CSE244A - FINAL PROJECT

Teammates:
1) Mrudula Reddy Atla
2) Shiva Ganesh Ramakrishnan
3) Milind Varma Penumathsa

Project Overview:
This project focuses on image classification. The goal is to classify images into 135 predefined categories, including 120 dog breeds and 15 leaf classes.

The approach used here is an ensemble of two high performing models namely ResNet-101 and ViT (Vision Transformer). The ensemble combines the strengths of both models to achieve improved performance.

The final two cells of the notebook provide a custom pipeline to test the ensemble model on any user-defined dataset.

Dataset Preparation

1) **Dataset**
   The dataset consists of labeled images grouped into 135 classes:

   120 Dog Breeds
   15 Leaf Classes

2) **Preprocessing**
   A) **Train-Validation Split:**
      The data is split into training (80%) and validation (20%) sets.
      Labels for both splits are saved in CSV files.
   B) **Transformations:**
      Images are resized to 224x224 pixels.

   Code for preprocessing is included in the notebook.

Model Architecture

3.1 Vision Transformer (ViT)
- Pretrained ViT model google/vit-base-patch16-224-in21k.
- Modified the final classification head to output 135 classes.

3.2 ResNet-101
- Pretrained ResNet-101 model from PyTorch.
- Modified the final fully connected (FC) layer to output 135 classes.

Ensemble
The predictions of ViT and ResNet-101 are averaged to form the final ensemble prediction:

1. Softmax outputs from both models are combined to compute the ensemble probability distribution.
2. The class with the highest probability in the ensemble output is the final prediction.

Experimental Setup

i) Hardware

GPU: CUDA-enabled GPU for training acceleration (Google Colab)
Batch Size: 64 for training and validation.
Optimizer: Adam with weight decay for regularization.
Loss Function: CrossEntropyLoss.

ii) Train Details

Learning Rate: 1e-3
Epochs: 15
One-Hot Encoding: Used for multi-class labels.

Custom Test Pipeline

Purpose
To test the ensemble model on custom datasets.

Steps
Images are preprocessed and then passed through the trained ensemble that does a weighted average of the predictions of the two trained models
Results are saved in a .csv file for easy reference

Instructions to Run the Code

i) Environment Setup

- Install necessary dependencies
- Ensure PyTorch, torchvision, and Hugging Face Transformers are installed.

ii) Dataset Preparation

- Download and extract the dataset to a folder.
- Update paths for training and testing directories in the code.

iii) Training

- Train ViT and ResNet-101 models using the provided training script.
- Save the model checkpoints after each epoch for validation.
- Take the fine tuned version of the models with best validation accuracy
- Train the ensemble with 0.5 as initial weight for both the models

- Pick the ensemble version with highest validation accuracy

iv) Testing

In the last cell of the notebook, replace the value of variable **test_dir** with the path to the folder with the testing images

Run the last two cells of the notebook to generate predictions in a CSV file.

Results and Observations

Metrics

- Accuracy: Training and validation accuracies for both models were logged.
- Loss: Monitored for convergence during training.

Observations

- The ensemble of ViT and ResNet-101 achieved better performance than either model individually.
- The combination of ResNet's strength in feature extraction and ViT's global attention mechanisms effectively handled the diverse dataset.

Link to the final model:
https://drive.google.com/file/d/1VgAOMFnHlQD2HkJkR-tWnesUF1kjPszo/view?usp=drive_link