February-June 2022 Semester
CS671: Deep Learning and Applications
Programming Assignment I

Date: Feb 28, 2022

Deadline for submission of code and report: **Sunday, March 20, 2022, 10:00 PM**

Classification tasks

Datasets:

Dataset 1: Linearly separable classes: 3 class, 2-dimensional linearly separable data is given. Each class has 500 data points.

Dataset 2: Nonlinearly separable classes: 2-dimensional data of 2 or 3 classes that are nonlinearly separable. The number of examples in each class and their order is given at the beginning of each file.

Divide the data from each class into training, validation and test data. From each class, train, validation and test split should be 60%, 20% and 20% respectively.

Models:

1. Perceptron with sigmoidal activation function for each of the datasets. Use one-against-one approach for 3-class classification. Implement gradient descent method for perceptron learning algorithm.
2. Fully connected neural network (FCNN) for each of the datasets. Try FCNN with one hidden layer for Dataset1 and 2 hidden layers for Dataset2. Try different number of hidden nodes for both datasets and observe the results. Implement stochastic gradient descent (SGD) for backpropagation algorithm. Use squared error as instantaneous loss function.

Presentation of results:

1) Plot of average error (y-axis) vs epochs (x-axis). For FCNN, give the plot only for the best architecture selected after cross validation.
2) Decision region plot superimposed by training data for each of the datasets.
   a) For perceptron, give the decision region plot between each pair of classes and also after combining.
   b) For FCNN, give the decision region plot for the best architecture selected after cross validation
3) Confusion matrix and classification accuracy. (For FCNN, give the confusion matrix and classification accuracy for each of the different architectures along with the best architecture selected after cross validation.)
4) Comparison of performance of different models for each dataset.
5) Plots of outputs for each of the hidden nodes and output nodes in FCNN for each of the datasets after the model is trained. Here, x and y axis are input variables of each example, z axis is output of hidden node/output node. Give the plots for training, validation and test data. (Give the plots for the best architecture selected after cross validation)
6) For FCNN, give the expressions for the change of weights in different layers (first and second hidden layer and output layer) of the network.

Regression tasks

Datasets

Dataset 1: 1-dimensional (Univariate) input data
Dataset 2: 2-dimensional (Bivariate) input data
Divide the data into training, validation and test data. Train, validation and test split should be 60%, 20% and 20% respectively.

Models:

1. Perceptron with linear activation function for each of the datasets. Implement gradient descent method for perceptron learning algorithm.
2. FCNN model for each of the datasets. Try FCNN with one hidden layer for Dataset1 and one as well as two hidden layers for Dataset2. Try different number of hidden nodes for both datasets and observe the results. Implement stochastic gradient descent (SGD) for backpropagation algorithm. Use squared error as instantaneous loss function.

Presentation of Results:

1. Plot of average error (y-axis) vs epochs (x-axis). For FCNN, give the plot only for the best architecture selected after cross validation.
2. Plots of the values of mean squared error (MSE) on training data, validation data and test data. (For FCNN, give the values for different model complexities i.e. different architectures.)
3. Plots of model output and target output for training data, validation data and test data. (For FCNN, give the plots for the best architecture selected after cross validation)
4. Scatter plot with target output on *x*-axis and model output on *y*-axis, for training data, validation data and test data. (For FCNN, give the plots for the best architecture selected after cross validation)
5. Plots of outputs for each of the hidden nodes and output nodes in FCNN for each dataset. Give the plots for training, validation and test data. (Give the plots for the best architecture selected after cross validation)
6. For FCNN, give the expressions for the change of weights in different layers (first and second hidden layer and output layer) of the network.

Each group of students must use the dataset identified for that group only.

Expectation of the assignment is to implement perceptron, FCNN and backpropagation learning algorithm from scratch using Python or MATLAB or any other programming language.
Note: You are not supposed to use libraries of neural network, backpropagation etc.

Selection of model complexity is to be done using the cross-validation method (only for FCNN). Initial weights should be same for each model during model selection.

Report should be in PDF form and report by a team should also include the observations about the results of studies.

Upload in Moodle, all your codes and report in a single zip file.

- **Give the name of the folder as Group<*number*>_Assignment1, Example: Group*01*_Assignment1.**
- **Give the name of the zip file as Group<*number*>_Assignment1.zip. Example: Group*01*_Assignment1.zip.**

We will not accept the submission, if you don't follow the given instruction.