**OpenAI Five with Dota2**

On April 13th, 2019, OpenAI Five became the first AI system to defeat the world champions in an esports game. The game of Dota 2 presents novel challenges for AI systems such as long-time horizons, imperfect information, and complex, continuous state-action spaces, all of which will become increasingly central to more capable AI systems. OpenAI Five leveraged existing reinforcement learning techniques, scaling to learn from batches of approximately 2 million frames every 2 seconds. We developed a distributed training system and tools for continual training, which allowed us to train OpenAI Five for 10 months. By beating Team OG, the world champion of Dota 2, OpenAI Five shows that self-play reinforcement learning can help a computer do a hard task better than a human.

**What is Dota 2?**

Dota 2 is played on a square map with opposing bases defended by opposing teams. Each side's base has a building known as an ancient; the game finishes when the other team destroys one of these ancients. Each team consists of five players, each of whom controls a hero unit with distinct powers. During the course of the game, both teams have a steady stream of uncontrolled, little "creep" troops that move towards the opposing base and assault any opponent units or buildings. Players collect resources, such as gold, from enemies in order to boost their hero's strength by purchasing things and enhancing their powers.

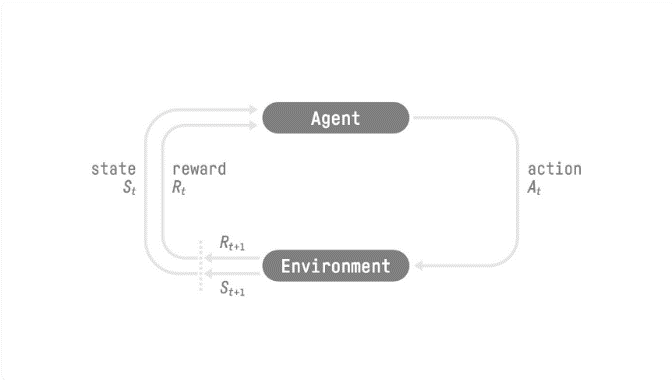
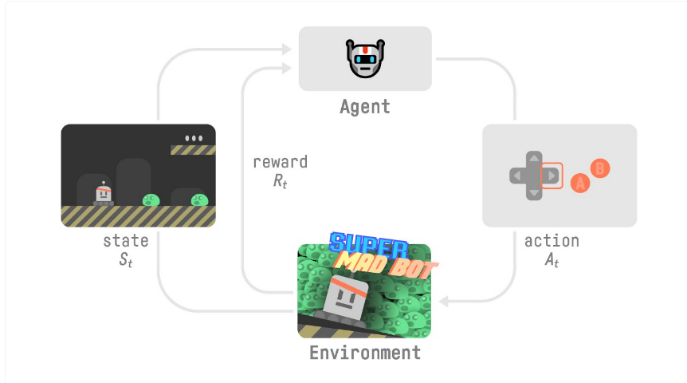
**What is OpenAI Five?**

OpenAI Five plays 180 years’ worth of games against itself every day, learning by self-play. It trains using a scaled-up version of Proximal Policy Optimization running on 256 GPUs and 128,000 CPU cores — a larger-scale version of the system we designed to play the much-simpler solo variation of the game last year. Using a distinct **LSTM** for each hero and no human data, it learns identifiable techniques. This indicates that **reinforcement learning** can yield long-term planning with large but achievable scale

**Diagram

Description automatically generatedSimplified OpenAI Five Model Architecture**

The complicated multi-array observation space is transformed into a single vector before being put through a 4096-unit LSTM. The LSTM state is anticipated to achieve the policy outcomes (actions and value function). Each of the five heroes on the squad is controlled by a duplicate of this network with inputs that are virtually identical, but each has its own concealed state. Due to a portion of the observation processing output identifying which of the five heroes is being commanded, the networks conduct distinct actions. The LSTM accounts for 84% of the model's total number of parameters.

**The Reinforcement Learning Framework**

The RL Process: a loop of state, action, reward, and next state

**What is LSTM Network?**

Long Short-Term Memory networks (LSTMs) are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997) and were refined and popularized by many people in following work. LSTMs work tremendously well on a large variety of problems and are now widely used.

LSTMs are designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

**What is Reinforcement learning?**

Reinforcement Learning is based on the premise that an agent (artificial intelligence) would learn from its environment by interacting with it (through trial and error) and obtaining incentives (positive or negative) as feedback for executing actions.

Graphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence

Fig 1

Fig 2

By clicking the correct button, Agent will interact with the surroundings (the video game) (action). He received a coin, which is a +1 reward. It's positive; he just realized that he must collect coins in this game. (Fig2) But then, he presses right again, and he touches an enemy, he just died -1 reward. (Fig 3)

Graphical user interface, text, application

Description automatically generated

Fig 3

By interacting with his environment and learning via trial and error, your agent realized he needed to obtain money while avoiding adversaries in this setting. Without supervision, the agent will become increasingly proficient at playing the game. Interaction is how both people and animals acquire knowledge. Reinforcement Learning is just a computational method for learning from experience.

