Deep Dive 2

Minerva University

CS113: Linear Algebra

Prof. Levitt

March 12, 2023

Deep Dive 3

Concept Exploration 3

Deep Dive problems 5

Part 1 5

Part 2 7

a) Demonstrate that these vectors form an orthogonal basis S of R^8 7

b) Representation in terms of the grayscale image 10

Part 3 12

a)Find the coordinates of v in terms of the new basis S 12

b)thresholds experiments 14

c)noticable changes 16

Reflection 19

Appendix A 20

Deep Dive 3

Concept exploration

A system of linear equations is independent if it has a unique solution, which means that there is only one possible set of values for the variables that satisfy all the equations simultaneously. In the space it means that there is one point where all the lines represented by the linear equations intersect.

A system of linear equations can also be dependent, when it has either no solution or infinitely many solutions. In space, no solution means that the lines represented by the linear equations are parallel to each other, so they will never intersect. Infinitely many solutions mean that the lines are actually merged, so there will be many points (indefinitely many) that all of the lines of the linear system share. Dependency of the equations implies that one or more equations are obtained by a linear combination of the others.

To determine whether S is linearly independent, we need to augment it with a zero vector to make our system a homogeneous matrix.

A homogeneous system of linear equations has the following forms :

$Ax = 0$

where A is the coefficient matrix, x is the vector of variables, and 0 is the zero vector. If we reduce the homogeneous matrix to RREF and the RREF is the identity matrix, it means that the system has a unique solution and is linearly independent so there is only 1 set of variables that satisfy the equations in the system.

As we do this we will have the system of equations in a following form:

$c1 * v1 + c2 * v2 + \ldots + cn * vn = 0$

To determine whether a vector w is in the span of a set of vectors S = {v1, v2, ..., vn}, we need to augment the matrix A that consists of the vectors v1,v2…vn with the vector w. After that we have to solve the system of linear equations that will look as follows:

$c1 * v1 + c2 * v2 + \ldots + cn * vn = w$

where c1, c2, ..., cn are the coefficients of the linear combination of the vectors that add up to ~w. If there exist values of c1, c2, ..., cn that satisfy this system, then ~w is in the span of S. Otherwise, if there are no values of c1, c2, ..., cn that satisfy this system, then ~w is not in the span of S. This process is equivalent to finding the coordinates of ~w in S, which is the set of coefficients that satisfy the equation. If the vectors are linearly independent, then the coefficients c1,c2…cn are the coordinates of ~w in S. If the system has infinitely many solutions, then ~w is in the span of S, but the coordinates may not be unique. In this case, we can express the coordinates of ~w in terms of a the parameters set by the free variables (variables from a non pivot column of the matrix when it is reduced to RREF)

For example, if we have one non pivot column, we express it as a free variable t and the solution for the coordinates of w in S will look as follows:

$c1 = 1 - 2t, c2 = t, c3 = 3t - 1$

which implies that any vector of the form

$w = (1 - 2t)v1 + tv2 + (3t - 1)vn$

is in the span of S, and each value of t corresponds to a different set of coordinates for w in S. Therefore, solving the system of linear equations is an essential step in determining whether w is in S, and finding the coordinates of w in S.

Deep Dive problems

1.Eight standard basis vectors represent the intensity of the white color (the degree of the presence of light) in each of the pixels of the grayscale image. Each vector encodes the information about the level of the light that is present in each of the pixels. The range of values can be determined by the sum of the possible combinations of the color. We have 8 pixels in total. Since we encode information in the binary system, it is represented as bits with the value of either 1 or 0. So we can find the number of all the possible encoding combinations with the following equation:

$C = 8^2 = 256$

Where C denotes the number of all the possible combinations. Now we can determine the range for the values (R) as follows:

$R = [0, 255]$

The first pixel has the value of 31. Since we know the upper bound, we can conclude that it has relatively little light intensity. When we look at the visualization provided, we can confirm the correctness of our reasoning, since the first pixel is very close to black.

Here are the standard vectors given:

$v_1 = [1, 1, 1, 1, 1, 1, 1, 1]$

$v_2 = [1, 1, 1, 1, -1, -1, -1, -1]$

$v_3 = [1, 1, -1, -1, 0, 0, 0, 0]$

$v_4 = [0, 0, 0, 0, 1, 1, -1, -1]$

$v_5 = [1, -1, 0, 0, 0, 0, 0, 0]$

v_6 = [0, 0, 1, -1, 0, 0, 0, 0]

v_7 = [0, 0, 0, 0, 1, -1, 0, 0,]

v_8 = [0, 0, 0, 0, 0, 0, 1, -1]

A linear combination of the standard vectors represents the matrix that is being multiplied by the coordinates to get the light intensity we need.

So the vector

$v = [31, 159, 9, 162, 233, 54, 217, 3]$

Represents the coordinates of the eight standard vectors.

