

# Foundation of AI and ML (4351601) - Summer 2024 Solution

Milav Dabgar

May 16, 2024

## પ્રશ્ન 1(a) [3 ગુણ]

Narrow AI અથવા Weak AI નો અર્થ શું છે?

### જવાબ

Narrow AI અથવા Weak AI એ specific અને limited કાર્યો માટે બનાવેલ artificial intelligence systems છે.

#### કોષ્ટક 1. Narrow AI ની લાક્ષણિકતાઓ

પાસું	વર્ણન
વ્યાપ્તિ	ફક્ત specific કાર્યો માટે
બુદ્ધિમત્તા	કાર્ય-વિશિષ્ટ કુશળતા
ઉદાહરણો	Siri, chess programs, recommendation systems
શૈખવાની પ્રક્રિયા	Domain માં pattern recognition

### મેમરી ટ્રીક

“Narrow = ફક્ત વિશિષ્ટ કાર્યો”

## પ્રશ્ન 1(b) [4 ગુણ]

વ્યાખ્યાયિત કરો: વગ્ાઈકરણ, રીપ્રેસન, કલસ્ટરિંગ, એસોસિએશન વિશ્લેષણ.

### જવાબ

#### કોષ્ટક 2. Machine Learning ની તકનીકો

તકનીક	વ્યાખ્યા	પ્રકાર	ઉદાહરણ
વગ્ાઈકરણ	Discrete categories/classes predict કરે છે	Supervised	Email spam detection
રીપ્રેસન	Continuous numerical values predict કરે છે	Supervised	House price prediction
કલસ્ટરિંગ	Similar data points ને group કરે છે	Unsupervised	Customer segmentation
એસોસિએશન વિશ્લેષણ	Variables વચ્ચે relationships શોધે છે	Unsupervised	Market basket analysis

### મેમરી ટ્રીક

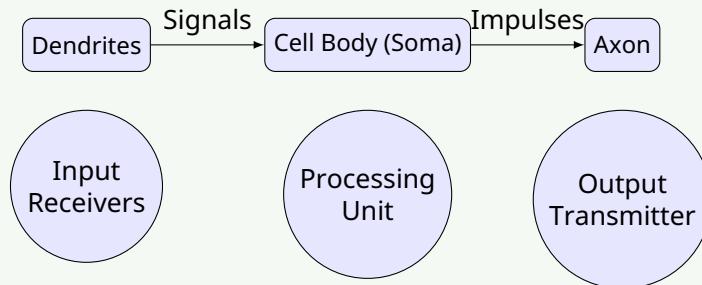
“CRCA - Categories, Real numbers, Clusters, Associations”

## પ્રશ્ન 1(c) [7 ગુણ]

ન્યુરોનના ત્રણ મુખ્ય ઘટકોને પ્રકાશિત કરો.

### જવાબ

Biological neuron ના ત્રણ મુખ્ય ઘટકો જે artificial neural networks ને inspire કરે છે:



આકૃતિ 1. ન્યુરોન ઘટકો

કોષ્ટક 3. ન્યુરોન ઘટકો

ઘટક	કાર્ય	AI માં સમકક્ષ
Dendrites	અન્ય neurons થી input signals receive કરે છે	Input layer/weights
Cell Body (Soma)	Signals ને process અને integrate કરે છે	Activation function
Axon	અન્ય neurons ને output signals transmit કરે છે	Output connections

### મુખ્ય મુદ્દાઓ:

- Dendrites: વિવિધ connection strengths સાથે input receivers તરીકે કામ કરે છે.
- Cell Body: Inputs ને sum કરે છે અને threshold function apply કરે છે.
- Axon: Processed signal ને આગામના neurons સુધી પહોંચાડે છે.

### મેમરી ટ્રીક

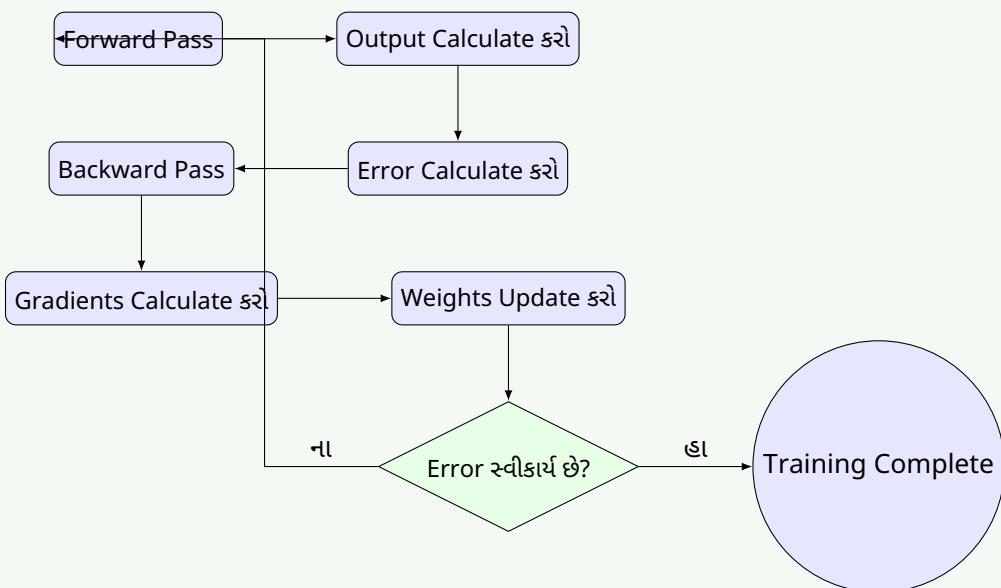
“DCA - Dendrites Collect, Cell-body Calculates, Axon Announces”

## પ્રશ્ન 1(c) OR [7 ગુણ]

Artificial Neural Network માં back propagation પદ્ધતિ સમજાવો.

