IMPERIAL

Department of Mathematics

Deep Reinforcement Learning for Ad Personalization

Martin Batěk

CID: 00951537

Supervised by Mikko Pakkanen

2 September 2024

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 there is a significant commercial incentive to design Ad-serving platforms that ensure that the content shown to each user is as relevent as possible, so as to maximize user engagement and platform revenues as much as possible.

Attaining accurate Click-Through Rate (CTR) prediction is a necessary first step for Ad personalization, 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. A number of Deep Learning models have been proposed, each using a different techniques for feature interaction modelling, ranging from Deep Learning extensions of Factorization Machines such as DeepFM (Guo et al., 2017), to novel methods such as AutoInt (Song et al., 2019). By employing a multi-towered neural network architecture, these models are able to capture both low-order and high-order feature interactions in the data, and therefore tend to achieve supperior predictive performance to their shallow counterparts.

However, irrespective of how well these models perform in a static environments, the reality is that user preferences and advertisment characteristics are constantly changing. Like most online reccomender systems, Ad personalization models must be able to adapt to these changes in order to continue to provide accurate predictions over the longer period (Zheng et al., 2018). This problem necessitates the use of Reinforcement Learning for Ad personalization.

Reinforcement Learning is a subdomain of Machine Learning in which the goal is for an agent to learn an optimal policy that maximizes the expected reward in an environment where the state-action-reward progression can be modelled as a Markov Decision Process (Puterman, 2014). Early Reinforcement Learning methods involved deriving a the transition probabilities for the state-action pairs on the basis of interactions with the environment and then using Dynamic Programming methods such as the Upper Confidence Bound RL (UCBRL) algorithm (Auer et al., 2008) and the the Thompson

Sampling algorithm for Reinforcement Learning (Pike-Burke, 2024). However, in cases where the state-action space is too sparce to be reasonably enumerated, it is often more practical to user a function approximator to directly estimate the expacted cumalative reward for each action in each state. This method of Reinfocement Learning is commonly referred to as Q-learning (Watkins, 1989), and has the advantage of being *model-free*, meaning that it does not require the agent to have a model of the environment thereby making it more scalable to large and sparce datasets.

Reasearch Question and Contributions

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.

Structure of the Report

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. A defining characteristic for this type of data is that many of the features are multi-value categories with a high degree of of cardinality (He and Chua, 2017). This in turn means that the ad marketplace datasets used for CTR predictions can be extremely sparse, which increases the difficulty of the classification problem at hand (Gu, 2021).

A key requirement for CTR modelling is therefore working out which of the many sparse features and feature interactions (combinations of two or more features) are significant for determining the correct prediction (Gu, 2021). Factorization Machines (Rendle, 2010) and Field-aware Factorization Machines (Juan et al., 2016) were popularized shallow modelling methods that explicitly account for first order interactions between features. However, these techniques do not capture higher order interactions (combinations of three or more features) and have thus been known to perform poorly in scenarios with highly sparse data (Zhang et al., 2021). Since 2015 the research cummunity has been increasingly turning to Deep Learning techniques to enhance prior CTR prediction techniques (such as in the case of DeepFM (Guo et al., 2017)), as well as to develop novel approaches. Neural based network models excel at simulataneoulsy extracting high-order and low-order feature interations virtue to the use of pooling layers, multiple hidden layers and activation units Gu (2021). Due the aforementioned importance of feature interation modelling, a number of Feature Interaction Operator layers have been developed to explicitly capture the key combinations of features. These layers are then typically incorporated with a supplimentary Deep Neural Network in a single or dual tower architecture, as shown in Figure 2.1.

