(12) PATENT APPLICATION PUBLICATION
(19) INDIA
(22) Date of filing of Application :28/03/2025

(21) Application No.202541030172 A
(43) Publication Date : 18/04/2025

(54) Title of the invention : FEATURE EXTRACTION USING ADAM OPTIMIZER ALGORITHM

(51) International classification	:G06Q0030020000, G06F0016953500, H04N0021466000, H04L0067500000, H04H0060460000
(86) International Application No	:NA
Filing Date	:NA
(87) International Publication No	: NA
(61) Patent of Addition to Application Number	:NA
Filing Date	:NA
(62) Divisional to Application Number	:NA
Filing Date	:NA

(71)Name of Applicant :
1)Dr. S. Sivapriya
Address of Applicant :Assistant Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600 072, Thiruvallur, Tamilnadu ---- ------- ----------
2)Dr. E. Saravanan
3)Dr. P. Ramesh
4)Dr. A.Kavitha
5)Dr. Puneet Kumar Gupta
6)Dr. P. Pirakatheeswari
7)Aabi Princy .A
8)Ms. Swati Raturi
9)Dr. Rakhi Shukla
10)Dr. C.Nithya Prabha
Name of Applicant : NA
Address of Applicant : NA
(72)Name of Inventor :
1)Dr. S. Sivapriya
Address of Applicant :Assistant Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600 072, Thiruvallur, Tamilnadu ---- ------- ----------
2)Dr. E. Saravanan
Address of Applicant :Research Supervisor & Guide, Associate Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600 072, Thiruvallur, Tamilnadu ----------- -----------
3)Dr. P. Ramesh
Address of Applicant :Professor, Department of Mathematics - Erode Sengunthar Engineering College, Erode, Tamilnadu ----------- -----------
4)Dr. A.Kavitha
Address of Applicant :Research Supervisor & Guide, Assistant Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600 072,Thiruvallur, Tamilnadu ----------- -----------
5)Dr. Puneet Kumar Gupta
Address of Applicant :Assistant Professor, ICFAI Business School, The ICFAI University, Rajawala Road, Selaqui, Dehradun, Uttrakhand - 248197 ----------- -----------
6)Dr. P. Pirakatheeswari
Address of Applicant :Associate Professor, Department of Commerce - PA, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore – 641006, Tamilnadu ----------- -----------
7)Aabi Princy .A
Address of Applicant :Student,III B.Com,Dr SNS Rajalakshmi College of Arts and Science,Coimbatore-641049, Tamilnadu ----------- -----------
8)Ms. Swati Raturi
Address of Applicant :Faculty Associate, ICFAI Education School, The ICFAI University, Rajawala Road, Selaqui, Dehradun, Uttarakhand - 248197 ----------- -----------
9)Dr. Rakhi Shukla
Address of Applicant :Assistant Professor, ICFAI Business School, The ICFAI University, Rajawala Road, Selaqui, Dehradun, Uttarakhand - 248197 ----------- -----------
10)Dr. C.Nithya Prabha
Address of Applicant :Asistant professor Department of Commerce with Information Technology Dr.SNS Rajalakshmi college of Arts and science , Coimbatore, Tamil Nadu ----------- -----------

(57) Abstract :
FEATURE EXTRACTION USING ADAM OPTIMIZER ALGORITHM ABSTRACT: In order to identify user preferences from a huge selection of available things, recommender systems use a variety of data mining techniques and algorithms. In contrast to static systems, recommender systems encourage greater interaction to provide a richer experience. These systems can automatically identify recommendations for specific users by assessing prior purchases, searches, and the activity of other users. This method uses user history data and the data of other users to forecast consumer preferences and generate tailored recommendations. This study focuses on the difficulties that recommender systems encounter, including the cold start issue, data sparsity, scalability, and accuracy. It explores content-based filtering in particular, which creates suggestions based on user activity. Content-based filtering use long-term, updateable user preference profiles, much like collaborative filtering. Recommender systems have become an essential tool in many online platforms, providing personalised recommendations to users to improve their experience and increase engagement. These systems utilize various data mining techniques and algorithms to analyse user behaviour, such as past purchases, searches, and interactions, to discern their preferences and make personalised recommendations. In recent years, recommender systems have evolved from static systems to more dynamic ones that foster increased interaction to offer a more enriched experience forusers. The cold start problem, which occurs when new users or objects have little to no previous data available for recommendations, is one of the major difficulties recommender systems must overcome. As a result, the system faces a serious issue in accurately predicting user preferences and making pertinent recommendations. The availability of scant data for recommendations is another difficulty, making it difficult to recognise trends and make precise predictions. Scalability is a problem since recommender systems must process a lot of data and give real-time recommendations to lots of consumers. Finally, accuracy is essential for recommender systems to earn user confidence and consistently give relevant recommendations. Content-based filtering, which creates suggestions based on the qualities or attributes of things rather than depending exclusively on user behaviour or collaborative filtering, is one prominent strategy for overcoming these difficulties. Content-based filtering makes use of an object's intrinsic characteristics to construct item profiles, such as genre, director, actors, keywords, or other metadata. The user profiles that are produced based on the user's past conduct or preferences are then matched with these item profiles. Even for new users or things with little historical data, content-based filtering can provide individualised recommendations by comparing the similarity between item and user profiles. This study's goal is to examine the difficulties recommender systems encounter, with a particular emphasis on content-based filtering as a solution. The different facets of content-based filtering, as well as its advantages and disadvantages, will be covered in detail in this paper, along with how it can be applied to enhance the functionality of recommender systems. The paper will also go through the various strategies and algorithmic frameworks employed in content-based filtering, as well as the most recent developments and fashions in this area.

