Stat601(Section001): Homework #4

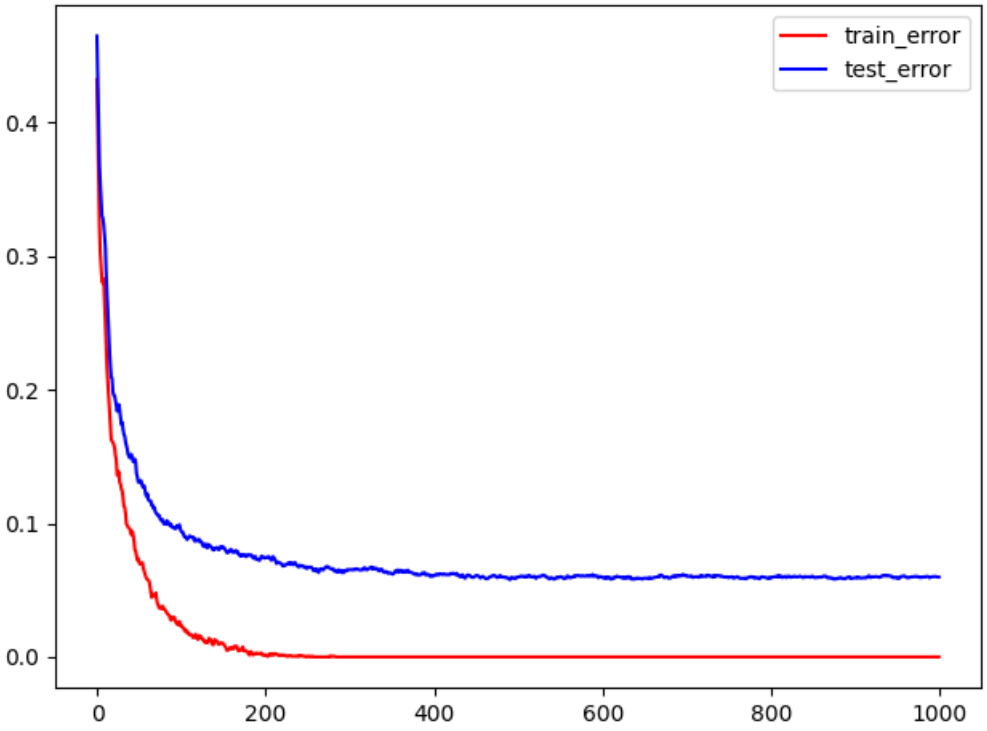
Due on Apr. 6, 2021 at 11:59pm

Instructor:Ziwei Zhu

Tiejin Chen **tiejin@umich.edu**

Problem 1

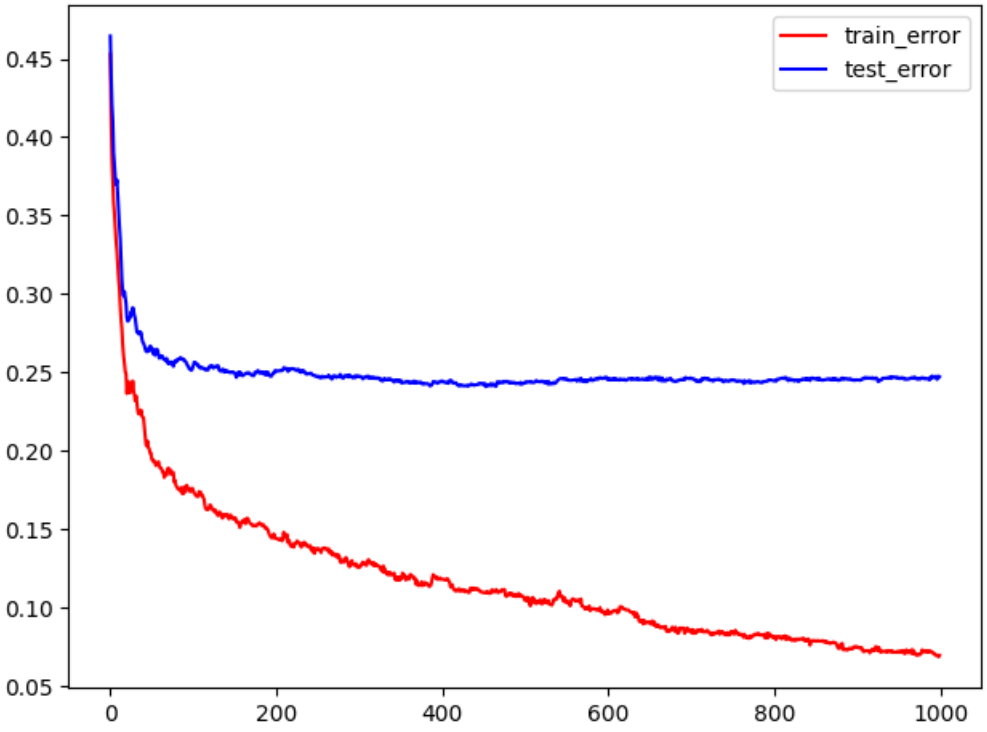
(b)
We can get the plot



where the x-axis is the num of iteration. Before interation around 200, both train and test error decrease as number of iteration increases. We can see the train error become near zero when iteration is around 200. Test error is a little bit larger than train error in all time, and it becomes around 0.07 after 200 iterations.

(c)
It seems that the test error will stablize around 0.07 after iteration 200. And it does not have a huge trend for test error to raise within 1000 iterations.

(d)
We get the plot



We can see that within 1000 interations, the train error always decreases with the increase of iterations. And it will reach near 0.06 around iteration 1000 which is not zero in the previous setting. We can say it is more harder for this setting to train on training set. And we can see the test error shows the similar trend with previous result, while the stabilized test error is much larger than it in (b). The satbilized test error is around 0.25. And we can see this setting has more volatile. Thus it is much harder setting to train and generelize, and causes a bad result.

