Weave&Rec : A Word Embedding based 3-D Convolutional Network for News Recommendation

Dhruv Khattar, Vaibhav Kumar*
International Institute of Information Technology
Hyderabad, India
{dhruv.khattar,vaibhav.kumar}@research.iiit.ac.in

Vasudeva Varma, Manish Gupta†
International Institute of Information Technology
Hyderabad, India
{vv,manish.gupta}@iiit.ac.in

Abstract

An effective news recommendation system should harness the historical information of the user based on her interactions as well as the content of the articles. In this paper we propose a novel deep learning model for news recommendation which utilizes the content of the news articles as well as the sequence in which the articles were read by the user. To model both of these information, which are essentially of different types, we propose a simple yet effective architecture which utilizes a 3-dimensional Convolutional Neural Network which takes the word embeddings of the articles present in the user history as its input. Using such a method endows the model with the capability to automatically learn spatial (features of a particular article) as well as temporal features (features across articles read by a user) which signify the interest of the user. At test time, we use this in combination with a 2-dimensional Convolutional Neural Network for recommending articles to users. On a real-world dataset our method outperformed strong baselines which also model the news recommendation problem using neural networks.

Keywords

Convolutional Neural Networks, News Recommendation

ACM Reference Format:
Dhruv Khattar, Vaibhav Kumar and Vasudeva Varma, Manish Gupta. 2018. Weave&Rec : A Word Embedding based 3-D Convolutional Network for News Recommendation. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269307

1 Introduction

Readers have access to a large quantity and variety of fresh news online. To solve the problem of information overload, it becomes desirable to have a recommender system that would point a user to the most relevant news and thus would maximize the user engagement with the site and minimize the time for finding relevant content.

A popular approach to the task of recommendation is called collaborative filtering which uses the user's past interaction to predict the most relevant content [11]. Another common approach is content-based recommendation, which uses features between items and/or users to recommend new items to the users based on the similarity between features. However, amongst the various approaches for collaborative filtering, matrix factorization is the most popular one, which projects users and items into a shared latent space, using a vector of latent features to represent a user or an item. Thereafter, a user's interaction with an item is modeled as the inner product of their latent vectors. Recent approaches for news recommendation are based on recurrent neural networks [8] which utilize the users' historical data to come up with better recommendations.

Each recommendation scenario comes with its own challenges. Methods based on matrix factorization for news recommendation which are solely based on implicit feedback are not suitable for news recommendation because of the large number of novel articles published each day. On the other hand content based methods suffer from the problem of over-specialization and also rely on certain heuristics for user-profile creation which may not be very effective in capturing the interests of the user.

In this paper, we come up with a deep learning model which utilizes the content of the articles and also takes the users' historical data into account in order to make better recommendations. We call our model Weave&Rec as we try to weave two different sorts of information together for providing recommendations. Weave&Rec tries to answer the following two questions: **Q.1** What is it that attracted the users' attention to this particular article? **Q.2** Can we infer the the reading pattern of the user based on the articles read by her in the past?

We divide Weave&Rec into two components. The first component of the model is based on a 3-dimensional convolutional neural network (CNN) which takes the word embeddings of the articles present in the user reading history as its inputs. 3D CNNs [7] have successfully been used for the task of action recognition where capturing spatial and temporal information is very important. In our case, applying 3D convolution helps us to identify both the spatial information (features of a particular article) as well as the temporal information (features present in the sequence of articles read by the user) which are pertinent to a user's interest without delving into the task of manual feature engineering. The second component of the model utilizes 2D CNN and takes the word embeddings of the

*Author had equal contribution. He can also be contacted at vaibhav2@andrew.cmu.edu
†Author is also a Principal Applied Researcher at Microsoft.

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269307

test article as its input. We then model the interaction between the outputs obtained from the two components using the Hadamard product. We evaluate our model over a real-world dataset and show that our model outperforms the state-of-the-art including the ones which are based on neural networks which have been shown to be suitable for this task.

Although a lot of work has been done using Recurrent Neural Networks for this task, none have attempted to utilize the capabilities of the 3D CNNs. With this paper, we also attempt to collaterally show the effectiveness of 3D CNNs for this task.

2 Related Work

There has been extensive study on recommendation systems with a myriad of publications.

2.1 Traditional Recommendation Systems

Recommendation systems in general can be divided into collaborative filtering based systems and content based systems. In collaborative filtering, an item is recommended to a user if similar users liked that item. Examples of such techniques include Bayesian matrix factorization [13], matrix completion [12], Restricted Boltzmann Machines (RBMs) [14], nearest neighbor modeling [1] etc. Another common approach for recommendations is content-based recommendations. In this approach, features from user's profile and/or items are extracted and are used for recommending items to users. The underlying assumption is that the users tend to like items similar to those they already liked.

