REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : IA 3/M2/2019

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence Artificielle**

A Deep Learning Approach for the analysis of feelings on social networks

---

Par :

KARABAGHLI MOUATEZ BELLAH

Soutenu le 06 juillet 2019, devant le jury composé de :

Aloui Ahmed	**MAA**	Président
Rezeg Khaled	**MCA**	Rapporteur
Boucetta Mebarak	**MAA**	Examinateur

Acknowledgement

First and foremost, praises and thanks to **my God**, **ALLAH** for everything.

I would like to express my sincere gratitude to my supervisor **Dr.REZEG Khaled** for his supervision, availability , follow-up, advice and constructive criticism despite his occupations, and I thank hurry **all the members of the jury.**

I want to thank anyone who has contributed closely or far to the completion of this work.

Finally I would like to thank **all my family** for their constant support.

Thank you

Abstract. Unstructured textual data produced on the Internet is growing rapidly and the analysis of feelings is becoming a challenge because of the limited contextual information they usually contain. Millions of people share daily real-time thoughts and opinions about everything, which generates an unstructured, informative and yet valuable information to data scientists. Traditional approaches are important to the world of consumer behavior because they require a large amount of time and resources, and lead to considerable losses for companies around the world. Text classification is an essential task for automatic natural language processing (NLP) with many applications, such as information retrieval, web search, ranking and spam filtering. The goal of the NLP is to process the text for analysis and extract information for decision support as a first step in our proposed work to propose an efficient and accurate approach for predicting sentiment from raw unstructured data in order to extract opinions from the Internet and predict online popular discussions using a deep learning approach, which can be valuable and decisive for economic and political researchers to serve the country and emerge from crises .

In this work we present an approach for the classification of social media discussions about real-world events like popular mobility in Algeria, and we propose an approach to analyze the feelings of social media messages in Algeria. Applying the different stages of the NLP through the use of deep learning.

Keywords: Analyze des sentiments, opinion mining, text mining, social networks, deep learning.

Summary

List of figures

General introduction

Social networks are a web based communication tools, allows people to have conversations, share information, comment about products, criticism of thoughts and create contents. There are numerous social network sites like twitter, facebook , social networking sites, instant messaging, photo-sharing sites, video-sharing sites and more. Billions of people across the world use social networks sites to share and make connections on a personal level. It becomes imperative that we have to define tools to use this massive data to extract useful information and make best decision and this is what we call sentiment analysis. Sentiment analysis on social network is a rapid and effective way for analyzing public opinion for business marketing or social studies or political cries. For example, political researchers can retrieve timely feedback on a new movement in the street or assembly areas by evaluating people's opinions on social network. As people recently talk about (government, popular movement, political parties, organizations, people, etc.) in a social networks plateformes like Twitter and Facebook. We mine people's opinions on specific entities in each post or discussion rather than the opinion about each whole sentence or whole post ,then we apply our methods and approaches to analyze the sentiments of posts to make best decision in final step that help for more accuracy prediction in the future.

Part 01

Basic concepts

1 Introduction

With the technical progress of the web and storage capacities and exchanges on the Internet. Social networks are experiencing an explosion in terms of data volume and the number of users around the world. This daily use of networks like Facebook ,Twitter and Instagram has changed the image of web 2.0 and given it a new dimension and also new challenges. With this large volume of data, Social networks have recently become a very attractive field of research for many societal actors, especially news agencies, companies, researchers in computer science and information science, psychologists and sociologists. These areas of research mainly concern market research, monitoring of advertising campaigns, trend analysis, analysis of human, social and individual behavior, detection of diseases and identification of influential people, etc.

The monitoring of local sporting, cultural and political events, on social networks increasingly arouse the authorities responsible for these events so a major goal for some lucrative organizations like advertising companies that seek to make the most of, is the extraction of relevant information from these community sites, in the interest of analyzing the sentiments of these events and to be able to improve their actions and sentiment interpretation. But this goal remains limited by the approaches and methods provided in the scientific world, and more particularly in the world of computer research using artificial intelligence

2 Artificial intelligence

Called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and animals. Colloquially, the term "artificial intelligence" is used to describe machines that mimic "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".[1]

2 1 AI problems

The overall research goal of artificial intelligence is to create technology that enables computers and machines to function in an intelligent manner. The general problem of simulating (or creating) intelligence has been broken down into sub-problems.

These consist of particular features or capabilities that researchers expect an intelligent system to display. The features given below have received the most attention.

2 1 1 Reasoning, problem solving

Early developers developed algorithms that imitated step-by-step reasoning that humans use when they solve puzzles or make logical deductions .By the late 1980s and 1990s, employing concepts from probability and economics.

2 1 2 Knowledge representation

Are central to classical AI research. Some "expert systems" attempt to gather together explicit knowledge acquired by experts in some narrow domain.

2 1 3 Planning

Intelligent agents must be able to set goals and achieve them .They need a way to visualize the future a representation of the state of the world and be able to make predictions about how their actions will change it—and be able to make choices that maximize the utility (or "value") of available choices .

2 1 4 Learning

Machine learning, a fundamental concept of AI research since the field's inception, is the study of computer algorithms that improve automatically through experience. Unsupervised learning is the ability to find patterns in a stream of input. Supervised learning includes both classification and numerical regression, which requires a human to label the data input first. Classification is used to determine what category of things belongs to a number of examples of things. Regression is the attempt to produce a function that describes the relationship between inputs and outputs and predicts change. Both classifiers and regression learners can be viewed as "function approximates" trying to learn an unknown (possibly implicit) function; for example, a spam classifier "spam" or "spam". Computational learning theory can be learned by computational complexity, by sample complexity, or by other notions of optimization. In reinforcement learning the agent is rewarded for good responses and punished for bad ones. The agent uses this sequence of rewards and punishments

2 1 5 Natural language processing (NLP)

NLP gives machines the ability to read and understand human language. A sufficiently powerful natural language processing system would enable natural-language user interfaces and the acquisition of knowledge directly from human-written sources, such as newswire texts. Some straightforward applications of natural language processing include information retrieval, text mining, question answering and machine translation.

2 1 6 Motion and manipulation

AI is extensively used in robotics. Robotic arms and other industrial robots, widely used in modern factories, can be used to reduce friction and slippage. A modern mobile robot, where given a small, static, and visible environment, can easily determine its location and map its environment; However, dynamic environments, such as (in endoscopy) the interior of a patient's breathing body, pose a greater challenge.

2 1 7 Social intelligence

SI is an interdisciplinary umbrella that includes systems that recognize, interpret, process, or simulate human affects . Moderate successes related to affective computing and the analysis of a problem, and a more recent study of multimodal sentiment analysis.

3 Machine learning (ML)

The first step in the study of computer algorithms and the use of mathematical methods in the field of computer science, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on a sample data, known as "training data", Machine learning algorithms are used in a wide variety of applications, such as email filtering, and computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.for example: Kernel machines are used to compute non-linearly separable functions into a higher dimension linearly separable function.figure1

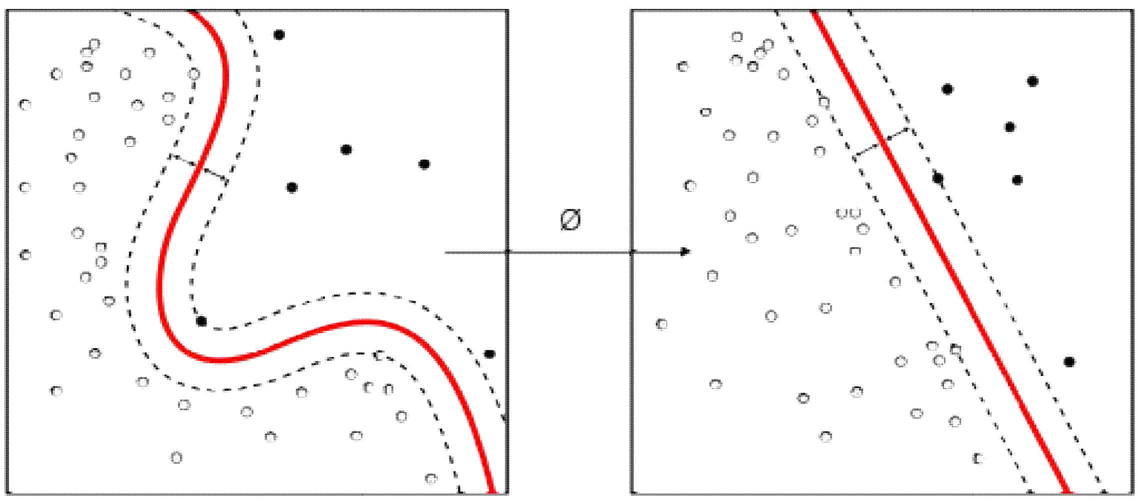


Figure 1: Machine learning

3 1 Machine learning tasks

At a high-level, machine learning is simply the study of teaching a computer program or algorithm how to progressively improve upon a set task that it is given. On the research-side of things, machine learning can be viewed through the lens of theoretical and mathematical modeling of how this process works. However, more practically it is the study of how to build applications that exhibit this iterative improvement. There are many ways to frame this idea, but largely there are three major recognized categories: supervised learning, unsupervised learning, and reinforcement learning.figure2.

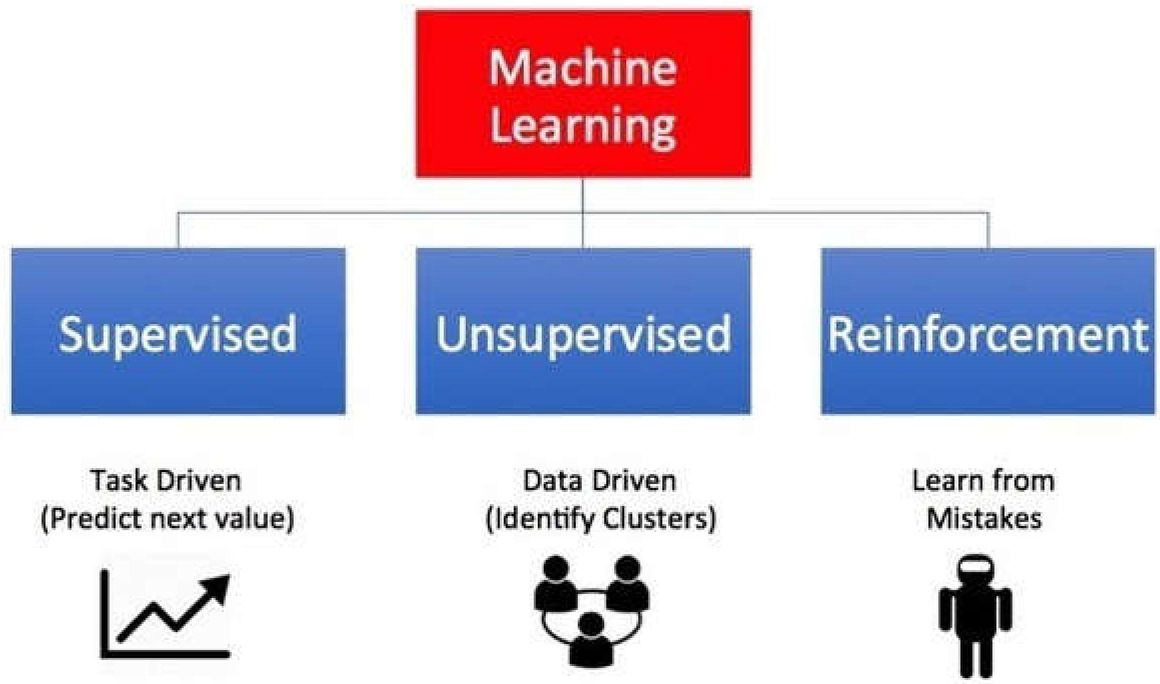


Figure 2: Machine learning tasks

3 1 1 Supervised Learning

Supervised learning is the most popular paradigm for machine learning. It is the easiest to understand and the simplest to implement. Given data in the form of examples with labels, we can feed a learning algorithm these example-label pairs one by one, allowing the algorithm to predict the label for each example, and giving it feedback as to whether it predicted the right answer or not. Over time, the algorithm will learn to approximate the exact nature of the relationship between examples and their labels. When fully-trained, the supervised learning algorithm will be able to observe a new, never-before-seen example and predict a good label for it.figure3.

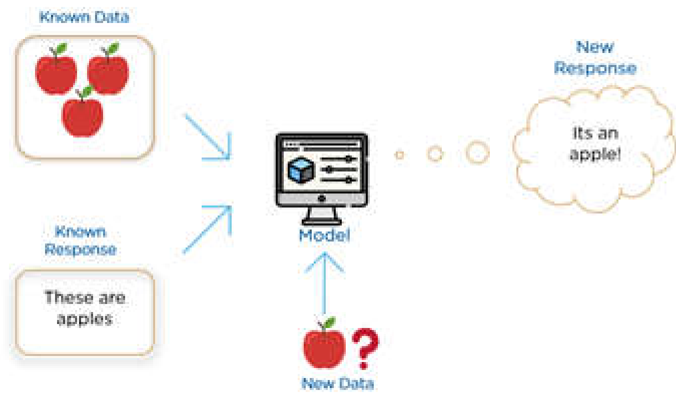


Figure03: Supervised Learning

Supervised learning is often described as task-oriented because of this. It is highly focused on a singular task, feeding more and more examples to the algorithm until it can accurately perform on that task. This is the learning type that you will most likely encounter, as it is exhibited in many of the following common applications:

- **Advertisement Popularity**: Selecting advertisements that will perform well is often a supervised learning task. Many of the ads you see as you browse the internet are placed there because a learning algorithm said that they were of reasonable popularity (and click ability). Furthermore, its placement associated on a certain site or with a certain query (if you find yourself using a search engine) is largely due to a learned algorithm saying that the matching between ad and placement will be effective.
- **Spam Classification**: If you use a modern email system, chances are you've encountered a spam filter. That spam filter is a supervised learning system. Fed email examples and labels (spam/not spam), these systems learn how to preemptively filter out malicious emails so that their user is not harassed by them. Many of these also behave in such a way that a user can provide new labels to the system and it can learn user preference.
- **Face Recognition**: Do you use Facebook? Most likely your face has been used in a supervised learning algorithm that is trained to recognize your face. Having a system that takes a photo, finds faces, and guesses who that is in the photo

(suggesting a tag) is a supervised process. It has multiple layers to it, finding faces and then identifying them, but is still supervised nonetheless.[2]

3 1 2 Unsupervised Learning

Unsupervised learning is very much the opposite of supervised learning. It features no labels. Instead, our algorithm would be fed a lot of data and given the tools to understand the properties of the data. From there, it can learn to group, cluster, and/or organize the data in a way such that a human (or other intelligent algorithm) can come in and make sense of the newly organized data.figure4

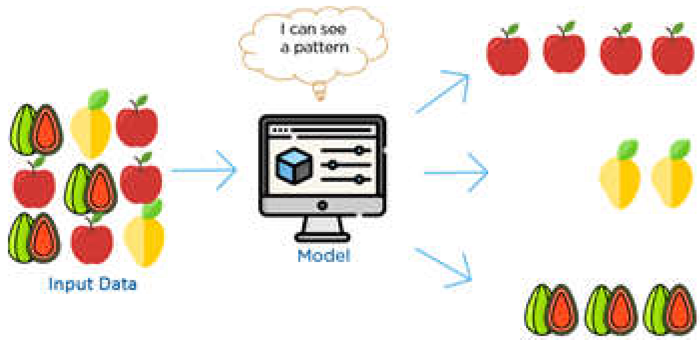


Figure4: unsupervised learning

What makes unsupervised learning such an interesting area is that an overwhelming majority of data in this world is unlabeled. Having intelligent algorithms that can take our terabytes and terabytes of unlabeled data and make sense of it is a huge source of potential profit for many industries. That alone could help boost productivity in a number of fields.

3 1 3 Reinforcement Learning

Reinforcement learning is fairly different when compared to supervised and unsupervised learning. Where we can easily see the relationship between supervised and unsupervised (the presence or absence of labels), the relationship to reinforcement learning is a bit murkier. Some people try to tie reinforcement learning closer to the two by describing it as a type of learning that relies on a time-dependent sequence of labels, however, my opinion is that that simply makes things more confusing.I prefer to look at reinforcement learning as learning from mistakes. Place a reinforcement learning algorithm into any environment and it will make a lot of mistakes in the beginning. So long as we provide some sort of signal to the algorithm that associates good behaviors with a positive signal and bad behaviors with a negative one, we can reinforce our algorithm to prefer good behaviors over bad ones. Over time, our learning algorithm learns to make less mistakes than it used to.figure5

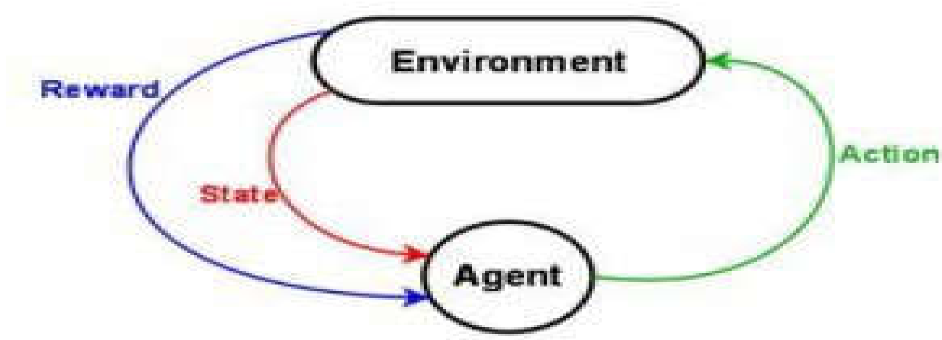


Figure 5: reinforcement learning

Reinforcement learning is very behavior driven. It has influences from the fields of neuroscience and psychology. If you've heard of Pavlov's dog, then you may already be familiar with the idea of reinforcing an agent, albeit a biological one. However, to truly understand reinforcement learning. For any reinforcement learning problem, we need an agent and an environment as well as a way to connect the two through a feedback loop. To connect the agent to the environment, we give it a set of actions that it can take that affect the environment. To connect the environment to the agent, we have it continually issue two signals to the agent: an updated state and a reward (our reinforcement signal for behavior).

