CS601 Machine Learning (RGPV) UNIT -04 :– Complete Notes WITH Easy Explaination

TOPIC 1: RECURRENT NEURAL NETWORK (RNN)

---

1. WHY THIS TOPIC EXISTS?

Real Life Problem

Imagine you are reading a sentence:

> "I went to the bank to deposit money."

When you read the word **money**, your brain remembers the previous words **bank** and **deposit** and understands that bank means a financial institution.

Now imagine:

```
Input 1 → I
Input 2 → went
Input 3 → bank
Input 4 → deposit
Input 5 → money
```

If a machine processes each word independently, it may not understand the meaning.

Problem:

Traditional Neural Networks cannot remember previous information.

Solution:

✅ Recurrent Neural Network (RNN)

RNN was developed to handle **sequence data** where previous information matters.

Why Engineers Need RNN?

RNN is used in:

- Google Translate
- Speech Recognition
- ChatGPT (basic concept)
- Stock Market Prediction
- Weather Forecasting
- Voice Assistants

---

2. BEGINNER EXPLANATION

Imagine a student solving a story problem.

A normal neural network behaves like:

```
Read one sentence
Forget it
Read next sentence
Forget it
```

RNN behaves like:

```
Read sentence
Remember it
Read next sentence
Use previous memory
```

This memory capability makes RNN special.

---

3. CORE EXAM DEFINITIONS

2 Marks Definition

A Recurrent Neural Network (RNN) is a neural network designed to process sequential data by maintaining information from previous inputs.

---

5 Marks Definition

RNN is a type of artificial neural network in which outputs from previous steps are fed back into the network, allowing it to remember past information and process sequential data effectively.

---

7 Marks Definition (RGPV Style)

Recurrent Neural Network (RNN) is a deep learning architecture specially designed for sequence processing. It contains recurrent connections that allow information from previous time steps to influence current outputs. RNN is widely used in language modeling, speech recognition, machine translation, and time-series forecasting.

---

4. DETAILED EXPLANATION

What is Sequential Data?

Sequential Data means:

Data where order matters.

Examples:

Example 1

Sentence:

```
I love Machine Learning
```

Order matters.

Example 2

Stock Prices

```
Day 1
Day 2
Day 3
Day 4
```

Previous prices influence future prices.

Example 3

Weather Forecast

Today's weather affects tomorrow's weather.

Architecture of RNN

Basic Structure

```
Input
  ↓
Hidden Layer
  ↓
Output
```

But RNN adds memory.

RNN Architecture

```
x1 → h1 → y1
      ↓
x2 → h2 → y2
      ↓
x3 → h3 → y3
      ↓
x4 → h4 → y4
```

Where:

x = Input

h = Hidden State (Memory)

y = Output

Unrolled RNN Diagram

```
Time 1

Input x1
   ↓
 Hidden h1
   ↓
Output y1

      ↓ Memory

Time 2

Input x2
   ↓
 Hidden h2
   ↓
Output y2
```

```
      ↓ Memory

Time 3

Input x3
   ↓
 Hidden h3
   ↓
Output y3
```

Hidden State

The hidden state acts as memory.

```
Current Input

+

Previous Memory

=

Current Output
```

Working of RNN

Step 1

Receive first input.

```
x1
```

Step 2

Generate hidden state.

```
h1
```

Step 3

Produce output.

```
y1
```

Step 4

Pass hidden state to next step.

```
h1 → h2
```

Step 5

Repeat process.

Mathematical Representation

```
ht = f(ht−1 , xt)
```

Where:

Symbol	Meaning
ht	Current Hidden State
ht−1	Previous Hidden State
xt	Current Input
f	Activation Function

Memory Trick

Remember:

```
RNN

R = Remember

N = Neural

N = Network
```

RNN = Network that remembers.

Types of RNN

One-to-One

```
Image
  ↓
Output
```

Example:

Image Classification

One-to-Many

```
Image
  ↓
Caption
```

Example:

Image Caption Generation

Many-to-One

```
Words
  ↓
Sentiment
```

Example:

Sentiment Analysis

Many-to-Many

```
English
  ↓
French
```

Example:

Machine Translation

Real Life Examples

Example 1: Google Translate

```
English
  ↓
RNN
  ↓
Hindi
```

Example 2: Speech Recognition

```
Voice
  ↓
RNN
  ↓
Text
```

Example 3: YouTube Captions

```
Speech
  ↓
RNN
  ↓
Subtitles
```

Advantages of RNN

1. Handles Sequential Data

Order is preserved.

---

2. Memory Capability

Stores previous information.

---

3. Flexible Inputs

Variable length inputs supported.

---

4. Effective for NLP

Useful in language tasks.

---

5. Time-Series Prediction

Predicts future values.

Disadvantages of RNN

1. Vanishing Gradient Problem

Learning becomes difficult.

---

2. Slow Training

Sequential processing required.

---

3. Short-Term Memory

Cannot remember very long sequences.

---

4. Complex Optimization

Hard to train.

Vanishing Gradient Problem

Why It Occurs?

During training:

Backpropagation sends gradients backward.

These gradients become smaller and smaller.

```
Output

↑
```

```
0.5
 ↑
0.1
 ↑
0.01
 ↑
0.0001
```

Eventually:

Learning stops.

Problems Caused

- Long sentences forgotten.
- Poor predictions.
- Low accuracy.

Solution

Two advanced architectures:

LSTM

(Long Short-Term Memory)

GRU

(Gated Recurrent Unit)

These are discussed in Chapter 2.

Applications of RNN

Natural Language Processing

Chatbots

Translation

Text Generation

Speech Recognition

Google Assistant

Alexa

Siri

Finance

Stock Prediction

Healthcare

Patient Monitoring

Disease Prediction

Weather Forecasting

Climate Analysis

PYQ INTELLIGENCE

Frequency

★★★★★ Very High

Years Asked

2020

2023

2024

2025

Last Asked

2025

Question Pattern

- Explain RNN
- Draw Architecture
- Applications
- Advantages
- Compare with LSTM

EXAMINER KEYWORDS

Write these words:

✅ Sequential Data

✅ Hidden State

✅ Memory

✅ Time Dependency

✅ Recurrent Connection

✅ NLP

✅ Speech Recognition

These keywords increase marks.

IMPORTANT QUESTIONS

7 Mark Questions

1. Explain RNN Architecture.
2. Explain Working of RNN.
3. Write Applications of RNN.
4. Explain Hidden State.

---

14 Mark Questions

1. Explain RNN with Architecture, Working, Advantages and Applications.
2. Explain Vanishing Gradient Problem in RNN.
3. Differentiate RNN and Traditional Neural Networks.

MOST EXPECTED QUESTIONS

95% Probability

Explain RNN Architecture.

90% Probability

Explain Working of RNN.

90% Probability

Explain Vanishing Gradient Problem.

85% Probability

Applications of RNN.

