Report

We are trying to build a Restricted Bolzmann Model for the ice-cream dataset. We have 120 training data with 10 features.

For every data sample we predict H0, V1, H1 and update the weights according to

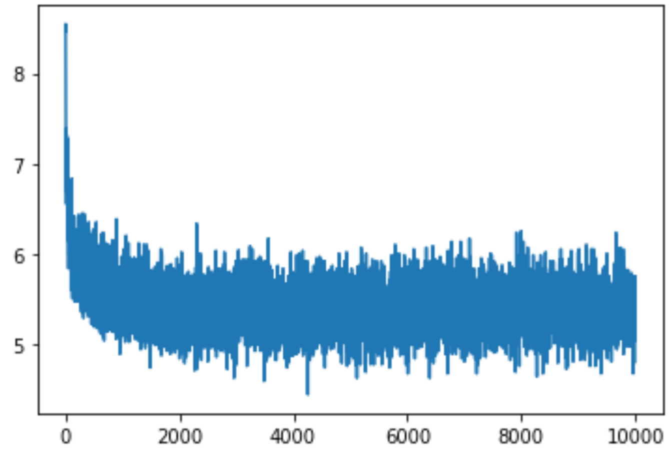
$$w_{ij} \leftarrow w_{ij} + \eta\left(<v_{0i}h_{0j}> - <v_{1i}h_{1j}>\right)$$

Our aim is to get a close match of V1 with V0(input).

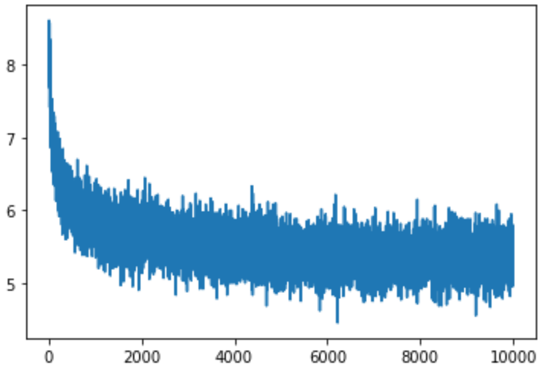
The measure of error is the mean absolute error over the training set.

For 4 hidden nodes I found the following observations of the MAE vs epoch length:

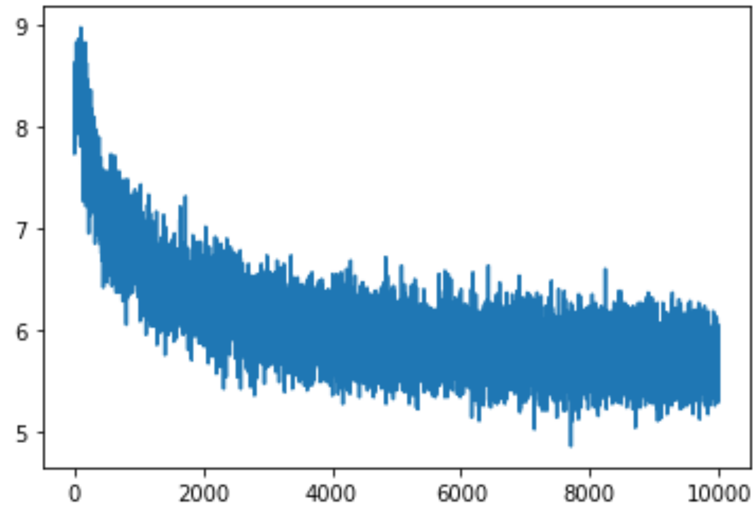
Eta=0.5



Eta =0.1



Eta = 0.01

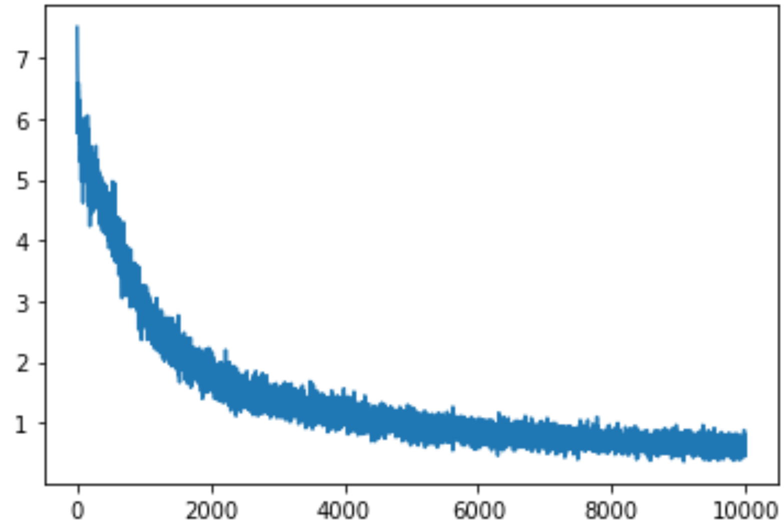


As can be seen from the above graphs, the error is quite high. The above graphs show that out of 10 features, the model is predicting 3 of them incorrectly (since the abs error is 2 for mismatch and 0 for match). Also, with smaller eta convergence is slow.