3 2 Machine Learning algorithms

Machine Learning algorithm is an evolution of the regular algorithm. It makes your programs “smarter”, by allowing them to automatically learn from the data you provide. The algorithm is mainly divided into:

- Training Phase
- Testing phase

Training Phase

You take a randomly selected specimen of apples from the market (**training data**), make a table of all the physical characteristics of each apple, like color, size, shape, grown in which part of the country, sold by which vendor, etc (**features**), along with the sweetness, juiciness, ripeness of that apple (**output variables**). You feed this data to the machine learning algorithm (**classification/regression**), and it learns a model of the correlation between an average apple’s physical characteristics, and its quality.

Testing Phase

Next time when you go shopping, you will measure the characteristics of the apples which you are purchasing(**test data**)and feed it to the Machine Learning algorithm. It will use the model which was computed earlier to predict if the apples are sweet, ripe and/or juicy. The algorithm may internally use the rules, similar to the one you manually wrote earlier (for eg, a **decision tree**). Finally, you can now shop for apples with great confidence, without worrying about the details of how to choose the best apples.[3]

3 2 1 types of Machine Learning Algorithms

So, Machine Learning Algorithms can be categorized by the following three types. As the following figure shows .figure6

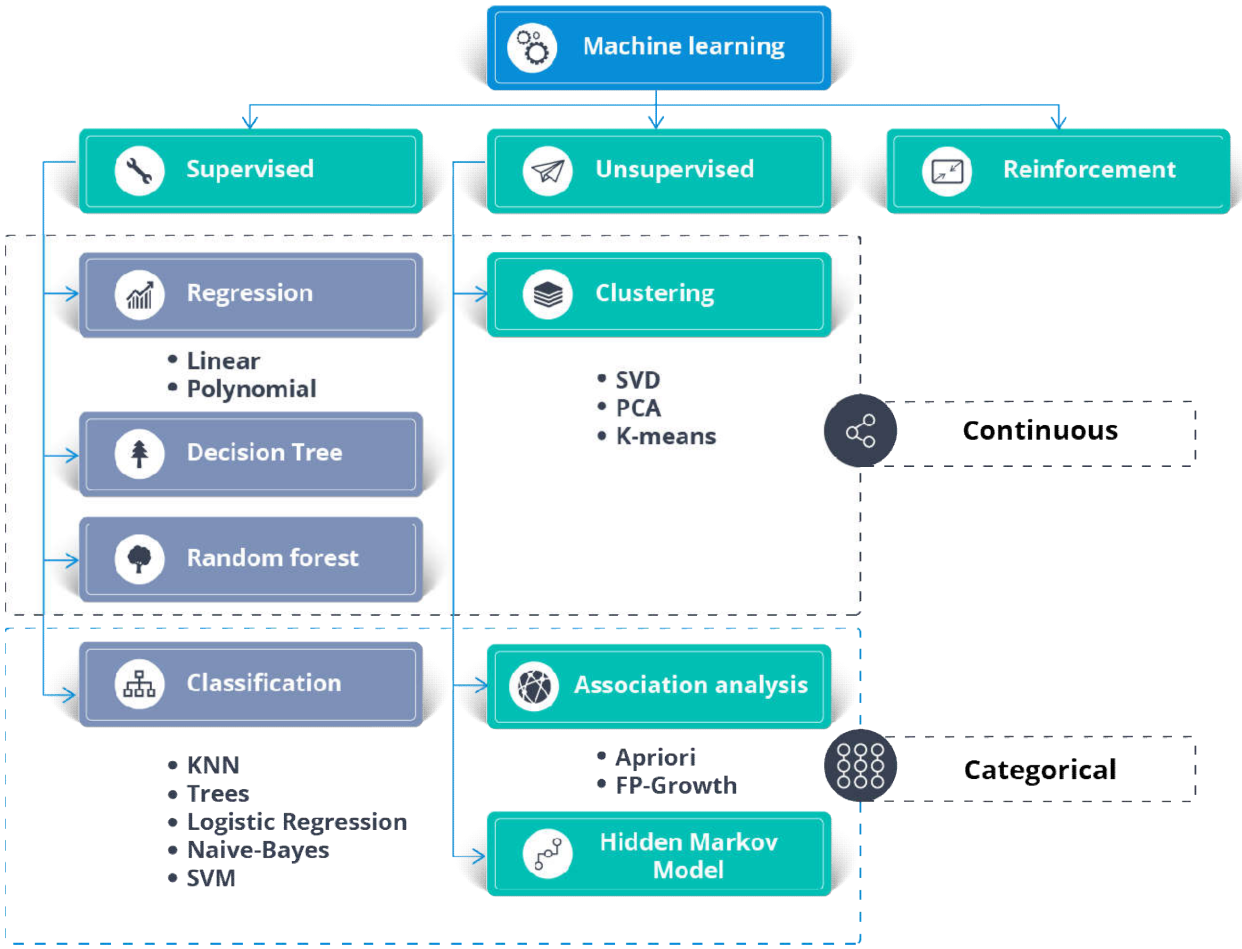


Figure 6: ML algorithmes types

3 2 2 List of Machine Learning Algorithms

Here is the list of 06 most commonly used machine learning algorithms.

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. Naive Bayes
5. Neural networks

3 2 2 1 Linear Regression

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variables. Here, we establish a relationship between the independent and dependent variables by fitting the best line. This best fit line is known as the *regression line* and represented by a linear equation **Y= aX + b**.The best way to understand linear regression is to relive this experience of childhood. Let us say, you ask a child in fifth grade to arrange people in his class by increasing order of weight, without asking them their weights! What do you think the child will do? He/she would likely look (visually analyze) at the height and build of people and arrange them using a combination of these visible parameters. This is a linear regression in real life! The child has actually figured out that height and build would be correlated to the weight by a relationship, which looks like the equation above.figure7

In this equation:

- **Y – Dependent Variable**
- **a – Slope**
- **X – Independent variable**
- **b – Intercept**

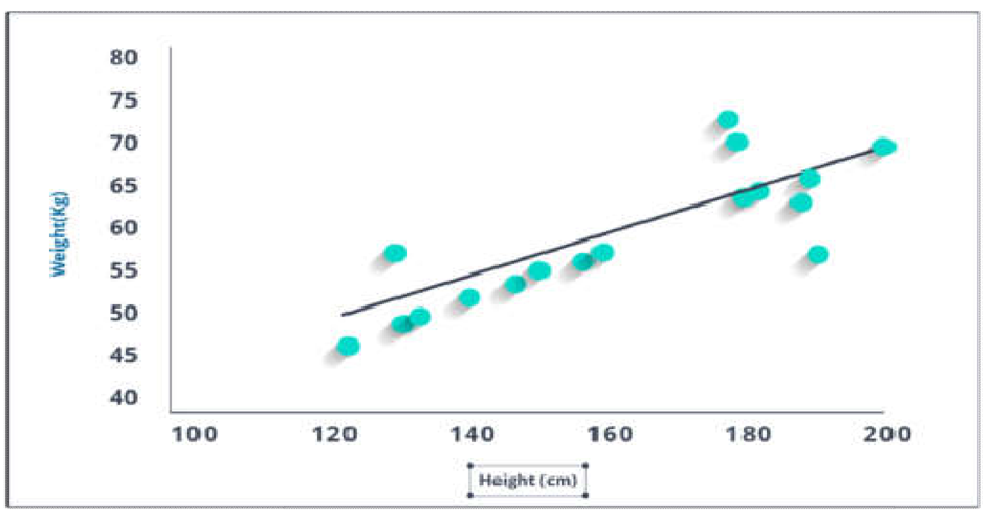


Figure 7: linear equation

These coefficients **a** and **b** are derived based on minimizing the 'sum of squared differences' of distance between data points and regression line.

3 2 2 2 Logistic Regression

It is a classification, and not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on a given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a *logit function*. Hence, it is also known as *logit regression*. Since it predicts the probability, its output values lie between 0 and 1.figure8

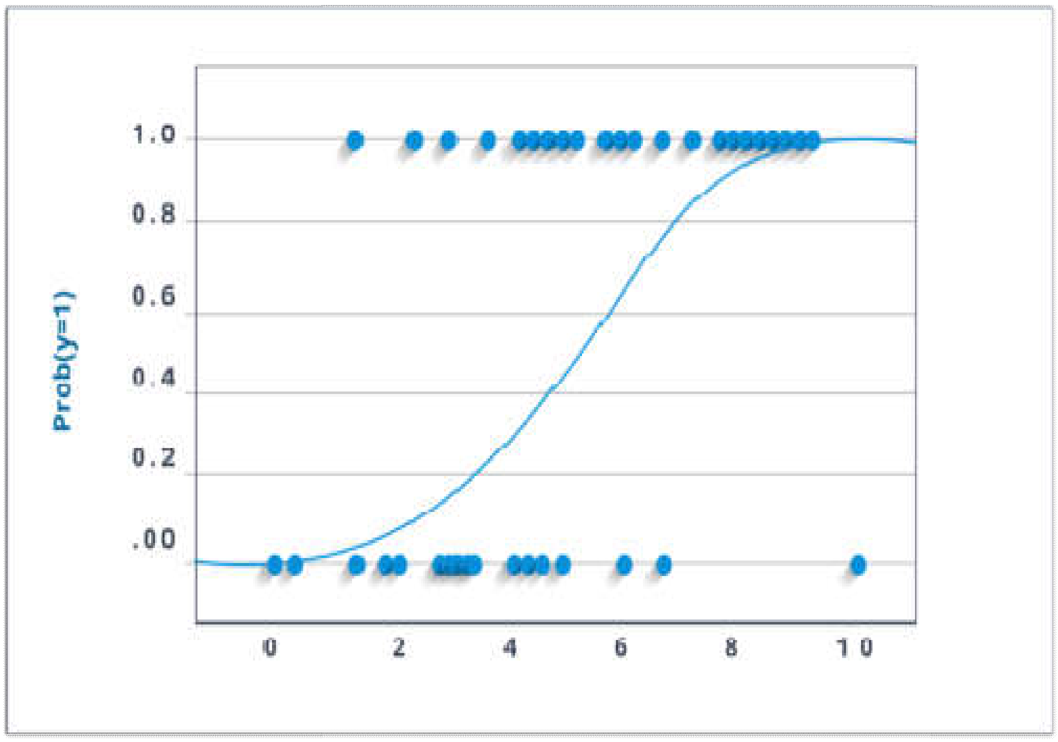


Figure 8 : *logit function*

3 2 2 3 Decision Tree

It is a type of supervised learning algorithm that is mostly used for classification problems. Surprisingly, it works for both categorical and continuous dependent variables. In this algorithm, we split the population into two or more homogeneous sets. This is done based on the most significant attributes/ independent variables to make as distinct groups as possible. look to the following figure .figure9

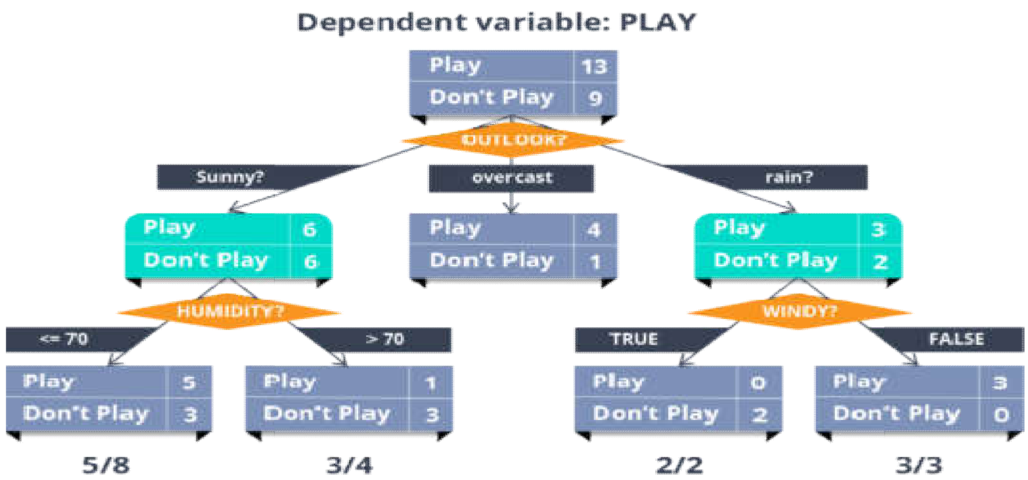


Figure 9: decision tree

In the image above, you can see that population is classified into four different groups based on multiple attributes to identify 'if they will play or not'.

3 2 2 4 Naive Bayes

This is a classification technique based on *Bayes' theorem* with an assumption of independence between predictors. In simple terms, a ***Naive Bayes classifier*** assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier would consider all of these properties to independently contribute to the probability that this fruit is an apple.Naive Bayesian model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.Bayes theorem provides a way of calculating posterior probability **P(c|x)** from **P(c)**, **P(x)** and **P(x|c)**. Look at the equation below:

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

3 2 2 5 KNN (k- Nearest Neighbors)

It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. ***K nearest neighbors*** is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function .These distance functions can be Euclidean, Manhattan, Minkowski and Hamming distance. First three functions are used for continuous function and the fourth one (Hamming) for categorical variables. If **K = 1**, then the case is simply assigned to the class of its nearest neighbor. At times, choosing K turns out to be a challenge while performing kNN modeling. Look figure10

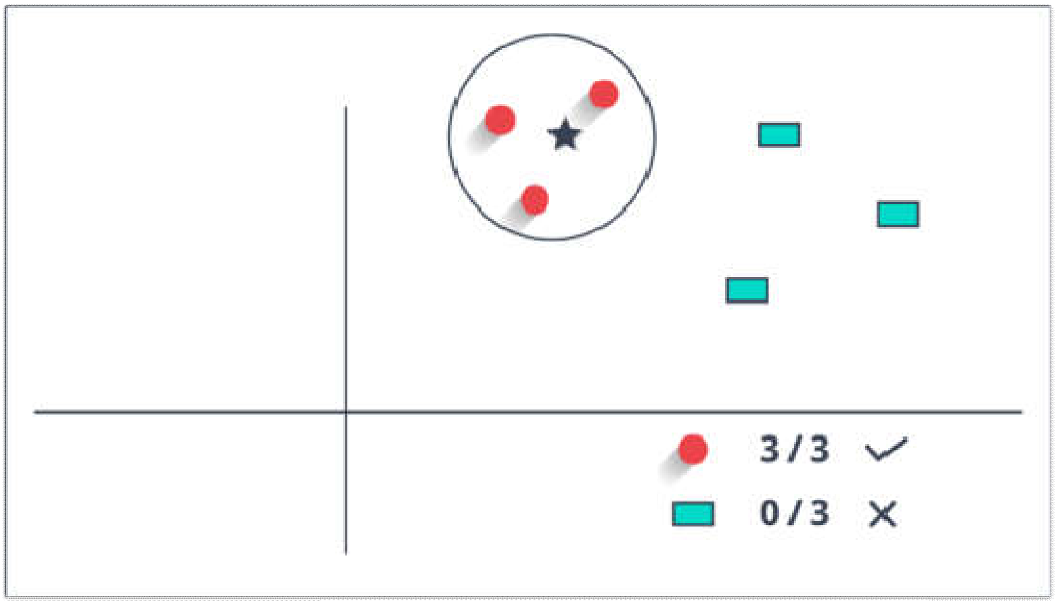


Figure 10: k- Nearest Neighbors

KNN can easily be mapped to our real lives. If you want to learn about a person, of whom you have no information, you might like to find out about his close friends and the circles he moves in and gain access to his/her information!

3 2 2 6 Neural networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

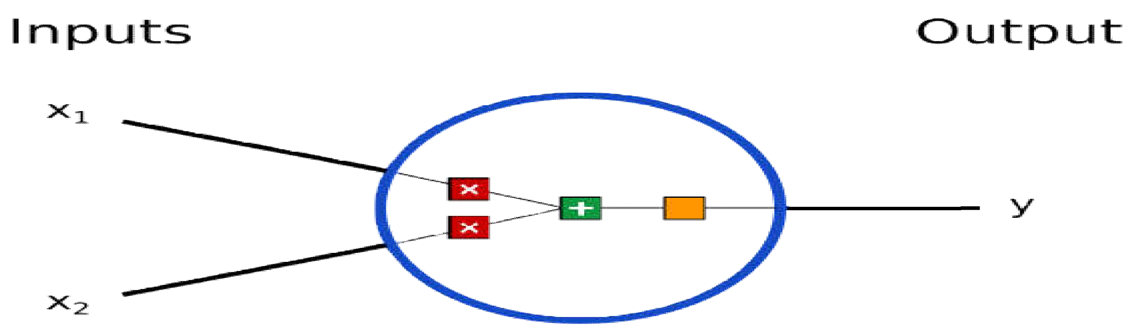


Figure 11: ANN

4 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 driverless 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.[4]

4 1 The importance of deep learning

In a word, accuracy. Deep learning achieves recognition accuracy at higher levels than ever before. This helps consumer electronics meet user expectations, and it is crucial for safety-critical applications like driverless cars. Recent advances in deep learning have improved to the point where deep learning outperforms humans in some tasks like classifying objects in images.

