

# Subject Name (Gujarati)

4351601 -- Summer 2025

Semester 1 Study Material

*Detailed Solutions and Explanations*

## પ્રશ્ન 1(અ) [3 ગુણ]

વડ એમ્બેડિંગ ટેકનિક શું છે? વિવિધ વડ એમ્બેડિંગ તકનીકોની સૂચિ બનાવો.

### જવાબ

વડ એમ્બેડિંગ એ એવી તકનીક છે જે શબ્દોને આંકડાકીય vectors માં રૂપાંતરિત કરે છે અને શબ્દો વચ્ચેના semantic સંબંધોને જાળવી રાખે છે. આ શબ્દોને high-dimensional space માં dense vectors તરીકે દર્શાવે છે.

ટેબલ: વિવિધ વડ એમ્બેડિંગ તકનીકો

તકનીક	વર્ણન	મુખ્ય લક્ષણ
TF-IDF	Term Frequency-Inverse Document Frequency	આંકડાકીય માપદંડ
Bag of Words (BoW)	આવર્તન-આધારિત રજૂઆત	સરળ ગણતરી પદ્ધતિ
Word2Vec	Neural network-આધારિત embedding	Semantic સંબંધો કેપ્ચર કરે
GloVe	Global Vectors for word representation	Global અને local આંકડા સંયોજન

### મુખ્ય પોઇન્ટ્સ:

- TF-IDF: દસ્તાવેજોમાં શબ્દની મહત્વતા માપે છે
- BoW: Vocabulary-આધારિત vectors બનાવે છે
- Word2Vec: CBOW અને Skip-gram models વાપરે છે
- GloVe: Global context સાથે pre-trained embeddings

### મેમરી ટ્રીક

“TB-WG” (TF-IDF, BoW, Word2Vec, GloVe)

## પ્રશ્ન 1(બ) [4 ગુણ]

આર્ટિફિશિયલ ઇન્ટેલિજન્સના વિવિધ પ્રકારોનું વર્ગીકરણ કરો અને તેને ડાયાગ્રામ વડે દર્શાવો.

### જવાબ

AI ને ક્ષમતાઓ અને કાર્યક્ષમતા આધારે વર્ગીકૃત કરી શકાય છે.

ડાયાગ્રામ:

### Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph TD
    A[Artificial Intelligence] --> B[Narrow AI/Weak AI]
    A --> C[General AI/Strong AI]
    A --> D[Super AI]

    B --> E[Reactive Machines]
    B --> F[Limited Memory]
```

```

C {-{-}{}} I [Theory of Mind]
C {-{-}{}} J [Self{-}Awareness]
{Highlighting}
{Shaded}

```

ટેબલ: AI પ્રકારોની તુલના

વર્ગ	પ્રકાર	વર્ણન	ઉદાહરણ
ક્ષમતાઓ	Narrow AI	કાર્ય-વિશિષ્ટ બુદ્ધિ	Siri, Chess programs
	General AI	માનવ-સ્તરની બુદ્ધિ	હજુ પ્રાપ્ત નથી
કાર્યક્ષમતા	Super AI	માનવ બુદ્ધિથી વધુ	સૈદ્ધાંતિક ઘ્યાલ
	Reactive	કોઈ ચાદરાશત નથી	Deep Blue
	Limited Memory	ભૂતકાળના ડેટાનો ઉપયોગ	Self-driving cars

### મેમરી ટ્રીક

“NGS-RLT” (Narrow-General-Super, Reactive-Limited-Theory)

## પ્રશ્ન 1(ક) [7 ગુણ]

તફાવત આપીને NLU અને NLG સમજાવો.

### જવાબ

**Natural Language Understanding (NLU)** અને **Natural Language Generation (NLG)** Natural Language Processing ના બે મુખ્ય ઘટકો છે.

ટેબલ: NLU vs NLG તુલના

પાસું	NLU	NLG
હેતુ	માનવી ભાષાને સમજવું	માનવી ભાષા જનરેટ કરવું
દિશા	Input processing	Output generation
કાર્ય	અર્થનું અર્થદાટન	ટેક્સ્ટ રચના
પ્રક્રિયા	વિશ્લેષણ અને સમજ	સંશ્લેષણ અને સર્જન
ઉદાહરણો	Intent recognition, sentiment analysis	Chatbot responses, report generation
પડકારો	અસ્પષ્ટતા નિવારણ	Natural text generation

### વિગતવાર સમજાવટ:

#### NLU (Natural Language Understanding):

- Unstructured text ને structured data માં કન્વર્ટ કરે છે
- Semantic analysis અને intent extraction કરે છે
- અસ્પષ્ટતા અને context ની સમજ હેન્ડલ કરે છે

#### NLG (Natural Language Generation):

- Structured data ને natural language માં કન્વર્ટ કરે છે
- સુસંગત અને contextually યોગ્ય ટેક્સ્ટ બનાવે છે
- વ્યાકરણની ચુસ્તતા અને પ્રવાહિતા સુનિશ્ચિત કરે છે

### મેમરી ટ્રીક

“UI-OG” (Understanding Input, Output Generation)

## પ્રશ્ન 1(ક) OR [7 ગુણ]

આઇન્ફોર્મેશન ઇન્ટેલિજન્સનો ઉપયોગ થાય છે તેવા વિવિધ ઉદ્યોગોની યાદી બનાવો અને કોઈપણ બેને સમજાવો.

## જવાબ

ટેબલ: ઉદ્યોગમાં AI એપ્લિકેશન

ઉદ્યોગ	AI એપ્લિકેશન	લાભો
આરોગ્ય	નિદાન, દવા શોધ	ચુસ્તતામાં સુધારો
ફાઇનાન્સ	છેતરપિંડી શોધ, ટ્રેડિંગ	જોખમ વ્યવસ્થાપન
ઉત્પાદન	ગુણવત્તા નિયંત્રણ	કાર્યક્ષમતા
પરિવહન	સ્વાયત્ત વાહનો	સુરક્ષા
રિટેલ	સુલેખન સિસ્ટમ	વ્યક્તિગત વ્યવસ્થાપન
શિક્ષણ	વ્યક્તિગત શિક્ષણ	અનુકૂલન શિક્ષણ

બે ઉદ્યોગોની વિગતવાર સમજાવટ:

1. આરોગ્ય ઉદ્યોગ:

- તબીબી નિદાન: AI તબીબી છબીઓ અને દર્દીના ડેટાનું વિશ્લેષણ કરે છે
- દવા શોધ: સંભવિત દવાઓની ઝડપી ઓળખ
- વ્યક્તિગત સારવાર: દર્દીના genetics આધારે ઉપયાર
- લાભો: ઝડપી નિદાન, ભૂલો ઘટાડવી, પરિણામોમાં સુધારો

2. ફાઇનાન્સ ઉદ્યોગ:

- છેતરપિંડી શોધ: Real-time માં શક્યાસ્પદ વ્યવહારો ઓળખવા
- Algorithmic Trading: બજારના patterns આધારે automated trading
- Credit Scoring: લોન ડિફોલ્ટ જોખમનું ચોક્કસ મૂલ્યાંકન
- લાભો: વર્દેલી સુરક્ષા, ઝડપી પ્રક્રિયા, વધુ સારું જોખમ વ્યવસ્થાપન

## મેમરી ટ્રીક

"HF-MR-TE" (Healthcare-Finance, Manufacturing-Retail-Transportation-Education)

## પ્રશ્ન 2(અ) [૩ ગુણ]

મશીન લર્નિંગ શબ્દને વ્યાખ્યાયિત કરો. મશીન લર્નિંગનું વર્ગીકરણ રેખાકૃતિ દોરો.

## જવાબ

મશીન લર્નિંગ AI નો ઉપબિભાગ છે જે કોમ્પ્યુટરોને સ્પષ્ટ રીતે પ્રોગ્રામ કર્યા વિના અનુભવથી શીખવા અને સુધારવા સક્ષમ બનાવે છે. આ ડેટાનું વિશ્લેષણ કરવા, patterns ઓળખવા અને predictions કરવા algorithms નો ઉપયોગ કરે છે.

ડાયગ્રામ:

### Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph TD
    A[Machine Learning] --> B[Supervised Learning]
    A --> C[Unsupervised Learning]
    A --> D[Reinforcement Learning]

    B --> E[Classification]
    B --> F[Regression]

    C --> G[Clustering]
    C --> H[Association]

    D --> I[Model-based]
    D --> J[Model-free]
```