We can rewrite it as:

$v = 31 * v1 + 159 * v2 + 9 * v3 + 162 * v4 + 233 * v5 + 54 * v6 + 217 * v7 + 3 * v8$

So basically the coordinates of the vector v represent the coefficients for each of the standard vectors that result in the light intensity of the image.

2. Consider the eight vectors

$$v_1 = \begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}, v_2 = \begin{bmatrix}1\\1\\1\\1\\-1\\-1\\-1\\-1\end{bmatrix}, v_3 = \begin{bmatrix}1\\1\\-1\\-1\\0\\0\\0\\0\end{bmatrix}, v_4 = \begin{bmatrix}0\\0\\0\\0\\1\\1\\-1\\-1\end{bmatrix}, v_5 = \begin{bmatrix}1\\-1\\0\\0\\0\\0\\0\\0\end{bmatrix}, v_6 = \begin{bmatrix}0\\0\\1\\-1\\0\\0\\0\\0\end{bmatrix}, v_7 = \begin{bmatrix}0\\0\\0\\0\\1\\-1\\0\\0\end{bmatrix}, v_8 = \begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\-1\end{bmatrix}.$$

Figure 1: *the figure represents the eight standard vectors given*

a) **Demonstrate that these vectors form an orthogonal basis S of R^8**

An important note here is that we have to prove the basis to be orthogonal, not orthonormal (thus we do not need to prove the property of the normality of the vectors).

An orthogonal basis of vectors is a set of vectors in which each vector is orthogonal (from 90 degree angle) to all the other vectors in the set. This means that any vector in the space can be expressed as a linear combination of the basis vectors, and the coefficients of this linear combination can be determined using a set of equations.

To demonstrate that these vectors form an orthogonal basis of R^8 we have to prove that all the vectors are orthogonal to each other, thus their dot product is equal to zero.

I used Sage math to prove this property:

```
Dot product of (1, 1, 1, 1, 1, 1, 1, 1) and (1, 1, 1, 1, -1, -1, -1, -1) is 0
Dot product of (1, 1, 1, 1, 1, 1, 1, 1) and (1, 1, -1, -1, 0, 0, 0, 0) is 0
Dot product of (1, 1, 1, 1, 1, 1, 1, 1) and (0, 0, 0, 0, 1, 1, -1, -1) is 0
Dot product of (1, 1, 1, 1, 1, 1, 1, 1) and (1, -1, 0, 0, 0, 0, 0, 0) is 0
Dot product of (1, 1, 1, 1, 1, 1, 1, 1) and (0, 0, 1, -1, 0, 0, 0, 0) is 0
Dot product of (1, 1, 1, 1, 1, 1, 1, 1) and (0, 0, 0, 0, 1, -1, 0, 0) is 0
Dot product of (1, 1, 1, 1, 1, 1, 1, 1) and (0, 0, 0, 0, 0, 0, 1, -1) is 0
Dot product of (1, 1, 1, 1, -1, -1, -1, -1) and (1, 1, -1, -1, 0, 0, 0, 0) is 0
Dot product of (1, 1, 1, 1, -1, -1, -1, -1) and (0, 0, 0, 0, 1, 1, -1, -1) is 0
Dot product of (1, 1, 1, 1, -1, -1, -1, -1) and (1, -1, 0, 0, 0, 0, 0, 0) is 0
Dot product of (1, 1, 1, 1, -1, -1, -1, -1) and (0, 0, 1, -1, 0, 0, 0, 0) is 0
Dot product of (1, 1, 1, 1, -1, -1, -1, -1) and (0, 0, 0, 0, 1, -1, 0, 0) is 0
Dot product of (1, 1, 1, 1, -1, -1, -1, -1) and (0, 0, 0, 0, 0, 0, 1, -1) is 0
Dot product of (1, 1, -1, -1, 0, 0, 0, 0) and (0, 0, 0, 0, 1, 1, -1, -1) is 0
Dot product of (1, 1, -1, -1, 0, 0, 0, 0) and (1, -1, 0, 0, 0, 0, 0, 0) is 0
Dot product of (1, 1, -1, -1, 0, 0, 0, 0) and (0, 0, 1, -1, 0, 0, 0, 0) is 0
Dot product of (1, 1, -1, -1, 0, 0, 0, 0) and (0, 0, 0, 0, 1, -1, 0, 0) is 0
Dot product of (1, 1, -1, -1, 0, 0, 0, 0) and (0, 0, 0, 0, 0, 0, 1, -1) is 0
Dot product of (0, 0, 0, 0, 1, 1, -1, -1) and (1, -1, 0, 0, 0, 0, 0, 0) is 0
Dot product of (0, 0, 0, 0, 1, 1, -1, -1) and (0, 0, 1, -1, 0, 0, 0, 0) is 0
Dot product of (0, 0, 0, 0, 1, 1, -1, -1) and (0, 0, 0, 0, 1, -1, 0, 0) is 0
Dot product of (0, 0, 0, 0, 1, 1, -1, -1) and (0, 0, 0, 0, 0, 0, 1, -1) is 0
Dot product of (1, -1, 0, 0, 0, 0, 0, 0) and (0, 0, 1, -1, 0, 0, 0, 0) is 0
Dot product of (1, -1, 0, 0, 0, 0, 0, 0) and (0, 0, 0, 0, 1, -1, 0, 0) is 0
Dot product of (1, -1, 0, 0, 0, 0, 0, 0) and (0, 0, 0, 0, 0, 0, 1, -1) is 0
Dot product of (0, 0, 1, -1, 0, 0, 0, 0) and (0, 0, 0, 0, 1, -1, 0, 0) is 0
Dot product of (0, 0, 1, -1, 0, 0, 0, 0) and (0, 0, 0, 0, 0, 0, 1, -1) is 0
Dot product of (0, 0, 0, 0, 1, -1, 0, 0) and (0, 0, 0, 0, 0, 0, 1, -1) is 0
```

