PRONOUN RESOLUTION WITH DEEP LEARNING

BY

MEHMET, TAZE

SEPTEMBER 2017

PRONOUN RESOLUTION WITH DEEP LEARNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
IZMIR UNIVERSITY OF ECONOMICS

BY
MEHMET TAZE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SEPTEMBER 2017

Approval of the Graduate School of Natural and Applied Sciences

(Assoc. Prof. Dr. Devrim Ünay)

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

(Assoc. Prof. Dr. Cem Evrendilek)

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

(Asst. Prof. Dr. Senem Kumova Metin)
Supervisor

Examining Committee Members Date: 05.09.2017

Asst. Prof. Dr. Senem Kumova Metin
Dept. of Software Engineering, IUE

Asst. Prof. Dr. Kaya Oğuz
Dept. of Computer Engineering, IUE

Asst. Prof. Dr. Tarık Kışla
Dept. of Computer Education and Instructional Technologies, Ege U.

ABSTRACT

PRONOUN RESOLUTION WITH
DEEP LEARNING

Taze, Mehmet

M.Sc. in Computer Engineering
Graduate School of Natural and Applied Sciences

Supervisor: Asst. Prof. Dr. Senem Kumova Metin
September 2017, 104 pages

In language, in order to prevent the repetitive use of an individual item, a referring pronoun or a noun phrase is employed instead. In such cases, the referred item is known as *antecedent* and the referring pronoun/noun phrase is named as *anaphor*. The problem of resolving references to earlier or later items, in other words the process of identifying relation between antecedent and anaphora is the *anaphora resolution.*

Anaphora resolution is used practically in a number of different natural language processing applications such as machine translation, text summarization, information extraction and question answering systems.

In this thesis, the task of anaphora resolution is simplified to pronoun resolution where only pronominal anaphora resolution is considered. We analyzed the performance of deep learning networks in Turkish pronoun resolution employing 12 features. Multilayer perceptron and convolutional neural networks are implemented with a number of different configurations. A data set of 593 positive samples (antecedent- anaphora pairs) is prepared from a collection of 10 child stories in Turkish. The experimental results showed that the highest performance in Turkish pronoun resolution is obtained by multilayer perceptron neural network with a medium number (9) of layers that employ too many neurons gives.

Keywords: Deep Learning, Anaphora Resolution, Pronoun Resolution, Natural Language Processing.

ÖZ

DERİN ÖĞRENME İLE ZAMİR ÇÖZÜMLEMESİ

Taze, Mehmet

Bilgisayar Mühendisliği, Yüksek Lisans
Fen Bilimleri Enstitüsü

Tez Yöneticisi: Asst. Prof. Dr. Senem Kumova Metin

Eylül 2017, 104 sayfa

Dilde, bir sözcüğün sürekli tekrar eden kullanımını önlemek için, ilgili sözcüğe atıfta bulunan bir zamir veya isim öbeği kullanılır. Bu gibi durumlarda, atıfta bulunulan sözcük *öncül*, atıf eden zamir veya isim öbeği ise *anafor* olarak adlandırılır. Önceki ve/veya sonraki atıfların çözümlenmesi bir diğer deyişle öncül ve anafor arası ilişkinin ortaya çıkartılması işlemi *anafor çözümlemesidir.*

Anafor çözümlemesi, makine çevirisi, metin özetleme, bilgi çıkarımı ve soru cevaplama sistemleri gibi birtakım farklı doğal dil işleme uygulamalarında kullanılır.

Bu tez çalışmasında, anafor çözümlemesi problemi zamir çözümlemesine indirgenerek Türkçe zamirlerin çözümlenmesinde derin öğrenme ağlarının başarımı incelenmiştir. Tez kapsamında, derin çok katmanlı algılayıcı ve derin konvolüsyonel sinir ağlarına 12 öznitelik girdi olarak verilerek pek çok farklı konfigürasyonda bu ağlar gerçeklenmiştir. Türkçe çocuk hikayelerinden derlenen 593 adet doğru örnek çifti (öncül – zamir) içeren bir veri seti oluşturulmuştur. Türkçe zamir çözümlemesinde en yüksek başarımın, her katmanda çok sayıda nöron içeren ve orta sayıda (9) katmana sahip çok katmanlı algılayıcı ağ ile elde edildiği görülmüştür.

Anahtar Kelimeler:Derin Öğrenme, Anafor Çözümleme, Zamir Çözümleme, Doğal Dil İşleme.

Acknowledgments

I would like to thank my supervisor Asst. Prof. Dr. Senem Kumova Metin who supported me about preparing and writing my thesis. Without her comments and inspiration, I would not have accomplished it. Not only during my master program but also all along my undergraduate years, she always welcomes me whenever I need her guidance.

Also, I would like to express my sincere gratitude to my mother, my father and my sister for supporting and encouraging me during all my education years as well as my thesis process.

I would like to thank the examining committee for their insightful comments and contributions to my thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In language, in order to keep the audience excited to listen the speech or to read the texts, speakers/authors tend to avoid repetitive use of same words or phrases. As a result, frequently they use words or phrases that refer to or replace earlier/later mentioned entities. Such words/phrases that are referring back/forward are known as anaphors. The entities that are pointed by anaphors are named as antecedents. And the process of identifying antecedent-anaphor pairs is the anaphora resolution.

In literature there exist several studies that discuss the notion of anaphora and focus on different points of view. In 1995, Grosz made the formal definition of anaphora [14]. He stated that making re-referencing to an event or an object that was mentioned before by a pronoun or name phrase is anaphora. In 1976, Halliday and Hasan discoursed that anaphora is providing cohesion by showing something that was mentioned before [36]. In 1997, Valin and LaPolla mentioned that the cohesion of text could be shaped by space, time and sameness/similarity of the action. Because of the cohesion in a text or in a speech, people perceive a text or speech in an integrated manner. By cohesion in discourse, items that are referenced by anaphora can be determined. By this, people can obtain new information about the thing that they encounter before [47]. For example, in the below sentences, the anaphoric relation is between the words “Onlar” and “Çiçekler”.

Çiçekler solmuştu. Onlar sulanmalıydı.

(The flowers were faded. They had to be watered.)

In this example, the pronoun "Onlar" is referencing to the noun "Çiçekler". The first sentence gives the current state of the item “Çiçekler” and the second sentence adds new information (an action to be performed) on the same item.

Detecting anaphora and resolving the reference can improve the understanding of a text or sentence but even detecting anaphora is difficult as there may be no indicator phrases or terms [41]. In some examples, some words are possibly anaphoric (e.g. pronouns) but not always so and anaphoric references may include a variety of structures. For example, though the below sentence includes the word “o” that commonly refers back to an item, in this sentence it is used in a phrase that emphasizes beauty of the roses instead of referring to something else.

Güller o kadar güzellermiş ki gören herkes perinin güllerine hayran kalırmış.

(The roses were such beautiful that everyone who saw them admired the roses of the fairy)

Anaphora resolution is used practically in many natural language understanding and processing applications such as machine translation, automatic abstracting, text summarization, information extraction and question answering systems [23]. Due to this wide range of tasks that require the resolution of anaphoric relations, a various amount of works are presented in years to solve this problem. These continuing works have led to a wide variety of anaphora resolution methods ranging from knowledge intensive technique heavily employing syntactic, semantic and real word information to knowledge-poor approaches resting on explicit information from the text.

As the works on anaphora resolution are examined, it is seen that the notion of anaphora is commonly categorized in four different ways

- The direction of referencing
- The capability of changing from expression to sound
- The grammatical category
- The location of antecedent and referencing item

The first categorization approach bases on the direction of the references. In this approach, two directions are considered. The first direction is “backward” meaning that the anaphora references something mentioned before as in our previous examples. This type of referencing items are named as anaphora. The second direction is “forward” where the item refers to something that will be mentioned later. In forward direction the referencing item is named as “cataphora”. In below sentences, a cataphoric relation is exemplified. The pronoun “onun” in first sentence is the cataphora that points forward to the noun “Sevgi” in next sentence.

Sabahtan beri onun hakkında konuşuyoruz. Fakat Sevgi ancak gelebildi.

(We have been talking about her since the morning. But Sevgi did not arrive yet.)

The categorization of anaphora in terms of sound content is being made by checking if the expression turning into sound in the superficial structure of a sentence. Anaphora that turned into sound (explicitly represented) is called as *"overt anaphora"* if it has not turned into sound, it is called as *"null anaphora"*.

In the below example,

Öğretmen$_1$, Mehmet$_2$'e bir kalem$_3$ verdi.

(Teacher gave a pencil to Mehmet)

$Q_1 Q_2$ Onu ders sonunda geri vermesini istedi.

(~~He~~ (Q_1) wanted ~~him~~ (Q_2) to return it back at the end of the course)

the word “Onu” in second sentence has its sound content but pronouns which are showed with the symbol Q_1 and Q_2 have not their sound content. In other words, two anaphors that refer to the items “Öğretmen” and “Mehmet” are not given explicitly in the second sentence. Therefore Q_1 and Q_2 are known as null pronouns.

The usage of overt/null pronouns varies according to the language. For example, Kornfilt (1997) mentioned that deleting of pronouns in Turkish is preferred when it is possible. As a result, he points that in Turkish anaphoric relations are being made more frequently over null pronouns. On the other hand, the usage of overt pronouns are much more common in English [17]. Because of this high number of null pronouns in Turkish anaphora resolution is stated to be much more difficult compared to English, (2007, Kılıçaslan vd.) [44].

In the third approach, the anaphora categorization is performed through grammatical categories of pointing items. The pointing can be done by a pronoun, noun, predicate or an indicator.

The studies that consider grammatical categories commonly use part of speech tags to resolve relations. In studies that consider only pronouns as anaphora, the anaphora is named as pronominal anaphora and the resolution task is given as pronoun resolution.

In the last categorization approach, some linguists also classify anaphora according to whether the antecedent and anaphora are in the same sentence or not. If they are in the same sentence, the anaphora is called as "*Intrasentential Anaphora*" and "*Intersentential Anaphora*" vice versa.

In this thesis, we considered overt pronouns (given in Table 1.1) that are referencing backward. The resolution experiments cover 4 types of overt pronouns (personal, locative, reflexive and reciprocal pronouns – given in Table 1.1) in Turkish. In our experiments all possible grammatical cases of mentioned pronouns(e.g, accusative, dative, ablative, locative) are considered.

Personal Pronouns	Locative Pronouns	Reflexive Pronouns	Reciprocal Pronouns
Ben	Bura	Kendim	Birbirimiz
Sen	Ora	Kendin	Birbiriniz
O	Şura	Kendi	Birbirleri
Biz	Burası	Kendimiz	
Siz	Şurası	Kendiniz	
Onlar	Orası	Kendileri	

Table 1.1: Overt Pronouns in Turkish

In the experiments, the overt pronominal anaphora are accepted to point backwards to previously mentioned antecedents in text. As a result, for each pronoun, the antecedent candidates are chosen from nouns or noun phrases that are observed prior to the regarding pronoun in a predetermined window.

Accepting the pronominal/pronoun resolution as a binary classification task, we built a data set of 593 positive and a maximum number of 17790 negative samples of pronoun-antecedent pairs and 12 different features (e.g, capital letter use in antecedent, number of tokens, number of characters) are determined to resolve the true pronoun-antecedent pairs. The features are provided to two different deep neural networks in a number of different settings. The resolution performances are analyzed.

The rest of this thesis is organized as follows. In chapter 2, we present the pronoun resolution and pronoun types in Turkish. Related works are demonstrated in chapter 3. In chapter 4, the data set preparation is explained. In chapter 5, deep learning in pronoun resolution is presented. In chapter 6, experimental setup is presented. In chapter 7, the deep learning machines' results are given and lastly, we conclude the study and suggest future works in Chapter 8.

Chapter 2

Pronoun Resolution

In grammar, a pronoun is described as a word or a phrase that may be substituted for a noun or a noun phrase, which once replaced, is known as the pronoun's antecedent. The process of identifying the pronoun and its regarding antecedent is the pronoun resolution.

In languages, there exist many different types of pronouns and each pronoun can hold many different roles in sentences (e.g. act as a subject, direct object, indirect object, object of the preposition, and more in a sentence,) that turn the pronoun resolution into a task that requires an understanding of the varieties specific to the regarding language. Since we work on the data sets built from Turkish texts in the thesis, in this section we will provide the required information on the pronoun types and the usage examples in Turkish.

In Turkish, there are 6 types of pronouns: Personal, Demonstrative, Reflexive, Indefinite, Interrogative and Relative pronouns. The pronouns may be in different grammatical forms(e.g, accusative, dative, ablative, locative) when they are used in different roles(e.g, subject, object) in sentences. As an example, in Table 2.1

the grammatical cases of personal pronouns are given. The further information on pronoun types will be given in subsections.

Nominative	**Dative**	**Accusative**	**Locative**	**Ablative**
Ben	Bana	Beni	Bende	Benden
Sen	Sana	Seni	Sende	Senden
O	Ona	Onu	Onda	Ondan
Biz	Bize	Bizi	Bizde	Bizden
Siz	Size	Sizi	Sizde	Sizden
Onlar	Onlara	Onları	Onlarda	Onlardan

Table 2.1: Grammatical Cases of Nouns

Besides the variety in types and their usages, Turkish has two major properties that must be considered during the identification of pronoun-antecedent pairs. These are:

1. Turkish allows the usage of null pronouns in subject and non-subject position of the sentence. In the Turkish language, both overt and null subject's number and person morphology agree with the verb. But if the third-person integration suffix is not overt or subject is not null, third-person suffix can be used optionally [23].

   For example, the below sentences may be used interchangeably in a text though they have different number of pronouns. The first sentence includes no pronouns. In the second one, both object and subject of the sentence are exchanged with the overt pronouns and in the last sentence, null pronouns are employed as the subject and the object.

   Çalışan-lar sorun-u bul–du-(lar) / Onlar o-nu bul-du-(lar) /

   Q_1Q_2Bul-du-lar

2. Turkish is not so informative in terms of signaling from third-person pronoun to its antecedent compared to other well-studied languages such as English. In Turkish third-person pronoun causes the ambiguity in terms of pointing to he, she or it.

   For example,

   AdamQ_1 kadınaQ_2 bir çiçekQ_3 verdi.

   (The man gave the woman a flower.)

   Q_1Arkadaşının onu$_{1/2/3}$ görmesini istemiyordu.

   (He did not want the child to see him/her/it.)

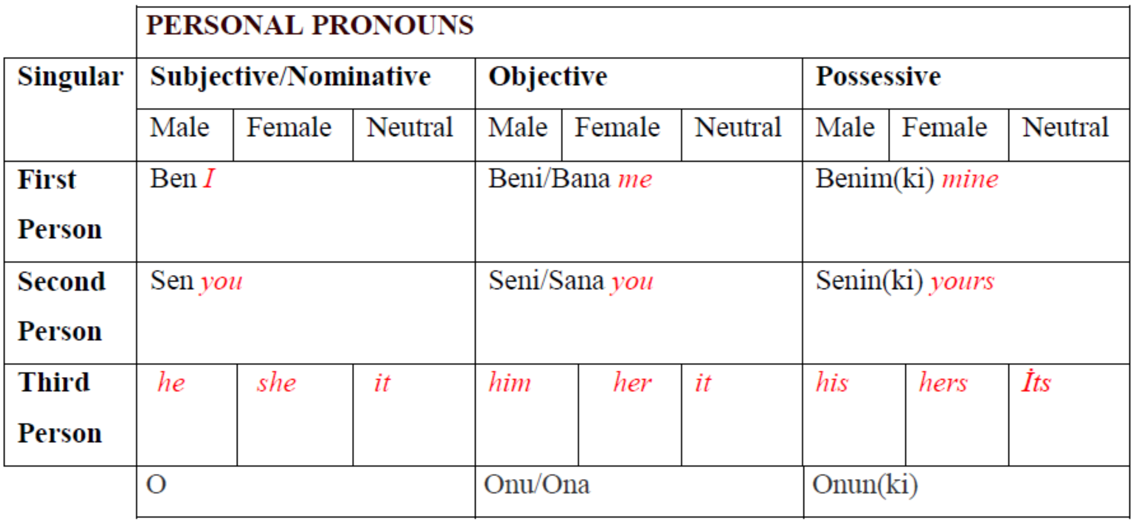
   At second sentence overt pronoun can refer to "adam", "kadına" or "çiçek" though this semantic ambiguity can be resolved when it is expressed in English [23].

In Appendix A and B, sample sentences (from our data set) that contain pronouns are provided. Appendix A includes the sentences where there is no antecedent for the annotated pronoun. On the other hand, Appendix B includes the sample sentences that true pronoun-antecedent pairs are detected.

2.1 Personal Pronouns

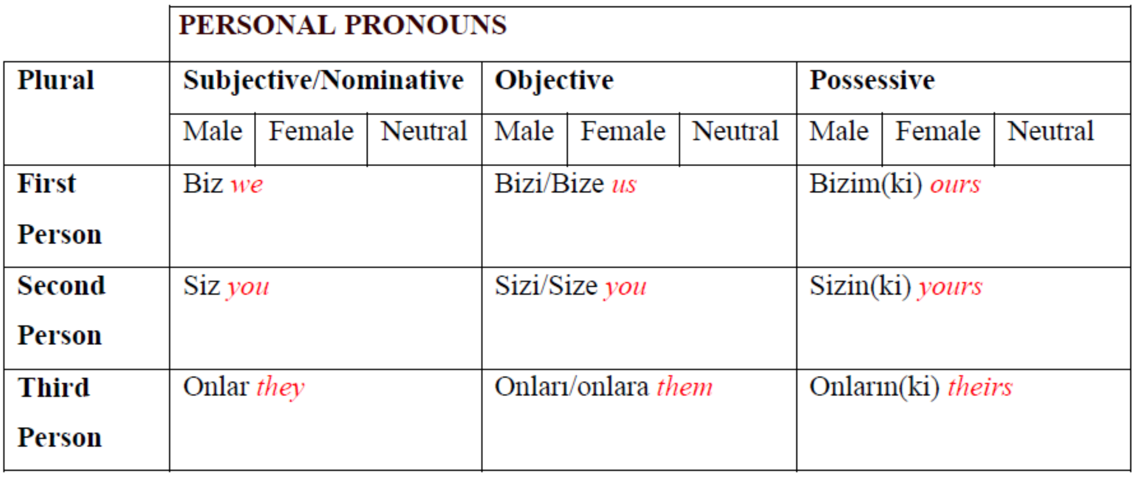
The personal pronoun is used instead of the names of persons. In order to use personal pronouns, it is important to know about the case (subject, object, and possessive), number (singular and plural), and person (first, second, and third). Table 2.2 and Table 2.3 give the singular and plural personal pronouns in Turkish respectively.

One of the difficulties in Turkish pronoun resolution is observed due to the third singular pronoun. As known, in Turkish third person singular pronoun "O" does not give any clue about gender on contrary to languages such as English and German. As a result, though the cases where gender identification is possible, it is not such easy to match the true antecedent with the regarding pronoun.



