RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEURE ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ IBN KHALDOUN - TIARET

MÉMOIRE

Présenté à:

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de:

MASTER

Spécialité: **Génie logiciel**

Par:

TAYEBI Imen
SMAILI Hayet

Sur le thème

Artificial Neural Networks for a Hybrid Recommendation System

---

Soutenu publiquement le 07 / 12 / 2020 à Tiaret devant le jury composé de :

Mr. KOUADRIA Abderrahmane	Grade	Maître assistant	Président
Mr. BOUDAA Boudjemaa	Grade	Maître de conférences	Encadreur
Mr. AID Lahcene	Grade	Maître de conférences	Examinateur

2019-2020

Acknowledgments

First of all, I would like to express my gratitude to Almighty Allah to enabling us to complete this thesis. Apart from the efforts of myself, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I would like to show my greatest appreciation to my research supervisor Mr. BOUDAA Boudjemaa. I can't say thank you enough for his support and help. Without his encouragement and guidance, this project would not have materialized.

Also, for the guidance and support received from my partner TAYEBI Imen who contributed to this project for its success. I am grateful for her constant support and help.

Lastly, I would like to express my sincere gratitude toward my parents for their kind co-operation and encouragement throughout my years of study and through my process of life, to my brother SMAILI Khalid for his support for me. To my bests friends: BOURWINA Khadidja, ZARHOUNI Kheira, and TRIKI Amina for their unconditional love, prayers, caring, and comfort.

My thanks and appreciations also go to my classmates and to the people who have willingly helped me with their abilities. Thank you.

Smaili Hayat

First and foremost, praises and thanks to Allah, the Almighty, for His showers of blessings throughout our research work to complete it successfully.

I would like to express my deep and sincere gratitude to my research supervisor, Mr.BOUDAA Boujemaa. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me.

I am extending my heartfelt thanks to my partner SMAILI Hayet for her support and cooperation during this research work and thesis preparation.

Also I would like to express my deep thanks to my father and mother for their unconditional support, love and prayers, for their soulful and precious support and for their constant standing by me, to my dear sisters Wiem and Ayat, for the days and nights that we stayed together, for their empathy, friendship and continuous encouragement, to my two younger brothers Mohammed and Ali for their encouragement and the fun days that we have. Thank you for your patience with me during my academic career and through my process of life.

Lastly and not last, my thanks and appreciations also go to my classmates and to the people who have willingly helped me with their abilities. Thank you.

T*ayebi Imen*

TABLE OF CONTENTS

Abstract

In the context of recommendation systems, metadata information from reviews written for businesses has rarely been considered in traditional systems developed using content-based and collaborative filtering approaches. Collaborative filtering and content-based filtering are popular memory-based methods for recommending new products to the users but suffer from some limitations and fail to provide effective recommendations in many situations. In this paper, we present a deep learning neural network framework that utilizes reviews in addition to content-based features to generate model-based predictions for the business-user combinations. We show that a set of content and collaborative features allows for the development of a neural network model with the goal of minimizing the rating misclassification error using stochastic gradient descent optimization algorithm. We empirically show that the hybrid approach is a very promising solution when compared to standalone memory-based collaborative filtering method.

Keywords:

Collaborative Filtering, Content-based Filtering, Artificial neural networks, Recommender systems

ملخص

في سياق أنظمة التوصية، نادرًا ما يتم النظر في معلومات البيانات الوصفية من المراجعات المكتوبة للشركات في الأنظمة التقليدية التي تم تطويرها باستخدام مناهج التصفية القائمة على المحتوى والتعاونية. تعتبر التصفية التعاونية والتصفية القائمة على المحتوى من الأساليب الشائعة المستندة إلى الذاكرة للتوصية بمنتجات جديدة للمستخدمين ولكنها تعاني من بعض القيود وتفشل في تقديم توصيات فعالة في العديد من المواقف. في هذه الأطروحة، نقدم إطار عمل شبكة عصبية للتعلم العميق يستخدم المراجعات بالإضافة إلى الميزات المستندة إلى المحتوى لإنشاء تنبؤات تستند إلى نموذج لمجموعات مستخدمي الأعمال. لقد أظهرنا أن مجموعة من المحتوى والميزات التعاونية تسمح بتطوير نموذج شبكة عصبية بهدف تقليل خطأ التصنيف الخاطئ باستخدام خوارزمية تحسين النسب العشوائية للتدرج. لقد أظهرنا بشكل تجريبي أن النهج الهجين هو حل واعد للغاية عند مقارنته بطريقة التصفية التعاونية القائمة على الذاكرة .

Résumé

Dans le contexte des systèmes de recommandation, les métadonnées des avis rédigés pour les entreprises ont rarement été prises en compte dans les systèmes traditionnels développés à l'aide d'approches de filtrage basées sur le contenu et collaboratives. Le filtrage collaboratif et le filtrage basé sur le contenu sont des méthodes populaires basées sur la mémoire pour recommander de nouveaux produits aux utilisateurs, mais souffrent de certaines limitations et ne parviennent pas à fournir des recommandations efficaces dans de nombreuses situations. Dans cette thèse, nous présentons un cadre de réseau neuronal d'apprentissage en profondeur qui utilise des critiques en plus des fonctionnalités basées sur le contenu pour générer des prédictions basées sur des modèles pour les combinaisons entreprise-utilisateur. Nous montrons qu'un ensemble de contenus et de fonctionnalités collaboratives permet le développement d'un modèle de réseau neuronal dans le but de minimiser l'erreur de classification erronée en utilisant un algorithme d'optimisation de la descente de gradient stochastique. Nous montrons empiriquement que l'approche hybride est une solution très prometteuse par rapport à la méthode de filtrage collaborative basée sur la mémoire autonome.

LIST OF FIGURES

LIST OF TABLES

General introduction

General introduction

1 Background and Problem Statement

The World Wide Web have presented to us overabundance information in shifted fields and of the information or data over-burdening, it is extremely exhausting to discover related information. So, Recommendation System comes into existence. Recommended systems or recommendation systems are a subset of information filtering system that used to expect the evaluation or preference that user would receive to an item. Recommended systems are further available for business experts, jokes, restaurants, financial services, life insurance and twitter followers. Recommended systems based on many techniques, the two traditional ones are content-based and collaborative filtering. While both methods have their advantages, they also have certain disadvantages like cold start and data-sparsity, which can be solved by combining both techniques to improve the quality of the recommendation. The resulting system is known as a hybrid recommender system. Recent research has demonstrable that a hybrid approach could be preferably effective in some cases. Basically, the main target of hybrid approach is to aggregate collaborative filtering and content-based filtering to improve recommendation accuracy.

As far as we know in this era of Web 2.0, user reviews are an integral part of web services like Trip Advisor, Amazon, Epinions and Yelp, where users can post their opinions about businesses, products and services through reviews consisting of free-form text or numeric as in the votes, and a numeric star rating usually in range of 1 to 5. These online reviews function as the 'online word-of-mouth' and a criterion for consumers to choose between similar products. Studies show that they have a significant impact on consumer purchase decisions as well as on product sales and business revenues. In general, an efficient recommender system should be able to model and capture the complex, nonlinear relationships between users and businesses, review and votes. ANNs are particularly well-suited to learn about these relationships. So the main research question of this thesis is:

Can we develop an ANN model for a Hybrid recommendation system

to predict the rating?

2 Approach

In this thesis we presents an Artificial Neural Network (ANN) model for a hybrid recommendation system that brings together content (user and business), collaborative (review and votes) features to train a deep learning model using ANNs to predict the rating and evaluate the improvement in recommendation predictions. Further, we demonstrate how well the proposed technique works by performing computational experiments using the Yelp Academic Dataset.

3 Outline

This thesis is structured in three chapters besides a general introduction and a general conclusion:

1. **Introduction**: An initiation to the background and problem statement, and the approach utilized in this thesis.

2. **Chapter one**: A theoretical introduction to recommender systems in general then we based on the hybrid recommender system its methods and challenges, therefore the issues that this approach overcomes and improves.

3. **Chapter two**: In this chapter we discussed the background of ANN (ML and DL), the BNN and the difference between it and ANN, types of ANN and the general concepts of ANN.

4. **Chapter three**: Takes an in-depth dive into the proposed approach a hybrid recommendation system using ANN and explore the experiments on our model using Yelp dataset.

5. **Conclusion**: It summarizes the contributions of this thesis, limitations of our work and present future research work.

Chapter – 1 – Hybrid recommender systems

1. Introduction

This chapter introduces recommender systems (commonly called RecSys), tools that recommend items to users. Many of the most popular uses of recommender systems involve to suggesting products to customers. Amazon, for example, uses recommender systems to choose which retail products to display. Recommender systems have changed the way people find products, information, and even other people. They study patterns of behavior to know what someone will prefer from among a collection of things he has never experienced. The technology behind recommender systems has evolved over the past 20 years into a rich collection of tools that enable the practitioner or researcher to develop effective recommenders.

2. A bit of background

Before we dive into the details, we need to set the stage and clarify some basic vocabulary: The three basic data sources for a recommender system are users, items, and the interactions among them. Interactions can be either implicit or explicit.

1.1 Explicit feedback

Is when a user directly likes or rates an item. Examples of explicit feedback include rating a movie on a 1–5 Likert scale, "liking" a friend's Facebook post, or writing a positive review for a restaurant [1].

1.2 Implicit feedback

Is more subtle and attempts to infer a user's preferences based on their indirect behaviour. Examples of implicit feedback include a user's purchase history, how many times they replayed a song, how quickly they binge-watched a series, or how long they read an article for [1].

3. What Is a Recommender System?

Recommender systems are a subclass of information filtering systems that present users with items he or she might be interested in based on preferences and behavior. They seek to predict your appreciation of an item and suggest the ones you are more likely to appreciate [2].

4. History of Recommender Systems

History: Before 1992

- Content Filtering
    - An architecture for large scale information systems (1985)
    - A rule-based message filtering system (1988)

History: 1992-1998

- Tapestry by Xerox Palo Alto (1992)
    - First system designed by collaborative filtering
- Grouplens (1994)
    - First recommender system using rating data

History: 1992-1998

- Fab : content-based collaborative recommendation
    - First unified recommender system
- Empirical Analysis of Predictive Algorithms for Collaborative Filtering (1998)
    - Systematically evaluate user-based collaborative filtering

History: 1999-2005

- Amazon proposed item-based collaborative filtering (Patent is filed in 1998 and issued in 2001)
- Pandora began music genome project (2000)
- Lastfm using Audioscrobbler to generate user taste profile on musics.

History: 2005-2009

- Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.
- Netflix Prize
    - Latent Factor Model (SVD, RSVD, NSVD, SVD++)
    - Temporal Dynamic Collaborative Filtering

- Yehuda Koren's team get prize
- Digg, Youtube try recommender system.

History: 2010-now

- Context-Aware Recommender Systems
- Music Recommendation and Discovery
- Recommender Systems and the Social Web
- Information Heterogeneity and Fusion in Recommender Systems
- Human Decision Making in Recommender Systems
- Personalization in Mobile Applications
- Novelty and Diversity in Recommender Systems
- User-Centric Evaluation
- Facebook launches instant personalization (2010)
  - Clicker
  - Bing
  - Trip Advisor
  - Rotten Tomatoes
  - Pandora [3]

5. Recommender System techniques

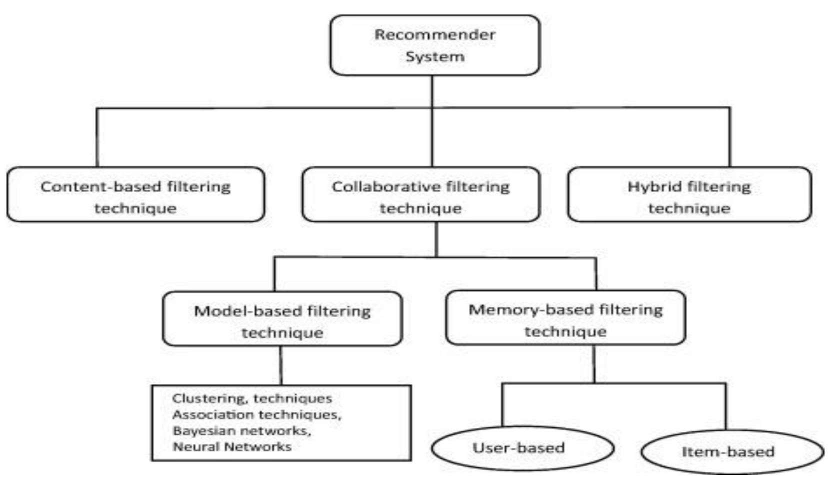


Figure 1 Recommender System techniques

As shown in Figure 1, the techniques in recommender systems is broadly classified into three categories:

- Collaboratives filtering techniques rely on user activity like ratings on items or buying patterns.
- Content based filtering techniques rely on user activity attributes like keywords during search or their profiles.
- Hybrid filtering techniques combine both above techniques to overcome their limitations and improves performance [4].

In the following sections we will cover each of the techniques and their limitations:

1.3 Collaborative Filtering Recommender System

Collaborative filtering Algorithm recommender system became one of the most researched techniques of recommender systems. if users shared the same interests in the past, they will also have similar tastes in the future. So, for example, if user A and user B have a purchase history that overlaps strongly and user A has recently bought an item that B has not yet been, the basic rationale is to propose this item also to B [5].

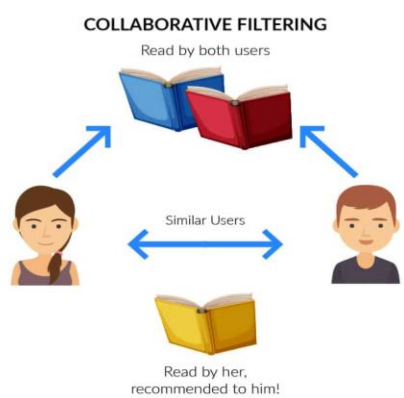


Figure 2 Collaborative Filtering Recommender System

Collaborative filtering models are two types:

1.3.1 Memory-based approaches

Compute the correlations between users and items to produce a preference score that predicts the likelihood of a user acquiring an item in the future and provide corresponding recommendation [6].

1.3.1.1 User-Based Approach

In the User-based approach, the user plays an important role. If a certain majority of the customer have the same taste then they join into the one group. Recommendations are given to user based on the evaluation of items by other users form the same group, with whom he/she shares common preferences. If the item was positively rated by the community, it will be recommended to the user [5].

1.3.1.2 Item-Based Approach

Here in Item-Based Approach, the items play an important role Recommendations are based on the evaluation of items. The system generates recommendations with items in the neighborhood that a user would prefer [5].

1.3.2 Model based approaches

These methods use predictive data mining and machine learning techniques to make recommendations. In case of parametrized models, these parameters are learned using optimization frameworks. Decision trees, Rule-based models, Bayesian methods and latent factor models are some examples. These models have high level of prediction coverage even for sparse ratings. However, these methods tend to be heuristic and don't perform well under all settings [4].

1.4 Content-Based Recommender System

These methods use both content description or descriptive attributes of items and user activity to make recommendations. Content-based methods works better for new items in the system since they find the similar items based on items descriptive attributes which are rated by active user [4].

