Learning Meters of Arabic Poems with Deep Learning

A Thesis Presented to the Faculty
of
Nile University

In Partial Fulfillment
of the Requirements for the Degree
of Master of SCIENCE

By
Moustafa Alaa Mohamed
July 2018

CERTIFICATION OF APPROVAL

Learning Meters of Arabic Poems with
Deep Learning

By
Moustafa Alaa Mohamed

_____ _____

Dr. Samhaa El-Beltagy (Chair) Date
Professor of Electrical and Computer Engineering

_____ _____

Dr. Waleed Yousef Date
Professor of Computer Science

ACKNOWLEDGMENT

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Thesis Outline

The coming chapters are arranged as follows:

- Chapter 1: Presents some basic introduction and background knowledge as regards the Arabic Poem and its definitions. Also, it contains details about the Arabic language and some feature used during our work.
- Chapter 2: Introduces the essential pre-processing steps, and the justification for their need. Pre-processing steps are data extraction, data cleansing and data format.
- Chapter 3: introduces the data encoding techniques used and the effect of each one. Also, it contains some comparisons between the three techniques used.
- Chapter 4: presents the model's details and how we chose the model and the architecture and hyper-parameters details.
- Chapter 5: Results and discussion.
- Chapter 6: Conclusion and future work

ABSTRACT

People can easily determine whether a piece of writing is a poem or prose, but only specialists can determine the class of poem.

In this thesis, We built a model that can classify poems according to their meters; a forward step towards machine understanding of Arabic language.

A number of different deep learning models are proposed for poem meter classification. As poems are sequence data, then recurrent neural networks are suitable for the task. We have trained three variants of them, LSTM, GRU with different architectures and hyper-parameters. Because meters are a sequence of characters, then we have encoded the input text at the character-level, so that we preserve the information provided by the letters succession directly fed to then models. Besides, We introduce a comparative study on the difference between binary and one-hot encoding regarding their effect on the learning curve. We also introduce a new encoding technique called *Two-Hot* which merges the advantages of both *Binary* and *One-Hot* techniques.

Artificial Intelligence currently works to do the human tasks such as our problem here. Our target in this thesis is to achieve the human accuracy which will make it easy for anyone to know the meter for any poem without referring to the language experts or to study the whole field to achieve it.

In this thesis, We will explain how to use the deep learning to classify the Arabic poem to classes. Also, explain in details the feature of Arabic poem and how to deal with this features. Besides, We explain how can anyone work with Arabic text encoding with a dynamic way to encode the text at the character level and deal with the Arabic text feature example the *Tashkeel*.

Chapter 1

INTRODUCTION

Arabic is the fifth most widely spoken language[1]. It is written from right to left. Its alphabet consists of 28 primary letters, and there are 8 more derived letters from the basic ones, so the total count of Arabic characters is 36 characters. The writing system is cursive; hence, most letters join to the letter that comes after them, a few letters remain disjoint.

1.1 Arabic Poetry

Arabic poetry(الشعر العربى) is the earliest form of Arabic literature. It dates back to the Sixth century. Poets have written poems without knowing exactly what rules which make a collection of words a poem. People recognize poetry by nature, but only talented ones can write poems. This was the case until *Al-Farahidi* (718 786 CE) has analyzed the Arabic poetry, then he came up with that the succession of consonants and vowels produce patterns or *meters*, which make the music of poetry. He has counted them fifteen meters. After that, a student of *Al-Farahidi* has added one more meter to make them sixteen. Arabs call meters بحور which means "*seas*". The study of Arabic Poems classification is named **Al-Arud (العَروض)**. It takes too much time for anyone to be an expert in this field.

1.2 Deep Learning

Deep Learning also named Deep Neural Network is part of Machine Learning algorithms. Deep Learning is trying to simulate the human brain into Neural dependency. Using Deep Learning, we can achieve better learning results from the data. Deep Neural Network needs a huge amount of data to achieve the expected learning curve and results. It also needs a massive amount of computation to build the networks which are based on an artificial neural network. We used the Recurrent Neural Network (RNN) to work on the Arabic Text which shown its ability to achieve outstanding performance over the text problem data. We also used LSTM to solve the long dependency issue in RNN. We will go deep into the Background section (add deep learning section reference).

[1] *according to the 20th edition of Ethnologue, 2017*

1.3 Thesis Objectives

In this study, we work to classify the poem and utilize the latest technologies check the class of poem. We also worked to achieve near human expert results which make our work is a breakthrough in the field concerning the results compared to the current achieved results. Figure 1.1 shows the steps.,

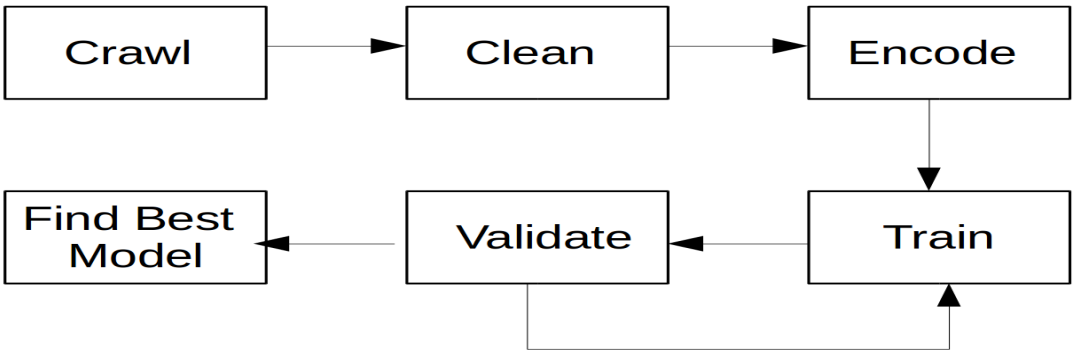


Figure 1.1: Thesis Working Steps.

- Crawling the data from the available sources with labeling.
- Clean and transform the data.
- Encode the data into a way to be input to the model to work on it. We used many encoding methods and compared each of them.
- Train the RNN model into the cleaned data.
- Validate and test the model.
- Enhance the model.

Chapter 2

BACKGROUND

Each Arabic letter represents a consonant, which means that short vowels are not represented by the 36 characters, for this reason, the need of *diacritics* rises. *Diacritics* are symbols that comes after a letter to state the short vowel accompanied by that letter. There are four diacritics ◌َ ◌ُ ◌ِ ◌ْ which represent the following short vowels /*a*/, /*u*/, /*i*/ and *no-vowel* respectively, their names are *fat-ha, dam-ma, kas-ra and sukun* respectively. The first three symbols are called *harakat*. Table 2.1 shows the 4 diacritics on a letter.

Diacritics	*without*	*fat-ha*	*kas-ra*	*dam-ma*	*sukun*
Shape	د	دَ	دِ	دُ	دْ

Table 2.1: *Diacritics on the letter* د

There are two more sub-diacritics made up of the basic four to represent two cases:

Definition 1 ***Shadaa***
to indicate the letter is doubled. Any letter with shaddah (ّ) the letter should be duplicated: first letter with a constant (sukoon) and second letter with a vowel (haraka) [1]; Table 2.2 shows the dal with shadda and the original letters.

Diacritics	*letter with Shadda*	*letters without shadaa*
Shape	دَّ	دْدَ

Table 2.2: *Shadaa diacritics on the letter* د

Definition 2 ***Tanween***
is doubling the short vowel, and can convert Tanween fathah, Tanween dhammah or Tanween kasrah by replacing it with the appropriate vowel (dhammah, fathah or kasrah) then add the Noon letter with constant to the end of the word [1]. Table 2.3 shows the difference between the original letter and the letter with Tanween

Diacritics	*letter with tanween*	*letters without tanween*
Tanween Fat-ha	دً	دَ+نْ
Tanween Dam-ma	دٌ	دُ+نْ
Tanween Kas-ra	دٍ	دِ+نْ

Table 2.3: *Tanween diacritics on the letter* د

Arabs pronounce the sound */n/* accompanied *sukun* at the end the indefinite words, that sound corresponds to this letter نْ, it is called *noon-sakinah*, however, it is just a phone, it is not a part of the indefinite word, if a word comes as a definite word, no additional sound is added. Since it is not an essential sound, it is not written as a letter, but it is written as *tanween* ◌ً ◌ٌ ◌ٍ. *Tanween* states the sound *noon-sakinah*, but as you have noticed, there are 3 *tanween* symbols, this because *tanween* is added as a diacritic over the last letter of the indefinite word, one of the 3 harakat*harakat* accompanies the last letter, the last letter's *harakah* needs to be stated in addition to the sound *noon-sakinah*, so *tanween* is doubling the last letter's *haraka*, this way the last letter's *haraka* is preserved in addition to stating the sound *noon-sakinah*; for example, رَجُلُ + نْ is written رَجُلٌ and رَجُلِ + نْ is written رَجُلٍ.

Those two definition, Definition 1 and Definition 2 will help us to reduce the dimension of the letter's feature vector as we will see in *preparing data* section.

Diacritics makes short vowels clearer, but they are not necessary. Moreover, a phrase without full diacritics or with just some on some letters is right linguistically, so it is allowed to drop them from the text.

In Unicode, Arabic diacritics are standalone symbols, each of them has its own unicode. This is in contrast to the Latin diacritics; e.g., in the set *{ê, é, è, ë, , , }*, each combination of the letter *e* and a diacritic is represented by one unicode.

2.1 Arabic Arud Science

Definition 3 ***Arud***
In Arabic Arud natively has many meanings (the way, the direction, the light clouds and Mecca and Madinah[1] *[4]. Arud is the science which studies The Arabic Poem meters and the rules which confirm if the Poem is sound meters & broken meters. If we need to su*

The Author of this science is *Al-Farahidi* (718 786 CE) has analyzed the Arabic poetry; then he came up with that the succession of consonants and vowels produce patterns or *meters*, which make the music of poetry. He was one of the famous people who know The melodies and the musical parts of speech. He has counted them fifteen meters. After that, a student of *Al-Farahidi* has added one more meter to make them sixteen. Arabs call meters بحور which means "*seas*" Poets have written poems without knowing exactly what rules which make a collection of words a poem.

The Reasons which makes *Al-Farahidi* put this science is

- Protect the Arabic Poems from the broken meters.
- Distinguish between the original Arabic Poem and the non-poem or from the prose.
- Make the rules clear and easy for anyone who needs to write a poem.

Some people said that the one-day Al-Farahidi was walking into the metal-market and he was said some of the poems and for some reasons the knock of the metals matched the musical sound of the poem he was saying then he got an idea to explore the Arud of the poems.