Figure 2: *dot product of each of 8 vectors with each other. The figure was produced using sage math. The code can be found in Appendix A.*

Figure 2 demonstrates that the property of orthogonality holds. 8 is the maximum number of vectors that will be orthogonal and linearly independent in R^8. It is impossible to have more orthogonal vectors than the dimensions of the space since in that case the remaining vectors will be just modifications of the existing ones and they will not be independent from each other anymore.

For example, let's consider R^2, a 2 dimensional space (for simplicity). There are only 2 possible arrows that will be orthogonal to each other. There might be a confusion because some people might think that all the arrows on figure 3 are orthogonal. They are. But red figures are produced from the black vectors by rotating them, thus they are the result of the linear combinations of the existing vectors and if we put all 4 vectors in the system, only 2 of them will be linearly independent.

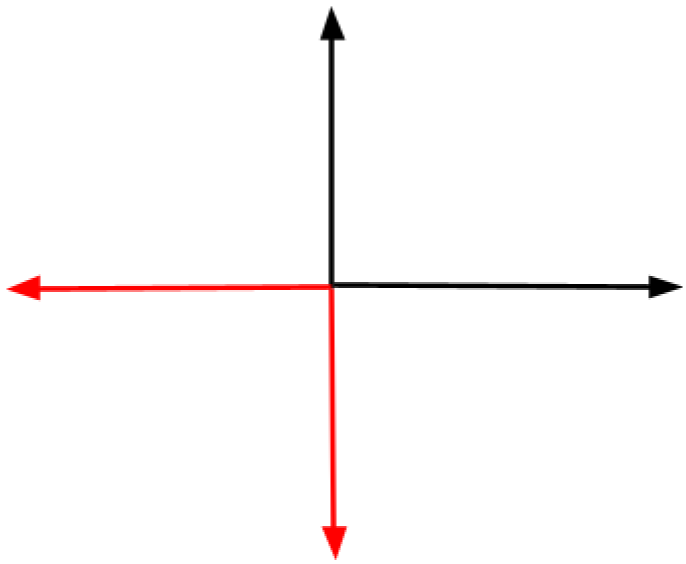

Figure 3*: visual representation of the analogy why it is impossible to have more than n orthogonal vectors in R^n dimensional space and keep the property of independence at the same time.*

b) **What do v1, 100v1 + 50v2, 128v1 − 64v3 + 32v5 − 16v7 represent in terms of a grayscale image?**

The linear combination of vectors represents a grayscale image where each pixel's value is determined by the coefficients of the given vectors. In our case, the vectors represent the brightness of the pixels. By adding or subtracting these vectors, we can create new patterns or adjust existing ones. Adding two vectors results in a new vector that is the sum of their corresponding components, and this new vector does not necessarily represent a combination of the properties of the original vectors but rather forms a new one.

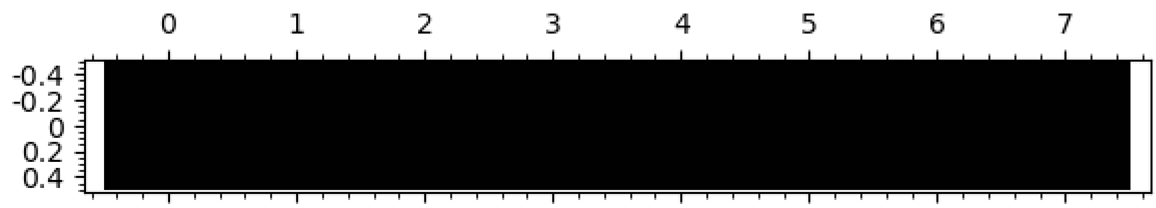

Figure 3: *the figure represents the visualization of the grayscale image corresponding to the vector v1 using matrix plot(). The figure was produced using SageMath. The code for the figure can be found in Appendix A.*