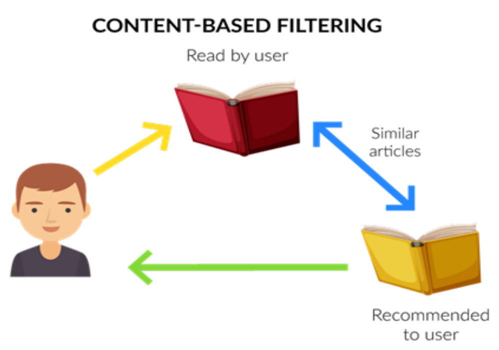


Figure 3 Content-Based Recommender System

1.5 Hybrid Recommender System

This is the best recommender system that combines two or more recommendation techniques for better performance with some drawbacks of any individual one. Various aspects from the above-mentioned recommender systems can be used to achieve the best results [4].

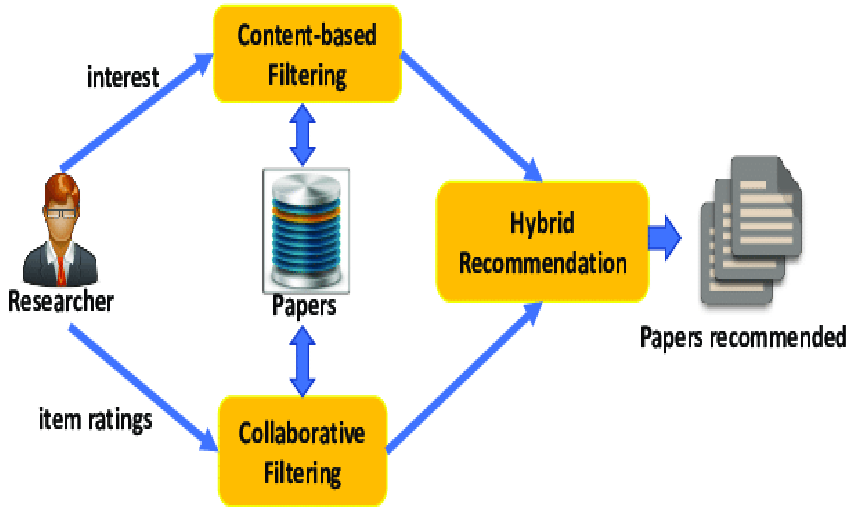


Figure 4 Hybrid Recommender System

6. Comparison of existing Recommender System techniques.

Type	Pros	Cons
Collaborative filtering	• No domain knowledge necessary • When data of № of users and № of items are available in large amount. It provides better results.	• Cold start problem arises when a new item or a new user is introduced. • Sparsity problems.
Content-based filtering	• It does not require knowledge about other users. It recommends only according to specific user. • Comparison between items is possible	• Cold start problem for new users. • Requires a lot of domain knowledge.
Hybrid filtering	• It overcomes some limitations of collaborative and content-based filtering.	• It is very expensive because it combines the two or more filtering.

Table 1 Comparison of existing recommender techniques.
[7]

7. A hybrid recommendation system using Content-based and Collaborative filtering

Content-based and collaborative recommendation techniques will always suffer from “new item” and/or “new user” problems since both techniques require prior history to produce effective predictions and the hybridization techniques discussed above alleviate some of these limitations. But cold star is just one problem, for example to analyze approximately 8 million ratings between 100,000 users and nearly 1.7 million vehicles. This translates into 5 out of 100,000 possible interactions challenging. So what does a recommender actually aim for? The basic goal of a recommender system is to predict future user-item interactions based on previous interactions and features. Referring to the image above, we want to quantify the question marks to see which car the user is most likely to see next. This goal is denoted as relevancy of recommendations and just one of many like trustworthiness, diversity, or robustness. Scalability is another important and production-oriented goal that forces us to provide recommendations to many users fast (within tens or hundreds of milliseconds). Perfect recommendations don’t matter if you need to grab a coffee while you wait [8].

8. Different Hybridization Methods

Hybrid methods are used to combine different recommendation methods in one or more ways as listed above, in order to achieve more accurate recommender output. Hybrids of trust based system with content-based methods are developed. Each of these hybrid methods are having its own advantages and disadvantages. The accuracy rate in the prediction depends on the type of dataset and other parameters. The weightages given to each of the constituents approaches of the hybrid methods also plays very important roles in deciding the accuracy of RS [9].

1.6 Weighted

A weighted hybrid recommender is one in which the score of a recommended item is computed from the results of all of the available recommendation techniques present in the system. For example, the simplest combined hybrid would be a linear combination of recommendation scores. The P-Tango system uses such a hybrid. It initially gives collaborative and content-based recommenders equal weight, but gradually adjusts the weighting as predictions about user ratings are confirmed or disconfirmed. Pazzani’s combination hybrid does not use numeric scores, but rather treats the output of each recommender (collaborative, content-based and demographic) as a set of votes, which are then combined in a consensus scheme. The benefit of a weighted hybrid is that all of the system's capabilities are brought to bear on the recommendation process in a straightforward way and it is easy to perform post-hoc credit assignment and adjust the hybrid accordingly. However, the implicit assumption in this technique is that the relative value of the different techniques is more or less uniform across the space of possible items. From the discussion above, we know that this is not always so: a collaborative recommender will be weaker for those items with a small number of raters [9].

1.7 Switching

Method Here in Switching method system uses some criterion to switch between recommendation techniques. The DailyLearner system uses a content/collaborative hybrid in which a content- based recommendation method is applied first. If the content-based system cannot make a recommendation with sufficient confidence, then a collaborative recommendation is attempted. this switching hybrid does not completely avoid problem [5].

1.8 Mixed Method

When large recommendations take place the mixed method come into the action. Here in this method is used in Television System used. First of all content based method is used for textual description of TV-shows and use of collaborative method for finding the preferences of the user and Recommendations from the two techniques lead to suggest a final program. With the help of this mixed method new item -start up problem can be overcome: the content-based component can be relied on to recommend new shows on the basis of their descriptions even if they have not been rated by anyone. It does not get around the "new user" start-up problem, since both the content and collaborative methods need some data about user preferences to get off the ground, but if such a system is integrated into a digital television, it can track what shows are watched (and for how long) and build its profiles accordingly [5].

1.9 Feature Combination

Another way to achieve the content/collaborative merger is to treat collaborative information as simply additional feature data associated with each example and use content-based techniques over this augmented data set. For example, Basu, Hirsh & Cohen report on experiments in which the inductive rule learner Ripper was applied to the task of recommending movies using both user ratings and content features, and achieved significant improvements in precision over a purely collaborative approach. However, this benefit was only achieved by hand filtering content features. The authors found that employing all of the available content features improved recall but not precision. The feature combination hybrid lets the system consider collaborative data without relying on it exclusively, so it reduces the sensitivity of the system to the number of users who have rated an item. Conversely, it lets the system have information about the inherent similarity of items that are otherwise opaque to a collaborative system [9].

1.10 Cascade

Here in Cascade method it involves the stage process In this technique, one recommendation technique is employed to produce a ranking of candidates and a second technique refines the recommendation from the candidate set. The restaurant recommender EntreeC, described below, is a cascaded knowledge-based and collaborative recommender. Like Entree, it uses its knowledge of restaurants to make recommendations based on the userÊs stated interests. The recommendations are placed in buckets of equal preference, and the collaborative technique is employed to break ties, further ranking the suggestions in each bucket [5].

1.11 Feature Augmentation

To improve the performance of a core system Feature Augmentation is used. For example Libra System makes content based recommendations of books based on data found in Amazon.com, using a naïve-bayes text classifier and this help in finding the quality of books. In Feature Augmentation one technique is used to produce a rating of an item and that information is then incorporated into the processing of the next recommendation technique. So the difference between the Cascade and augmentation are as follows: in feature augmentation the feature used by second recommendation is the one which is the output of the first one where as in cascading second recommender doesnot use the output of first one but the results of the two recommenders are combined in a prioritized manner [5].

1.12 Meta-level

Another way that two recommendation techniques can be combined is by using the model generated by one as the input for another. We use a learned model to generate features for input to a second algorithm in a meta-level hybrid, the entire model becomes the input. The first meta-level hybrid was the web filtering system Fab. In Fab, user-specific selection agents perform content-based filtering using Rocchio's method to maintain a term vector model that describes the user's area of interest. Collection agents, which garner new pages from the web, use the models from all users in their gathering operations. So, documents are first collected on the basis of their interest to the community as a whole and then distributed to particular users. In addition to the way that user models were shared, although the collaborative step only created a pool of documents and its ranking information was not used by the selection component. A meta-level hybrid that focuses exclusively on recommendation is described by Pazzani as "collaboration via content". A content-based model is built by Winnow for each user describing the features that predict restaurants the user likes. These models, essentially vectors of terms and weights, can then be compared across users to make predictions. More recently, Condliff have used a two-stage Bayesian mixed-effects scheme: a content-based naive Bayes classifier is built for each user and then the parameters of the classifiers are linked across different users using regression. LaboUr uses instance-based learning to create content-based user profiles which are then compared in a collaborative manner. The benefit of the meta-level method, especially for the content/collaborative hybrid is that the learned model is a compressed representation of a user's interest, and a collaborative mechanism that follows can operate on this information-dense representation more easily than on raw rating data [9].

Hybridization method	Description
Weighted	The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation.
Switching	The system switches between recommendation techniques depending on the current situation.
Mixed	Recommendations from several different recommenders are presented at the same time
Feature combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm.

Cascade	One recommender refines the recommendations given by another.
Feature augmentation	Output from one technique is used as an input feature to another.
Meta-level	The model learned by one recommender is used as input to another.

Table 2 Hybridization Methods**.**
[9]

9. Challenges and Issues

1.13 Cold-Start Problem

It's difficult to give recommendations to new users because his profile is almost empty and he has not rated any items so the taste of user remains unknown to the system so this type of the problem is called as cold-start problem. In some recommender systems this problem can be solve with the survey at the time of creating a profile. Another problem is when user has not rated before when new to the system. Both of these problems can be overcome with the help of hybrid approaches [5].

1.14 Data-Sparsity

Sparsely is the problem of lack of information. Suppose we have a huge amount of users and items but user have rated only few items. If a user has evaluated only few items then it's difficult to determine the taste of the user.so to overcome this we use collaborative and hybrid approach to create neighborhoods of users based on their profiles [5].

1.15 Scalability

Due to increase of numbers of users and items, the system needs more resources for processing information and forming recommendations. Majority of resources is consumed with the purpose of determining users with similar tastes, and goods with similar descriptions. This problem is also solved by the combination of various types of filters and physical improvement of systems [5].

1.16 Gray Sheep

Gray sheep problem means where user does not consistently agree or disagree to the group of the people and due to this reason for such user recommendation seems to be difficult [5].

10.Conclusion

In this chapter, we discussed the recommendation system generally, its definition, history, and techniques. We comprised the different techniques: Collaborative Filtering, Content-Based, and Hybrid Recommender System. We based on the hybrid approach which is taken between context-based filtering and collaborative filtering to implement the system and we mention different Hybridization Methods and challenges and issues that this approach overcomes and improves the performance of the system.

Chapter – 2 – Artificial neural networks

1. Introduction

The long course of evolution has given the human brain many desirable characteristics not present in Von Neumann or modern parallel computers. These include massive parallelism, distributed representation and computation, learning ability, generalization ability, adaptivity, inherent contextual information processing, fault tolerance, and low energy consumption. It is hoped that devices based on biological neural networks will possess some of these desirable characteristics. On this basic the scientists come out with the concept of artificial neural network. An artificial neural network, often just called a neural network, is a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connection approach to computation. Neural networks have emerged in the past few years as an area of unusual opportunity for research, development and application to a variety of real world problems. Indeed, neural networks exhibit characteristics and capabilities not provided by any other technology.

2 Background

2.1 Machine Learning

Machine learning is the branch of artificial intelligence (AI) that provides the ability to learning automatically learn and improve from experience (or simply that teaches computers to do what comes naturally to humans and animals: learn from experience). It was first introduced by Arthur Samuel. Its primary aim is to allow the computer to learn automatically without human involvement or assistance and adjust actions accordingly. Many problems are historical very easy for humans, and very difficult for networks, Machine learning (deep learning in particular) is currently our best solution for many of those problems. For example, medical diagnosis, image processing, prediction, classification, regression, etc [10] [11].

2.1.1 Need for Machine Learning

The demand for machine learning is rapidly increasing day by day. As a human, we have many limitations as we cannot access a large amount of data manually, so for that, we need some computer systems. And machine learning to make things easy for us. It's use cases can easily understand machine learning. Right now, machine learning is used in self-driving cars, cyber fraud detection, face recognition, and friend suggestion on Facebook. Some top companies, like Amazon and Netflix, have built machine learning models, which are using a large amount of data to analyze the user interest and recommend the product correctly. It is also used to finding a hidden pattern and extracting useful information from data [10] [11].

2.1.2 Why Machine Learning Matters?

With the rise in big data, machine learning has become a key technique for solving problems in areas, such as:

- Computational finance, for credit scoring and algorithmic trading.
- Image processing and computer vision, for face recognition, motion detection, and object detection.
- Computational biology, for tumor detection, drug discovery, and DNA sequencing.
- Energy production, for price and load forecasting.
- Automotive, aerospace, and manufacturing, for predictive maintenance.
- Natural language processing, for voice recognition applications [10] [11].

2.1.3 When Should You Use Machine Learning?

Consider using machine learning when you have a complex task or problem involving a large amount of data and lots of variables, but no existing formula or equation **[10] [11]**. For example, machine learning is a good option if you need to handle situations like these:

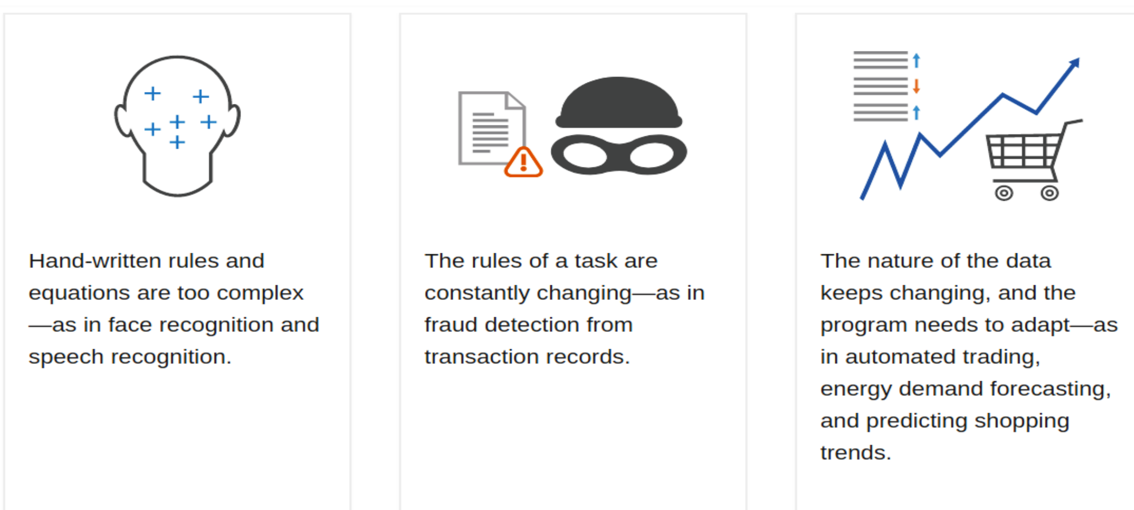


Figure 5 Situations to use Machine Learning.

2.1.4 Types of Machine Learning

- ***Supervised Learning- "Train me!"***

Supervised Learning is the type of machine learning, where we can consider a teacher guides the learning. The dataset which we have will acts as a teacher and use it to train the model and the machines. Once the model gets trained, it starts making the prediction or decision when new data is given to it [12].