There are many reason for this science name

- It named Arud because some people said he put this science in Arud place العَروض *with fat-ha, not with dam-ma such as the science name* العُروض between Mecca and Al-Ta'if[4].
- Arud in Arabic is noun come from verb يعرض which means here to be assessed. They said because of Any poem should be assessed by Al-Arud science so, it named Al-Arud [3].

[1] *Mecca and Madinah are two cities in Saudi Arabia.*

2.1.1 Al-Farahidi and Pattern Recognition

This subsection is our opinion in Al-Farahidi and his method he followed during working on Arabic Poem Classifications.

1. Al-Farahidi thought there is a pattern for every collection of the poem by chance; however, He scientifically worked into this problem. He started analyzing the poem and add every group with the same tafa'il to the same class.
2. He analyzed the outliers and the particular case from every class and added it to his model.
3. He revised the Bohor and get the cases and generalize his case to be fit into all Poems.
4. His student once he found some Poems which weren't fit into any model to be a model for a new class.

The best essential point which made us admired by Al-Farahidi is his way of research and his passion for getting an indeed succession model. Also, his model is general and followed all the steps currently any Data scientist follows to explore new pattern. Some people state that He died when he was thinking about the problem he hit a wall which made trouble for him. His die story shows that he was thinking in profoundly about this problem. One of the most interest thing I found during this research is how he found this pattern and Al-Farahidis way to find a new thing.

2.1.2 Feet Representation

A meter is an ordered sequence of feet. Feet are the basic units of meters; there are ten of them.

Definition 4 ***Feet***
A Foot consists of a sequence of ***Sukun*** *(Consonants) represented as (0) and* ***Harakah*** *(Vowels) (/). Traditionally, feet are represented by mnemonic words called tafail* تفاعيل.

Feets consists of three parts (Reasons أسباب, Wedge وتد, Breaks فواصل).

- **Reasons (أسباب)**: It has two types
  1. **Light (سبب خفيف)** which happens when we have the first letter is harakah and the second is sukun (/0) example (هَبْ, لَمْ).
  2. **Heavy (سبب ثقيل)** which happens when we have two harakah letter (//) example (لَكَ, بِكَ).
- **Wedge (وتد)**: It has two types
  1. **Combined Wedge (وتد مجموع)** which happens when we have two harakah letters followed by sukun (//0) example (مَشَى, عَلَى).
  2. **Separated Wedge (وتد مفروق)** which happens when we have two harakah and in between a sukun letter (/0/) example (مُنْذُ, مِصْرُ).
- **Breaks (فواصل)**: It has two types
  1. **Small Break (فاصلة صغرى)** which happens when we have three harakah letters followed by a sukun letter (///0) example (ذَهَبُوا, سُفُناً).
  2. **Big Break (فاصلة كبرى)** which happens when we have four harakah letters followed by a sukun letter (////0) example (جَعَلَهُمْ).[2]

[2] *Some of Arab linguistic scientist assume the small Breaks as a combination between big reason and small reason. Same for the Big Breaks assumed to be a combination between Big reason and Combined Wedge. So, they didn't assume we have three types of feet it is only pure two and any other feets constructed from this two. In this thesis we assume there are three feets .*

2.1.2.1 Rules for Arabic Letters Representation

Arabic Arud has one general rule in the poem representation which is we represent only the letters which is (spoken) not the written which means the letters with phonatics not the written. We have give the below rules as a results of the general rule.

- Any letter with *harakah* represented as (/).
- Any letter with *sukun* represented as (0).
- Any letter with shaddah represented by two letters the first one will be *sukun* and the second letter will be *harakah* represented as (0/) example (مُحَمَّد) will be (//0//0).
- Any letter with tanween represented by two letters the first one is *haraka* (/) and the second is *sukun*.
- Alef without hamze (همزة الوصل) and Wow Algmaa are not represented example (وُاعلَموا) will be (/0//0)
- If we have a letter which is not written but (spoken) so, we will represent it example (هذا) it include Alef but not written (هاذا) the representation will be (/0/0).
- If we have *Meem Aljamaa* with harakah so, it represented with *Mad* example (هُمُ) will be (//0) .
- *Alef Mad* (آ) will be two letters *Alef with harakah* and *Alef with sukun* example (آدَمُ) will be (/0//).
- if the verse ended with *harkah* we will add *sukun* to it.

Example: (note: the below representation first line is simliar the second one but with Arud language style).

أرَاكَ عَصِيَّ الدّمعِ شِيمَتُكَ الصّبرُ، *** أما للهوى نهيٌ عليكَ ولا أمرُ ؟
أرَاكَ عَصَيْيَ دّمعِ شِيمَتُكَ صّبرُو، *** أما للهوى نهينْ عليكَ ولا أمرو ؟

2.1.3 Arabic Poetry Feets

Arabic poetry feets has ten tafa'il تفاعيل (scansion) any peom constructed from these feets. They are eight from writing (syntax) perspective, But it ten in the rules.

#	Feet	Scansion	Construction
1	فَعُولُنْ	0/0//	combined wedge (فعو) and small reason (لن)
2	مَفاعِيلُنْ	0/0/0//	combined wedge (مفا) and two light reasons (عي) (لن)
3	مُفَاعَلَتُنْ	0///0//	combined wedge (مفا), heavy reason (عل) and light reason (تن)
4	فَاعِلاتُنْ	0/0//0/	light reason (فا), combined wedge (علا) and light reason (تن)
5	فَاعِ لاتُنْ	0/0//0/	Separated wedge (فاع) and two light reason (لا)(تن) [3]
6	فَاعِلُنْ	0//0/	light reason (فا) and combined wedge (علن)
7	مُتَفَاعِلُنْ	0//0///	heavy reason (مت), light reason (فا) and combined wedge (علن)
8	مَفْعُولاَت	0//0///	two light reason (مف)(عو) and separated wedge (لات)
9	مُسْتَفْعِلُنْ	0//0/0/	two light reason (مس)(تف) and combination wedge (علن)
10	مُسْتَفْعِ لُنْ	0//0/0/	light reason (مس), separated wedge (تفع) and light reason (لن) [4]

Table 2.4: The ten feet of the Arabic meters.

[3] *We separated the letters (ع) and (لا) in (فاع لاتن) to show that this part is separated wedge and distinguish between this feet and (فاع لاتن) which contains combined wedge .*

[4] *We separated the letters (ع) and (ل) in (مستفع لن) to show that it ends with a separated wedge and distinguish between this feet and (مستفعلن) which contains combined wedge*

Definition 5 ***Meter***
Poetic meters define the basic rhythm of the poem. Each meter is described by a set of ordered feet which can be represented as ordered sets of consonants and vowels [2].

ولد الهدى فالكائنات ضياء *** وفم الزمان تبسم وثناء انشاء
الروح والملأ الملائك حوله *** للدين والدنيا به بشراء

Definition 6 ***Arabic Verse***
refers to "poetry" as contrasted to prose. Where the common unit of a verse is based on meter or rhyme, the common unit of prose is purely grammatical, such as a sentence or paragraph [5]*. A verse know as Bayt in Arabic* بيت

Definition 7 ***Shatr***
A verse consists of two halves, each of them is called shatr and carries the full meter. We will use the term shatr to refer to a verse's half; whether the right or the left half.

Definition 8 ***Poem***
is a set of verses has the same meter and rhyme.

[5] *https://en.wikipedia.org/wiki/Verse_(poetry).*

2.1.4 Arabic Poetry Meters

2.1.4.1 Al-Taweel الطويل

Why it named Al-Taweel? *Al-Taweel is named Al-Taweel for two reasons; first, It is the longest meter between all meters. Second, It starts with Wedge then Reasons and Wedge is longer than Reasons. So, it named Al-Taweel. We need here to note later in the encoding section we will pad all other meters by zeros to make it all the same length. Example if the max Bayt is 82 so, any Bayt less than 82 will be padded by zeros to have the same length.[3]*

tafa'il

فعلون مفاعيلن فعولن مفاعلن *** فعولن مفاعيلن فعولن مفاعيلن

Example:

اذا جادَ أقوامٌ بِمالٍ رأيْتَهُمْ *** يَجودونَ بالأرواحِ مِنْهُم بلا بُخل
0/0//0/0/0// 0/0// 0/0//0/0/0// 0/0// *** 0//0// 0/0//0/0/0// 0/0// 0/0//0/0/0// 0/0//
فعلون مفاعيلن فعولن مفاعلن *** فعولن مفاعيلن فعولن مفاعيلن

2.1.4.2 Al-Madeed المديد

Why it named Al-Madeed?

Al-Madeed is named because of the reasons الأسباب *is represented in all its seven parts of tafa'il, One in the first part and the other in the second part. So, it named Madeed* مديداً *[3]*

tafa'il

فاعلاتن فاعلِن فاعلاتن *** فاعلاتن فاعلن فاعلاتن

Example:

يَا لِبَكْرٍ أَنْشِرُوا لِي كُلَيْباً يَا لِبَكْرٍ أَيْنَ أَيْنَ الْفِرَارُ
0/0//0/ 0//0/ 0/0//0/ *** 0/0//0/ 0//0/ 0/0//0/ 0/0//0/
فاعلاتن فاعلِن فاعلاتن *** فاعلاتن فاعلن فاعلاتن

2.1.4.3 Al-Baseet البسيط

Why it named Al-Baseet? Al-Baseet there is a different idea behind this name

- The Reasons الأسباب expanded into it is tafa'il. So, We will find at the beginning of every part two reasons so; it named Al-Baseet.
- The other reasons which may be the more logic are the harkat الحركات expanded in its tafa'il.[3]

tafa'il

مستفعلن فعلن مستفعلن فاعلن *** مستفعلن فاعلن مستفعلن فاعلن

Example:

لَيْسَ الجَمَال بأَثْواب تُزَيِّنُنَا *** إن الجمال جمال العلم والأدب
0/// 0//0/0/ 0/// 0//0/0/ *** 0/// 0//0/0/ 0/// 0//0/0/
مستفعلن فعلن مستفعلن فاعلن *** مستفعلن فاعلن مستفعلن فاعلن

2.1.4.4 Al-Wafer الوافر

Why it named Al-Wafer? Al-Wafer there are different ideas behind this name

- There is much harakat in its parts because there is no tafa'il has part includes harakat more than the word مفاعلتن.
- There are many parts to its base.[3]

tafa'il

مفاعلتن مفاعلتن مفاعلتن *** مفاعلتن مفاعلتن مفاعلتن

Example:

إذَا بَلَـغَ الفِطَـامَ لَنَا صَبِـيٌّ *** تَخِـرُّ لَهُ الجَبَـابِرُ سَاجِدِيْنَـا
0//0/ 0///0// 0///0// *** 0//0/ 0///0// 0///0//
مفاعلتن مفاعلتن فعولن *** مفاعلتن مفاعلتن فعولن

