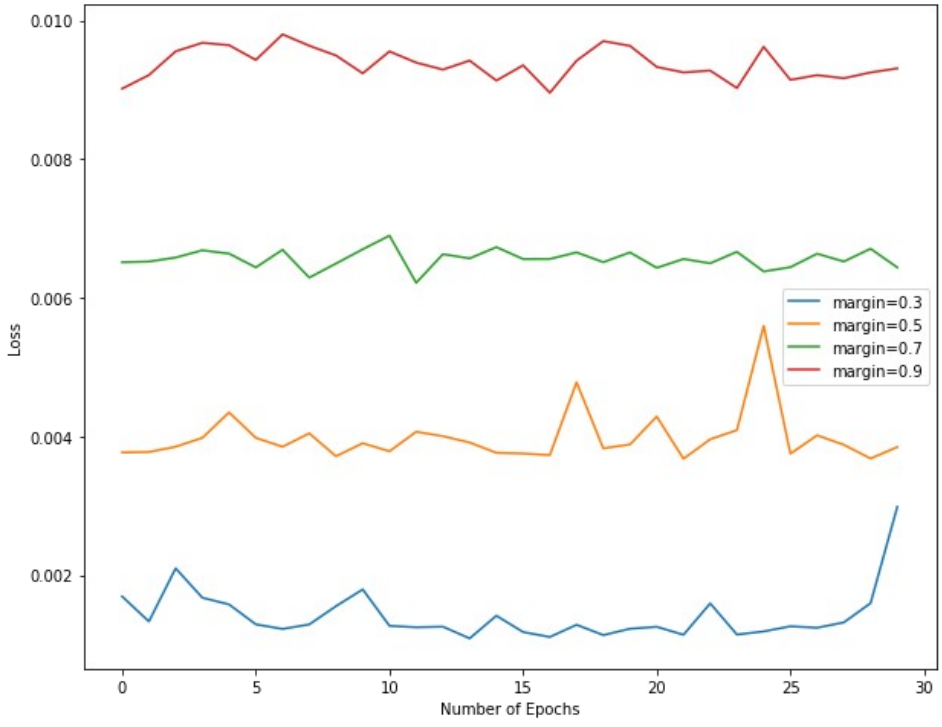
Optimizing AUC directly

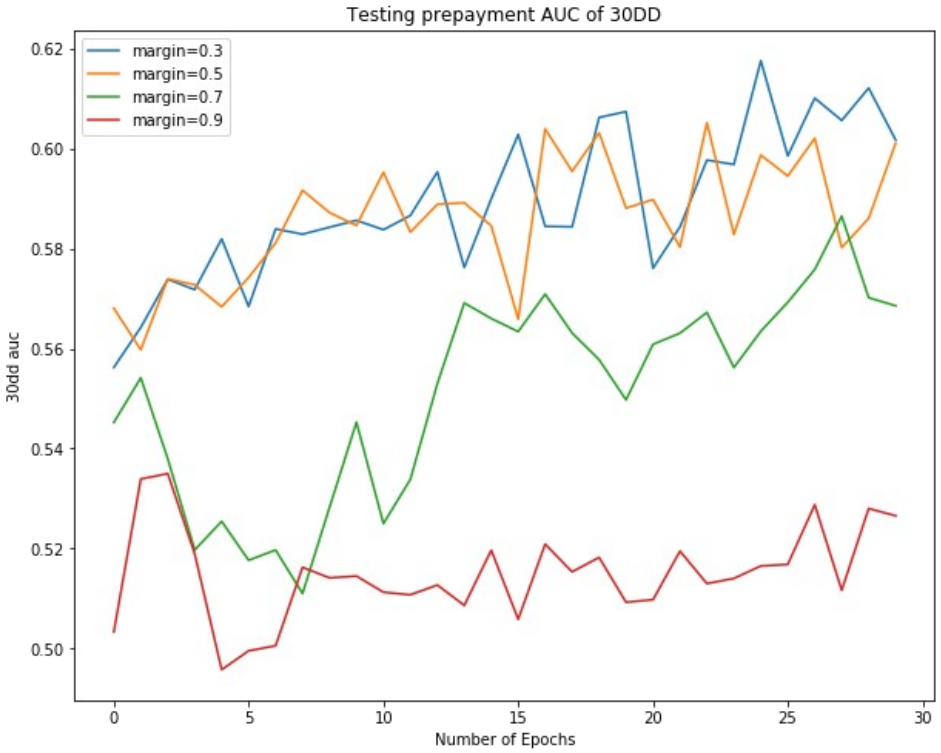
According to this paper [Optimizing Classifier Performance via an Approximation to the Wilcoxon-Mann-Whitney Statistic] I used **MultiMarginLoss** in pytorch to train logistic regression and feed forward network. The results look noisy and the AUC is less compared to that when I trained with cross entropy loss.

