

Practical AI: AI as a function

Stanislav Protasov for Harbour.Space



Refresh

- What is AI?
- What is ML?
- How do ANNs work?
- How do companies distribute AI?

Agenda

- What is function
- Brute force method
- Algorithmic approaches
 - Naive
 - Heuristic
 - Game trees, minimax
- Vectors and features

AI as a function

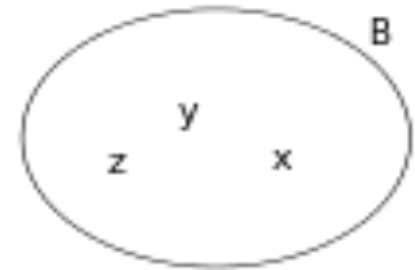
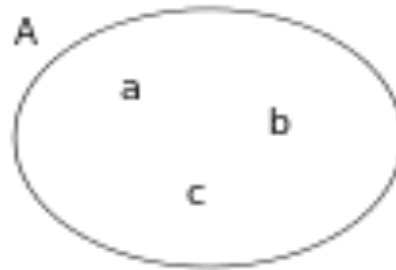
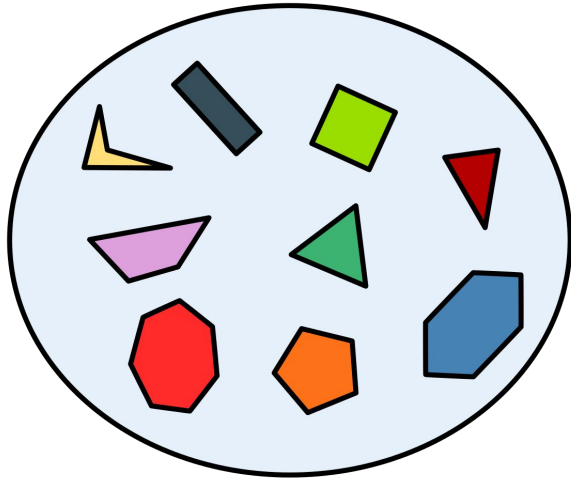
- **In general:** AI is a set of technologies, that make people better/smarter/...
- **Model:** we delegate our cognitive **functions** to AI and computers in general.

$$\mathbf{F}(\text{data} \mid \text{context})$$

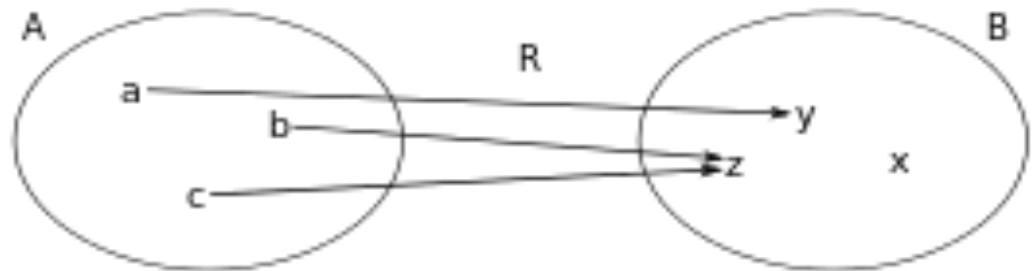
Examples:

- $\text{move} = \mathbf{F}(\text{board} \mid \text{rules, debuts, ...})$
- $\text{theorem proof} = \mathbf{F}(\text{theorem} \mid \text{axioms, production rules})$
- $\text{identity} = \mathbf{F}(\text{face image} \mid \text{feature extractor, id database})$
- ...

Sets, Cartesian product, relations, functions



$$A \times B = \{(a,x), (b,x), (c,x), (a,y), (b,y), (c,y), (a,z), (b,z), (c,z)\}$$



$$R = \{(a,y), (b,z), (c,z)\}$$



Whiteboard time



Explicit approach: brute force

- Memorize a *function* (Data x Answers)



- How much memory do we need to memorize with function?

Algorithmic approach (1)

- Memorize *winning strategies/algorithms, patterns* - write a function
 - Matches examples
- Or you can hire human professionals :)

NB these approaches will only work if the problem is considered **solved**, otherwise they will not have **generalization power**. If your system memorized a function given examples, in ML with is called **overfitting**.

