SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai
Accredited by NAAC-UGC with 'A++' Grade (Cycle III) &
Accredited by NBA (B.E - CSE, EEE, ECE, Mech&B.Tech.IT)
COIMBATORE-641 035, TAMIL NADU

Reg. No: | | | | | | | | | | | | |

B.E/B.Tech- Internal Assessment – II
Academic Year 2024-2025 (Even Semester)
Sixth Semester
19CST302-NEURAL NETWORKS & DEEP LEARNING

A

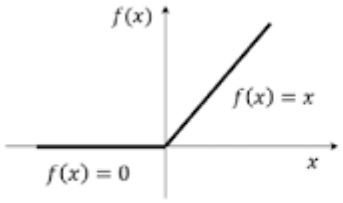
Time: 1 1/2 Hours **Maximum Marks: 50**

Answer All Questions

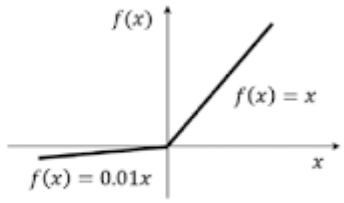
PART-A (5 x 2 = 10 Marks)

		CO	Blooms
		CO2	Ana

1. Differentiate ReLU and Leaky ReLU with neat diagram.



ReLU activation function



LeakyReLU activation function

2. List out the causes of Exploding gradient. CO2 Rem

The primary causes include:

1. Deep Network Architecture

- When a neural network has too many layers, the gradients can exponentially increase as they propagate backward.
- This results in extremely large weight updates, making training unstable.

2. Improper Weight Initialization

- If weights are initialized with very large values, the gradients can also become large.
- This can cause instability right from the beginning of training.

3. What is the significance of Deep Recurrent Networks? CO3 Und

Capturing Long-Term Dependencies

Unlike shallow recurrent models, deep recurrent networks stack multiple recurrent layers, allowing them to capture more complex

patterns over extended sequences. This is crucial in applications like speech recognition, where understanding a word depends on context from many previous words.

2. Improved Representation Learning

By introducing depth in recurrent architectures, DRNs can learn hierarchical features, which improve performance in tasks like machine translation, text generation, and video analysis.

4. What is the role of pooling in deep learning? CO3 Rem

Pooling is a critical operation in deep learning, particularly in Convolutional Neural Networks (CNNs), where it helps improve efficiency, reduce computational complexity, and enhance feature extraction. Pooling layers downsample feature maps, retaining essential information while reducing spatial dimensions.

5. Illustrate how filter is applied in CNN with an Example. CO3 Ana

Filters (also called kernels) in CNNs are small matrices that slide over an image to extract features like edges, textures, and patterns. The process of applying a filter to an image is called **convolution**.

PART–B (13 + 13+ 14 = 40 Marks)

6. (a) How do Recurrent Neural Networks (RNNs) function, and what are their architectural components? Provide a detailed analysis with a relevant diagram. 13 CO2 Ana

1. Functioning of RNNs

Step 1: Input Processing

At each time step ttt, an input xtx_txt is fed into the network. This input can be a word in a sentence (for NLP) or a frame in a video (for computer vision).

Step 2: Hidden State Computation

The network maintains a hidden state hth_tht, which serves as memory. This state is updated at each time step using the current input and the previous hidden state:

ht=f(Whht−1+Wxxt+b)h_t = f(W_h h_{t-1} + W_x x_t + b)ht=f(Whht−1+Wxxt+b)

where:

- WhW_hWh and WxW_xWx are weight matrices for the hidden state and input, respectively.

- bbb is the bias term.
- fff is an activation function (commonly tanh or ReLU).
- ht−1h_{t-1}ht−1 is the previous hidden state, maintaining past context.

Step 3: Output Generation

The network produces an output yty_tyt at each time step, typically computed as:

yt=g(Wyht+c)y_t = g(W_y h_t + c)yt=g(Wyht+c)

where:

- WyW_yWy is the output weight matrix.
- ccc is the output bias.
- ggg is often a softmax function (for classification tasks).

(or)

(b) What are the derivative and mathematical equations of the Sigmoid and Rectified Linear Unit (ReLU) activation functions? Provide a detailed analysis.

. Sigmoid Activation Function

Mathematical Equation:

The **Sigmoid function** (also called the logistic function) is given by:

σ(x)=11+e−x\sigma(x) = \frac{1}{1 + e^{-x}}σ(x)=1+e−x1

where:

- xxx is the input to the neuron.
- e−xe^{-x}e−x ensures smooth and bounded output between **(0,1)**.

