Deep Learning

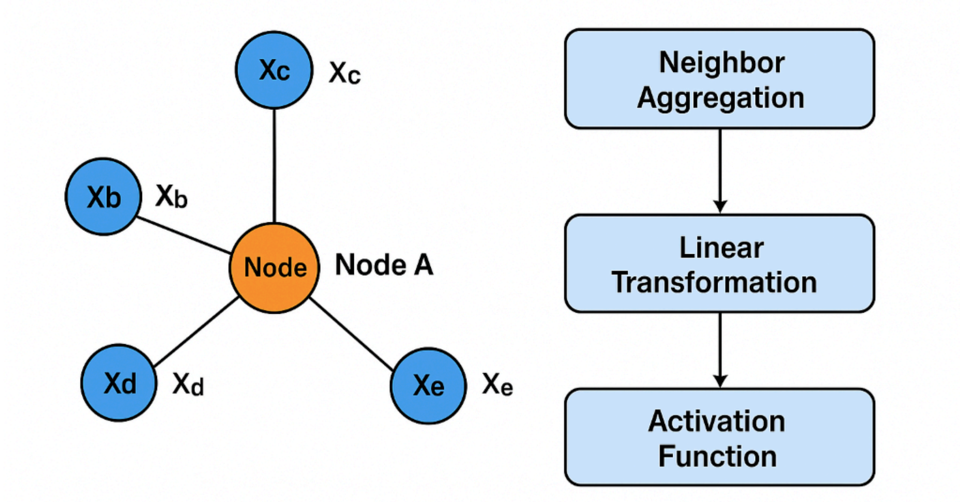
Tuesday, December 09, 2025 3:42 PM

Unit 06: Applications of Deep Learning

Nov – Dec 2022

Q7) a) Explain graph convolution approach for social network analysis? Describe RNN based framework for NLP. Write any four applications of NLP.
Ans.

Graph Convolution Approach for Social Network Analysis



Graph Convolution Approach for Social Network Analysis

Graph Convolutional Networks (GCNs) apply convolution operations on graph-structured data. Unlike traditional CNNs that work on grid-like data (e.g., images), GCNs operate on nodes and edges in a graph.

Key ideas

- In a social network, **users are represented as nodes**, and **relationships (friendships, interactions, etc.) are represented as edges**.
- GCNs learn node representations by aggregating information from a node's neighbors.
- Each node's feature is updated by combining its own attributes and those of its neighbors through a convolution-like operation.

GCN process

1. **Input graph** (G = (V, E)) with node features.
2. **Neighborhood aggregation**: For every node, gather features from connected nodes.
3. **Linear transformation + activation function** (ReLU).
4. **Propagation through multiple layers** to learn higher-level structure.

Uses in social network analysis

- Community detection (finding groups of users)
- Link prediction (predicting future friendships/ interactions)
- Node classification (detecting spam accounts, user profiling)
- Influence prediction and recommendation systems

RNN-Based Framework for NLP

Recurrent Neural Networks (RNNs) are deep learning models designed to handle **sequential data**, making them suitable for Natural Language Processing.

How RNN works

- Processes input **word by word (token by token)**.
- Maintains a **hidden state** that stores past information (memory).
- Hidden state at time *t* depends on input at *t* and hidden state from *t-1*.

Architecture

```
Input sequence → Embedding layer → RNN/GRU/LSTM layer → Output layer
```

Advantages

- Captures temporal and contextual dependencies in sentences.
- Useful for variable-length text.

Applications

- Language modeling and next-word prediction
- Machine translation
- Sentiment analysis
- Named Entity Recognition (NER)
- Speech recognition

Any Four Applications of NLP

Application	Description
Machine Translation	Converts text from one language to another (e.g., Google Translate).
Speech Recognition	Converts spoken language into text (e.g., Siri, Alexa).
Sentiment Analysis	Determines emotional tone of text (e.g., product review analysis).
Chatbots & Virtual Assistants	Human-like interaction with users for services.
Text Summarization	Creates condensed summaries of long documents.
Information Retrieval	Search engines retrieving relevant documents.

Final Summary

- **Graph Convolution Networks** help analyze social networks by aggregating neighbor node information to learn node embeddings.
- **RNN frameworks** capture sequential word relationships in NLP tasks.
- NLP has wide applications such as **translation, sentiment analysis, speech recognition, and chatbots**.

Q7) b) What are the application areas of image classification? Explain CNN for image Classification.
Ans.

Application Areas of Image Classification

Image classification is widely used in many fields where recognizing and categorizing visual objects is necessary. Major application areas include:

1. Medical Imaging

- Classifying X-ray, MRI, or CT images to detect diseases (e.g., tumors, COVID-19 detection, diabetic retinopathy).

2. Security & Surveillance

- Identifying suspicious activities or detecting objects like weapons, or recognizing individuals in CCTV footage.

3. Autonomous Vehicles

- Recognizing traffic signs, pedestrians, vehicles, and road obstacles for safe navigation.

4. Agriculture

- Identifying crop diseases, plant types, soil conditions, or classifying fruits and vegetables.

5. Retail & E-commerce

- Automated product categorization, visual search, and inventory monitoring.

6. Face Recognition & Biometrics

- Used in authentication systems like face unlock, attendance marking, border control.

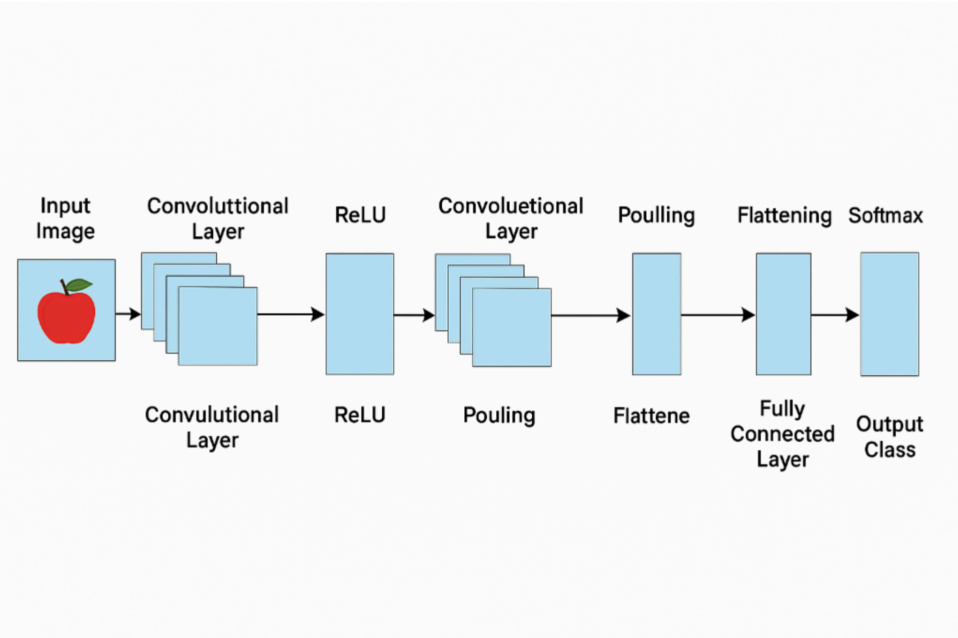
7. Satellite and Remote Sensing

- Land cover classification (water, vegetation, buildings), disaster monitoring, climate study.

8. Industrial & Manufacturing

- Quality control, defect detection, sorting and counting products in factories.
-

Explain CNN for Image Classification



Convolutional Neural Networks (CNNs) are deep learning models specifically designed to process and classify visual data such as images. They automatically learn important features (edges, textures, shapes, objects) from images without manual feature engineering.

Working of CNN

A CNN usually consists of several key layers:

1. Input Layer

- Takes the raw image (e.g., 224×224×3 for RGB).

2. Convolution Layer

- Applies learnable filters (kernels) across the image to detect features.

- Produces feature maps by sliding filters over the image.

Example operation:

```
Feature Map = Input Image * Filter (convolution)
```

3. Activation Function (ReLU)

- Introduces non-linearity and removes negative values.

4. Pooling Layer (Downsampling)

- Reduces spatial size to decrease computation and prevent overfitting.
- Common: Max Pooling selects maximum value from a region.

5. Fully Connected (FC) Layer

- Flattens the feature maps and connects them to neurons like a traditional neural network.
- Interprets the extracted high-level features.

6. Output Layer

- Uses **Softmax** to provide probabilities for each class.

Flow Diagram of CNN

```
Input Image → Convolution → ReLU → Pooling → Convolution → ReLU →
Pooling
               ↓
         Flattening → Fully Connected Layer → Softmax → Output Class
```

Advantages of CNN

- Learns features automatically
- High accuracy in image tasks
- Translation and rotation tolerant
- Reduced parameters due to shared weights

Example Applications Using CNN

Task	Model
Face recognition	VGG, ResNet
Object detection	Faster R-CNN, YOLO
Image classification	AlexNet, MobileNet

Summary

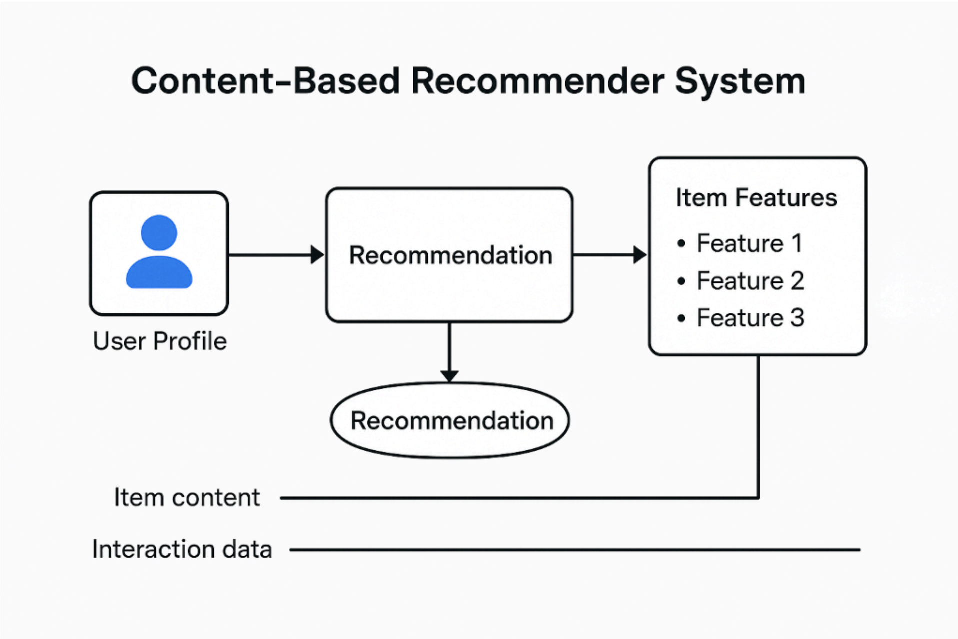
CNNs classify images by automatically learning hierarchical features through convolution and pooling operations, then predicting the proper class via fully connected layers and softmax. They are essential in various domains like medical detection, security, autonomous vehicles, and industrial automation.

Q8) a) Explain content based, collaborative and hybrid recommender system with pros and cons.
Ans.

Recommender Systems

Recommender systems suggest relevant items (movies, products, courses, music, news, etc.) to users based on their preferences. The three primary types are:

1. Content-Based Recommender System



Concept

Recommends items similar to those a user has liked or interacted with in the past.

- Uses item features (genre, category, keywords, description, attributes).
- Builds a profile for each user based on their preferences.

Example

If a user watches **action movies**, the system recommends other **action movies** with similar actors or plot themes.