COMMON MISTAKES

❌ Writing RNN = ANN

❌ Forgetting hidden state

❌ Not drawing architecture

❌ Ignoring memory concept

❌ Missing applications

RGPV ANSWER WRITING FORMAT

Introduction

RNN is designed for sequential data.

Definition

Write standard definition.

Diagram

Draw architecture.

Working

Step-by-step explanation.

Advantages

Minimum 5 points.

Disadvantages

Minimum 4 points.

Applications

Minimum 5 applications.

Conclusion

RNN is widely used for sequence processing but suffers from vanishing gradient problem.

MARKS OPTIMIZATION

2 Marks

Definition only.

---

5 Marks

Definition + Architecture + Working.

---

7 Marks

Definition + Diagram + Working + Applications.

---

14 Marks

Introduction + Definition + Diagram + Working + Advantages + Disadvantages + Applications + Conclusion.

Minimum:

4–5 pages answer.

TOPPER NOTES

✅ RNN processes sequential data.

✅ Hidden State acts as memory.

✅ Used in NLP and Speech Recognition.

✅ Main limitation: Vanishing Gradient Problem.

✅ LSTM and GRU were developed to solve this problem.

✅ Most repeated Unit 4 topic.

MEMORY RETENTION TECHNIQUES

Story Method

Imagine reading a novel.

Each page depends on previous pages.

RNN remembers previous pages.

Analogy Method

Human Brain = RNN

Memory = Hidden State

Visual Association

```
Input

↓

Memory

↓

Output
```

Remember this flow.

ONE-MINUTE REVISION BOX

```
RNN

↓

Sequential Data

↓

Hidden State

↓

Memory

↓

Applications:
NLP
Speech
Translation

↓

Problem:
Vanishing Gradient

↓



Solution:
LSTM
GRU
```

IF EXAM IS TOMORROW

Learn:

1. Definition
2. Architecture Diagram
3. Hidden State
4. Working
5. Applications
6. Vanishing Gradient Problem

These alone can fetch most of the marks from RNN questions.

10-CGPA MODE

Must Learn

- Definition
- Architecture
- Working
- Hidden State
- Applications

Should Learn

- Mathematical Formula
- Types of RNN

Optional

- Advanced Variants

FINAL EXAM SUMMARY

Read This 10 Minutes Before Exam

✅ RNN handles sequence data.

✅ Hidden State stores memory.

✅ Previous information influences future output.

✅ Applications:
NLP, Translation, Speech Recognition.

✅ Main drawback:
Vanishing Gradient Problem.

✅ Solution:
LSTM and GRU.

✅ Most expected question:
"Explain RNN with Architecture and Applications."

Next Topic– LSTM & GRU Complete Premium Notes (most important topic of Unit 4 and highly repeated in RGPV exams). 🚀📘

TOPIC 2: LONG SHORT-TERM MEMORY (LSTM) & GATED RECURRENT UNIT (GRU)

CS601 Machine Learning (RGPV) – Complete Premium Notes

WHY THIS TOPIC EXISTS?

Real-Life Problem

Imagine your teacher asks:

> "What was the first word of a paragraph you read 5 minutes ago?"

Most students may forget.

Similarly, RNN also forgets old information.

Example:

```
Input 1 → Ram
Input 2 → went
Input 3 → to
Input 4 → school
Input 5 → yesterday
```

By the time the network reaches "yesterday", it may forget "Ram".

This problem is called:

Vanishing Gradient Problem

To solve this problem:

Scientists developed:

✅ LSTM

✅ GRU

WHY ENGINEERS NEED LSTM?

Used in:

- Google Translate
- Siri
- Alexa
- ChatGPT Foundations
- Speech Recognition
- Time Series Forecasting
- Self Driving Cars

LONG SHORT-TERM MEMORY (LSTM)

BEGINNER EXPLANATION

Imagine your brain has:

Short-Term Memory

Remembers recent information.

Long-Term Memory

Remembers important information for a long time.

LSTM works exactly like this.

It decides:

```
What to Remember?

What to Forget?

What to Output?
```

CORE EXAM DEFINITIONS

2 Marks Definition

LSTM is a special type of Recurrent Neural Network that can learn long-term dependencies using memory cells and gates.

---

5 Marks Definition

LSTM is an advanced RNN architecture that solves the vanishing gradient problem by using memory cells, forget gates, input gates and output gates.

---

7 Marks Definition

Long Short-Term Memory (LSTM) is a recurrent neural network architecture designed to capture long-range dependencies in sequential data. It uses memory cells and gate mechanisms to selectively store, update and retrieve information, thereby overcoming the limitations of traditional RNNs.

LSTM ARCHITECTURE

```
        +---------------+
        | Memory Cell   |
        +---------------+
                ↓

        +---------------+
        | Output Gate   |
        +---------------+
                ↓

             Output
```

COMPONENTS OF LSTM

---

1. Forget Gate

Purpose

Decides:

```
What information should be forgotten?
```

Example:

Old irrelevant information removed.

Diagram

```
Previous Memory

↓

Forget Gate

↓
```

```
Useful Information Remains
```

---

2. Input Gate

Purpose

Decides:

```
What new information should be stored?
```

Example

When reading a sentence:

```
The capital of India is New Delhi
```

Input Gate stores:

```
India
New Delhi
```

---

3. Memory Cell

Purpose

Stores important information.

Acts as long-term memory.

Diagram

```
Important Information

↓
```

```
Memory Cell

↓

Future Use
```

---

4. Output Gate

Purpose

Decides:

```
What information should be sent to output?
```

WORKING OF LSTM

Step 1

Receive Input

```
xt
```

Step 2

Forget Gate removes irrelevant information.

Step 3

Input Gate stores useful information.

Step 4

Memory Cell updates itself.

Step 5

Output Gate generates output.

Step 6

Pass memory to next step.

MEMORY TRICK

Remember:

FIMO

```
F → Forget

I → Input

M → Memory

O → Output
```

Most students remember LSTM using:

FIMO Rule

VISUAL LEARNING

```
Input
```

```
↓

Forget

↓

Input

↓

Memory

↓

Output
```

WHY LSTM IS BETTER THAN RNN?

Feature	**RNN**	**LSTM**
Memory	Short	Long
Gradient Problem	High	Low
Accuracy	Lower	Higher
Long Sentences	Poor	Excellent
Stability	Moderate	High

ADVANTAGES OF LSTM

1.

Stores long-term information.

2.

Solves vanishing gradient problem.

3.

High accuracy.

4.

Excellent for NLP tasks.

5.

Handles long sequences.

DISADVANTAGES OF LSTM

1.

Complex architecture.

2.

Slow training.

3.

High computational cost.

4.

More memory consumption.

APPLICATIONS OF LSTM

NLP

Language Modeling

Translation

Google Translate

Speech Recognition

Siri

Alexa

Stock Prediction

Market Analysis

Healthcare

Patient Monitoring

PYQ INTELLIGENCE

Frequency:

★★★★★ Very High

Years Asked:

2020

2023

2024

2025

Expected:

2026

EXAMINER KEYWORDS

Write:

- Memory Cell
- Forget Gate
- Input Gate
- Output Gate
- Long-Term Dependency
- Sequential Data

These keywords fetch extra marks.