While deep learning was first theorized in the 1980s, there are two main reasons it has only recently become useful:

1. Deep learning requires large amounts of **labeled data**. For example, driverless car development requires millions of images and thousands of hours of video.
2. Deep learning requires substantial **computing power**. High-performance GPUs have a parallel architecture that is efficient for deep learning. When combined with clusters or cloud computing, this enables development teams to reduce training time for a deep learning network from weeks to hours or less.

4 2 How Deep Learning Works

Most deep learning methods use **neural network** architectures, which is why deep learning models are often referred to as **deep neural networks** .The term “deep” usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150.Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.

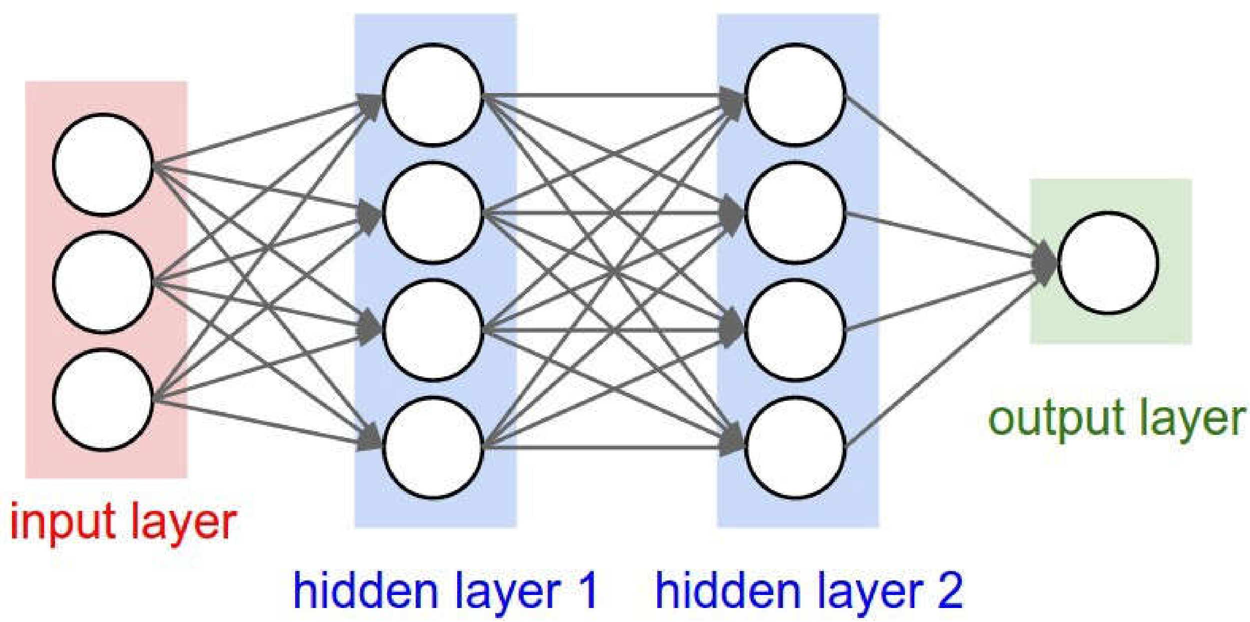


Figure 12: Neural networks

4 3 Difference Between Machine Learning and Deep Learning

Deep learning is a specialized form of machine learning. A machine learning workflow starts with relevant features being manually extracted from images. The features are then used to create a model that categorizes the objects in the image. With a deep learning workflow, relevant features are automatically extracted from images. In addition, deep learning performs “end-to-end learning” – where a network is given raw data and a task to perform, such as classification, and it learns how to do this automatically .Another key difference is deep learning algorithms scale with data, whereas shallow learning converges. Shallow learning refers to machine learning methods that plateau at a certain level of performance when you add more examples and training data to the network .A key advantage of deep learning networks is that they often continue to improve as the size of your data increases.[5]

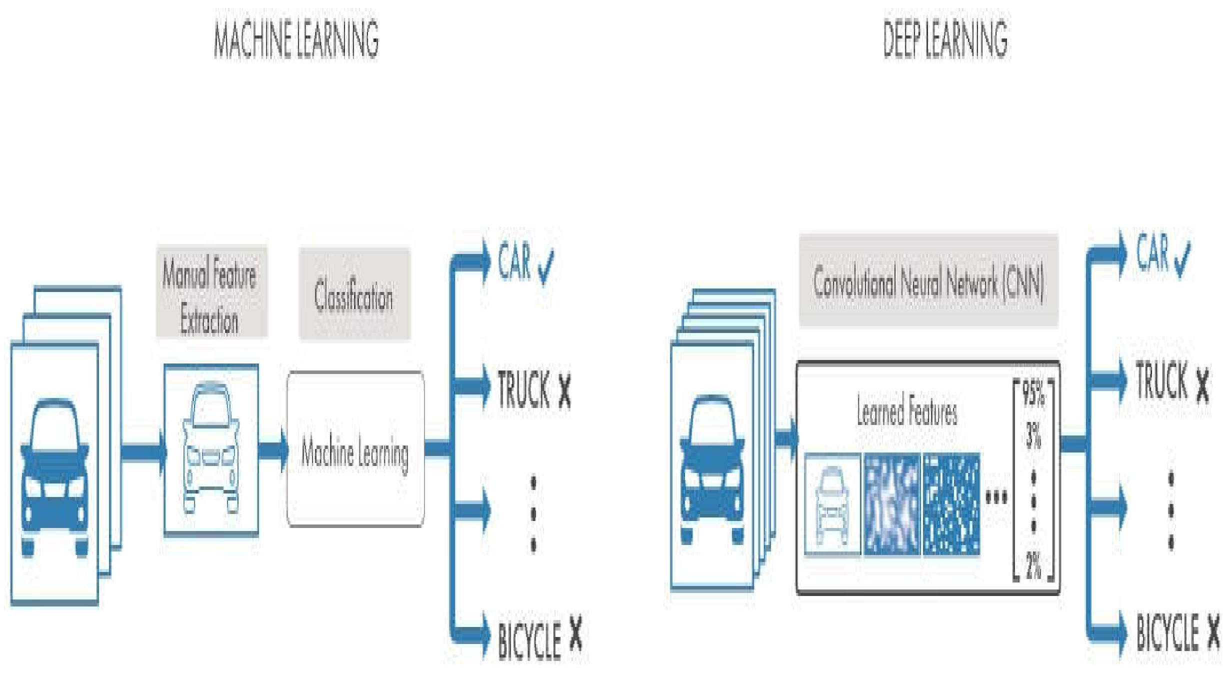


Figure 13: difference between Machine Learning and Deep Learning

4 4 How to Create and Train Deep Learning Models

The three most common ways people use deep learning to perform object classification are:

Training from Scratch

To train a deep network from scratch, you gather a very large labeled data set and design a network architecture that will learn the features and model. This is good for new applications, or applications that will have a large number of output categories. This is a less common approach because with the large amount of data and rate of learning, these networks typically take days or weeks to train.

Transfer Learning

Most deep learning applications use the transfer learning approach, a process that involves fine-tuning a pretrained model. You start with an existing network, such as AlexNet or Google Net, and feed in new data containing previously unknown classes. After making some tweaks to the network, you can now perform a new task, such as categorizing only dogs or cats instead of 1000 different objects. This also has the advantage of needing much less data (processing thousands of images, rather than millions), so computation time drops to minutes or hours .Transfer learning requires an interface to the internals of the pre-existing network, so it can be surgically modified and enhanced for the new task.

Feature Extraction

A slightly less common, more specialized approach to deep learning is to use the network as a **feature extractor**. Since all the layers are tasked with learning certain features from images, we can pull these features out of the network at any time during the training process. These features can then be used as input to a machine learning model such as support vector machines (SVM).

4 5 Deep Neural Network

Deep Neural Networks (DNNs) are typically Feed Forward Networks (FFNNs) in which data flows from the input layer to the output layer without going backward[3] and the links between the layers are one way which is in the forward direction and they never touch a node again.

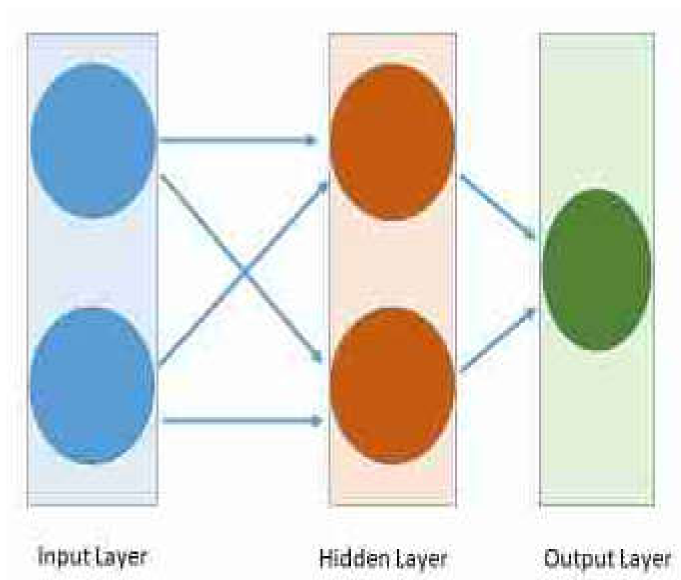


Figure 14: DNN

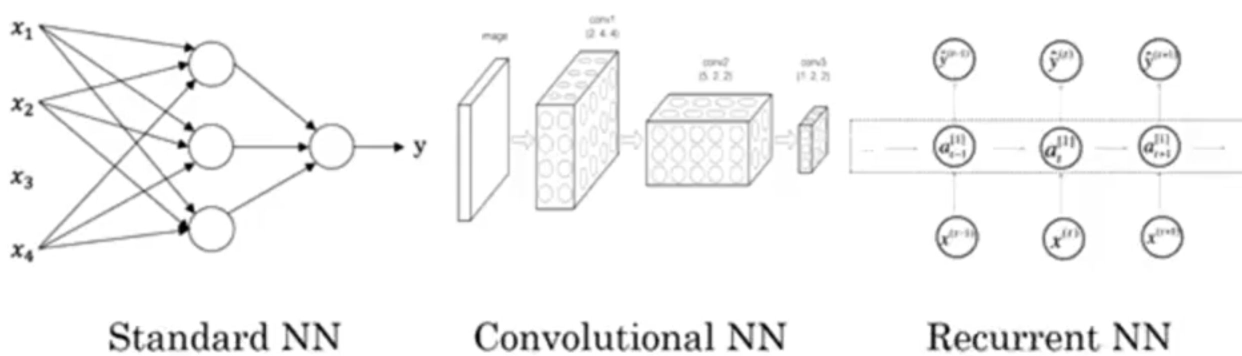


Figure 15: DNN types

4 5 1 Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) addresses this issue which is a FFNN with a time twist. This neural network isn't stateless, has connections between passes and connections through time. They are a class of artificial neural network where connections between nodes form a directed graph along a sequence like features links from a layer to previous layers, allowing information to flow back into the previous parts of the network thus each model in the layers depends on past events, allowing information to persist

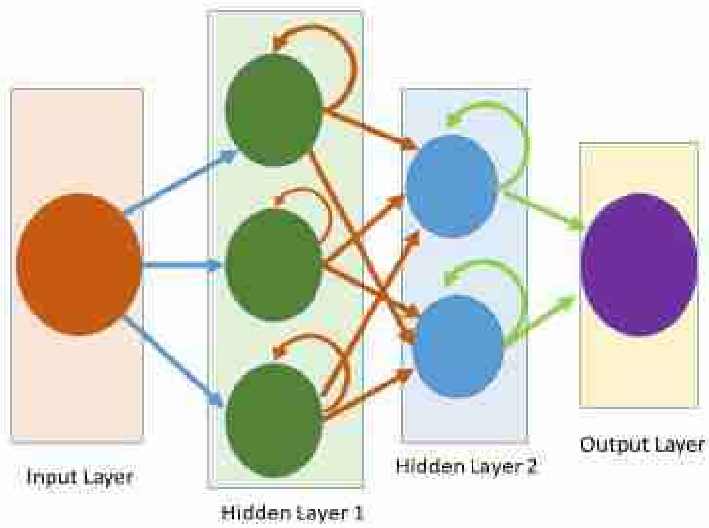


figure 16: RNN

In this way, **RNNs** can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. But they not only work on the information you feed but also on the related information from the past which means whatever you feed and train the network matters, like feeding it ‘chicken’ then ‘egg’ may give different output in comparison to ‘egg’ then ‘chicken’. **RNNs** also have problems like vanishing (or exploding) gradient/long-term dependency problem where information rapidly gets lost over time. Actually, it’s the weight which gets lost when it reaches a value of 0 or 1 000 000, not the neuron. But in this case, the previous state won’t be very informative as it’s the weight which stores the information from the past.

4 5 2 Long Short Term Memory (LSTM)

breakthroughs like **Long Short Term Memory (LSTM)** don’t have this problem! **LSTMs** are a special kind of **RNN**, capable of learning long-term dependencies which make **RNN** smart at remembering things that have happened in the past and finding patterns across time to make its next guesses make sense. **LSTMs** broke records for improved Machine Translation, Language Modeling and Multilingual Language Processing.

4 5 3 Convolutional Neural Network (CNN)

Next comes the Convolutional Neural Network (CNN, or ConvNet) which is a class of deep neural networks which is most commonly applied to analyzing visual imagery. Their other applications include video understanding, speech recognition and understanding natural language processing. Also, **LSTM** combined with **Convolutional Neural Networks (CNNs)** improved automatic image captioning like those are seen in Facebook. Thus you can see that **RNN** is more like helping us in data processing predicting our next step whereas **CNN** helps us in visuals analyzing.[6]

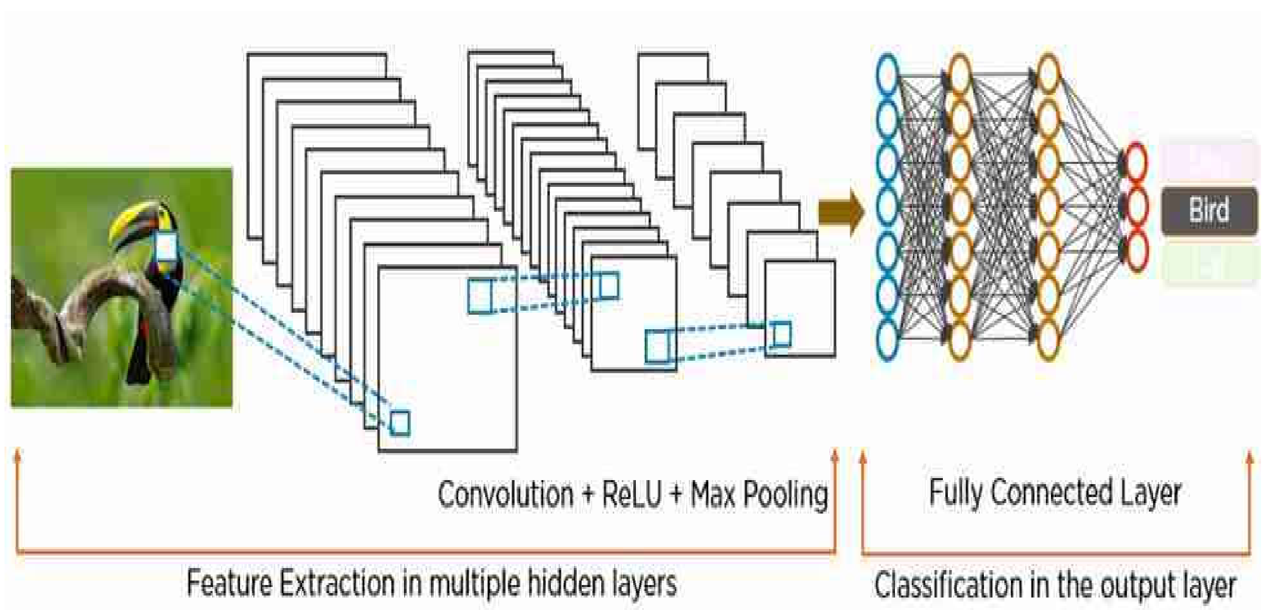


Figure 17: CNN

4 5 4 Types of neural networks to solve problems

Input (X)	Output (y)	Application	Type of Neural Network
Home Features	Price	Real Estate	Standard Neural Network
Ad, user info	Click prediction (0/1)	Online Advertising	Standard Neural Network
Image	Image Class	Photo Tagging	CNN
Audio	Text Transcript	Speech Recognition	RNN
English	Chinese	Machine Translation	RNN
Image, Radar info	Position of car	Autonomous Driving	Custom / Hybrid NN

Figure 18: types of NN to solve problems

5 Natural Language Processing

Natural Language Processing, usually shortened as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language .The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable .Most NLP techniques rely on machine learning to derive meaning from human languages.

5 1 Use of NLP

Natural Language Processing is the driving force behind the following common applications:

- Language translation applications such as Google Translate
- Word Processors such as Microsoft Word and Grammarly that employ NLP to check grammatical accuracy of texts.
- Interactive Voice Response (IVR) applications used in call centers to respond to certain users' requests.
- Personal assistant applications such as OK Google, Siri, Cortana, and Alexa.

