Introduction to Data Processing and Representation (236201)

Spring 2022

Homework 1

Submitted by:
Shelly Golan 209290477
Ofer Nissim 312367576

May 10, 2022

Part I

Theory

1. Solving the L^p problem using the L^2 solution.
The weighted L^p sampling problem consists in solving the following optimization problem:

$$\min_{\hat{f}} \varepsilon^p \left(f, \hat{f}\right) = min \int_0^1 \left|f(x) - \hat{f}(x)\right|^p w(x)dx$$

(a) We assume here that w is a constant function. So, the optimal $\hat{f}_p$ when $p = 1$ is the median over each interval and when $p = 2$ it's the average over each interval.

(b) For a general w, we'll want to find the optimal $\hat{f}_p$for $p = 2$.
Since $\hat{f}$ is a piece-wise constant function we can denote for each interval that $(1)\ \forall x \in I_i :$ $\hat{f}(x) = \hat{f}_i(x)$ when $\hat{f}_i(x)$ is the value of $\hat{f}$ in the interval I_i.
We'll set $p = 2$ in the optimization problem above:

$$\varepsilon^2 \left(f, \hat{f}\right) = \int_0^1 \left|f(x) - \hat{f}(x)\right|^2 w(x)dx \underset{when\, function\, raised\, by\, power\, of\, 2,\, it's\, positive}{=}$$

$$= \int_0^1 \left(f(x) - \hat{f}(x)\right)^2 w(x)dx \underset{(1)}{=} \sum_{i=1}^{N} \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left(f(x) - \hat{f}_i(x)\right)^2 w(x)dx$$

To find the optimal value $\hat{f}_p$ which minimizes the optimization problem we'll take a derivative of $\varepsilon^2 \left(f, \hat{f}\right)$ w.r.t $\hat{f}_i$ and take it to zero:

$$\frac{\partial \varepsilon^2 \left(f, \hat{f}\right)}{\partial \hat{f}_i} = \frac{\partial}{\partial \hat{f}_i} \sum_{i=1}^{N} \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left(f(x) - \hat{f}_i(x)\right)^2 w(x)dx = -2 \cdot \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left(f(x) - \hat{f}_i(x)\right) w(x)dx$$

$$\frac{\partial \varepsilon^2 \left(f, \hat{f}\right)}{\partial \hat{f}_i} = -2 \cdot \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left(f(x) - \hat{f}_i(x)\right) w(x)dx = 0$$

$$\int_{\frac{i-1}{N}}^{\frac{i}{N}} f(x)w(x)dx - \int_{\frac{i-1}{N}}^{\frac{i}{N}} \hat{f}_i(x)w(x)dx = 0$$

$$0 \underset{(*)}{=} \int_{\frac{i-1}{N}}^{\frac{i}{N}} f(x)w(x)dx - \hat{f}_i(x) \int_{\frac{i-1}{N}}^{\frac{i}{N}} w(x)dx$$

$$\hat{f}_i(x) = \frac{1}{\int_{\frac{i-1}{N}}^{\frac{i}{N}} w(x)dx} \int_{\frac{i-1}{N}}^{\frac{i}{N}} f(x)w(x)dx$$

(*) over each interval $\hat{f}_i(x)$ is a constant function.
Since for each interval I_i, $\hat{f}_2 = \hat{f}_i$

$$\hat{f}_2(x) = \frac{1}{\int_{\frac{i-1}{N}}^{\frac{i}{N}} w(x)dx} \int_{\frac{i-1}{N}}^{\frac{i}{N}} f(x)w(x)dx$$

Which is the weighted average over each interval.

(c) For a general w, we'll want to find the optimal $\hat{f}_p$for $p = 1$.
As is in the previous question, $(1)\ \forall x \in I_i : \hat{f}(x) = \hat{f}_i(x)$ when $\hat{f}_i(x)$ is the value of $\hat{f}$ in the interval I_i.
We'll set $p = 1$ in the optimization problem above:

$$\varepsilon^1\left(f, \hat{f}\right) = \int_0^1 \left|f(x) - \hat{f}(x)\right|^1 w(x)dx =$$

$$= \sum_{i=1}^{n} \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f(x) - \hat{f}(x)\right| w(x)dx =$$

We'll denote in the following manner:
$\delta_i^+ \triangleq \left\{x \,|\, \forall x \in I_i \, s.t \, f(x) \geq \hat{f}_i(x)\right\}$
$\delta_i^- \triangleq \left\{x \,|\, \forall x \in I_i \, s.t \, f(x) < \hat{f}_i(x)\right\}$
So, we'll get that:

$$\frac{\partial}{\partial \hat{f}_i}\varepsilon^1\left(f, \hat{f}\right) = \int_{\frac{i-1}{N}}^{\frac{i}{N}} sign\left(f(x) - \hat{f}(x)\right) w(x)dx =$$

$$\underset{using\, the\, above}{=} \int_{\delta_i^+} w(x)dx - \int_{\delta_i^-} w(x)dx \underset{to\, find\, the\, minimum}{=} 0$$

$$\Longrightarrow \int_{\delta_i^+} w(x)dx = \int_{\delta_i^-} w(x)dx$$

The result we got will be "satisfied" or "the best" when $\hat{f}_i$ is the weighted median over the interval I_i , therefore $\hat{f}_p$ will be the weighted median over each interval I_i.

(d) Let $\varepsilon_i^p\left(f_i, \hat{f}_i\right) = \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f_i(x) - \hat{f}_i(x)\right|^p w(x)dx$. Where f_i and $\hat{f}_i$are the functions f and $\hat{f}$ restrained to the interval I_i. Therefore, we can rewrite our optimization problem $\varepsilon^p\left(f, \hat{f}\right)$ as:

$$\varepsilon^p\left(f, \hat{f}\right) = \int_0^1 \left|f(x) - \hat{f}(x)\right|^p w(x)dx = \sum_{i=1}^{N} \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f_i(x) - \hat{f}_i(x)\right|^p w(x)dx = \sum_{i=1}^{N} \varepsilon_i^p\left(f_i, \hat{f}_i\right)$$

(e) Let $i \in \{1, ..., N\}$.

i. Assume that $f_i(x) \neq \hat{f}_i(x)$ for all $x \in I_i$ for our fixed i.

$$w_{f_i,\hat{f}_i}(x) = \frac{\left|f(x) - \hat{f}(x)\right|^p}{\left(f(x) - \hat{f}(x)\right)^2}$$

We can devide the two because $f_i(x) \neq \hat{f}_i(x)$. And they are both positive thus w is positive.

ii. We'll rewrite our problem:

$$\varepsilon_i^p\left(f_i, \hat{f}_i\right) = \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f_i(x) - \hat{f}_i(x)\right|^p w(x)dx =$$

$$= \int_{\frac{i-1}{N}}^{\frac{i}{N}} \frac{\left(f_i(x) - \hat{f}_i(x)\right)^2}{\left(f_i(x) - \hat{f}_i(x)\right)^2} \cdot \left|f_i(x) - \hat{f}_i(x)\right|^p w(x)dx =$$

$$= \int_{\frac{i-1}{N}}^{\frac{i}{N}} \frac{\left|f_i(x) - \hat{f}_i(x)\right|^p}{\left(f_i(x) - \hat{f}_i(x)\right)^2} \cdot \left(f_i(x) - \hat{f}_i(x)\right)^2 w(x)dx =$$

$$\underset{Q1.i}{=} \int_{\frac{i-1}{N}}^{\frac{i}{N}} w_{f_i,\hat{f}_i}(x) \cdot \left(f_i(x) - \hat{f}_i(x)\right)^2 w(x)dx =$$

If we denote $w'_{f_i,\hat{f}_i}(x) = w_{f_i,\hat{f}_i}(x) \cdot w(x)$ we'll get that:

$$= \int_{\frac{i-1}{N}}^{\frac{i}{N}} w'_{f_i,\hat{f}_i}(x) \cdot \left(f_i(x) - \hat{f}_i(x)\right)^2 dx$$

as needed.