GATED RECURRENT UNIT (GRU)

WHY GRU EXISTS?

LSTM works well.

But:

It is computationally expensive.

Researchers created:

GRU

Simpler

Faster

Efficient

DEFINITION

GRU is a simplified version of LSTM that uses fewer gates while maintaining good performance.

ARCHITECTURE

```
Input

↓

Update Gate

↓

Reset Gate

↓

Output
```

COMPONENTS OF GRU

Update Gate

Decides:

```
What should be remembered?
```

Reset Gate

Decides:

```
What should be forgotten?
```

WORKING OF GRU

Step 1

Receive Input.

Step 2

Update Gate selects important information.

Step 3

Reset Gate removes unnecessary information.

Step 4

Generate Output.

Step 5

Pass information forward.

MEMORY TRICK

Remember:

RU

```
R → Reset

U → Update
```

Only 2 gates.

Easy to remember.

ADVANTAGES OF GRU

1.

Faster training.

2.

Less memory usage.

3.

Simpler architecture.

4.

Good accuracy.

5.

Suitable for large datasets.

DISADVANTAGES OF GRU

1.

Less expressive than LSTM.

2.

May perform poorly on very long sequences.

APPLICATIONS OF GRU

- Chatbots
- NLP
- Time-Series Prediction
- Recommendation Systems
- Speech Processing

LSTM VS GRU

Feature	LSTM	GRU
Gates	3	2
Memory Cell	Yes	No
Complexity	High	Low

Speed	Slow	Fast
Accuracy	Slightly Higher	Slightly Lower
Memory Usage	High	Low

RNN VS LSTM VS GRU

Feature	RNN	LSTM	GRU
Memory	Low	High	High
Gradient Problem	High	Very Low	Low
Speed	Fast	Slow	Faster
Accuracy	Low	High	High

REAL LIFE EXAMPLES

Example 1

Google Translate

Uses LSTM.

Example 2

Voice Assistant

Uses LSTM/GRU.

Example 3

Stock Prediction

Uses GRU.

Example 4

Chatbots

Use LSTM and GRU.

IMPORTANT QUESTIONS

7 Mark Questions

1. Explain LSTM Architecture.
2. Explain Forget Gate.
3. Explain GRU Architecture.
4. Compare LSTM and GRU.

---

14 Mark Questions

1. Explain LSTM with Architecture and Working.
2. Explain Vanishing Gradient Problem and how LSTM solves it.
3. Differentiate RNN, LSTM and GRU.
4. Explain GRU with diagram.

MOST EXPECTED QUESTIONS

95% Probability

Explain LSTM Architecture.

95% Probability

Differentiate RNN and LSTM.

90% Probability

Explain GRU.

90% Probability

Compare LSTM and GRU.

85% Probability

Explain Forget Gate.

COMMON MISTAKES

❌ Confusing gates.

❌ Forgetting Memory Cell.

❌ Not drawing architecture.

❌ Writing GRU = LSTM.

❌ Missing applications.

RGPV ANSWER WRITING FORMAT

For 14 Marks:

Introduction

Need for LSTM.

Definition

Standard definition.

Architecture Diagram

Must draw.

Working

Step-by-step.

Advantages

5 points.

Disadvantages

4 points.

Applications

5 applications.

Conclusion

LSTM solves long-term dependency problems.

TOPPER NOTES

✅ LSTM solves vanishing gradient problem.

✅ Uses Forget, Input and Output Gates.

✅ Memory Cell stores long-term information.

✅ GRU is simplified LSTM.

✅ LSTM = Higher Accuracy.

✅ GRU = Faster Training.

✅ Very frequently asked in RGPV.

ONE-MINUTE REVISION BOX

```
RNN

↓

Problem

↓

Vanishing Gradient

↓

Solution

↓

LSTM

↓

Forget Gate
Input Gate
Output Gate

↓

GRU

↓
```

```
Reset Gate
Update Gate
```

IF EXAM IS TOMORROW

Learn:

1. LSTM Diagram
2. Forget Gate
3. Input Gate
4. Output Gate
5. GRU Diagram
6. LSTM vs GRU Table

These alone can secure most marks from Chapter 2.

FINAL EXAM SUMMARY

Read This Before Exam

✅ LSTM solves RNN limitations.

✅ FIMO Rule:

Forget

Input

Memory

Output

✅ GRU uses:

Reset Gate

Update Gate

✅ Most Expected Question:

"Explain LSTM Architecture with neat diagram."

🚀 **NextTopic :– Machine Translation, Beam Search, Beam Width, BLEU Score and Attention Model (Complete Premium Notes)**.

TOPIC 3 : MACHINE TRANSLATION, BEAM SEARCH, BEAM WIDTH, BLEU SCORE & ATTENTION MODEL

CS601 Machine Learning (RGPV) – Complete Premium Notes

WHY THIS TOPIC EXISTS?

Real Life Problem

Suppose you type:

```
How are you?
```

Google Translate converts it into:

```
आप कैसे हैं?
```

Question:

How does a machine understand one language and generate another language?

This problem led to the development of:

✅ Machine Translation

✅ Beam Search

✅ Attention Mechanism

✅ BLEU Score

These technologies are used in:

- Google Translate
- ChatGPT
- DeepL Translator
- Alexa
- Siri

MACHINE TRANSLATION

Beginner Explanation

Machine Translation means:

Converting text from one language to another using AI.

Example:

```
English

↓

Machine Learning Model

↓
```

```
Hindi
```

Input:

```
I love India
```

Output:

```
मुझे भारत पसंद है
```

Definition

2 Marks

Machine Translation is the automatic conversion of text from one language to another using computer algorithms.

5 Marks

Machine Translation is an NLP technique that automatically translates text from a source language into a target language.

7 Marks

Machine Translation is a Natural Language Processing application that uses deep learning models such as RNNs, LSTMs and Transformers to convert text from one language to another while preserving meaning and context.

ARCHITECTURE OF MACHINE TRANSLATION

Encoder-Decoder Model

```
English Sentence

↓

Encoder

↓

Context Vector

↓

Decoder

↓

Hindi Sentence
```

Encoder

Purpose:

Understand input sentence.

Example:

```
I Love India
```

Encoder converts it into meaningful numerical representation.

Decoder

Purpose:

Generate translated sentence.

Example:

```
मुझे भारत पसंद है
```

Working

Step 1

Input sentence received.

↓

Step 2

Encoder processes sentence.

↓

Step 3

Context vector created.

↓

Step 4

Decoder generates output.

↓

Step 5

Translation completed.

Advantages

1. Fast translation.
2. Supports multiple languages.
3. Scalable.
4. Used globally.
5. Improves communication.

Disadvantages

1. May lose context.
2. Ambiguous words create errors.
3. Requires large datasets.

BEAM SEARCH

Why Beam Search Exists?