No. of Pages : 10 No. of Claims : 8

D.R.B.C.C.C. HINDU COLLEGE
IQAC/NAAC
24-25

PRINCIPAL
DHARMAMURTHI RAO BAHADUR CALAVALA
CUNNAN CHETTY'S HINDU COLLEGE,
DHARMAMURTHI NAGAR, PATTABIRAM,
CHENNAI - 600 072.

FORM 5
THE PATENTS ACT 1970
(39 of 1970)
&
The Patents rules, 2003
DECLARATION AS TO INVENTORSHIP
[See section 10(6) and rule 13(6)]

1. NAME: OF APPICANT (S)	1. Dr. S. Sivapriya 2. Dr. E. Saravanan 3. Dr. P. Ramesh 4. Dr. A.Kavitha 5. Dr. Puneet Kumar Gupta 6. Dr. P. Pirakatheeswari 7. Aabi Princy .A 8. Ms. Swati Raturi 9. Dr. Rakhi Shukla 10. Dr. C.Nithya Prabha

Hereby declare that the truth and first inventor (s) of the invention disclosed in the provisional specification filed in pursuance of my application numbered **2025**__________ dated ______ are.

2. INVENTOR (S)

NAME	NATIONALITY	ADDRESS
Dr. S. Sivapriya	Indian	Assistant Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600 072, Thiruvallur, Tamilnadu
Dr. E. Saravanan	Indian	Research Supervisor & Guide, Associate Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600 072, Thiruvallur, Tamilnadu
Dr. P. Ramesh	Indian	Professor, Department of Mathematics - Erode Sengunthar Engineering College, Erode, Tamilnadu
Dr. A.Kavitha	Indian	Research Supervisor & Guide, Assistant Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600 072,Thiruvallur, Tamilnadu
Dr. Puneet Kumar Gupta	Indian	Assistant Professor, ICFAI Business School, The ICFAI University, Rajawala Road, Selaqui, Dehradun, Uttrakhand - 248197

Dr. P. Pirakatheeswari	Indian	Associate Professor, Department of Commerce - PA, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore – 641006, Tamilnadu
Aabi Princy .A	Indian	Student,III B.Com,Dr SNS Rajalakshmi College of Arts and Science,Coimbatore-641049, Tamilnadu
Ms. Swati Raturi	Indian	Faculty Associate, ICFAI Education School, The ICFAI University, Rajawala Road, Selaqui, Dehradun, Uttarakhand - 248197
Dr. Rakhi Shukla	Indian	Assistant Professor, ICFAI Business School, The ICFAI University, Rajawala Road, Selaqui, Dehradun, Uttarakhand - 248197
Dr. C.Nithya Prabha	Indian	Asistant professor Department of Commerce with Information Technology Dr.SNS Rajalakshmi college of Arts and science , Coimbatore, Tamil Nadu

Date 28/03/2025

Saurabh Kumar Jain
(IN/PA-3637)
Agent for Applicant

3. DECLARATION TO BE GIVEN WHEN THE APPLICATION IN INDIA IS FILED BY THE APPLICANT (S) IN THE CONVENTION COUNTRY:-

-NA-

~~We the applicant(s) in the convention country hereby declare that our right to apply for a patent in India is by way of assignment from the true and first inventor(s).~~

Dated this____day of ____, 2022.

Signature:-NA

Name: of signatory:- NA

To,

The controller of patent
The patent office, at Delhi/Mumbai/Chennai/Kolkata.

FORM 2

THE PATENTS ACT 1970

39 OF 1970

&

THE PATENT RULES 2003

COMPLETE SPECIFICATION

(SEE SECTIONS 10 & RULE 13)

1. TITLE OF THE INVENTION

FEATURE EXTRACTION USING ADAM OPTIMIZER ALGORITHM

2. APPLICANTS (S)

NAME	NATIONALITY	ADDRESS
Dr. S. Sivapriya	Indian	Assistant Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600 072, Thiruvallur, Tamilnadu
Dr. E. Saravanan	Indian	Research Supervisor & Guide, Associate Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600 072, Thiruvallur, Tamilnadu
Dr. P. Ramesh	Indian	Professor, Department of Mathematics - Erode Sengunthar Engineering College, Erode, Tamilnadu
Dr. A.Kavitha	Indian	Research Supervisor & Guide, Assistant Professor, PG& Research Department of Commerce, Dharmamurthi Rao Bahadur Calavala Cunnan Chetty's Hindu College, Pattabiram, Chennai - 600

		072,Thiruvallur, Tamilnadu
Dr. Puneet Kumar Gupta	Indian	Assistant Professor, ICFAI Business School, The ICFAI University, Rajawala Road, Selaqui, Dehradun, Uttrakhand - 248197
Dr. P. Pirakatheeswari	Indian	Associate Professor, Department of Commerce - PA, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore – 641006, Tamilnadu
Aabi Princy .A	Indian	Student,III B.Com,Dr SNS Rajalakshmi College of Arts and Science,Coimbatore-641049, Tamilnadu
Ms. Swati Raturi	Indian	Faculty Associate, ICFAI Education School, The ICFAI University, Rajawala Road, Selaqui, Dehradun, Uttarakhand - 248197
Dr. Rakhi Shukla	Indian	Assistant Professor, ICFAI Business School, The ICFAI University, Rajawala Road, Selaqui, Dehradun, Uttarakhand - 248197
Dr. C.Nithya Prabha	Indian	Asistant professor Department of Commerce with Information Technology Dr.SNS Rajalakshmi college of Arts and science , Coimbatore, Tamil Nadu

