Code No.: R25CS58111PE | R25 | H.T.No. | | | 8 | R | | | | | |

CMR ENGINEERING COLLEGE: : HYDERABAD
UGC AUTONOMOUS
I–M.TECH–I–Semester End Examinations (Regular) - February- 2026
DEEP LEARNING
(CSE)

[Time: 3 Hours] **[Max. Marks: 60]**

Note: This question paper contains two parts A and B.
Part A is compulsory which carries 10 marks. Answer all questions in Part A.
Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

PART-A **(10 Marks)**

1.	a)	What is ReLU activation function?	[2M]
	b)	Define convolution in CNNs.	[2M]
	c)	Mention two applications of deep learning in computer vision.	[2M]
	d)	What is Natural Language Processing (NLP)?	[2M]
	e)	Define Named Entity Recognition (NER).	[2M]

PART-B **(50 Marks)**

2.	Explain the working of a feed-forward neural network and back propagation algorithm with diagrams.	[10M]
	OR	
3.	Explain heuristics for avoiding bad local minima and achieving faster training.	[10M]
4.	Explain CNN architecture and its components with suitable examples.	[10M]
	OR	
5.	Apply convolution and pooling operations to a given input image and compute the output feature map dimensions at each layer.	[10M]
6.	Describe object detection approaches and automatic image captioning using CNN-based models.	[10M]
	OR	
7.	Analyze an object detection system and explain how region proposals and classification are handled in CNN-based detectors.	[10M]
8.	Explain vector space models and word embeddings used in NLP.	[10M]
	OR	
9.	Apply the CBOW and Skip-Gram models to generate word embeddings for a small corpus and explain the training process in detail with examples.	[10M]
10.	Explain Named Entity Recognition and lemmatization using deep learning approaches.	[10M]
	OR	
11.	Apply CNN-based models for sentence classification and discuss feature extraction and pooling strategies.	[10M]

****

PART-A

1) a) What is ReLU Activation Function?

ReLU (Rectified Linear unit) is an activation function used in neural networks.

Def:

$$f(x) = \max(0, x)$$

It outputs:

- 0 if input is negative
- Input value itself if input is positive

purpose:

- Introduces non-linearity
- Reduces Vanishing gradient problem
- Makes training faster.

b) Define Convolution in CNNs.

Convolution: Convolution is a mathematical operation used in Convolutional Neural Networks (CNNs) to extract important features from input data (like images).

Def:

It is the process of sliding a small matrix called a filter (kernel) over the input image and computing

element-wise multiplication and sum to produce a feature map.

purpose :-

- Detect edges
- Detect patterns.
- Extract spatial features.

c) Mention Two applications of Deep Learning in Computer Vision.

1. Image classification
2. Object Detection
3. Face Recognition
4. Image segmentation
5. Medical Image Analysis.

d) What is Natural Langugе processing (NLP)?

Natural Language processing is a branch of Artificial Intelligence that enables computers to understand, interpret, and generate human language.

It combines :

- Linguistic
- Machine Learning
- Deep Learning.

Examples : Chatbots, Machine Translation, Sentiment Analysis.

e) Define Named Entity Recognition (NER).

Named Entity Recognition (NER) is a task in NLP that identifies and classifies important entities in text into predefined categories.

Common categories:

- person names
- Locations
- Organizations
- Dates

Example:

sentence: "Sundar pichai is the CEO of google!"

- Sundar pichai → person
- Google → Organization.

PART-B

(2) Explain the working of a feed-forward neural Network and back propegating algorithm with diagrams.

→ Ans Feedforward Neural Network (FNN)

1) Defination:

A feed-forward Neural Network is a type of artifical neural network in which information flows in one direction only:

Input layer → Hidden Layers (s) → output layers.

There are no loops or feedback connections.

It is mainly used for:

- Image classification
- Speech recognition
- pattern recognition
- Regression problems

2) Structure of FNN

FNN has a layered structure

4) Neurons and weights.

- weights
  → Every connection has a weight
  → weight shows strength of connection
  → similar to regression coefficients
  → Updated during training

- Artificial Neurons

Each neuron performs two steps:

1. weighted sum
2. Apply activation function.

Activation functions

Activation functions introduce non-linearity.

Common activation functions:

- Sigmoid
- ReLU
- Tanh
- Softmax.

Without activation → network behaves like linear regression.

Training of feed-forward Neural network.

Training involves adjusting weights to reduce error.