5 2 Difficulty of NLP

Natural Language processing is considered a difficult problem in computer science. It's the nature of the human language that makes NLP difficult.The rules that dictate the passing of information using natural languages are not easy for computers to understand .Some of these rules can be high-leveled and abstract; for example, when someone uses a sarcastic remark to pass information. On the other hand, some of these rules can be low-leveled; for example, using the character "s" to signify the plurality of items .Comprehensively understanding the human language requires understanding both the words and how the concepts are connected to deliver the intended message. While humans can easily master a language, the ambiguity and imprecise characteristics of the natural languages are what make NLP difficult for machines to implement.[7]

5 3 How does NLP Works

NLP entails applying algorithms to identify and extract the natural language rules such that the unstructured language data is converted into a form that computers can understand. When the text has been provided, the computer will utilize algorithms to extract meaning associated with every sentence and collect the essential data from them. Sometimes, the computer may fail to understand the meaning of a sentence well, leading to obscure results .For example, a humorous incident occurred in the 1950s during the translation of some words between the English and the Russian languages .Here is the biblical sentence that required translation:

"The spirit is willing, but the flesh is weak."

Here is the result when the sentence was translated to Russian and back to English:

"The DRINK *is good, but the meat is rotten."*

5 4 NLP techniques

Syntactic analysis and semantic analysis are the main techniques used to complete Natural Language Processing tasks .Here is a description on how they can be used.

5 4 1 Syntax

Syntax refers to the arrangement of words in a sentence such that they make grammatical sense. In NLP, syntactic analysis is used to assess how the natural language aligns with the grammatical rules. Computer algorithms are used to apply grammatical rules to a group of words and derive meaning from them.

Here are some syntax techniques that can be used:

- **Lemmatization**: It entails reducing the various inflected forms of a word into a single form for easy analysis.
- **Morphological segmentation**: It involves dividing words into individual units called morphemes.
- **Word segmentation**: It involves dividing a large piece of continuous text into distinct units.

- **Part-of-speech tagging**: It involves identifying the part of speech for every word.
- **Parsing**: It involves undertaking grammatical analysis for the provided sentence.
- **Sentence breaking**: It involves placing sentence boundaries on a large piece of text.
- **Stemming**: It involves cutting the inflected words to their root form.

5 4 2 Semantics

Semantics refers to the meaning that is conveyed by a text. Semantic analysis is one of the difficult aspects of Natural Language Processing that has not been fully resolved yet.

It involves applying computer algorithms to understand the meaning and interpretation of words and how sentences are structured.

Here are some techniques in semantic analysis:

- **Named entity recognition (NER):** It involves determining the parts of a text that can be identified and categorized into preset groups. Examples of such groups include names of people and names of places.
- **Word sense disambiguation:** It involves giving meaning to a word based on the context.
- **Natural language generation**: It involves using databases to derive semantic intentions and convert them into human language.

6 Sentiment Analysis

Sentiment analysis is the automated process of understanding an opinion about a given subject from written or spoken language .In a world where we generate 2.5 quintillion bytes of data every day, sentiment analysis has become a key tool for making sense of that data. This has allowed companies to get key insights and automate all kind of processes.[8]

6 1 what is Sentiment Analysis

Sentiment Analysis also known as *Opinion Mining* is a field within Natural Language Processing (NLP) that builds systems that try to identify and extract opinions within text. Usually, besides identifying the opinion, these systems extract attributes of the expression e.g.:

- *Polarity*: if the speaker express a *positive* or *negative* opinion,
- *Subject*: the thing that is being talked about,
- *Opinion holder*: the person, or entity that expresses the opinion.

Currently, sentiment analysis is a topic of great interest and development since it has many practical applications. Since publicly and privately available information over Internet is constantly growing, a large number of texts expressing opinions are available in review sites, forums, blogs, and social media.With the help of sentiment analysis systems, this unstructured information could be automatically transformed into structured data of public opinions about products, services, brands, politics, or any topic that people can express opinions about. This data can be very useful for commercial applications like marketing analysis, public relations, product reviews, net promoter scoring, product feedback, and customer service.

6 2 Sentiment Analysis Scope

Sentiment analysis can be applied at different levels of scope:

- **Document level** sentiment analysis obtains the sentiment of a complete document or paragraph.
- **Sentence level** sentiment analysis obtains the sentiment of a single sentence.
- **Sub-sentence level** sentiment analysis obtains the sentiment of sub-expressions within a sentence.

6 3 Types of Sentiment Analysis

There are many types and flavors of sentiment analysis and SA tools range from systems that focus on polarity (positive, negative, neutral) to systems that detect feelings and emotions (*angry*, *happy*, *sad*, etc) or identify intentions (e.g. *interested* v. *not interested*). In the following section, we'll cover the most important ones.

6 3 1 Fine-grained Sentiment Analysis

Sometimes you may be also interested in being more precise about the level of polarity of the opinion, so instead of just talking about *positive*, *neutral*, or *negative* opinions you could consider the following categories:

- Very positive
- Positive
- Neutral
- Negative
- Very negative

This is usually referred to as fine-grained sentiment analysis. This could be, for example, mapped onto a 5-star rating in a review, e.g.: Very Positive = 5 stars and Very Negative=1 star.

Some systems also provide different flavors of polarity by identifying if the positive or negative sentiment is associated with a particular feeling, such as, anger, sadness, or worries (i.e. negative feelings) or happiness, love, or enthusiasm (i.e. positive feelings).

6 3 2 Emotion detection

Emotion detection aims at detecting emotions like, happiness, frustration, anger, sadness, and the like. Many emotion detection systems resort to lexicons (i.e. lists of words and the emotions they convey) or complex machine learning algorithms.

One of the downsides of resorting to lexicons is that the way people express their emotions varies a lot and so do the lexical items they use. Some words that would typically express anger like *shit* or *kill* (e.g. *in your product is a piece of shit* or *your customer support is killing me*) might also express happiness (e.g. in texts like *This is the shit* or *You are killing it*).

6 3 3 Aspect-based Sentiment Analysis

Usually, when analyzing the sentiment in subjects, for example products, you might be interested in not only whether people are talking with a positive, neutral, or negative polarity about the product, but also which particular aspects or features of the product people talk about. That's what aspect-based sentiment analysis is about. In our previous example:

"The battery life of this camera is too short."

The sentence is expressing a negative opinion about the camera, but more precisely, about the battery life, which is a particular feature of the camera.

6 3 4 Intent analysis

Intent analysis basically detects what people want to do with a text rather than what people say with that text. Look at the following examples:

“Your customer support is a disaster. I’ve been on hold for 20 minutes”.

“I would like to know how to replace the cartridge”.

“Can you help me fill out this form?”

A human being has no problems detecting the complaint in the first text, the question in the second text, and the request in the third text. However, machines can have some problems to identify those. Sometimes, the intended action can be inferred from the text, but sometimes, inferring it requires some contextual knowledge.

6 3 5 Multilingual sentiment analysis

Multilingual sentiment analysis can be a difficult task. Usually, a lot of preprocessing is needed and that preprocessing makes use of a number of resources. Most of these resources are available online (e.g. sentiment lexicons), but many others have to be created (e.g. translated corpora or noise detection algorithms). The use of the resources available requires a lot of coding experience and can take long to implement .An alternative to that would be detecting language in texts automatically, then train a custom model for the language of your choice (if texts are not written in English), and finally, perform the analysis.

6 4 Sentiment Analysis Algorithms

There are many methods and algorithms to implement sentiment analysis systems, which can be classified as:

- **Rule-based** systems that perform sentiment analysis based on a set of manually crafted rules.
- **Automatic** systems that rely on machine learning techniques to learn from data.
- **Hybrid** systems that combine both rule based and automatic approaches.

6 4 1 Rule-based Approaches

Usually, rule-based approaches define a set of rules in some kind of scripting language that identify subjectivity, polarity, or the subject of an opinion. The rules may use a variety of inputs, such as the following:

- Classic NLP techniques like *stemming*, *tokenization*, *part of speech tagging* and *parsing*.
- Other resources, such as lexicons (i.e. lists of words and expressions).

A basic example of a rule-based implementation would be the following:

1. Define two lists of polarized words (e.g. negative words such as *bad*, *worst*, *ugly*, etc and positive words such as *good*, *best*, *beautiful*, etc).
2. Given a text:
    1. Count the number of positive words that appear in the text.
    2. Count the number of negative words that appear in the text.
3. If the number of positive word appearances is greater than the number of negative word appearances return a positive sentiment, conversely, return a negative sentiment. Otherwise, return neutral.

This system is very naïve since it doesn't take into account how words are combined in a sequence. A more advanced processing can be made, but these systems get very complex quickly. They can be very hard to maintain as new rules may be needed to add support for new expressions and vocabulary. Besides, adding new rules may have undesired outcomes as a result of the interaction with previous rules. As a result, these systems require important investments in manually tuning and maintaining the rules.

6 4 2 Automatic Approaches

Automatic methods, contrary to rule-based systems, don't rely on manually crafted rules, but on machine learning techniques. The sentiment analysis task is usually modeled as a classification problem where a classifier is fed with a text and returns the corresponding category, e.g. positive, negative, or neutral (in case polarity analysis is being performed).

Said machine learning classifier can usually be implemented with the following steps and components:

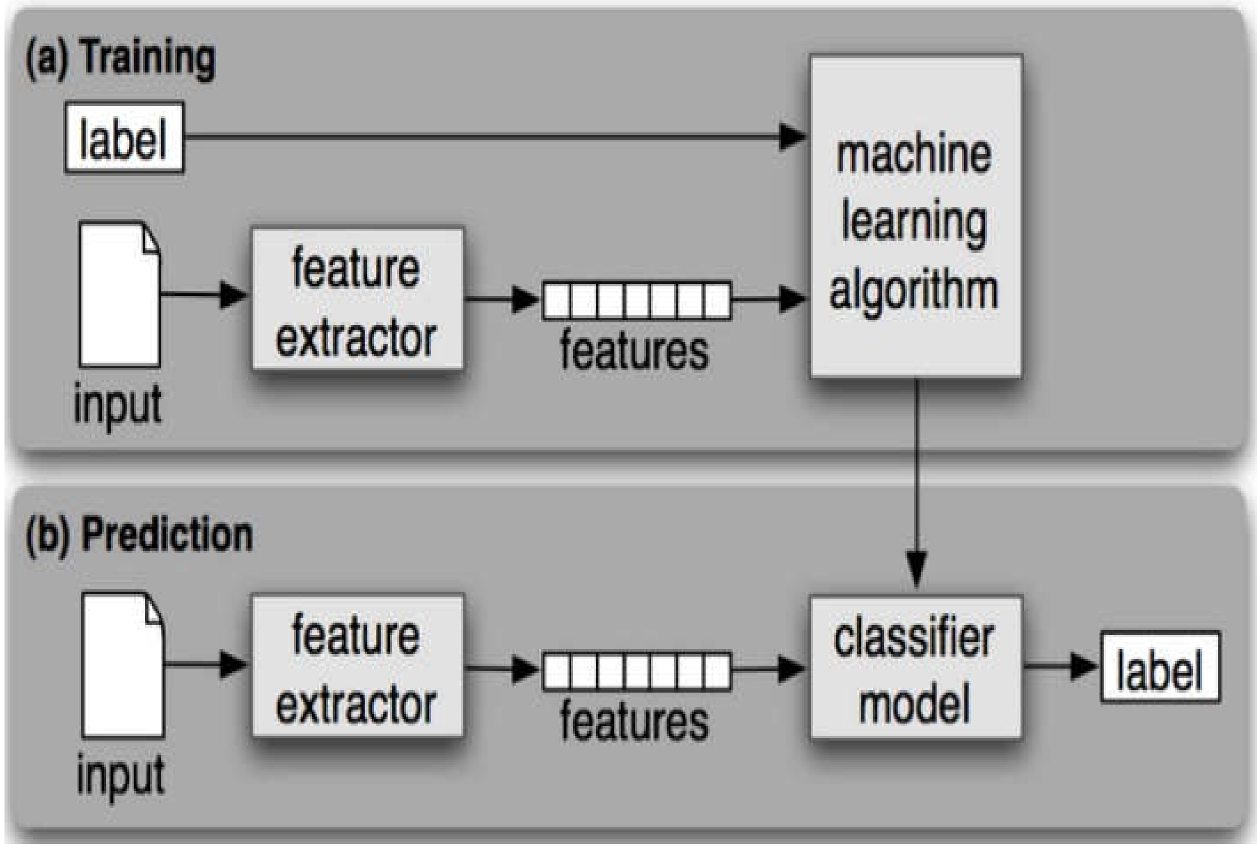


Figure 19: ML classifier

6 4 3 The Training and Prediction Processes

In the training process (a), our model learns to associate a particular input (i.e. a text) to the corresponding output (tag) based on the test samples used for training. The feature extractor transfers the text input into a feature vector. Pairs of feature vectors and tags (e.g. *positive*, *negative*, or *neutral*) are fed into the machine learning algorithm to generate a model.

In the prediction process (b), the feature extractor is used to transform unseen text inputs into feature vectors. These feature vectors are then fed into the model, which generates predicted tags (again, *positive*, *negative*, or *neutral*).

6 4 4 Feature Extraction from Text

The first step in a machine learning text classifier is to transform the text into a numerical representation, usually a vector. Usually, each component of the vector represents the frequency of a word or expression in a predefined dictionary (e.g. a lexicon of polarized words). This process is known as feature extraction or text vectorization and the classical approach has been bag-of-words or bag-of-ngrams with their frequency.

More recently, new feature extraction techniques have been applied based on word embeddings (also known as *word vectors*). This kind of representations makes it possible for words with similar meaning to have a similar representation, which can improve the performance of classifiers.

6 4 5 Classification Algorithms

The classification step usually involves a statistical model like Naïve Bayes, Logistic Regression, Support Vector Machines, or Neural Networks:

- **Naïve Bayes** : a family of probabilistic algorithms that uses Bayes's Theorem to predict the category of a text.
- **Linear Regression** : a very well-known algorithm in statistics used to predict some value (Y) given a set of features (X).
- **Support Vector Machines** : a non-probabilistic model which uses a representation of text examples as points in a multidimensional space. These examples are mapped so that the examples of the different categories (sentiments) belong to distinct regions of that space.. Then, new texts are mapped onto that same space and predicted to belong to a category based on which region they fall into.
- **Deep Learning** : a diverse set of algorithms that attempts to imitate how the human brain works by employing artificial neural networks to process data.

7 Sentiment Analysis Challenges and importance

In view of the global events, especially the Middle East and the Arab region of conflicts and wars Projections of government systems and the exit of people to the street to hear his voice and can be an example of what is happening in Algeria the popular movement(الحراك الشعبي) entered in the third month of his goal to make radical changes can be said, Of the crisis, which requires the intervention of the wise and experts and researchers to get the country to safety with minimal damage and why not enter the researchers in artificial intelligence to propose solutions and smart proposals as soon as, we see here the importance of Sentiment Analysis using NLP and accuracy approaches as Deep learning to benefit of :Improved accuracy and efficiency of documentation, the ability to automatically make a readable summary text and allows an organization to use chatbots for popular support which we can put it to the city hall to understand the point of popular and exit the country from the cries in the perfect time with most damage .

8 Conclusion

We have so far demonstrated the theoretical part of our research, which has to be mentioned to understand the scientific fundamentals and rules, which allows an understanding of our work proposed in the following sections .The rest of this research is organized as follows: part02 discusses the problem and some of the related work, and we will presents our proposed approach and the results obtained, and finally, we conclude our research with a conclusion and references at the end.

Part 02

Design and implementation

1 Problem

Posts posted on Twitter or Facebook or in social network in general reflect the interaction of users with actual events happening around the world, such as elections, popular movement, sporting and cultural events, natural disasters, etc. These actual events have a direct impact on the amount of tweets uploaded. Tracking these events on social networks generally and on Twitter more specifically is a bold challenge for researchers, first of all because a subject on Twitter is characterized by multiple terms (these terms may be hashtags) that may dynamically change where some may become less used and others may appear . So it is essential to find a way to cover all these post or discussion used during the sentiment analysis process. This is one of our goals in this work. But before look for the the sentiment of posts , we need to be able to identify sets of tweets that are talking about the same subject and that represent a thread, which defines the main objective of our work .We are supposed to collect the data set according to the events currently taking place in Algeria from elections, demonstrations, rallies, strikes and popular movements. To carry out the necessary analysis of our proposed work and to find it difficult to obtain and collect sufficient data for the success of the proposed model in time ,for that problem we have to develop the model depending on the data set of other purpose only to ensure the success of the model developed in waiting for the necessary and sufficient data set, which we have only replaced with the previous in the future because the mode developed successful.

