CHAPTER 4

Unit IV

Natural Language Processing and ANN

Syllabus Topics

Natural Language Processing : Introduction, Stages in natural language Processing, Application of NLP in Machine Translation, Information Retrieval and Big Data Information Retrieval. Learning : Supervised, Unsupervised and Reinforcement learning.

Artificial Neural Networks (ANNs) : Concept, Feed forward and Feedback ANNs, Error Back Propagation, Boltzmann Machine.

Syllabus Topic : Natural Language Processing - Introduction

4.1 Natural Language Processing

Introduction

- Natural Language Processing (NLP) is a form of artificial intelligence that helps machines "read" text by simulating the human ability to understand language.
- NLP techniques incorporate a variety of methods, including linguistics, semantics, statistics and machine learning to extract entities, relationships and understand context, which enables an understanding of what's being said or written, in a comprehensive way. Rather than understanding single words or combinations of them, NLP helps computers understand sentences as they are spoken or written by a human.
- It uses a number of methodologies to decipher ambiguities in language, including automatic summarization, part-of-speech tagging, disambiguation, entity extraction and relations extraction, as well as disambiguation and natural language understanding and recognition.
- Natural Language Processing involves machines or robots to understand and process the language that human speak, and infer knowledge from the speech input. It also involves the active participation from machine in the form of dialog i.e. NLP aims at the text or verbal output from the machine or robot. The input and output of an NLP system can be speech and written text respectively.

4.1.1 Components of NLP

Mainly there are two components of NLP.

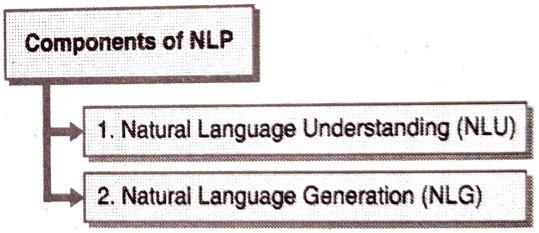


Fig. 4.1.1 : Components of NLP

1. Natural Language Understanding (NLU)

- In this part of the process, the speech input gets transformed into the useful representations in order to analyze various aspects of the language. As the natural language is very reach in forms and structures, it is also very ambiguous.
- There can be different forms of ambiguities like **lexical ambiguity**, which is a very basic i.e. word level ambiguity. For example the "document" can be a noun or verb. It's a complicated process.
- Secondly, there can be **syntactical ambiguity**, which is about parsing the sentence. For example, a sentence like "Madam

said on Monday she would give an exam". Thirdly, there can be **referential ambiguity**. Check a sentence, "Meera went to Geeta. She said "I am Hungry"."Who is hungry, is not well referred from this sentence. In many cases we observe that one sentence can have meanings. And reversely, many sentences mean the same. Hence NLU a complicated process.

➜ **2. Natural Language Generation (NLG)**

In order to generate the output text, the intermediate representation requires to be converted back to the natural language format. Hence, in this process there are multiple sub processes involves. They are as follow :

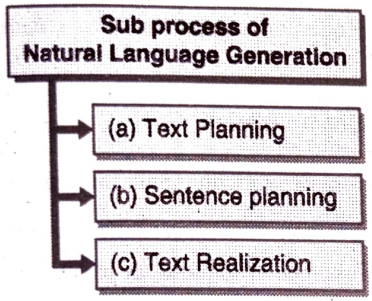


Fig. 4.1.2 : Sub process of Natural Language Generation

➜ **(a) Text Planning**

It includes extracting relevant contents from knowledge base.

➜ **(b) Sentence planning**

This process involves selecting correct words, forming meaningful sentence following language grammar and setting tone for the same.

➜ **(c) Text Realization**

This is the process of mapping the planned sentence into a structure.

Syllabus Topic : Stages in Natural Language Processing

4.1.2 Stages in Natural Language Processing

Q. 4.1.1 Explain all the steps in a NLP with an example. (Refer section 4.1.2) **(8 Marks)**

Q. 4.1.2 Explain steps in NLP. (Refer section 4.1.2) **(8 Marks)**

As natural language is very reach in forms and structures, NLP by default is a complicated process. In general, it can be divided into five steps as shown in Fig. 4.1.3.

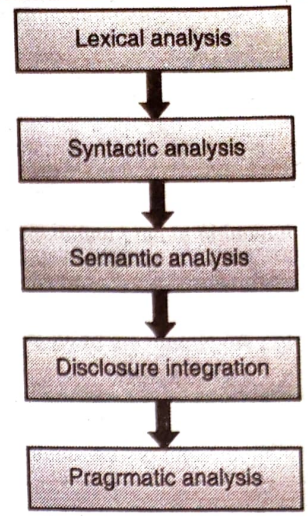


Fig. 4.1.3 : Steps of NLP process

1. Lexical Analysis

Lexicon is the words and phrases in language. Lexicon analysis deals with the recognition and identification of structure of the sentences. It divides the paragraphs in sentences, phrases and words.

2. Syntactic Analysis

In syntactic analysis the sentences are parsed as noun, verbs, adjectives, and other parts of sentences. In this phase the grammar of the sentence is analyzed in order to get the relationships among different words in the sentence. For example, "mongo eats me" will be rejected by syntactic analyzer.

3. Semantic Analysis

In this phase, the actual meaning of the sentence is extracted from the structure and the words used. It checks whether the sentence is meaningful. It maps the object with their syntactic structure to decide the correctness of the sentence. For example, "bitter sugar" will be termed as a wrong sentence by the analyzer.

4. Discourse Integration

The meaning of discourse with respect to NLP is nothing but the context of the sentence or a word. Generally, meaning or interpretation of a sentence majorly depends or changes according to the context i.e. the sentences before and after the given sentence. In discourse integration, the meaning of a sentence gets verified with the sentence before it.

5. Pragmatic Analysis

Pragmatic deals with meaning of the sentence in various situations. In pragmatic analysis, the sentences are re-interpreted to verify the correctness of the meaning in the given context. It requires to have real world knowledge of the language.

Syllabus Topic : Applications of NLP in Machine Translation

4.1.3 Applications of NLP

Q. 4.1.3 What are the applications of NLP? (Refer section 4.1.3) **(4 Marks)**

NLP has a huge number of applications. Few to mention are as follows :

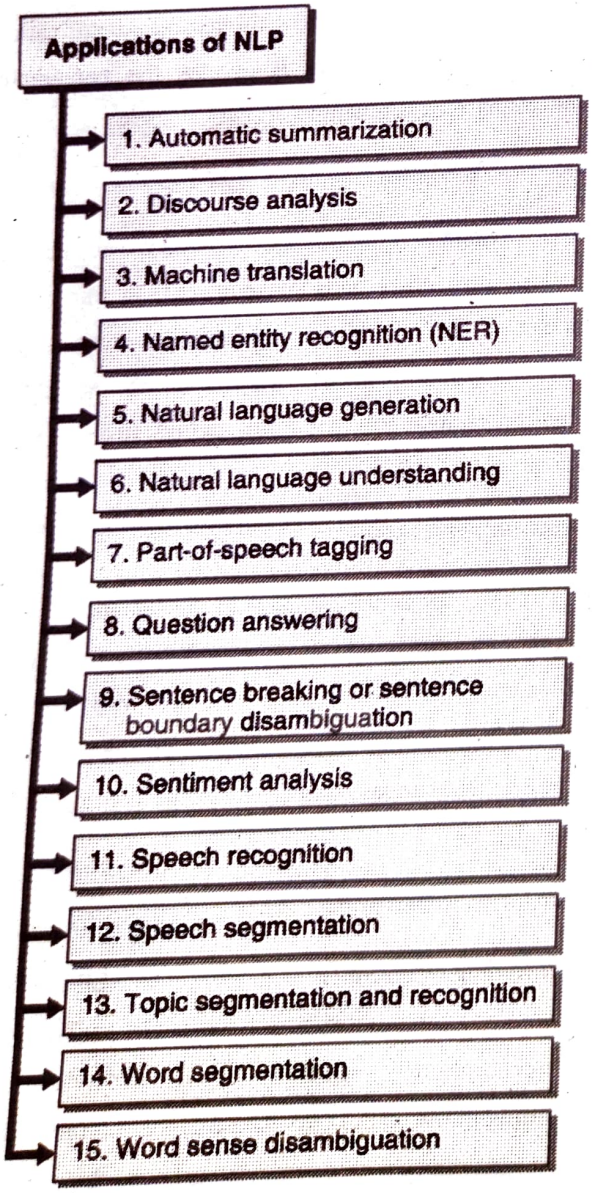


Fig. 4.1.4 : Application of NLP

1. Automatic summarization

NLP techniques are used to produce a readable summary of a chunk of text. This application is mainly found in summarizing of text of a known type such as articles in the financial section of a newspaper.

