Usage of deep neural networks as recommendation engines. Review of selected approaches

Filip Wójcik
Wroclaw University of Economics and Business

CREATIVETIME
doktorant.com.pl

18 ZAGADNIENIA AKTUALNIE PORUSZANE PRZEZ MŁODYCH NAUKOWCÓW

Redaktor Wydania: Krzysztof Piech
AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, al. A. Mickiewicza 30, Kraków 30-059.

Skład tekstów i projekt graficzny okładki: Marcin Kuczera

Korekty: Krzysztof Piech, Marcin Kuczera

Opracowanie pt. ZAGADNIENIA AKTUALNIE PORUSZANE PRZEZ MŁODYCH NAUKOWCÓW 18 zawiera recenzowane prace naukowe Młodych Naukowców współpracujących z CreativeTime. Wybrane prace zostały nadesłane przez poszczególnych Uczestników Konferencji Młodych Naukowców nt. NOWE TRENDY W BADANIACH NAUKOWYCH - WYSTĄPIENIE MŁODEGO NAUKOWCA - I edycja – 20-21.06.2020 i II Edycja – 20-22.11.2020, Konferencji Młodych Naukowców nt. NOWE WYZWANIA DLA POLSKIEJ NAUKI - VII Edycja – 5-7.09.2020 oraz Konferencji Młodych Naukowców nt. ANALIZA ZAGADNIENIA, ANALIZA WYNIKÓW - WYSTĄPIENIE MŁODEGO NAUKOWCA – II edycja cd – 17-18.10.2020.

Skład opracowania wykonano na podstawie dostarczonych przez autorów tekstów. Wszystkie artykuły zostały opublikowane na odpowiedzialność ich autorów. Za treść odpowiadają autorzy poszczególnych tekstów.

ISBN: 978-83-66772-02-1

Opracowanie

Niniejsza książka elektroniczna DVD ma służyć młodym naukowcom. Propagujemy podejmowane działania wśród młodych naukowców, wiedzę, innowacyjne badania oraz rozwój nauki. Nauka musi charakteryzować się ciągłym rozwojem. Dzisiejsi naukowcy korzystają z coraz to nowocześniejszych metod badawczych, prowadzą różnego rodzaju projekty, których efekty w nieodległej przyszłości mają służyć całej społeczności i otaczającemu nas środowisku. Niniejsze opracowanie zawiera zbiór zagadnień prezentujących zainteresowania naukowe młodych adeptów nauki.

Młody naukowiec

Absolwenci studiów drugiego stopnia coraz częściej podejmują decyzję o rozpoczęciu studiów doktoranckich. Decyzja ta często podyktowana jest chęcią pozostania na uczelni w charakterze naukowca i wykładowcy. Niestety po otrzymaniu dyplomu doktora nauk tylko część młodych naukowców pozostanie na uczelni macierzystej. Część młodych doktorów zasili inne uczelnie i jednostki naukowe, a zdecydowana większość rozpocznie kolejny etap swojego życia w instytucjach państwowych i firmach prywatnych. Dlatego też obok realizacji własnych badań naukowych i pisania pracy, doktoranci powinni podjąć wszelkie możliwe działania zmierzające do nawiązania współpracy z firmami prywatnymi, aby realizować dalszą karierę zawodową. Włączanie się doktorantów w różnego rodzaju projekty międzyuczelniane, współpracę w modelu naukowiec--firma, udział we wszelkich konferencjach i szkoleniach o charakterze biznesowo-naukowym zwiększa szanse doktorantów na rozwój naukowy i zawodowy, a przede wszystkim może przynieść upragnioną satysfakcję.

Młodzi naukowcy, którzy pozostali na uczelni wyższej w charakterze często asystenta, adiunkta mają również wiele możliwości nawiązania współpracy ze stale rozwijającym się polskim biznesem. Należy zastanowić się, w jaki sposób przenieść własne dokonania i pomysły naukowe do realizacji w biznesie.