	PERSONAL PRONOUNS								
Singular	Subjective/Nominative			Objective			Possessive		
	Male	Female	Neutral	Male	Female	Neutral	Male	Female	Neutral
First Person	Ben *I*			Beni/Bana *me*			Benim(ki) *mine*		
Second Person	Sen *you*			Seni/Sana *you*			Senin(ki) *yours*		
Third Person	*he*	*she*	*it*	*him*	*her*	*it*	*his*	*hers*	*İts*
	O			Onu/Ona			Onun(ki)		

Table 2.2: Singular Personal Pronouns



	PERSONAL PRONOUNS								
Plural	Subjective/Nominative			Objective			Possessive		
	Male	Female	Neutral	Male	Female	Neutral	Male	Female	Neutral
First Person	Biz *we*			Bizi/Bize *us*			Bizim(ki) *ours*		
Second Person	Siz *you*			Sizi/Size *you*			Sizin(ki) *yours*		
Third Person	Onlar *they*			Onları/onlara *them*			Onların(ki) *theirs*		

Table 2.3: Plural Personal Pronouns

The other difficulty is examined since the same words are used in different types of pronouns for different purposes. For example, the third singular and plural personal pronouns “o” and “onlar” may be used either as personal pronouns or as demonstrative pronouns. If the third pronoun is related with the person in the sentence then it is accepted that it is a personal pronoun. On the other hand if it is related to an object then it turns to be a demonstrative pronoun. For example in below sentences, the pronoun “onlar” is firstly used as a personal pronoun, secondly as a demonstrative pronoun.

*Genç peri ve prens gülleriyle çok mutluymuş, ama onları üzen birşey varmış.→ Personal pronoun (Here “onları” is used for referencing "genç peri ve prens")

(Young fairy and the prince were so happy but there was something that makes them sad.)

*Peri güllerini çok sever, sabahları onları sularmış.→ Demonstrative pronoun (Here "güllerini" is the object of the sentence and "onları" is demonstrating "güllleri")

(Fairy has liked her roses so much and water them in the mornings)

2.2 Reflexive Pronouns

There are two forms of reflexive pronouns in Turkish which are “kendi” and its derivation of "kendisi", "kendileri" and other forms. Similar to the third personal pronoun, the reflexive pronoun in Turkish is genderless.

In our dataset, the following examples are located:

*Prensin başını devamlı suyun üstünde tutmaya çalışmış .Kendini onunla birlikte suyun akışına bırakmış.

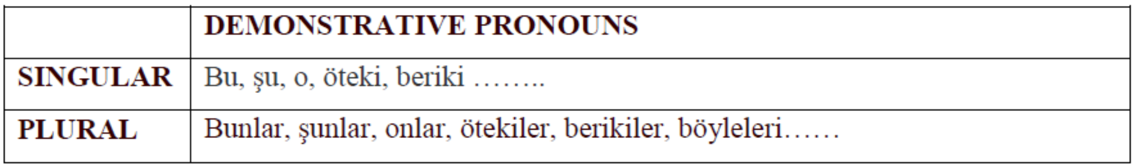
(She/He tried to keep prince's head above the water, then let her/him into the flow with the prince)

*Al oğlum bu parayla kendine defter kalem al. Güzelce okuluna git, demiş.

("Take this money son, go buy yourself a pencil and a notebook and keep up with your school and your studies" he said.)

2.3 Demonstrative Pronouns

Demonstrative pronouns point out specific persons, animals, places, things or ideas. Table 2.4 depicts the demonstrative pronouns in Turkish.



	DEMONSTRATIVE PRONOUNS
SINGULAR	Bu, şu, o, öteki, beriki ……..
PLURAL	Bunlar, şunlar, onlar, ötekiler, berikiler, böyleleri……

Table 2.4: Demonstrative Pronouns

Below, some examples of demonstrative pronouns are given from our data set:

*Orada insanlar olsa gerek. Bunlar beni gece yanlarında misafir ederler diye düşünmüş; ışığa doğru ilerlemiş. (Here “Bunlar” demonstrating “insanlar”)

(There must be someone ahead, as to accept me as their guests through the night, he said, as he headed towards the light.)

*Ne zaman birkaç orman hayvanını bir arada görüp yanlarına gitmeye kalksa huzursuzluğu çoğalıyordu. Çünkü onlar Gadro'ya sıradan biriymiş gibi davranıyorlardı. (Here "onlar" points "birkaç orman hayvanı")

(Whenever he sees an animal down in the forest and tries to get closer, he felt the uncomfort as it is going to treat Gadro as someone normal.)

Similar to the third personal pronouns, some of the demonstrative pronouns may be used also as determinants. Below, two sentences from the texts that we used to construct our dataset are given. For these usages, there exist no true pronoun-antecedent pairs.

*Gadro bir gün yine bu tepeye çıkmıştı. (Here "bu" is determinant)

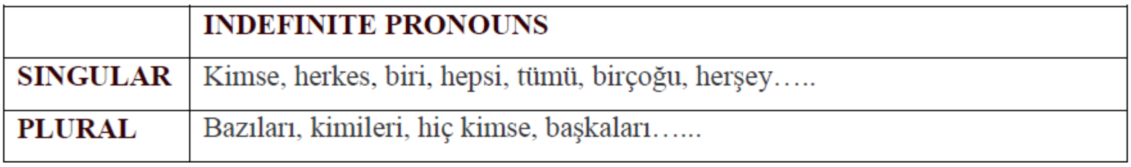
(Once, Gadro climbed up that hill again.)

*Yarışmalara bu yıl da ilgi çok azdı.(Here "bu" is also determinant)

(The interest in the competition was again slight this year.)

2.4 Indefinite Pronouns

Indefinite pronouns do not refer to any particular person or place or thing. They replace nouns without specifying which noun they replace. Table 2.5 depicts the some indefinite pronouns in Turkish.



	INDEFINITE PRONOUNS
SINGULAR	Kimse, herkes, biri, hepsi, tümü, birçoğu, herşey…..
PLURAL	Bazıları, kimileri, hiç kimse, başkaları…...

Table 2.5: Indefinite Pronouns

There are a lot of different forms of indefinite pronouns in Turkish. Examples of indefinite pronouns in used dataset are given below:

*Ertesi gün gölde binlerce ördek toplanmıştı. Hepsi büyük bir sabırsızlıkla Gadro'yu bekliyordu.

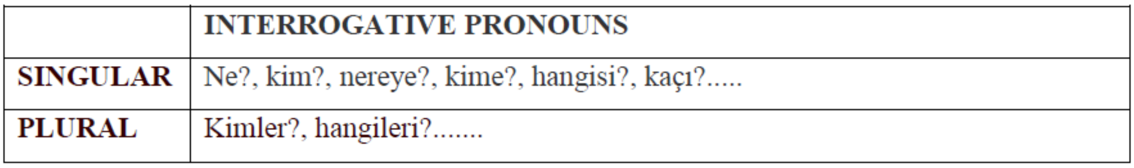
(The other day, thousands of ducks gathered in the lake, waiting for the Gadro impatiently.)

*Padişah hikayelerin hepsini dikkatle dinlemiş, adamlara acımış.

(Sultan listened to each of all of the stories carefully and felt sorry for the men.)

2.5 Interrogative Pronouns

Interrogative pronouns are used to begin or introduce interrogative sentences. Table 2.6 gives the regarding pronouns in Turkish.



	INTERROGATIVE PRONOUNS
SINGULAR	Ne?, kim?, nereye?, kime?, hangisi?, kaçı?.....
PLURAL	Kimler?, hangileri?.......

Table 2.6: Interrogative Pronouns

In our experiments, interrogative pronouns are ignored since it is observed that they are generally not giving a strong clue about the antecedent of the anaphora and the number of samples was not enough to run the experiments effectively. These pronouns are annotated in data set and it is planned to study on them as a further work. An example of Interrogative pronouns in the used dataset is given below:

*Kim ola ki? Nereden çıktı birdenbire?

(Who is he? Where did he come suddenly?)

2.6 Relative Pronouns

Relative pronouns begin with a subordinate clause and connect that clause to another noun that precedes it in the sentence. In other words, relative pronouns are used instead of its predecessor of the word. And it exists as an appendix and holds the place of the word before it. The disadvantage of relative pronoun "-ki" is that it may be used either as a conjunction as a adjective or a pronoun. Below are three sentences where "-ki" is used in different purposes. It is used as a pronoun, adjective and conjunction respectively in sentences 1,2, and 3.

Sentence 1: Şemsiyen yoksa benimkini alabilirsin.

(If you do not have an umbrella you can take mine.)

Sentence 2: Benim saçlarım güzel onunki çirkin.

(My hair is beautiful but hers is not)

Sentence 3: Senin ki güzel de benim ki değil mi ?

(Yours is beautiful but mine is not?)

Chapter 3

Related Work

In this section, firstly an overview of anaphora resolution approaches will be provided and some example studies will be explained briefly. Following, anaphora resolution studies on Turkish will be examined.

3.1 Anaphora Resolution Approaches

The main goal in anaphora resolution is determining the pairs of anaphora and the antecedents in a given text or speech. The proposed approaches commonly find possible antecedents for each anaphora in discourse and choose the most suitable one with the highest success. The resolution studies differ in the way they follow to reach the goal, the methods and features that are used. Also the language that the experiments are run on is an important factor that may change the structure of the resolution system. Anaphora resolution studies may be categorized in three main groups:

1. Knowledge Based Approach
2. Learning Based Approach
3. Hybrid Approach

3.1.1 Knowledge Based Approaches

Knowledge based approaches are based on formal, syntactic, lexical, and discourse source information. According to the level of information used during resolution, they are commonly divided into 2 groups: knowledge poor and knowledge rich.

Two of the earliest examples of knowledge-based approaches are proposed by Hobbs (1978). The first is a simple and effective method that runs on parse trees of a sentence in the text. Parse trees that show the grammatical structure of the sentence are generated without changing locations of the words in a sentence. Algorithm traverses the tree from left to right and searches a name phrase that has a compatible number and gender features. Hobbs [20] used a novel of Arthur Hailey and a part of Newsweek Magazine in his experiments and achieved 88.3% success rate. The second approach proposed by Hobbs is a modified version where the meaning analysis is added to the first approach. It is reported that the 91.7% success rate is achieved with this approach.

In 1994, Lappin and Leass introduced the resolution of anaphora procedure (RAP) that uses remarkable criteria that are obtained from a syntactic structure and striking state model. In RAP, during the resolution process, some of the candidate name phrases are filtered by using formal and syntactic filters. Then, the remaining candidates are assigned with specific values through predefined criteria. The criteria used in this algorithm are proximity, the priority of subject, the priority of existence, and the priority of name phrase. It is reported that 72% success rate for interjectional and 89% success rate for intra sentence was obtained [26].

Kenndy and Boguraev (1996) modified the RAP. The modified algorithm differs from RAP since it does not need the fully unbundled structures works by determining only name phrases. In addition, in modified algorithm, the candidate antecedents are sorted by only syntactic information instead of grammatical role. Kenndy and Boguraev obtained 75% success rate on third person pronouns [22].

3.1.2 Learning Based Approaches

Learning based approaches use machine learning techniques to solve anaphora resolution studies. These approaches may be either supervised or unsupervised methods. For example, in the study of Aone and Bennet(1995), a learning system based on decision tree is proposed. The system is trained by 66 features that are lexical, syntactic, semantic, and location information features. Some of these features refer to only discrete features of antecedents or anaphora. Although, the other part of features refers to the relationship between the anaphora and antecedent. It is reported that 90% success rate is obtained by the system [3]. The RESOLVE system is presented by McCarthy and Lehnert (1995). In RESOLVE system, eight sematic features are provided to a decision tree [30] and the experiments are run on the data set obtained from MUC-5 texts. The success rates are 87.6% for pruned and 92.4% for un-pruned trees. Another study where decision trees are utilized is introduced by Soon, Ng and Lim (2001). Soon, Ng and Lim (2001) used 12 features (lexical, syntactic, semantic and location features) and made experiments on MUC-6 and MUC-7 corpora. The contribution of each property was also measured in this study, and they obtained 68% success rate by using only 3 features [42].

3.1.3 Hybrid Approaches

Hybrid approaches were emerged by using a combination of learning based approaches and knowledge based approaches. The studies of Mitkov, Evans and Orasan [31] and Preiss [33] may be given as hybrid approach examples.

In 2002, Mitkov developed a system whose name is MARS (Mitkov's Anaphora Resolution System). This system works fully automatically. Unmarked texts are taken and parsing and anaphora resolution operations are made. Firstly, this system specifies the name phrases by the syntactic parsing. Then it determines anaphoric items with machine learning methods. For all anaphoric items, system sums up the candidate antecedents up to two sentences and applies person, number and gender integration filters on these. Extracted candidate antecedents are sorted themselves by using the various factors and real antecedent will be determined. Mitkov, Evans and Orasan obtained 61.6% success rate in this study [31].

Preiss improved Kennedy and Boguraev's algorithm in his study (2002). Preiss modified the previous algorithm by machine learning methods and measured the effectiveness of the memory-based approach in this way. In experiments, four states of the art probabilistic parser are used. It is reported that after the modification the system returned approximately same performance with the original rule-based approach [33].

3.2 Anaphora Resolution Studies in Turkish

To the best of our knowledge, the earliest study on anaphora resolution in Turkish is presented by Turan (1996). In (Turan, 1996), a system based on centering theory is proposed. As different from English, sorting in Turkish should be based on thematic roles instead of sorting based on grammatical rules. The purpose of the study is detecting an aspect of discourse consistency which includes anaphoric relations between utterances with specific emphasis on the topic in Turkish. In this work, three complementary questions hold for concerning discourse anaphora. The first question is the noun phrases contribution to the Cf-list which investigated in Turkish. The second question is the study of investigation of the determining the most salient entity which is the reasonable antecedent for definite anaphoric reference at next utterance and the last question is the study of determining the discourse functions of null vs. overt pronouns vs full noun phrase in subject position in Turkish[45]. Yüksel and Bozşahin suggest that the planning system which creates an anaphora that appropriate to the context. Their system combines with a set of rules for binding relations and Centering Theory for modeling local and also nonlocal references. In the system accordance with the binding rule, an antecedent of reflexive pronouns should be in the local domains. Their system is based on Binding Theory (Chomsky 1981) which is concerned with interpolation of anaphora, pronouns and referring statements and also system based on Centering Theory (Grosz et al 1995) which is an attempt to relate focus on attention and the choice of antecedent's reference and comprehended with coherence of the discourse [48]. Yıldırım and Kılıçarslan create a machine learning based approach for predicting antecedents of anaphora. They use personal pronouns in Turkish sentences with a decision tree algorithm classification couple with the crowd learning methods. They presented the first results of some experiments by applying a decision tree classification on a text. The used Weka j-48 decision tree classifier with some of the setting in their experiments [49]. Tüfekçi and Kılıçarslan presented a work which is an implementation of a version of Hobbs (1978) naive algorithm for pronoun resolution algorithm adapted for Turkish. Their system processes nominal level knowledge by using syntactic information to gather possible antecedent of given pronouns. Their paper's aim is narrowing down the scope of the anaphora phenomena. They focus on underneath problem of anaphora resolution which is the resolution of the third person singular pronominal anaphora to antecedents in the form of a noun [44]. (Erguvanlı Taylan, 1986) studies are about explaining how an overt or null pronoun in a sentence can be used and where these null or overt pronouns can be used. Also, they state that anaphoric relations can occur by using mandatory overt and mandatory zero pronoun and also overt or null pronoun by selection. Enç argues that the choice between overt and null subjects is determined by the criterion of whether the topic of discourse is a newly established one(zero pronoun) or continuation of the old one(overt pronoun) [43]. (Tın, Akman, 1998) has the big importance of contextual factors for anaphora resolution in Turkish. In addition, they describe the computational framework called BABY-SIT, an anaphoric ambiguity is resolved by related contextual information into effect by means of forward and backward chaining constraints. They define a new and novel approach for the analysis of pronominal anaphora in Turkish[10].

Chapter 4

Dataset Preparation

In this study, a collection of 10 Turkish child stories (totally 1274 sentences) is used as a source of pronoun-antecedent sample pairs set. The collection is chosen to include child stories since they are;

- More clear and understandable than other types of writing
- Not handled psychological analysis and long description and depiction
- Shorter than other types of writing
- Preprocessed easier than other types of writing

The dataset preparation process requires for tokenization, part of speech (POS) and manual annotation tasks to be performed.

10 child stories are collected from web. There existed no selection criteria and each text is chosen randomly. The collection includes 11613 tokens (any string between two blanks) and a number of punctuations. The sentences in the collection (totally 1274 sentences) are determined and each sentence is given POS-tagger machine proposed in [25] in order to be tagged by POS.

The POS tagger assigned a POS tag from a set of 13 types to each word in collection. The POS tags are *Adj* (Adjective), *Noun*, *Det* (Determinant), *Verb*, *Punc* (Punctutiaon), *Adv* (Adverb), *Conj* (Conjuctuion), *Pron* (Pronoun), *Postp* (Postposition), *Num* (Number), *Interj* (interjection), *Ques* (Question) and *Dup* (Duplication). In Figure 4.1, the ratio of each part of speech tags in our collection is given.

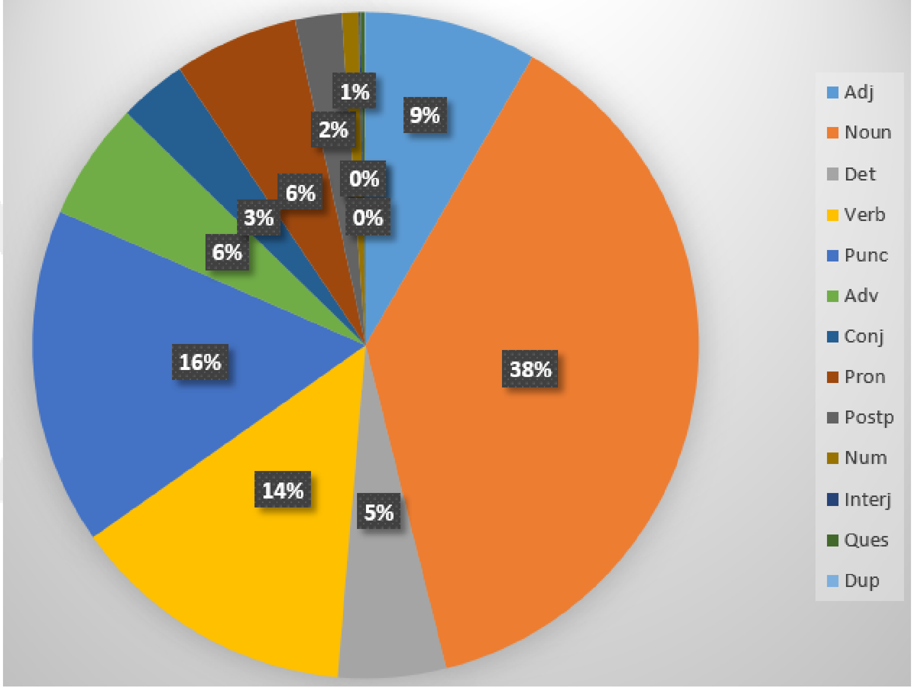
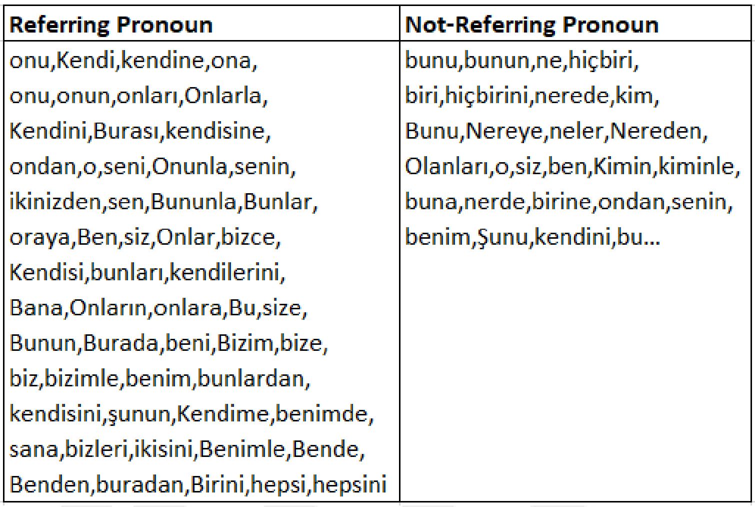


Figure 4.1: POS Tags in Collection

In order to prepare the dataset of pronoun–antecedent pairs, each word/token tagged as pronoun (totally 442 token) by the POS tagger is examined manually. This examination is actually done to check whether the pronoun tagged token has a proper antecedent or not. After this examination, each pronoun tagged token is classified as *referring* or *not-referring* pronoun. *Referring* pronouns are the ones that have a valid antecedent. In contrast, *not-referring* pronouns (totally 107 tokens) are the ones that do not have a valid antecedent.