### જવાબ

Back Propagation એ supervised learning algorithm છે જે gradient descent દ્વારા error minimize કરીને multi-layer neural networks ને train કરે છે.



આકૃતિ 2. Back Propagation ફ્લો

## કોષ્ટક 4. Back Propagation Steps

સ્ટેપ	પ્રક્રિયા	ફોર્મ્યુલા
Forward Pass	Layer દ્વારા layer outputs calculate કરો	$y = f(\sum(w_i x_i + b))$
Error Calculation	Loss function compute કરો	$E = \frac{1}{2}(\text{target} - \text{output})^2$
Backward Pass	Error gradients calculate કરો	$\delta = \partial E / \partial w$
Weight Update	Learning rate વાપરીને weights adjust કરો	$w_{new} = w_{old} - \eta \cdot \delta$

## મુખ્ય લાક્ષણિકતાઓ:

- Gradient Descent: Minimum error શોધવા માટે calculus વાપરે છે.
- Chain Rule: Layers દ્વારા error ને backward propagate કરે છે.
- Learning Rate: Weight updates ની speed control કરે છે.

## મેમરી ટ્રીક

"FEBU - Forward, Error, Backward, Update"

## પ્રશ્ન 2(a) [3 ગુણ]

Machine Learning માં ઉપયોગમાં લેવાતા કોઈપણ પાંચ લોકપ્રિય algorithms ની સૂચિ બનાવો.

## જવાબ

## કોષ્ટક 5. લોકપ્રિય ML Algorithms

Algorithm	અકાર	Application
Linear Regression	Supervised	Continuous values નું prediction
Decision Tree	Supervised	Classification અને regression
K-Means Clustering	Unsupervised	Data grouping
Support Vector Machine	Supervised	Margins સાથે classification
Random Forest	Supervised	Ensemble learning

## મેમરી ટ્રીક

``LDKSR - Learn Data, Keep Samples, Run''

## પ્રશ્ન 2(b) [4 ગુણા]

નિષ્ણાત સિસ્ટમ શું છે? તેની મર્યાદાઓ અને applications ની યાદી બનાવો.

## જવાબ

**Expert System** એ AI program એ જે specific domains માં complex problems solve કરવા માટે human expert knowledge ને mimic કરે છે.

કોષ્ટક 6. Expert System Overview

પાસું	વિગતો
વ્યાખ્યા	Domain-specific expertise સાથે AI system
ઘટકો	Knowledge base, inference engine, user interface

## Applications:

- **Medical Diagnosis:** રોગ identification systems.
- **Financial Planning:** Investment advisory systems.
- **Fault Diagnosis:** Equipment troubleshooting.

## મર્યાદાઓ:

- **Limited Domain:** ફક્ત specific areas માં કામ કરે છે.
- **Knowledge Acquisition:** Expert knowledge extract કરવું મુશ્કેલ.
- **Maintenance:** Rules update અને modify કરવા મુશ્કેલ.

## મેમરી ટ્રીક

``EXPERT - Explains Problems, Executes Rules, Tests''

## પ્રશ્ન 2(c) [7 ગુણા]

ટોકનાઇઝેશન શું છે? યોગ્ય ઉદાહરણ સાથે સમજાવો.

## જવાબ

Tokenization એ text ને smaller units (tokens) માં break down કરવાની process એ NLP processing માટે.

કોષ્ટક 7. Tokenization ના પ્રકારો

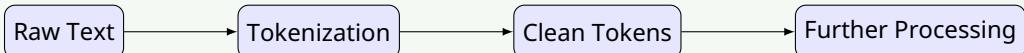
પ્રકાર	વર્ણન	ઉદાહરણ
Word Tokenization	Words દ્વારા split કરે છે	"Hello world" → ["Hello", "world"]
Sentence Tokenization	Sentences દ્વારા split કરે છે	"Hi. How are you?" → ["Hi.", "How are you?"]
Subword Tokenization	Subwords માં split કરે છે	"unhappy" → ["un", "happy"]

## Code ઉદાહરણ:

```

1 import nltk
2 text = "Natural Language Processing is amazing!"
3 tokens = nltk.word_tokenize(text)
4 # Output: ['Natural', 'Language', 'Processing', 'is', 'amazing', '!']

```



આંકડત 3. Process Flow

## મુખ્ય ફાયદા:

- **Standardization:** Text ને uniform format માં convert કરે છે.
- **Analysis Ready:** ML algorithms માટે text prepare કરે છે.
- **Feature Extraction:** Statistical analysis enable કરે છે.

## મેમરી ટ્રીક

“TOKEN - Text Operations Keep Everything Normalized”

## પ્રશ્ન 2(a) OR [3 ગુણ]

સુપરવાઇઝડ અને અનસુપરવાઇઝડ લાર્નિંગની સરખામણી કરો.