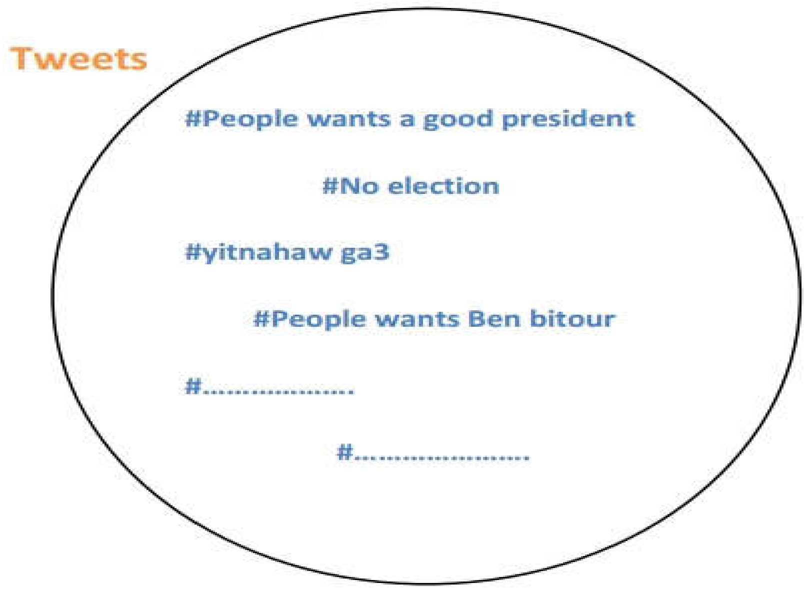


Figure 1: illustrate the problem

2 Related works

The proposed approach is in the field of sentiment analysis. To determine whether a piece of text expresses a positive or negative sentiment, three main approaches are commonly used: the lexicon-based approach, the machine learning-based approach and approach that combine lexicon-based methods and machine learning algorithms. (WEDJDANE WEDJDANE NAHILI, KHALED REZEG Sentiment Analysis. Business Intelligence et Big Data - EDA 2018.). The lexicon-based approach (Hu and Liu 2004, Kim and Hovy 2004)[12] defines the sentiment polarity using a dictionary to capture the set opinion words in the document or the sentence. The machine learning-based approach typically trains sentimental classifiers using features such as unigrams or bigrams (Pang et al., 2002; Khanaferov et al. 2014; Aliaksei et al. 2015; Debjyoti et al., 2017).[13]The approach that combine lexicon-based methods and machine learning algorithms combines natural language processing techniques with Naive Bayes to classify users data .Most techniques use some form of supervised learning by applying different learning techniques such as Naive Bayes, Maximum Entropy and

Support Vector Machines. Although machine learning methods proved high accuracy in previous work, unfortunately, manual labeling for training examples are needed for every application domain. [14][15][16]

Year	Application	Domain/Authors
2011	**L**	Movie reviews. Taboada et al. (2011)
2013	**ML**	Topic trending. Ostrawski et al. Ostrowski (2013)
2014	**ML**	Healthcare. Khanaferov et al. Khanaferov (2014)
2015	**ML**	Text document. Aliaksei Severyn et al. Aliaksei Severyn (2015)
	L	Movie, hotel and product reviews. Vitares et al.
2016	**L**	Movie reviews. Cambria et al. (2016)
2017	**ML**	US Elections 2016. Debjyoti Paul et al. Debjyoti Paul (2017)
2018	**Naïve bayes+ML**	Wedjdane Nahili, Khaled Rezeg.Sentiment Analysis. BusinessInt elligence et Big Data - EDA 2018.

Tab. 1 – *Previous related works*

L: lexicon-based, ML: machine Learning[17] [18][19] [20] [21]

The issue in analyzing social media unstructured data in this manner lacks accuracy and gives a generalized idea. In order to make it more specific and more accuracy, we suggest proposing an approach of analysis of the feelings of the messages of social networks by applying the various stages of the NLP through the use of the deep learning. to perform a location-based sentiment analysis on tweets since they are a reliable source of information, mainly because people tweet about anything and everything, either it is about political discussion , buying new products or reviewing them.

3 Design

In order to validate our methods of study of the syntactic evolution of the subjects and to make emerge threads of Tweets, we set up an experimental framework having the architecture of the (Figure 2) which represents the big parts of our contribution. Each phase has an input, an output and a series of parameters that we will vary to find the best values. This framework is un-cut into the collection, tweeting, and cleaning phases. For the purpose of grouping tweets we have transformed them into digital vectors before applying deep learning approach. The results of these methods are then analyzed and visualized

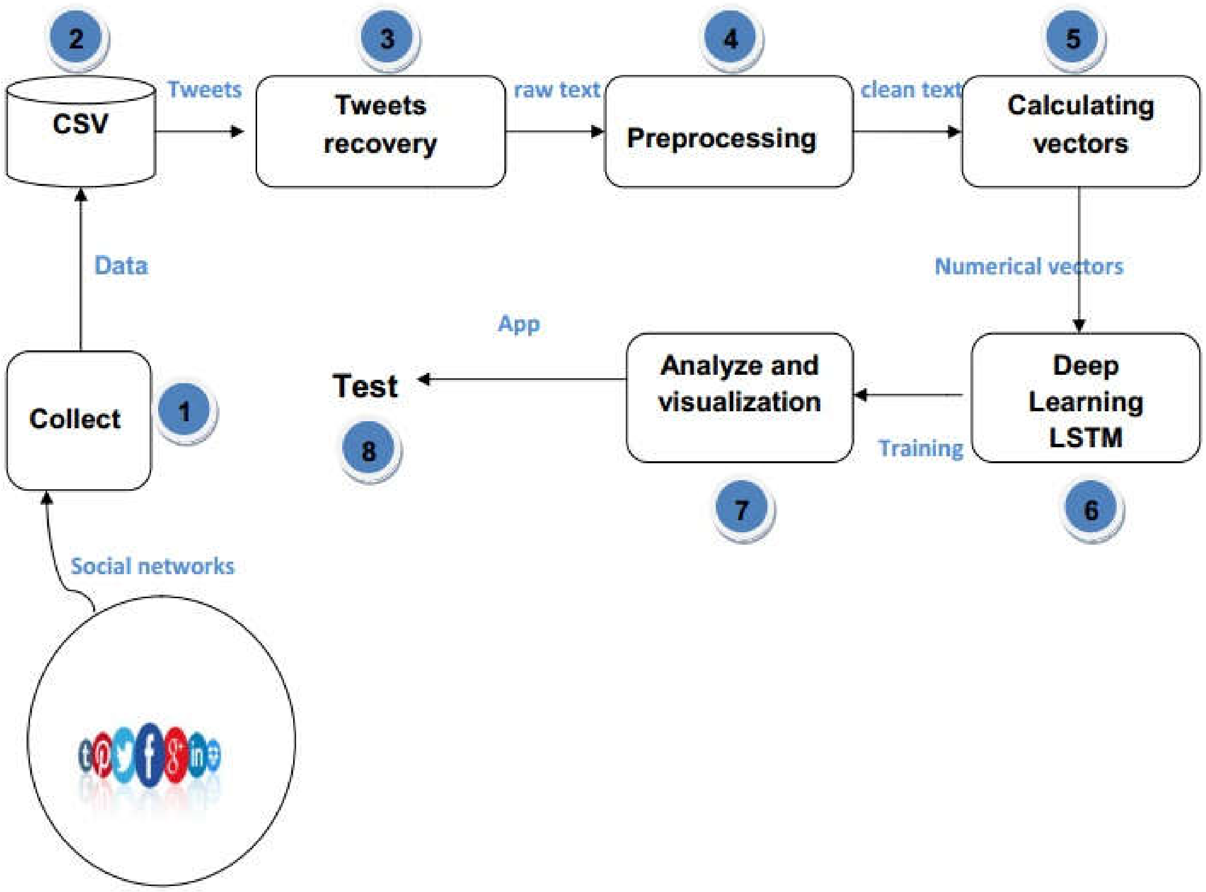


Figure 2: Architecture design

In what follows, we will specify each phase of the architecture.

3 1 Phase 01: Collect

In this project we will work on a specified collections of tweets, We initially relied on data collection ourselves and the difficulty of completing it we focused in a specified data set to complete our project until our data set be ready .

We started with the collection ‘Twitter US Airline Sentiment’ described below:

A sentiment analysis job about the problems of each major U.S. airline. Twitter data was scraped from February of 2015 and contributors were asked to first classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as “late flight” or “rude service”).

Description of the collection, this collection contains:

- 30000 tweets
- Collected in February 12, 2015
- Size 03,5 Mo
- Labeled : positive(01) , negative (0)

For each Tweet collected we have:

- attributes that concern the tweet itself:
    - the content (of the text).
    - hashtags if they exist.
    - the date of publication.
    - the textual properties of words and hashtags.
    - URL of the tweet.
- attributes relating to the geographical position:
    - the geographic coordinates of the user when publishing tweets if they exist.
    - the country.
- Attributes that relate to the user profile:
    - the date of account creation.
    - The pseudonym.
    - The user name.
    - The time zone.
    - The number of previous Tweets

3 2 Phase 02: Retrieve posts

First, we started by retrieving text from tweets only, with a given number of tweets set to 20000. In this case, the output of the first step is a CSV file that contains all the tweets selected in plain text. And before going to the pre-processing stage, we had to extract each word by itself, tweet by tweet to be able to apply the cleaning with the regular expressions and others. For that we use a step of transformation of the tweets towards words, which takes as input the list of text tweets and returns a list of words.
The set of modifiable parameters during this phase:

- the collection name.
- The type of post (Twitter).
- the type of data (Text, Hashtags, geographic coordinates, user names).
- The temporal dimension (between two dates, per time slot).
- The number of tweets

3 4 Phase 04: Preprocessing

Analyzing tweets will be a major challenge for us and other researchers working on the Twitter feed. This stems from the nature of these published posts, where the tweets are totally different from other documents such as newspaper articles, official speeches, web pages, etc. For cleaning methods, we are inspired by the works evoked in the state of the art that we have completed with original methods. Among the features that can be found in the tweets we quote :

- users use a language that is not formal, and a mix between jargon, abbreviations and multiple_languages_in_the_same_tweet.
- Tweets are full of spelling errors, lexical errors and syntactic errors.
- The existence of links and identifiers complicates the analysis operation. For these reasons it was decided to apply a minimum number of cleanings and filtering on tweets during_this_phase,_which_consists_of:
- remove links: in this context, a link in a tweet does not have a semantic weight for the subject followed (unless we analyze the link if there are more details possibly in the page pointed_by_the_link).

- delete identifiers: we have implemented this type of optional cleaning because a Twitter identifier is not a keyword that concerns the subject. At the same time, it can give more similarity between two tweets of the same subject who are often close and posted by the same person.

- remove hashtags: One of the possible types of cleanup is the removal of hashtags to work_only_on_text_in_natural_language.

- update the hashtags: this operation consists in putting a sharp (#) in front of each word that is identical to a hashtag (the word becomes a hashtag too) for an interest to give more context to a tweet, which probably improves the identification of the_subject_to_which_it_belongs_this_tweet.

- removal of non-hashtags words (this cleaning can be combined with the previous one)

- to solve the problem of uppercase and lowercase we decided to apply two types of treatments, the first is the conversion of all the words in lowercase. This transformation poses a problem for abbreviations_identical_to_existing_words_that_will_be_changed.

- the second treatment is the conversion of the first character in uppercase and the rest in lowercase (like the case of the first word of a sentence in French) of all words that contain only alphabetic characters and are not all in capital letters, and this prevents a word like 'La' (at the beginning of the sentence) and 'la' from being considered as different words, but at the same time we guarantee not to convert the abbreviations to lowercase

- Another type of treatment is to clean words by removing quotation marks, commas, periods, exclamation points, question marks, and so on.

-delete words that contain only a single character.

-for cleaning the numbers it is not easy

All these types of cleaning are not always applied at once, but for an experiment we can specify the set of cleanings to be applied, to study the impact of the variation of pre-treatments on the final result. this step is the list of lists of raw words extracted from each tweet, and the output is the list of lists of cleaned words. If cleaning removes all words from a tweet, the tweet will be deleted.

3 5 Phase 05: Calculating vectors

This phase takes as input:

- the list of unique words existing in the csv file to be analyzed after preprocessing.
- The list of the word lists of each tweet after the preprocessing.
- The type of method to apply.

This step provides as output digital vectors corresponding to each tweet which is a statistical representation of the different words of each tweet selected according to the formula used.

3 6 Phase 06: Deep learning 'LSTM' training

We'll be using a deep learning approach called in model word by a Long Short-term Memory (LSTM). LSTMs overcome this by having an extra recurrent state called a cell, which can be thought of as the "memory" of the LSTM and the use use multiple gates which control the flow of information into and out of the memory. Fig3

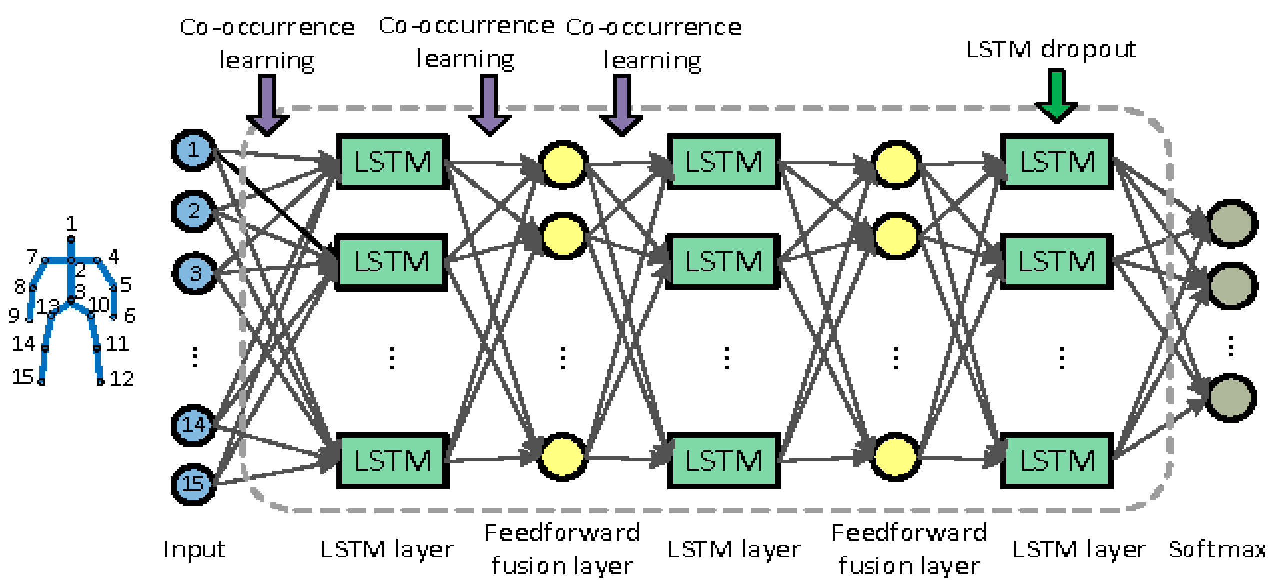


Figure 3: LSTM

3 7 Phase 07: Analyze and visualization

In this part we have carried out a statistical analysis on our learning and model training flow. For simple goals for analyze and discuss the accuracy of our model, we will use a python based library to make this task easier called matplotlib .

3 7 Phase 08: Test

Arrive to this phase mean that our model is ready and we have close the learning and training steps. Now in this phase we create a simple application 'interface user-machine' to test the accuracy of our work .

4 Conclusion

Until now we described and explained our methodology in theoretical mode, in the next section we will detail the implementation of each part of our application with the mention of used tools and techniques.

5 Proposed approach

In this work, we propose a real time localized sentence-level Twitter sentiment analysis approach using Deep learning, where we analyze a dataset (tweets) expressing reactions and opinions. Our analysis is done as follows: we extracted Tweets contain a lot a noise so preliminary processing is needed beforehand then the tweets are stored. In our approach, we apply a natural language processing (NLP) tasks . Although this tasks give a pre-trained data that let our data ready to have a numerical format which allow our deep learning model train our data, to improve accuracy we do the following: We first extract tweets that contain opinionated terms (sentences and tokens) in a csv file classified in positive ,negative and neutral sentences. Secondly, we use NLP to pre-process our data to assign tweets in one and clean format. After that, we use Keras a python package that allows to vectorize our dataset, by turning each text into either a sequence of integers based on word count, based on tf-idf. we take insight of the results using Matplotlib to derive high-quality information that was not that obvious beforehand and visually display the results. Our proposed approach is illustrated in Fig.4.