### મુખ્ય પોઈન્ટ્સ:

- Supervised:** Labeled training data વાપરે છે
- Unsupervised:** Unlabeled data માં patterns શોધે છે
- Reinforcement:** Rewards અને penalties દ્વારા શીખે છે

## પ્રશ્ન 2(બ) [4 ગુણ]

Positive reinforcement અને Negative reinforcement નો તફાવત દર્શાવો.

## જવાબ

## ટેબલ: Positive vs Negative Reinforcement

પાસું	Positive Reinforcement	Negative Reinforcement
વ્યાખ્યા	સારા વર્તન માટે રિવોર્ડ ઉમેરવું	અપ્રિય stimulus દૂર કરવું
કિયા	કંઈક આનંદાયક આપવું	કંઈક અપ્રિય દૂર કરવું
હેતુ	ઇચ્છિત વર્તન વધારવું	ઇચ્છિત વર્તન વધારવું
ઉદાહરણ	સારા પ્રદર્શન માટે બોનિસ	જાગ્યા પછી alarm બંધ કરવું
અસર	Rewards દ્વારા પ્રેરણા	રાહત દ્વારા પ્રેરણા
Agent પ્રતિસાદ	કિયા પુનરાવર્તન કરવી	નકારાત્મક પરિણામો ટાળવા

## મુખ્ય પોઇન્ટ્સ:

- Positive Reinforcement: Positive stimulus ઉમેરીને વર્તન મજબૂત બનાવે છે
- Negative Reinforcement: Negative stimulus દૂર કરીને વર્તન મજબૂત બનાવે છે
- બંને પ્રકાર: ઇચ્છિત વર્તનની સંભાવના વધારવાનું લક્ષ્ય છે
- તફાવત: પ્રોત્સાહનની પદ્ધતિ (ઉમેરવું VS દૂર કરવું)

## મેમરી ટ્રીક

## પ્રશ્ન 2(ક) [7 ગુણ]

Supervised અને Unsupervised learning ની તુલના કરો.

## જવાબ

## ટેબલ: Supervised vs Unsupervised Learning

પેરામેટર	Supervised Learning	Unsupervised Learning
ડેટા પ્રકાર	Labeled data (input-output pairs)	Unlabeled data (માત્ર inputs)
શીખવાનું લક્ષ્ય	પરિણામોની આગાહ	છુપા patterns શોધવા
Feedback	સાચા જવાબો છે	સાચા જવાબો નથી
Algorithms	SVM, Decision Trees, Neural Networks	K-means, Hierarchical clustering
એપ્લિકેશન	Classification, Regression	Clustering, Association rules
ચોક્સાઈ	માપી શકાય છે	માપવી મુશ્કેલ
જટિલતા	ઓછી જટિલ	વધુ જટિલ
ઉદાહરણો	Email spam detection, કિંમત આગાહ	Customer segmentation, Market basket analysis

વિગતવાર તુલના:

**Supervised Learning:**

- જાણીતા પરિણામો સાથે training data ની જરૂર
- પ્રદર્શનનું મૂલ્યાંકન સરળતાથી કરી શકાય છે
- આગાહીના કાર્યો માટે વપરાય છે

**Unsupervised Learning:**

- પૂર્વ-નિર્ધારિત labels વિના ડેટા સાથે કામ કરે છે
- ડેટામાં છુપાયેલા structures શોધે છે
- અન્વેષણાત્મક ડેટા વિશ્લેષણ માટે વપરાય છે

મેમરી ટ્રીક

“LP-PF” (Labeled Prediction, Pattern Finding)

**પ્રશ્ન 2(અ) OR [3 ગુણ]**

વાખ્યાયિત કરો: Classification, Regression અને clustering.

જવાબ

ટેબલ: ML કાર્યોની વાખ્યાઓ

કાર્ય	વાખ્યા	આઉટપુટ પ્રકાર	ઉદાહરણ
<b>Classification</b>	Discrete categories/classes ની આગાહ	Categorical	Email: Spam/Not Spam
<b>Regression</b>	સતત અંકડાકીય મૂલ્યોની આગાહ	અંકડાકીય	ઘરની કિંમત આગાહ
<b>Clustering</b>	સમાન ડેટા points ને જૂથ બનાવવા	જૂથો/Clusters	Customer segmentation

વિગતવાર વાખ્યાઓ:

- Classification:** શીખેલા patterns આધારે input data ને પૂર્વ-નિર્ધારિત વર્ગોમાં સૌંપે છે
- Regression:** સતત મૂલ્યોની આગાહ કરવા variables વચ્ચેના સંબંધોનો અંદાજ કાઢે છે
- Clustering:** જૂથોની પૂર્વ જાણકારી વિના ડેટામાં ફુદરતી જૂથો શોધે છે

મેમરી ટ્રીક

“CRC” (Categories, Real numbers, Clusters)

**પ્રશ્ન 2(બ) OR [4 ગુણ]**

Artificial Neural Network અને Biological Neural Network ની તુલના કરો.