iii. Since our wieght is depended of $\hat{f}_i$ we can't calculate directly the integral above. If we would have a weight independed of $\hat{f}_i$, we could use the closed form expression above to calculate the integral.

iv. In this section we remove the previous assumption. The main change due to the removal of that assumption is that now, if $f_i(x) = \hat{f}_i(x)$ we'll get that $w_{f_i,\hat{f}_i}(x)$ will be ∞ if $p < 2$ and 0 if $p > 2$, or 1 if $p = 2$.
Using the suggested function

$$\tilde{w}_{f_i,\hat{f}_i}(x) = min\left\{\frac{1}{\varepsilon}, w_{f_i,\hat{f}_i}\right\}$$

is useful now because we might have a situation in which $f_i(x) = \hat{f}_i(x)$ and instead of having a weight function that is unstable and ∞, we'll get that our weight function is $\frac{1}{\varepsilon}$ for some $\varepsilon > 0$. Even if $\frac{1}{\varepsilon}$ is a large number, it's still a constant and our problem is well defined.

v. Our pseudo-code:

A. Initialize $\hat{f}_i = 0$

B. Calculate $w'_{f_i,\hat{f}_i}(x) = min\left\{\frac{1}{\varepsilon}, w_{f_i,\hat{f}_i}(x)\right\}$

C. Set $\hat{f}_i^{next} = \frac{\int_{\frac{i-1}{N}}^{\frac{i}{N}} f(x) w'_{f_i,\hat{f}_i}(x)dx}{\int_{\frac{i-1}{N}}^{\frac{i}{N}} w'_{f_i,\hat{f}_i}(x)dx}$

D. Use $\hat{f}_i^{next}$ as $\hat{f}_i$ and repeat the calculation from B.

(f) Our pseudo-code:

i. For each $i \in \{1, ..., N\}$ we'll set the interval $I_i = \left[\frac{i-1}{N}, \frac{i}{N}\right)$ and do the following:

A. Calculate $\hat{f}_i$ using the algorithm above.

B. Set $\hat{f}(x) = \hat{f}_i(x)$

ii. return $\hat{f}(x)$

2. Signal Discretization using a Piecewise-Linear Approximation.

   (a) We want to show that for a positive integer k:

$$\int_{t\in\Delta_i}(t-t_i)^k dt = \begin{cases} 0 & k\,is\,odd \\ \frac{|\Delta_i|}{2^k(k+1)} & k\,is\,even \end{cases}$$

an integral computation yields the following:

$$\int_{t\in\Delta i}(t-t_i)^k dt = \frac{1}{k+1}(t-t_i)^{k+1}|_{t\in\Delta_i=(t_i-\frac{|\Delta_i|}{2},t_i+\frac{|\Delta_i|}{2})} =$$

$$= \frac{1}{k+1}\left[(t_i+\frac{|\Delta_i|}{2}-t_i)^{k+1}-(t_i-\frac{|\Delta_i|}{2}-t_i)^{k+1}\right] =$$

$$= \frac{1}{k+1}\left(\frac{|\Delta_i|}{2}\right)^{k+1} - \left(-\frac{|\Delta_i|}{2}\right)^{k+1} = \frac{1}{2^{k+1}(k+1)}\left|\Delta_i\right|^{k+1} - (-\left|\Delta_i\right|)^{k+1}$$

separating to cases by the parity of k we obtain **(2)**:

$$\int_{t\in\Delta i}(t-t_i)^k dt = \begin{cases} 0 & if\;k\,is\,odd \\ \frac{2|\Delta_i|^{k+1}}{2^{k+1}(k+1)} = \frac{|\Delta_i|^{k+1}}{2^k(k+1)} & if\;k\,is\,even \end{cases}$$

   (b) We want to find the optimal coefficients a_i, c_i that minimize the MSE of representing the entire signal using N intervals.

$$MSE = \int_0^1 \phi(t) - \phi^{opt}(t)dt = \frac{1}{N}\sum_{i=1}^{n}\int_{\Delta_i}\left(\phi_i(t)-\phi_i^{opt}(t)\right)^2 dt$$

Where $\phi_i^{opt} = a_i(t-t_i)+c_i$, so we want to find the derivative w.r.t a_i and c_i, in order to find the coefficients that minimize the MSE

$$\frac{\partial}{\partial a_i}\left(\frac{1}{N}\sum_{i=1}^{n}\int_{\Delta_i}\left(\phi_i(t)-a_i(t-t_i)-c_i\right)^2 dt\right) =$$

$$= -2\int_{\Delta_i}\left(\phi_i(t)-a_i(t-t_i)-c_i\right)(t-t_i)dt = 0$$

$$\Rightarrow \int_{\Delta_i}\phi_i(t)(t-t_i)dt - a_i\int_{\Delta_i}(t-t_i)^2 dt - c_i\int t-t_i dt = 0$$

and as we showed in (a) for odd k values $\int t-t_i = 0$ so

$$\int_{\Delta_i}\phi_i(t)(t-t_i)dt = \int_{\Delta_i}a_i^{opt}(t-t_i) = a_i^{opt}\cdot\frac{|\Delta_i|^3}{12}$$

$$a_i^{opt} = \frac{12\cdot\int_{\Delta_i}\phi_i(t)(t-t_i)dt}{|\Delta_i|^3}$$

We'll find c_i:

$$\frac{\partial}{\partial c_i}\left(\frac{1}{N}\sum_{i=1}^{n}\int_{\Delta_i}\left(\phi_i(t)-a_i(t-t_i)-c_i\right)^2 dt\right)$$

$$= -2\int_{\Delta_i}\left(\phi_i(t)-a_i(t-t_i)-c_i\right)dt = 0$$

$$\int_{\Delta_i}\phi_i(t)dt - \int_{\Delta_i} a_i(t-t_i)dt - \int_{\Delta_i} c_i dt = 0$$

Again, as we showed in (a) for odd k values $\int t - t_i = 0$ so

$$\int_{\Delta_i}\phi_i(t)dt = \int_{\Delta_i} c_i^{opt} dt$$

$$\Rightarrow c_i^{opt} = \frac{\int_{\Delta_i}\phi_i(t)dt}{|\Delta_i|}$$

(c) using the optimal coefficients, the minimal MSE of representing the entire signal using N intervals would be:

$$\Psi_{MSE}(\phi \to \hat{\phi}_{opt}) = \int_0^1 (\phi(t) - \hat{\phi}_{opt}(t))^2 dt = \frac{1}{N}\sum_{i=1}^{N}\int_{\Delta_i}(\phi(t) - [\phi(t_i) + \phi'(t_i)(t-t_i)])^2 dt$$

(d) To compare the minimal MSE's we'll compute them first:
We saw the piecewise constant MSE function in class,

$$MSE = \int_o^1 \phi(t)^2 dt - \frac{1}{N^2}\sum_{i=1}^{N}\int_{\Delta_i}\phi_i(t)^2 dt$$

We'll calculate integrals first:

$$\int_{\Delta_i}\left(a_i^{opt}(t-t_i)+c_i^{opt}\right)^2 dt = \frac{\left(a_i^{opt}(t-t_i)+c_i^{opt}\right)^3}{3a_i^{opt}}\Big|_{\frac{i-1}{N}}^{\frac{i}{N}} = \frac{\left(a_i^{opt}+2Nc_i^{opt}\right)^3 - \left(2Nc_i^{opt}-a_i^{opt}\right)^3}{24a_i^{opt}N^3} =$$

$$= \frac{2a_i^{(opt)3} + 24N^2 c_i^{(opt)^2} a_i^{opt}}{24N^3 a_i^{opt}} = \frac{a_i^{(opt)2}}{12N^3} + \frac{c_i^{opt}}{N}$$

