Homework 2

Selma Wanna
SLW3429
slwanna@utexas.edu

February 26, 2019

1 Introduction

Homework 2 focuses on leveraging Independent Component Analysis (ICA) in conjunction with Gradient Descent to solve the Blind Signal Separation (BSS) problem.

BSS involves separating source signals after they have been altered by an unknown mixing matrix: A. In our assignment, five source signals (as sound waves) are provided with 44000 time samples. We then mix these sounds with a randomly generated: A then use ICA to resolve the original signals from the sound mixture.

Given a s by t matrix: S, of original sound signals and a n by t matrix: A, which mixes the original signals by means of randomly generated coefficients, the mixed signal matrix: X is determined by the equation below.

$$X = AS \tag{1}$$

In order to recover the original signal matrix S, we must approximate the reconstruction matrix W as shown below.

$$\hat{S} = WX \tag{2}$$

From the inspection of equations (1) and (2), W should be approximated as A^{-1}; however, because A is an unknown in this problem, ICA is selected to determine W.

The first step in ICA is recognizing that the source signals that make up S are independent from each other. Thus, we can treat the resulting mixed signal's pdf as follows.

$$p(x) = \prod_{i=1}^{n} p_s(w_i^T x \cdot |W|) \tag{3}$$

In Equation (3), w_i^T refers to the rows of matrix W as shown in Fig. 1 on the next page. Equation (4) demonstrates how the i-th source signal can be determined by leveraging the information in the rows of W.

$$s_i = w_i^T x \tag{4}$$

$$W = \begin{bmatrix} — w_1^T — \\ \vdots \\ — w_n^T — \end{bmatrix}.$$

Figure 1: Description of the Reconstruction Matrix: W

Because the pdf: p_s in Equation (3) is unknown, a cdf commonly used in audio applications will be used to derive an appropriate pdf for the ICA model. Equation (5) provides the cdf to be used in Homework 2.

$$g(s) = \frac{1}{1+e^{-s}} \tag{5}$$

Therefore, $g'(s)$ provides the pdf, which is shown in Equation (6).

$$g'(s) = \frac{e^{-s}}{(1+e^{-s})^2} = e^{-s} \cdot g(s)^2 \tag{6}$$

The log of the likelihood function for m samples of n mixed signals is then computed in Equation (7). However, in order to leverage gradient descent, we must take the derivative of $l(W)$ with respect to W to determine our update.

$$l(W) = \sum_{i=1}^{m}(\sum_{j=1}^{n} log(g'(w_j^T x^{(i)})) + log(|W|)) \tag{7}$$

The process for deriving the update step is listed in Equations (8)-(12) below.

$$\frac{\partial}{\partial W} l(W) = \frac{\partial}{\partial W}(\sum_{i=1}^{m}(\sum_{j=1}^{n} log(g'(w_j^T x^{(i)})) + log(|W|))) \tag{8}$$

The partial derivative can be distributed to each log term in the double summation. Equations (9) through (12) will focus on the left log term. For the sake of simplicity, the terms $w_j^T x^{(i)})$ will be represented as s. Thus the derivation begins by substituting Equation (6) into Equation (8).

$$\frac{\partial}{\partial W}(log(e^{-s} \cdot g(s)^2)) = \frac{\partial}{\partial W}(log(e^{-s} + log(g(s)^2)) \tag{9}$$

Simplifying the log expression and then re substituting $w_j^T x^{(i)})$ for s yields Equation (10).

$$\frac{\partial}{\partial W}(-s + 2log(g(s)) = \frac{\partial}{\partial W}(-w_j^T x^{(i)} + 2log(g(w_j^T x^{(i)})) \tag{10}$$

Taking the derivative of the above expression results in the following.

$$x^{(i)} + 2x^{(i)} \frac{g'(w_j^T x^{(i)})}{g(w_j^T x^{(i)})} = x^{(i)} + 2x^{(i)} \frac{e^{-w_j^T x^{(i)}}}{1 + e^{-w_j^T x^{(i)}}} = x^{(i)} + 2x^{(i)} e^{-w_j^T x^{(i)}} g(w_j^T x^{(i)}) \tag{11}$$

Further simplifying Equation (11) results in Equation (12) below.

$$(1 + 2e^{-w_j^T x^{(i)}} g(w_j^T x^{(i)}))X_i^T = (1 - 2g(w_j^T x^{(i)}))X_i^T \tag{12}$$

Next,the derivative of the right log expression is taken.

$$\frac{\partial}{\partial W} log(|W|) = (W^T)^{-1} \tag{13}$$

Now that both partial derivatives have been taken, the two expressions can now be summed in the dimensions of time m and the number of mixed signals n. These sums are consolidated into a matrix representation in Equation (14).

$$\Delta W = ([1 - 2g(WX)]X^T) + (n \cdot m \cdot (W^T)^{-1}) \tag{14}$$

It is apparent from the right hand expression of Equation (14) that the $(W^T)^{-1}$ can easily explode the ΔW evaluation for very large numbers of time samples: m. To combat this, I have multiplied the maximum likelihood summation by a normalization constant of $\frac{1}{M}$. This will be shown later on in the derivation. For now, the next step is to multiply the expression in Equation (14) by $W^T W$.

$$\Delta W = (([1-2g(WX)]X^T) + (n \cdot m \cdot (W^T)^{-1}))W^T W = (([1-2g(WX)])(WX)^T + n \cdot m \cdot I)W \tag{15}$$

Now it is clear from Equation (15) that multiplying by $W^T W$ will make the gradient descent more robust by enabling our algorithm to deal with situations where W is not a square matrix.