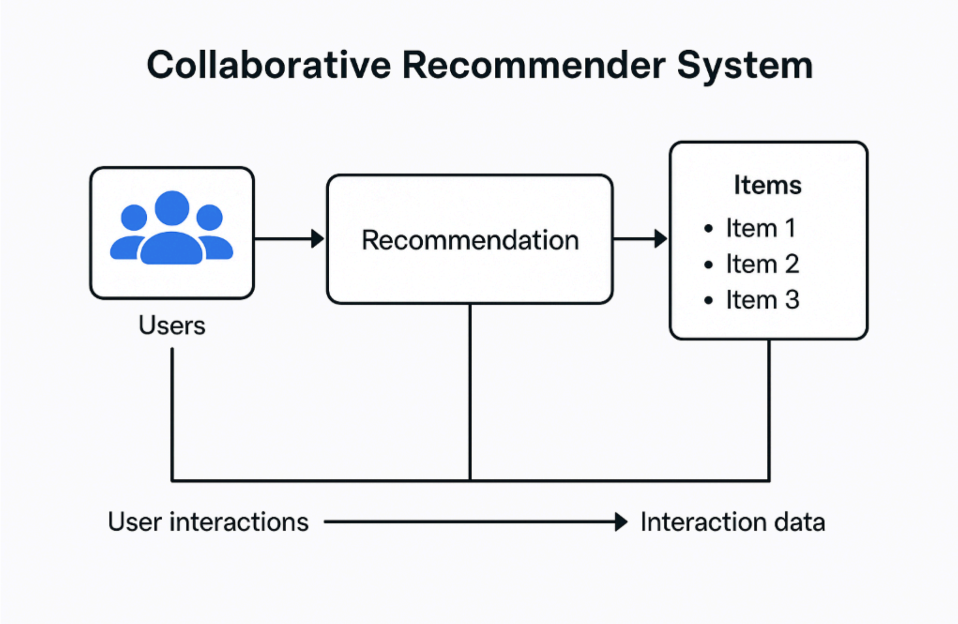
Pros

Advantages
No need for large user community data
Works well for new items (no cold-start for items)
Personalized recommendations
Transparent — explanations easily available ("recommended because similar to…")

Cons

Disadvantages
Limited ability to suggest diverse content (recommendation loop)
Depends heavily on item feature extraction
Cannot handle new users without history (cold-start for users)

2. Collaborative Filtering Recommender System



Concept

Recommends items based on similarity between **users** or **items** using past interactions and ratings.

- **User-based CF**: Users with similar tastes like similar items.
- **Item-based CF**: Items liked by similar users are recommended.

Example

Two users with similar movie rating histories → use one user's preferences to recommend movies to the other.

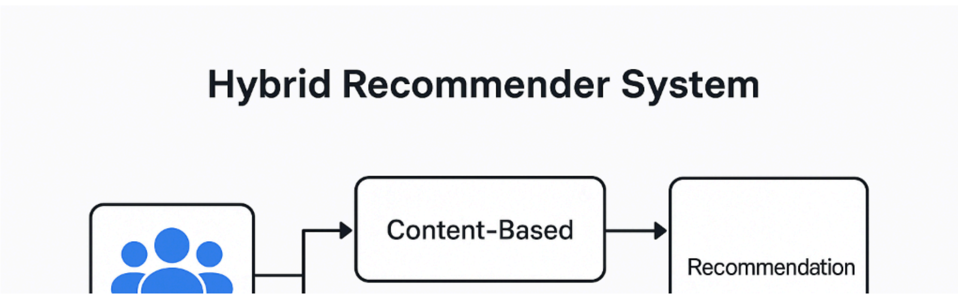
Pros

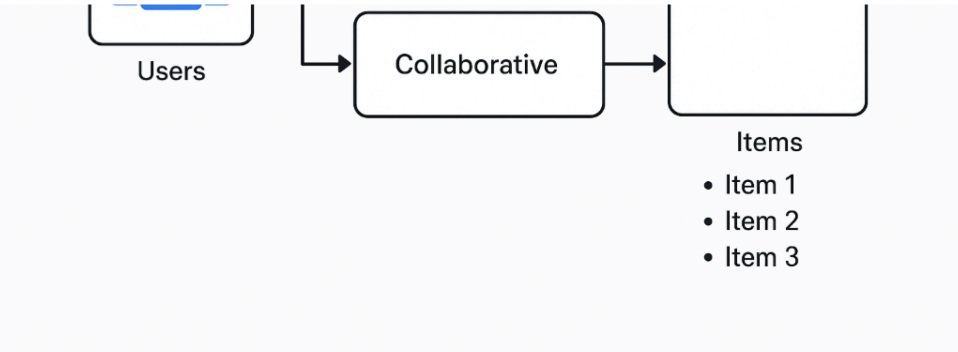
Advantages
No need for item features or domain knowledge
Can produce diverse and surprising recommendations
Improves with large community data

Cons

Disadvantages
Cold-start problem for new users/items
Sparsity problem (insufficient ratings)
Scalability issues for large datasets
Popularity bias (famous items over-recommended)

3. Hybrid Recommender System





Concept

Combines **content-based** and **collaborative filtering** methods to overcome limitations of each.

- Can mix, cascade, or switch between both approaches.

Example

Netflix uses a hybrid approach: user behavior + movie metadata + trends.

Pros

Advantages
More accurate and robust recommendations
Reduces cold-start and sparsity problems
Combines strengths of both methods
Better diversity and novelty

Cons

Disadvantages
Higher complexity
Increased computational cost
More difficult to implement and maintain

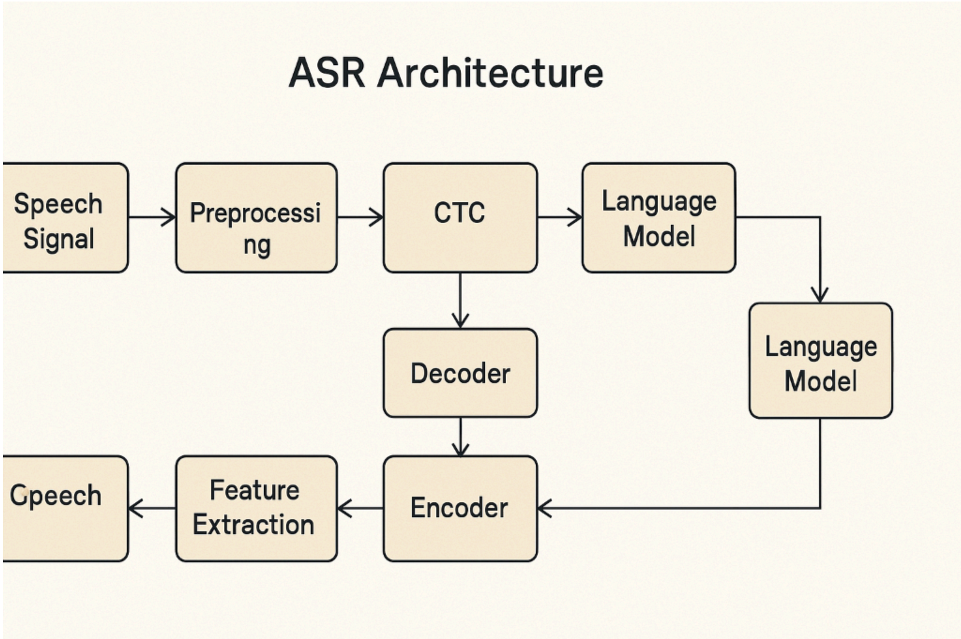
Summary Table

Type	Uses	Pros	Cons
Content-Based	Similar item recommendation	Personalized, no large data needed	Limited diversity, cold-start (user)
Collaborative Filtering	Similar user or item ratings	Diverse results, domain-independent	Cold-start, sparsity, scalability
Hybrid System	Mix of both methods	Accurate, robust, solves weaknesses	Complex and costly

Q8) b) Explain basic architecture of Automatic Speech Recognition system. Why RNN is suitable for speech recognition? How bidirectional RNNs are used in automatic speech recognition?
Ans.

1. Basic Architecture of an Automatic Speech Recognition (ASR) System

An **Automatic Speech Recognition system** converts spoken speech into text. Although modern ASR pipelines are often end-to-end, the basic (classical + neural) architecture has the following components:



Step-by-Step Architecture

1. Speech Signal Acquisition

- Microphone captures the user's voice.
- Signal is digitized (sampling + quantization).

2. Preprocessing

- Removes noise and enhances the signal.
- Common steps:
- Pre-emphasis
- Framing and windowing
- Noise reduction

3. Feature Extraction

Converts raw audio waveform into meaningful numerical features.

- Common features:
- **Mel Frequency Cepstral Coefficients (MFCCs)**
- Spectrograms
- Mel-Spectrograms
- Filter banks

These features capture frequency, pitch, and articulation patterns.

4. Acoustic Model

- Maps short segments of features to phonemes or subword units.
- Traditionally used HMM + GMM.
- Modern systems use:

- RNNs (LSTM, GRU)
- CNNs + RNNs
- Transformers

5. Language Model

- Predicts word sequences and improves grammar coherence.
- Models used:
- N-grams
- RNN-based models
- Transformer-based models (BERT, GPT)

6. Decoder

- Combines acoustic model + language model.
- Finds the most likely word sequence.
- Uses algorithms such as:
- Viterbi decoding
- Beam search

7. Output Text

Final text transcription is generated.

2. Why RNN Is Suitable for Speech Recognition?

Recurrent Neural Networks (RNNs) are well-suited for ASR due to the following reasons:

1. Speech is Sequential

- Speech is a **time-dependent** signal.
- Each sound depends on previous sounds.
- RNNs maintain a **hidden state** that stores past information.

2. Ability to Model Temporal Dependencies

- Speech phonemes unfold over time.
- RNNs (especially LSTMs/GRUs) capture **long-range dependencies**, such as:
- Coarticulation effects
- Speaking rate changes
- Intonation variations

3. Variable-Length Input Handling

- Speech inputs vary in length.
- RNNs naturally process variable-length sequences.

4. Context Preservation

- The meaning of the present sound depends on earlier sounds (e.g., "ca" vs "cha").
- RNNs learn context across multiple time steps.

5. Good for Frame-by-Frame Acoustic Modeling

- RNNs compute output for each frame of audio.
- Align naturally with speech frames (10–25 ms windows).

3. How Bidirectional RNNs Are Used in Automatic Speech Recognition

Bidirectional RNNs (BiRNNs, BiLSTMs, BiGRUs) process sequence information from both directions:

How They Work

- A **forward RNN** reads input from (t = 1 \rightarrow T)
- A **backward RNN** reads input from (t = T \rightarrow 1)
- Their outputs are combined (concatenation or summation)

This allows the model to use **past AND future context** for each time step.

Why BiRNNs Help in ASR

1. Phoneme Ambiguity Needs Future Context

Example:

- The sound /k/ in "key" vs "car" differs based on following vowel.
- BiRNNs know **what comes after** to identify sounds more accurately.

2. Better Acoustic Modeling

- Future frames help disambiguate noisy or unclear frames.
- Reduces recognition errors in fast, unclear, or co-articulated speech.

3. Used in Modern Systems

Many modern ASR systems (CTC-based, sequence-to-sequence, encoder-decoder models) use:

- **BiLSTM encoders**
- **BiRNN acoustic models**

Where Bidirectional RNNs Are Used in ASR

- **Acoustic Model Encoder** in CTC (Connectionist Temporal Classification)
- **Encoder** in sequence-to-sequence ASR models
- **Feature enhancement networks**

Summary

ASR Architecture

5. Signal acquisition
6. Preprocessing
7. Feature extraction
8. Acoustic model

9. Language model
10. Decoder
11. Output text

Why RNN?

