ENGR-E 511 Fall 2018: Assignment #4

Due on Sunday, November 11th, 11:59P

Professor Minje Kim

Abhilash Kuhikar (akuhikar@iu.edu)

December 9, 2018

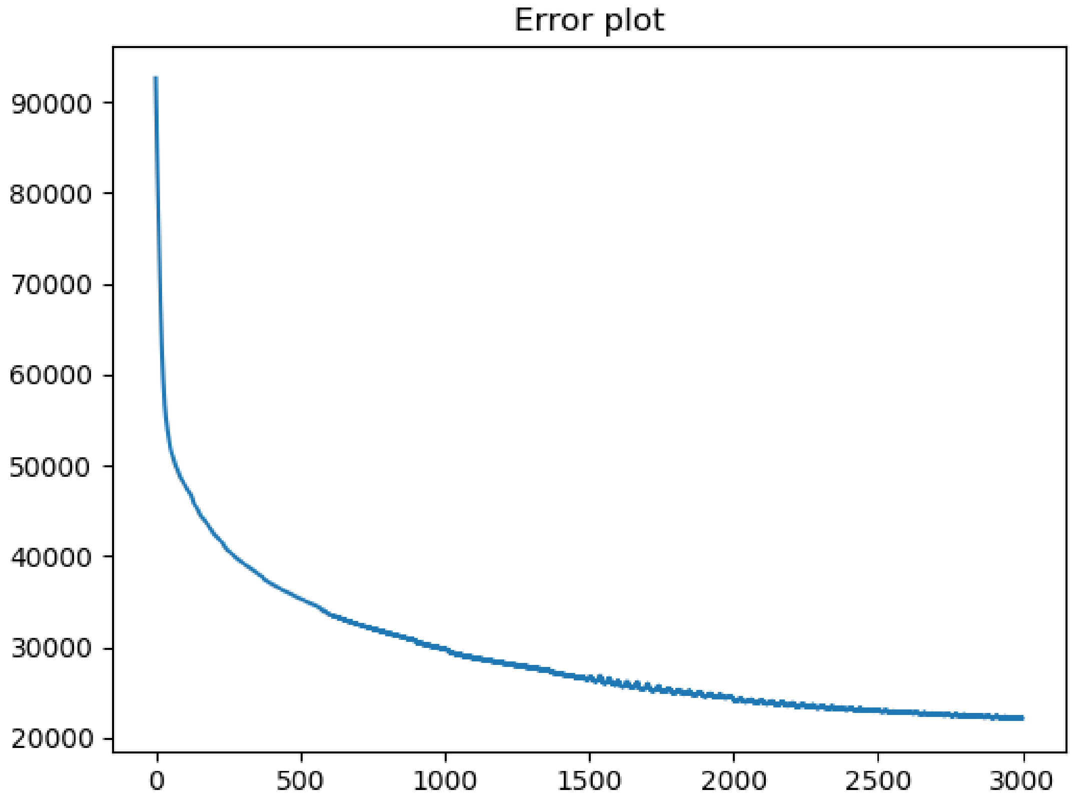
Contents

Problem 1: Neural Network for Source Separation

The objective is to use neural network to separate the sources.

(a) **Neural network architecture**:
I have used the architecture as the one mentioned in the assignment.
an input layer
hidden layer with 80 hidden units (tanh activation function)
Output layer (sigmoid activation function)

(b) **Number of epochs and SNR**
The number of epochs I am currently using is 3000
Here's the error plot for 3000 epochs.