2. Discourse analysis

In this application there are various related tasks. One is to identify the discourse structure of connected text. i.e. the nature of relationship among sentences.(e.g. explanation, contrast, elaboration, etc.) Another possible task is recognizing and classifying the speech acts in a chunk of text (e.g. yes-no question, content question, statement, assertion, etc.).

3. Machine translation

This is the most difficult application of NLP. Machine translation is automatically translating text from one human language to another. It requires all of the different types of knowledge that humans possess (e.g. grammar, semantics, facts about the real world, etc.) in order to carry out the translation in a proper manner.

4. Named entity recognition (NER)

In NER, given a stream of text, NLP system determines which items in the text map to proper names, such as people or places, and what the type of each such name is (e.g. person, location, organization). Although capitalization can aid in recognizing named entities in languages such as English, this information cannot aid in determining the type of named entity. For example, the first word of a sentence is also capitalized, and named entities often span several words, only some of which are capitalized. Furthermore, many other languages in non-Western scripts (e.g. Chinese or Arabic) do not have any capitalization at all, and even languages with capitalization may not consistently use it to distinguish names.

5. Natural language generation

In this application of NLP, information from computer databases is converted into readable human language.

6. Natural language understanding

In Natural language understanding, chunks of text is converted into more formal representations such as first-order

logic structures that are easier for computer programs to manipulate. It involves the identification of the intended semantic from the multiple possible semantics which can be derived from a natural language expression.

➜ **7. Part-of-speech tagging**

In this application, the part of speech for each word is determined for given sentences. Many words, especially common ones, can serve as multiple parts of speech. For example, "book" can be a noun ("the book on the table") or verb ("to book a flight"); "set" can be a noun, verb or adjective; and "out" can be any of at least five different parts of speech. Some languages have more such ambiguity than others. Languages with little inflectional morphology, such as English are particularly prone to such ambiguity.

➜ **8. Question answering**

Using NLP, one can determine answers of human-language questions. Typical questions have a specific right answer (such as "What is the capital of India?"), but sometimes open-ended questions are also considered (such as "What is the meaning of life?"). Recent works have looked at even more complex questions.

➜ **9. Sentence breaking or sentence boundary disambiguation**

As the name suggest in this application of NLP, sentence boundaries are found from the given chunk of text. Sentence boundaries are often marked by periods or other punctuation marks, but these same characters (i.e. period or any punctuation mark) can serve other purposes (e.g. marking abbreviations). In such cases there can be ambiguities.

➜ **10. Sentiment analysis**

In sentiment analysis, subjective information is extracted usually from a set of documents, to determine "polarity" about specific objects. It is especially useful for identifying trends of public opinion in the social media, for the purpose of marketing. It is one of the big applications of NLP.

➜ **11. Speech recognition**

In this application, for a given sound clip of a person or people speaking, the textual representation of the speech is generated. This is the opposite of text to speech and is one of the extremely difficult problems.

➜ **12. Speech segmentation**

In speech segmentation, from the given sound clip of a person or people speaking, words are separated. Speech segmentation is a subtask of speech recognition and typically grouped with it.

➜ **13. Topic segmentation and recognition**

In this application, from the given a chunk of text, segments are separated, each of which is devoted to a topic, and thus the topic of the segment is identified.

➜ **14. Word segmentation**

This is a significant application for foreign languages like Chinese, Japanese and Thai which do not have word separators (like space) used. Word segmentation separates the words in order to extract knowledge.

➜ **15. Word sense disambiguation**

While using words that have more than one meaning; we have to select the meaning which makes the most sense in context. For this problem, we are typically given a list of words and associated word senses, e.g. from a dictionary or from an online resource such as WordNet.

4.2 Important Applications of NLP

There are several applications of NLP as follows :

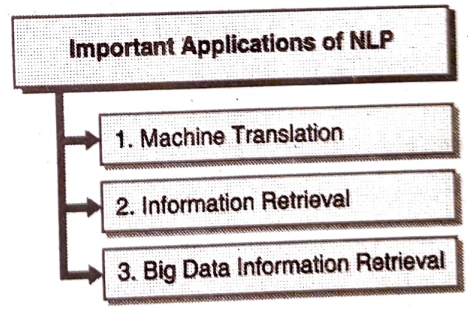


Fig. 4.2.1 : Important Applications of NLP

4.2.1 Machine Translation

– This is the most difficult application of NLP. Machine Translation (MT) was one of the first applications envisaged for computers Machine translation is automatically translating text from one human language to another, preserving the meaning of the input text, and producing fluent text in the output language.

- It requires all of the different types of knowledge that humans possess (e.g. grammar, semantics, facts about the real world, etc.) in order to carry out the translation in a proper manner. Machine translation was first demonstrated by IBM in 1954 with a basic word-for-word translation system.
- While machine translation is one of the oldest subfields of artificial intelligence research, the recent shift towards large-scale empirical techniques has led to very significant improvements in translation quality.

Various Approaches to MT

- Word-for-word translation
- Syntactic transfer
- Inter-lingual approaches
- Example-based translation
- Statistical translation

Syllabus Topic : Information Retrieval and Big Data Information Retrieval

4.2.2 Information Retrieval

- As the reader has probably already deduced, the complexity associated with natural language is especially key when retrieving textual information [Baeza-Yates, 1999] to satisfy a user's information needs. This is why in Textual Information Retrieval, NLP techniques are often used [Allan, 2000] both for facilitating descriptions of document content and for presenting the user's query, all with the aim of comparing both descriptions and presenting the user the documents that best satisfy their information needs.
- In other words, a textual information retrieval system carries out the following tasks in response to a user's query as shown in Fig. 4.2.2.

1. Indexing the collection of documents : In this phase, NLP techniques are applied to generate an index containing document descriptions. Normally each document is described through a set of terms that, in theory, best represents its content.
2. When a user formulates a query, the system analyses it, and if necessary, transforms it with the hope of representing the user's information needs in the same way as the document content is represented.
3. The system compares the description of each document with that of the query, and presents the user with those documents whose descriptions are closest to the query description.
4. The results are usually listed in order of relevancy, that is, by the level of similarity between the document and query descriptions.

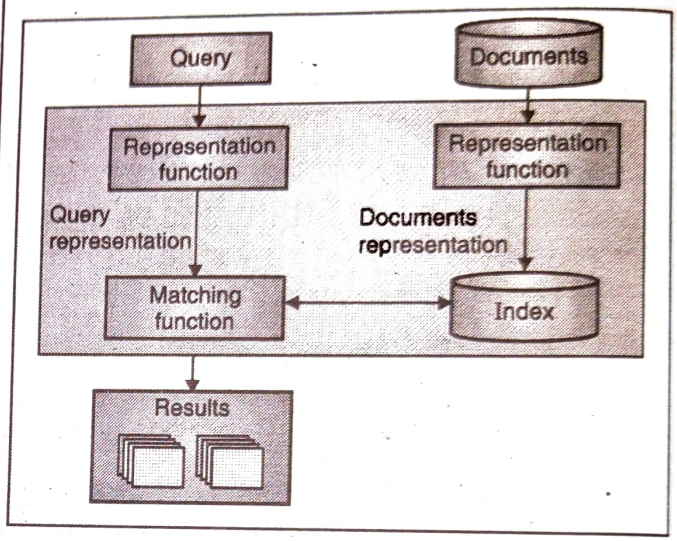


Fig. 4.2.2 : The architecture of an information retrieval system

- As of now there are no NLP techniques that allow us to extract a document's or query's meaning without any mistakes. In fact, the scientific community is divided on the procedure to follow in reaching this goal. In the following section we will explain the functions and peculiarities of the two key approaches to natural language processing : a statistical approach and a linguistic focus. Both proposals differ considerably, even though in practice natural language processing systems use a mixed approach, combining techniques from both focuses.