Table 4.1 gives the lists of both types of pronoun examples in the collection. In our experiments we employed only referring pronouns in the collection. From now on, referring pronouns will be named as pronoun in the thesis.



Referring Pronoun	**Not-Referring Pronoun**
onu,Kendi,kendine,ona,	bunu,bunun,ne,hiçbiri,
onu,onun,onları,Onlarla,	biri,hiçbirini,nerede,kim,
Kendini,Burası,kendisine,	Bunu,Nereye,neler,Nereden,
ondan,o,seni,Onunla,senin,	Olanları,o,siz,ben,Kimin,kiminle,
ikinizden,sen,Bununla,Bunlar,	buna,nerde,birine,ondan,senin,
oraya,Ben,siz,Onlar,bizce,	benim,Şunu,kendini,bu...
Kendisi,bunları,kendilerini,	
Bana,Onların,onlara,Bu,size,	
Bunun,Burada,beni,Bizim,bize,	
biz,bizimle,benim,bunlardan,	
kendisini,şunun,Kendime,benimde,	
sana,bizleri,ikisini,Benimle,Bende,	
Benden,buradan,Birini,hepsi,hepsini	

Table 4.1: Referring and Not-Referring Anaphora

While labeling the pronoun-antecedent pairs, it is observed that some of the pronouns in dataset do not refer to a single word (token). Below, an example for such a case is given.

Güllerinin en güzeli solmamış. İyi yürekli peri, her gün onu evinin penceresinden seyrediyormuş.

(The most beautiful of his roses did not fade. The kind-hearted fairy watched it from the window everyday.)

In above sentences the pronoun "onu" in second sentence refers to the phrase "güllerinin en güzeli" in first sentence. We assume that it is a hit (a correct classification/ true pair) if a machine pairs one of the words in phrase with the regarding pronoun. In such cases where pronoun refers to a phrase of multiple words, the regarding phrase is split into words and for every word in the phrase, a pair is inserted to the dataset which includes the regarding pronoun. As a result, for the given sentence, (onu, güllerinin), (onu, en), (onu, güzeli) pairs will be inserted to the data set.

Figure 4.2 shows that the numbers of words used in antecedents in our collection. There are totally 335 pronouns (referring to a single word or multiple words). 191 of them refer to a single word, 67 of them refer to two words, 59 of them refer to three words, 9 of them refer to four words phrase, 5 of them refer to five words phrase, 3 of them refer to seven words phrase and 1 of them refers to eight words phrase.

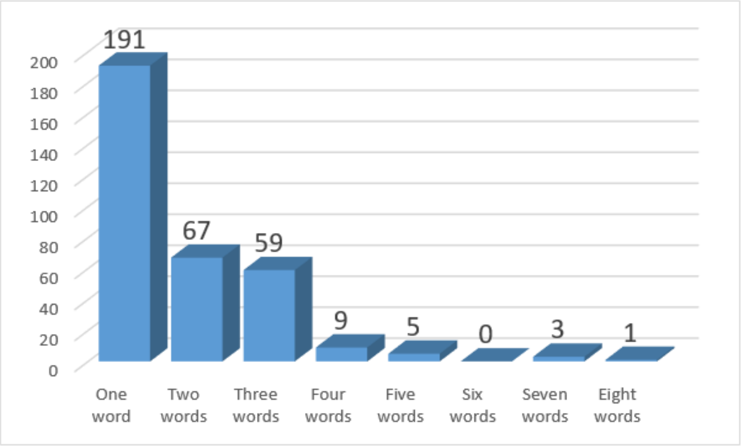


Figure 4.2: The statistics of Antecedent Length (In Terms of Number of Words)

The resulting data set contains 593 pronoun-antecedent pairs where there exist 227 unique antecedents and 56 unique pronouns. The set of 593 pronouns together with their true antecedents are used as positive samples in our experiments. The set of negative samples are built in experiments with different sized windows. The term window here represents some number of tokens that is between the pronoun and its candidate antecedent.

Figure 4.3 shows the total number words observed between pronouns and their true antecedents in different windows sizes. In the figure horizontal axis gives the windows size. For example, in our collection, there exist 250 pronoun-antecedent pairs (positive samples) where 0 to 10 words are observed between the regarding pronoun and the antecedent. In this example 0 to 10 means window size is between 0 and 10. As expected, it is observed that most of the pronoun-antecedent pairs are so close to each other (window size is in range [0 10]).

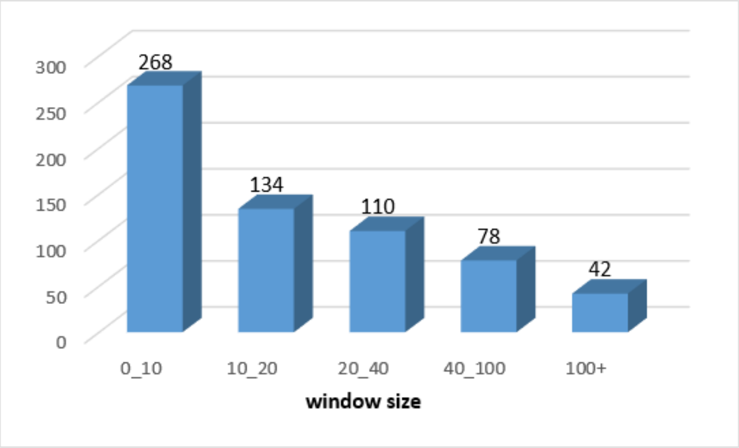


Figure 4.3: Number of words between Pronouns and their Antecedents

In our data set, there are 593 pronoun-antecedent pairs that will be employed as positive samples during classification. In order to train the machine properly, negative examples are also required for train and test processes on deep learning machine. In order to collect negative samples, we used a window-size based approach. Window size (W) is set to 7 different values (W= 1, 5, 10, 15, 20, 25 and 30). For example, if window size set to 15, there will be fourteen negative pronoun-antecedent pairs for each positive pair. All training and testing experiments are re-performed for each W value. In window-based approach, briefly, W-1 nearest tokens preceding the pronoun are employed in negative samples.

The window-sized based approach will be explained briefly with the example in Table 4.2. The example text contains 23 tokens including punctuation. The last token is the pronoun (word no: 23) and its true antecedent is “güllerini” (Token no=17). “onları-güllerini” is set as the positive pronoun antecedent pair. If window size=15, there must be fourteen negative pronoun-antecedent pairs. The negative samples are build by matching the pronoun with all the preceding tokens in the given window size. The resulting pronoun-antecedent pairs for this example are given in Table 4.3.

Token no	Token	POS tag
1	Güller	Noun
2	o	Det
3	kadar	Noun
4	taze	Noun
5	ve	Conj
6	güzellermiş	Verb
7	ki	Conj
8	gören	Adj
9	herkes	Noun
10	perinin	Noun
11	güllerine	Noun
12	hayran	Noun
13	kalırmış	Verb
14	.	Punc
15	Peri	Noun
16	de	Conj
17	**güllerini**	Noun
18	çok	Adj
19	sever	Noun
20	,	Punc
21	her	Det
22	sabah	Noun
23	**onları**	Pron

Table 4.2: Example Text

No	Antecedent	Antecedent POS tag	Pronoun	Sample (Pair) Type
1	güllerini	Noun	onları	Positive
2	gören	Adj	onları	Negative
3	herkes	Noun	onları	Negative
4	perinin	Noun	onları	Negative
5	güllerine	Noun	onları	Negative
6	hayran	Noun	onları	Negative
7	kalırmış	Verb	onları	Negative
8	.	Punc	onları	Negative
9	Peri	Noun	onları	Negative
10	de	Conj	onları	Negative
11	çok	Adj	onları	Negative
12	sever	Noun	onları	Negative
13	,	Punc	onları	Negative
14	her	Det	onları	Negative
15	sabah	Noun	onları	Negative

Table 4.3: Negative and Positive pronoun-Antecedent Pairs

Excluding the positive sample, 14 negative pronoun-antecedent pairs are determined in this window size since there exists less number of words between true antecedent and the pronoun. In cases, where window size is less than the number of tokens between the pronoun-antecedent pair, W number of nearest tokens are employed in negative pairs. The resulting data sets for each W value is given in Table 4.4.

Datas set no	Window Size	Sample size (number of pairs)
1	1	1180
2	5	3474
3	10	6307
4	15	9168
5	20	12061
6	25	14962
7	30	17891

Table 4.4: Window Sizes and Number of Pairs

Chapter 5

Deep Learning in Pronoun Resolution

Within the scope of this thesis, pronoun resolution is accepted as a classification problem to be solved by deep learning machines. In this section, mathematical background on deep learning machines will be given; features used in pronoun resolution will be presented.

5.1 Deep Learning

Donald Hebb is a neurologist who has worked on how the brain learned and also known as the father of the artificial neural networks [7]. His studies began with by taking the neuron, which is the most basic unit of the brain. Hebb examined how two neurons correlate with each other and placed the neural network theory on this basis. This base is implicitly not the only truth. Because even now, how the brain works is explained by the help of some theories. However, with the help of Hebb's studies, this idea has been set in motion and has become appealing to a wide range of people with hundreds of discrete theories today. There are many different models of artificial neural networks that are currently used in real life and expressed with the success rate 99%. All these developed models aimed at solving the problems that are described as unsolvable or np complete in the computer world and even some of these models solve them with a success [11].

5.1.1 Artificial Neural Network

Artificial neural network is a way of modeling the human brain in which all the functions of the layered and parallel structure of neurons are carried out in the numerical world together with all their functions. The numeric world can be specified with hardware and software. In other words, the artificial neural network can be modeled by both hardware and software. In this context, artificial neural networks firstly were tried to be installed with the help of the electronic circuits but these attempts slowly left the field to software area. The reason for such a situation is that electronic circuits cannot be changed flexibly and dynamically, and these different units cannot be brought together [4].

Though the software is a better choice to model the neural networks, artificial neural network models established with electronic circuits will achieve a faster result compared to the models installed with the software. For this reason, artificial neural networks are currently being established, operated and tested, and all necessary changes and dynamic updates are made with software, and decisions are made according to the results of software. If the results can be expressed as 99% and if it is deemed necessary, then the model can be implemented in electronic circuits [4].

Artificial neural network is an information processing system that tries to mimic the biological neural networks. Artificial neural networks have been developed for generalizations of mathematical and numerical models of human cognition and neural biology [11]. As a result, understanding of how biological neural networks works is very crucial is required before understanding the artificial neural networks. Figure 5.1 shows a structure of the biological neuron.

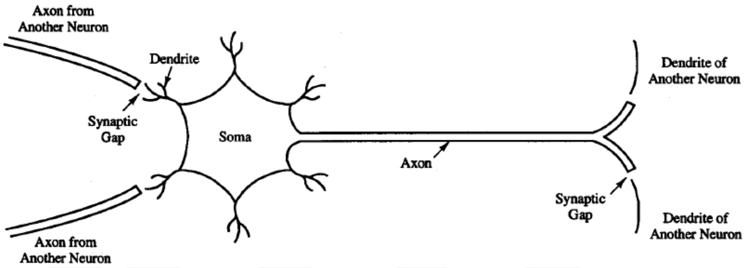


Figure 5.1: Simple Biological Neuron Structure [11]

A biological neuron has four types of components: 1. Dendrites, 2. Soma, 3. Axon and 4. Synapsis. Dendrite's mission is to transmit signals from other neurons to the nucleus of the neuron. Although this structure seems simple, there are up-to-date discussions on the complexity of the dendrites' tasks. There is a different communication between the cell's nucleus and each dendrite. For this reason, it is known that some dendrites have a dominant share in the interaction and others are recessive which means that an important phenomenon such as selectivity in the signals coming from outside is realized by the neuron.

Soma, known as the center and/or the cell nucleus, collects all the signals transmitted through the dendrites. The nucleus transmits the information to the axon to transmit the total incoming signal to the other neuron. Axon is responsible for distributing the total information from the cell nucleus/soma to the next neuron. However, it prevents the transfer of this total signal to the other neuron without preprocessing. Axon transmits the total signal to the unit which is named synapsis.

Synapsis is responsible for transmitting the total information from the axon to the dendrites of the other neurons after being preprocessed. This preprocessing consists of changing the total incoming signal to a certain threshold value. Thus, the total signal is transmitted to the other neurons, reduced to a certain range, rather than as it is. From this point of view, a correlation is created between the sum of each incoming signal and the signal transmitted by the dendrite. The idea that learning takes place in synapses has been put forward and this hypothesis has become a theory for today's artificial neural networks. "Learning" on artificial neural network models is based on this theory and is known as the updating of the coefficients of the weights between synapses and dendrites [4].

Figure 5.2 shows a structure of the artificial neuron that mimics a biological neuron.

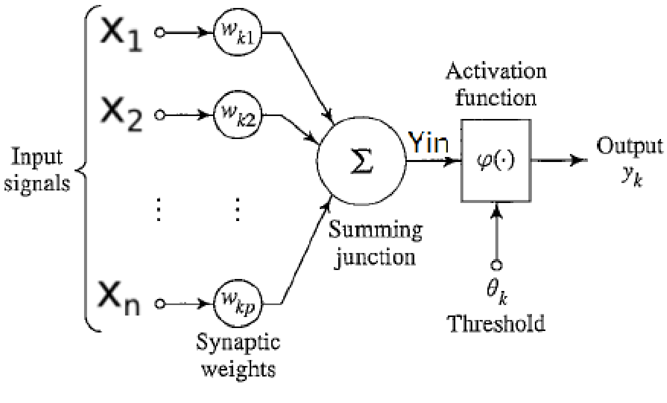


Figure 5.2: The structure of a Simple Artificial Neuron [37]

The dendrites of the artificial neuron shown in Figure 5.2 are indicated by X_n and each dendritic weight coefficient indicated by W_n. Thus, X_n carries the input signals and W_n carries the values of the weight coefficients of those signals. The kernel namely transfer function obtains the weighted sum of all input signals. The summed up signal is indicated by "Y_{in}" and is directed at the input to the synapsis, which has the transfer function. The resultant signal from the transfer function on synapsis is indicated by "*Output*" and directed to feed to the other cell [37].

Briefly, the task of a single artificial neuron is that neuron employs X_n input pattern to generate the "*Output*" signal and transmits this signal to the other cells. The weights W_n representing the correlation between each X_n and "*Output*" are re-adjusted according to each new input pattern and output signal. This adjustment process is called learning. In order to determine the completion of the learning; the input patterns are fed to the system until the change in weight of W_n is stabilized. When stabilization is provided, the cell is accepted to complete the learning. Artificial neural network model is established with the layered structure of at least one artificial neuron. In other words, the term "network" can be used to touch on to any systems of the artificial neurons. A network can range from something as simple as a single node to the very large collection of nodes in which each one is connected to the other every nodes in the created system. Figure 5.3 shows one type of network [15].

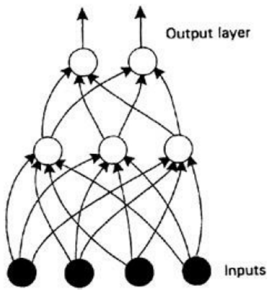


Figure 5.3: Simple Neural Network [15]

5.1.2 Artificial Neural Network Learning

The decision/classification step of artificial learning is known as the activation. [11] The thing to check in activation is whether the sum of the signals entering the neuron has a degree that can activate the regarding cell or not. If the total signal is high enough to fire the cell, it is high enough to exceed the threshold, then the cell is active otherwise the cell is passive. This result is accepted to be also the classification result of the artificial network. In other words; such a network, which can respond by the values 1 or 0 to input patterns, is considered to have decided on whether given input pattern belongs to class 1 or class 0.

During learning, for each new input signal/sample, the last (up-to-date) values of W_n, denoted as the weights of the neurons and shown in Figure 5.2, are employed. The W_n values are adjusted each time with each input signal received. The weighting factors constitute the most determinative points of the geometric pattern in which the regression curve is represented, which minimizes the sum of the distances of all input patterns, which best tries to represent all the input patterns. In this way the system gives optimum answer to taken inputs [4]. Each input value is multiplied with its weight for calculating the activation.

$$a = w_1x_1 + w_2x_2 + ... + w_nx_n \tag{5.1.1}$$

For instance, considering five input signals with weights w_1 = 0.5, w_2 = 1.0, w_3 = -1.0, w_4 = 0, w_5 = 1.2 and the five input values are $x_1 = 1$, $x_2 = 1$, $x_3 = 1$, $x_4 = 0$, $x_5 = 0$. Using the above given formula the activation result is obtained as follows

$$a = (0.5 \times 1) + (1.0 \times 1) + (-1.0 \times 1) + (0 \times 0) + (1.2 \times 0) \tag{5.1.2}$$

To decide on the responding action, a predetermined value -threshold value- is needed such that, if activation value exceeds or equals to it, the node generates the value 1 to the output and if it is less than threshold value then it emits a 0 [15].

In neural networks, the weights (assigned to each input) are re-arranged in each iteration during learning. In other words after each new sample (input pattern) the weights are reconsidered. For example, assume that a network has the initial weights that are set to w_1=w_2=w_1=0.5. Two samples are given to the network sequentially. The first sample has the values $x_1 = 10$, $x_2 = 15$, $x_3 = 20$ and second has x_1=14, x_2=15, x_3=18. If two samples are classified to same group by network, then the weights are changed such

- w_1 will be increased since the second x_1 value is greater than the first one,
- w_2 will not be changed since x_2 values of both samples are equal,
- w_3 will be decreased since the second x_3 value is less than the first one.

$$y_{in} = \sum_{i=1}^{P} X_i \times W_i \tag{5.1.3}$$

As mentioned before Figure 5.1 and Figure 5.2, the X_n signal is added to the summation signal by multiplying it by its own weight. The value in the form of "y_{in}" is sent to the synapse using the axon on the core side. Synapsis gives the output value by thresholding the total signal value [11].

$$y = f(y_{in}) \tag{5.1.4}$$

The function given above may correspond to any mathematical function. At this formula our “y” value can be anything between minus to positive infinitive. At that moment neurons do not know the bounds of the “y”. That is why neuron should have a pattern for fire or not fire. Because of this reason artificial neurons has activation function for to check the “y” value.

Though in literature there exist several different activation functions (e.g. “Relu”, “Than”, “Leakly Relu”, “Sigmoid”, “Hard Than”, “Softmax”, “Identity”, “Softplus”, “Softsign”, ”Threshold”) as some of them are depicted in Figure 5.4 [2], three functions (given in Figure 5.5) are popular on artificial neural network models.