2.1.4.5 Al-Kamel الكامل

Why it named Al-Kamel? *Al-Kamel is named because its harakat is fully integrated and it is 30 harakah which is not similar to any other Bahr has this numbers of harakat. However, Al-wafer has much harakat in its parts but not the same number as Al-Kamel. Al-Wafer has the harakat but it not overwritten into its source but Al-Kamel it is written into its source* أصله*. So, Al-Kamel in Arabic named due to it is more integrated* أكمل *than Al-Wafer [3].*

tafa'il

متفاعلن متفاعلن متفاعلن ⁂ متفاعلن متفاعلن متفاعلن

Example:

وَلَقَدْ شَفَا نَفْسِي وَأَبْرَأَ سُقْمَهَا ⁂ قِيلُ الفَوَارِسِ وَيكَ عَنْتَرَ أَقْدِمِ
0//0/// 0//0/// 0//0/0/ ⁂ 0//0/// 0//0/0/ 0//0///
متفاعلن مستفعلن متفاعلن ⁂ مستفعلن متفاعلن متفاعلن

2.1.4.6 Al-Hazaj الهزج

Why it named Al-Hazaj? *Al-Hazaj is named because of the sound reverberation in its parts. Al-Hazaj in Arabic is sound frequency. So, due to the sound reverberation, it named AL-Hazaj. Also, Because every part ends for two reasons so, it made some of sound like a piece of music to be Al-Hazaj [3].*

tafa'il

مفاعيلن مفاعيلن ⁂ مفاعيلن مفاعيلن

Example:

أَيَا مَنْ لاَمَ فِي الحُبِّ ⁂ وَلَمْ يَعْلَمْ جَوى قَلبِي
0/0/0// 0/0/0// ⁂ 0/0/0// 0/0/0//
مفاعيلن مفاعيلن ⁂ مفاعيلن مفاعيلن

2.1.4.7 Al-Rejz الرجز

Why it named Al-Rejz? *Al-Rejz is named Al-Rezj because it constructed from three parts. If there is an animal and someone pull this animal by one leg and the animal walk into three legs in Arabic named Rejz* رجزاً*, this is the reason it named Rejz because of it has three parts similar than animal pulled by one leg and walk into three legs. Also, in Arabic, if we have a camel when it stands up*

has some disturbances due to it sick or has any issue it named Rejz Camel رجز جمل*, and in Arabic disturbance* إضطراب *means Rejz for example [3].*

tafa'il

مستفعلن مستفعلن مستفعلن ⁂ مستفعلن مستفعلن مستفعل

Example:

دار لسلمى إذ سليمى جارة ⁂ قفر ترى آيانها مثل الزّبر
0//0/0/ 0//0/0/ 0//0/0/ ⁂ 0//0/0/ 0//0/0/ 0//0/0/
مستفعلن مستفعلن مستفعلن ⁂ مستفعلن مستفعلن مستفعلن

2.1.4.8 Al-Raml الرمل

Why it named Al-Raml? *Al-Raml is named Al-Raml because Al-Raml is a type of the singing which constructed from this Bahr. Another reason is the Wedges is appeared in between the Reasons so; it named Al-Raml due to the diversity between the Reasons and the wedges inside the tafa'il [3].*

tafa'il

فاعلاتن فاعلاتن فاعلاتن ⁂ فاعلاتن فاعلاتن فاعلاتن

Example:

انما الدنيا غرور كلها ⁂ مثل لمغ اش□ل في أرض الفقار
0/0//0/ 0/0//0/ 0/0//0/ ⁂ 0//0/ 0/0//0/ 0/0//0/
فاعلاتن فاعلاتن فاعلن ⁂ فاعلاتن فاعلاتن فاعلاتن

2.1.4.9 Al-Sarea السريع

Why it named Al-Sarea? *Al-Sarea is named Al-Sarea in Arabic Al-Sarea means the fastest. It named these because its speed in teste* الزوق *or its parts*التقطيع*. The other reason because its three parts have 7 reasons and the reasons are faster than the wedges.[3]*

tafa'il

مستفعلن مستفعلن مفعولات ⁂ مستفعلن مستفعلن مفعولات

Example:

أزمان سلمى لا يرى مثلها الر ⁂ راؤون في شام ولا عراق
00//0/ 0//0/0/ 0//0/0/ ⁂ 0//0/ 0//0/0/ 0//0/0/
مستفعلن مستفعلن مفعولات ⁂ مستفعلن مستفعلن مفعولات

2.1.4.10 Al-Monsareh المنسرح

Why it named Al-Monsareh? *to be written later :) :) [3].*

tafa'il

مستفعلن مفعولات مستفعلن ⁂ مستفعلن مفعولات مستفعلن

Example:

ان ابن زيد لا زال مستعملا ⁂ للخير يفشي في مصره العرفا
0///0/ /0/0/0/ 0//0/0/ ⁂ 0//0/0/ /0/0/0/ 0//0/0/
مستفعلن مفعولات مستفعلن ⁂ مستفعلن مفعولات مفتعلن

2.1.4.11 Al-Khafeef الخفيف

Why it named Al-Khafeef? *Al-Khafeef name in Arabic means light. The reason behind the name is its wedge last harkah connected to its reason so, it became light* خفت*. The other reason is its light in teste* الزوق *or its parts* التقطيع *because it has a part with three reasons and the reasons are lighter than wedges[3].*

tafa'il

فاعلاتن مستفع لن فاعلاتن *** فاعلاتن مستفع لن فاعلاتن

Example:

حل أهلي ما بين دوني فبادو *** لي وحلّت علوية بالسخال
0/0//0/ 0//0/0/ 0/0//0/ *** 0/0//0/ 0//0/0/ 0/0//0/
فاعلاتن مستفع لن فاعلاتن *** فاعلاتن مستفع لن فاعلاتن

2.1.4.12 Al-Modarea المضارع

Why it named Al-Modarea? *Al-Modarea in Arabic means the present. It named by this name because it is the present version of the Al-Hazaj. Also, This Bahr wasn't famous in Arabic Poem and there weren't any popular peams or poetry used this Bahr before [3].*

tafa'il

مفاعلين فاع لاتن *** مفاعلين فاع لاتن

Example:

كأن لم يكن جديراً *** بحفظ الذي أضاعا
0/0//0/ /0/0// *** 0/0//0/ /0/0//
مفاعيل فاع لاتن *** مفاعيل فاع لاتن

2.1.4.13 Al-Moktadeb المقتضب

Why it named Al-Moktadeb ? *Al-Moktadeb in Arabic means reproduced from another thing* الإقتطاع *and because this Bahr word is all appeared into Al-Monsareh in all its words but also there is a difference in the order of the parts. So, it named Al-Moktadeb because it seems to reproduce from Al-Monsareh. We will have another section which will focus on the relation between the Bohor [3].*

tafa'il

مفعولات مستفعلن *** مفعولات مستفعلن

Example:

حف كأسها الحبب *** فهي فضة ذهب
0///0/ /0//0/ *** 0///0/ /0//0/
فاعلات مفتعلن *** فاعلات مفتعلن

2.1.4.14 Al-Mojtaz المجتز

Why it named Al-Mojtaz? *Al-Mojtaz in Arabic name is similar meaning for Al-Moktadeb it means reproduced from another thing* الإقتطاع أو الإجتزاز *and because these Bahr words are all appeared into Al-khafeef in all its words but also there is a difference in the order of the parts. So, it named Al-Mojtaz because it seems it reproduced from Al-khafeef* إجتُزَّ من بحر الخفيف *[3].*

tafa'il

مستفع لن فاعلاتن *** مستفع لن فاعلاتن

Example:

أنت الذى ولدتك *** أسماء بنت الحباب
0/0//0/ 0//0/0/ *** /0/// 0//0/0/
مستفع لن فعِلاتُ *** مستفع لن فاعلاتن

2.1.4.15 Al-Motaqareb المتقارب

Why it named Al-Motaqareb? *Al-Motaqareb in Arabic means convergent and is named by this name because the wedges are convergent to each other this is because between every two wedges one reason, so the wedges are convergent to each other. Also, another reason is its parts are similar to each other, so it named Al-Motaqareb [3].*

tafa'il

فعولن فعولن فعولن فعولن *** فعولن فعولن فعولن فعولن

Example:

فأمَّا تَميمٌ، تَميمُ بنُ مُرٍّ *** فألفاهُمُ القَومُ رَوْبَى، نِياما
0/0// 0/0// 0/0// 0/0// *** 0/0// 0/0// 0/0// 0/0//
فعولن فعولن فعولن فعولن *** فعولن فعولن فعولن فعولن

2.1.4.16 Al-Motadarek المتدارك

Why it named Al-Motadarek? There are different ideas behind this name.

- Al-Motadarek in Arabic means explored because Al-Farahidi forgets this Bahr and his student Al-Akhfash Al-Awsat has explored it named by this name.
- Al-Motadarek in Arabic also means followed by something يُدرك, and because this Bahr came after Al-Motaqareb, it names Al-Motadarek [3].

tafa'il

فَاْعِلُنْ فَاْعِلُنْ فَاْعِلُنْ فَاْعِلُنْ *** فَاْعِلُنْ فَاْعِلُنْ فَاْعِلُنْ فَاْعِلُنْ

Example:

لَمْ يَدَعْ مَنْ مَضَى لِلَّذِيْ قَدْ غَبَرْ *** فَضْلَ عِلْمٍ سَوَى أَخْذِهِ بِالأَثَرْ
0//0/ 0//0/ 0//0/ 0//0/ *** 0//0/ 0//0/ 0//0/ 0//0/
فَاْعِلُنْ فَاْعِلُنْ فَاْعِلُنْ فَاْعِلُنْ *** فَاْعِلُنْ فَاْعِلُنْ فَاْعِلُنْ فَاْعِلُنْ

2.1.5 Bohor Relations

to be added :)

2.2 Deep Learning Recurrent Neural Networks

What is Deep Learning? *Deep Learning is a new approach of Machine Learning research which focus on learning and understanding from the data without the needs for the human operator to formally specify all the knowledge that the computer needs. This method built using a hierarchy of concept which enables the computer to learn complex concepts by building them layer by layer from simpler ones. If there is a graph which shows how this concept built we will figure out a very deep graph with many layers, for this reason, we call this approach to AI deep learning [5]*

There was many of early trials to utilize the AI into real life problems. For Example, IBM's Deep Blue chess-playing system which defeated world champion Garry Kasprov in 1997 (Hsu , 2002).

