Comparison of CNN Architectures on Different Datasets

Skills Takeaway From This Project

In this project, learners will gain skills in:
- Understanding and implementing various Convolutional Neural Network (CNN) architectures
- Applying CNNs to different types of datasets
- Evaluating model performance using various metrics
- Analyzing the impact of dataset characteristics on model performance
- Utilizing popular deep learning frameworks such as PyTorch and TensorFlow

Domain

Machine Learning, Deep Learning, Computer Vision

Problem Statement

The goal of this project is to compare the performance of different CNN architectures on various datasets. Specifically, we will evaluate LeNet-5, AlexNet, GoogLeNet, VGGNet, ResNet, Xception, and SENet on MNIST, FMNIST, and CIFAR-10 datasets. The comparison will be based on metrics such as loss curves, accuracy, precision, recall, and F1-score.

Business Use Cases

The insights from this project can be applied in various business scenarios, including:
- Choosing the appropriate CNN architecture for specific computer vision tasks
- Improving model performance by understanding the impact of dataset characteristics
- Optimizing resource allocation by selecting models that offer the best trade-off between performance and computational cost

Approach

1. Load and preprocess the datasets (MNIST, FMNIST, CIFAR-10).
2. Implement the following CNN architectures: LeNet-5, AlexNet, GoogLeNet, VGGNet, ResNet, Xception, and SENet.
3. Train each model on each dataset, recording the loss and accuracy metrics.
4. Evaluate the performance of each model on the test sets using accuracy, precision, recall, and F1-score.
5. Plot the loss curves and other performance metrics for comparison.
6. Analyze the results to understand the impact of different architectures and datasets on model performance.

Results

The expected outcomes of this project include:
- Comparative loss curves for each model on each dataset
- Accuracy, precision, recall, and F1-score for each model on each dataset
- Analysis of the results to determine the strengths and weaknesses of each architecture on different datasets

Project Evaluation Metrics

The success and effectiveness of the project will be evaluated using the following metrics:
- Accuracy: The proportion of correct predictions out of the total predictions made.
- Precision: The proportion of true positive predictions out of all positive predictions made.
- Recall: The proportion of true positive predictions out of all actual positives.
- F1-score: The harmonic mean of precision and recall.
- Loss: The value of the loss function during training and testing.

Technical Tags

CNN, Deep Learning, PyTorch, TensorFlow, Computer Vision, Image Classification, Model Evaluation

Data Set

The datasets used in this project are:
- MNIST: Handwritten digits dataset consisting of 60,000 training images and 10,000 testing images. Each image is 28x28 pixels in grayscale.
- FMNIST: Fashion MNIST dataset consisting of 60,000 training images and 10,000 testing images of fashion products. Each image is 28x28 pixels in grayscale.
- CIFAR-10: Dataset consisting of 60,000 32x32 color images in 10 classes, with 50,000 training images and 10,000 testing images.

Data Set Explanation

The datasets are chosen to cover a variety of image classification tasks:
- MNIST and FMNIST provide simpler tasks with grayscale images, allowing for the evaluation of basic image recognition capabilities.
- CIFAR-10 offers a more complex task with color images, testing the models ’abilities to handle more detailed and varied data.

Project Deliverables

Learners need to submit the following upon project completion:
- Source code for implementing and training the models
- Documentation detailing the approach, results, and analysis
- Plots of loss curves and performance metrics

- Final report summarizing the findings and conclusions

Project Guidelines

Follow these guidelines and best practices for project development:
- Use version control (e.g., Git) to manage code changes
- Adhere to coding standards and write clean, readable code
- Regularly validate and test the models to ensure they work correctly
- Document the code and approach clearly for ease of understanding