Multiplying by the normalization constant: $\frac{1}{M}$ and adding an additional update parameter: the learning rate, results in the following equation.

$$\Delta W = \eta(\frac{1}{M}([1 - 2g(WX)])(WX)^T) + nI)W \tag{16}$$

Now that the update equation has been calculated, the theoretical procedure for determining W is fairly simple and can be observed in Algorithm 1 on the following page.

Algorithm 1: Gradient Descent for Determining Reconstruction Matrix: W

```
Data: $W$, $w$, $X$, $\eta$, $isConverged$
Result: Find W
$W \supset \mathrm{w}_{i,j}$
$w_{i,j} \in \mathbb{R} : 0 \leq \mathrm{w}_{i,j} \leq 5$
$\eta = 0.01$
$isConverged = False$
while not $isConverged$ do
    $Y = WX$
    $Z = \frac{1}{1+e^{-y_{i,j}}}$
    $\Delta W = \eta(I + \frac{1}{M}(1 - 2Z)(WX)^T)W$
    $W = W + \Delta W$
    if $\|\Delta W\| < 0.0001$ then
        $isConverged = True$
    end
end
```

The general solution is listed in Algorithm 1, however, the code used for Homework 2 varied certain aspects of the gradient descent algorithm to ensure an infinite while loop was never met. These methods are explained in following section.

2 Methods

In the Homework 2 assignment, a 5 by 44000 matrix of source signals: U is given. The five rows which span U make up the 5 original source signals in our BSS problem. In the assignment, we generate a random mixing matrix: A, and ensure the columns of the matrix are not greater than its rows, in order to avoid a problem that contains infinite solutions. The mixing matrix is multiplied by U to generate a matrix X of mixed signals. The purpose of the assignment is to determine W, the reconstruction matrix, which will retrieve the original source signals from the matrix X.

Gradient descent is used in conjunction with Independent Component Analysis to find the Maximum Likelihood of the weights in the reconstruction matrix. The procedure is as follows.

First, initialize a guess of W's weights. In my algorithm, I selected the weights to be random numbers in the range [0, 2]. Then create an update loop for the gradient descent algorithm which, calculates ΔW in every iteration, then increments W by ΔW. Algorithm 1 describes this general procedure for determining the weights of W. There is one major difference between that algorithm block and the code implemented for this assignment. Instead of creating a while loop that solely relies on exiting when the system converges, I added an artificial timer in the form of a for loop which ends at some large constant. I typically used 10000. This decision was made for debugging purposes and to ensure I could efficiently complete the assignment before the submission deadline. However, there are possible discrepancies in my graphs as a result of this stopping artifact that will be mentioned in subsections 3.4 and 3.5.

For full access to the code, please see click on the link: ICA GitHub

3 Results

The following section discusses the accuracy of source signal reconstruction using ICA and Gradient Descent. Additionally, the identification task of assigning an original input signal to its reconstructed output signal is addressed. Furthermore, experimentation regarding the learning rate parameter was conducted. The effects of the learning rate on the accuracy of reconstruction and on gradient descent convergence were evaluated.

3.1 Identifying Reconstructed Signals with the Appropriate Sources and Accuracy Metric

By the nature of Independent Component Analysis, the order of the input signals, their scales, and their magnitudes are not known. However, it is still possible to match an input signal with an output signal using external similarity checks. In this paper, the normalized cross correlation metric was used to evaluate input signals with all reconstructed signals. The normalized cross correlation values closest to 1 resulted in an assignment of an input signal to an output. The equation of cross correlation is provided below.

$$(f \star g)[n] = \sum_{m=-\infty}^{\infty} f[\bar{m}]g[m+n] \tag{17}$$

The normalized cross correlation takes the same form, but is then divided by the magnitude of the individual signals.

Once the signals have been properly identified, the accuracy of the reconstruction can be calculated. This is done using the mean squared error (MSE) metric after all signals have been normalized between 0 and 1. The MSE equation is provided in Equation (18) below.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \tag{18}$$

It is important to note, the cross correlation value influences how Equation (18) is performed. If the normalized cross correlation is approximately 1, then Equation (18) is used as written. If the normalized cross correlation is approximated negative 1, that indicates the signal during reconstruction has been flipped across the x-axis. Therefore, before evaluating the match with Equation (18), the reconstructed signal is flipped before calculating the MSE.