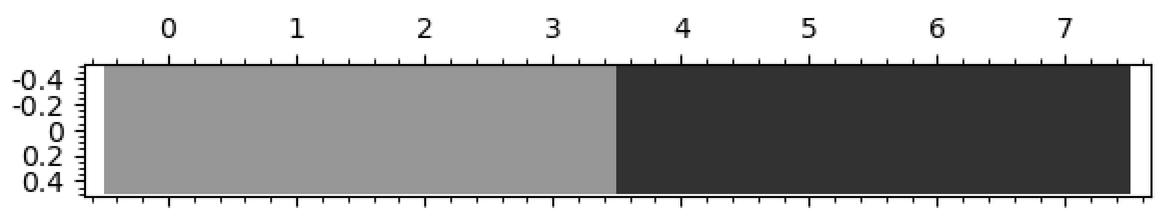

Figure 4*: the figure represents the visualization of the grayscale image corresponding to the linear combination 100v1 + 50v2 using matrix plot().The figure was produced using SageMath. The code for the figure can be found in Appendix A.*

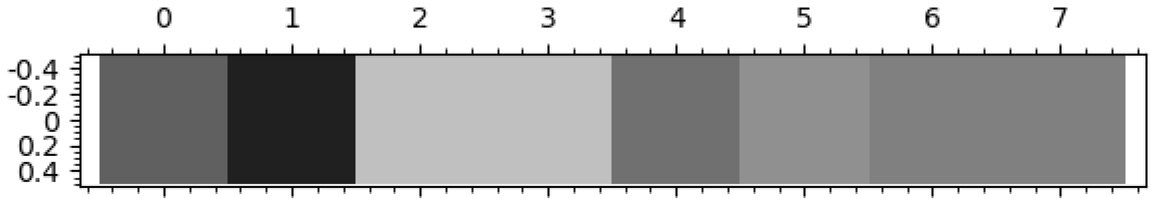

Figure 5*: the figure represents the visualization of the grayscale image corresponding to the linear combination 128v1 − 64v3 + 32v5 − 16v7 using matrix plot().The figure was produced using SageMath. The code for the figure can be found in Appendix A.*

On figure 1, we represent the whole image with only 1 vector, so the whole image is colored with the 1 color that is given by v1 since no other vectors are used for producing this image.

On figure 2 we used the linear combination of 2 vectors (vector addition is used), so the image consists of 2 components given by the linear combination of v1 and v2.

Figure 3 is a bit more complicated since we have both addition and subtraction of the vectors. In this case, the value of the brightness of each pixel in the image is determined by the weighted sum of the components of the vectors v1, v3, v5, and v7.

This linear combination can be thought of as a way to represent a grayscale image using a combination of different patterns, where each vector corresponds to a different pattern.

3. **Reducing Storage Space and Improving Human Interpretability of Images Using Sparse Matrices and Thresholding**

a) Find the coordinates of v in terms of the new basis S = {v1, v2, . . . , v8}.

To find the coordinates of the vector in the basis we have to create a matrix that contains all the vectors in the basis and augment it with the vector we want to find the coordinates of. After that we have to reduce the matrix to RREF. If there is a unique solution (the number of pivots is equal to the number of pivot columns and RREF is an identity matrix) the augmented side would represent the coordinates of v in terms of the new basis S.

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}
\times
\begin{bmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \\ c7 \\ c8 \end{bmatrix}
=
\begin{bmatrix} 31 \\ 159 \\ 9 \\ 162 \\ 233 \\ 54 \\ 217 \\ 3 \end{bmatrix}
$$

Figure 6*: the figure represents matrix A multiplied by the vector of coefficients and equated to the vector v. The figure was produced using SageMath. The code can be found in Appendix A.*

```
[  1   1   1   1   1   1   1   1| 31]
[  1   1   1   1  -1  -1  -1  -1|159]
[  1   1  -1  -1   0   0   0   0|  9]
[  0   0   0   0   1   1  -1  -1|162]
[  1  -1   0   0   0   0   0   0|233]
[  0   0   1  -1   0   0   0   0| 54]
[  0   0   0   0   1  -1   0   0|217]
[  0   0   0   0   0   0   1  -1|  3]
```

***Figure** 7:the figure demonstrates the matrix A augmented with the vector v. The figure was produced using SageMath. The code can be found in Appendix A.*

```
[      1      0      0      0      0      0      0      0| 285/2]
[      0      1      0      0      0      0      0      0|-181/2]
[      0      0      1      0      0      0      0      0|  97/2]
[      0      0      0      1      0      0      0      0| -11/2]
[      0      0      0      0      1      0      0      0|   133]
[      0      0      0      0      0      1      0      0|   -84]
[      0      0      0      0      0      0      1      0|   -55]
[      0      0      0      0      0      0      0      1|   -58]
```

