Linear Optimization – Homework 1

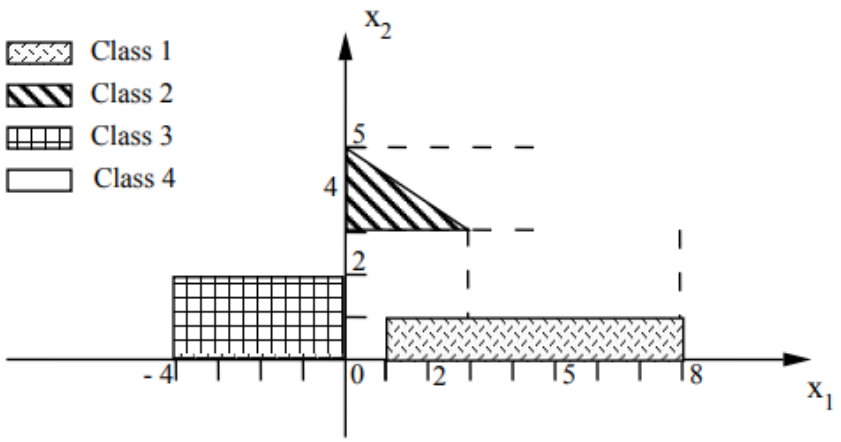
Due: Sunday, April 1

Instruction: Write a report and complete code. Zip and upload them to ftp.

- Upload:
  - Address: 10.13.72.84
  - Username: opt Passwd: opt18 Port: 21
- Download:
  - Address: 10.13.72.84
  - Username: opt Passwd: opt18 Port: 21

Problem 1. A McCulloch-Pitts (M-P) neuron accepts bipolar input $x_i \in \{-1, 1\}$ and gives $y = \text{sgn}(\sum_{i=1}^{m} w_i x_i + b)$.
Give weights and bias for a M-P neuron with inputs $x, y, z \in \{-1, 1\}$ and whose output is z if $x = -1$ and $y = 1$, and is -1 otherwise.

Problem 2. The following figure shows the decision regions of four classes. Design a classifier for these linearly inseparable classes, using a network consists of M-P neurons with three output units. For class i $(1 \leq i \leq 3)$, classification requires that $y_i = 1$, with $y_j = -1$ for $j \neq i$. Class 4 is recognized when $y_i = -1$ for $1 \leq i \leq 3$.



Problem 3. MLP and BackPropogation:

- Download "hw2.zip", The code pass test under Python 2.7.14 and Python 3.6.3. Complete code surrounded by "TODO" in "net.py" and "opt.py". For example,

  ##

```
# TODO: Implement the affine forward pass. Store the result in out. You   #
# will need to reshape the input into rows.                               #
###########################################################################


###########################################################################
#                             END OF YOUR CODE                            #
###########################################################################
```

- Run "python main.py" and paste the results to your report. Make sure your code pass gradient check, *i.e.*, relative error between Numerical gradient and analytic gradient is small, *q.g.* smaller than 1e-7. We use centered formula $df(x)/x = [(]f(x+h) - f(x)]/h$ to compute numerical gradient since it has an error on order of $O(h)$ and use relative error as metric.

- Do something extra surrounding the topics in this assignment, using the code you developed. For example, is there some other interesting question we could have asked? Is there any insightful visualization you can plot? We did not use large cifar10 data. You can select some samples in cifar10 or use the dataset in Problem 4 to do real case analysis.

Probelm 4. Predict House Prices: Given 79 explanatory variables describing (almost) every aspect of residential homes, such as the size and the location of the house, please predict the final price of each home.

- The dataset is cleaned to become simple train/test format. You can try to modify the process of data clean.

- Encourage to do model selection. For example, try lasso regression or humble loss.

- Make sure implement by yourself and do not machine learning related library. You almost implement the algorithm in Problem 3.