Instructions:

- Answer all questions **clearly and concisely.**
- Support answers with diagrams, equations, or examples where necessary.
- Cite sources (especially book chapters mentioned) where applicable.
- Write in your **own words** — avoid copying from the internet.
- Submit your completed assignment in **Github Repo Link (PDF+Code Notebook)**

Section A: Fundamentals of Deep Learning

Q1. What is Deep Learning?

Explain the concept of deep learning.

- How is it different from traditional machine learning?

  ✎ **Answer:** The traditional machine learning compels the manual inputs from the user while deep learning makes it completely automatic. Manual feature extraction and singular algorithms makes it more simple.

- Why is it called "deep"?

  ✎ **Answer:** The deep in the deep learning doesn't mean that we have to go to deep inside the model rather it means that how deep we have those hidden layers.

Q2. Key Components of Deep Learning

List and describe the **main components** of a deep learning model (e.g., neural networks deep neural networks (DNNs), layers, activation functions, loss functions, optimizers).

✎ **Answer:**

Neural network : It is the combination of neurons linked with each other, each neuron has weight and bias. Input from each neuron is then multiplied by weight and then bias is added to it.

Layers: Each layer have multiple neurons. Each layer can contain 1 to many neurons.

Activation function: Activation function is mathematical formula which introduces the non linearity in the data. Each neuron in the hidden layer sends the output inside the activation function most common are Relu, sigmoid, linear etc.

Loss function: Loss function calculates how correct our neural network is. The loss is then calculated and model uses it to optimise most common loss function are; crossentropy, sigmoid etc

Optimizers: Those are functions which do backtracking to make the model correct based on the loss function, most common is Adam, gradient descent

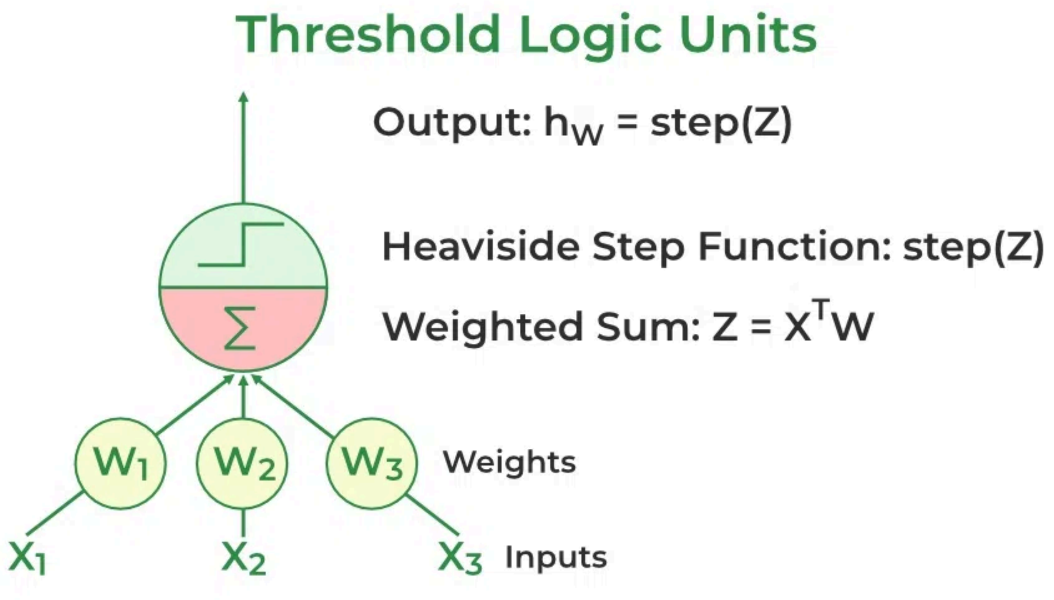
Q3. Understanding Neural Networks, Neurons, and the Perceptron

Explain the structure and working of:

- A **neuron** in a neural network

  ✎ **Answer:** A neuron is a function which contains weight and bias along with activation function. It is a building block of hidden layers, also called a hidden unit.

- The **perceptron model** (include formula and diagram)



- How multiple perceptrons form a **neural network**
  ✎ **Answer:** To make a neural network we connect multiple perceptrons, Each perceptron receives the output from every single perceptron and then individual output is then input for the next perceptron. Each neuron has its own weights and a bias.