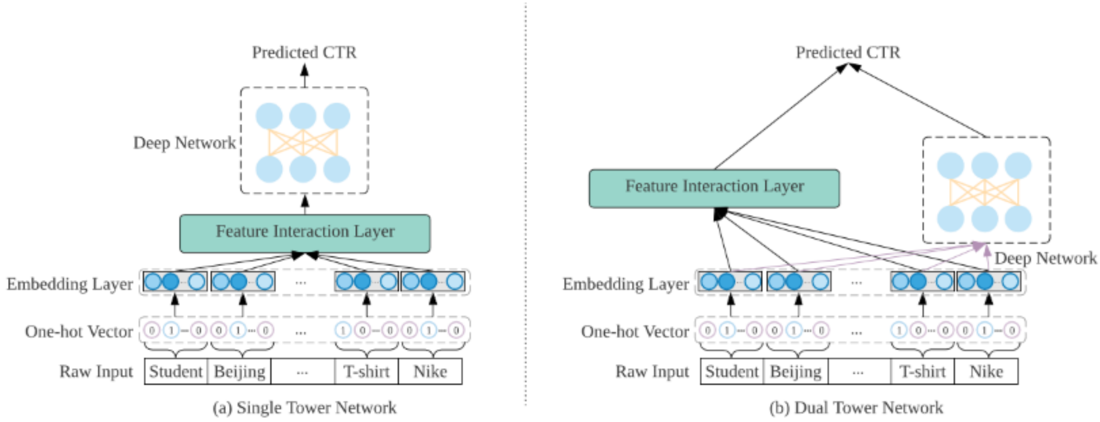

Figure 2.1: Deep Neural Network Architecture for CTR prediction. Image taken from Zhang et al. (2021)

Feature Interaction Operators can be categorized as either Product Operators, Convolutional Operators or Attention Operators. Product Operators such as the Product-based Neural Network (PNN) (Qu et al., 2016), Neural Factorization Machines (NFM) (He and Chua, 2017), Deep and Cross Network (Wang et al., 2017) and Gated Deep Cross Network (GDCN) (Wang et al., 2023) introduce a product layer between the categorical feature embedding layer and the rest of the neural network in order to explicitly model the important feature interactions. Convolutional Operators such as the Convolutional Click Prediction Model (CCPM) (Liu et al., 2015) utilized convolution, pooling and non-linear activation in order to calculate arbitrary-order interactions. Finally, Attention Operators such as Attentional Factorization Machines (AFM) (Xiao et al., 2017), AutoInt Song et al. (2019) and Interpretable CTR prediction model with Hierarchical Attention (InterHAt) (Li et al., 2020) utilize the attention mechanism to enable different feature interactions to contribute differently to the prediction.

2.2 Deep Reinforcement Learning

In their survey, (Wang et al., 2024) describe how deep reinforcement learning combines the aforementioned feature extraction capabilities of DNN's with the decision-making capability of reinforcement learning, which aims to learn an optimal state-action policy which maximizes the expected reward gained in a given environment. In the context of recommendation systems, a significant amount of research has been dedicated to formulating the recommendation problem as a Contextual Multi-Armed Bandit (MAB) problem setting, where the context consists of user, site and item features (Bouneffouf et al., 2012; Li et al., 2010; Zeng et al., 2016). However, a shortcoming for the MAB approach is that it does not explicitly model the future expected reward for the policy, which may be detrimental in the longer term (Zheng et al., 2018). Markov Decision Process (MDP) models solve for this issue by modelling the state-action progression as a Markov Process, allowing for the stochastic valuation of the future potential rewards for a given recommendation policy (Lu and Yang, 2016; Mahmood and Ricci, 2007). DRN (Zheng et al., 2018) is a MDP framework that leverages a Deep Neural Network to approximate the expected total user response for each recommendation at each state. The two major advantages of DRN are firstly that it is composed on the basis of a continuous state and action representation, meaning that it can be scaled to large and sparse datasets, and secondly that the proposed reward function consists of both the immediate reward (user click) as well as the future expected reward (long term user engagement), thereby allowing for better recommendations over a user's lifetime.

3 Deep CTR model Evaluation

3.1 Model Selection Methodology

As explained above, I will explore a number of deep learning models. I selected five popular models on the basis of the following criteria

- Competitive predition accuracy in the KDD12, Criteo and Avazu datasets as published on Papers with Code.
- Ideally, I was looking for a representitive set of models for each model type as discussed in (Zhang et. al. 2021). Therefore I was looking for models that employed Product Interaction Opetators, Attention Operators and Factorization Machines as a basis.
- The code for the model has to be accessible and intuitive to use.