Now we'll use the coefficients we found in the previous question and use the fact that $\Delta_i = \frac{1}{N}$:

$$\frac{1}{N^2}\sum_{i=1}^{N}\int_{\Delta_i}\left(a_i^{opt}(t-t_i)+c_i^{opt}\right)^2 dt - \frac{1}{N^2}\sum_{i=1}^{N}\int_{\Delta_i}\phi_i(t)^2 dt =$$

$$=$$

$$= \frac{\left(12N^3\int_{\Delta_i}\phi_i(t)(t-t_i)dt\right)^2}{12N^3} + \frac{\left(N\int_{\Delta_i}\phi_i(t)dt\right)^2}{N} - \left(\frac{1}{N}\int_{\Delta_i}\phi_i(t)dt\right)^2 =$$

$$= 12N^3 \left(\int_{\Delta_i} \phi_i(t)(t-t_i)dt \right)^2 + \frac{N^3-1}{N^2} \left(\int_{\Delta_i} \phi_i(t)dt \right)^2 \geq 0$$

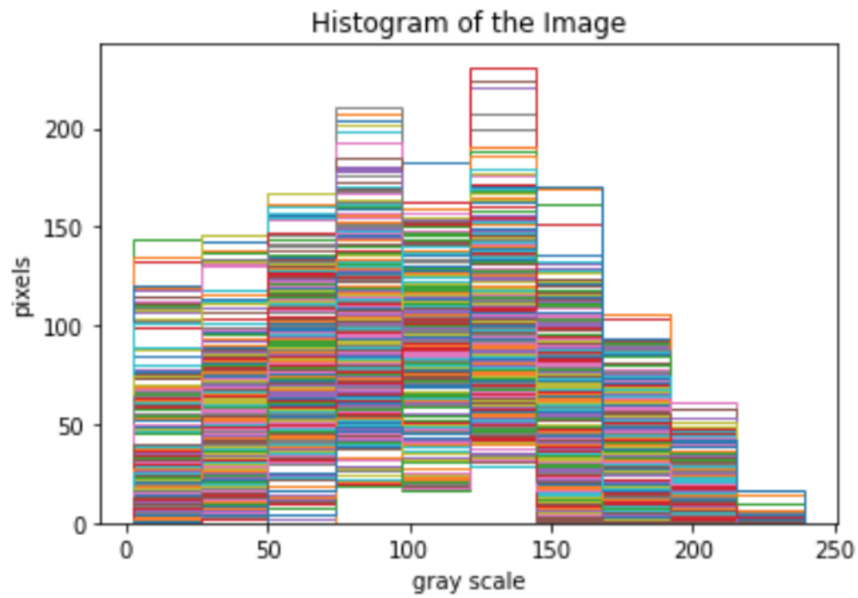
Both components are positive since $N > 0$ and we have squared components as well. Therefore, $MSE_{peicewise} - MSE_{linear} \geq 0$.

Part II

Implementation

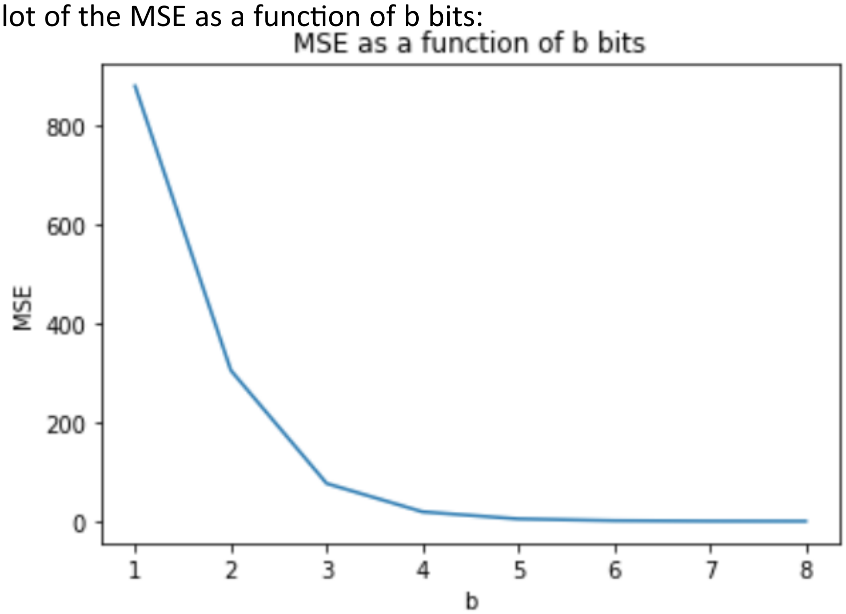
1. Question 1:

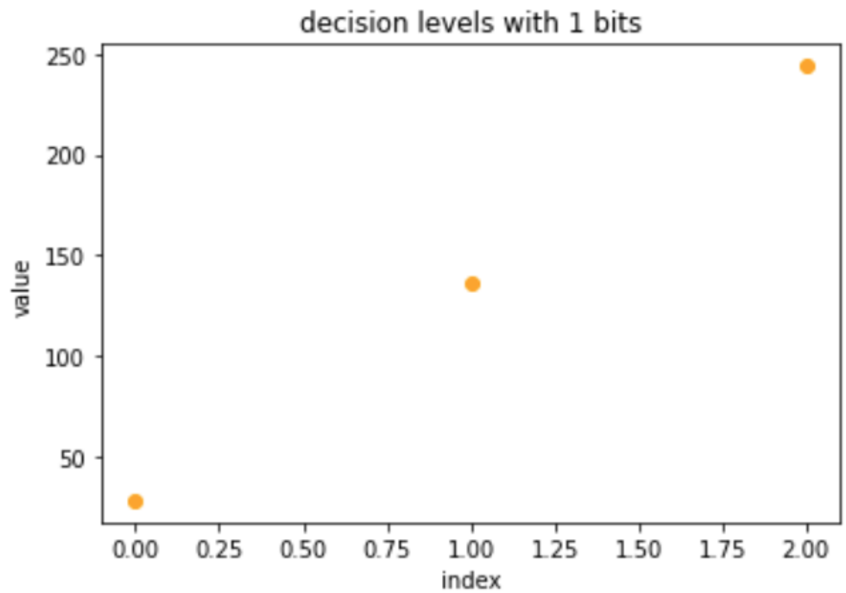
   (a) Plot of the histogram of the image:



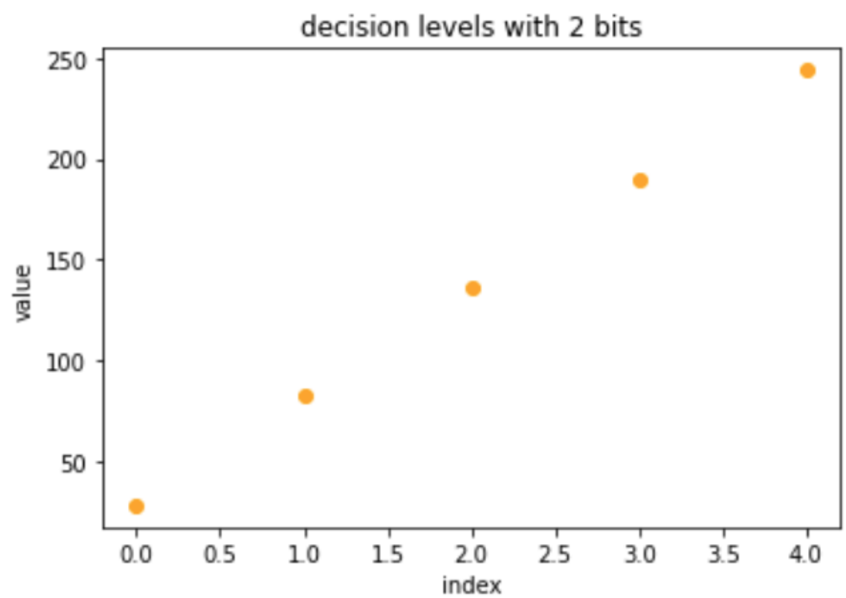
   (b) We applied uniform

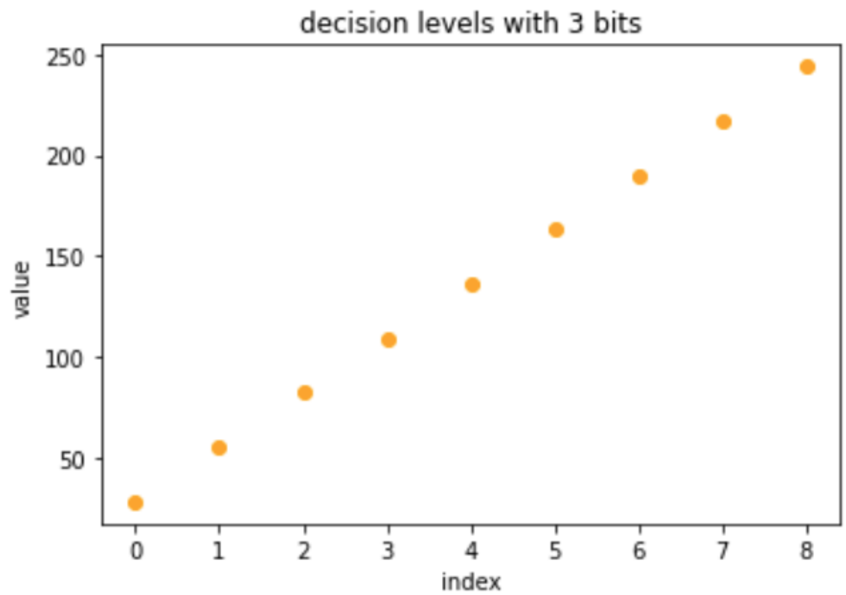
      i. Plot of the MSE as a function of b bits:

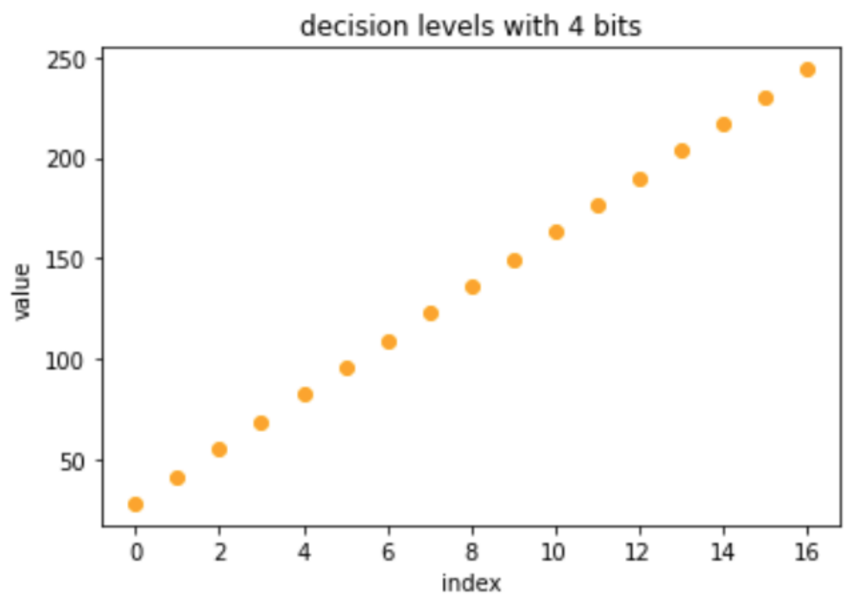


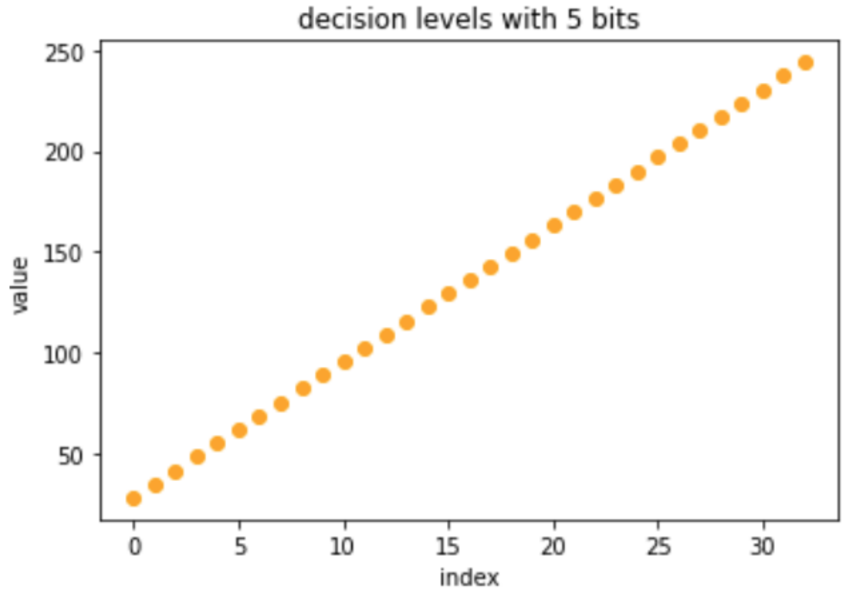

decision levels with 1 bits
value
index
250
200
150
100
50
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

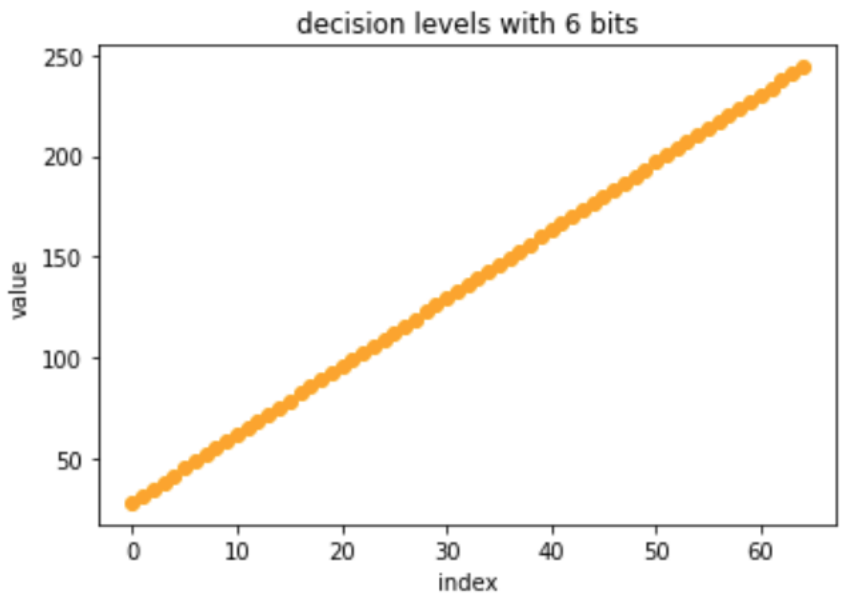
ii.

