Training Artificial Intelligent

Tarik Gasser

Georgia Institute of Technology

July 2020

Training Artificial Intelligence

Introduction:

Artificial Intelligence is the theory and development of a computer system able to perform tasks that normally require human intelligence, such as visualization, speech recognition, decision-making, and translation between languages (**Oxford**). In our research we are focused on building an Artificial Intelligence that is capable of decision-making. In Artificial Intelligence world, you can build a strong AI or weak AI. The big difference between strong and weak AI is with strong AI, machines can actually think and carry out tasks on their own, just like humans do. With weak AI, the machines cannot do this on their own and rely heavily on human interference (**Magazine**). In our research we are building a strong AI that has the ability to make decision on its own without given explicit answers. Additionally, Artificial Intelligence is defined as a study of rational agents. A rational agent could be anything which makes decisions, as a person, firm, machine, or software. It carries out an action with the best outcome after considering past and current percepts (**Bansall**).

The way anyone builds an AI agent is through Machine Learning using data and training. There some variations of how to define the types of Machine Learning Algorithms but commonly they can be divided into categories according to their purpose and the main categories are the following: Supervised learning, Unsupervised Learning, Semi-supervised Learning, and Reinforcement Learning (**Fumo**). In our research we are focused on Reinforcement learning. Secondly, the data that is being used for the Machine Learning part is heavily depending on the type of work that this AI is being build for. For example, if we are building an AI for trading in the stock market, then the data will be the previous numbers of stocks and their fluctuation over specific period. In our research our data is policies and actions that the agent will be solving or learning from while going through the environment.

The environment that the AI will be trained on is one of the absolute key elements to determine the right models for an AI solution (**Rodriguez**). Therefore, the characteristics of the environment are one of the absolutely key elements to determine the right models for an AI solution. There are several categories we use to group AI problems based on the nature of the environment. Examples of Artificial Intelligence Environments: Complete vs Incomplete, Fully Observable vs Partially Observable, Competitive vs Collaborative, Static vs Dynamic, Discrete vs Continue, Deterministic vs Stochastic (**Rodriguez**). In our research we are using the Deterministic vs Stochastic environment and we are using the environment for training just like in the real world. For example, a football player to become better at a football game, they usually practice in the field that they will be playing in, same idea goes to the artificial Intelligence, it needs an environment to be able to train on. In our research our environment is built like a map with nodes, each node is called a state and each state has different policies and directions for the AI agent to follow, solve or to learn from. This research is completely software, therefore, it's all code-based research besides the theories. We are building the AI agent and the environment using python as a coding language because of its extensive libraries. My contribution to the research was solving issues with the environment.

Lit review:

Building an AI has become easy using code because it is a software and there are lots of resources on how to build an AI online. However, training an AI has been a difficult task to achieve. Furthermore, research and discoveries rely on three different methods to train an AI agent to learn: supervised learning, unsupervised learning, and reinforcement learning. Each method is built differently and operates completely different than the other. Supervised learning is where the AI agent can map or assign the given input to output, such as images of things and they are labelled with what they are. Unsupervised learning is where the AI agent is responsible to find an output for a given input, such as a lot of data about fruits clustered together and the agent must identify each one. Reinforcement learning where the agent is exposed to an environment where it gets trained by trial and error (**GN**). Since we are focusing on decision-making part of Artificial Intelligence, then we are using Reinforcement Leaning from Machine Learning. Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, an artificial intelligence faces a game-like situation (**Osiński**).

Our research talks about Imitating latent policies from observation. Imitation Learning (IL) and Reinforcement Learning (RL) are often introduced as similar, but separate problems. Imitation learning involves a supervisor that provides data to the learner. Reinforcement learning means the agent must explore in the environment to get feedback signals. This crude categorization makes sense as a start, but as with many things in life, the line between them is blurry (**Seita**). The main approach of our research is Imitation Learning using Reinforcement learning. Furthermore, Reinforcement learning (RL) is one of the most interesting areas of machine learning, where an agent interacts with an environment by following a policy. In each state of the environment, it takes action based on the policy, and as a result, receives a reward and transitions to a new state. The goal of RL is to learn an optimal policy which maximizes the long-term cumulative rewards (**Lőrincz**). The way we are building this Imitation learning in our research is we are building strong AI agent with a mechanism for learning policies from observation alone without requiring access to expert actions with only a few interactions within the environment. Therefore, besides the AI agent that we are building and training, we have an expert agent that is there to help, but we are restricting our AI agent to not to access the expert actions. We can accomplish that by using Reinforcement Learning, such as give then AI agent a negative reward if he did. The expert agent is fully equipped with all of the information on what to do and what direction to go to through the environment. For example, a professor always knows the answers to the exam they give to student, but the students are not allowed to ask the professor for the answers during the exam.