Another approach which used to use AI but using hard-code knowledge about the world informal language. A computer can understand statements from the formal language automatically using logical inference rules. This is known as the knowledge base approach to artificial intelligence rules. None of these projects has achieved significant success. For Example, Cyc is tried to gather a comprehensive ontology and knowledge base about the basic concepts about how the world works Cyc (Lenat and Guha, 1989). Cyc is an inference engine and a database of statements in a language called Cycl. A staff of human supervisors enters these statements. People struggle to devise formal rules with enough complexity to describe the world accurately[5].

The difficulty faced in the previous system is due to the hard-coded knowledge has shown up the AI need to acquire their knowledge from the data itself. This capability is known as machine learning. This approach has introduced some algorithms which solve and tackle the problems from which we can, for example, check the email is spam or not. Also, it used for other problems for price predictions for housing Example of this algorithms is (Naive Bayes, Logistic regression).

This simple machine learning approach is working in the data but not with its original format it required some different representation to be input for the model. This different representation named feature engineering. Feature Engineering example: in case of email spam or not spam example it can be word frequency, char frequency, class attributes, capital letters frequency, some other data processing such as remove stop words from the input lemmatization. So, all the previous feature provided by a human expert which know the problem in details and analyzing which features it affect the data then add it as a feature to the input model.

However, for many tasks, it is difficult to identify the features which should be extracted. For example, we need to detect cars in photographs. We know every car have wheels. So, to detect cars, we can check if there is a wheel to be a feature for car detection. However, to detect or to describe wheels in terms of pixel values is a difficult task. The image may be not clear or may be complicated by shadows, the sun glaring off the metal parts of the wheel, the blurring in images may not make it clear sometimes, and so on[5].

One solution to solve this problem is to use machine learning itself to discover not only the output of the model but also the features which are the input for the model. This approach is known as representation learning. Learned representation can achieve better results than hard-designed representation. This approach also allows AI systems to rapidly adapt to new tasks or be automatically identify it from any new data. A representation learning can discover many features automatically fast or can take more times in case complex tasks, but at least it will get an excellent set of features which adapt for any complex problem without the need for manual features. In this research, we used the AI to identify the features for our model which make this model get a breakthrough results than the old fashion of manual feature machine learning used.

If we go back to the image example, we can show that it is not an easy task to extract features to detect the car from an image. So, Deep learning is trying to solve this problem in feature engineering by introducing representation learning that are build complex representations in terms of another simpler layer of representations Figure 2.1 shows how deep learning represents an image of a person by combining simpler representation example the edges and contours which led to understanding complex representations. The benefit from allowing the computer to understand the data and building the representation is the ability now for building and understanding very complex representation and also, to utilize and combine features from simpler to deep representations with many ways such as recurrent or sequences.

Modern deep learning provides a compelling framework for learning data problems. This model becomes more complex by the adding more layers and more units within a layer. Deep Learning model is working perfectly on the big dataset which allows the model to learn the data features in a good way.

In the remaining parts in this section we will start introducing the main concepts and component used in deep learning, Also the basic unit into Recurrent Neural networks and LSTM.

Figure 2.1: Illustrations on how can Deep Learning work based on images figure presented from [5] [6].

2.2.1 Logistic Regression

Logistic Regression is a machine learning algorithm which we can assume has the basic idea behind the deep learning we will explain it later. Also, Logistic Regression is one of the most used machine learning techniques for binary classification.

A simple example of logistic regression it would be if we have an algorithm for fraud detection. It takes some raw data input and detect if it is a fraud case or not lets assume fraud case is one and a non-fraud case is zero. David Cox developed logistic regression in 1958 [7]. The logistic name came from its core function logistic function which also named as *Sigmoid function* function (2.1). The Logistic function is shaped as S-shape. Also, one of these function features it can take any input real number and convert it into a value between 1 and 0.

Let's take an Example, Given x, we want to get the predictions of $\widehat{y}$ which is the estimate of y when $\widehat{y}$ is presented in equation (2.2). So,to calculate the output function for logistic regression using equation (2.3). Note: if we remove the Sigmoid function σ from the equation it will be Linear Regression model and $\widehat{y}$ can be greater than 1 or negative. Figure XXXX show the Sigmoid function output.

$$x = \frac{1}{1-e^{-x}} \quad \text{where} \quad x \in \mathbb{R}^{n_x} \tag{2.1}$$

$$\widehat{y} = P(y = 1|x) \quad \text{where} \quad 0 \leq \widehat{y} \leq 1 \tag{2.2}$$

$$\widehat{y} = \sigma(w^t x + b) \quad \text{where:} \quad \sigma(z) = \frac{1}{1 - e^{-z}}, w \in \mathbb{R}^{n_x}, b \in \mathbb{R} \tag{2.3}$$

2.2.1.1 Loss Error Function

Loss Error Function is the function which describes how well our algorithm can understand $\widehat{y}$ y b when the true label is y. It also can be defined as the difference between the true value of y and the estimated value of $\widehat{y}$. [6]. Equation (2.4) describe the loss function for Logistic Regression. There are another functions can represent the loss functions but we take the below as example. As we know y is the label which should be 1 or 0. So, The reason why this function make sense to describe the loss function as below

- in case (y = 1) equation (2.5) we need $\widehat{y}$ to be big as possible to be equal or near y true which is 1. So, $-(\log \widehat{y})$ will get the value. Note as explained before Sigmoid function can't be greater than 1 or less than 0.
- in case (y = 0) equation (2.6) we need $\widehat{y}$ to be small as possible to be equal or near y true which is 0. So, $-\log(1 - \widehat{y})$ will get the value.

$$\ell(y, \widehat{y}) = -(y \log \widehat{y} + (1 - y) \log(1 - \widehat{y})) \tag{2.4}$$

$$\begin{aligned} \text{(if y = 1)} \quad \ell(y, \widehat{y}) &= -(y \log \widehat{y} + (1 - y) \log(1 - \widehat{y})) \\ &= -(1 \log \widehat{y} + (1 - 1) \log(1 - \widehat{y})) \\ &= -(\log \widehat{y}) \end{aligned} \tag{2.5}$$

$$\begin{aligned} \text{(if y = 0)} \quad \ell(y, \widehat{y}) &= -(y \log \widehat{y} + (1 - y) \log(1 - \widehat{y})) \\ &= -(0 * \log \widehat{y} + (1 - 0) \log(1 - \widehat{y})) \\ &= -\log(1 - \widehat{y}) \end{aligned} \tag{2.6}$$

2.2.1.2 Cost Function

To predict y from $\widehat{y}$ we learn from the input parameters in this case it will be ***(w,b)*** from Equation (2.3) as ***(w,b)*** is the parameters which define the relation between input dataset X and the output Y. So, Cost Function will measure how well you are doing an entire training set and the ability to understand the relation between X,Y.

Cost function ***J*** in equation (2.7) is the average of loss function applied to every training example which equal the sum of the lost for each training example divided on the total number of training example.

[6] *Parts of this subsections are explained into Andrew NG Coursera courses in deep learning and It written using our understanding to this topic but the equations and the idea taken from the course https://www.coursera.org/learn/neural-networks-deep-learning/*

$$
\begin{aligned}
J(w,b) &= \frac{\sum_{i=1}^{m} \ell(y^i, \widehat{y^i})}{m} \quad \text{where m is the total number of training example} \\
&= \frac{-\sum_{i=1}^{m} [(y^i \log \widehat{y^i} + (1-y^i) \log(1-\widehat{y^i}))]}{m}
\end{aligned}
\tag{2.7}
$$

2.2.1.3 Convex Function vs Non-Convex Function

2.2.1.4 Gradient Descent

As we explained in the previous parts, we need to find the relation between X,Y from the input parameters ***(w,b)*** which will make the cost function (2.7) to the minimum. In other words we need to find the best value of ***J(w,b)*** which will represent the relation and reduce the error between y and $\widehat{y}$ So, we need to minimize ***J(w,b)***.

To illustrate the relation between ***J(w,b)*** we will assume for simplicity the relation will be function of one variable ***J(w)***. As shown in Figure XXXX we have a curve which represent the function ***J(w)*** we need to find the minimum point in this curve which is the local minimum assuming it is a ***convex function***. We will use equation to find the local minimum.

To explain how this equation works let's take a random point p from Figure XXXX let's take derivative *(which by definition is the slope of the function at the point)* The slope of this function is the height (h) divided into the width (w) it is the tangent of J(w) at this point. If the derivative is positive so, w will be update minus the derivative multiplied by learning rate alpha α as (2.8). We will repeat the previous step until value of w get the lowest minimum. When w get the lowest minimum the derivative will be negative so, w will start to increase again at this step the algorithm will stop.

$$\begin{aligned} w &:= w - \alpha dw \quad \textit{alpha is learning rate} \\ &:= w - \alpha \frac{dJ(w)}{dw} \quad \textit{d represent the derivative wrt w} \end{aligned} \tag{2.8}$$

Now, Let's generalize the above equation assume we have two parameters ***(w,b)*** and we need to calculate the cost function for ***J(w,b)*** we will work on as two steps first function (2.9) wrt ***(w)*** and second function (2.10) wrt ***(b)***

$$w := w - \alpha \frac{dJ(w,b)}{dw} \tag{2.9}$$

$$b := b - \alpha \frac{dJ(w,b)}{db} \tag{2.10}$$

2.2.1.5 Logistic Regression derivatives

As described we need to calculate the gradient descent to get the best $\widehat{y}$ which minimizes the total cost in equation (2.11). So, we will do backpropagation to get the value of dz we need to calculate da in equation (2.12) then we will calculate dz based on the output of da from equation (2.13). After that, We will start to take the derivative for z function parameters w_1, w_2, b. Once we got the values of dw_1, dw_2, db we can use it to calculate the estimated values of w_1, w_2, b in the equations (2.14), (2.15), (2.16)

$$\boxed{\widehat{y} = \sigma(z) = a} \longrightarrow \boxed{z = w^t x + b = w_1 x_1 + w_2 + x_2 + b} \longrightarrow \boxed{\ell(a, y)} \tag{2.11}$$

$$\boxed{da = \frac{d\ell}{da} = \frac{d\ell(a,y)}{da} = -\frac{y}{a} + \frac{1-y}{1-a}} \tag{2.12}$$

$$\boxed{dz = \frac{d\ell}{dz} = \frac{d\ell(a,y)}{dz} = \frac{d\ell}{da}.\frac{da}{dz}} = \boxed{(-\frac{y}{a} + \frac{1-y}{1-a}).a(a-1)} = \boxed{a-y} \tag{2.13}$$

$$\boxed{dw_1 = \frac{\partial \ell}{dw_1} = x_1 dz} \longrightarrow \boxed{w_1 := w_1 - \alpha dw_1} \tag{2.14}$$

$$\boxed{dw_2 = \frac{\partial \ell}{dw_2} = x_2 dz} \longrightarrow \boxed{w_2 := w_2 - \alpha dw_2} \tag{2.15}$$

$$\boxed{db = \frac{\partial \ell}{db} = dz} \longrightarrow \boxed{b := b - \alpha db} \tag{2.16}$$

