DA5402 - Assignment 5

Anirudh Rao be21b004

Task 1

The CIFAR-10 dataset was downloaded from Kaggle. The corresponding sub-folders under the `train` and `test` folders were merged. The resulting `cifar10` folder contains 10 sub-folders, each containing 6000 images. This folder has been included as part of the submission for this assignment.

Task 2

A Python script called `partition_dataset.py` was created for this task. This script performs the following tasks in order:

- Creates a folder called `data`
- Initializes `git` and `dvc`
- Creates a list of all 60000 images present in the sub-folders of the `cifar10` folder
- Partitions the images into three non-overlapping sets (`v1`, `v2`, `v3`) of 20000 each after randomly shuffling the list of all images
- For each partition, copies the 20000 images in the partition from `cifar10` to `data`, adds the data to `dvc`, adds `data` to `.gitignore`, commits `data.dvc` to `git`, and tags the commit with the appropriate partition name (`v1`, `v2`, `v3`)
- Clears the `data` folder once all partitions have been added to `dvc`

We now have three versions of the data that have been checked into `dvc` and `git`.

Task 3

For this task, the following files were created:

- `dvc.yaml` - Contains the stages of the pipeline along with the parameters and metrics to be tracked by `dvc`.

- `params.yaml` - Contains the configuration / parameters to be used by the different stages in the pipeline.
- `pull_data.py` - Checks out data from `dvc` according to the version name(s) provided in `params.yaml` and creates a `dataset.pkl` file. If more than one version of the data is to be used, it has to be specified in a space-separated manner, e.g. `v1 v2 v3`.
- `prepare_data.py` - Divides the checked out data in `dataset.pkl` into three partitions (`train.pkl`, `val.pkl`, `test.pkl`) whose proportions are provided in `params.yaml`.
- `train_model.py` - Builds a CNN classifier according to the hyperparameters provided in `params.yaml`, trains the model using `train.pkl`, fine-tunes (with accuracy as the metric) the model using `val.pkl` and the hyperparameters listed for tuning in `params.yaml`. Returns the fine-tuned model as `model.pkl` and the fine-tuned hyperparameters and validation accuracy as `model_params.json`. As instructed in the assignment problem statement, two hyperparameters (number of convolutional layers and number of filters) are set for fine-tuning with three values each.
- `evaluate.py` - Tests the performance of `model.pkl` on `test.pkl`. Reports the test accuracy and confusion matrix in `evaluation_report.json`.

Each stage of the pipeline is associated with the same `random_seed` parameter as instructed in the assignment problem statement.

Task 4

To run the set of experiments as instructed, a `run_experiments.py` script was created. This iteratively modifies the `params.yaml` file as required (with the mentioned dataset versions `v1`, `v2`, `v3`, `v1+v2`, `v1+v2+v3` and three different random seeds) and runs `dvc exp run` to run an experiment with those parameters. After running all experiments, it runs `dvc exp show` and stores the output at `dvc_exp_show.txt`. An example of `dvc_exp_show.txt` that was obtained after running `run_experiments.py` has been attached with the submission for this assignment.

Steps to run code

- Ensure `git` is installed on the local system.
- Run `pip install -r requirements.txt`.
- Download `cifar10.zip` from here. Unzip `cifar10.zip` and move the core folder (which contains 10 sub-folders) that is inside the unzipped folder to the working directory.
- Run `partition_dataset.py`.
- Run `run_experiments.py`.
- Check the output in `dvc_exp_show.txt` or run `dvc exp show --md` to view it in the command window.

The parameters of the stages can be modified manually in `params.yaml` if the user wishes to do so.