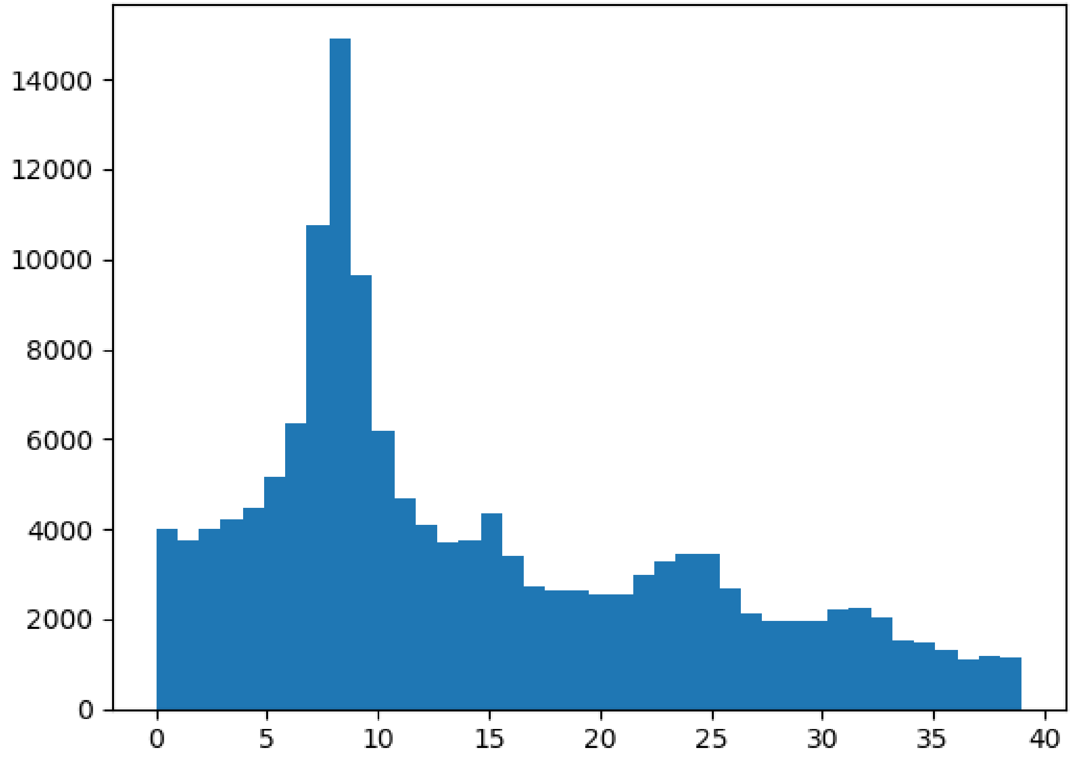
The SNR varies with random initializations of the weights but the best so far I have found was SNR = 6.057038904367814 and now I have set the random seed so that it always gives this SNR.
The learning rate is 0.0005

The recovered speech signal can be found with this submission named recoveredSpeech.wav.

Problem 2: Stereo Matching (revisited)

The goal here is to use the GMM and MRF on disparity map of the displacement of the pixels to find out the depth map in the image.

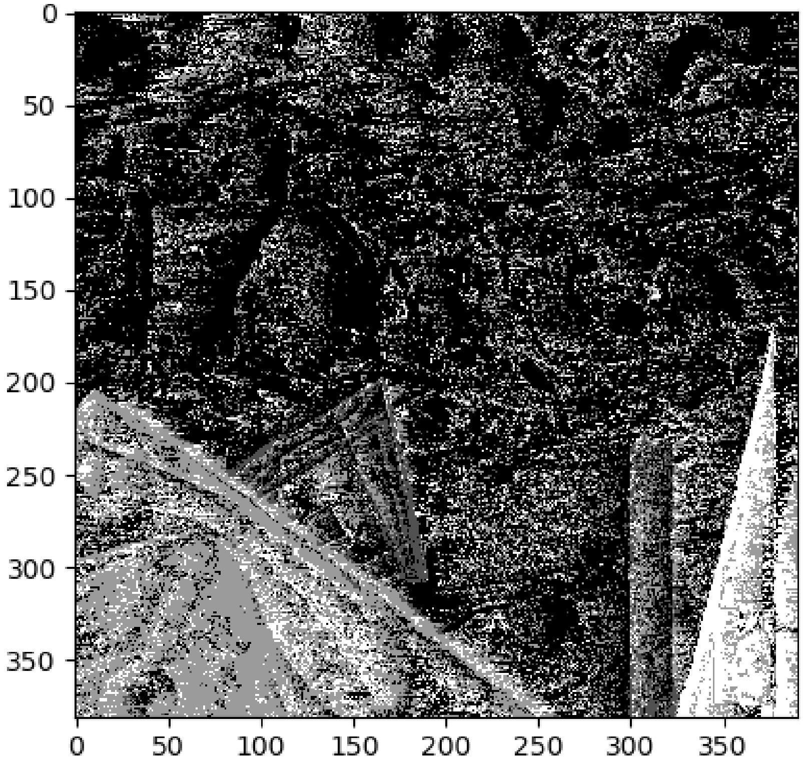
(a) Histogram of the disparities.



Looking at the histograms we can see about 4 peaks may be corresponding to 4 clusters.
I have used 4 clusters to cluster the data in 4 depths(clusters).
The results from GMM using 4 clusters are as follows:
The clusters means and their Sds found from GMM are as follows:
Means: 33.4664468 , 7.23917279, 15.57783642, 23.21886311
SDs: 3.27560671, 3.68598722, 1.71207253, 3.93675652

Problem 2: Stereo Matching (revisited) continued on next page. . .