Example: <http://sprotasov.ru/gomoku/>

Algorithmic approach (2): heuristics

Heuristic = “mental shortcut”

A* — heuristic greedy-like algorithm for finding **path**, working with lower bound estimate.

$$f(n) = g(n) + h(n)$$

- ***n*** is the next node on the path,
- ***g(n)*** is the cost of the path from the start node to *n*
- ***h(n)*** is a heuristic function that estimates the cost of the cheapest path from *n* to the goal

Lab #1. Write intelligent function

Write A* algorithm's missing parts that finds a path in a labyrinth. Use [this template](#).

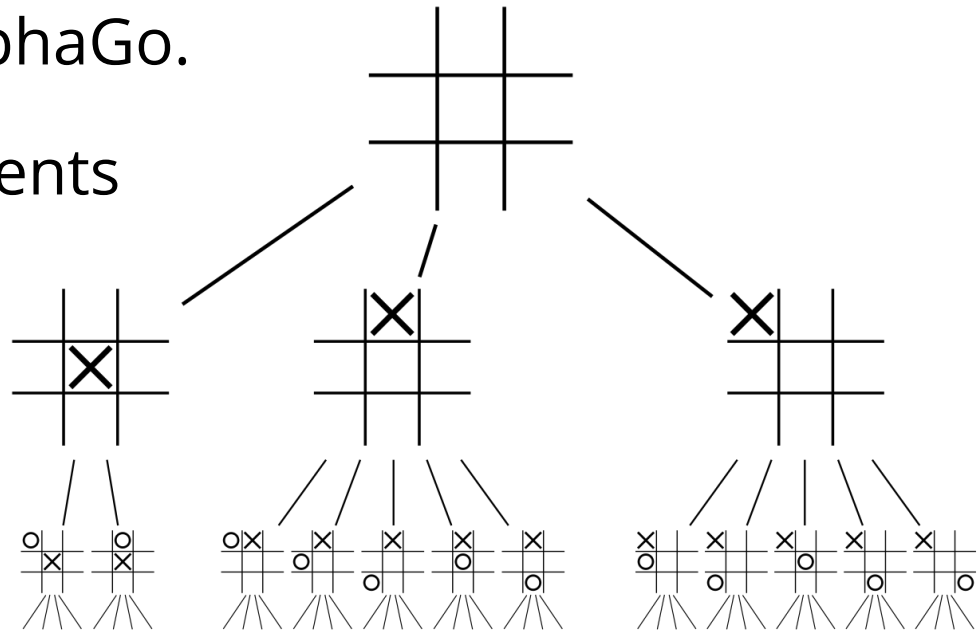
Algorithmic approach (3): trees

Board games is a special case of problem. Search space can be represented as a tree.