Biznes

Niewątpliwie szansą dla biznesu są innowacje, które niosą ze sobą między innymi młodzi naukowcy. Każdy dobry biznesmen powinien zdać sobie sprawę, że nie ma innowacji bez nowych pomysłów i badań naukowych.

Sami spróbujmy zachęcić właścicieli polskich firm, osoby decyzyjne, menedżerów do nawiązywania współpracy z nami - Młodymi Naukowcami.

Wydawca:
Wydawca: CREATIVETIME, www.creativetime.pl
biuro@creativetime.pl
Skrytka Pocztowa nr 92, 30-093 Kraków 23

Nakład 55 egzemplarzy

Wydanie ISBN

ZAGADNIENIA AKTUALNIE PORUSZANE PRZEZ MŁODYCH NAUKOWCÓW 18

Wydawca: CREATIVETIME

Kraków 2021

USAGE OF DEEP NEURAL NETWORKS AS RECOMMENDATION ENGINES. REVIEW OF SELECTED APPROACHES

Filip Wójcik

Abstract: This paper reviews selected neural network architectures used in the context of e-commerce recommendation engines. Subsequent approaches are presented in order from the most straightforward applications to the latest advancements. It compares the models in terms of generalization capabilities as well as the possibility to utilize external features. The formalization of the essential learning procedures is presented, as well as the derivation of equivalence between autoencoder neural networks and the matrix factorization approach. In conclusion - modern deep learning architectures composed of the autoencoder component and dense low-dimensional features (called encodings) have much stronger predictive power and can replicate, without the loss of generality, the behavior of matrix-factorization approaches.

Keywords: deep learning, neural networks, machine learning, recommendation systems

1. Introduction

International market digitalization changed the way how e-commerce companies operate and acquire new customers or users. During the last decade, traditional marketing was slowly replaced by personalized, user-specific targetting, intended to proactively identify potential new clients (Brynjolfsson & McAfee, 2014). Automated solutions allowed companies to speed up the whole process and make it more flexible and reactive when faced with an unstable and ever-shifting market environment.

Simultaneously, in the face of the growing popularity of deep neural networks, new use cases for that family of algorithms emerged. Initially intended to be used as classification, regression, or image recognition tools, they started to be used in many different fields, including recommendation engines. The following sections present taxonomy and analysis of selected neural network architectures to recommend new products or services.

2. Methods and materials

Early approaches

Early approaches to use neural networks as recommendation engines utilized the so-called "content-based" or "context-aware" recommendation paradigm. A system was presented with explicit features of each user-item combination, and its task was to predict rating or classify it as like/dislike. Such architectures were often used as plug-ins supporting other methods and acting as, e.g., meta-classifiers, combining predictions of other methods (Biancalana et al., 2011). Exemplification of such architecture is presented below:

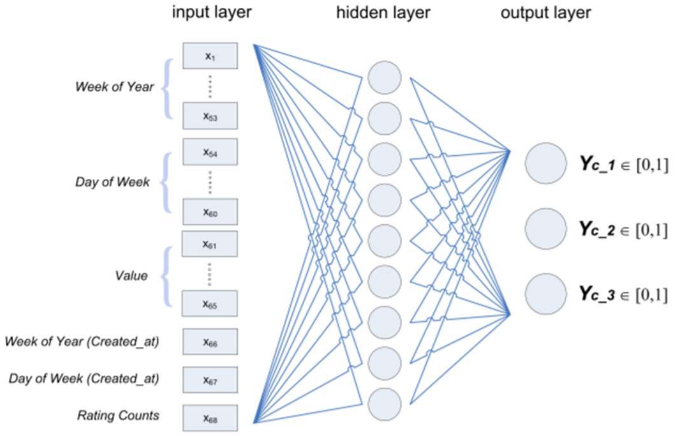

Fig.1: Neural ensemble
Source: (Biancalana et al., 2011)

While such use of neural networks as recommendation engines is correct, it can still be categorized as a "typical" classification/regression problem, not utilizing the full generalization capabilities of deep learning.

