



## Service Recommendations with Deep Learning: A Study on Neural Collaborative Engines

Pasquale De Rosa<sup>1,\*</sup>, Michel Deriaz<sup>2</sup>, Marco De Marco<sup>3</sup>, Luigi Laura<sup>4</sup>

<sup>1,\*</sup>University of Geneva, Switzerland, [pasquale.derosa@unige.ch](mailto:pasquale.derosa@unige.ch)

<sup>2</sup>University of Geneva, Switzerland, [michel.deriaz@unige.ch](mailto:michel.deriaz@unige.ch)

<sup>3</sup>International Telematic University Uninettuno, Italy, [marco.demarco@uninettunouniversity.net](mailto:marco.demarco@uninettunouniversity.net)

<sup>4</sup>International Telematic University Uninettuno, Italy, [luigi.laura@uninettunouniversity.net](mailto:luigi.laura@uninettunouniversity.net)

### Abstract

**Background:** *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.*

**Method:** *Our research represents an exploratory investigation on the adoption of Neural Networks for Recommendation Systems, inspired by the findings of a recent study on service science that highlighted the suitability of those models for designing cutting-edge recommenders capable of overcoming stable traditional benchmarks like the Singular Value Decomposition and the k-Nearest Neighbors algorithms. Following this study, we designed a more "complex" Feed-Forward Neural Network, trained on the "Movielens 100K" dataset using the Mean-Squared Error function to approximate the model loss generated and the Adaptive Moment Estimation algorithm (Adam) for the parameters optimization.*

**Results:** *The results of this study demonstrate the primary role of Feed-Forward Neural Networks for designing advanced Collaborative recommenders, consolidating and even improving the outcomes of the work that inspired our research.*

**Conclusion:** *Given these assumptions, we 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.

This research article was submitted on 17-Feb-2021 and under one revision, accepted on 22-May-2021.

Citation: De Rosa, P., Deriaz, M., De Marco, M., & Laura, L. (2022). Service Recommendations with Deep Learning: A Study on Neural Collaborative Engines. *Pacific Asia Journal of the Association for Information Systems*, 14(2), 59-70. <https://doi.org/10.17705/1pais.14205>

Copyright © Association for Information Systems.

