A report on

IMAGE CLASSIFICATION USING FEED FORWARD NEURAL NETWORKS

By:

Aditya Patel (2021A7PS2433P)

Sugeet Sood (2021B4A71746P)

Made as a part of the course
Deep Learning (CS F425)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
(September 2024)

Table of Contents

Introduction

This project is part of the Deep Learning course, focusing on image classification using deep learning techniques. The primary task was to classify images into distinct categories without using traditional convolutional neural networks (CNNs). Instead, the focus was solely on developing a feedforward neural network for this classification task. Task 1 involved exploring various approaches to enhance the classification accuracy of the model by leveraging different feature extraction methods, rather than relying on the inherent spatial learning capabilities of CNNs.

The dataset provided consisted of images from 60 different classes, with each class containing only five examples. The challenge was to accurately classify these images while preventing overfitting due to the limited number of examples per class. Throughout the project, various techniques were employed to preprocess the images, extract meaningful features, and optimize the model's performance. The progression of this project involved starting with a simple neural network, experimenting with different feature extraction techniques like wavelet transforms, and eventually using Histogram of Oriented Gradients (HOG) to achieve the final results.

Dataset Description

The dataset used in this project comprises images of flowers, categorized into 60 distinct classes. Each class contains five examples, making a total of 300 images. One noteworthy aspect of this dataset is the inherent challenge posed by the categorization of certain flowers, which may exhibit a variety of colors but are classified under a single label. For instance, a flower species may appear in shades of red, yellow, or white, yet all color variations are grouped together under the same class.

This categorization presents both an opportunity and a challenge for the classification model. On one hand, the diversity of colors within a single class can provide the model with a broader understanding of the features that define that species. On the other hand, the model must learn to generalize across these color variations, which can introduce complexity and potentially lead to misclassifications if not adequately addressed.

Given the limited number of examples per class, it is essential to leverage effective feature extraction methods to capture the relevant characteristics of the flowers. This necessitates a careful examination of color distribution and texture within the images to ensure that the model is robust enough to handle the variability in appearance while still accurately classifying them into their respective categories. Overall, this dataset serves as an excellent basis for exploring the capabilities of a feedforward neural network in the context of image classification, while also highlighting the importance of effective feature representation

Example Image from the Dataset

Data Augmentation

To enhance the performance of the feedforward neural network and mitigate overfitting, several data augmentation techniques were employed:

- Flipping: Random horizontal and vertical flips of images.
- Zooming: Random zooming in and out of images.
- Translation : Random translation done on the images.
- Random Cropping: Taking random crops of the images.
- Noise Addition: Adding Gaussian noise to some images.

These augmentation methods increased the diversity of the training dataset, allowing the model to generalize better across the various classes.

Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) is a feature extraction technique commonly used in image processing and computer vision, particularly for object detection and classification. The fundamental idea behind HOG is to capture the shape and structure of objects in an image by examining the distribution of gradient orientations.

The HOG algorithm works by dividing an image into small, connected regions called cells. Within each cell, the gradients (changes in intensity) are computed, and a histogram of gradient orientations is created. These histograms provide a representation of the local shape and texture of the object. To improve the robustness of the feature representation, the HOG descriptor can be normalized by grouping adjacent cells into larger blocks, which helps to reduce the influence of local lighting variations.

HOG has several advantages, including its ability to maintain spatial information while being computationally efficient. It is particularly effective for detecting objects with well-defined edges, making it suitable for tasks such as human detection and, in this project, flower classification. By employing HOG features, the model can capture essential characteristics of the flower images, leading to improved classification performance.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique widely used in data preprocessing for machine learning and deep learning applications. The primary objective of PCA is to transform a high-dimensional dataset into a lower-dimensional space while preserving as much variance (information) as possible.

PCA works by identifying the directions (principal components) in which the data varies the most. These principal components are orthogonal to each other and are ranked according to the amount of variance they capture from the original dataset. By projecting the data onto a smaller number of principal components, PCA reduces the dimensionality of the dataset while retaining the essential features.

In the context of this project, PCA was applied after feature extraction to compress the high-dimensional HOG features into a more manageable form. This not only helps speed up the training process but also reduces the risk of overfitting by eliminating redundant features that may not contribute significantly to the model's performance. By leveraging PCA, the model can focus on the most informative aspects of the data, resulting in enhanced classification accuracy and efficiency

Model Architecture

The final model developed for the flower classification task is a feedforward neural network built using the Keras API in TensorFlow. The model architecture is designed to effectively learn and classify images of flowers, comprising several key components:

1. **Input Data Preparation**: The model accepts the processed image data, which consists of PCA-reduced features derived from the HOG extraction method. The training and validation datasets are represented as X_train and X_val, respectively, while the corresponding labels are one-hot encoded to create y_train and y_val for multi-class classification.
2. **Layer Structure**: The architecture of the model consists of multiple dense layers, enabling it to learn complex relationships within the data:
    - **First Dense Layer**: The model begins with a dense layer containing 2048 neurons, activated using the ReLU (Rectified Linear Unit) function. L1 and L2 regularization is applied to mitigate overfitting.
    - **Dropout Layer**: A dropout layer with a rate of 0.5 follows the first dense layer, randomly deactivating half of the neurons to prevent overfitting.
    - **Batch Normalization**: This layer normalizes the activations to stabilize and speed up the training process.

