Service Recommendations with Deep Learning: a Study on Neural Collaborative Engines

Pasquale De Rosa[1], Michel Deriaz[1], Marco De Marco[2], and Luigi Laura[2]

[1] Information Science Institute, GSEM/CUI
University of Geneva
Geneva, Switzerland
{pasquale.derosa, michel.deriaz}@unige.ch
[2] International Telematic University UNINETTUNO
Rome, Italy
{marco.demarco, luigi.laura}@uninettunouniversity.net

Abstract. The present paper aims to investigate the adoption of Neural Networks for recommendation systems and to propose Deep Learning architectures as advanced frameworks for designing Collaborative Filtering engines. Recommendation systems are data-driven infrastructures which are widely adopted to create effective and cutting-edge smart services, allowing to personalize the value proposition and adapt it to changes and variations in customers' preferences. For this purpose we will introduce a Collaborative Filtering algorithm based on the adoption of a "deep" Feed-Forward Network, inspired by a recent research on neural-based service recommenders; given these assumptions, we will confirm the suitability of Feed-Forward Neural Networks as effective recommendation algorithms, laying the foundations for further studies in neural-based recommendation science.

Keywords: Neural Networks · Recommendation Systems · Deep Learning · Smart Services · Collaborative Filtering · Service Science

1 Introduction

This paper will investigate the suitability of a Deep-Learning-based approach for designing advanced collaborative recommendation systems. Recommendation engines can be regarded as noticeable examples of "smart services" [1] enablers, data-driven architectures designed to facilitate the users' decision-making process, in accordance with a customer-centric perspective [2].
The role of data science and advanced analytics in the smart service design process in definitely prominent [3] [4], providing techniques, instruments and sophisticated algorithmic tools for mapping and describing properly a dynamic world made up by dynamic customers [3]. Given these assumptions, the role of recommendation systems as smart and personalized architectures that make use of previously collected and labeled customer data to provide them with effective service suggestions is surely relevant, and the adoption of a "Neural-based" approach can lead to outstanding performances also if compared with more "traditional" methodologies [5].

2 Theoretical Framework

In this section we will provide the theoretical background of our research study: in particular, in Sect. 2.1 we will describe Recommendation Systems and the Collaborative Fltering approach, in Sect. 2.2 we will focus on Neural Networks and fundamentals of Deep Learning and in Sect. 2.3 we will provide a brief overview of the recent advancements in the scientific literature.

2.1 Recommendation Engines and Collaborative Filtering

In the introductory chapter, we provided a brief overview of the impact of data science and big data analytics in shaping a new era for service science.
One noticeable example is represented by "recommendation systems", powerful algorithmic engines designed in order to simplify the customers' decision-making process providing them with relevant and effective service suggestions; several different approaches to recommendation emerged, like Collaborative Filtering [10], Content-based Filtering [10] and Hybrid architectures [6] [8] [9] [10] [11].
Among those paradigms, is worth focusing on Collaborative Filtering, an approach to recommendation based on the convergence between the preferences of different users, that allows to "extend" the customers' purchase intentions to unknown and/or unexplored service categories [7]. However, the adoption of a Collaborative perspective when designing a service recommendation engine could lead to some disadvantages, like the "cold-start problem" (common for new, unrated goods), the "sparse" nature of user ratings and the computational complexity.
When building an effective Collaborative engine, in literature is widely adopted the "user/rating" matrix, sparse by definition, in which the customers' preferences are represented by a $m \times n$ structure, where m are the overall service users and n the total number of services previously rated by the clients [10].