2.2.1.6 Implementing Logistic Regression on m example

To implement a simple 1 iteration example below sample code simulate the program structure. First, assume $J = 0, dw_1 = 0, dw_2 = 0, db = 0$. Then calculate the feedforward step. Then backpropagation calculate. Finally, update the parameters. We can transfer the above equation into the below python sample code.

```
import numpy as np
J = 0, dw_1 = 0, dw_2 = 0,db = 0, alpha = .02
# FEED FORWARD PROPAGATION
A = 1 / (1 + np.exp(-(np.dot(w.T,X) + b))#    Z = np.dot(w.T,X) + b
cost = (- 1 / m) * np.sum(Y * np.log(A) + (1 - Y) * (np.log(1 - A)))
# BACKWARD PROPAGATION (TO FIND GRADIENT)
dw = (1 / m) * np.dot(X, (A - Y).T) #      dz = A - Y
db = (1 / m) * np.sum(A - Y)
# UPDATE THE PARAMETERS
w = w - alpha * dw
b = b - alpha * db

```

2.2.2 The Neuron

As we all know, Most computer research is trying to simulate the human brain as it is the most advanced smartest creation. If we are trying to check how the model understands the new information regarding for example bananas photo we can give a baby two bananas then ask him about it baby can remember it with all it new shapes. Same case if you inform any human about some information and trying to get a new inference it will automatically detect this information. So, The new research trying to simulate the human brain model into an Artificial Intelligence model to trying to get this performance. In this subsection, we will try to give an overview of the relation between the new research era and the human brain.

The neuron is the foundation unit of the brain. The size of the brain is as about the size of a grain of rice. The brain contains more over 10000 neurons with average 6000 connections with other neurons[7]. These massive networks allow our brain to build its knowledge about the world around us. The neuron is work by receiving the information from other neuron and process it uniquely then pass the output to other neurons this process is shown in figure 2.2.

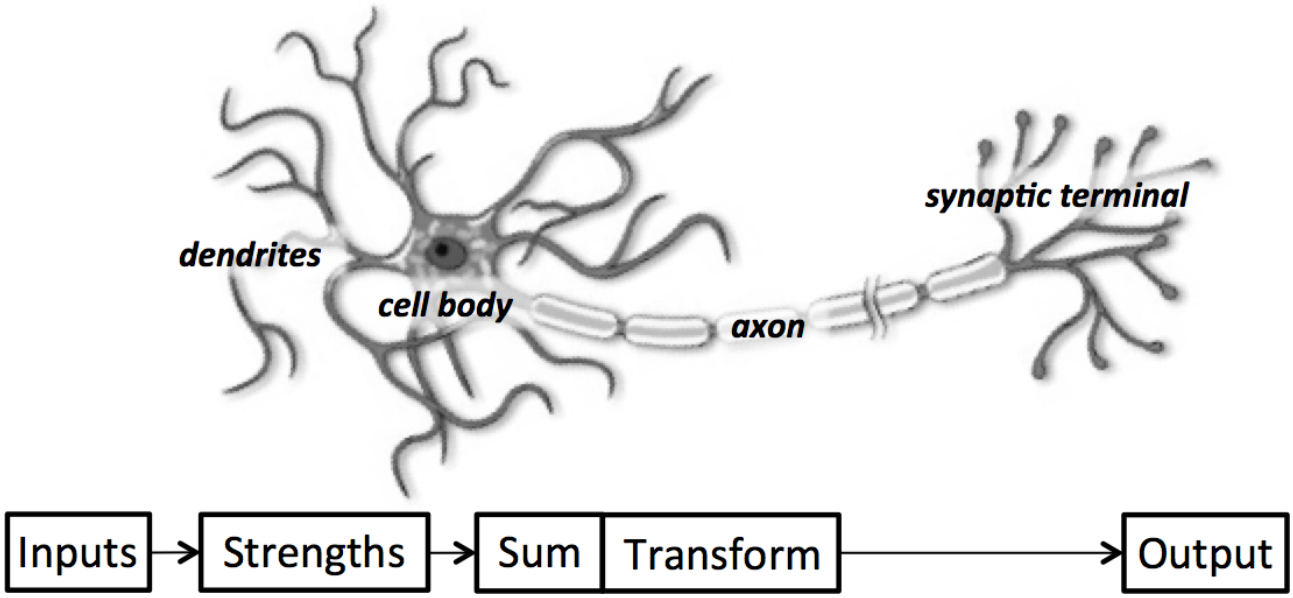


Figure 2.2: Description of neuron's structure this figure from [8]

How do we learn a new concept? *The neuron receives its input from dendrites. The incoming neuron connection is dynamically strengthened or weakened based on how often it is used, and the strength of each connection determines the contribution of the input to the neuron's output. Based on the connection strength it will have weight then the input is summed in the cell body. This sum is transformed into a new signal which is propagated along the cell's axon and sent to other neurons[8].*

The above biological model can be translated into an Artificial Neural Network as described in figure XXXX. We have an input $x_1, x_2, x_3, ..., x_n$ every input has its own strength (weight) $w_1, w_2, w_3, ..., w_n$. We Sum the multiplication of X and W to get the logit of the neuron, $z = \sum_{i=0}^{n} x_i w_i$. The logit is passed throw a function f to produce the output $y = f(z)$ the output will be the input to other neurons. Note: In many cases, the logit can also include a bias constant. So, in this case the function will be

$$y = f(\sum_{i=0}^{n} x_i w_i + b)$$

[7] *Restak, Richard M. and David Grubin. The Secret Life of the Brain. Joseph Henry Press, 2001.*

2.2.3 The Neural Network Representation

As explained previously, We have been trying to simulate the human brain model into our research work in Deep Neural Network. So, We will have multi-layers to allows the model to get in-depth knowledge and more computation performance to simulate the human brain.

Now, we will represent the functions per layer as below equations where *l is refer to layer number, i refer to the node number in the layer*(2.17)

$$\boxed{z^l = W^l x + b^l} \longrightarrow \boxed{a_i^l = \sigma(z^l)} \longrightarrow \boxed{\ell(a^l, y)} \tag{2.17}$$

What is the Neural Networks component?

Input Layer: Input layers is the input data raw for the network it is denoted as a^0. **Hidden Layers:** The layers between the input layers and the output layer it can be any number of layers. It also has a set of weighted input and produces an output through an activation function. Every layer in the hidden layer transmits the output to the other hidden layer as an input feature figure XXXX shows this relations.

Output Layer: It is one output layer with have the final results from the hidden layers.

2.2.4 Neural Network Computation

In this subsection, We will show as example on how we can compute the Neural Networks for every layer. In figure XXXX we have example of one layer we will continue explain on this example(2.18).

$$Z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]}, a_1^{[1]} = \sigma(Z_1^{[1]}) \tag{2.18a}$$
$$Z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]}, a_2^{[1]} = \sigma(Z_2^{[1]}) \tag{2.18b}$$
$$Z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]}, a_3^{[1]} = \sigma(Z_3^{[1]}) \tag{2.18c}$$
$$Z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]}, a_4^{[1]} = \sigma(Z_4^{[1]}) \tag{2.18d}$$

If we need to compute the above equations it will be simply be represented as vectorized way below matrix shows how we can implement it.

$$z^{[1]} = \begin{bmatrix} w_1^{[1]T} \\ w_2^{[1]T} \\ w_3^{[1]T} \\ w_4^{[1]T} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} = \begin{bmatrix} w_1^{[1]T} x + b_1^{[1]} \\ w_2^{[1]T} x + b_2^{[1]} \\ w_3^{[1]T} x + b_3^{[1]} \\ w_4^{[1]T} x + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_3^{[1]} \end{bmatrix}$$

2.2.4.1 Linear Neurons and Their Limitations

Now, We explained the equations for the feedforward Neural Network. We have only one point we need to discuss it which is the Activation function. Let's assume we will continue use linear function $y = wx + b$. So, if we have mutli-layer networks for example equation (2.19) it will end as linear function because composition of two linear function will be linear function. So, we will not compute deep computation and we will get limited information from the networks. So, to be able to detect the deep information we will use different function for the hidden layers example: Tanh(2.20), Sigmoid(2.1) and Relu(2.21). Most of binary classification problems use Sigmoid function for output layer. Also, we can use the same functions for the output but we can also use the linear for activation function in some cases.

$$Z^{[1]} = w_1^{[1]T} x + b_1^{[1]}, a_1^{[1]} = \sigma(Z_1^{[1]}) \tag{2.19a}$$
$$Z^{[2]} = w^{[2]T} a^1 + b^{[2]} = w^{[2]T}(w^{[1]T} x + b^{[1]}) + b^{[2]} \tag{2.19b}$$
$$= (w^{[1]T} W^{[2]T}) x + (w^{[2]} b^{[1]} + b^{[2]}) \tag{2.19c}$$
$$= W' x + b' \tag{2.19d}$$

$$a = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2.20}$$

$$a = max(0, z) \tag{2.21}$$

2.2.4.2 Softmax Output Layers

Sometimes our problem has multi-output results not only 1 or 0. For example, we have a problem to recognize the characters from 0 to 9 in MNIST dataset, But we will not be able to recognize digits with 100% confidence. So, we will use the probability distribution to give us a better idea of how confident we are in our predictions. The result will be an output vector of the form of the $\sum_{i=0}^{9} P_i = 1$

This is achieved by using a special output layer named softmax layer. This layer is differ from the other as the output of a neuron in a softmax layer is depending on the output of all the other neurons in its layer. This because its sum of all output equal 1. If we assume z_i be the logit of i^{th} softmax neuron, we can normalize by setting its output to represented from eq (2.22):

$$y_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2.22}$$

The strong prediction will have a value entry in the vector close to 1, while the other entries will be close to 0. The weak prediction will have multiple possible labels has almost the equal values[8].

2.2.4.3 Forward-Propagation in a Neural Networks

We will take the below figure XXXX as example of Deep Neural Network. So, to calculate the Forward propagation we will follow the below equation (2.23). Note: we assume $X = a^{[0]}$ as initial function notation. Also, $\widehat{Y} = g(Z^{[4]} = A^{[4]}$ as the final output layer.

$$Z^{[l]} = w^{[l]} a^{l-1} + b^{[l]}, A^{[1]} = g^l(Z^{[l]}) \tag{2.23}$$

2.2.4.4 Back-Propagation in a Neural Networks

We explained previously, how neural networks could learn their weights using gradient descent algorithm. In this part, we will explain how to compute the gradient of the cost function.