***Figure 8**:the figure demonstrates the matrix A augmented with the vector v and reduced to RREF. The figure was produced using SageMath. The code can be found in Appendix A.*

As we have the matrix A reduced to RREF (figure 8) we can see that there is a unique solution. The augmented side represents the coordinates of the vector v in the space S. This is the solution:

$C = [285/2, -181/2, 97/2, -11/2, 133, -84, -55, -58]$

When we multiply the vector v by the vector c we will get the vector v represented as a linear combination of the vectors in S.

b) Choose three thresholds $\epsilon1$,$\epsilon2$,$\epsilon3$. Let c1, c2, c3 be the vectors you get by compressing v using thresholds $\epsilon1$, $\epsilon2$, and $\epsilon3$ respectively. Express c1,c2,c3 in the standard basis. What do these new vectors represent? Create visual representations of v, c1, c2, c3 in the standard basis vectors.

To compress vector v using a threshold value, we set any coordinate in our given vector v with absolute value less than the threshold to zero. We already found the representation of v in the set S and denoted it as c and we will use this vector. To express c1, c2, and c3 in the standard basis, we need to find the coefficients of the linear combinations of the standard basis vectors that result in these new vectors. The standard basis vectors are the 8 given vectors. To express c1 in the standard basis, we need to find the coefficients of the linear combination:

$c1 \ = \ a1 \ * \ v1 \ + \ a2 \ * \ v2 \ + \ a3 \ * \ v3 \ + \ ... \ + \ an \ * \ vn$

where a1, a2, a3, are the coefficients we're looking for.

Let's set the first thresholds $\epsilon1$ to be equal to 10:

$\epsilon1 \ = \ 10$

We go loop through the vector v and each value that is lover than $\epsilon1$ we set it to 0.

$C1 \ = \ [285/2, 0, 97/2, 0, 133, 0, 0, 0]$

Since we need the vector v to be represented with c1, we will multiply the matrix A that consists of the 8 basis vector by c1 and plot the resulting grayscale image. After that we plot the result. Figures 9-11 represent the output of this algorithm for the threshold values $\epsilon1 \ = \ 10$,

$\epsilon2 \ = \ 30, \epsilon3 \ = \ 50$

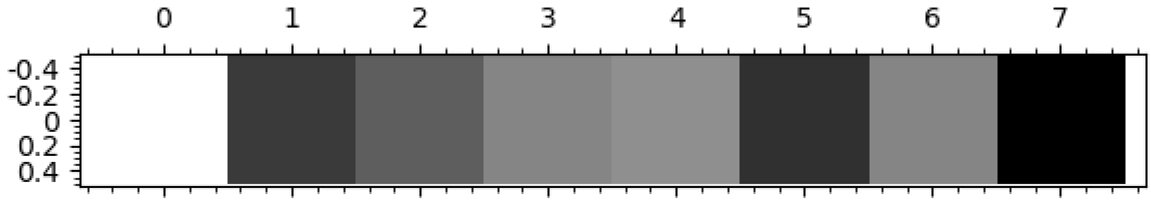

Figure 9*: the figure represents the visualization of the grayscale image produced by setting the threshold value to e1 = 10 and setting to 0 all the values below the threshold . The figure was produced using the matrix plot() function in SageMath. The code for the figure can be found in Appendix A.*

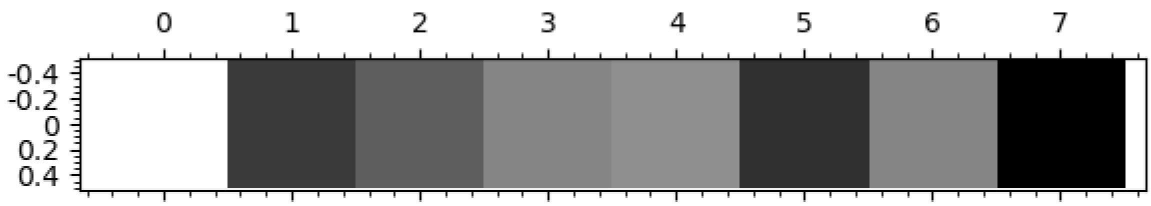

Figure 10*: the figure represents the visualization of the grayscale image produced by setting the threshold value to e1 = 30 and setting to 0 all the values below the threshold . The figure was produced using the matrix plot() function in SageMath. The code for the figure can be found in Appendix A.*