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1n} \\ r_{21} & r_{22} & r_{23} & \dots & r_{2n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ r_{m1} & r_{m2} & r_{m3} & \dots & r_{mn} \end{pmatrix}$$

The user/rating matrix allows to create effective algorithms capable of generating predictions about the customer ratings, that can be used to define the basis of subsequential recommendations.
Collaborative Filtering techniques are generally sub-divided in two different "families": Neighborhood-based and Model-based [10].
The Neighborhood-based algorithms originate from the "nearest neighbors" concept: a subset consisting of the k most similar users to a specific customer, whose ratings are defined as a weighted combination of the reviews expressed by his "nearest neighbors" in the past. The Neighborhood-based models are widely adopted in practical applications for their characteristic computational efficiency,

for the proven stability when dealing with variations in the data structure and for the capability to arouse the customers' interest in new services (serendipity) [12].
The "Model-based" recommenders, on the other hand, make use of statistical techniques to provide an estimation of user ratings [10]. Among these, it is worth mentioning the "Latent Factors Models", like the "Singular Value Decomposition" (SVD), based on the assumption that the similarity between users is determined by the presence of latent and hidden structures in the data, and Artificial Neural Networks, sophisticated machine learning algorithms capable under certain conditions of overcoming in effectiveness more "traditional" approaches to recommendation [5].

2.2 Neural Networks and Deep Learning

In the previous section, we affirmed that model-based techniques constitute a significant advance in the development of cutting-edge Collaborative engines; more specifically, a new milestone in this field could be represented by Artificial Neural Networks, machine learning architectures whose suitability for recommendation systems has already been investigated in recent studies [5] .
A first "prototype" of Neural Network was theorized by the psychologist Frank Rosenblatt in 1958, and was named "Perceptron" [13] [14]: the original Perceptron was a classification algorithm that, starting from a number n of inputs, $x_1, x_2, ..., x_n$, each one assigned with a weight $\omega_1, \omega_2, ..., \omega_n$, produced a binary outcome as explained in the following equation:

$$\text{Output} = \begin{cases} 0, & \text{if } \sum_n x_n \omega_n \leq t \\ 1, & \text{if } \sum_n x_n \omega_n > t \end{cases} \tag{1}$$

Where t is an exogenous treshold value determined by the researcher in accordance with the purposes of the study; in practical applications, the threshold usually appears in the other side of the inequality, "replaced" by what's known as the Perceptron's bias b, defined as $-t$ [14]:

$$\text{Output} = \begin{cases} 0, & \text{if } \sum_n x_n \omega_n + b \leq 0 \\ 1, & \text{if } \sum_n x_n \omega_n + b > 0 \end{cases} \tag{2}$$

The second equation represents the activation conditions of the Perceptron, and in literature is generally defined the "activation function" of the Neural Network (more specifically, this expression is also known as the "Heaviside step function") [14].
The leftmost, first layer in the network is also called "input layer", while the final activation layer contains the output neuron; in the past years, several enhancements were made to the original Perceptron's architecture, introducing middle layers between the inputs and the final activation, also known as "hidden layers".

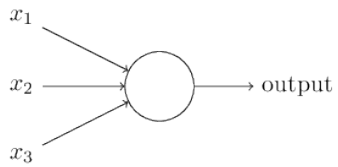

Fig. 1: A graphical representation of the Perceptron [14].

A Neural Network consisting of one or multiple hidden layers between the first and the final neurons is also called "Multi-layer Perceptron" or "Feed-Forward Neural Network", if the output from each layer is used as input to the next one, and information is never fed back [14].

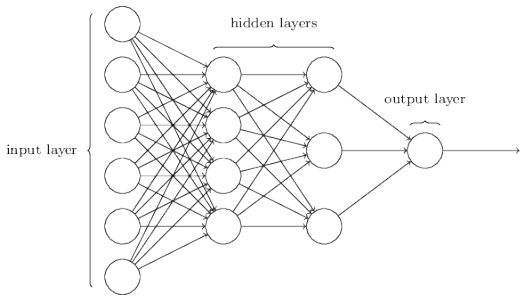

Fig. 2: A graphical representation of the Multi-Layer Perceptron [14].

The optimal parameters of the Network (weights and biases) can be determined as a result of an algorithmic process, defining a non-negative "cost function" $C(\omega, b)$ and minimizing it by finding a combination of weights and biases that generates the lowest achievable model loss [14]. A well-known example of minimization algorithm for Neural Networks is represented by the "gradient descent" technique and its most commonly adopted variant, the "stochastic gradient descent".
In addiction to the basic Multi-Layer Perceptron architecture, other several neural structures emerged in the scientific literature, among which is worth mentioning Convolutional Neural Networks and Recurrent Neural Networks [14].

