[F24] MACHINE LEARNING ASSIGNMENT 2

Due Date: 17th November 2024, (23:59)
Submission : Jupyter notebooks to moodle (**2 notebooks + TensorBoard files in .zip file**).

TASK 1 (25 POINTS)

Detection of inconsistencies in flower descriptions in online floristry and delivery platforms is essential for success, customer retention, and satisfaction. Many companies providing online floristry services are increasingly utilizing deep learning solutions to ensure that a flower image displayed on their platform matches the given description or category.

Objective

To implement a flower classification convolutional neural network (CNN) trained on the **Flowers102 dataset** (Dataset description: [1]).

Steps to Complete the Task

(1) **Data Loading & preprocessing:** Load the Flowers dataset using torchvision.datasets with appropriate transformations and preprocessing (i.e resizing, normalization, etc..)
(2) **Baseline Model implementation:** Implement CNN architecture shown in Table 1 which is a baseline model.
(3) **Baseline Models training:** Train the baseline model for 10 epochs using the SGD optimizer with a learning rate of 0.001, applying only min-max scaling for preprocessing and relu as activation function. Model performance at each training epoch should be logged to TensorBoard.
(4) **Model Evaluation:** Evaluate the model training and validation loss, accuracy, and F1 score.
(5) **Baseline model improvement:** Improve the model's performance using a minimum of three techniques presented in the course (e.g., Dropout, Batch Normalization, etc.), excluding transfer learning.
(6) **Transfer Learning:** Utilize transfer learning to achieve better performance than the improved baseline model above. A suggested pretrained model is ‘resnet50‘, which can be loaded from **torchvision.models.resnet50**; however, you are not limited to using resnet50 and may explore other suitable pretrained models as well.

NOTE: Overall 3 models are to be implemented and compared

Layer	Layer Type	Kernel size	Stride	Padding	Out channels
0	Input	-	-	-	3
1	Convolutional	3 x 3	1	1 x 1	32
2	Max-pooling	2 × 2	2	-	-
3	Convolutional	3 x 3	1	1 x 1	64
4	Max-pooling	2 × 2	2	-	-
5	Convolutional	3 x 3	1	1 x 1	128
6	Max-pooling	2 × 2	2	-	-
7	Flatten	-	-	-	-
8	Fully connected	-	-	-	512 (output)
9	Output Layer	-	-	-	102 (output)

Table 1. The details of the baseline CNN model

TASK 2 (25 POINTS)

In the field of computer vision, the ability to group similar objects based on their visual characteristics is fundamental for tasks such as image classification, retrieval, and organization. Clustering allows for the exploration of unlabelled data and helps to identify inherent patterns without prior knowledge of the categories. This task aims to leverage deep learning techniques to extract meaningful features from flower images and categorize them into distinct clusters.

Objective

To cluster images of flowers into distinct groups based on their visual features using unsupervised learning techniques. The goal is to utilize a pre-trained convolutional neural network (CNN) to extract features from the images and then apply a clustering algorithm to categorize the flowers.

Steps to Complete the Task

(1) **Data Loading & preprocessing:** Load the Flowers dataset using torchvision.datasets with appropriate transformations and preprocessing (i.e resizing, normalization, etc..)

(2) **Feature Extraction:** Use a pre-trained CNN model (e.g., ResNet-50, or Custom CNN from Task 1) to extract feature vectors from the images.

(3) **Clustering:** Cluster the extracted features using K-means. Choose an appropriate number of clusters.

(4) **Clusters visualization:** Use dimensionality reduction approach such as PCA on feature vectors for visualization purposes and plot the clustered data points in a 2D or 3D space, using different colors to represent different clusters

BONUS TASK: OUTLIER DETECTION & FUTURE ORDERS FORECASTING

For this bonus task, you will use data from assignment 1 to develop a model to forecast the number of new orders (for next *n* minutes) for a group of stores in a region. Additionally, you will identify and handle outliers in the dataset to improve the robustness of your model. Jupyter Notebook for this task should be separate.

Subtasks

- **Outlier Detection** (5 points): Analyze the dataset for outliers that could negatively impact the performance of your model. Implement outlier detection techniques for time-series data to identify and handle these outliers [1, 2, 3].
- **Forecasting Future orders** (5 points): Build a forecasting model to predict the number of new orders for each region over a specified future time period (i.e for next *n* minutes). Where *n* is a hyper-parameter.

GRADING

Each of the task mentioned above will be graded as based on the following :

(1) Exploratory Data Analysis, Feature engineering & preprocessing (20%);
(2) Machine learning or DNN model definition, training and hyper-parameters turning (35%);
(3) Comparison of ML models using appropriate evaluation metrics (30%);
(4) Reporting the goal, approach for training model and the attained ML model results in Jupyter notebook (use markdown) (15%)

SOURCE CODE (JUPYTER NOTEBOOK FILE)

The implementation should be in python. All deep learning tasks should be implemented using PyTorch framework and Tensorboard should be used to monitor the model performance. All the implementation should be presented in Jupyter notebook files. One jupyter notebook per task. The jupyter notebook file should contain have the following main sections : (i) Data Reading, Exploration and preprocessing; (ii) Machine learning or Deep learning model defining, training and hyper-parameters turning; (iii) Model performance evaluation; (iv) Conclusion and possible improvements;