Q4. Hierarchical Representations

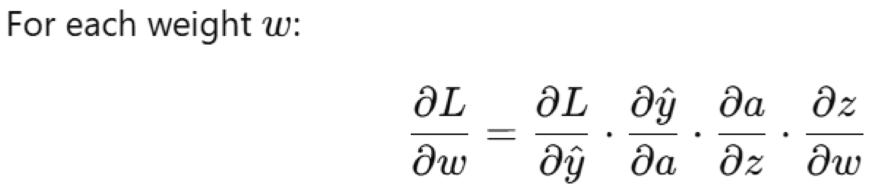
What are hierarchical representations in deep learning?
Explain how features evolve from low-level (edges, lines) to high-level (objects, faces).

✎ **Answer:** In a neural network model learns from generic to specific. The layer which is closer to the input has a generic understanding of the dataset. The more we move towards the last layer the more neurons get specific to certain details.

Q5. Fitting Parameters using Backpropagation

Describe the backpropagation algorithm.
How does it adjust network parameters during training?

✎ **Answer:** In the forward pass, when the neural network multiples its weight and bias it provides the predicted output after each epoch. Then model predicts the difference of predicted and actual output. The optimizer then calculates the partial derivative of the loss and then teaches each neuron what if this particular weight fet decreases will my prediction become accurate.



For each weight w:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w}$$

Q6. Non-Convex Functions

What are non-convex functions, and why do they make optimization challenging in deep learning?

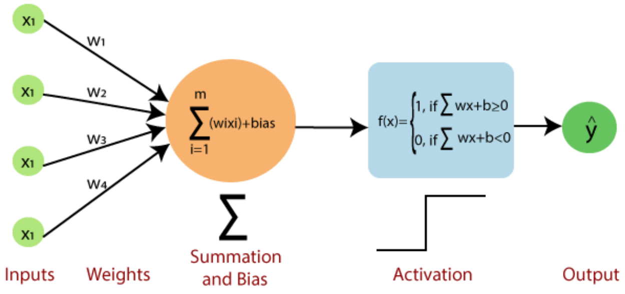
✎ **Answer:** The non-Convex functions are the biggest reasons to use the deep learning models. Those functions add non linearity to the models. Non- Convex functions are tanh sigmoid, softmax those activation functions add non-linearity. Due to those models optimizers face following :

1. Optimizers stuck to the local minimum
2. The can stuck into the plateau
3. The sophisticated gradient descent can tackle this by adding the randomizing and generalizing.

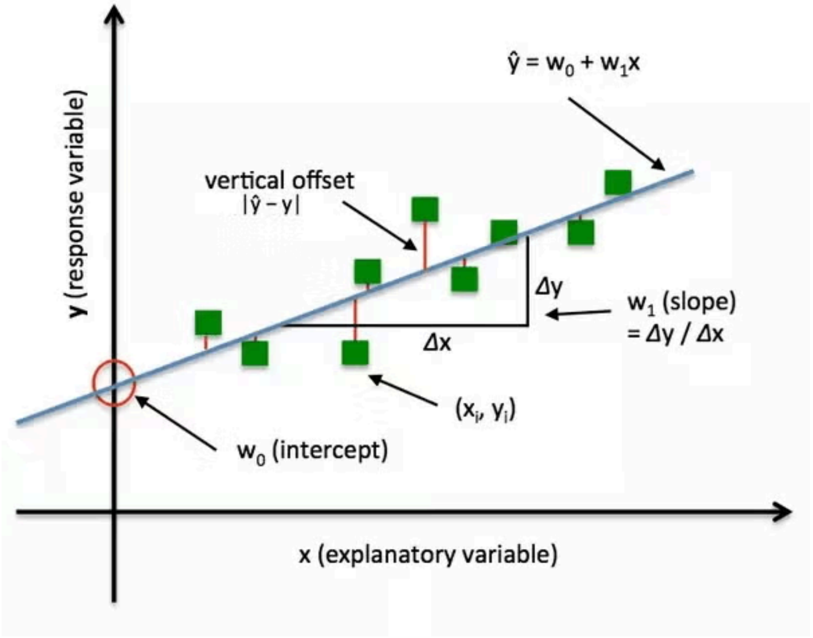
Q7. Training and Model Optimization

Explain the **training process** of a neural network (forward pass, loss calculation, backward pass, parameter update).
Include techniques used for model optimization (e.g., dropout,SGD, learning rate scheduling, batch normalization).
✎ **Answer:** Training of neural network involves 4 steps:

