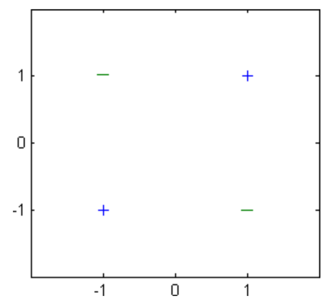
Machine Learning and Data Mining I (DS 4400)
Homework 4

Instructor: Ehsan Elhamifar

Due Date: April 13, 2020, 11:59pm

1. SVM. Consider a supervised learning problem in which the training examples are points in 2-dimensional space. The positive examples (samples in class 1) are $(1, 1)$ and $(-1, -1)$. The negative examples (samples in class 0) are $(1, -1)$ and $(-1, 1)$. Are the positive examples linearly separable from the negative examples in the original space? If so, give the coefficients of $\boldsymbol{w}$.



1. For the example above, consider the feature transformation $\phi(\boldsymbol{x}) = [1, x_1, x_2, x_1x_2]$, where x_1 and x_2 are, respectively, the first and second coordinates of a generic example $\boldsymbol{x}$. Can we find a hyperplance $\boldsymbol{w}^\top\phi(\boldsymbol{x})$ in this feature space that can separate the data from positive and negative class. If so, give the coefficients of $\boldsymbol{w}$ (You should be able to do this by inspection, without significant computation).

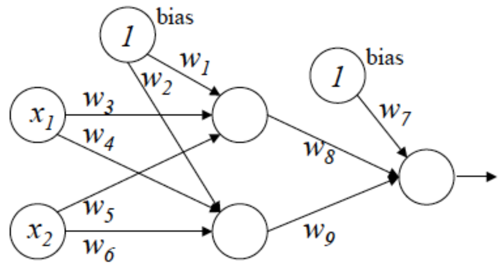
2. What is the kernel corresponding to the feature map $\phi(\cdot)$ in the last part. In other words, provide the kernel function $K(\boldsymbol{x}, \boldsymbol{z}) : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$.

2. Neural Network. Consider a neural net for a binary classification which has one hidden layer as shown in the figure below. We use a linear activation function $a(z) = cz$ at hidden units and a sigmoid activation function $a(z) = 1/(1 + e^{-z})$ at the output unit to learn the function for $P(y = 1|\boldsymbol{x}, \boldsymbol{w})$ where $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{w} = (w_1, w_2, \ldots, w_9)$.

1. What is the output $P(y = 1|x, \boldsymbol{w})$ from the above neural net? Express it in terms of x_i, c and weights w_i. What is the final classification boundary?

2. Draw a neural net with no hidden layer which is equivalent to the given neural net, and write weights $\tilde{\boldsymbol{w}}$ of this new neural net in terms of c and w_i.

3. Is it true that any multi-layered neural net with linear activation functions at hidden layers can be represented as a neural net without any hidden layer? Explain your answer.



3. Feed Forward Neural Network Implementation. For this exercise, you can use PyTorch using which building and training neural networks are more convenient. You can also use Python to fully implement the neural network (it would be much harder). Implement a Feed Forward Neural Network with n layers using either PyTorch, that takes as input the number of layers n and the number of nodes in each layer, S_ℓ for $\ell \in \{1, \ldots, n\}$. The network must receive a dataset $\{\boldsymbol{x}^i, y^i\}_{i=1}^N$, and must output all the learned weights and biasses of all layers as well as the activations of the last layer. The code must allow specifying the following activation functions: sigmoid, hyperbolic tangent, rectifier linear and identity activation. The loss function must be the cross-entropy loss.

4. Testing Algorithms on Data.

(a) Download the dataset of HW04. Plot the two-dimensional data in the training set. Use the training labels (-1 and +1) to plot data in different classes with different colors and markers. Are the data linearly separable?

(b) Run the SVM code from HW03 (with linear kernel and RBF kernel, where for the RBF kernel you need to try a few values of σ) on the training dataset to train and then evaluate the learned model on the test dataset. Report the classification error on the training set and the test set. Plot the data (use different colors for data in different classes) and plot the classification results (points predicted to be in class $+1$ with circles and points predicted to be in class -1 with squares).

(c) Run the NN code (with one hidden layer and with two hidden layers) on the training dataset and evaluate on the test dataset. Here, you need to explore the effect of different number of neurons in the hidden layer(s). Report classification error on the training set and on the test set, when using 10, 30 and 50 neurons in the hidden layer(s). Plot the data (use different colors for data in different classes) and plot the classification results (points predicted to be in class $+1$ with circles and points predicted to be in class -1 with squares).

Homework Submission Instructions: Please submit both the analytical part and the programming part of the homework via email by the DEADLINE. To submit, please send an email to the instructor and cc the TA.

– The title of your email must be “DS4400: HW04:Your-Last-Name”.

– Please attach a single zip file to your email that contains the pdf of your analytical solution, all python codes and plots and a readme file on how to run your files.

– Please name your zip file as “HW04:Your-Last-Name”.