- Handles sequential data
- Learns temporal dependencies
- Stores context across time
- Works with variable-length input

How BiRNNs help

- Use past + future context
- Improve phoneme recognition
- Reduce noise and ambiguity
- Enhance accuracy in acoustic modeling

May – Jun 2023

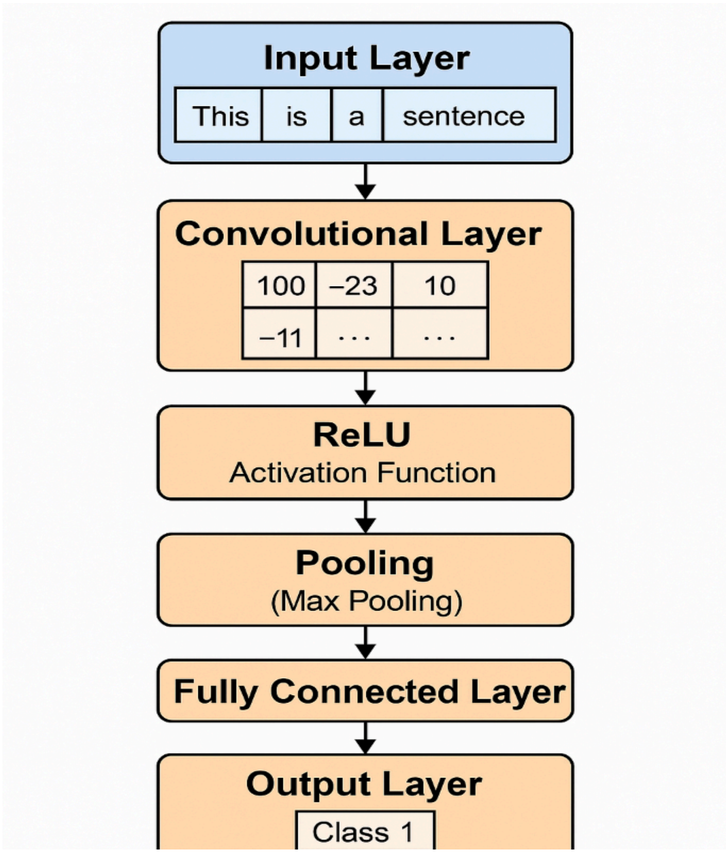
Q7) a) Explain CNN based and RNN based framework for natural language processing.

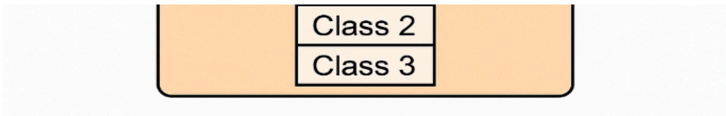
Ans.

CNN-Based Framework for Natural Language Processing

Although Convolutional Neural Networks (CNNs) were originally designed for image processing, they are also highly effective for **text classification, sentiment analysis, sentence modeling, question answering, and document categorization** in NLP.

CNNs work in NLP by treating text like a **1-dimensional signal**, applying convolution filters over sequences of word embeddings to detect important local features such as key phrases or n-grams.





1. Basic Idea

CNNs capture **local patterns in text**, such as:

- Important phrases (“very good”, “not bad”, “high risk”, etc.)
- Word order relationships in small regions
- Task-specific features that help classification

This makes CNNs good at tasks where **specific key phrases** determine meaning.

2. Architecture of CNN-Based NLP Framework

(1) Input Layer

A sentence is converted into a matrix of word embeddings:

- Each word → vector (using Word2Vec, GloVe, or trainable embeddings)
- Sentence length = *n*
- Embedding dimension = *d*

So input becomes:

Sentence Matrix: n x d

(2) Convolution Layer

Filters slide over the sentence matrix to capture patterns.

- A filter size (e.g., 2, 3, 4 words) corresponds to **bi-grams, tri-grams, 4-grams**, etc.
- Multiple filters detect multiple types of important features.

Example:

- 3-word filter learns phrase patterns
- 5-word filter detects longer contexts

Convolution outputs a **feature map**.

(3) Activation Function

ReLU is commonly used:

ReLU(x) = max(0, x)

It introduces nonlinearity and keeps only significant activations.

(4) Pooling Layer

Pooling reduces dimensionality and selects the most important features.

Most common: **max pooling**

It picks the *maximum* value from each feature map → the strongest signal.

Max pooling helps CNN identify:

- The most important phrase in the entire sentence
- Position-invariant features (phrase location does not matter)

(5) Fully Connected Layer

Outputs a fixed-length vector representing the entire text.

This layer combines all extracted local features.

(6) Output Layer

Uses **Softmax** (for multi-class) or **Sigmoid** (for binary classification).

Example outputs:

- Sentiment score (positive/negative)
- Topic label
- Spam vs non-spam
- Intent category

3. Why CNN Works Well in NLP

✓ Detects Key Phrases

Convolution filters automatically learn important n-grams.

✓ Position Invariance

Max pooling ensures the location of the phrase does not matter.

✓ Fast and Parallelizable

Unlike RNNs, CNNs:

- Don’t depend on previous time steps
- Train faster
- Work efficiently on long sentences

✓ Great for Classification Tasks

CNNs build robust sentence/document representations.

4. Applications of CNNs in NLP

- Sentiment analysis
- Sentence/document classification
- Topic categorization
- Intent detection in chatbots
- Spam detection
- Fake news classification
- Relation extraction
- Named Entity Recognition (CNN + CRF models)

5. Example CNN Architecture (Simplified)

Sentence → Embeddings → Convolution → ReLU → Max Pooling → Fully Connected Layer → Softmax

This is the standard architecture used in Kim's CNN model (Yoon Kim, 2014), a foundational CNN text classification model.

Summary

CNN-based NLP framework includes:

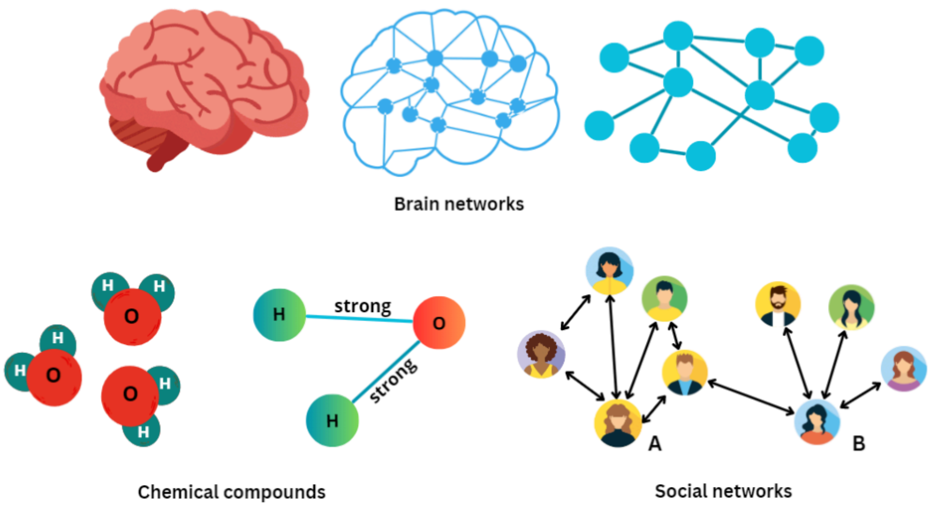
12. Word Embedding Input
13. Convolution Filters (n-gram detectors)
14. Activation (ReLU)
15. Pooling (Max Pooling)
16. Fully Connected Layer
17. Output (Softmax / Sigmoid)

CNNs excel at capturing local textual patterns like key phrases, making them powerful for classification tasks.

Q8) a) Illustrate the social network analysis using deep learning and enlist the applications of social network analysis.
Ans.

1. Illustration: Social Network Analysis Using Deep Learning

Below is a conceptual diagram **explaining how deep learning is used for analyzing a social network**.



Explanation of the Illustration

A typical **deep learning-based SNA pipeline** includes the following steps:

(1) Social Network Graph Construction

- Users → **Nodes**
- Connections (friendship, follow, messages) → **Edges**

- Node attributes (age, interests) + edge attributes (interaction frequency)

(2) Feature Representation

- Node features:
- User profile details
- Posting behavior
- Activity patterns
- Edge features:
- Interaction frequency
- Sentiment exchange
- Type of relationship

(3) Deep Learning Model (usually a GNN)

Deep learning models used:

- **Graph Neural Networks (GNN)**
- **Graph Convolutional Networks (GCN)**
- **Graph Attention Networks (GAT)**
- **RNN/LSTM** for temporal interactions
- **CNN** for text/image content of posts
- **Transformers** for social sequence patterns

These models learn:

- Node embeddings (user representations)
- Community structures
- Influence patterns
- Similarity relationships

(4) Prediction / Analysis Layer

Deep learning enables:

- Link prediction (future interaction)
- Community detection
- User classification (bot vs real user)
- Recommendation generation

(5) Output

Results may include:

- Groups of similar users
- Possible future connections
- User behavior prediction
- Content recommendation

This provides actionable insights for social platforms.

2. Applications of Social Network Analysis (SNA)

SNA is used across many fields — from social media to cybersecurity.

(A) Social Media Platforms

- Friend recommendation (Facebook)
- Content recommendation (Instagram, TikTok)
- Influencer detection
- Trend prediction

- Fake news detection

(B) Marketing & Business

- Customer segmentation
- Viral marketing strategies
- Identifying key opinion leaders
- Targeted advertising

(C) Cybersecurity

- Bot detection
- Fraud detection
- Identifying suspicious or malicious accounts

(D) Healthcare & Epidemiology

- Disease spread modeling
- Contact tracing
- Understanding transmission networks

(E) Sociology & Behavioral Science

- Studying group formation
- Social influence analysis
- Understanding human interaction patterns

(F) Recommendation Systems

- Social recommendation (friends, groups, interests)
- Collaborative filtering enhanced by social links

(G) E-commerce

- Product recommendation
- User similarity analysis
- Purchase pattern prediction

(H) Communication Networks

- Analyzing email networks
- Detecting community-based communication patterns

(I) Criminal Network Analysis

- Terrorism network detection
- Crime pattern discovery
- Identifying central suspects

(J) Education

- Understanding student collaboration networks
- Detecting isolated students
- Improving group learning strategies

Summary

Deep learning enhances Social Network Analysis by enabling:

- Better modeling of complex user interactions
- Learning from graph structures
- Predicting user behaviors
- Detecting communities and anomalies
- Recommending content effectively

Applications include:

✓ Social media analytics

✓ Marketing
✓ Cybersecurity
✓ Healthcare
✓ Sociology
✓ Recommendation systems
✓ Crime detection

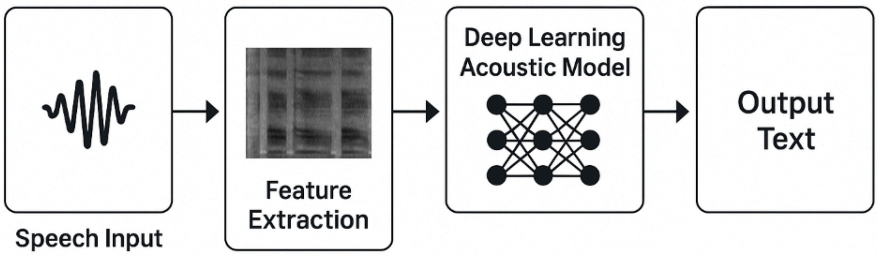
Nov– Dec 2023

Q7) a) Discuss Speech Recognition using Deep Learning.
Ans.

Speech Recognition Using Deep Learning

Speech Recognition (also called Automatic Speech Recognition – ASR) is the process of converting spoken language into text. Deep learning has significantly improved ASR quality by enabling systems to automatically learn acoustic, linguistic, and contextual patterns from large amounts of speech data.

Deep learning–based ASR replaced older statistical models (HMM + GMM) and now dominates modern applications such as Siri, Alexa, Google Assistant, and transcription systems.