On the basis of the above critea, I have chosen the following models to explore:

- Factorization Supported Neural Networks
- Product Based Neural Networks
- Wide and Deep
- DeepFM
- Automatic Feature Interaction (AutoInt)

In the section below, I briefly introduce each of the models, and evaluate against the benchmark datasets loaded and preprocessed above.

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.

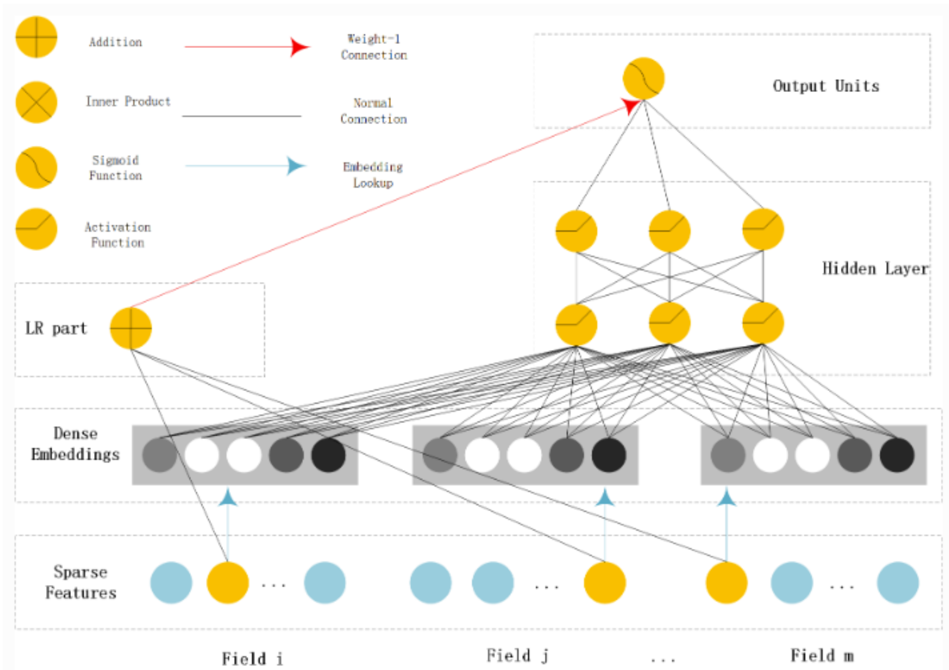

Figure 3.1: Factorization Supported Neural Network as proposed by Zhang et al. (2016). Image taken from Shen (2017)

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 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.

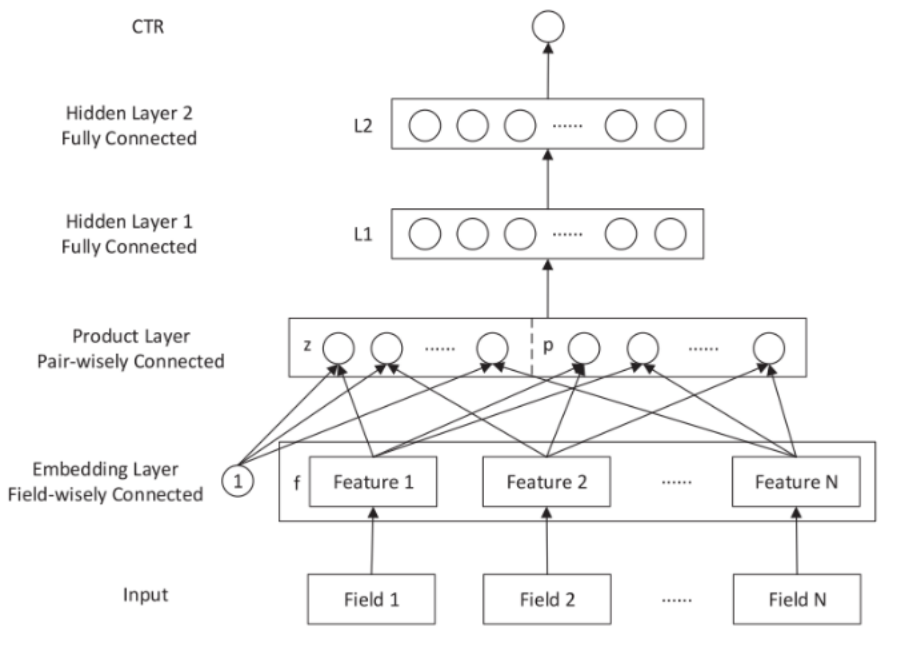

Figure 3.2: Product Based Neural Network as proposed by Qu et al. (2016). Image taken from Shen (2017)

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.

