Teaching computer to play Doom from visual input

Anssi Kanervisto

Teaching computer to play Doom from visual input

Why teaching?

Why visual input?

Why Doom?

Teaching computer to play Doom from visual input

Why teaching?

- Computer can learn previously unseen environments
- Possibly not limited by human knowledge

Why visual input?

Why Doom?

Teaching computer to play Doom from visual input

Why teaching?

- Computer can learn previously unseen environments
- Possibly not limited by human knowledge

Why visual input?

- No need to design features per environment
- Commonly available in physical world (cameras)

Why Doom?

Teaching computer to play Doom from visual input

Why teaching?

- Computer can learn previously unseen environments
- Possibly not limited by human knowledge

Why visual input?

- No need to design features per environment
- Commonly available in physical world (cameras)

Why Doom?

- Secure, modifiable and compact environment
- Fun

Teaching computer to play

Teaching computer to play

Instead of programming computer's decision by hand, we let machine learning to find a good way to play the game!

E.g. **Reinforcement learning** includes methods and definitions for **learning from intercation**.

Teaching computer to play

Instead of programming computer's decision by hand, we let machine learning to find a good way to play the game!

E.g. **Reinforcement learning** includes methods and definitions for **learning from intercation**.

Example: Training your dog to sit on command.

1) Initial state: Dog is standing (also dog likes treats)
2) You command dog to sit.
3) Dog sits down (or trots away).
4) You give dog a treat (or not).

Repeating this with enough treats will result dog to always sit down on command.

Reinforcement learning

In terms of reinforcement learning and Doom...

Reinforcement learning

In terms of reinforcement learning and Doom...

The state: Visual image

$$s_t \in \mathbb{R}^{\text{height} \times \text{width} \times \text{channels}}$$

Time (discrete integers)

Reinforcement learning

In terms of reinforcement learning and Doom...

The action: Positive integer*

$$a_t \in \mathbb{N}$$

Time (discrete integers)

(*) Also can represent any action between earth and the heaven**
(**) Withing the limits of the provided environment***
(***) Which can also be the physical world

Reinforcement learning

In terms of reinforcement learning and Doom...

The treat: Reward function

Previous state

$$R(s_t, a_t, s_{t+1}) = \begin{cases} 1 & \text{if collects a medkit} \\ -1 & \text{if agent dies} \\ \dots & \dots \end{cases}$$

Action taken

Next state

Reinforcement learning

In terms of reinforcement learning and Doom...

The goal: Maximize treats

Interact till terminal state.
E.g. Play one game of Doom

$$\max \sum_{t=1}^{T} R(s_t, a_t, s_{t+1})$$

Learning to take good actions

Learning to take good actions

After training, the dog sits down on command as it expects to receive a treat [citation needed]. **Q-learning** takes a similar approach.

Learning to take good actions

After training, the dog sits down on command as it expects to receive a treat [citation needed]. **Q-learning** takes a similar approach.

The value of an action

$$Q(s_t, a_t) = \mathbb{E}\left[\sum_{i=t}^{T} R(s_i, a_i, s_{i+1})\right]$$

Read: Expected sum of the future reward

Learning to take good actions

After training, the dog sits down on command as it expects to receive a treat [citation needed]. **Q-learning** takes a similar approach.

Learning from the experience

Discount factor.
How short sighted agent is.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_a Q(s_t, a) - Q(s_t, a_t) \right)$$

Learning rate

Reward obtained
after taking the action

Expected reward in next state.
Connection to the future.

Learning to take good actions

After training, the dog sits down on command as it expects to receive a treat [citation needed]. **Q-learning** takes a similar approach.

Taking knowledged actions

$$a_t = \arg\max_a Q(s_t, a)$$

The iron is hot

The iron is hot

- 2013: Original Deep Q Network paper.

The iron is hot

- 2013: Original Deep Q Network paper.
- 2016, 2017: AlphaGo beats top human Go players.

