IMPERIAL

Department of Mathematics

Deep Reinforcement Learning for Ad Personalization

Martin Batěk

CID: 00951537

Supervised by Mikko Pakkanen

1 May 2023

Submitted in partial fulfilment of the requirements for the
MSc in Machine Learning and Data Science of
Imperial College London

The work contained in this thesis is my own work unless otherwise stated.

Signed: Martin Batěk   Date: 17 July 2024

Abstract

ABSTRACT GOES HERE

Acknowledgements

ANY ACKNOWLEDGEMENTS GO HERE

Contents

1 Introduction

The global digital advertising market is worth approximately $602 billion today. Due to the increasing rate of of online participation since the COVID-19 pandemic, this number has been rapidly increasing and is expected to reach $871 billion by the end of 2027 (eMarketer, 2023). Many of the of the major Ad platforms such as Google, Facebook and Amazon operate on a cost-per-user-engagement pricing model, which usually means that advertisers get charged for every time a user clicks on an advertisment. This means that these platforms are incentivized to make sure that the content shown to each user is as relevent as possible in order to maximize the number of clicks in the long term. Attaining accurate Click-Through Rate (CTR) prediction is a necessary first step for Ad persionalization, which is why study of CTR prediction methods have been an extremely active part of Machine Learning research over the past through years.

Initially, shallow prediction methods such as Logistic Regression, Factorization Machines (Rendle, 2010) and Field-Aware Factorization Machines (Juan et al., 2016) have been used for CTR prediction. However, these methods have often been shown to be unable to capture the higher order feature interactions in the sparse multi-value categorical Ad Marketplace datasets (Zhang et al., 2021). Since then, Deep Learning methods have been shown to show superior predictive ability on these datasets. The focus of my reasearch project is therefore to explore the merits of different Deep Learning architechtures for click-through rate prediction.

A number of Deep Learning models have been proposed for CTR prediction, some of which will be explored in this report. Each of these models outperform their shallow counterparts in terms of predictive ability. In a static environment, these models are able to serve the CTR prediction function of Ad personalization, but in a dynamic environment, the model must be able to adapt to the changing user preferences. This is where Reinforcement Learning comes in. Reinforcement Learning is a type of Machine Learning that is used to make a sequence of decisions in an environment in order to maximize some notion of cumulative reward. In the context of Ad personalization, the environment is composed of the user, the Ad platform and the advertisments, whereas the reward is the users' engagement with the advertisments and with the Ad platform.

In this report, I aim to construct a Deep Reinforcement Learning model for Ad personalization that is able to adapt to the changeing user preferences and advertisment characteristics available on the platform. In chapter 2, I begin by providing a background to the problem of Click-Through Rate prediction in the context of Ad personalization, and explore the unique challenges posed by the typically sparse multi-value categorical datasets that are common in the Ad marketplace. I then proceed to review the literature on Deep Learning models for CTR prediction, highlighting the different techniques that each framework uses to capture the key feature interactions in the data. I also review the literature on Deep Reinforcement Learning and its applications across different domains. In chapter 3, I evaluate the performance of different Deep Learning models for CTR prediction on three well-known benchmark datasets, Criteo (Tien et al., 2014), KDD12 (Aden, 2012) and Avazu (Wang and Cukierski, 2014). In chapter 4, I construct a Deep Reinforcement Learning model for Ad personalization and evaluate its performance on the same benchmark datasets. Finally, in chapter 5, I discuss the results of the experiments and provide some concluding remarks.

2 Background

2.1 Deep CTR Prediction

In their respective surveys on the use of Deep Learning methods for CTR prediction, Gu (2021) and Zhang et al. (2021) outline the problem of CTR prediction as one that essentially boils down to a binary (click/no-click) classification problem utilizing user/ad-view event level online session records. The goal of CTR prediction is to predict the probability of a user clicking on an advertisment given the information available about the user, advertisment and the context in which the advertisement is shown. Suppose that $\mathbf{x} \in \mathbb{R}^n$ is a vector of features that describes the user, ad and platform for a given instance, and $y \in \{0, 1\}$ is the binary label indicating whether the user clicked on the ad or not. The goal of CTR prediction is to learn a function $f : \mathbb{R}^n \to (0, 1)$ such that:

$$f(\mathbf{x}) = \mathbb{P}(y = 1|\mathbf{x}) = \mathbb{P}(\text{click}|\mathbf{x})$$

In other words, for a given set of features $\mathbf{x}$, the model should output the probability that the user will click on the ad.

