Reinforcement Learning for Neural Collaborative Filtering

2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) | 978-1-6654-5818-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICAIIC54071.2022.9722673

Alexandros I. Metsai
My Company Projects O.E.
Thessaloniki, Greece
alexandros.metsai@mycompany.com.gr

Konstantinos Karamitsios
My Company Projects O.E.
Thessaloniki, Greece
kk@mycompany.com.gr

Konstantinos Kotrotsios
My Company Projects O.E.
Thessaloniki, Greece
kotrotsios@mycompany.com.gr

Periklis Chatzimisios
International Hellenic University
Thessaloniki, Greece
pchatzimisios@ihu.gr

George Stalidis
International Hellenic University
Thessaloniki, Greece
stalidgi@ihu.gr

Kostas Goulianas
International Hellenic University
Thessaloniki, Greece
gouliana@ihu.gr

***Abstract*—Artificial Intelligence (AI) has become an integral part of many modern technologies, with significant advances in real-world applications. With the rise of deep learning methods in the past decade, systems that utilize artificial neural networks have produced remarkable results in a variety of fields, such as computer vision, natural language processing and voice recognition, with performance exceeding that of humans in many cases. Examples of practical applications include self-driving cars, state-of-the-art text translators and generators, and robust object detection algorithms. The field of Recommender Systems has also taken advantage of this progress, with a plethora of novel neural networks being proposed that achieve significant improvements in providing automatic recommendations regarding the preferences of users. Aiming to further explore this area and the capabilities of different deep neural networks, we train top-performing neural collaborative filtering recommender systems under a reinforcement learning setting, which has been largely unexplored in favor of supervised learning for these models. Experimental evaluation on the MovieLens-1m dataset showcases the behavior of different neural architectures under this setting, and how the introduction of sophisticated components contributes to improved performance.**

Keywords—artificial intelligence, deep learning, reinforcement learning, recommender systems

I. Introduction

Over the last decade, Artificial Intelligence (AI) has witnessed a great rise in popularity and adoption, with various modern architectures being characterized as state-of-the-art and top-performing solutions in many different domains. Specifically, a plethora of systems based on deep neural networks (or "deep learning" in the relevant literature) have been applied outside of basic research, in real-world industrial settings, with outstanding results [20]. Examples include self-driving cars, significant improvements in computer vision related fields and major advances in natural language processing. A notable field that has made extensive use of these advances in AI is the field of Recommender Systems [16]. These systems concern the automatic generation of recommendations that satisfy a user's preferences by utilizing various factors, such as their item selection history, ratings, demographic information, seasonal conditions, etc. [2].

A major category of algorithms for implementing Recommender Systems are Collaborative Filtering techniques. In essence, these are methods for performing predictions regarding the preferences of a single user by taking into account the information regarding the preferences of many users [20]. These methods usually require large amounts of data; however, this restriction is usually satisfied for this particular domain, since e-commerce and related online services have access to an abundance of user-item interaction data. As an example, the volume of interactions that platforms such as Spotify and Netflix record is enormous, with the same being true for online retail platforms such as Amazon and eBay. Influenced by the rise of deep learning described above, many implementations of collaborative filtering methods that utilize neural networks were introduced in recent years, with promising results [4, 6, 7, 8].

Regarding the training or learning process of a neural network (and of a machine learning model in general), the three basic learning paradigms are supervised learning, unsupervised learning, and reinforcement learning [5]. Most of the algorithms in the collaborative filtering for recommender systems domain utilize supervised learning, with the unsupervised learning setting being less practical in this scenario, while the reinforcement learning setting has not been extensively explored. Motivated by this, we aim to evaluate the performance of modern neural collaborative filtering algorithms under a reinforcement learning setting.

II. Related Work

In both research and industry settings, many Recommender Systems have been proposed, with the first attempts dating to the late 1970s [19]. The first truly mature methods began to emerge in the mid-1990s, with the systems GroupLens [18], Video Recommender [9] and Ringo [21] providing remarkable solutions in automatic recommendations. The GroupLens system started as a recommender for relevant articles in the Usenet [11] platform, by taking into consideration a user's previous ratings. Similarly, the Video Recommender system was tasked with selecting the most relevant videos from a larger set. Finally, the Ringo system's goal was the personalized recommendations of relevant music artists and records.

The above contributions led to the first commercial recommender systems by the end of the decade. One of the first and most significant examples is the recommender system integrated in Amazon's platform [14]. This platform provided recommendations in the form of lists "also seen by other users". In the following years, even more commercial platforms followed this example, and by the middle of the next decade the field of recommender systems had become an area

This work has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T2EDK-03843).

highly active in both research and adoption by the industry. The Netflix Prize [1], organized in 2006 and offering 1 million dollars to the best collaborative filtering algorithm, is another example of this high activity, which continued in the following years.

