February-June 2023 Semester
CS671: Deep Learning and Applications
Programming Assignment II

Date: Feb 21, 2023

Deadline for submission of code and report: **Tuesday, March 07, 2022, 10:00 PM**

Classification tasks

Datasets: (Same as Assignment I)

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.

Model: 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)*: 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*: Give the decision region plot for the best architecture selected after cross validation.
3) *Confusion matrix and classification accuracy*: Give the confusion matrix and classification accuracy for each of the different architectures along with the best architecture selected after cross validation.
4) 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)
5) Comparison of performance with that of the single neuron model (Assignment-1) for each dataset.
6) Inferences on the plots and inferences on the results observed.

Regression tasks

Datasets: (Same as Assignment I)

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.

Model: 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)*: 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 (like a bar chart)*: Give the values for different model complexities i.e. for different architectures.
3. *Plots of model output and target output for training data, validation data and test data*: 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*: 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. Comparison of performance with that of the single neuron model (Assignment-1) for each dataset.
7. Inferences on the plots and inferences on the results observed.

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

The expectation of the assignment is to implement FCNN from scratch using Python or MATLAB or any other programming language.
Note: You are not supposed to use libraries of TensorFlow, neural network, gradient descent etc.

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

Instruction:

Upload in Moodle all your codes in a single zip file. Note that code(s) should be in a .py file, if you are coding in Python.

- **Give the name of the code folder as Group<*number*>_Assignment2_code**
  Example: Group01_Assignment2_code.
- **Give the name of the zip file as Group<*number*>_Assignment2_code.zip**
  Example: Group01_Assignment2_code.zip

Upload the report as PDF file.

- **Give the name to the report file as Group<*number*>_Assignment2_report.pdf**
  Example: Group01_Assignment2_report.pdf

We will not accept the submission if you don't follow the above instructions.