International Journal of English Literature and Social Sciences
Vol-9, Issue-6; Nov-Dec, 2024

Peer-Reviewed Journal
Journal Home Page Available: https://ijels.com/
Journal DOI: 10.22161/ijels

YouTube's Two-Stage Deep Learning Framework for Video Recommendations

Sameena Begum[1], Swati Chauhan[2], Sana Sarwar[3], Dr Gulnaz Fatma[4]

[1]Language Instructor, Dept of Foreign Languages, College of Arts and Humanities, Jazan University, KSA
sbmohammed@jazanu.edu.sa
[2]Language Instructor, Dept of Foreign Languages, College of Arts and Humanities, Jazan University, KSA
schauhan@jazanu.edu.sa
[3]Language Instructor, College of Science for Girls, Jazan University, Jazan KSA
ssarwar@jazanu.edu.sa
[4]Language Instructor, University College, Al Ardah, Jazan University, Jazan, Saudi Arabia
gulnaz.fatima15@gmail.com

Received: 23 Nov 2024; Received in revised form: 20 Dec 2024; Accepted: 24 Dec 2024; Available online: 31 Dec 2024

Abstract*— One of the most popular and technologically advanced commercial recommendation systems is YouTube, which makes it one of the most popular. In addition to providing an overview of the system, it shows the extraordinary advances in efficiency that have been made possible by advanced learning techniques. The standard "two-stage" architecture is utilized in this research for the purpose of isolating and obtaining related material. First, a comprehensive model for the generation of candidates should be provided, and then a model for ranking applicants should be outlined. Both of these should be done in the order that they are stated. Important consideration should be given to the sequence in which these two activities should be finished. Additionally, it provides instruction that can be put into practice as well as useful insights that have been gathered from the process of designing, iterating, and maintaining a large-scale recommendation system that has a substantial impact on the experience of the user. In addition, it offers training that can be put into practice.*

Keywords*— Deep Learning, Framework, Ranking, Creating, Iterating.***

I. INTRODUCTION

YouTube has gained widespread recognition as the most prominent global platform for the creation, dissemination, and exploration of material that is based on video technology. By pulling from a vast library of videos, the recommendation system on YouTube plays a vital part in allowing the discovery of material that is specifically customized to the preferences of over one billion users. The purpose of this research is to investigate the major impact that machine learning has had on the video rating system that YouTube has utilized over the past few years. A representation of the suggestions that appear on the home screen of the YouTube application for mobile devices is shown in Figure 1. There are three primary areas in which it is challenging to create recommendations for videos on YouTube: A great number of the existing recommendation algorithms, which have been demonstrated to perform effectively in limited circumstances, require increased scalability in order to be helpful for their requirements. Scattered learning algorithms and speedy, dependable portion systems are essential for YouTube because of the enormous user base and material collecting that makes it possible for YouTube to function. Considering that YouTube uploads a growing number of videos each and every second, the material of the website is always evolving. When it comes to accurately processing freshly delivered content and user actions, the efficiency of the reference system needs to be sufficient. Maintaining a sufficient amount of fresh information at all times.

Fig. 1.Home screen YouTube app recommendations

Due to sparsity and other external variables that are difficult to see, it is challenging to forecast how users have behaved on YouTube in the past. Instead of asking users about their level of pleasure, simulate their chaotic implicit feedback signals. The metadata that describes content needs to be better organised with an explicit ontology. The algorithms must be robust enough to handle these irregularities in the training data. YouTube has undertaken a similar paradigm shift, with deep learning being seen as a silver bullet for all learning problems, as other Google products have. TensorFlow, an offshoot of Google Brain, is the foundation of the system. Using large-scale distributed training, TensorFlow allows for exploring several deep neural network designs. About a billion parameters are learned by these models, which are trained using hundreds of billions of data points. While matrix factorization techniques have seen many studies, deep neural networks for recommendation systems have seen minimal development. News recommendations, citation suggestions, and rating suggestions are just some of the many uses for neural networks. Both autoencoders and the deep neural network formulation of collaborative filtering used therein use deep learning to represent users across domains. Applying deep neural networks to the Task of Music Recommendation in a Content-based Environment This paper follows the following structure: Section 2, provides a high-level overview of the system. The model's advantages concerning deep layers of unseen units and extra varied signals will be shown experimentally. The ranking model is described in Section 4, along with the adjustments made to traditional logistic regression to train a model to predict the typical viewing duration. Section 3, explains the candidate generation model, and Section 4 depicts its training and application. As shown by experiments, the depth of the concealed layer is also valuable here. Finally, discuss the findings and take-out in Section 5.