1) Input layer.

- Receives input data
- Each neuron represents one feature

Ex: If input = image with 3 features → 3 input neurons.

2) Hidden Layer(s).

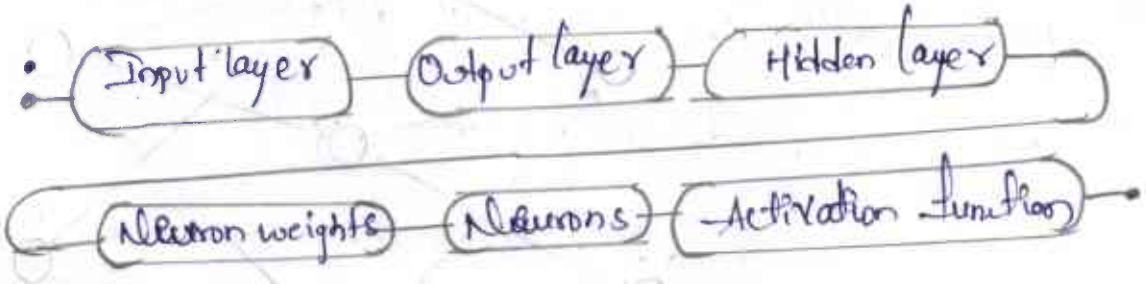
- one or more layers.
- Learns complex patterns
- Each neuron:
  - Takes weighted sum of inputs.
  - Applies activation function.

3) Output layer

- produces final result
- Number of neurons depends on problem

Ex:

- 10 neurons for digit classification (0-9)
- 1 neuron for binary classification



~~Input layer~~

~~output layer~~.

~~Hidden layer~~

Step1: Forward propagation.

- Input passes through network.
- Output is calculated

Step2: Loss Calculation.

Error between actual and predicted output.

common loss functions:

- MSE → Regression
- Cross-Entropy → classification.

Step3: Back propogation

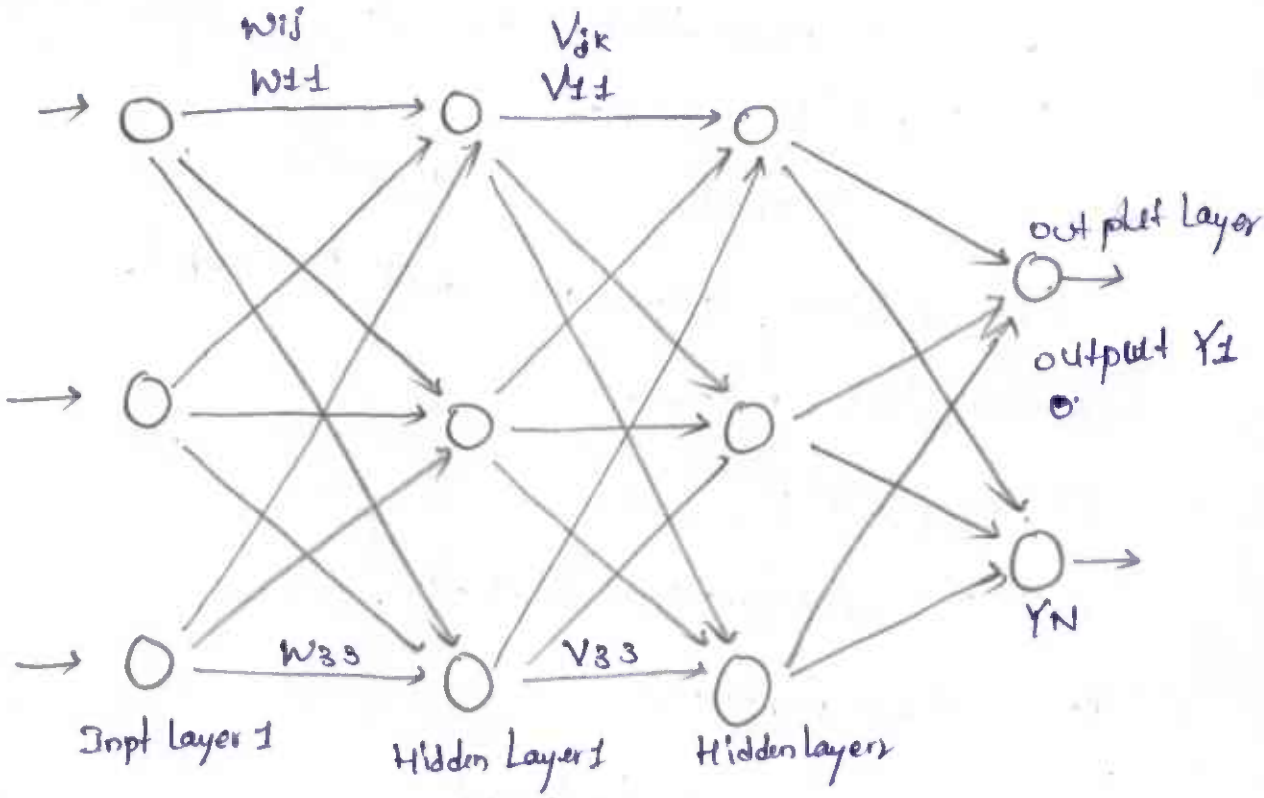
- Error propogated backword
- Gradients calculated
- Weights updated using Gradient Descent.