Problem 2

Proof. Considering forward stagewise additive modeling with exponential loss function. We have:

$$\begin{aligned}(\beta_m, G_m) &= argmin_{\beta,G} \sum_{i=1}^{N} exp(-(y_i(f_{m-1}(x_i) + \beta G(x_i)))) \\ &= argmin_{\beta,G} \sum_{i=1}^{N} exp(-y_i f_{m-1}(x_i)) exp(-\beta y_i G(x_i))\end{aligned}$$

Now consider $y_i G(x_i)$, since $y_i = 1, -1, G(x_i) = 1, -1$. if $y_i = G(x_i), y_i G(x_i) = 1$, otherwise we have $y_i G(x_i) = -1$. Now, for loss function, we have:

$$\begin{aligned}\sum_{i=1}^{N} exp(-y_i f_{m-1}(x_i)) exp(-\beta y_i G(x_i)) &= e^{-\beta} \sum_{y_i = G(x_i)} exp(-y_i f_{m-1}(x_i)) + e^{\beta} \sum_{y_i \neq G(x_i)} exp(-y_i f_{m-1}(x_i)) \\ &= (e^{\beta} - e^{-\beta}) \sum_{i=1}^{N} exp(-y_i f_{m-1}(x_i)) I(y_i \neq G(x_i)) + e^{-\beta} \sum_{i=1}^{N} exp(-y_i f_{m-1}(x_i))\end{aligned}$$

For any fixed value of β, $G_m = argmin_G (e^{\beta} - e^{-\beta}) \sum_{i=1}^{N} exp(-y_i f_{m-1}(x_i)) I(y_i \neq G(x_i)) + e^{-\beta} \sum_{i=1}^{N} exp(-y_i f_{m-1}(x_i))$ is equvalent to $G_m = argmin_G \sum_{i=1}^{N} exp(-y_i f_{m-1}(x_i)) I(y_i \neq G(x_i))$. And we can find that $exp(-y_i f_{m-1}(x_i))$

Problem 2 continued on next page...