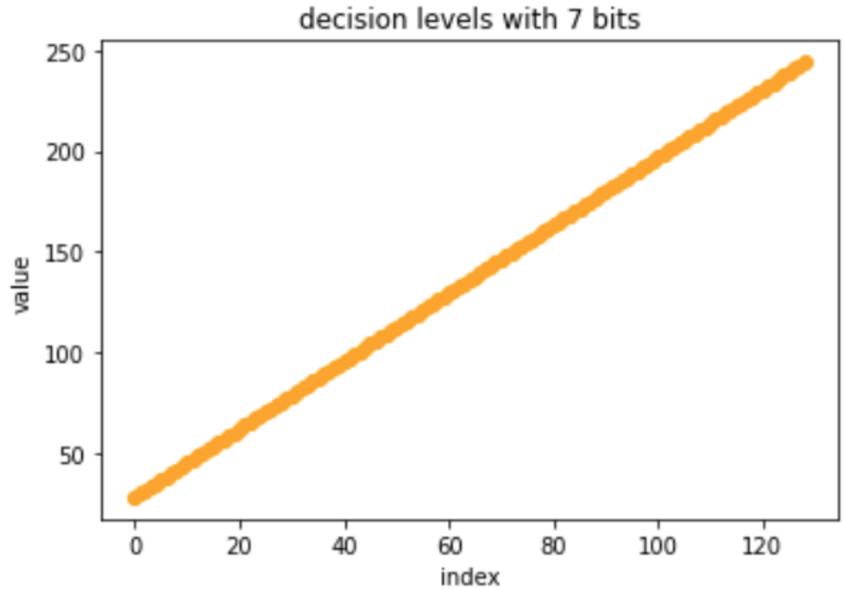

decision levels with 2 bits
value
index
250
200
150
100
50
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