Another approach was proposed in the publication "Restricted Boltzmann machines for collaborative filtering" (Salakhutdinov et al., 2007). The solution proposed in that paper (so-called RBM's) can be described as a network of symmetrically connected units (similar to neurons) divided into the visible and the hidden layer. There are no connections amongst units within the same layer. Such "neurons" make stochastic decisions, whether to fire or not. Their behavior is modeled through a process similar to free-energy formulation in physics. The picture below presents a schematic of RBMs architecture:

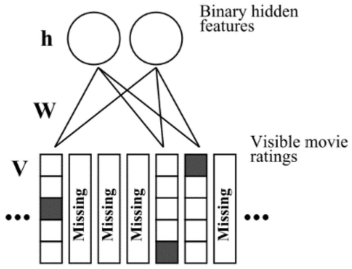

Fig.2: Restricted Boltzman Machine
Source**:** (Salakhutdinov et al., 2007)

The main mathematical formalization elements are the following:

$$p(h_j = 1|\boldsymbol{V}) = \sigma\left(b_j + \sum_{i=1}^{m}\sum_{k=1}^{K} v_i^k W_{ij}^k\right) \quad [1]$$

with:
h_j activation of j-th neuron;
$p(\cdot)$ probability of j-th neuron activation given an energy function;
$\sigma(\cdot)$ sigmoid activation function;
$\boldsymbol{V}$ matrix of visible ratings/reviews;
W_{ij}^k a symmetric interaction parameter between hidden neuron j and rating k of movie i.
Source**:** (Salakhutdinov et al., 2007)

In this context, RBMs can be considered a generative, probabilistic model. Two main drawbacks made that class of models less popular than other approaches. The first one is that RBMs can only model binary data (due to the probabilistic output character) (Salakhutdinov et al., 2007; Sedhain et al., 2015). The second one is computational complexity due to Monte Carlo Sampling and infeasible log-likelihood computation (Zhang et al., 2018).

Autoencoders

AutoRec algorithm, published in 2015 (Sedhain et al., 2015), significantly improved over existing implementations. This approach transferred the knowledge from the image recognition field directly into recommendation systems to mimic the behavior of a classic approach called "latent factors model" or "collaborative filtering based on matrix factorization" (Schafer et al., 2007). One of the classic approaches, called "matrix factorization" (or latent factors model), was based on matrix decomposition techniques. While implementation details varied between papers (Canny, 2002; Goldberg et al., 2001), the general idea remained the same - given a large (customer x item) rating matrix, k-latent (hidden) factors were selected, such that k is much lower than initial dimensionality. The general formulation, capturing the idea of almost all approaches from this family of models, can be described with the following mathematical formulation (based on (Koren et al., 2009)):

Given a u – as a number of users, i – as a number of items, R as a matrix of ratings, such that $R \in \mathbb{R}^{u \times i}$, a classic matrix decomposition approach aims to find f – "hidden" (latent) factors, capable of minimizing reconstruction error, interpreted as unknown user taste's and unknown "features" of items.

$$\hat{R}_{mf} = \boldsymbol{PQ^T} + \boldsymbol{b} \quad [2]$$

with
$\hat{R}_{mf}$ reconstructed ratings matrix, $\hat{R}_{mf} \in \mathbb{R}^{u \times i}$;
$\boldsymbol{Q}$ projected (simplified) matrix of items "described by" hidden features, $Q \in \mathbb{R}^{i \times f}$;
$\boldsymbol{P}$ projected (simplified) matrix of users, "described by" hidden features, $P \in \mathbb{R}^{u \times f}$.
$\boldsymbol{b}$ bias, usually depicted as a vector for user $b_u \in \mathbb{R}^{u \times 1}$, a vector for an item $b_i \in \mathbb{R}^{i \times f}$ or as a combined matrix, where $b_{ui} \in \mathbb{R}$, $\boldsymbol{b} \in \mathbb{R}^{u \times i}$, sometimes with global rating average μ added