The iron is hot

- 2013: Original Deep Q Network paper.
- 2016, 2017: AlphaGo beats top human Go players.
- 2015-2017: Multiple complex environments/APIs created for training agents by using games designed for humans: Minecraft, Doom, Starcraft 1 and 2, Quake, Atari.

The iron is hot

- 2013: Original Deep Q Network paper.
- 2016, 2017: AlphaGo beats top human Go players.
- 2015-2017: Multiple complex environments/APIs created for training agents by using games designed for humans: Minecraft, Doom, Starcraft 1 and 2, Quake, Atari.
- Last Friday: OpenAI bot beats top human players in 1v1 games in Dota 2

The iron is hot

- 2013: Original Deep Q Network paper.
- 2016, 2017: AlphaGo beats top human Go players.
- 2015-2017: Multiple complex environments/APIs created for training agents by using games designed for humans: Minecraft, Doom, Starcraft 1 and 2, Quake, Atari.
- Last Friday: OpenAI bot beats top human players in 1v1 games in Dota 2

Thank you for your attention!

anssi.kanervisto@uef.fi

Learning from interaction

Current state
s_t

Process the image, produce action to be taken. Action can be currently known optimal, or e.g. random action.

Action (fire)
a_t

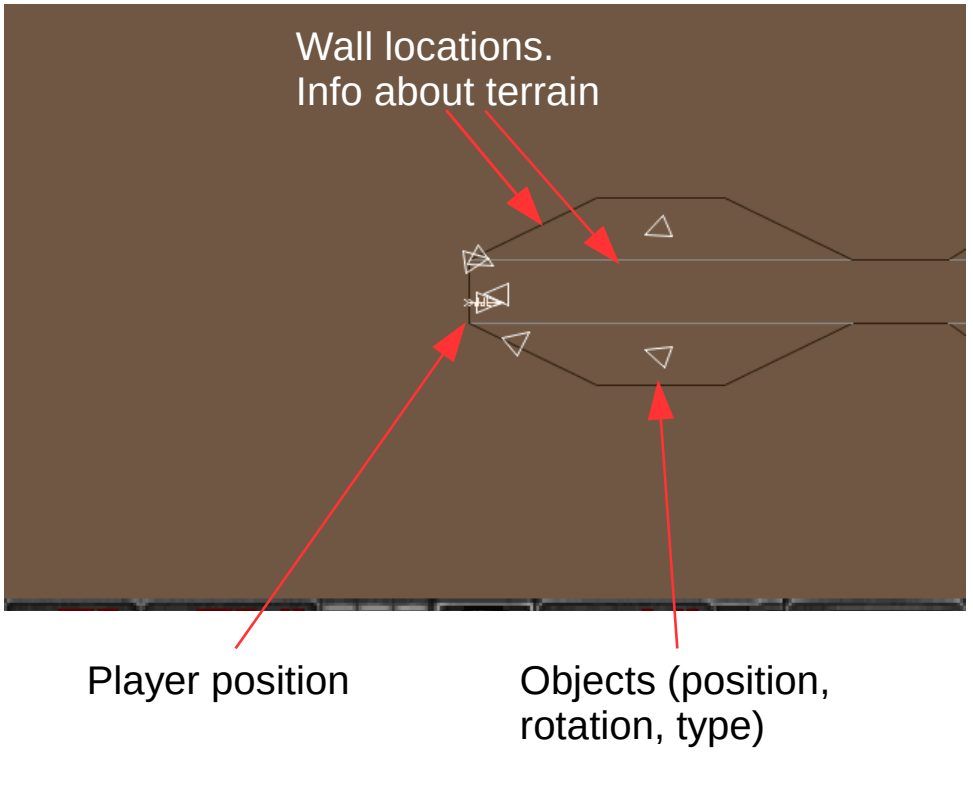
Environment processes the action and moves to next state

Successor state
s_{t+1}
r_t

Proceeding state and the reward obtained.
In this example, agent killed an enemy and thus is reward with positive reward.

Visual input

Instead of giving computer this...



… we give this.

Screen buffer / pixels.
= What human player would see