1. Core Components of a Deep Learning Speech Recognition System

Modern ASR systems generally follow this pipeline:

1. Speech Input & Feature Extraction

Raw audio is first converted into meaningful features:

- **MFCCs** (Mel-frequency cepstral coefficients)
- **Spectrograms**

- **Mel-Spectrograms**
- **Log-mel filter banks**

These features compress speech into a frame-wise representation capturing pitch, tone, and articulation patterns.

2. Deep Learning Acoustic Models

Deep learning models map audio features to probability distributions over phonemes, characters, or subword units. Common architectures:

(a) Deep Neural Networks (DNNs)

- Early approach
- Multiple fully connected layers
- Still dependent on HMMs for temporal modeling

(b) Convolutional Neural Networks (CNNs)

- Good at extracting local temporal-frequency patterns
- Used for spectrogram analysis
- Capture stable features like formants & harmonics

(c) Recurrent Neural Networks (RNNs)

Special kinds used:

- **LSTM** (Long Short-Term Memory)
- **GRU** (Gated Recurrent Unit)

Why used?

- Speech is sequential and time-dependent
- RNNs handle long-term dependencies and variable-length input

(d) Bidirectional RNNs (BiLSTM / BiGRU)

- Capture both past and future context
- Improve phoneme recognition accuracy

(e) Connectionist Temporal Classification (CTC)

- Loss function designed for speech sequence alignment
- Eliminates need for frame-level labels
- Produces character/phoneme sequences directly

This combination (BiLSTM + CTC) was used in early Deep Speech models.

3. End-to-End ASR Architectures

Deep learning enables **End-to-End (E2E)** systems that directly map speech → text, without hand-designed pipelines.

Two major E2E architectures:

(a) CTC-based Models

Examples: DeepSpeech

Characteristics:

- Simpler training
- Directly maps acoustic frames to output tokens
- Works well for real-time ASR

(b) Encoder–Decoder With Attention

Examples: Listen, Attend, and Spell (LAS)

How it works:

- **Encoder** extracts high-level features from audio
- **Attention** aligns input frames to output text dynamically
- **Decoder** generates text character-by-character or word-by-word

Advantages:

- Learns alignment automatically
- Better for long utterances

(c) Transformer-based Models

Examples:

- **wav2vec 2.0**
- **Whisper**
- Conformer (Conv + Transformer hybrid)

Benefits:

- Self-attention captures global relationships
- State-of-the-art performance
- Robust even with noisy or accented speech

Transformers with large datasets now dominate ASR.

4. Training Deep Learning Models for Speech Recognition

Training involves:

- Large speech corpora (LibriSpeech, CommonVoice, Switchboard, etc.)
- Data augmentation:
- Noise addition
- Speed perturbation
- SpecAugment (masking spectrogram regions)

Deep learning learns:

- Acoustic patterns
- Linguistic patterns
- Contextual dependencies
- Speaker variations (accent, pitch, speaking rate)

5. Advantages of Deep Learning-Based Speech Recognition

✓ High Accuracy

Learns complex relationships not possible with traditional HMM-based methods.

✓ End-to-End Learning

No need for manual feature engineering or phoneme dictionaries.

✓ Handles Noisy and Real-World Speech

Deep models learn robustness to:

- Background noise
- Echo

- Variants in speaking style

✓ Scales with Data

More data → better performance.

✓ Language Independence

Can learn any language using the same architecture.

6. Applications of Deep Learning Speech Recognition

1. Virtual Assistants

Alexa, Siri, Google Assistant.

2. Voice Search

YouTube, Google, in-car voice systems.

3. Real-Time Transcription

Zoom, Teams, call centers.

4. Accessibility Tools

Speech-to-text for hearing-impaired users.

5. Dictation & Documentation

Medical transcription, legal transcription.

6. Home Automation / IoT

Smart home devices responding to voice commands.

7. Customer Support Automation

Voicebots and conversational AI.

7. Summary

Deep learning revolutionized ASR through:

- CNNs for feature extraction
- RNNs/LSTMs/GRUs for sequence modeling
- BiRNNs for contextual phoneme recognition
- CTC for alignment-free training
- Transformers (wav2vec2.0, Whisper) for state-of-the-art performance

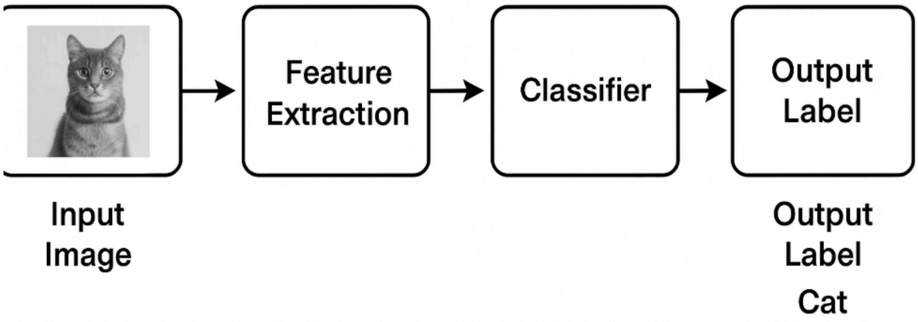
Modern speech recognition is efficient, scalable, and highly accurate due to deep learning architectures.

Q8) a) What are Image Classification Techniques.
Ans. Image classification techniques are methods used to automatically categorize images into predefined classes based on their visual content. These techniques have evolved from traditional machine-learning approaches to powerful deep-learning models.

Image Classification Techniques

Image classification techniques can be broadly divided into **two categories**:

- **1. Traditional (Handcrafted Feature-Based) Techniques**
- **2. Deep Learning-Based Techniques**



1. Traditional Image Classification Techniques

These approaches rely on **handcrafted features** extracted from the image. Common techniques include:

A. Feature Extraction Methods

(i) Histogram of Oriented Gradients (HOG)

- Captures edge direction and gradient patterns.
- Useful for object detection and recognition.

(ii) Scale-Invariant Feature Transform (SIFT)

- Detects keypoints that are invariant to scale and rotation.
- Produces feature descriptors for matching.

(iii) Speeded-Up Robust Features (SURF)

- Faster version of SIFT.
- Used for object tracking and recognition.

(iv) Local Binary Patterns (LBP)

- Captures texture by comparing a pixel with its neighbors.
- Used in face recognition and texture analysis.

(v) Color Histograms / Texture Features

- Simple statistical features.
- Useful when color or texture is discriminative.

B. Classification Algorithms (using extracted features)

After extraction, features are fed into a traditional classifier:

(i) Support Vector Machine (SVM)

- Widely used for high-dimensional data.
- Works well with HOG/SIFT features.

(ii) k-Nearest Neighbors (k-NN)

- Simple distance-based method.

(iii) Decision Trees / Random Forests

- Ensemble methods used for multi-class classification.

(iv) Naïve Bayes

- Probabilistic classifier assuming independence.

Limitations of Traditional Methods:

- Require manual feature engineering.
- Struggle with large variations in real-world images.
- Performance lower compared to deep learning.

2. Deep Learning-Based Image Classification Techniques

Deep learning eliminates manual feature engineering by learning hierarchical features directly from images.

A. Convolutional Neural Networks (CNNs)

CNNs are the most widely used deep-learning architecture for image classification.

Key operations:

- Convolution
- ReLU activation
- Pooling
- Fully connected layers

Popular CNN architectures:

- **AlexNet**
- **VGG16 / VGG19**
- **GoogLeNet / Inception**
- **ResNet**
- **DenseNet**
- **MobileNet**
- **EfficientNet**

Advantages:

- Automatically learn features.
- High accuracy.
- Robust to variations.

B. Transfer Learning

Using pre-trained CNNs (ResNet, VGG, etc.) on large datasets (ImageNet) and fine-tuning them for new tasks.

Benefits:

- Faster training
- High accuracy with small datasets

C. Data Augmentation

Not a classifier but an important technique:

- Rotation, flipping, cropping, noise addition
- Increases dataset size
- Improves generalization

D. Vision Transformers (ViT)

A modern technique that uses **self-attention** instead of convolution.

Advantages:

- Captures long-range dependencies
- SOTA (state-of-the-art) on many benchmarks

E. Hybrid CNN–Transformer Models

Combines CNN feature extraction with transformer attention.

Examples:

- ConViT
- CoAtNet

F. Capsule Networks (CapsNets)

Proposed to preserve spatial relationships better than CNNs.

Summary Table

Technique Type	Methods	Characteristics
Traditional	SIFT, SURF, HOG, LBP, Color Histogram + SVM/kNN	Require manual features; limited accuracy
Deep Learning	CNNs, ViT, Transfer Learning, CapsNets	End-to-end learning; high accuracy; current standard
Hybrid	CNN + Transformer	Best of both worlds; high performance

Conclusion

Modern image classification heavily relies on **deep learning**, especially CNNs and Vision Transformers, due to their ability to automatically learn hierarchical features and achieve high accuracy across a wide range of applications.

Q8) b) Explain Natural Language Processing in detail.
Ans.

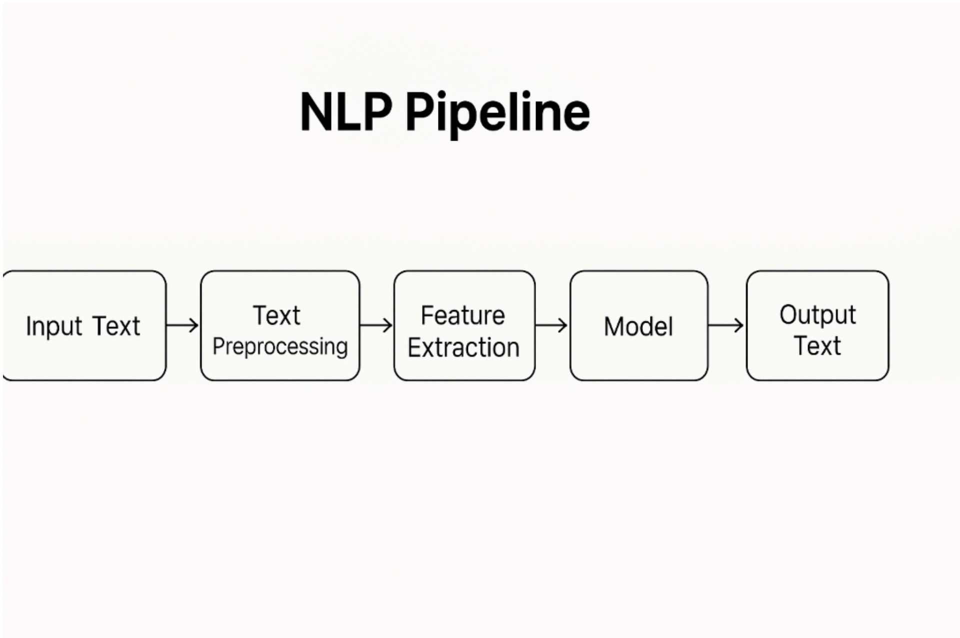
Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field of **Artificial Intelligence (AI)** and **Computational Linguistics** that focuses on enabling computers to **understand, interpret, generate, and interact using human languages** such as English, Hindi, Tamil, etc.

It lies at the intersection of:

- **Computer Science**
- **Linguistics**
- **Machine Learning / Deep Learning**

NLP allows machines to read text, hear speech, understand meaning, determine sentiment, translate languages, and communicate with humans naturally.



Why NLP Is Important?

Human language is:

- Ambiguous
- Context-dependent
- Full of grammar rules and exceptions

NLP builds models that help computers deal with this complexity.

Components of NLP

NLP involves two major components:

1. Natural Language Understanding (NLU)

Helps computers **understand** and interpret human language.

Includes:

- Syntax analysis (grammar, structure)
- Semantics (meaning)

- Pragmatics (context)
- Sentiment analysis
- Entity recognition

2. Natural Language Generation (NLG)

Helps computers **generate** natural-sounding language.