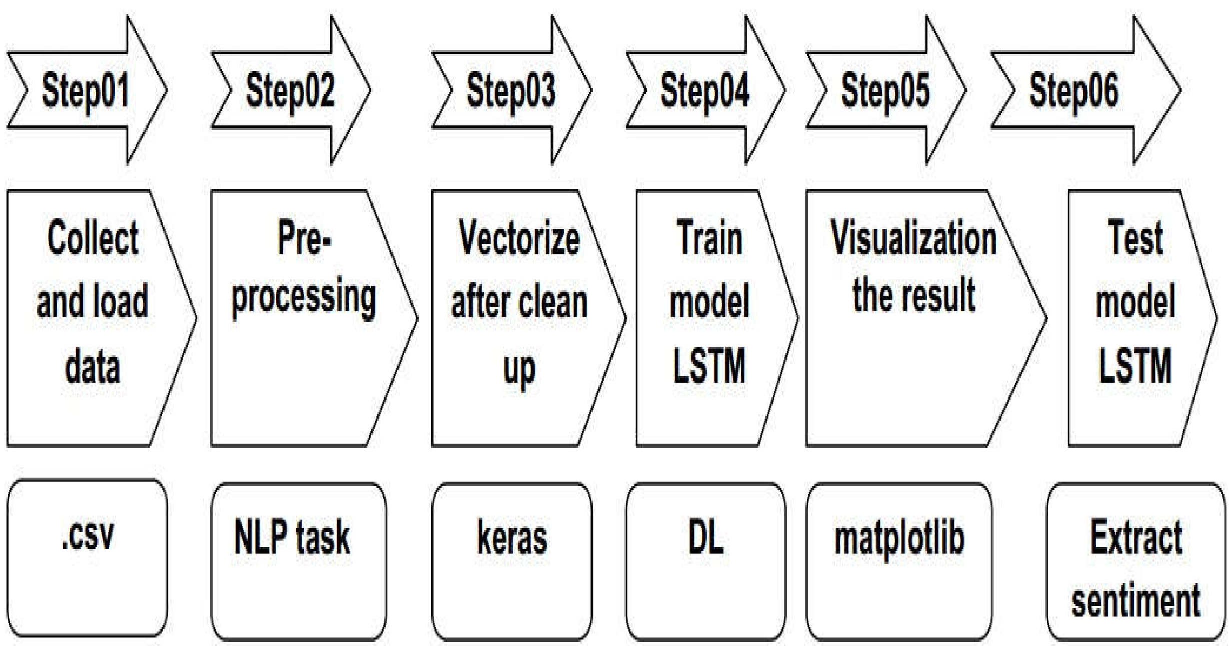


Figure 4: Proposed approach

6 Keras

We will use a Python library named Keras to help us with our work and summarize many of our work which allows us to reach our goal as soon as possible with the desired results. Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.*

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.[9]

7 Proposed approach steps

7 1 Collect and load data

This is the most important phase in our sentiment analysis model. It is divided into two steps: the first step is a way of collect posts(sentences). The second step, we labeling each post (sentence) with a score depending on how positive or negative it can be and finally we trained a deep learning model to train on our pre-labeled dataset because it is easy to build and particularly functional for large data sets, and it is known to outperform even highly sophisticated classification methods. Since Deep learning is all about using high scalable algorithms to build models which are complex and difficult for machine learning algorithms like naïve bayes , logistic regression, vector support machine e.t.c, to model. In our work, were supposed to use the data of the political events currently taking place in Algeria such as popular movement, elections and strikes. However, due to the difficulty of collecting sufficient data for their use in our work, we used to use another open source data to teach our modal accordingly Until we prepare and collect the necessary data according to our project and replace them only with previous data and re-education only. So we chose a dataset of about 20,000 tweets to be trained from the kaggle website which is a data set of airline client' s posts(tweets) .[22]

	B	K	
1	***Sentiment***	***text***	***tweet_coord***
2	neutral	@VirginAmerica What @dhepburn said.	
3	positive	@VirginAmerica plus you've added commercials to the experience... tacky.	
4	neutral	@VirginAmerica I didn't today... Must mean I need to take another trip!	
5	negative	@VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces &	
6	negative	@VirginAmerica and it's a really big bad thing about it	
7	negative	@VirginAm	
8	positive	@VirginAmerica yes, nearly every time I fly VX this â€œear wormâ€ wonâ€™t go away :)	
9	neutral	@VirginAmerica Really missed a prime opportunity for Men Without Hats parody, there. https:/	
10	positive	@virginamerica Well, I didn'tâ€¦but NOW I DO! :-D	
11	positive	@VirginAmerica it was amazing, and arrived an hour early. You're too good to me.	
12	neutral	@VirginAmerica did you know that suicide is the second leading cause of death among teens 10	
13	positive	@VirginAmerica I <3 pretty graphics. so much better than minimal iconography. :D	
14	positive	@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia &am	

Figure 5: a sample of our data set

7 2 NLP tasks (pre-processing)

It is often necessary to normalize the text for any NLP task. Since the tweets are often represented in a cryptic and informal way, systematic preprocessing of tweets is required to enhance the accuracy of the sentiment analyzer. The tweets are preprocessed to extract all valid terms that have immense significance to determine the polarity. At this level, we analyze the extracted tweets relevant to the prefixed keyword inserted in the search query. This analysis is done according to two following phases:

—**Tokenization:** is the process of converting the sequence of 280 characters composing the tweets, into a sequence of words (tokens), in our case only the tweets containing matching terms with the predefined set of opinion words are kept for further processing. In simple terms, tokenization means dividing a given text into smaller and meaningful elements like sentences and words. For example, let us assume we have the following review on a airline trip : #**This is such a great deal! Already thinking about my 2nd trip**.. After tokenization, the sentence could take the following form **"This" , "such" ,"great", "deal" ,"Already" ,"thinking" ,"about" , "nd", "trip" .**

— **Filtering:** the dataset obtained obviously contains a lot of non relevant data (noise). Therefore, very basic and rudimentary cleanup needs to be performed. Arbitrary characters and other useless information such as punctuation, emoticons substitution, stop words, special characters and text normalization are applied using regular expressions, finally links/URL, hashtags(#) and words that start with '@' character were removed since we found no significance in our scoring approach. When these two steps are completed, the processed tweets are then stored in a comma separated values (csv) file for further processing.[23]

In this step we use keras that help us to make the nlp task in few codes and best result , for an example :

```
from keras.preprocessing.text import Tokenizer
```

```
keras.preprocessing.text.one_hot(text, n, filters='!"#$%&()*+,-
./:;<=>?@[\\]^_`{|}~\t\n', lower=True, split=' ')
```

```
data['text'] = data['text'].apply(lambda x: x.lower())

data['text'] = data['text'].apply((lambda x: re.sub('[^a-zA-z0-9\s]','',x)))
```

7 3 Vectorize dataset

Text is one of the most widespread form of sequence data. It can be understood either as a sequence of characters, or a sequence of words, albeit it is most common to work at the level of words. The deep learning sequence processing models that we'll introduce can use text to produce a basic form of natural language understanding, sufficient for applications ranging from document classification, sentiment analysis, author identification, or even question answering (in a constrained context). Keep in mind throughout this article that none of the deep learning models you see truly "understands" text in a human sense, rather, these models are able to map the statistical structure of written language, which is sufficient to solve many simple textual tasks. Deep learning for natural language processing is pattern recognition applied to words, sentences, and paragraphs, in much the same way that computer vision is pattern recognition applied to pixels. Like all other neural networks, deep learning models don't take as input raw text: they only work with numeric tensors. Vectorizing text is the process of transforming text into numeric tensors. This can be done in multiple ways:

- By segmenting text into characters, and transforming each character into a vector.
- By extracting "N-grams" of words or characters, and transforming each N-gram into a vector. "N-grams" are overlapping groups of multiple consecutive words or characters.

Collectively, the different units into which you can break down text (words, characters or N-grams) are called "tokens", and breaking down text into such tokens is called "tokenization". All text vectorization processes consist in applying some tokenization scheme, then associating numeric vectors with the generated tokens. These vectors, packed into sequence tensors, are what get fed into deep neural networks. There are multiple ways to associate a vector to a token. In this section we will present two major ones: **one-hot encoding of tokens**, and **token embeddings** (typically used exclusively for words, and called "**word embeddings**"). In the remainder of this section, we will explain these techniques and show concretely how to use them to go from raw text to a Numpy tensor that you can send to a Keras network.[24]

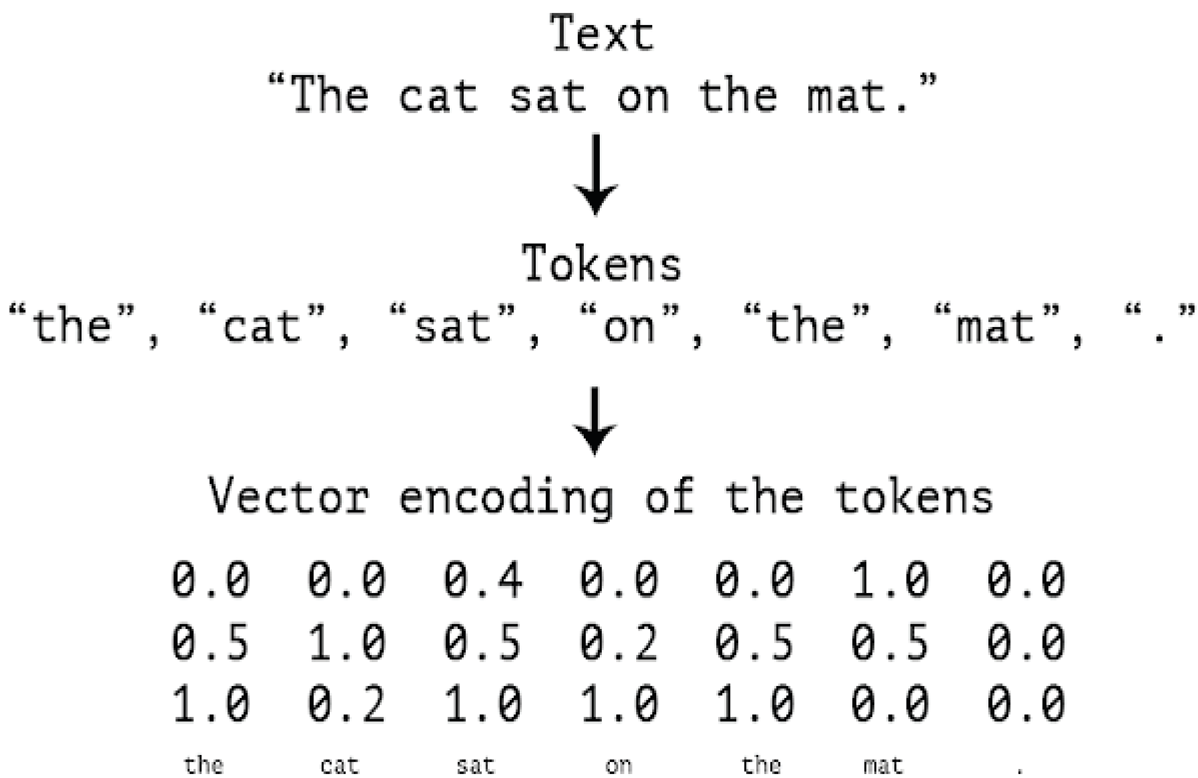


Figure 6: From text to tokens to vectors

In this step, vectorization step we used Keras for this operation that help us to vectorize the data set rapidly and in few instructions, for an example :

```
from keras.preprocessing.text import Tokenizer
# define 5 documents
docs = ['Well done!',
     'Good work',
     'Great effort',
     'nice trip ,
     'Excellent trip !']
# create the tokenizer
t = Tokenizer()
# fit the tokenizer on the documents
t.fit_on_texts(docs)
# summarize what was learned
print(t.word_counts)
print(t.document_count)
print(t.word_index)
```

```
print(t.word_docs)
# integer encode documents
encoded_docs = t.texts_to_matrix(docs, mode='count')
print(encoded_docs)
```

Result :

```
OrderedDict([('well', 1), ('done', 1), ('good', 1), ('work', 2), ('great', 1),
('effort', 1), ('nice', 1), ('excellent', 1)])
5
{'work': 1, 'effort': 6, 'done': 3, 'great': 5, 'good': 4, 'excellent': 8, 'well': 2,
'nice': 7}
{'work': 2, 'effort': 1, 'done': 1, 'well': 1, 'good': 1, 'great': 1, 'excellent': 1,
'nice': 1}
[[ 0.  0.  1.  1.  0.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  1.  1.  0.  0.]
 [ 0.  1.  0.  0.  0.  0.  0.  1.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  1.]]
```

After the success of vectorize our data set, that allow us to build our model and train it by the previous vectorized data set .

7 4 Deep learning step

We'll be using a different RNN architecture called a Long Short-Term Memory (LSTM). LSTMs overcome this by having an extra recurrent state called a cell, which can be thought of as the "memory" of the LSTM and the use use multiple gates which control the flow of information into and out of the memory.

Our model will be present as neural network started by an input and a two hidden layers ended by an output that have two binary classes, in our case positive or negative

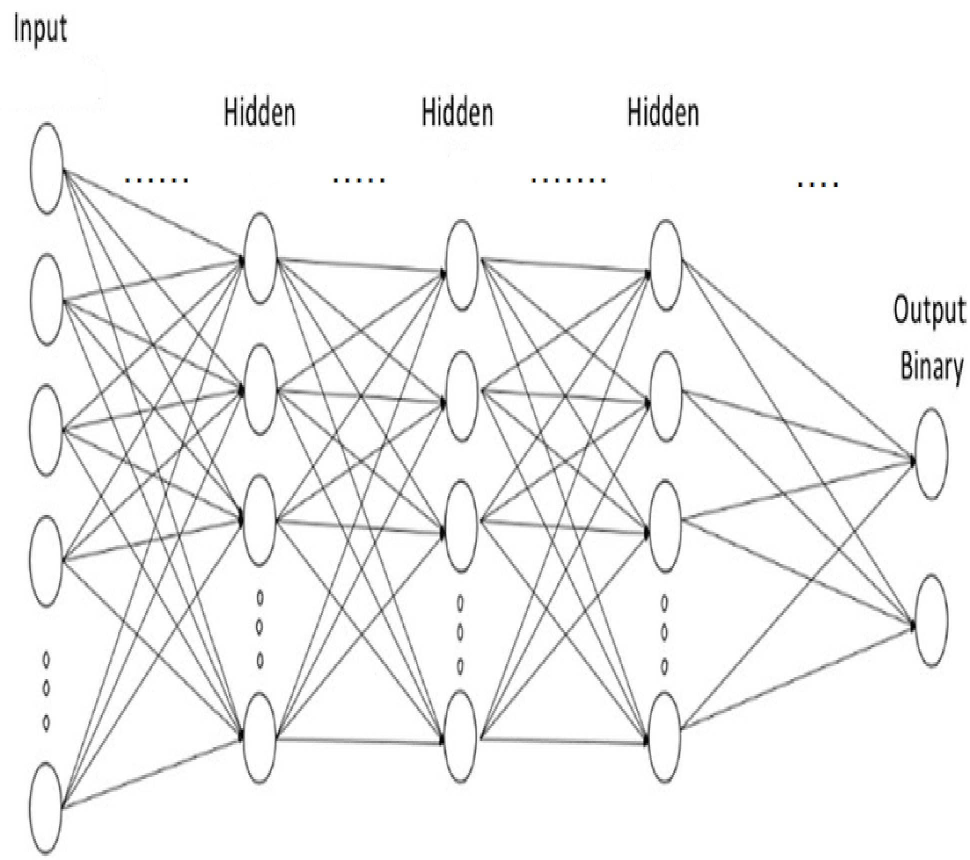


Figure 7: an abstract two classes model

Below shows an example sentence, with the RNN predicting zero, which indicates a negative sentiment. The RNN is shown in orange and the linear layer shown in silver. Note that we use the same RNN for every word, i.e. it has the same parameters. The initial hidden state h0 is a tensor initialized to all zeros. Fig8

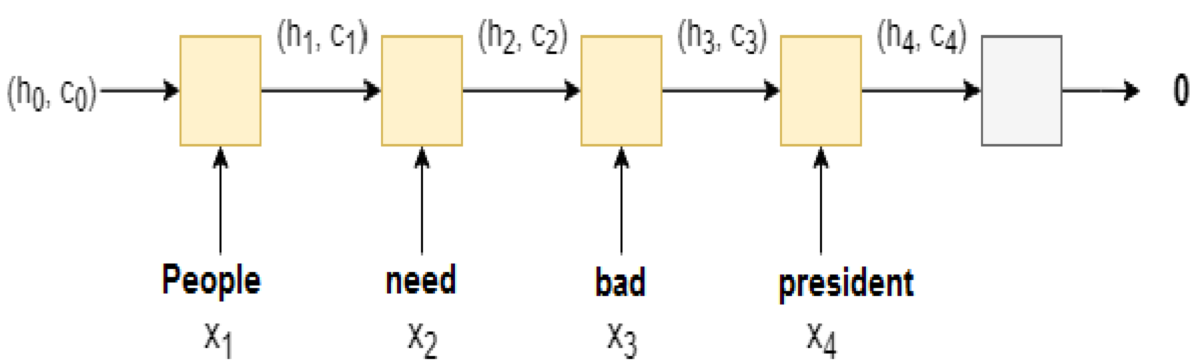


Figure 8: LSTM model

Long Short-Term Memory networks—usually just called "*LSTMs*"—are a special kind of *RNN*, capable of learning long-term dependencies. *LSTMs* don't have a fundamentally different architecture from *RNNs*, but they incorporate additional components.

