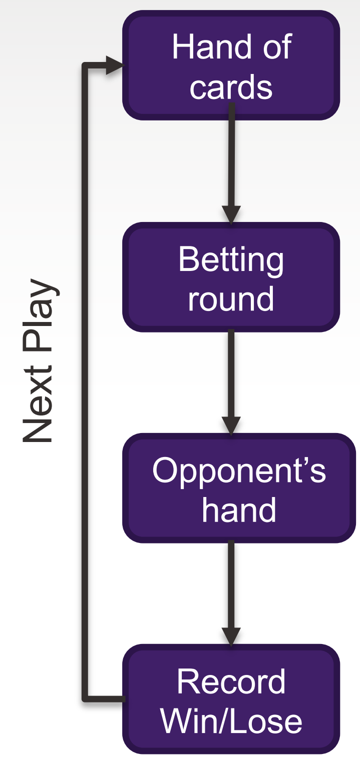
Autonomous Cars

Airline Revenue Management

- Two classes: Basic economy and refundable ticket
  - Limited number of seats on aircraft
- Passenger places a booking request
- Action
  - Accept the request or reject
  - Why not always accept?
- State
  - Available seats in aircraft
- Reward
  - Price of itinerary
- Environment
  - Do not know future booking requests
- Dilemma
  - Economy request
  - If accept, a refundable booking request might be coming
  - If reject, seat might be left unfilled

Use Cases

- Production planning
  - Current state of a plant
  - Schedule production for next hour
  - Reward
  - Machines go down, electricity rates spike
- Chess (games in general)
  - State of the board
  - Next move
  - Leads to a win
  - Opponent makes the move
- Shortest path
- AlphaGo



Contextual Bandit

- User with profile access a web page
- Action
  - Ads to display
- Reward
  - Click on ad
  - Caveat: revealed only after action made
- State
  - Features of user and candidate ads
- Next state?

- Not reinforcement learning
  - Next state independent of action and current state
  - No notion of future reward based on current action

Single Agent vs Multi Agent

- Single agent
  - Single decision maker
  - In games other players captured in environment
  - Reasonable when many participants
    - Individual behaviors neutralize
- Multi agent
  - Explicitly model competing agents
  - Use when powerful dominant competing agents
    - Large trading firm
  - Texas Hold'em
    - Imperfect information game

- Multi agent modeling
  - State captures all agents
  - Action for each individual agent
  - Reward for each agent
  - Next state depends on state and actions of each agent
- Agents can compete or collaborate

Summary

- Reinforcement learning: state, action, reward, environment
- Imitation learning
  - Reward not explicitly given, episodes are given
- Contextual bandit
  - State, action, reward revealed after action
  - No notion of next state
- Single agent vs multi agent