MACHINE LEARNING INTRODUCTION

NGUYEN BINH ARTHUR
MAKARISON JOLYANE

The goal of this course is to provide an introduction to the field of machine learning with the example of reinforcement learning.

What is Machine Learning?

Machine learning (ML) is a subset of artificial intelligence (AI) that enables computers to learn from data and improve their performance over time without being explicitly programmed for every task. At its core, machine learning algorithms use statistical methods to find patterns in data and make predictions or decisions. The beauty of machine learning lies in its ability to adapt to new data independently, allowing for applications ranging from email filtering to more complex decisions in robotics, financial modeling, and beyond.

Types of Machine Learning

Machine learning can be broadly categorized into three main types, each with its own approach to learning and application areas:

1. Supervised Learning

Supervised learning involves learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. Each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. Common applications include spam detection in emails, sentiment analysis, and image recognition.

2. Unsupervised Learning

Unsupervised learning, in contrast, deals with learning patterns from untagged data. Here, the system tries to learn without a teacher. It’s left on its own to find structure in its input. Unsupervised learning can discover hidden patterns in data like customer segmentation in marketing or detecting anomalous transactions in finance. Common methods include clustering, dimensionality reduction, and association rules.

3. Reinforcement Learning

Reinforcement learning (RL) is a type of machine learning where an agent learns to make decisions by taking actions in an environment to achieve some objectives. The learner is not told which actions to take but instead must discover which actions yield the most reward by trying them. Reinforcement learning is different from supervised learning in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected.

In reinforcement learning, an agent learns from the consequences of its actions, rather than from being taught explicitly. It selects actions based on past experiences (exploitation) and also by trying new things (exploration). The goal is to find a balance between exploration of uncharted territory and exploitation of current knowledge.

What is Reinforcement Learning?

Reinforcement learning is both a machine learning paradigm and a statement about learning by interacting with an environment to achieve a goal. The learner, or agent, discovers through trial and error which actions yield the most rewards. These rewards can be immediate or cumulative, allowing for both short-term gains and long-term strategy development.

Key Concepts in Reinforcement Learning:

Agent: The learner or decision-maker.
Environment: Everything the agent interacts with.
Action: All possible moves the agent can make.
State: A situation returned by the environment.
Reward: Feedback from the environment to assess the last action.

Applications of Reinforcement Learning:

Reinforcement learning has been successfully applied in various fields, including robotics for industrial automation and autonomous vehicles, game playing (e.g AlphaGo), recommendation systems, and more.

Before we conclude, it's important to understand how the principles of reinforcement learning can be applied in practical scenarios. One of the most exciting ways to see reinforcement learning in action is through the use of MLAgents in Unity. Unity, a popular game development engine, offers the MLAgents Toolkit: a flexible and powerful platform for training intelligent agents in complex, interactive environments. This toolkit allows developers and researchers to implement and test their reinforcement learning algorithms in a visual and intuitive setting.

By leveraging MLAgents, we can create simulations where agents navigate environments, solve puzzles, or even play games, learning optimal behaviors through the trial-and-error process inherent in reinforcement learning. In the following demonstrations, we will explore how to set up a basic environment in Unity and train an agent using the reinforcement learning techniques discussed. This hands-on approach will not only solidify your understanding of reinforcement learning concepts but also illustrate the potential of these algorithms to solve real-world problems in creative and innovative ways.

Course Questions

1. Train the Model Using ML-Agents

Using the default neural network provided in ML-Agents, configure and initiate the training of your agent in Unity. Remember to utilize multiple environments simultaneously to accelerate the training process. Describe the steps you took to start the training and any parameters you adjusted to improve learning efficiency.

2. Load the Trained Model and Launch a Test Run

After training your model, ML-Agents outputs the trained neural network as a .onnx file, commonly referred to as a "brain". Load this brain into your Unity project and initiate a test run with the agent using this trained model. Outline the process of importing the .onnx file and starting the test run in Unity.

3. Observe and Analyze the Results

During the test run, closely observe your agent's behavior and performance. Compare the agent's actions and decision-making process before and after training. What improvements or changes do you notice? Are there areas where the agent still struggles? Provide a detailed analysis of the test run's results.

4. Implement an Additional Dimension for Agent Displacement

Enhance the complexity of your agent's environment by adding another dimension to its displacement abilities. This could involve allowing the agent to move in a new direction, jump, or any other form of movement that was not previously part of its action space. Describe the changes you made to both the agent and the environment to support this new dimension of displacement. Reflect on how this addition impacts the agent's learning process and its ability to navigate the environment.

Conclusion

This introduction has skimmed the surface of machine learning, highlighting its main types and delving into reinforcement learning. The vast and dynamic nature of machine learning makes it an exciting field, with reinforcement learning opening up avenues for developing systems that improve autonomously through experience. With practical tools like MLAgents in Unity, we have the opportunity to witness firsthand the power of reinforcement learning in creating intelligent, adaptive agents. As we move forward, the potential applications of reinforcement learning and other machine learning methods continue to expand, promising significant advancements in technology and automation.