2.2 Neural Network-based Recommendation Systems

There has been some work on exploring neural networks for recommendation systems. In [14], a two-layer RBM was used to model users' explicit ratings on items. Recently, auto-encoders have become a popular choice for building recommendation systems [2, 16, 17]. AutoRec [16] learns hidden structures that can reconstruct a user's ratings given her historical ratings as inputs. In terms of user personalization, this approach is similar to the item-item models [10, 15] that represent a user using features of her rated items. While previous work has focused on modeling CF, they have modeled the *observed* ratings data only. As a result, they can easily fail to learn users preference from the positive-only implicit data. Deep Semantic Structured Model (DSSM) [6] which was originally used for ranking web documents has been extended to recommendation scenarios in [3], where the first neural network contains user's query history and the second neural network contains implicit feedback on items. The resulting model is named multi-view DNN. However, it is not directly adaptable for news recommendation because it requires a lot of information outside the news domain. In [18] a collaborative denoising auto-encoder (CDAE) for CF with implicit feedback is presented. In contrast to the DAE-based CF [17], CDAE additionally plugs a user node to the input of auto-encoders for reconstructing the user's ratings. While CDAE is solely based on item-item interaction, our proposed model is based on user-item interactions. The Neural Collaborative Filtering (NCF) model [4] replaces the user-item inner product with a multi-layer perceptron architecture that can learn an arbitrary function from the given data which can then be used for generating recommendations.

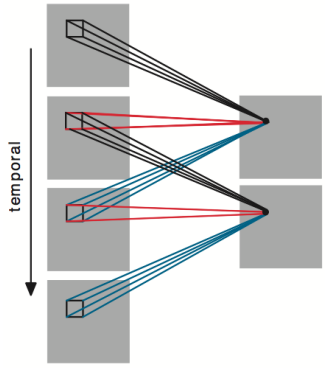

Figure 1: An illustration of 3D Convolution. Size of the Convolutional kernel is 3, and the set of weights are color coded so that shared weights are in the same color.

2.3 3D Convolutional Neural Networks

In [7], authors develop a novel 3-D CNN model for action recognition. Unlike the 2-D CNN, in which convolutions are applied over the 2-D feature map, this model extracts features from both spatial and temporal dimensions by performing 3-D convolutions, thereby capturing the motion information encoded in multiple adjacent frames of a video. The developed model generates multiple channels of information from the input frames, and the final feature representation is obtained by combining information from all channels. An example of 3D Convolutions can be seen in Figure 1.

Although the model is very useful in capturing temporal patterns, it has not been utilized in the news recommendation scenario so far. In this paper, we adapt this model for capturing the temporal interests from the user's reading behaviour and make recommendations accordingly.

3 Proposed Model

3.1 Word Embeddings

We combine the title and text of the news articles in our training sample and then learn a word2vec representation for each word. We set the dimension of the vectors to be 300.

3.2 Model Architecture

We divide the architecture into two components as shown in Figure 2. For the first component (user-history component), we choose a specific amount of reading history R for each user. For each article present in the user reading history, we create a 2D matrix. This 2D matrix comprises of the word2vec representations of the first 50 words of the news article. Doing so for each news article in the reading history gives us a series of matrices over which we apply 3D convolutions alternating it with pooling layers. We then flatten the outputs. For the second component, we create a 2D matrix for the test sample, i.e., the item that is to be considered for recommendation. We then apply 2D convolutions and alternate it with pooling layers and finally flatten the outputs. We then perform an

element-wise product between these outputs from the two components. Finally, we apply a fully connected layer with 128 hidden units followed by a logistic unit.

Formally in a 3D CNN, the value at position (x, y, z) on the j^{th} feature map of the i^{th} layer is given by,

$$v_{ij}^{xyz} = \tanh(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{pqr}^{ijm} x_{(i-1)m}^{(x+p)(y+q)(z+r)}) \quad (1)$$

where R_i is the size of the 3D kernel along the temporal dimension, w_{pqr}^{ijm} is the $(p, q, r)^{\text{th}}$ value of the kernel connected to the m^{th} feature map in the previous layer.