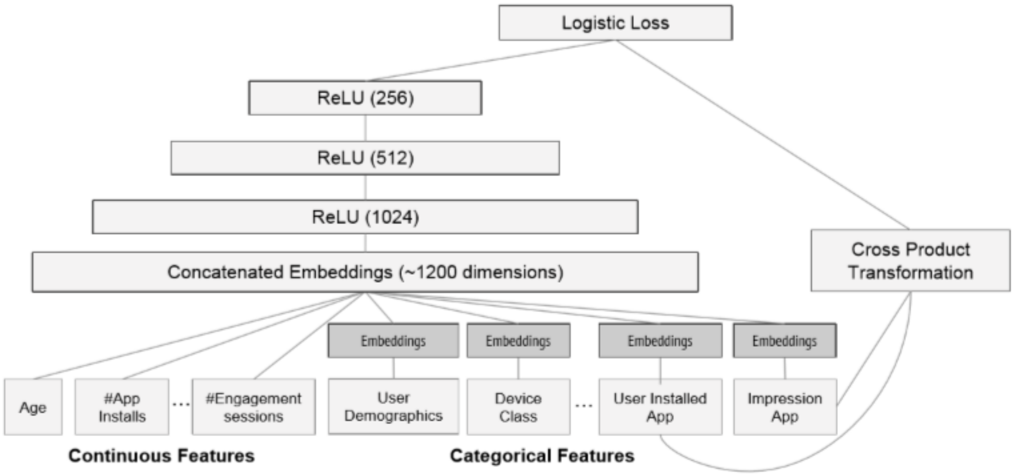

Figure 3.3: Wide & Deep Learning model as proposed by Cheng et al. (2016)

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

DeepFM

The DeepFM model proposed by Guo et al. (2017) can be thought of as an imporvement of the aforementioned FNN (Zhang et al., 2016) and WDL (Cheng et al., 2016) models. Like the FNN model, the DeepFM model usilises the Factorization Machine model (Rendle, 2010) to learn lower-order feature interactions. However, it also employs a dual-tower architecture like the WDL model, with the Wide component being the FM model and the Deep component being a fully connected DNN model. The DeepFM model is therefore able to avoid the limitations on capturing low-order interactions that are inherent in the FNN model. In addition, due the the application of the FM to all feature embeddings, the DeepFM model eliminates the need to choose which features to feed through the wide component, as is the case in the WDL model. The architecture of the DeepFM model is shown in Figure 3.4.

Automatic Feature Interaction Learning

The Autotomatic Feature Interaction Learning (AutoInt) model proposed by Song et al. (2019) makes use of a multi-head self attention network to model the important feature interactions in the data. The initial paper separates the model into three parts: an