To compute the gradient descent in Neural Networks, we use an algorithm named *backpropagation*. The backpropagation algorithm was initially invented in the 1970s, but it wasn't shining until one of the most important papers in this field published in 1986 which describes several neural networks where backpropagation has a significant performance better than the earlier approaches and making it possible to use neural networks to solve problems which were previously not possible to be solved. Now, the backpropagation is the backbone for the learning in neural networks.

The backpropagation not only an algorithm which gives us the expression for partial derivative of the cost function C with respect to wights w and bias b but also it gives is an intuations about the change of the cost function while changing its variables w&b and its effect to the overall network.

As explained in logistic regression section (2.2.1.5) how we can calculate the derivatives for logistic regression with one layer using this equations(2.11),(2.12),(2.13),
(2.14),(2.15),(2.16).
We will generalize the derivatives equations to be for l layers from the below equations(2.24).

$$dz^{[l]} = da^{[l]} * g^{[l]'}(z^l) \tag{2.24a}$$
$$dw^{[l]} = dz^{[l]} \cdot a^{[l-1]} \tag{2.24b}$$
$$db^{[l]} = dz^{[l]} \tag{2.24c}$$
$$da^{[l-1]} = W^{[l]T} \cdot dz^{[l]} \tag{2.24d}$$

We can vectorize the above equation for Neural Network implementation as below equations(2.25).

$$dz^{[l]} = dA^{[l]} * g^{[l]'}(z^l) \tag{2.25a}$$
$$dw^{[l]} = \frac{1}{m} dz^{[l]} \cdot A^{[l-1]T} \tag{2.25b}$$
$$db^{[l]} = \frac{1}{m} \text{np.sum}(dz^{[l]}, \text{axis=1}, \text{keepdims} = \text{true}) \tag{2.25c}$$
$$dA^{[l-1]} = W^{[l]T} \cdot dz^{[l]} \tag{2.25d}$$

If we checked the input variable in the backpropagation we will find it is da^l and this is the derivative of (2.4) which we can get it as explained previously from (2.12) this is the formula for final layer in the feedforward step. If we need to calculate the vectorize version of this equation we can use equation(2.26)

$$da = \frac{d\ell}{da} = \frac{d\ell(a,y)}{da} = (-\frac{y^{[1]}}{a^{[1]}} + \frac{1-y^{[1]}}{1-a^{[1]}} \ldots - \frac{y^{[m]}}{a^{[m]}} + \frac{1-y^{[m]}}{1-a^{[m]}}) \tag{2.26}$$

2.2.4.5 How we Initialize the Wights

As we explained previously in Logistic regression, We initialized the weights to Zero. However, in Deep Neural Networks it will not work. Note: It is okay to initialize the Bias to Zero but the wights it will not works. Let's see what will happen if we initialize the weights and Bias to Zero.

Assume from figure XXXX we have two input vectors x_1, X_2 if we initialize $W^{[1]}$ to Zero from equation(2.27) and $b^{[1]}$ to Zeros. So, $a_1^{[1]} = a_2^{[1]}$ because both of the hidden units compute the same functions. Also, $W^{[2]}$ = [00] Then when we will compute the backpropagation we will find that $dz_1^{[1]} = dz_2^{[2]}$. So, After every iteration, we will find that the two hidden units calculate the same function and we will not get more information from this Deep Neural Network. We need to highlight that the main idea from Neural Networks as explained before is every hidden unit should work to get a new piece of information. The more hidden unit, the more hidden information we will get but if we initialize it to Zero. It will be the same function which is calculated, and we will not get any new information[8].

$$W^{[1]} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad b^{[1]} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.27a}$$

$$W^{[2]} = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{2.27b}$$

$$a_1^{[1]} = a_2^{[1]} \quad dz_1^{[1]} = dz_2^{[1]} \tag{2.27c}$$

$$dw = \begin{bmatrix} u & u \\ v & v \end{bmatrix} \quad W^{[1]} = W^{[1]} - \alpha dw \tag{2.27d}$$

To initialize weights and to get the maximum value of the neural network computation we should initialize the weight by any small random numbers to avoid the big weights which will tend to get the small slope from the Z where $Z^{[1]} = W^{[1]}X + b^{[1]}$ For example, if we use tanh we will get the big tail values $a^{[1]} = g^{[1]}(Z^{[1]})$. So, the big weights we more likely to get slow learning rate.

[8] *Parts of this subsection are explained into Andrew NG Coursera course in deep learning and It written using our understanding to this topic but the equations and the idea taken from the course https://www.coursera.org/learn/neural-networks-deep-learning/*

2.2.5 Recurrent Neural Networks (RNNs)

Deep Neural Networks shows its ability to solve many problems. However, in some use cases, Naive Neural Network architecture cannot works or get the expected results. One of the famous example related to this issue in the NLP tasks when working on a text problem for example, If we say our Harry is the king and Elizabeth is the queen, and we need our model to understand from the sentence that, Harry is he and Elizabeth is she. Also, if this word appears again, we need the model to detect that Harry is a person.

This type of problem has a dependency on the input text and how to get the output prediction based on the provided information from the input.

As explained previously, Most of the research in this area trying to simulate human brains. So, we will not find anyone every time trying to think about something start from scratch it always starts from another related point. Example, What is the human do if he tries to connect the information to generate the knowledge about something.

RNN shows its ability to work on sequence data and its related application problems such as natural language[10]. showed the effective of RNN on language modeling. There are many problems which based on this idea of dependency. For example,

- Time series anomaly detection.
- Speech recognition.
- Music Composition.
- Image captioning.
- Stock market prediction.
- Translation.

So, What are the problems in the Naive Neural Network architecture?

- Input and output length can be the different length in a different example.
- The most important issue is that the Naive architecture cannot share features learned across different positions of text. In this case, we will lose the learned feature, and the lake of dependency, in this case, will affect the overall performance.

What is the new proposed architecture which can provide a way to share the features between the Network?

- First, Assume we have input features x_1, x_2, x_n in the old architecture we input all these features to the Neural Network but now we will input for example x_1 and take the output activation from $a^{<1>}$ to be a feature input with x_2 then take the output activation from $a^{<2>}$ as input to x_3 similar till x_n figures 2.3, 2.4 shows an example. So, This new change will allow us to share the learned feature between the networks input data. Also, we can thing about it as multiple copies of the same network, each passing a message to a successor[9].
- Second, The feedfarward will be compute for time t and then we will calculate the loss at step t. The final loss is the sum of loss at every step t eq(2.28) explains the steps for feedfarward.

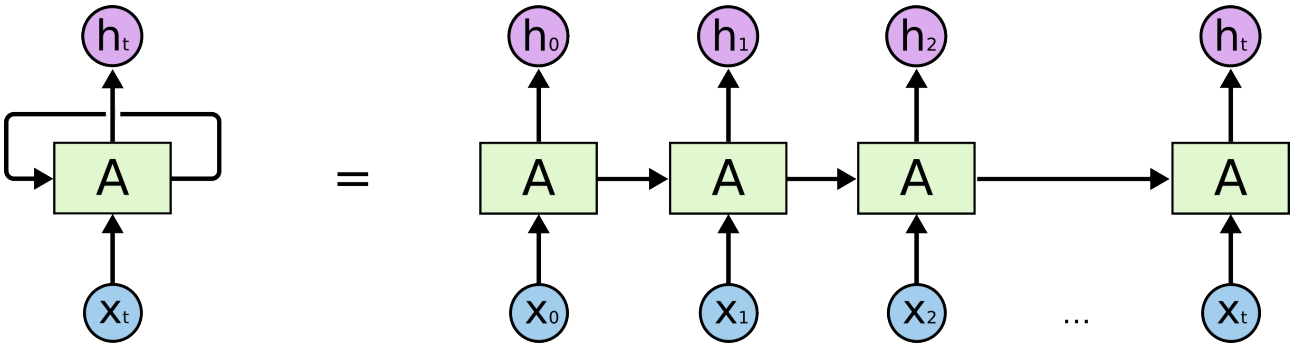


Figure 2.3: Recurrent Neural Networks Loops[9]

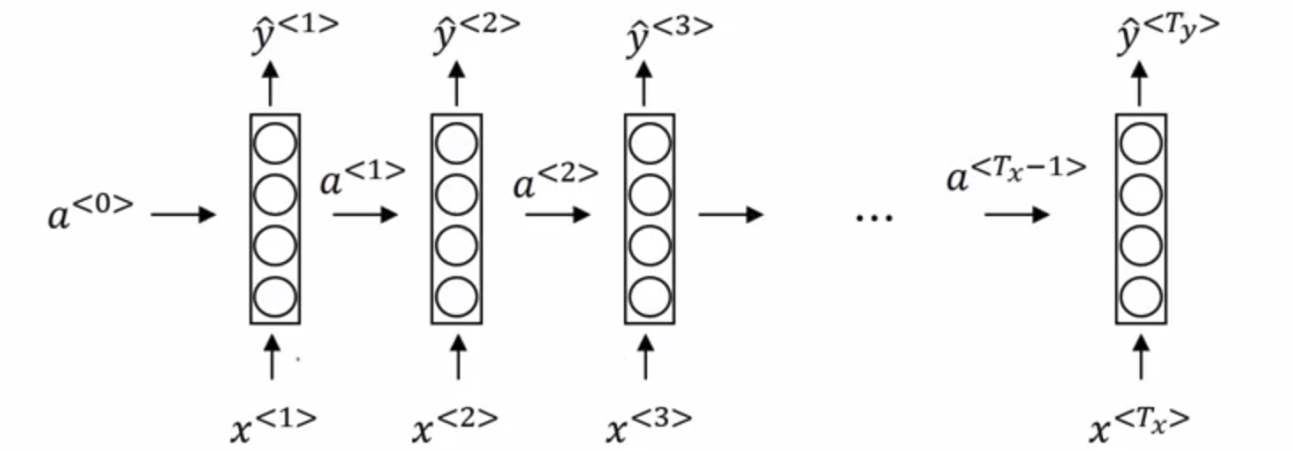


Figure 2.4: Recurrent Neural Networks feedfarward *This figure from Andrew NG course sequence models https://www.coursera.org/learn/nlp-sequence-models/*

Note: The backpropagation here will be calculated though time at every step.

$$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \tag{2.28a}$$
$$= g(W_a[a^{<t-1>}, x^{<t>}] + b_a) \tag{2.28b}$$
$$\hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y) \tag{2.28c}$$
$$\ell^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -(y^{<t>}\log\hat{y}^{<t>} + (1-y^{<t>})\log(1-\hat{y}^{<t>})) \tag{2.28d}$$
$$\ell(\hat{y}, y) = \sim\sum_{t=1}^{T_m} \ell^{<t>}(\hat{y}^{<t>}, y^{<t>}) \tag{2.28e}$$

2.2.5.1 Vanishing Gradient with RNNs

As we explained, RNN works on sequential data, and the idea is to predict new output not only based on the input data vector but also, other input vectors. Due to the recurrent structure in RNNs, it tends to suffer from long-term dependency to simplify this point lets have an example, the following sentence
*Waleed Yousef who is Associate Professor at Helwan University and teaching Data Science courses and its dependencies **was** got Ph.D. in Computer Engineering from GWU at 2006..*

In the previous example, to predict the word was is depending on long dependency to check if Waleed is singular or not to be consistent. Also, shows how some problems need the long-term dependencies handling.[Bengio et al.,1994][11] showed that Basic RNNs has a problem in long-term dependency. Another problem which may happen into basic Neural Networks is gradient exploding. One of the side-effects of gradient exploding is exponentially large gradient which causes our parameters to be so large. So, the Neural Networks parameters will have a server problem. Another fetal problem with Basic Neural Networks is overfitting problems [Zaremba et al., 2014][12].