3.2 Binary Source Signal Reconstruction

In this subsection, every combination of the five original source signals provided in the homework assignment were mixed together in pairs by randomly generated matrices: A.

The normalized cross correlation metric was used to match and color original signals from the top plot to the bottom plot in Figures 2-8. The top plot shows the original source signals. The middle plot shows the signals after multiplying by the random mixing matrix: A. The bottom signal shows the reconstructed sources. In subsections 3.2 and 3.3 the learning rate was fixed to 0.01.

The accuracies of the reconstructions are summarized in Table 1 on Page 9.

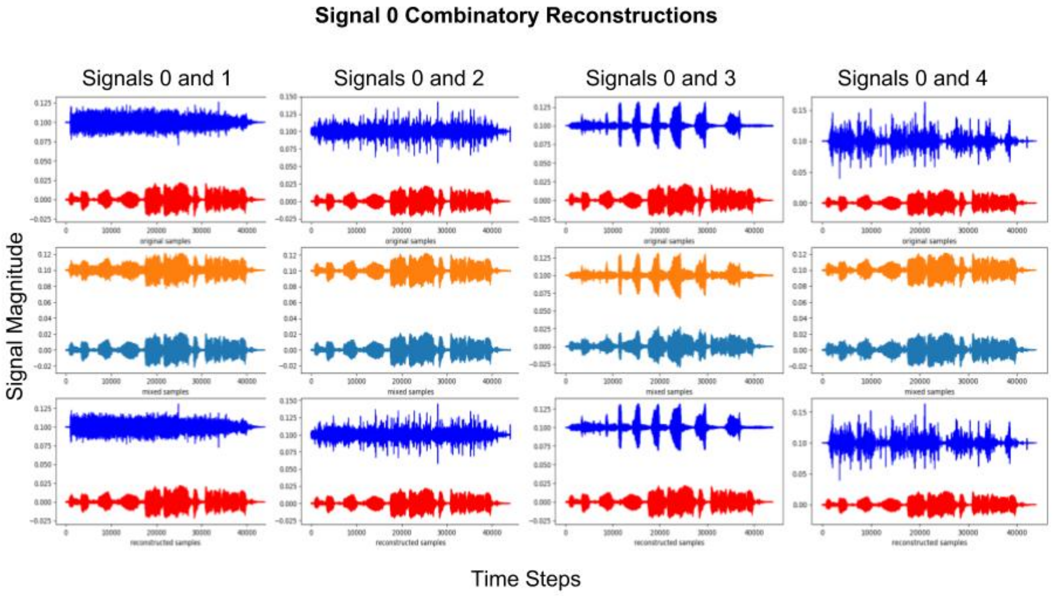

Figure 2: Reconstruction of Signal 0 w.r.t Signals 1, 2, 3, and 4.

In Figure 2 above, we can see that the mixture of Signals 0 and 1 and in the mixture of Signals 0 and 2, the blue signals (signal 1 and signal 2 respectively) are inverted in their reconstruction. This can be expected when using Independent Component Analysis.