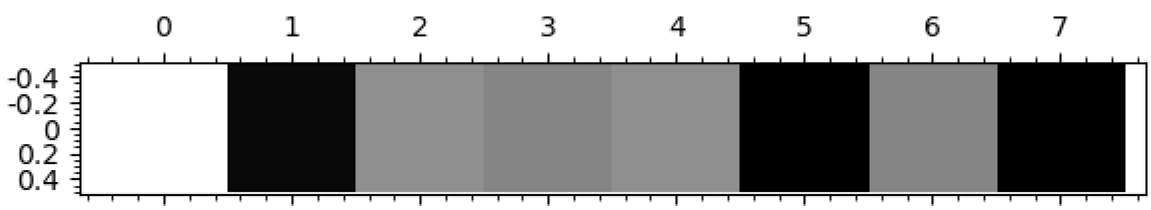

Figure 11*: the figure represents the visualization of the grayscale image produced by setting the threshold value to e1 = 50 and setting to 0 all the values below the threshold . The figure was produced using the matrix plot() function in SageMath. The code for the figure can be found in Appendix A.*

c) How large can you make ϵ before the compression is noticeable?

Using the method error and trial I found the key values that make the noticeable changes to the compressed image. The first noticeable change happens at the value of $\epsilon 2 \;=\; 49$. As we can see on figure 12, as we change the threshold value from 48 to 49 the pixels under the numbers 1,5 and 7 change significantly and it becomes difficult to ignore the change. Pixel 2 also changes but its change is not that drastic compared to the ones addressed before.

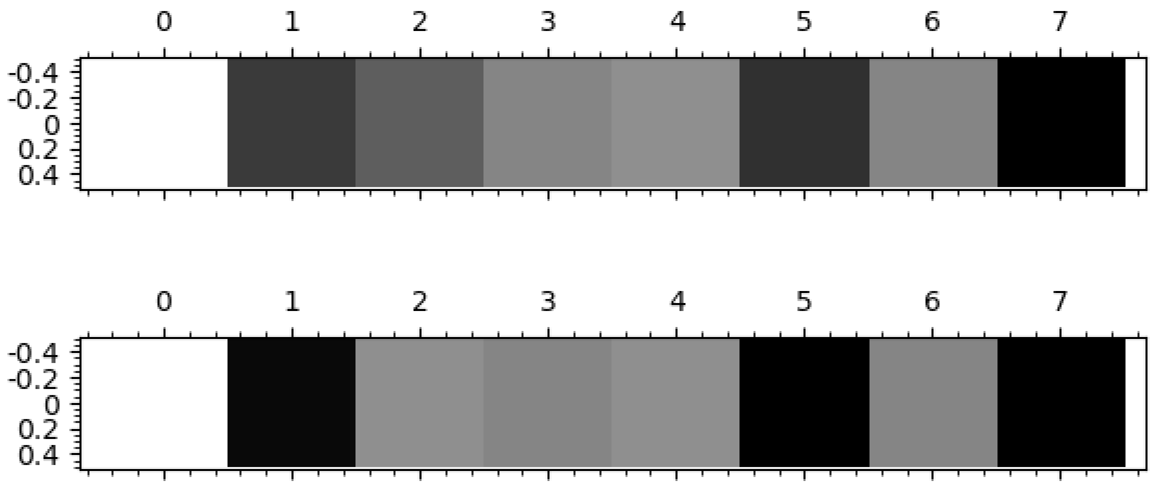

Figure 11*: the figure represents the comparison of the visualizations of the grayscale image produced by setting the threshold values to e = 48 and e = 49 and setting to 0 all the values below the threshold . The figure was produced using the matrix plot() function in SageMath. The code for the figure can be found in Appendix A.*

The next noticeable change happened at the value of 134. As we can see on figure 12 the compressed images at the threshold value of 133 and 49 from the figure 11 have barely any differences. However as we increase the value to 134 most of the pixels get compressed and we end up with only 2 possible colors for the grayscale image. When we look at the vector of compressed image we can see why

$c134 = [285/2\ 285/2\ 285/2 \quad 0\ 285/2 \quad 0 \quad 0 \quad 0])$

There are only 2 types of values, so it is reasonable to expect the grayscale representation of the image to have only 2 levels of brightness.