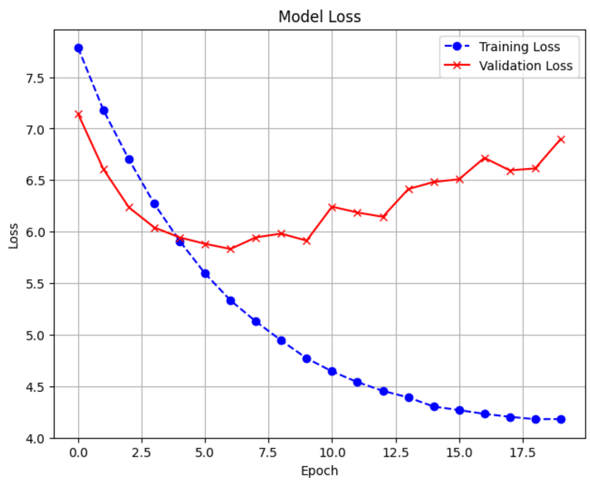
Subsequent layers progressively reduce the number of neurons while retaining the model's ability to learn:

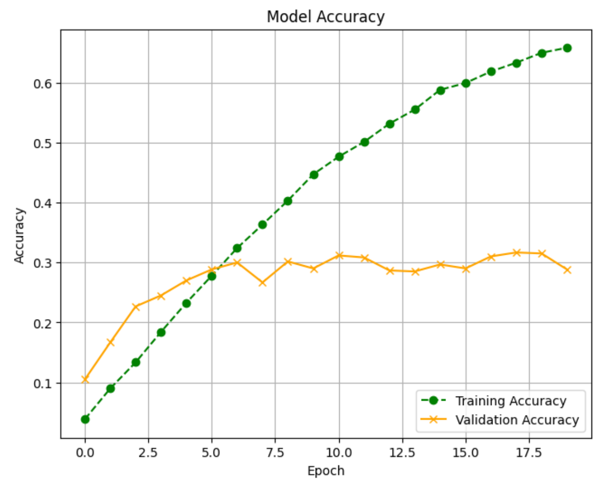
- **Second Dense Layer**: 1024 neurons, followed by a dropout layer (0.4) and batch normalization.
- **Third Dense Layer**: 512 neurons with a dropout rate of 0.3 and batch normalization.
- **Fourth Dense Layer**: 256 neurons with a dropout layer (0.3), which helps maintain essential information as the network reduces in size.
- **Final Dense Layer**: A dense layer with 60 neurons corresponds to the 60 flower classes, utilizing the softmax activation function to produce class probabilities.

3. **Regularization Techniques**: The model incorporates L1 and L2 regularization techniques across the dense layers to enhance generalization and prevent overfitting. This is particularly important given the complex nature of the classification task with multiple classes.
4. **Optimization and Training**: The model is compiled using the Stochastic Gradient Descent (SGD) optimizer, with a learning rate of 0.01 and momentum of 0.9. The categorical cross-entropy loss function is utilized, as it is suitable for multi-class classification tasks. The model undergoes training for 20 epochs, with a batch size of 32, while monitoring validation accuracy.
5. **Model Saving**: Once training is complete, the model is saved in the HDF5 format as 'flower_model.h5', allowing for easy reloading and inference in future applications.

Results

After training the feedforward neural network on the flower classification task, the model achieved an accuracy ranging from approximately 30% to 33% on the test dataset. This level of accuracy reflects the model's ability to distinguish between the different flower species based on the features extracted from the images. Following graphs show the loss and accuracy for the models for training and validation sets.





Area of Improvement

To enhance the performance of the flower classification model, incorporating Convolutional Neural Networks (CNNs) could be a significant improvement. CNNs are specifically designed for image data and excel at automatically learning spatial hierarchies and local features through convolutional layers. By utilizing parameter sharing and pooling layers, CNNs can reduce the number of parameters while capturing essential patterns in the images. This end-to-end learning approach allows for better generalization and accuracy in classification tasks, making CNNs a promising alternative to the current feedforward neural network architecture. Transitioning to a CNN model could lead to improved performance in distinguishing between the various flower species in the dataset.