## જવાબ

કોષ્ટક 8. Supervised vs Unsupervised Learning

પાસું	Supervised Learning	Unsupervised Learning
<b>Training Data</b>	Target outputs સાથે labeled data	Targets બિના unlabeled data
<b>લક્ષ્ય</b>	Specific outcomes predict કરવા	Hidden patterns discover કરવા
<b>ઉદાહરણો</b>	Classification, Regression	Clustering, Association rules
<b>મૂલ્યાંકન</b>	Accuracy, precision, recall	Silhouette score, elbow method
<b>Applications</b>	Email spam, price prediction	Customer segmentation, anomaly detection

## મેમરી ટ્રીક

“SU - Supervised Uses labels, Unsupervised Uncovers patterns”

## પ્રશ્ન 2(b) OR [4 ગુણ]

હેલ્થકેર, ફાઇનાન્સ અને મેન્યુફ્યુઝચરિંગમાં AI applications વિશે બધું સમજાવો.

## જવાબ

કોષ્ટક 9. Industry પ્રમાણે AI Applications

Industry	Applications	ફાયદા
Healthcare	Medical imaging, drug discovery, diagnosis	Improved accuracy, faster treatment
Finance	Fraud detection, algorithmic trading, credit scoring	Risk reduction, automated decisions
Manufacturing	Quality control, predictive maintenance, robotics	Efficiency, cost reduction

## Healthcare ઉદાહરણો:

- **Medical Imaging:** X-rays અને MRIs માં AI cancer detect કરે છે.
- **Drug Discovery:** AI નીવિ medicine development ની accelerate કરે છે.

**Finance ઉદાહરણો:**

- Fraud Detection:** Real-time transaction monitoring.
- Robo-advisors:** Automated investment management.

**Manufacturing ઉદાહરણો:**

- Quality Control:** Automated defect detection.
- Predictive Maintenance:** Equipment failure prediction.

**મેમરી ટ્રીક**

"HFM - Health, Finance, Manufacturing benefit from AI"

**પ્રશ્ન 2(c) OR [7 ગુણ]**

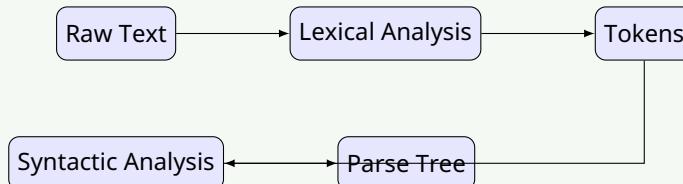
સિન્ટેક્ટિક વિશ્લેષણ શું છે અને તે લેક્સિકલ વિશ્લેષણથી કેવી રીતે અલગ છે?

**જવાબ**

Syntactic Analysis sentences ના grammatical structure ને examine કરે છે, જ્યારે Lexical Analysis text ને meaningful tokens માં break કરે છે.

કોષ્ટક 10. Lexical vs Syntactic Analysis

પાસું	Lexical Analysis	Syntactic Analysis
હેતુ	Text ને words માં tokenize કરવા	Grammatical structure parse કરવા
Input	Raw text	Lexical analysis થી tokens
Output	Tokens, part-of-speech tags	Parse trees, grammar rules
ધ્યાન	Individual words	Sentence structure
ઉદાહરણ	"The cat runs" → [The, cat, runs]	Noun-verb relationship દર્શાવું parse tree બનાવે છે



આકૃતિ 4. Process Flow

**ઉદાહરણ:**

- Lexical:** "She reads books" → ["She", "reads", "books"]
- Syntactic:** "She" ને subject, "reads" ને verb, "books" ને object તરીકે identify કરે છે.

**મુખ્ય તફાવતો:**

- Scope:** Lexical words પર કામ કરે છે, Syntactic sentence structure પર.
- જિટિલતા:** Syntactic analysis lexical કરતાં વધુ complex છે.
- Dependencies:** Syntactic analysis lexical analysis પર depend કરે છે.

**મેમરી ટ્રીક**

"LEX-SYN: LEXical extracts, SYNtactic structures"

**પ્રશ્ન 3(a) [3 ગુણ]**

પ્રતિક્રિયાશીલ મશીનોની વિવિધ લાક્ષણિકતાઓની યાદી બનાવો.

### જવાબ

#### કોષ્ટક 11. Reactive Machines ની લાક્ષણિકતાઓ

લાક્ષણિકતા	વર્ણન
કોઈ મેમરી નથી	Past experiences store કરી શકતા નથી
વર્તમાન-કેન્દ્રિત	ફક્ત current input ને respond કરે છે
નિર્ધારિત	Same input માટે same output આપે છે
કાર્ય-વિશિષ્ટ	Particular functions માટે design કરેલ
કોઈ શીખવું નથી	Experience થી improve કરી શકતા નથી

ઉદાહરણો:

- Deep Blue: IBM નું chess computer.
- Game AI: Tic-tac-toe programs.

### મેમરી ટ્રીક

“REACT - Responds Exactly, Always Consistent Tasks”

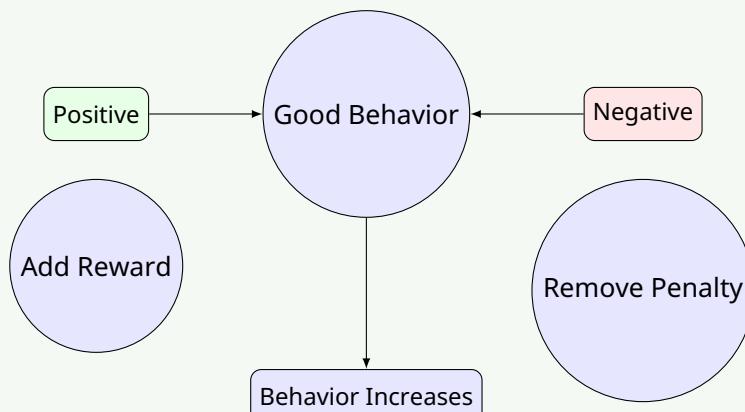
### પ્રશ્ન 3(b) [4 ગુણ]