Algorithm:

* Back propogation stands for "backward propogation of errors"
* It is a supervised learning algorithm used for training multilayer neural networks.

- It works by computing the gradient of the loss function with respect to each weight using the chain role of calculus.

Draw



BPN Algorithm:

Back propogation works in four steps:

1. forward pass
2. Compute error at otput layer
3. Backward pass [Error propogation]
4. Update weights.

---

(3) Explain heuristic for avoiding bad local minima and achieving faster training.

Ans Heuristics for avoiding Bad Local Minima:

Def:

Heuristics are practical techniques used to improve training and reduce the change of getting stuck in poor local minima.

1. Leaky ReLU & parametric ReLU (PReLU).

→ Leaky ReLU

Instead of making all negative values zero (like ReLU), Leaky ReLU allows a small hope.

$$f(x) = \begin{cases} x & x > 0 \\ 0.01x & x < 0 \end{cases}$$

simple example:

If input = -5

ReLU → 0

Leaky ReLU → 0.05

small gradient still flows → neuron does not "die"

→ parametric ReLU (PReLU).

similar to Leaky ReLU, but slope is learned during training

$$f(x) = \begin{cases} x & x > 0 \\ ax & x < 0 \end{cases}$$

Here a learned automatically

→ model decides best slope.

2. Xavier (Glorot) Initialization.

Def:

Weights are initialized so that variance remains stable across layers.

3. Batch Normalization.

Def:

Normalizes layer input to mean = 0 and Variance = 1.

Simple Ex:

If layers output are

[100, 120, 130]

After Normalization

[-1.2, 0.3, 0.9]

④ skip connections (Residual Networks).

Used in deep networks like Resnet

Def:

Adds shortcut connection from earlier layer to later layer.

Instead of learning: $H(x)$

Networks learns $F(x) + x$

Ex:

Input → Layer 1 → Layer 2 → output

With skip connection

Input directly added to output

-) Gradient can directly flow backward

→ prevents Vanishing

Regularization (L1/L2).

Def:

Adds penalty to large weights to prevent Overfitting.

L1 → Adds |w|

L2 → Adds w²

Ex:

Without regularization

weight = 50

With L2 → penality added → weight becomes smaller

→ prevents complex model

→ Avoids Overfitting

→ Helps better minima.

Heuristics for achieving faster Training.

Introduction

faster training means the neural network reaches the minimum loss Quickly with fewer epochs and stable convergence

1. Momentum

2. Nesterov Accelerated Gradient (NAL).

3. Mini - Batch Gradient Descent.

4. proper Learning Rate & Learning Rate scheduling

5. Batch Normalization.

6. ReLU Activation Function.

7 Good weight Initialization.

4) 4. Explain CNN architecture and its components with suitable examples.

Ans Introduction:

A. convolutional Neural Network (CNN) is a type of deep learning model mainly used for:

→ Image classification

→ Object detection

→ Image Segmentation

→ Face recognition.

CNN is specially designed to process image data by automatically spatial features.

1. Input layer.

→ Takes image as input

→ Image represented as matrix of pixel values.

Example:

→ A 28 × 28 grayscale image

→ Input size = 28 × 28 × 1

2. Convolution Layer.

Purpose :

Extracts important features from image

- Uses small filters (kernels)
- Slides across image
- produces feature maps.

Example:

If filters detects vertical edges

$$\begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix}$$

It highlights vertical edges in image.

3) Activitation function (ReLU).