2. PREAMBLE TO THE DESCRIPTION

COMPLETE SPECIFICATION

The following specification particularly describes the invention and the manner in which it is to be performed

FEATURE EXTRACTION USING ADAM OPTIMIZER ALGORITHM

ABSTRACT:

In order to identify user preferences from a huge selection of available things, recommender systems use a variety of data mining techniques and algorithms. In contrast to static systems, recommender systems encourage greater interaction to provide a richer experience. These systems can automatically identify recommendations for specific users by assessing prior purchases, searches, and the activity of other users. This method uses user history data and the data of other users to forecast consumer preferences and generate tailored recommendations. This study focuses on the difficulties that recommender systems encounter, including the cold start issue, data sparsity, scalability, and accuracy. It explores content-based filtering in particular, which creates suggestions based on user activity. Content-based filtering use long-term, updateable user preference profiles, much like collaborative filtering.

Recommender systems have become an essential tool in many online platforms, providing personalised recommendations to users to improve their experience and increase engagement. These systems utilize various data mining techniques and algorithms to **analyse user behaviour**, such as past purchases, searches, and interactions, to discern their preferences and make personalised recommendations. In recent years, recommender systems have evolved from static systems to more dynamic ones that foster increased interaction to offer a more enriched experience forusers.

The cold start problem, which occurs when new users or objects have little to no previous data available for recommendations, is one of the major difficulties recommender systems must overcome. As a result, the system faces a serious issue in accurately predicting user preferences and making pertinent recommendations. The availability of scant data for recommendations is another difficulty, making it difficult to recognise trends and make precise predictions. Scalability is a problem since recommender systems must process a lot of data and give real-time recommendations to lots of consumers. Finally, accuracy is essential for recommender systems to earn user confidence and consistently give relevant recommendations.

Content-based filtering, which creates suggestions based on the qualities or attributes of things rather than depending exclusively on user behaviour or collaborative filtering, is one prominent strategy for overcoming these difficulties. Content-based filtering makes use of an object's intrinsic characteristics to construct item profiles, such as genre, director, actors, keywords, or other metadata. The user profiles that are produced based on the user's past conduct or preferences are then matched with these item profiles. Even for new users or things with little historical data, content-based filtering can provide individualised recommendations by comparing the similarity between item and user profiles.

This study's goal is to examine the difficulties recommender systems encounter, with a particular emphasis on content-based filtering as a solution. The different facets of content-based filtering, as well as its advantages and disadvantages, will be covered in detail in this paper, along with how it can be applied to enhance the functionality of recommender systems. The paper will also go through the various strategies and algorithmic frameworks employed in content-based filtering, as well as the most recent developments and fashions in this area.

CLAIMS:

1. Recommended systems are widely used in various domains such as online booking, online shopping, and audio and video recommendations, among others, to generate personalised preferences to support decision- making for users.

2. Despite being a well-established concept, the increasing number of users and choices available has made the task of providing relevant recomendations more challenging. The cold start problem, which occurs when a new user or product enters the system and lacks sufficient user rating history, further complicates the performance of recommender systems.

3. To address this challenge, a proposed solution is a hybrid recommender system that leverages demographic attributes such as age, gender, occupation, and similarity to existing users to

generate more accurate and relevant recommendations compared to traditional methods. The recommender system has become one of the most important methods for providing personalised documents, merchandise, and services to fulfil user requirements in information retrieval, e-commerce, and online services.

4. With the increasing volume of data and information available daily, the problem of information overload arises, making it challenging to identify customers' requirements. While search engines were an initial solution to this problem, they lacked personalisation. Recommendation systems were then introduced to utilize users' history to understand their interests and preferences from a large set of options.

5. The main objective of a recommendation system is to provide meaningful suggestions and recommendations for items of interest, such as book recommendations on Amazon, which utilize recommendation systems to identify users' preferences and attract them to engage more. Various methods and algorithms are available for creating personalised recommendations in recommendationsystems.

6. In this system, the recommendation process begins with gathering information about items, such as author, title, and cost, and utilising feature extraction and in- formation indexing. Content-based filtering is employed to process information and data from various sources, extracting useful features and elements about the contents of items [1]. Constraint-based filtering, on the other hand, utilises item features to determine their relevance.

7. Feature extraction and representation can be achieved automatically, such as extracting news from papers, or manually, where human editors need to insert features from items such as movies and songs. Recommender systems facilitate matching users with items, and different types of recommender systems are designed based on available data, implicit and explicit user feedback, domain characteristics, etc. These recommender systems are classified according to the approach or paradigm used for predicting preferences in the researchfield.

8. Two major approaches commonly employed in Recommender Systems are **Collaborative Filtering** and**Content-Based Filtering**. In this paper, we have adopted the Content-Based Filtering approach to enhance recommendation accuracy.