4.2.2(A) Statistical Processing of Natural Language

- Statistical processing of natural language [Manning, 1999] represents the classical model of information retrieval systems, and is characterized from each document's set of key words, known as the terms index.
- This is a very simple focus based on the "bag of words." In this approach, all words in a document are treated as its index terms. Moreover, each term is assigned a weight in function of its importance, usually determined by its appearance

frequency within the document. This way the word's order, structure, meaning, etc., are not taken into consideration.

- These models are then limited to pairing the documents' words with that of the query's. Its simplicity and efficacy has become the most commonly used contemporary models in textual information retrieval systems.
- This document processing model involves the following stages :

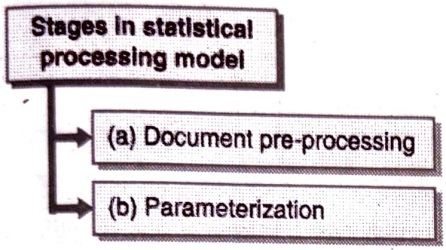


Fig. 4.2.3 : Stages in Statistical Processing model

(a) Document pre-processing

Fundamentally consisting in preparing the documents for its parameterisation, eliminating any elements considered as superfluous.

(b) Parameterization

A stage of minimal complexity once the relevant terms have been identified. This consists in quantifying the document's characteristics (that is, the terms).

Below we will illustrate their function using this paper's first paragraph as an example, assuming that it is XML tagged. So the document on which we would apply the pre-processed and parameterisation techniques would be the following :

```
<document document_ID="000127"
source=http://www.hipertext.net>
<title>
Processing Natural Language in Textual Information
Retrieval and related topics
<title>
<body>
1. Introduction
"Natural Language Processing" (NLP) as a field has been
developing for many years. It was formed in 1960 as a sub-
field of Artificial Intelligence and Linguistics, with the aim
of studying problems in the automated generation and
understand of natural language.
...
</body>
</document>
```

Document pre-processing consists of three basic phases

1. Elimination of the elements in the document that are not for indexing (stripping), such as some document tags or headers (Example 1).

Example 1 : Document without headers or tags

```
Processing Natural Language in Textual Information
Retrieval and related topics
1. Introduction
"Natural Language Processing" (NLP) as a field has been
developing for many years. It was formed in 1960 as a sub-
field of Artificial Intelligence and Linguistics, with the aim
of studying problems in the automated generation and
understand of natural language.
...
```

2. Text standardising, consisting in homogenising the whole text in the complete collection of documents to be worked on, including the consideration of capitalised or non-capitalised terms, checking specific parameters like numerals or dates; abbreviations or acronyms, eliminating empty words by applying the lists of functional words (prepositions, articles, etc.) identifying N-Grams, (the example's terms and underlined terms). (Example 2).

Example 2 : Standardised document

```
Processing natural language in textual information retrieval
and related topics
StringNumber Introduction
Processing natural language (NLP) has been developing for
StringNumber sub-area linguistics artificial intelligence aim
of study problems in the automated generation and
understanding of natural language.
...
```

3. Stemming terms is a linguistic process that attempts to determine the base (lemma) of each word in a text. Its aim is to reduce a word to its root, so that the key words in a query or document are represented by their roots instead of the original words. The lemma of word is its basic form along with its inflected forms. For example, "inform" could be the lemma of "information" or "inform." The stemming process (Example 3) is carried out by using algorithms that can represent the different variants of a term at once, while also reducing the amount of vocabulary and as a consequence improving the capacity of storage in systems, as well as document processing time. However, these algorithms have the inconvenience of sometimes not grouping words that should be grouped, and vice versa; erroneously presenting words as equals.

Example 3 : Documents with stemmed terms

natural language text information retrieval similar topics StringNumber Introdu

Process natural language (NLP) has been developing for StringNumber sub-area linguistics artificial intelligence aim study problem in generate automatic natural language understand.

...

Parametrising documents consists in assigning a weight to each one of the relevant terms associated to a document. A term's weight is usually calculated as a function of its appearance frequency in the document, indicating the importance of these terms as the document's content description (Example 4).

Related	1	linguist	1
Area	1	From	1
Automate	1	Aim	1
Understand	1	NLP	1
Develop	1	Problem	1
In	1	Process	2
Field	1	information retrieval	1
Study	1	StringNumber	--
generate	1	Subarea	1
artificial intelligence	1	Text	1
Introd	1	Years	1
many	1	[...]	
natural language	3		

Example 4 : Fragment of a parametrised document (see how the frequencies of each term changes as the quantification of the remaining terms in the document continues)

- One of the most often used methods to estimate the importance of a term is the TF.IDF system (Term Frequency, Inverse Document Frequency). It is designed to calculate the importance of a term relative to its appearance frequency in a document, but as a function of the total appearance frequency for all of the corpus' documents. That is, the fact that a term appears often in one document is indicative that that term is representative of the content, but only when that term does not appear frequently in all documents. If it appeared frequently in all documents, it would not have any discriminatory value (for example, it would be absurd to represent the content of a document in a recipe database by the frequency of the word food, even though it appears often).
- Finally, and as we have already mentioned, we must describe two commonly used techniques in the statistical processing of natural language :

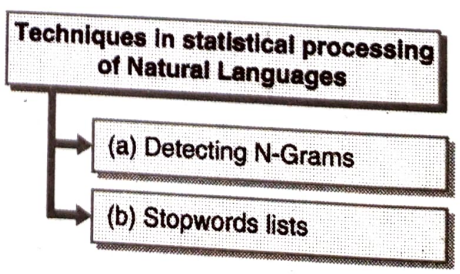


Fig. 4.2.4 : Techniques in the statistical processing of natural language

➔ **(a) Detecting N-Grams**

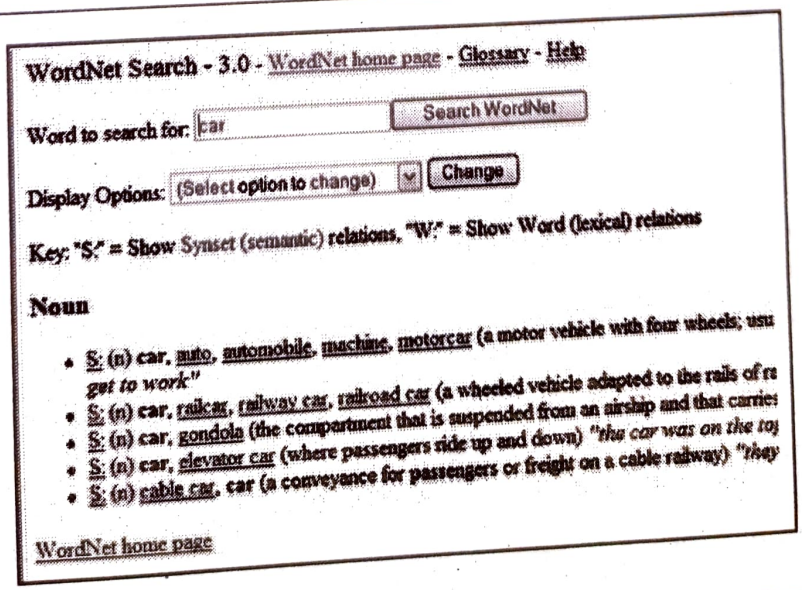
This consists in identifying words that are usually together (compound words, proper nouns, etc.) to be able to process them as a single conceptual unit. This is usually done by estimating the probability of two words that are often together make up a single term (compound). These techniques attempt to identify compound terms such as "accommodation service" or "European Union."