Includes:

- Text summarization
- Report generation
- Dialogue generation

Key NLP Tasks

1. Text Preprocessing

- Tokenization
- Stop-word removal
- Lemmatization/Stemming
- Normalization
- POS tagging

2. Text Classification

- Sentiment analysis
- Spam detection
- Topic categorization

3. Machine Translation

- English ↔ Spanish
- Google Translate style systems

4. Information Extraction

- Named Entity Recognition (NER)
- Relation extraction

5. Speech-Based NLP

- Speech recognition
- Text-to-speech generation

6. Question Answering & Chatbots

- Siri, Alexa, ChatGPT
- Customer support bots

7. Summarization

- Extractive or abstractive summarization

How NLP Works (Technical View)

NLP uses techniques from:

- **Linguistic rules**
- **Machine learning**
- **Deep learning**

Two major approaches are:

1. Traditional NLP (Rule-based + Machine Learning)

Uses:

- Regular expressions
- Part-of-speech tagging
- TF-IDF
- Naïve Bayes, SVM, decision trees

Limitations:

- Requires manual feature engineering
- Struggles with ambiguity and large data

2. Deep Learning–Based NLP

Uses neural networks to learn patterns directly from large datasets.

Popular models:

- **RNN / LSTM / GRU**
- **CNN for text**
- **Transformers (BERT, GPT, T5)** → State of the art

Deep learning improves:

- Context understanding
- Machine translation
- Speech recognition
- Summarization

Applications of NLP

✓ Chatbots & Virtual Assistants

Alexa, Siri, Google Assistant

✓ Sentiment Analysis

Used by companies to analyze customer feedback

✓ Machine Translation

Google Translate, DeepL

✓ Spam Detection

Filters unwanted emails

✓ Search Engines

Google search understands meaning and context

✓ Healthcare

Medical text analysis, diagnosis support

✓ Social Media Monitoring

Hate speech detection, trend analysis

✓ Document Summarization

Automatic summarization of reports and articles

Advantages of NLP

- Automates language-based tasks
- Improves human–computer interaction
- Handles huge volumes of text data
- Enables intelligent decision-making

Challenges in NLP

- Ambiguity in natural language
- Sarcasm and idioms
- Multilingual and code-mixed text
- Domain-specific vocabulary

Summary

Natural Language Processing is a technology that enables machines to understand and generate human language using a combination of **linguistic knowledge**, **machine learning**, and **deep learning**. It powers modern applications like chatbots, translation engines, sentiment analysis tools, and intelligent assistants.

May – Jun 2024

Q7) a) Explain the traditional approach and deep learning approach for Automatic Speech Recognition.
Ans.

1. Traditional Approach for Automatic Speech Recognition

Traditional ASR systems (used before deep learning) relied on **hand-crafted features**, **probabilistic models**, and **linguistic rules**. The most common framework was **HMM (Hidden Markov Model) + GMM (Gaussian Mixture Model)**.

Architecture of Traditional ASR

18. **Speech Signal Input**
19. **Feature Extraction**
    - MFCC (Mel Frequency Cepstral Coefficients)
    - PLP (Perceptual Linear Prediction)
    - Filter banks
20. **Acoustic Model** → **HMM + GMM**
21. **Language Model** → N-grams

22. **Decoder** → Uses Viterbi algorithm
23. **Output Text**

Key Components

(A) Feature Extraction

Hand-crafted features such as:

- MFCC
- LPC (Linear Predictive Coding)
- Delta + Delta-Delta coefficients

These capture speech characteristics but lose raw acoustic details.

(B) Acoustic Modeling using HMM-GMM

- **HMM** models temporal sequence of speech frames.
- **GMM** models distribution of speech features for each phoneme.

Limitations:

- GMM struggles with high-dimensional non-linear data.
- HMM assumes independence between frames → unrealistic for speech.
- Requires heavy feature engineering.

(C) Language Modeling

- N-gram probabilistic models
- Limited ability to capture long-range dependencies

Advantages of Traditional Approach

✓ Simple and interpretable

✓ Works with limited data
✓ Efficient to compute

Disadvantages

✗ Heavy manual feature engineering

✗ HMM assumptions are unrealistic
✗ Poor handling of variability (accent, noise, speaker differences)
✗ Cannot capture long-term context

2. Deep Learning Approach for Automatic Speech Recognition

Deep learning revolutionized ASR by learning features **automatically** and modeling speech **end-to-end**.

Deep learning architectures:

- **DNNs (Deep Neural Networks)**
- **CNNs**

- **RNNs / LSTM / GRU**
- **Bi-directional RNNs / BiLSTMs**
- **CTC (Connectionist Temporal Classification)**
- **Encoder–Decoder with Attention**
- **Transformers (wav2vec 2.0, Whisper, Conformer)**

Architecture of Deep Learning ASR

24. **Speech Input**
25. **Feature Extraction** (or raw waveform)
26. **Deep Learning Acoustic Model**
    - CNN / RNN / Transformer
27. **CTC or Attention Decoder**
28. **Language Model** (Neural LM)
29. **Output Text**

Key Components

(A) Feature Learning

Deep learning learns features automatically:

- CNNs extract local time-frequency patterns
- RNNs model temporal structure
- Transformers capture global context

Some modern models take **raw waveforms directly**, skipping MFCC.

(B) Acoustic Modeling using Neural Networks

DNN

- Replaces GMM in HMM systems.
- Handles nonlinear data better.

RNN / LSTM / GRU

- Speech is sequential → RNNs are ideal.
- Captures long-term dependencies (e.g., co-articulation effects).

Bidirectional RNN

- Uses past + future context for each frame.
- Boosts recognition accuracy.

(C) CTC (Connectionist Temporal Classification)

- Aligns input audio and text without frame-level labels.
- Used in Deep Speech models.

(D) Encoder–Decoder with Attention

- Encoder creates high-level speech representation.
- Attention aligns speech frames with output text.
- Decoder generates characters or words.

Examples: LAS (Listen, Attend & Spell)

(E) Transformer-Based Models

State-of-the-art ASR:

- **wav2vec 2.0**
- **Whisper (OpenAI)**
- **Conformer**

Advantages:

- Self-attention captures global context
- Very high accuracy
- Robust to noise

Advantages of Deep Learning Approach

✓ No need for manual feature engineering

✓ Learns complex patterns in speech
✓ High accuracy even with noisy data
✓ End-to-end training possible
✓ Better generalization across speakers and languages

Disadvantages

✗ Requires large datasets

✗ High computational cost
✗ Training time is long

3. Comparison: Traditional vs Deep Learning ASR

Aspect	Traditional (HMM-GMM)	Deep Learning ASR
Feature Extraction	Manual (MFCC, LPC)	Automatic (learned features)
Acoustic Modeling	HMM + GMM	DNN / CNN / RNN / Transformer
Sequence Modeling	HMM	RNN / LSTM / Attention
Accuracy	Moderate	Very high
Robustness	Low	High
Data Requirement	Low	High
Computation	Low	High
End-to-End	No	Yes

4. Summary

Traditional Approach

- Based on HMM-GMM

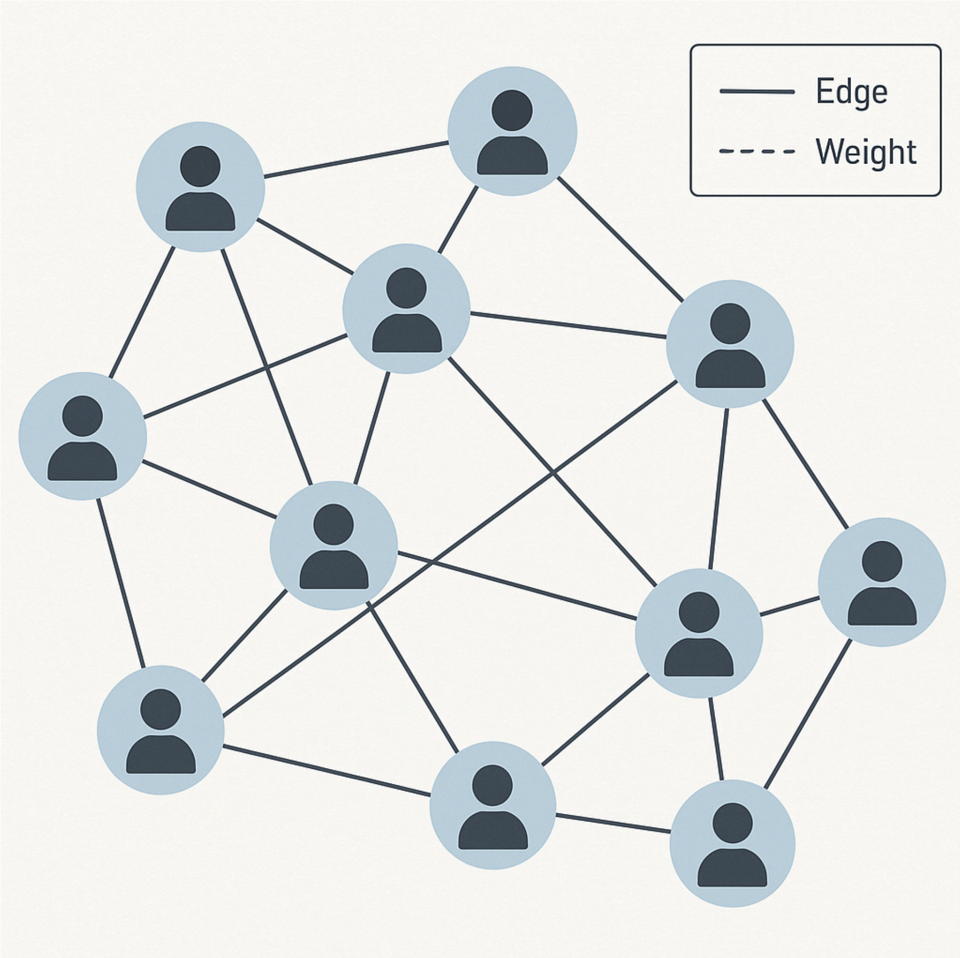
- Uses handcrafted features
- Limited accuracy and robustness

Deep Learning Approach

- Uses neural networks and end-to-end learning
- Learns features + acoustic structure automatically
- Achieves state-of-the-art performance

Deep learning has **completely transformed** modern speech recognition and is used in Siri, Alexa, Google Assistant, Zoom transcription, and many AI-powered voice systems.

Q8) a) Explain the following social network analysis terminologies i) Nodes & Edges ii) Centrality Measures iii) Weight iv) Network Level Measures
Ans.



i) Nodes & Edges

Nodes (Vertices)

- Nodes represent the **individual entities** in a social network.
- These can be:
- People
- Organizations
- Devices
- Webpages
- Events

Example:

Each user on Facebook is a **node**.

Edges (Links)

- Edges represent the **relationships or interactions** between nodes.
- These can be:
- Friendships
- Follows
- Messages
- Co-authorship
- Transactions

Edges can be:

- **Directed** (A → B) → e.g., Twitter follow
- **Undirected** (A – B) → e.g., Facebook friendship

Together, **nodes and edges form the graph** used in SNA.

ii) Centrality Measures

Centrality measures indicate the **importance, influence, or position** of a node in the network.

1. Degree Centrality

- Number of edges connected to a node.
- Represents how many direct connections a node has.

Example: A user with 1,000 friends has high degree centrality.

2. Betweenness Centrality

- Measures how often a node lies on the shortest path between other nodes.
- Represents a **broker / bridge / gatekeeper**.

Example: A node connecting two communities has high betweenness.

3. Closeness Centrality

- Measures how close a node is to all others in the network (based on shortest paths).
- Indicates how fast information spreads from that node.

4. Eigenvector Centrality

- Measures node influence based on the importance of its neighbors.
- A node connected to influential nodes gets a high score.

Example: Google's PageRank is a form of eigenvector centrality.

iii) Weight

Edges in a network may have **weights** representing the **strength, frequency, or importance** of a relationship.

