February-June 2021 Semester
CS671: Deep Learning and Applications
Programming Assignment I

Date: March 1, 2021

Deadline for submission of code and report: Tuesday, March 23, 2021, 10:00 PM

Classification tasks

Datasets:

Dataset 1: 2-dimensional input data:

(a) **Linearly separable classes**: 2-dimensional artificial data of 3 classes that is linearly separable. Each class has 500 data points.

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

Divide the data into training, validation and test data. Consider 60% of the data from each class as training data, 20% of the data from each class as validation data and remaining 20% of data from each class as test data.

Dataset 2: **Image data**: Scene image data corresponding to 3 different classes

Consider 40 images from each class in training folder as training data, remaining 10 images from each class in training folder as validation data.

Models:

1. Perceptron with sigmoidal activation function for Dataset 1(a) and 1(b)
2. MLFFNN for Datasets 1(a), 1(b) and 2

Presentation of Results:

1. Plot of epochs (x-axis) vs average error (y-axis).
2. Decision region plot for Datasets 1(a) and 1(b)
3. Confusion matrix and average classification accuracy
4. Comparison of performance of different models for each dataset
5. Plots of outputs for each of the hidden nodes and output nodes in MLFFNN for Dataset 1(a) and 1(b) 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.

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. Consider 60% of the data from each class as training data, 20% of the data from each class as validation data and remaining 20% of data from each class as test data.

Models:

1. Perceptron with linear activation function for Datasets 1 and 2.
2. MLFFNN model for Datasets 1 and 2.

Presentation of Results:

1. Plot of epochs (x-axis) vs average error (y-axis).
2. Plots of the values of mean squared error (MSE) on training data, validation data and test data, for different model complexities.
3. Plots of model output and target output for training data, validation data and test data.
4. Scatter plot with target output on *x*-axis and model output on *y*-axis, for training data, validation data and test data.

5. Plots of outputs for each of the hidden nodes and output nodes in MLFFNN after different number of epochs during training. (For Dataset 1 and 2)

Extract Bag-of-visual-words (BoVW) feature each of the images. The description and steps of extracting BoVW feature is as follows:

1. Consider 32 x 32 nonoverlapping patches on every image (from training and test sets). For example, if image size is 256 x 256, there will be 32 numbers of 32 x 32 nonoverlapping patches. If the image size is not the divisible of 32, then you can copy the pixels to make those patches of size 32.
2. Extract 8-bin **colour histogram** from every colour channel (R, G and B) from a patch. It results in 3, 8-dimentional feature vectors. Concatenate them to form 24-dimentional feature vector.
3. Similarly extract 24-dimentional feature vector from every patch.
4. Repeat the above steps to all the images in training and test set of all the classes.
5. Take the 24-dimentional colour histogram feature vectors of all the training examples of all the classes.
6. Group them into 32 clusters using K-means clustering algorithms.
7. Now take an image, assign each 24-dimentional colour histogram feature vector to a cluster.
8. Count the number of feature vectors assigned to each of the 32 clusters.
9. ***This results in a 32-dimentional vector representation for that image.***
10. ***Divide elements of 32-dimensional vector by the number of patches considered for that image.***
11. ***This results in a 32-dimentional BoVW representation for that image.***
12. ***Repeat this (Step 7 to 11) for every image in training and test set.***

Colour histogram is computed as follows from a colour channel:

1. When the given image is read, it will be read as 3-dimentioanl matrix of pixel values. Each dimension is corresponding to a colour channel. The pixel values in each colour channel are in the range 0 to 255.
2. For a colour channel,
   (a) Divide this range into 8 equal bins.
   (b) Count the number of pixels falling into each bin. This results in a vector of 8 values.
   (c) This is the 8-dimentional colour histogram (from a colour channel) feature vector.
3. Do the same for other colour channels. Concatenate those three 8-dimentional colour histogram vectors to form 24-dimentional vector.

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

Expectation of the assignment is to implement perceptron, MLFFNN and backpropagation learning algorithm from scratch using Python or MATLAB or any 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 MLFFNN).

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

Send/share all your codes and report in a single zip file with me and all the TAs.

- **Give the name of the folder as Group<*number*>_Assignment1, Example: Group*1*_Assignment1.**
- **Give the name of the zip file as Group<*number*>_Assignment1.zip. Example: Group*1*_Assignment1.zip.**

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