Reconstruction is achieved by minimizing reconstruction error (usually Mean Sequared Error MSE):

$$\min_{Q,P} \frac{1}{N}\left(R - (PQ^T + b)\right)^T\left(R - (PQ^T + b)\right) = \frac{1}{N}\left(R - \hat{R}_{mf}\right)^T\left(R - \hat{R}_{mf}\right) = \frac{1}{N}\, \boldsymbol{e}^T\boldsymbol{e} + \lambda\left(\|Q\|_F{}^2 + \|\, P\|_F^2 + \|b\|_F^2\right) \quad [3]$$

with

$\hat{R}_{mf}$ reconstructed ratings matrix, $\hat{R}_{mf} \in \mathbb{R}^{u\times i}$ as in [2];
$\boldsymbol{Q}$ projected (simplified) matrix of items “described by” hidden features, $Q \in \mathbb{R}^{i\times f}$;
$\boldsymbol{P}$ projected (simplified) matrix of users, “described by” hidden features, $P \in \mathbb{R}^{u\times f}$.
$\boldsymbol{b}$ bias for a user and item combined
e error matrix (difference between observed ratings and reconstructed ratings);
$\boldsymbol{\lambda}$ regularization rate
$||\boldsymbol{x}||_{\boldsymbol{F}}$ Frobenius norm for matrix.
Source: own work on based on (Koren et al., 2009)

The minimization described by equation [3], typically can be achieved via iterative (stochastic) gradient descent, which is also a standard way of training neural networks. In that setting, partial matrix updates are defined as: (uxi) * (ixf)

$$Q \leftarrow Q + \gamma(\boldsymbol{e} \cdot \boldsymbol{P} - \lambda \quad) \qquad [4]$$
$$P \leftarrow P + \gamma(\boldsymbol{e} \cdot \boldsymbol{Q} - \lambda \quad)$$

with
$\boldsymbol{Q}$ projected (simplified) matrix of items “described by” hidden features, $Q \in \mathbb{R}^{i\times f}$;
$\boldsymbol{P}$ projected (simplified) matrix of users, “described by” hidden features, $P \in \mathbb{R}^{u\times f}$.
e error matrix (difference between observed ratings and reconstructed ratings) as in equation [3];
$\boldsymbol{\lambda}$ regularization rate;
$\boldsymbol{\gamma}$ learning rate
Source: own work on based on (Koren et al., 2009)

AutoRec, proposed in a publication (Sedhain et al., 2015), could replicate that behavior, extending it further with an opportunity to add more dense neural layers, introducing complicated nonlinearities. The architecture of the model can be visualized as follows:

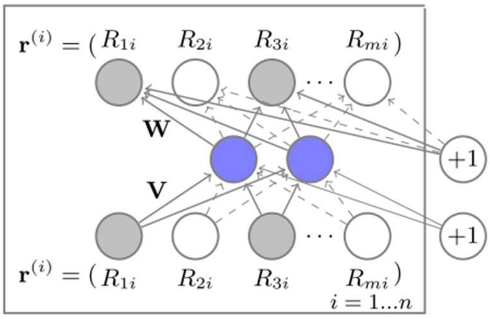
$\mathbf{r}^{(i)} = (R_{1i} \quad R_{2i} \quad R_{3i} \quad R_{mi})$
W
V
+1
+1
$\mathbf{r}^{(i)} = (R_{1i} \quad R_{2i} \quad R_{3i} \quad R_{mi})$
$i = 1...n$

Fig.3: AutoRec
Source: (Sedhain et al., 2015)