3.3 Training

Typically in matrix factorization, to learn the model parameters, existing pointwise methods perform regression with a squared loss. This is based on the assumption that observations are generated from a Gaussian distribution. The final output layer has the predicted score $y_{\hat{ux}}$, and training is performed by minimizing the pointwise loss between $y_{\hat{ux}}$ and its target value y_{ux}. Considering the one-class nature of implicit feedback, we can view the value of y_{ux} as a label 1 meaning the item x is relevant to a user u, and 0 otherwise. The prediction score $y_{\hat{ux}}$ then represents how likely an item x is relevant to u. Hence in order to constrain the values between 0 and 1 we use the logistic function. We then define the likelihood function as,

$$p(\gamma, \gamma^{-}|M, \Theta_m) = \prod_{(u,i)\in\gamma} y_{\hat{ui}} \prod_{(u,j)\in\gamma^{-}} (1 - y_{\hat{uj}}) \quad (2)$$

where γ, γ^{-} represent the positive (observed interactions) and negative (unobserved interactions) samples/items, M represents the similarity tensor and Θ_m represents the parameters of the model. Taking the negative log likelihood we get,

$$L = -\sum_{u,i\in\gamma\cup\gamma^{-}} y_{ui} \log y_{\hat{ui}} + (1 - y_{ui})(1 - \log y_{\hat{ui}}) \quad (3)$$

This is the objective function to minimize, and its optimization can be done by performing stochastic gradient descent (SGD). Careful readers might have realized that it is the same as the binary cross-entropy loss. Overall, by employing a probabilistic treatment, we address recommendation with implicit feedback as a binary classification problem. This is similar to what has been done in [4].

4 Experiments

4.1 Dataset

For this work we use the dataset published by CLEF NewsREEL 2017. As a part of their evaluation for offline setting, CLEF shared a dataset which captures interactions between users and news stories. The dataset includes information like the title and text of each news articles. For our task we considered users who had read more than 10 news articles (22229 users). We make the code publicly available[1].

[1]https://github.com/dhruvkhattar/WE3CN

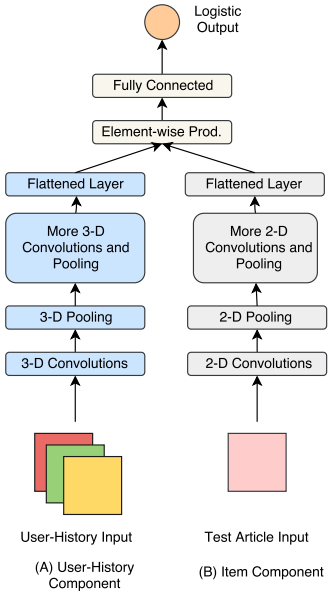

Figure 2: Model Architecture

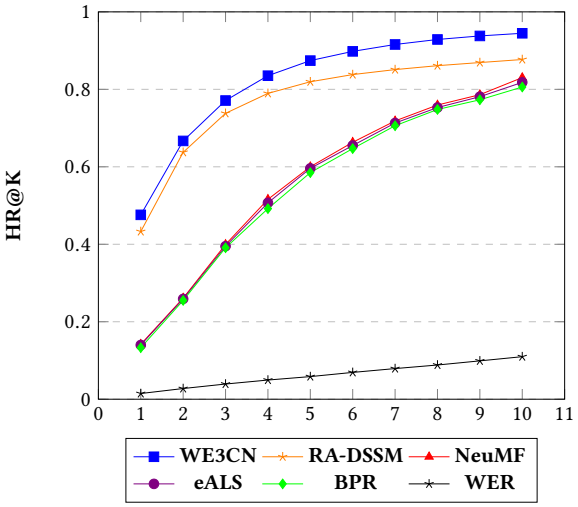

Figure 3: HR of our Model vs state-of-the-art

4.2 Evaluation Protocol

To evaluate the performance of the recommended item we use the leave-one-out evaluation strategy which has been widely adopted in literature [4]. For each user we held-out her latest interaction as the test instance and utilized the remaining data for training using a sliding window approach. We then recommend a ranked list of items and judge the position of the test item by Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG).

4.3 Experimental Settings

For training, we divide the training and validation set in a 4:1 ratio. We tuned the hyper-parameters of our model using the validation

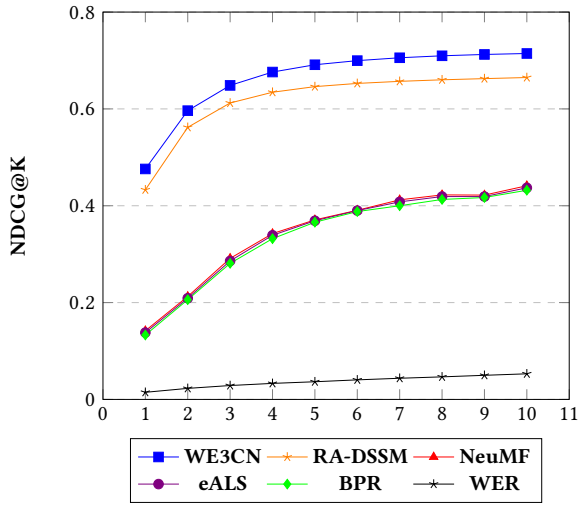

Figure 4: NDCG of our model vs state-of-the-art

set. We used a batch size of 256 and used AdaDelta as a gradient based optimizer.