Examples of weight meanings:

- **Interaction frequency** (how often two users communicate)
- **Strength of a tie** (close friend vs acquaintance)
- **Similarity score** (shared interests)
- **Transaction amount** (in a financial network)

A weighted network gives more detailed understanding than a simple binary (connected/not connected) network.

Example:

If Alice messages Bob 20 times and Charlie 2 times, the edges have weights **20** and **2** respectively.

iv) Network Level Measures

These measure properties of the **entire network**, not individual nodes.

1. Density

- Proportion of actual connections to all possible connections.
- Shows how connected the network is.
  High density → tightly knit network.

2. Diameter

- The longest shortest-path distance between any two nodes.
- Indicates the "size" of the network in terms of connectivity.

3. Average Path Length

- Average number of steps needed to connect any two nodes.
- Smaller average path length → faster information spread.

4. Clustering Coefficient

- Measures the tendency of nodes to form groups or clusters.
- High clustering → strong community structure.

5. Modularity

- Quantifies how well the network is divided into distinct communities.
- High modularity → clear, well-separated communities.

6. Network Centralization

- Indicates how much the network is dominated by a few central nodes.

Summary Table

Term	Meaning
Nodes & Edges	Basic elements of a graph; represent entities & relationships
Centrality Measures	Identify important or influential nodes
Weight	Strength or frequency of connections
Network Level Measures	Properties describing the entire network (density, diameter, modularity, etc.)

Q8) b) b) How does image classification works? Describe various image classification techniques and enlist the four advantages of using deep learning in image classification.
Ans.

1. How Does Image Classification Work?

Image classification is the process of assigning an input image to one of several predefined categories (e.g., cat, dog, car). The system analyzes visual features in the image and maps them to the correct label.

Steps in Image Classification:

(1) Image Acquisition

The input image is collected via sensors, cameras, or datasets.

(2) Preprocessing

Image enhancements to improve quality:

- Noise removal
- Resizing & normalization
- Color correction

(3) Feature Extraction

The system identifies distinguishing patterns such as:

- Edges
- Corners
- Textures
- Color information

In deep learning, features are **automatically learned** by the model.

(4) Model Training

A classifier is trained using labeled images.

(5) Classification

The trained model predicts the class of a new input image.

(6) Evaluation

Performance measured using:

- Accuracy
- Precision
- Recall
- F1-score

2. Image Classification Techniques

These techniques can be grouped into:

A. Traditional Image Classification Techniques

In early approaches, human-designed (handcrafted) features were extracted and fed into machine learning models.

1. Feature Extraction Methods

(i) SIFT (Scale-Invariant Feature Transform)

- Detects key points and descriptors
- Robust to scale, rotation, and noise

(ii) SURF (Speeded-Up Robust Features)

- Faster alternative to SIFT

- Useful for object recognition

(iii) HOG (Histogram of Oriented Gradients)

- Captures gradient orientation
- Common in pedestrian/object detection

(iv) LBP (Local Binary Patterns)

- Texture-based feature
- Used in face recognition

2. Traditional Classifiers

After extracting features, classifiers are applied:

(i) SVM (Support Vector Machine)

- Separates classes using hyperplanes
- Works well for high-dimensional data

(ii) k-NN (k-Nearest Neighbors)

- Classifies based on nearest neighbors

(iii) Random Forest

- Ensemble of decision trees

(iv) Naïve Bayes

- Probabilistic model

B. Deep Learning-Based Image Classification Techniques

Deep learning revolutionized image classification by learning features automatically.

1. Convolutional Neural Networks (CNNs)

Most widely used architecture.

Key layers:

- Convolution
- ReLU
- Pooling
- Fully connected layer
- Softmax

Popular CNN Models:

- **AlexNet**
- **VGG16**
- **GoogLeNet (Inception)**
- **ResNet**
- **DenseNet**
- **MobileNet**
- **EfficientNet**

2. Transfer Learning

Using pretrained models (e.g., ResNet50) on ImageNet and fine-tuning them for new tasks.

Benefits:

- Requires less data
- Faster training
- High accuracy

3. Vision Transformers (ViT)

New architecture replacing CNNs with:

- Self-attention mechanisms
- Patch-based processing

Provides state-of-the-art classification.

4. Ensemble Models

Combining multiple deep learning models to improve accuracy.

3. Advantages of Using Deep Learning in Image Classification

(1) Automatic Feature Extraction

- No manual feature engineering
- Learns hierarchical features from basic edges to complex shapes

(2) High Accuracy

- Outperforms traditional methods in almost all image tasks
- Robust to variations in lighting, angle, scale, and noise

(3) Scalability

- Performs well when trained on large datasets
- Can handle millions of images

(4) End-to-End Learning

- Processes raw images to final predictions in a single workflow

Additional advantages include:

- Supports real-time classification
- Adapts easily to different domains using transfer learning

Summary

How Image Classification Works

- Preprocess image → extract features → train model → classify → evaluate

Techniques

- **Traditional**: SIFT, SURF, HOG, LBP, SVM, kNN
- **Deep Learning**: CNNs, Transfer Learning, Vision Transformers

Advantages of Deep Learning

Automatic feature learning
High accuracy

Scalable to big data
End-to-end architecture

Nov– Dec 2024

Q7) b) State different applications of Deep Learning? With suitable diagram explain use of CNN for image classification.
Ans.

Applications of Deep Learning

Deep Learning is widely used across multiple domains because it can automatically learn complex patterns from large datasets.

1. Computer Vision

- Image classification
- Object detection (YOLO, Faster R-CNN)
- Face recognition
- Medical image diagnosis

2. Natural Language Processing (NLP)

- Sentiment analysis
- Machine translation
- Chatbots and virtual assistants
- Text summarization

3. Speech Processing

- Speech recognition (Siri, Alexa)
- Speaker identification
- Speech-to-text systems

4. Autonomous Vehicles

- Lane detection
- Pedestrian and traffic sign recognition
- Collision avoidance

5. Healthcare

- Disease prediction
- Medical imaging (CT, MRI, X-ray)
- Drug discovery

6. Finance

- Fraud detection
- Algorithmic trading
- Credit risk prediction

7. Robotics

- Vision-based navigation
- Motion control
- Human–robot interaction

8. Recommendation Systems

- Product/movie recommendations
- Personalized ads

Use of CNN for Image Classification

A **Convolutional Neural Network (CNN)** is a deep learning model designed to process visual data.

CNNs automatically learn hierarchical image features—from edges → textures → objects.

🔹 How CNN Works for Image Classification

1. Input Layer

Receives the raw image (e.g., 224×224×3 RGB image).

2. Convolution Layer

- Learns low-level features like edges, corners, colors
- Uses **filters/kernels** to slide over image
- Produces **feature maps**

3. Activation Function (ReLU)

- Introduces non-linearity
- Converts negative values to zero

4. Pooling Layer

- Reduces spatial size
- Makes model robust to position changes
- Usually **Max Pooling** is used

5. Fully Connected (FC) Layer

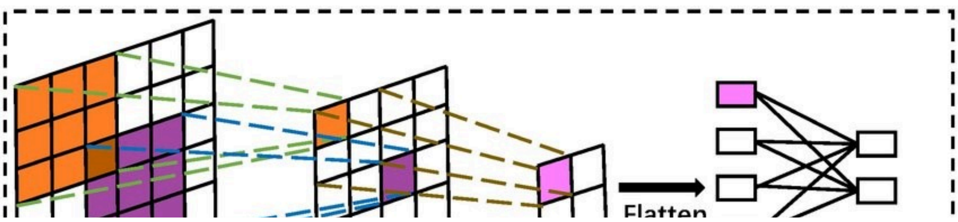
- Acts like a traditional neural network
- Combines extracted features to classify the image

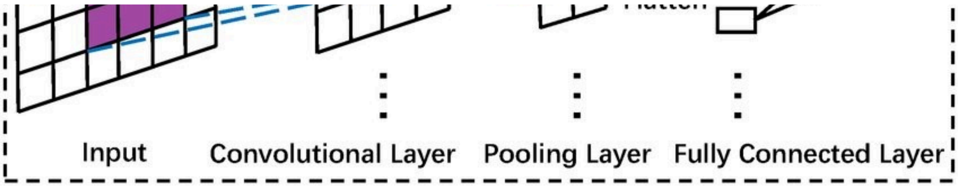
6. Output Layer

- Uses **softmax** to give final class probabilities
- Example: {Cat: 90%, Dog: 8%, Car: 2%}

Diagram: CNN for Image Classification

Below is the **diagram image illustrating the CNN workflow**:





Summary

Component	Purpose
Convolution	Extract local features (edges, textures)
ReLU	Add non-linearity
Pooling	Reduce size + prevent overfitting
FC Layer	Final classification
Softmax	Outputs class label

CNNs are ideal for image tasks because they learn powerful hierarchical features automatically, without manual feature engineering.

Q8) a) State applications of NLP. Why RNN is preferred as compare to CNN for Natural Language Processing? State different NLP tasks where RNN is used.
Ans.

Applications of NLP (Natural Language Processing)

NLP is used in many real-world applications that involve understanding or generating human language.

1. Machine Translation

- Converts text from one language to another (e.g., Google Translate).

2. Sentiment Analysis

- Determines emotion or polarity of text (positive, negative, neutral).

3. Chatbots & Virtual Assistants

- Siri, Alexa, Google Assistant
- Customer service chatbots

4. Text Summarization

- Automatic creation of summaries (extractive or abstractive).

5. Information Retrieval

- Search engines retrieving relevant documents (Google search).

6. Named Entity Recognition (NER)

- Identifying names, locations, organizations in text.

7. Part-of-Speech (POS) Tagging

- Assigning grammatical labels to words.

8. Speech Recognition

- Converting spoken speech to text.

9. Spam Detection

- Filtering unwanted emails or messages.

10. Question Answering

- Systems like ChatGPT, QA bots, reading comprehension models.

Why RNN Is Preferred Over CNN for NLP?

Although CNNs can process text, **RNNs are better suited for sequential data** like natural language.

1. RNNs handle sequential (time-ordered) data

Language is inherently sequential:

- Meaning of a word depends on previous words
- RNN stores **hidden states** containing past information

CNNs only capture **local patterns**, not long-term relationships.

2. RNNs maintain context and memory

RNNs (especially LSTMs/GRUs) remember:

- Long-term dependencies
- Grammar
- Context from earlier in a sentence

Example: In

“The boy who lived in the big house is playing”,
the subject "boy" appears far from the verb "is" → RNNs handle this.

CNNs struggle with long-range dependencies.

3. RNNs handle variable-length input

Sentences differ in length.

RNNs naturally process sequences token-by-token.

CNNs require fixed-size kernels and may need padding.

4. RNNs model temporal relationships

For speech or text input, the order of words matters.

RNNs capture:

- Sequence
- Rhythm
- Time-step dependencies

CNNs treat text more like images → local pattern matching.

NLP Tasks Where RNN Is Used

RNNs (and LSTM/GRUs) are widely used for sequence modeling tasks:

1. Language Modeling

Predicting next word in a sequence.

2. Machine Translation

Sequence-to-sequence RNN models (Encoder–Decoder with attention).

3. Speech Recognition

Converting speech waveforms into text.

4. Text Generation

Generating paragraphs, poetry, and summaries.

5. Sentiment Analysis

Understanding emotional tone of text.

6. Named Entity Recognition (NER)

Identifying people, places, and organizations.

7. POS Tagging

Assigning grammatical labels (noun, verb, adjective, etc.).

8. Sequence Labeling

Any task where output is a sequence aligned to input.

9. Question Answering & Dialog Systems

Understanding and generating human-like responses.

10. Time-Series NLP Tasks

Predicting sequences where order matters.

Summary

Applications of NLP

- Translation, summarization, chatbots, search, sentiment analysis, speech recognition, etc.

