Exercise 01 for MA-INF 2213 Advanced Computer Vision SS25
11.04.2025
Submission deadline: TBD

Important: We recommend to use **Python 3.12** for your solutions in this exercise. You are **not** allowed to use any additional Python modules for the parts that specifically stated as "using only **NumPy** library". Otherwise you will not get any points in all related parts.

Grading: Submissions that generate runtime errors or produce invalid results (e.g. nans, inf, incoherent visualization) might not be graded.

Plagiarism: Plagiarism in any form is prohibited. If your solution contains code copied from any source, you will receive **0 points** for the entire exercise sheet.

Submission: You can complete the exercise in groups of two, but only one submission per group is allowed. Include a *README.txt* file with your group members into each solution. Points for solutions without readme file will only be given to the uploader.

1 Price prediction

In this task, you are given a subset (/data/price) of DVM-CAR [1] dataset. This subset includes raw images of cars, so the first step is feature extraction. Thereafter, you are expected to build regression models from scratch by using only NumPy to learn entry price from training data. Then you will optimize the model parameters by using validation data and evaluate their performance on the test data.

1. **Data Loading and Feature Extraction:** Instead of using directly raw pixel values, you will use HOG method to extract features as vector. Therefore, you should start with extracting HOG features for training data, validation data and test data. You can use OpenCV for this purpose. Please do not forget to obtain price labels from filenames to use them as ground truth. The labels of each car image is embedded in the filename, formatted as [brand]$$[model]$$[year]$$...$$[price_label].jpg.
   (1 Points)

2. **Linear Regression**: Build a linear regression model which for price prediction. Do not use regularization, i.e. simply perform regression in the feature space. The regression model must be built using only **NumPy** library. Off-the-shelf models from other libraries (scikit-learn, etc.) are **not** allowed to use.
   (2 Points)

3. **Gaussian Process Regression**: Build a Gaussian process regression model with
   - a linear kernel

- a polynomial kernel
- a radial basis function (RBF) kernel

for price prediction. The regression model must be built using only **NumPy** library. Off-the-shelf models from other libraries (scikit-learn, etc.) are **not** allowed to use.

(4 Points)

4. **Evaluate Models and Visualize Results**: Evaluate performance of the regression model with different kernels on the test data with MSE (Mean Squared Error) metric. Evaluation function must be built using only **NumPy** library. Off-the-shelf evaluation functions from other libraries (scikit-learn, etc.) are **not** allowed to use. After evaluation, you should visualize a random subset of your predictions for test set. Your visualization should include the original images, the predicted values, and the ground truth values. You can use matplotlib for this purpose.

   (1 Points)

5. **Optimization of the results:** The main objective of this part is to get the best result with the Gaussian process regression model with RBF kernel you implemented in the previous part. From this point on, you are not limited to HOG features. You can extract any features to use, except features obtained by using Neural Networks. You should run your model with different parameters. Please make sure you only use validation data to evaluate your model during these experiments. Once you have your best parameters, you can use test data to report your result. You should again visualize the results of your best model.

   (3 Points)

2 Brand Classification

In this binary classification task, another subset (/data/class) of the DVM-CAR [1] dataset will be used.

1. **Data loading and Feature extraction** You should use similar feature extraction function from the regression task. You should adapt it to new label for brand name. If a filename starts with "Audi", its class should be 0. If it starts with "BMW", its class should be 1.

   (1 points)

2. **Nonlinear Logistic Regression:** Build a non-linear logistic regression model with RBF kernel. The loss function is the negative log-likelihood. You should use gradient descent for optimization. The model must be built using only **NumPy** library. Off-the-shelf models from other libraries (scikit-learn, etc.) are **not** allowed to use. Train your classifier for at least 10000 iterations with training data. In these iterations, use also validation

data to check your validation loss. You can plot train and validation loss values using matplotlib to understand trend better, and also print these values for every 200 iterations. Please observe the both loss values and decide on the best values for the number of iterations (less or more), and the step size of the gradient descent (smaller or larger). Then train your model with the latest parameters.

(4 Points)

3. **Evaluate Model:** Finally, evaluate performance of the classifier on the test data with accuracy metric. Accuracy for binary classification is defined as $(TP + TN)/(TP + FP + TN + FN)$ where TP and FN stand for true positive and false negative respectively. You will use test data as input to get predictions and will define a decision boundary (can be taken as 0.5) for probability outputs of your model. If your model output is greater than or equal to this value, the corresponding input will be predicted as positive class (1), otherwise negative class (0). Evaluation function must be built using only **NumPy** library. Off-the-shelf evaluation functions from other libraries (scikit-learn, etc.) are **not** allowed to use.

(1 Points)

3 Derivatives of Log-Likelihood

In logistic regression, the log-likelihood is given by

$$L = \sum_{i=1}^{I} y_i log\left(\frac{1}{1 + exp(-w^T x_i)}\right) + \sum_{i=1}^{I} (1 - y_i) log\left(\frac{exp(-w^T x_i)}{1 + exp(-w^T x_i)}\right)$$

Show the gradient to be

$$\frac{\partial L}{\partial w} = -\sum_{i=1}^{I} \left(\frac{1}{1 + exp(-w^T x_i)} - y_i\right) x_i \tag{1}$$

Provide a comprehensible solution.

(3 Points)

References

[1] Huang, J., Chen, B., Luo, L., Yue, S., & Ounis, I. (2022, December). DVM-CAR: A large-scale automotive dataset for visual marketing research and applications. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 4140-4147). IEEE.