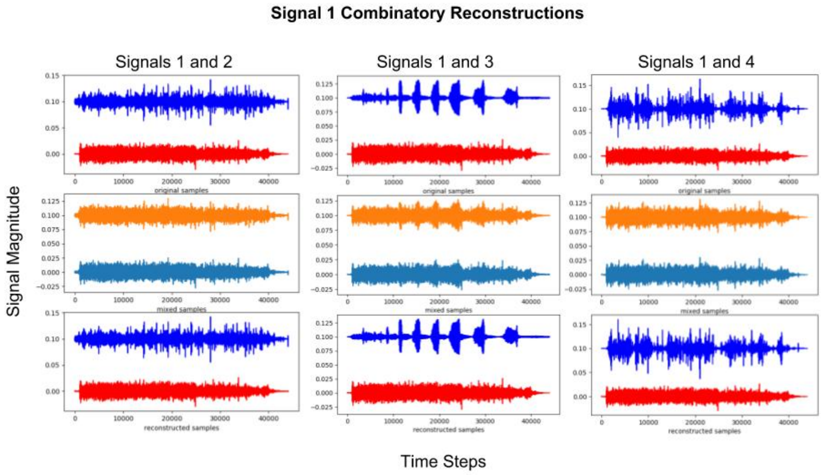

Figure 3: Reconstruction of Signal 1 w.r.t Signals 2, 3, and 4

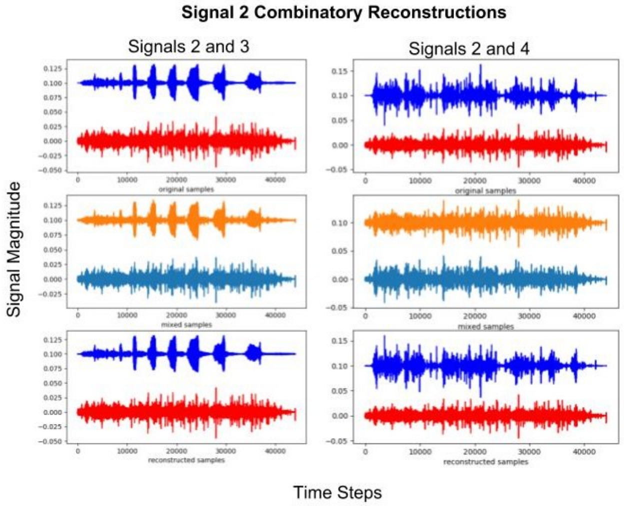

Figure 4: Reconstruction of Signal 2 w.r.t 3 and 4

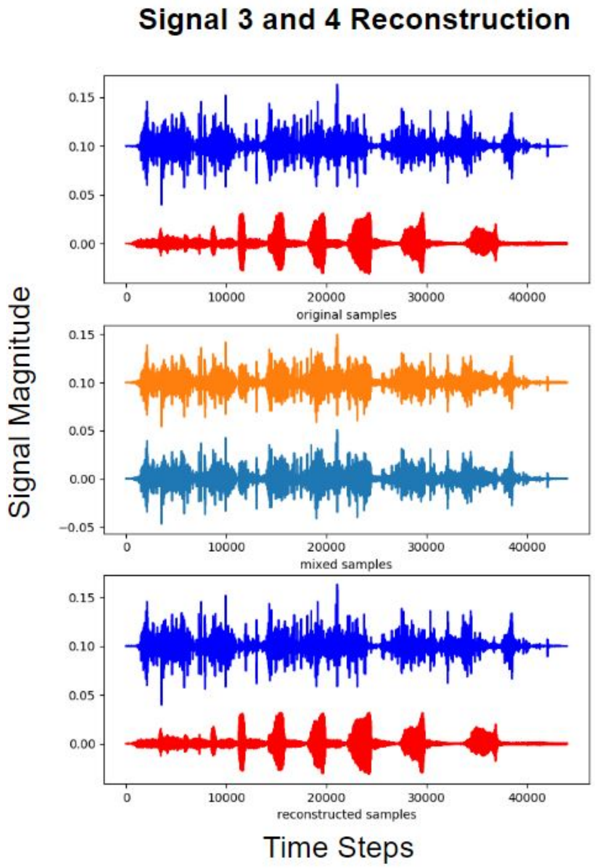

Figure 5: Reconstruction of Signal 3 and 4

Mean Squared Error of Mixed Signal Pairs		
Signal Pair	MSE First Signal	MSE Second Signal
Signal 0 and 1	2.04609e-07	9.06863e-08
Signal 0 and 2	1.40674e-09	9.09086e-08
Signal 0 and 3	2.01816e-08	9.25416e-09
Signal 0 and 4	4.31857e-11	3.73872e-10
Signal 1 and 2	1.36941e-07	1.35549e-07
Signal 1 and 3	2.46275e-10	3.18763e-10
Signal 1 and 4	9.40809e-10	9.09088e-08
Signal 2 and 3	2.67193e-09	1.21463e-08
Signal 2 and 4	1.06596e-08	9.08947e-08
Signal 3 and 4	4.81071e-12	1.48309e-10

Table 1: Mean Squared Error of Signal Pairs

It is important to note for Table 1 above that the mixing matrix A was not kept constant across measurements of accuracy. Therefore, it is difficult to judge which signal mixtures are easier to reconstruct over others. It is also important to note that the Mean Squared Error

(MSE) accuracy measurement is a distance measurement. Given that the magnitudes of the signals are roughly [0, 0.5], the MSE will always look small (see Equation (18)). Generally speaking, the magnitudes of the errors (e-7 or smaller) are still far less than the magnitudes of the signals, thus the overall reconstruction using ICA and Gradient Descent is still very impressive.

3.3 Multiple Source Signal Reconstruction

The same metrics for signal identification, accuracy, and learning rate in the pair mixing in subsection 3.2 are used in this subsection. Figures 6-8 show BSS problems involving three mixed signals, four mixed signals, and five mixed signals respectively. Table 2 on page 11 summarizes the MSE accuracies from the experiments performed.

