Capturing and Reusing Experience

2019-11-18

Andrew Ng – The State of AI (December 15, 2017)

- "99% of the economic value created by AI today is through one type of AI: which is learning a mapping A → B, or input to output maps"
  - Falls under category of supervised learning
- Other types (ordered falloff)
  - Transfer learning
  - Unsupervised learning
  - Reinforcement learning

Input	Output
Picture	Is it you? (0/1)
Loan application	Will the applicant repay the loan? (0/1)
Online: (Ad, User)	Will you click? (0/1)
Voice input	Text transcript
English	French
Car: image, radar/lidar	Positions of other cars

https://www.youtube.com/watch?v=NKpuX_yzdYs

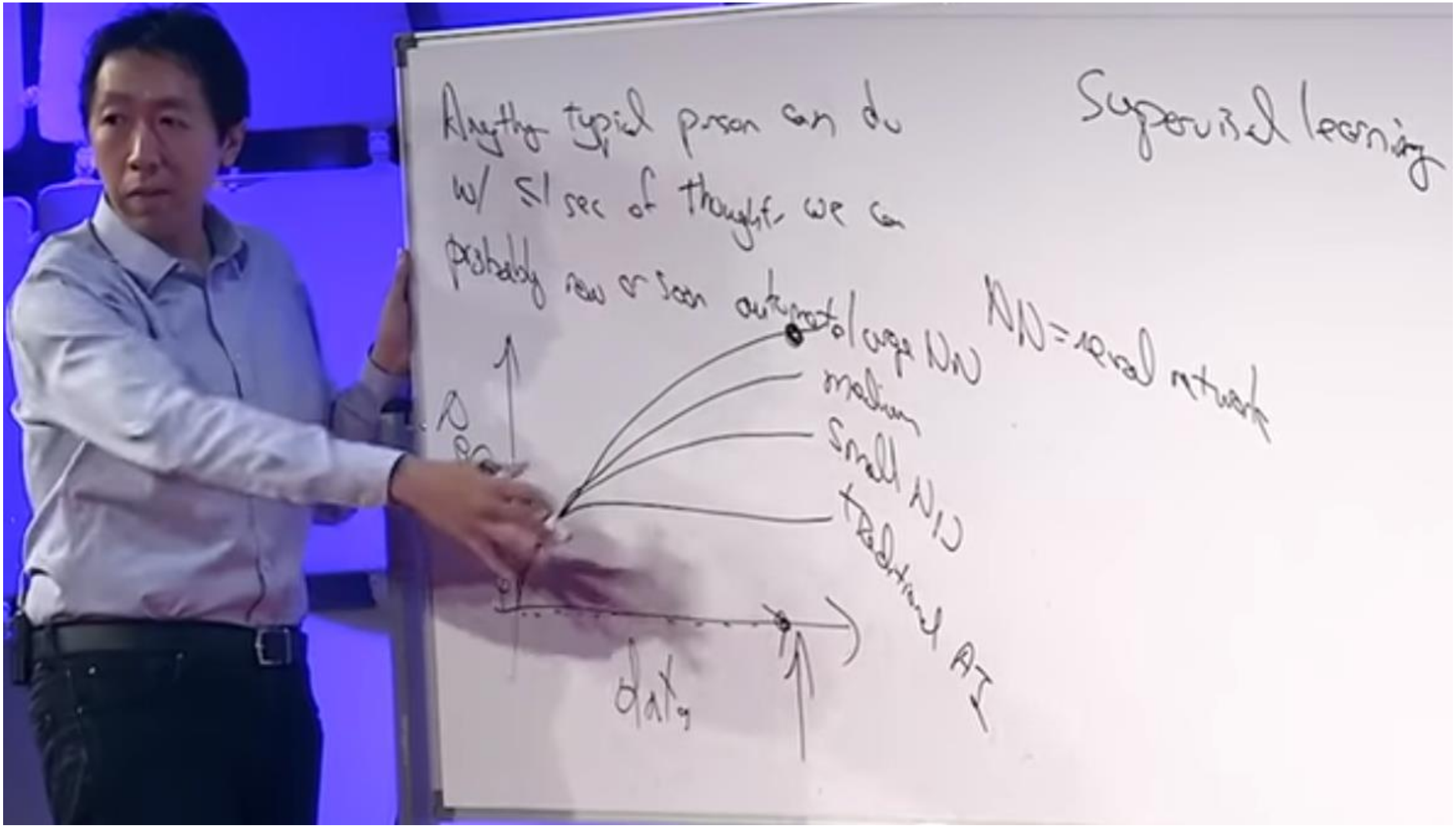
ML & Function Approximation

- Arthur Samuel (1959), checkers:
  - Machine Learning: Field of study that gives computer the ability to learn without being explicitly programmed (recognize board patterns that lead to wins/losses)
- Dr. Mark Riedl, what is a deep neural network:
  - Function approximation via stacks of complete acyclic weighted bipartite graphs
- Target function *f* may be known or unknown
  - **If known**, we seek functions that trade accuracy for desirable properties (inexpensive computation, continuity, integral and limit values, etc.)
  - **If unknown**, only a set of points of the form (x, *f*(x)) is provided.