So, to solve this learning problem [Hochreiter and Schmidhuber, 1997] introduced Long Short-Term Memory which helps to reduce the dependency problem using memory cell and forget gate.

2.2.6 Long Short Term Memory networks (LSTMs)

Long Short Term Memory networks aka LSTMs are a special type of RNN, capable of learning long-term dependencies. To solve the vanishing gradient problem for long-term dependencies, [Hochreiter and Schmidhuber, 1997][14] suggested new cell architecture for RNN by adding Long Short Term Memory which significantly reduced the long-term dependency problem using memory cell and forget gate.

LSTMs designed to help solving the long-term dependency problem and to hold information in memory for long periods of time. It also, use same RNNs sequential model but with adding some gating mechanism structure to every cell.

Both Basic RNNs and LSTM have the form of a chain of repeating modules of neural network. The main difference is the structure of the Networks.

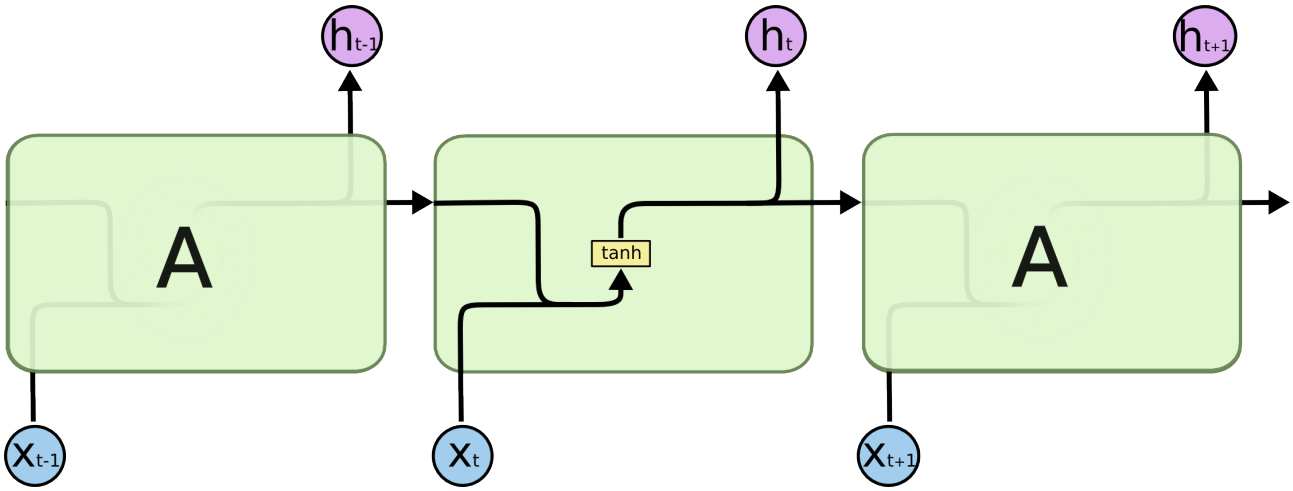


Figure 2.5: The repeating module in a standard RNN contains a single layer.[9]

In Basic RNNs it is very simple structure for every layer with simple output function 2.5. But in LSTMs it has four interacting layers 2.6.

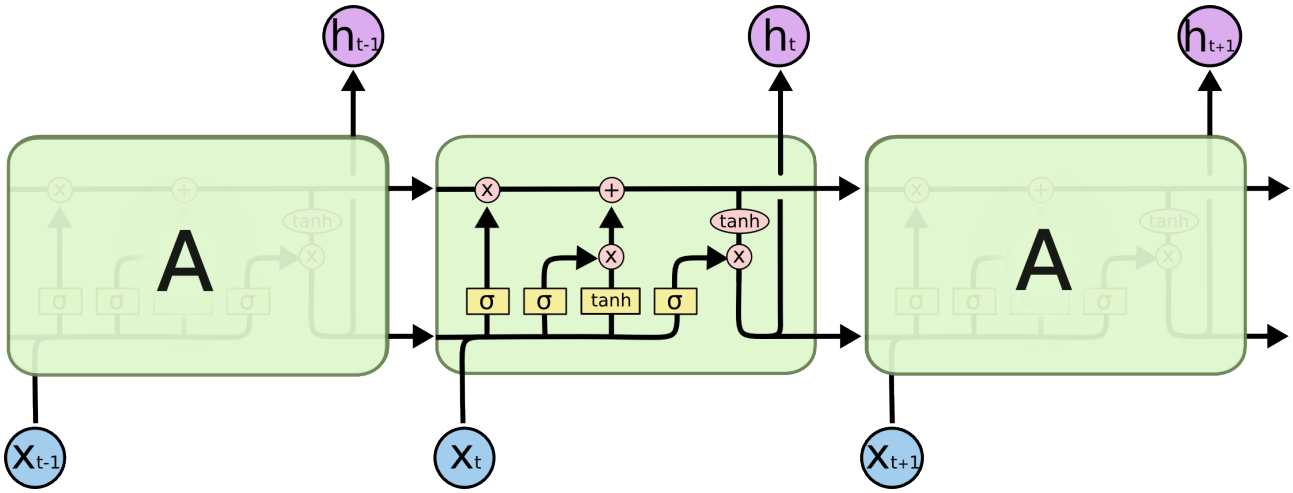


Figure 2.6: The repeating module in an LSTM contains four interacting layers.[9]

2.2.6.1 LSTM Gate Mechanism

The main component of LSTM is the cell state; It allows the information to pass through along it unchanged. In figure xxx the top line show the information flow through the cell. The LSTM cell can add or remove information to the cell state using the Gating mechanism.

Gates's idea is a methodology to manage the way how and which information pass or not. It controls information flow through the cell. It has three of these gates. They are consist of a sigmoid neural network layer 2.7 and a pointwise multiplication operation 2.8.

Sigmoid function output values between zero and one. If the value is one these means that everything should pass, while if the value is zero these means do not pass anything. So, the value output from the sigmoid function refers to the amount of each component should be passed.

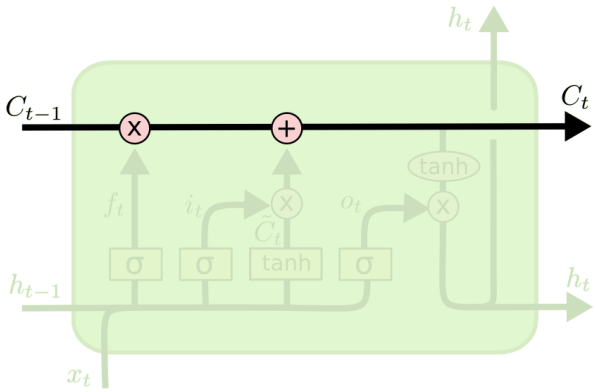


Figure 2.7: LSTM top horizontal line working as the medium for information flow [9]

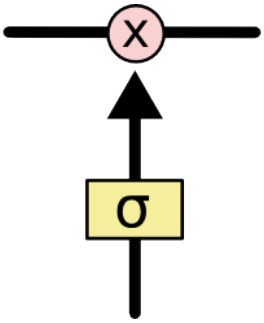
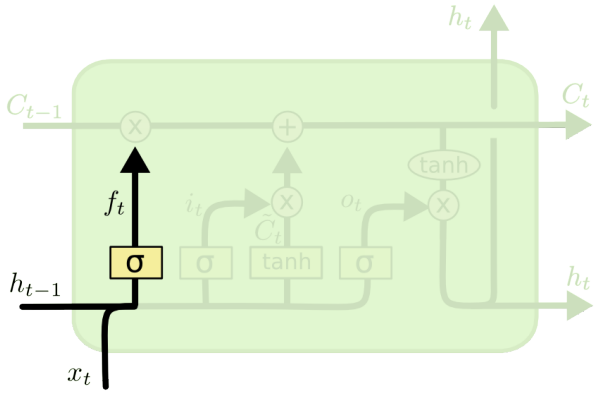


Figure 2.8: Cell gate with sigmoid function and a pointwise multiplication operation [9]

2.2.6.2 How LSTM Works?

We have explained LSTM has three gates with some

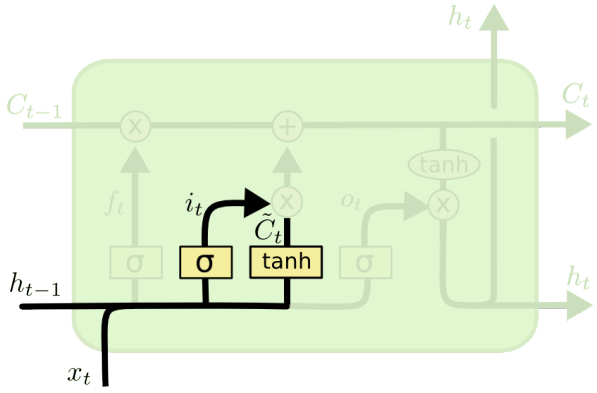
- **Forget Gate Layer** a Sigmoid layer 2.9 decides which information will be allowed to pass and which will not. It looks at h_{t-1} and x_t, and calculate the output from Sigmoid function between zero and one. As explained if one *everything should pass*, while if zero *do not pass anything*. The value zero or one depends on the value of the cell state if it includes a gender type and we need to predict the pronouns so, it will pass else it will ride of this state.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 2.9: LSTM sigmoid forget gate [9]

- **Input Gate Layer** is a combination between *sigmoid layer* which works to decide which values we should be updated, and *tanh layer* creates a new vector of the new information $\tilde{C}_t$ which should be stored for the next state 2.10. The previous combination controls the update state. This layer used when we have new input information. For example, We have a new subject named Elizabeth we need to store it for the next input. The next step is the pointwise multiplication and addition operations.