તફાવત કરો: હકારાતમક મજબૂતીકરણ V/S નકારાતમક મજબૂતીકરણ

### જવાબ

#### કોષ્ટક 12. Positive vs Negative Reinforcement

પાસું	Positive Reinforcement	Negative Reinforcement
વ્યાખ્યા	Good behavior માટે reward add કરવું	Good behavior માટે penalty remove કરવું
Action	કંઈક desirable આપવું	કંઈક undesirable દૂર કરવું
લક્ષ્ય	Desired behavior increase કરવું	Desired behavior increase કરવું
ઉદાહરણ	Correct answer માટે treat આપવું	Good performance માટે extra work દૂર કરવું



આકૃતિ 5. Reinforcement પ્રકારો

મુખ્ય મુદ્દાઓ:

- બંને behavior increase કરે છે પરંતુ વિવિધ mechanisms દ્વારા.
- Positive કંઈક pleasant add કરે છે.
- Negative કંઈક unpleasant remove કરે છે.

## મેમરી ટ્રીક

“PN - Positive adds Nice things, Negative removes Nasty things”

## પ્રશ્ન 3(c) [7 ગુણ]

ટર્મ-ફ્રીકવન્સી-ઇનવર્સ ડોક્યુમેન્ટ ફ્રીકવન્સી (TF-IDF) word embedding technique વિશે બધું સમજાવો.

## જવાબ

TF-IDF એ numerical statistic છે જે documents ના collection માં કોઈ document માટે word કેટલું important છે તે reflect કરે છે.

ફોર્મ્યુલા:

$$\text{TF-IDF} = \text{TF}(t, d) \times \text{IDF}(t)$$

જ્યાં:

- $\text{TF}(t, d) = \frac{\text{Document } d \text{ માં term } t \text{ કેટલી વાર આવે છે}}{\text{Document } d \text{ માં total terms}}$
- $\text{IDF}(t) = \log \left( \frac{\text{Total documents}}{\text{Term } t \text{ ધરાવતી documents}} \right)$

કોષ્ટક 13. TF-IDF ઘટકો

ઘટક	ફોર્મ્યુલા	હેતુ
Term Frequency (TF)	$tf(t, d) = count(t, d) /  d $	Document માં word frequency measure કરે છે
Inverse Document Frequency (IDF)	$idf(t) = \log(N / df(t))$	Corpus માં word importance measure કરે છે
TF-IDF Score	$tf-idf(t, d) = tf(t, d) \times idf(t)$	Final word importance score

#### ઉદાહરણ Calculation:

- Document: "cat sat on mat"
- Term: "cat"
- TF = 1/4 = 0.25
- જો "cat" 10 માંથી 2 documents માં આવે છે: IDF =  $\log(10/2) = 0.699$
- TF-IDF =  $0.25 \times 0.699 = 0.175$

#### Applications:

- Information Retrieval: Search engines.
- Text Mining: Document similarity.
- Feature Extraction: ML preprocessing.

#### ફાયદા:

- Common words ને low scores મળે છે (the, and, is).
- Rare પરંતુ important words ને high scores મળે છે.
- સરળ અને અસરકારક text analysis માટે.

## મેમરી ટ્રીક

“TF-IDF - Term Frequency x Inverse Document Frequency”

## પ્રશ્ન 3(a) OR [3 ગુણ]

ફ્લોડિંગ લોજિક સિસ્ટમ્સ વ્યાખ્યાયિત કરો. તેના મુખ્ય ઘટકોની ચર્ચા કરો.

**જવાબ**

**Fuzzy Logic Systems** uncertainty અને partial truth handle કરે છે, completely true અને completely false વચ્ચે values allow કરે છે.

**કોષ્ટક 14. Fuzzy Logic ઘટકો**

ઘટક	કાર્ય	ઉદાહરણ
<b>Fuzzifier</b>	Crisp inputs ને fuzzy sets માં convert કરે છે	Temperature 75°F → "Warm" (0.7)
<b>Rule Base</b>	If-then fuzzy rules ધરાવે છે	IF temp is warm THEN fan is medium
<b>Inference Engine</b>	Inputs પર fuzzy rules apply કરે છે	Multiple rules combine કરે છે
<b>Defuzzifier</b>	Fuzzy output ને crisp value માં convert કરે છે	"Medium speed" → 60% fan speed

**મેમરી ટ્રીક**

``FRID - Fuzzifier, Rules, Inference, Defuzzifier''

**પ્રશ્ન 3(b) OR [4 ગુણ]**

મજબૂતીકરણ શિક્ષણના ઘટકો સમજાવો: નીતિ, પુરસ્કાર સંકેત, મૂલ્ય કાર્ય, મોડેલ

**જવાબ****કોષ્ટક 15. Reinforcement Learning ઘટકો**

ઘટક	વાખ્યા	હેતુ
<b>Policy (નીતિ)</b>	Actions select કરવાની strategy	Agent ના behavior ને define કરે છે
<b>Reward Signal (પુરસ્કાર સંકેત)</b>	Environment તરફથી feedback	Good/bad actions indicate કરે છે
<b>Value Function (મૂલ્ય કાર્ય)</b>	Expected future rewards	Long-term benefit estimate કરે છે
<b>Model (મોડેલ)</b>	Environment નું agent representation	Next state અને reward predict કરે છે

**વિગતવાર સમજૂતી:**

- Policy ( $\pi$ ):** Deterministic અથવા stochastic હોઈ શકે.
- Reward Signal (R):** Immediate feedback (positive/negative).
- Value Function (V):** Expected long-term return.
- Model:** Environment dynamics predict કરે છે.