With the linear activation function, stochastic gradient descent training scheme and regularization of autoencoder weights, it can be proven that AutoRec exactly replicates the simplified matrix factorization approach:

$$h(\boldsymbol{r};\boldsymbol{\theta}) = f(W \cdot g(Vr + \mu) + b) \qquad [5]$$

with
$\boldsymbol{r}$ an observable (actual) matrix of ratings r $\in \mathbb{R}^{u\times i}$;
$\boldsymbol{\theta}$ symbolic representation of an autoencoder in parametric form;
V matrix of weights mapping visible ratings matrix into latent space with k dimensions, $V \in \mathbb{R}^{k\times i}$;
$\boldsymbol{\mu}, \boldsymbol{b}$ global rating average/user and item combined bias $\mu, b \in \mathbb{R}^{u\times i}$;
$\boldsymbol{W}$ matrix of weights used for reconstruction, $W \in \mathbb{R}^{u\times k}$;
f, g activation functions for encoder/decoder respectively, preferably nonlinear.

Source: (Sedhain et al., 2015)

Reconstruction is achieved by minimizing reconstruction error (usually Mean Squared Error MSE), like in matrix factorization approach, formalized by [3] and [4].

It is visible that (given the identity activation functions *f,g,* such that $f(x) = x$ and $g(x) = x$), the matrix $(Vr + \mu) \in \mathbb{R}^{k\times i}$ used in [5] can be treated as similar to the matrix $Q^T \in \mathbb{R}^{f\times i}$ used in [2] for matrix factorization. Simultaneously, AutoRec's reconstruction weights $W \in \mathbb{R}^{u \times k}$ used in [5] can be treated as similar to the matrix $P \in \mathbb{R}^{u \times f}$ used in [2] for matrix factorization. After such substitutions, the only differences can be attributed to notation conventions, but the reconstruction mechanism and training policy remain the same for both models, which can be formalized as follows:

$$h(\boldsymbol{r};\boldsymbol{\theta}) = f(W \cdot g(Vr + \mu) + b) = f(W \cdot g(Q^T) + b) = f(P \cdot g(Q^T) + b) = PQ^T + b = \hat{R}_{mf} \ \blacksquare \qquad [6]$$

with symbols as in previous equations [5], [4], [3], [2], [1].
Source: own work.

The derivation presented in [6] demonstrates that AutoRec can replicate a matrix factorization approach. Extensions are possible mainly due to the usage of nonlinear activation functions *f, g.*

This type of model can be compared to RBMs discussed in a previous section. In that context, AutoRec can be thought of as a discriminative model that minimizes reconstruction error instead of maximizing (intractable) log-likelihood (Sedhain et al., 2015). Additionally, AutoRec can be trained with a much faster gradient descent and backpropagation method than contrastive divergence used in RBM. Additionally, its flexible architecture (no limitation on the number of hidden nodes and layers) makes it easily extensible.

AutoRec extensions

A natural extension to AutoRec was the addition of more dense layers to a shallow neural network. A new algorithm, called DeepRec (Kuchaiev & Ginsburg, 2017), was composed of two deep networks - encoder and decoder, replicating the behavior of AutoRec, but with more nonlinear activations, increasing the expressiveness of the whole solution. The architecture proposed in a research paper is presented below:

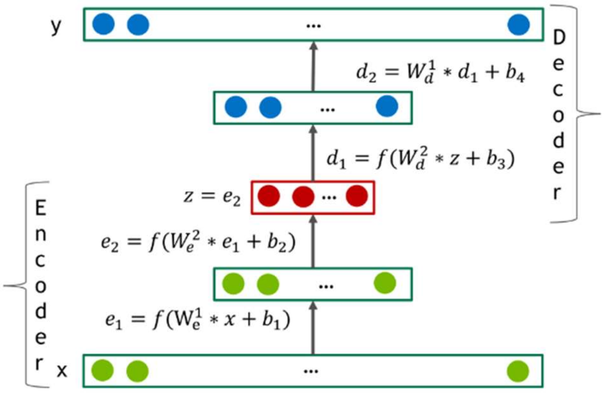
y
$d_2 = W_d^1 * d_1 + b_4$
Decoder
$d_1 = f(W_d^2 * z + b_3)$
$z = e_2$
Encoder
$e_2 = f(W_e^2 * e_1 + b_2)$
$e_1 = f(W_e^1 * x + b_1)$
x