Convolutional Neural Networks Convolutional Neural Networks (CNNs) are complex neural architectures specially suitable for image recognition and computer vision applications, in which the hidden units are not "fully connected" to each input neuron, but connections are built only in small localized regions (called "local receptive fields") [14]; for each layer, several "feature maps" are created through a mathematical convolution, building a "convolutional layer"

which can detect n different characteristics of the input data.
The convolutional layers are usually followed by structures known as "pooling layers", whose aim is to is simplify the information in the previous output applying several transformations like max-pooling or L2 pooling [14]; lastly, for the final activation layer (which is "fully-connected"), in image recognition tasks are usually adopted functions like the Sigmoid (for binary classifications) or the Softmax (for multiple classifications).

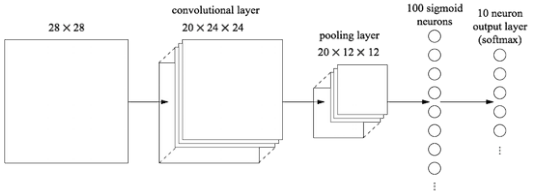

Fig. 3: A graphical representation of a Convolutional Neural Network [14].

Recurrent Neural Networks Recurrent, or "Feed-back" Neural Networks (RNNs) are Deep Learning architectures in which the behaviour of each neuron is not only determined by the activation in the previous hidden layer, but also by the earlier states [14].
More specifically, the activation function for each hidden layer of a Recurrent Neural Network can be represented by the following expression [15]:

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}, \theta) \tag{3}$$

In which the hidden layer a the time t, $h^{(t)}$, is function of the previous state, $h^{(t-1)}$, of the current input $x^{(t)}$ and of the activation function adopted, θ.

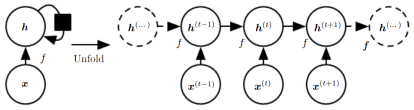

Fig. 4: A graphical representation of a Recurrent Neural Network [15].

The training process of Recurrent Neural Networks is often characterized by the "unstable gradient problem": the gradient of the adopted cost function

tends to get smaller or bigger as it is propagated back through layers, resulting in a final "vanishing" or "explosion" and making RNNs unable to model "long term dependencies" between data [14]. For this purpose, in practical applications are commonly adopted other complex architectures known as "gated RNNs": Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), models specifically designed to be capable of accumulating information over a long time duration [15].

2.3 Neural Collaborative Filtering: a Literature Overview

The adoption of Neural Networks for recommendation tasks is widely attested in the scientific literature: among the most recent works in this field is certainly worth mentioning the contribution of Vassiliou et al. [16], that introduced an hybrid framework for recognizing implicit patterns between user profiles and items in order to provide personalized suggestions; moreover, the survey conducted by Zhang et al. [17] provided a taxonomy of neural-based recommendation models and a comprehensive overview of both the current trends and the new perspectives of this scientific field.
Lastly, is worth to cite the research from Bobadilla et al. [18], that provided an innovative deep-learning based framework introducing the "reliability" concept to improve the model's predictive capability and the quality of recommendations, and the work that inspired the present study, "Collaborative Recommendations with Deep Feed-Forward Networks" [5], that analyzed the better performances of neural-based recommenders in comparison with more "traditional" approaches like k-Nearest Neighbors and Singular Value Decomposition.

3 Research Study

The present section will discuss the results of our research, analyzing in depth the effectiveness of a Neural-based Collaborative Filtering algorithm: more specifically, in Sect. 3.1 we will provide an overview of the "Movielens 100K" dataset used for the training process, in Sect. 3.2 we will discuss in details the architecture of the Neural Network and in Sect. 3.3 we will describe the experimental results and the findings of our study.

3.1 Preliminaries and Data Structure

We based our study on the findings of the paper [5], aiming to extend its scope and investigate further the suitability of Feed-Forward Neural Networks for Collaborative Filtering by analyzing the performance of a "deeper" neural recommender.
In order to guarantee a methodological coherence with the previous study, we trained our model on the "Movielens 100K" dataset, whose main characteristics are listed below [19]:

- 100.000 ratings (from 1 to 5), collected from 943 users on 1682 movies.