1. Forward pass: Each layer transforms input data by applying
    a. A = w[t]x + b
    b. Z = sigmoid (A)


x_1
x_1
x_1
x_1
w_1
w_2
w_3
w_4
$\sum_{i=1}^{m}$(wixi)+bias
$f(x)=\begin{cases}1, \text{ if } \sum wx+b\geq0 \\ 0, \text{ if } \sum wx+b<0\end{cases}$
$\hat{y}$
$\sum$
Inputs
Weights
Summation and Bias
Activation
Output

2. Loss Calculation: The model compares the predicted output with the actual output with few of the algorithms are:
    a. L = 1/ n Sum(y_pred - y_actual)^2
    b. L = -1/n Sum(y_actual log(y_pred)



3. Backward pass: The gradients are calculated and then weights are updated some of the gradients are

$$\frac{\partial \mathcal{L}}{\partial W^{[l]}}, \quad \frac{\partial \mathcal{L}}{\partial b^{[l]}}$$

4. Update: Weights are updated accordion to the calculations

$$W^{[l]} \leftarrow W^{[l]} - \eta \frac{\partial \mathcal{L}}{\partial W^{[l]}}$$

$$b^{[l]} \leftarrow b^{[l]} - \eta \frac{\partial \mathcal{L}}{\partial b^{[l]}}$$

Q8. Challenges and Requirements

List the **main challenges** in deep learning (e.g., data requirements, interpretability, model complexity, computational cost).

What are the essential **requirements** for building effective deep learning models?
✎ **Answer:**

1. Deep-learning require a lot of data even transfer learning need thousands of images in order to perform well
2. Deep learning have millions of parameters to understand in order to know the background of the data we have to understand the model summary which is the uphill task
3. Model complexity is also an issue. We have to understand which layer to add after the sequence and also we have to handle millions of parameters. Models can overfit if we don't care about the regularization.
4. Cost for the computation takes a lot of money we have to prepare the gpu for the models to make it quick.

Section B: Deep Learning Frameworks & Implementation

Q9. Deep Learning Frameworks

List at least **three deep learning frameworks** (e.g., TensorFlow, PyTorch, Keras).
Describe their main features and why they are used.
✎ **Answer:**

1. Pytorch**:** Pytorch is used for the research purpose we can handle the parameters and provides the extra control over it
2. Tensorflow: It is much fast and low code as compare to pytorch it is used for the development purpose it is more automatic then the pytorch
3. Keras is a part of tensorflow which is used for the sequential model creations and give flexibility to change / add / update models with its user friendly syntax

Q10. Building Neural Networks with Keras and TensorFlow (Book Reference: Chapter 3)

Explain how neural networks can be built using **Keras** and **TensorFlow**.
Include key steps such as:

- Defining layers

- Compiling the model

- Training and evaluating performance
  ✎ **Answer:**

```
model = sequential ([

        Dense(64, activation = ‘inear’, input_shape(4, 0)

        Dense(64, activation = ‘inear’,),

        Dense(64, activation = ‘relu’)

])

model.compile(optimizer = ‘adam’, loss = ‘binary_crossentropy’’, ‘metrics = [‘’accuracy’]

hist = model.fit(X_train, y_train, epochs = 20, batch_Size = 32, validation_split= 0.2)
```

Q11. Data Preprocessing, Feature Engineering, and Feature Learning (Book Reference: Chapter 4)

Define the following in the context of neural networks:

- **Data preprocessing**

- **Feature engineering**

- **Feature learning**
  Explain their importance in preparing data for deep learning models.
  ✎ **Answer:**

1. Data Preprocessing involves:

   a. Vectorizing : Only vectorized data is processed in the neural network
   b. Missing values imputation : Missing values are imputed to make the homogenous
   c. Normalization: Make it scale 0-1. So that model should learn from scratch
   d. Homogenous data:

2. Feature Engineering:

   a. Feature engineering involves manual or hard-coding the features and knowing which one is correct. It includes the correlation maps to select the best ones

3. Feature Learning:

a. Next layers learn batter then the previous layers
b. Automatic process means model learns all the features in only one pass
c. Joint Learning, all the layers learn at the same time
d. Representation, early layers are generic and next layers are specific

---

Section C: Image Classification Concepts

Q12. What is Image Classification?

Define image classification.
Give two examples where image classification is applied in real-world scenarios.
✎ **Answer:**

1. Medicine: AI and image processing for disease prediction
2. Cars: Object detection in cars to prevent accidents

Q13. Introduction to ImageNet

What is the ImageNet dataset, and why is it important for deep learning research?
Mention the **ImageNet Challenge** and how it transformed computer vision.

