Statistical Methods in AI (CSE/ECE 471)
Spring-2020
Assignment-3 (200 points)
Posted on: 23/03/20
Due on: 11:59 P.M. 01/04/20

Instructions

1. Assignment must be implemented in Python 3 only.
2. Your submission should have `logisitc_regression.py`, `mnist.py`, `regression_time_series.py` and `report.pdf` and this should be in a zipped directory `RollNo_A3.zip`.
3. Any attempts at plagiarism will be penalized heavily.
4. A basic laptop without a GPU should be sufficient for all the questions. However, if you need additional compute, feel free to use Google Colab (`https://colab.research.google.com`)

Questions

1. (40 points) PCA

    1. You are provided a subset of original dataset of faces with labels. `<a>_<b>.jpg` image is bth image of ath class.
    2. Please find the link to the dataset here: `https://bit.ly/39eUAJi`.
    3. Perform PCA over all the images in the dataset. You may downscale the image and convert it to grayscale for ease of computation.
    4. We can reconstruct the image back using a small number of components. Plot a graph showing the total mean square error over all train images vs the number of principal components used to reconstruct. Include this plot in your submission. (Use a reasonable range for number of components)
    5. Decide N, the number of principal components required such that the reconstructed images will have mean squared error less than 20% over all train images. Display those N principal components as reconstructed images. You will see some base structures of the faces. Include these images in your report.
    6. Use scatterplots to examine how the images are clustered in the 1D, 2D and 3D space using the required number of principal components.
    7. This part will be evaluated qualitatively from your report. Also you will use some code of this method to transform images to reduced representation for Q2 and Q3 of this assignment.

2. (40 points) Logstic Regression

    1. Implement logistic regression to classify the images provided in the dataset.
    2. You are not allowed to use scikit-learn or any other machine learning library/framework for this purpose. You may use numpy(for Eigendecomposition and other linear algebra functionality) and OpenCV for Image I/O.
    3. For this process, you will first perform PCA on the training set and use the feature vectors obtained after dimensionality reduction to perform logistic regression.
    4. Report accuracy score, Confusion matrix and any other metrics you feel useful.
    5. For automated evaluation, your predicted labels will be generated by running the following: `python3 logistic_regression.py <train_file_location> <test_file_>`. The output should be the predicted labels which should be sent to standard output.

3. (60 points) MNIST Classification

    1. Implement Multilayer Perceptron(MLP), Convolutional Neural Network(CNN) as well as Support Vector Machines(SVM) to classify digits from the MNIST dataset. You can find the data here `http://yann.lecun.com/exdb/mnist/`.
    2. You are allowed to use scikit-learn as well as deep learning frameworks such as PyTorch, keras etc, for this purpose.

3. Report accuracy , Confusion matrix and any other metrics you feel may be useful.
4. Experiment with different architectures(number of hidden layers, activation functions etc., kernels in the case of SVM) and see the impact on performance. Summarize your findings in the report.
5. For automated evaluation, put your best performing method in `mnist.py` and your predicted labels for the test set will be generated by running the following: `python3 mnist.py <directory_location>`. The output should be the predicted labels which should be sent to standard output. You are encouraged to use python-mnist `https://pypi.org/project/python-mnist/` for easy loading of the data.

4. (60 points) Regression

1. In this question, you are expected to perform regression over the dataset of global active power values. You are supposed to take the active power values in the past one hour and predict the next active power value.
2. Implement Multilayer Perceptron(MLP) as well as a linear regression model for this question. You can find the data here `https://archive.ics.uci.edu/ml/machine-learning-databases/00235/household_power_consumption.zip`.
3. Compare and contrast the performance of both the models on metrics like Root Mean Squared Error(RMSE), Mean Absolute Percentage Error(MAPE) score and any other metrics you feel may be useful.
4. It would be sufficient if you consider only the Global_active_power field.
5. You are allowed to use scikit-learn as well as deep learning frameworks such as PyTorch, keras etc, for this purpose.
6. Experiment with different architectures(number of hidden layers, activation functions etc) and see the impact on performance. Summarize your findings in the report.
7. Also experiment on taking some more window of past power values and report the performance (For example taking a window of two hours instead of one).
8. For automated evaluation, put your best performing method in `regression_time_series.py` and your predicted values for the test set will be generated by running the following: `python3 regression_time_series.py <directory_location>`. The output should be the predicted power consumption values which should be sent to standard output.