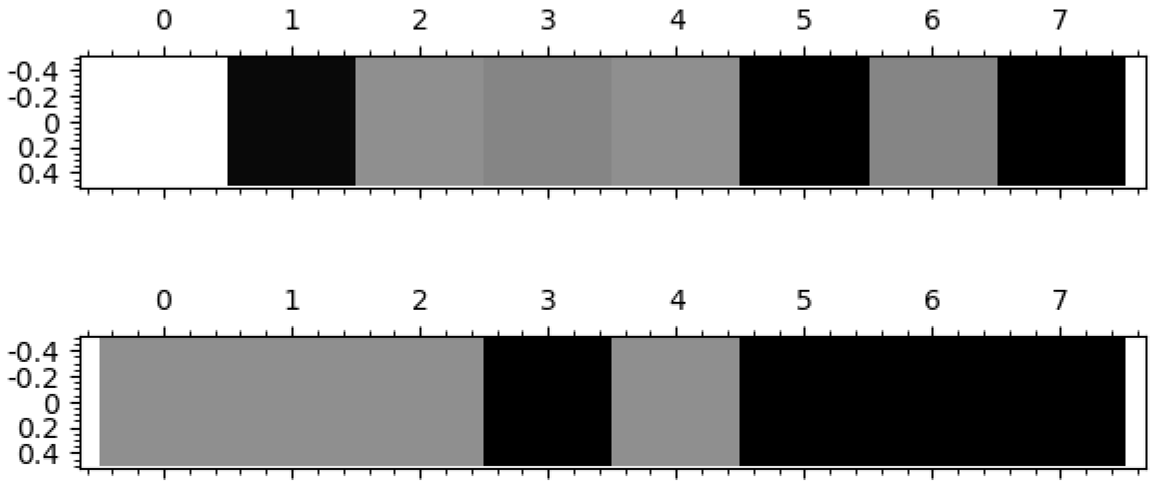

__Figure 12__: the figure represents the comparison of the visualizations of the grayscale image produced by setting the threshold values to e = 133 and e = 134 and setting to 0 all the values below the threshold . The figure was produced using the matrix plot() function in SageMath. The code for the figure can be found in Appendix A.

The last noticeable change happened at the value of 143. As we can see on figure 13 the compressed images at the threshold value of 133 and 142 from the figure 12 have barely any differences. However, as we increase the value to 143 all the pixels become black. This is the final stage of the compressed image. No matter what the value of the next threshold the grayscale representation will not change anymore. We can see it based on the vector that corresponds to this visualization.

$C143 = [0, 0, 0, 0, 0, 0, 0, 0]$

It consists solely of zeros, so it does not have any information from the original image.

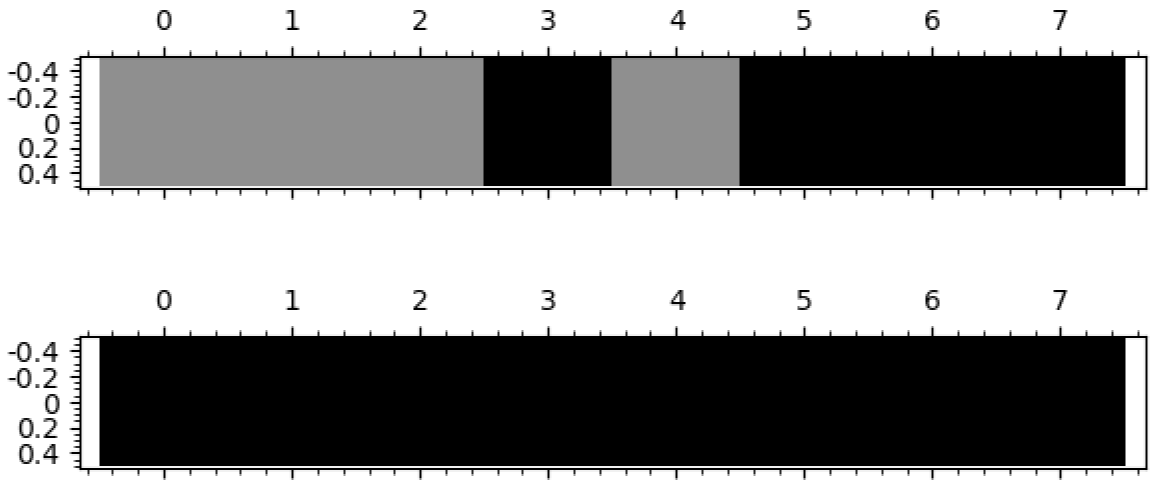

Figure 13*: the figure represents the comparison of the visualizations of the grayscale image produced by setting the threshold values to e = 142 and e = 143 and setting to 0 all the values below the threshold . The figure was produced using the matrix plot() function in SageMath. The code for the figure can be found in Appendix A.*

Reflection

1. #selfawareness: it was not shown in the written art of the assignment itself but I used this HC a lot while I was working on the assignment. I had difficult times last week and I knew that the deadline was approaching. As a responsible student I started working on an assignment 2 days before the deadline. However, because of some life circumstances I was not feeling mentally well and I could not concentrate on the assignment. Instead of pushing myself and making myself even more energy drained I decided that I will allow my body and mind to rest and I will take extensions. Although it is not considered an achievement from the academic point of view, I do consider it to be my small win. I understood that the quality of the time I spend on the assignment is more important than the quantity and that I am not productive when I am not feeling good and there is no point in pushing myself even more and getting in an even worse state. After taking a break I felt better and what is more important I felt excited about working on the assignment and I did it with joy and pleasure although 2 days past the deadline.

