Faculty of Computer and Informatics Department of Computer Engineering

Graduation Project Charter BLG4901(E)

Project Title

Recommendation Systems based on Reinforcement Learning

Short Summary (Abstract)

The graduation project focuses on developing an advanced recommender system utilizing reinforcement learning techniques, including Q-learning, deep Q-learning, deep double Q-learning, and graph neural networks (GNNs). Deep Q-learning enhances traditional Q-learning by using neural networks to approximate the Q-value function, enabling the model to manage high-dimensional state spaces and learn optimal policies through environmental interaction [1]. Deep double Q-learning further improves upon this by addressing overestimation bias through the use of two separate networks for action selection and value estimation, leading to more accurate updates [2]. Meanwhile, GNNs excel at modeling complex user-item relationships by leveraging graph structures, where nodes represent users and items, and edges denote interactions [3]. They aggregate information from neighboring nodes to enhance representation learning, effectively tackling challenges such as cold-start problems and scalability [3]. By integrating these methods, the project aims to create a robust and efficient recommender system that showcases the effectiveness of combining various machine learning strategies in real-world applications.

Keywords

Reinforcement Learning, Deep Q-learning, Graph Neural Networks, Deep Double Q-learning, Recommender System, Machine Learning, Cold-start Problem, Policy Optimization, Feedback Loop.

Project Team

Id	Name	E-mail	Major
150200097	M. Can ÇALIŞKAN	caliskanmu20@itu.edu.	Computer Engineering
150200016	Yusuf ŞAHİN	sahiny20@itu.edu.tr	Computer Engineering

Hardware/Software Requirements

The project does not require any additional hardware components beyond a standard setup. Several libraries and frameworks are necessary, including TensorFlow or PyTorch for deep learning models, reinforcement learning environments, and Numpy, Pandas, and Scikit-learn for data processing and machine learning tasks. For data visualization, Matplotlib and Seaborn will be utilized. Version control will be managed using Git, with repositories hosted on GitHub. Docker is optional but can be used for environment management.

Work Breakdown, Deliverable and Time Plan

#	Task	Start Date	Due Date	Deliverable
1	Literature Review	Nov 1st, 2024	Jan 10th, 2025	Research notes
2	Environment Setup	Jan 16th, 2025	Feb 15th, 2025	Configured environment
3	Recommender System Design	Feb 16th, 2025	Mar 20th, 2025	Design documentation
4	Algorithm Implementation	Mar 1st, 2025	Apr 25th, 2025	Models and results
5	Testing and Optimization	May 1st, 2025	May 13th, 2025	Performance reports
6	Final Evaluation	May 15th, 2025	May 28th, 2025	Final model
7	Paper and Presentation	May 15th, 2025	May 28th, 2025	Paper and slides

Evaluation Criteria and Project Objectives

Deliverable	Evaluation Criteria	Objective
Literature Review	Depth of analysis	Gain insights into reinforcement learning
Environment Setup	Functionality of tools	Ensure proper setup for development
Learning Model	Performance metrics	Establish a foundational model
Algorithm Integration	Efficiency improvements	Optimize system performance
Final Report	Clarity and completeness	Effectively communicate project results

Bibliography

[1] J. Nuer, "A Comprehensive Guide to Deep Q-Learning", Medium, May 20, 2022. [Online]. Available: https://medium.com/@jereminuerofficial/a-comprehensive-guide-to-deep-q-learning-8aeed632f52f. [Accessed: Oct. 24, 2024].

[2] C. Yoon, "Double Deep Q Networks: Tackling Maximization Bias in Deep Q-Learning", Towards Data Science, Jul. 17, 2019. [Online]. Available: https://towardsdatascience.com/double-deep-q-networks-tackling-maximization-bias-in-deep-q-learning-905dd8325412. [Accessed: Oct. 24, 2024].

[3] R. Merritt, "What Are Graph Neural Networks?", NVIDIA Blog, Oct. 24, 2022. [Online]. Available: https://blogs.nvidia.com/blog/what-are-graph-neural-networks/. [Accessed: Oct. 24, 2024].