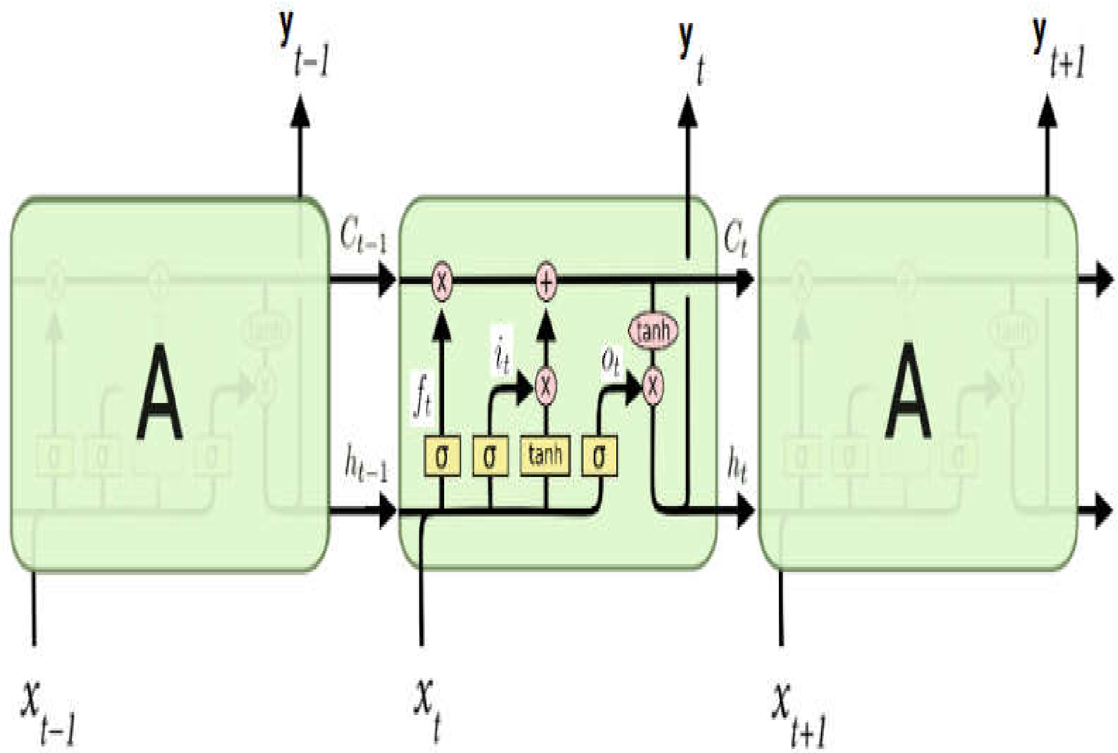


Figure 9: LSTM detail

The key to *LSTMs* is the cell state ***C(t)***, the horizontal line running through the top of the diagram. A cell state is an additional way to store memory, besides just only using the hidden state ***h(t)***. However, ***C(t)*** makes it possible that *LSTMs* can work with much longer sequences in opposite to vanilla *RNNs*.

Furthermore, *LSTMs* have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. An *LSTM* has three of these gates, to protect and control the cell state.

- **Forget Gate**: After getting the hidden state ***h(t-1)*** of the previous input ***x(t-1)***, Forget gate helps us to make decisions about what must be removed from ***h(t-1)*** state and thus keeping only relevant stuff.
- **Input Gate:** In the input gate, we decide to add new stuff from the present input ***x(t)*** to our present cell state ***C(t)***.
- **Output Gate:** The output gate as the name suggests, decides what to output from the current cell state ***C(t)*** to the next ***C(t+1)***. For the language model example, since it just saw a subject, it might want to output information relevant to a verb, in case that's what is coming next. For example, it might output whether the subject is singular or plural so that we know what form a verb should be conjugated into if that's what follows next.[25]

Behind each of these states are separate neural networks. As you can imagine this makes LSTMs quite complex. At this point, I won't go much more into the detail about LSTMs.[10]

7 4 1 Model building

In our sentiment classification task the data consists of both the raw string of the review and the sentiment, either "positive" or "negative".

To build our LSTM model, The steps to do this are:

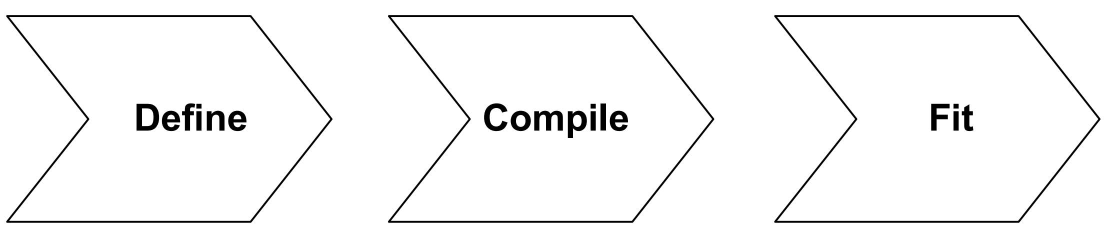


Figure 10: build LSTM

7 4 1 1 Define Model

For building a deep learning model, we need to define the layers (Input, Hidden, and Output). Here, we will go ahead with a LSTM model, which means that we will define layers automatically using Keras . Also, we will be going ahead with a fully connected network.

We will focus on **defining the input layer**. This can be specified while creating the first layer with the *input dim* argument and setting it to **33** for the **130** independent variables.

Next, **define the number of hidden layer(s)** along with the number of neurons and activation functions. The right number can be achieved by going through multiple iterations. Higher the number, more complex is your model. To start with, I'm using two hidden layers. One has 130 neurons and the other has 200 with the same activation function - "**softmax**".

Softmax activation function

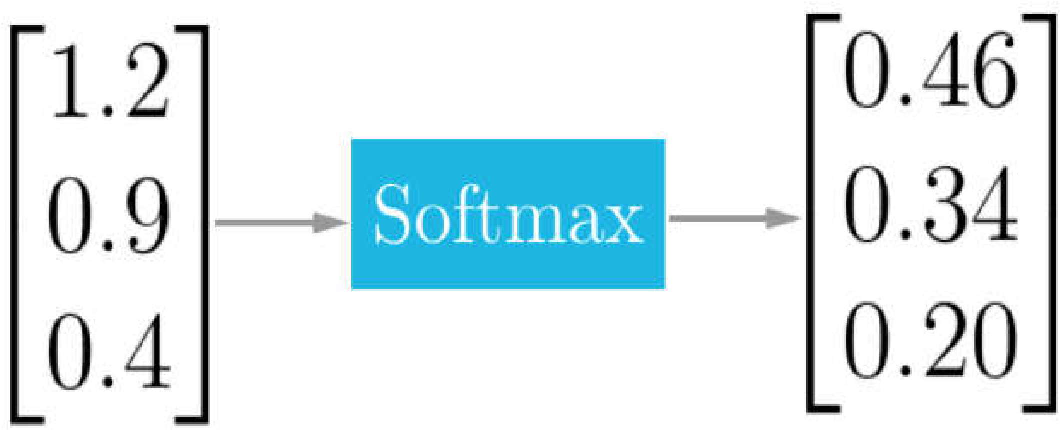


Figure 11:softmax function

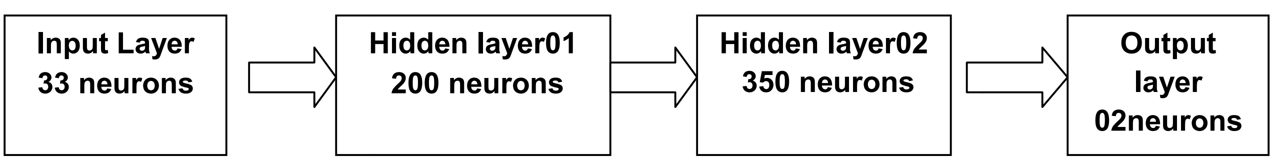
The softmax function squashes the outputs of each unit to be between 0 and 1, just like a sigmoid function. But it also divides each output such that the total sum of the outputs is equal to 1 (check it on the figure above).The output of the softmax function is equivalent to a categorical probability distribution, it tells you the probability that any of the classes are true. Mathematically the softmax function is shown below, where z is a vector of the inputs to the output layer (if you have 10 output units, then there are 10 elements in z). And again, j indexes the output units, so j = 1, 2, ..., K.[26]

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Finally, we need to **define the output layer** with 2 neurons to predict the purchase amount. The problem in hand is a regression challenge so we can go ahead with a linear transformation at the output layer. Therefore, there is no need to mention any activation function .

By using keras we can resume a lot of work in some of python codes as bellow :

```
embed_dim = 200

lstm_out = 350

model = Sequential()

model.add(Embedding(max_fatures, embed_dim,input_length = X.shape[1]))

model.add(Dropout(0.5))

model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))

model.add(Dense(2,activation='softmax'))

model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy'])
```



I show in figure figure12 a Keras description of our LSTM model:

```
"C:\Program Files\Python36\python.exe" C:/Users/KMB/PycharmProjects/A
Using TensorFlow backend.
The positive tweets :  4726
The negative tweets :  18356

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 33, 200)           1400000
_________________________________________________________________
dropout_1 (Dropout)          (None, 33, 200)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 350)               771400
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 702
=================================================================
Total params: 2,172,102
Trainable params: 2,172,102
Non-trainable params: 0
_________________________________________________________________
```

Figure 12: keras lstm model description

7 4 1 2 Compile Model

At this stage, we will configure the model for training. We will set the optimizer to change the weights and biases, and the loss function and metric to evaluate the model's performance. Here, we will use "**adam**" as the optimizer, "**accuracy** " as the loss metric. Depending on the type of problem we are solving, we can change our loss and metrics. For binary classification, we use "**categorical_crossentropy**" as a loss function.

Keras code :

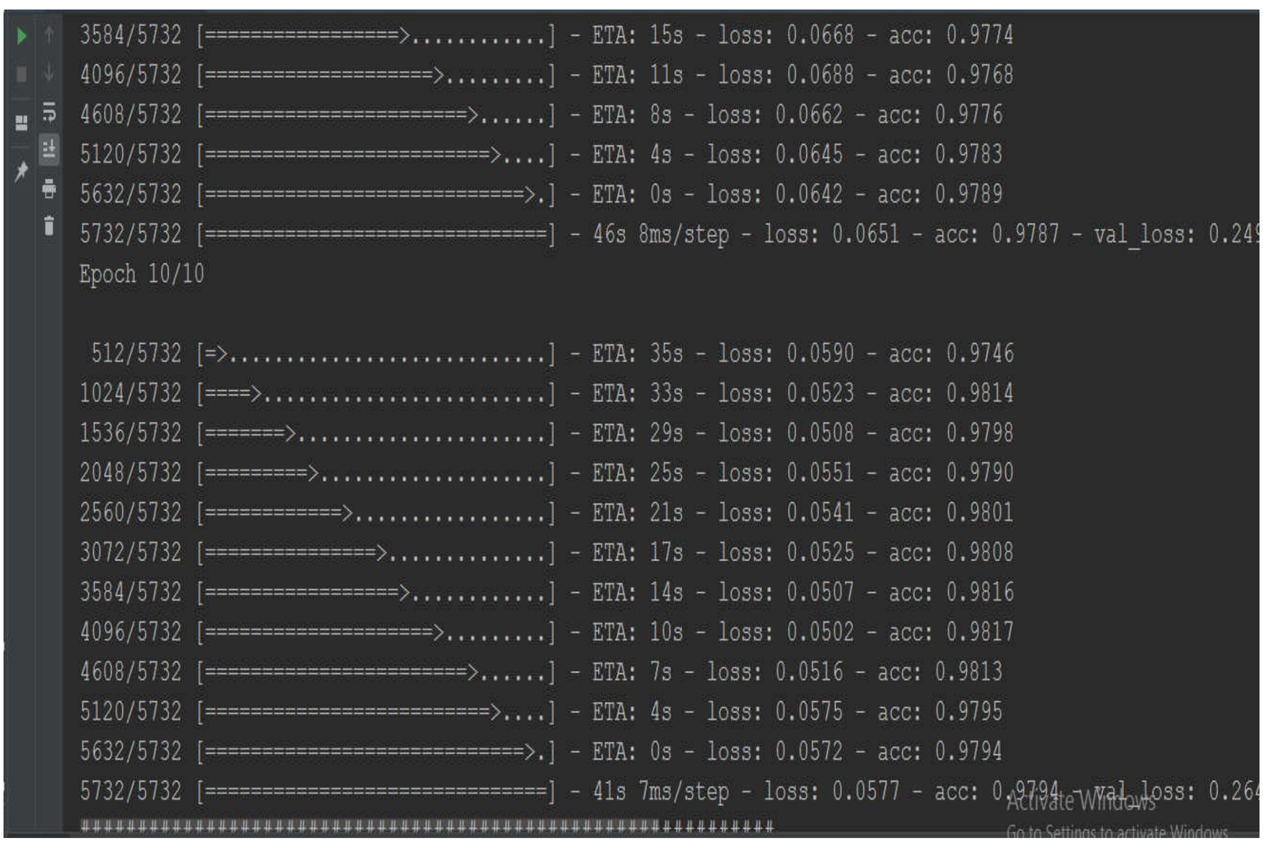
```
model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy'])
```

7 4 1 3 Fit model

The final step of model building is fitting the model on the training dataset. We need to provide both independent and dependent variables along with the number of training iterations, i.e. epochs. Here, we have taken **10 epochs.**

```
history = model.fit(partial_X_train,

                    partial_Y_train,

                    epochs =10,

                    batch_size=batch_size,

                    validation_data=(X_val, Y_val))
```

show of keras training :



```
3584/5732 [=================>...........] - ETA: 15s - loss: 0.0668 - acc: 0.9774
4096/5732 [====================>.........] - ETA: 11s - loss: 0.0688 - acc: 0.9768
4608/5732 [======================>.......] - ETA: 8s - loss: 0.0662 - acc: 0.9776
5120/5732 [=========================>....] - ETA: 4s - loss: 0.0645 - acc: 0.9783
5632/5732 [============================>.] - ETA: 0s - loss: 0.0642 - acc: 0.9789
5732/5732 [==============================] - 46s 8ms/step - loss: 0.0651 - acc: 0.9787 - val_loss: 0.24
Epoch 10/10

 512/5732 [=>............................] - ETA: 35s - loss: 0.0590 - acc: 0.9746
1024/5732 [====>.........................] - ETA: 33s - loss: 0.0523 - acc: 0.9814
1536/5732 [=======>......................] - ETA: 29s - loss: 0.0508 - acc: 0.9798
2048/5732 [=========>....................] - ETA: 25s - loss: 0.0551 - acc: 0.9790
2560/5732 [============>.................] - ETA: 21s - loss: 0.0541 - acc: 0.9801
3072/5732 [===============>..............] - ETA: 17s - loss: 0.0525 - acc: 0.9808
3584/5732 [=================>............] - ETA: 14s - loss: 0.0507 - acc: 0.9816
4096/5732 [====================>.........] - ETA: 10s - loss: 0.0502 - acc: 0.9817
4608/5732 [======================>.......] - ETA: 7s - loss: 0.0516 - acc: 0.9813
5120/5732 [=========================>....] - ETA: 4s - loss: 0.0575 - acc: 0.9795
5632/5732 [============================>.] - ETA: 0s - loss: 0.0572 - acc: 0.9794
5732/5732 [==============================] - 41s 7ms/step - loss: 0.0577 - acc: 0.9794 - val_loss: 0.264
```

Figure 13: Fit training

7 5 Visualization the result

We have built our LSTM model of deep learning techniques using Keras, and we got a learned deep learning model after 20 Epochs, now we use a python library to visualize the results

7 5 1 Matplotlib

a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab.[27]

Use of the library

```
import matplotlib.pyplot as plt
```

- Training and validation loss

Python code

Next python code let to show the plot of our training and validation Loss

```
plt.plot(epochs, loss, 'bo', label='*Training*

loss',color='r')

plt.plot(epochs, val_loss, 'b', label='*Validation* loss',color='r')

plt.title('(*Training* and *validation* loss)')

plt.xlabel('Epochs')

plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
plt.clf()
```

Result

The code execution gives the following plot of the training and validation **loss** after the training of our learned LSTM model .figure14

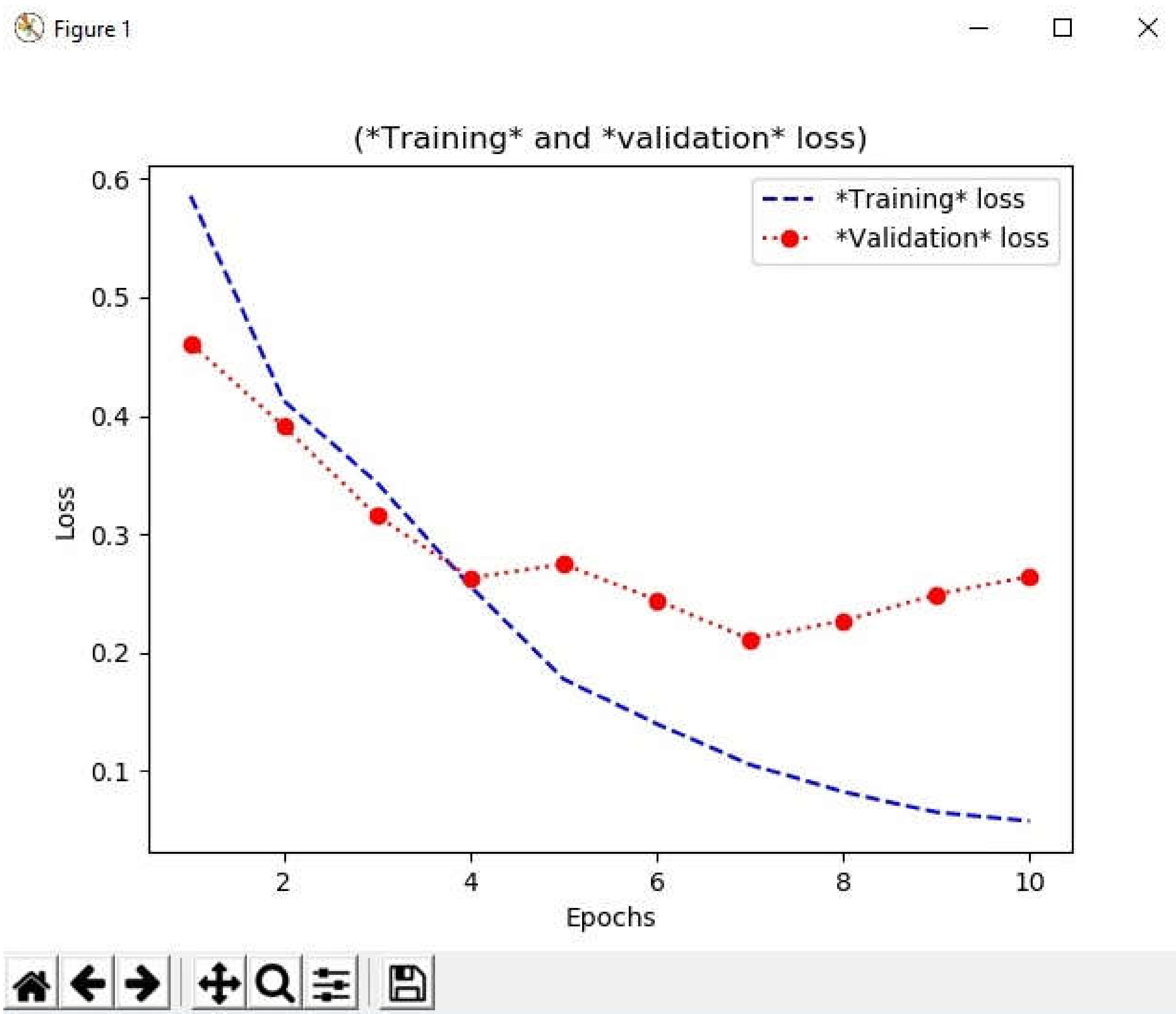


Figure 14: training and validation loss

- Training and validation accuracy

Python code

Next python code let to show the plot of our training and validation **accuracy**.

```
acc = history.history['acc']

val_acc = history.history['val_acc']

plt.plot(epochs, acc, 'r--', label='*Training acc*')

plt.plot(epochs, val_acc, 'b:o', label='*Validation acc*')

plt.title('(*Training* and *validation* accuracy)')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.show()
```