COLLABORATIVE FILTERING

Collaborative filtering is a method used in recommendation systems to model user behaviour based on past interactions. This model can be built from an individual user's behaviour or similar users' behaviour. Collaborative filtering leverages group knowledge to make recommendations based on users who exhibit similar preferences or behaviours. By analysing the information from multiple users who subscribe to and read blogs, for example, users can be grouped based on their preferences. From this information, the most popular blogs that are read by that group can be identified. Then, for a particular user within the group, a recommendation can be made for the most popular blog that they have not yet read or subscribed to.

Collaborative filtering methods are commonly used to recommend objects or items to customers based on their interests and categorisations. For example, collaborative filtering has been employed in email and document filtering systems, allowing customers to customise their filters based on their preferences. In collaborative filtering, algorithms identify close neighbours of active customers, using patterns that express customers' preferences and interactions to match them with other customers who share similar information and documents. Once a potential match is identified, recommendations are generated using the algorithm. Collaborative filtering algorithms also utilize patterns to predict values for empty cells in amatrix.

CONTENT-BASED FILTERING

Content-based methods provide recommendations by analysing the descriptions of rated items by users and the descriptions of items to be recommended. Numerous algorithms have been proposed for analysing the content of text documents and identifying similarities that can serve as the basis for making recommendations. The main goal of classification learners is to learn a function that predicts the class of a document, while other algorithms may use regression to predict the numeric rating value of a document. There are two key sub-problems in designing a content-based filtering system. The first is finding a representation of documents, and the next is making recommendations for unseendocuments.

The approach described in this paper is based on using deep learning for content-based filtering in recommendation systems. The idea is to use neural networks to compute feature vectors for users and movies, and then use these feature vectors to make recommendations. The neural networks, called the user network and the movie network, take as input the features of users and movies, respectively, and output feature vectors **(v_u and v_m)** that describe the users and movies, respectively.
The user network and the movie network can have multiple layers of dense neural network layers, and the output layer of each network has a fixed size (in this case, 32 units). The feature vectors v_u and v_m are then used to predict the rating of a user on a movie by taking the dot product of the two vectors. This prediction can be used to recommend movies to users.

To train the model, a **cost function** is defined that measures the squared difference between the predicted ratings and the actual ratings given by users for movies. This cost function is used to optimise the parameters of the neural networks (i.e., the weights and biases) using gradient descent or other optimisation algorithms.

Regularisation can also be added to the cost function to prevent overfitting and encourage smaller parameter values.

DESCRIPTION:

Regularisation is a well-known technique used in ma- chine learning algorithms, including content-based filtering, to **prevent overfitting** and improve the generalisation performance of the model. One common regularisation technique used in content-based filtering algorithms is L2 regularisation, also known as **RIDGE REGULARISATION**. It adds a penalty term to the loss function, which is based on the squared L2 norm of the modelparameters.

The L2 regularisation term discourages the model from assigning too much importance to any particular feature during training, as it penalises large parameter values. This helps in preventing the model from becoming overly reliant on a single feature, which can lead to overfitting. By adding a penalty for large parameter values, L2 regularisation promotes a smoother distribution of importance among all the features, ensuring that no single feature dominates the model's decision-making process.

Furthermore, L2 regularisation helps in keeping the model parameters small, as it minimises the squared L2 norm of the parameters. Smaller parameter values can lead to a simpler model, which is less prone to overfitting. This is especially beneficial in content-based filtering, where the model typically relies on feature similarity to make recommendations. Smaller parameter values help in maintaining a balanced and unbiased representation of features, preventing any single feature from having too much influence on the recommendation process.

ADAM'S OPTIMIZER

The Adam optimiser, as an optimisation algorithm, is commonly used in training machine learning models, including content-based filtering algorithms with regularisation. During training, the Adam optimiser updates the model parameters based on the gradients of the loss function concerning the parameters, allowing the model to learn optimal weights for its features.

In content-based filtering algorithms with regularisation, the Adam optimiser can be used to optimise the model parameters while taking into account the regularisation term. The regularisation term is usually added to the loss function as a weighted sum of the original loss function and the regularisation term. The weight of the regularisation term is controlled by a hyper-parameter known as the regularisation strength, which determines the balance between the original loss function and the regularisationterm.

Several Optimisation Algorithms Efficiency

The Adam optimiser can update the model parameters in a way that minimises the overall loss function, including the regularisation term. By considering both the original loss function and the regularisation term, the Adam optimiser can help in finding a set of model parameters that strike a balance between fitting the training data and preventing overfitting. This is achieved by adjusting the model parameters based on the gradients of the loss function while taking into account the regularisation term.

One of the key features of the Adam optimiser is its ability to calculate adaptive learning rates for each parameter based on historical gradients. This allows the optimiser to adaptively adjust the learning rates for different parameters, depending on their update patterns. Additionally, the Adam optimiser includes momentum, which helps in accelerating the optimisation process by accumulating the historical gradients and updating the parameters accordingly.

The adaptive learning rates and momentum of the Adam optimiser are beneficial in optimising the model parameters while considering the regularisation term in content-based filtering algorithms. The adaptive learning rates ensure that the regularisation termis takeninto account during parameter updates, helping in preventing overfitting by controlling the magnitude of parameter updates. The momentum feature of the Adam optimiser also aids in achieving a faster convergence towards the optimal parameter values, while accounting for the regularisationterm.

Overall, using deep learning for content-based filtering allows the model to learn complex patterns in the features of users and movies, and can potentially lead to more accurate recommendations compared to traditional content-based filtering methods that rely on handcrafted features.