After convolution

$f(x) = \max(0, x)$ $f(x) = \max(0, x)$ $f(x) = \max(0, x)$.

Removes negative values

Introduce non-linearity.

Helps model learn complex patterns.

4. pooling Layer.

purpose:

Reduces size of features maps.

common type → Max polling.

Example:

from 4×4 matrix → becomes 2×2 matrix

By taking maximum value from each region.

→ Reduces computation

→ controls Overfitting

→ keeps important features

5. fully connected (fc) Layer.

. Flattens feature maps into vector.

. performs classification.

Example

If 10 classes → 10 out put neurons

6. Out put layer.

Usually uses:

. softmax (for multi-class classification).

. Sigmoid (for binary classification).

Softmax converts outputs into probabilities.

Example: Handwritten Digit classification (MNIST).

Input 28x28 image.

CNN Architecture:

1. conv Layer (32 filters)
2. ReLU
3. Max poding
4. Conv Layer (64 filters)
5. ReLU
6. Max pooling
7. Flatten
8. Dense Layer
9. Output (10 neurons - digit - 0-9).

Output: probability of each digit.

Advantages:

- Automatic feature extraction.
- parameter sharing
- translation invariance.
- High accuracy for images.

5) Ans

Introduction to convolution and pooling in CNN.

In convolution Neural Networks (CNNs). convolution and pooling are two important operations for feature extraction from images.

CNNs are mainly used in image processing, computer vision, face recognition, object detection. etc. Instead of directly analyzing raw pixel values, CNNs automatically learn important patterns like edges, textures and shapes.

1. Convolution operation.

- convolution is the core operation of CNN
- It used a small matrix called a filter (kernel).
- The filter slides over the input image.
- At each position, element-wise multiplication is performed
- The results are summed to produce a single value in the feature map.

purpose:

- Detect features such as edges, corners, textures
- Reduce parameters compared to fully concentrated layers.

2. pooling operation.

- pooling is used after convolution
  - It reduces the spatial size (height & width) of the feature map.
  - common types:
    - ⇒ max pooling → selects maximum value
    - ⇒ Average pooling → selects average value.

Purpose:

- Reduces computation
- control Overfiting
- makes model Invariant to small translations.

Now let us apply convolution and polling operations and compute output feature map.

Example problem:

Given:

Input Image = 32 x 32 x 3

1. convolution Layer.

Filter size = 5x5

Number of filter = 6

Stride = 1

padding = 0

Formula :

$$\text{Output} = \frac{(N - F + 2P)}{S} + 1$$

$$= \frac{(32 - 5 + 0)}{1} + 1 = 28$$

Output Height = 28

Output Width = 28

Output Depth = 6

After Convolution :

28 x 28 x 6

② pooling Layer.

pool size = 2x2

stride = 2

$(28-2)2+1 = 14$ | frac {(28-2)} {2} +1 = 14

$$\frac{(28-2)}{2} + 1 = 14$$

After pooling :

14 x 14 x 6

Final output Summary.

Layer	Output Size
Input	32 x 32 x 3
convolution	28 x 28 x 3
pooling	14 x 14 x 6

---

(6) Ans

Object Detection Approaches

Introduction:

Object Detection is a computer vision task that identifies and locates objects in an image

It performs:

- classification (what is the object?)
- localization (where is the object?)

Unlike image classification (which predicts one label for the whole image), object detection draws bounding boxes around multiple objects.

Main Object Detection Approaches:

Object detection methods are mainly divided into :-

A). Two-stage Detectors.

These methods first generate region proposals and then classify them.

1. R-CNN

- Extracts region proposals using selective search.
- Applies CNN to each region.
- slow because CNN runs multiple times.

2. Fast R-CNN

- Runs CNN once on entire image.
- Uses ROI pooling
- faster than R-CNN

3. Faster R-CNN

- Introduces Region proposal Network (RPN)
- Much faster and accurate.

B. One-stage Detectors.

These detect objects in a single step (no separate proposal stage).

1. YOLO (you only look once).

- Divides image into grid cells.
- predicts bounding boxes and class probabilites simultaneously.
- Very fast (real-time detection).

2. SSD (single shot Detector).

- Uses Multi-Scale feature maps.
- faster than two-stage detectors.

2. Automatic Image Captioning Using CNN-Based models.

Introduction.

Automatic Image Captioning is the task of generating a meaningful natural Language description for an image.