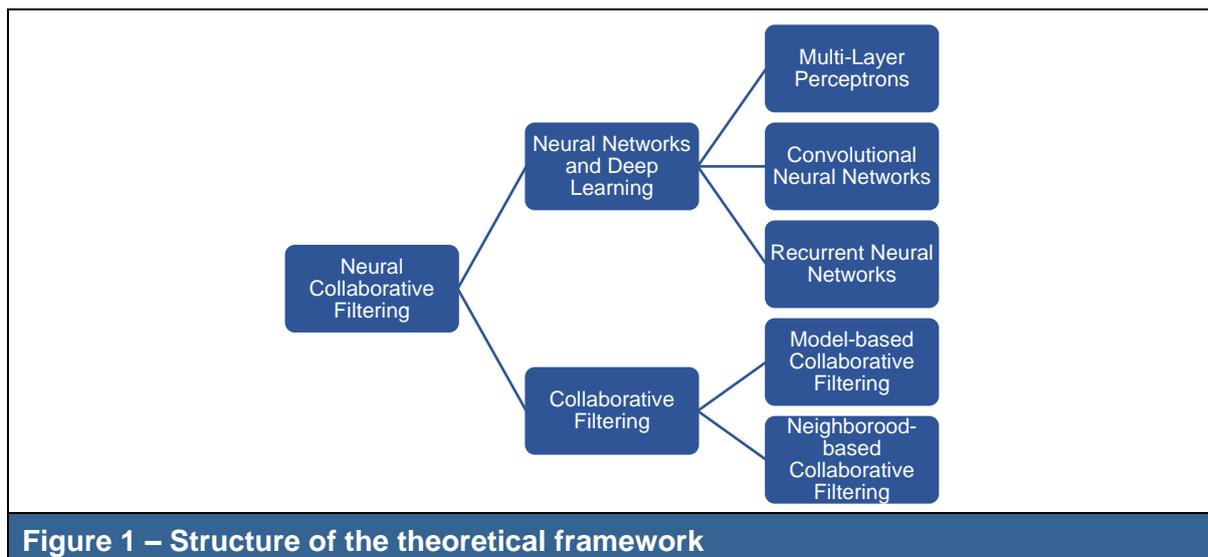
## Introduction

This paper investigates the suitability of a Deep-Learning-based approach for designing advanced collaborative recommendation systems. More in depth, we start from the findings of a recent paper in service recommendation science, “Collaborative Recommendations with Deep Feed-Forward Networks” (Cascio Rizzo et al., 2020) that analyzed the use of Deep Learning methodologies to build effective recommenders, in order to further enrich its scope by introducing a new, more complex and effective neural Collaborative Filtering engine.

Recommendation engines can be regarded as noticeable examples of “smart services” (Alt et al., 2019) enablers, data-driven architectures designed to facilitate the users’ decision-making process, in accordance with a customer-centric perspective (Blöcher & Alt, 2018). The role of data science and advanced analytics in the smart service design process is definitely prominent (Demirkan et al., 2015; Meierhofer & Meier, 2017), providing techniques, instruments and sophisticated algorithmic tools for mapping and describing properly a dynamic world made up by dynamic customers (Demirkan et al., 2015). 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 (Cascio Rizzo et al., 2020).

## Theoretical Framework

In this section we provide the theoretical background of our research study: in particular, in the sect. “Recommendation Engines and Collaborative Filtering” we describe Recommendation Systems and the Collaborative Filtering approach, in the sect. “Neural Networks and Deep Learning” we majorly focus on Neural Networks and fundamentals of Deep Learning and in the sect. “Neural Collaborative Filtering: a Literature Overview” we provide a brief overview of the recent advancements in the scientific literature. The following schema summarizes the structure of this chapter, highlighting the organization of the theoretical framework.



## Recommendation Engines and Collaborative Filtering

In the introductory chapter, we gave to the reader 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, Content-based Filtering (Melville & Sindhvani, 2017) and Hybrid architectures (Basu et al., 1998; Cotter & Smyth, 2000; Melville et al., 2002; Melville & Sindhvani, 2017; Pazzani, 1999). 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 (Bhatnagar, 2016). 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 (Melville & Sindhvani, 2017).

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

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 (Melville & Sindhvani, 2017).

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) (Ricci et al., 2011). The "Model-based" recommenders, on the other hand, make use of statistical techniques to provide an estimation of user ratings (Melville & Sindhvani, 2017). 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 (Cascio Rizzo et al., 2020).

## 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 (Cascio Rizzo et al., 2020). A first "prototype" of Neural Network was theorized by the psychologist Frank Rosenblatt and was named "Perceptron" (Nielsen, 2015; Rosenblatt, 1959). The original Perceptron was a classification algorithm that, starting from a number  $n$  of

inputs,  $(x_1, x_2, \dots, x_n)$ , each one assigned with a weight  $(\omega_1, \omega_2, \dots, \omega_n)$ , produced a binary outcome as explained in the following equation:

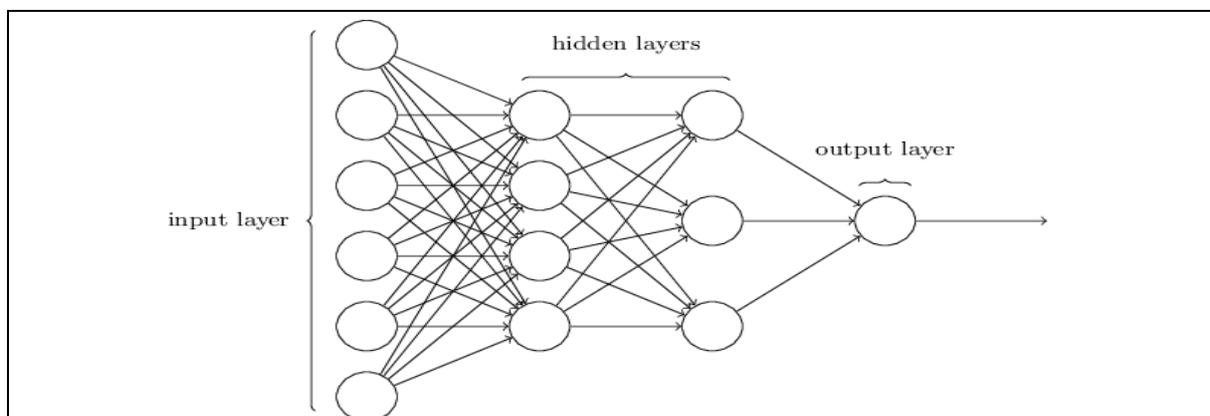
$$y = \begin{cases} 0, & \text{if } \sum_n x_n \omega_n \leq t \\ 1, & \text{if } \sum_n x_n \omega_n > t \end{cases}$$

Where  $t$  is an exogenous threshold 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  $-t$  (Nielsen, 2015):

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

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”) (Nielsen, 2015). 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”. 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 (Nielsen, 2015). 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 (Nielsen, 2015). 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 addition 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 (Nielsen, 2015).



**Figure 2 - A graphical representation of the Multi-Layer Perceptron**

## 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") (Nielsen, 2015); 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 simplify the information in the previous output applying several transformations like max-pooling or L2 pooling (Nielsen, 2015); 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).

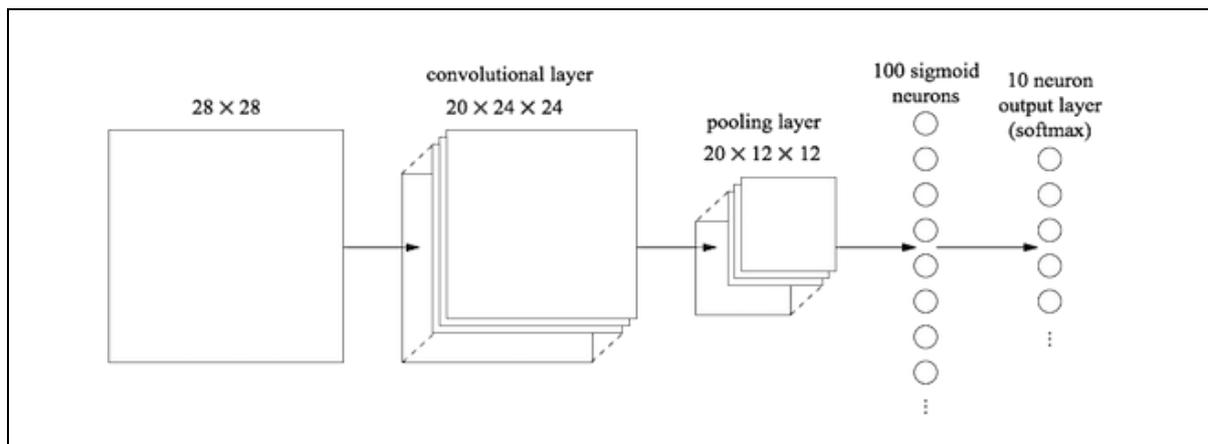


Figure 3 - A graphical representation of a Convolutional Neural Network

## Recurrent Neural Networks

Recurrent, or "Feed-back" Neural Networks (RNNs) are Deep Learning architectures in which the behavior of each neuron is not only determined by the activation in the previous hidden layer, but also by the earlier states (Nielsen, 2015).

More specifically, the activation function for each hidden layer of a Recurrent Neural Network can be represented by the following expression (Goodfellow et al., 2016):

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

In which the hidden layer at 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,  $\theta$ . 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 (Nielsen, 2015). 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 (Goodfellow et al., 2016).