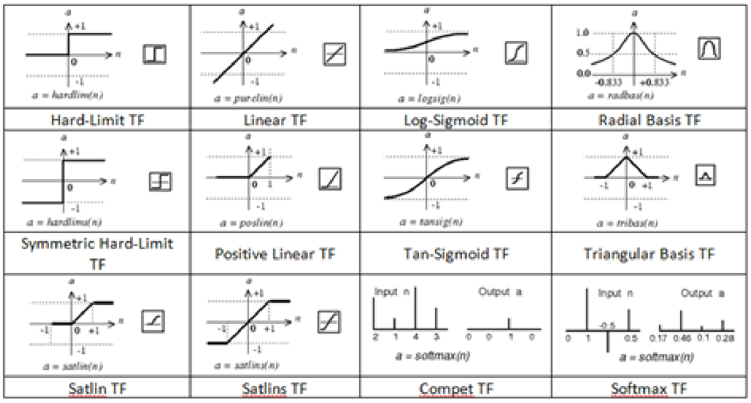


Figure 5.4: Examples of Activation Functions

First of these functions one is the Sigmoid function that provides continuous, ongoing responses to the input pattern. The answers/responses of the function are definitely not separate. Since it is accepted to be the most appropriate function to apply for problems where sensitive assessments are to be used, the sigmoid function has a common use. As an alternative to the sigmoid function, continuity tangent functions or similar functions can be used. The important point in such choice is that the function's derivative can be taken [28].

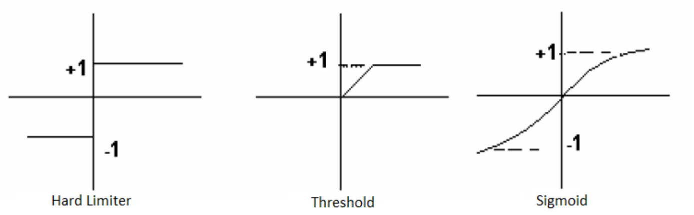


Figure 5.5: Three Popular Activation Functions [4]

The second is the hard limiter/limit function. The step/hard limit function transfer is used to obtain discrete results considering the values of the input patterns. In other words, whatever the input is the output is set either to +1 or -1. Thus a definite limit is provided. If the value entered is less than the threshold value, the result is true, and if the value is greater than the threshold value, the result is false.

The third function is the threshold function. The main aim of function is that input pattern responds to the total value by linearly increasing values up to a certain threshold value. When the upper limit is reached, the answer is now again as definite. It does not show an increasing tendency.

$$W_{i,new}(t+1) = W_{i,old}(t) + (\mu \times [d(t) - f(y_{in})] \times x_i(t)) \tag{5.1.5}$$

Above dedicated formula show that the difference between the response obtained with the thresholding result and the expected value of (y_{in}), which is the expected value of *d(t)*, is multiplied by the learning coefficient μ with the input signal and added to the old weight to determine the new weight. This formula can be express with the delta rule for updating the weights[18] and also with this formula, the process of updating the artificial neural cell to learn it is explained. With this formula, the process of updating the artificial neuron for learning can be described [28]. Further formulization and mathematical details on learning in neural networks may be found in [11].

5.1.3 Deep Neural Network

Artificial neural networks are employed in a wide range of applications from detecting the objects in images, to processing natural language and too many other working areas. In recent years, increasingly these applications of neural networks operate a class of techniques that are commonly named as deep learning

[27]. Deep learning is currently a popular method in image recognition [40], text processing [19], speech recognition [34], and data mining [16]. Though deep learning appears to be a novel approach, its dates back to the 1940s.

If the key historical trends of deep learning are examined, first of all, it is seen that deep learning has a long and rich history and it has become more useful when the amount of available training data is raised. Deep learning models have grown over time at size as infrastructure of hardware and software for deep learning has improved and also it solved increasingly complicated problems and applications with accuracy over time [13].

Deep learning gives changes to computers to build complex and difficult concepts out of simple concepts. For example, Figure 5.6 shows a deep learning system that tries to assign the given image into three classes (person, animal and car) learn an image of a person by combining more simple concepts like corners, circles and also other basic objects in the environment [13].

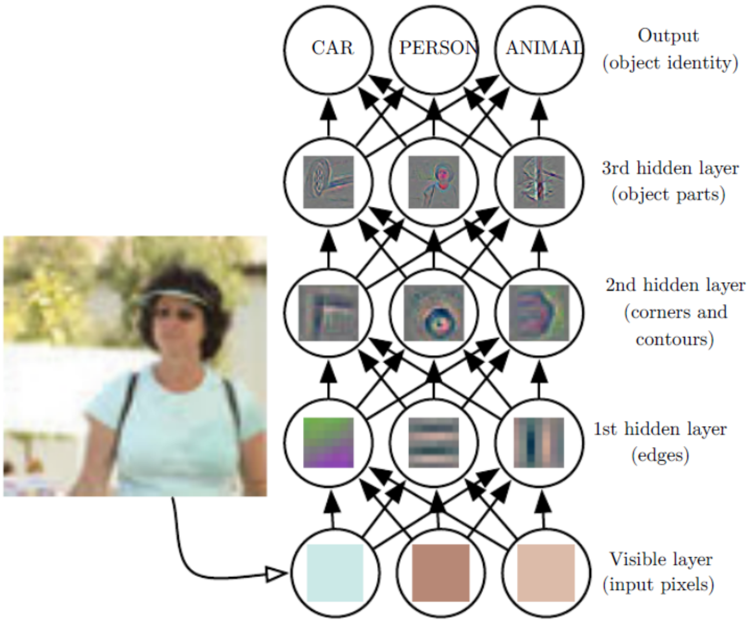


Figure 5.6: An example - Deep Learning Model Example [13]

The aim of deep learning is to extract complex features on high dimensional data in order to use them to build models that relates given input to expected output. It is known that the architectures of deep learning are generally created by multilayer networks that use many hidden layers, that is why abstract features can be computed as non-linear functions of low-level features [39] .

Before examining the multilayer structure of deep networks, the notion of the perceptron should be explained. Perceptron is known as a simple two-layered neural network that contains a few neurons in input layer and one or more than neurons at the output layer. All neurons basically use step function and network based on delta rule which is mentioned in subsection 5.1.2. Figure 5.7 shows the basic structure of perceptron. The network can be used for linear classification and also it can just be applied to linearly separable problems.

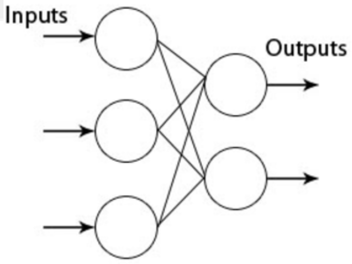


Figure 5.7: Simple Perceptron Structure

The most important factor in the Perceptron model is the threshold value. The threshold value can be determined according to the characteristic of the problem.

A single layer perceptron can solve the logical operations which are *AND* and *OR* because they are known as linearly separable. For example, if two input values *x1* and *x2* as (0, 0), (0, 1), (1, 0), (1, 1) and the output will be (0, 0, 0, 1) when *AND* operation is applied. The output will be (0, 1, 1, 1) when *OR* operation is solved. Because these types of problems are linearly separable problems, perceptron can give solutions to the problems but if a problem such as *XOR* logical operation is to be solved by a single perceptron, it is not possible to obtain the correct results. In order to solve *XOR* problem or similar problems that are linearly inseparable, instead of the single layer approach, multilayer perceptron that requires for at least two layers must be employed [9]. In Figure 5.8, AND, OR and XOR solution spaces are depicted.

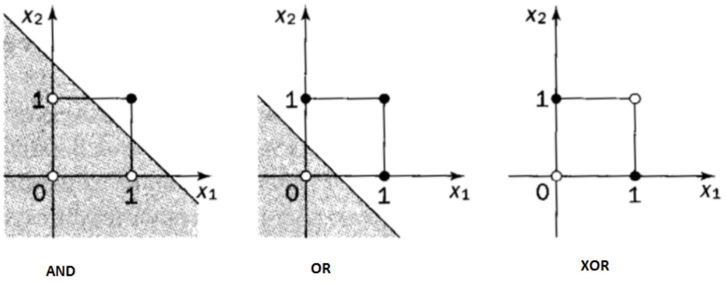


Figure 5.8: AND, OR, XOR Operations

5.1.4 Multilayer Perceptron Neural Network

As mentioned before multilayer neural networks are the mostly used form of the deep learning machines/architectures. These networks are known also as deep feed forward networks. The simplest form of multilayer perceptron is the one that includes a single hidden layer (given in Figure 5.9) [8]. In this architecture, there is exactly one hidden layer between input and output.

There exist also other types of multilayer perceptrons that have multiple layers (known as hidden layer) with nodes in the directed graph. In this type of multilayer perceptrons, every layer is fully connected to the next layer. Each node in the layer has a nonlinear activation function except input layer. Multilayer perceptron uses back propagation algorithm that is a supervised learning algorithm for training the network. Multilayer perceptron is accepted to be a modified version of single layer perceptron that has the ability to distinguish linearly inseparable data.

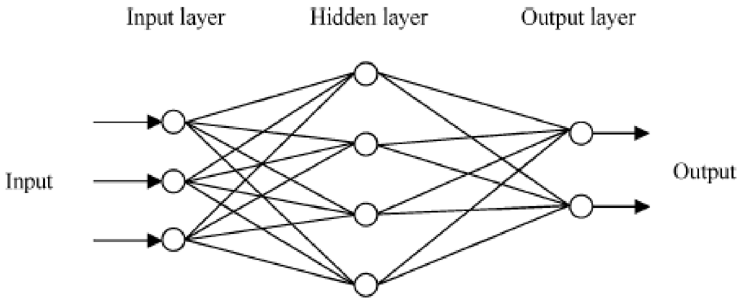


Figure 5.9: Structure of Single Hidden Layer Multilayer Perceptron [8]

A multilayer perceptron is a model which consists of a finite number of sequential layers and each layer occurs with a finite number of neurons. Each neuron of each layer is connected to each neuron of the following and previous layers. These connections generally are called synapsis. The flow of knowledge from one layer to the next is named as feed forwarding in literature. The first layer is the input layer that consists of inputs. In the structure, there can be one or more intermediate layers. The intermediate layers construct the hidden layer(s) in the network. The resulting output is transferred by the last layer (output layer) [32]. Figure 5.10 shows a simple structure of multilayer perceptron.

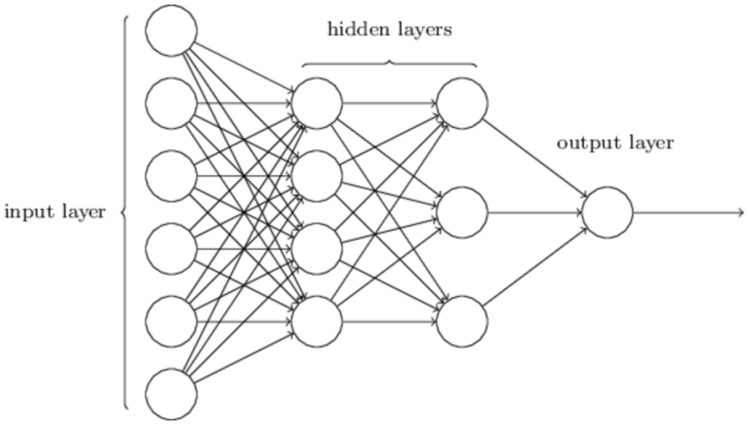


Figure 5.10: Structure of Multilayer Perceptron

Error back propagation algorithm is generally used for training of multilayer feed forward networks. Back propagation recalculates the weights and thresholds in a backwards manner to eliminate the error in multilayer networks. Error patterns are propagated sequentially during back propagation, just as in the simple case of delta learning rule, as examined in previous subsections.

In the task of classification, the trained neural network runs forward. However, the weight settings applied by the learning rules propagate backward from the output layer to the input layer via hidden layers [5]. Generally, the mapping (between the input and output) error is cumulative and it is calculated over the full training set.

5.1.5 Convolutional Neural Network

Convolutional neural network is a category of artificial neural networks that have been recently used efficiently and they reach the state of the art in a lot of different areas like image and video recognition, speech recognition and classification and natural language processing [46]. Convolutional neural networks are very thriving at identifying objects in a given picture or face recognition and it has been applied for many years for handwritten characters and also robots and self-driving cars technologies for powering the vision [38]. It also known as *space invariant* artificial neural network and *shift invariant* at the literature. The convolutional neural network is a variation of multilayer perceptron model which is designed for intended use of least amount of preprocessing.

Convolutional neural networks are additionally known as multilayer neural network with at least one convolutional layer with standard (fully-connected) hidden layers that are suitable for visual processing since they utilize the topology of the inputs. The basic structure of convolutional neural networks is shown at Figure 5.11.

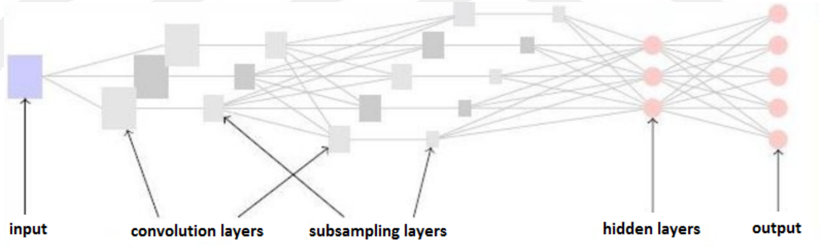


Figure 5.11: Structure of Convolutional Neural Network

Convolutional layers generally come immediately after the input layer. In Convolutional Neural Networks, next to its every convolution layer, the subsampling process may be performed in order to reduce the complexity. Following the convolutional layers, commonly fully connected hidden layers are utilized in the network. All of the layers transmit the output values that are generated by itself to the next layer.

Convolution neural networks are generally preferred for image processing. Input images are spatially divided into an optional number (related to kernel value) of relatively smaller zones in the convolutional neural networks. These zones as a whole cover all the input field. Each zone is mapped to a successive zone in following layers. Usually deeper in the network fully connected layers are utilized as opposed to zones [29]. Convolutional neural networks contain a different number of convolutional and sub-sampling layer and optionally following that fully connected layers. In image processing applications, the given input to convolutional layer is generally a form of A x A x R image where A is the height and width of image and R is depth of image which is also known as number of channels like RGB (red, green, blue).

The focus of this work does not cover images in other words, we work on vectors which have only one channel, height value equal one and width value equals the number of features. Convolutional layer has M kernels or filters with size of B x B x Q where Q is smaller value than dimension of the image and Q can be the same as channel number R besides it can be smaller and vary for every kernel. Throughout the forward feeding, each kernel is convolved against the width and height of the input volume and dot product is calculated between entries of the kernel and input for generating a two dimensional activation map of that kernel. As a result of this network learning, kernel or filter that can be activated when kernel specifies some particular type of features at some special position at the input [21].

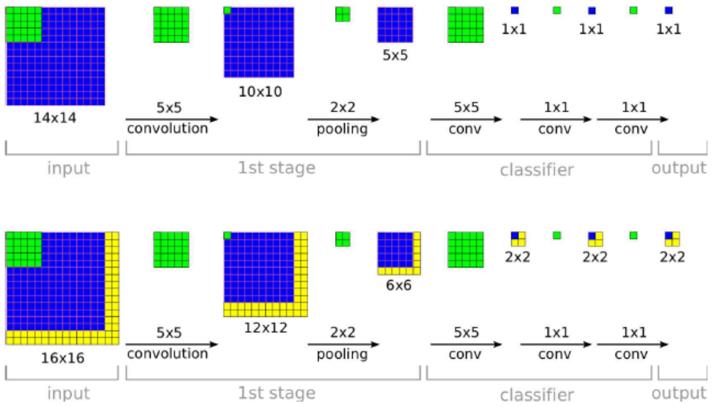


Figure 5.12: Convolutional and Sub-Sampling Layer Example [21]

Convolutional neural network occurs with one or more than one pair of convolutional and sub-sampling layers shown at Figure 5.12. Convolutional layer applies set of filters to small local parts of the taken input where these filters are repeated by sliding window approach with strides [46]. Sub-sampling layer generates a lower resolution pattern of the convolutional layer activations with extracting the maximum value of activations from different positions inside of defined window. This contributes to the invalidity and tolerance of pieces of objects with small differences. Top fully connected layers finally combine inputs from all positions to classify all entries [38]. This hierarchical organization generates good results in convolutional neural networks [1].

At Figure 5.12, convolutional neural network takes 14x14 and 16x16 images as inputs and it converts them to expected outputs. There is a formula for calculating the spatial size of output value as a function of the input value at each layer shown below. "O" is the size of output volume, "W" is the size of input volume, "K" is the size of the filter(kernel), "P" is the padding and "S" is the value of stride.

$$O = \frac{W - K + 2P}{S} + 1 \tag{5.1.6}$$

There are two principal parameters that change and modify the behavior of each layer. After selecting the filter(kernel) size, stride value should be selected. Stride value controls, how filter convolves around the input volume. For the first input image from the Figure 5.12, the size of input value equals to 14 and filter convolves around the input volume with shifting one unit at a time which is referred to stride value equal to 1 and kernel size of convolution layer equal to 5 in given example. This input does not need any padding value which means that zero padding applied. After applying a formula 5.1.6, the expected output value of this convolution layer will be 10. Subsequently pooling layer comes and when given formula applied again at pooling layer with kernel size and stride value equal to 2, the output value will be 5 and the process continues like this way.

In this thesis work, experiments are applied with "deeplearning4j" deep learning tool which is distributed deep learning project in Java and Scala and created by the Skymind company [6]. When multilayer neural network configuration and convolutional neural network configuration are created with this tool for experiments, some of neural network functions and settings should be initialized. General information about some of the terms and meanings frequently used in neural networks are given below.

The term *seed* can be examined like this, when inputs are given the machine for learning progress with train and test parts, machine takes input data line by line but this can negatively affect the learning progress because input data cannot be the form of stirred that is why for deep learning machines, data taken with seeds for to be sure of the input data with shuffle form. The term *epoch* can be explained with one forward pass to data and then one backward pass all of the training data and also the term *batch* size can be determined as the number of training data in one forward and backward pass. The term *iteration* means that all of the given data is passed through once. For example, if training data has 2000 example and the initialized batch size equal 200, then it will take 10 iterations for completing the one epoch. *Optimization algorithm* term is a way of helping the minimizing or maximizing the error function and optimize the gradient descent during the network training. Error function is a mathematical function which depends on the internal parameters on used algorithm model for calculating the target values from model's set of predictors. Algorithm model's internal parameters have very important role at efficient training an algorithm and also generate a correct results.

The term *regularization* is an important component for preventing overfitting. Any change made to the learning algorithm aims to reduce the overall error, but it does not reduce the training error. Regularization allows regulating the weights during the update process. The term *learning rate* is a value which is used by the learning algorithm for determining how rapidly weight values are arranged during the back propagation process. Learning rate detect the time of acquiring for neurons with weights which are trained using an algorithm and also *learning rate policy* is a strategy of how fast the weights are adjusted with different learning rates. Generally in the first iterations learning rate set to high values and then this value gradually reduced for last iterations. The term *weightInit* is initially assigning values to the weights in the neural network. There are several different weightInit functions like previously explained activation functions in artificial neural networks.

5.2 Features in Pronoun Resolution

In classification, machine-learning methods require a number of features that help the machine to distinguish the samples that belong to different classes. We determined 12 features whose values may be obtained from the collection and/or the data sets with different window sizes. In below subsections, the features that are used in pronoun resolution are introduced.

5.2.1 Capital Letter Use in Antecedent (CLU)

During the manual annotation of the collection, it is observed that the proper nouns in text tend to be antecedents for the pronouns. In order to put this observation in use, we accept that the words that begin with a capital letter are commonly proper nouns and are expected to be true antecedents. As a result, CLU is defined as a binary feature that gets the value 1 if the antecedent starts with a capital letter, and 0 otherwise.

5.2.2 Number of Tokens (NT)

NT feature represents the number of tokens between the antecedent candidate and the pronoun in sample pair. The range of NT values is limited by the window size.

5.2.3 Number of characters (NC)

NC feature is the number of characters (excluding blanks) between the antecedent candidate and the pronoun.