RETRIEVAL & RANKING

In today's recommendation systems, selecting a handful of items to recommend from a large catalogue of thou- sands, millions, or even tens of millions of items can be computationally challenging. For example, a movie streaming site may have thousands of movies to choose from, an ad recommendation system may have millions of ads, a music streaming site may have tens of mil- lions of songs, and an online shopping site can have millions or even tens of millions of products.

Running neural network inference on such a massive number of items for each user visit can become compu- tationally infeasible. Processing thousands or millions of items every time a user visits a website to determine which products to recommend is resource-intensive and inefficient.

To address this challenge, we used a two-step approach called **retrieval and ranking**. This two-step approach enables more efficient computation by reducing the number of items that need to undergo resource-intensive processing, making large-scale recommendation systems computationally feasible while maintaining the accuracy and relevance of the recommendations.

RETRIEVAL

In the retrieval step of recommendation systems, the goal is to generate a large list of plausible item candidates that can cover a wide range of possible recommendations for the user. This step is designed to be efficient and quick, as it involves pre-computing similarity scores or other relevant features for items.

One common approach in the retrieval step is to com- pute similarity scores between items based on various criteria, such as content-based features, collaborative filtering, or user-item interactions. For example, in a movie streaming site, the retrieval step may involve computing similarity scores between movies based on genre, director, actors, or other relevant features. larly, in an ad recommendation system, the retrieval step may involve computing similarity scores between ads based on contextual information, user preferences, or adattributes.

These pre-computed similarity scores or features allow for fast retrieval of a large list of item candidates that are likely to be relevant to the user. For example, the retrieval step may identify the most similar movies to a

Several Optimisation Algorithms Efficiency

The Adam optimiser can update the model parameters in a way that minimises the overall loss function, including the regularisation term. By considering both the original loss function and the regularisation term, the Adam optimiser can help in finding a set of model parameters that strike a balance between fitting the training data and preventing overfitting. This is achieved by adjusting the model parameters based on the gradients of the loss function while taking into account the regularisation term.

One of the key features of the Adam optimiser is its ability to calculate adaptive learning rates for each parameter based on historical gradients. This allows the optimiser to adaptively adjust the learning rates for different parameters, depending on their update patterns. Additionally, the Adam optimiser includes momentum, which helps in accelerating the optimisation process by accumulating the historical gradients and updating the parameters accordingly.

The adaptive learning rates and momentum of the Adam optimiser are beneficial in optimising the model parameters while considering the regularisation term in content-based filtering algorithms. The adaptive learning rates ensure that the regularisation termis takeninto account during parameter updates, helping in preventing overfitting by controlling the magnitude of parameter updates. The momentum feature of the Adam optimiser also aids in achieving a faster convergence towards the optimal parameter values, while accounting for the regularisationterm.

Overall, using deep learning for content-based filtering allows the model to learn complex patterns in the features of users and movies, and can potentially lead to more accurate recommendations compared to traditional content-based filtering methods that rely on handcrafted features.

RETRIEVAL & RANKING

In today's recommendation systems, selecting a handful of items to recommend from a large catalogue of thou- sands, millions, or even tens of millions of items can be computationally challenging. For example, a movie streaming site may have thousands of movies to choose from, an ad recommendation system may have millions of ads, a music streaming site may have tens of mil- lions of songs, and an online shopping site can have millions or even tens of millions of products.

Running neural network inference on such a massive number of items for each user visit can become compu- tationally infeasible. Processing thousands or millions of items every time a user visits a website to determine which products to recommend is resource-intensive and inefficient.

To address this challenge, we used a two-step approach called **retrieval and ranking**. This two-step approach enables more efficient computation by reducing the number of items that need to undergo resource-intensive processing, making large-scale recommendation systems computationally feasible while maintaining the accuracy and relevance of the recommendations.

RETRIEVAL

In the retrieval step of recommendation systems, the goal is to generate a large list of plausible item candidates that can cover a wide range of possible recommendations for the user. This step is designed to be efficient and quick, as it involves pre-computing similarity scores or other relevant features for items.

One common approach in the retrieval step is to com- pute similarity scores between items based on various criteria, such as content-based features, collaborative filtering, or user-item interactions. For example, in a movie streaming site, the retrieval step may involve computing similarity scores between movies based on genre, director, actors, or other relevant features. larly, in an ad recommendation system, the retrieval step may involve computing similarity scores between ads based on contextual information, user preferences, or adattributes.

These pre-computed similarity scores or features allow for fast retrieval of a large list of item candidates that are likely to be relevant to the user. For example, the retrieval step may identify the most similar movies to a given movie, or the top movies in a user's most viewed genres or country. This large list of plausible item candidates provides a diverse set of recommendations that can cover a wide range of user interests and preferences.

The efficiency of the retrieval step is crucial in handling large catalogues of items, as it helps to quickly narrow down the list of candidates from thousands or millions of items to a more manageable subset for further processing in the ranking step. This allows the recommendation system to provide timely and relevant recommendations to users without incurring excessive computational costs.