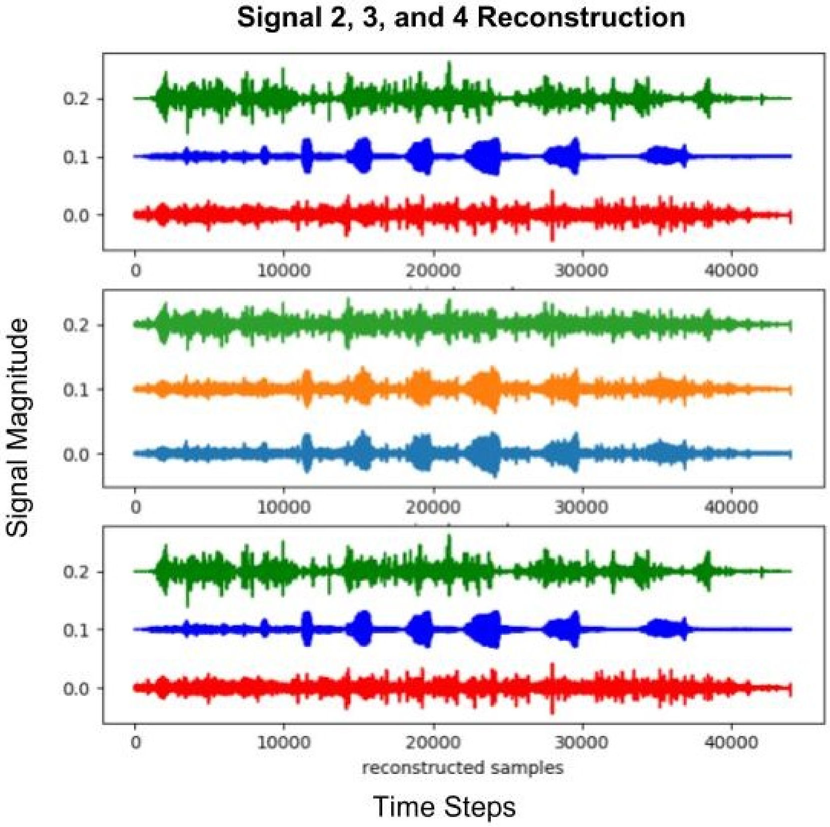

Figure 6: Reconstruction of Signals 2, 3, and 4

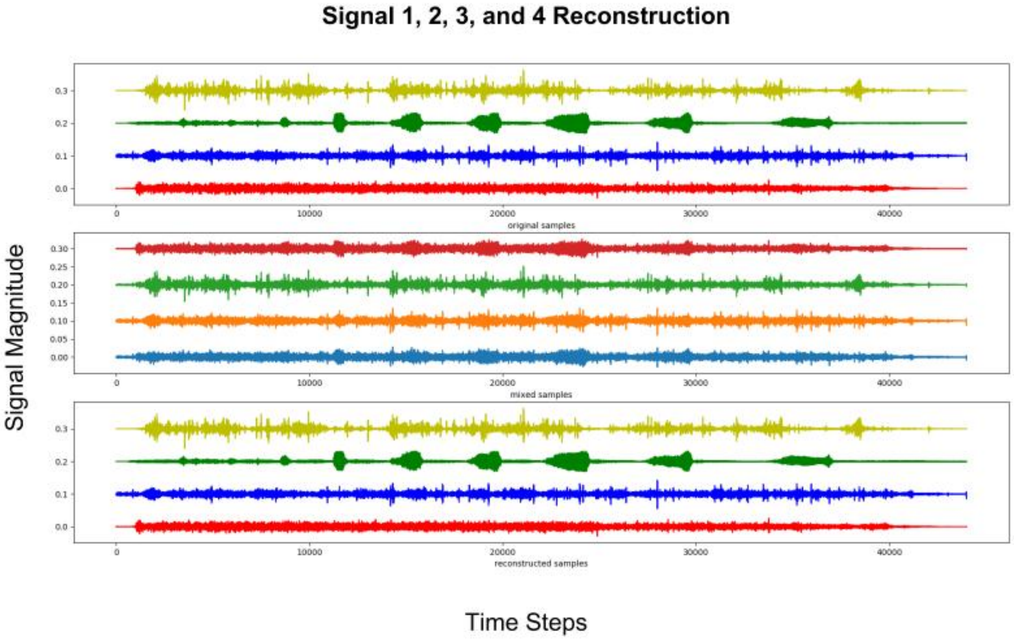

Figure 7: Reconstruction of Signals 1, 2, 3, and 4

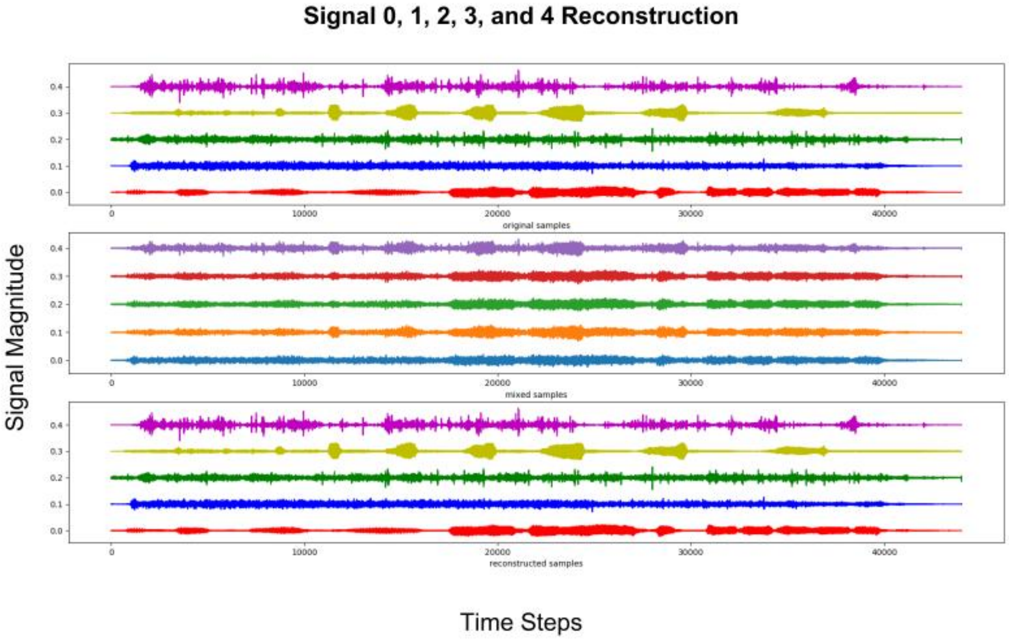

Figure 8: Reconstruction of Signals 0, 1, 2, 3, and 4

Mean Squared Error of Mixed Signal Pairs					
Signal Pair	MSE First Signal	MSE Second Signal	MSE Third Signal	MSE Fourth Signal	MSE Fifth Signal
Signals 2,3, and 4	1.27534e-09	1.48327e-09	1.61670e-09	N/A	N/A
Signals 1,2,3 and 4	2.82484e-08	2.74507e-08	4.96278e-09	6.42436e-09	N/A
Signals 0,1,2,3 and 4	1.50935e-09	3.63869e-08	2.96254e-08	2.80287e-09	2.58715e-09

Table 2: Mean Squared Error of Signal Pairs

Impressively, the mean squared errors for the multiple signal reconstruction stayed within the same order as the MSE for the pair signal measurements. However, the computational effort and ability to reach the optimum likelihood weights for W during gradient descent took much longer when adding additional signals to the mixing problem.