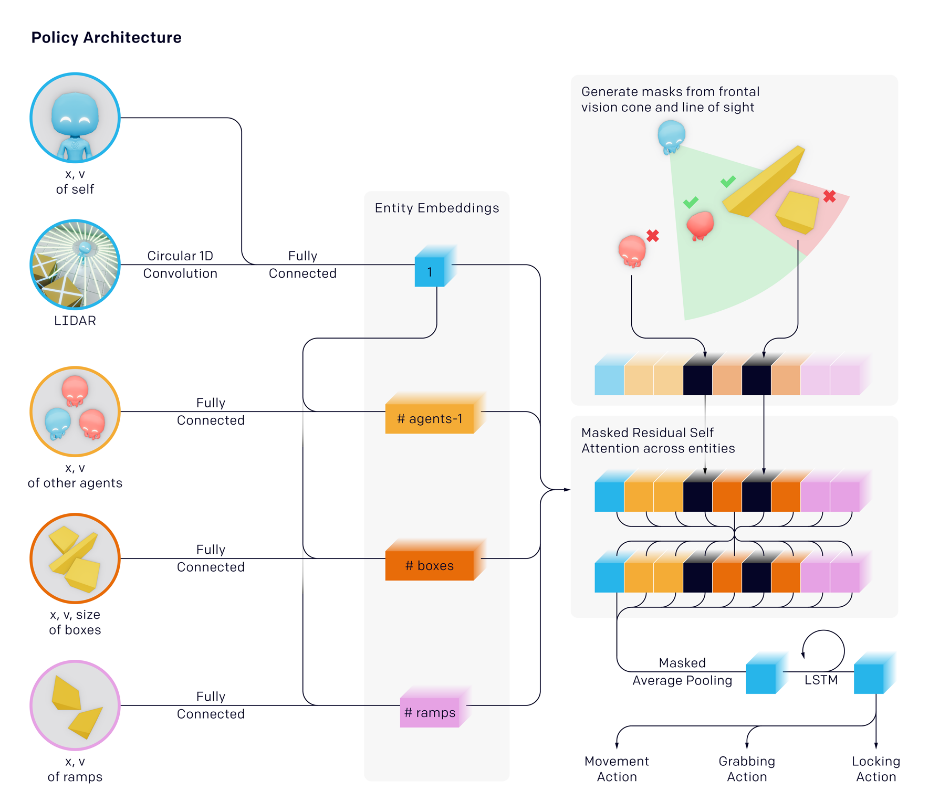
**How are they different from the class content?**

Using the minimax search technique in a competition is distinct from reinforcement learning (RL). Minimax search is based on competitive settings in which two players compete for the highest score. It may be comparable to a turn-based game in which one offensive player and one defensive player battle for the least number of points, such as chess, hex games, etc., in order to win. The algorithm for minimal search involves rounding. However, Reinforcement Learning (RL) is a computer method to action-based learning. We develop an agent that interacts with its surroundings through trial and error and receives negative or positive incentives as feedback. Any RL agent's objective is to maximize its predicted cumulative reward (also called expected return).

**Example of game**

**Multi-agent Hide and Seek from OpenAI**

Multi-agent Hide and Seek uses the same training infrastructure and algorithms used to train OpenAI Five.



Each agent in our environment behaves autonomously based on its own observations and concealed memory state. Agents adopt a permutation-invariant entity-centric state-based representation of the world with regard to objects and other agents. Similar to transformers, each item is embedded and then processed via a masked residual attention block. Objects not in the agent's line of sight and in front of the agent are disguised so that the agent is unaware of their existence.

**References**

OpenAI. 2022. *OpenAI Five*. [online] Available at: [https://openai.com/blog/openai-five/](https://openai.com/blog/openai-five/%20)

Colah.github.io. 2022. *Understanding LSTM Networks -- colah's blog*. [online] Available at: [http://colah.github.io/posts/2015-08-Understanding-LSTMs/#lstm-networks](http://colah.github.io/posts/2015-08-Understanding-LSTMs/%23lstm-networks)

Huggingface.co. 2022. *An Introduction to Deep Reinforcement Learning*. [online] Available at: [https://huggingface.co/blog/deep-rl-intro#the-reinforcement-learning-framework](https://huggingface.co/blog/deep-rl-intro%23the-reinforcement-learning-framework)

Arxiv.org. 2022. [online] Available at: <https://arxiv.org/pdf/1912.06680.pdf>

OpenAI. 2022. *Emergent Tool Use from Multi-Agent Interaction*. [online] Available at: <https://openai.com/blog/emergent-tool-use/>