During translation:

Many possible outputs exist.

Example:

```
I love India
```

Possible translations:

```
Option 1
मुझे भारत पसंद है

Option 2
मैं भारत से प्रेम करता हूँ
```

```
Option 3
मुझे इंडिया अच्छा लगता है
```

Question:

Which translation should AI choose?

Answer:

Beam Search.

Definition

Beam Search is a heuristic search algorithm that keeps only the most promising candidate sequences during decoding.

Beginner Explanation

Imagine:

You are solving MCQs.

Instead of checking every option,

you focus only on the best options.

This is Beam Search.

Working

Step 1

Generate possible words.

↓

Step 2

Assign probability.

↓

Step 3

Keep top candidates.

↓

Step 4

Expand those candidates.

↓

Step 5

Select best sentence.

Diagram

```
Start

↓

Option A (0.8)

Option B (0.7)

Option C (0.3)

↓

Keep A and B
```

```
↓

Expand

↓

Best Translation
```

Beam Width

Definition

Beam Width is the number of candidate sequences retained at each step during Beam Search.

Example

Beam Width = 1

```
Only Best Option
```

This becomes Greedy Search.

Beam Width = 2

```
Top 2 Options Retained
```

Beam Width = 3

```
Top 3 Options Retained
```

Memory Trick

Remember:

```
Beam Width

=

Number of Paths Kept
```

Advantages of Beam Search

1. Better translation quality.
2. More accurate than greedy search.
3. Efficient.

Disadvantages

1. Computational cost.
2. Not always globally optimal.

BLEU SCORE

Why BLEU Exists?

Question:

How do we evaluate translation quality?

Need:

A numerical score.

Solution:

BLEU Score.

Definition

BLEU (Bilingual Evaluation Understudy) Score is a metric used to evaluate machine translation quality.

Beginner Explanation

Suppose:

Reference Translation:

```
I love India
```

Machine Translation:

```
I like India
```

BLEU checks similarity.

Higher similarity

↓

Higher BLEU Score

BLEU Score Range

```
0 to 1
```

or

```
0% to 100%
```

Interpretation

BLEU Score	Quality
0.9+	Excellent
0.8	Very Good
0.6	Good
0.4	Average
Below 0.3	Poor

Advantages

1. Fast evaluation.
2. Automatic scoring.
3. Widely used.

Disadvantages

1. Does not fully understand meaning.
2. Human evaluation still important.

ATTENTION MODEL

Why Attention Exists?

Problem:

Long sentences.

Example:

```
The student who won the competition and received many awards visited Delhi.
```

RNN/LSTM may forget important words.

Need:

Focus only on important words.

Solution:

Attention Mechanism.

Definition

Attention is a neural network mechanism that allows the model to focus on important parts of the input sequence while generating output.

Beginner Explanation

Imagine reading a book.

Before exam:

You highlight important lines.

Attention works exactly like highlighting.

Diagram

```
Input Sentence

↓

Important Words Selected

↓

Attention Layer

↓

Output
```

Example

Input:

```
The capital of India is New Delhi
```

Attention focuses on:

```
India

New Delhi
```

Working

Step 1

Input sequence received.

↓

Step 2

Importance score calculated.

↓

Step 3

Important words selected.

↓

Step 4

Output generated.

Advantages

1. Better accuracy.
2. Handles long sequences.
3. Improves translation.
4. Reduces information loss.

Applications

- ChatGPT
- Google Translate
- Text Summarization
- Question Answering
- Speech Recognition

PYQ INTELLIGENCE

Machine Translation

★★★★☆

Asked:

2022

2024

Beam Search

★★★★☆

Asked:

2024

BLEU Score

★★★☆☆

Asked:

2024

Attention Model

★★★★★

Asked:

2022

2025

Expected:

2026

EXAMINER KEYWORDS

Write these:

Translation

- Encoder
- Decoder
- Context Vector

Beam Search

- Candidate Sequence
- Probability
- Beam Width

BLEU

- Translation Quality
- Evaluation Metric

Attention

- Focus Mechanism
- Importance Score
- Long Sequence

IMPORTANT QUESTIONS

7 Mark Questions

1. Explain Beam Search.
2. Explain Beam Width.

3. Explain BLEU Score.
4. Explain Machine Translation.

---

14 Mark Questions

1. Explain Encoder-Decoder Architecture.
2. Explain Attention Mechanism with diagram.
3. Explain Machine Translation process.
4. Explain Beam Search and BLEU Score.

MOST EXPECTED QUESTIONS

95% Probability

Explain Attention Mechanism.

90% Probability

Explain Beam Search.

90% Probability

Explain Machine Translation.

85% Probability

Explain BLEU Score.

COMPARISON TABLE

Beam Search vs Greedy Search

Feature	Greedy Search	Beam Search
Candidates	One	Multiple
Accuracy	Lower	Higher
Speed	Faster	Slower
Quality	Moderate	Better

TOPPER NOTES

✅ Machine Translation uses Encoder-Decoder.

✅ Beam Search selects best candidate sequences.

✅ Beam Width = Number of paths retained.

✅ BLEU Score evaluates translation quality.

✅ Attention focuses on important words.

✅ Attention is one of the most important Unit 4 topics.

ONE-MINUTE REVISION BOX

```
Machine Translation

↓

Encoder

↓
```

```
Decoder

↓

Beam Search

↓

Beam Width

↓

BLEU Score

↓

Attention

↓

Focus Important Words
```

IF EXAM IS TOMORROW

Learn:

1. Encoder-Decoder Diagram
2. Beam Search
3. Beam Width
4. BLEU Score
5. Attention Mechanism

These topics can easily fetch a large portion of marks from Chapter 3.

FINAL EXAM SUMMARY

Read This 10 Minutes Before Exam

✅ Translation = Language Conversion

✅ Encoder Understands Input

✅ Decoder Generates Output

✅ Beam Search = Best Path Selection

✅ Beam Width = Number of Candidates

✅ BLEU = Translation Quality Metric

✅ Attention = Focus on Important Words

✅ Most Expected Question:

"Explain Attention Mechanism with neat diagram."

🚀 **Next Topics– Reinforcement Learning (RL Framework), Agent, Environment, Reward System & Markov Decision Process (MDP) Complete Premium Notes**.

TOPIC 4: REINFORCEMENT LEARNING (RL) FRAMEWORK & MARKOV DECISION PROCESS (MDP)

CS601 Machine Learning (RGPV) – Complete Premium Notes

WHY THIS TOPIC EXISTS?

Real-Life Problem

Imagine you are teaching a dog.

When the dog performs a correct action:

```
Give Reward ✓
```

When it performs a wrong action:

```
No Reward ✗
```

After some time:

Dog learns correct behavior.

Question:

Can a machine learn the same way?

Answer:

✅ Reinforcement Learning

WHY ENGINEERS NEED RL?