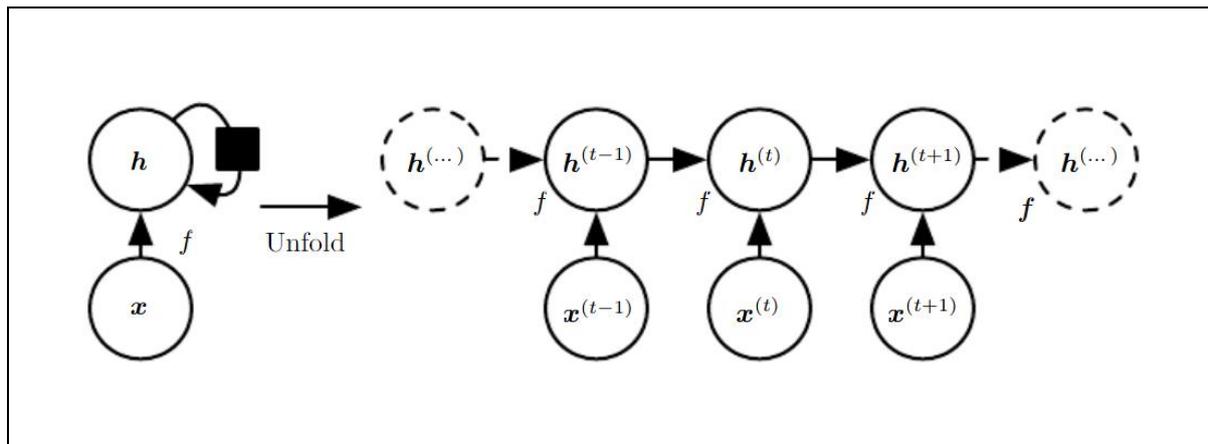


Figure 4 - A graphical representation of a Recurrent Neural Network

### Neural Collaborative Filtering: A Literature Overview

The role of Neural Networks and Deep Learning is surely prominent in almost every recent Machine Learning trend (Khamparia & Singh, 2019), with relevant applications and popularity in artificial intelligence domains such as sentiment analysis and opinion mining (Yadav & Vishwakarma, 2020), object recognition and language modeling (Miikkulainen et al., 2019), speech and image recognition (Bashar, 2019), natural language processing (Bashar, 2019), feature extraction and visualization (Khamparia & Singh, 2019), pattern recognition and computer vision (Liu et al., 2017). Furthermore, 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 (Vassiliou et al., 2006) that introduced an hybrid framework for recognizing implicit patterns between user profiles and items in order to provide personalized suggestions; moreover, another research survey (Zhang et al., 2019) 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 (Bobadilla et al., 2020) 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" (Cascio Rizzo et al., 2020), that analyzed the better performances of neural-based recommenders in comparison with more "traditional" approaches like k-Nearest Neighbors and Singular Value Decomposition.

### Research Study

The present section discusses the results of our research, analyzing in depth the effectiveness of a Neural-based Collaborative Filtering algorithm: more specifically, in the sect. "Preliminaries and Data Structure" we provide an overview of the "Movielens 100K" dataset used for the training process, in the sect. "Model Description and Training Process" we discuss in details the architecture of the Neural Network and in the sect. "Experimental Results and Research Findings" we describe the experimental results and the findings of our study.

## Preliminaries and Data Structure

We based our study on the findings of the paper "Collaborative Recommendations with Deep Feed-Forward Networks" (Cascio Rizzo et al., 2020), 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 (Harper & Konstan, 2015), whose main characteristics are listed below:

- 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	3.0	881250949
186	302	3.0	891717742
22	377	1.0	878887116
244	51	2.0	880606923
166	346	1.0	886397596