Here's the disparity map obtained from GMM:

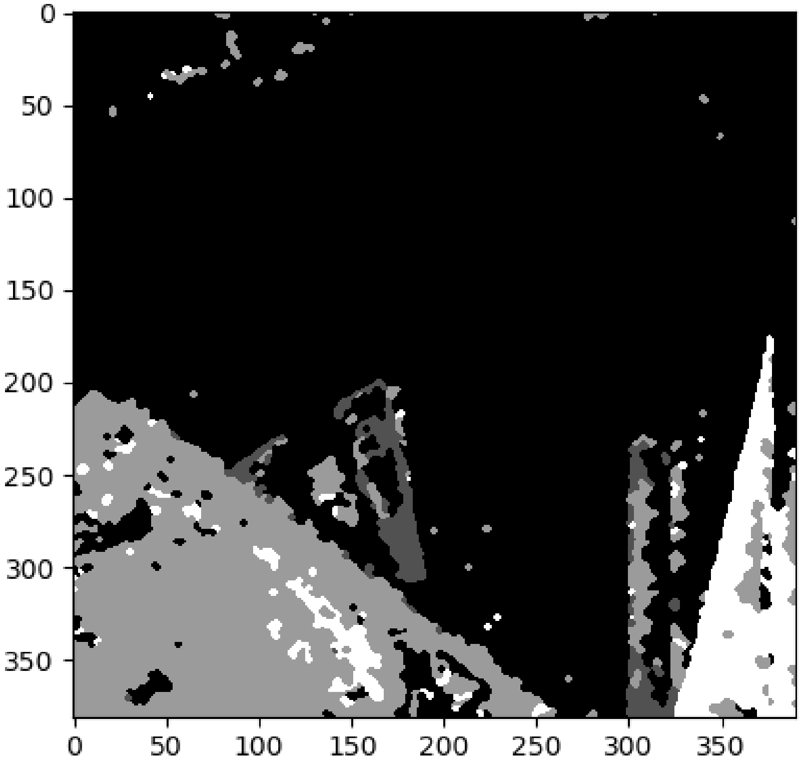


However we can see that there is a lot of noise and this can be smoothened with MRF smoothening. For smoothening I have used Gibbs sampling. However as we sample for each pixel and I couldn't specd much time for optimizing the code, it is taking a long time to create samples.(about 10-15 minutes for 30 samples).

I observed that only after 5-6 samples the depth map was smoothened enough.

I have created 30 samples using Gibbs sampling(12-15 minutes) and then have considered the last 10 samples for calculating the probability(by counting).

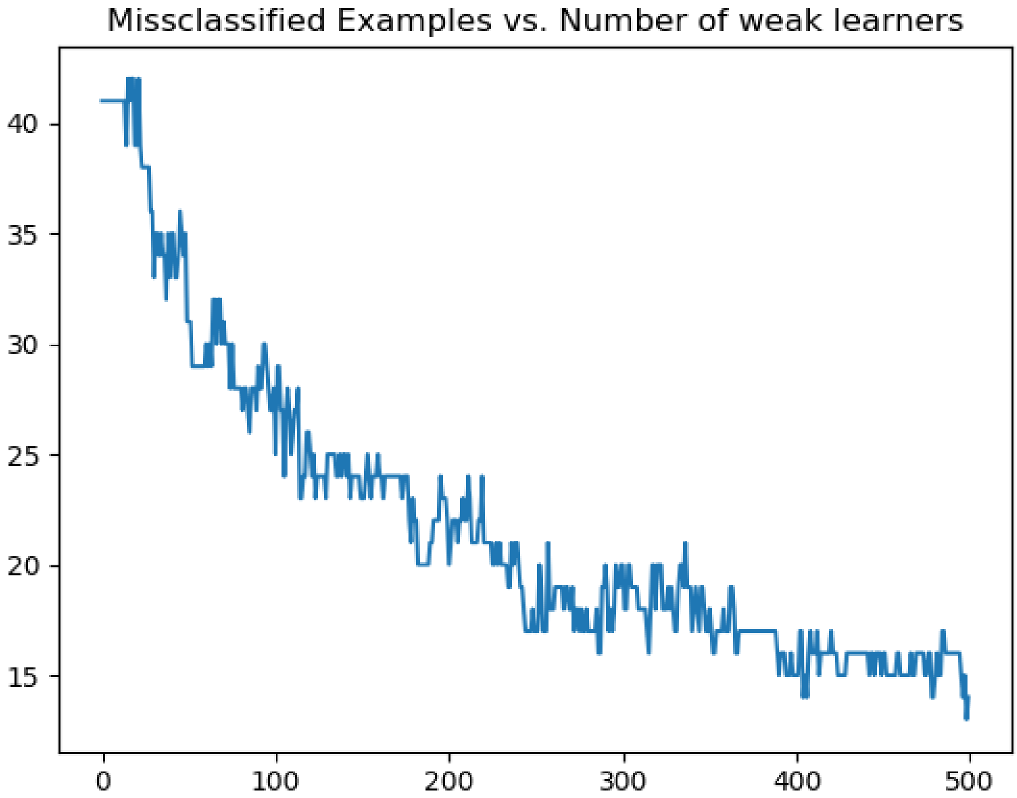
Here's the smoothened depth map of disparity using MRF by Gibbs sampling.