3.4 Learning Rate's Effect on Accuracy (Mean Squared Error)

In this section, the learning rate parameter is adjusted from 10^{-2} to 5. It can be observed in Figure 9 below that the effect of learning rate on the mean squared error is minimal until reaching a learning rate of 3. Once this point is reached, an almost periodic triangle wave is produced.

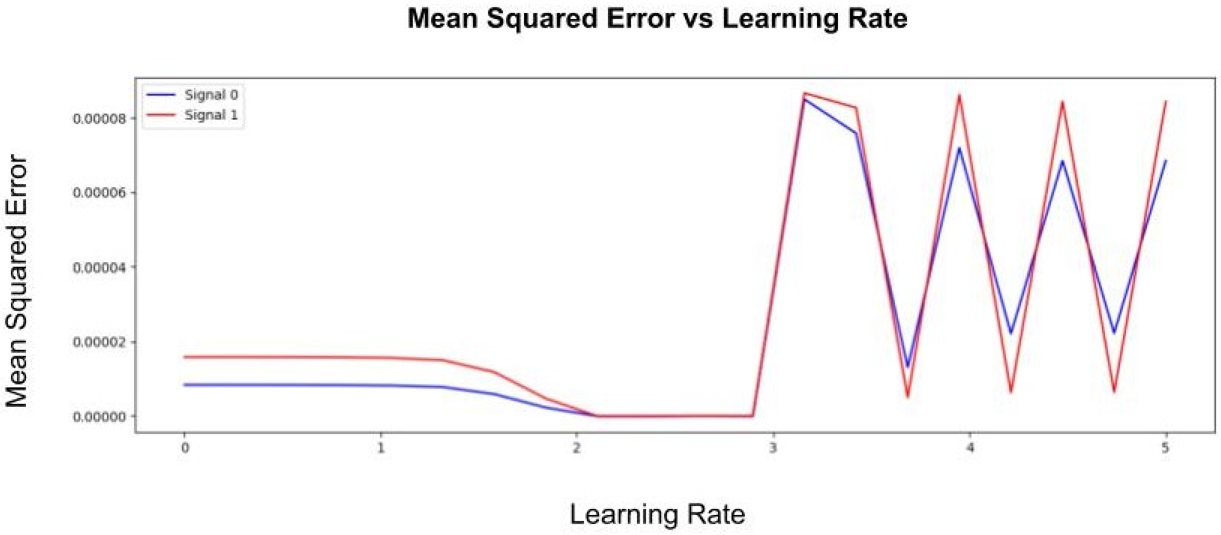

Figure 9: The Effect of the Learning Rate Parameter [0.001, 5] on Mean Squared Error

This graph feature is an artifact of the artificial timer I set in my gradient descent loop. If my model does not converge after 10,000 iterations, my system will automatically return the signals in their current form. In some cases, this form is very close to maximum likelihood guess and produces a MSE minimum. In other cases, the form moves far away from the MSE minima as a result of iterating over a large step size (thus overshooting the maximum likelihood optimum.)

