Machine Learning

Lab 6 : CNN

Name :Mohammed Musharraf

SRN : PES2UG23CS915

SECTION: F

1. Introduction

The objective of this lab was to design, train, and evaluate a Convolutional Neural Network (CNN) to classify rock, paper, and scissors images. Using a dataset containing images of hand gestures representing the three classes, we built a custom CNN architecture and trained it using PyTorch. The model's performance was tested based on its ability to accurately predict the gesture shown in new images.

2. Model Architecture

Our CNN model consists of **three convolutional blocks** followed by a **fully connected classifier**. Here's a breakdown of the architecture:

Convolutional Block Details Each

block consists of:

- Conv2d Layer: Convolution with a 3×3 kernel and padding of 1.
- ReLU Activation.
- MaxPool2d: Reduces spatial dimensions by a factor of 2.

Block Input Channels Output Channels Kernel Size Padding Pooling

1 3 16 3×3 1 MaxPool2d(2×2)

2 16 32 3×3 1 MaxPool2d(2×2) 3 32 64 3×3 1

MaxPool2d(2×2)

After 3 pooling layers, the input image of size 128×128 is reduced to 16×16, with 64 channels.

Fully Connected Classifier

This block is designed as follows:

- Flatten: Converts 64×16×16 feature map to 16384 features.

- Linear Layer: 16384 → 256
- ReLU Activation
- Dropout(p=0.3)
- Linear Layer: 256 → 3 (representing rock, paper, scissors classes)

3. Training and Performance Key

Hyperparameters :

- **Optimizer**: Adam
- **Loss Function**: CrossEntropyLoss
- **Learning Rate**: 0.001
- **Epochs**: 10
- **Batch Size**: 32 Final Test Accuracy :
- The model achieved a final test accuracy of **98.17 %**

4. Conclusion and Analysis

The CNN performed well on the rock-paper-scissors classification task, achieving a test accuracy of **98.17 %**. The architecture was simple yet effective, demonstrating the power of convolutional layers for image-based classification.

Challenges Faced:

- Ensuring proper pre-processing and normalization of the images.
- Avoiding overfitting given the limited dataset size.

Suggestions for Improvement:

- Increase model depth or complexity (e.g., more convolutional layers).
- Apply data augmentation to increase generalization.
- Use techniques like learning rate scheduling or early stopping.