**મેમરી ટ્રીક**

``PRVM - Policy chooses, Reward judges, Value estimates, Model predicts''

**પ્રશ્ન 3(c) OR [7 ગુણ]**

તફાવત કરો: આવૃત્તિ-આધારિત v/s આગાહી-આધારિત word embedding તકનીકો.

**જવાબ****કોષ્ટક 16. Frequency-based vs Prediction-based Word Embeddings**

પાસું	Frequency-based	Prediction-based
<b>Approach</b>	Count-based statistics	Neural network prediction
ઉદાહરણો	TF-IDF, Co-occurrence Matrix	Word2Vec, GloVe
<b>Computation</b>	Matrix factorization	Gradient descent
<b>Context</b>	Global statistics	Local context windows
<b>Scalability</b>	Matrix size દ્વારા limited	Vocabulary સાથે scales
<b>Quality</b>	Basic semantic relationships	Rich semantic relationships

#### Frequency-based Methods:

- TF-IDF: Term frequency × Inverse document frequency.
- Co-occurrence Matrix: Word pair frequency counts.

#### Prediction-based Methods:

- Word2Vec: Skip-gram અને CBOW models.
- GloVe: Global Vectors for Word Representation.

#### ફાયદા:

- Frequency-based: સરળ, જડપી (small data), basic similarity માટે સારં.
- Prediction-based: Dense vectors, બહેતર semantics, scalable.

#### મેમરી ટ્રીક

“FP - Frequency counts, Prediction learns”

### પ્રશ્ન 4(a) [3 ગુણ]

પ્રતિક્રિયાશીલ મશીનની મુખ્ય લાક્ષણિકતાઓની ચાદ્રી બનાવો.

#### જવાબ

કોષ્ટક 17. Reactive Machine મુખ્ય લાક્ષણિકતાઓ

લાક્ષણિકતા	વર્ણન
<b>Stateless</b>	Past interactions ની કોઈ memory નથી
<b>Reactive</b>	ફક્ત current inputs ને respond કરે છે
<b>Deterministic</b>	Same inputs માટે consistent outputs
<b>Specialized</b>	Specific tasks માટે designed
<b>Real-time</b>	Stimuli ને immediate response

#### ઉદાહરણો:

- Deep Blue: Chess-playing computer.
- Google AlphaGo: Go-playing system (early version).

#### મેમરી ટ્રીક

“SRDSR - Stateless, Reactive, Deterministic, Specialized, Real-time”

### પ્રશ્ન 4(b) [4 ગુણ]

વિવિધ પૂર્વ-પ્રોસેસિંગ તકનીકોની સૂચિ બનાવો. તેમાંથી કોઈપણ એકને python code વડે સમજાવો.

## જવાબ

### કોષ્ટક 18. Text Pre-processing તકનીકો

તકનીક	હેતુ	ઉદાહરણ
Tokenization	Text ને words માં split કરવું	"Hello world" → ["Hello", "world"]
Stop Word Removal	Common words remove કરવા	"the", "and", "is" remove કરવા
Stemming	Words ને root form માં reduce કરવા	"running" → "run"
Lemmatization	Dictionary form માં convert કરવા	"better" → "good"

**Stemming સમજૂતી:** Stemming suffixes remove કરીને words ને root form માં reduce કરે છે.

Stemming માટે Python Code:

```

1 import nltk
2 from nltk.stem import PorterStemmer
3
4 # Stemmer initialize કરો
5 stemmer = PorterStemmer()
6
7 # Example words
8 words = ["running", "flies", "dogs", "churches", "studying"]
9
10 # Stemming apply કરો
11 stemmed_words = [stemmer.stem(word) for word in words]
12 print(stemmed_words)
13 # Output: ['run', 'fli', 'dog', 'church', 'studi']

```

Stemming ના ફિયદા:

- ML models માટે vocabulary size reduce કરે છે.
- Related words ને together group કરે છે.
- Text analysis efficiency improve કરે છે.

## મેમરી ટ્રીક

“TSSL - Tokenize, Stop-words, Stem, Lemmatize”

## પ્રશ્ન 4(c) [7 ગુણા]

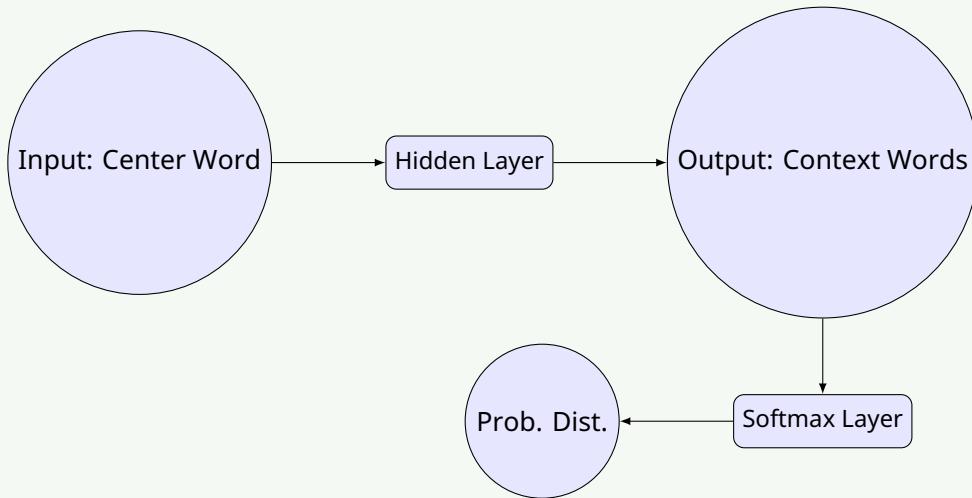
Word2vec તકનીકને વિગતવાર પ્રકાશિત કરો.