– Four variables of interest (the IDs for users and movies, ratings and timestamps).

This dataset, which constitutes a stable benchmark in recommendation science, was collected through the MovieLens web site between September 1997 and April 1998 and subsequently cleaned up, removing all users with less than 20 ratings or devoid of complete demographic information.

User ID	Movie ID	Rating	Timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596

Table 1: Summary of the first five elements in the training dataset, randomly ordered.

3.2 Model Description and Training Process

The present paragraph aims to describe in depth the structure of our Neural Collaborative Filtering algorithm: more specifically, in the following sub-sections will be provided further indications on the model architecture, in addiction to a detailed explanation of the optimization techniques adopted for the training process.

Model Architecture The first step of our research was to turn all Movies and Users IDs into categoricals, in order to create Entity Embedding tensors of shape ($batch\_size$, 1, 256). The adoption of an Embedding Layer allows not only to reduce the memory usage if compared with one-hot encoding, but also to reveal the intrinsic properties of the input variables [20].
Directly after the Embedding Layers, we created two Flatten Layers in order to reduce the dimensionality of the previous output, making it suitable for the subsequent computations.
The previously generated Embedding Layers were subsequently concatenated into one Merged Layer of shape ($batch\_size$, 512), before the first ReLu (Rectified Linear) Activation.
Moreover, we decided to use a ReLu Activation also for the final layer of the Network, since rating predictions were bounded to non-negative values between 1 and 5.
In addiction to simple hidden Dense activations, we added to our Network several Dropout Layers: those architectures were specifically developed in order to

address overfitting by randomly dropping units from the Neural Network during the training process [21].
For our research purposes, we decided to apply Dropouts (with a ratio of 0.5 unities dropped) directly after each Dense/ReLu Activation Layer in the Network.

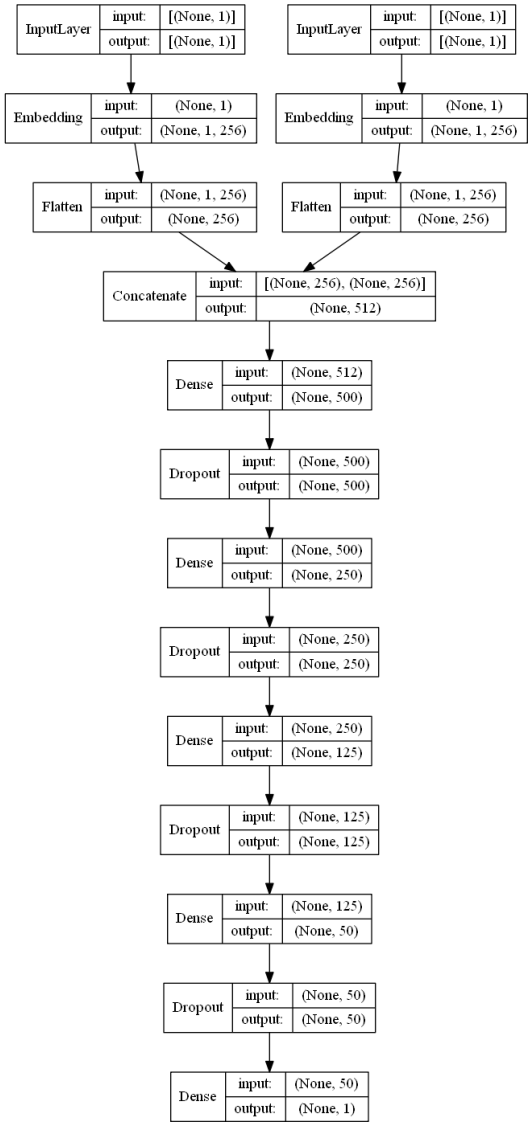

Fig. 5: Architecture of the Feed-Forward Neural Network

Optimization and Training Once defined the model structure, we initialized the training process by selecting the most appropriate cost function and a proper optimization algorithm.
Since the rating prediction was a regression task, we decided to use the MSE (Mean-Squared Error) function to provide an estimation of the model loss; moreover, for the model optimization we used the Adam (Adaptive Moment Estimation) algorithm [22], setting the learning rate to 0.001.
Lastly, we opted to train our model on the 80% of overall data, using the remaining 20% for validation.