Why RNN is preferred over CNN

- Handles sequential data
- Maintains context (memory)
- Captures long-term dependencies
- Naturally processes variable-length inputs

RNN-based NLP Tasks

- Machine translation
- Language modeling
- Speech recognition
- Text generation
- NER, POS tagging
- Sentiment analysis

Q8) b) State and Explain different centrality measures used in social network analysis.
Ans.

Centrality Measures in Social Network Analysis

Centrality measures quantify the **importance, influence, or prominence** of nodes in a network.

Different measures capture different aspects of importance.

The most widely used centrality measures are:

- **Degree Centrality**
- **Betweenness Centrality**
- **Closeness Centrality**
- **Eigenvector Centrality**
- **PageRank Centrality** (variant of eigenvector)

1. Degree Centrality

Definition:

Degree centrality is the number of edges (connections) a node has.

Types:

- **In-degree** (for directed networks): Number of incoming links
- **Out-degree**: Number of outgoing links

Interpretation:

- High degree → the node is **highly active** or **popular**.
- It measures **immediate influence** in the network.

Example:

On Facebook, a user with many friends has high degree centrality.

Formula:

$C_D(v) = \deg(v)$

2. Betweenness Centrality

Definition:

Measures how often a node lies on the **shortest paths** between other pairs of nodes.

Interpretation:

- A node with high betweenness acts as a **bridge**, **broker**, or **gatekeeper**.
- It controls information flow between network groups.

Applications:

- Finding key connectors
- Detecting bottlenecks
- Identifying influential mediators

Formula:

$C_B(v) = \sum (\sigma\_st(v) / \sigma\_st),\ \text{for all } s \neq v \neq t$

Where:

σ_st = total number of shortest paths from node s to node t

$\sigma\_st(v)$ = number of those shortest paths that pass through node v

3. Closeness Centrality

Definition:

Measures how close a node is to all other nodes based on shortest paths.

Interpretation:

- High closeness → node can **spread information quickly**.
- Measures efficiency of communication.

Applications:

- Efficient information broadcasters
- Nodes that can reach others with minimum steps

Formula:

$C_c(v) = 1 / \sum d(v, t)$

Where:

- (d(v,t)) = shortest path distance

4. Eigenvector Centrality

Definition:

Measures node importance based on the importance of its neighbors.

Interpretation:

- A node connected to **important or influential nodes** gets more weight.
- Not just quantity of connections, but their quality.

Applications:

- Influencer identification
- Detecting authority figures in networks

Example:

Google PageRank is based on eigenvector centrality.

Formula:

$C_e(v) = (1/\lambda) \times \sum C_e(t)$

5. PageRank Centrality

Definition:

A variant of eigenvector centrality used by Google to rank webpages.

Key Idea:

- A node is important if many important nodes link to it.
- Includes a **damping factor** to account for random surfing.

Formula:

$PR(v) = (1 - d)/N + d \times \sum (PR(u) / Out(u))$

Where:

- d = damping factor (usually 0.85)
- N = total nodes

Summary Table of Centrality Measures

Centrality Type	What It Measures	Indicates
Degree	Number of connections	Popularity / activity
Betweenness	Control over shortest paths	Broker, mediator, bottleneck
Closeness	Distance to others	Speed of information spread
Eigenvector	Influence of neighbors	Quality of connections
PageRank	Weighted importance of incoming links	Authority, web ranking

When to Use Each Centrality Measure

Goal	Best Centrality
Identify most connected users	Degree
Identify influencers or bridges	Betweenness
Find nodes that spread info fastest	Closeness
Measure overall influence in network	Eigenvector / PageRank

May – Jun 2025

Q7) a) Explain how deep learning is used in recommender systems?
Ans.

How Deep Learning Is Used in Recommender Systems

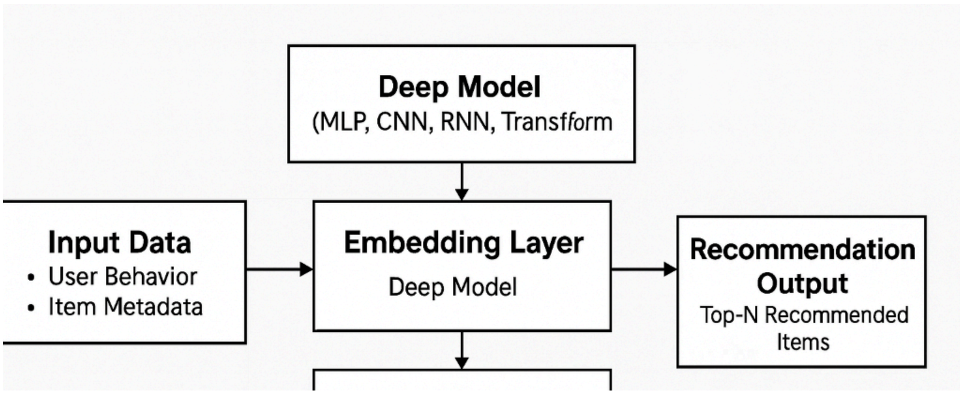
Recommender systems help predict user preferences and suggest relevant items such as movies, products, songs, or news articles.

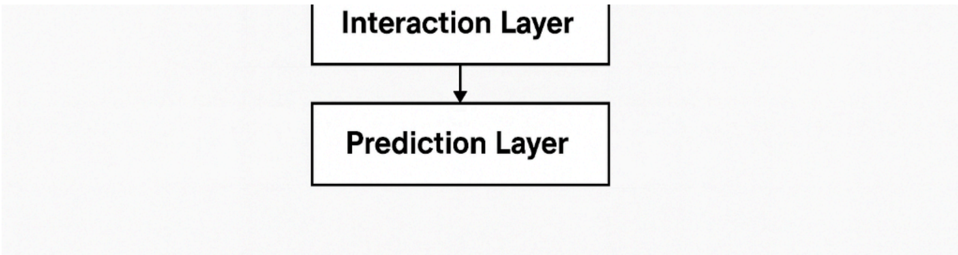
Deep Learning has dramatically improved recommender systems by enabling them to learn:

- complex user behavior
- nonlinear patterns
- contextual information
- temporal changes in user interest

Traditional recommender systems (content-based, collaborative filtering) rely on simple similarity measures or matrix factorization.

Deep learning goes beyond these by learning rich **feature representations** of users and items.





1. Deep Learning Techniques Used in Recommender Systems

1. Neural Collaborative Filtering (NCF)

Deep learning replaces traditional matrix factorization with neural networks.

- User and item embeddings are fed into a **multi-layer neural network (MLP)**
- Learns nonlinear interaction patterns
- Improves rating prediction and click-through rate (CTR)

Example: Neural Matrix Factorization (NeuMF)

2. Autoencoders for Recommendations

Autoencoders learn compressed representations of user-item interactions.

How used:

- Input: user rating vector
- Hidden layer: learns a latent representation
- Output: reconstructed ratings used to fill missing values

Useful for:

- Collaborative filtering
- Denoising sparse user-item matrices

3. Convolutional Neural Networks (CNNs)

CNNs are applied when items contain **images or text**.

Examples:

- Product recommendation → CNN extracts product image features
- Movie recommendation → CNN processes poster images
- News recommendation → CNN extracts local text features

CNN-generated features are combined with user preferences.

4. Recurrent Neural Networks (RNNs) for Sequential Recommendation

RNN variants (LSTM, GRU) capture **temporal patterns** in user behavior.

Used for:

- Next-item prediction
- Session-based recommendation
- News/music streaming platforms

Example: Predicting the next song based on listening history.

5. Attention Mechanisms

Attention models focus on the most relevant user interactions.

Used in:

- Personalized ranking
- Context-aware recommendations
- Sequence modeling

Helps identify which past activities influence the next recommendation.

6. Transformers in Recommender Systems

Transformers (self-attention models) excel at modeling long sequences.

Applications:

- Session-based recommendation
- Search ranking
- Large-scale personalization (Amazon, YouTube)

Transformers outperform RNNs on long user histories.

7. Graph Neural Networks (GNNs)

Used in **social recommendation** and modeling user-item interaction as graphs.

Advantages:

- Captures relationships between users
- Detects communities and mutual interests
- Learns user/item embeddings through graph convolution

Example: Pinterest's PinSage uses GNNs for content recommendation.

2. Why Deep Learning Improves Recommender Systems

✓ Learns nonlinear and complex patterns

Deep networks model complicated relationships between users and items.

✓ Handles large and high-dimensional data

Images, text, clicks, watch history—deep networks process all.

✓ Supports multi-modal recommendation

Combines:

- text
- images
- audio
- video
- user behavior logs

✓ Automatically learns feature representations

No need for manual feature engineering.

✓ Improves personalization

Models individual preferences more accurately.

✓ Adapts to evolving user behavior

RNN/Transformer models capture temporal changes in interests.

3. Deep Learning Recommender System Pipeline (Summary)

Input Data
User behavior
Item metadata
Images, text, social data
Embedding Layer
Converts users and items into dense vectors
Deep Model
MLP, CNN, RNN, Transformer, GNN
Interaction Layer
Learns how users and items relate
Prediction Layer
Rating prediction, ranking, similarity scores
Recommendation Output
Top-N recommended items

4. Examples of Deep Learning in Real Systems

Netflix

- Deep neural networks for ranking, personalization
- Uses CNNs for video thumbnails
- RNNs for user viewing sequences

Amazon

- Uses deep learning for product embeddings
- Sequential models for personalized suggestions

YouTube

- Deep neural networks for user ranking & candidate generation
- Processes billions of interactions daily

Spotify

- RNNs & Transformers for next-track prediction

Conclusion

Deep learning is used in recommender systems to:

- Learn rich user and item features
- Capture sequential and contextual behavior
- Integrate multi-modal data
- Improve accuracy and personalization

Techniques like **NCF, Autoencoders, CNNs, RNNs, Transformers, and GNNs** have made recommender systems far more powerful and intelligent than traditional approaches.

Q7) b) Explain deep learning-based framework for NLP.
Ans.

Deep Learning–Based Framework for NLP

Deep Learning has transformed NLP by enabling models to automatically learn **semantic, syntactic, and contextual patterns** from text without manual feature engineering.

A typical deep learning NLP framework includes the following steps:

1. Text Input

Raw text is provided as input:

- Sentences
- Paragraphs
- Documents
- Speech-to-text transcripts

Deep learning models cannot work directly with raw text, so **tokenization** is applied.

2. Text Preprocessing

Common preprocessing steps:

- Tokenization (splitting into words/subwords)
- Lowercasing
- Stop-word removal (optional in deep learning)
- Padding sequences
- Handling unknown words

Note: Deep learning models rely less on heavy preprocessing compared to traditional NLP.

3. Word Representation (Embeddings)

Text is converted into dense numerical vectors known as **embeddings**.