Fig.4: DeepRec architecture
Source: (Kuchaiev & Ginsburg, 2017)

Apart from a more complex structure, the DeepRec training loop addressed sparsity in the user-item matrix. In most cases, the rating matrix is very sparse - most of the users do not rate the majority of items. That circumstance makes training harder, as the algorithm has less information available. For DeepRec, authors proposed a procedure called "dense re-feeding" - rating matrix reconstructed by a model is dense by nature, and such a dense matrix is fed again in every iteration, as a training input, and compared against the original matrix. Such an augmentation gives a substantial advantage over competing solutions, suffering from sparsity and lack of training material.

Several other research papers extended the idea of the autoencoder as a recommendation system. The most important are Denoising Collaborative Autoencoders (Wu et al., 2016), which utilizes negative sampling technique for better generalization, or Variational Autoencoders (Liang et al., 2018), based on Bayesian inference.

Hybrid approaches

The last family of deep learning models used as recommendation engines can be labeled as "the hybrid approach." Most of the time, models belonging to that family combine matrix-factorization with classic feature-based reasoning, where the algorithm learns correlations between product/customer attributes and assigned ratings. "Wide and Deep Learning for Recommender Systems" (Cheng et al., 2016) is one of such solutions. It exploits two fundamental properties of machine learning systems - memorization and generalization. Memoization is a capability to learn frequent co-occurrences or correlations present in historical data. On the other hand, generalization can extend knowledge about existing correlations and explore new possibilities. The former can be achieved by the generalized linear model, tuned using binarized, sparse features. The latter, by the deep learning model, capable of identifying higher-order co-occurrences. The model architecture described in a publication was depicted in the figure below:

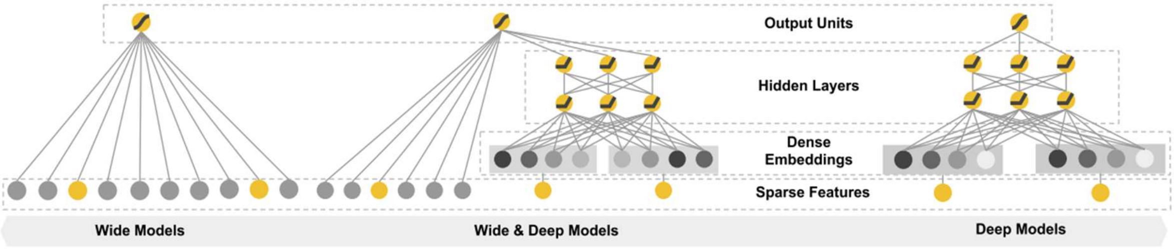

Fig.5: Wide & Deep Architecture
Source: (Cheng et al., 2016)

The "wide component" is a shallow linear model, replicating a simple rating matrix reconstruction behavior. It utilizes both - single binary features and their interactions. The "deep component" is a deep feed-forward neural network, fed with sparse high-dimensional features. These features are first converted into low-dimensional dense, real-valued vectors, called "the embeddings" (Cheng et al., 2016). Subsequent layers of the network with nonlinear activations learn higher-order representations for enhanced generalization.

The sigmoid activation function combines wide and deep parts for binary prediction (e.g., like/dislike). Formalization of this procedure is presented below:

$$P(Y = 1|x) = \sigma\left(\boldsymbol{w}_{\boldsymbol{wide}}^{T}[\boldsymbol{x}, \phi(\boldsymbol{x})] + \boldsymbol{w}_{\boldsymbol{deep}}^{T} a^{(l_f)} + \boldsymbol{b}\right) \qquad [7]$$

with

Y	binary target label (like/dislike or recommend/do not recommend);
$\boldsymbol{w_{wide}}, \boldsymbol{w_{deep}}$	weights of wide and deep components respectively;
$\boldsymbol{x}$	input ratings matrix;
$\boldsymbol{\phi}(\boldsymbol{x})$	Cross product transformations of the original matrix x;
$\boldsymbol{b}$	bias term;
$\boldsymbol{a}^{(l_f)}$	final layer activation weights;
$\sigma(\cdot)$	Sigmoid activation function.