➔ **(b) Stopwords lists**

A list of empty words in a terms list (prepositions, determiners, pronouns, etc.) considered to have little semantic value, and are eliminated when found in document, leaving them out of the terms index to be analysed. Deleting all of these terms avoids document noise problems and saves on resources, since in documents few elements are repeated frequently.

4.2.2(B) Linguistic Processing of Natural Language

- This approach is based on the application of different techniques and rules that explicitly encode linguistic knowledge [Sanderson, 2000]. The documents are analyzed through different linguistic levels (as previously mentioned) by linguistic tools that incorporate each level's own annotations to the text. Below we show the different steps to take in a linguistic analysis of documents, even though not all systems use them.
- The morphological analysis is performed by taggers that assign each word to a grammatical category according to the morphological characteristics found.
- After having identified and analyzed the words in a text, the next step is to see how they are related and used together in making larger grammatical units, phrases and sentences. Therefore a syntax analysis of the text is performed. This is when parsers are applied : descriptive formalism that demonstrate the text's syntax structure. The techniques used to apply and create parsers vary and depend on the aim of the syntax analysis. For information retrieval it is often used for a superficial analysis aiming to only identify the most meaningful structures : nominal sentences, verbal and prepositional sentence, values, etc. This level of analysis is usually used to optimize resources and not slow down the system's response.
- From the text's syntax structure, the next aim is to obtain the meaning of the sentences within it. The aim is to obtain the sentence's semantic representation from the elements that make it up.
- One of the most often used tools in semantic processing is the lexicographic database WordNet. This is an annotated semantic lexicon in different languages made up of synonym groups called synsets which provide short definitions along with the different semantic relationships between synonym groups.



4.2.2(C) Problems Regarding NLP in IR

- Linguistic techniques must be essentially perfect
- Errors occurs in linguistic processing e.g. POS tagging, sense resolution, parsing etc.
- Effect of these errors on retrieval performance must be considered.
- Incorrectly resolving two usages of the same sense differently is disastrous for retrieval effectiveness.
- Disambiguation accuracy of at least 90% is required just to avoid degrading retrieval effectiveness.
- Queries are difficult. Queries are especially troublesome for most NLP processing.
- They are generally quite short and offer little to assist linguistic processing.
- But to have any effect whatsoever on retrieval queries must also contain the type of index terms used in documents.
- Compensated by query expansion and blind feedback.
- Linguistic knowledge is implicitly exploited.
- Statistical techniques implicitly exploit the same information the linguistic techniques make explicit.
- So linguistic techniques may provide little benefit over appropriate statistical techniques.

4.2.3 Big Data Information Retrieval

- No longer just a buzzword, the phrase "big data" describes the growing volume of structured and unstructured, multi-source information that is too large for traditional applications to handle. In terms of its usefulness, the 2013 book "Big Data : A Revolution That Will Transform How We Live, Work, and Think," by Viktor Mayer-Schönberger and Kenneth Cukier refers to big data as "the ability of society to harness massive amounts of information in novel ways to produce useful insights or goods or services of significant value."
- Regardless of the sector, every business today relies on large volumes of text information. For example, a law firm works with large amounts of research, past and ongoing legal transaction documents, notes, email correspondence as well as large volumes of governmental and specialized reference information. A pharmaceutical company will have large volumes of clinical trial information and data, doctor notes, patient information and data, patent and regulatory information as well as the latest research on competitors.
- Because these types of information are largely made up of language, natural language processing for big data presents an opportunity to take advantage of what is contained in especially large and growing stores of content to reveal patterns, connections and trends across disparate sources of data.

Interactions

- Today, natural language processing technologies are already at work in a variety of commonly used interactive applications such as smartphone assistants like Apple's Siri, in online banking and retail self-service tools and in some automatic translation programs.
- Users ask questions in everyday language and receive immediate, accurate answers. It's a win-win for both customers, who can easily communicate with companies they do business with whenever and wherever they want, and for companies who increasingly realize savings by reducing the number of calls handled by traditional live assistance.

Business Intelligence

- Natural language processing for big data can be leveraged to automatically find relevant information and/or summarize the content of documents in large volumes of information for collective insight.
- Users are no longer limited by having to choose or know the "right" keywords to retrieve what they're looking for but can interact with the content via search using queries in their own words. Faster, more thorough access to information speeds up all downstream processes that depend on timely information and enable its use for real time, actionable business intelligence.

Sentiment analysis

- With an increasingly online customer base, social channels are a rich, if noisy source of invaluable information. Using natural language processing for sentiment analysis,

organizations can understand what is being said about their brand and products, as well as "how" it's being talked about how users feels about a service, product or concept/idea. This is a powerful way to discover information about the market and about current and potential customers (opinions, but also information about customer habits, preferences and needs/wants, as well as demographic information) that would otherwise remain out of reach. This information can then be applied to product development, business intelligence and market research.

- If estimates by IDC come true, by 2020, we'll be looking at around 44 trillion gigabytes of digital knowledge worldwide (an IDC Digital Universe Study reports that by 2020, for every human in the world, approximately 1.7 megabytes of new information will be created every second; that's around 44 trillion gigabytes). Fourty four trillion gigabytes is a lot of potential. No matter where you apply it, natural language processing for big data will be an essential feature to build into your analysis pipeline in order to capture the value of this information for insight, reduced costs and increased productivity.

Syllabus Topic : Learning - Supervised, Unsupervised and Reinforcement Learning

4.3 Learning : Supervised, Unsupervised and Reinforcement Learning

4.3.1 General Model of Learning Agents

Q. 4.3.1 Explain general model of learning agents. (Refer section 4.3.1) **(8 Marks)**

Learning agent can be classified into four theoretical components as shown in Fig. 4.3.1. Fig. 4.3.2 shows the architecture of learning agent, thereby detailing the interconnections among all the components and there working.

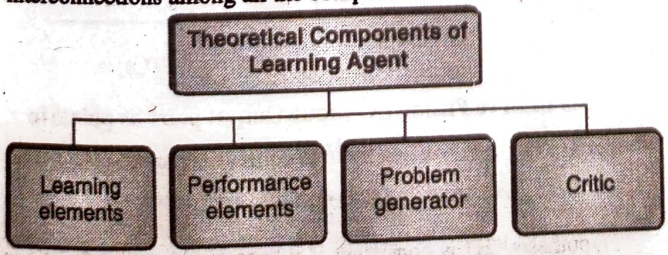


Fig. 4.3.1 : Theoretical components of Learning Agent

1. Learning Elements

- Component which plays an important role in improvising an agent is called as Learning Element.
- Learning element helps in improvising an agent by taking knowledge about performance element and feedback (we will learn about types of feedback in upcoming sections) from Critic.
- After that it evaluates how to revise performance element for optimizing the results.

2. Performance Elements

- Performance element is the main component of an agent like a Neuron in human body.
- It takes input from the environment with the help of sensors and decides the appropriate action based on the interaction with learning element, and then it performs the action with the help of effectors.

3. Problem Generator

- Problem generator experiments on performance element by suggesting actions which can generate new instances or experiences.
- This is useful for training an agent further for better results.

4. Critic

- Critic follows some basic performance standard after receiving input from the sensors (input can be a success or a failure).
- It compares this performance standard with input and based on the comparison. Critic gives feedback to the learning agent.