Used in:

- Self Driving Cars
- Robotics
- ChatGPT Reinforcement Training
- Chess AI
- AlphaGo
- Recommendation Systems

BEGINNER EXPLANATION

Most ML algorithms learn from examples.

Example:

```
Input → Output
```

But RL learns through:

```
Trial
↓
Error
↓
Reward
↓
Learning
```

Like humans.

WHAT IS REINFORCEMENT LEARNING?

---

2 Marks Definition

Reinforcement Learning is a machine learning technique where an agent learns by interacting with an environment and receiving rewards.

---

5 Marks Definition

Reinforcement Learning is a learning paradigm in which an agent performs actions in an environment and learns an optimal policy by maximizing cumulative rewards.

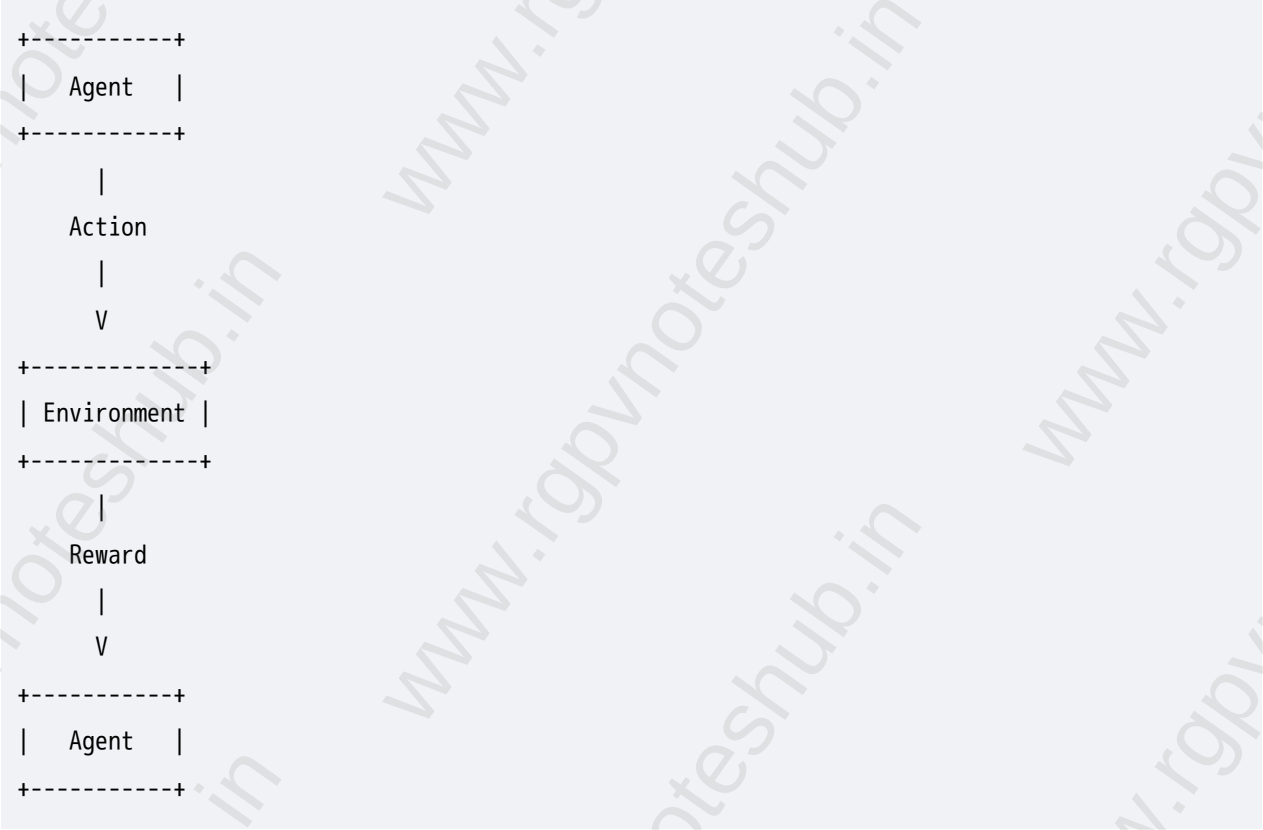
---

7 Marks Definition

Reinforcement Learning (RL) is a branch of machine learning where an intelligent agent learns optimal behavior through interaction with an environment by receiving rewards or penalties. The objective is to maximize long-term cumulative reward.

RL FRAMEWORK

Basic Architecture



RL COMPONENTS

---

1. Agent

The learner or decision maker.

Example:

```
Robot
Self Driving Car
Game Player
```

---

2. Environment

The world where agent operates.

Examples:

```
Road
Game
Factory
Robot World
```

---

3. State (S)

Current situation.

Example:

```
Robot Location
```

---

4. Action (A)

Decision taken by agent.

Example:

```
Move Left
Move Right
Move Forward
```

---

5. Reward (R)

Feedback received.

Example:

```
Correct Action → +10

Wrong Action → -10
```

---

6. Policy (π)

Strategy used by agent.

Simple meaning:

```
What action to take in a state?
```

RL WORKING

Step 1

Agent observes state.

↓

Step 2

Agent chooses action.

↓

Step 3

Environment changes.

↓

Step 4

Reward received.

↓

Step 5

Agent learns.

↓

Step 6

Process repeats.

VISUAL LEARNING

```
State

 ↓

Action

 ↓

Environment

 ↓

Reward

 ↓
```

```
Learning

↓

Better Action
```

REAL-LIFE EXAMPLE

Self Driving Car

State:

```
Traffic Signal Red
```

Action:

```
Stop
```

Reward:

```
+10
```

Wrong Action:

```
Move Forward
```

Reward:

```
-100
```

Eventually:

Car learns correct behavior.

TYPES OF RL

Positive Reinforcement

Good reward given.

```
Correct Answer

↓

Reward
```

Negative Reinforcement

Penalty removed after correct action.

ADVANTAGES OF RL

1

Learns automatically.

2

No labeled data required.

3

Handles dynamic environments.

4

Useful for decision making.

5

Improves continuously.

DISADVANTAGES OF RL

1

Needs lots of training.

2

High computational cost.

3

Slow convergence.

4

Requires reward design.

APPLICATIONS OF RL

Robotics

Robot Navigation.

Gaming

AlphaGo.

Finance

Trading Systems.

Healthcare

Treatment Optimization.

Recommendation Systems

Netflix

Amazon

YouTube

MARKOV DECISION PROCESS (MDP)

WHY MDP EXISTS?

Question:

How can we mathematically represent RL problems?

Solution:

MDP

DEFINITION

2 Marks

MDP is a mathematical model used for decision making in Reinforcement Learning.

---

5 Marks

Markov Decision Process is a framework that models states, actions, rewards and transitions for sequential decision making.

---

7 Marks

A Markov Decision Process (MDP) is a mathematical framework used in Reinforcement Learning to model environments where outcomes depend on current states, actions and rewards.