0
50
100
150
200
250
300
350
0
50
100
150
200
250
300
350

Problem 3: Rock or Metal

The goal here is to implement Adaboost using perceptrons as weak learners who will classify the inputs as Rock or Metal depending on the loudness and noisiness of the input signal. The final classifier will be the addition of all these weak learners.

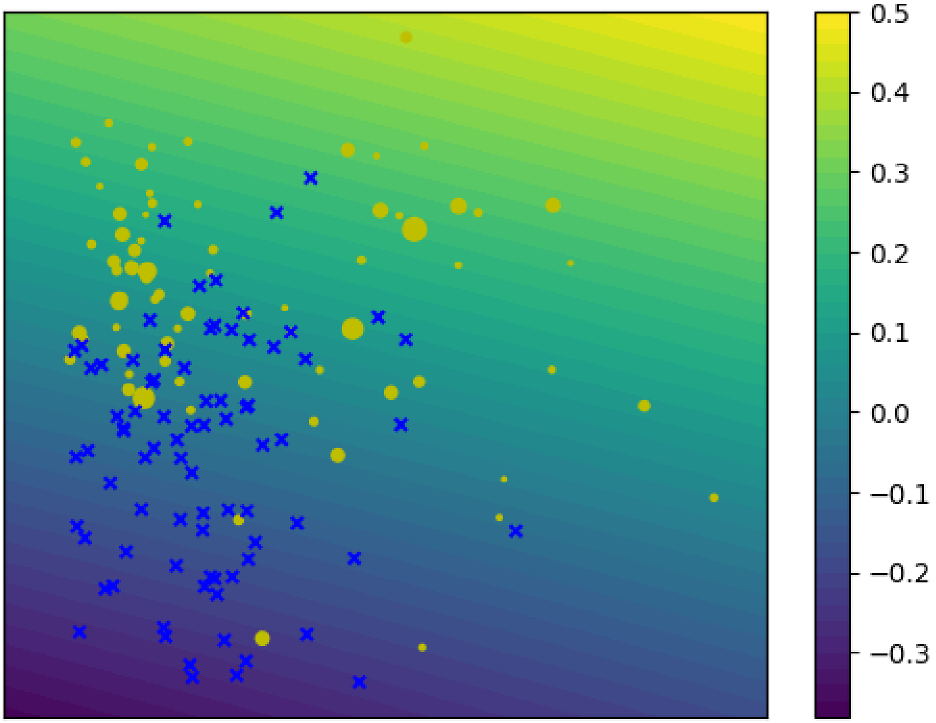
(a) Hyperparameters:
I have trained 500 weak learners(perceptrons) to classify the music as Rock or Metal.
Here's the error plot of the combined classifiers.



We can see that the number of misclassified examples goes down as the number of weal learners increases.
The best training accuracy is 91.875 % with 147 out of 160 examples classified correctly.

Problem 3: Rock or Metal continued on next page...

The plot for adaboost training where the size of the plotted points represents the weights of the examples considering the training from all the weak learners(perceptrons)



PLSI for Analyzing Twitter Stream

The goal here is to use PLSI for analyzing Twitter stream. The training data contains term frequency matrix which are classified as neutral, positive or negative sentiments. We have to learn the PLSI topics and their weight distributions using PLSI

(a) Training and Testing Classification accuracies:
Training Accuracy: 0.6093143596377749
Testing Accuracy: 0.5492227979274611

References