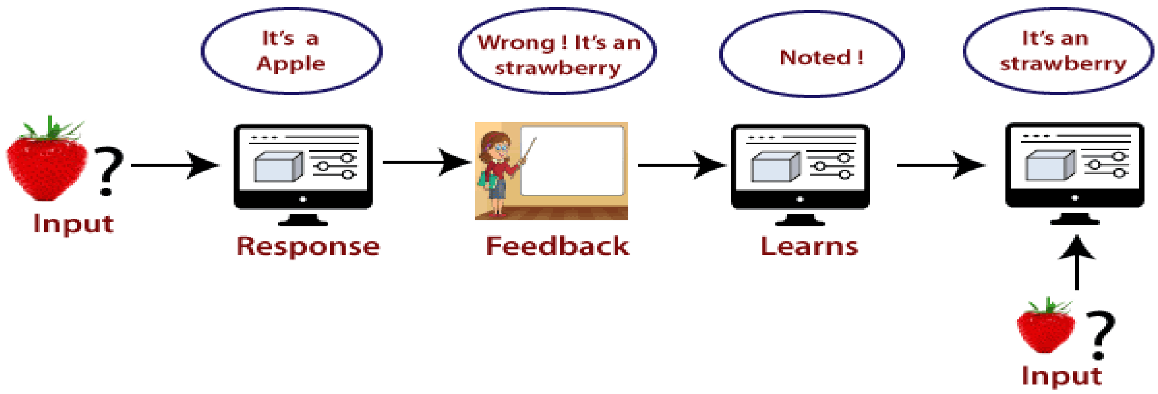


Figure 6 Example of Supervised Learning.

It can be grouped into two types:

- Classification

It is a technique that aims to reproduce class assignments.
It produces the response value, and the data separated into "classes."
Example: Recognition of a type of car in a photo [12].

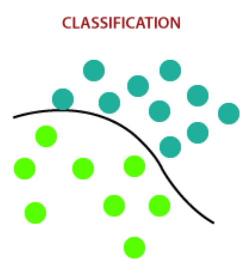


Figure 7 Classification

- Regression

Regression is a technique that aims to produce the output value, we can use it.
Example: Use to predict the price of a different product [12].

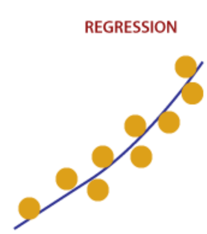


Figure 8 Regression

- ***Unsupervised Learning- "I am self-sufficient in learning!"***

The unsupervised machine learning algorithm is used when the training information is neither classified nor labelled. If the model is given a dataset, it automatically finds patterns and relationships in the dataset by creating clusters in it. Supposed we presented images of apples, bananas, and mangoes to the model, based on some patterns and relationships it creates cluster and divides the dataset into clusters. Now if a new data is delivered to the model, it adds it to one of the generated groups [12].

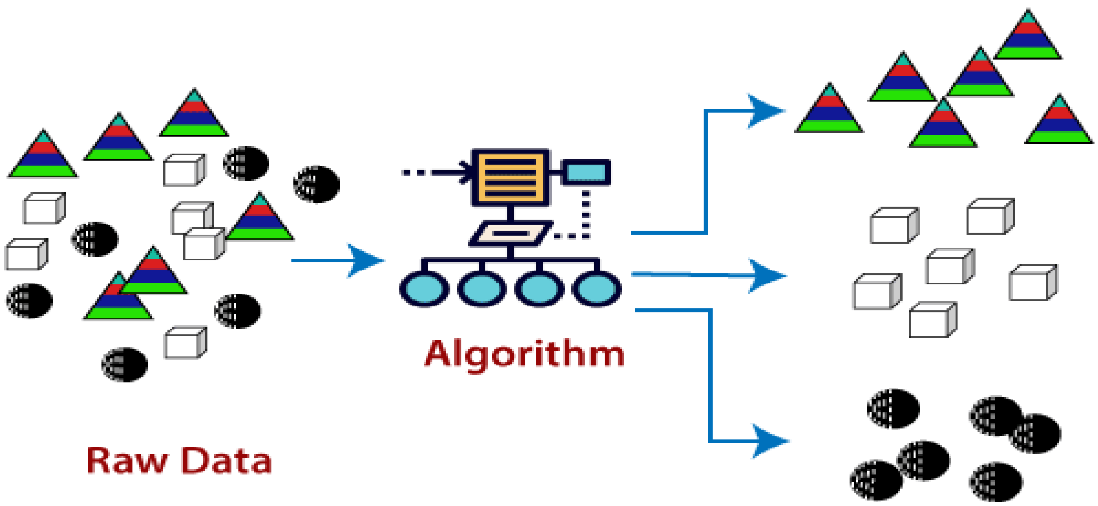


Figure 9 Unsupervised Learning example

It also has two types:

- Clustering

Clustering is used to find likeness and differences in a particular thing. It groups similar things. This algorithm can help us to solve many obstacles.
Example: Create clusters of similar tweets based on their content, find a group of photos with similar cars, or identify different types of news [12].

- Association

Association rules mining is another key of unsupervised data mining method, after clustering, which finds interesting associations (relationships, dependencies) in a large set of data items [12].

♦ ***Semi-Supervised Learning***

It falls somewhere between supervised and unsupervised learning. So, they use both labeled and unlabeled data for training where a small amount of labeled data and a big amount of unlabeled data are used. Commonly, semi-supervised learning is chosen when the acquired labeled data requires skilled and significant resources to train it [12].

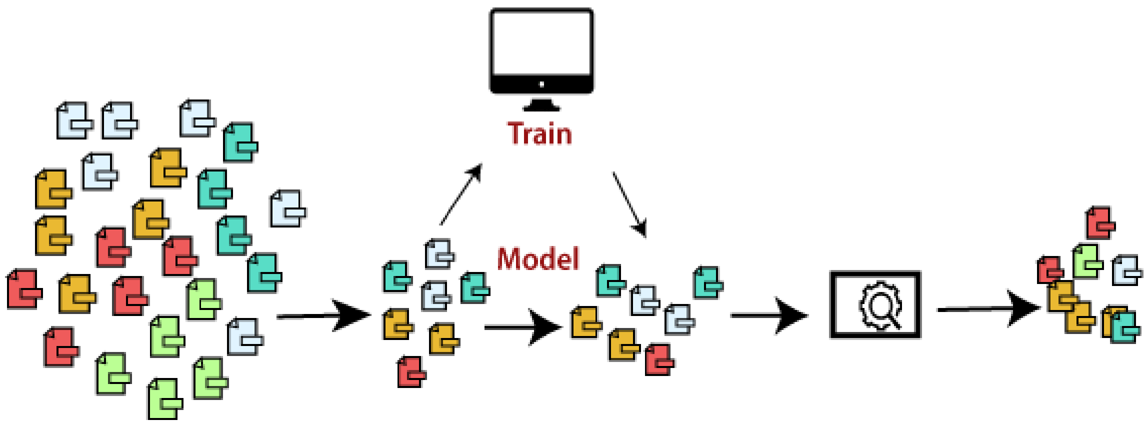


Figure 10 Semi-Supervised Learning

♦ ***Reinforcement Learning- "My life my rules (Hit and Trial)!"***

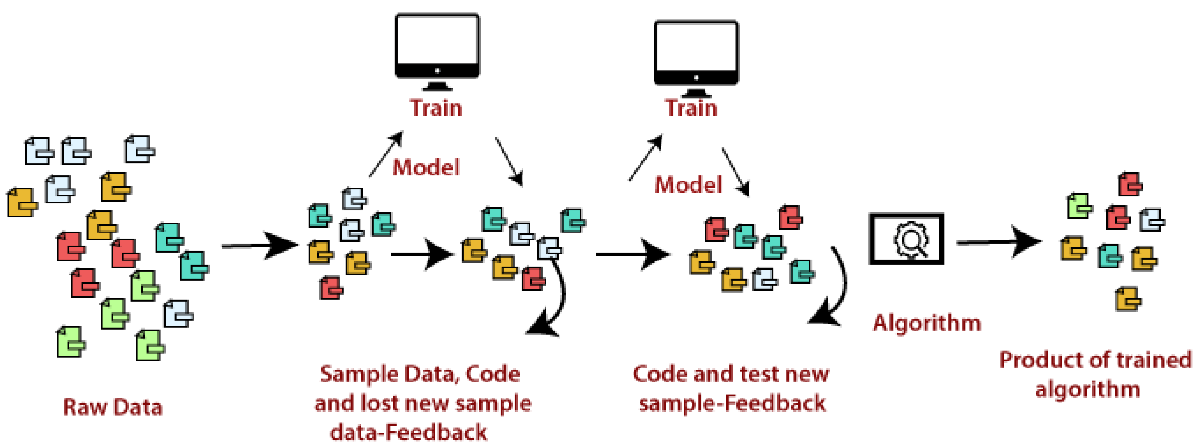


Figure 11 Reinforcement Learning

Reinforcement learning is the ability of an agent to interact with the environment and find out the best outcome. It chases the concept of the hit and trial method. The agent is rewarded or condemned with a point for a correct or a wrong answer, and based on the positive rewards points gained the model train itself. And once again it trained to predict the new data presented to it. The goal of the agent is to get the most reward points and to improve its performance [12].

2.2 Deep Learning

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driver-less cars, enabling them to recognize a stop sign or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers (Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks) [11].

2. Artificial Neural Network

2.3 What is a neural network?

A neural network can be defined as a model of reasoning based on the human brain. The brain consists of a densely interconnected set of nerve cells, or basic information-processing units, called neurons. The human brain incorporates nearly 10 billion neurons and 60 trillion connections, synapses, between them (Shepherd and Koch). By using multiple neurons simultaneously, the brain can perform its functions much faster than the fastest computers in existence today. Although each neuron has a very simple structure, an army of such elements constitutes a tremendous processing power. A neuron consists of a cell body, soma, a number of fibers called dendrites, and a single long fiber called the axon. While dendrites branch into a network around the soma, the axon stretches out to the dendrites and somas of other neurons.

Our brain can be considered as a highly complex, nonlinear, and parallel information-processing system. Information is stored and processed in a neural network simultaneously throughout the whole network, rather than at specific locations. In other words, in neural networks, both data and its processing are global rather than local. Owing to the plasticity, connections between neurons leading to the ‘right answer’ are strengthened while those leading to the ‘wrong answer’ weaken. As a result, neural networks have the ability to learn through experience. Learning is a fundamental and essential characteristic of biological neural networks. The ease and naturalness with which they can learn led to attempts to emulate a biological neural network on a computer.

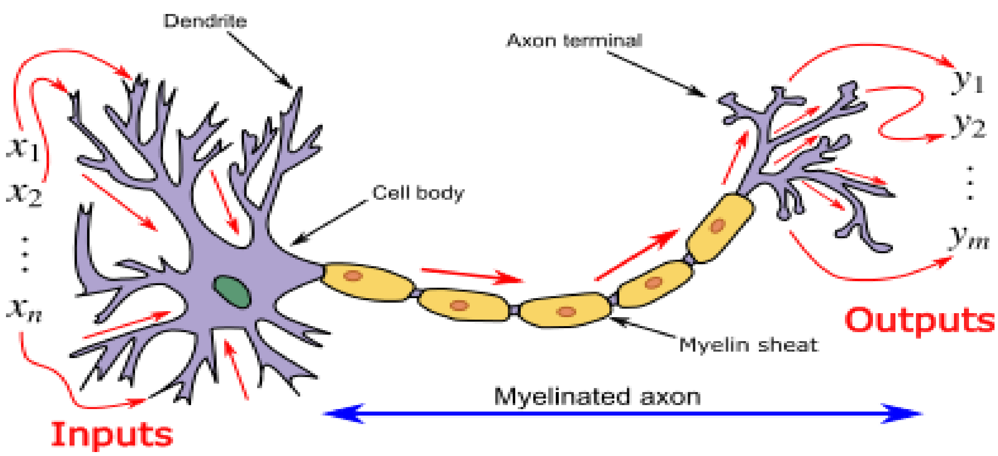


Figure 12 Biological Neuron

Although a present-day artificial neural network (ANN) resembles the human brain much as a paper plane resembles a supersonic jet, it is a big step forward. ANNs are capable of ‘learning’, that is, they use the experience to improve their performance. When exposed to a sufficient number of samples, ANNs can generalize to others they have not yet encountered. They can recognize hand-written characters, identify words in human speech, and detect explosives at airports. Moreover, ANNs can observe patterns that human experts fail to recognize. For example, Chase Manhattan Bank used a neural network to examine an array of information about the use of stolen credit cards and discovered that the most suspicious sales were for women’s shoes costing between $40 and $80 [13].

2.4 ANN versus BNN

Before taking a look at the differences between Artificial Neural Network and Biological Neural Network, let us take a look at the similarities based on the terminology between these two [14].

Biological Neural Network	Artificial Neural Network
Soma	Node
Dendrites	Input
Synapse	Weights or Interconnections
Axon	Output

Table 3 ANN versus BNN.

The following table shows the comparison between ANN and BNN based on some criteria mentioned [14].

Criteria	BNN	ANN
Processing	Massively parallel, slow but superior than ANN.	Massively parallel, fast but inferior than BNN.
Size	10^{11} neurons and 10^{15} Interconnections.	10^2 to 10^5 nodes.
Learning	They can tolerate ambiguity.	Very precise, structured and formatted data is required to tolerate ambiguity.
Fault tolerance	Performance degrades with even partial damage.	It is capable of robust performance, hence has the potential to be fault tolerant.
Storage capacity	Stores the information in the Synapse.	Stores the information in continuous memory locations.

Table 4 The comparison between ANN and BNN.

2.5 How do artificial neural nets model the brain?

An artificial neural network consists of a number of very simple and highly interconnected processors, also called neurons, which are analogous to the biological neurons in the brain. The neurons are connected by weighted links passing signals from one neuron to another. Each neuron receives a number of input signals through its connections; however, it never produces more than a single output signal. The output signal is transmitted through the neuron's outgoing connection (corresponding to the biological axon). The outgoing connection, in turn, splits into a number of branches that transmit the same signal (the signal is not divided among these branches in any way). The outgoing branches terminate at the incoming connections of other neurons in the network [13].

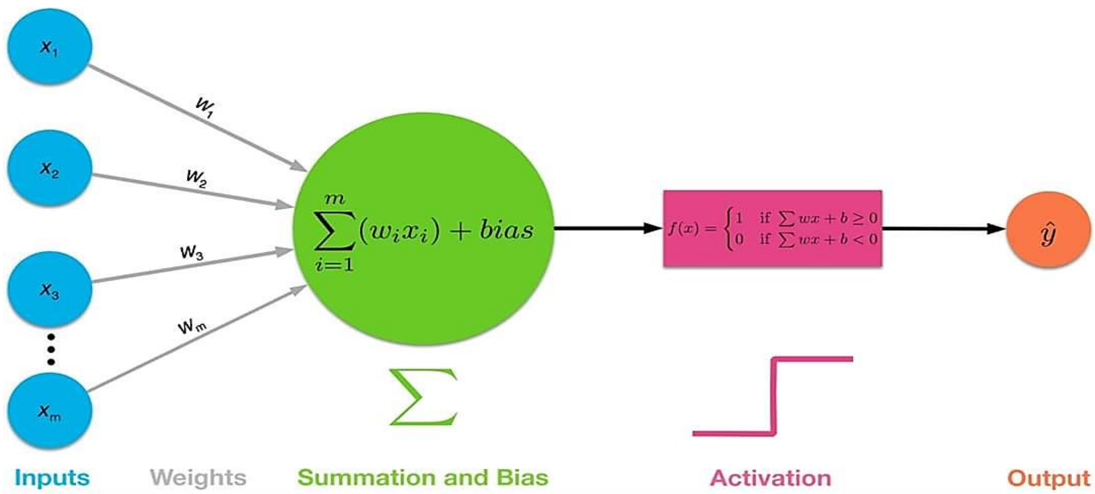


Figure 13 Elements of neuron of ANN**.**

And basically, each neuron from a network can be implemented as shown above and it is possible to see that the artificial neuron is composed of six basic elements, namely:

- Inputs: it represents features and basically the dataset entering to the networks.
- Weight: It represents the dimension or strength of the connection between units. If the weight to node 1 to node 2 has a higher quantity, then neuron 1 has a more considerable influence on the neuron 2.
- Bias: It is the same as the intercept added in a linear equation. It is a special neuron added to each layer in the neural network, which simply stores the value of 1 which task is to modify the output along with the weighted sum of the input to the other neuron.
- Net sum: It calculates the total sum.
- Activation Function: A neuron can be activated or not, is determined by an activation function. The activation function calculates a weighted sum and further adding bias with it to give the result [15].
- Output: Consists of the final value produced by the neuron given a particular set of input signals [16].

