ITU Computer and Informatics Faculty
BLG 454E Learning From Data, Spring 2018
Homework #4
Due May 27, 2018 11pm

1. [**Multilayer Perceptron**]

   You will use the dataset *dataTrain.csv* and *dataTest.csv* given for question 1. The last column of the file represents the class label:

   dataTrain.csv: Training data.

   dataTest.csv: Test data.

   You will use dataTrain.csv for training a multilayer perceptron model which has 64 inputs, 10 hidden units and 10 outputs. You will use Alpaydin's book backpropagation algorithm for training this multilayer perceptron(Figure 1). This is a multiclass classification problem therefore you should use softmax outputs different from this algorithm. In addition please don't forget to add the bias units to your code and shuffle training data for improving accuracy.

   (a) (25 pts) Report cross entropy error value for 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50, 100, 200th iteration in the backpropagation(training) phase.

   (b) (25 pts) Use trained multilayer perceptron model and calculate the accuracy for test set(dataTest.csv) and report it.

   Note: You must use backpropagation algorithm(Figure 2) therefore you don't use any built in multilayer perceptron function however you can use built in functions for mathematical operations(e.g. numpy.exp, numpy.argmax, numpy.random, numpy.dot, numpy.zeros, numpy.log etc.)

Initialize all v_{ih} and w_{hj} to rand$(-0.01, 0.01)$
Repeat
  For all $(\mathbf{x}^t, r^t) \in \mathcal{X}$ in random order
    For $h = 1, \ldots, H$
      $z_h \leftarrow \text{sigmoid}(\mathbf{w}_h^T \mathbf{x}^t)$
    For $i = 1, \ldots, K$
      $y_i = \mathbf{v}_i^T \mathbf{z}$
    For $i = 1, \ldots, K$
      $\Delta \mathbf{v}_i = \eta (r_i^t - y_i^t) \mathbf{z}$
    For $h = 1, \ldots, H$
      $\Delta \mathbf{w}_h = \eta (\sum_i (r_i^t - y_i^t) v_{ih}) z_h (1 - z_h) \mathbf{x}^t$
    For $i = 1, \ldots, K$
      $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta \mathbf{v}_i$
    For $h = 1, \ldots, H$
      $\mathbf{w}_h \leftarrow \mathbf{w}_h + \Delta \mathbf{w}_h$
Until convergence

Figure 1: Backpropagation algorithm for training a multilayer perceptron for regression with K outputs. This code can easily be adapted for two-class classification (by setting a single sigmoid output) and to K > 2 classification (by using softmax outputs).

2. [**Clustering**]

   You will use the dataset *Cluster.csv* given for question 2.

   Firstly, reduce the dimensionality of dataset to 2 using PCA(you can use built in PCA function) and then Apply Alpaydin's book k-means algorithm to this 2 dimensional dataset in Figure 2:

(a) (25 pts) Sum of Squared Error(SSE) is the sum of the squared differences between each observation and its group's mean. It can use for evaluating clustering:

$$\textbf{SSE} = \sum_{i=1}^{n} \sum_{x \in C_i} dist^2(m_i, x)$$

x is a object in C_i cluster

$\boldsymbol{m_i}$ is the center of C_i cluster

Calculate Sum of Squared Error(SSE) value for k=1, 5, 10 and 20 in this dataset and report them. Decide which k value is the best clustering parameter for this dataset and report it.

(b) (25 pts) Cluster this dataset using finded the best k value in section a. Draw the decision boundaries and report it.

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$
Repeat
  For all $\boldsymbol{x}^t \in \mathcal{X}$
    $b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$
  For all $\boldsymbol{m}_i, i = 1, \ldots, k$
    $\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$
Until $\boldsymbol{m}_i$ converge

Figure 2: k-means algorithm(Alpaydin, E., 2014. Introduction to machine learning. MIT press.)

Submission Policy

- Prepare the report and code. Only electronic submissions through Ninova will be accepted no later than May, 27 at 11pm.
- You may discuss the problems at an abstract level with your classmates, but you should not **share or copy code** from your classmates or from the Internet. You should submit your **own, individual** homework.
- Note that your codes and reports will be checked with the plagiarism tools **including previous years submissions**!
- Academic dishonesty, including cheating, plagiarism, and direct copying, is unacceptable.
- If a question is not clear, please first read Alpaydin's book relevant parts then send email to (**cebeci16@itu.edu.tr**).
- You can only use Python or Matlab programming languages for this assignment.

Bonus marks (10pts)

- Clarity and nicely described report
- Using Latex template for the report

Deductions (-10pts)

- Spelling errors.
- Messiness
- Lack of content.
- Irrelevant / mistaken content.