HW 4

Wednesday, November 17, 2021 5:07 PM

Problem 1:

A:

$$\min_{\vec{w}} \sum_{i=1}^{n} \max_{\vec{w}}(\nabla_{\vec{w}}[0], \nabla_{\vec{w}}[y_i \vec{w}^T x_i]) \rightarrow \min_{\vec{w}} \sum_{i=1}^{n} \max_{\vec{w}}(0, -y_i \overrightarrow{x_i})$$

So we see that the gradient of this function swings between $-y_i\vec{x}_i$ and 0 due to the max() function. If the feature is less that 0 it outputs 0 and if the feature is greater than 0 it outputs the feature. This means the gradient is not continous and therefore it is not smooth.

B:

$w_{t+1} = w_t + \eta y_i \overrightarrow{x_i}$ we have that $\eta = 1 \rightarrow w_{t+1} = w_t + y_i \overrightarrow{x_i}$ For $t = 0\ \overrightarrow{w_t} = \vec{0}$.

Now since the perception algorithm only updates when $\overrightarrow{w_t}$ makes a mistake, and we start with $\vec{0} = \overrightarrow{w_0}$ we can say that for each update. $\overrightarrow{w_t} = \vec{0} + y_i \overrightarrow{x_i} + y_{i+1} \overrightarrow{x_{i+1}} + \cdots$ where i represents elements of the set of failure cases.

For a particular $(y_i, \overrightarrow{x_i})$ if the algorithm runs multiple times before converging the update will be $\overrightarrow{w_t} = y_i \overrightarrow{x_i} + y_i \overrightarrow{x_i} + \cdots$ some a number of times.

Combining both previous steps we see that the full update will be $\overrightarrow{w_t} = \vec{0} + a_i y_i \overrightarrow{x_i} + a_{i+1} y_{i+1} \overrightarrow{x_{i+1}} + \cdots$

Thus for a set of n iterations before convergence and a_i iterations on each $(y_i, \overrightarrow{x_i})$ $\widehat{\overrightarrow{w_t}} = \sum_{i=1}^{n} a_i y_i \overrightarrow{x_i}$ as desired.

C:

Initialize the weight vector by some value;
Immediately classify the success rate of these weights (number of correction predictions vs. number of incorrect predictions.);
store results of classification
Choose a suitable learning parameter as follows:
    Choose a fixed size parameter defined as $\eta = \frac{R}{L\sqrt{T}}$;
    Where R and L are user defined constants and the T value of our stopping counter.
For loop:
    uniformly at random choose a feature vector x from the training set ;
    perform stochastic gradient descent update on the weights with learning parameter ;
    run the classification performance test ;
    if new results are better than old results
        keep new weights ;
    else:
        increase stopping counter by 1 ;
        keep old weights ;

end for loop if (results > acceptance threshold) or (stopping counter > pre-defined constant T)

Expected rate of convergence
We have a fixed constant learning parameter η
$E[L(\overrightarrow{w_T})] - L(\overrightarrow{w^*}) \leq \frac{RL}{\sqrt{T}}$ Where $R \geq \left\|\overrightarrow{w^*}\right\|$