2.6 Main Architectures of Artificial Neural Networks

In general, an artificial neural network can be divided into three parts, named layers, which are known as [16]:

- **Input Layer** This layer is responsible for receiving information (data), signals, features, or measurements from the external environment. These inputs (samples or patterns) are usually normalized within the limit values produced by activation functions. This normalization results in better numerical precision for the mathematical operations performed by the network.
- **Hidden Layer** These layers are composed of neurons which are responsible for extracting patterns associated with the process or system being analyzed. These layers perform most of the internal processing from a network.
- **Output Layer** This layer is also composed of neurons, and thus is responsible for producing and presenting the final network outputs, which result from the processing performed by the neurons in the previous layers.

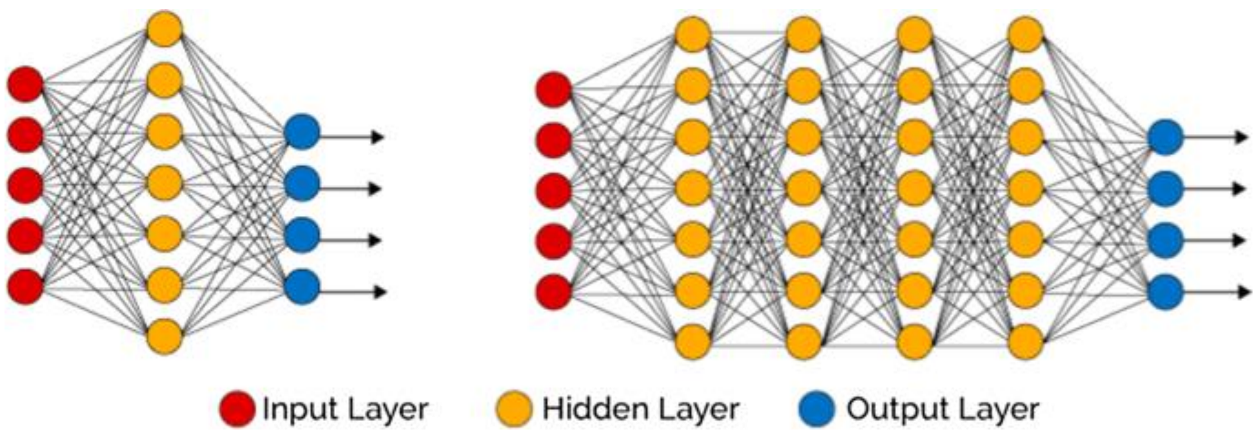


Figure 14 Shallow (simple) vs deep neural network

A shallow neural network has three layers of neurons that process inputs and generate outputs. A Deep Neural Network (DNN) has two or more "hidden layers" of neurons that process inputs. According to Goodfellow, Bengio, and Courville, and other experts, while shallow neural networks can tackle equally complex problems, deep learning networks are more accurate and improve in accuracy as more neuron layers are added [16].

The main architectures of artificial neural networks, considering the neuron disposition, as well as how they are interconnected and how its layers are composed, can be divided as follows:

♦ **Single-Layer Feedforward Architecture**

This artificial neural network has just one input layer and a single neural layer, which is also the output layer. The figure bellow illustrates a simple-layer feedforward network composed of n inputs and m outputs. The information always flows in a single direction (thus, unidirectional), which is from the input layer to the output layer. It is possible to see that in networks belonging to this architecture, the number of network outputs will always coincide with its amount of neurons. These networks are usually employed in pattern classification and linear filtering problems. Among the main network types belonging to feedforward architecture are the Perceptron and the ADALINE [16].

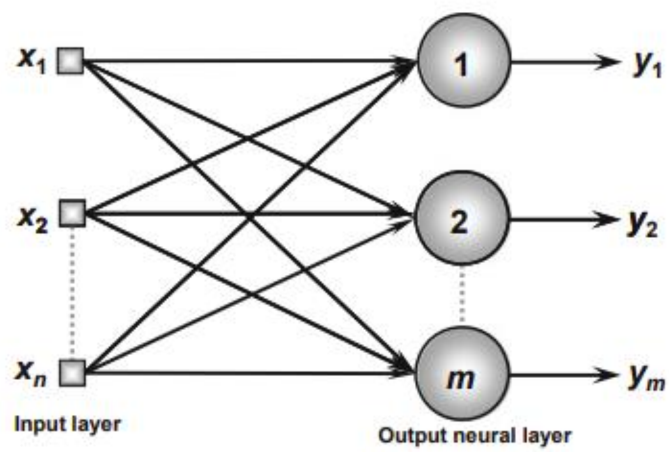


Figure 15 Example of a single-layer feedforward network

♦ **Multiple-Layer Feedforward Architectures**

Differently from networks belonging to the previous architecture, feedforward networks with multiple layers are composed of one or more hidden neural layers. They are employed in the solution of diverse problems, like those related to function approximation, pattern classification, system identification, process control, optimization, robotics, and so on. We have bellow a feedforward network with multiple layers composed of one input layer with n sample signals, two hidden neural layers consisting of n1 and n2 neurons respectively, and, finally, one output neural layer composed of m neurons representing the respective output values of the problem being analyzed.

Among the main networks using multiple-layer feedforward architectures are the Multilayer Perceptron (MLP) and the Radial Basis Function (RBF), and from figure bellow it is possible to understand that the amount of neurons composing the first hidden layer is usually different from the number of signals composing the input layer of the network. In fact, the number of hidden layers and their respective amount of neurons depend on the nature and complexity of the problem being mapped by the network, as well as the quantity and quality of the available data about the problem. Nonetheless, likewise for simple layer feedforward networks, the amount of output signals will always coincide with the number of neurons from that respective layer [16].

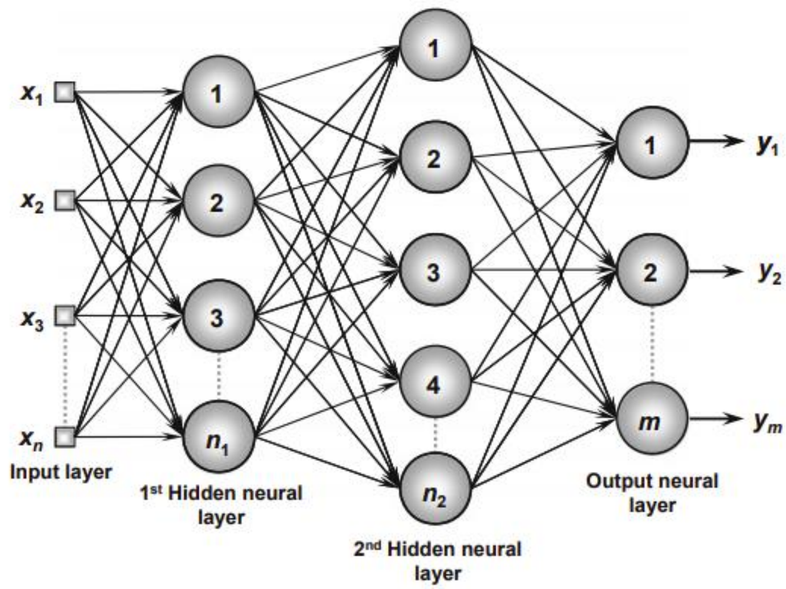


Figure 16 Example of a feedforward network with multiple layers

- **Recurrent or Feedback Architecture**

In these networks, the outputs of the neurons are used as feedback inputs for other neurons. The feedback feature qualifies these networks for dynamic information processing, meaning that they can be employed on time-variant systems, such as time series prediction, system identification and optimization, process control, and so forth. Among the main feedback networks are the Hopfield and the Perceptron with feedback between neurons from distinct layers, Figure 11illustrates an example of a Perceptron network with feedback, where one of its output signals is fed back to the middle layer. Thus, using the feedback process, the networks with this architecture produce current outputs also taking into consideration the previous output values [16].

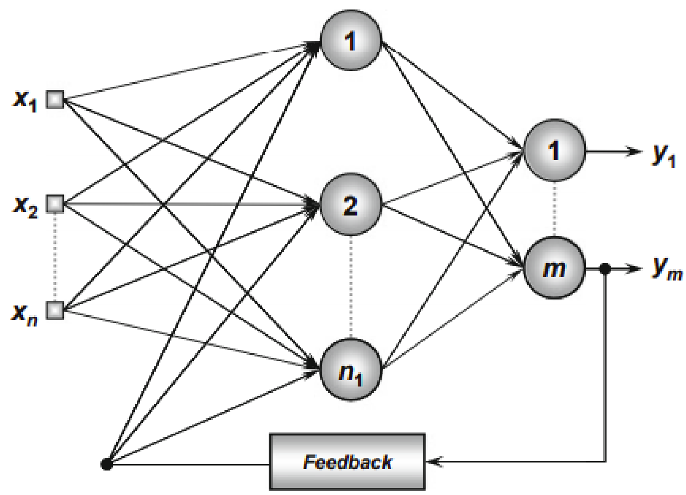


Figure 17 Example of a recurrent network

2.7 Activation functions

Activation functions are mathematical equations that determine the output of a neural network. The function is attached to each neuron in the network, and determines whether it should be activated ("fired") or not, based on whether each neuron's input is relevant for the model's prediction. Activation functions also help normalize the output of each neuron to a range between 1 and 0 or between -1 and 1.

2.7.1 Types of Activation Functions

♦ **Binary Step Function**

A binary step function is a threshold-based activation function. If the input value is above or below a certain threshold, the neuron is activated and sends exactly the same signal to the next layer. The problem with a step function is that it does not allow multi-value outputs—for example, it cannot support classifying the inputs into one of several categories [15].

♦ **Linear Activation Function**

A linear activation function takes the form: A = cx. It takes the inputs, multiplied by the weights for each neuron, and creates an output signal proportional to the input. In one sense, a linear function is better than a step function because it allows multiple outputs, not just yes and no. However, a linear activation function has two major problems:

- Not possible to use backpropagation (gradient descent) to train the model, the derivative of the function is a constant, and has no relation to the input, X. So it's not possible to go back and understand which weights in the input neurons can provide a better prediction.
- All layers of the neural network collapse into one with linear activation functions, no matter how many layers in the neural network, the last layer will be a linear function of the first layer (because a linear combination of linear functions is still a linear function). So a linear activation function turns the neural network into just one layer [15].

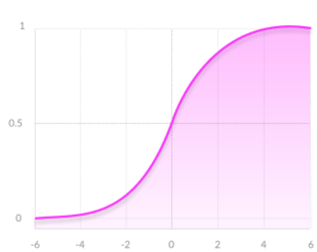
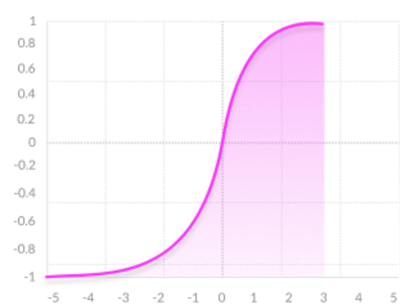
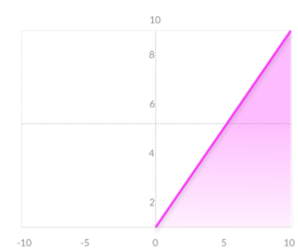
♦ **Non-Linear Activation Functions**

Modern neural network models use non-linear activation functions. They allow the model to create complex mappings between the network's inputs and outputs, which are essential for learning and modeling complex data, such as images, video, audio, and data sets which are non-linear or have high dimensionality. Almost any process imaginable can be represented as a functional computation in a neural network, provided that the activation function is non-linear. Non-linear functions address the problems of a linear activation function:

- They allow backpropagation because they have a derivative function which is related to the inputs.
- They allow "stacking" of multiple layers of neurons to create a deep neural network. Multiple hidden layers of neurons are needed to learn complex data sets with high levels of accuracy [15].

2.7.2 Common Nonlinear Activation Functions

♦ ***Curves*** [15]

		
Sigmoid / Logistic:	TanH / Hyperbolic Tangent:	ReLU (Rectified Linear Unit):
$f(x) = \frac{1}{(1+e^{-x})}$	$\tanh x = \frac{(e^{x}-e^{-x})}{(e^{x}+e^{-x})}$	f(x) = {0 for x<0, x for x≥0}

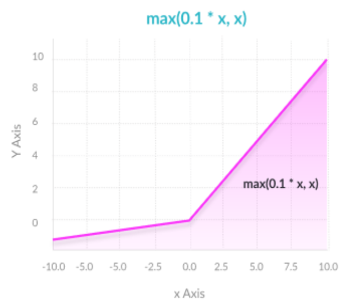
 Leaky ReLU	$f(x)=max(a∂,∂)$ Parametric ReLU	$\sigma(z)j=\frac{ezj}{\sum_{k=1ezk}^{k}}$ for $j=1....k.$ Softmax

Table 5 Nonlinear Activation Functions.

- ***Advantage and disadvantage*** [15]

Function	Advantages	Disadvantages
Sigmoid / Logistic	Smooth gradient, preventing "jumps" in output values. Output values bound between 0 and 1, normalizing the output of each neuron. Clear predictions—For X above 2 or below -2, tends to bring the Y value (the prediction) to the edge of the curve, very close to 1 or 0. This enables clear predictions.	Vanishing gradient—for very high or very low values of X, there is almost no change to the prediction, causing a vanishing gradient problem. This can result in the network refusing to learn further, or being too slow to reach an accurate prediction. Outputs not zero centered. Computationally expensive.
TanH / Hyperbolic Tangent	Zero centered: making it easier to model inputs that have strongly negative, neutral, and strongly positive values. Otherwise like the Sigmoid function.	Like the Sigmoid function.
ReLU (Rectified Linear Unit)	Computationally efficient: allows the network to converge very quickly. Non-linear: although it looks like a linear function, ReLU has a derivative function and allows for backpropagation.	The Dying ReLU problem: when inputs approach zero, or are negative, the gradient of the function becomes zero, the network cannot perform backpropagation and cannot learn.

Leaky ReLU	Prevents dying ReLU problem: this variation of ReLU has a small positive slope in the negative area, so it does enable backpropagation, even for negative input values. Otherwise like ReLU.	Results not consistent: leaky ReLU does not provide consistent predictions for negative input values.
Parametric ReLU	Allows the negative slope to be learned unlike leaky ReLU, this function provides the slope of the negative part of the function as an argument. It is, therefore, possible to perform backpropagation and learn the most appropriate value of α. Otherwise like ReLU.	May perform differently for different problems.
Softmax	Able to handle multiple classes only one class in other activation functions: normalizes the outputs for each class between 0 and 1, and divides by their sum, giving the probability of the input value being in a specific class. Useful for output neurons: typically Softmax is used only for the output layer, for neural networks that need to classify inputs into multiple categories.	