is just weight, and we can use $w_i^m = exp(-y_i f_{m-1}(x_i))$ to present them. Then we have $G_m = argmin_G \sum_{i=1}^N w_i^m I(y_i \neq G(x_i))$. Then we plug G_m in loss function to get:

$$\beta_m = argmin_\beta (e^\beta - e^{-\beta}) \sum_{i=1}^N w_i^m I(y_i \neq G_m(x_i)) + e^{-\beta} \sum_{i=1}^N w_i^m$$

$$\frac{\partial (e^\beta - e^{-\beta}) \sum_{i=1}^N w_i^m I(y_i \neq G_m(x_i)) + e^{-\beta} \sum_{i=1}^N w_i^m}{\partial \beta} = (e^\beta + e^{-\beta}) \sum_{i=1}^N w_i^m I(y_i \neq G_m(x_i)) - e^{-\beta} \sum_{i=1}^N w_i^m$$

Let it to be zero, we can have:

$$e^{2\beta} = \frac{\sum_{i=1}^N w_i^m}{\sum_{i=1}^N w_i^m I(y_i \neq G_m(x_i))} - 1$$

let $err_m = \frac{\sum_{i=1}^N w_i^m I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i^m}$. We have:

$$e^{2\beta} = \frac{1 - err_m}{err_m}$$

We take log to get:

$$2\beta_m = log \frac{1 - err_m}{err_m} \rightarrow beta_m = \frac{1}{2} log \frac{1 - err_m}{err_m}$$

Then we plug $f_m(x) = f_{m-1}(x) + \beta_m G_m(x)$ to w_{i+1}^m to get:

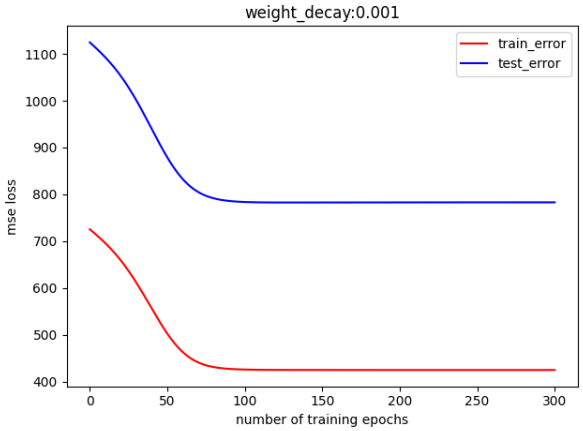
$$w_{i+1}^m = exp(-y_i(f_{m-1}(x) + \beta_m G_m(x))) = w_i^m exp(-y_i \beta_m G_m(x)) = w_i^m exp(\alpha_m I(y_i \neq G_m(x_i))) exp(-\beta_m)$$

where $\alpha_m = 2\beta_m$. And $exp(-\beta_m)$ is a constance, thus it does not affect final result, and we can ignore it. Then all parameter's update rule is the same as adaboost. Thus adaboost is equivalent to forward stagewise additive modeling using the exponential loss function. □

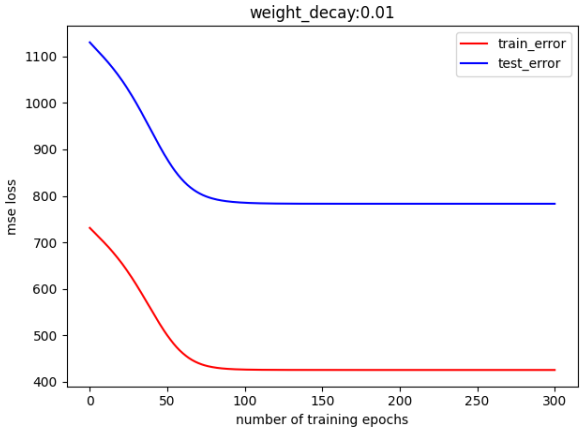
Problem 3

(b)