COMPONENTS OF MDP

MDP consists of:

```
S = States

A = Actions

R = Rewards

P = Transition Probability
```

MDP DIAGRAM

```
State S1

↓

Action A

↓

State S2
```

```
↓

Reward R
```

MARKOV PROPERTY

Definition

Future depends only on present state.

Not on complete history.

Example

Correct

```
Current State

↓

Future State
```

Wrong

```
Past History

↓

Future State
```

Memory Trick

Remember:

```
Markov

=

Present Matters
```

Not past.

TRANSITION PROBABILITY

Definition

Probability of moving from one state to another.

Example

```
State A

↓

0.8

↓

State B
```

Means:

80% chance.

MDP WORKING

Step 1

Observe current state.

Step 2

Select action.

Step 3

Move to next state.

Step 4

Receive reward.

Step 5

Repeat.

VISUAL LEARNING

```
State

 ↓

Action
```

```
↓

Next State

↓

Reward

↓

Learning
```

RL vs MDP

Feature	RL	MDP
Type	Learning Method	Mathematical Model
Purpose	Learn Behavior	Model Environment
Focus	Rewards	States & Actions
Usage	Training Agent	Problem Representation

PYQ INTELLIGENCE

Reinforcement Learning

Frequency:

★★★★★

Asked:

2020

2022

2023

2024

Expected:

2026

MDP

Frequency:

★★★★★

Asked:

2020

2022

2024

Expected:

2026

EXAMINER KEYWORDS

Write:

RL

- Agent
- Environment
- Reward

- Action
- State
- Policy

MDP

- Markov Property
- State
- Action
- Reward
- Transition Probability

IMPORTANT QUESTIONS

7 Mark Questions

1. Explain RL Framework.
2. Explain Agent and Environment.
3. Explain Reward System.
4. Explain MDP Components.

---

14 Mark Questions

1. Explain Reinforcement Learning Framework with diagram.
2. Explain Markov Decision Process.
3. Explain RL Components.
4. Differentiate RL and MDP.

MOST EXPECTED QUESTIONS

95% Probability

Explain Reinforcement Learning Framework.

95% Probability

Explain MDP with diagram.

90% Probability

Explain Markov Property.

90% Probability

Applications of RL.

COMMON MISTAKES

❌ Confusing RL with Supervised Learning.

❌ Forgetting reward concept.

❌ Not drawing RL architecture.

❌ Not explaining Markov Property.

TOPPER NOTES

✅ RL learns through rewards.

✅ Agent interacts with Environment.

✅ Goal = Maximize reward.

✅ MDP models RL mathematically.

✅ MDP = State + Action + Reward + Transition.

✅ Most repeated Unit 4 topic.

ONE-MINUTE REVISION BOX

```
Agent

↓

Action

↓

Environment

↓

Reward

↓

Learning

----------------

MDP

↓

State

↓

Action

↓

Reward

↓
```

```
Transition
```

IF EXAM IS TOMORROW

Learn:

1. RL Framework Diagram
2. RL Components
3. MDP Components
4. Markov Property
5. Applications

These topics alone can secure most marks from Chapter 4.

FINAL EXAM SUMMARY

Read This 10 Minutes Before Exam

✅ RL learns using rewards.

✅ Agent takes actions.

✅ Environment provides feedback.

✅ MDP models RL mathematically.

✅ Markov Property:

```
Future depends on Present
```

✅ Most Expected Question:

"Explain Reinforcement Learning Framework and Markov Decision Process with neat diagrams."

🚀 **Next Topics – Bellman Equation, Discount Factor, Value Iteration & Policy Iteration (Complete Premium Notes)**.

TOPIC 5 : BELLMAN EQUATION, DISCOUNT FACTOR, VALUE ITERATION & POLICY ITERATION

CS601 Machine Learning (RGPV) – Complete Premium Notes

WHY THIS TOPIC EXISTS?

Real-Life Problem

Imagine you have two choices:

Option 1

Get ₹100 Today

Option 2

Get ₹120 After 1 Month

Question:

Which option is better?

To make intelligent decisions we must consider:

✅ Immediate Reward

✅ Future Reward

This is exactly what Reinforcement Learning does.

To calculate future rewards mathematically, Bellman Equation was introduced.

BEGINNER EXPLANATION

Suppose you are playing a game.

Every move gives points.

```
Move 1 → +10

Move 2 → +20

Move 3 → +50
```

Question:

Which path gives maximum total reward?

Bellman Equation helps find the best path.

WHAT IS BELLMAN EQUATION?

---

2 Marks Definition

Bellman Equation is a recursive equation used to calculate the value of a state in Reinforcement Learning.

---

5 Marks Definition

Bellman Equation expresses the value of a state as the sum of immediate reward and future discounted rewards.

---

7 Marks Definition

The Bellman Equation is a fundamental equation in Reinforcement Learning that relates the value of a state to the reward received and the value of future states. It is used for solving Markov Decision Processes and finding optimal policies.

WHY BELLMAN EQUATION IS IMPORTANT?

Without Bellman Equation:

❌ Agent cannot evaluate future rewards.

With Bellman Equation:

✅ Agent chooses optimal actions.

BELLMAN EQUATION FORMULA

```
V(s) = R + γV(s')
```

Where:

Symbol	Meaning
V(s)	Current State Value
R	Immediate Reward
γ	Discount Factor

V(s')	Future State Value

MEMORY TRICK

Remember:

```
Bellman

=

Present Reward

+

Future Reward
```

VISUAL LEARNING

```
Current State

↓

Immediate Reward

+

Future Reward

↓

State Value
```

EXAMPLE

Suppose:

```
Reward = 10

Future Value = 100

Discount Factor = 0.9
```

Then:

```
V(s)

=

10 + (0.9 × 100)

=

100
```

ADVANTAGES OF BELLMAN EQUATION

1. Simplifies decision making.
2. Used in RL algorithms.
3. Finds optimal policy.
4. Evaluates future rewards.
5. Supports MDP solutions.

DISADVANTAGES

1. Computationally expensive.
2. Large state spaces are difficult.
3. Requires repeated calculations.

DISCOUNT FACTOR (γ)

WHY DISCOUNT FACTOR EXISTS?

Question:

Should future rewards be valued equally as present rewards?

Answer:

Usually No.

Future rewards are discounted.

Definition

Discount Factor determines the importance of future rewards.

Range

$$0 \leq \gamma \leq 1$$

Interpretation

γ = 0

Agent only cares about immediate reward.

γ = 1

Agent values future rewards completely.

γ = 0.9

Agent considers future rewards significantly.

Example

```
Reward Today = 100

Future Reward = 100
```

With:

```
γ = 0.5
```

Future value becomes:

```
50
```

MEMORY TRICK

```
γ

↓

Future Importance
```

Higher γ

↓

More future focused.

VALUE FUNCTION

Definition

Value Function measures how good a state is.

Formula

```
V(s)
```

Example

State A:

```
Value = 50
```

State B:

```
Value = 200
```

State B is better.

VALUE ITERATION

WHY VALUE ITERATION EXISTS?