If the for loop time out did not exist, there is a reasonable chance that the system would not converge, making error calculations impossible to perform.

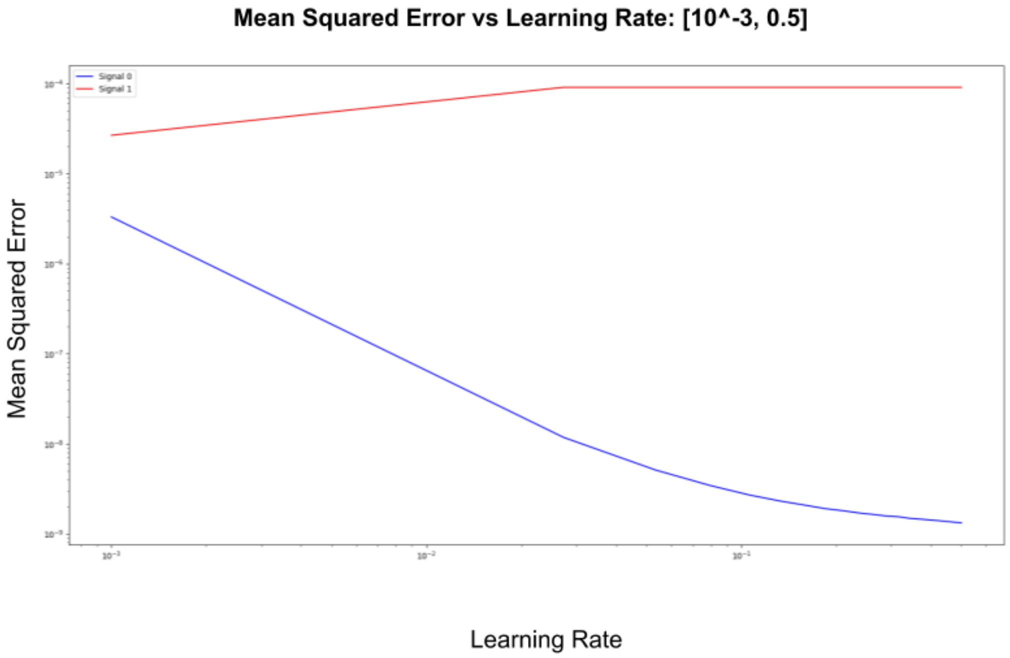

Figure 10: The Effect of the Learning Rate Parameter [0.0001, 0.5] on Mean Squared Error

By narrowing the learning rate parameter's range to [0.0001, 0.5] and by observing the effects of MSE on a log-log plot, we can see that the MSE's of signal 0 and signal 1 respond differently (Figure 10.) Signal 1's error seems to increase with greater learning rates while Signal 0's error seems to decrease. Even though there is an adverse effect on Signal 1's MSE, the change in MSE magnitude is very small compared to the improvement in signal 0's MSE.

Further experimentation revealed that the improvement in MSE magnitude between runs were just as dependent on the initial mixing matrix as the learning parameter value. A collection of MSE vs learning rate graphs are provided in Figures 11 and 12 on the following page.

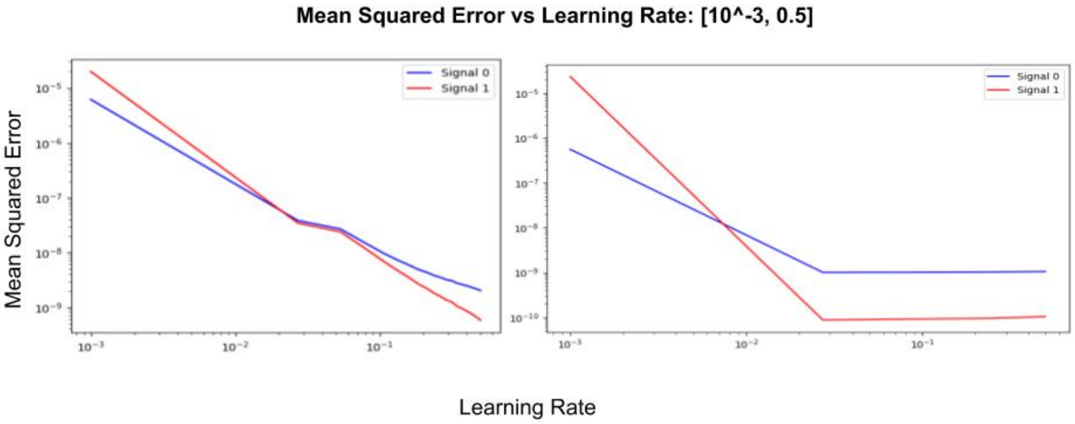

Figure 11: The Effect of the Learning Rate Parameter [0.0001, 0.5] on Mean Squared Error Cont.