3.3 Experimental Results and Research Findings

The study outcomes highlighted the noticeable performance of our Neural recommender, capable of providing accurate rating predictions also if compared to the research that inspired the present paper [5]. After an iterative process, we decided to train our model for 10 epochs with a batch size of 64, since it was the optimal training duration in order to prevent the model from overfitting.

Epoch	Training Loss	Validation Loss
1	1.7421	0.9738
2	1.2687	0.9237
3	1.1357	0.9308
4	1.0462	0.9645
5	0.9845	0.9219
6	0.9390	0.9215
7	0.8995	0.8887
8	0.8761	0.8923
9	0.8550	0.9014
10	0.8279	0.8822

Table 2: Summary of the research findings sorted by the training epoch.

As it can be observed in the table, in fact, the Feed-Forward Neural Network registered in the last training epoch a MSE of 0.8822 (RMSE = 0.9392), an improvement in terms of predictive ability respect to the neural recommender proposed in [5]. Those results confirm the primary role of Neural Networks for the design of successful and cutting-edge recommendation algorithms, with a proven stability and a noticeable accuracy in rating predictions.

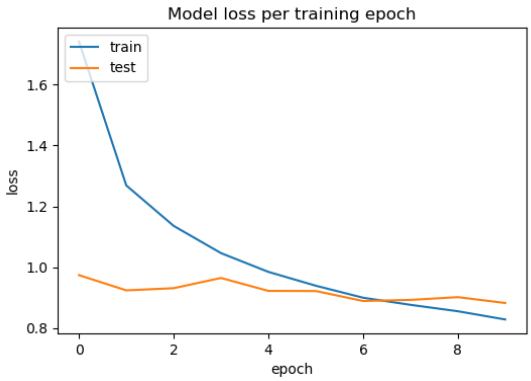

Fig. 6: Loss function trend per epoch of training.

4 Conclusions

The results of this study confirmed the suitability of Feed-Forward Neural Networks for designing advanced Collaborative recommenders: in fact, adopting a "deeper" and more complex model architecture, it was possible to consolidate and even improve the outcomes, in terms of predictive ability, of the study that inspired our research [5].
These assumptions lead us to suppose that the adoption of Neural Networks for service recommendations should be extended also to a broader range of techniques, like sequential "Feed-Back" architectures as the "Long-Short Term Memories" (LSTM) and the "Gated Recurrent Units (GRUs).
The service consumer behaviour can also be analyzed as a dynamic process, instead of a static sequence of unrelated actions over a certain time frame [23]: in accordance with this statement, we can assume that the adoption of a sequence-based framework, like a Recurrent Neural Network, could lead to relevant performances and even lay the foundations for significant advances in service recommendation design [24] [25].

References

1. Alt, Rainer; Demirkan, Haluk; Ehmke, Jan Fabian; Moen, Anne & Winter, Alfred. (2019). Smart services: The move to customer orientation, Electron Markets 29, 1–6, https://doi.org/10.1007/s12525-019-00338-x.
2. Blöcher, Katharina & Alt, Rainer. (2018). An Approach for Customer-Centered Smart Service Innovation Based on Customer Data Management, 9th International Conference, IESS 2018, Karlsruhe, Germany, September 19–21, 2018, Proceedings.
3. Dermikan, Haluk; Bess, Charlie; Spohrer, C. James; Rayes, Ammar; Allen, Don; Moghaddam, Yassi. (2015). Innovations with Smart Service Systems: Analytics, Big Data, Cognitive Assistance, and the Internet of Everything, CAIS, 37, 35.