The code execution gives the following plot of the training and validation **accuracy** after the training of our learned LSTM model .figure15

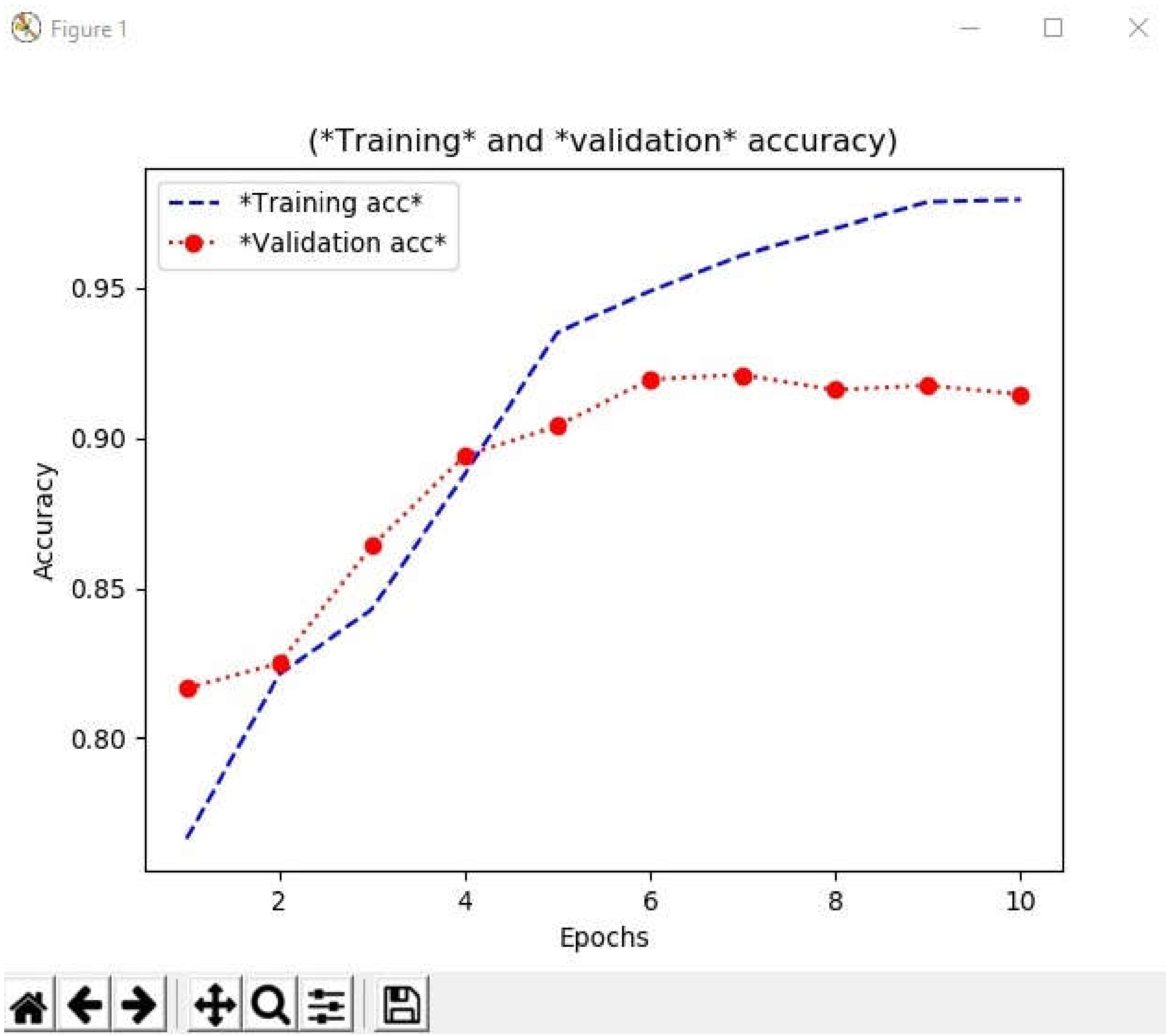


Figure 15: training and validation accuracy

7 5 2 Discussion

I trained this deep learning model 'LSTM' I used my personal computer environment without any kind of GPU because Nvidia graphics card missed in my laptop. I used a laptop that characterized by: **CPU intel core i5-3230 206 GHZ with a Ram of 08 GB .**

As we can see from the graphs:" **loss**" and "**accuracy**", the **04th** epoch is the best before the network start to over fitting the data.

7 5 3 Training validation

After evaluating our LSTM model and finalizing the model parameters, we can go ahead with the prediction on the test data. Below is the code to do this :

```
pos_cnt, neg_cnt, pos_correct, neg_correct = 0, 0, 0, 0

for x in range(len(X_val)):

   result = model.predict(X_val[x].reshape(1, X_test.shape[1]),
batch_size=1, verbose=2)[0]

   if np.argmax(result) == np.argmax(Y_val[x]):

      if np.argmax(Y_val[x]) == 0:

         neg_correct += 1

      else:

         pos_correct += 1

   if np.argmax(Y_val[x]) == 0:

      neg_cnt += 1

   else:

      pos_cnt += 1
```

Result

After the validation process of our LSTM model we get this prediction from our learned model

- **Accuracy of positives** = 71.25603864734299**%**
- **Accuracy of negatives** = 96.72131147540983**%**

```
############################################################
Accuracy of positives 71.25603864734299 %###
------------------------------------------------------------
Accuracy of negatives  96.72131147540983 %##
############################################################

Process finished with exit code 0
                                                    Activate Windows
```

Figure 16: result

Discussion

The accuracy of prediction with our deep learning model 'LSTM' arrived to 71% for positives tweets and to 97% for negatives tweets. Despite the accuracy increase with this kind of our deep learning model, I am sure the accuracy can be improved, and this is the goal of our perspectives works.

7 6 Test: Obtain the Sentiment

We have seen how to preprocess the data and how to feed in the LSTM network. Now, let's discuss how we can finally get the sentiment of a given post.

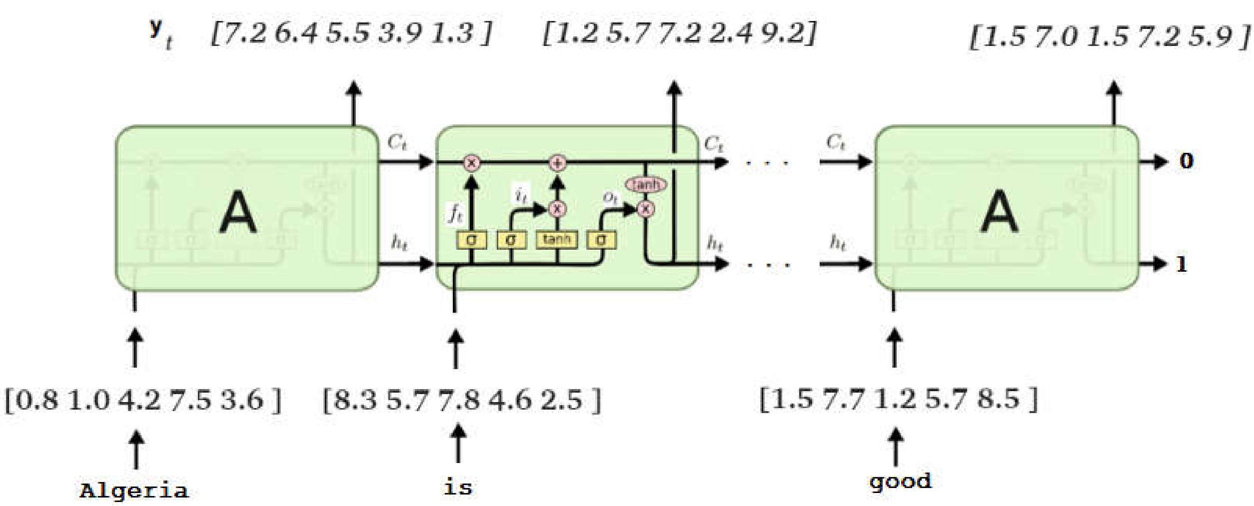


Figure 17: LSTM network

For each time step **t**, the LSTM network receives an input vector **x(t)** which results in the output vector **y(t)**. This process is repeated until **x(n), n** being the number of words in the review. Let's say **n**=20 words. Until **x(n)** the *LSTM* network produced **y(n)** output vectors. Each of these 20 vectors represents something, but not the sentiment we are looking for**.** Rather the vectors y are an encoded representation of features of the review that (according to the neural network) will be important in determining the sentiment.

y(8) represents the features the neural networks recognized for the first 8 words of the review. **y(20)**, on the other hand, represents the features for the whole review. Although it is sufficient to use only the last output vector **y(20)** in practice, I have found that it leads to more accurate results if we use all vectors **y(0)—y(20)** for determining of the sentiment. This can be achieved by computing the mean value over all vectors. Let's call this mean value vector **y_mean.**

Finally, the feature representation of the review that is encoded in **y_mean** can be used to classify the review into the categories of being positive or being negative. In order to do so, it is required to add a final classification layer, which is nothing else than the dot product between **y_mean** and another weight matrix **W**.

Test application

We test our model by ask a user for a post then our model will test it and give a result if positive or negative

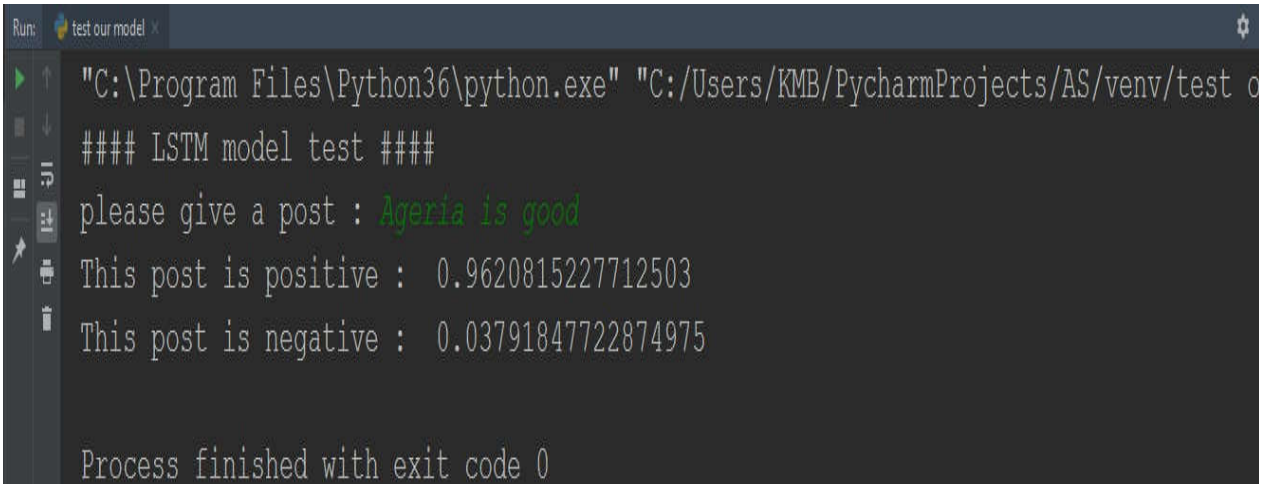


Figure 18: result of test

7 7 Conclusion

Sentiment analysis has provided more importance to the mass of opinions that leads to better decision and good prediction. Researchers can take advantage of this understanding and therefore, they will position themselves as they want it to be, and improve their decision-making process. In this work, we introduced a sentiment analysis methodology for automatically extracting and analyzing opinion from tweets using a deep learning approach. To do so, we have used a free data set of voyagers' opinion of trips about **30000** tweets are extracted. For storing tweets we used a CSV format. Due to the fact that in our and methodology location was only used for user segmentation for one and academic language 'English' , further perspectives would be working on a combination of academic language and Dialect language and more for emotion sentiments . Another perspective is improving our model for more accuracy sentiment analysis for better NLP filtering techniques for sarcasm detection, seeing that it is frequently used in natural language, in addition to expanding the terms domain in order to reflect real-life complex scenarios.

General conclusion

Social networks may contains opinions, facts, thoughts and other information that required to be analyzed using sentiment analysis tools.Sentiment analysis in social networks is the process of retrieving opinions, information, feeling of users or people in general about products, events,…etc to extract and make best decision for future achievements and learn from mistakes .In this work, we introduced a sentiment analysis methodology based in deep learning approach for automatically extracting and analyzing opinion from tweets. To do so, we have considered a specified data set that at latest 20000 tweets. For storing tweets we used a CSV format to apply our deep learning approach to improve the result's accuracy of sentiment analysis.

References

[1]: https://en.wikipedia.org/wiki/Artificial_intelligence

[2]: https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f

[3]: https://www.edureka.co/blog/machine-learning-algorithms/

[4]: https://www.mathworks.com/discovery/deep-learning.html

[5]: https://www.mathworks.com/discovery/deep-learning.html

[5]: https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32

[6] : https://monkeylearn.com/sentiment-analysis/

[7]: https://mc.ai/understanding-deep-learning-dnn-rnn-lstm-cnn-and-r-cnn/

[8]: https://freecontent.manning.com/deep-learning-for-text/

[9]: https://keras.io/

[10]: https://towardsdatascience.com/sentiment-analysis-with-deep-learning-62d4d0166ef6

[11]: Ahmed REMADNA , Classification des posts sur des réseaux sociaux, INSA-Lyon .

[12]: Wedjdane Nahili∗, Khaled Rezeg∗∗, Lyna Miloudi∗∗∗Towards Better Decision-making with Twitter Sentiment Analysis , LINFI Laboratory, Computer Science, Mohamed Khider University, Biskra

[13]: Aliaksei Severyn, A. M. (2015). Combining lexicon-based and learning-based methods for twitter sentiment analysis. HP Laboratories HPL-2011-89.

[14]: Khanaferov, D., L. C. W. T. (2014). Social network data mining using natural language processing and density based clustering. In IEEE International Conference on Semantic Computing (ICSC) (pp. 250,151).

[15]: Lei Zhang, Riddhiman Ghosh, M. D. M. H. B. L. (2015). Twitter sentiment analysis
with deep convolutional neural networks. *In proceedings Severyn 2015 Twitter SA,SIGIR*.

[16]: .Ostrowski, D. (2013). Semantic social network analysis for trend identification. *In IEEE Sixth International Conference on Semantic Computing (pp 215,222)*.

[17]: Taboada, M., B. J.-T. M. V. K. a. S. M. (2010). Lexicon-based methods for sentiment
analysis. *Journal of Computational Linguists 2010*.

[18]: Tan, S., W. Y. and X. Cheng. *Journal of Computational Linguists, SIGIR2008*.
Wang B, L. M. *https://cs224d.stanford.edu/reports/WangBo.pdf* .

[19]: Hu M, L. B. (2004). Mining and summarizing customer reviews. *In KDD'04, pp168-177*

[20]: Lei Zhang, Riddhiman Ghosh, M. D. M. H. B. L. (2015). Twitter sentiment analysis
with deep convolutional neural networks. *In proceedings Severyn 2015 Twitter SA,*
SIGIR.

[21]: Ostrowski, D. (2013). Semantic social network analysis for trend identification. *In*
IEEE Sixth International Conference on Semantic Computing (pp 215,222)

[22]: https://www.kaggle.com/datasets

[23]: Wedjdane Nahili∗, Khaled Rezeg∗∗, Lyna Miloudi∗∗∗Towards Better Decision-making with Twitter Sentiment Analysis , LINFI Laboratory, Computer Science, Mohamed Khider University, Biskra

[24]: https://machinelearningmastery.com/prepare-text-data-deep-learning-keras/

[25]: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[26]: https://towardsdatascience.com/deep-learning-concepts-part-1-ea0b14b234c8

[27]: https://matplotlib.org/3.1.1/contents.html