II. METHODOLOGY

Figure 2 depicts the entire architecture of the recommendation system. The system comprises two neural networks—one for coming up with potential candidates and another for rating them. A massive repository of YouTube viewing history is used as the source for the candidate generation network, which selects a very small subset (hundreds of videos). These potentials are typically extremely accurate and applicable to the user. The contender generation network offers generalizations of individualized features, such as collaborative filtering, to its users. Coarse information, such as video watch identifiers, search query tokens, and demographics, describe the degree of similarity between users. The population composition, etc. The ranking network uses a wealth of characteristics about the video and the viewer to score each video based on the intended objective function. The highest-rated videos are shown in order of their score. Employ a two-stage recommendation process to ensure that the few movies shown on the device are tailored to the user's interests and preferences. This enables us to make suggestions from a massive corpus (millions of videos). Candidate mixing from other sources, such as those outlined in a previous paper, is also possible with this system. To drive the iterative refinement of the system throughout development, rely on offline measures (such as accuracy, recall, ranking loss, etc.). However, depends on A/B testing via real-world trials to determine an algorithm or model's efficacy. Subtle changes in metrics like click-through rate and watch duration are easily measurable in a live experiment. This is significant since offline trials don't always correspond with findings from live A/B testing.

III. PRODUCTION OF CANDIDATES

During contender, thousands of videos in YouTube's massive corpus are narrowed down to hundreds that may interest the viewer. A matrix decomposition method, trained using rank loss, was the recommender's forerunner. The first neural network prototypes were shallow networks that just stored the user's historical watching habits, emulating this factorization tendency. This method may be considered a non-linear extension of factorization approaches.

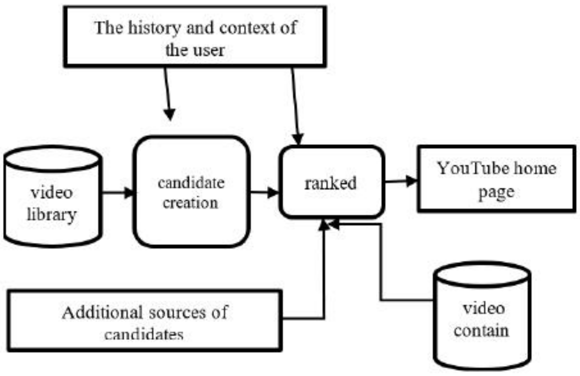
The history and context of the user
video library
candidate creation
ranked
YouTube home page
Additional sources of candidates
video contain

Fig. 2.In the System for Recommendations, A "Funnel" Is Used to Locate and Rate Potential Movies Before Displaying Just A Handful To The Viewer

A. *TYPES OF RECOMMENDATIONS:*

As an extreme form of multiclass classification, defining the recommendation of the problem, determining which of millions of movies in a corps server chooses to watch at a given time

$$p(W_T = I|u,c) = \frac{E^{V_I U}}{\sum_{J\epsilon v} E^{V_J U}}$$

The user's "embedding" (represented by $U \in R^N$), the context pair, and the embedding of each potential video (represented by $V_J \in R^N$), all have high dimensionality. An embedding in this context is equivalent to a mapping from the discrete entities (videos, users, etc.) to a dense vector R^N. The goal of the deep neural network is to train a set of user embeddings u that are helpful for softmax video classification based on the user's past actions and current circumstances. Despite this, YouTube has built-in systems for providing direct input. For model training, we rely on watches' implicit feedback, where a user's successful film completion serves as a benchmark. We made this decision because we have access to orders of extent more implicit user data, which allows us to provide suggestions in the tail, where obvious input is very scarce