13 CO2 Ana

Derivative of Sigmoid:

To compute the derivative of the sigmoid function:

dσ(x)dx=σ(x)·(1−σ(x))\frac{d\sigma(x)}{dx} = \sigma(x) \cdot (1 - \sigma(x))dxdσ(x)=σ(x)·(1−σ(x))

Explanation of Derivative:

- The derivative is **maximum at x=0x = 0x=0** and equals **0.250.250.25**.
- For very large or very small values of xxx, the gradient **approaches zero**, leading to the **vanishing gradient problem**.

- This is problematic in deep networks because it slows down learning.

Graph of Sigmoid & Its Derivative

- The sigmoid function smoothly maps input values to a range of (0,1).
- The derivative is highest at x=0x=0x=0 and flattens at extreme values.

2. ReLU (Rectified Linear Unit) Activation Function

Mathematical Equation:

The **ReLU function** is defined as:

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

which means:

- If x>0x > 0x>0, then f(x)=xf(x) = xf(x)=x.
- If x≤0x \leq 0x≤0, then f(x)=0f(x) = 0f(x)=0.

Derivative of ReLU:

The derivative of ReLU is:

f′(x)={1,x>00,x≤0f'(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}f′(x)={1,0,x>0x≤0

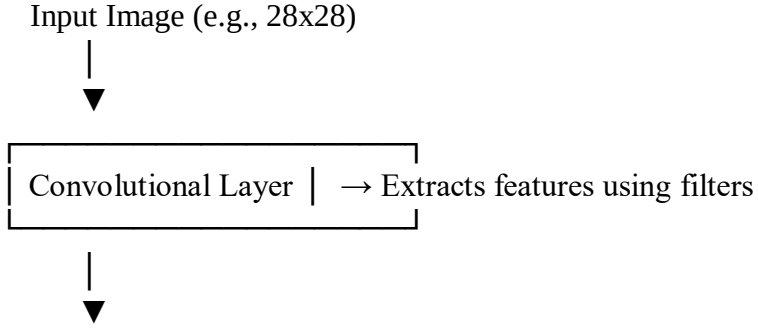
Explanation of Derivative:

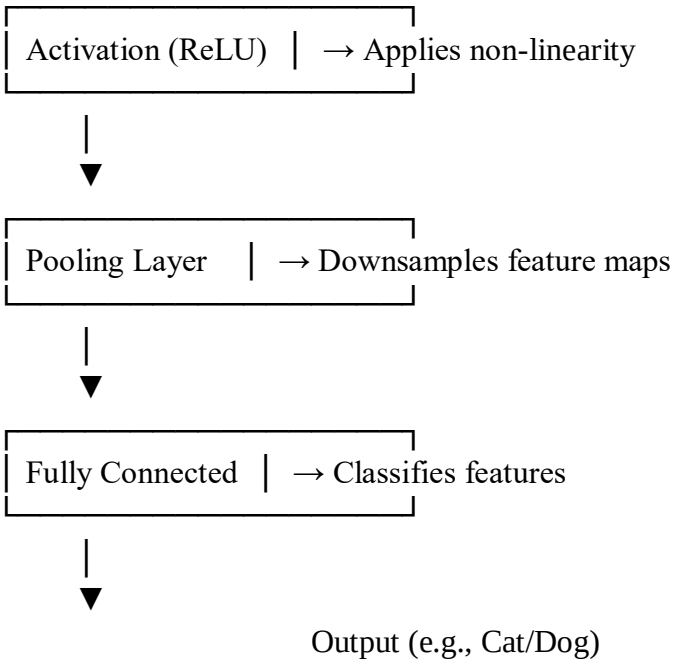
- For **positive values**, the gradient is **1**, which helps avoid the vanishing gradient problem.
- For **negative values**, the gradient is **0**, meaning neurons that receive negative inputs do not update during backpropagation. This issue is called the **dying ReLU problem**.

Graph of ReLU & Its Derivative

- The function is **linear for positive values** and **zero for negative values**.
- The derivative is either **1 (for x>0x > 0x>0)** or **0 (for x≤0x \leq 0x≤0)**.

7. (a) What is a Convolutional Neural Network (CNN), and how do its layers function? Provide a detailed explanation along with a clear diagram. — 13 — CO3 — Rem





Architecture of a CNN

- A CNN consists of multiple types of layers, each with a specific function. The key layers in a CNN include:
- 1. Convolutional Layer (Feature Extraction)
- The core building block of a CNN is the convolution operation, where a filter (kernel) slides over the input image to extract features like edges and textures.
- Mathematically, a convolution is performed as:
- $Y(i,j)=\sum m\sum nX(i+m,j+n)\cdot W(m,n)Y(i, j) = \sum_{m} \sum_{n} X(i+m, j+n) \cdot W(m, n)Y(i,j)=m\sum n\sum X(i+m,j+n)\cdot W(m,n)$
- where:
- XXX is the input image matrix,
- WWW is the filter (kernel),
- YYY is the output feature map.
- Purpose: Detects low-level features (edges, textures) in early layers and complex patterns (shapes, objects) in deeper layers.