જવાબ

ટેબલ: ANN vs Biological Neural Network

પાસું	Artificial Neural Network	Biological Neural Network
પ્રોસેસિંગ	Digital/Binary	Analog
જડપ	જડપી પ્રોસેસિંગ	ધીમી પ્રોસેસિંગ
શીખવું	Backpropagation algorithm	Synaptic plasticity
મેમરી	અલગ સ્ટોરેજ	ક્રેક્શનમાં વિતરિત
સ્ટક્ચર	સ્તરવાર આર્કિટેક્ચર	જટિલ 3D structure
ઓટ સહન	ઓછું	વધુ
ઉર્જા	વધુ પાવર consumption	ઓછો ઉર્જા વપરાશ

સમાંતર  
પ્રક્રિયા

મધ્યદિન parallel processing

વિશાળ parallel processing

#### મુખ્ય તફાવતો:

- **ANN:** મગજથી પ્રેરિત ગાળિતિક મોડલ
- **Biological:** વાસ્તવિક મગજના neural networks
- **હેતુ:** ANN computation માટે, Biological cognition માટે
- **અનુકૂળનક્ષમતા:** Biological networks વધુ flexible

#### મેમરી ટ્રીક

“DSML-CFEP” (Digital-Speed-Memory-Layer vs Complex-Fault-Energy-Parallel)

### પ્રશ્ન 2(ક) OR [7 ગુણ]

Supervised, unsupervised અને reinforcement learning ની વિવિધ applications ની સૂચિ બનાવો.

#### જવાબ

#### ટેબલ: વિવિધ Learning પ્રકારોની Applications

Learning પ્રકાર	Applications	વાસ્તવિક જગતના ઉદાહરણો
Supervised	Email classification, તબીબી નિદાન, Stock prediction, Credit scoring	Gmail spam filter, X-ray analysis, Trading algorithms
Unsupervised	Customer segmentation, Anomaly detection, Data compression	Market research, Fraud detection, Image compression
Reinforcement	Game playing, Robotics, Autonomous vehicles, Resource allocation	AlphaGo, Robot navigation, Self-driving cars

#### વિગતવાર Applications:

##### Supervised Learning:

- **Classification:** Spam detection, sentiment analysis, image recognition
- **Regression:** કિંમત આગાહ, હવામાન આગાહ, વેચાણ અંદાજ

##### Unsupervised Learning:

- **Clustering:** Market segmentation, gene sequencing, recommendation systems
- **Association:** Market basket analysis, web usage patterns

##### Reinforcement Learning:

- **Control Systems:** Robot control, traffic management
- **Optimization:** Resource scheduling, portfolio management

#### મેમરી ટ્રીક

“SCR-CRO” (Supervised-Classification-Regression,  
Reinforcement-Control-Optimization)

Unsupervised-Clustering-Association,

### પ્રશ્ન 3(અ) [3 ગુણ]

સિંગલ લેયર ફોરવર્ક નેટવર્કને યોગ્ય ડાયાગ્રામ સાથે સમજાવો.

## જવાબ

સિંગલ લેચર ફોરવર્ક નેટવર્ક (Perceptron) એ સૌથી સરળ neural network છે જેમાં input અને output વચ્ચે weights નો એક સ્તર હોય છે.

ડાયગ્રામ:

### Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    X1[Input X1] -->|W1| S1[S]
    X2[Input X2] -->|W2| S1
    X3[Input X3] -->|W3| S1
    B[Bias b] -->|S| S1
    S1 -->|A| A[Activation Function]
    A -->|Y| Y[Output Y]
{Highlighting}
{Shaded}
```

ઘટકો:

- Inputs:** X1, X2, X3 (feature values)
- Weights:** W1, W2, W3 (connection strengths)
- Bias:** Threshold adjustment માટે વધારાણું parameter
- Summation:** Inputs નો weighted sum
- Activation:** Output બનાવવા માટેનું function

ગાણિતિક સૂત્ર:  $Y = f(\sum(Wi \times Xi) + b)$

## મેમરી ટ્રીક

“IWSA” (Input-Weight-Sum-Activation)

## પ્રશ્ન 3(બ) [4 ગુણ]

Backpropagation પર ઢૂંઢી નોંધ લખો.

## જવાબ

Backpropagation એ supervised learning algorithm છે જે error calculation આધારે weights adjust કરીને neural networks ને train કરવા માટે વપરાય છે.

ટેબલ: Backpropagation પ્રક્રિયા

તબક્કો	વર્ણન	કિયા
<b>Forward Pass</b> <b>Error Calculation</b>	Input network દ્વારા આગળ વધે છે Output ને target સાથે compare કરવું	Output ની ગણતરી Error/loss શોધવો
<b>Backward Pass</b> <b>Weight Update</b>	Error પાછળની દિશામાં વધે છે Gradient વાપરીને weights adjust કરવા	Weights update કરવા Error ઘટાડવો

મુખ્ય લક્ષણો:

- Gradient Descent:** Optimal weights શોધવા માટે calculus વાપરે છે
- Chain Rule:** દરેક weight ની error contribution ની ગણતરી
- Iterative Process:** Convergence સુધી પુનરાવર્તન
- Learning Rate:** Weight updates ની ઝડપ નિયમિત કરે છે

પગલાં:

- Random weights initialize કરવા
- Output મેળવવા forward propagation
- Actual અને predicted વચ્ચે error ની ગણતરી
- Weights update કરવા backward propagation

## પ્રશ્ન 3(ક) [7 ગુણ]

ફીડ ફોરવર્ક ન્યૂરોન નેટવર્કના આર્કિટેક્ચરના components સમજાવો.

## જવાબ

ફીડ ફોરવર્ક ન્યૂરલ નેટવર્ક અનેક સ્તરો ધરાવે છે જ્યાં માહિતી input થી output સુધી એક દિશામાં વહે છે.  
ડાયગ્રામ:

## Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph TD
    subgraph Input Layer
        I1[X1]
        I2[X2]
        I3[X3]
    end

    subgraph Hidden Layer
        H1[N1]
        H2[N2]
        H3[N3]
    end

    subgraph Output Layer
        O1[Y1]
        O2[Y2]
    end

    I1 {--{-}{-->} H1}
    I1 {--{-}{-->} H2}
    I1 {--{-}{-->} H3}
    I2 {--{-}{-->} H1}
    I2 {--{-}{-->} H2}
    I2 {--{-}{-->} H3}
    I3 {--{-}{-->} H1}
    I3 {--{-}{-->} H2}
    I3 {--{-}{-->} H3}

    H1 {--{-}{-->} O1}
    H1 {--{-}{-->} O2}
    H2 {--{-}{-->} O1}
    H2 {--{-}{-->} O2}
    H3 {--{-}{-->} O1}
    H3 {--{-}{-->} O2}

{Highlighting}
{Shaded}
```