B. *Effective Multiclass Extreme Learning:*

To effectively train a model with many classes, use a "candidate sampling" method to choose antagonistic classes from the background distribution. Subsequently, to address the bias introduced by this selection via significance weighting the cross-entropy loss is minimised for the actual label and the putative antagonistic paths. In practical implementation, a substantial number of negative samples are selected, resulting in a performance improvement of over 100 times compared to the conventional SoftMax approach. One often used alternative method is hierarchical SoftMax; however, the attempts to get similar levels of accuracy were unsuccessful. In the context of hierarchical SoftMax, visiting each node in the tree requires distinguishing between groups of often unrelated classes. This introduces additional complexity to the classification task and leads to a decline in performance. During the serving phase, it is necessary to calculate the N most probable classes (videos) to choose the top N for presentation to the user. To achieve efficient scoring of a large number of items under a tight serving latency constraint of tens of milliseconds, it is necessary to use an approximation scoring method that exhibits sublinear scaling concerning the number of classes. The preceding methods used at YouTube depended on hashing, and the classifier described in this paper employs a similar methodology. When serving, there is no need for calibrated probabilities derived from the SoftMax yield layer. Consequently, the scoring issue may be simplified to the closest neighbour exploration in the dot produce space, which can be efficiently addressed using general-purpose libraries. The findings indicate that selecting the closest neighbour search technique did not significantly impact the A/B outcomes.

C. *Model Structure:*

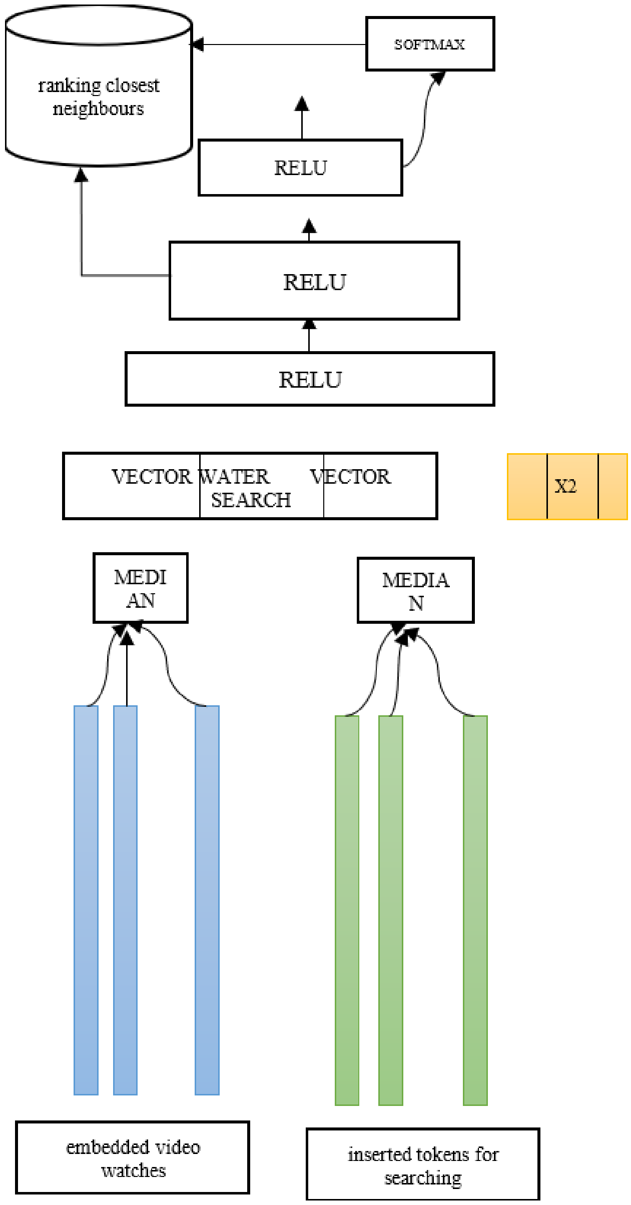
ranking closest neighbours
SOFTMAX
RELU
RELU
RELU
VECTOR WATER SEARCH VECTOR
X2
MEDIAN
MEDIAN
embedded video watches
inserted tokens for searching

Fig. 3.Embedded sparse features connected to dense features in a deep candidate generation model architecture. To make uniform input vectors of constant width for the hidden layers, variable-sized bags of sparse IDs are embedded and concatenated. All unseen levels are intricately interwoven. Training involves using gradient descent to minimize a cross-entropy loss on the sampled softmax's output. At the time of service, hundreds of potential videos are generated using an estimated closest neighbour query.