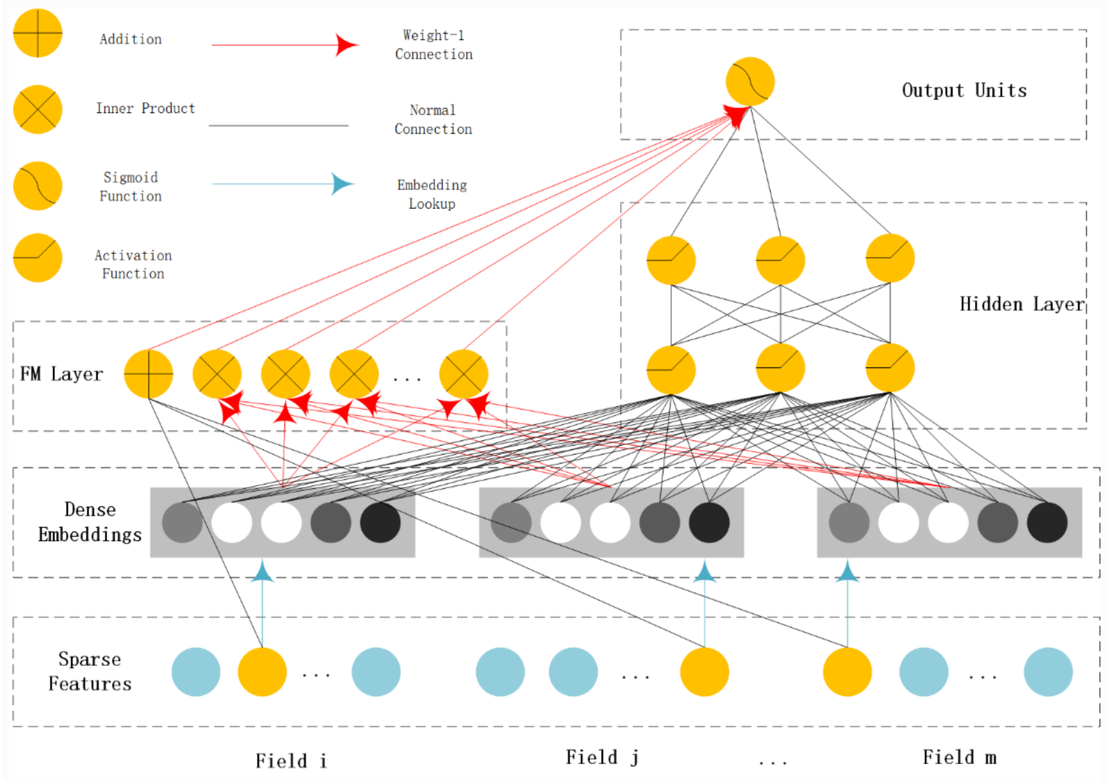

Figure 3.4: DeepFM model as proposed by Guo et al. (2017). Image taken from Shen (2017)

embedding layer, an interaction layer and an output layer. The embedding layer aims to project each sparse multi-value categorical a and dense numerical feature into a lower dimensional space, as per the equation 3.2:

$$\mathbf{e_i} = \frac{1}{q}\mathbf{V_i}\mathbf{x_i} \tag{3.2}$$

where $\mathbf{V_i}$ is the embedding matrix for the i-th field, x_i is a multi-hot vector, and q is the number of non-zero values in x_i. The interaction layer employs the multi-head mechanism to determine which higher order feature interaction are meaningful in the data. This not only improves the efficiency of model traning, but it also improves the model's explainability. Lastly, the output layer is a fully connected layer that takes in the concatinated output of the interaction layer, and applies the sigmoid activation function to produce the final prediction. The architecture of the AutoInt model is shown in Figure 3.5.

