

Machine Learning

Assignment 3: SVM & ANN

1. Perform the following implementation and analysis of binary SVM classification on the provided data 40 marks
 - a. Randomly pick 80% of the data as a training set and the rest as a test set. Implement the binary SVM classifier using the following kernels: 20 marks
 - i. Linear
 - ii. Quadratic
 - iii. Radial basis function
 - b. Report both the training and test set classification accuracies for the most suitable value of generalization constant C for each of the three kernels. Consider C empirically. Report the accuracy in the form of a comparison table with corresponding C values (C value which provides the best test set accuracy is the most suitable one out of all those different trial values that you try.) 20 marks
2. Build a MLP classifier for the given dataset. Use stochastic gradient descent optimiser for the models. Find the number of nodes in the input and output layer according to the dataset and justify it in the report. Randomly pick 80% of the data as a training set and the rest as a test set. Use packages of pytorch. 55 marks
 - a. Vary the number of hidden layers and number of nodes in each hidden layer as follows. 20 marks
 - i. 0 hidden layer
 - ii. 1 hidden layer with 2 nodes
 - iii. 1 hidden layer with 6 nodes
 - iv. 2 hidden layers with 2 and 3 nodes respectively
 - v. 2 hidden layers with 3 and 2 nodes respectively
 - b. For each of the architectures, vary the learning rates as 0.1, 0.01, 0.001, 0.0001, 0.00001. Plot graph for the results with respect to accuracy. (Learning rate vs accuracy for each model and model vs accuracy for each learning rate.) 20 marks
 - c. Mention the architecture and hyper parameters (including optimizer and learning rate) of the best found model in the report. Try to justify it. 7.5 marks
 - d. Compare execution time in CPU and GPU for each model. Conclude your findings. 7.5 marks
3. Compare the performances of both the classifiers. 5 marks

Submission instructions:

1. Submit your codes by implementing them only in PYTHON. No other programming language is allowed. You may use inbuilt library functions to perform the tasks.
2. Submit a README file which will contain the instructions on how to execute your code.
3. Submit a report briefly explaining the procedure and the results.

If required, any data cleaning or pre-processing can be done. Mention them in the report. The source code, README file, and the report must be uploaded as a single compressed file (.tar.gz or .zip). The compressed file should be named as: {Group_Number}_ML_A3.zip or {Group_NUMBER}_ML_A3.tar.gz.

Datasets:

The links for the available datasets are provided below. The descriptions related to these datasets are provided in the corresponding pages.

1. <https://archive.ics.uci.edu/ml/datasets/Thoracic+Surgery+Data>