For instance, in following sentences for the antecedent candidate “Ali” and the pronoun “Onu”, NC (Ali, Onu)=9.

Ali eve geldi. Onu özlediğimi biliyordu.

(Ali came to home. He knows that I missed him)

5.2.4 Number of unique characters (NUC)

NUC feature represents the number of unique (different) characters (excluding blanks) between the antecedent candidate and the pronoun.

For instance, in example sentences (Ali eve geldi. Onu özlediğimi biliyordu) given in above section, NUC (Ali, Onu)=6

5.2.5 Capital Letter Use between Pronoun and Antecedent (CLUB)

In CLUB feature, the number of capital letters between pronoun and the antecedent are counted. If CLUB value is lower, it is expected that less number of proper nouns is used between the pronoun and the antecedent candidates that leads us to think that the regarding antecedent is the true antecedent.

5.2.6 Number of Nouns (NN)

NN feature represent the number of noun-tagged tokens between the antecedent candidate and the pronoun. It is expected to observe lower NN values between true pronoun-antecedent pairs. For example, in the following sentences for the antecedent candidate "Ali" and the pronoun "Onu", NN (Ali, Onu)=2 (Nouns between "Ali" and "Onu" are "eve" and "çantasıyla").

Ali eve çantasıyla geldi. Onu unuttuğunu bilmiyordu.. *(Ali came to home with his bag. He did not know that he forgot it.)*

5.2.7 Number of Pronouns (NP)

In NP feature, the number of pronouns between pronoun and the antecedent candidate is measured. It is assumed that there exist many pronouns between the constituents of the sample pair; the pair is much closer being negative sample.

5.2.8 Number of Punctuations (NPU)

In this thesis, the use of punctuations between a pronoun and its antecedent candidate is accepted to indicate a negative pair. As the number of punctuations increases, it is expected that the constituents of the pair lose their chance to be connected. As a result, in NPU feature, the number of punctuations observed between the pronoun and antecedent is measured based on the POS tagger results and provided to the learning machine as an indicator of classification.

5.2.9 The Length of Antecedent (LA)

Zipf states that due to the least effort principle in language, people tend to use shorter words more frequently [50]. Grounding from Zipf's claim, it may be stated that the longer words are more prone to be exchanged by their shorter forms, in other words by the pronouns. Starting from this point of view, we decided to employ the length of antecedent candidates as a pronoun resolution feature. LA represents the length of antecedent candidate in characters. To exemplify LA values, In Table 5.1 the antecedents for a pronoun together with their LA values are given.

Antecedent	Anaphor	Antecedent Length
güllerini	onları	9
gören	onları	5
herkes	onları	6
perinin	onları	7
güllerine	onları	9
hayran	onları	6
kalırmış	onları	8
.	onları	1
Peri	onları	4
de	onları	2
çok	onları	3
sever	onları	5
,	onları	1
her	onları	3

Table 5.1: LA Values

5.2.10 Plural Pronoun (PP)

In this thesis, as mentioned before, the results where the classifier matches one of the constituting tokens of a multi-worded antecedent with the regarding pronoun are accepted to be true positives. For example in below sentences, the pronoun "onlar" refers to the multi-word antecedent "Ali ve Ayşe".

Ali ve Ayşe eve geldi. Onlar çok yorgundu.

(Ali and Ayşe came to home. They were so tired.)

Detection of multi-word expressions in text to consider them as a single token is not in the scope of this thesis. As a result, we assume that if any of the candidate pairs ("Ali, Onlar") (ve, Onlar) and (Ayşe, Onlar) is assigned as a pronoun-antecedent pair, it is a success (a hit). PP feature bases on this assumption. PP value is assigned as 1 if the pair includes a plural pronoun since the number of positive pairs for the plural pronouns will be higher compared to singular pronouns. In other words the antecedent candidates that are paired with plural pronouns has a higher chance to be truly referred by the pronoun.

5.2.11 Number Agreement (NUMA)

In number agreement feature, the pronoun and antecedent candidate are examined if they are both in same for in terms of plurality. In order to measure the number agreement of the antecedent candidate and the pronoun, that there are six important cases considered for each sample pair:

Case 1: If antecedent candidate includes "-lar", "-ler" or if it is a number and given pronoun is plural, then NA_1=1; otherwise NA_1=0.

Example: Dün Aliler bize gelecekti, fakat son anda onlar vazgeçti.

Case 2: If antecedent candidate is followed by a conjunction token such as "ve", "ile", "," and given pronoun is plural , then NA_2=0.75; otherwise NA_2=0

Example: Pinokyo ve dedesi çok iyi anlaşılardı. Onlar asla kavga etmezdi.

Case 3: If there is no syntactic or morphological evidence of a plural antecedent but the pronoun is plural, NA_3=0.5; otherwise NA_3=0

Example: Kuş sürüsü havada süzülüyordu. Onlar çok güzel gözüküyordu.

Case 4: If there is no evidence of a plural antecedent and the pronoun is singular, NA_4=0.75 ; otherwise NA_4=0

Case 5: If antecedent candidate includes "-lar", "-ler" and given pronoun is singular, then NA_5=0.25; otherwise NA_5=0

Case 6: If an antecedent candidate is followed by a conjunction token such as "ve", "ile", "," or if it is a number; and given pronoun is singular, then NA_6=0.25; otherwise NA_6=0

After measuring the results of these 6 cases (NA_1 to NA_6), these six values are summed up and divided by 3.5 to be normalized to range [0 1]. The division result is employed as NA value of the pronoun-antecedent pair.

5.2.12 Plural Antecedent (PA)

PA is a feature that represents if the antecedent candidate is plural or a part of a plural expression. The feature is measured considering two cases:

Case 1: If antecedent candidate ends with the one of the strings "-lar" or "-ler", PA_1 is set to 1. If not, the value is assigned to 0.

Case 2: If the token that follows (comes after) antecedent candidate is “ve” or “ile”, PA_2= 1 otherwise PA_2=0.

After applying these cases to every positive or negative pair in prepared data set with different window sizes, there will be four different possible combinations of PA_1PA_2 values which are “00”, “01”, “10” and “11”. If the combination is “10”,”01” or ”11”, candidate antecedent is accepted to be plural and the corresponding PA values is set to 1; otherwise PA=0 is assigned.

Chapter 6

Experimental Setup

In this section, the experimental settings to evaluate the performance of multilayer perceptron and convolutional neural networks will be presented together with the evaluation metrics.

6.1 Experimental Settings

In this thesis work, experiments are applied with "deeplearning4j" deep learning tool, which is a distributed deep learning project in Java and Scala. The tool is built by the Skymind company [6]. As mentioned in Chapter 4, we utilized seven datasets with different number of negative samples. In our deep learning experiments, datasets are divided into two parts: train and test parts. The experiments are performed by 6-fold cross validation technique.

K-fold cross validation is a technique that splits the data set into k subsets. In each iteration, one of the k subsets is used as the testing set and the other k-1 subsets are merged to form a training set. After that, the average error across all k trials is calculated. Figure 6.1 shows train-test sets in 6.1 cross validation.

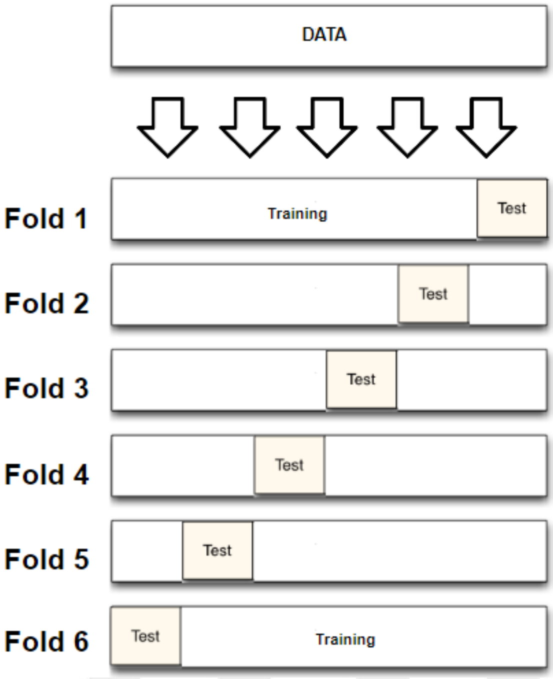


Figure 6.1: 6-Fold Cross Validation

In this study, for each of seven data sets, six training and six testing sets are compiled for the deep learning machine. All the experiments are performed individually for two different deep learning machines (multilayer perceptron neural networks and convolutional neural networks) with number of layers and different number of neurons. In subsections, experimental settings specific two machines will be explained in detail.

6.1.1 Multilayer Perceptron Neural Network Experimental Setup

A multilayer perceptron deep neural network on deeplearning4j tool requires for some adjustments to be done before learning process. First of all, the number of inputs and the outputs of the deeplearning4j machine must be determined. In our problem, input size is set to 12. The other initial values are set as

- the number of iterations=1000,
- the batch size= 10% of input size,
- the seed value=6,
- learning rate of the algorithm =0.1.

As mentioned in Chapter subchapter 5.1, among the wide range of different activation functions in the literature for multilayer perceptron model, "*tanh*" activation function is preferred in our experiments since it is stated that "*tanh*" quickly converges compared to other activation functions and also achieves better accuracy results on multilayer perceptron neural networks [35]. *“Xaiver”* initialization is used to make weight initialization in multilayer perceptron model in the deeplearning4j machine because it keeps the signals in a reasonable range of values through many layers which make sure the weights are just right and it considers the distribution of output activations in point of input activations [12].

There are 2 main configurations on multilayer perceptron deep neural networks. First is the number of layers in deep networks and the second one is the number of neurons in each layer with fixed size layers. To examine the change in performance due to the change in number of layers, we run experiments on multilayer perceptron networks with

1. few number of layers *(Total number of layers =2 hidden + 1 input + 1 output)*
2. medium number of layers *(Total number of layers =7 hidden + 1 input + 1 output)*
3. too many layers *(Total number of layers =18 hidden + 1 input + 1 output)*

Figure 6.2 shows these different type of networks where the number of neurons is fixed.

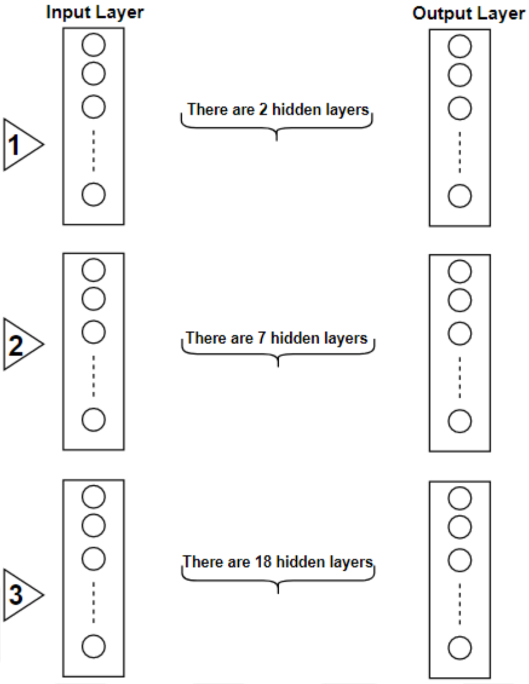
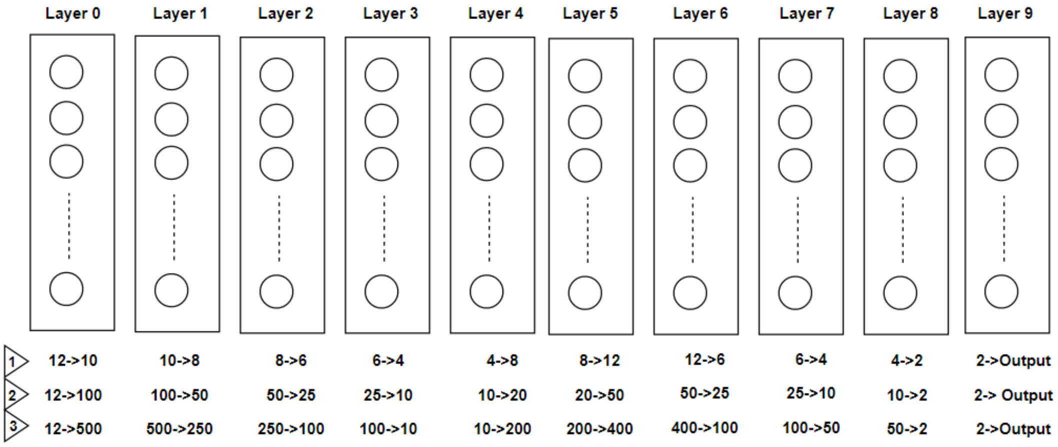

Input Layer
Output Layer
1
There are 2 hidden layers
2
There are 7 hidden layers
3
There are 18 hidden layers

Figure 6.2: Multilayer Perceptron network- Layer Size Configurations

Figure 6.3 shows the different networks generated by changing the neuron size. In these experiments number of layers in network is set to 10. There are three different combinations/settings applied to deep learning machine. These are 10-layered neural networks with

1. small number of neurons,
2. medium number of neurons,
3. too many neurons .

For each combination/setting, input layer takes the input and it firstly reduces the neurons size and then it increases neurons size and then it decreases neurons size again for output layer that is pre-decided to return the result of binary classification. For example, in first setting, the first layer gets 12 input values and generates 10. In second layer, 10 inputs are lowered to 8 output values and so on.



	Layer 0	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Layer 7	Layer 8	Layer 9
1	12->10	10->8	8->6	6->4	4->8	8->12	12->6	6->4	4->2	2->Output
2	12->100	100->50	50->25	25->10	10->20	20->50	50->25	25->10	10->2	2-> Output
3	12->500	500->250	250->100	100->10	10->200	200->400	400->100	100->50	50->2	2->Output

Figure 6.3: Multilayer Perceptron network - Neuron Size Configurations

6.1.2 Convolutional Neural Network Experimental Setup

In this subsection, the experimental settings to run convolutional neural network will be given. Similar to multilayer perceptron network, input size is set to 12 and the other initial values are set as

- the number of iterations=1000,
- the batch size= 10% of input size,
- the seed value=6,
- learning rate of the algorithm =0.01 (In later experiments, it is set as 0.005 and in the last iteration it is equalized to 0.001 with learning rate decay policy which can control the learning rate ratio between different iterations).

In the literature, too many different activation functions are used for convolutional neural network model and it is not certain which activation function to use for which types of problem like multilayer perceptron neural network model. When neural network configuration on deep learning machine is prepared for convolutional neural network, *"relu"* activation function is used at each convolutional layer since it is known to be less computationally expensive compared to than "sigmoid" activation function and *"tanh"* activation function. In addition as mentioned in [24], convolutional neural networks generally have higher time complexities compared to multilayer perceptron neural networks that is why using *"relu"* activation function for convolutional neural networks is a good point to consider when designing deep neural networks. We utilized *"Xaiver"* initialization to make weight initialization in convolutional neural networks in the deeplearning4j machine because it is known to keep the signals in a reasonable range of values through many layers ensuring that the weights are just right and it considers the distribution of output activations in point of input activations [12]. Stochastic gradient descent was preferred as optimization algorithm for the convolutional neural network in this study because it optimizes gradient descent and also it minimizes the loss function throughout network training. There is an additional factor for determining how fast an optimization algorithm converging on the optimum point (known as "momentum"). Momentum affects the direction of the weight adjustment that is why for convolutional neural network's experiments, it can be considered as a weight updater.

Similar to multilayer perceptron experiments, there are 2 main configurations are determined for on convolutional deep neural network experiments. First covers the changes on the number of layers through filter (kernel) values. Secondly, fixing the total layer size, the numbers of neurons are to be changed.

When creating a convolutional neural network with convolutional layers and subsampling layers, kernel and stride values should be pre-calculated for determining the number of layers with previously given formula **??** since when the size of each layer's output volume is calculated with kernel and stride value, it can change the number of layers in the system.

To examine the change in performance due to the change in number of layers, we run experiments on convolutional neural networks with

1. very few layers,
2. few number of layers,
3. medium number of layers,
4. too many layers.

The number of neurons in each layer is same to obtain accurate results for different number of layers. Figure 6.4 shows convolutional neural network with very few layers. There are totally 3 layers in this combination. *Layer 0* which is convolutional layer takes 12 inputs with the kernel value of 12 and stride value equal to 1. Convolutional layer tries to find any correlation between among all of these consecutive 12 values and then it gives 50 outputs with a size of 1 to the fully connected layer. Fully connected layer takes 50 inputs with a size of 1 and it increases the volume of input to the 100 and sent to the output layer for deciding the output value.

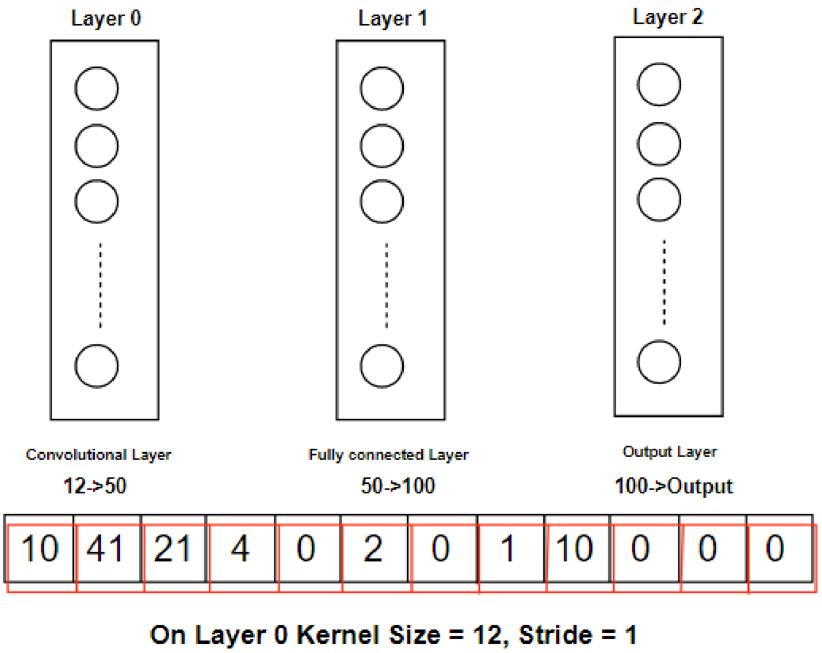


Figure 6.4: Convolutional Neural Network with Very Few Layers

Figure 6.5 shows convolutional neural network with few layers. There are totally 5 layers in this network. *Layer 0* which is convolutional layer takes input value as 12 with the kernel value of 6 and stride value equal to 1. Convolutional layer tries to find any correlation between among all of these consecutive 6 values with seven times. First of all, it takes 10, 41, 21, 4, 0, 2 values and then with one unit shifting it takes 41, 21, 4, 0, 2, 0 values and so on as given in Figure 6.5. In *Layer 1* (max pooling subsampling layer) 50 inputs are received in 7 unit lengths and applying the Equation .5.1.6 with kernel size equal to 2 and stride value equal to 1, it converts these 50 inputs in 7 unit lengths to 50 outputs in 6 unit lengths. *Layer 2* takes inputs in 6 unit lengths with kernel size equal to 6 and stride value equal to 1 then it converts these inputs to 1 unit length outputs for the fully connected layer.

