**MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY, JAIPUR**

## Department of Computer Science and Engineering

Machine Learning Lab 2022

*Exercise Sheet*

**Note:**
(1) All the problems are to be implemented in Python. The exercises are to be done individually.
(2) The submission date is the last lab day for the exercise. The number of labs for each exercise is mentioned alongside.

Following is a list of exercises to be done as part of Machine Learning Lab course.

1. (Labs - 2) Generate a linearly separable data (random) set of size 20. Plot the examples $\{(x_n, y_n)\}$ as well as the target function $f$ on a plane. Be sure to mark the examples from different classes differently, and add labels to the axes of the plot.

   (a) Run the perceptron learning algorithm on the data set above. Report the number of updates that the algorithm takes before converging. Plot the examples $\{(x_n, y_n)\}$, the target function $f$, and the final hypothesis $g$ in the same figure. Comment on whether $f$ is close to $g$.

   (b) Repeat everything in $(a)$ with another randomly generated data set of size 100. Compare your results with $(a)$.

2. (Labs - 3) Write a python script that can find $w_0$ and $w_1$ for an arbitrary dataset of number of hours studied versus rank of a students as $\{(x_n, y_n)\}$ pairs. Find the linear model, $y = w^T x$, that minimizes the squared loss. Derive the optimal parameter value, $\widehat{w}$, for the total training loss: $L = \sum_{n=1}^{N}(y_n - w^T x_n)^2$ using the closed form solution and gradient descent algorithm. Using the model predict the rank for the number of hours studied. Plot the training error v/s number of iterations (for gradient descent).

   (a) Now fit a $4^{th}$ order polynomial function $f(x; w) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$ to this data. What do you notice about magnitude of the coefficients from $w_0$ to $w_4$?

   (b) Fit high order polynomials like $[6, 8, 10, 12, 14$ and $16$ degree$]$. Plot the training error (for all degrees) of the polynomial on the same plot. Using different colors for different degrees and also show legend on the graph. What do you notice about the magnitude of the coefficients? Comment about generalization and overfitting.

(c) (Labs - 2) Linear regression can also be augmented with the $l_2$-norm regularization:

$$\min_{w} E(w) + \lambda ||W||_2^2,$$

where $E(w)$ is the linear loss. Please change your gradient descent algorithm accordingly and use cross-validation to determine the best regularization parameter.

(d) Plot the training and testing performance curves.

(e) Indicate in the plot the best regularization parameter you obtained (using cross validation).

3. (Labs - 4) The task is to predict possibility of a patient having a heart attack based on his attributes. Use the heart attack dataset given in file Heart.csv dataset to build a logistic regression model. Take 75% of the data as training and 25% as test set.

- Familiarize yourself with the data by giving a plot of any three features of the two classes in two different colors.
- Implement logistic regression for classification using gradient descent to find the best separator you can using the training data only (use your 2 features from the above question as the inputs). The output is $+1$ if the example is a *yes* and $-1$ for a *no*.
- Give separate plots of the training and test error, together with the separators.

OR
Handwritten Digits Data: You should download the two data files with handwritten digits data: training data (ZipDigits.train) and test data (ZipDigits.test). Each row is a data example. The first entry is the digit, and the next 256 are grayscale values between $-1$ and 1. The 256 pixels correspond to a $16 \times 16$ image. For this problem, we will only use the 1 and 4 digits, so remove the other digits from your training and test examples. Please submit your Python code implementing the logistic regression for classification using gradient descent.

- Familiarize yourself with the data by giving a plot of two of the digit images.
- Develop two features to measure properties of the image that would be useful in distinguishing between 1 and 4. You may use symmetry and average intensity (as discussed in class).
- As in the text, give a 2-D scatter plot of your features: for each data example, plot the two features with a red x if it is a 4 and a blue o if it is a 1.
- Classifying Handwritten Digits: 1 vs. 4. Implement logistic regression for classification using gradient descent to find the best separator you can using the training data only (use your 2 features from the above question as the inputs). The output is $+1$ if the example is a 1 and $-1$ for a 4.
- Give separate plots of the training and test data, together with the separators.

4. (Labs - 2) The third classification assignment is based on the $K$ nearest neighbours method which you will need to implement in Python. Load the data given in NearestNeighbor.csv file. Draw the Voronoi regions for each data. Generate and illustrate the decision boundary by aggregating the neighboring local models (Voronoi regions) that make the same prediction. Estimate the minimum distance between test feature vector and sample vector in the training/test set by using Euclidean distances.

   - Classify the test samples into appropriate class label using minimum distance $K = 3$-nearest neighbor classifier. Compute the Euclidean distance between the test data point and all the training data. Sort the calculated distances in ascending order. Get the $k$ nearest neighbors by taking top $k$ rows from sorted array. Find the majority class of these rows. Return predicted class. Save the predicted class in a vector $\hat{Y}$.
   - Give the accuracy of your model. You already have the true values ($Y$) and you have your predicted values $\hat{Y}$ from your $K$-NN model. The accuracy is calculated by dividing the correctly classified samples count by total samples.
   - Experiment with different values of $k$. What happend when $k = 1$ and when $k = n$?

5. (Labs - 3) Consider a theoretical biometric matcher that generates distance scores in the range $[-\infty, \infty]$. Assume that the genuine and impostor score distributions due to this matcher can be approximately modeled as $N(30, 10)$ and $N(60, 15)$, respectively. Here, $N(\mu, \sigma^2)$ denotes normal distribution with mean, $\mu$, and variance, $\sigma^2$. Suppose the following decision rule is employed: $s$ is classified as a genuine score if $s \leq \eta$; else it is classified as an impostor score. Here, $\eta \in [0, 100]$.

   - Plot the genuine and impostor distributions in a single graph. The distributions should be contained in the range $[0, 100]$.
   - If $\eta = 50$, what is the FMR (i.e., FAR) and FNMR (i.e., FRR) of the biometric matcher?
   - Given $s$ is classified as a genuine score if $s \leq \eta$; else it is classified as an impostor score. If $\eta = 75$, what is the FMR (i.e., FAR) and FNMR (i.e., FRR) of the biometric matcher?
   - Write a program to compute the *DET* and *ROC* curves based on these two distributions. Also compute the AUC of this matcher.

6. (Labs - 4) In this problem you will implement forward and backward propagation methods for learing an XOR gate using a multi-layer neural network with $K = 2$ hidden layers. Assume that $K$ is a user input less than 10. Implement the networks with sigmoid as the activation function. Assume that the last layer has a linear activation function and the loss function is $l(y, \hat{y}) = ||y - \hat{y}||_2^2$. Submit your code (along with any instructions necessary to run it), the forward pass outputs at each layer and the gradients of the parameters $\left(W_{ij}^k, b_i^k\right)$. Use 10 nodes in each layer and initialize the weights randomly.

- Report the training and validation accuracy as a function of iterations (with 5 hidden layers). Report the convergence speed of the training procedure (with 5 hidden layers) for the Stochastic Gradient Descent optimization algorithm.
- Report the training and validation accuracy for cross-validation.
- Finally, report the best test error that you can achieve.