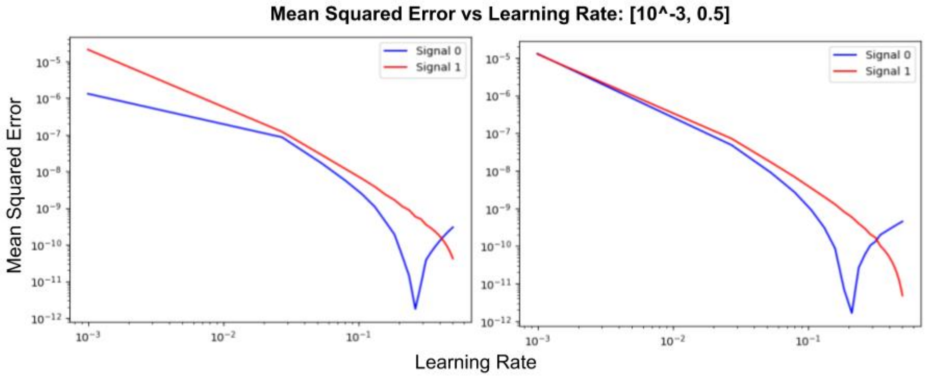

Figure 12: The Effect of the Learning Rate Parameter [0.0001, 0.5] on Mean Squared Error Cont.

It can be determined that within this learning parameter range, that greater learning parameter values will generally decrease the overall MSE in both signals.

3.5 Learning Rate's Effect on Convergence (Number of Update Iterations)

The effect of the learning rate on convergence basically amounts to finding a ”goldilocks” zone. In most stochastic gradient descent problems, the engineering trade off for selecting the learning rate involves minimizing the speed to converge by increasing the learning rate step size, but not massively over shooting the optima when getting closer to convergence.

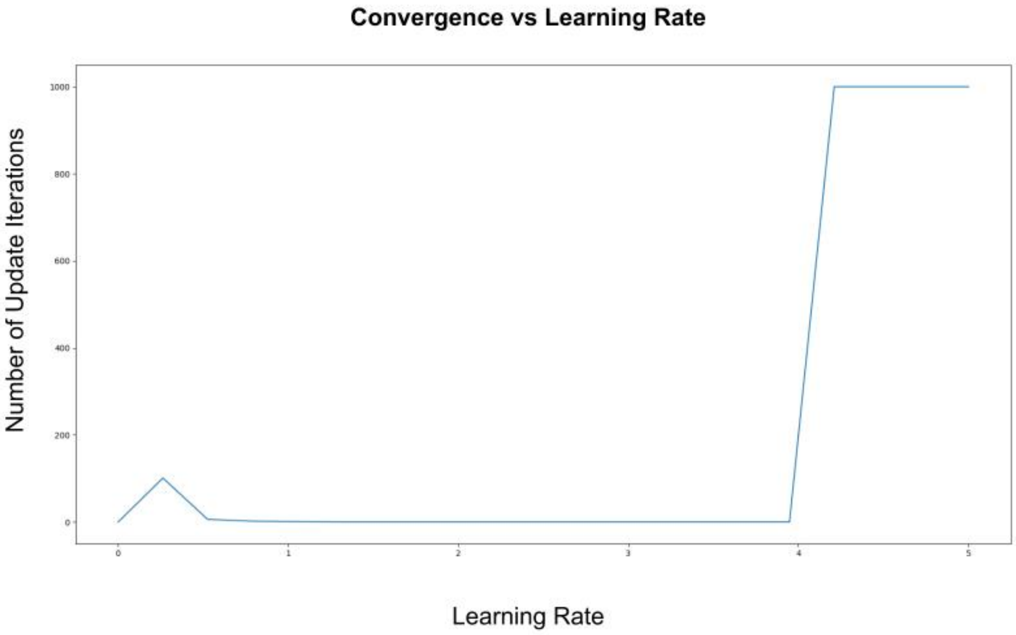

Figure 13: The Effect of the Learning Rate Parameter [0.001, 5] on Convergence

In Figure 13 above, the ”goldilocks” zone is around 0.5 to 3.5. In that range of values, it typically take 5-10 iterations for the system to converge. Once the learning rate surpasses 4, the system does not converge. The plateau at 1000 iterations is due to the for loop timer maxing out at 1000.

4 Summary

To solve the Blind Source Separation problem, Independent Component Analysis and Gradient Descent were used. On average, with a learning rate of 0.01 and a convergence condition of $\|\Delta W\| < 0.0001$, the mean squared error of the reconstructed signals were on the order of 10^{-8}, thus indicating that the overall reconstruction was very good.

By experimenting with the learning rate parameter, it was observed that it effects both the overall errors of signal reconstruction and convergence of the gradient descent algorithm. In both cases, an optimal zone of the learning parameter value exists. This is because a balance must be met between reaching the optimal zone quickly with large learning parameter step sizes versus not constantly overshooting the optimum after quickly getting close to it. It is common in stochastic gradient descent to address this issue by incorporating adaptive learning rates which scale down as you approach the optimum.