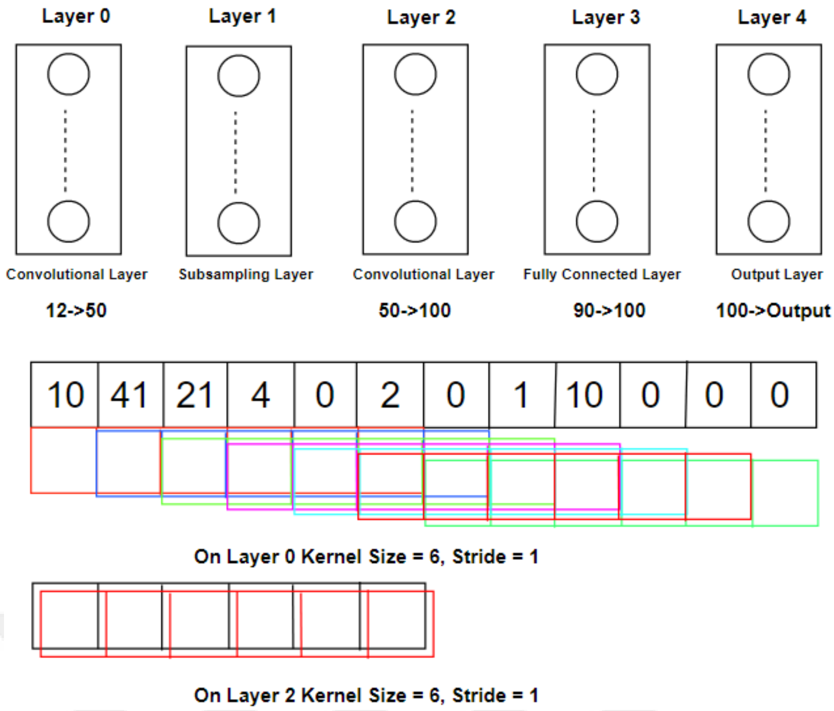


Figure 6.5: Convolutional Neural Network with Few Layers

Figure 6.6 shows the convolutional neural network with medium number of layers. There are totally 8 layers in this combination. *Layer 0* which is convolutional layer takes input value as 12 with the kernel value of 2 and stride value equal to 2. Convolutional layer tries to find any correlation between among all of these consecutive 2 values with six times which is shown at Figure 6.6. In the *Layer 1*, 50 inputs in 6 unit lengths are received and applying the Equation ?? with kernel size equal to 2 and stride value equal to 1, 50 inputs in 6 unit lengths are concerted to 50 outputs in 5 unit lengths for the upcoming layer. Operations on layers continue with same settings for convolutional and subsampling layers until each outputs unit lengths equal to one for giving these outputs to the fully connected layer and then fully connected layer to output layer.

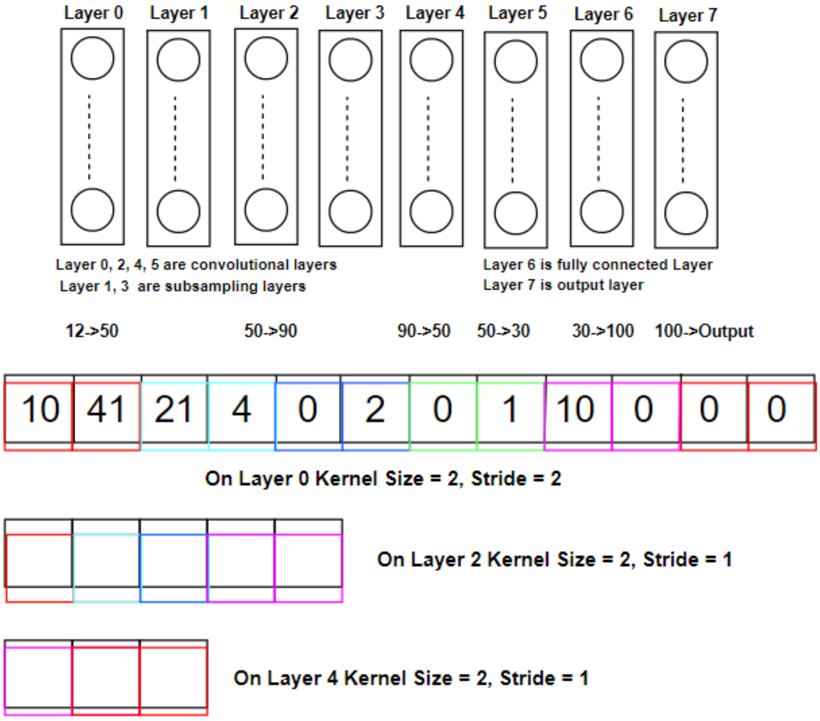

Layer 0
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Layer 0, 2, 4, 5 are convolutional layers
Layer 1, 3 are subsampling layers
Layer 6 is fully connected Layer
Layer 7 is output layer
12->50
50->90
90->50
50->30
30->100
100->Output
10
41
21
4
0
2
0
1
10
0
0
0
On Layer 0 Kernel Size = 2, Stride = 2
On Layer 2 Kernel Size = 2, Stride = 1
On Layer 4 Kernel Size = 2, Stride = 1

Figure 6.6: Convolutional Neural Network with Medium Number of Layers

Figure 6.7 shows the last combination (convolutional neural network with too many layers). There are totally 13. *Layer 0* takes 12 input values with the kernel value of 2 and stride value equal to 1. In all subsequent convolutional and max pooling subsampling layers, kernel value and stride values are same as *Layer 0* since it is desired to reduce the unit lengths of the inputs by 1 unit in each layer for creating a maximum number of layers convolutional neural network with given input volume.

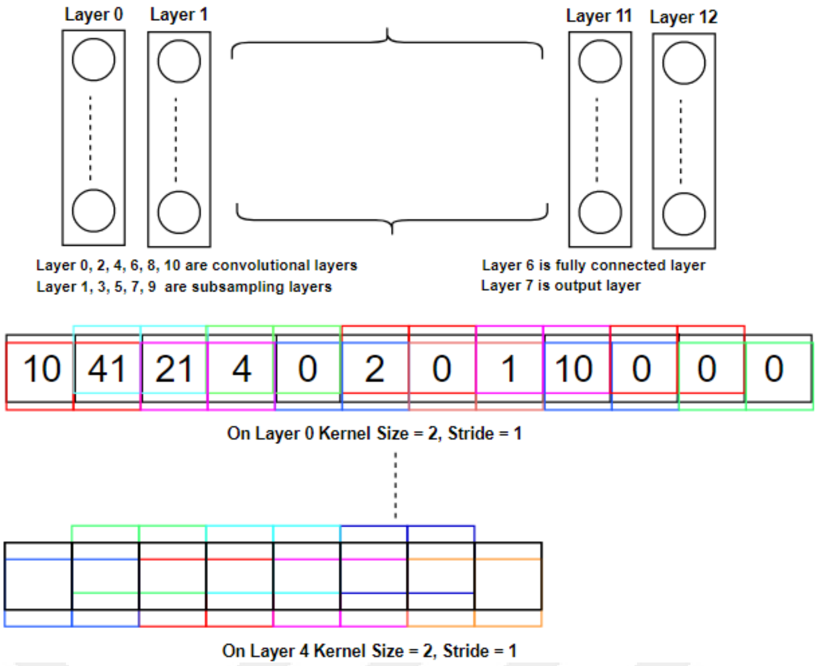

Layer 0
Layer 1
Layer 11
Layer 12
Layer 0, 2, 4, 6, 8, 10 are convolutional layers
Layer 1, 3, 5, 7, 9 are subsampling layers
Layer 6 is fully connected layer
Layer 7 is output layer
10
41
21
4
0
2
0
1
10
0
0
0
On Layer 0 Kernel Size = 2, Stride = 1
On Layer 4 Kernel Size = 2, Stride = 1

Figure 6.7: Convolutional Neural Network with Too Many Layers

Figure 6.8 shows the different convolutional neural networks configurations built by changing the neuron size. These are networks with

1. small number of neurons,
2. medium number of neurons,
3. too many neurons.

The number of layers does not change during the experiments and the number of layers equalized to 7 for each setting. In all settings, Layer 0 is the convolutional layer that takes 12 input values with the kernel value of 2 and stride value equal to 2. In all subsequent convolutional and max pooling subsampling layers, kernel value equal to 2 and stride value equal to 1.

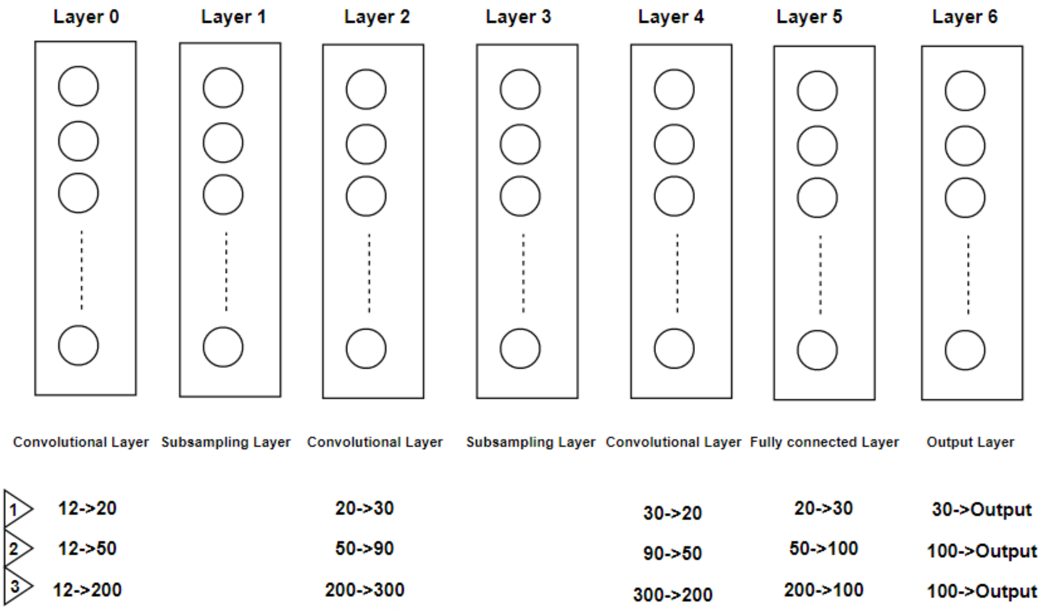


Figure 6.8: Convolutional Neural Network Neuron Size Configurations

6.2 Evaluation Measures

In this thesis, the performance of pronoun resolution with deep learning is evaluated by accuracy, precision, recall and F-measures. Accuracy metric permits measuring the percentage of correct predictions with regard to the entire data set. Accuracy allows one to consider both positive and negative pronoun-antecedent pairs by paying equal attention to all types of error. It is measured as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6.2.1}$$

where TP is number of the true positives (pairs that are classified and observed to be true pairs), TN is the number of true negatives (pairs that are both observed and classified to be false pairs), FP is the number of false positives (pairs that are classified as a true pronoun-antecedent pair incorrectly) and FN is the false negatives (pairs that are actually true pairs but classified as false pairs).

The other two measures are precision and recall. They are given as

$$Precision = \frac{TP}{TP + FP} \tag{6.2.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.2.3}$$

In the statistical analysis, F1-measure (shortly F-score or F-measure) is defined as the combination of the precision and the recall values by calculating their harmonic mean. F-score is used as a weighted average of the precision and recall where F-score owns its highest value at 1 and lowest value at 0. The formula of F-measure is given below.

$$F - measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{6.2.4}$$

Chapter 7

Experimental Results

In this thesis, pronoun-antecedent datasets with different number of samples are utilized to observe the performance of two different deep learning machines (multilayer perceptron and convolutional neural network) with a total of 13 number of settings/configurations. The tests are all performed in a 6-fold cross validation manner as mentioned before.

In following subsections, the experimental results of multilayer perceptron neural network and convolutional neural network will be presented and the weighted values of evaluation metrics (Accuracy, Precision, Recall and F1-Score) will be examined in detail.

7.1 Multilayer Perceptron Neural Network Experimental Results

A total of 6 (3 different number of layers + 3 different number of neurons) number of tests is performed over multilayer perceptron neural network, including all the folds in cross validation. The results of the experiments with different number of layers (few-medium-too many layers) are given in Table 7.1, 7.2 and 7.3.

In Table 7.1, 7.2 and 7.3

1. F1, precision, recall and accuracy columns represent the weighted averages of regarding measures that are obtained from 6-fold cross validation results.
2. W columns represent the window sizes (mentioned before in Chapter 4). Simply, as W values gets higher, number of negative samples in data set increase.
3. The bold values in each column presents the maximum value in regarding metric.
4. The last row gives the average values of evaluation metrics for all window sizes.

W	Accuracy	Precision	Recall	F1 Score
1	**0,975**	0,974	**0,936**	**0,954**
5	0,965	0,969	0,905	0,936
10	0,96	0,976	0,782	0,868
15	0,964	0,98	0,711	0,824
20	0,967	0,983	0,653	0,784
25	0,971	**0,985**	0,619	0,759
30	0,974	0,982	0,602	0,745
Average	0,968	0,979	0,744	0,839

Table 7.1: Experimental Results -Multilayer Perceptron Network with Few Number of Layers

W	Accuracy	Precision	Recall	F1 Score
1	**0,992**	**0,993**	**0,992**	**0,993**
5	0,969	0,974	0,915	0,943
10	0,964	0,966	0,81	0,881
15	0,965	0,952	0,737	0,83
20	0,969	0,964	0,683	0,798
25	0,971	0,969	0,633	0,765
30	0,975	0,96	0,615	0,748
Average	0,972	0,968	0,769	0,851

Table 7.2: Experimental Results -Multilayer Perceptron Network with Medium Number of Layers

W	Accuracy	Precision	Recall	F1 Score
1	0,941	0,954	**0,941**	**0,948**
5	0,967	0,962	0,917	0,939
10	0,963	0,954	0,816	0,879
15	0,965	0,943	0,738	0,827
20	0,967	0,925	0,684	0,785
25	**0,975**	**0,984**	0,671	0,797
30	0,972	0,906	0,62	0,731
Average	0,964	0,947	0,769	0,844

Table 7.3: Experimental Results -Multilayer Perceptron Network with Too Many Layers

The experiments on multilayer perceptron networks with different number of layers show that

1. Considering F1 and recall measures, among different configurations in terms of layer numbers, deep network succeeded most when same number of positive and negative examples (in other word when W=1) are provided.
2. Examining the average evaluation values (last row in tables), it is observed that the maximum F1, accuracy and recall values are obtained when the neural network has the medium number of layers.
3. Considering all tests that utilize multilayer perceptron neural network the highest classification performance is obtained with medium number of layers and equal number of positive and negative samples in data set.

Figure 7.1 depicts F1 curves for multilayer perceptron networks with different number of layers where horizontal axis represents W (window size) value. In Figure 7.1, it is clear that the curve FEW that refers to the F1 curve of network with few number of layers is continuously lower than the alternatives (MEDIUM referring to medium number of levels and TOO_MANY to a large number of layers in network). This result is actually an expected one since it is known that deepening of the multilayer perceptron network (in other words, increasing the number of network layers) improves the performance of network.

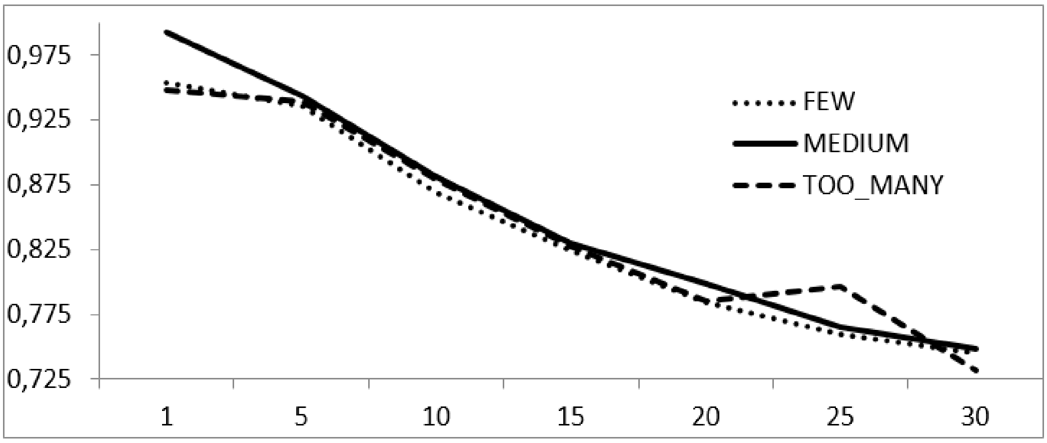


Figure 7.1: F1 Curves for Multilayer Perceptron Networks with Different Number of Layers

The evaluation results of network are given in Table 7.4, 7.5 and 7.6 respectively for few number of neurons, medium number of neurons, too many neurons. The structure of the Table 7.4, 7.5 and 7.6 are similar to the Table 7.1, 7.2 and 7.3 in terms of column names, technical terms etc.

W	Accuracy	Precision	Recall	F1 Score
1	**0,991**	**0,991**	**0,991**	**0,991**
5	0,966	0,952	0,923	0,937
10	0,96	0,946	0,803	0,868
15	0,964	0,941	0,731	0,822
20	0,968	0,947	0,675	0,788
25	0,97	0,95	0,619	0,749
30	0,968	0,968	0,5	0,659
Average	0,969	0,956	0,749	0,831

Table 7.4: Experimental Results -Multilayer Perceptron Network with Few Number of Neurons

W	Accuracy	Precision	Recall	F1 Score
1	0,972	**0,976**	**0,972**	**0,974**
5	0,968	0,963	0,922	0,942
10	0,964	0,967	0,813	0,883
15	0,966	0,953	0,742	0,834
20	0,969	0,963	0,681	0,798
25	0,973	0,961	0,651	0,775
30	**0,974**	0,922	0,618	0,738
Average	0,97	0,958	0,771	0,849

Table 7.5: Experimental Results -Multilayer Perceptron Network with Medium Number of Neurons

W	Accuracy	Precision	Recall	F1 Score
1	**0,991**	**0,991**	**0,991**	**0,991**
5	0,971	0,971	0,922	0,946
10	0,966	0,973	0,818	0,888
15	0,968	0,97	0,744	0,842
20	0,97	0,974	0,686	0,804
25	0,973	0,987	0,653	0,785
30	0,975	0,949	0,62	0,749
Average	0,973	0,974	0,776	0,858

Table 7.6: Experimental Results -Multilayer Perceptron Network with Too Many Neurons

The experiments on multilayer perceptron networks with different number of neurons show that

1. In all settings (except the accuracy when medium number of neurons are employed), the highest scores are obtained when window size is set to one (as given in Figure 7.2) in other words same number of positive and negative samples are provided.

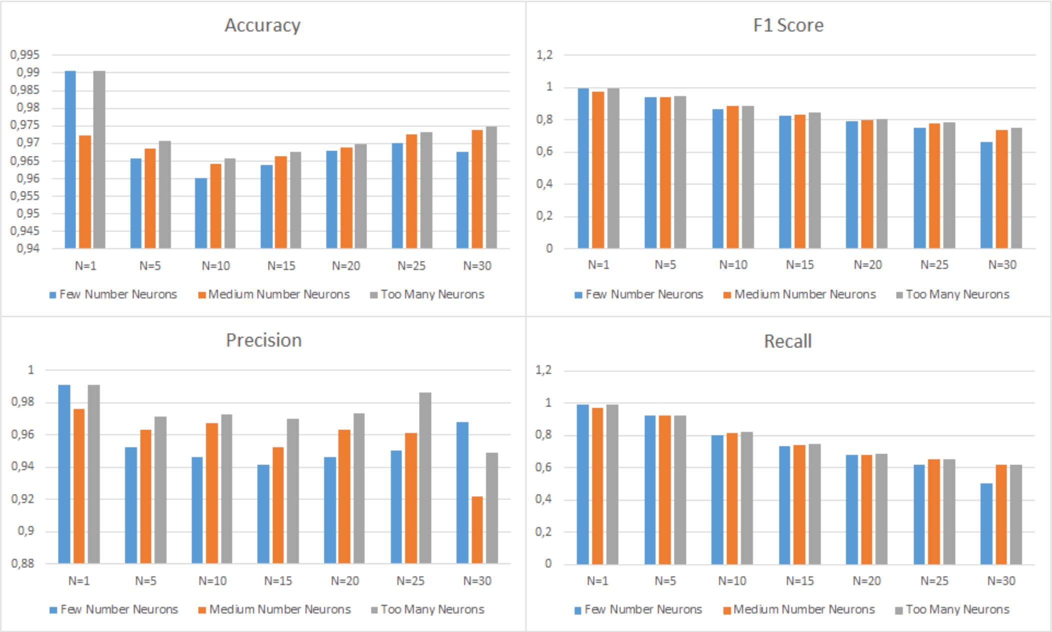


Figure 7.2: The Evaluation Scores of Multilayer Perceptron Network with Different Neuron Sizes

2. Considering the average values of accuracy, precision, recall and F1 scores, it is observed that when too many neurons are employed the system reaches is maximum evaluation scores.