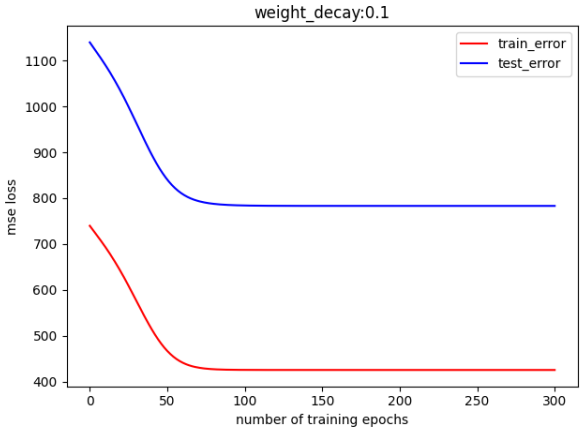
we use learning rate 0.001, And choose weight decay from $[1e-3, 1e-2, 0.1, 0.5, 1, 5, 10]$ to discuss the behavior.And we can get:



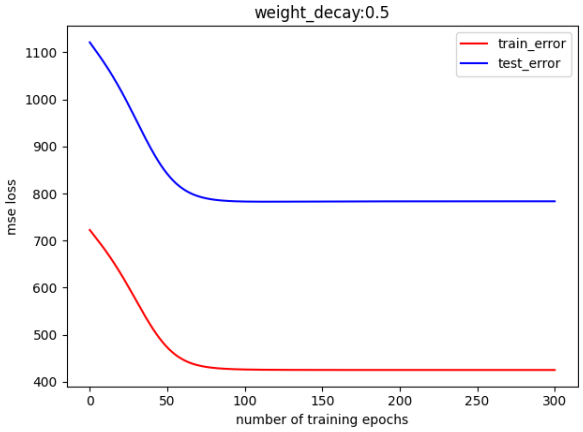
(a) 0.001



(b) 0.01

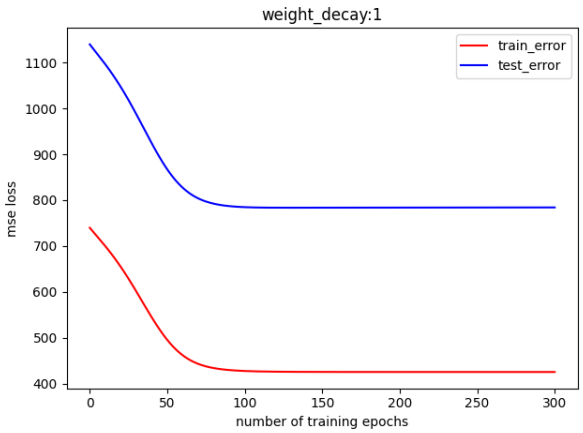


(c) 0.1

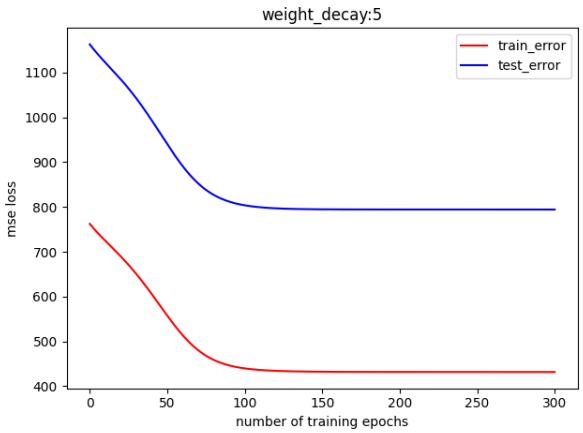


(d) 0.5

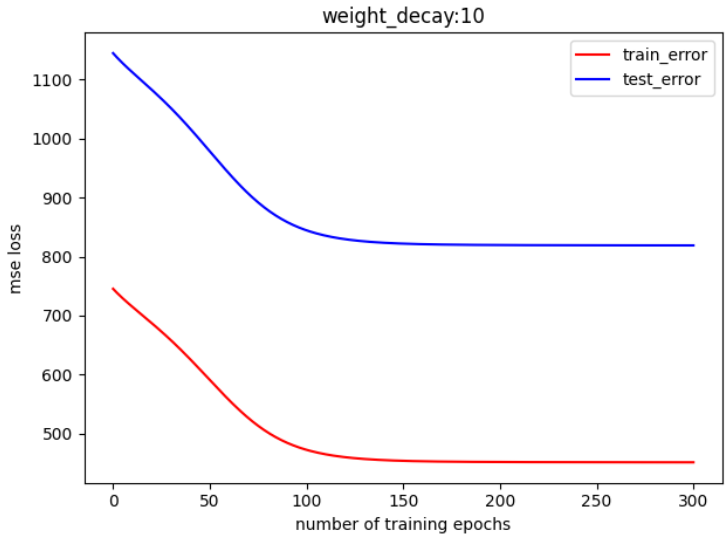
Problem 3 continued on next page. . .



(e) 1



(f) 5

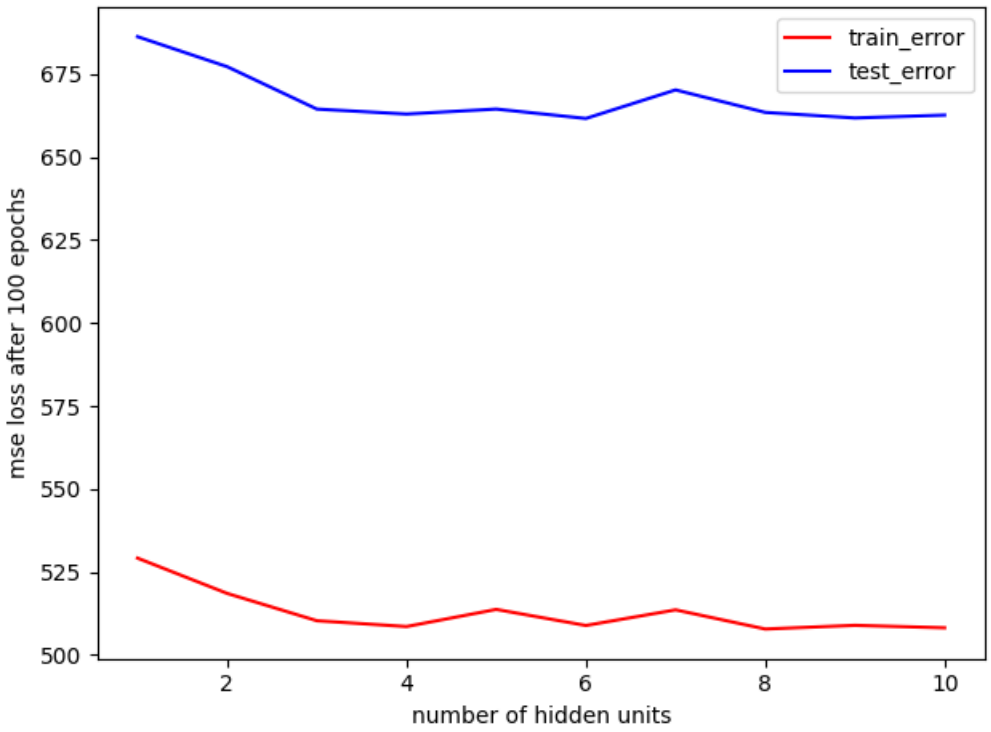


(g) 10

From all the figures, we can see that all of them have similar trends. And we can find that when weight dacay is less than 0.1, as we increase the weight decay,the converge becomes faster. when weight decay is 0.1, it takes about 50 epoches to converges. And also, the test error will decrease for a little bit. But it just a small improvment and difference. However, after 0.1, when weight decay increases, the coverges become slower. When weight decay is 10, model converges after 100 epoches. And both train error and test error are increasing when weight decay increase when it is greater than 0.

(c)

We use $weight\ decay = 0.1, epoch = 100$ here. And we can get the plot



From the figure we can see that when hidden units are less than 3, model will benefit from increasing number of hidden units. Thus minimum number needed to perform well for this task is 3. However, 3 hidden units only performs better a little bit than 1 hidden units .