Types of embeddings:

(a) Traditional embeddings

- Word2Vec
- GloVe
- FastText

(b) Deep contextual embeddings

- ELMo
- BERT embeddings

- Transformer-based representations

These embeddings capture:

- Meaning of words
- Context
- Semantic similarity

4. Deep Learning Model

Different neural architectures are used to learn patterns in text:

A. RNN-Based Models (LSTM, GRU)

- Capture sequential structure of language
- Handle long-range dependencies
- Good for translation, sentiment analysis, speech recognition

RNN pipeline:

Input → Embedding → LSTM/GRU → Output

B. CNN-Based Models

Used for:

- Text classification
- Detecting key phrases

CNN pipeline:

Embedding → Convolution → Max Pooling → Dense → Output

C. Transformer-Based Models

Transformers revolutionized NLP. Examples:

- **BERT**
- **GPT**
- **T5**
- **XLNet**

Benefits:

- Capture long-range dependencies using self-attention
- Parallel processing (faster than RNNs)
- State-of-the-art for most NLP tasks

Transformer pipeline:

Embedding → Multi-head Attention → Feed-forward layers → Output

5. Sequence Modeling / Output Layer

Depending on the task, the output changes:

Text Classification

→ Softmax output

(e.g., spam vs non-spam)

Sentiment Analysis

→ Positive / negative / neutral

Machine Translation

→ Decoder generates translated sentence word-by-word

NER / POS Tagging

→ Tag for each token

Text Generation

→ Predict next word tokens

Question Answering

→ Start and end token probabilities

6. Training and Optimization

Deep NLP models are trained using:

- Backpropagation
- Loss functions: cross-entropy, CTC, next-token prediction
- Large datasets (Wikipedia, books, online text)

Optimization techniques:

- Adam optimizer
- Gradient clipping
- Regularization and dropout

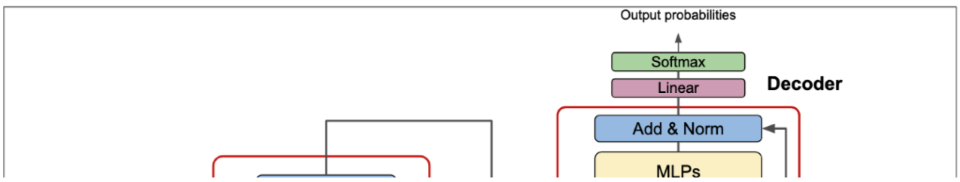
7. Final Output

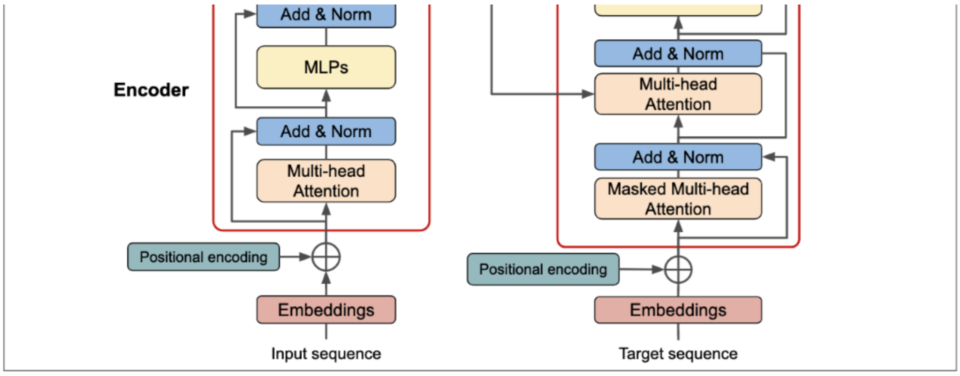
The system produces:

- Classified text
- Generated text
- Summaries
- Translated text
- Named entities
- Answers to questions

Deep learning models generalize well and deliver high accuracy.

Diagram: Deep Learning-Based NLP Framework





Summary

Component	Role
Input & Preprocessing	Prepare text for modeling
Embeddings	Convert words into numerical vectors
Deep Model (RNN/CNN/Transformer)	Learn language patterns
Output Layer	Produce predictions such as class labels or sentences
Training	Optimize model parameters

Deep learning allows NLP systems to understand **context**, **meaning**, and **relationships** much better than traditional rule-based or statistical models.

Q8) a) Explain the role of CNNs as an image classifier.
Ans.

Role of CNNs as an Image Classifier

Convolutional Neural Networks (CNNs) are deep learning models specifically designed for visual data. They perform exceptionally well in **image classification**, where the goal is to assign an input image to one of several predefined categories (e.g., cat, dog, car).

CNNs automatically learn useful patterns from images, such as:

- edges
- textures
- shapes
- objects

This makes them far superior to traditional image classification methods that rely on manual feature extraction.

Why CNNs Work Well for Image Classification

CNNs classify images by processing them through multiple layers, each learning increasingly abstract features.

1. Convolution Layers Learn Local Features

The convolution operation uses filters/kernels that slide across the image to detect:

- edges
- lines
- corners
- color patterns

Each filter learns a specific visual pattern from the image.

Role:

✓ Extracts meaningful local features automatically

✓ Reduces the need for manual feature engineering

2. Activation Functions Add Non-linearity

After convolution, activation functions such as **ReLU** introduce non-linearity.

Role:

✓ Allows network to learn complex patterns

✓ Removes negative values improving efficiency

3. Pooling Layers Reduce Spatial Dimensions

Max pooling or average pooling reduces the size of feature maps.

Role:

✓ Makes CNN more computationally efficient

✓ Reduces risk of overfitting

✓ Provides translation invariance (feature remains recognizable even if moved)

4. Multiple Convolution + Pooling Layers Build a Hierarchy

CNNs learn features at different levels:

CNN Layer	Learned Feature
Early Layers	Edges, corners
Middle Layers	Shapes, textures
Deep Layers	Entire objects like faces, animals

Role:

✓ Hierarchical feature learning extracts complete object patterns

✓ Helps accurately classify complex images

5. Fully Connected Layers Perform Final Classification

Flattened feature maps are fed into fully connected layers.

Role:

✓ Combines all learned features

✓ Predicts the final class label using Softmax

6. Softmax Layer Outputs Probabilities

Softmax produces probability scores for each class.

Example:

- Cat: 0.90
- Dog: 0.08
- Car: 0.02

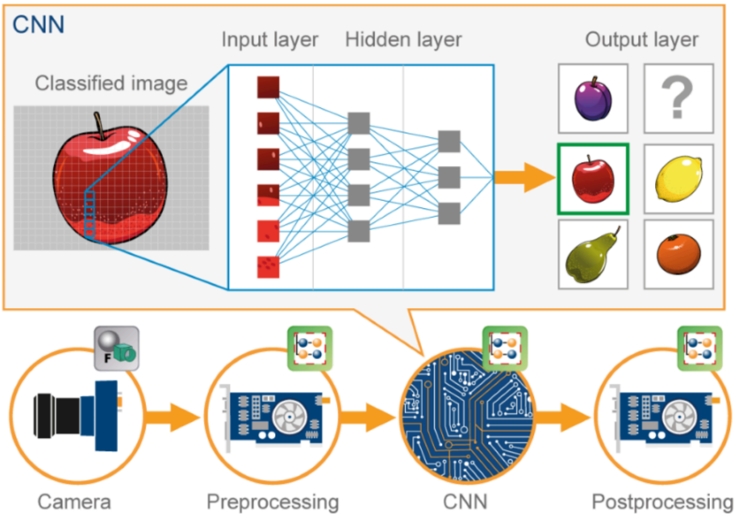
The highest probability becomes the predicted label.

Overall Role of CNN in Image Classification

CNNs:

- Automatically learn and extract features
- Handle variations in pose, lighting, and scale
- Avoid manual feature engineering
- Reduce computation using shared weights and pooling
- Achieve state-of-the-art accuracy in real-world tasks

Diagram: CNN as an Image Classifier



Summary

The role of CNNs as image classifiers is to:

Extract local and global image features through convolution.
Reduce dimensionality using pooling.
Learn complex visual patterns in hierarchical layers.
Combine features through fully connected layers.

Output the predicted class using softmax.

CNNs are the backbone of modern computer vision and power applications like:

- Face recognition
- Medical image diagnosis
- Self-driving cars
- Retail and surveillance systems

Q8) b) How deep learning is used in social network analysis?
Ans.

How Deep Learning Is Used in Social Network Analysis (SNA)

Deep learning enhances Social Network Analysis by allowing computers to learn complex patterns from large-scale social data such as user interactions, friendships, posts, comments, and network structures.

Traditional SNA methods rely on graph statistics, but deep learning can capture:

- Non-linear relationships
- Community structures
- User behavior patterns
- Influence propagation
- Temporal dynamics in interactions

1. Graph Representation Learning

Deep learning helps convert social networks into meaningful numerical representations (embeddings).

Techniques Used:

- **Graph Neural Networks (GNNs)**
- **Graph Convolutional Networks (GCN)**
- **Graph Attention Networks (GAT)**
- **DeepWalk, Node2Vec**

Role:

Generate **node embeddings** that encode:

- User similarity
- Influence
- Community membership

These embeddings are used for downstream tasks like classification, clustering, and recommendation.

2. Community Detection

Deep learning models identify clusters of users with similar behavior.

Methods:

- GNNs detect communities by learning network structures
- Autoencoders reconstruct graph connections

Applications:

- Detecting interest groups
- Identifying fraud/collusion groups
- Finding influencer clusters

3. Link Prediction

Deep learning predicts whether two users are likely to interact in the future.

Uses:

- Friend recommendation
- Social connection growth prediction
- Fake account detection

Models used:

- Graph Autoencoders
- GNN-based link predictors
- Siamese neural networks

4. Influence Modeling & Information Diffusion

Deep learning predicts:

- How information spreads
- Who influences whom
- Viral content propagation

Models:

- RNNs / LSTMs to model temporal interaction patterns
- Attention models to identify key influencers

5. Node Classification

Deep learning can label users based on their:

- Interests
- Age group
- Role in the network
- Spam/bot behavior

Techniques:

- GCNs
- GATs

6. Social Recommendation Systems

Deep learning enhances recommendations by combining:

- User social connections
- Interaction history
- Content data (text, images, video)

Models:

- GNN-based recommenders
- Neural collaborative filtering
- Transformers

7. Sentiment and Content Analysis

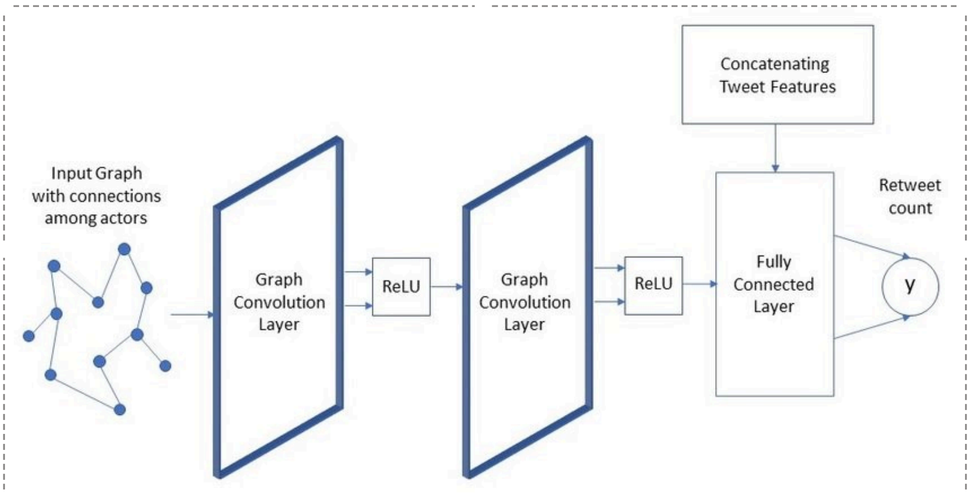
Used for analyzing:

- Social media posts
- Comments
- Reviews
- Emotions & opinions

Deep learning models:

- CNNs
- RNNs / LSTMs
- Transformers (BERT, GPT)

Deep Learning–Based SNA Framework Diagram



Summary of How Deep Learning Helps in SNA

Deep Learning Technique	Role in SNA
GNN/GCN/GAT	Learn graph structure & detect communities
RNN/LSTM	Model temporal user interactions
CNN	Analyze text, images, videos in posts
Autoencoders	Link prediction, anomaly detection
Transformers	Social recommendations, sequence modeling
Embeddings	Represent users & relationships numerically

Final Answer Summary (Exam-Ready)

Deep learning is used in social network analysis to:

- Learn user and network embeddings
- Detect communities
- Predict future links and interactions
- Analyze user-generated content
- Identify influential users
- Provide personalized recommendations
- Detect spam, bots, and anomalies

This enables powerful, accurate, and scalable analysis of complex social platforms like Facebook, Twitter, LinkedIn, Instagram, and YouTube.