|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table 1 | | | | |
| Data | Xi1 | Xi2 | Yi | αi |
| X1 | 4 | 2.9 | 1 | 0.414 |
| X2 | 4 | 4 | 1 | 0 |
| X3 | 1 | 2.5 | -1 | 0 |
| X4 | 2.5 | 1 | -1 | 0.018 |
| X5 | 4.9 | 4.5 | 1 | 0 |
| X6 | 1.9 | 1.9 | -1 | 0 |
| X7 | 3.5 | 4 | 1 | 0.018 |
| X8 | 0.5 | 1.5 | -1 | 0 |
| X9 | 2 | 2.1 | -1 | 0.414 |
| X10 | 4.5 | 2.5 | 1 | 0 |

1. **[25 pts] Support Vector Machines**: Given 10 points in Table 1, along with their classes and their Lagrangian multipliers (αi), answer the following questions. The solution is already provided in the lecture slide.
   1. [10 pts] Using NumPy, implement a simple SVM that returns the coefficients of the hyperplane. **The simple SVM does not require an SDG approach.** The simple SVM has three arguments, X, Y, and , as simple\_SVM(X,Y,alpha). What is the equation of h(x)? Draw the hyperplane with the 10 points.
   2. [10 pts] Write a method that calculates the distance of each point from the hyperplane and returns a binary result if a point is within the margin. If a point is in the margin, the binary result is 1 and 0 otherwise. What is the maximized margin? Return the result in the table format as shown.

|  |  |  |
| --- | --- | --- |
| Data | Distance | In Margin? |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. [5 pts] Classify the point z = (3, 3)T using h(x) from above.

1. **[55 pts] Neural Networks**: Apply a Neural Network (NN) model to predict handwritten digit images from 0 to 9. The pickled file represents a tuple of 3 lists : the training, validation, and test sets. Each of the three lists is a pair formed from a list of images and a list of class labels for each image. An image is represented as a numpy 1-dimensional array of 784 (28 x 28) float values between 0 and 1 (0 stands for black, 1 for white). The labels are numbers between 0 and 9, indicating which digit the image represents. The code block below shows how to load the dataset.

import cPickle, gzip, numpy  
# Load the dataset  
f = gzip.open(’mnist.pkl.gz’, ’rb’)  
train\_set, valid\_set, test\_set = cPickle.load(f)  
f.close()

* 1. [15 pts] Using Scikit-learn, train a neural network model using the training data and predict the labels of data samples in the test set.
  2. [15 pts] Using GridSearchCV, tune your model hyper-parameters (number of hidden neurons in your hidden layer) on the validation set. Report the best model.
  3. [10 pts] Plot the train, validation, and test errors as a function of the epochs.
  4. [5 pts] Report the best accuracy on the validation and test data sets.
  5. [5 pts] Apply early stopping using the validation set to avoid overfitting.
  6. [5 pts] Give a brief description of your observations.