UCLA CS 145 Homework #2

DUE DATE: Sunday 10/29/2017 11:59 pm

Note:

- You are expected to submit both a report and code. The submission format is specified on CCLE under HW2 description.
- "`########## Please Fill Missing Lines Here ##########`" is used where input from you is needed in the code file.

1. Support Vector Machine

The table shown below contains 20 data points and their class labels.

Point #	x1	x2	Class (y)
1	0.52	-1	1
2	0.91	0.32	1
3	-1.48	1.23	1
4	0.01	1.44	1
5	-0.46	-0.37	1
6	0.41	2.04	1
7	0.53	0.77	1
8	-1.21	-1.1	1
9	-0.39	0.96	1
10	-0.96	0.08	1
11	2.46	2.59	-1
12	3.05	2.87	-1
13	2.2	3.04	-1
14	1.89	2.64	-1
15	4.51	-0.52	-1

16	3.06	1.3	-1
17	3.16	-0.56	-1
18	2.05	1.54	-1
19	2.34	0.72	-1
20	2.94	0.13	-1

Suppose by solving the dual form of the quadratic programming of svm, we can derive the $\boldsymbol{\alpha}$'s for each data point as follows:

$\boldsymbol{\alpha}_2$ = 0.9492
$\boldsymbol{\alpha}_{18}$ = 0.3030
$\boldsymbol{\alpha}_{19}$ = 0.9053
Others = 0

(1) Please point out the support vectors in the training points.
(2) Calculate the normal vector of the hyperplane: w
(3) Calculate the bias b, according to $b = \Sigma_{k:\alpha_k \neq 0} (y_k - w^T x_k)/N_k$ where $x_k = (x_{k1}, x_{k2})^T$ indicate the support vectors and N_k is the total number of support vectors.
(4) Write down the learned decision boundary function $f(x) = w^T x + b$ (the hyperplane) by substituting w and b with learned values in the formula.
(5) Suppose there is a new data point $x = (-1, 2)$, please use the decision boundary to predict its class label.
(6) Show a plot of the data points and your decision boundary line (x1 feature on x-axis, x2 feature on y-axis) in your report. Plot both data points and decision boundary in the same graph, and use different colors to represent points in different classes (y).

2. Artificial Neural Network

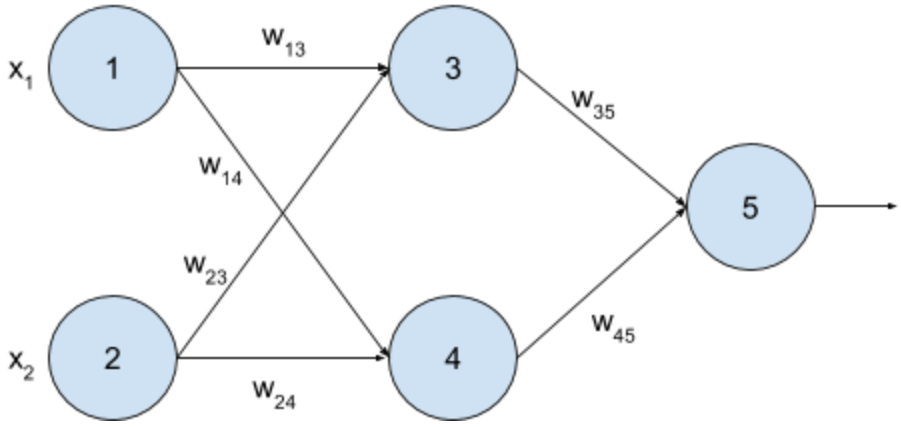
2.1 Suppose you have a fully-connected multilayer neural network with 1 input, 2 hidden and 1 output layers. If your dataset has *p* features, the two hidden layers have 3 and 4 neurons respectively, and the output layer has *k* outputs, calculate the number of parameters in the neural network in terms of *p* and *k*. Assume that the bias terms have not been considered in the specified neurons and need to be added to the parameter count.

2.2 Write down the major steps involved in backpropagation algorithm.

2.3 Given the following multilayer neural network, a training data point $x = (x_1 = 0, x_2 = 1)$, and the target value $T = 1$, please calculate weights and bias after 1 iteration of backpropagation algorithm (show your calculations and fill out the empty tables given below). The learning rate $\eta = 0.8$. The initial weights and bias are in the following table.

w_{13}	w_{14}	w_{23}	w_{24}	w_{35}	w_{45}	θ_3	θ_4	θ_5
-0.3	0.2	0.4	-0.1	-0.2	-0.3	0.2	-0.4	0.1

Multilayer neural network:



The following tables need to be completed:

Net Input and Output Calculations

Unit, j	Net Input, I_j	Output, O_j
3		
4		
5		

Calculation of the error at each node

Unit, j	Err_j
5	
4	
3	

Calculations for weight and bias updating

Weight or Bias	New Value
w_{35}	
w_{45}	
w_{13}	
w_{14}	
w_{23}	
w_{24}	
θ_5	
θ_4	
θ_3	

3. K Nearest Neighbors

Fill in the missing lines of code in KNN.py to implement KNN algorithm by using Euclidean distance as distance measure. Use 5-fold cross validation to calculate the average accuracy on "iris.data" for different values of K.

(1) Report the best value of K obtained.
(2) Show a plot of K value vs. average accuracy over 5-folds. K values must be on x-axis, and average accuracies must be on y-axis. You may try different ranges of K to observe the graph.
(3) Why does your reported best K value give a better average accuracy than extreme K values (using just nearest point i.e. K=1, or using all points in training data as K nearest points i.e. K=120)?
(4) Is there anything you would like to specify about your code? (optional)

Note:

(1) Your code will be graded based on your code implementation (yes, partial grading for the missing lines). The questions for the report above have a separate weightage.
(2) You may observe different best K values for different runs. Just report the one observed from any of the runs. Make sure (1) and (2) are from the same run.
(3) For verification purposes, accuracy for best K will be pretty high for the given dataset.