The field of recommender systems has utilized a broad spectrum of machine learning techniques for the implementation of relevant algorithms. These include classification techniques and clustering, as well as methods for dimensionality reduction. Algorithms for matrix factorization [12] and factorization machines [17], which improve upon the former, were some of the most prominent solutions that first gained widespread attention after their exploitation during the Netflix prize.

During the past decade, AI algorithms based on deep neural networks have witnessed a great surge in popularity and adoption, with many applications and improvements in different domains. Influenced by these advances, many works in recommender systems adopted deep learning techniques for implementing collaborative filtering methods and factorization machines. In 2016, the authors of [4] introduced an influential neural network called "Wide and Deep". As the title suggests, this model utilizes a linear model (wide component) combined with a deep network for modeling both high and lower-level relations. The deep component is constituted by three fully connected layers with ReLU activations, while the wide component is described by a cross-product transformation. The authors reported that this architecture was successfully put in production and evaluated on Google Play, a commercial application distribution service with more than a billion active users.

Motivated by the idea of combining a shallow and a deep component with the aim of modeling high and lower-level user and item relationships, similar works that improved upon the "Wide and Deep" network architecture emerged. In 2017, the authors of [6] proposed a variation of the model, that allows for a larger degree of flexibility and efficiency regarding the overall architecture. These improvements include the replacement of the wide model with a Factorization Machine and the introduction of an embedding vector for representing the input, that constitutes the shared input to both of the wide and deep components. Shortly after this work, in 2018, a model that further improves upon the latter was introduced [13]. This architecture, called xDeepFM, utilized a liner model combined with a Compressed Interaction Network (CIN) and a deep neural network. A different approach was taken by the authors of [7] that attempted to construct an improved factorization machine model with a neural network architecture, called Neural Factorization Machine.

III. Proposed Approach

Aiming to investigate the capabilities of different deep learning networks for utilizing recommender systems, we attempt to train the most prominent networks under a reinforcement learning setting. Though supervised learning has been the norm for training these architectures, reinforcement learning can be more suitable for modeling real-world applications, by utilizing the network as an agent that is tasked with performing actions in an environment defined by the user and the possible items that can be recommended.

A. Factorization Machines

Matrix Factorization techniques have been an important contribution to the field of recommender systems and have gained a significant adoption [10]. However, due to the way they function, which is by deconstructing a user-item interaction matrix to two matrices of lower dimensionality, the product of which approximates the original, they are limited to modeling lower-order relationships. Since these relationships may be better described by considering higher-order relationships too, there arises the need for a more descriptive modelling method.

Factorization Machines, first described in 2010 [17], have been widely adopted by the industry and influenced the relevant research. These models map any input features to vectors of lower dimensions and are able to estimate parameters using very sparse data, thus being able to scale into larger datasets. Equation (1) describes the output of a Factorization Machine:

$$\hat{y}_{FM}(x) = w_0 + \sum_{i=1}^{n} x_i w_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j \qquad (1)$$

Where the input vector is $x \in R^n$, with the value $x_i = 0$ signifying that the ith feature is not present in the current input, and $w_0, w_i, < v_i, v_j >$ are values that must be configured during the training/fitting processes. The first term represents the global bias, the second term represents the weights of the ith variable, and the third term represents the dot product of the ith and jth elements of the user-item interaction matrix.

B. Neural Network Architectures

As mentioned previously, DeepFM [6] is a neural network architecture that extends the core idea presented by [4], that is combining a wide component with a deep component, by replacing the wide part of the architecture, which was previously a linear model, with a Factorization Machine. This allows for modelling both first and second order relationships, while at the same time being more robust to sparse data, as is the case with most user-item interaction matrices. Moreover, the authors introduced an embedding layer for representing the input and used this new representation as the shared input to the deep and wide components. The model thus allows for end-to-end training of the network instead of the manual feature engineering by human experts that the original method required.

Authors of [13] further extended the above work by introducing xDeepFM (eXtreme Deep Factorization Machine). This architecture utilized a linear model coupled with a deep model, as proposed in [4], and adopted the embedding layer introduced in DeepFM. However, they introduced an additional component, the Compressed Interaction Network (CIN). This component's properties include user-item interactions being applied at a vector-wise level, measuring high-order feature interactions explicitly, and its complexity increasing in a non-exponential manner as the dimensionality of interactions increases.