RANKING

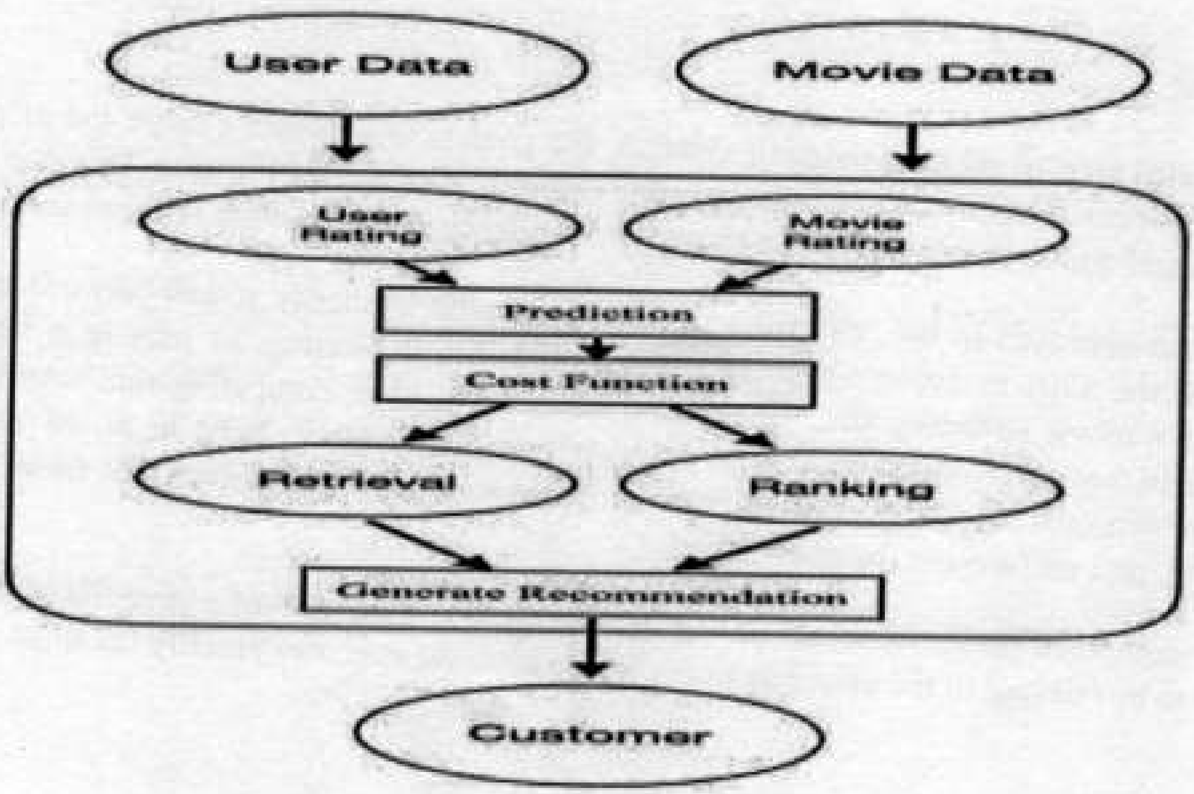
Once the retrieval step is completed, the list of retrieved items is combined and filtered to remove duplicates, items the user has already interacted with, or items that may not be relevant to the user based on their history or preferences. This filtered list of items is then passed to the ranking step, where the learned model, such as a neural network, is utilised to compute predicted ratings or relevance scores for each user-item pair. This step involves running the neural network inference on the filtered list of items, which is typically smaller than the original catalogue of items, making it computationallyfeasible.