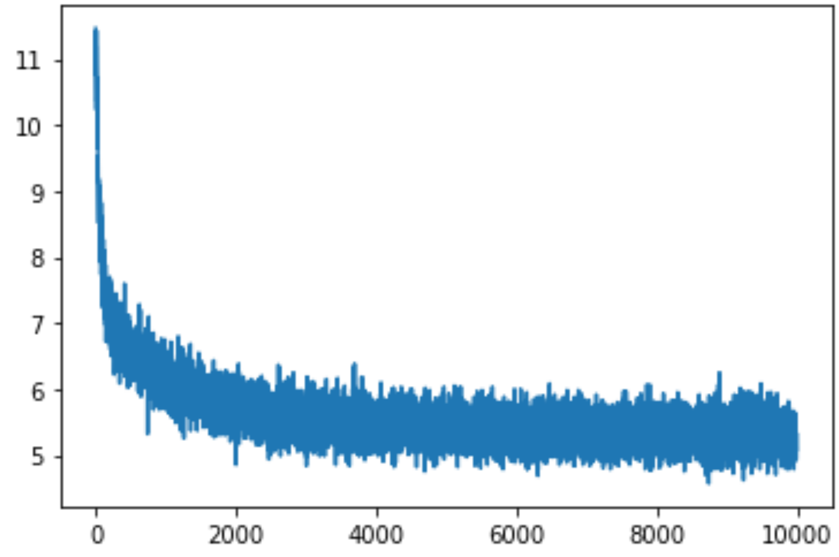
To get a better prediction, I tried increasing the number of hidden units to 15.

This has given close to zero error, which may also point towards overfitting.

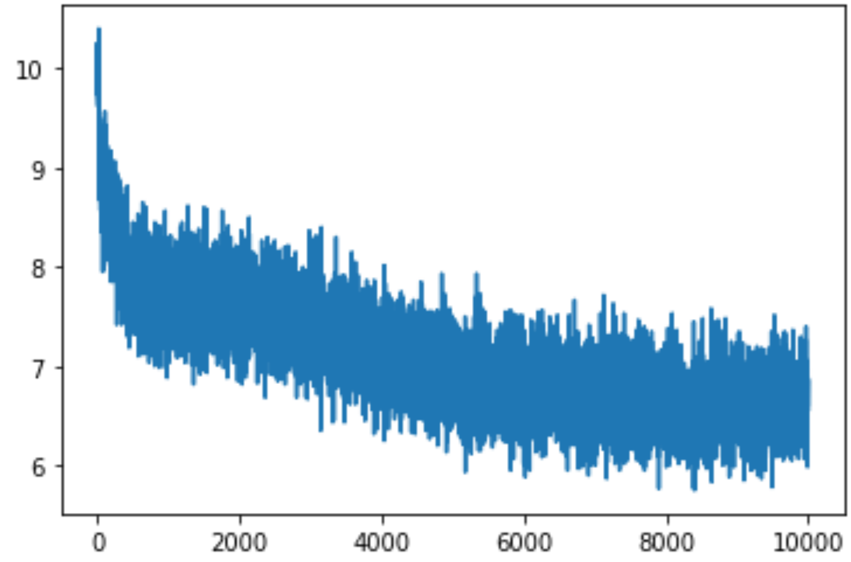
Eta=0.1, hidden nodes = 15



I also tried adding bias to the input layer with 4 hidden nodes but this yielded no improvement:



Increasing the number of iterations in contrastive divergence estimation as the learning progresses also did not yield good results. I went upto 3 hidden layer and observed the following MAE:



The error value has also increased in this case.

Thus, Increasing the number of hidden layers was able to get close to zero error on the training data.

Bonus 1

We are now trying to see if the model can recommend one thing over the other as MAE measures just the ability to reconstruct the test data.

Note: It was a bit confusing to understand if the question asked the probality of only the changed nodes or the overall probability of predicting 1. I have given the below analysis only for the changed nodes.

I split the input data set on 90% training and 10% split set. So from the test set I assign one of the 1 and -1 to 0. I do this randomly at every test epoch length.
Probability of positive instance is calculated as the sigmoid of that index which changed from 1 to 0. Probability of negative instance is calculated as the sigmoid of that index which changed from -1 to 0. Comparing these two gives the count probability of 1 being greater than -1.

For training with 15 hidden nodes, I have the following observations:

Number of times probability of 1 was greater than probability of -1 = 48.75%
Number of times the changed 0 was correctly detected as 1 and coming from 1 = 50.67%
Overall Accuracy of the system = 85%

On test set we have accuracy of 85%, so the taking hidden layer size of 15 is indeed correct.

Thus, we see that by changing one of the inputs we still were able to recover back the original input. The above model thus has a good ability to recommend one item over the other.

Bonus 2

The jester data set has 24983 examples with 100 features.

I split this into 90% train and 10% test data.

I noticed that it takes a lot of time just to train over even 1000 epochs. But the idea here would be similar to that of previous problem. In this data set we have a mix of 1, 0, -1.

For jokes with no rating, the model predicts a rating for that. It’s not clear on what to consider for the accuracy. We can estimate the number of times 1 and -1 was predicted correctly, but 0s could be predicted as 1 or -1 and that is the gist of recommender system. Calculating accuracy for no rating jokes may not be straightforward.

Accuracy would be measure of how many 1s and -1s we got right.