UNI FR
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

eXascale Infolab

UNIVERSITY OF FRIBOURG

BACHELOR THESIS

Thesis Title

Author:
David Gauch

Supervisor:
Prof. Dr. Philippe Cudré-Mauroux
Giuseppe Cuccu

January 01, 1970

eXascale Infolab
Department of Informatics

Boulevard de Pérolles 90 • 1700 Fribourg • Switzerland
phone +41 (26) 300 84 65 • diuf-secr@unifr.ch • www3.unifr.ch/inf

Abstract

David Gauch

Thesis Title

Write the thesis abstract here. Should be between half-a-page and one page of text, no newlines.

Keywords: keywords, list, here

Contents

List of Figures

Chapter 1

Introduction

1.1 Reinforcement learning

Reinforcement learning is a type of machine learning that focuses on training agents to make decisions in dynamic environments in order to maximize a reward signal. It is distinct from other paradigms such as supervised learning, which uses labeled data to predict outputs for unseen data, and unsupervised learning, which seeks to find patterns in unlabeled data. In addition to being a problem that can be addressed with specific solution methods, reinforcement learning is also the field of study that examines this problem and its potential solutions. Overall, the goal of reinforcement learning is to teach agents to make optimal decisions in order to achieve a desired outcome or reward.(Sutton and Barto, 2018).

Classical reinforcement learning involves the interaction between two main components: an environment and an agent. The environment is represented by a current state that provides information about the "world" in which the agent is located, and the agent is the individual who performs actions within this environment. In each interaction, the agent receives observations based on the current state of the environment and selects an action to take. This action is then transmitted to the environment, which updates its internal state and provides feedback to the agent in the form of observations and a reward. The reward signal indicates whether the action was suitable for completing the task, while the observations provide an overview of the updated environment. Figure 1.1 illustrates one timestep of interaction between the agent and the environment.

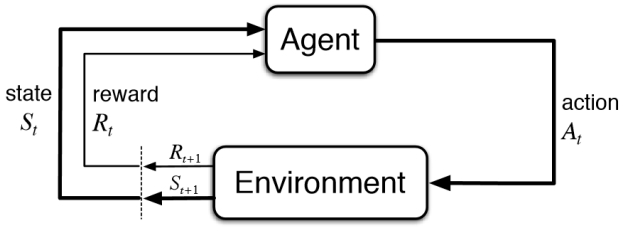

FIGURE 1.1: **Main interaction of the agent and the environment in reinforcement learning.** At the beginning (timestep t) the agent gets the observation S_t and the reward R_t from the environment. The agent performs then action A_t and sends it to the environment. The environment changes its state and returns a new observation S_{t+1} and a new reward R_{t+1}.

In reinforcement learning, the policy is a key element of the framework that determines the actions an agent should take in different states of the environment. The policy is represented by a mapping from states to actions, and it can be either deterministic or stochastic. Deterministic policies specify a single action to take in each state, while stochastic policies specify probabilities for different actions to occur. The reward an agent receives depends on the chosen policy, and the sequence of states reached by the agent is called a Markov chain. This mathematical concept models systems that change over time in a way that depends only on the current state and not on the history of past states.(Sutton and Barto, 2018).

The value function is a key concept in reinforcement learning that allows us to evaluate the effectiveness of different policies.$V_\pi(s)$ defines the expected total reward that an agent can expect to receive by following the policy π, starting from state s. One way to compute the value function is using the Bellman equation, which expresses the value of a state in terms of the values of its successors(Barron and Ishii, 1989). However, the Bellman equation does not have a closed-form solution, which makes it challenging to compute the value function in practice. The value function can be used to define an optimal policy, which is the policy that is expected to maximize the reward over time. Another way to analyze policies is using the Q-function.$Q_\pi(s, a)$ is defined as the expected total reward acquired by the agent following policy π starting from state s and taking action a. The Q-function can be related to the value function through the equation $V_\pi(s) = Q_\pi(s, \pi(s))$. A common method for finding the optimal Q-function is Q-learning, which is an iterative process that updates the Q-function based on experience (Watkins and Dayan, 1992).

Reinforcement learning is well-suited for autonomous systems that learn to achieve a desired outcome through trial and error. However, this paradigm presents a unique challenge that is not encountered in supervised or unsupervised learning: balancing exploitation and exploration. Exploitation refers to the process of repeating actions that have resulted in positive rewards in the past, in order to maximize the cumulative reward. On the other hand, exploration involves trying new actions in order to potentially discover higher rewards and avoid getting stuck in a local optimum. Finding the right balance between these two approaches is crucial for the success of the learning process.

While reinforcement learning has been effective in solving a range of tasks, it has also encountered challenges in real-world applications (Zhu et al., 2020). In this thesis, however, the paradigm is sufficient for addressing the desired tasks. Overall, reinforcement learning offers a powerful tool for training agents to make decisions in dynamic environments and optimize for a given reward signal.

1.2 Black-Box Optmization

In mathematics, optimization refers to the process of finding the maximum or minimum value of an objective function. Neural networks, for example, try to find the best weights for approximating an underlying function using techniques such as backpropagation and gradient descent. However, these techniques require knowledge of the derivative of the function, which may not always be available or may be too complex to compute (Schaul, n.d.). Black-box optimization is a method that does not rely on any assumptions about the function or its properties, and can be

used to optimize any function approximator. It is based on a feedback score similar to reinforcement learning, and the parameter set is improved based on this score (Anderson, 1995).