2.2 Deep Reinforcement Learning

3 Deep CTR model Evaluation

3.1 Model Selection Methodology

3.2 Model Summaries

3.2.1 Shallow Models

Logistic Regression

Factorization Machines

Factorization Machines were first introduced in (Rendle, 2010) as a model class that "combines the advantages of Support Vector Machines (SVM) with factorization models". The model is able to capture the second order feature interactions in the data, which is a key advantage over Logistic Regression. The model is defined as follows:

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \tag{3.1}$$

where w_0 is the bias term, w_i are the weights for the i-th feature, $\mathbf{v}_i$ are the latent vectors for the i-th feature. Rendle (2010) shows that the learned biases and weights of the FM model can be computed in linear time, "and can be learned efficiently by gradient descent methods", such as Stochastic Gradient Descent (SGD).

3.2.2 Deep Models

Factirization Supported Neural Networks

The first Deep Learning model that we will consider is the Factorization Supported Neural Network (FNN) model proposed by Zhang et al. (2016). The model works by first training a Factorization Machine model on the sparse-encoded categorical input features. It then uses the latent vectors learned by the FM model (see $\mathbf{v}_i$ in equation 3.1) as inputs to a Neural Network, as shown in Figure 3.1. In doing so, the FNN model is effectively using the FM latent factors to initialize the embedding layer of the Neural Network. The DNN is then able to learn the higher order feature interactions in the data, which the FM model is unable to capture.

Product Based Neural Networks

The Product Based Neural Network (PNN) model proposed by Qu et al. (2016) is another Deep Learning model that was developed around the same time as the FNN model. The

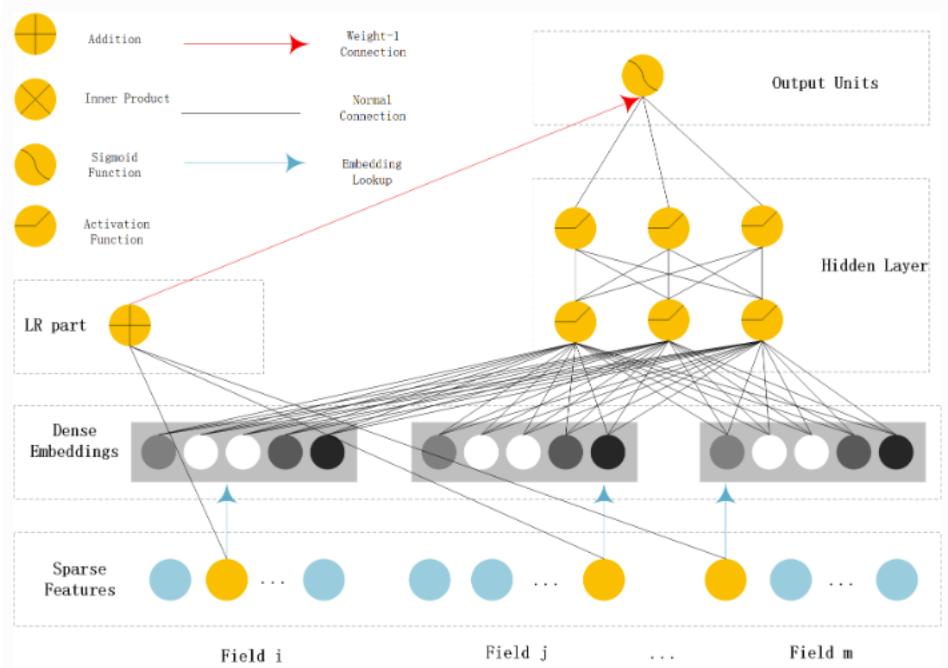

Figure 3.1: Factorization Supported Neural Network as proposed by Zhang et al. (2016). Image taken from Shen (2017)

key innovation of the PNN moel is the use of a pair-wisely connected Product Layer after a field-wise connected embetting layer for the categorical features, as shown in Figure 3.2. The Product Layer is able to directly model inter-field feature interaction by means of either an inner product or outer production operation, and then further distill higher feature inturactions by passing the output of the Product Layer through fully connected MLP layers.