Source: (Cheng et al., 2016)

There are several other modifications of the architecture described above. Deep Factorization Machine (Guo et al., 2017) simplifies the feature engineering process by adding dot-product multiplication and addition layers, which serve higher-order feature generation purposes. The schematic of such a network is presented below:

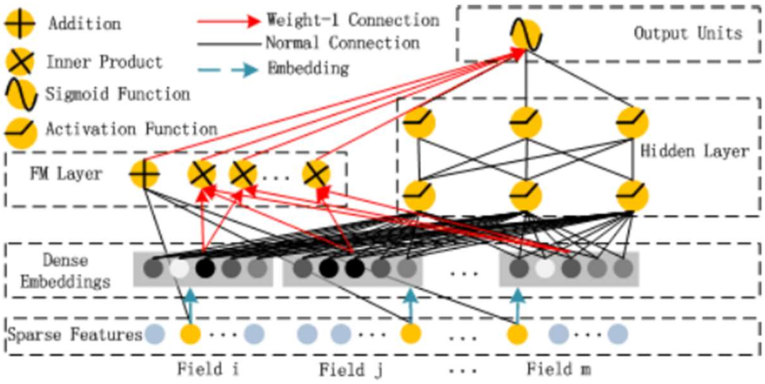

Fig. 6. Deep Factorization Machine architecture
Source: (Guo et al., 2017)

That architecture was enriched with Compressed Interaction Network CIN (Lian et al., 2018), with further generalization improvement and easier integration with item/user features.

Other approaches, like Deep Hybrid Collaborative Filtering with Content (Wójcik & Górnik, 2020), explicitly combine user-item interaction embedding as dense real-values vectors, with an additional item attributes information (also encoded in the form of embeddings), used as an external source of the data for the neural network.

3. Results

Publications describing the aforementioned neural network architectures were published at different points in time, and authors do not always refer directly to competing approaches when reporting experimental results. Additionally, the source code is not always available or actively maintained. Probably the most recent and complete quantitative comparative study is a part of "xDeepFM" paper (Lian et al., 2018), where the authors compared on three datasets their proposed hybrid solution (The Extreme Deep Factorization Machine) with:

- Logistic regression (as a baseline model);
- Plain deep neural network;
- Wide & Deep Network;
- Factorization Machine;
- Deep Factorization Machine.

The shortened comparison results from this publication are presented in a table below.

Tab.1: Shortened algorithms performance comparison on three datasets

Dataset	Criteo		Dianping		Bing News	
Model name\metric	AUC	Logloss	AUC	Logloss	AUC	Logloss
LR	0.7577	0.4854	0.8018	0.3608	0.7988	0.2950
FM	0.7900	0.4592	0.8165	0.3558	0.8223	0.2779
DNN	0.7993	0.4491	0.8318	0.3382	0.8366	0.2730
DCN	0.8026	0.4470	0.8391	0.3379	0.8379	0.2677
Wide & Deep	0.8000	0.4490	0.8361	0.3364	0.8377	0.2668
PNN	0.8038	0.4927	0.8445	0.3424	0.8321	0.2775
DeepFM	0.8025	0.4468	0.8481	0.3333	0.8376	0.2671
xDeepFM	0.8052	0.4418	0.8639	0.3156	0.8400	0.2649

Source: (Lian et al., 2018)

The authors reported that all differences between the models in question were found to be statistically significant.