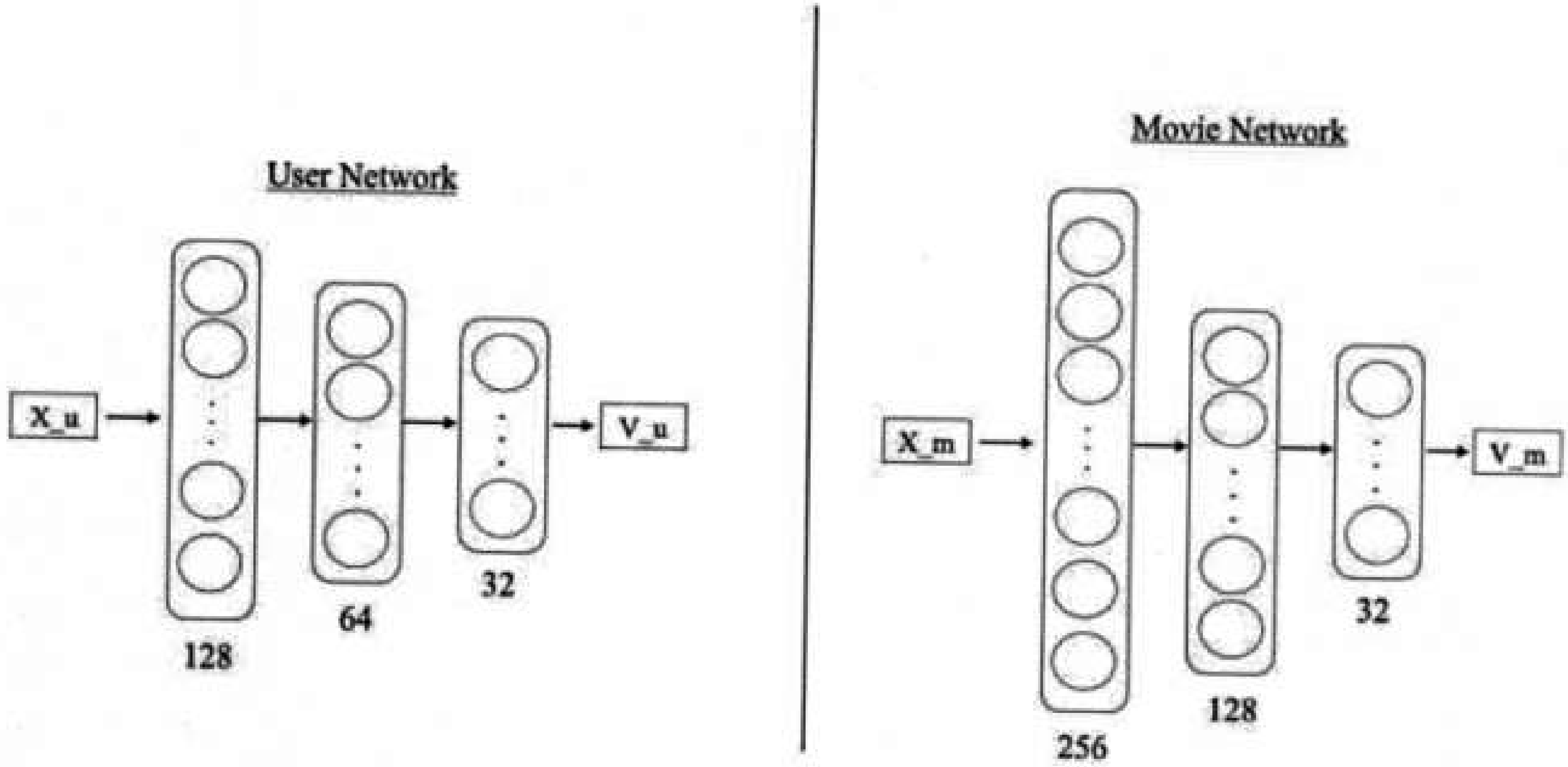
An additional optimisation that can be employed is to pre-compute features or representations for all items in the catalogue in advance. This way, during the ranking step, only the user feature vector needs to be computed in real time, and the inner product between the user feature vector and the pre-computed item feature vectors can be quickly calculated. This approach can further enhance the efficiency of the ranking step and streamline the recommendation process.

Efficiently handling large catalogues of items is critical in modern recommendation systems, as it allows for quick and relevant recommendations without incurring excessive computational costs. The retrieval step, by generating a large list of plausible item candidates and filtering them based on user-specific criteria, helps to narrow down the options for further processing in the ranking step. Additionally, pre-computing features or representations for items in advance can significantlyreduce the computational overhead during the ranking step, making it more feasible for real-time recommendation scenarios.

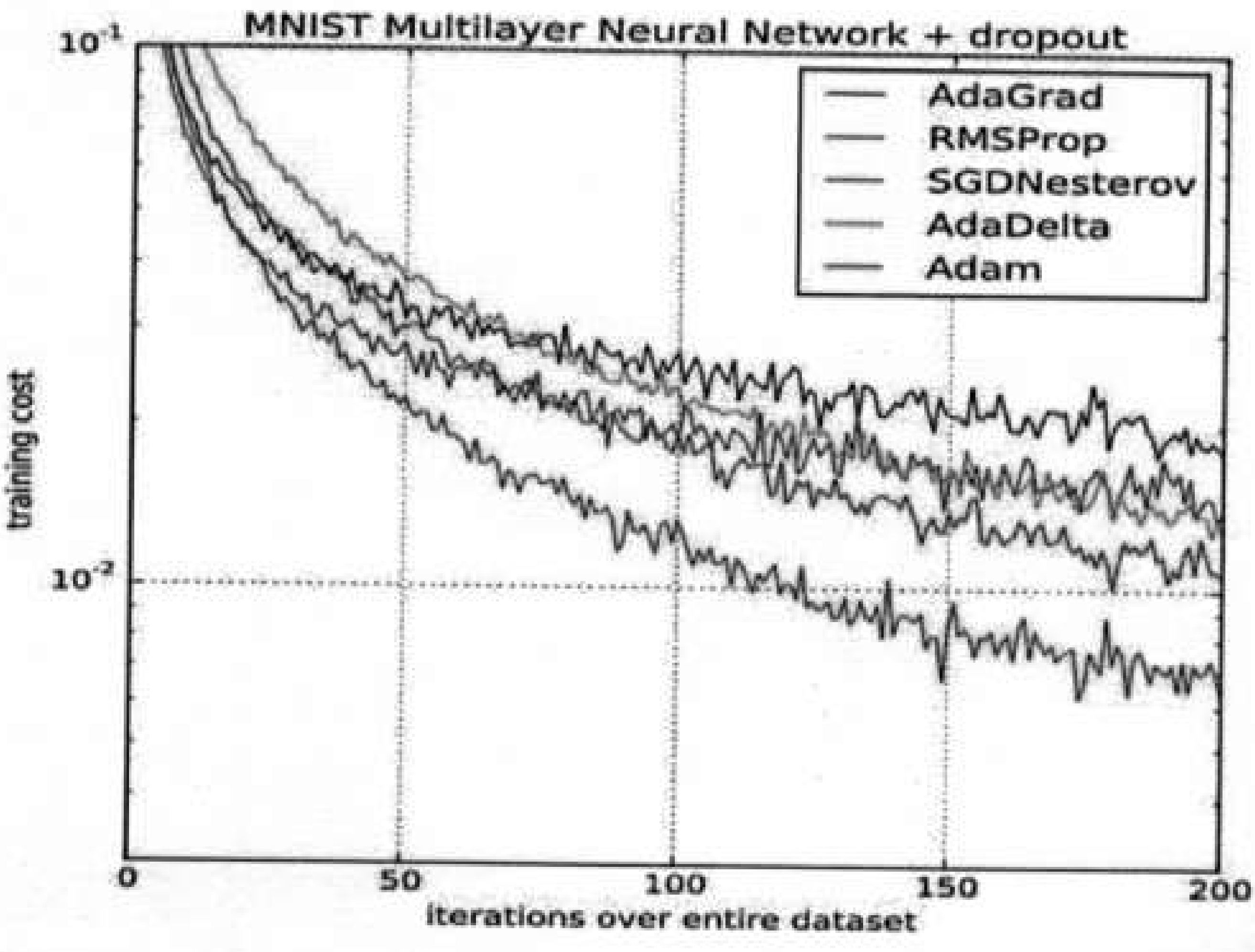
In conclusion, the combination of retrieval and ranking steps, along with pre-computing item features, is the approach that is used in recommendation systems to efficiently handle large catalogues of items. These optimisations enable the recommendation process to be computationally feasible and provide timely and relevant recommendations to users in various online applications, such as movie streaming sites, ad recommendation systems, music streaming sites, and online shoppingsites.