Imitation learning approaches usually requires both observation and actions to learn the policies along with extensive interactions with the environment. Therefore, we are using some coding algorithm to equip the AI agent with the ability to quickly solve problems. Artificial Intelligence (AI) refers to the simulations of human intelligence in machines that are programmed to think like humans and mimic their actions (**Frankenfield**). Comparatively, human often learn from and develop experiences through mimicry. This is one of the approaches we took in our research to be able to train our AI agent. This AI agent is being trained to produce a prediction with each cycle, and each time the parameters can be adjusted to ensure that the prediction becomes more accurate with each training step (**Revain**).

Our research is built on two steps: First we teach the AI agent offline. Secondly, Learning by Observation as the agent is going through the environment. Learning by observation has been implemented many times in the previous and recent papers. One recent approach is “learning by observation,” which can be of key importance whenever agents sharing similar features want to learn from each other (**Costa**). Costa’s approach depends on two methods, learning from the observed information and categorizing the information. Also, Costa proposes that while the agent is performing a task, it needs an expert agent to learn from. Liu talks about “imitation learning” as an effective approach for autonomous systems to acquire control policies when explicit reward function is unavailable (**Liu**). A very similar approach is autonomous imitation learning techniques called Behavioral Cloning from observation, which aims to provide improved performance (**Torabi**). Behavior Cloning is considered the straight-forward approach is Behavior Cloning which is used sometimes for Autonomous Driving Cars. However, Behavior Cloning treats imitation learning as supervised learning. This Behavior Cloning usually rely heavily on the expert’s intent in a form of a reward function. Instead we are trying to build our AI agent offline before we put it through the environment and build more experience as its going through the environment.

My contribution to the research was working on the remapping issues we had with the environment. Our environment is built with different state and each state has a pair of actions and directions. Unfortunately, we noticed an issue with the mapping as the states kept flip-flopping and the wrong state would show up with the wrong actions and directions. I proposed a solution to solve the mapping issues by presenting a recent study that had a better implementation of an environment that could solve our issues. Wu built a neural net that is trained such that when it takes as input the first demonstration and a state samples from the second demonstrations. It should predict the action corresponding to the sampled state (**Wu**). Using this type of neural net with our action map could solve the re-mapping issues we are facing.

Methodologies:

In the AI industry everything is code-based because Artificial Intelligence are software. Therefore, we are using Python as a coding language because of the large number of libraries that it contains. That makes any issues that rise easier to be-bug or trace. However, not all AI are built in Python, but most researchers tends to use Python because there are a lot of resources online that can help with the code.

Our plan for the agent is to first learn the policy offline in a latent space that best describes the observed transition. Then it takes a limited number of steps in the environment to ground this latent policy to the true action labels. Since each label has the state action pair, then AI agent should be able to know its next state from the state its in.

Another method we used is leaning latent policies where the AI agent can predict the different between states while going through the environment. This can be achieved by simply make prediction based on the observed next state after taking each action. We also used action remapping in order to imitate from expert observation. The AI agent needs to learn a mapping from the latent policy learned in the previous step to the correct action. Note: the expert sometimes is used like a mentor to the agent while its being trained to help the agent to pick the best solution since the expert is implemented known the correct path, but the agent does not.

We evaluated our approach in four environments: Classic control with Cartpole, Acrobot, Mountain Car, and a recent game by OpenAI. These different types of environments help to show how well the AI is trained and evaluate its performance. Those online resources are very good with accuracy and known in the AI industry. Therefore, we are using the results of how well our AI agent perform to compare with the other recent research.

The method that I added to the research is the One-Shot Imitation Learning neural net by Wu. The architecture of the neural net is learning block stacking which the neural net will be able to receive a demonstration trajectory as input and produce an embedding of the demonstration to be used by the policy. In addition, the size of this embedding grows linearly as a function of the length of the demonstration as well as the number of states in the environment. The implementation of the neural net will take some code manipulation to be able to integrate this into our research.