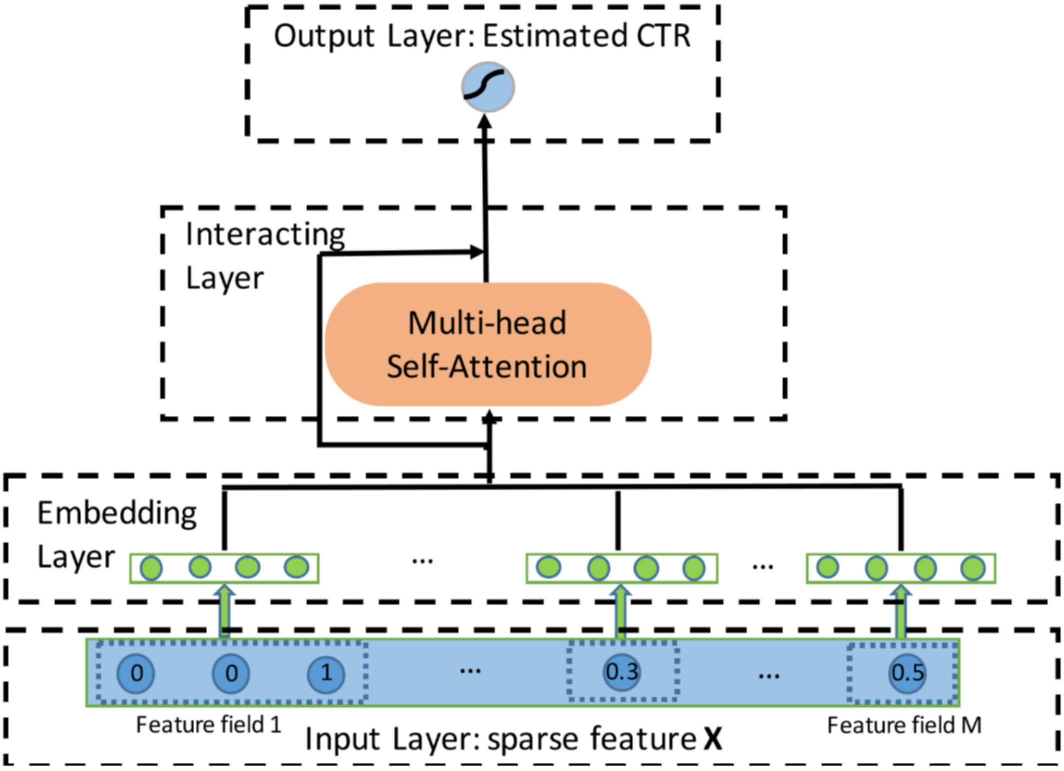

Figure 3.5: AutoInt model as proposed by Song et al. (2019)

3.3 Benchmark Datasets and Exploratory Data Analysis

3.4 Model Evaluation

3.5 Deep CTR Model Results

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.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2008.

Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gancarski. A contextual-bandit algorithm for mobile context-aware recommender system. In Tingwen Huang, Zhigang Zeng, Chu Li, ong, and Chi Sing Leung, editors, *Neural Information Processing*, pages 324–331, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-34487-9.

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.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. *CoRR*, abs/1703.04247, 2017. URL http://arxiv.org/abs/1703.04247. 1703.04247.

Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics, -08-16 2017.

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`.

Li, Cheng, Chen, Chen, and Wang. Interpretable click-through rate prediction through hierarchical attention, -01-20 2020.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 661–670, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605587998. doi: 10.1145/1772690.1772758. URL `https://doi.org/10.1145/1772690.1772758`.

Liu, Yu, Wu, and Wang. A convolutional click prediction model, -10-17 2015.

Zhongqi Lu and Qiang Yang. Partially observable markov decision process for recommender systems. *CoRR*, abs/1608.07793, 2016. URL `http://arxiv.org/abs/1608.07793`. 1608.07793.

Tariq Mahmood and Francesco Ricci. Learning and adaptivity in interactive recommender systems. In *Proceedings of the Ninth International Conference on Electronic Commerce*, ICEC '07, page 75–84, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937001. doi: 10.1145/1282100.1282114. URL `https://doi.org/10.1145/1282100.1282114`.

Ciara Pike-Burke. Optimism/thompson sampling, 2024.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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`.

Song, Shi, Xiao, Duan, Xu, Zhang, and Tang. Autoint, -11-03 2019.

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

Fangye Wang, Hansu Gu, Dongsheng Li, Tun Lu, Peng Zhang, and Ning Gu. Towards deeper, lighter and interpretable cross network for ctr prediction. ACM, 2023. doi: 10.1145/3583780.3615089.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, ADKDD'17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450351942. doi: 10.1145/3124749.3124754. URL `https://doi.org/10.1145/3124749.3124754`.

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

Xu Wang, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5064–5078, 2024. doi: 10.1109/TNNLS.2022.3207346.

Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, 1989.

Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks *, 2017. URL `https://arxiv.org/abs/1708.04617`.

Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 2025–2034, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939878. URL `https://doi.org/10.1145/2939672.2939878`.

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`.

Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *2018 World Wide Web Conference*, pages 167–176, Lyon, France, 2018. International World Wide Web Conferences Steering Committee. doi: 10.1145/3178876.3185994. URL `https://doi.org/10.1145/3178876.3185994`.