## Model Description and Training Process

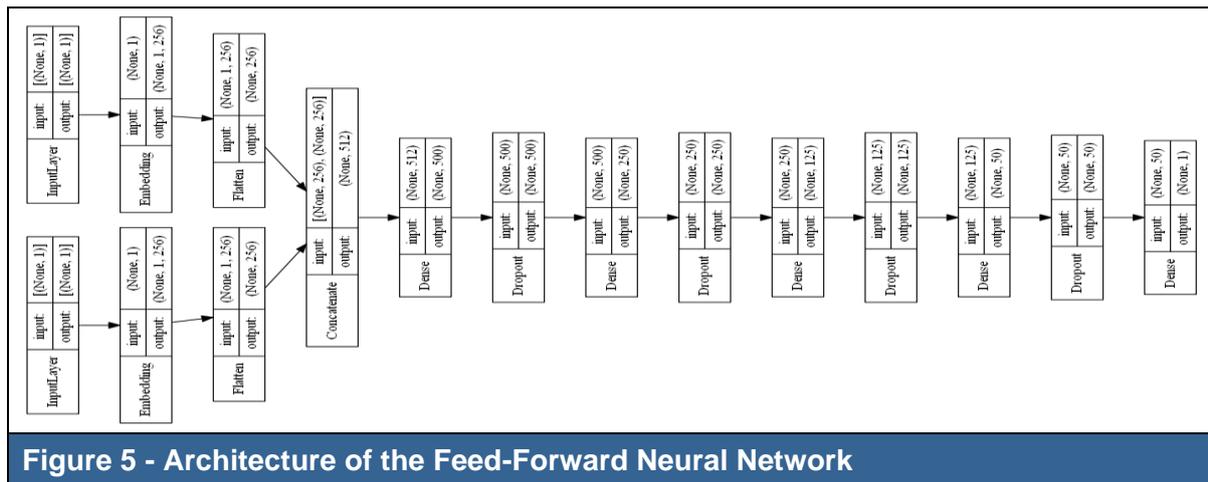
The present paragraph aims to describe in depth the structure of our Neural Collaborative Filtering algorithm: more specifically, in the following sub-sections we provide further indications on the model architecture, in addition to a detailed explanation of the optimization techniques adopted for the 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 (Guo & Berkahn, 2016).

Directly after the Embedding Layers, we created two Flatten Layers in order to reduce the dimensionality of the previous output tensor, making it suitable for the subsequent computations. The previously generated Embedding Layers were subsequently concatenated into one Merged Layer, representing both users and movies, 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 addition 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 (Srivastava et al., 2014). 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.



### 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 (Kingma & Ba, 2014), setting the initial learning rate to 0.001. Lastly, we opted to train our model on the 80% of overall data, using the remaining 20% for validation.

### 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 (Cascio Rizzo et al., 2020). 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.

**Table 2 - Summary of the research findings**

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

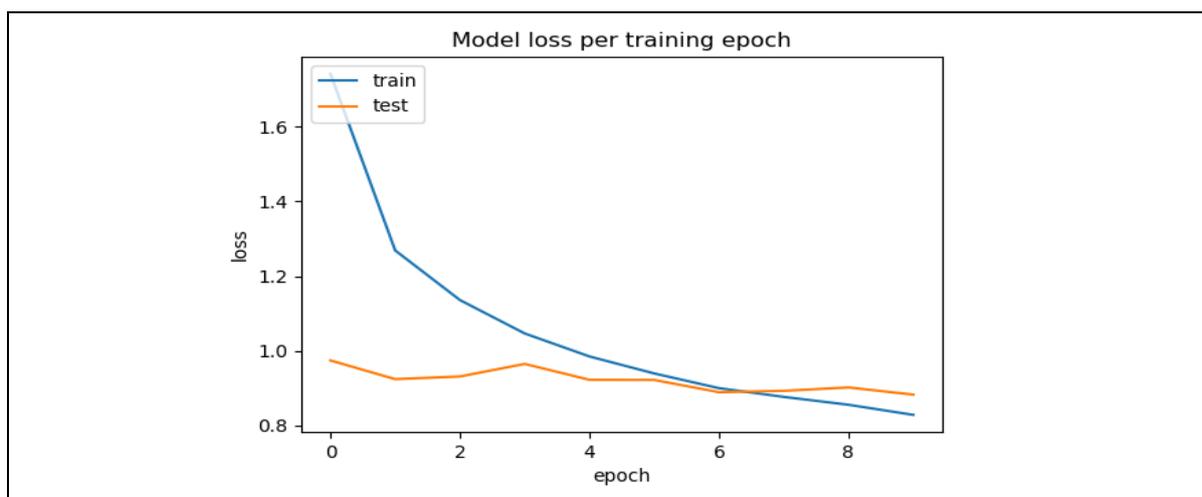
As it can be observed in the table, in fact, the Feed-Forward Neural Network registered in the last training epoch a MSE of 0.8822 (RMSE = 0.9392), an improvement in terms of predictive ability also respect to the best recommendation system proposed in the previous research (Cascio Rizzo et al., 2020). More in depth, the following table synthesizes the major differences emerged between our neural model and the benchmark recommenders:

**Table 3 – Model comparison with the benchmark**

Model	Validation Loss (lowest value)
Deep Neural Recommender*	0.9392
Deep Neural Recommender**	0.9593
Shallow Neural Recommender**	0.9624
SVD Recommender**	0.9809
k-NN Recommender**	0.9813

Notes: \*Research model. \*\* Benchmark model.

Those results confirm the primary role of Neural Networks (especially more complex and “deeper” structures) for the design of successful and cutting-edge recommendation algorithms, with a proven stability and a noticeable accuracy in rating predictions.

**Figure 6 – Loss function trend per epoch of training**

## 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 (Cascio Rizzo et al., 2020).

This led us to observe that the use of “deep-learning-based” models can also be extended beyond the range of collaborative filtering, with possible empirical applications in the domain of “content-based” architectures and “hybrid” recommendation systems: in the long term, the adoption of a similar framework for recommendation engines can represent a further, significant and valuable innovation for the entire system of customized service solutions “3.0”.

Moreover, the study outcomes 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).

In fact, the service consumer behavior can also be analyzed as a dynamic process, instead of a static sequence of unrelated actions over a certain time frame (Jacoby et al., 1976): 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 (Hidasi et al., 2015) (Devoogh & Bersini, 2016).

## References

- Alt, R., March, Demirkan, H., Ehmke, J. F., Moen, A., & Winter, A. (2019). Smart services: The move to customer orientation. *Electron Markets*, 29, 1-6.
- Bashar, A. (2019). Survey on evolving deep learning neural network architectures. *Journal of Artificial Intelligence*, 1(02), 73-82.
- Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In the *Proceedings of the 15th AAAI Conference on Artificial Intelligence 1998*, Madison, Wisconsin, USA.
- Bhatnagar, V. (2016). *Collaborative Filtering Using Data Mining and Analysis*. IGI Global.
- Blöcher, K., & Alt, R. (2018). An approach for customer-centered smart service innovation based on customer data management. In the *Proceedings of the 9th International Conference on Exploring Service Science*, Karlsruhe, Germany.
- Bobadilla, J., Alonso, S., & Hernando, A. (2020). Deep learning architecture for collaborative filtering recommender systems. *Applied Sciences*, 10, 2441.
- Rizzo, G. L. C., Marco, M. D., Rosa, P. D., & Laura, L. (2020). Collaborative recommendations with deep feed-forward networks: An approach to service personalization. In the *Proceedings of the 10th International Conference on Exploring Service Science*, Porto, Portugal.
- Cotter, P., & Smyth, B. (2000). PTV: Intelligent personalized TV guides. In the *Proceedings of the 12th Conference on Innovative Applications of Artificial Intelligence 2000*, Austin, Texas, USA.
- Demirkan, H., Bess, C., Spohrer, J., Rayes, A., Allen, D., & Moghaddam, Y. (2015). Innovations with smart service systems: Analytics, big data, cognitive assistance, and the internet of everything. *Communications of the association for Information Systems*, 37(1), 35.
- Devoogh, R., & Bersini, H. (2016). Collaborative Filtering with Recurrent Neural Networks. *arXiv:1608.07400*.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Guo, C., & Berkhahn, F. (2016). Entity Embeddings of Categorical Variables. *arXiv:1604.06737*.
- Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4), 1-19.
- Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.
- Jacoby, J., Kohn Berning, C., & Szybillo, G. (1976). Time and consumer behavior: An interdisciplinary overview. *Journal of Consumer Research*, 2(4), 320-339.
- Khamparia, A., & Singh, K. M. (2019). A systematic review on deep learning architectures and applications. *Expert Systems*, 36(3), 1-12.
- Kingma, P. D., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980*.
- Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11-26.
- Meierhofer, J., & Meier, K. (2017). From data science to value creation. In the *Proceedings of the 8th International Conference on Exploring Service Science*, Rome, Italy.