4. Discussion

There are a couple of conclusions that can be drawn from it:

- All models based on deep neural networks perform better than baselines, "classic" approaches like linear regression or matrix factorization.
- As the models go deeper and utilize more advanced user/item attributes, as well as their interactions - the better scores they achieve.
- More complex deep neural network models are flexible enough to use richer feature data than straightforward approaches.

These conclusions are aligned with the intuition, following the "universal approximation theorem" (Hornik et al., 1989) applied to deep neural networks. Descriptive comparison of features and capabilities of model families described above are presented in a table below:

Tab.2: Deep neural networks as recommendation engines comparison

Model family\features	Generalization	Nonlinearity	Flexible architecture	Utilization of external features
Matrix factorization	Moderate	Moderate	No	No
Autoencoders, Deep autoencoders	High	High, depending on selected architecture	Moderate, possibility to choose network depth	No
Hybrid Networks (Wide&Deep, Deep Factorization Machines, etc.)	High	High	High, wide and deep layers can be combined	Yes, via utilization of feature encoding

Source: own work

One should keep in mind that there is no single, optimal neural network architecture that will address all the issues in recommendation tasks. Every problem is unique and requires proper tuning and model design. Nevertheless, the fact that autoencoder-based neural networks can replicate matrix factorization and collaborative filtering operation without any loss of generality, one can conclude that this family of algorithms will eventually replace "classic" training methods. Hybrid approaches, which utilize external feature information (if available), are their natural extension, capable of delivering significantly better results.

The flexibility of neural network architecture design is promising in terms of future enhancements, as this family of models can operate with, e.g., time-series data, sequential data, images, audio, etc.

5. Conslusions

Presented research results seem to be indicators of a significant trend in recommendation systems design - namely, the comprehensive utilization of neural networks. Without any loss of generality, they can replace matrix factorization and Collaborative Filtering approaches, extending them with additional capabilities like external features utilization and implicit interactions.

6. Literature

Biancalana, C., Gasparetti, F., Micarelli, A., Miola, A., & Sansonetti, G. (2011). Context-aware movie recommendation based on signal processing and machine learning. In Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation (pp. 5–10).

Brynjolfsson, E., & McAfee, A. (2014). The second machine age: Work, progress, and prosperity in a time of brilliant technologies. WW Norton & Company.

Canny, J. (2002). Collaborative filtering with privacy via factor analysis. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 238–245.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., & others. (2016). Wide & deep learning for recommender systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 7–10.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval, 4(2), 133–151.

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: a factorization-machine based neural network for CTR prediction. ArXiv Preprint ArXiv:1703.04247.

Hornik, K., Stinchcombe, M., White, H., & others. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359–366.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30–37.

Kuchaiev, O., & Ginsburg, B. (2017). Training deep autoencoders for collaborative filtering. ArXiv Preprint ArXiv:1708.01715.

Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., & Sun, G. (2018). xdeepfm: Combining explicit and implicit feature interactions for recommender systems. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1754–1763.

Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018). Variational autoencoders for collaborative filtering. Proceedings of the 2018 World Wide Web Conference, 689–698.

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. Proceedings of the 24th International Conference on Machine Learning, 791–798.

Schafer, J., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative Filtering Recommender Systems. In The Adaptive Web (Vol. 4321, pp. 291–324)

Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. Proceedings of the 24th International Conference on World Wide Web, 111–112.

Wójcik, F., & Górnik, M. (2020). Improvement of e-commerce recommendation systems with deep hybrid collaborative filtering with content: A case study. Econometrics, 24(3), 37–50.

Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. (2016). Collaborative denoising auto-encoders for top-n recommender systems. Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, 153–162.

Zhang, N., Ding, S., Zhang, J., & Xue, Y. (2018). An overview on Restricted Boltzmann Machines. Neurocomputing, 275, 1186–1199

Institution: Wroclaw University of Economics and Business, Faculty of Management Sciences
Research supervisor: dr hab. Iwona Chomiak-Orsa, prof. UE
Correspondence address: filip.wojcik@ue.wroc.pl