Machine Learning Pipeline for Image Classification with CIFAR-10

Due: June 1st, 2023 *** **Late Accepted Until:** June 4, 2023 w. 5pts off every day ***

Goal

The goal of the CS412-Machine Learning project is to gain hands-on experience on the ML project pipeline.

In this project, you will have the opportunity to apply several classification methods you have learned to one of the most popular machine learning datasets, *CIFAR-10*. By working on this project, you will gain a better understanding of various classification algorithms (their performances, advantages and limitations), as well as proper model selection and evaluation approaches.

Dataset

The CIFAR-10 dataset is a collection of 60.000 RGB-images in 10 classes, with 6.000 images per class. Each image has a dimension of 32 × 32 × 3, representing the height, width, and color channels. The 10 classes in the dataset are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. It was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton and has been widely used in the machine learning community for evaluating image classification algorithms.

The CIFAR-10 dataset is publicly available and can be downloaded from the official CIFAR-10 website. The dataset is provided by six binary files with five training data batches and one test batch for evaluation. Each batch consists of 10.000 images with 1.000 images per class. The data is stored in a dictionary format with keys: *batch_label*, *labels*, *data*, and $filenames$. The *labels* and *data* parts are relevant for our purposes, and the *data* field contains flattened images.

Note that you do not need to perform any train-test-split operation as the dataset provided already has preset 5-folds and a test set. You must use them exactly as is.

Methodology

Here is a list of the classification algorithms that you can apply to the problem (you should choose at least 3 that you deem suitable for the project):

- K-Nearest Neighbors (KNN)
- Decision Trees (DT)
- Multilayer Perceptron (MLP)
- Convolutional Neural Networks (CNN)
- Support Vector Machines (SVM)

For each method above, you are expected to:

- Choose hyperparameters (at least 1, up to 3) using 5-folds
- Train the final model using all folds with the best hyper-parameters
- Evaluate the final model's performance on the test data batch

Some of the methods above may or may not require a data preprocessing step (scaling, normalization, feature selection or extraction), we left you to decide what kind of data preprocessing strategy might be used for a given method.

Use a seed at the first cell in your notebook (before any random number generation) so that your results are comnparable/reproducable. We can say seed=42.

Bonus: Up to 10pts

Once you have completed training and evaluation of 3 baseline techniques discussed in the previous section, you may want to explore more advanced techniques to further improve your results. This could involve applying various dimensionality reduction techniques such as PCA, t-SNE, and UMAP, and observe the performance differences of unsupervised learning algorithms on the image data. Alternatively, you could try more sophisticated algorithms such as boosting or ensemble learning approaches to improve your classification accuracy.

Report

Your report should include (with the appropriate section titles in this order):

- **Summary/Abstract**: A 1-2 paragraph summary of your work. E.g. ”We tried .... algorithms for the CIFAR-10 problem and observed their performances after hyper-parameter optimization for each one separately. The best one was .... with %x.... accuracy. ”

- **Introduction**: Briefly explain the task and its importance (mention classification, briefly mention dataset and number of classes...)

- **Dataset**: Explain the dataset in a bit more detail (number of samples in each subset, train-test split, ...) and include at least one example for all different classes, so that your report is self-contained

- **Methodology**: Summarize the algorithms you have worked on, mention if any preprocessing is applied, and which hyperparameters are tuned and why/how. If a method was unsuccessful, you can summarize the finding saying that ”we have also tried ...., but it diverged/took very long/....” without going into more detail.

- **Experiments**: Include tables, figures for all hyperparameter tuning phases. You should have one big table where you present the final performances of all algorithms you tried. If applicable (e.g. in NN based approaches), you should also include a training curve (train and validation loss per epoch).

- **Discussion**: Here you can briefly discuss the best algorithm (no need to go into details of the others if CNNs clearly outperform for instance) and do a brief error analysis (you can give confusion matrix here along with the type of most seen errors...).

- **Bonus**: Describe if you have done anything above the basic requirements, such as those discussed in the Bonus section above.

- **Conclusion**: Briefly explain your final thoughts, did the experiments yield the same results as in your intuition, were you surprised in any of your findings? etc.

Important: Do not give a chronological account of your work (e.g. ”we first tried this, and then ...”. Instead, give how those algorithms works without the chronology of events. (e.g. ”we have tried these 3 approaches and ... worked best with an accuracy of ....”)

Grading

Your project will be graded based on:

- **Work**: Amount of work and whether have you followed the right steps in fine-tuning, evaluation, model selection...: 40pts
- **Report**: 40pts
- **Results compared to others**: 20pts (best group will have 20pts and grade will drop 2pts for approximately 0.1% drop in accuracy difference, down to the minimum of 10pts).
- **Bonus**: Up to 10pts

Submission

Person with the smallest ID number in the group will submit the **PDF report** and **ipynb files** to SUCourse. For each deliverable mentioned, please include your names and student IDs.