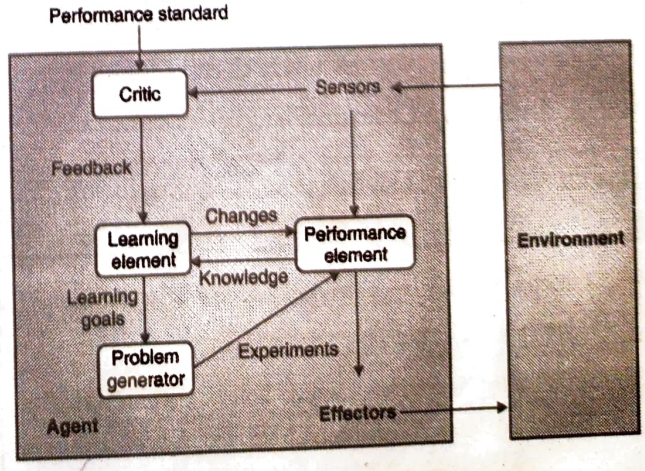


Fig. 4.3.2 : Learning Agent Architecture

Let's take on example of automated car agent, theoretical components of automated car learning agent can be given as follows :

1. **Learning elements :** Create objective (e.g. Learn routing various places, Learn when to use break, accelerator, gear, etc.)
2. **Performance element has information about the procedure for driving a car** (e.g. actions like : Turning with steering, accelerating to increase speed, changing gears, stop car with break, etc.)
3. **Problem generator :** Trying different routes to travel is the work of problem generator.
4. **Critic :** Observes surrounding environmental conditions and gives this information to the learning element. (e.g. check reaction of other drivers if there is slope on the road or if there is a speed breaker, or else if there is a quick left turn across two lanes of traffic).

Take another example of automated air conditioner. Theoretical components of automated air conditioner learning agent can be given as follows :

1. **Learning elements :** Create objective (e.g. Learn when to when to switch temperature levels, etc.)
2. **Performance element has information about the procedure for using a air conditioner** (e.g. actions like : how to switch temperature levels, etc.)
3. **Problem generator :** Trying different temperature levels is the work of problem generator.
4. **Critic :** Observes surrounding environmental conditions and gives this information to the learning element (e.g. check reaction of people if temperature levels are varied).

4.3.2 Inductive Learning

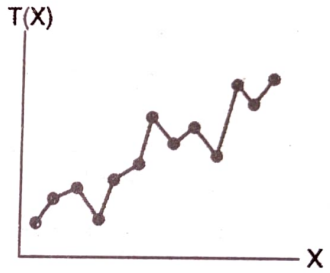
Q. 4.3.2 Write a short note on Inductive learning. (Refer section 4.3.2) **(8 Marks)**

- Inductive learning is a supervised learning technique because it learns through examples.
- It is called inductive learning because, it induces (i.e. stimulates) general protocols from the set of instances observed by agent from given examples. Learning method extracts these protocols through massive data sets.
- Inductive learning is also called as concept learning. It constructs class definitions by various classification methods like : Versions Spaces, Decision Trees, etc.
- Inductive learning tries to find a hypothesis (supposition) which approximates a set of samples for defining a target function "T".
- Let's put this in an equation format :

  "T" is a target function

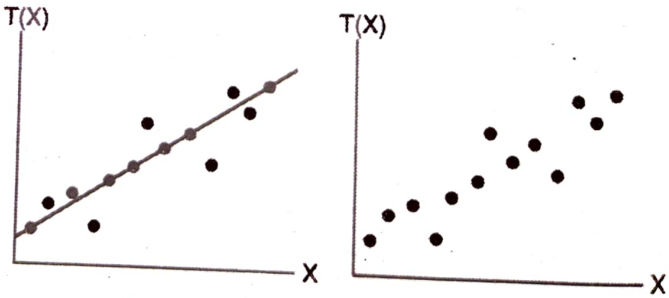
  (x, T(x)) is a sample input-output pair
- The problem is to find a hypothesis H

  such that $H \approx T$ (for given training set of examples)
- Inductive learning is a most simplified model of real learning. It ignores prior knowledge and assumes that the examples are given as input.



(A)

- It constructs hypothesis "H" to agree with a target function "T" on training set. You can understand this with the help of Fig. 4.3.3.



(B) (C)

Fig. 4.3.3 : Curve Fitting (A) Input-Output pairs as points in plane, (B) "H" function consisting of linear segments, (C) "H" expressed as polynomial function

- Hypothesis "H" is consistent if it agrees with target function "T" on all examples. If we have multiple hypotheses which

are consistent with the data then simplest one should be selected.

- Take one more example of Tic-Tac-Toe. T(x) is our target function and our goal is to find a hypothesis such that H ≈ T. Our target is that "X" mark should win the game. In first case it can be seen that "X" mark is winning the game so T(x) = + 1. In second case it can be seen that "X" mark is losing the game so T(x) = – 1 and in last case it can be seen that neither "X" mark nor "O" can win the game, so the match will be drawn thus T(x) = 0.

x = [+ 1 – 1 – 1 0 + 1 0 0 0 + 1]

T(x) = + 1

X	O	O
	X	
		X

x = [0 – 1 – 1 + 1 + 1 0 0 0 + 1]

T(x) = – 1

X	O	O
	X	
	X	

x = [+ 1 – 1 – 1 0 + 1 0 0 + 1 0]

T(x) = 0

	O	O
X	X	
		X

4.3.3 Type of Feedback

Q. 4.3.3 Explain Supervised, Unsupervised and reinforcement learning with example. (Refer sections 4.3.4 to 4.3.6) **(8 Marks)**

Feedback gives information about the actual outputs of performed actions. There are three main types of learning as shown in Fig. 4.3.4.

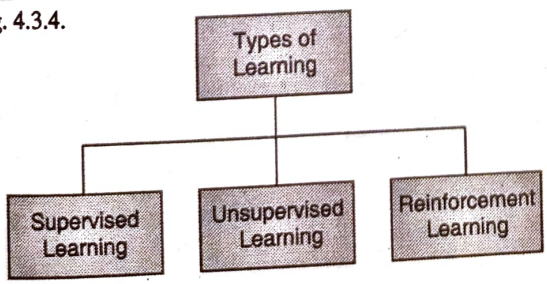


Fig. 4.3.4 : Types of Learning

4.3.4 Supervised Learning

Q. 4.3.4 Write a short note on : Supervised learning. (Refer section 4.3.4) **(6 Marks)**

Q. 4.3.5 Explain the working of a learning agent with example. (Refer section 4.3.4) **(8 Marks)**

- Learning agent and teacher both take input from surrounding environment.
- Teacher has the desired answer key for the given problems.
- Whereas, learning agent processes the given input and gets an output.
- This actual output is subtracted from the desired output by the adder, which shows the error in the actual output.

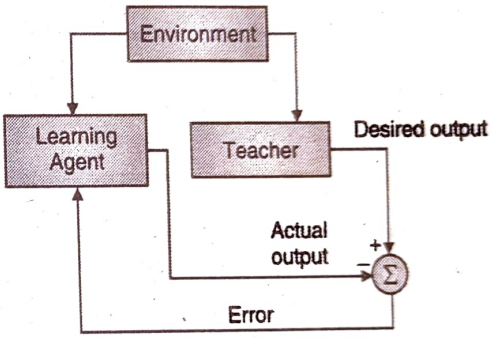


Fig. 4.3.5 : Supervised Learning

- This error is given as input to the learning agent, so that it can learn from this error and while generating output for the next time it can try to remove that error.
- As per the name Supervised Learning is performed under supervision of a Teacher.
- Teacher can be an agent/system which has a correct answer for each example. This answer can be a variable in numeric form, a categorical variable, etc.
- Let's take one example where Learning Agent has to identify living and nonliving things.
- Supervised Learning will have correct answers see labels shown in Fig. 4.3.6.

Fig. 4.3.6 : Supervised Learning Agent with correct answer

- In short, the way in real life teacher guides a student to get better results; supervised learning method guides learning agent with the help of teacher to get better results.

4.3.5 Unsupervised Learning

Q. 4.3.6 Write a short note on : Unsupervised learning. (Refer section 4.3.5) **(6 Marks)**

Q. 4.3.7 Describe unsupervised learning with example. (Refer section 4.3.5) **(8 Marks)**

- Unsupervised learning does not have any teacher which means, that correct answer is not known to the learning agent.
- Unsupervised learning agent tries to learn on its own from the patterns without corresponding output values, as shown in Fig. 4.3.7.