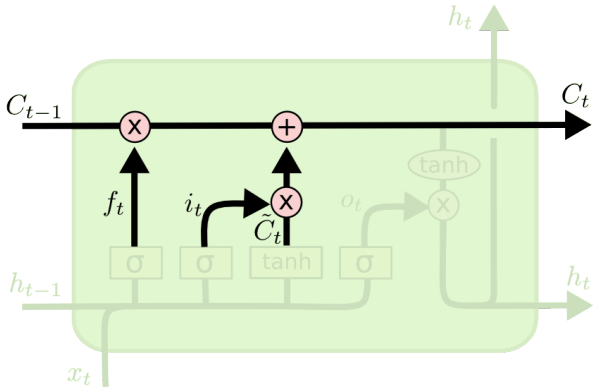
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 2.10: LSTM Input gate a combination of Sigmoid and Tanh layers [9]

- **Multiplication and Addition operations** This step is to apply the actions recommended by the previous gates. This step is the actions applying the forget of the old information and add the new information, as we decided in the previous steps. Let's look into the upper line in 2.11 there are two operations,

  1. Multiplication Operation: This operation to apply the forget gate step by multiplying the old state $\tilde{C}_{t-1}$ by the f_t.

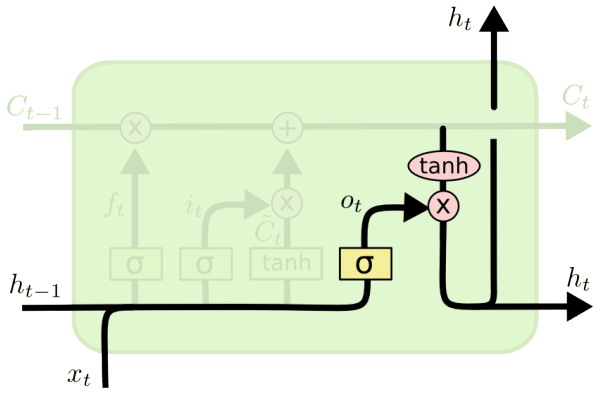
2. Addition Operation: This operation will add the output from the previous multiplication with the new input information scaled by how much we need to update each state value $i_t * \tilde{C}_t$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 2.11: LSTM Multiplication and Addition Operation in LSTM [9]

- **Output Gate** This gate is a combination of *sigmoid* layer and *tanh* layer. *Sigmoid* layer decides the information which should be output. Then the output of the *sigmoid* function will be multiplying with the output of the *tanh* layer of the cell state. This *tanh* will make the values between -1 and 1. The output of the multiplication of *sigmoid* and *tanh* will be the final output. In practice, this gate responsible for deciding which information should be the output. For example, if it saw a subject such as Elizabeth, it might want to output a verb to be relevant to her as a singular.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

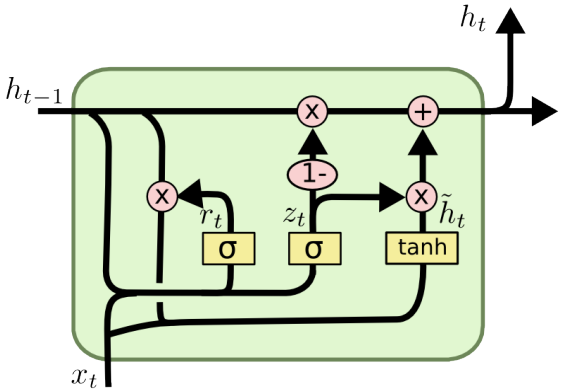
Figure 2.12: LSTM Multiplication and Addition Operation in LSTM [9]

We have explained the normal LSTM. Also, we need to mention that there are much research proposed different modifications of the normal LSTM type. We will not explain all the types, but we will give a small overview of one of these modifications named Gated Recurrent Unit (GRU) in the next part.

2.2.6.3 Gated Recurrent Units (GRUs)

In RNN Gated recurrent units (GRUs) are a gating mechanism, introduced in 2014 by Kyunghyun Cho et al. [13]. It works to overcome the problem for long-term dependencies. It also aimed to solve the vanishing gradient problem from Basic RNNs.It proposed a new architecture 2.13 similar than the LSTM but with some major variants as below,

- It combines the forget gate and input gates into a single gate named update gate and reset gate.
- The GRU unit controls the flow of information without having to use a memory unit. It just exposes the full hidden content without any control.
- It also merges the cell state and hidden state.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 2.13: GRU cell architecture [9]

The result of this modifications is GRUs are simpler and easier for modifications in the design. GRUs trains faster and in some case, it performs better than LSTMs on less training data mainly in language modeling. However, LSTMs has some benefits over GRUs in case longer sequences than GRUs in tasks requiring modeling long-distance relations.

Chapter 3

LITERATURE REVIEW

Chapter 4

DATASET

We have scrapped the Arabic dataset from two big poetry websites: الديوان[1], الموسوعة الشعرية[2]. Both are merged into one large dataset. It is important to note that the verses' diacritic states are not consistent, this means that a verse can carry full, semi diacritics or it can carry nothing. The total number of verses is 1,862,046 poetic verses; each verse is labeled by its meter, the poet who wrote it, and the age which it was written in. There are 22 meters, 3701 poets and 11 ages; and they are Pre-Islamic, Islamic, Umayyad, Mamluk, Abbasid, Ayyubid, Ottoman, Andalusian, era between Umayyad and Abbasid, Fatimid and modern. We are only interested in the 16 classic meters which are attributed to *Al-Farahidi*, and they are the majority of the dataset with a total number of 1,722,321 verses[3].

4.1 Preparing Data

4.1.1 Data Cleaning

[1] *aldiwan.net*
[2] *poetry.tcaabudhabi.ae*
[3] https://wwww.github.com/tahamagdy

Chapter 5

DATA ENCODING

5.0.1 Arabic Poem Encoding

5.0.1.1 One-Hot encoding

5.0.1.2 Binary Encoding

5.0.1.3 Two-Hot encoding

Chapter 6

MODEL TRAINING

Chapter 7

RESULTS AND DISCUSSION

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Future Work

REFERENCES

[1]

[2] Abdulrahman Almuhareb

[3] Al-Khatib Al tabrisi 1994. Al-Kafi in Al-Arud and Al-Quafi. Al-Khangi Press.

[4] ,

[5] Deep Learning, author=Ian Goodfellow and Yoshua Bengio and Aaron Courville, publisher=MIT Press, note=http://www.deeplearningbook.org , year=2016

[6] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In ECCV14.

[7] Cox, D. R. "The Regression Analysis of Binary Sequences." Journal of the Royal Statistical Society. Series B (Methodological) 20, no. 2 (1958): 215-42. http://www.jstor.org/stable/2983890.

[8] Fundamentals of Deep Learning by Nikhil Buduma and Nicholas Locascio (OReilly).

[9] Colah, Understanding Lstm Networks, 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[10] Mikolov et al., 2010 Tomas Mikolov, Martin Karafiát,Lukas Burget, Jan Cernock, and Sanjeev Khudanpur. Recurrent neural network based language model. In Inter-speech, volume 2, page 3, 2010.

[11] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2):157166, 1994.

[12] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329, 2014

[13] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):17351780, 1997.

APPENDICES

APPENDIX A

Phase Correlation Theory

Let $D_1(x,y)$ and $D_2(x,y)$ be the dilated images to be registered, the Fourier transform for both $F_1(u,v)$ and $F_2(u,v)$ is given by:

$$\begin{aligned} F_k(u,v) &= \mathcal{F}\{D_k(x,y)\} \\ &= \int_{y=-\infty}^{y=\infty}\int_{x=-\infty}^{x=\infty} D_k(x,y)\exp^{(-i2\pi\omega xy)}\,dxdy \end{aligned} \tag{8.1}$$

where, $\mathcal{F}$ is the Fourier operator, K denotes image 1 or 2, ω is the frequency (in hertz), x and y are the spatial domain coordinates, u and v are the frequency domain coordinates of the two images.

Given two images of size $N \times M$ shifted against each other, according to the Fourier shift property, their Fourier becomes:

$$F_2(u,v) = F_1(u,v)\exp^{\left(-i2\pi\left(\frac{u\Delta x}{M}+\frac{v\Delta y}{N}\right)\right)} \tag{8.2}$$

The Normalized Cross Power Spectrum ($C(u,v)$) is defined as:

$$C(u,v) = \frac{F_1(u,v)\cdot F_2(u,v)^*}{|F_1(u,v)\cdot F_2(u,v)^*|} \tag{8.3}$$

where '.' denotes the element-wise product, '*' denotes the complex conjugate.

Using equation 8.2:

$$C(u,v) = \frac{F_1(u,v)\cdot F_1(u,v)^*\exp^{\left(i2\pi\left(\frac{u\Delta x}{M}+\frac{v\Delta y}{N}\right)\right)}}{\left|F_1(u,v)\cdot F_1(u,v)^*\exp^{\left(i2\pi\left(\frac{u\Delta x}{M}+\frac{v\Delta y}{N}\right)\right)}\right|} \tag{8.4}$$

Since the phase term of $F_1(u,v)\cdot F_1(u,v)^*$ is zero, only the magnitude remains, i.e. $F_1(u,v)\cdot F_1(u,v)^* = |F_1(u,v)\cdot F_1(u,v)^*|$ and since the magnitude of any complex exponential is 1, the equation drops to:

$$\begin{aligned} C(u,v) &= \frac{|F_1(u,v)\cdot F_1(u,v)^*|\exp^{\left(i2\pi\left(\frac{u\Delta x}{M}+\frac{v\Delta y}{N}\right)\right)}}{|F_1(u,v)\cdot F_1(u,v)^*|} \\ &= \exp^{\left(i2\pi\left(\frac{u\Delta x}{M}+\frac{v\Delta y}{N}\right)\right)} \end{aligned} \tag{8.5}$$

the inverse Fourier transform of which is a delta function, i.e. a single peak.

The Normalized Cross Correlation (c) equals:

$$c = \mathcal{F}^{-1}\{C\} = \delta(x+\Delta x, y+\Delta y) \tag{8.6}$$

The shift in x and y between the two images $(\Delta x, \Delta y)$ takes the location of the maximum peak in c, such that:

$$(\Delta x, \Delta y) = \underset{x,y}{\text{argmax}}\{c\} \tag{8.7}$$