Table 6 Advantage and disadvantage of Nonlinear Activation Functions.

2.8 The Backward Propagation Algorithm

If we are planing to build a Neural Network, we will definitely need to understand how to train it. Backpropagation is a commonly used technique for training neural network, in this section we are going to understand what is forward pass, error function (loss function), backpropagation and gradient descent [17].

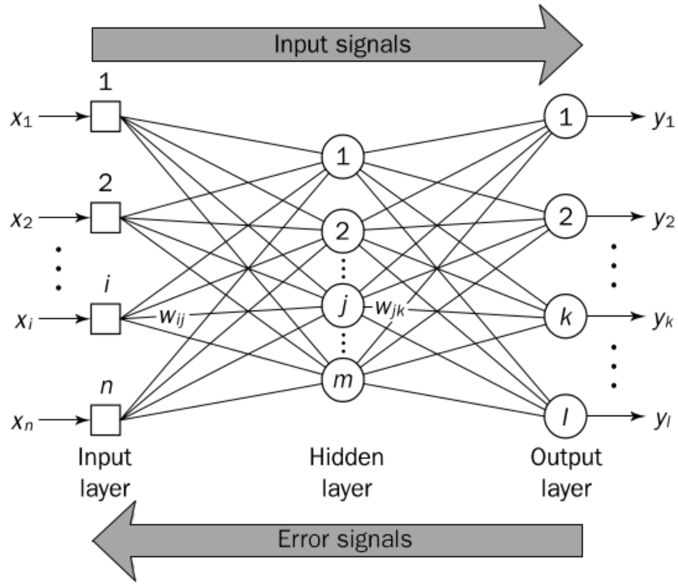


Figure 18 Neural Network with three layers

- **Overview**

We will build a neural network with three layers [17]:

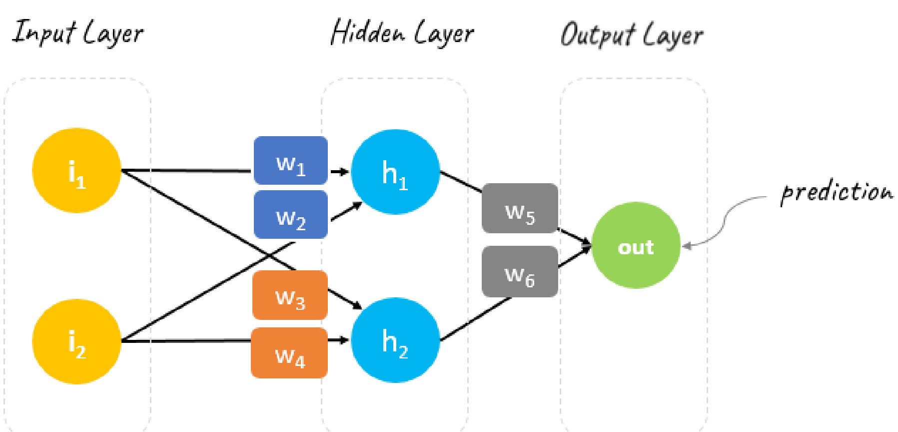


Figure 19 Neural network with three layers

Neural network training is about finding weights that minimize prediction error. We usually start our training with a set of randomly generated weights. Then, backpropagation is used to update the weights in an attempt to correctly map arbitrary inputs to outputs. Our initial weights will be as following: w1 = 0.11, w2 = 0.21, w3 = 0.12, w4 = 0.08, w5 = 0.14 and w6 = 0.15 [17].

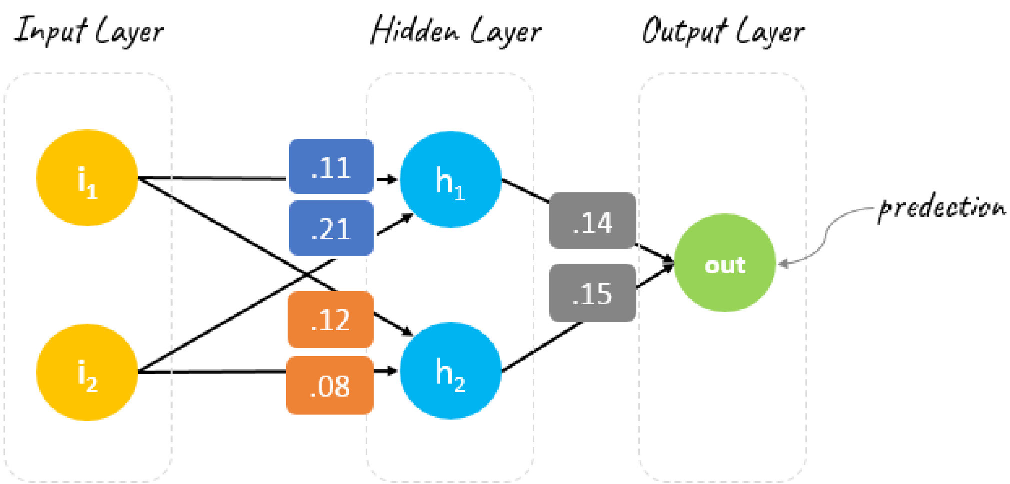


Figure 20 Initial weights of neural network with three layers

- **Dataset**

Our dataset has one sample with two inputs and one output [17].

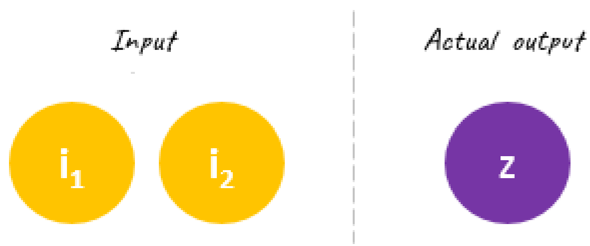


Figure 21 Dataset: Inputs and Outputs

Our single sample is as following inputs=[2, 3] and output=[1] [17].

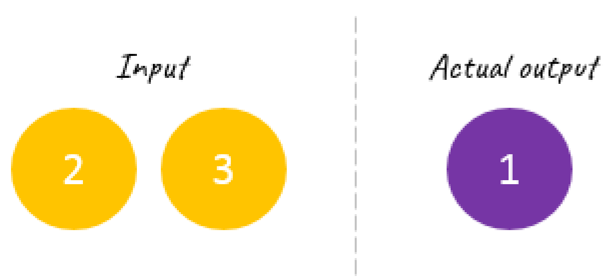
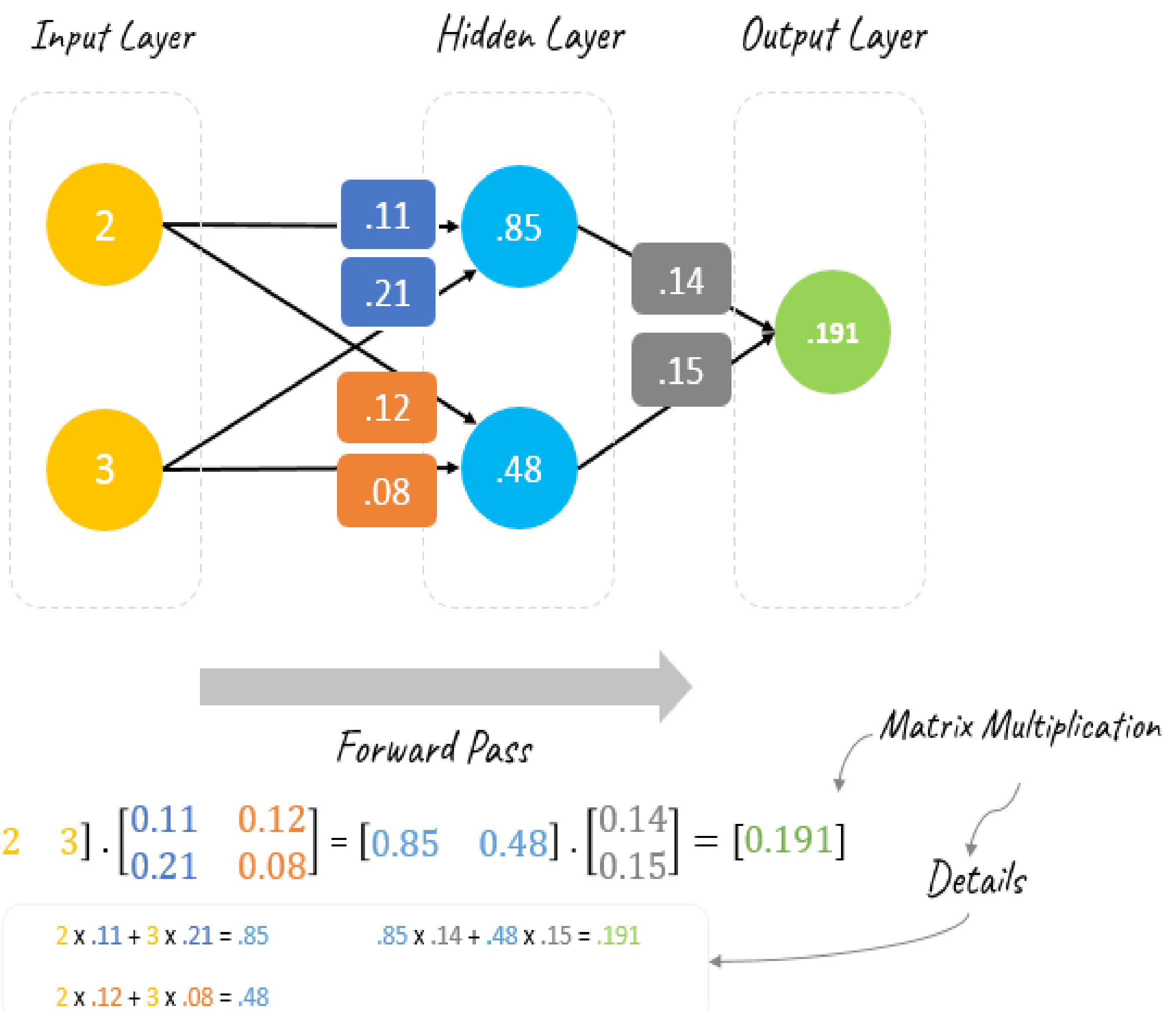


Figure 22 Initialization of the dataset

- **Forward Pass**

We will use given weights and inputs to predict the output. Inputs are multiplied by weights; the results are then passed forward to next layer [17].



$$[2 \quad 3] \cdot \begin{bmatrix} 0.11 & 0.12 \\ 0.21 & 0.08 \end{bmatrix} = [0.85 \quad 0.48] \cdot \begin{bmatrix} 0.14 \\ 0.15 \end{bmatrix} = [0.191]$$

Figure 23 Forward pass

- **Calculating Error**

Now, it's time to find out how our network performed by calculating the difference between the actual output and predicted one. It's clear that our network output, or prediction, is not even close to actual output. We can calculate the difference or the error as following [17].

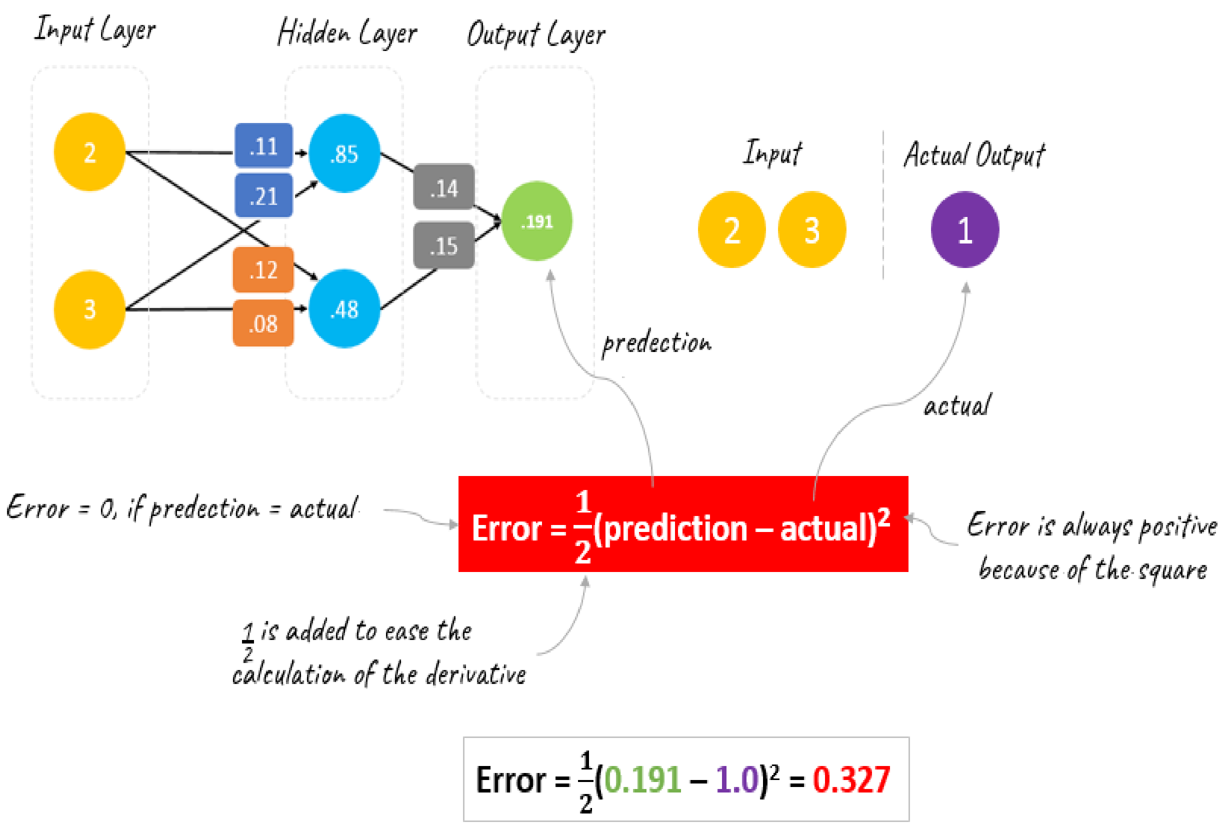


Figure 24 Error function (Loss function)

- **Reducing Error**

Our main goal of the training is to reduce the error or the difference between prediction and actual output. Since actual output is constant, "not changing", the only way to reduce the error is to change prediction value. The question now is, how to change prediction value? By decomposing prediction into its basic elements we can find that weights are the variable elements affecting prediction value. In other words, in order to change prediction value, we need to change weights values [17].

$$prediction = out$$

$$prediction = (h_1)\, w_5 + (h_2)\, w_6$$

$h_1 = i_1 w_1 + i_2 w_2$

$h_2 = i_1 w_3 + i_2 w_4$

$$prediction = (i_1\, w_1 + i_2\, w_2)\, w_5 + (i_1\, w_3 + i_2\, w_4)\, w_6$$

to change ***prediction*** value,
we need to change ***weights***

Figure 25 Reducing error

- **Backpropagation**

Backpropagation, short for "backward propagation of errors", is a mechanism used to update the weights using gradient descent. It calculates the gradient of the error function with respect to the neural network's weights. The calculation proceeds backwards through the network.

Gradient descent is an iterative optimization algorithm for finding the minimum of a function; in our case we want to minimize the error function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point [17].