Motivated by the language models known as continuous bags of words, acquire embeddings of high dimensionality for every video inside a predetermined vocabulary. These embeddings are then inputted into a feedforward neural network. A user's watch history is denoted by a series of video IDs, which might vary in length and sparsity. These video IDs are then transformed into a dense vector representation using embeddings. The network necessitates inputs that are dense and of constant size. Among several tactics, such as sum and component-wise max, it was shown that the highest performance was achieved by simply averaging the embeddings. The embeddings and the rest of the model parameters are learned using standard gradient descent backpropagation updates, which is of vital relevance. There is a vast initial layer where the characteristics are integrated, followed by several Rectified Linear Units that are all connected. Figure 3 depicts the overarching network architecture, which encompasses supplementary functionalities about non-video viewing attributes, as elaborated upon afterwards.

D. DIFFERING SIGNALS

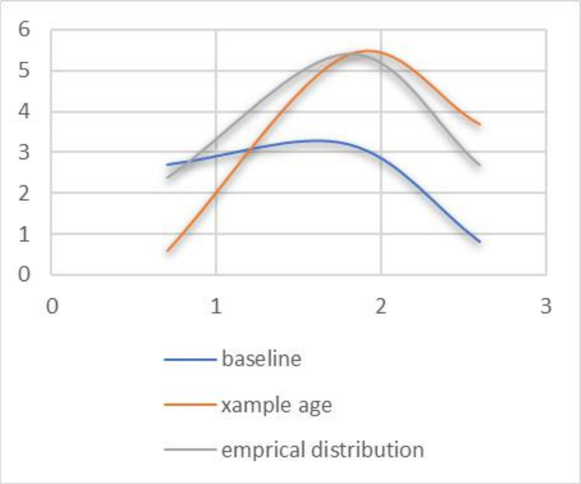

Fig. 4.A Video Algorithm Trained Using Age as A Feature Properly Represents Uploading Time and Interest Over Time. The Model Predicted the Training Window Median Probability Without the Feature.

One notable benefit of using deep neuronal networks as an extension of matrix factoring is their ability to include diverse continuous and categorical variables into the model easily Unigrams and bigrams are separated from each query. and then embedded with associated representations, much to how watch history is handled. After being averaged, the tokenized and embedded queries of the user serve as a condensed representation of their whole search history. The inclusion of demographic variables is crucial in establishing past information, so ensuring that the suggestions exhibit reasonable behaviour for users who are new to the system. The geographic area and device of the user are combined and joined together. The network receives input in real numbers normalized to [0, 1], Examples of primary binary and continuous data include the user's gender, login status, and age.

E. SELECTING LABELS AND CONTEXT

It is crucial to highlight that making recommendations often entails resolving a surrogate issue and applying the obtained solution to a specific setting. One such instance is the presumption that the precise prediction of ratings indicates the efficacy of movie recommendations. The selection of a surrogate learning issue has been seen to impact the effectiveness of A/B testing significantly. However, accurately measuring this impact via offline tests poses considerable challenges. Training examples are derived from all instances of YouTube video views, including those hosted on other websites, rather than just focusing on views related to the suggestions given. Otherwise, the emergence of fresh material would be significantly hindered, and the recommender system would exhibit an excessive inclination towards favouring exploitative content. If users see films via channels other than recommended suggestions, must promptly disseminate this finding to other users using collaborative filtering. A further significant observation that contributed to the enhancement of live metrics was implementing a strategy to provide a consistent quantity of training instances for each user, ensuring equal weighting of our users inside the loss function. This measure effectively mitigated the potential for a select group of highly engaged users to influence the outcome excessively. To mitigate the problem of overfitting in the surrogate issue and effectively use the site's structure, it is essential to refrain from providing specific information to the classifier. To illustrate the current scenario, consider the case of an individual searching for information about the renowned artist Taylor Swift. The videos with the highest likelihood of being seen are prominently featured on the search results page for "Taylor Swift. “Therefore, a classifier trained with this data may predict which videos will be watched next. Predictably, homepage suggestions that mirror a user's most recent search result fare badly. Classifiers are blinded to the context of a label when sequence information is discarded, and queries are represented as an unsorted collection of tokens. Co-viewing probabilities are very asymmetric due to viewers' natural viewing habits. Users often find musicians in a genre by starting with the most widely popular and working their way down to the more specialized acts The user's text needs to be longer to be rewritten academically. As a result of this investigation, it was determined that anticipating the user's subsequent timepiece was much more efficient than predicting a randomly withheld wristwatch, as seen in Figure 5. In many collaborative filtering systems, labels and context are chosen invisibly by picking an item at random and making a prediction based on the user's previous interactions with similar things (5a).

