(B.TECH) Semester-VII AY 2023-24

DL Lab Assignment No. 07

Student Name: Rohit Saini_____ PRN No.: 1032200897_____

Date: (Perform) 25-11-23 Faculty: Prof. Anita Gunjal

Problem Statement: Implement a Convolution Neural Network (CNN) for image dataset using Python.

Objectives:

1. To understand the architecture and working of Convolution Neural Network
2. To implement the Convolution Neural Network

Theory

Applications of Convolutional Neural Networks (CNN):

1. **Image Classification:** CNNs excel in classifying objects within images. They've been pivotal in competitions like ImageNet, achieving human-level performance.
2. **Object Detection:** CNNs can identify and locate multiple objects within an image, making them fundamental in applications like autonomous vehicles and surveillance systems.
3. **Facial Recognition:** CNNs are used to recognize and verify faces, contributing to applications in security, authentication, and entertainment.
4. **Medical Image Analysis:** CNNs assist in detecting and diagnosing medical conditions from images such as X-rays, MRIs, and CT scans.
5. **Natural Language Processing (NLP):** In NLP, CNNs are employed for tasks like text classification and sentiment analysis.
6. **Autonomous Vehicles:** CNNs play a crucial role in detecting and recognizing objects for autonomous driving systems, ensuring safety.
7. **Style Transfer:** CNNs are utilized in artistic applications, transforming images into different artistic styles, contributing to the field of computational creativity.
8. **Video Analysis:** CNNs can be applied to analyze and understand video content, enabling applications like video summarization and action recognition.
9. **Image Generation:** CNNs are part of architectures that generate realistic images, contributing to the field of generative adversarial networks (GANs).
10. **Augmented Reality:** CNNs enhance the capabilities of augmented reality applications by accurately recognizing and tracking objects in real-time.

Architecture of Convolutional Neural Networks (CNN):

1. **Convolutional Layers:** These layers apply convolutional operations to capture local patterns in the input data. Filters are learned to detect features like edges, textures, and more complex structures.
2. **Pooling Layers:** Pooling layers reduce the spatial dimensions of the input data, capturing the most important information while reducing computational complexity.
3. **Fully Connected Layers:** After extracting features, fully connected layers make predictions based on the learned representations. These layers are often present towards the end of the network.
4. **Activation Functions:** Non-linear activation functions (e.g., ReLU) introduce non-linearity to the model, enabling it to learn complex relationships.
5. **Batch Normalization:** Batch normalization normalizes the input of a layer, reducing internal covariate shift and accelerating training.
6. **Dropout:** Dropout is a regularization technique used to prevent overfitting by randomly dropping a fraction of connections during training.

Description about Image Dataset Used (e.g., MNIST dataset):

MNIST (Modified National Institute of Standards and Technology) Dataset:

Description: MNIST is a widely used dataset in the machine learning community, consisting of 28x28 grayscale images of handwritten digits (0-9).

Size: 60,000 training images and 10,000 testing images.
Application: Primarily used for digit recognition tasks, serving as a benchmark dataset for evaluating the performance of various machine learning algorithms.
Usage: Often employed for training and testing the efficacy of CNNs due to its simplicity and accessibility.
Challenges: While MNIST is straightforward, it helps in understanding the basic principles of CNNs and serves as a starting point for more complex image datasets.

Operations to be performed:

1. Import the required Python libraries and dataset
2. Normalizing dataset (reshaping training and testing images)
3. Build the model
   Identifying model requirement (number of convolution layers pooling layers, fully connected layers, size of filters/kernel used, activation function used)
4. Compile the model
   Optimize the model to adjust the weights to minimize the losses (which optimizer to be used)
   Identifying optimizer and loss function.
5. Train the model with Image dataset
6. Predict/Test the model
7. Model performance visualization in terms of accuracy and loss

Program code: (paste your program code)

Output: (paste output screen & graphs plotted)

FAQs:

1. What is the difference between a regular neural network and Convolutional neural network?
2. What is the need of striding and padding in a Convolutional neural network?
3. Explain any two regularization techniques used in deep neural networks to avoid overfitting problems of training models?

PC41 Rohit Saini
1032200897

Page No.
Date

DL Lab 6

FAQ:

Q1. What is the difference between a regular neural network and convolutional neural networks?

Ans: In regular neural network, each neuron in one layer is connected to every neuron in the adjacent layers. This results in a large number of parameters, which can make it challenging to process structured data like images efficiently.