In a similar manner to the above works, [7] proposed a novel architecture, called Neural Factorization Machine (NFM), that combines the functionality of a Factorization Machine with the abilities for modeling nonlinear higher order relationships that characterize a deep neural network. By definition, a NFM is more flexible than a Factorization

Machine since the latter can be considered as a special case of the former. Given a sparse vector $x \in R^n$ as input, with the value $x_i = 0$ signifying that the ith feature is not present in the current input, a NFM calculates it input as:

$$\hat{y}_{NFM}(x) = w_0 + \sum_{i=1}^{n} x_i w_i + f(x) \qquad (2)$$

Where the first two terms are known from equation (1), that describes a Factorization Machine, while the third term, $f(x)$, is a deep neural network, the core component that this architecture introduces. Finally, as was the case with both neural network architectures described previously, the input is passed through an embedding layer in order to reduce dimensions and sparsity.

C. Reinforcement Learning

Reinforcement learning is a machine learning paradigm that describes the process where an intelligent agent performs actions in an environment, with the aim of maximizing some reward. Fields of application include robotics, autonomous driving, natural language processing, etc. [20]. For the needs of this work, we define the environment as a user and a list of items on which the agent (neural network model) will perform actions, which correspond to recommending some of the available items. The agent will be rewarded for each item that would be selected by the user, and therefore will aim to maximize its total reward during the training process.

In more detail, for a given user and a list of items, for which the user's preference is known, the agent iterates through the list, recommending some items to the user and discarding others. After the whole list of items has been processed, the agent will have produced a subset corresponding to the model's recommendations, with each selection or rejection being treated as a separate action. Actions that led to a correct recommendation, that is the selection of a positively labeled item and the rejection of a negatively labeled item, yield a positive reward, while incorrect actions yield zero reward.

IV. Experiments

A. Dataset

The MovieLens-1m [15] is an established benchmarking dataset in the Recommender Systems literature. It is provided by the GroupLens research lab of the university of Minnesota, and its goal is the facilitation and testing of relevant algorithms. This dataset contains 1 million anonymized ratings for 4,000 movies from 6,000 unique users, as well as additional information concerning the users and the movies (user gender, age, movie title, description, etc.). In correspondence with our research goals, we only utilize the information regarding the ratings (ranging from 1 to 5), user IDs and item IDs. Ratings with a score lower or equal to 3 are considered as negative samples, while ratings with a larger score are considered as positive samples.

B. Evaluation Approach

For evaluating the performance of each architecture, we utilize the Precision, Recall and AUC, which are common metrics in the field of machine learning [3]. Precision concerns the percentage of relevant samples among the retrieved samples, i.e., the percentage of samples that belong in the positive class from the total samples classified as positive. Recall describes the percentage of positive samples retrieved, i.e., the percentage of positive samples retrieved from the total positive samples in the dataset. The Receiver Operator Characteristic (ROC) is a curve that plots the true positive rate against the false positive rate in various threshold values. Resulting from the ROC curve, the Area Under the Curve (AUC) metric describes an algorithm's ability to separate in between two classes, with values higher than 50% signifying an increasingly better performance in distinguishing between positive and negative samples.

C. Implementation Details

We proceed to describe our implementation details. We follow the typical evaluation setting of splitting the dataset into two non-overlapping sets, keeping 80% of the data for training and 20% for validation and testing of the models. For fair comparison, the learning rate is set to 10^{-3} for all models using the Adam optimizer, the maximum number of epochs is set to 100, while we also apply early stopping when performance ceases to improve in the validation set.

The neural network part of DeepFM consists of 2 hidden layers, each with 16 neurons and ReLU activations, and an output layer with a sigmoid activation function, with the same applying for the xDeepFM model. The Neural Factorization Machine model consists of an embedding layer followed by a Bi-Interaction pooling layer and two hidden layers, with the output being extracted by a single neuron with linear activation. All of the presented deep neural networks are implemented using the PyTorch framework and are trained in an end-to-end manner.

D. Performance Evaluation

In table 1, we present the results of the comparative evaluation of the three deep neural models, along with the results yielded by random selection. We observe that the DeepFM and xDeepFM models exhibit significantly better results than those of the Neural Factorization Machine model. This can be attributed to the more sophisticated structure of the first two architectures since they consist of both wide and deep components. The xDeepFM model yields generally better results than DeepFM, especially for the Recall metric, which further strengthens the significance of the addition of the CIN network. This is also a reasonable outcome, since xDeepFM builds and improves upon the DeepFM architecture, as described previously. We conclude that xDeepFM is the best performing architecture for this reinforcement learning setting, with DeepFM being a close second. It this therefore recommended to still test the performance of both methods if another dataset is evaluated in future work. Finally, we note that all methods are significantly better than the performance of random selection.

TABLE I. Performance comparisson on the MovieLens-1m dataset [15].

