Project Task 1
Deep Learning (CS F425)

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

September 2024

Team Members

Akshay Shukla	2022A7PS0087P
Ishaan Kale	2022A7PS0084P
Siddhartha Gotur	2022A7PS0070P

Table of Contents

Description of Dataset

The dataset given to us consists of images of flowers belonging to 60 different classes. The train dataset contains 3000 images with 50 images belonging to each of the classes. There is also a separate text file for each of the train and validation datasets which contains the labels corresponding to each of the images. The images are 3 channel images (RGB) and have a dimension of 256x256. The validation dataset has 600 images with 10 images for each class.

Feature Extraction Methods

We explored the different feature extraction methods available for the image classification purpose and decided to use PCA for our purpose. Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space by identifying the directions (principal components) that maximize variance. This method helps to simplify datasets, reduce noise, and visualize complex data while preserving important relationships and structures.

Model Architecture

We used a multilayer perceptron. The description of the architecture is as follows:-
The hidden layers have 256, 128, and 64 neurons each. The input layer uses the dimensions obtained after performing PCA on the training dataset while the output layer gives a result from one of 60 classes. We also added batch normalization and dropout to improve model performance and prevent overfitting. The 256*256*3 flattened image is reduced to just 1024 features after applying PCA, substantially improving the training time process.
The model uses a multi layer perceptron architecture. We use a Relu activation function. ReLU (Rectified Linear Unit) is an activation function defined as $f(x) = \max(0, x)$, which allows only positive values to pass through while setting negative values to zero. It introduces non-linearity into neural networks, helping them learn complex patterns while maintaining computational efficiency due to its simplicity.

Hyperparameters used for Training

We performed the training of the model in two stages. The first 50 epochs were with a learning rate of 0.01 and the next 50 were with a learning rate of 0.001. The other hyperparameters were as follows -
Loss Function - Cross Entropy
Optimizer - Adam
Batch Size - 64

Here is a description of some of the methods we used -

Cross-entropy is a loss function commonly used in classification tasks that measures the dissimilarity between the predicted probability distribution and the true distribution of labels. It quantifies how well the predicted probabilities match the actual class labels, with lower values indicating better model performance.

Adam (Adaptive Moment Estimation) is an optimization algorithm that combines the benefits of two other methods: AdaGrad and RMSProp, by maintaining separate learning rates for each parameter and using both first and second moments of the gradients. It adapts the learning rate for each parameter based on the estimates of first and second moments, enabling faster convergence and improved performance on various types of data.

Final Result after Training

Training Accuracy - 63%
Training Loss - 1.1
Validation Accuracy - 27.84%
Validation Loss - 4.05

We have saved the best performing model as a pth file and the PCA model as a pickle file. Additionally, we have added a cell which can be used for loading the testing data, performing all the pre-processing and evaluating the testing dataset on the model.