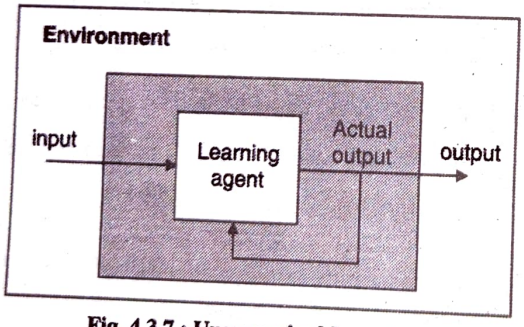


Fig. 4.3.7 : Unsupervised Learning

- Take same example where Learning Agent has to identify living and non-living things. Unsupervised learning will not have correct answers as can be seen in Fig. 4.3.8 which is without labels.

Fig. 4.3.8 : Unsupervised Learning Agent without correct answer

4.3.6 Reinforcement Learning

Q. 4.3.8 Explain reinforcement learning. (Refer section 4.3.6) **(8 Marks)**

- Reinforcement learning is based on occasional rewards. Reinforcement learning agent does not have the exact output for given inputs, but it accepts feedback on the desirability of the outputs. This feedback can be provided by the environment or the agent itself.
- Feedback generally occurs after a sequence of actions, so there can be a delay in getting respective improved action immediately.
- Reinforcement learning agent knows that the results are right (or wrong), but it does not know what action caused the results.

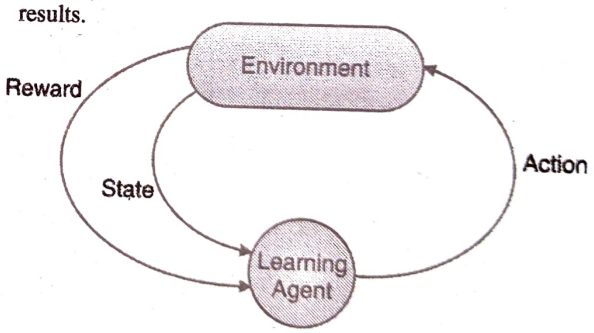


Fig. 4.3.9 : Reinforcement Learning

Syllabus Topic : Artificial Neural Network

4.4 Neural Networks

The basic computational unit in the nervous system is the nerve cell, or neuron. Human brain contains around 10^{11} neurons. Each neuron is connected to approximately 10^4 others. A neuron has :

- Dendrites (inputs)
- Cell body
- Axon (output)

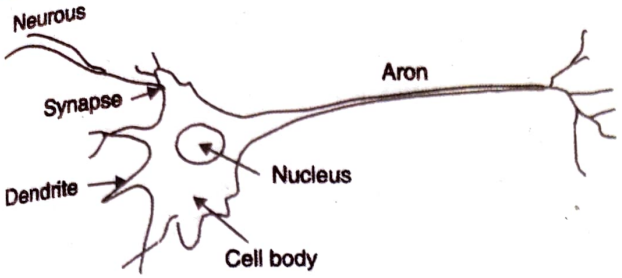


Fig. 4.4.1

- Experiments and evidences suggest that neurons receive, analyze and transmit information. A neuron receives input from other thousands of neurons. The information in transmitted in a form of electro-chemical signals (pulses). Once input exceeds a critical level; the neuron discharges a spike - an electrical pulse that travels from the body, down the axon, to the next neuron(s) or other receptors. When a neuron sends the information we say that a neuron 'fires'. This spiking event is also called depolarization.
- The axon endings (Output Zone) almost touch the dendrites or cell body of the next neuron. Transmission of an electrical signal from one neuron to the next is effected by neurotransmitters, chemicals which are released from the first neuron and which bind to receptors in the second. The receptors of a neuron are called synapses, and they are located on many branches, called dendrites.
- There are many types of synapses, but roughly they can be divided into two classes :

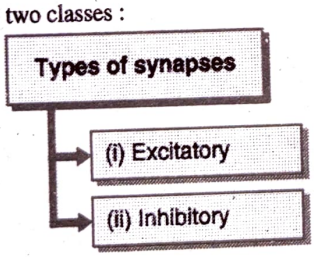


Fig. 4.4.2 : Types of synapses

➜ **(i) Excitatory**

A signal received at this synapse 'encourages' the neuron to fire

➜ **(ii) Inhibitory**

A signal received at this synapse inhibits the neuron (as if asking to 'shut up')

- The neuron analyses all the signals received at its synapses. If most of them are 'encouraging', then the neuron gets 'excited' and fires its own message along a single wire, called axon. The axon may have branches to reach many other neurons.
- Computational neurobiologists have constructed very elaborate computer models of neurons in order to run detailed simulations of particular circuits in the brain. People have implemented model neurons in hardware as electronic circuits, often integrated on VLSI chips.

Syllabus Topic : ANN - Concept

4.5 Artificial Neural Network

They typically consist of many simple processing units, which are wired together in a complex communication network as human brain. Each of the processing unit is called a s neuron. It consists of input signals, weights assigned to each of the input, processing function f which computes the summation of weighted inputs, and output signal. Basic structure of neuron is as shown in Fig. 4.5.1.

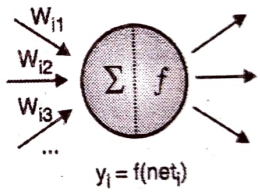


Fig. 4.5.1

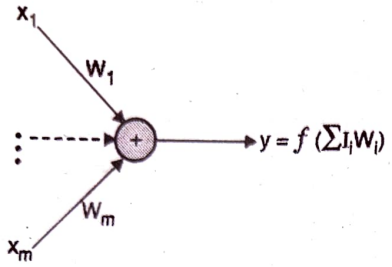
☞ **Significance of NN in AI**

- NN have the ability to learn from experience in order to improve their performance and to adapt themselves to changes in the environment, which resembles with human learning and inference patterns.
- In addition to that they are able to deal with incomplete information or noisy data and can be very effective especially in situations where it is not possible to define the rules or steps that lead to the solution of a problem as it is expected from an intelligent human to do.
- So as NN are mimicking activities of human brain, they have a significant application in artificial intelligence field.
- There are many AI applications like intelligent washing machine, Air conditioner, driverless car which uses NN

techniques to a complex level to develop and implement such real time systems.

A Model of a Single Neuron (Unit)

- McCulloch and Pitts (1943) proposed the 'integrate and fire' model, which is named after him as McCulloch and Pitts model.



- The m input values are denoted by $x_1, x_2, \ldots, x_m$.
- Each of the m inputs (synapses) has a weight $w_1, w_2, \ldots, w_m$.
- The input values are multiplied by their weights and summed

$$v = w_1 x_1 + w_2 x_2 + \ldots + w_m x_m = \sum_{i=1}^{m} w_i x_i$$

- The output is some function $y = f(v)$ of the weighted sum.

Example 1

Let $x = (0, 1, 1)$ and $w = (1, -2, 4)$.

Then

$v = 1 \cdot 0 - 2 \cdot 1 + 4 \cdot 1 = 2$

Activation Function

- The output of a neuron (y) is a function of the weighted sum $y = f(v)$. This function is often called the activation function.
- There are different types of activation functions.

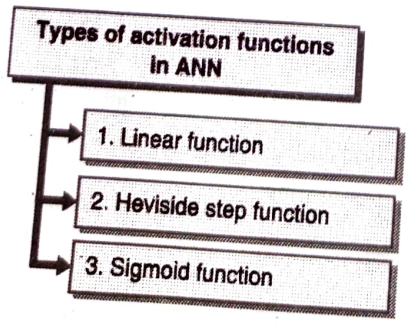


Fig. 4.5.2 : Types of activation functions

1. Linear function

$$f(v) = a + v = a + \sum w_i x_i$$

where parameter a is called bias.

Notice that in this case, a neuron becomes a linear model with bias a being the intercept and the weights, $w_1, \ldots, w_m$, being the slopes.