✎ **Answer:** Imagenet is an already trained model with millions of images which have its own weights. In deep learning models we use transfer learning by removing the last layer and adding the desired one of our own. It belongs to 1k categories. When we don't have enough dataset we use transfer learning for the best results

Q14. Classification using a Single Linear Threshold (Perceptron)

Explain how a **single-layer perceptron** works for binary classification.
Include the mathematical formula and how it decides the output.

✎ **Answer:** A single layer perceptron is enough for the binary classification because it gives output in range of 0-1 in the case of sigmoid if it is <0.5 then it is class A otherwise it is class B. In the multiclass classification we use sigmoid.

Q15. How Interpretable Are Deep Learning Features?

Why are deep learning models considered “black boxes”?
Describe one method (e.g., Grad-CAM, feature visualization) used to interpret model features.

✎ **Answer:** The deep learning model can learn very accurately. It have millions of parameters when the model goes overfitting or underfitting we dont really know which feature and which parameter is responsible for that therefore it is called blackbox.

1. Grad-CAM: Used for CNN, Computes gradients for the target class with respect to feature maps in last convolution layer. It calculates the target class and maps with the image. It is used in masking.

Q16. Manipulating Deep Nets

Explain what **adversarial examples** are and how they can trick deep learning models.
Suggest one way to make neural networks more robust against such attacks.

✎ **Answer:** The slightly modified and altered data to fool the system to make it more accurate. Slightly noise in the dataset model thinks its the opposite class therefore predicts wrong. I.e slightly modified cat image is not dog.

1. Way to make it robust: Train the model with the real data and adversarial data. Like adding some noise in the real dataset and checking the results if it is matching or not.

Q17. Transfer Learning

What is **transfer learning**, and how does it help when data is limited?
Mention two pre-trained models commonly used for transfer learning.

✎ **Answer:** Training on the dataset with already trained models is called transfer learning. Transfer learning makes the model more accurate by removing the last layer and introducing its own. According to the goals. i.e. softmax for the multiclass sigmoid for the binary with only one neuron

---

Section D: Applications of Deep Learning

Q18. Applications in Data Science

Explain at least **three applications of deep learning in data science** (e.g., Speech Recognition, Image Classification, NLP, Predictive Analytics, Computer vision, Healthcare).

✎ **Answer:**

1. Speech Recognition: Pilot cockpit opens only when the air hostess speaks
2. Disease classification and surgery
3. Stock market analysis
4. Self driving cars
5. Robotic surgeries i.e cornea transplant

Q19. Case Study 1: Data Scientist Employee Attrition

Read the scenario and answer:

A company wants to predict whether a **data scientist** will leave the organization.
They collect data such as age, salary, experience, satisfaction, and commute distance.

a. What kind of problem is this (classification or regression)?
b. Suggest an appropriate deep learning model architecture.
c. What type of loss function should be used?
d. How can the results help HR management?
✎ **Answer:**

1. It is the classification problem, more specifically it is a binary one**.**
2. The appropriate model would be a neural network with 32 neurons optimally
3. The binary cross entropy would be best
4. The results will be determined by the accuracy metrics

Section E: Practical Project

Q20. Project – Handwritten Digit Classification (MNIST Dataset)

Objective:
Build an **Artificial Neural Network (ANN)** using **Keras** and **TensorFlow** to classify handwritten digits (0–9) from the **MNIST dataset**.

Requirements:

- Load and preprocess the MNIST dataset.
- Design a suitable **ANN architecture**.
- Train and evaluate the model for accuracy.
- Visualize the results (confusion matrix, accuracy/loss curves).

Deliverables:

- Model summary and architecture diagram
- Accuracy and loss graphs
- Sample predictions (images + predicted labels)
- Short explanation of results

✎ **Answer / Attach Code Link:**

Section F: Reflection (Bonus)

Q23. Your Thoughts on Deep Learning

In your opinion:

- What makes deep learning powerful?
- What ethical or practical issues should be considered in its real-world use?

Answer: In machine learning we only train one neuron (regression). But in case of deep learning we use multiple regressions (casually stated), making it more non linear. In deep learning we can break down the problem and give it to smaller problem solvers (perceptrons). Make them more efficient. Instead a full model focuses on complex problems. We break down model and each has its own learning and prediction