Need:

Best state values.

Solution:

Value Iteration.

Definition

Value Iteration is an algorithm that repeatedly updates state values using the Bellman Equation until convergence.

WORKING

Step 1

Initialize values.

```
All States = 0
```

Step 2

Apply Bellman Equation.

Step 3

Update values.

Step 4

Repeat.

Step 5

Stop when values stabilize.

DIAGRAM

```
Initialize

↓

Bellman Update

↓

Update Values

↓

Repeat

↓

Optimal Values
```

ADVANTAGES

1. Finds optimal values.
2. Guarantees convergence.
3. Easy implementation.

DISADVANTAGES

1. Slow for large problems.

2. Computationally intensive.

POLICY

Definition

Policy is a strategy that tells the agent what action to take in a state.

Example

```
State A

↓

Move Right
```

POLICY ITERATION

WHY POLICY ITERATION EXISTS?

Instead of directly finding values,

we improve policy repeatedly.

Definition

Policy Iteration is an algorithm that alternates between policy evaluation and policy improvement until an optimal policy is obtained.

WORKING

Step 1

Initialize policy.

Step 2

Evaluate policy.

Step 3

Improve policy.

Step 4

Repeat.

Step 5

Get optimal policy.

DIAGRAM

```
Initial Policy

↓

Policy Evaluation

↓

Policy Improvement

↓

Repeat

↓

Optimal Policy
```

POLICY EVALUATION

Question:

How good is current policy?

Answer:

Calculate state values.

POLICY IMPROVEMENT

Question:

Can a better action be chosen?

If Yes:

Update policy.

VALUE ITERATION VS POLICY ITERATION

Feature	Value Iteration	Policy Iteration
Focus	Values	Policies
Updates	Value Function	Policy + Value
Speed	Faster per iteration	Fewer iterations
Complexity	Lower	Higher

REAL-LIFE EXAMPLE

Navigation App

State:

```
Current Location
```

Action:

```
Choose Route
```

Reward:

```
Shortest Path
```

Bellman Equation helps find optimal route.

RL CONNECTION

```
RL

↓

MDP

↓

Bellman Equation

↓

Value Function

↓

Value Iteration

↓

Policy Iteration

↓

Optimal Policy
```

PYQ INTELLIGENCE

Bellman Equation

★★★★★

Asked:

2020

2022

2024

Expected:

2026

Value Iteration

★★★★★

Asked:

2022

2024

Expected:

2026

Policy Iteration

★★★★★

Asked:

2020

2024

Expected:

2026

EXAMINER KEYWORDS

Write:

- Bellman Equation

- Discount Factor
- Value Function
- Policy
- Policy Evaluation
- Policy Improvement
- Convergence
- Optimal Policy

IMPORTANT QUESTIONS

7 Mark Questions

1. Explain Bellman Equation.
2. Explain Discount Factor.
3. Explain Value Iteration.
4. Explain Policy Iteration.

---

14 Mark Questions

1. Explain Bellman Equation with example.
2. Explain Value Iteration Algorithm.
3. Explain Policy Iteration Algorithm.
4. Differentiate Value Iteration and Policy Iteration.

MOST EXPECTED QUESTIONS

95% Probability

Explain Bellman Equation.

95% Probability

Explain Value Iteration.

90% Probability

Explain Policy Iteration.

90% Probability

Explain Discount Factor.

COMMON MISTAKES

❌ Forgetting Bellman formula.

❌ Confusing Value Iteration with Policy Iteration.

❌ Ignoring Discount Factor.

❌ Not drawing diagrams.

TOPPER NOTES

✅ Bellman Equation is the heart of RL.

✅ Current reward + Future reward.

✅ Discount Factor controls future importance.

✅ Value Iteration updates values.

✅ Policy Iteration improves policies.

✅ Most repeated RL topic in RGPV.

ONE-MINUTE REVISION BOX

```
Bellman

↓

V(s)=R+γV(s')

↓

Discount Factor

↓

Future Importance

↓

Value Iteration

↓

Optimal Values

↓

Policy Iteration

↓

Optimal Policy
```

IF EXAM IS TOMORROW

Learn:

1. Bellman Equation Formula
2. Discount Factor

3. Value Iteration Steps
4. Policy Iteration Steps
5. Comparison Table

These alone can secure most marks from Chapter 5.

FINAL EXAM SUMMARY

Read This 10 Minutes Before Exam

✅ Bellman Equation = Immediate Reward + Future Reward

✅ γ = Discount Factor

✅ Value Iteration finds optimal values.

✅ Policy Iteration finds optimal policy.

✅ Policy Evaluation + Policy Improvement = Policy Iteration.

✅ Most Expected Question:

"Explain Bellman Equation, Value Iteration and Policy Iteration with neat diagrams."

🚀 **Next TOPICS – Actor-Critic Model, Q-Learning and SARSA (Most Important & Highest PYQ Frequency Topic of Unit 4)**.

TOPIC 6: ACTOR-CRITIC MODEL, Q-LEARNING & SARSA

CS601 Machine Learning (RGPV) – Complete Premium Notes

> ⭐ **Most Important Chapter of Unit 4**
>
> PYQ Frequency: ★★★★★
>
> Asked Multiple Times in RGPV Papers

WHY THIS TOPIC EXISTS?

Real-Life Problem

Imagine you are learning to play cricket.

Initially:

```
Wrong Shot
↓
Get Out
```

Later:

```
Correct Shot
↓
Score Runs
```

Over time you learn:

- Which shot is good
- Which shot is bad

This is exactly how Reinforcement Learning works.

Question:

How does an AI decide:

```
Which action is best?
```

Answer:

Using:

✅ Q-Learning

✅ SARSA

✅ Actor-Critic

BEGINNER EXPLANATION

Imagine a rat inside a maze.

Goal:

```
Find Food
```

Initially:

Rat makes random decisions.

After many attempts:

Rat learns:

```
Left = Bad

Right = Good
```

Eventually:

Rat finds shortest path.

RL algorithms perform exactly the same learning process.

PART 1 : Q-LEARNING

WHAT IS Q-LEARNING?

---

2 Marks Definition

Q-Learning is a model-free reinforcement learning algorithm used to learn optimal actions using rewards.

---

5 Marks Definition

Q-Learning is a value-based RL algorithm that learns the quality of actions in different states and helps an agent select optimal actions.

---

7 Marks Definition

Q-Learning is a model-free reinforcement learning algorithm that learns an optimal policy by estimating Q-values for state-action pairs and maximizing cumulative future rewards.

WHY IS IT CALLED Q-LEARNING?

Q stands for:

```
Quality
```

Meaning:

```
How good is an action?
```

Q VALUE

Q-Value tells:

```
How useful an action is in a state
```

Example:

State	Action	Q Value
S1	Left	10
S1	Right	50

Agent chooses:

```
Right
```

Because:

```
50 > 10
```

Q-TABLE

Q-Learning stores knowledge inside:

Q Table

Example:

State	Left	Right
A	5	10
B	15	50
C	20	100

Best action:

Highest Q Value.