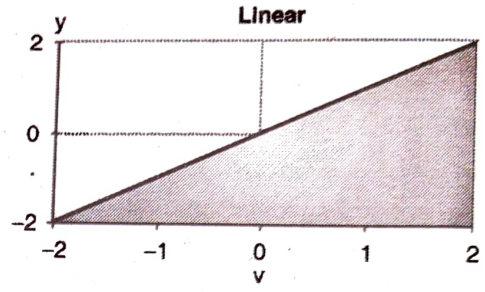


Fig. 4.5.3 : Linear Activation Function

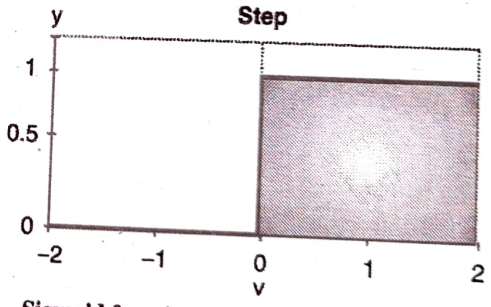
2. Heviside step function

$$f(v) = \begin{cases} 1 & \text{if } v \geq a \\ 0 & \text{otherwise} \end{cases}$$

Here a is called the threshold.

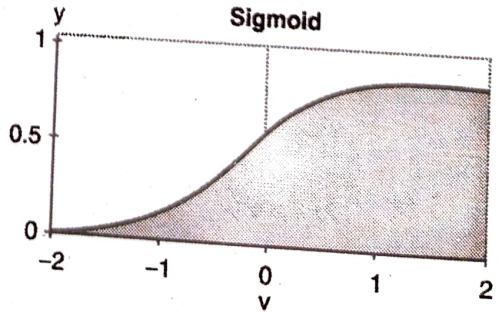
Example 2

If $a = 0$ and $v = 2 > 0$, then $f(2) = 1$, the neuron fires



3. Sigmoid function

$$f(v) = \frac{1}{1 + e^{-v}}$$



Syllabus Topic : Feed Forward Neural Network

4.5.1 Feed Forward Neural Network

- Fig. 4.5.4 depicts a neural network consisting of single layer feed forward neural network. The most common structure of connecting neurons into a network is by layers. The simplest form of layered network is shown in Fig. 4.5.4.
- These are the networks without cycles or feedback loops, hence are called as feed-forward networks or perceptron.
- The shaded nodes on the left are in the so-called *input layer*. The input layer neurons are to only pass and distribute the inputs and perform no computation. Thus, the only true layer of neurons is the one on the right. Each of the inputs $x_1, x_2, \ldots x_N$ is connected to every artificial neuron in the output layer through the connection weight. Since every value of outputs $y_1, y_2, \ldots y_N$ is calculated from the same set of input values, each output is varied based on the connection weights. Even if, the presented network is *fully connected*, the true biological neural network may not have all possible connections - the weight value of zero can be represented as "no connection".

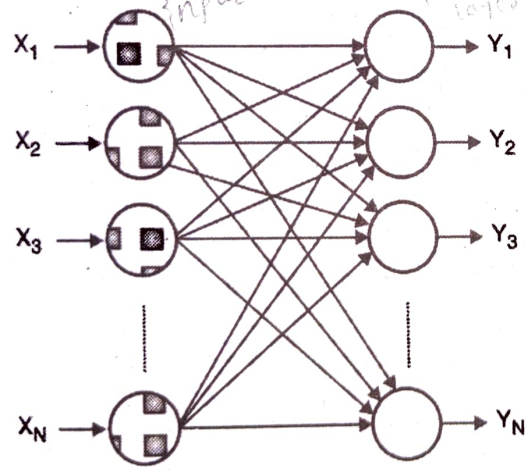


Fig. 4.5.4 : Single Layer Feed Forward Neural Network

- To achieve higher level of computational capabilities, a more complex structure of neural network is required. Fig. 4.5.5 shows the *multilayer neural network* which distinguishes itself from the single-layer network by having one or more *hidden layers*. In this multilayer structure, the input nodes pass the information to the units in the first hidden layer, then the outputs from the first hidden layer are passed to the next layer, and so on.
- Multilayer network can be also viewed as cascading of groups of single-layer networks. The level of complexity in computing can be seen by the fact that many single-layer networks are combined into this multilayer network. The designer of an artificial neural network should consider how many hidden layers are required, depending on complexity in desired computation.

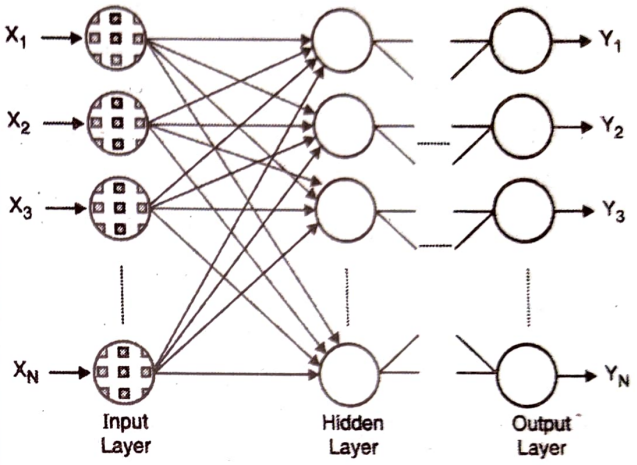


Fig. 4.5.5 : Multilayer Feed forward Network

Syllabus Topic : Feedback ANNs

4.5.2 Feedback ANNs

- Feedback networks, and multi layered perceptrons, in general, are feedforward networks with distinct input, output, and hidden layers. The units function basically like perceptrons, except that the transition (output) rule and the weight update (learning) mechanism are more complex.
- The Fig. 4.5.6 presents the architecture of back propagation networks.

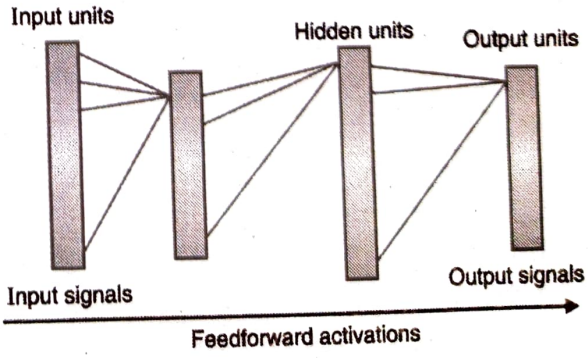


Fig. 4.5.6 : Architecture of feedback network

- There may be any number of hidden layers, and any number of hidden units in any given hidden layer. Input and output

units can be binary {0, 1}, bi-polar {– 1, + 1}, or may have real values within a specific range such as [– 1, 1]. Note that units within the same layer are not interconnected.

Syllabus Topic : Error Back Propagation

4.5.3 Error Back Propagation

- In feedforward activation, units of hidden layer 1 compute their activation and output values and pass these on to the next layer, and so on until the output units will have produced the network's actual response to the current input. The activation value ak of unit k is computed as follows.
- This is basically the same activation function of linear threshold units (McCulloch and Pitts model).

$$a_k = \sum_{i=1}^{n} w_k x_i$$

- As illustrated above, x_i is the input signal coming from unit i at the other end of the incoming connection. wk_i is the weight of the connection between unit k and unit i. Unlike in the linear threshold unit, the output of a unit in a back propagation network is no longer based on a threshold. The output y_k of unit k is computed as follows :

$$y_k = f(a_k) \text{ and } f(x) = 1/(1 + e^{-x})$$

- The function f(x) is referred to as the output function. It is a continuously increasing function of the sigmoid type, asymptotically approaching 0 as x decreases, and asymptotically approaches 1 as x increases. At x = 0, f(x) is equal to 0.5.

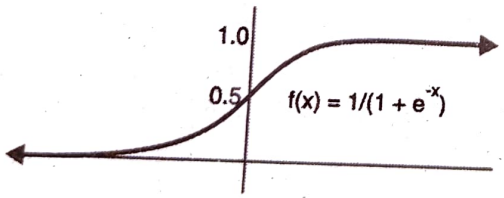


Fig. 4.5.7

- The output function f(x) is a continuously increasing function of the "sigmoid" type, asymptotically approaching 0 as x decreases, and approaching 1 as x increases. At x equal to 0, f(x) is equal to 0.5.
- In some implementations of the back propagation model, it is convenient to have input and output values that are bi-polar.

In this case, the output function uses the hyper tangent function, which has basically the same shape, but would be asymptotic to –1 as x decreases. This function has value 0 when x is 0.

