Reinforcement Learning Project Proposal

Angad Singh, U20220011
Bharat Jain, U20220026
Jiya Agrawal, U20220047
Pratham Arora, U20220068
Plaksha University
{angad.singh, bharat.jain, jiya.agrawal, pratham.arora}@plaksha.edu.in

This proposal consists of two sub-projects. The first focuses on comparing and analyzing 4-5 different reinforcement learning models on a common dataset. The second aims to optimize ad placement by minimizing its negative impact on user experience while maximizing ad revenue. Our implementation is contingent upon receiving access to the required dataset, which we have formally requested.

***Abstract*—This project explores reinforcement learning for optimizing ad selection using the Criteo dataset. We compare contextual bandits (Epsilon-Greedy, UCB, SoftMax) and deep RL models (DQN, DDPG) to maximize user engagement. Evaluation uses Inverse Propensity Scoring (IPS), Click-Through Rate (CTR), and other metrics. Our approach enhances ad placement strategies, balancing revenue, and engagement.**

***Index Terms*—Reinforcement Learning, Contextual Bandits, Deep Q-Networks (DQN), Deep Deterministic Policy Gradient (DDPG), Ad Selection, Click-Through Rate (CTR), Inverse Propensity Scoring (IPS), Online Advertising, Counterfactual Learning, User Engagement Optimization.**

I. Introduction

The Criteo Ad Placement Challenge provides a dataset that captures real-world online advertising scenarios where a policy selects an ad to display to maximize user engagement (measured via click-through rate, CTR). Each impression consists of multiple candidate ads, and the goal is to design a policy that intelligently selects the most effective ad based on user context and candidate features. Unlike standard CTR prediction, where the goal is to estimate the probability of a click for a given ad, this challenge involves counterfactual learning—evaluating and optimizing policies using logged interactions, where the effect of alternative ad choices is unknown. Reinforcement learning (RL) offers a natural framework to address this problem by learning an optimal selection strategy over time.

The dataset consists of logged interactions formatted as ¡user impression, candidate set, selected product, click/no-click¿ pairs. Each impression is represented by M candidate ads, each described by a 74,000-dimensional sparse vector encoding 33 categorical features and 2 numeric features. Additionally, inverse propensity weighting (IPW) information is provided to allow unbiased evaluation of counterfactual policies.

II. Proposed Evaluation

The winning approach in the Criteo Ad Placement Challenge was based on the Follow-The-Regularized-Leader Proximal (FTRL-Proximal) algorithm, a powerful online learning method designed for large-scale, sparse classification problems. FTRL-Proximal is particularly well-suited for high-dimensional datasets, such as those encountered in computational advertising, due to its ability to efficiently update weights without requiring full batch gradient computations.

The key advantage of FTRL-Proximal lies in its use of adaptive per-coordinate learning rates combined with L1/L2 regularization, allowing it to maintain sparse and stable weight updates. In this competition, the winning model was an ensemble of 10 FTRL-Proximal models, each trained with the following hyperparameters:

- Learning rate parameters: $\alpha = 0.1, \beta = 1$
- Regularization terms: $L_1 = 75, L_2 = 25$
- Additional hyperparameters: $C = 850100, M = 15$

By averaging the predictions of multiple trained models, the approach helped mitigate high variance issues and improved policy stability. The effectiveness of FTRL-Proximal in this setting highlights its suitability for real-time ad selection, where models must adapt to dynamic user interactions while efficiently handling large-scale feature representations.

We will leverage reinforcement learning techniques to develop an optimized ad selection policy using the Criteo dataset. Our approach will focus on:

A. Contextual Bandits

- **Epsilon-Greedy**: A simple yet effective strategy that balances exploration and exploitation by selecting random ads with a small probability.
- **Upper Confidence Bound (UCB)**: A method that prioritizes ads with higher uncertainty in their estimated reward.
- **SoftMax (Boltzmann Exploration)**: A probabilistic approach that assigns higher selection probabilities to ads with greater expected rewards.

B. Deep Reinforcement Learning (DRL) Approaches

Deep Q-Networks (DQN): Inspired by the DEAR framework, we will treat ad placement as a Markov Decision Process (MDP), where an agent sequentially selects ads to maximize long-term reward. The DQN will approximate the Q-function to evaluate ad selections based on the given user context and candidate set.

- **State (S)**: The feature vector representing the user context and candidate ads.
- **Action (A)**: Selecting an ad from the candidate set.
- **Reward (R)**: Whether the user clicks on the ad (1) or not (0).
- **Transition (P)**: The next state after ad selection is influenced by user engagement.
- **Discount Factor (γ)**: Determines how much future rewards influence current decisions.