Wide & Deep Learning

The Wide & Deep Learning (WDL) model proposed by Cheng et al. (2016) introduces the concept of dual-tower model architecture (Zhang et al., 2021). While both the FNN and the PNN models generally tend to be constructed as a single fully connected DNN model, the Wide & Deep model consists of a wide component, consisting of a three layer Deep Neural Network that takes the concatinated embedding vectors of the categorical features as input, and a deep component, consisting of a cross product transformation of selected sparse categorical features. The logits from the wide and deep components are added together to produce the final prediction. The architecture of the WDL model is shown in Figure 3.3.

The purpose behind the Dual-Tower architecture is to counteract the tendancy of the fully connected single tower DNN models to lose the ability to capture low-order feature interactions (Zhang et al., 2021). The Wide component is able to capture the

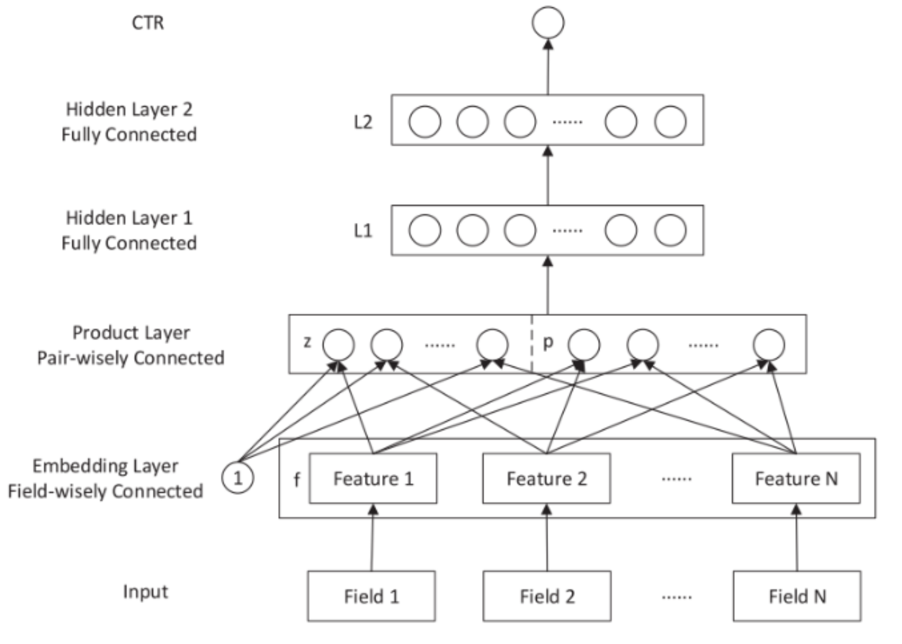

Figure 3.2: Product Based Neural Network as proposed by Qu et al. (2016). Image taken from Shen (2017)

low-order feature interactions, while the Deep component is able to capture the higher order feature interactions.