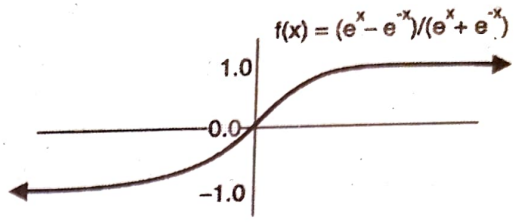


Fig. 4.5.8

- Once activation is fed forward all the way to the output units, the network's response is compared to the desired output ydi which accompanies the training pattern. There are two types of error. The first error is the error at the output layer. This can be directly computed as follows :

$$e_i = y_i^d - f(a_i)$$

- The second type of error is the error at the hidden layers. This cannot be computed directly since there is no available information on the desired outputs of the hidden layers. This is where the retropropagation of error is called for.
- Essentially, the error at the output layer is used to compute for the error at the hidden layer immediately preceding the output layer. Once this is computed, this is used in turn to compute for the error of the next hidden layer immediately preceding the last hidden layer. This is done sequentially until the error at the very first hidden layer is computed. The retropropagation of error is illustrated in the Fig. 4.5.9.

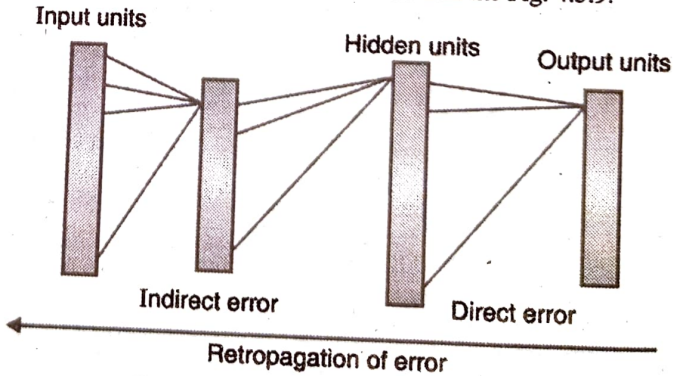


Fig. 4.5.9 : Retropropagation of error

- Computation of errors e_i at a hidden layer is done as follows :

$$e_h = \sum_{i=1}^{n} w_{ih} e_i$$

- The errors at the other end of the outgoing connections of the hidden unit h have been earlier computed. These could be error values at the output layer or at a hidden layer. These error signals are multiplied by their corresponding outgoing connection weights and the sum of these is taken.

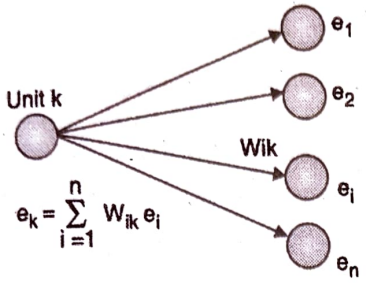


Fig. 4.5.10

- The error values at the output layer or the hidden layer are already computed. These errors are multiplied by their corresponding outgoing connection weights and the weighted sum is computed.
- After computing for the error for each unit, whether it be at a hidden unit or at an output unit, the network then fine-tunes its connection weights $w_{kj}^{t} + 1$. The weight update rule is uniform for all connection weights.

$$w_{kj}^{t+1} = w_{kj}^{t} + \alpha e_k f(a_k) f'(a_k)$$

- The learning rate a is typically a small value between 0 and 1. It controls the size of weight adjustments and has some bearing on the speed of the learning process as well as on the precision by which the network can possibly operate. f'(x) also controls the size of weight adjustments, depending on the actual output f(x). In the case of the sigmoid function above, its first derivative (slope) f'(x) is easily computed as follows :

$$f'(x) = f(x)(1 - f(x))$$

- We note that the change in weight is directly proportional to the error term computed for the unit at the output end of the incoming connection. However, this weight change is controlled by the output signal coming from the input end of the incoming connection. We can infer that very little weight change (learning) occurs when this input signal is almost zero.

  The weight change is further controlled by the term f'(ak). Because this term measures the slope of the function, and knowing the shape of the function, we can infer that there will likewise be little weight change when the output of the unit at the other end of the connection is close to 0 or 1. Thus, learning will take place mainly at those connections with high pre-synaptic signals and non-committed (hovering around 0.5) post-synaptic signals.

Syllabus Topic : Boltzmann Machine

4.6 Boltzmann Machine

- The Hopfield Network or Hopfield Model is one good way to implement an associative memory. It is simply a fully connected recurrent network of N McCulloch-Pitts neurons. Activations are normally ±1, rather than 0 and 1, so the neuron activation equation is :

$$x_i = \operatorname{sgn} \sum_i w_{ih} x_i - \theta_i$$

where $\operatorname{sgn}(x) - \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$

- Unlike the earlier feed-forward McCulloch-Pitts networks, the activations here depend on time, because the activations keep changing till they have all settled down to some stable pattern. Those activations can be updated either synchronously or asynchronously. It can be shown that the required associative memory can be achieved by simply setting the weights w_{ij} and thresholds θ j in relation to the target outputs t^p as follows :

$$w_{ij} = \frac{1}{N} \sum_{p=1}^{p} t_i^P t_j^P, \theta_i - 0$$

- A stored pattern t will then be stable if the neuron activations are not changing, i.e.,

$$r_i^q = \operatorname{sgn}\left(\sum_i w_{ih} r_i^q - \theta_i\right) - \operatorname{sgn}\left(\sum_i \frac{1}{N} \sum_p t_i^P t_j^q\right)$$

which is best analyzed by separating out the q term from the p sum to give

$$t_i^q = \operatorname{sgn}\left(t_i^q + \frac{1}{N} \sum_i \sum_{p=q} t_i^P t_j^q\right)$$

- If the second term in this is zero, it is clear that pattern number q is stable. It will also be stable if the second term is non-zero but has magnitude less than 1, because that will not

be enough to move the argument over the step of the sign function sgn(x).

- In practice, this happens in most cases of interest as long as the number of stored patterns P is *small enough*. Moreover, not only will the stored patterns be stable, but they will also be ***attractors*** of patterns close by. Estimates of what constitutes a small enough number P leads to the idea of the ***storage capacity*** of a Hopfield network.

Boltzmann Machines

- A ***Boltzmann Machine*** is a variant of the Hopfield Network composed of N units with activations $\{x_i\}$. The state of each unit i is updated asynchronously according to the rule :

$$x_i = \begin{cases} +1 & \text{with probability } p_i \\ -1 & \text{with probability } 1 - p_i \end{cases}$$

$$p_i = \frac{1}{1 + \exp - \left(\sum_{j=i}^{N} w_{ij} x_j - \theta_i \right) T}$$

- with positive temperature constant T, network weights wij and thresholds θj.
- The fundamental difference between the Boltzmann Machine and a standard Hopfield
- Network is the ***stochastic activation*** of the units. If T is very small, the dynamics of the
- Boltzmann Machine approximates the dynamics of the discrete Hopfield Network, but when T is large, the network visits the whole state space. Another difference is that the nodes of a Boltzmann Machine are split between visible input and output nodes, and hidden nodes, and the aim is to have the machine learn input-output mappings.
- Training proceeds by updating the weights using the ***Boltzmann learning algorithm.***

$$\Delta w_{ij} = -\frac{n}{T} [(x_i x_j)_{fixed} - (x_i x_j)_{free}]$$

where $<x_i x_j>_{fixed}$ is the expected/average product of x_i and x_j during training with the input and output nodes fixed at a training pattern and the hidden nodes free to update, and $xix\,j$ *free* is the corresponding quantity when the output nodes are also free.

- For both Hopfield Networks and Boltzmann Machines one can define an ***energy function***

$$E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} x_i x_j + \sum_{i=1}^{N} \theta_i x_i$$

and the network activation updates cause the network to settle into a local minimum of this energy. This implies that the stored patterns will be local minima of the energy. If a Boltzmann Machine starts off with a high temperature and is gradually cooled (known as ***simulated annealing***), it will tend to stay longer in the basins of attraction of the deepest minima, and have a good change of ending up in a global minimum at the end.