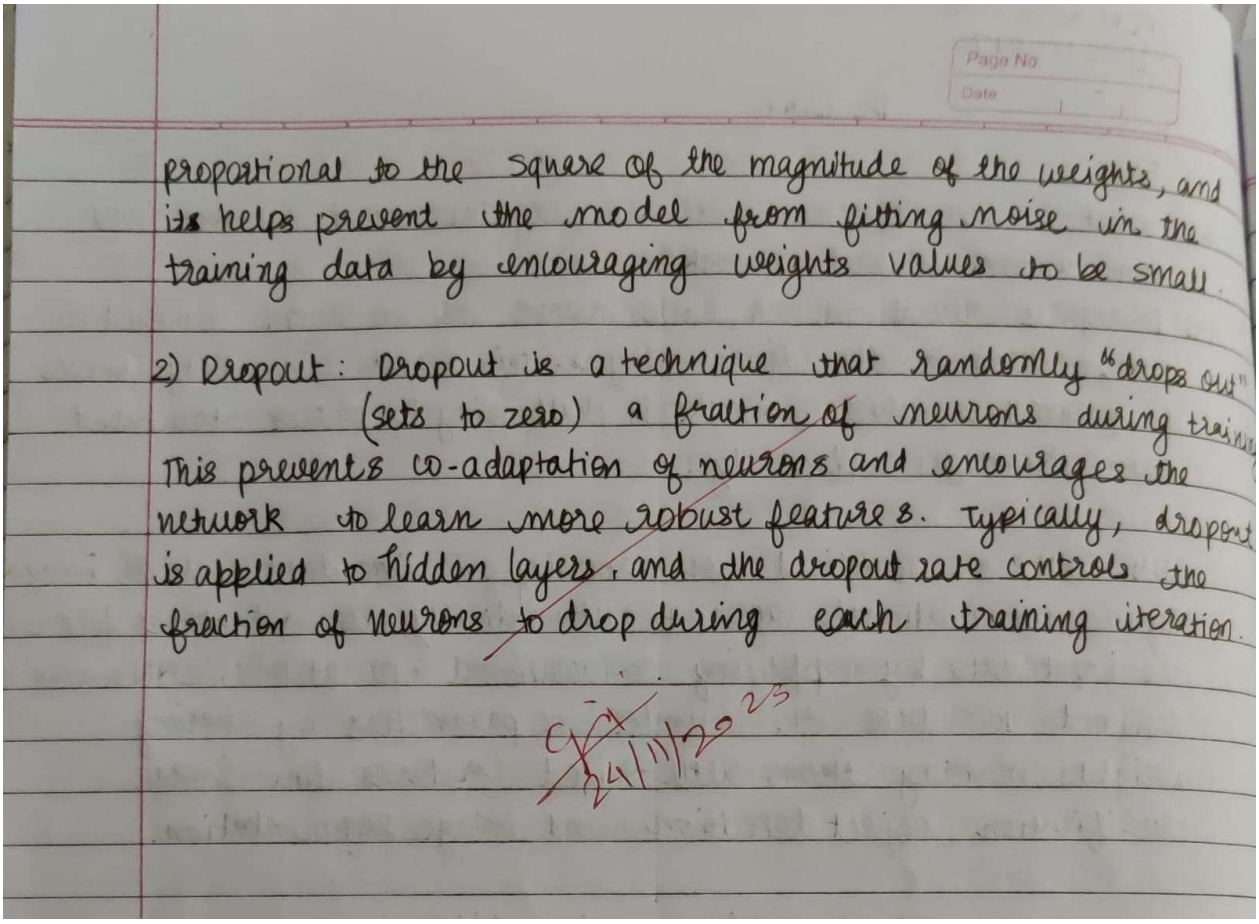
CNN: CNN's are specialized for processing grid like data such as images. They use convolutional layers to automatically learn features from the input data by applying convolutional operations. CNNs are designed to reduce the number of parameters by sharing weights, making them well-suited for tasks like image classification, object detection and image segmentation.

Q2. What is the need of striding and padding in a convolutional neural network?

Ans Striding can help downsample feature maps and reduce computational complexity. It is useful for controlling the spatial resolution of the output.
Padding is important to preserve the spatial dimensions when performing convolution, especially when maintaining spatial ~~dimensions when performing convolution~~ information is critical for the task. Padding can also help prevent information loss at the image edges.

Q3. Explain any two regularization techniques used in deep neural network to avoid overfitting problems of training models.

Ans: 1) L2 Regularization (Weight Decay): L2 regularization adds a penalty term to the loss function that discourages large weights in the network. This term is



Conclusion:

The architecture of Convolutional Neural Network was studied and the implementation of CNN was performed successfully.