4. Meierhofer, Jürg & Meier, Kevin. (2017). From Data Science to Value Creation, 8th International Conference, IESS 2017, Rome, Italy, May 24–26, 2017, Proceedings.
5. Cascio Rizzo, Giovanni Luca; De Marco, Marco; De Rosa, Pasquale & Laura, Luigi. (2020). Collaborative Recommendations with Deep Feed-Forward Networks: An Approach to Service Personalization, In: Nóvoa H., Drăgoicea M., Kühl N. (eds) Exploring Service Science, IESS 2020, Lecture Notes in Business Information Processing, vol 377, Springer, Cham.
6. Basu, Chumki; Hirsh, Haym & Cohen, William. (1998). Recommendation as classification: Using social and content-based information in recommendation, In: Proceedings of the Fifteenth National Conference on Artificial Intelligence, pp. 714–720, 1998.
7. Bhatnagar, Vishal. (2016). Collaborative Filtering Using Data Mining and Analysis, IGI Global, 2016.
8. Cotter, Paul & Smyth, Barry. (2000). PTV: Intelligent personalized TV guides, In: Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 957–964, 2000.
9. Melville, Prem; Mooney, Raymond J. & Nagarajan, Ramadass. (2002). Content-boosted collaborative filtering for improved recommendations, In: Proceedings of the Eighteenth National Conference on Artificial Intelligence, pp. 187-192, Edmonton, 2002.
10. Melville, Prem & Sindhwani, Vikas. (2017). Recommender Systems, In: Sammut, Claude & Webb, Geoffrey I. (eds), Encyclopedia of Machine Learning and Data Mining, Springer, Boston, MA, 2017.
11. Pazzani, Michael J. (1999). A framework for collaborative, content-based and demographic filtering, Artificial Intelligence Review, Volume 13, Issue 5-6, pp. 393–408, 1999.
12. Ricci, Francesco; Rokach, Lior; Shapira, Bracha & Kantor, Paul B. (2011) Recommender Systems Handbook, Springer Science+Business Media, LLC 2011.
13. Rosenblatt, Frank. (1959. The perceptron: a probabilistic model for information storage and organization in the brain,Psychological review, 65 (6), pp. 386-408.
14. Nielsen, Michael A. (2015). Neural Networks and Deep Learning, Determination Press, 2015.
15. Goodfellow, Ian; Bengio, Yoshua & Courville, Aaron. (2016). Deep Learning, MIT Press.
16. Vassiliou, Charalampos; Stamoulis, Dimitris & Martakos, Drakoulis. (2006). A Recommender System Framework combining Neural Networks and Collaborative Filtering, Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, pp. 285-290.
17. Zhang, Shuai; Yao, Lina; Sun, Aixin; & Tay, Yi. (2018). Deep Learning based Recommender System: A Survey and New Perspectives, ACM Comput. Surv. 1, 1, Article 1, 35 pages.
18. Bobadilla, Jesus; Alonso, Santiago & Hernando, Antonio. (2020). Deep Learning Architecture for Collaborative Filtering Recommender Systems, Applied Sciences, 10, 2441, 10.3390/app10072441.
19. Harper, Maxwell F. & Konstan, Joseph A. (2015). The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages.
20. Guo, Cheng & Berkhahn, Felix. (2016). Entity Embeddings of Categorical Variables, arXiv:1604.06737.

21. Srivastava, Nitish; Hinton, Geoffrey; Krizhevsky, Alex; Sutskever, Ilya & Salakhutdinov, Ruslan. (2014). Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research, Volume 15 Issue 1, pp. 1929-1958.
22. Kingma, P. Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization, arXiv:1412.6980.
23. Jacoby, Jacob; Kohn Berning, Carol & Szybillo, George. (1976). Time and Consumer Behavior: An Interdisciplinary Overview, Journal of Consumer Research, Vol. 2, No. 4 (Mar. 1976), pp. 320-339.
24. Hidasi, Balázs; Karatzoglou, Alexandros; Baltrunas, Linas; Tikk, Domonkos. (2015). Session-based Recommendations with Recurrent Neural Networks, CoRR abs/1511.06939.
25. Devoogh, Robin & Bersini, Hugues. (2016). Collaborative Filtering with Recurrent Neural Networks, arXiv:1608.07400.