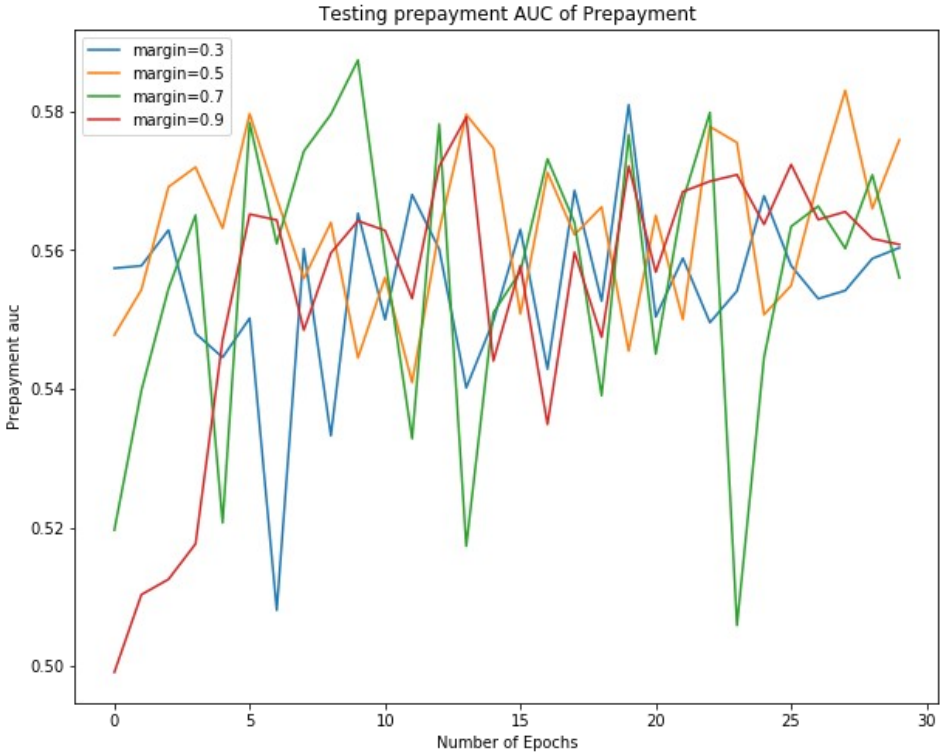
> For each mini-batch sample, the loss in terms of the 1D input *x* and scalar output *y* is:
>
> $$\text{loss}(x, y) = \frac{\sum_i \max(0, \text{margin} - x[y] + x[i]))^p}{\text{x.size}(0)}$$
>
> where *i* == *o* to *x.size*(*o*) and $i \neq y$.

This loss function has one hyper parameter **margin,** we plot the results for different values of margin.