Training Process: The DQN model learns by:

- Collecting experiences (s, a, r, s') from logged impressions.
- Storing experiences in a replay buffer to break temporal correlations.
- Using a target network to stabilize Q-value updates and avoid divergence.

Why Effective? The DEAR framework applied DQN to jointly optimize ad interpolation, positioning, and selection. In our case, we simplify the problem to focus on selecting the best ad for each impression.

Deep Deterministic Policy Gradient (DDPG): Following the CS229 study, we will explore how actor-critic methods can optimize ad pacing and ensure efficient budget utilization, maintaining both revenue and user engagement.

Why DDPG? Unlike DQN, which works well for discrete action spaces (selecting from a fixed set of ads), DDPG extends to continuous action spaces, making it suitable for ad pacing optimization—ensuring budget is spent evenly while maximizing engagement.

How It Works: DDPG uses an actor-critic architecture-

- **Actor Network**: Learns the optimal policy (π) by mapping states to actions (ad selections).
- **Critic Network**: Evaluates the Q-value of actions taken by the actor.
- Updates are performed using experience replay and target networks, similar to DQN.

Application in Ad Placement:

- Can be used to control ad pacing, ensuring an optimal trade-off between revenue generation and user engagement.
- Helps dynamically adjust selection probabilities based on available budget constraints.

Evaluation Criterion: The policy function we develop will take M candidate ads, each represented as a 74,000-dimensional sparse vector, and assign scores to them, ranking them from 1 to M. The selected ad will then be evaluated based on whether a user clicked on it (reward = 1) or not (reward = 0). Since the dataset was generated from a stochastic logging policy, evaluation will use Inverse Propensity Scoring (IPS) to ensure an unbiased estimate of the aggregate reward over a held-out test set.

By leveraging these techniques, we aim to design the above-discussed models and compare and analyze the results of all the 4-5 models we can build. We aim to utilize metrics like IPS, as mentioned above, in addition to Click-Through-Rates, Expected Cumulative Rewards, Exploration Ratio, and more to conduct an in-depth analysis of the different models implemented on this dataset.

Our work contributes to the broader field of online advertising optimization by providing a reinforcement learning-based approach that can enhance user engagement and maximize advertiser revenue.

REFERENCES

[1] Zhao, X., Gu, C., Zhang, H., Yang, X., Liu, X., Tang, J., & Liu, H. (2021, May). DEAR: Deep reinforcement learning for online advertising impression in recommender systems. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, No. 1, pp. 750-758).

[2] Chen, Y. CS229: Apply Reinforcement Learning on Ads Pacing Optimization.

[3] https://www.aicrowd.com/challenges/nips-17-workshop-criteo-ad-placement-challenge

A. Advertisement Placement – 2

I. Introduction and Related Work

The goal is to minimize the negative influence of ads on user experience while maximizing ad revenue. The advertising system must make three decisions: (A) whether to interpolate an ad in the recommendation list (rec-list), (B) choosing the optimal ad, and (C) placing it in the optimal location.

Classical models fail to simultaneously perform these three sub-actions. Hence, the DEep reinforcement learning framework (DEAR) with a novel DQN architecture for online advertising in recommender systems was designed to address this challenge efficiently.

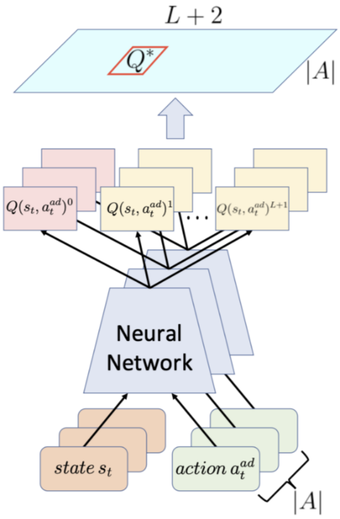

Fig. 1. The Novel DQN architecture for online advertising.

II. Dataset

The dataset originates from a short video site, Douyin. It includes two types of videos: normal (recommended items) and ad videos. Normal video features include id, like score, finish score, comment score, follow score, and group score, while ad videos include id, image size, bid-price, hidden-cost, predicted-CTR, and predicted-recall. We split the dataset into 70% training/validation and 30% testing.