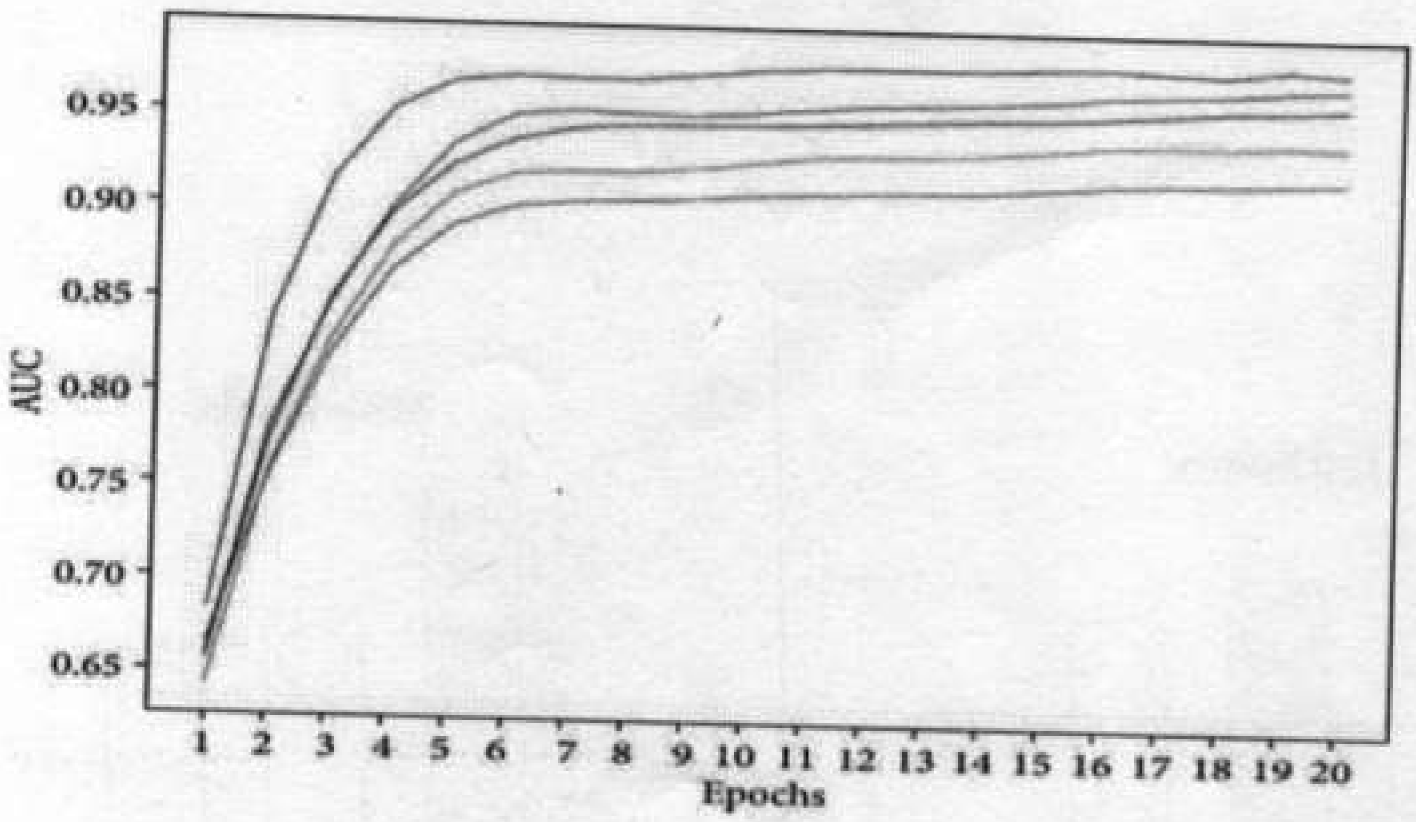
DIAGRAM:





Prediction : V_u (j) . V_m (i)




0.95
0.90
0.85
0.80
0.75
0.70
0.65
AUC
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Epochs