#analogies: I used analogical thinning to explain why it is impossible to have more than n orthogonal vector in R^n dimensional space. I created an image that is easy to understand and used a simplified version with 4 vectors and 2 dimensional space.

2.I am not sure if it can be categorized as creative but I find it absolutely beautiful how we were constructing regression lines with matrices in session 8.2. It was my favourite class at Minerva. I would really like to know more about it and get more hands-on experience in this. Would be very grateful for any further literature/online courses/minerva classes recommendations.

Appendix A:

Code for figure 2

```
#define vectors

v1 =  vector([1, 1, 1, 1, 1, 1, 1, 1])

v2 = vector([1, 1, 1, 1, -1, -1, -1, -1])

v3 = vector([1, 1, -1, -1, 0, 0, 0, 0])

v4 = vector([0, 0, 0, 0, 1, 1, -1, -1])

v5 = vector([1, -1, 0, 0, 0, 0, 0, 0])

v6 = vector([0, 0, 1, -1, 0, 0, 0, 0])

v7 = vector([0, 0, 0, 0, 1, -1, 0, 0,])

v8 = vector([0, 0, 0, 0, 0, 0, 1, -1])


lst = [v1,v2,v3,v4,v5,v6,v7,v8]

#create space for used vectors to avoid doube repetition

used = []

for i in lst:

    used.append(i)

    for j in lst:

        if i!=j and j not in used:

            #find dot procuct

            r = j.dot_product(i)

            print("Dot product of", i, 'and', j, 'is',r)
```

Code for figures 3-5

```
#define vectors
v1 =  vector([1, 1, 1, 1, 1, 1, 1, 1])
v2 = vector([1, 1, 1, 1, -1, -1, -1, -1])
v3 = vector([1, 1, -1, -1, 0, 0, 0, 0])
v4 = vector([0, 0, 0, 0, 1, 1, -1, -1])
v5 = vector([1, -1, 0, 0, 0, 0, 0, 0])
v6 = vector([0, 0, 1, -1, 0, 0, 0, 0])
v7 = vector([0, 0, 0, 0, 1, -1, 0, 0,])
v8 = vector([0, 0, 0, 0, 0, 0, 1, -1])

m1 = matrix(v1)
matrix_plot(m1,cmap = "gray",vmin = 0,vmax=255)
m2 = matrix(100*v1+50*v2)
matrix_plot(m2,cmap = "gray",vmin = 0,vmax=255)
m3 = matrix(128*v1 - 64*v3 + 32*v5 - 16*v7)
matrix_plot(m3,cmap = "gray",vmin = 0,vmax=255)
```

Code for figures 6-8

```
v1 =  vector([1, 1, 1, 1, 1, 1, 1, 1])
v2 = vector([1, 1, 1, 1, -1, -1, -1, -1])
v3 = vector([1, 1, -1, -1, 0, 0, 0, 0])
v4 = vector([0, 0, 0, 0, 1, 1, -1, -1])
v5 = vector([1, -1, 0, 0, 0, 0, 0, 0])
v6 = vector([0, 0, 1, -1, 0, 0, 0, 0])
v7 = vector([0, 0, 0, 0, 1, -1, 0, 0,])
v8 = vector([0, 0, 0, 0, 0, 0, 1, -1])



v = vector([31, 159, 9, 162, 233, 54, 217, 3])


A = matrix([v1,v2,v3,v4,v5,v6,v7,v8])
A = A.augment(v, subdivide = True)
A.rref()
```

Code for figures 8 - 13

```
#define vectors
v1 =  vector([1, 1, 1, 1, 1, 1, 1, 1])
v2 = vector([1, 1, 1, 1, -1, -1, -1, -1])
v3 = vector([1, 1, -1, -1, 0, 0, 0, 0])
v4 = vector([0, 0, 0, 0, 1, 1, -1, -1])
v5 = vector([1, -1, 0, 0, 0, 0, 0, 0])
v6 = vector([0, 0, 1, -1, 0, 0, 0, 0])
v7 = vector([0, 0, 0, 0, 1, -1, 0, 0,])
v8 = vector([0, 0, 0, 0, 0, 0, 1, -1])
A = matrix([v1,v2,v3,v4,v5,v6,v7,v8])
c = vector([285/2, -181/2,97/2,-11/2,133,-84,-55,-58])
c_list = [285/2, -181/2,97/2,-11/2,133,-84,-55,-58]

def compress(e,c_list,c,A):
    for i in c_list:
        if i < e:
            c[c_list.index(i)] = 0
    result = matrix([A*c])
    return matrix_plot(result, cmap = 'gray', vmin = 0,vmax = 255)


compress(49,c_list,c,A)
```