It combines :

- Computer Vision (CNN).
- Natural Language processing (RNN/LSTM/transformer)

CNN-Based Image Captioning Architecture

Step1: Feature Extraction (CNN).

- A pre-trained CNN like :

- VGG 16
- ResNet.

- Extracts high-level image features.
- Finally fully connected layer is removed.

Step 2: Caption Generation (LSTM).

- Feature Vector is given to an LSTM network.
- LSTM generates words one by one
- Each word is predicted based on.
  - Image features.
  - previously generated words.

Working Example

For an image of
→ family sitting at dinning table.

CNN extracts features →

LSTM generates.

"A family is sitting at a dining table."

- Basic Architecture Diagram (Text Representation).

Image → CNN → Feature Vector → LSTM → caption Output

Final conclusion:

* Object detection locates and classifies objects using two-stage approaches.

* Automatic image captioning combines CNN for vision and LSTM / Transformer for language generation.

CNN-based models play a key role in both tasks.

---

(7) Ans

Introduction:

An object Detection System Identifies and localizes multiple object in an image. It performs two main tasks:

1. Localization → Draw bounding boxes around objects.
2. classification → Assign class labels to detected objects.

Modern CNN-based detectors handle the task using either.

- Two stage detection.
- One stage detection.

1. Two stage CNN-Based Detectors.

Example: Faster R-CNN.

Two stage detectors separate the process into:

Stage 1: Region proposal.

Stage 2: classification and Bounding Box Refinement.

working flow:

Input Image

→ CNN features Extraction

→ Region proposal Network

→ ROI Pooling

→ Classification + Bounding Box Regression.

→ final Detections.

-Advantages:

1) High detection accuracy

2) Better localization.

Disadvantages:

* slower due to two stages.

2. One-stage CNN-Based Detectors.

Example: YOLO

One-stage detectors Combine region proposal and classification into a single step.

How it works:

- The image is divides into grid cells.
- Each grid cell predicts.
  - Bounding box coordinates
  - Object Confidence score.
  - class probabilites.

There is no separate region proposal stage.

Output prediction Include:

- (x, y, width, height).
- Object Confidence
- Class label.

Final Analysis.

In CNN - Based object detection Systems:

- Region proposals identify potential object locations.
- classification assigns lables to the those regions.
- Two-stage detectors explicity separate these steps.
- One-stage detectors combine both into a unified prediction.

(8)

Ans

The Vector Space model of semantics is a foundational model in Natural language processing (NLP) and Information Retrieval (IR) that represents units such as words, phrases, or documents as vectors in a continous high-dimensional semantic space

Theoritical foundation

VSM is based on the distributional hypothesis, which states: words that occur in similar contexts tend to have similar meanings

Vector Space Construction

Let a Vocabulary Contain in unique terms.

Each document or word is represented as a vector in an n-dimensional space

$$\vec{V} = (w_1, w_2, w_3 \ldots\ldots w_n)$$

Term Weighting Theory.

* Term Frequency (TF).

Tf measures how often a term appears in a document

$$Tf(t,d) = \frac{\text{frequency of } t \text{ in } d}{\text{total terms in } d.}$$

* Inverse Document frequency (IDF).

IDF measures how informative a terms is across the corps.

$$IDF(t) = Log\left[\frac{N}{df(t)}\right]$$

* TF - IDF (Theortical Justification)

$$Tf - IDF(t,d) = Tf(t,d) \times IDF(t)$$

Semantic similar theory:

VSM assumes that semantic similarity corresponds to spatial proximity.

Cosine Similarity:

Measures the angle between two vectors.

$$\cos\theta = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}|\,|\vec{B}|}$$

Example:

consider two documents.

* D1 : "Deep learning is powerful!"
* D2 : "Machine learning is powerful!"

Vocabulary:

Term	D1	D2
deep	1	0
Machine	0	1
learning	1	1
is	1	1
powerful	1	1

So,

* D1 = (1, 0, 1, 1, 1)
* D2 = (0, 1, 1, 1, 1)

So,

* D_1 = (1, 0, 1, 1, 1), D_2 = (0, 1, 1, 1, 1)

These Vector can now be compared using cosine similarity. to determine how similar the documents are.

Word Embeddings.

Def:

Word Embeddings are dense Vector representation of words words that capture semantic meaning. Unlike VSM, Embeddings place similar words close together in Vector space.

popular word Embedding Models.