Old weight

Derivative of Error with respect to weight

$$\underbrace{{}^{*}W_x}_{\text{New weight}} = W_x - \underbrace{a}_{\text{Learning rate}} \left(\frac{\partial Error}{\partial W_x}\right)$$

Figure 26 Backpropagation

For example, to update w6, we take the current w6 and subtract the partial derivative of error function with respect to w6. Optionally, we multiply the derivative of the error function by a selected number to make sure that the new updated weight is minimizing the error function; this number is called learning rate [17].

$$^{*}W_6 = W_6 - a\left(\frac{\partial Error}{\partial W_6}\right)$$

Figure 27 Updating weight 6 (w6)

The derivation of the error function is evaluated by applying the chain rule as following:

$$\frac{\partial Error}{\partial W_6} = \frac{\partial Error}{\partial prediction} * \frac{\partial prediction}{\partial W_6} \quad \text{chaine rule}$$

$$\text{Error} = \tfrac{1}{2}(\text{prediction} - \text{actual})^2$$

$$\text{prediction} = (i_1 w_1 + i_2 w_2)\, w_5 + (i_1 w_3 + i_2 w_4)\, w_6$$

$$\frac{\partial Error}{\partial W_6} = \frac{\frac{1}{2}(predictoin - actula)^2}{\partial prediciton} * \frac{\partial\,(i_1 w_1 + i_2 w_2)\, w_5 + (i_1 w_3 + i_2 w_4)\, w_6}{\partial W_6}$$

$$\frac{\partial Error}{\partial W_6} = 2 * \frac{1}{2}(\text{predictoin} - \text{actula}) \frac{\partial(predictoin - actula)}{\partial prediciton} * (i_1 w_3 + i_2 w_4) \quad h_2 = i_1 w_3 + i_2 w_4$$

$$\frac{\partial Error}{\partial W_6} = (\text{predictoin} - \text{actula}) * (h_2) \quad \Delta = \text{prediction} - \text{actual} \quad \text{delta}$$

$$\frac{\partial Error}{\partial W_6} = \Delta h_2$$

Figure 28 Updating weight 6 (w6)

So to update w6 we can apply the following formula [17]:

$$^{*}W_6 = W_6 - a\, \Delta h_2$$

Figure 29 formula of updating

Similarly, we can derive the update formula for w5 and any other weights existing between the output and the hidden layer [17].

$$^{*}W_5 = W_5 - a\, \Delta h_1$$

Figure 30 formula of updating

However, when moving backward to update w1, w2, w3 and w4 existing between input and hidden layer, the partial derivative for the error function with respect to w1, for example, will be as following [17].

$$\frac{\partial Error}{\partial W_1} = \frac{\partial Error}{\partial prediction} * \frac{\partial prediction}{\partial h_1} * \frac{\partial h_1}{\partial W_1}$$

chain rule

$$\text{Error} = \tfrac{1}{2}(\text{prediction} - \text{actual})^2$$

$$\text{prediction} = (h_1)\, w_5 + (h_2)\, w_6$$

$$h_1 = i_1 w_1 + i_2 w_2$$

$$\frac{\partial Error}{\partial W_1} = \frac{\partial \frac{1}{2}(predictoin - actula)^2}{\partial prediciton} * \frac{\partial\, (h_1)\, w_5 + (h_2)\, w_6}{\partial h_1} * \frac{\partial\, i_1 w_1 + i_2 w_2}{\partial w_1}$$

$$\frac{\partial Error}{\partial W_1} = 2 * \frac{1}{2}(\text{predictoin} - \text{actula}) \frac{\partial (predictoin - actula)}{\partial prediciton} * (w_5) * (i_1)$$

$$\frac{\partial Error}{\partial W_1} = (\text{predictoin} - \text{actula}) * (w_5 i_1)$$

$$\Delta = \text{prediction} - \text{actual}$$

delta

$$\frac{\partial Error}{\partial W_1} = \Delta\, w_5 i_1$$

Figure 31 Updating of weight 1

We can find the update formula for the remaining weights w2, w3 and w4 in the same way. In summary, the update formulas for all weights will be as following [17]:

Updated weight

$$\begin{aligned}
{}^*w_6 &= w_6 - a\,(h_2 \,.\, \Delta)\\
{}^*w_5 &= w_5 - a\,(h_1 \,.\, \Delta)\\
{}^*w_4 &= w_4 - a\,(i_2 \,.\, \Delta w_6)\\
{}^*w_3 &= w_3 - a\,(i_1 \,.\, \Delta w_6)\\
{}^*w_2 &= w_2 - a\,(i_2 \,.\, \Delta w_5)\\
{}^*w_1 &= w_1 - a\,(i_1 \,.\, \Delta w_5)
\end{aligned}$$

Figure 32 Updated weight

We can rewrite the update formulas in matrices as following [17]:

$$\begin{bmatrix} w_5 \\ w_6 \end{bmatrix} = \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} - a\,\Delta \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} - \begin{bmatrix} a h_1 \Delta \\ a h_2 \Delta \end{bmatrix}$$

$$\begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} = \begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} - a\,\Delta \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} . \begin{bmatrix} w_5 & w_6 \end{bmatrix} = \begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} - \begin{bmatrix} a\, i_1 \Delta w_5 & a\, i_1 \Delta w_6 \\ a\, i_2 \Delta w_5 & a\, i_2 \Delta w_6 \end{bmatrix}$$

Figure 33 Updated weights as matrix

- **Backward Pass**

Using derived formulas, we can find the new weights [17].

$\Delta = 0.191 - 1 = -0.809$ ← **Delta** = pradection - actual

$a = 0.05$ ← **Learning rate**, we smatly guess this number

$$\begin{bmatrix} w_5 \\ w_6 \end{bmatrix} = \begin{bmatrix} 0.14 \\ 0.15 \end{bmatrix} - 0.05(-0.809)\begin{bmatrix} 0.85 \\ 0.48 \end{bmatrix} = \begin{bmatrix} 0.14 \\ 0.15 \end{bmatrix} - \begin{bmatrix} -0.034 \\ -0.019 \end{bmatrix} = \begin{bmatrix} 0.17 \\ 0.17 \end{bmatrix}$$

$$\begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} = \begin{bmatrix} .11 & .12 \\ .21 & .08 \end{bmatrix} - 0.05(-0.809)\begin{bmatrix} 2 \\ 3 \end{bmatrix}.\begin{bmatrix} 0.14 & 0.15 \end{bmatrix} = \begin{bmatrix} .11 & .12 \\ .21 & .08 \end{bmatrix} - \begin{bmatrix} -0.011 & -0.012 \\ -0.017 & -0.018 \end{bmatrix} = \begin{bmatrix} .12 & .13 \\ .23 & .10 \end{bmatrix}$$

Figure 34 Backward pass

Now, using the new weights we will repeat the forward passed. We can notice that the prediction 0.26 is a little bit closer to actual output than the previously predicted one 0.191. We can repeat the same process of backward and forward pass until error is close or equal to zero [17].

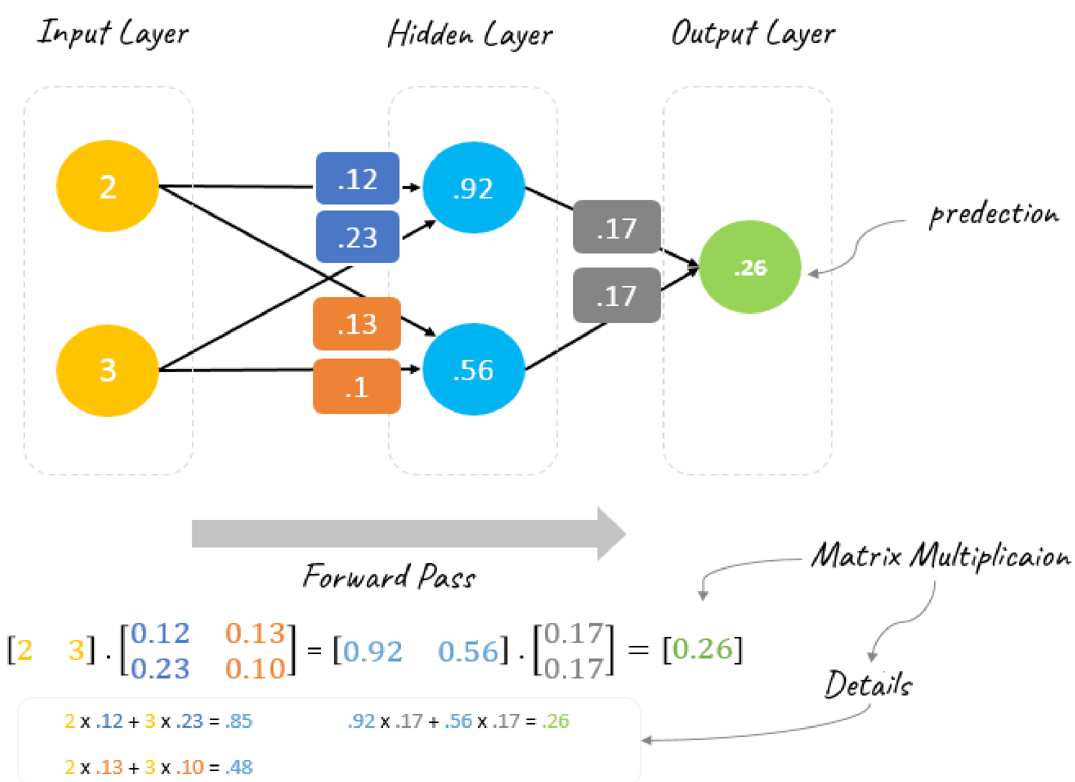


Figure 35 Forward pass

2.9 Types of ANNs

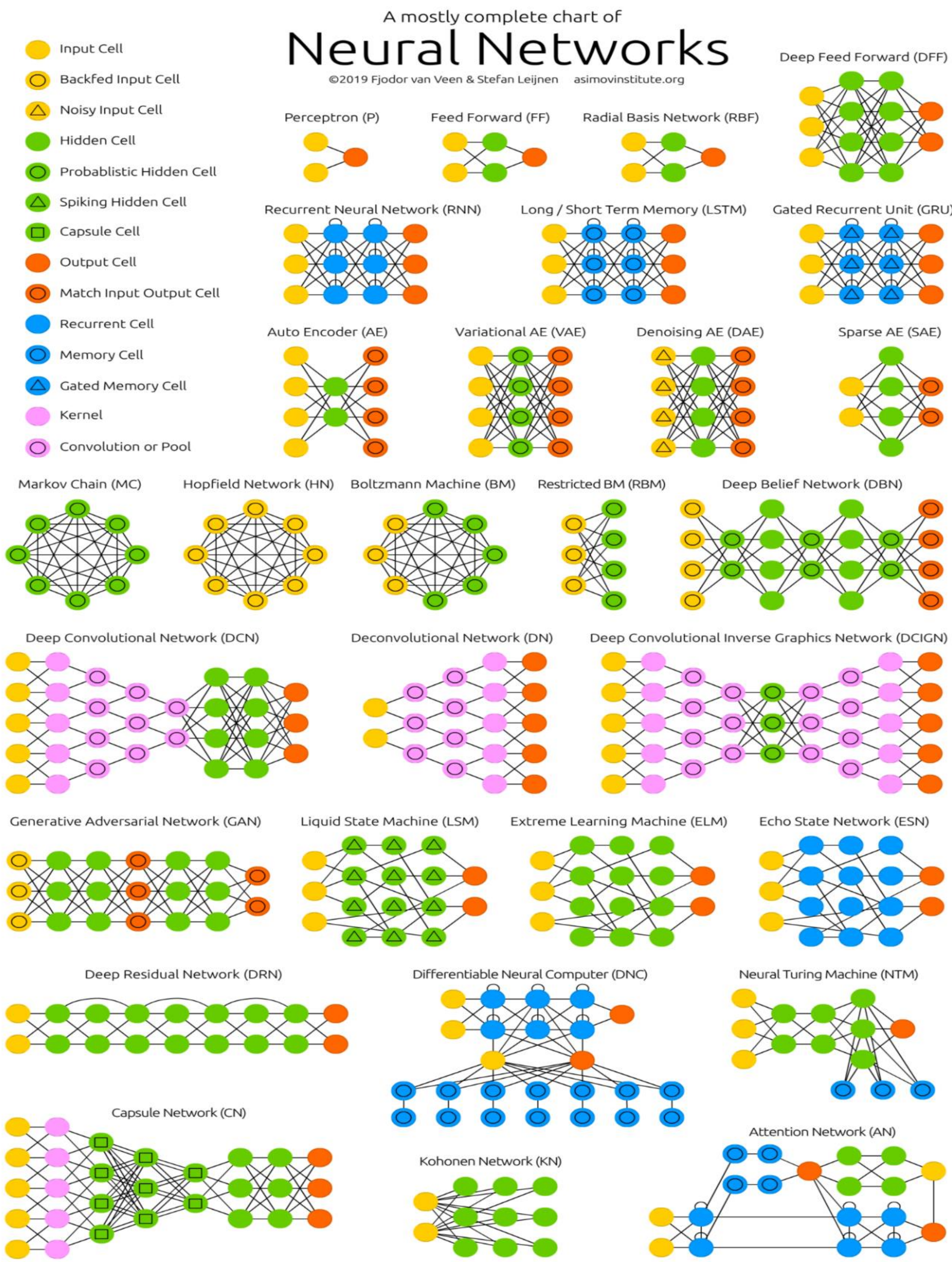


Figure 36 Types of ANN

As shown above in figure 36 there are different types of neural networks, in this section we will discuss the famous ones, namely [18]:

- **Feed-forward neural networks**

FNN is the purest form of ANN in which input and data travel in only one direction. Data flows in an only forward direction; that's why it is known as the Feed-forward Neural Network. The data passes through input nodes and exit from the output nodes. The nodes are not connected cyclically. It doesn't need to have a hidden layer. In FNN, there doesn't need to be multiple layers. It may have a single layer also. It has a front propagate wave that is achieved by using a classifying activation function. In FNN, the sum of the product's input and weight are calculated, and then it is fed to the output. Technologies such as face recognition and computer vision are used FNN [18].

- **Multilayer Perceptron**

A Multilayer Perceptron has three or more layer. The data that cannot be separated linearly is classified with the help of this network. This network is a fully connected network that means every single node is connected with all other nodes that are in the next layer. A Nonlinear Activation Function is used in Multilayer Perceptron. It's input and output layer nodes are connected as a directed graph. It is a deep learning method so that for training the network it uses backpropagation. It is extensively applied in speech recognition and machine translation technologies [18].

- **Convolutional Neural Network**

In image classification and image recognition, a CNN plays avital role, or we can say it is the main category for those. Face recognition, object detection etc., are some areas where CNN are widely used. It is similar to FNN, learn-able weights and biases are available in neurons. CNN takes an image as input that is classified and process under a certain category. As we know, the computer sees an image as pixels and depends on the resolution of the picture. Based on image resolution, it will see h * w * d, where h=height w=width and d=dimension. In CNN, each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, filters (Also known as kernels). And apply Soft-max function to classify an object with probabilistic values 0 and 1 [18].

- **Recurrent Neural Network**

back to the input. It will help to predict the outcome of the layer. In Recurrent Neural Network, the first layer is formed in the same way as FNN's layer, and in the subsequent layer, the recurrent neural network process begins. Both inputs and outputs are independent of each other, but in some cases, it required to predict the next word of the sentence. Then it will depend on the previous word of the sentence. RNN is famous for its primary and most important feature, i.e., Hidden State. Hidden State remembers the information about a sequence. RNN has a memory to store the result after calculation. RNN uses the same parameters on each input to perform the same task on all the hidden layers or data to produce the output. Unlike other neural networks, RNN parameter complexity is less [18].