Black-box optimization methods are generally less efficient than traditional techniques such as gradient descent because they do not take advantage of information about the structure of the function being optimized. This means they must explore a larger space of possible solutions, which can be time-consuming. However, black-box optimization methods can be effective in situations where the function being optimized is highly complex or has a large number of variables, and traditional methods may not be applicable. They are also flexible and can be applied to a wide range of problems without requiring any knowledge of the function being optimized

1.2.1 Evolution strategies

Evolution strategies (ES) is a class of evolutionary algorithms that is specialized for optimization of continuous variables. Inspired by natural evolution, ES is a black-box optimization algorithm that uses a process of mutation and selection to search for good solutions to a given problem. In the main loop of the algorithm, new individuals are created by mutating the parent individuals of the current generation. An individual in the context of ES refers to a specific set of parameters being optimized by the algorithm. A population is a group of individuals being considered by the algorithm at a given time, and a generation refers to one iteration of the main loop. The fitness of an individual is a measure of its performance or quality, based on the feedback score provided by the algorithm.

The main loop of the ES algorithm consists of creating new individuals from the parent individuals of the current generation, evaluating their fitness, and selecting the best-performing individuals to be the parent individuals for the next generation. This process continues until a sufficient solution is found, as determined by a stopping criterion. Algorithms differ in the number of offsprings created per generation, the number of selected individuals for the next generation, and how the mutation process is performed (Salimans et al., 2017). Other than gradient-descent based methods, ES generates multiple individuals and by that explores different areas or paths of the optimization space independently, which can be beneficial for avoiding local optima and solving real-world problems that may require sophisticated exploration mechanisms.

1.2.2 Covariance Matrix Adapation Evolution Strategy

1.3 Binary trees

Trees are a commonly used data structure in mathematics and computer science that are composed of nodes and edges. A tree is an undirected, connected, acyclic graph, which means that it consists of a set of nodes that are connected by edges, but there are no loops or cycles in the graph. In a tree, a node that connects other nodes is called a parent node, and the nodes that are connected to it are called children nodes. The node at the top of the tree, which has no parent nodes, is called the root node, and the nodes at the bottom of the tree, which have no children, are called leaf nodes. The levels of a tree are determined by the distance from the root node, with the root node being at level 0 and the nodes connected to it being at level 1, and so on. Nodes on the same level are called sibling nodes.

Binary trees are a special type of tree in which all nodes except for the leaf nodes have at most two children nodes. These children nodes are typically referred to as the left node and the right node, respectively. An example of a binary tree is shown in Figure 1.2.

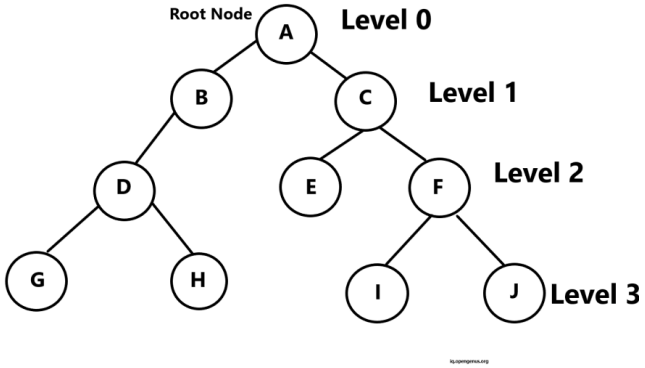

FIGURE 1.2: **Binary tree with letters representing the data**

Bibliography

Anderson, James A. (1995). *An Introduction to Neural Networks*. en. Google-Books-ID: _ib4vPdB76gC. MIT Press. ISBN: 978-0-262-51081-3.

Barron, E.N. and H. Ishii (Sept. 1989). "The Bellman equation for minimizing the maximum cost". en. In: *Nonlinear Analysis: Theory, Methods & Applications* 13.9, pp. 1067–1090. ISSN: 0362546X. DOI: 10.1016/0362-546X(89)90096-5. URL: https://linkinghub.elsevier.com/retrieve/pii/0362546X89900965 (visited on 12/16/2022).

Salimans, Tim et al. (Sept. 2017). *Evolution Strategies as a Scalable Alternative to Reinforcement Learning*. en. arXiv:1703.03864 [cs, stat]. URL: http://arxiv.org/abs/1703.03864 (visited on 12/19/2022).

Schaul, Tom (n.d.). "Studies in Continuous Black-box Optimization". en. In: ().

Sutton, Richard S. and Andrew G. Barto (Nov. 2018). *Reinforcement Learning, second edition: An Introduction*. en. Google-Books-ID: uWV0DwAAQBAJ. MIT Press. ISBN: 978-0-262-35270-3.

Watkins, Christopher J. C. H. and Peter Dayan (May 1992). "Q-learning". en. In: *Machine Learning* 8.3, pp. 279–292. ISSN: 1573-0565. DOI: 10.1007/BF00992698. URL: https://doi.org/10.1007/BF00992698 (visited on 12/16/2022).

Zhu, Henry et al. (2020). "THE INGREDIENTS OF REAL-WORLD ROBOTIC REINFORCEMENT LEARNING". en. In: p. 21.