1. Word2 Vec.

Developed by Google

It has two architecture:

- CBOW (Continous Bag of words) → predicts target word from contex.
- Skip-Gram → predicts Context words from target word.

2. GloVe.
   - Use global word co-occurance statistics.
   - Combines advantages of matrix. factorization and neural networks

3. Fast Text.
   - Represents words as character n-grams.
   - Handles rare and Unkown words better.

Example of word Embedding Relationship.

King - Man + Women = Queen.

This shows Embeddings capture Semantic relationship mathematically.

(9) Introduction:

Ans Word2Vec is a popular word embedding technique that converts words into dense vectors.

- CBOW (continous Bag of words)
- Skip-Gram.

Both models are shallow neural networks trained to predict words based on context.

2. Small Corpus example.

Let take a small corpus:

"the cat is sits on the mat"

Step1: Build Vocabulary.

Vocabulary = {the, cat, sits, on, the, mat}

Vocabulary size = 5.

Word	Index.
the	0
cat	1
sits	2
on	3
mat	4

Step 2 : Choose Window Size.

Let window size = 1

This Means :

- One word to left
- one word to Right.

3. CBOW Model

⇒ CBOW predicts the target word using Sorrounding context words.

Example :

Sentence : the cat sits on the mat

for target words = "sits"

Context words = {cat, on}

So training pair :

Input → (cat, on)

Output → sits.

Training pairs (window = 1).

Context	Target
(the)	cat
(cat, sits)	on
(sits the)	mat.

CBOW Architecture

Input Layer → Hidden Layer → Output Layer.

- Input size = Vocabulary size (5).
- Hidden layer = Embedding dimension (say 3).
- Output layer = Vocabulary size (5).

4. Skip-Gram Model

* Skip-Gram predicts context words from a target word.

Example:

Target word = "sits"

Context words = {cat, on}

Training pairs:

Input → sits

Output → cat.

Input → sits

Output → on

* Skip-Gram Training pairs.

Input (Target)	output (Context).
cat	the
sits	cat
sits	on
on	sits.

Skip-Gram Architecture:

Input → Hidden → output

Same structure as CBOW but objective differs.

Final Embedding Example (After training).

Suppose Embedding dimension 3.

Word	Vector
the	[0, 2, 0, 1, 0, 4]
cat	[0, 9, 0, 3, 0, 8]
sits	[0, 8, 0, 4, 0, 7]
on	[0, 7, 0, 2, 0, 6]
mat	[0.85, 0.35, 0.75]

Here:

- cat and mat Vectors are close
- Sits and one are contextually related.

Conclusion:

1. CBOW predicts a word using context.
2. Skip-Gram predicts context using a word.
3. Both use shallow neural network.

(10) Ans

Named Entity Recognition [NER].

Def:

Named Entity Recognition [NER] is an NLP task that identifies and classifies named entities in text into predefined categories such as:

- person [PER]
- Organization (ORG)
- Location (LOC)
- Date (DATE)
- Money (MONEY)

Example:

Sentence:

"Ravi works at Infosys in Hyderabad"

NER output:

Word	Entity type.
Ravi	PERSON
Infosys	ORGANIZATION
Hyderabad	LOCATION

Traditional Vs Deep learning Approaches.

Earlier methods.

- Rule-base Systems.

- CRF (Conditional Random Fields).

Modern Systems use Deep Learning.

Deep Learning Approaches for NER

1. BiLSTM - Based NER.

Architecture:

Input Sentence
→ word Embeddings
→ Bidirectional LSTM
→ Fully Connected Layer
→ Entity Tags.

→ Why BiLSTM?

- LSTM captures long-term dependencies.
- Bidirectional LSTM reads:
  - forward context
  - Backward context.

This helps ensures Valid tag Sequences.

Example:

- "B-PER" must be followed by "I-per", not "B-Loc."

This improves accuracy.

3. Transformer - Based NER.

Modern models use:

- BERT.

BERT:

- pre trained on large text corpus.
- fine - turned for NER
- produces Contextual embeddings.

Deep Learning NER Training process.

1. Convert words into embedding.
2. pass through LSTM / Transformer.
3. predict tag for each word.
4. Compute loss (cross - Entropy).

2. Lemmatization.

Def:

Lemmatization is the process of converting a word into its base or root form (Lemma).

Example

Word	Lemma.
running	run
better	good
cars	car

Deep Learning Approaches for Lemmatization.