2.10 Potential Application Areas

Artificial neural networks can be employed in several problems related to engineering and sciences. The potential application areas can be divided as follows [16]:

- **Universal curve fitting (function approximation)**

The goal is to map the functional relationship between variables (usually real numbers) of a particular system from a known set of meaningful values. These applications are as diverse as possible, and often involve mapping processes that are difficult to model using traditional methods [16].

- **Process control**

This application category consists of identifying control actions capable of meeting quality, efficiency, and security requirements. Among the multiple available applications, neural controllers are of particular interest to robotics, airplanes, elevators, appliances, satellites, and so on [16].

- **Pattern recognition/classification**

The purpose is to associate a given input pattern (sample) to one of the previously defined classes, as in the case of image, speech and writing recognition. In this case, the problem being addressed has a discrete and known set of possible desired outputs [16].

♦ **Data clustering**

On this circumstance, the goal is to detect and identify similarities and particularities of the several input patterns to allow their grouping (clustering). Some examples, to cite a few, are applications involving automatic class identification and data mining [16].

♦ **Prediction system**

The purpose of this system category is to estimate future values of a particular process, taking into account several previous samples observed in its domain. Among the known applications, it is possible to find systems for time series prediction, stock market projection, weather forecast, and so on [16].

♦ **System optimization**

The goal is to minimize or maximize a cost function (objective) obeying eventual constraints to correctly map a problem. Among the optimization tasks which can benefit from artificial neural networks, the most important includes constrained optimization problems, dynamic programming, and combinational optimization [16].

♦ **Associative memory**

The objective is to recover a correct pattern even when its inner elements are uncertain or inaccurate. Some examples include image processing, signal transmission, written character identification, and so forth [16].

2.11 Advantages and disadvantages of ANN

Advantages [19]**:**

• A neural network can perform tasks in which a linear program cannot perform.

• When an element of the neural network fails, it can continue without any problem by their parallel nature.

• A neural network does not need to be reprogrammed as it learns itself.

• It can be implemented in an easy way without any problem.

• As adaptive, intelligent systems, neural networks are robust and excel at solving complex problems. Neural networks are efficient in their programming and the scientists agree that the advantages of using ANNs outweigh the risks.

• It can be implemented in any application.

Disadvantages [19]**:**

• The neural network requires training to operate.

• Requires high processing time for large neural networks.

• Requires large amount of data for a good performance.

3. Conclusion

The computing world has a lot to gain from neural networks. Their ability to learn by examples make them very flexible and powerful. Furthermore, there is no need to devise an algorithm in order to perform a specific task, they are also very well suited for real time systems because of their fast response and computational times which are due to their parallel architecture. In this chapter we discussed the background of ANN (ML and DL), the BNN and the difference between it and ANN, types of ANN and the general concepts of ANN.

Chapter – 3 – A hybrid recommender system using artificial neural networks

1. Introduction

Recommendation systems algorithms that use rating information are hugely popular but in this era of Web 2.0 where user reviews are available for most of the products/services, this information can be mined efficiently and used in combination with rating data to deliver high quality recommendations. Reviews are typically written in text form describing their assessment of the product or experience of the service provided. However, rating is a homogeneous value and does not capture the sentiment and/or context behind users' experience in the way that a review would. As far as we know, there is very little research done that uses the combination of reviews and ratings in conjunction with user and item metadata to develop a supervised learning rating prediction model. In general, an efficient recommender system should be able to model and capture the complex, nonlinear relationships between users and items. In this chapter, we develop a new hybrid RS technique that builds on the capabilities provided by traditional approaches like collaborative filtering and content-based filtering by making use of user reviews to train and build an ANN to predict "enhanced rating (EnhR)", which is a rating variable derived from the user's reputation, review helpfulness, and item rating and evaluate the improvement in recommendation predictions. Further, we demonstrate how well the proposed technique works by performing computational experiments using the Yelp Academic Dataset.

2 Artificial neural network for Hybrid Recommendation System

Review websites, such as TripAdvisor and Yelp, allow users to post online reviews for various businesses, products and services, and have been recently shown to have a significant influence on consumer shopping behavior. An online review typically consists of text, and a star rating out of 5. The problem of predicting a user's star rating for a product, given the user's review for that product, is called Review Rating Prediction and has lately become a popular, albeit hard, problem in machine learning. For our research, we consider the problem of predicting the rating as a multi-class classification problem where each rating is treated as class and we build a different prediction model which is an Artificial Neural Network for a hybrid recommendation system.

As far as we know the term "hybrid recommender system or recommendation system" is used to describe any recommender system that combines multiple recommendation techniques together to produce its output, and in this thesis we combined features from content based recommendation system (user and business) and collaborative recommendation system (review and votes) and fed it into our model, this method of combining features from different recommendation sources together and give it to a single recommendation algorithm known as feature combination [20].

Therefor for the purpose of the recommendation system, we estimate the probabilities of a customer rating a business into one of the discrete classes in $c \in [1, \ldots, 5]$, and we use an ANN with 5 neurons in the output layer such that each output neuron corresponds to a rating and we evaluated the predicted rating using Confusion matrix on the test dataset or a validation dataset.

2.1 Artificial neural network - supervised learning Model

ANN were originally motivated by the goal of having machines that could mimic the brain, and they are an excellent nonlinear modeling tools for approximating any non-linear relationships and finding patterns within the data [21]. Therefore, a model built using an ANN is well positioned to learn the complex relationships between users and items, as well as predict better recommendations. Deep learning is a new and evolving field in machine learning where powerful models are developed by utilizing a deep structured neural network and we consider our model a Deep learning algorithm since we use more than a single hidden layer in our Feedforward Neural Network model as we will explore in the section below.

2.2 A feedforward neural network

Feedforward neural network is a nonlinear function of its inputs, in which information flows from input nodes to output nodes. Each neuron in the network is a nonlinear combination of inputs x_i weighted by the parameters w_i.

The yelp dataset was trained using network topology that has 3 hidden layers with [32,32,32] hidden nodes cause we observed when the number of nodes increased, the model performance and accuracy of rating predictions improved too but we did not observe a significant improvement in overall performance or validation loss between a model with 12 and 192 hidden nodes, and the increasing of the number of hidden nodes raised the model training time . In addition, all networks are fully connected and contain a bias node in the hidden layers, and each network was initialized with a random set of weights following a uniform probability distribution. For the purpose of our research, an activation function with nonlinear properties is important because of its ability to discriminate relationships in the feature space and strongly influences the complexity and performance of our mode, we use softmax activation function for the output layer with the constraint that there are 5 nodes y in the output layer, one for each class of rating as shown in figure 37, where p = 13 and x represents the input, and we performed experiments with our model using different activation functions, such as tanh, sigmoid, relu for hidden layers [21] .

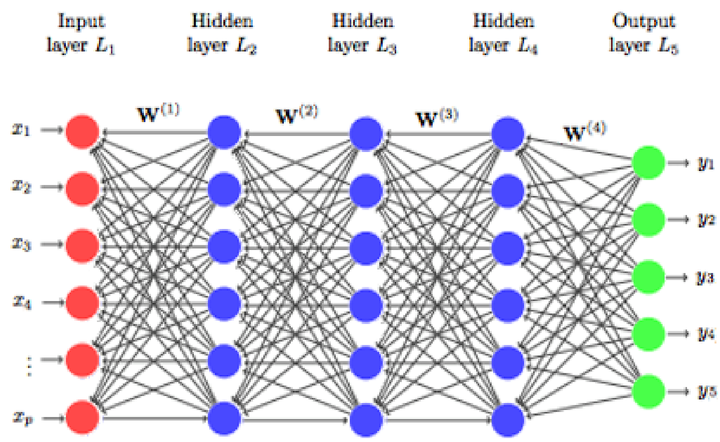


Figure 37 ANN Architecture

4. Experiments

2.3 Data Model

We demonstrate the efficiency of our approach using Yelp Dataset [22]. This data source contains rich set of relational data objects for business, users, reviews, tips and checkins. For our research, business, users and reviews data objects are used, and since the dataset has a total of 783 categories such as Appliances & Repair, Car Rental, Doctors, Restaurants, etc. we choose the category Restaurants.

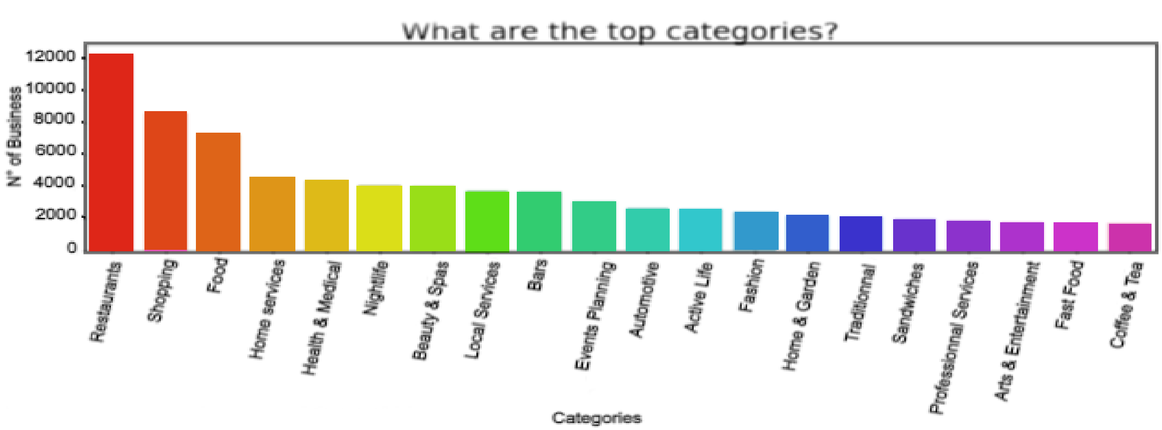


Figure 38 Top categories in Yelp Dataset

When using data, most people agree that your insights and analysis are only as good as the data you are using. Essentially, garbage data in is garbage analysis out. Data cleaning, also referred to as data cleansing and data scrubbing, is one of the most important steps for your organization if you want to create a culture around quality data decision-making. And since proper data cleaning can make or break your project and the professional data scientists usually spend a very large portion of their time on this step. We believe that is an important section this is why we talked about how we explored the yelp dataset to achieve our dataset (the final dataset) which is going to be feed into our model.

First, we start by handling missing data since we cannot simply ignore missing values in our dataset by filling the original missing value with 0 just to meet the technical requirement, then we filter out restaurants that are closed. Note that, businesses, users, reviews are relational objects that are stored in separate files and to enable model training, these objects are merged to form a flat structure such that each row represents an observation. This is achieved by performing an "inner-join" between these datasets; which is an operation between two database tables such that all rows that satisfy the join condition are returned. Open restaurants are filtered from businesses and joined against reviews using the attribute "business_id". This intermediate dataset (that is the result of joining businesses and reviews), is further joined against users using the attribute "user_id" to produce the final dataset used for model training.

Finally, the dataset used for model training has the following attributes: business average rating, total reviews for business, total fans for a user, total friends for a user, days user has been yelping since, total compliments received by the user, total votes received by users' reviews, total reviews written by the user, age of the review, total cool, funny, useful votes received by a review and rating associated with the review. The relational model of these objects for attributes used to build our model is shown in Figure 39.

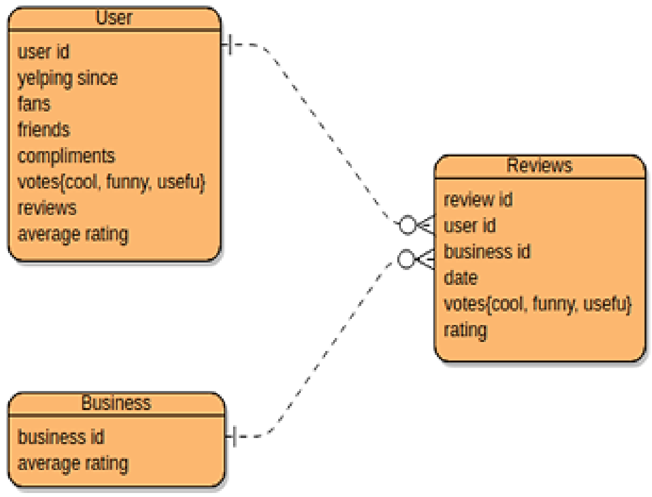


Figure 39 Yelp dataset entity-relationship diagram

The final dataset is split into train/test dataset such that 70% are used to create the training dataset and remaining 30% is used to generate the test dataset. Descriptive statistics for the features used to train the ANN model are shown in Table 7:

Dataset	Feature	Mean	StdDev	Median	Min	Max	Skew
Business	Avg. rating	3,45	0.83	3.5	1	5	-0.53
	Total review	92,75	229.09	28	3	10129	11.64
User	Total fans	12.12	64.25	1	0	11568	18.72
	Total friends	122.91	373.92	20	0	14994	9.59
	Yelping since	2717.61	1016.42	2706	382	5922	0.07
	Total compliments	169.88	1665.63	2	0	263889	29.46
	Total votes	900.12	6996.75	33	0	554350	26.12
	Total reviews	121.03	351.31	28	0	14455	13.94
	Avg. rating	3,75	0.77	3.84	1	5	-1.06
Review	Days since review	1637.59	936.66	1440	382	5922	0.92
	Total cool votes	56.24	2.49	0	0	321	21.81
	Total funny votes	41.49	1.88	0	0	786	37.24
	Total useful votes	1.04	3.04	0	0	758	19.62
	Rating	3,76	1.39	4	1	5	-0.85

Table 7 Feature Descriptive statistics

Since Outliers are data objects that have some characteristics that are different from most of the other data objects as shown in figure 40 below , and they will have an influence on the accuracy of a supervised learning model [23].

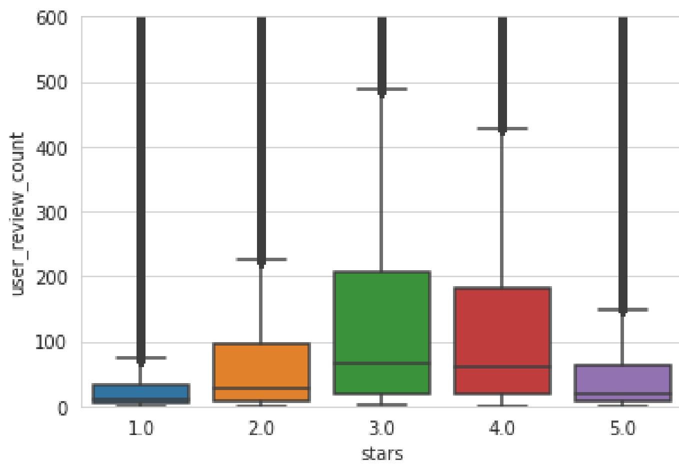


Figure 40 Outliers in the number of reviews written per star rating

We minimize the influence of these outliers on the model estimates, by transforming the data using winsorizing, that is, extreme values are clipped to a threshold percentile of 95.