Q-LEARNING ARCHITECTURE

```
Current State

↓

Choose Action

↓

Perform Action

↓

Receive Reward

↓

Update Q Value

↓

Learn
```

WORKING OF Q-LEARNING

Step 1

Observe State

```
S
```

Step 2

Choose Action

```
A
```

Step 3

Receive Reward

```
R
```

Step 4

Move to Next State

```
S'
```

Step 5

Update Q Table

Step 6

Repeat

Q-LEARNING FORMULA

```
Q(S,A)

=

Q(S,A)

+
```

```
α [R + γ maxQ(S',A') − Q(S,A)]
```

Memory Trick

Remember:

```
Old Knowledge

+

New Experience

=

Updated Knowledge
```

ADVANTAGES OF Q-LEARNING

1

Simple algorithm.

2

No environment model required.

3

Finds optimal policy.

4

Easy implementation.

5

High performance.

DISADVANTAGES

1

Large Q tables.

2

Slow learning.

3

Poor scalability.

REAL-LIFE EXAMPLE

Robot Navigation

State:

```
Current Position
```

Action:

```
Move Right
```

Reward:

```
+100
```

Robot learns best path.

PART 2 : SARSA

WHAT IS SARSA?

Full Form

```
State

Action

Reward

State

Action
```

Definition

SARSA is an on-policy reinforcement learning algorithm that updates values using the action actually taken by the agent.

Beginner Explanation

Q-Learning says:

```
Learn from BEST future action.
```

SARSA says:

```
Learn from ACTUAL future action.
```

This is the main difference.

SARSA ARCHITECTURE

```
State

↓

Action

↓

Reward

↓

Next State

↓

Next Action

↓

Update
```

SARSA FORMULA

```
Q(S,A)

=

Q(S,A)

+

α [R + γQ(S',A') − Q(S,A)]
```

IMPORTANT DIFFERENCE

Q-Learning

Uses:

```
Best Future Action
```

SARSA

Uses:

```
Actual Future Action
```

MEMORY TRICK

SARSA =

```
State

Action

Reward

State

Action
```

Simply remember initials.

Q-LEARNING VS SARSA

Feature	Q-Learning	SARSA

Type	Off Policy	On Policy
Learning	Best Action	Actual Action
Exploration	Less Safe	More Safe
Speed	Faster	Slower
Convergence	Faster	Moderate

REAL-LIFE EXAMPLE

Imagine driving.

Q-Learning:

```
What is BEST route?
```

SARSA:

```
What route am I ACTUALLY taking?
```

PART 3 : ACTOR-CRITIC MODEL

WHY ACTOR-CRITIC EXISTS?

Problem:

Q-Learning becomes slow in large environments.

Need:

Faster learning.

Solution:

Actor-Critic.

BEGINNER EXPLANATION

Imagine:

A cricket coach.

Two people involved:

Actor

Plays the shot.

Critic

Evaluates the shot.

ACTOR-CRITIC IDEA

```
Actor

↓

Takes Action

↓

Critic

↓

Gives Feedback

↓

Actor Improves
```

DEFINITION

Actor-Critic is a reinforcement learning architecture that combines policy-based learning (Actor) and value-based learning (Critic).

COMPONENTS

Actor

Responsible for:

```
Choosing Action
```

Critic

Responsible for:

```
Evaluating Action
```

ARCHITECTURE

```
Environment

↓

State

↓

Actor
```

```
↓

Action

↓

Environment

↓

Reward

↓

Critic

↓

Feedback

↓

Actor Update
```

WORKING

Step 1

Actor selects action.

Step 2

Environment returns reward.

Step 3

Critic evaluates reward.

Step 4

Critic provides feedback.

Step 5

Actor improves policy.

Step 6

Repeat.

ADVANTAGES

1

Fast learning.

2

Handles large environments.

3

Stable training.

4

Efficient policy learning.

5

Used in modern RL systems.

DISADVANTAGES

1

Complex implementation.

2

More computational cost.

APPLICATIONS

Robotics

Robot control.

Self Driving Cars

Navigation.

Gaming

AlphaGo.

Finance

Portfolio optimization.

Recommendation Systems

Netflix.

Amazon.

PYQ INTELLIGENCE

Q-Learning

★★★★★

Asked:

2020

2022

2025

Expected:

2026

SARSA

★★★★★

Asked:

2020

2022

2025

Expected:

2026

Actor-Critic

★★★★☆

Asked:

2024

Expected:

2026

EXAMINER KEYWORDS

Write these:

Q-Learning

- Q Value
- Q Table
- Optimal Policy
- Reward

SARSA

- On Policy
- Actual Action
- State Action Reward

Actor-Critic

- Actor
- Critic
- Policy Learning
- Feedback

- Value Function

IMPORTANT QUESTIONS

7 Mark Questions

1. Explain Q-Learning.
2. Explain SARSA.
3. Explain Actor-Critic Model.
4. Differentiate Q-Learning and SARSA.

---

14 Mark Questions

1. Explain Q-Learning with architecture and formula.
2. Explain SARSA with working and formula.
3. Explain Actor-Critic architecture.
4. Compare Q-Learning, SARSA and Actor-Critic.

MOST EXPECTED QUESTIONS

95% Probability

Explain Q-Learning.

95% Probability

Differentiate Q-Learning and SARSA.

90% Probability

Explain Actor-Critic Model.

90% Probability

Explain Q Table.

COMMON MISTAKES

❌ Confusing Q-Learning with SARSA.

❌ Forgetting Q Table.

❌ Missing formulas.

❌ Not drawing architecture.

❌ Mixing Actor and Critic roles.

TOPPER NOTES

✅ Q-Learning uses BEST future action.

✅ SARSA uses ACTUAL future action.

✅ Actor chooses actions.

✅ Critic evaluates actions.

✅ Q-Learning = Off Policy.

✅ SARSA = On Policy.

✅ Actor-Critic combines both approaches.

ONE-MINUTE REVISION BOX

```
Q-Learning

↓

Q Table

↓

Best Action

----------------

SARSA

↓

State Action Reward

State Action

↓

Actual Action

----------------

Actor-Critic

↓

Actor = Action

Critic = Evaluation
```

IF EXAM IS TOMORROW

Learn:

1. Q-Learning Formula
2. SARSA Formula
3. Q-Learning vs SARSA Table
4. Actor-Critic Diagram
5. Q Table Concept

These topics alone can fetch a very large portion of Unit 4 marks.

UNIT 4 COMPLETE ✔

You now have all 6 TOPICS

1. RNN
2. LSTM & GRU
3. Translation, Beam Search, BLEU, Attention
4. Reinforcement Learning & MDP
5. Bellman Equation, Value Iteration, Policy Iteration
6. Actor-Critic, Q-Learning, SARSA

These cover essentially the entire RGPV CS601 Unit 4 syllabus and the most frequently asked PYQ areas.