## જવાબ

Word2Vec એ neural network-based તકનીક છે જે context predict કરીને words ના dense vector representations શીખે છે.

### કોષ્ટક 19. Word2Vec Architectures

Architecture	Approach	Input	Output
Skip-gram	Center word થી context predict કરે છે	Center word	Context words
CBOW	Context થી center word predict કરે છે	Context words	Center word



આકૃતિ 6. Skip-gram Model

**Training Process:**

- **Sliding Window:** Text पર window move કરો.
- **Word Pairs:** (center, context) pairs બનાવો.
- **Neural Network:** Context predict કરવા માટે train કરો.
- **Weight Matrix:** Word vectors extract કરો.

**Mathematical Concept:**

$$\text{Objective} = \max \sum \log P(\text{context}|\text{center})$$

$$P(\text{context}|\text{center}) = \frac{\exp(v_{\text{context}} \cdot v_{\text{center}})}{\sum \exp(v_w \cdot v_{\text{center}})}$$

**Applications:**

- **Similarity:** Similar words શોધવા.
- **Analogies:** King - Man + Woman = Queen.
- **Feature Engineering:** ML input features.

**મેમરી ટ્રીક**

“W2V - Words to Vectors via neural networks”

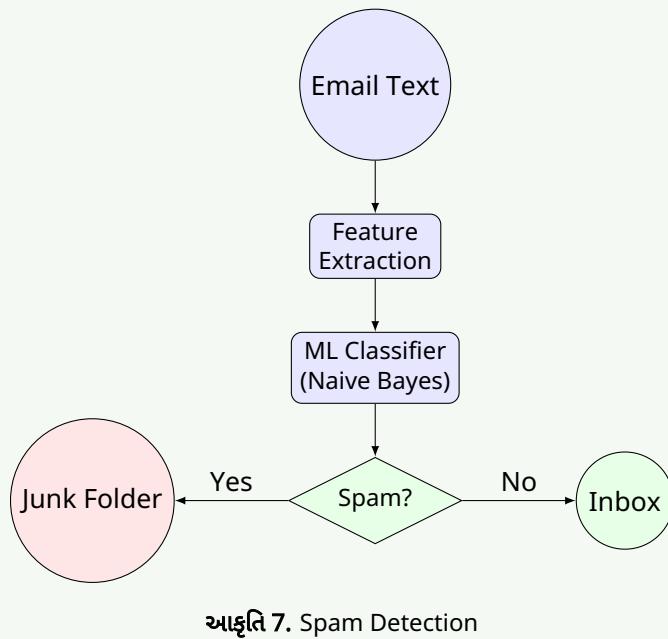
**પ્રશ્ન 4(a) OR [3 ગુણ]**

નેચરલ લેંગ્વેજ પ્રોસેસિંગની કોઈપણ ચાર applications ની યાદી બનાવો. સ્પામ શોધને વિગતવાર સમજાવો.

**જવાબ****કોષ્ટક 20. NLP Applications**

Application	વર્ણન
Spam Detection	Unwanted emails identify કરવા
Sentiment Analysis	Emotional tone determine કરવા
Machine Translation	Languages વચ્ચે translate કરવા
Chatbots	Automated conversation systems

Spam Detection વિગતો:



મેમરી ટ્રીક

“SMTP - Spam, Machine Translation, Sentiment, Phishing detection”

## પ્રશ્ન 4(b) OR [4 ગુણ]

પ્રવચન સંકલન અને વ્યવહારિક વિશ્લેષણ વિશે સમજાવો.

જવાબ

### કોષ્ટક 21. Discourse Integration vs Pragmatic Analysis

પાસું	Discourse Integration	Pragmatic Analysis
ધ્યાન	Text coherence અને structure	Context અને intention
વ્યાપ્તિ	Multiple sentences/paragraphs	Speaker નો intended meaning
ઘટકો	Anaphora, cataphora, connectives	Implicature, speech acts
લક્ષ્ય	Text flow understand કરવું	Real meaning understand કરવું

ઉદાહરણો:

- Discourse: "Mary owns a car. The vehicle is red." ("vehicle" refers to "car").
- Pragmatic: "Can you pass the salt?" (Request, ability વિશે question નથી).

મેમરી ટ્રીક

“DP - Discourse connects, Pragmatics interprets context”

## પ્રશ્ન 4(c) OR [7 ગુણ]

બેગ ઓફ વર્ડ્સ word embedding technique વિશે વિગતવાર અર્થાં કરો.

**જવાબ**

**Bag of Words (BoW)** એ simple text representation method છે જે documents ને unordered collections of words તરીકે treat કરે છે.

કોષ્ટક 22. BoW Process

Step	Description	Example
Vocabulary	Collect unique words	["cat", "sat", "mat"]
Vector	Count occurrences	[1, 1, 1] for "cat sat mat"
Representation	Document → Vector	Vectors form Matrix

**ફાયદા:**

- સરળતા: Understand અને implement કરવા માટે સરળ.
- અસરકારકતા: Spam detection જેવા tasks માટે સારું.