Model	*Precision*	*Recall*	*AUC*
DeepFM [6]	**75.01 %**	65.27 %	71.10 %
xDeepFM [13]	73.50 %	**73.30 %**	**72.10 %**
NFM [7]	69.98 %	57.71 %	63.27 %
Random	57.19 %	49.96 %	49.72 %

V. Conclusions

In this work, we compared three state-of-the art collaborative filtering recommender system architectures, based on deep neural networks, under a reinforcement

learning setting. The popular MovieLens-1m dataset was utilized to benchmark the selected models using common metrics for evaluating the performance machine learning algorithms. Our experiments showcased that the training of all the networks converges successfully under this setting, yielding significantly better performance than random selection, with the xDeepFM architecture exhibiting the best results. Future work could utilize additional datasets, suitable for further evaluating the performance and overall properties of the recommender system algorithms. Moreover, the whole learning process could benefit from the introduction of a trainable critic for facilitating a more robust evaluation of the actor's actions, which for our case are the system's recommendations, in an actor-critic reinforcement learning setting.

ACKNOWLEDGMENT

This work has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T2EDK-03843).

REFERENCES

[1] J. Bennett and S. Lanning, "The Netflix Prize," in Proceedings of KDD Cup and Workshop, 2007.

[2] S. Blanda, "Online Recommender Systems – How Does a Website Know What I Want?" American Mathematical Society, 2015.

[3] C. D. Brown and H. T. Davis, "Receiver operating characteristics curves and related decision measures: A tutorial," Chemometrics and Intelligent Laboratory Systems, Volume 80, Issue 1, 2006, Pages 24-38. ISSN 0169-7439. https://doi.org/10.1016/j.chemolab.2005.05.004

[4] [3] H. T. Cheng, et al., "Wide & Deep Learning for Recommender Systems," in Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016, DOI:https://doi.org/10.1145/2988450.2988454

[5] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[6] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: a factorization-machine based neural network for CTR prediction," in Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17), AAAI Press, 2017, 1725–1731.

[7] X. He and T. S. Chua, "Neural Factorization Machines for Sparse Predictive Analytics," in Proceedings of SIGIR '17, Shinjuku, Tokyo, Japan, 2017.

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural Collaborative Filtering," in Proceedings of the 26th International Conference on World Wide Web (WWW '17), International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017, 173–182. DOI:https://doi.org/10.1145/3038912.3052569

[9] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95). ACM Press/Addison-Wesley Publishing Co., USA, 1995, 194–201. DOI:https://doi.org/10.1145/223904.223929

[10] KismetK. Netflix: Recommendations Worth a Million. Retrieved March 28, 2021 from https://rstudio-pubs-static.s3.amazonaws.com/190289_5089121940cc4f74b7e87c963f6e3b65.html

[11] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, GroupLens: applying collaborative filtering to Usenet news. Commun. ACM 40, 3 (March 1997), 77–87. DOI:https://doi.org/10.1145/245108.245126

[12] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, Aug. 2009. doi: 10.1109/MC.2009.263.

[13] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "XDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18), Association for Computing Machinery, New York, NY, USA, 2018, 1754–1763. DOI:https://doi.org/10.1145/3219819.3220023

[14] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," in IEEE Internet Computing, vol. 7, no. 1, 2004, pp. 76-80.

[15] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," in ACM Trans. Interact. Intell. Syst. 5, 4, Article 19 (January 2016), 2015, 19 pages. DOI:https://doi.org/10.1145/2827872

[16] A. I. Metsai, I. Tabakis, K. Karamitsios, K. Kotrotsios, P. Chatzimisios, G. Stalidis, and K. Goulianas, "Customer Journey: Applications of AI & Machine Learning in E-Commerce," in Proceedings of the International Conference on Interactive Mobile and Communication Technologies and Learning (IMCL), 2021.

[17] S. Rendle, "Factorization Machines," in Proceedings of the IEEE International Conference on Data Mining, 2010. DOI:https://doi.org/10.1109/ICDM.2010.127

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in Proceedings of the 1994 ACM conference on Computer supported cooperative work (CSCW '94). Association for Computing Machinery, New York, NY, USA, 1994, 175–186. DOI:https://doi.org/10.1145/192844.192905

[19] E. Rich, "User modeling via stereotypes," in Cognitive Science, vol. 3, no. 4, 1979, p. 329–354.

[20] T. J. Sejnowski, The Deep Learning Revolution. MIT Press, 2018.

[21] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating word of mouth," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95). ACM Press/Addison-Wesley Publishing Co., USA, 1995, 210–217. DOI:https://doi.org/10.1145/223904.223931