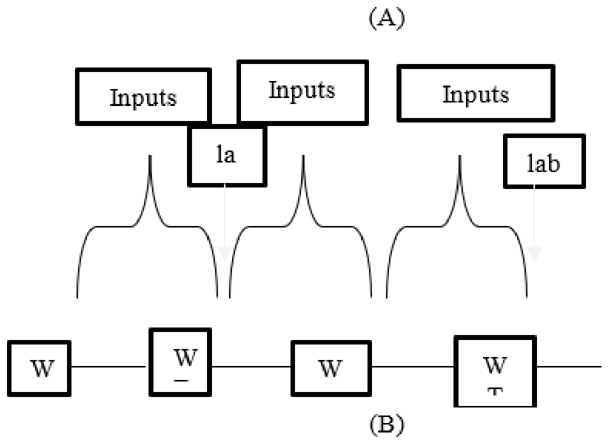

Fig. 5.(a) Performed Better. In 5b), The Age Given Is T max − Tn, Where T max Is the Extreme Observed Time in The Training Data.

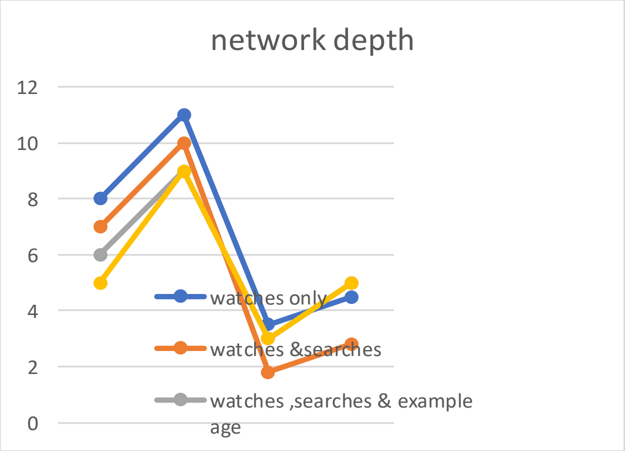

Fig. 6.Adding data beyond video embeddings improves retention MAP, and height layers enhance expression for successful interaction modelling.

The current system's performance exhibited a high degree of similarity to that of its predecessor. The width and depth dimensions were increased until the additional advantages were less significant, and the convergence process became challenging. (Figure 6)

- At depth 0, the linear layer is responsible for transforming the concatenation layer to align its dimensions with the SoftMax layer, which has a size of 256.
- First depth: 257 Re L U
- Deepness 2: 412 Re LU → 256 Re LU
- Re LU depth 2: 1034 → 512 → 256
- Depth 4: 2048 Re LU → 1024 → 512 → 256.

IV. COMPARISON

Ranking's primary function is to fine-tune and specialize candidate predictions based on impression data for the given user interface. For instance, despite the user's strong propensity to view a particular video, the homepage impression, including the video's thumbnail, is unlikely to elicit a click. Since just a few hundred movies are being evaluated during ranking instead of millions during candidate creation, access much more information defining the video and the user's connection to the video. The position is also essential for combining many applicant sources with incomparable ratings for each video impression, using logistic reversion and a deep neural network with an architecture identical to applicant generation (Figure 7). The customer is, after that, provided with a collection of videos, which are organized based on their respective scores. The primary objective of our ranking strategy is to optimize the expected view time per impression. However, consistently refine this approach based on the results obtained from live A/B testing. In contrast to watch duration, which more accurately reflects user involvement, rankings based on click-through rate tend to promote misleading films that the viewer does not finish ("clickbait").

A. REPRESENTING CHARACTERISTICS

There is a clear delineation between the categorical and continuous/ordinal properties we use. There is a broad range of cardinality among the categorical characteristics; some have just two potential values (such as whether or not the user is signed in), while others have millions. Features are further classified as either "univalent" (contributing a single value) or "multivalent" (contributing many values). Univalent category features include the video ID of the impression being evaluated. In contrast, multivalent features could include a bag of the user's most recent N video IDs. We also categorize features based on whether they describe item qualities (an "impression") or user/context properties (a "query"). Instead of being calculated for each item rated, impression characteristics are calculated once for each query.