3. Considering the average values of accuracy, precision, recall and F1 scores, it is seen that though the accuracy values are similar in different configurations, F1 value (F1=0.858) is much higher in when too many neurons are used in the system.

4. Examining all experiments with different number of neurons, the highest scores are achieved with too many or few neurons when W is set to one. But to generalize, due to higher scores in alternative W values, it may be stated that the best configuration is obtained with too many neurons.

7.2 Convolutional Neural Network Experimental Results

In this subsection, the experimental results of tests that are applied employing convolutional neural network deep learning machine with different configurations (network with different number of layers or network with different number of neurons) are presented. A total of 7*6 (4 different number of layers + 3 different number of neurons) number of tests is performed including all the folds in cross validation.

The results of the experiments with different number of layers (very few-few-medium-too many layers) are given in Table 7.7, 7.8, 7.9 and 7.10. W in tables represents the window size, the performance scores in columns are all weighted averages of 6-folds, bold values in columns are the highest values for the regarding column. The average rows give the averages of values in columns. In Figure 7.3, the results are given with bar diagrams.

W	Accuracy	Precision	Recall	F1 Score
1	**0,992**	**0,992**	**0,992**	**0,992**
5	0,964	0,966	0,904	0,934
10	0,961	0,972	0,791	0,872
15	0,965	0,98	0,715	0,826
20	0,967	0,98	0,658	0,786
25	0,971	0,985	0,617	0,758
30	0,974	0,987	0,595	0,742
Average	0,97	0,98	0,753	0,844

Table 7.7: Experimental Results –Convolutional Neural Network with Very Few Layers

W	Accuracy	Precision	Recall	F1 Score
1	**0,992**	**0,992**	**0,992**	**0,992**
5	0,963	0,961	0,905	0,932
10	0,96	0,969	0,786	0,868
15	0,965	0,963	0,711	0,817
20	0,967	0,962	0,662	0,784
25	0,971	0,975	0,625	0,761
30	0,974	0,987	0,599	0,744
Average	0,97	0,973	0,754	0,843

Table 7.8: Experimental Results –Convolutional Neural Network with Few Layers

W	Accuracy	Precision	Recall	F1 Score
1	**0,992**	**0,992**	**0,992**	**0,992**
5	0,964	0,963	0,906	0,933
10	0,958	0,954	0,78	0,858
15	0,964	0,962	0,715	0,82
20	0,966	0,942	0,655	0,773
25	0,97	0,94	0,62	0,746
30	0,974	0,968	0,597	0,738
Average	0,97	0,96	0,752	0,837

Table 7.9: Experimental Results –Convolutional Neural Network with Medium Number of Layers

W	Accuracy	Precision	Recall	F1 Score
1	**0,992**	**0,992**	**0,992**	**0,992**
5	0,962	0,956	0,905	0,929
10	0,958	0,957	0,781	0,86
15	0,964	0,958	0,719	0,82
20	0,967	0,96	0,659	0,78
25	0,968	0,909	0,611	0,729
30	0,971	0,917	0,588	0,712
Average	0,969	0,95	0,751	0,832

Table 7.10: Experimental Results –Convolutional Neural Network with Too Many Layers

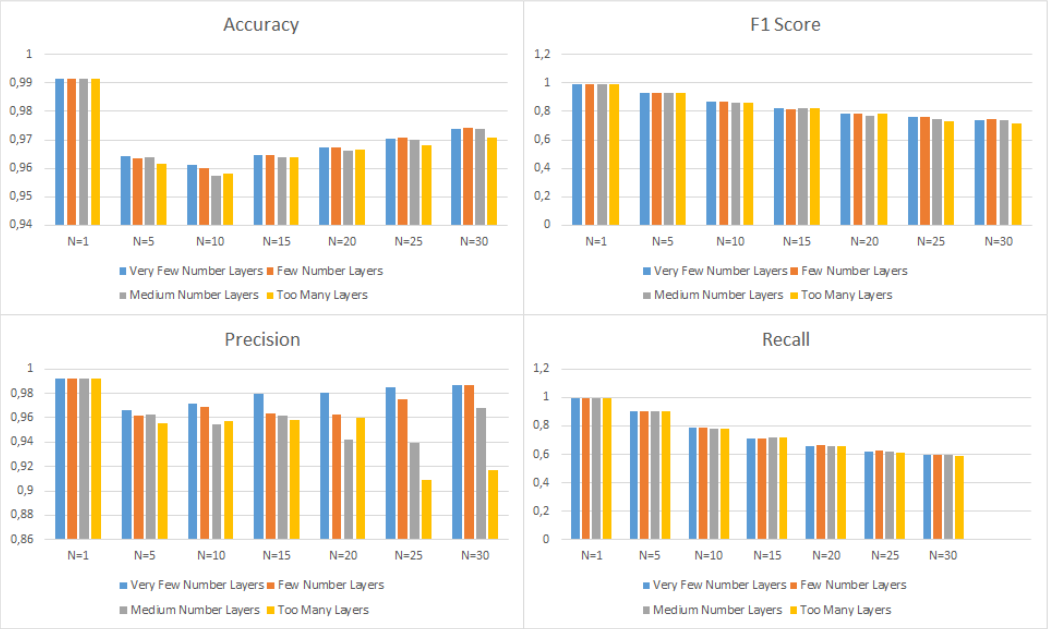


Figure 7.3: The Evaluation Scores of Convolutional Neural Network with Different Number of Layers

The experiments on convolutional neural networks with different number of layers show that considering all evaluation measures the network succeeds most when W=1 in all configurations. In Table 7.11, the average evaluation scores for each configuration that holds different number of layers are given. For example VERY_FEW in Table 7.11 refers to the configuration where very few number of layers are employed. In each column of Table 7.11 the maximum scores are bold. It is observed that no single configuration outperforms the alternatives and the scores are (approximately) equal, especially for accuracy and recall measures. It can only be stated that there is no need for too many layers in convolutional network in the pronoun resolution problem.

Number of Layers	Accuracy	Precision	Recall	F1 Score
VERY_FEW	**0,97**	**0,98**	0,753	**0,844**
FEW	**0,97**	0,973	**0,754**	0,843
MEDIUM	**0,97**	0,96	0,752	0,837
TOO_MANY	0,969	0,95	0,751	0,832

Table 7.11: Average Evaluation Scores for Convolutional Neural Networks with Different Number of Layers

The result of the experiments on convolutional neural network model to observe the affect of different number of neurons are shown in Table 7.12, Table 7.13 and Table 7.14. Similar to Tables 7.8-7.10, W represents the window size; the bold values in columns are the highest values for the regarding column and the average values are given in the last row. Examining the results in Tables 7.12-7.14, it is clearly seen that equal number of positive and negative samples in data set is required to obtain the highest evaluation scores. For all different configurations considering different number of neurons achieved the same maximum evaluation results.

W	Accuracy	Precision	Recall	F1 Score
1	**0,992**	**0,992**	**0,992**	**0,992**
5	0,964	0,962	0,906	0,933
10	0,957	0,952	0,779	0,857
15	0,962	0,944	0,709	0,809
20	0,965	0,955	0,642	0,767
25	0,969	0,94	0,609	0,738
30	0,974	0,975	0,592	0,735
Average	0,969	0,96	0,747	0,833

Table 7.12: Experimental Results –Convolutional Neural Network with Few Number of Neurons

W	Accuracy	Precision	Recall	F1 Score
1	**0,992**	**0,992**	**0,992**	**0,992**
5	0,965	0,966	0,907	0,935
10	0,958	0,957	0,781	0,86
15	0,964	0,963	0,715	0,82
20	0,966	0,936	0,653	0,769
25	0,97	0,936	0,617	0,743
30	0,974	0,965	0,598	0,737
Average	0,97	0,959	0,752	0,837

Table 7.13: Experimental Results –Convolutional Neural Network with Medium Number of Neurons

W	Accuracy	Precision	Recall	F1 Score
1	**0,992**	**0,992**	**0,992**	**0,992**
5	0,965	0,964	0,908	0,935
10	0,959	0,96	0,783	0,862
15	0,964	0,964	0,716	0,821
20	0,966	0,95	0,653	0,773
25	0,97	0,948	0,615	0,746
30	0,974	0,968	0,595	0,736
Average	0,97	0,964	0,752	0,838

Table 7.14: Experimental Results –Convolutional Neural Network with Too Many Neurons

Figure 7.4 provides the bar graphics of the same results in Table 7.12-7.14. Examining the Figure 7.4, it is seen that F1 and recall values decrease, as W value gets higher for different configurations. But not continuous decrease or increase is observed for accuracy and precision values as the dataset size changes.

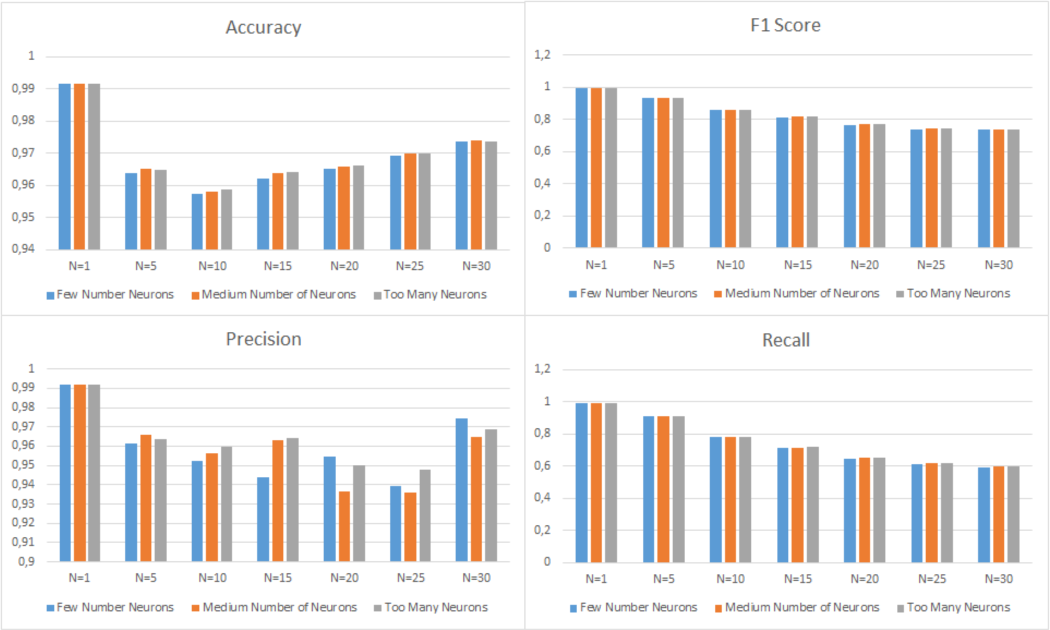


Figure 7.4: The Evaluation Scores of Convolutional Neural Network with Different Number of Neurons

In order to compare the performance in terms of average values Table 7.15 is prepared. In Table 7.15 average scores (from Table 7.12, 7.13 and 7.14) are given for few number of neurons (FEW), medium number of neurons (MEDIUM) and too many neurons (TOO_MANY). Average results extract hat in terms of accuracy and F1 scores, employing medium number of or too many neurons must be preferred instead of few number of neurons.

Number of Neurons	Accuracy	Precision	Recall	F1 Score
FEW	0,969	0,96	0,747	0,833
MEDIUM	0,97	0,959	0,752	0,837
TOO_MANY	0,97	0,964	0,752	0,838

Table 7.15: Average Evaluation Scores for Convolutional Neural Networks with Different Number of Neurons

7.3 Comparison of Multilayer Perceptron and Convolutional Neural Network's Experimental Results

In this thesis two different deep neural networks are employed to determine pronoun-antecedent pairs. In order to compare their performance, F1 and accuracy scores for different configurations will be considered. F1 and accuracy scores (given in this section) are all measured by averaging the scores obtained from data sets of different size. In Table 7.16, F1 and accuracy scores for networks that have different number of layers are given. The average row in Table 7.16 represent the average of the values in regarding column and the bold values represent the maximum values obtained in regarding measure. It is observed that Multilayer perceptron neural network outperforms convolutional network considering two experimental results. These are

1. F1 and Accuracy scores that are maximized when the network is built by medium number of layers
2. The average F1 scores

On the other hand, it is seen that the average accuracy of convolutional network is 0.02 units higher than the average accuracy of multilayer perceptron network.

	Convolutional Neural network		Multilayer perceptron Neural network	
Number of Layer	Accuracy	F1	Accuracy	F1
VERY-FEW	0,97	0,844	-	-
FEW	0,97	0,843	0,968	0,839
MEDIUM	0,97	0,837	**0,972**	**0,851**
TOO_MANY	0,969	0,832	0,964	0,844
Average	0,97	0,839	0,968	0,845

Table 7.16: Average Accuracy and F1 Scores for Both Deep Networks with Different Number of Layers

In Table 7.17, F1 and accuracy scores for networks that have different number of neurons are given. The same structure (notation) is used in table 7.17 with Table 7.16. From Table 7.17, it is seen that the maximum F1 and Accuracy values are obtained by Multilayer perceptron neural network when too many neurons are employed. When the average scores are compared again multilayer perceptron neural network is examined to be better in pronoun resolution problem.

	Convolutional Neural network		Multilayer perceptron Neural network	
Number of Neurons	Accuracy	F1	Accuracy	F1
FEW	0,969	0,833	0,969	0,831
MEDIUM	0,97	0,837	0,97	0,849
TOO_MANY	0,97	0,838	**0,973**	**0,858**
Average	0,969	0,836	0,971	0,846

Table 7.17: Average Accuracy and F1 Scores for Both Deep Networks with Different Number of Neurons

Based on the experimental results, it may be stated that multilayer perceptron neural network with too many neurons and medium number of layers achieved the highest performance. As a result, instead of convolutional neural network that requires for a longer run time, multilayer perceptron neural network may be employed in pronoun resolution problem in Turkish.

Chapter 8

Conclusion and Future Study

In this thesis, we accept pronoun resolution as a binary classification problem. The feature values of Turkish pronoun-antecedent pair candidates are provided as inputs to deep learning machines. The performance of deep learning machines in solving the regarding problem is investigated. In order to run the experiments, a collection of ten child stories is compiled and a data set of 593 positive pronoun-antecedent pairs is prepared. The data set is enlarged with different number of negative samples to obtain data sets with different sizes and annotated manually. 12 features (e.g, capital letter use in antecedent, number of tokens, number of characters) are determined to be used in deep learning.

The performance of two deep learning machines, multilayer perceptron neural network and convolutional neural network, is measured considering two different configurations. In first set of experiments, the number of layers are varied in networks. The performances are measured for (very) few, medium and high number of layers. In second set, number of neurons in each layer is set as few, medium or high. The evaluation is performed by four different metrics (accuracy, recall, precision, F1 score). It is observed that multilayer perceptron neural network with a medium number (9) of layers that employ too many neurons gives the highest performance results.

As a future study, we plan to enlarge the data set, implement other deep neural networks and enrich the feature set employed in this study.

Appendices

Appendix A

Not-Referring Anaphora Example Sentences

1. Her sabah ona küçük sarı tomurcuk büyüyecek , kocaman güzel bir gül olacak diye güzel sözler söylüyormuş . Tomurcuk da bunu anlıyormuş gibi günden güne daha da güzelleşerek büyümüş .[bunu]

2. O kadar güzelmiş ki onu görenler sarı güle bakmaya doyamıyorlarmış . Peri de bunun farkındaymış ve çok mutluymuş .[bunun]

3. Ama o , ne konuşabiliyor ne de şarkı söyleyebiliyormuş .[ne]

4. Ama hiçbiri yeryüzünü görmek için onun kadar sabırsızlanmıyormuş .[hiçbiri]

5. Orada bulunan diğer kızlar prensin ve kral ailesinin önünde şarkı söylemişler .İçlerinden biri diğerlerinden daha güzel şarkı söylüyormuş .[biri]

6. Gemiden kopan kalaslar ve direkler azgın dalgalara karışıyor küçük deniz kızına zor anlar yaşatıyormuş . Tahtalar çarpabilir hatta ezilebilirmiş . Ama bunların hiçbirini düşünecek durumda değilmiş .[hiçbirini]

7. Geceyi nerede geçirirsen geçir ![nerede]

8. Kim oluyorlardı da onun çapında birine gülüyorlardı ?[kim]

9. Bütün gün bunu yaptığı için hiç müşterisi kalmamış zavallı adamın .[Bunu]

10. Böyle neşeli neşeli nereye gidiyorsun ?[Nereye]

11. Ne nereye gittiğini öğrendiler , ne de neler yaptığını duydular .[neler]

12. Sesin nereden geldiğini anlamak için başını çevirmiş .[Nereden]

13. Pinokyo içeride olanları çok merak ettiğinden , Geppetto ustanın okula gitmesi için verdiği parayı uzatmış .[Olanları]

14. Güller o kadar taze ve güzellermiş ki gören herkes perinin güllerine hayran kalırmış .[o]

15. Oturdukları beyaz evin bahçesi öyle güzel çiçeklerle bezeliymiş ki , kokuları siz deyin on mahalle , ben diyeyim yirmi mahalle öteden duyulurmuş .[siz]

16. Oturdukları beyaz evin bahçesi öyle güzel çiçeklerle bezeliymiş ki , kokuları siz deyin on mahalle , ben diyeyim yirmi mahalle öteden duyulurmuş .[ben]

17. Kimin iyi , kimin kötü olduğunu ise bilebilmek pek zormuş .[Kimin]

18. O günden sonra da kiminle karşılaştıysa , saçının tellerini yaşmağının ucundan gösterip birşeyler geveler , birşeyler anlatmak istermiş .[kiminle]

19. Ne dersiniz buna siz ?[buna]

20. Geceyi nerde geçirirsen geçir ![nerde]

21. Kim oluyorlardı da onun çapında birine gülüyorlardı ?[birine]

22. Fakat ondan sonra evin içinde bir karışıklık olmuş .[ondan]

23. O sokak senin , bu sokak benim dolaşıyorlarmış .[senin]

24. O sokak senin , bu sokak benim dolaşıyorlarmış .[benim]

25. Şunu hiç aklından çıkarma :[Şunu]

26. Bir kendini bilmez yanında getirdiği şişenin içindekini içmiş , giderken de atmış şişeyi kırmıştı .[kendini]

27. Pulları gümüş gibi parlak , gözleri cam gibi aydınlık , güzel mi güzel bir balıkmış bu. . . [bu. . .]

Appendix B

Referring Anaphora Example Sentences

1. Yaşlı bir kurbağa ayağından yaralanmış az ilerde yatıyor . Ne olur benimle gelin ona yardın edin , onu kurtarın .[onu]