Results:

Our research described and shows how the agent can learn to imitate latent policies from only the expert agent with very few interactions to the environment. We also show that our method “Latent Policy from Imitation Learning” is able to outperform “Behavior Cloning Observation Policy”. Our research will be able to produce faster and more efficient results by training the agent offline before putting it through the environment.

While evaluating our approach within classic control environment we can show that our research performs better than the standard approaches. Furthermore, we show that our approach in four environments: classic control with cartpole, acrobots, and mountain car and recent platform by OpenAI. The results from these environments shows that our AI agent is performing very well and learning the correct policies. We show that our approach can perform as well as the expert after just a few steps of interaction with the environment and performs better than a recent approach for imitating form observation such as behavior coning form observation.

Discussion and conclusion:

We believe that this research will serve a step-in stone to teach the agent offline with fewer interactions to the environment not just for similar agent, but also to any agent whose actions are unknown. Lastly, our research could be used for pre-training imitation by observation.

Future implementations, we can test that the neural net from “One-Shot Imitation Learning” is effective and efficient enough to solve our remapping issues. This will help enforcing stronger local consistencies between latent actions and generate prediction across different states

Citation

Bansall, S. (n.d.). Agents in Artificial Intelligence. *Machine Learning*. Retrieved from https://www.geeksforgeeks.org/agents-artificial-intelligence/.

Magazine, E., & Techjury, S. (2020, February 06). Strong vs. Weak Artificial Intelligence - e3zine – E-3 Magazine International. Retrieved from https://e3zine.com/strong-artificial-intelligence/

Oxford. (n.d.). Artificial Intelligence. Retrieved from https://www.oxfordreference.com/view/10.1093/oi/authority.20110803095426960

Fumo, D. (2017, August 17). Types of Machine Learning Algorithms You Should Know. Retrieved from https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861

Rodriguez, J. (2017, January 12). 6 Types of Artificial Intelligence Environments. Retrieved from https://medium.com/@jrodthoughts/6-types-of-artificial-intelligence-environments-825e3c47d998

Lőrincz, Z. (2019, September 19). A brief overview of Imitation Learning. Retrieved from https://medium.com/@SmartLabAI/a-brief-overview-of-imitation-learning-8a8a75c44a9c

Osiński, B. (2020, July 23). What is reinforcement learning? The complete guide. Retrieved from https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/

Seita, D. (n.d.). Combining Imitation Learning and Reinforcement Learning. Retrieved from https://danieltakeshi.github.io/2019/04/30/il-and-rl/

Wu, Duan, and Yiannis Demiris. "One Shot Learning by Imitation." 2010 IEEE International Conference on Robotics and Automation. Retrieved from http://papers.nips.cc/paper/6709-one-shot-imitation-learning.pdf

Revain. (2019, March 28). How to Train Your AI. Retrieved from https://medium.com/revain/how-to-train-your-ai-98113bdac101.

Frankenfield, J. (2019, November 18). How Artificial Intelligence Works. Retrieved June 13, 2019, from https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp.

Marr, B. (2019, April 1). What Is The Importance Of Artificial Intelligence (AI). Retrieved from https://bernardmarr.com/default.asp?contentID=1829.

GN, C. K. (2018, September 6). Machine Learning Types and Algorithms. Retrieved from https://towardsdatascience.com/machine-learning-types-and-algorithms-d8b79545a6ec.

Costa, Paulo. "Learning by Observation of Agent Software Images." Https://Arxiv.org/Ftp/Arxiv/Papers/1402/1402.0591.Pdf, May 2013.

Liu, Gupta, Abhishek, Abbeel, Pieter, Levine, & Sergey. (2018, June 18). Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation. Retrieved from https://arxiv.org/abs/1707.03374.

Torabi, Faraz, Warnell, Garrett, & Peter. (2018, May 11). Behavioral Cloning from Observation. Retrieved from https://arxiv.org/abs/1805.01954

Wu, Duan, and Yiannis Demiris. "One Shot Learning by Imitation." 2010 IEEE International Conference on Robotics and Automation. Retrieved from http://papers.nips.cc/paper/6709-one-shot-imitation-learning.pdf