DeepFM

Feature Generation by Convolutional Neural Network

Automatic Feature Interaction Learning

3.3 Benchmark Datasets and Exploratory Data Analysis

3.4 Model Evaluation

3.5 Deep CTR Model Results

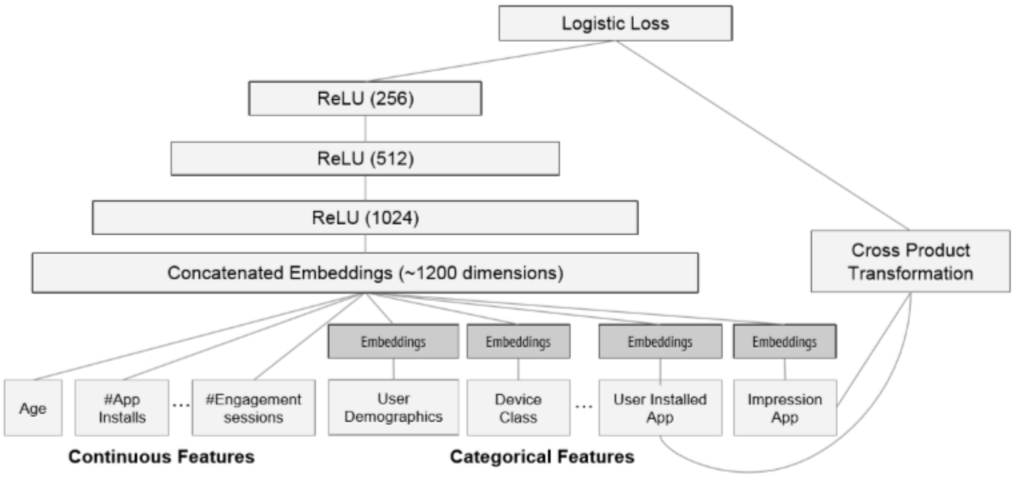

Figure 3.3: Wide & Deep Learning model as proposed by Cheng et al. (2016)

4 Deep Reinforcement Learning for Ad Personalization

4.1 DeepCTR-RL Framework

4.2 Experiment Setup

4.3 Results

5 Discussion

Discussion goes here.

6 Conclusion

Conclusion goes here.

Bibliography

Yi Wang Aden. Kdd cup 2012, track 2, 2012. URL https://kaggle.com/competitions/kddcup2012-track2.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, page 7–10, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450347952. doi: 10.1145/2988450.2988454. URL https://doi.org/10.1145/2988450.2988454.

eMarketer. Digital advertising spending worldwide from 2021 to 2027 (in billion u.s. dollars). Technical report, Statista Inc., 2023. URL https://www-statista-com.iclibezp1.cc.ic.ac.uk/statistics/237974/online-advertising-spe

Liqiong Gu. Ad click-through rate prediction: A survey. In Christian S. Jensen, Ee-Peng Lim, De-Nian Yang, Chia-Hui Chang, Jianliang Xu, Wen-Chih Peng, Jen-Wei Huang, and Chih-Ya Shen, editors, *Database Systems for Advanced Applications. DASFAA 2021 International Workshops*, pages 140–153, Cham, 2021. Springer International Publishing. ISBN 978-3-030-73216-5.

Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. Field-aware factorization machines for ctr prediction. In *10th ACM Conference on Recommender Systems*, RecSys '16, page 43–50, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340359. doi: 10.1145/2959100.2959134. URL https://doi.org/10.1145/2959100.2959134.

Yanru Qu, Han Chai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1149–1154. IEEE, 2016. ISBN 2374-8486. doi: 10.1109/ICDM.2016.0151. ID: 1.

Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000, 2010. ISBN 1550-4786. doi: 10.1109/ICDM.2010.127. ID: 1.

Weichen Shen. Deepctr: Easy-to-use,modular and extendible package of deep-learning based ctr models, 2017. URL https://github.com/shenweichen/deepctr.

Jean-Baptiste Tien, joycenv, and Olivier Chapelle. Display advertising challenge, 2014. URL https://kaggle.com/competitions/criteo-display-ad-challenge.

Steve Wang and Will Cukierski. Click-through rate prediction, 2014. URL `https://kaggle.com/competitions/avazu-ctr-prediction`.

Weinan Zhang, Tianming Du, and Jun Wang. Deep learning over multi-field categorical data: A case study on user response prediction, 2016. URL `https://arxiv.org/abs/1601.02376`. 1601.02376.

Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. Deep learning for click-through rate estimation, 21 Apr 2021. URL `https://arxiv.org/abs/2104.10584`.