1. Sequence-to-sequence (seq2seq) Models.

Lemmatization can be treated as a sequence transformation task.

Input → "running"

Output → "run"

Input characters.

→ Encoder (LSTM).

→ Decoder (LSTM).

→ output lemma.

This works well for morphologically rich languages.

2. Character-Level LSTM models.

- Words are broken into characters.
- LSTM learns morphological patterns.
- Useful for:
  - Irregular forms.
  - Rare words.

3. Transformer-Based models.

Models like BERT can be fine-turned for lemmatization.

- predict base form based on sentence content.
- Helps resolve ambiguity.

Example:

"Saw"

- Verb → see
- Noun → saw.

Context helps decide correct lemma.

---

(11) Ans

Introduction:

Sentence classification is the task of assigning a label to a sentence, such as:

- Sentiment Analysis (positive / Negative).
- Spam Detection.
- Topic classification
- Question classification.

CNNs, originally designed for images, are successfully applied to text classifications because they can capture local patterns (n-grams) in sentences.

2. CNN - Based Sentence Classification Model.

popular approach inspired by.

Yoon Kim (2014 paper on CNN for sentence classification).

Step-by-step Architecture.

Example sentence.

"The Movie was Very interesting".

Step 1: Word Embedding Layer.

Each word is converted into a dense vector using.

- word2vec.
- Glove.

Assume embedding size = 5.

Sentence matrix becomes:

word	5-D Vector.
The	[0.2, 0.1, 0.5, 0.3, 0.6]
Movie	[...]
was	[...]
Very	[...]
interesting	[...]

If Sentence Length = 5

Embedding size = 5

Input matrix size = 5x5.

3. convolution Layer (feature Extraction).

CNN uses 1D convolution over text.

Filter operation.

Filter size = 2 (bi-gram detector)

The filter slides over adjacent words:

- (The, movie)
- (movie, was).
- (was, very).
- (very, interesting).

Each filter extracts important phrase-level features.

Feature Extraction.

Filters detect pattern like:

- "very good".
- "not bad"
- "extremely boring"

Different filter sizes capture different n-grams:

4. Fully Connected + Softmax Layer.

After pooling:

→ concentrate pooled features

→ pass to fully connected layer.

→ Apply Softmax.

Output:

- positive (0.85)
- Negative (0.15).

Complete Architecture Flow.

Sentence :

→ word embeddings.

→ convolution (multiple filters)

→ ReLU Activation

→ Max pooling

→ fully connected Layer

→ class Label.

*Why CNN works Well for sentence classification?

1) captures local word patterns

2) Handles Variable-length sentences.

3) parallel computation (faster than RNN).

4) Less computationally expensive.

Example sentiment : sentiment Analysis.

Sentence :

"The product is not good"

CNN filter can capture:

. "not good" → negative feature.

pooling ensures this strong feature dominates prediction.

Advantages of CNN in NLP.

→ Automatic feature extraction.

→ No manual feature engineering

→ Good performance on short texts

Limitations:

* Cannot capture very long dependencies (better handled by LSTM/Transformer).

Conclusion:

CNN-based models for sentence classification:

- Use word embeddings as input.
- Extract n-gram features using convolution
- Use pooling to select important features.
- classify using fully connected + softmax.

Feature extraction and pooling are key steps that enable CNN to effectively classify text.

M.TECH I-I SEM END EXAM (Regular) FEB-2026

SCHEME OF EVALUATION-DEEP LEARNING

PART A

SL NO	THEORY	MARKS	TOTAL
a	What is ReLu activation function?	2	2
b	Define convolution in CNNs.	2	2
c	Mention two application of deep Learning	2	2
d	What is Natural processing (NLP).	2	2
e	Define Named Entity Recognition(NER)	2	2

PART B

SL NO	THEORY	MARKS	TOTAL
2.	Explain the working of a feed-forward neural network and back propagation	10	10
3.	Explain heuristics for avoiding bad local minima and achieving	10	10
4.	Explain CNN Architecture.	10	10
5.	Apply convolution and pooling approaches.	10	10
6.	Describe object detection system approaches and automatic image	05 05	10
7.	Analyze an Object detect system and explain how region proposals	10	10
8.	Explain Vector Space models	10	10
9.	Apply the CBOW and Skip-Gram models to generate word embeddings	10	10
10.	Explain Name Entity Recognition and lemmatization using deep learning	10	10
11.	Apply CNN-based models for sentence classification and discuss feature	07 03	10