## ઘટકો:

## 1. Input Layer:

- Raw data મેળવે છે
- કોઈ processing નથી, માત્ર વિતરણ
- Neurons ની સંખ્યા = features ની સંખ્યા

## 2. Hidden Layer(s):

- Computation અને transformation કરે છે
- Activation functions ધરાવે છે

- અનેક hidden layers હોઈ શકે છે

### 3. Output Layer:

- અંતિમ પરિણામો ઉત્પન્ન કરે છે
- Neurons ની સંખ્યા = outputs ની સંખ્યા
- Task પ્રકાર માટે થોળું activation વાપરે છે

### 4. Weights અને Biases:

- **Weights:** Neurons વચ્ચેની connection strengths
- **Biases:** Threshold adjustment parameters

### 5. Activation Functions:

- Non-linearity દાખલ કરે છે
- સામાન્ય પ્રકારો: ReLU, Sigmoid, Tanh

## મેમરી ટ્રીક

"IHO-WA" (Input-Hidden-Output, Weights-Activation)

## પ્રશ્ન 3(અ) OR [3 ગુણ]

મલ્ટિલેયર ફીડ ફોરવર્ક ANN ને ડાયાગ્રામ સાથે સમજાવો.

### જવાબ

મલ્ટિલેયર ફીડ ફોરવર્ક ANN માં input અને output layers વચ્ચે અનેક hidden layers હોય છે, જે જટિલ pattern recognition સક્ષમ બનાવે છે.

ડાયાગ્રામ:

### Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    subgraph Input
        I1[X1]
        I2[X2]
    end

    subgraph Hidden1
        H11[H1]
        H12[H2]
    end

    subgraph Hidden2
        H21[H1]
        H22[H2]
    end

    subgraph Output
        O1[Y]
    end

    I1 {--{-}{-->} H11}
    I1 {--{-}{-->} H12}
    I2 {--{-}{-->} H11}
    I2 {--{-}{-->} H12}

    H11 {--{-}{-->} H21}
    H11 {--{-}{-->} H22}
    H12 {--{-}{-->} H21}
    H12 {--{-}{-->} H22}

    H21 {--{-}{-->} O1}
```

H22 {-{-}{}} 01  
 {Highlighting}  
 {Shaded}

#### લક્ષણો:

- **Deep Architecture:** અનેક hidden layers
- **જાટિલ Patterns:** Non-linear relationships શીખી શકે છે
- **Universal Approximator:** કોઈપણ સતત function નો અંદાજ લગાવી શકે છે

#### મેમરી ટ્રીક

“MDC” (Multiple layers, Deep learning, Complex patterns)

### પ્રશ્ન 3(બ) OR [4 ગુણ]

સમજાવો ‘ReLU એ સૌથી વધુ ઉપયોગમાં લેવાતું Activation function છે.’

#### જવાબ

**ReLU (Rectified Linear Unit)** તેની સરળતા અને deep networks માં અસરકારકતાને કારણે વ્યાપક રીતે વપરાય છે.  
 ટેબલ: ReLU કેમ લોકપ્રીય છે

ફાયદો	વર્ણન	લાભ
<b>Computational Efficiency</b>	સરળ max(0,x) operation	જડપી processing
<b>Gradient Flow</b>	Positive values માટે vanishing gradient નથી	વધુ સારું learning
<b>Sparsity</b>	Negative inputs માટે zero output	કાર્યક્ષમ representation
<b>Non-linearity</b>	Non-linear વર્તન દાખલ કરે છે	જાટિલ pattern learning

ગાણિતિક વ્યાખ્યા:  $f(x) = \max(0, x)$

અન્ય Functions સાથે તુલના:

- vs Sigmoid: Saturation સમસ્યા નથી, જડપી computation
- vs Tanh: સરળ ગણતરી, વધુ સારો gradient flow
- મર્યાદાઓ: Negative inputs માટે dead neurons સમસ્યા

સૌથી સામાન્ય કેમ:

- Vanishing gradient સમસ્યા હલ કરે છે
- Computationally કાર્યક્ષમ
- વ્યવહારમાં સારું કામ કરે છે
- Hidden layers માટે default પસંદગી

#### મેમરી ટ્રીક

“CGSN” (Computational, Gradient, Sparsity, Non-linear)

### પ્રશ્ન 3(ક) OR [7 ગુણ]

Artificial Neural Network ની સ્ટેપ બાય સ્ટેપ લર્નિંગ પ્રક્રિયા સમજાવો.

#### જવાબ

**ANN Learning Process** માં prediction error ઘટાડવા માટે iterative weight adjustment સામેલ છે.  
 ટેબલ: સ્ટેપ-બાય-સ્ટેપ લર્નિંગ પ્રક્રિયા

સ્ટેપ	પ્રક્રિયા	વર્ણન
<b>1. Initialization</b>	Random weights સેટ કરવા	નાના random values
<b>2. Forward Propagation</b>	Output ની ગણતરી	Input → Hidden → Output
<b>3. Error Calculation</b>	Target સાથે સરખામણી	Loss function computation

4. Backward Propagation	Gradients ની ગણતરી	Error → Hidden ← Input
5. Weight Update	Parameters adjust કરવા	Gradient descent
6. Iteration	પ્રક્રિયા પુનરાવર્તન	Convergence સુધી

#### વિગતવાર સ્ટેપ્સ:

##### સ્ટેપ 1: Weights Initialize કરવા

- બધા weights અને biases ને નાના random values સોંપવા
- Symmetry breaking સમસ્યા અટકાવે છે

##### સ્ટેપ 2: Forward Propagation

- Input data network layers દ્વારા આગળ વહે છે
- દરેક neuron weighted sum + activation ની ગણતરી કરે છે

##### સ્ટેપ 3: Error ની ગણતરી

- Network output ને desired output સાથે compare કરવું
- MSE અથવા Cross-entropy જેવા loss functions વાપરવા

##### સ્ટેપ 4: Backward Propagation

- દરેક weight માટે error gradient ની ગણતરી
- Error પાછળની દિશામાં propagate કરવા chain rule વાપરવું

##### સ્ટેપ 5: Weights Update કરવા

- Gradient descent વાપરીને weights adjust કરવા
- New\_weight = Old\_weight - (learning\_rate × gradient)

##### સ્ટેપ 6: પ્રક્રિયા પુનરાવર્તન

- Error converge થાય અથવા maximum epochs સુધી ચાલુ રાખવું
- Overfitting ટાજવા validation performance monitor કરવું

#### મેમરી ટ્રીક

"IFEBWI" (Initialize-Forward-Error-Backward-Weight-Iterate)

#### પ્રશ્ન 4(અ) [3 ગુણ]

નેચરલ લેંગ્વેજ પ્રોસેસિંગના વિવિધ ફાયદા અને ગેરફાયદાની યાદી બનાવો.

#### જવાબ

##### ટેબલ: NLP ફાયદા અને ગેરફાયદા

ફાયદા	ગેરફાયદા
Automation of text processing	અસ્પેષ્ટતા in human language
24/7 ઉપલબ્ધતા customer service માટે	Context Understanding પડકારો
બહુભાષીય સપોર્ટ ક્ષમતાઓ	સાંસ્કૃતિક સૂક્ષ્મતાઓ મુશ્કેલી
સ્ક્રેલેબિલિટી મોટા datasets માટે	ઉચ્ચ Computational જરૂરિયાતો
સુસંગતતા responses માં	કેટા ગોપીનીયતા ચિંતાઓ
લાગત ઘટાડવી operations માં	મધ્યાદિત સર્જનાત્મકતા responses માં

#### મુખ્ય પોઇન્ટ્સ:

- ફાયદા: કાર્યક્ષમતા, સુલભતા, સુસંગતતા
- ગેરફાયદા: જટિલતા, resource જરૂરિયાતો, મર્યાદાઓ
- સંતુલન: ધાર્યી applications માં ફાયદા પડકારો કરતાં વધુ છે

#### મેમરી ટ્રીક

"AMS-ACC" (Automation-Multilingual-Scalability vs Ambiguity-Context-Computational)

#### પ્રશ્ન 4(બ) [4 ગુણ]

NLP માં પ્રી-પ્રોસેસિંગ તકનીકોની સૂચિ બનાવો અને પાયથોન પ્રોગ્રામ વડે કોઈપણ એકને demonstrate કરો.

## જવાબ

ટેબલ: NLP પ્રીપોરેસિંગ તકનીકો

તકનીક	હેતુ	ઉદાહરણ
Tokenization	ટેક્સ્ટને words/sentences માં વિભાજન	“Hello world” → ["Hello", "world"]
Stop Words Removal	સામાન્ય શબ્દો દૂર કરવા	“the”, “is”, “and” દૂર કરવા
Stemming	શબ્દને root form માં ઘટાડવા	“running” → “run”
Lemmatization	Dictionary form માં કન્વર્ટ કરવું	“better” → “good”
POS Tagging	Parts of speech ઓળખવા	“run” → verb
Named Entity Recognition	Entities ઓળખવા	“Apple” → Organization

Python પ્રોગ્રામ - Tokenization:

```
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize

# Sample text
text = "Natural Language Processing . . ."

# Word tokenization
words = word_tokenize(text)
print("Words:", words)

# Sentence tokenization
sentences = sent_tokenize(text)
print("Sentences:", sentences)
```

## મેમરી ટ્રીક

“TSSL-PN” (Tokenization-Stop-Stemming-Lemmatization, POS-NER)

## પ્રશ્ન 4(ક) [7 ગુણ]

NLP ના phases સમજાવો.

## જવાબ

NLP Phases natural language ને process અને સમજવા માટેના વ્યવસ્થિત અભિગમને દર્શાવે છે.

ટેબલ: NLP Phases

તબક્કો	વર્ણન	પ્રક્રિયા	ઉદાહરણ
Lexical Analysis	Tokenization અને word identification	ટેક્સ્ટને tokens માં વિભાજન	“હું ખુશ છું” → [“, ”, ””]
Syntactic Analysis	વ્યાકરણ અને વાક્ય structure	Parse trees, POS tagging	Noun, verb, adjective ઓળખવા
Semantic Analysis	અર્થ extraction	Word sense disambiguation	“બેંક” → financial vs
Discourse Integration	વાક્યો પારના context	Pronouns, references resolve કરવા	“તે” refers to “જોન”
Pragmatic Analysis	Intent અને context understanding	Situation/culture consider કરવું	વંગા, idioms interpretation

### વિગતવાર સમજાવદ:

#### 1. Lexical Analysis:

- NLP pipeline નો પ્રથમ તબક્કો
- Character stream ને tokens માં કન્વર્ટ કરે છે
- Punctuation અને special characters દૂર કરે છે

#### 2. Syntactic Analysis:

- વ્યાકરણનું structure વિશ્લેષણ કરે છે
- Parse trees બનાવે છે
- વાક્યના ઘટકો ઓળખે છે

#### 3. Semantic Analysis:

- ટેક્સ્ટમાથી અર્થી extract કરે છે
- શબ્દની અસ્પષ્ટતા handle કરે છે
- શબ્દોને concepts સાથે map કરે છે

#### 4. Discourse Integration:

- વાક્ય સ્તર પાર ટેક્સ્ટનું વિશ્લેષણ કરે છે
- વાક્યો પાર context જાળવે છે
- References અને connections resolve કરે છે

#### 5. Pragmatic Analysis:

- વાસ્તવિક જગતનો context consider કરે છે
- Speaker નો intent સમજે છે
- રૂપક ભાષા handle કરે છે

મેરોડિએક ડાયગ્રામ:

Mermaid Diagram (Code)