- Melville, P., & Sindhvani, V. (2017). Recommender Systems. In: Sammut, C. & Webb, G. I. (Eds), *Encyclopedia of Machine Learning and Data Mining* (pp. 1056-1066). Springer, Boston, Massachusetts, USA.
- Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In the *Proceedings of the 18th AAAI Conference on Artificial Intelligence*, Edmonton, Alberta, Canada.
- Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzyan, A., Duffy, N., & Hodjat, B. (2019). Evolving deep neural networks. In *Artificial intelligence in the age of neural networks and brain computing* (pp. 293-312). Academic Press.
- Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. San Francisco, CA, USA. Determination Press.
- Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6), 393-408.
- Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2011). *Recommender Systems Handbook*. Boston, MA, USA. Springer Science+Business Media, LLC.
- Rosenblatt, F. (1959). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958
- Vassiliou, C., Stamoulis, D., & Martakos, D. (2006). A recommender system framework combining neural networks and collaborative filtering. In the *Proceedings of the 5th WSEAS International Conference on Instrumentation, Measurement, Circuits and Systems*, Hangzhou, China.
- Yadav, A., & Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: A review. *Artificial Intelligence Review*, 53(6), 4335-4385.
- Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1), 1-38.

## About the Authors

**Mr. Pasquale De Rosa** is a research scientist at the Travelling and Mobility (TaM) R&D group at the University of Geneva, where he conducts applied research on several ML and AI projects, covering their full lifecycle from the initial idea to an operational prototype. Before joining TaM, Pasquale De Rosa graduated in Data Science and Marketing from LUISS University of Rome in 2018, with a thesis on deep learning-based recommendation systems, and spent two years in a consulting firm where he gained the experience to handle with long and short-term industrial projects. He has been an active AIS member since 2020.

**Dr. Michel Deriaz**, after an engineer degree in telecommunications and a master in computer science, did a PhD in economical and social sciences. He spent then three years in the industry before coming back at University as lab director. Among his main realizations there are FoxyTag, a worldwide speed camera warning system with thousands of users, and FoxyTour, a museum and city guide that adapts automatically to the preferences of its user.

**Dr. Marco De Marco**, after having served 30 years at the Catholic University of Milan up to the top of the academic career, today is full professor of Organization and Information Systems at the International Telematic University Uninettuno in Rome. He is also dean of the faculty of economics. Marco De Marco is the author of five books that discuss the development of information systems, the computer industry, and the impact of technology on organizations, as well as the writer of several articles and essays. He is also a member of the editorial board of several journals. His major research interests are: systems development, e-government, banking information systems, IT and organizations. For his contribution to the discipline, he received in 2010 the award of AIS Fellow.

**Dr. Luigi Laura** is currently associate professor of Computer Science Engineering in the International Telematic University Uninettuno, where he teaches "Big Data Platforms" and "Theoretical Computer Science". Previously he lectured several courses in Sapienza, Tor Vergata and LUISS universities, where he has been the thesis advisor of more than one hundred students. Since 2020 he is the president of the committee for the Italian Olympiads in Informatics (OII) and since 2006 he is the trainer of the Italian team for the International Olympiads in Informatics (IOI).

Copyright © 2022 by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints, or via email from [publications@aisnet.org](mailto:publications@aisnet.org).