4.4 Baselines

- **BPR** [11]: This method optimizes the matrix factorization method with a pairwise ranking loss, which is tailored to learn from implicit feedback.
- **eALS** [5]: This is a state-of-the-art matrix factorization method for item recommendation. It optimizes the squared loss (between actual item ratings and predicted ratings) and treats all unobserved interactions as negative instances, weighting them non-uniformly by item popularity.
- **RA-DSSM** [8]: Bi-directional LSTMs are used to model the interests of the user and make predictions.
- **WER** [9]: Word embedding of individual articles are used for the creation of a user profile.
- **NeuMF** [4]: This is the state-of-the-art neural matrix factorization model. It treats the problem of generating recommendation using implicit feedback as a binary classification problem.

5 Results

Fig. 3 and Fig. 4 show the graphs for HR@K and NDCG@K when K is varied from 1 to 10. Our model shows significant improvements over the baselines across all positions. One of the reasons for this is that collaborative filtering models do not account for the sequential information present in the users' historical data. From the results it can also be inferred that our model performs better than RADSSM which utilizes Bi-LSTMs to model the user reading history. This suggests that a 3D CNN has the capability of learning dependencies while extracting meaningful features. We also replaced the LSTMs used in the baseline with RNNs and GRUs and found no improvement in the performance. Apart from this, we also jumbled the original sequence of articles used as inputs to our model and found that the performance deteriorated. It is clear from the results that the user-history component of our model which is based on 3D CNNs seems to be fairly capturing the sequential information required for producing good quality recommendations.

We also varied the reading history R from 10–15 and found R=10 to be performing the best. We then experimented by varying the number of words of an article used as input to our model. We experimented with 25, 50, 75, 100 and found that choosing a size of 50 gave best results. The reason for this can be attributed to the fact that most of the important information in news is covered in the title and its first paragraph.

6 Conclusions

In this paper, we investigated the application of 3D CNNs for providing news recommendation. We found them to significantly outperform the existing methods. The historical user data modeled using 3D convolutions neural networks led to best results. In the future we would like to investigate more on 3D CNNs for personalization.

References

[1] Robert M Bell and Yehuda Koren. 2007. Improved Neighborhood-based Collaborative Filtering. In *KDD*. 7–14.

[2] Minmin Chen, Zhixiang Xu, Fei Sha, and Kilian Q Weinberger. 2012. Marginalized Denoising Autoencoders for Domain Adaptation. In *ICML*. 767–774.

[3] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *WWW*. 278–288.

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proc. of the* 26^{th} *Intl. Conf. on World Wide Web (WWW '17)*.

[5] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In *Proc. of the* 39^{th} *Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*. ACM, 549–558.

[6] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In *CIKM*. 2333–2338.

[7] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (2013), 221–231.

[8] Vaibhav Kumar, Dhruv Khattar, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Neural Architecture for News Recommendation. In *Working Notes of the* 8^{th} *Intl. Conf. of the CLEF Initiative*.

[9] Cataldo Musto, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. 2016. Learning Word Embeddings from Wikipedia for Content-based Recommender Systems. In *ECIR*. Springer, 729–734.

[10] Xia Ning and George Karypis. 2011. Slim: Sparse Linear Methods for Top-n Recommender Systems. In *ICDM*. 497–506.

[11] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proc. of the* 25^{th} *Conf. on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.

[12] Jasson DM Rennie and Nathan Srebro. 2005. Fast Maximum Margin Matrix Factorization for Collaborative Prediction. In *Proc. of the 22nd Intl. Conf. on Machine Learning*. ACM, 713–719.

[13] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo. In *Proc. of the* 25^{th} *Intl. Conf. on Machine Learning*. ACM, 880–887.

[14] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann Machines for Collaborative Filtering. In *Proc. of the* 24^{th} *Intl. Conf. on Machine Learning*. ACM, 791–798.

[15] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proc. of the* 10^{th} *Intl. Conf. on World Wide Web*. ACM, 285–295.

[16] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet Collaborative Filtering. In *Proc. of the* $24^{th} on World Wide Web. ACM$, 111 – –112.

[17] Florian Strub and Jeremie Mary. 2015. Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs. In *NIPS Workshop on ML for eCommerce*.

[18] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-n Recommender Systems. In *WSDM*. 153–162.