```

{Shaded}
{Highlighting} []
graph LR
    A[Raw Text] --> B[Lexical Analysis]
    B --> C[Syntactic Analysis]
    C --> D[Semantic Analysis]
    D --> E[Discourse Integration]
    E --> F[Pragmatic Analysis]
    F --> G[Understanding]
    {Highlighting}
    {Shaded}
  
```

### મેમરી ટ્રીક

“LSSDP” (Lexical-Syntactic-Semantic-Discourse-Pragmatic)

### પ્રશ્ન 4(અ) OR [3 ગુણ]

નેચરલ લેંગ્વેજ પ્રોસેસિંગ શું છે? તેની applications ની યાદી બનાવો.

#### જવાબ

નેચરલ લેંગ્વેજ પ્રોસેસિંગ (NLP) AI ની એક શાખા છે જે કોમ્પ્યુટરોને માનવી ભાષાને અર્થપૂર્ણ રીતે સમજવા, અર્થધારણ કરવા અને generate કરવા સક્ષમ બનાવે છે.

ટેબલ: NLP Applications

વર્ગ	Applications	ઉદાહરણો
કોમ્પ્યુનિકેશન	Chatbots, Virtual assistants	Siri, Alexa, ChatGPT
અનુવાદ	ભાષા અનુવાદ	Google Translate
વિશ્લેષણ	Sentiment analysis, Text mining	Social media monitoring
શોધ	માહિતી પુનઃપ્રાપ્તિ	Search engines
લેખન	વ્યાકરણ તપાસ, Auto-complete	Grammarly, predictive text
પિંગનેસ	દસ્તાવેજ processing, Spam detection	Email filtering

### મુખ્ય Applications:

- મશીન અનુવાદ: ભાષાઓ વચ્ચે ટેક્સ્ટ કન્વર્ટ કરવું
- સ્પીચ રેકાયિશન: વાણીને ટેક્સ્ટમાં કન્વર્ટ કરવું
- ટેક્સ્ટ સમરાઇઝન: સંક્ષિપ્ત સારાંશ બનાવવા
- પ્રશ્ન જવાબ: પ્રશ્નોના જવાબો આપવા

### મેમરી ટ્રીક

“CTAS-WB” (Communication-Translation-Analysis-Search, Writing-Business)

### પ્રશ્ન 4(બ) OR [4 ગુણ]

NLTK માં WordNet સાથે કરવામાં આવતા કાર્યોની સૂચિ બનાવો અને python code વડે કોઈપણ એકને demonstrate કરો.

#### જવાબ

ટેબલ: NLTK માં WordNet કાર્યો

કાર્ય	વર્ણન	હેતુ
Synsets	સમાનાર્થી શબ્દો શોધવા	શબ્દ સમાનતા
Definitions	શબ્દના અર્થો મેળવવા	Context સમજવા
Examples	ઉપયોગના ઉદાહરણો	વ્યવહારિક application
Hyponyms	Specific terms શોધવા	વશવેલો સંબંધો
Hypernyms	સામાન્ય terms શોધવા	Category identification
Antonyms	વિરોધી શબ્દો શોધવા	Contrast analysis

Python કોડ - Synsets અને Definitions:

```
from nltk.corpus import wordnet

\# {      synsets      }
synsets = wordnet.synsets({good})
print("Synsets:", synsets)

\#
definition = synsets[0].definition()
print("Definition:", definition)