1) Engineered Features

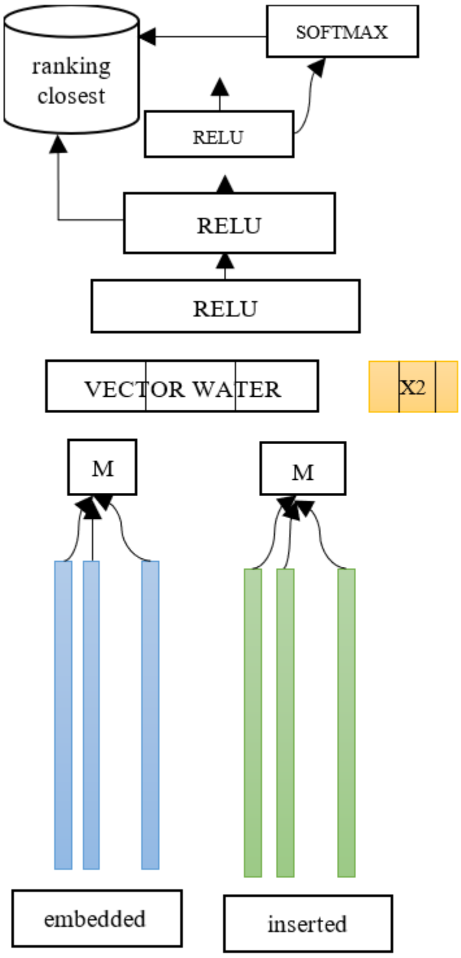

Fig. 7.Through the Use of Common Embeddings and Powers of Normalized Continuous Features, This Deep-Ranking Network Design Can Accommodate Both Univalent and Multivalent Categorical Data. All Levels Are Interconnected. The Network Learns from Experience and Is Fed Hundreds of Features.

The characteristics of ranking algorithms often include numbers in the hundreds, with the majority being either categorical or continuous. The raw data does not naturally fit into feedforward neural networks, even though deep learning can potentially reduce the requirement for hand-engineered features. It still takes technical work to transform user and video input into usable functionality. The fundamental difficulty is in accurately displaying a user's behaviours throughout time and how those actions affect the video's overall impression. Consistent with other researchers' findings in ad ranking, find that signals describing a user's prior engagement with the item and related things are the most essential. Continuous characteristics that describe prior user behaviours on connected objects are mainly potent since they simplify effectively across things. It is also discovered that it's vital to spread knowledge through features from the candidate generation phase into the ranking phase. It is also important to include features that describe the frequency of previous video impressions to introduce "churn". If a user recently suggested a movie but did not watch it, the algorithm would automatically lower this impression on the subsequent page visit. Providing real-time access to user feedback and viewing habits is a significant technical challenge outside the scope of this article but essential for generating relevant suggestions

B. *CATEGORICAL FEATURES EMBED*

Like the candidate generation process, embeddings transform sparse categorical information into compact representations well-suited for utilization in neural networks. Each distinct identifier space, or "vocabulary," is associated with an independently learnt embedding. The dimension of this embedding grows in a manner that is roughly equal to the logarithm of the total number of exceptional values inside the vocabulary. The relationship is directly proportional to the logarithm of the number of distinct values. The vocabulary mentioned above consists of uncomplicated lookup tables constructed by traversing the data once before the training process. To manage the vast cardinality of ID spaces, such as video IDs or search query phrases, a common approach is to truncate the space by selecting just the top N elements. This truncation is often performed by sorting the elements based on their frequency in clicked impressions and retaining only the highest-ranking ones. Values not present in the vocabulary are assigned to the zero embedding. In the process of candidate creation, the multivalent categorical feature embeddings undergo an averaging operation before inputting into the network. It's vital to remember that ID space-independent categorical features have joint embeddings. The video ID linked with the impression, the video ID of the last film the user saw, and the video ID that kicked off the recommendation process are all part of a global embedding of video IDs included in this instance. Despite the shared embedding, the neural network receives input from each feature separately. This allows the deeper layers to learn unique representations for each feature. Pooling resources, including embeddings, may improve generalization, speed up training, and alleviate memory restrictions. The majority of model parameters favour a high cardinality embedding space. To illustrate, a 32-dimensional space that embeds one million IDs has seven times more parameters than ultimately linked layers with a width of 2048 units.

V. CONCLUSIONS

Deep neural network architecture is presented in detail, focusing on its application in suggesting YouTube videos. The design is divided into two components: candidate generation and rating, each addressing specific challenges.