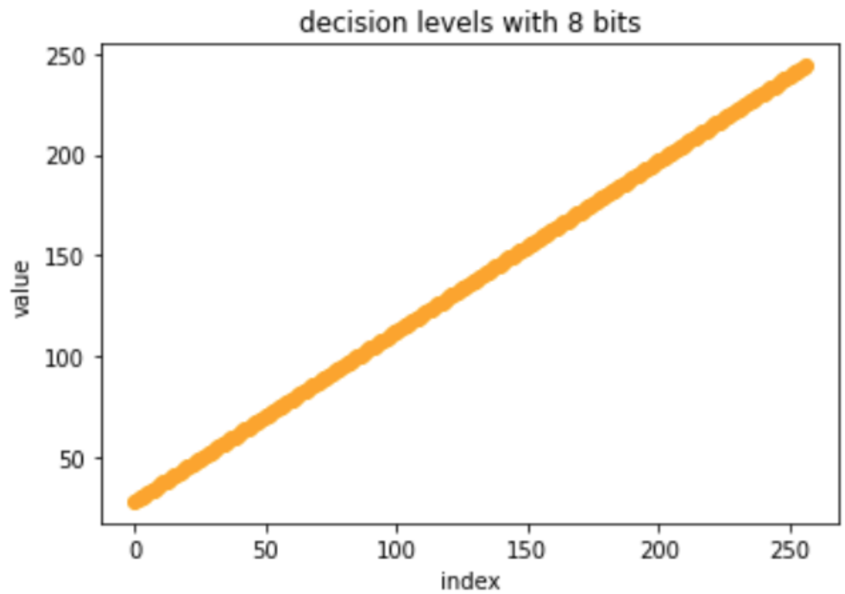

decision levels with 3 bits
value
250
200
150
100
50
0
1
2
3
4
5
6
7
8
index

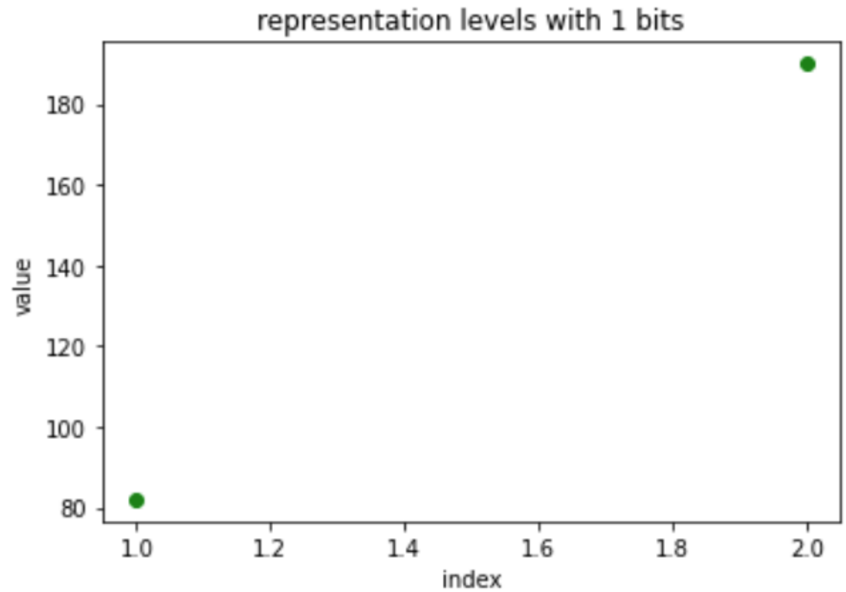

decision levels with 4 bits
value
250
200
150
100
50
0
2
4
6
8
10
12
14
16
index

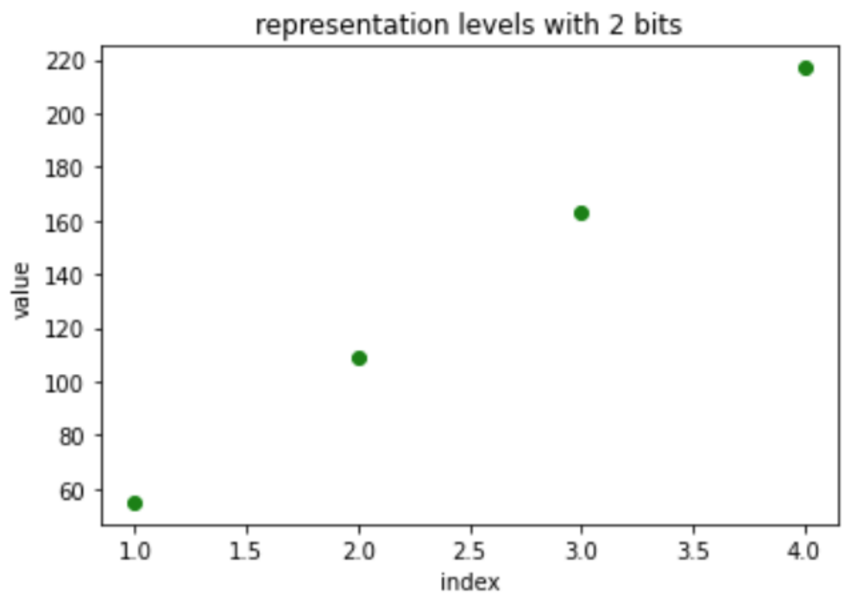

decision levels with 5 bits
value
250
200
150
100
50
0
5
10
15
20
25
30
index

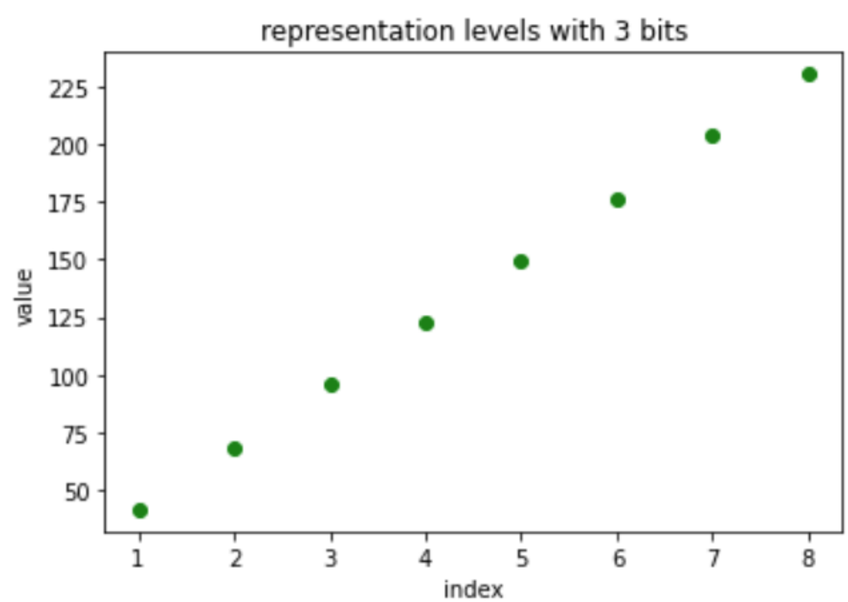

decision levels with 6 bits
value
250
200
150
100
50
0
10
20
30
40
50
60
index

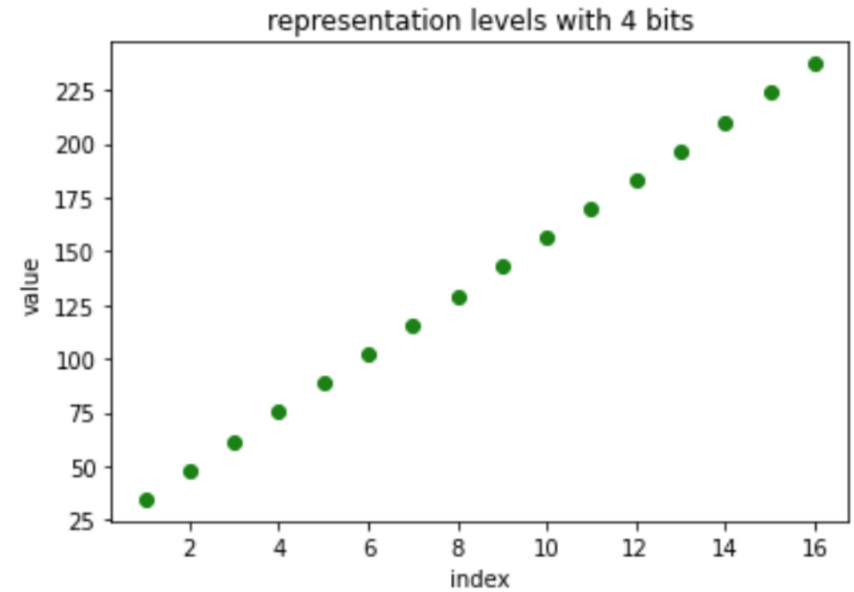

decision levels with 7 bits
value
250
200
150
100
50
0
20
40
60
80
100
120
index


decision levels with 8 bits
value
250
200
150
100
50
0
50
100
150
200
250
index

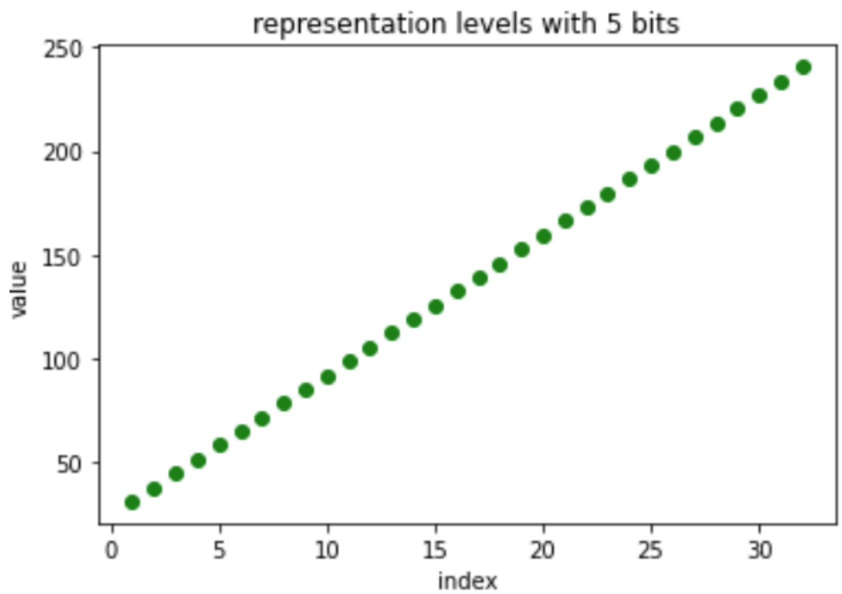

representation levels with 1 bits
value
180
160
140
120
100
80
1.0
1.2
1.4
1.6
1.8
2.0
index


representation levels with 2 bits
value
220
200
180
160
140
120
100
80
60
1.0
1.5
2.0
2.5
3.0
3.5
4.0
index

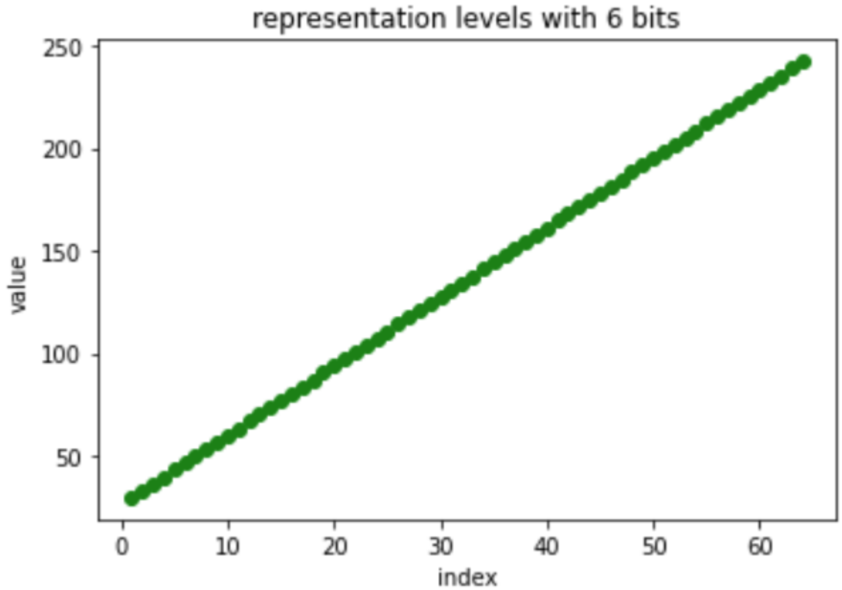

representation levels with 3 bits
value
225
200
175
150
125
100
75
50
1
2
3
4
5
6
7
8
index