- Several techniques for approximating *f*
  - If *f* is an operation on the real numbers, techniques of **interpolation, extrapolation, regression analysis, and curve fitting** can be used
  - If the codomain (range or target set) of g is a finite set, one is dealing with a **classification** problem instead.
- To some extent, the different problems (regression, classification, fitness approximation) have received a **unified treatment in statistical learning theory, where they are viewed as supervised learning problems**.
- The multilayer perceptron (a class of feedforward ANN) is **a universal function approximator**, as proven by the universal approximation theorem

https://www.youtube.com/watch?v=UzxYlbK2c7E

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


Anything typical person can do
w/ ≤1 sec of thought, we can
probably now or soon automate
Supervised learning
large NN
medium
small NN
traditional AI
NN = neural network
Perf
data

Decision Learning in M&F

- See Millington & Funge
  - 7.4 through 7.8
- Naive Bayes classification (7.5)
  - Try first, simple to implement
  - Good baseline
- Decision tree learning (7.6)
  - Output is interpretable
- Reinforcement learning (7.7)
  - “hot topic in game AI”
  - “a good starting point is the Q-learning algorithm. Q-learning **is simple to implement**, has been widely tested on non-game applications, and can be **tuned without a deep understanding of its theoretical properties**” (p631)
- Neural Networks (7.8)
  - “Very little neural network theory is applicable to games, however” (p646)

Decision Making Questions

1. How can we describe decision making?
2. What do the algorithms we've seen share?
3. What are the dimensions we tend to assess?

PCG Questions

1. PCG can be used to p____ or a____ game aspects
2. F(player model, designer constraints, instance) → fitness

RL Questions

Q1: What is the state transition function? Do we need it as input for Q-learning?

Q2: For the multi-armed bandit problem, which statement below best summarizes why we can't just use exploitation or exploration?

(A.) Exploration means it will take a long time to find the optimal solution

(B.) Both will minimize our total regret

(C.) Pure exploration doesn't learn, pure exploitation learns too fast

(D.) Pure exploration means we can never do better than chance, pure exploitation can get an agent caught in a local maximum strategy

No free lunch

"One of the greatest challenges in applying reinforcement learning to real-world problems is deciding **how to represent and store value functions and/or policies**. Unless the state set is finite and small enough to allow exhaustive representation by a lookup table […] **one must use a parameterized function approximation scheme**. […]

Most successful applications of reinforcement learning **owe much to sets of features carefully handcrafted based on human knowledge** and intuition about the specific problem to be tackled. […]

in all the examples of which we are aware, the most impressive demonstrations required the network's input to be **represented in terms of specialized features handcrafted for the given problem**"

Sutton & Barto, 2nd ed, section 16.5

Millington 7.3

CAPTURING AND REUSING EXP: ACTION PREDICTION

Action Prediction

- Guess what player will do next
  - E.g. waypoint, weapon, tactic, cover point, melee
  - Make more realistic, challenging (helpful) NPC
  - Can do with little observation
  - Can transfer from other players
- Humans bad at random (Psychology). Furthermore…
  - “We have shared characteristics that run so deep that learning to anticipate one player’s actions can often lead to better play against a completely different player.”

Naïve Algorithm

- Predict using raw probability
  - Keep a tally, use to predict
  - Pro
    - Easy, fast
    - Gives a lot of feedback to player
    - Can learn from many different players
  - Con
    - Player can “game” the system
    - Eventually can reduce to equal probabilities
- “Left or right” game
  - Object in either L or R hand
  - Another persons guesses hand
- Incremental update of average
  - Keep mean (m_{n-1}), and count (n)
  - $m_n = m_{n-1} + (1/n)(v_n – m_{n-1})$

String Matching

- Choice made several times
  - Encode as string “LRRLRLLLRRLRLRR”
  - Predict → find substring, return subsequent choice
  - Example: “RR”. What next?
  - Window size: 2
- Rarely implemented by matching against a string
  - Use a set of probabilities similar to the naïve algorithm

Prediction: *N*-Grams

- String matching + probabilities
  - N is window size + 1 (e.g. 3-gram from before)
  - Record Prob of each move for all windows
  - Must sum to 1
  - E.g. “LRRLRLLLRRLRLRR”