The deep collaborative filtering model demonstrates proficiency in including several signals and capturing their intricate relationships via layered depth. This surpasses the performance of prior matrix factorization methods employed by YouTube. The process of picking a surrogate problem for suggestions involves a greater emphasis on artistic judgment rather than scientific methodology. The study focused on classifying a future watch that would exhibit strong performance based on real-time measurements. This was achieved by recording the asymmetric co-watch behaviour and implementing measures to avoid leaking future information. The exclusion of discriminative signals from the classifier was crucial in obtaining favourable outcomes. Otherwise, the model would exhibit overfitting towards the surrogate issue and need to effectively generalize to the home page. Using the training example's age as an input feature reduces the inherent bias towards the past, as shown by the research. This adjustment allows the model to describe well-known films' time-dependent behaviour correctly. The implementation of offline holdout accuracy has led to significant improvements in performance metrics, particularly in increased view time for freshly uploaded movies, as seen via A/B testing. Ranking is a traditional topic in the field of machine learning. However, our deep learning strategy outperformed earlier approaches based on linear and tree-based models for predicting watch time. Specialized attributes that describe previous user behaviour with objects are particularly advantageous for recommendation systems. Deep neural networks need specialized representations for both categorical and continuous information. These representations are achieved via embedding embeddings for categorical features and quantile normalization for continuous features. The study demonstrated the efficacy of including many layers of depth to accurately represent and capture non-linear relationships among a multitude of data. The logistic regression model was modified by including a weighting scheme for training examples, where positive instances were assigned, weights based on their watch time, and negative examples were assigned a weight of unity. This modification enables the model to effectively estimate the probabilities that closely align with the anticipated watch time. The methodology above exhibited superior performance in assessment criteria weighted by watch time instead of directly forecasting click-through rate.

REFERENCES

[1] Christakopoulou, Konstantina, et al. "Q&R: A two-stage approach toward interactive recommendation." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018.

[2] Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for YouTube recommendations." *Proceedings of the 10th ACM conference on recommender systems*. 2016.

[3] Wang, Guoqiang, et al. "Revisiting TAM2 in behavioural targeting advertising: a deep learning-based dual-stage SEM-ANN analysis." *Technological Forecasting and Social Change* 175 (2022): 121345.

[4] Huang, Zhenhua, et al. "A novel group recommendation model with two-stage deep learning." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.9 (2021): 5853-5864.

[5] Ni, Juan, et al. "A two-stage embedding model for recommendation with multimodal auxiliary information." *Information Sciences* 582 (2022): 22-37.

[6] Baluja, Shumeet, et al. "Video suggestion and discovery for YouTube: taking random walks through the view graph." *Proceedings of the 17th International Conference on World Wide Web*. 2008.

[7] Li, Chao, et al. "multi-interest network with dynamic routing for recommendation at Tmall." *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019.

[8] Zhao, Zhe, et al. "Recommending what video to watch next: a multitask ranking system." *Proceedings of the 13th ACM Conference on Recommender Systems*. 2019.

[9] Li, Qian, et al. "Deep learning-based short video recommendation and prefetching for mobile computing users." *Proceedings of the ACM SIGCOMM 2019 Workshop on Networking for Emerging Applications and Technologies*. 2019.

[10] Zhang, Shuai, et al. "Deep learning-based recommender system: A survey and new perspectives." *ACM computing surveys (CSUR)* 52.1 (2019): 1-38.

[11] Tsai, Chieh-Yuan, Yi-Fan Chiu, and Yu-Jen Chen. "A two-stage neural network-based cold start item recommender." *Applied Sciences* 11.9 (2021): 4243.

[12] Rani, Shalli, et al. "Detection of shilling attack in recommender system for YouTube video statistics using machine learning techniques." *Soft Computing* (2021): 1-13.

[13] Vishwakarma, Anish Kumar, and Kishor M. Bhurchandi. "No-Reference Video Quality Assessment using Novel Hybrid Features and Two-stage Hybrid Regression for Score level Fusion." *Journal of Visual Communication and Image Representation* 89 (2022): 103676.

[14] Gao, Tianhan, Lei Jiang, and Xibao Wang. "Recommendation system based on deep learning." *Advances on Broad-Band Wireless Computing, Communication and Applications: Proceedings of the 14th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2019) 14*. Springer International Publishing, 2020.

[15] Baccouche, Moez, et al. "Sequential deep learning for human action recognition." *Human Behavior Understanding: Second International Workshop, HBU 2011, Amsterdam, The Netherlands, November 16, 2011. Proceedings 2*. Springer Berlin Heidelberg, 2011.