**ગેરફાયદા:**

- કોઈ Word Order નથી.
- Sparse Vectors.
- કાઈ Semantic નથી.

**મેમરી ટ્રીક**

“BOW - Bag Of Words counts occurrences”

**પ્રશ્ન 5(a) [3 ગુણા]**

ન્યુરલ નેટવર્કમાં સંક્ષિપ્ત રીતે કાર્યોની ભૂમિકા શું છે?

**જવાબ**

કોષ્ટક 23. Activation Function ભૂમિકાઓ

ભૂમિકા	વર્ણન
બિન-રૈચીયતા	Complex patterns શીખવાને enable કરે છે
આઉટપુટ નિયંત્રણ	Neuron firing threshold determine કરે છે
Gradient Flow	Backpropagation efficiency ને affect કરે છે
રેઝ મર્યાદા	Output values ને bounds કરે છે

**મેમરી ટ્રીક**

“NOGL - Non-linearity, Output control, Gradient flow, Limiting range”

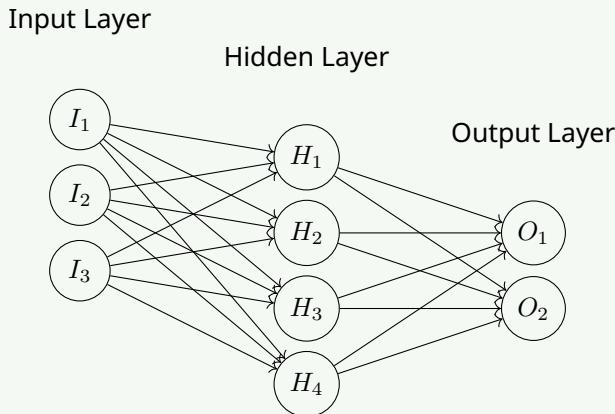
**પ્રશ્ન 5(b) [4 ગુણા]**

ન્યુરલ નેટવર્કના આર્કિટેક્ચરનું વિગતવાર વર્ણન કરો.

**જવાબ**

કોષ્ટક 24. Neural Network Architecture ઘટકો

ઘટક	કાર્ય	ઉદાહરણ
Input Layer	Input data receive કરે છે	Features/pixels
Hidden Layers	Information process કરે છે	Pattern recognition
Output Layer	Final result produce કરે છે	Classification/prediction
Connections	Layers વચ્ચે neurons ને link કરે છે	Weighted edges



આકૃતિ 8. Neural Network Architecture

**Information Flow:**

- Forward Pass: Input → Hidden → Output.
- Weighted Sum:  $\sum(w_i \times x_i + bias)$ .
- Activation: Activation function apply કરો.

**મેમરી ટ્રીક**

"IHOC - Input, Hidden, Output, Connections"

**પ્રશ્ન 5(c) [7 ગુણ]**

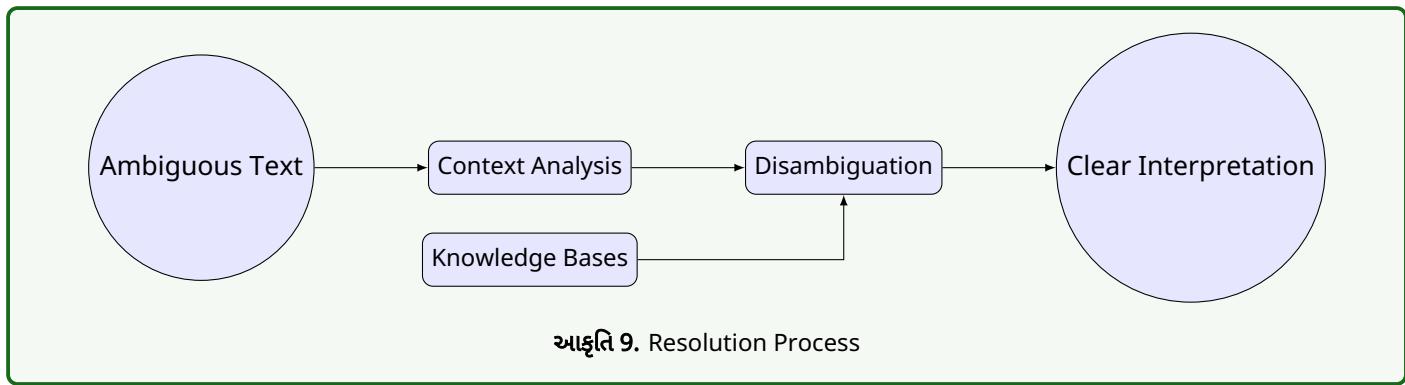
નેચરલ લેંગવેજ પ્રોસેસિંગમાં અસ્પષ્ટતાના પ્રકારોની યાદી બનાવો અને સમજાવો.

**જવાબ**

Ambiguity NLP માં ત્યારે થાય છે જ્યારે text ના multiple possible interpretations હોય છે.