	..R	..L
LL	1/2	1/2
LR	3/5	2/5
RL	3/4	1/4
RR	0/2	2/2

Prediction: *N*-Grams

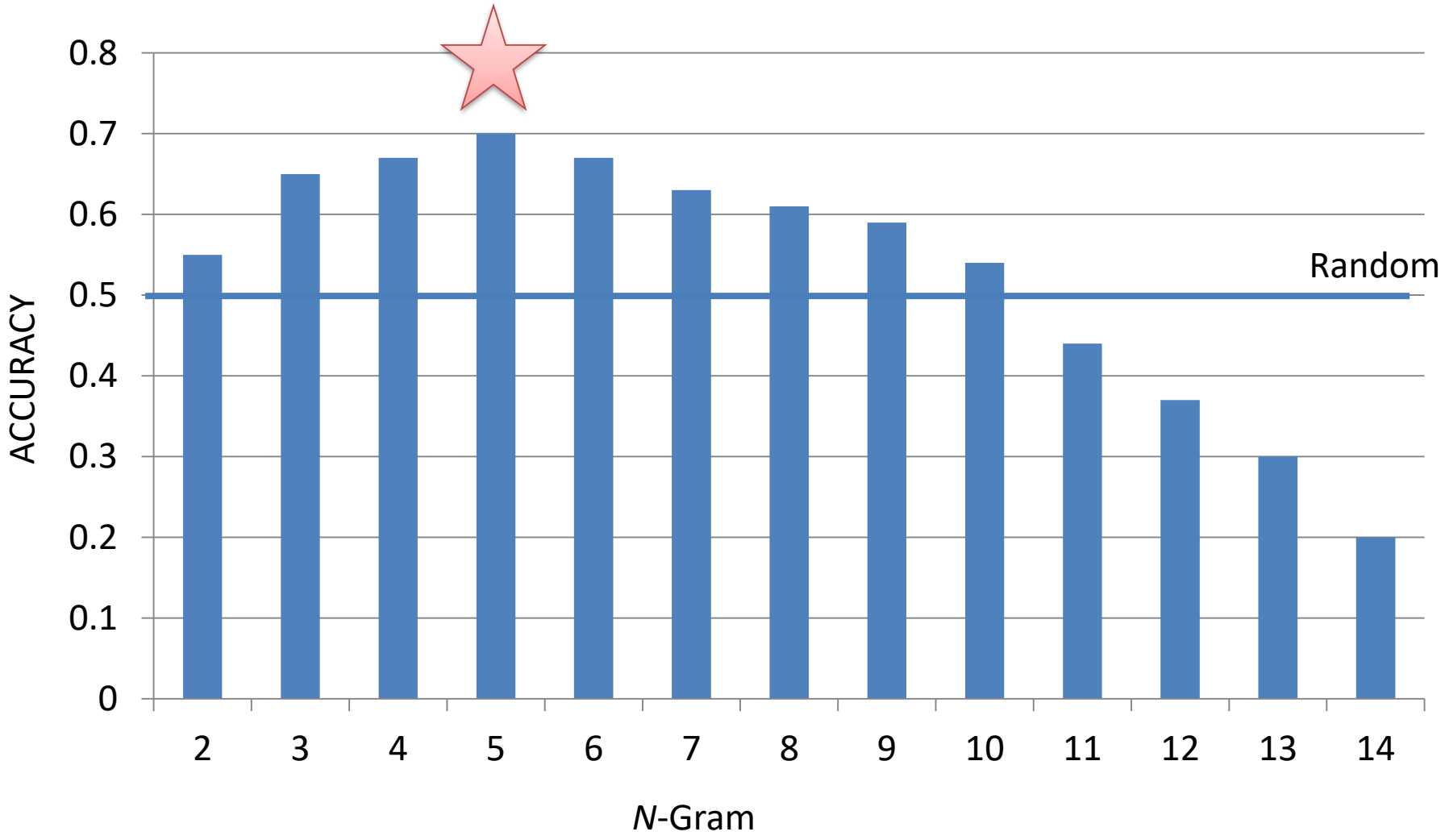
- String matching + frequencies
  - N is window size + 1 (e.g. 3-gram from before)
  - Record count of each move for all windows
  - Must sum to count
  - E.g. “LRRLRLLLRRLRLRR”

	..R	..L
LL (2)	1	1
LR (5)	3	2
RL (4)	3	1
RR (2)	0	2

Question

How do we choose the window size?

Window Size


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
ACCURACY
Random
2
3
4
5
6
7
8
9
10
11
12
13
14
N-Gram

Window Size

- Increase size helps initially, hurts later. Why?
  - Future actions predicted by *short* causal process
  - Similar to Markov assumption?
  - Psychology?
  - Degree of randomness in actions
    - (↑ random ↓window)
- How to tune?

Hierarchical *N*-Grams

- Online learning approach
- Balances max predictive power and alg. perf.
  - Large window, better potential, slower coverage
- Essentially several parallel *N*-grams
  - E.g. Hierarchical 3-gram: 1, 2, and 3 gram
  - When prediction requested, look up window with
    - sufficient examples
    - highest predictive accuracy
  - What is sufficient number of examples?

N-gram summary

- Simple, effective prediction mechanism
- Synonymous with combo-based melee games
  - Can make unbeatable (no fun) AI
  - Often is intentionally weakened
- Many other uses
  - statistical analysis techniques (e.g. language)
  - [Weapon, location, unit] selection…