(or)

7. (b) Examine how the Recursive Neural Network is used for sentiment analysis in natural language sentences. Discuss about it using a real-world example.

Recursive Neural Network (RecNN) for Sentiment Analysis

A **Recursive Neural Network (RecNN)** is a type of deep learning model that processes data hierarchically by applying the same neural network structure recursively over a **tree-like structure**.

1. How RecNN Works for Sentiment Analysis

- hierarchical structures like **parse trees** of sentences.
- Words and phrases are recursively combined based on grammatical structure to form **higher-level**
- **representations** (phrases → sentences).
- These representations are used for **sentiment prediction**.
- [Root: Neutral]
- / \
- [Positive] [Negative]
- / \ / \
- "The" "movie" "was" "visually"
- \
- "stunning"
- \
- "but lacked emotional depth"

13 CO2 App

Mathematical Formulation:

Given a **binary parse tree**, each node in the tree is computed as:

hp=f(W·[hl,hr]+b)h_p = f(W \cdot [h_l, h_r] + b)hp =f(W·[hl,hr]+b) where:

- hl,hrh_l, h_rhl,hr = hidden representations of left and right child nodes.
- WWW = weight matrix.
- bbb = bias term.
- fff = non-linear activation function (e.g., **tanh** or **ReLU**).

8. (a) Imagine you are training a deep learning model for image recognition, but despite multiple training attempts, the model gets stuck in suboptimal performance. The loss function decreases initially but then plateaus at a higher value than expected, indicating the model might be trapped in a **local minimum** instead of reaching the optimal **global minimum**.

Can you provide a real-world scenario where overcoming local minima significantly improved model performance?

Real-World Scenario: Overcoming Local Minima in Image Recognition

Scenario: Self-Driving Car's Pedestrian Detection System

A major self-driving car company was developing a **pedestrian detection model** using a deep **Convolutional Neural Network (CNN)**. The goal was to accurately classify pedestrians in various environments (day, night, urban, rural). However, despite extensive training, the model consistently reached a **suboptimal accuracy plateau** (~85%) and failed to improve further.

14 CO2 APP

Identifying the Problem: Local Minima in Loss Function

- **Observations:**
  - The loss function **decreased initially** but plateaued at a high value.
  - The model **struggled with low-light pedestrian detection**, missing crucial features.
  - Training on complex backgrounds resulted in **high false negatives** (pedestrians misclassified as background).
- **Possible Causes of Local Minima:**
  - **Poor Weight Initialization:** The initial parameters biased the network toward certain features.
  - **Limited Data Augmentation:** The model overfitted to well-lit pedestrian images.
  - **Suboptimal Learning Rate:** The optimizer

failed to escape a shallow local minimum.

(OR)

(b) Imagine you are developing a speech-to-text application. Initially, you use a **Feedforward Neural Network (FNN)** to process audio signals and convert them into text. However, you notice that the model struggles with long sentences because it treats each sound independently and does not retain context.

To improve performance, you switch to a **Recurrent Neural Network (RNN)**, which remembers previous sounds and words, allowing it to understand the relationship between them. Now, when processing a sentence like *"I am going to the market"*, the RNN retains the words it has already processed, ensuring more accurate predictions. How does the ability of an RNN to retain past information help improve speech-to-text accuracy compared to an FNN?

How RNNs Improve Speech-to-Text Accuracy

1. Capturing Dependencies Between Words

- In an **FNN**, the model processes the phrase **"going to the market"** as individual words without linking them.
- An **RNN**, however, remembers previous words, allowing it to correctly predict that **"market"** is the likely next word in the sentence **"I am going to the ..."**

14 CO3 App

2. Handling Accents and Variations in Pronunciation

- FNNs struggle with variations in pronunciation because they process phonemes in isolation.
- RNNs **learn temporal relationships**, allowing them to predict words correctly even if pronunciations vary slightly.

3. Disambiguating Similar Sounds Based on Context

- Example: The phrase **"recognize speech"** vs. **"wreck a nice beach"** sounds similar.
- An **FNN** might confuse these phrases because it

doesn’t retain prior words.

- An **RNN** understands the context and correctly predicts **"recognize speech"** in a professional setting.

Bloom's Taxonomy:

REM – Remember **UND** – Understand **APP**– Apply **ANA**–Analyse

Faculty Incharge **Teaching Coordinator** **HoD/Dean**