III. Related Work

Recent work in RL-based online advertising and recommender systems can be broadly divided into two categories. For online advertising, Salomatin, Liu, and Yang (2012) focus on guaranteed delivery models, while Yang and Lu (2016), Nuara et al. (2018), Gasparini et al. (2018), Tang et al. (2013), Xu, Qin, and Liu (2013), Yuan, Wang, and van der Meer (2013), and Schwartz, Bradlow, and Fader (2017) address real-time bidding using multi-armed bandit and MDP formulations. In addition, Wu et al. (2018a) propose a multi-agent RL approach for cooperative publisher policies, with further MDP-based methods offered by Cai et al. (2017), Wang et al. (2018a), Rohde et al. (2018), Wu et al. (2018b), and Jin et al. (2018). On the recommender systems side, Zhao et al. (2019b) and Zhang et al. (2020) develop frameworks that jointly leverage positive (click/purchase) and negative (skip) feedback, while Zhao et al. (2017, 2018a) introduce a page-wise recommendation model, Zhao et al. (2020a) propose the multi-agent DeepChain framework, and Zhao et al. (2019a) create a GAN-based user simulator, with additional contributions from Fan et al. (2020), Zou et al. (2020), and others.

IV. Problem Statement

This project addresses the challenge of ad interpolation in recommendation lists, modeling it as a Markov Decision Process (MDP) in which an Advertising-Agent (AA) interacts with environment ϵ (or users) by sequentially interpolating ads into a sequence of rec-lists over time, so as to maximize the cumulative reward from the environment. The MDP consists of:

- **State (S)**: User's browsing history, recommendation list, and contextual information.
- **Action (A)**: Deciding three internally related tasks, whether to insert an ad, which ad to place, and its optimal position in the list.
- **Reward (R)**: Balancing ad revenue and user experience based on user feedback.
- **Transition Probability (P)**: Defining how states evolve after each action.
- **Discount Factor** (γ): Weighing immediate vs. future rewards.

The goal is to develop an optimal advertising policy that maximizes ad revenue while minimizing the negative impact on user experience.

Given the historical Markov Decision Process (MDP), i.e., (S, A, P, R, γ), the goal is to find an advertising policy $\pi : S \to A$, which can maximize the cumulative reward from users, i.e., maximizing the income of ads and minimizing the negative influence on user experience.

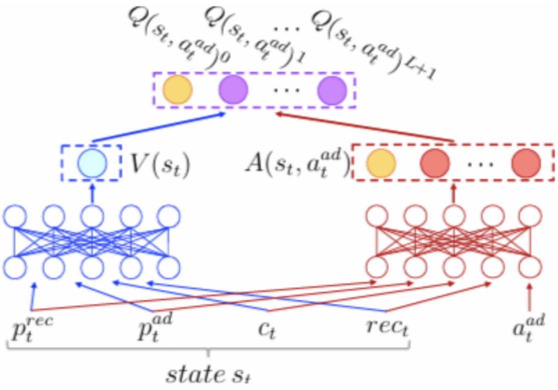

Fig. 2. TIllustration of the reinforcement learning-based ad selection framework.

V. Evaluation

Our evaluation aims to answer three key questions:

- **Performance Comparison**: How does the proposed architecture compare to representative baselines?
- **Component Contribution**: How do individual components impact performance?
- **Hyperparameter Sensitivity**: How do different hyperparameter settings affect effectiveness?

To assess performance, we use accumulated rewards as the primary metric. Our model is compared against established baselines such as W&D (Cheng et al., 2016), DFM (Guo et al., 2017), and GRU (Hidasi et al., 2015).

To analyze component contributions, we evaluate modified versions of the model, systematically removing specific components and measuring the resulting impact. Additionally, hyperparameter tuning is conducted to identify the optimal parameter configurations that maximize effectiveness.

Our objective is to:

- **(i)** Jointly determine whether to interpolate an ad in a recommendation list, and if so, select the optimal ad and placement.
- **(ii)** Balance ad revenue maximization with minimizing negative effects on user experience.

References

[1] Zhao, X., Gu, C., Zhang, H., Yang, X., Liu, X., Tang, J., & Liu, H. (2021). DEAR: Deep reinforcement learning for online advertising impression in recommender systems. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*.

[2] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10. ACM.

[3] Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1725–1731.

[4] Li, W.-J. (2019). SVD: A Large-Scale Short Video Dataset for Near-Duplicate Video Retrieval. *Proceedings of International Conference on Computer Vision*.

[5] Bellman, R. (2013). *Dynamic programming*. Courier Corporation.

[6] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI Gym. *arXiv preprint arXiv:1606.01540*.

[7] Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y., & Guo, D. (2017). Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 661–670. ACM.

[8] Chen, H., Dai, X., Cai, H., Zhang, W., Wang, X., Tang, R., Zhang, Y., & Yu, Y. (2018a). Large-scale interactive recommendation with tree-structured policy gradient. *arXiv preprint arXiv:1811.05869*.

[9] Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., & Chi, E. (2018b). Top-k off-policy correction for a REINFORCE recommender system. *arXiv preprint arXiv:1812.02353*.