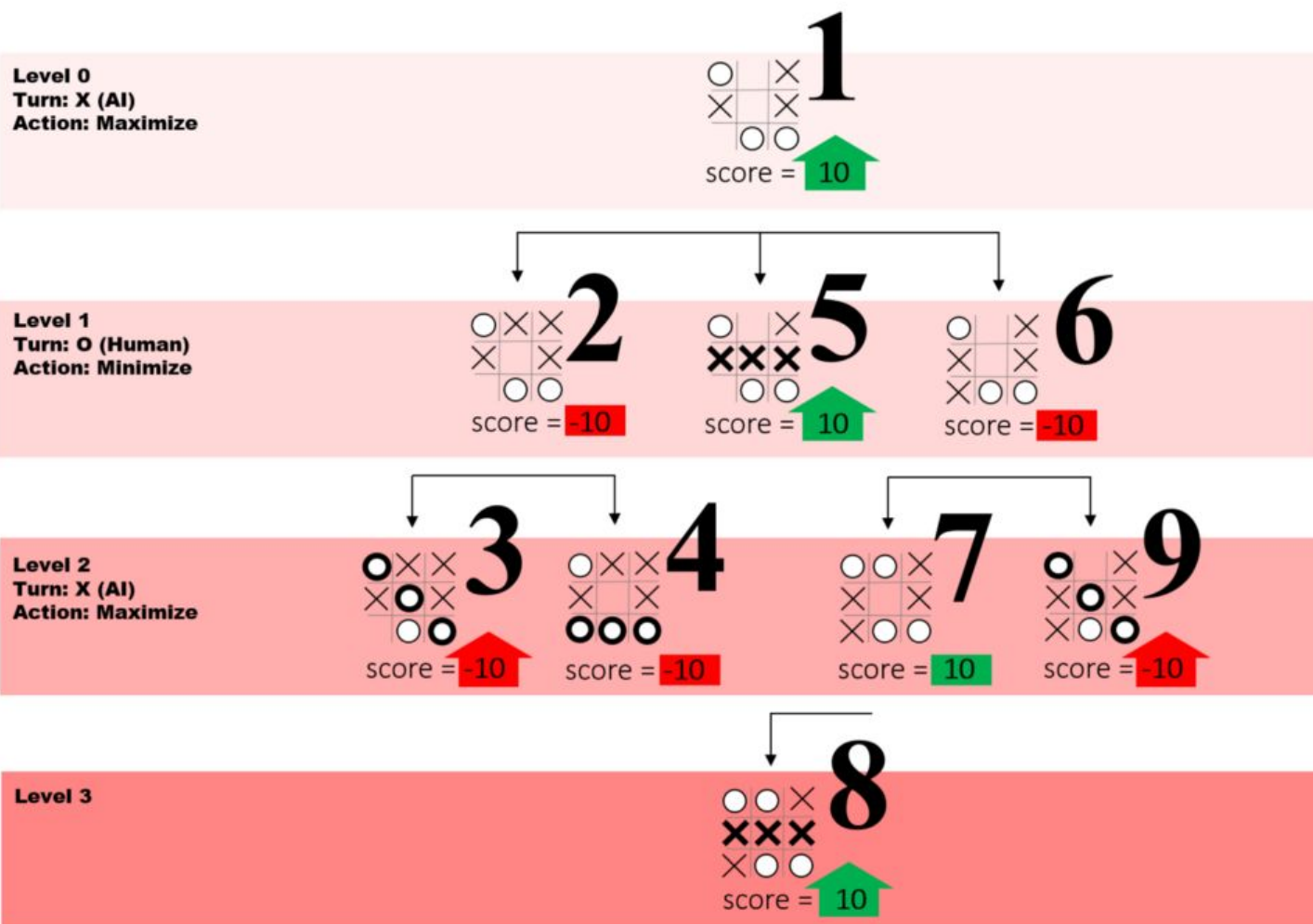
Major game bots implement tree search, e.g. [MCTS](#) is AlphaGo.

There are lots of improvements to work with such trees:

- [Alpha-beta pruning](#)
- [Branch-and-bound](#)



Minimax



Lab #2. Game tree

- 1) Write the code that builds whole tic-tac-toe game tree.
- 2) Write a function, that estimates field for being win/lose/tie.
- 3) Write a function that updated minimax weights for the whole tree.
- 4) ???
- 5) Implement minimax bot!

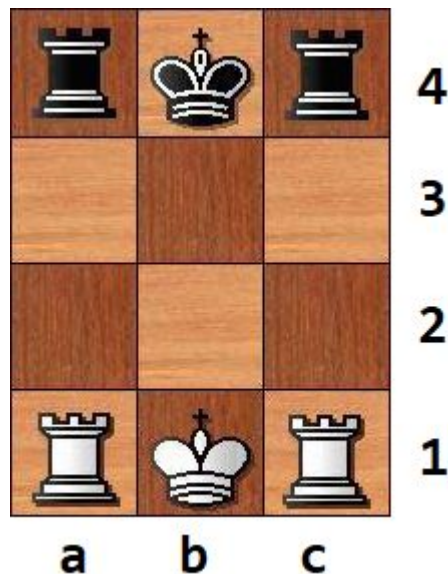
Homework #1

Write a bot that plays 3x3x3 cubic tic-tac-toe.

Consider you have a cube, where you can put a piece only on top of the floor or other piece (rule of gravity). Any row of 3 is a win. Write a bot that plays this version of XO.

Homework #2

Imaging chessboard like this



Using normal chess rules write a bot that plays these chess **for white**.
Implement any game strategy you prefer. Print the sequence (any, or shortest)
of moves that leads to win.