કોષ્ટક 25. NLP Ambiguities ના પ્રકારો

પ્રકાર	વ્યાખ્યા	ઉકેલ
Lexical	Word ના multiple meanings	Context analysis
Syntactic	Multiple parse structures	Grammar rules
Semantic	Multiple sentence meanings	Semantic analysis
Pragmatic	Context-dependent meaning	Intent recognition
Referential	Unclear pronoun reference	Anaphora resolution



## મેમરી ટ્રીક

“LSSPR - Lexical, Syntactic, Semantic, Pragmatic, Referential”

## પ્રશ્ન 5(a) OR [3 ગુણ]

ન્યૂરલ નેટવર્કમાં ઉપયોગમાં લેવાતા કેટલાક લોકપ્રિય સંક્ષિપ્તકરણ કાર્યોના નામોની સૂચિ બનાવો.

## જવાબ

કોષ્ટક 26. લોકપ્રિય Activation Functions

Function	ફોર્મ્યુલા	Range	વપરાશ
ReLU	$f(x) = \max(0, x)$	$[0, \infty)$	Hidden layers
Sigmoid	$f(x) = 1/(1 + e^{-x})$	$(0, 1)$	Binary classification
Tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1, 1)$	Hidden layers
Softmax	$f(x_i) = e^{x_i} / \sum e^{x_j}$	$(0, 1)$	Multi-class output

## મેમરી ટ્રીક

“RSTSL - ReLU, Sigmoid, Tanh, Softmax, Leaky ReLU”

## પ્રશ્ન 5(b) OR [4 ગુણ]

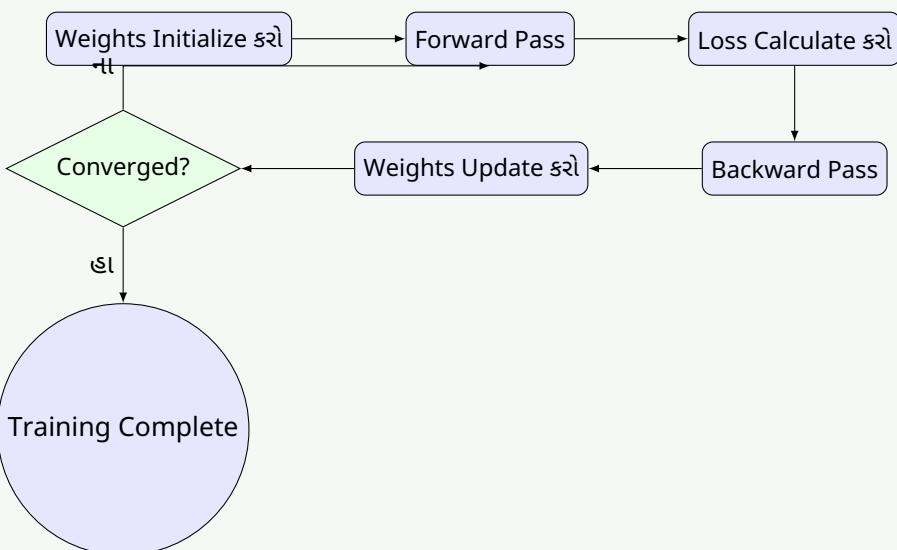
ફૂન્ડેશનિયલ ન્યૂરલ નેટવર્કમાં શીખવાની પ્રક્રિયા સમજાવો.

## જવાબ

**Learning Process** neural networks માં iterative training દ્વારા error minimize કરવા માટે weights અને biases ને adjust કરવાનો સમાવેશ કરે છે.

કોષ્ટક 27. Learning Process Steps

સ્ટેપ	પ્રક્રિયા	વર્ણન
Initialize	Random weights	Small random values સાથે start કરો
Forward Pass	Output calculate કરો	Network દ્વારા input propagate કરો
Calculate Error	Target સાથે compare કરો	Loss function વાપરો
Backward Pass	Gradients calculate કરો	Backpropagation વાપરો
Update Weights	Parameters adjust કરો	Gradient descent apply કરો



આકૃતિ 10. Learning Flow

## મેમરી ટ્રીક

“IFCBU - Initialize, Forward, Calculate, Backward, Update”

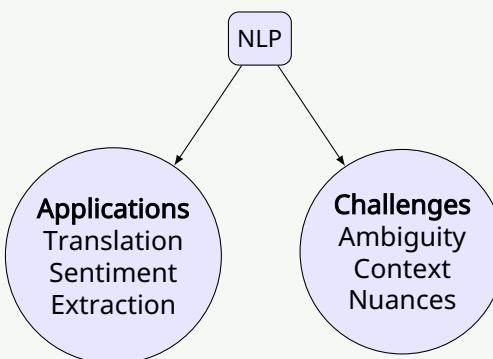
## પ્રશ્ન 5(c) OR [7 ગુણ]

નેચરલ લેંગ્વેજ પ્રોસેસિંગના વિવિધ ફાયદા અને ગેરફાયદાની યાદી બનાવો.

## જવાબ

## કોષ્ટક 28. NLP ફાયદા અને ગેરફાયદા

ફાયદા	ગેરફાયદા
સ્વચાલિત ટેક્સ્ટ વિશ્લેષણ	અસ્પષ્ટતા હેન્ડલિંગ
ભાષા અનુવાદ	સંદર્ભ સમજ
માનવ-કમ્પ્યુટર કિયાપ્રતિક્ષિયા	સાંસ્કૃતિક સૂનઘારતા
માહિતી નિષ્કર્ષણ	કોમ્પ્યુટેશનલ જટિલતા
ભાવના વિશ્લેષણ	ડેટા આવશ્યકતાઓ



આકૃતિ 11. Applications vs Challenges

**મેમરી ટ્રીક**

"ALICE vs ACHDR - Automated, Language, Interaction, Content, Extraction vs Ambiguity, Context, Human-nuances"