2. Pinokyo başına gelenlerin kendi suçu olduğunu Geppetto ustanın sözünü dinleyip okula gitse bunların hiçbirinin olmayacağını düşünerek , ağlamaya başlamış .[Kendi]

3. Padişah hayretler içinde kalmış . Acaba bu insanlar delirmiş de benim mi haberim yok , diye kendi kendine sorar olmuş .[kendine]

4. Bu yüzden , genç peri sarı tomurcuğa daha özenli bakmaya başlamış . Her sabah ona küçük sarı tomurcuk büyüyecek , kocaman güzel bir gül olacak diye güzel sözler söylüyormuş .[ona]

5. Küçük kız , gemiye yaklaşmış . Dalgalar onu yükseltince de yuvarlak pencerelerden içerisini görebilmiş .[onu]

6. Prens biraz kendine gelir gibi olmuş . Ama gözleri hala kapalı , yüzü ise solgunmuş . Küçük kız onun güzel ve geniş alnını öpmüş .[onun]

7. Baksana beş ördek yarışıyor , taş çatlasa elli ördek onları alkışlayıp gayrete getirmeye çalışıyor .[onları]

8. Peri de güllerini çok sever , her sabah onları hem sular hem de onlarla konuşurmuş .[Onlarla]

9. Gadro , arkadaşları oyun oynarken tek başına antrenman yapmış , hırsla kendini büyük bir şampiyon olacağım diyerek yetiştirmişti .[Kendini]

10. Yemyeşil kıyıların önünde büyük bir bina yükseliyormuş . Burası eski bir şatoymuş .[Burası]

11. Bunun içinde bir kurbağa vardı ve o kurbağa da kendisine bakıyordu .[kendisine]

12. Kız başından geçenleri yaşlı adama anlatmış . Geceyi geçirmek için ondan bir yer istemiş .[ondan]

13. Kızcağız bütün gün ormanda dolaşıp durmuş . Akşam olunca o da yaşlı adamın evine varmış .[o]

14. Prens , küçük deniz kızına : - Ne kadar mutluyum . Onu bulduğuma inanamıyorum .Benim mutluluğum seni de sevindirsin , demiş .[seni]

15. Prensin başını devamlı suyun üstünde tutmaya çalışmış . Kendini onunla birlikte suyun akışına bırakmış .[Onunla]

16. Ablaları hançeri küçük kıza uzatıp : - Bu hançeri güneş doğmadan prensin kalbine sapla . Kanı senin ayaklarını ıslatınca tekrar deniz kızı olabileceksin .[senin]

17. Ablaları hançeri küçük kıza uzatıp : Bu hançeri güneş doğmadan prensin kalbine sapla . Gün doğmadan önce ikinizden birinin ölmesi gerek .[ikinizden]

18. Eski durumuma dönmem için yalnızca insanlara değil ; hayvanlara da iyilik etmeyi seven , temiz yürekli bir kızın yanıma gelmesi gerekti . İşte bu kız sen oldun .[sen]

19. İşte tam aradığım gibi bir kütük . Bununla çok güzel bir kukla yapacağım , diye sevinerek kütüğü sırtladığı gibi oyuncakcı dükkanına taşımış .[Bununla]

20. Bu kez bezelye götüreceğim . Yollara serpeceğim . Bunlar mercimekten daha iridirler .[Bunlar]

21. Bu padişah çok uzak memleketlerin birisinde yaşıyormuş . Bu ülke öyle uzakmış ki , oraya varmak için yüz tane dağ , elli tane ova , beş - yüz tane de ırmak geçmek gerekiyormuş .[oraya]

22. Adam geri geldiğinde yüzündeki ifade değişmişti . Kuşu çağırdı , Bu adamı nereden getirdiysen oraya götür . Ben böyle bir evlat istemiyorum . dedi .[Ben]

23. Adam hayvanlara seslenmiş güzel tavuk , güzel horoz , alacalı güzel inek ! Ne dersiniz buna siz ?[siz]

24. Ne zaman birkaç orman hayvanını bir arada görüp yanlarına gitmeye kalksa huzursuzluğu çoğalıyordu . Çünkü onlar Gadro'ya sıradan biriymiş gibi davranıyorlar , bazı konularda ileri sürdüğü fikirlere gülüp geçiyorlardı .[Onlar]

25. Hayvanlar hep bir ağızdan bizce uygun demişler .[bizce]

26. Keloğlan onu tutmak için eğilince kendisi de ırmağa yuvarlanmış .[Kendisi]

27. Yanıma bir torba mercimek alıyorum . Taneleri darınınkinden iridir . Kız bunları daha iyi görür , yolunu şaşırmaz ![bunları]

28. Yarışmacıların hepsinin üstünde Gadro'nun emeği vardı . O , gece gündüz demeden kendilerini bu yarışa hazırlamıştı .[kendilerini]

29. Tam kanadı yiyecekken kedi konuşmaya başladı : Bana o kanadı verirsen , karşılığında sana yüz tane altın veririm .[Bana]

30. Daha sonra düzenlenen yarışmaya kadar Gadro , genç ördeklere gölde antrenman yaptırdı . Onların iyi birer yarışmacı olmaları için sonsuz gayret gösterdi .[Onların]

31. Padişah hikayelerin hepsini dikkatle dinlemiş , adamlara acımış . Hemen onlara hazineden para verdirmiş .[onlara]

32. Keloğlan çok şaşırmış . Bir kaç kere denemiş , hep altın akıyormuş tastan . Bu , sihirli bir tas galiba . Hemen anama haber vereyim demiş .[Bu]

33. Parlak şeye baktığında çok şaşırdı . Bunun içinde bir kurbağa vardı ve o

kurbağa da kendisine bakıyordu .[Bunun]

34. Afiyetle yiyin sevgili hayvanlar ! Susadığınız zaman içersiniz diye size serin su da getireyim demiş .[size]

35. Çocuklar gittikten sonra kurbağacık yaşlı kurbağaya destek oldu ve onu kuytu bir yere götürdü . Burada yaşlı kurbağa , kurbağacığa yaptığı yardımlardan dolayı teşekkür ettikten sonra :[Burada]

36. Yatakta doğrulmuş ben bir prensim demiş , kötü bir cadı beni ak saçlı , ak sakallı bir yaşlı kılığına sokarak ormanda yaşamaya zorlamıştı .[beni]

37. Paraları gören kurnaz tilki ve kedi bir oyun oynayıp bu paraları almaya karar vermişler . Pinokyo'ya : – Okula gidip de ne yapacaksın ? Bizim dediklerimizi yaparsan zengin olursun .[Bizim]

38. Hayvanlar hep bir ağızdan bizce uygun demişler .Yaşlı adam kıza dönerek burada her şeyden bol bol var ! Haydi ocağa git , bize akşam yemeği pişir demiş .[bize]

39. Paraları gören kurnaz tilki ve kedi bir oyun oynayıp bu paraları almaya karar vermişler . Pinokyo'ya : – Okula gidip de ne yapacaksın ? Sen o paraları bize ver , biz de götürüp sihirli tarlaya ekelim .[biz]

40. Bir güzel dualar etmiş ki kadın oturduğu yerden , Bukle ve Menzile pek sevinmişler . Menzile evin yoksa kal bizimle , yoldaş olursun bize demiş .[bizimle]

41. Padişah hayretler içinde kalmış . Acaba bu insanlar delirmiş de benim mi haberim yok , diye kendi kendine sorar olmuş .[benim]

42. Biraz sonra kırk elli ördeğin göl kıyısına gelerek , bunlardan ayrılan beş ördeğin göle girip birbirleriyle yarıştıklarını gördü .[bunlardan]

43. Yekta ay ışığı altında , yavaş bir tempo tutturmuş olarak kilometrelerce koştuktan sonra birden ürperdi . Sol tarafında bir karartı vardı ve kendisini geçmeye çalışıyordu .[kendisini]

44. Pinokyo'nun şarkı söyleyerek yürüdüğünü gören kurnaz tilki ve arkadaşı kedi Bu kukla ne kadar da neşeli , şunun bir yanına gidelim diyerek Pinokyo'nun önüne çıkmışlar .[şunun]

45. Pinokyo da : – Kendime defter kalem alıp okula gideceğim , demiş .[Kendime]

46. Geppetto usta , karşısında Pinokyo'yu bu şekilde görünce dünyalar onun olmuş . En sonunda benimde gerçek bir oğlum oldu diyerek sevinç gözyaşları içerisinde oğluna sarılmış .[benimde]

47. Ey padişah kızı , bu gece sana uzun bir masal anlatacağım .[sana]

48. Hayvanlar seslenmişler onunla yedin içtin bizleri düşünmedin .[bizleri]

49. Çırağımla bu tavukları eve gönderdim . Çırağa , Hemen ikisini de pişirsinler .[ikisini]

50. Onu hemen durdurup iki tane tavuk satın aldım . Çırağımla bu tavukları eve gönderdim . Çırağa , Hemen ikisini de pişirsinler . Birini kendileri yesin

, diğerini de bana göndersinler . [Birini]

51. Adam altınların yarısını teklif etti , ama kabul etmedim . İlle de hepsi olacak diye tutturmuştum . [hepsi]

52. Padişah hikayelerin hepsini dikkatle dinlemiş , adamlara acımış . [hepsini]

53. "Demircinin hikayesini dinledikten sonra sıra bahçıvana gelmiş . O da başına gelenleri şöyle anlatmış : - Bir sabah meyveleri toplamak için bahçeye girdim . Elma ağacının başına çıkmış bir bir meyveleri topluyordum . Bu sırada tam karşımda duran çok güzel bir kuş gözüme çarptı . Daha önce böylesine güzel bir kuşu hiç görmemiştim . Kuşu yakalamak için elimi uzattım , fakat o daha hızlı davrandı ve beni yakaladığı gibi havalandı . Bir süre uçtuktan sonra kocaman bir gül bahçesine indik . Daha önce bu kadar güzel bir gül bahçesi de görmemiştim . Güller öyle güzel açmıştı ki , o renkte güllerin varlığını bile bilmiyordum . Akılım başımdan uçtu gitti . Bahçede deli - divane gezinirken bir ihtiyar çıktı karşıma . Beraberce bir köşeye oturduk . Benimle konuşmaya başladı : [Benimle]

54. "Tüccar karşı çıktı : İşim çok acele , durmadan devam etmeliyiz .Fakat ben onu dinlemiyordum . Seni öldüreceğim ve bütün altınlar benim olacak . diyordum adama . Adam altınların yarısını teklif etti , ama kabul etmedim . İlle de hepsi olacak diye tutturmuştum . Hem adamı bırakırsam beni şikayet etmesinden korkuyordum . Gözüm hiçbir şey görmüyordu . Bu kadar kötü kalpli olduğumu ben de bilmiyordum . Meğer öyleymiş . Tam elimdeki bıçağı saplayacaktım ki , adam beni durdurdu . Dur dedi . Bende

bir sürme var . Göze sürüldüğü zaman toprak altında ne kadar hazine varsa hepsi görülüyor . [Bende]

55. Dünya bu kadar küçücük mü sanki ? Neden kurtarmazsın kendini buradan , çekip gitmezsin buralardan ? [buradan]

56. Yekta tüm çabasına karşılık ikinci sırada kalmıştı . Tüh be , Tavşan'ı kaçırdım ! Bu Tavşan'ı zaten son düzlüğe kadar kimse geçemezmiş . Yarışın ortasına gelmeden onu mutlaka geçmeliyim . Haydi Yekta , daha hızlı , daha hızlı... 1500 startı geçildiğinde Tavşan ikinci durumdaki Yekta'nın üç boy kadar önündeydi . Bomba nerelerde ki , dönüp bakmalı . Tavşan bu süratiyle yarışı tamamlayamaz . Vay , Bomba hemen arkamdaymış ! Ne oluyor ya , ne dümen çeviriyor bunlar ? Son düzlüğe kadar orta sıralarda saklanırmış bu . Benden huylandılar muhakkak . [Benden]

Chapter 9

Bibliography

[1] Ossama Abdel-hamid, Hui Jiang, and Gerald Penn, *APPLYING CONVOLUTIONAL NEURAL NETWORKS CONCEPTS TO HYBRID NN-HMM MODEL FOR SPEECH RECOGNITION Department of Computer Science and Engineering , York University , Toronto , Canada Department of Computer Science , University of Toronto , Toronto , Canada*, Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on (2012), 4277–4280.

[2] Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vi, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan

Yu, and Xiaoqiang Zheng, *TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems*, (2015).

[3] Chinatsu Aone and Sw Bennett, *Evaluating automated and manual acquisition of anaphora resolution strategies*, Proceedings of the 33rd annual meeting on . . . (1995), 122–129.

[4] M.G Cinsdikici, *Neural Network Solutions for ATM Routing & Multicasting Problems*, (1997), no. 1949.

[5] Geri Davis and Barbara Farabaugh, *Introduction to Art if icial Neural Sy Indexing :.*

[6] Deeplearning4j.org, *Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org.*

[7] D.O.Hebb, *The Organization of Behavior a Neuropsychological Theory*, (1949).

[8] Peter A Dowd and Eulogio Pardo-ig, *Estimating the boundary surface between geologic formations from 3D seismic data using neural networks and geostatistics*, **70** (2005), no. 1, 1–11.

[9] D Elizondo, *The Linear Separability Problem: Some Testing Methods*, 1–13.

[10] Tın Erkan and Varol Akman, *Situated Processing of Pronominal Anaphora*, Verarbeitung natürlicher Sprache (KONVENS) (1998), 369–378.

[11] Laurene Fausett, *Fundamentals of Neural Networks*, Igarss 2014 (2014), no. 1, 1–5.

[12] Xavier Glorot and Yoshua Bengio, *Understanding the difficulty of training deep feedforward neural networks*, Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) **9** (2010), 249–256.

[13] Ian Goodfellow, *Deep Learning.*

[14] Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi, *Centering: A Framework for Modelling the Local Coherence of Discourse Centering: A Framework for Modelling the Local Coherence of Discourse*, Computational Linguistics **21** (1995), no. 2, 203–225.

[15] Kevin Gurney, *An introduction to neural networks An introduction to neural networks.*

[16] Robert L. Hale, *Cluster analysis in school psychology: An example*, Journal of School Psychology **19** (1981), no. 1, 51–56.

[17] Jakob Hamann and Jakob Hamann, *On the Syntax and Morphology of Double Agreement in Lavukaleve*, (2010), 197–225.

[18] Robert Hecht-nielsen, *the Backpropagation Neural Network*, 593–605.

[19] Naacl Hlt, *Tutorial Abstracts*, Services - Part I, 2008. IEEE Congress on (2008), –.

[20] Jerry R. Hobbs, *Resolving pronoun references*, Lingua **44** (1978), no. 4, 311–

338.

[21] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen, *Convolutional Neural Network Architectures for Matching Natural Language Sentences*, Advances in Neural Information Processing Systems 27 (2014), 2042–2050.

[22] Christopher Kennedy and Branimir Boguraev, *Anaphora for Everyone : Pronominal Anaphora Resolution without a Parser*, Proceedings of the 16th International Conference on Computational Linguistics, 1996, pp. 113–118.

[23] Yilmaz Kiliçaslan, Edip Serdar Güner, and Savaş Yildirim, *Learning-based pronoun resolution for Turkish with a comparative evaluation*, Computer Speech and Language **23** (2009), no. 3, 311–331.

[24] Alex Krizhevsky, Ilya Sutskever, and Hinton Geoffrey E., *ImageNet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems 25 (NIPS2012) (2012), 1–9.

[25] Tarık Kışla and Bahar Karaoğlan, *A hybrid Statistical Approach to Stemming in Turkish: An Agglutinative Language*, Anadolu University Journal of Science and Technology-A Applied Sciences and Engineering **17** (2016), no. 2, 401–412.

[26] Shalom Lappin and Herbert J Leass, *An Algorithm for Pronominal Anaphora Resolution*, Computational Linguistics **20** (1994), no. 4, 535–561.

[27] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton, *Deep learning*, (2015).

[28] Richard P Lippmann, *An Introduction ' to Computing with Neural Nets.*

[29] Mohammad Manthouri, *HIERARCHICAL STRUCTURE BASED CONVOLUTIONAL NEURAL NETWORK FOR FACE RECOGNIT...: EBSCOhost*, (2017), no. September 2013.

[30] Joseph F McCarthy and Wendy G Lehnert, *Using Decision Trees for Coreference Resolution*, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI) (1995), no. 1, 1–5.

[31] Ruslan Mitkov, Richard Evans, and Constantin Orasan, *A New, Fully Automatic Version of Mitkov's Knowledge-Poor Pronoun Resolution Method*, Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics CICLing2002 **2276** (2002), no. IV, 168–186.

[32] Allan Pinkus, *Approximation theory of the MLP model in neural networks*, (1999).

[33] Judita Preiss, *Choosing a Parser for Anaphora Resolution*, Proceedings of the 4th Discourse Anaphora and Anaphora Resolution Colloquium (DAARC 2002) (2002), 175–180.

[34] L.R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, Proceedings of the IEEE **77** (1989), no. 2, 257–286.

[35] Dennis W. Ruck, Steven K. Rogers, Matthew Kabrisky, Mark E. Oxley, and Bruce W. Suter, *Letters: The Multilayer Perceptron as an Approximation*

to a Bayes Optimal Discriminant Function, IEEE Transactions on Neural Networks **1** (1990), no. 4, 296–298.

[36] T. Sanders and H. Pander Maat, *Cohesion and Coherence: Linguistic Approaches*, 2006, pp. 591–595.

[37] Tapan Sau and Information Technology, *BREATH ACETONE-BASED NON-INVASIVE DETECTION OF BLOOD GLUCOSE LEVELS*, (2015), no. June.

[38] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun, *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks*, (2013).

[39] Reza Shokri and Cornell Tech, *Privacy-Preserving Deep Learning.*

[40] Karen Simonyan and Andrew Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv preprint (2014), 1–10.

[41] a F Smeaton, *Progress in the application of natural language processing to information retrieval tasks*, The computer journal **35** (1992), no. 3, 268.

[42] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim, *A Machine Learning Approach to Coreference Resolution of Noun Phrases*, Computational Linguistics **27** (2001), no. 4, 521–544.

[43] E. E. Taylan, *Pronominal vs. Zero Representation of Anaphora in Turkish.*, In D. I. Slobin & K. Zimmer (Eds.), Studies in Turkish Linguistics. Amster-

dam: John Benjamins. (pp. 206-233).

[44] Pınar Tüfekçi and Yılmaz Kılıçaslan, *A Computational Model for Resolving Pronominal Anaphora in Turkish Using Hobbs' Naïve Algorithm*, International Journal of Computer, Information Science and Engineering **1** (2007), no. 5, 854–858.

[45] Umit Deniz Turan, *Null Vs. Overt Subjects in Turkish Discourse: A Centering Analysis*, (1996), no. May.

[46] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen, *Deep content-based music recommendation*, Electronics and Information Systems department (ELIS) (2013), 9.

[47] Robert D. Van Valin, *A Summary of Role and reference Grammar*, Role and Reference Grammar Web Page, . . . (2005), 1–30.

[48] Özgür Yüksel and Cem Bozsahin, *Contextually appropriate reference generation*, Natural Language Engineering **8** (2002), no. 1, 69–89.

[49] Savaş Yıldırım and Yılmaz Kılıçaslan, *A Machine Learning Approach to Personal Pronoun Resolution in Turkish*, FLAIRS Conference (2007), 269–270.

[50] George Kingsley Zipf, *Human Behaviour and the Principle of Least Effort*, 1949.