As far as we know in an ANN model, if the range of a feature is too high, any change to the weights will have a relative influence on the features with larger magnitudes. As shown in figure 4 above, several features have a high range of values and will lead to inefficient training without transforming the data. There is no standard approach for scaling features, but we adopt min-max normalization technique because it linearly transforms the data and preserves the relationship from original data. This is a computationally simple technique that can fit data within a uniform range. To improve model performance and convergence rate, we use the formula below to normalize the data such that the feature vector is transformed to have values in the range [0, 1].

$$\boldsymbol{x} = \frac{(\boldsymbol{x} - \mathbf{min}(\boldsymbol{x}))}{(\mathbf{max}(\boldsymbol{x}) - \mathbf{min}(\boldsymbol{x}))}$$

2.4 Implementation

We developed our hybrid ANN model using Keras [24] for the API (Application Programming Interface) layer and TensorFlow as the back-end computation engine. We ran our experiments on Windows 10 system with 8 GB RAM and 4 CPUs.

2.4.1 Language and libraries

2.4.1.1 R Language

R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing [25]. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, data mining surveys, and studies of scholarly literature databases show substantial increases in popularity; as of September 2020, R ranks 9th in the TIOBE index, a measure of popularity of programming languages [26].

2.4.1.2 Python Language

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. The best options for utilizing Python are web development, simple scripting and data analysis [27] .

Figure 41 Python logo.

2.4.1.3 Keras

Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license. Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles:

Modularity: A model can be understood as a sequence or a graph alone. All the concerns of a deep learning model are discrete components that can be combined in arbitrary ways.

Minimalism: The library provides just enough to achieve an outcome, no frills and maximizing readability.

Extensibility: New components are intentionally easy to add and use within the framework, intended for researchers to trial and explore new ideas.

Python: No separate model files with custom file formats. Everything is native Python [28].

Figure 42 Keras deep learning library logo

2.4.1.4 TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state of-the-art in ML and developers easily build and deploy ML-powered applications. TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization to conduct machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well. TensorFlow provides stable Python and C++ APIs, as well as no guaranteed backward compatible API for other languages [29].

Figure 43 TensorFlow deep learning library logo

2.4.1.5 Scikit-learn

Scikit-learn is open source python library built on NumPy, SciPy, and matplotlib, it has simple and efficient tools for predictive data analysis and is accessible to everybody, and reusable in various contexts which used in many fields such as classification, regression and clustering, etc [30].

Figure 44 Scikit-learn logo

2.5 Running Environment

2.5.1 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. For our program we use Anaconda Individual which is a free Edition, easy-to-install package manager, environment manager, and Python distribution with a collection of 1,500+ open source packages with free community support [31].

Figure 45 Anaconda logo

2.5.2 Spyder

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package [32].

Figure 46 Spyder logo

2.6 Results and Discussion

Results from the initial set of experiments are presented in Table 8 sorted by decreasing validation loss. All experiments were run using the stochastic gradient descent optimization algorithm by varying the parameters such as the activation functions: rectified linear unit (relu), tanh, sigmoid for the hidden layers with Learning Rate (which is size of the step towards minimum values along the gradient and determines how fast or slow the model iterates towards the optimal weights) of 0.001 and varying the dropout probabilities of 0.1, 0.2 and 0.3 but this parameter did not have a significant impact on the model performance.

Architecture	Dropout	Validation loss (min)	Epoch	Train loss (at epoch)
3 hidden layers (32-32-32) with tanh activation function.	0.1	0.5488	184	0.9918
	0.2	0.4519	192	0.7593
	0.3	0.3786	198	0.5140
3 hidden layers (32-32-32) with relu activation function.	0.1	0.1951	179	0.3971
	0.2	0.1810	191	0.3561
	0.3	0.1627	199	0.2979
3 hidden layers (32-32-32) with sigmoid activation function.	0.1	1.4513	15	1.8317
	0.2	1.4364	19	1.6872
	0.3	1.4279	25	1.5399

Table 8 Model performance comparison using tanh, relu and sigmoid activation function.

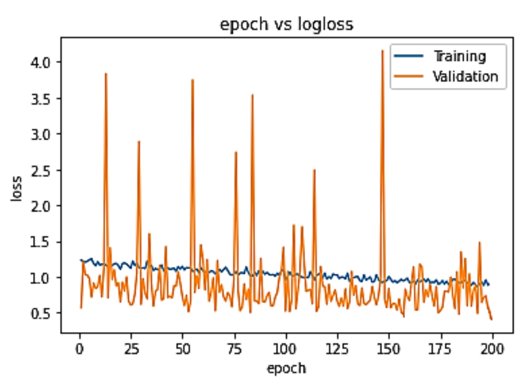


Figure 47 Comparison of model performance between training and validation datasets with learning rate of 0.1 using relu activation function.

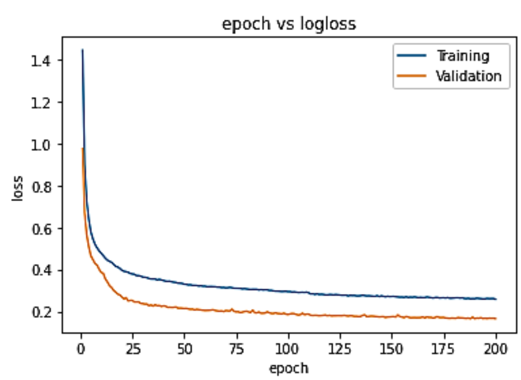


Figure 48 Comparison of model performance between training and validation datasets with learning rate of 0.001 using relu activation function.

T-Model32- L0.001-D0.3 which is a model with activation function relu, 32 hidden nodes with a learning rate 0.001 and dropout probability of 0.3 at each hidden layer, has the best performance with lowest validation loss of 0.1627 achieved at epoch 199/200 and the training error at this epoch is 0.2979. We also observed that validation loss continues to get better even at 200 epochs as shown by the very high epoch at which lowest validation loss is observed. The model's performance is evaluated against test dataset of restaurants from the Yelp dataset. For this test data, model results in an accuracy of 91.22%. Accuracy of a classification model is the percentage of correct predictions in the total population, where it is often presented as a percentage:

classification accuracy = (correct predictions / total predictions) * 100

and misclassification rate or error rate:

error rate = (1 - (correct predictions / total predictions)) * 100

The main problem with classification accuracy, it can hide the detail you need to diagnose the performance of your model. But thankfully we can tease apart this detail by using a confusion matrix. So the confusion matrix is a summary of prediction results on a classification problem, where the number of correct and incorrect predictions are summarized with count values and broken down by each class. It shows the ways in which your classification model is confused when it makes predictions and gives you insight not only into the errors being made by your classifier but more importantly the types of errors that are being made.

Our confusion matrix for the results from test dataset is shown in the table 9 below, it is observed that the error rate is primarily contributed by misclassifying 8,78% of the ratings as 2 when the actual rating was 1. Note that, for developing a RS, ratings 1, 2, 3 belong to the class of not recommending the business and misclassification of ratings within this group can be safely ignored. We infer the reason for incorrect classification of the rating 1 that is the ratings in the dataset are skewed with over 68.99% of the overall ratings classified as 4, 5 and only 10% of restaurants receiving a rating of 1 ; this makes it really difficult for the model to predict the rating 1 compared to other ratings.

Restaurant	Predicted rating				
Actual rating	1	2	3	4	5
1	15 456	111 233			
2		109 539			
3			156 588		
4				463 092	
5					410 982

Table 9 Confusion matrix of test dataset for restaurants

Finally, we can say that predicting ratings as a multi-class classification problem instead of a binary yes/no recommendation has its advantages. It allows the engine to perform deeper analysis on the predictions to build more efficient recommendation systems. For instance, an engine can rank the recommendations such that businesses with prediction rating of 5 are presented to the user ahead of the businesses with prediction rating of 4 which would not be possible if the model only presented yes or no prediction. It is also worth noting that, 12.36% of the predictions belong to the rating 3, which can also be treated as neutral instead of a negative. In general, predicting ratings provides flexibility for the engine to interpret and generate a recommendation customized to the line of business.

5. Conclusion

In this chapter, we proposed a novel approach for predicting the ratings using an artificial neural network framework, where both collaborative and content-based features are used to train the model. This methodology brings together content (user and business), collaborative (review and votes) and metadata associated with ratings under the framework of a unified supervised learning. We performed a number of experiments using the Yelp academic dataset by varying the model hyper-parameters, such as the number of hidden layers, number of nodes, number of epochs, learning rate, and dropout probability. We optimized these parameters by validating the model performance against test and validation datasets such that the misclassification error rate was reduced to less than 9%.

General Conclusion

1 Summary

First, we presented an overview about recommendation system generally, its definition, history, and techniques. Then, we based on the hybrid approach which is taken between content-based filtering and collaborative filtering to implement the system and we mention the different hybridization methods, as we discussed the challenges and the issues that this approach overcomes to improve the performance of the system.

In Chapter 2, we discussed the background of Artificial Neural Network (ANN): Machine Learning (ML) and Deep Learning (DL), types of ANN and the general concepts of ANN.

Finally, to answer our thesis question, which was:

Can we develop an ANN model for a Hybrid recommendation system to predict the rating?

We proposed a novel approach for predicting the ratings using an artificial neural network framework, where both collaborative and content-based features are used to train the model. This methodology brings together content (user and business), collaborative (review and votes) and metadata associated with ratings under the framework of a unified supervised learning model to produces better prediction results. We showed that because of the effective learning capabilities of an ANN framework, from the collaborative and content-based features we can significantly improve the overall effectiveness of the recommendation system. We performed several experiments using the Yelp academic dataset by varying the model hyper-parameters, such as the activation function in the hidden layers, learning rate, and dropout probability. We optimized these parameters by validating the model performance against test and validation datasets such that the misclassification error rate was reduced to less than 9% and none of the predictions crossed boundaries between the binary classification category of recommending (ratings 4 or 5) and not recommending (ratings 1 or 2 or 3) a business to the user.

2 Limitations

Our thesis didn't contain Baseline model where we were supposed to use collaborative filtering model and compare the accuracy of classification and misclassification rate with our model. Also, review texts which can also play an important role in understanding users' interests and can be more useful compared to star rating. By applying Natural Language Processing methods, we can identify latent factors within the review text to further enhance the quality of recommendations.

3 Future work

In future work, several avenues of this research still need to be explored, such as:

- Geospatial data of the businesses can play an important role in the development of location aware recommendation systems. It adds another dimension to the problem, and it can be extremely useful in developing a user-specific, real-time recommendation system by leveraging geospatial data from a users' GPS enabled device.

- During the last few years, there has been significant technological developments in the domain of distributed storage and computation using big data analytic frameworks like Hadoop, MapReduce and Spark. Most of the recommender systems available today are designed to run on a single server; it is an interesting idea to analyze and compare the cost, performance and efficiency of a recommendation system while utilizing a distributed processing framework.

- Therefore, it will be interesting to analyze the performance of an ANN for Hybrid recommendation system using other publicly available datasets such as Amazon reviews, TripAdvisor and Epinions databases.

Bibliography

[1] J. Cates, «Recommender Systems 101,» 6 Aug 2019 . [En ligne]. Available: https://medium.com/@topspinj/recommender-systems-101-bcbdfbe1e6e7.

[2] T. Daigle, «Introduction to Recommender Systems,» 31 Jul 2017. [En ligne]. Available: https://dzone.com/articles/introduction-to-recommender-systems.

[3] L. Xiang, «Recommender system introduction,» 15. avr 2012. [En ligne]. Available: https://fr.slideshare.net/xlvector/recommender-system-introduction-12551956.

[4] N. R. Pinnapareddy, «Deep Learning based Recommendation Systems,» 2018.

[5] L. S. Jaimeel M. Shah1, «A Hybrid Based Recommendation System based on Clustering and,» *Binary Journal of Data Mining & Networking*, 2015.

[6] M. A. A. A. R. ,. K. Hael Al-bashiri, «An improved memory-based collaborative filtering method based on the TOPSIS technique,» *PLoS One*, 2018 Oct 4.

[7] R. S. S. R. a. D. G. Babaljeet Kaur, «Recommender System: Towards Classification of HumanIntentions in E-Shopping Using Machine Learning,» *Journal ofComputational and Theoretical Nanoscience*, 30 August 2019.

[8] J. L. Tulasi K.Paradarami.Nathaniel D.Bastian, «A hybrid recommender system using artificial neural,» 2017.

[9] R. Burke, Hybrid Recommender Systems:, Department of Information Systems and Decision Sciences , November 2002.

[10] «Edureka,» [En ligne]. Available: “https://www.edureka.co/blog/what-is-machine-learning/”..

[11] «MathWorks,» [En ligne]. Available: https://www.mathworks.com/discovery/machine-learning.html.

[12] «Javatpoint,» [En ligne]. Available: "https://www.javatpoint.com/what-is-machine-learning-in-tensorflow"..

[13] M. Negnevitsky, Artificial Intelligence A Guide to Intelligent Systems, Second Edition published 2005 éd.

[14] «tutorialspoint,» [En ligne]. Available: https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_basic_concepts.htm.

[15] «missinglink.ai,» [En ligne]. Available: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/.

[16] D. H. S. R. A. F. L. H. B. L. ,. S. F. d. R. A. Ivan Nunes da Silva, Artificial Neural Networks A Practical Course. Springer International Publishing Switzerland., 2017.

[17] «HMKCODE,» [En ligne]. Available: https://hmkcode.com/ai/backpropagation-step-by-step/.

[18] «Artificial Neural Network in TensorFlow - Javatpoint,» [En ligne]. Available: https://www.javatpoint.com/artificial-neural-network-in-tensorflow#:~:text=Feedforward%20Neural%20Network%20(Artificial%20Neuron,exit%20from%20the%20output%20nodes.

[19] «ElProCus – Electronic Projects for Engineering Students,» [En ligne]. Available: https://www.elprocus.com/artificial-neural-networks-ann-and-their-types/.

[20] R. Burke, Hybrid Web Recommender Systems, School of Computer Science, Telecommunications and Information Systems DePaul University, 243 S. Wabash Ave. Chicago, Illinois, USA, 2005.

[21] T. K. Paradarami, «A hybrid recommender system using artificial neural networks,» 2017.

[22] Yelp, «Yelp Dataset,» 2020. [En ligne]. Available: https://www.yelp.com/dataset.

[23] P. J. &. L. A. M. Rousseeuw, Robust regression and outlier detection, New York, NY, USA: John Wiley & Sons, Inc, 1987.

[24] «GitHub - keras-team/keras: Deep Learning for humans,» 2015. [En ligne]. Available: https://github.com/keras-team/keras.

[25] K. Hornik, «The Comprehensive R Archive Network. 2.1 What is R?,» 1998-2020. [En ligne]. Available: https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f.

[26] TIOBE, «TIOBE Index - The Software Quality Company,» 9 September 2020. [En ligne]. Available: https://www.tiobe.com/tiobe-index/.

[27] A. Malakooth, «What can you do with Python? Medium,» [En ligne]. Available: https://medium.com/@mmafrar/what-can-you-dowith-python-8d901376db3e.

[28] J. Brownlee, «Introduction to Python Deep Learning with Keras,» 10 May 2016. [En ligne]. Available: https://machinelearningmastery.com/introduction-python-deep-learning-library-keras/ .

[29] Tensorflow, «Tensorflow,» [En ligne]. Available: https://www.tensorflow.org/.

[30] «scikit-learn,» [En ligne]. Available: https://scikit-learn.org/stable/index.html.

[31] Anaconda, «Anaconda Documentation,» 2020. [En ligne]. Available: https://docs.anaconda.com/.

[32] S. W. Contributors, «Home — Spyder IDE,» 2020. [En ligne]. Available: https://www.spyder-ide.org/.

[33] C.-T. S. a. E. M. J. -S. R. Jang, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Englewood Cliffs, NJ: Prentice-Hall, 1997.