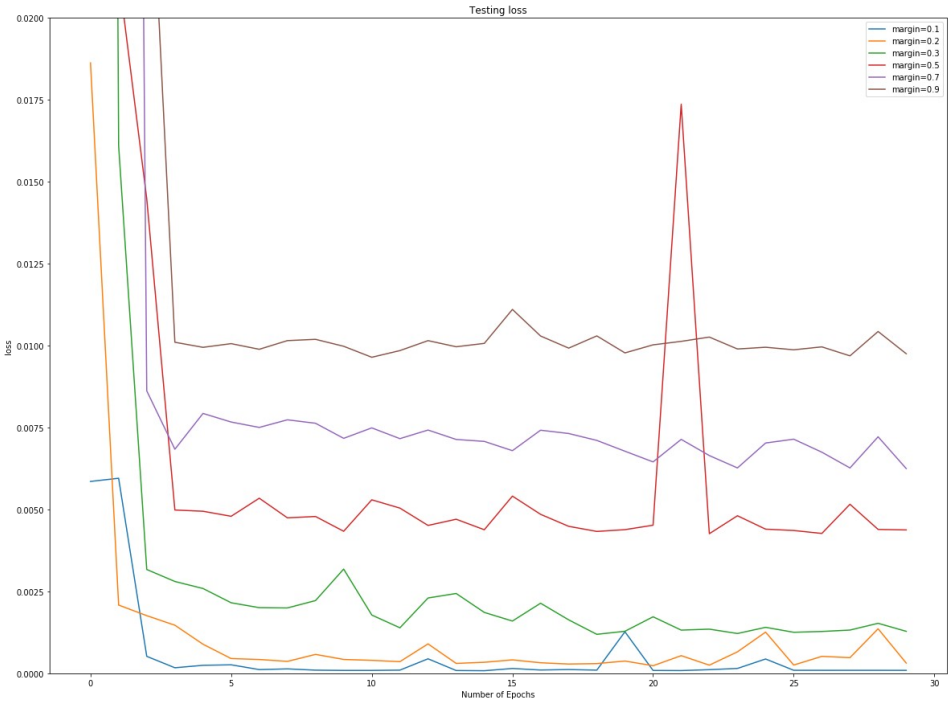
Logistic Regression

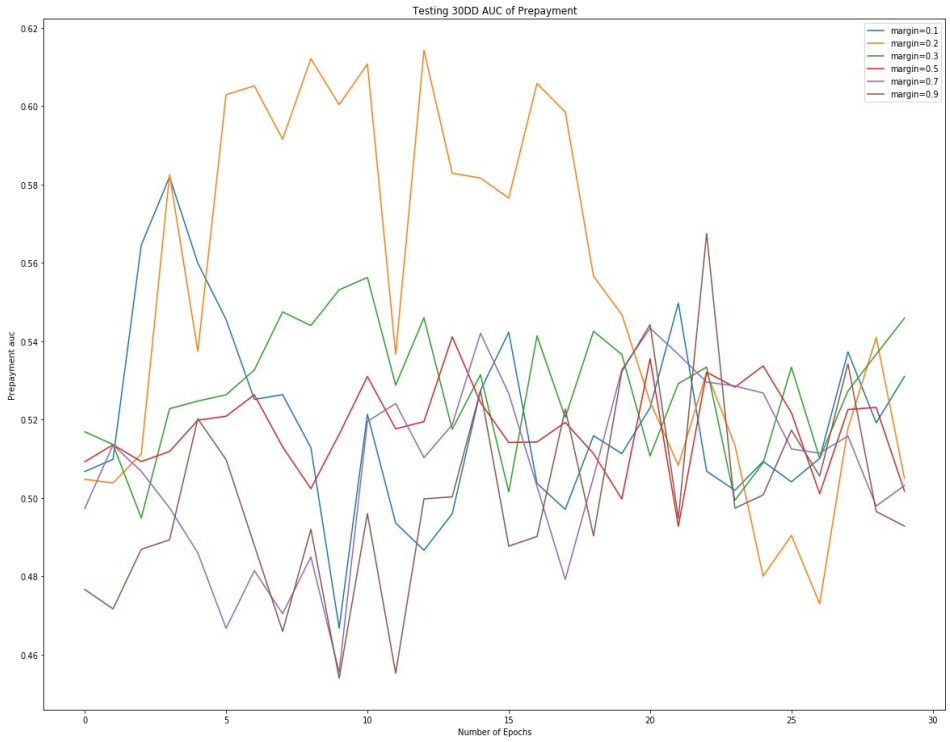


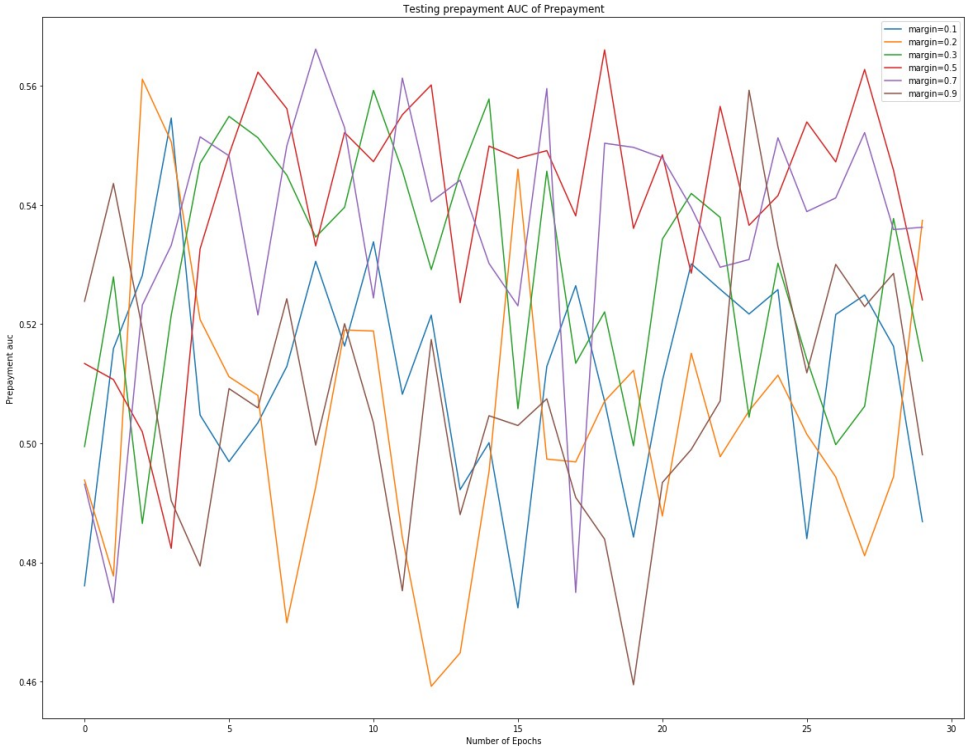

Testing prepayment AUC of 30DD
margin=0.3
margin=0.5
margin=0.7
margin=0.9
30dd auc
Number of Epochs


Testing prepayment AUC of Prepayment
margin=0.3
margin=0.5
margin=0.7
margin=0.9
Prepayment auc
Number of Epochs

Feed forward network






Testing prepayment AUC of Prepayment
margin=0.1
margin=0.2
margin=0.3
margin=0.5
margin=0.7
margin=0.9
Prepayment auc
Number of Epochs
0.56
0.54
0.52
0.50
0.48
0.46
0
5
10
15
20
25
30