\#
examples = synsets[0].examples()
print("Examples:", examples)
```

### મેમરી ટ્રીક

“SDEHA” (Synsets-Definitions-Examples-Hyponyms-Antonyms)

### પ્રશ્ન 4(ક) OR [7 ગુણ]

NLP માં ambiguities ના પ્રકારો સમજવો.

#### જવાબ

**NLP Ambiguities** ત્યારે થાય છે જ્યારે ટેક્સ્ટનું અનેક રીતે અર્થધટન થઈ શકે છે, જે automated understanding માટે પડકારો બનાવે છે.

ટેબલ: Ambiguities ના પ્રકારો

પ્રકાર	વર્ણન	ઉદાહરણ	નિવારણ
<b>Lexical</b>	એક શબ્દના અનેક અર્�	"બેંક" (financial/નદીનો કિનારો)	Context analysis
<b>Syntactic</b>	અનેક વ્યાકરણ અર્થધટન	"ઉડતા વિમાનો ખતરનાક હોઈ શકે છે"	Parse trees
<b>Semantic</b>	વાક્ય સ્તરે અનેક અર્થ	"સમય તીરની જેમ ઉડે છે"	Semantic analysis
<b>Pragmatic</b>	Context-આધારિત અર્થધટન	"શું તમે મીઠું આપી શક્શો?"	Situational context
<b>Referential</b>	અસ્પષ્ટ pronoun references	"જોને બોબને કહ્યું તે ખોટો હતો"	Discourse analysis

**વિગતવાર સમજાવદ:**

- 1. Lexical Ambiguity:**
  - એક જ શબ્દ, વિવિધ અર્થો
  - Homonyms અને polysemes
  - ઉદાહરણ: "બેટ" (પ્રાણી/રમત સાધન)
- 2. Syntactic Ambiguity:**
  - અનેક વ્યાકરણ structures
  - વિવિધ parse trees શક્ય
  - ઉદાહરણ: "મૈં દૂરબીન સાથે એક માણસને જોયો"
- 3. Semantic Ambiguity:**
  - વાક્ય-સ્તરે અર્થની ગુંચવણા
  - અનેક અર્થધટન શક્ય
  - ઉદાહરણ: "સંબંધીઓની મુલાકાત કંટાળાજનક હોઈ શકે છે"
- 4. Pragmatic Ambiguity:**
  - Context અને intent આધારિત
  - સાંસ્કૃતિક અને પરિસ્થિતિગત પરિબળો
  - ઉદાહરણ: વંગ અને indirect requests
- 5. Referential Ambiguity:**
  - Entities ના અસ્પષ્ટ references
  - Pronoun resolution પડકારો
  - ઉદાહરણ: અનેક શક્ય antecedents

**નિવારણ વ્યૂહરચનાઓ:**

- Context analysis અને machine learning
- અંકડાકીય disambiguation પદ્ધતિઓ
- Knowledge bases અને ontologies

### મેમરી ટ્રીક

"LSSPR" (Lexical-Syntactic-Semantic-Pragmatic-Referential)

### પ્રશ્ન 5(અ) [3 ગુણ]

બેગ ઓફ વર્ડ્સને ઉદાહરણ સાથે સમજાવો.

#### જવાબ

બેગ ઓફ વર્ડ્સ (BoW) એ ટેક્સ્ટ પ્રતિનિધિત્વ પદ્ધતિ છે જે શબ્દની આવર્તન આધારે ટેક્સ્ટને અંકડાકીય vectors માં કન્વર્ટ કરે છે, વ્યાકરણ અને શબ્દ ક્રમને અવગાળીને.

ટેબલ: BoW પ્રક્રિયા

સ્ટેપ	પ્રક્રિયા	વર્ણન
1. Tokenization	ટેક્સ્ટને શબ્દોમાં વિભાજન	Vocabulary બનાવવી
2. Vocabulary Creation	અનન્ય શબ્દોનો સંગ્રહ	Terms નો શબ્દકોશ
3. Vector Creation	શબ્દ આવર્તન ગણવી	અંકડાકીય પ્રતિનિધિત્વ

## ઉદાહરણ:

દસ્તાવેજો:

- Doc1: "હું મશીન લર્નિંગ પરસંદ કરું છું"
- Doc2: "મશીન લર્નિંગ અફ્ટું છું"

Vocabulary: [હું, પરસંદ, મશીન, લર્નિંગ, અફ્ટું, છે, કરું, છું]

BoW Vectors:

- Doc1: [1, 1, 1, 1, 0, 0, 1, 1]
- Doc2: [0, 0, 1, 1, 1, 1, 0, 0]

લક્ષણો:

- ક્રમ સ્વતંત્ર: શબ્દ ક્રમ અવગારાવમાં રહાવે છે
- આવર્તન આધારિત: શબ્દ occurrences ગણે છે
- Sparse Representation: ઘણા શૂન્ય મૂલ્યો

## મેમરી ટ્રીક

"TVC" (Tokenize-Vocabulary-Count)

## પ્રશ્ન 5(બ) [4 ગુણ]

Word2Vec શું છે? તેના steps સમજાવો.

### જવાબ

Word2Vec એ neural network-આધારિત તકનીક છે જે મોટા text corpora માં તેમના context થી શીખીને શબ્દોના dense vector representations બનાવે છે.

ટેબલ: Word2Vec Models

મોડલ	અભિગમ	આગાહી
CBOW	Continuous Bag of Words	સંદર્ભ →
Skip-gram	Skip-gram with Negative Sampling	લક્ષ્ય શબ્દ →

Word2Vec ના સ્ટેપ્સ:

1. મોટા તૈયારી:

- મોટો text corpus એકત્ર કરવો
- ટેક્સ્ટ સાફ કરવું અને preprocess કરવું
- Training pairs બનાવવા

2. મોડલ આર્કિટેક્ચર:

- Input layer (one-hot encoded words)
- Hidden layer (embedding layer)
- Output layer (prediction માટે softmax)

3. Training પ્રક્રિયા:

- CBOW: context થી target word ની આગાહ
- Skip-gram: target word થી context ની આગાહ
- Weights update કરવા backpropagation વાપરવું

4. Vector Extraction:

- Hidden layer થી weight matrix extract કરવું
- દરેક row word embedding દર્શાવે છે
- સામાન્યતા: 100-300 dimensions

લાભો:

- Semantic relationships કેપ્ચર કરે છે
- સમાન શબ્દોના સમાન vectors હોય છે
- Arithmetic operations સપોર્ટ કરે છે (રાજા - પુરુષ + સ્ત્રી = રાણી)

## મેમરી ટ્રીક

"DMAT" (Data-Model-Architecture-Training)

## પ્રશ્ન 5(ક) [7 ગુણ]

NLP ની applications ની યાદી બનાવો અને કોઈપણ એકને વિગતવાર સમજાવો.

### જવાબ

#### ટેબલ: NLP Applications

Application	વર્ણન	ઉદ્યોગ ઉપયોગ
મશીન અનુવાદ Sentiment Analysis	ભાષા રૂપાંતરણ મત ખનન	વૈશ્વિક કોમ્યુનિકેશન Social media monitoring
Chatbots	વાતચીત AI	Customer service
ટેક્સ્ટ સારાંશ	સામગ્રી સંકુચન	સમાચાર, સંશોધન
અપીય રેક્ગ્રિશન	અવાજ થી ટેક્સ્ટ	Virtual assistants
માહિતી નિર્જારીણ	ટેક્સ્ટમાંથી ડેટા માઇનિંગ	Business intelligence
પ્રશ્ન જવાબ	સ્વચાલિત પ્રતિસાદો	Search engines
સ્પામ શોધ	Email filtering	Cybersecurity

#### વિગતવાર સમજાવો: Sentiment Analysis

Sentiment Analysis એ ટેક્સ્ટ ડેટામાં વ્યક્ત કરાયેલ ભાવનાત્મક tone અને મંતવ્યો નક્કી કરવાની પ્રક્રિયા છે.

#### ઘટકો:

- ટેક્સ્ટ પ્રીપોરેસિંગ: સફાઈ અને tokenization
- Feature Extraction: TF-IDF, word embeddings
- વર્ગીકરણ: સકારાત્મક, નકારાત્મક, તટસ્થ
- વિશ્વાસ સ્કોરિંગ: Sentiment ની મજબૂતાઈ

#### પ્રક્રિયા સ્ટેપ્સ:

- ડેટા સંગ્રહ: Reviews, social media થી ટેક્સ્ટ એકત્ર કરવું
- પ્રીપોરેસિંગ: Noise દૂર કરવો, ટેક્સ્ટ normalize કરવું
- Feature Engineering: ટેક્સ્ટને આંકડાકીય features માં કન્વર્ટ કરવું
- મોડલ ટ્રેનિંગ: વર્ગીકરણ માટે ML algorithms વાપરવા
- આગાહી: નવા ટેક્સ્ટના sentiment ની વર્ગીકરણ
- મૂલ્યાંકન: ચોકસાઈ અને કામગીરી માપવી

#### Applications:

- બ્રાન્ડ મોનિટરિંગ: ગ્રાહક મંતવ્યોને ટ્રેક કરવા
- પ્રોડક્ટ રિવ્યુ: ગ્રાહક પ્રતિસાદનું વિશ્લેષણ
