Foundations of Machine Learning: Assignment 4

1 AdaBoost

1.1

Suppose, in the usual boosting set-up, that the weak learning condition is guaranteed to hold so that $\epsilon_t \leq \frac{1}{2} - \gamma$ for some $\gamma > 0$ which is *known* before boosting begins. Describe a modified version of AdaBoost whose final classifier is a simple (unweighted) majority vote, and show that its training error is at most $(1-4\gamma^2)^{T/2}$.
Note: For k classifiers $\{g_i : \mathcal{X} \to \{-1,+1\} | i = 1, \ldots, k\}$, majority vote means a classifier f that $f(x) = sgn(\sum_{j=1}^{k} g_j(x))$

1.2

(Exercise 6.2 in *Foundations of Machine Learning*) Alternative objective functions.
This problem studies boosting-type algorithms defined with objective functions different from that of AdaBoost. We assume that the training data are given as m labeled examples $(x_1, y_1), \ldots, (x_m, y_m) \in X \times \{-1,+1\}$. We further assume that Φ is a strictly increasing convex and differentiable function over $\mathbb{R}$ such that: $\forall x \geq 0$, $\Phi(x) \geq 1$ and $\forall x < 0$, $\Phi(x) > 0$.
(a) Consider the loss function $L(\alpha) = \sum_{i=1}^{m} \Phi(-y_i g(x_i))$ where g is a linear combination of base classifiers, i.e., $g = \sum_{t=1}^{T} \alpha_t h_t$ (as in AdaBoost). Derive a new boosting algorithm using the objective function L. In particular, characterize the best base classifier h_u to select at each round of boosting if we use coordinate descent.
(b) Consider the following functions: (1) zero-one loss $\Phi_1(-u) = 1_{u \leq 0}$; (2) least squared loss $\Phi_2(-u) = (1-u)^2$; (3) SVM loss $\Phi_3(-u) = \max\{0, 1-u\}$; and (4) logistic loss $\Phi_4(-u) = \log(1 + e^{-u})$. Which functions satisfy the assumptions on Φ stated earlier in this problem?
(c) For each loss function satisfying these assumptions, derive the corresponding boosting algorithm. How do the algorithm(s) differ from AdaBoost?

2 Neural Network

Try to implement a simple fully-connected neural network and backpropagation by yourself. You are required to complete the **Q4: Two-Layer Neural Network** in Stanford cs231n 2016 winter assignment 1. The python code is included in the attachment. The original description can be found at `http://cs231n.github.io/assignments2016/assignment1/`. Although, the total assignment is provided in attachment, you are only required to complete Q4. Submit your codes when you complete.