- સોશિયલ મીડિયા: જાહેર sentiment ને monitor કરવું
- બજાર સંશોધન: ગ્રાહક પસંદગીઓ સમજવી

### મેમરી ટ્રીક

“MSCTSIQ-S” (Machine-Sentiment-Chatbot-Text-Speech-Information-Question-Spam)

## પ્રશ્ન 5(અ) OR [3 ગુણ]

ઉદાહરણ સાથે TFIDF સમજાવો.

### જવાબ

TF-IDF (Term Frequency-Inverse Document Frequency) દસ્તાવેજોના સંગ્રહ સાપેક્ષે દસ્તાવેજમાં શબ્દની મહત્વતા માપે છે.

સૂત્ર:  $TF-IDF = TF(t,d) \times IDF(t)$

જ્યાં:

- $TF(t,d) = (\text{દસ્તાવેજ } d \text{ માં term } t \text{ કેટલી વખત આવે છે}) / (\text{દસ્તાવેજ } d \text{ માં કુલ terms})$
- $IDF(t) = \log(\text{કુલ દસ્તાવેજો} / \text{term } t \text{ ધરાવતા દસ્તાવેજો})$

#### ઉદાહરણ:

દસ્તાવેજો:

- Doc1: “મશીન લર્નિંગ સારું છે”
- Doc2: “લર્નિંગ algorithms સારા છે”
- Doc3: “મશીન algorithms સારું કામ કરે છે”

ટેબલ: “મશીન” માટે TF-IDF ગણતરી

દસ્તાવેજ	TF	IDF	TF-IDF
Doc1	$1/4 = 0.25$	$\log(3/2) = 0.18$	$0.25 \times 0.18 = 0.045$
Doc2	$0/4 = 0$	$\log(3/2) = 0.18$	$0 \times 0.18 = 0$
Doc3	$1/4 = 0.25$	$\log(3/2) = 0.18$	$0.25 \times 0.18 = 0.045$

### મુખ્ય પોઇન્ટ્સ:

- ઉચ્ચ TF-IDF: વિશિષ્ટ દસ્તાવેજમાં મહત્વપૂર્ણ શબ્દ
- નીચો TF-IDF: દસ્તાવેજનો પાર સામાન્ય શબ્દ
- Applications: માહિતી પુનર્પ્રાપ્તિ, text mining

### મેમરી ટ્રીક

"TF-IDF" (Term frequency, Inverse Document frequency)

## પ્રશ્ન 5(બ) OR [4 ગુણ]

TFIDF અને BOW સાથેના challenges વિશે સમજાવો.

### જવાબ

#### ટેબલ: TF-IDF અને BOW સાથેના પડકારો

પડકાર	TF-IDF	BOW	અસર
Semantic Understanding	અર્થ કેંચર કરી શકતું નથી	શબ્દ સંબંધો અવગારો છે	Context ની ભરાબ સમજ
શબ્દ ક્રમ	સ્થિતિ અવગારો છે	ક્રમ ખોવાઈ જાય છે	વ્યક્તરાશ અર્થ ખોવાઈ જાય છે
Sparsity Vocabulary માપ	ઉચ્ચ-dimentional vectors મોટો feature space	ઘણા શૂન્ય મૂલ્યો Corpus સાથે વધે છે	મેમરી અકાર્યક્ષમ Computational જટિલતા
Out-of-Vocabulary	અજાણ્યા શબ્દો અવગારો છે	નવા શબ્દો handle કરતું નથી	મર્યાદિત સામાન્યીકરણ
Polysemy	અનેક અર્થો	વિવિધ senses માટે સમાન વર્તન	અસ્પષ્ટતા સમસ્યાઓ

#### વિગતવાર પડકારો:

##### 1. Semantic Understanding નો અભાવ:

- શબ્દો સ્વતંત્ર features તરીકે ગણાય છે
- Synonyms અથવા સંબંધિત concepts સમજી શકતા નથી
- "સારં" અને "ઉત્તમ" અલગ રીતે ગણાય છે

##### 2. શબ્દ ક્રમ ખોવાઈ જવો:

- "કૃતરો માણસને કરડે છે" vs "માણસ કૃતરાને કરડે છે" સમાન representation
- Context અને વ્યક્તરાશની માહિતી ખોવાઈ જાય છે
- વાક્ય structure અવગણવામાં આવે છે

##### 3. ઉચ્ચ Dimensionality:

- Vector size vocabulary size ની બરાબર
- મોટાભાગે શૂન્યો સાથે sparse matrices
- Storage અને computation સમસ્યાઓ

##### 4. Context Insensitivity:

- એક જ શબ્દ વિવિધ contexts માં સમાન ગણાય છે
- "Apple" કંપની vs ફળ સમાન representation
- Polysemy અને homonymy સમસ્યાઓ

#### ઉક્ળેદ:

- Word Embeddings: Word2Vec, GloVe
- Contextual Models: BERT, GPT
- N-grams: કેટલાક શબ્દ ક્રમ કેંચર કરવા
- Dimensionality Reduction: PCA, SVD

## પ્રશ્ન 5(ક) OR [7 ગુણ]

GloVe ની કામગીરી સમજાવો.

### જવાબ

**GloVe (Global Vectors for Word Representation)** word embeddings બનાવવા માટે global અંકડાકીય માહિતીને local context windows સાથે જોડે છે.

ટેબલ: GloVe vs અન્ય પદ્ધતિઓ

પાસું	GloVe	Word2Vec	પરંપરાગત પદ્ધતિઓ
અભિગમ	Global + Local statistics	Local context windows	આવર્તન-આધારિત
Training	Matrix factorization	Neural networks	ગાણતરી પદ્ધતિઓ
કાર્યક્ષમતા	જડપી training	ધીમી training	ખૂબ જડપી
પ્રદર્શન	ઉચ્ચ ચોકસાઈ	સારી ચોકસાઈ	મયર્ફિંડિટ પ્રદર્શન

### કામગીરી પ્રક્રિયા:

#### 1. Co-occurrence Matrix Construction:

- Context windows માં શબ્દ co-occurrences ગણવા
- Global statistics matrix બનાવવો
- $X_{ij}$  = word j, word i ના context માં કેટલી વખત આવે છે

#### 2. Ratio ગણતરી:

- સંભાવના ratios ની ગણતરી
- $P(k|i) = X_{ik} / X_i$  (word i આપવામાં આવે તો word k ની સંભાવના)
- સંભાવનાઓ વચ્ચેના અર્થપૂર્ણ ratios પર ધ્યાન

#### 3. Objective Function:

- Weighted least squares objective minimize કરવું
- $J = \frac{1}{2} \sum f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2$
- જ્યાં f(x) weighting function છે

#### 4. Vector Learning:

- Objective optimize કરવા gradient descent વાપરવું
- Word vectors  $w_i$  અને context vectors  $w_j$  શીખવા
- અંતિમ representation બંને vectors combine કરે છે

### મુખ્ય લક્ષણો:

#### Global Statistics:

- સમગ્ર corpus માહિતી વાપરે છે
- Global word relationships કેચર કરે છે
- Local પદ્ધતિઓ કરતાં વધુ સ્થિર

#### કાર્યક્રમતા:

- Co-occurrence statistics પર train કરે છે
- Neural network પદ્ધતિઓ કરતાં જડપી
- મોટા corpora માટે scalable

#### પ્રદર્શન:

- Analogy tasks પર સારું પ્રદર્શન
- Semantic અને syntactic બંને સંબંધો કેચર કરે છે
- Similarity tasks પર સારી કામગીરી

### ગાળિતિક આધાર:

$$J = \sum_{(i,j=1 \text{ to } V)} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2$$

#### જ્યાં:

- $V$  = vocabulary size
- $X_{ij}$  = co-occurrence count
- $w_i, w_j$  = word vectors
- $b_i, b_j$  = bias terms
- $f(x)$  = weighting function

#### ફાયદા:

- લાભો સંયોજન: Global statistics + local context
- સમજી શક્ય તેવું: સ્પષ્ટ ગાળિતિક આધાર
- કાર્યક્રમ: Word2Vec કરતાં જડપી training
- અસરકારક: વિવિધ tasks પર સારી કામગીરી

#### Applications:

- શબ્દ સમાનતા: સંબંધિત શબ્દો શોધવા
- **Analogy Tasks:** રાજા - પુરુષ + સ્ત્રી = રાણી
- **ટેક્સ્ટ વર્ગીક્રણ:** Feature representation
- **મશીન અનુવાદ:** Cross-lingual mappings

### મેમ્ટ્રી ટ્રીક

"CROF-PGAE" (Co-occurrence-Ratio-Objective-Function, Performance-Global-Advantage-Efficiency)