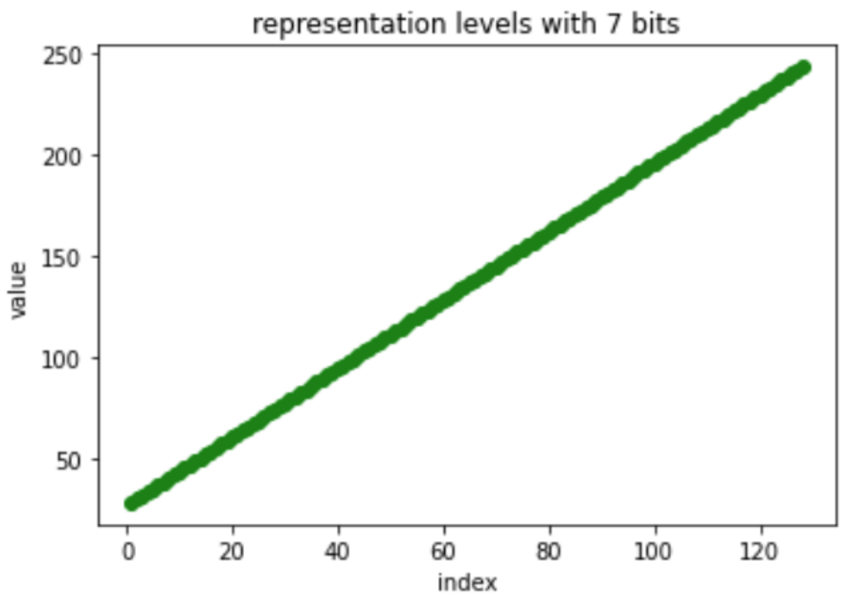


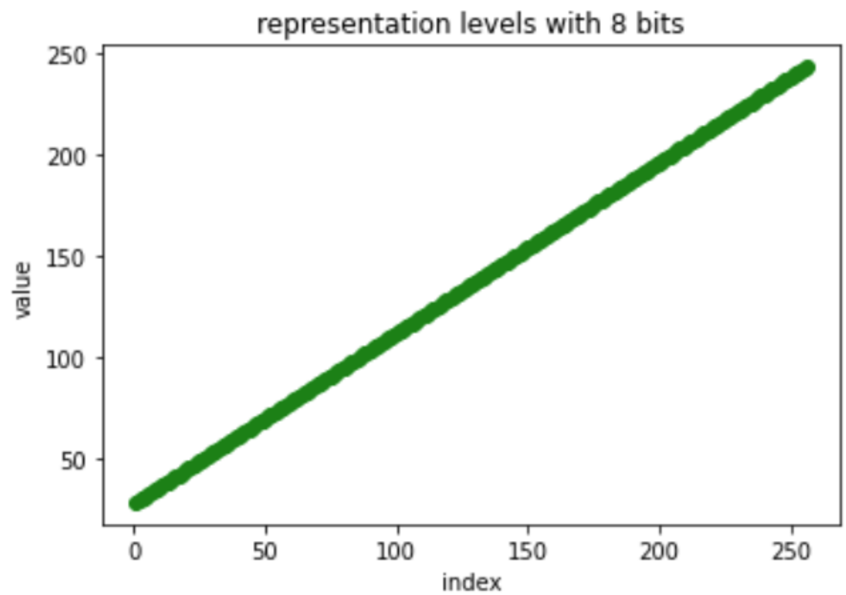
representation levels with 4 bits
value
index



representation levels with 5 bits
value
index


representation levels with 6 bits
value
index
250
200
150
100
50
0
10
20
30
40
50
60


representation levels with 7 bits
value
index
250
200
150
100
50
0
20
40
60
80
100
120

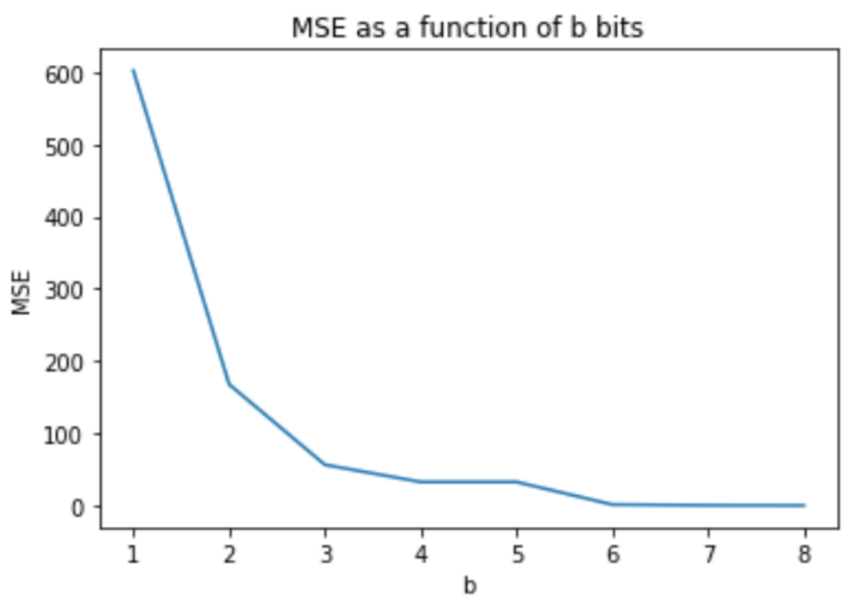


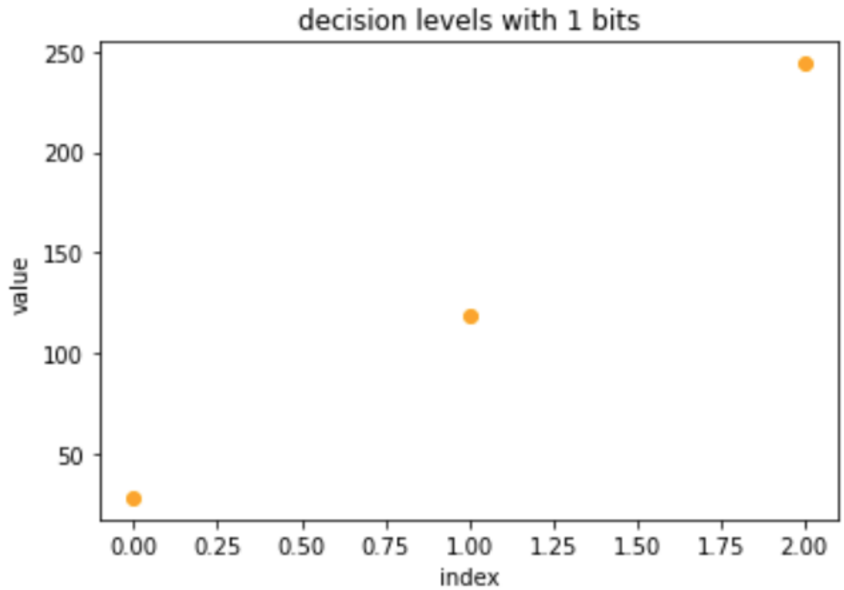
These are all the plots of the representation and desicion levels.

(c) I implemented the Max-Loyd function as needed.

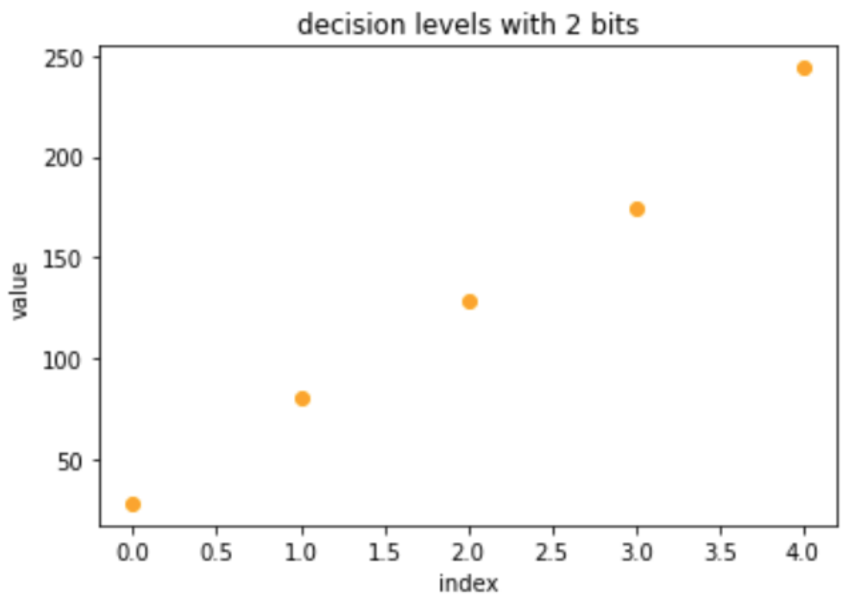
(d) We applied the max-loyd quantizer starting with the uniform quantizer:

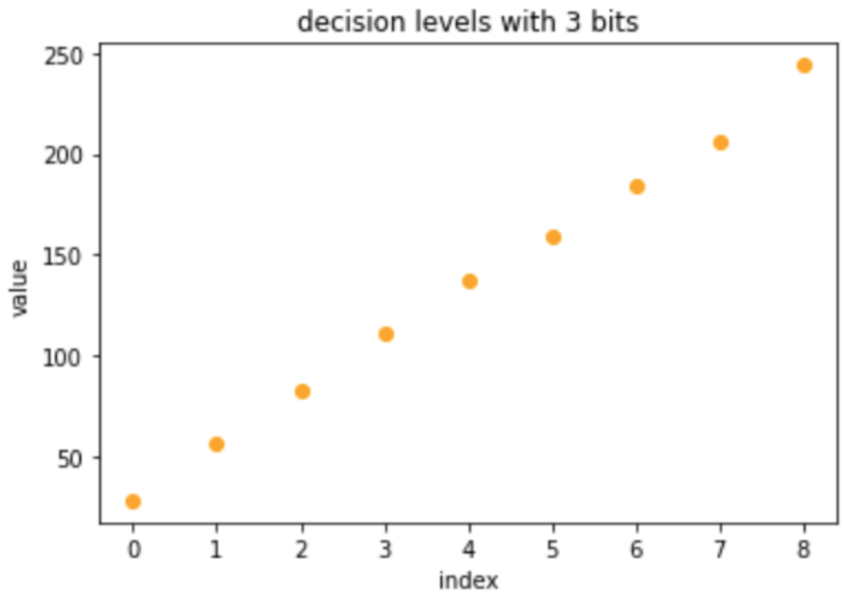
i. MSE as a function of b bits:

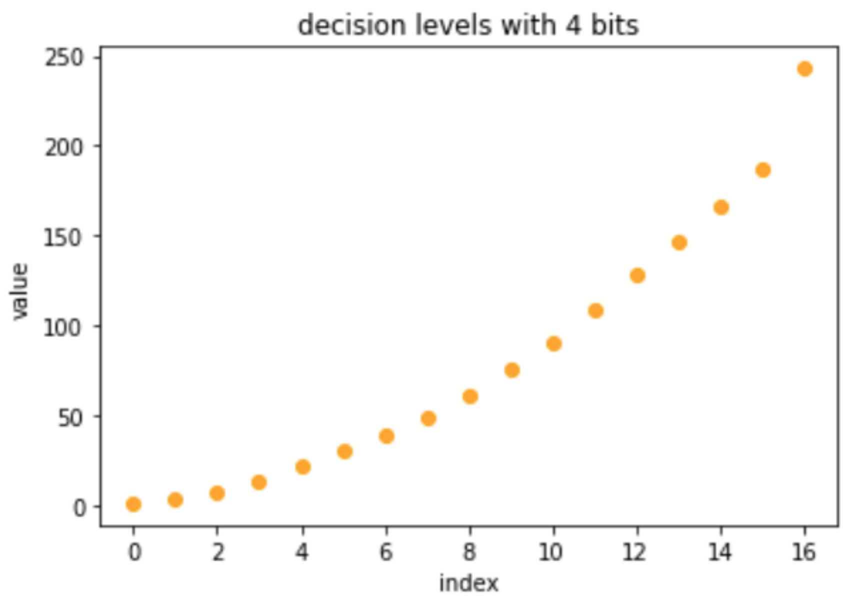


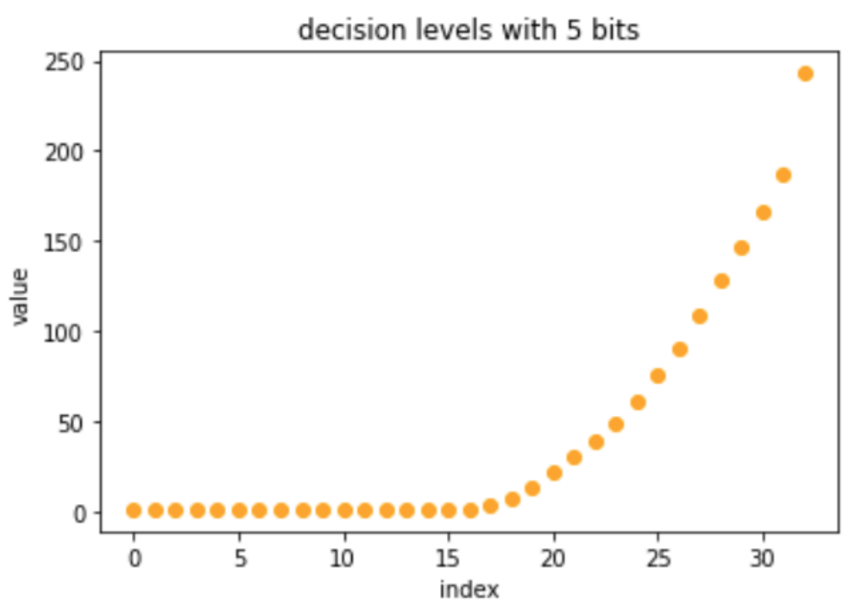

decision levels with 1 bits
value
index
250
200
150
100
50
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

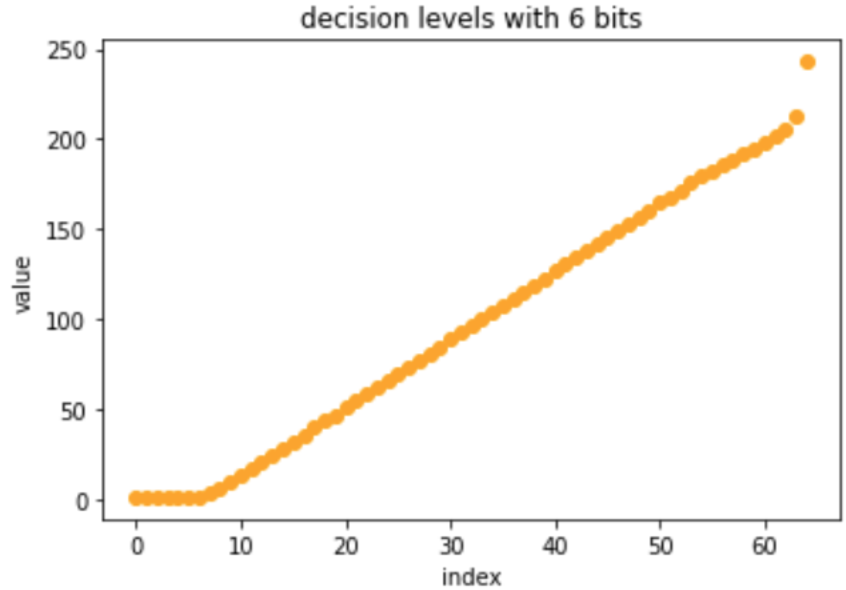
ii.

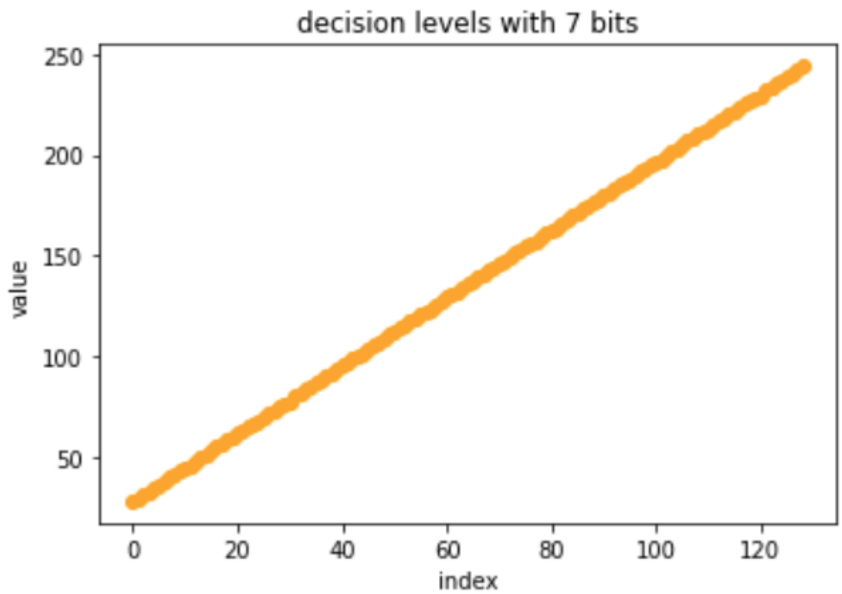

decision levels with 2 bits
value
index
250
200
150
100
50
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

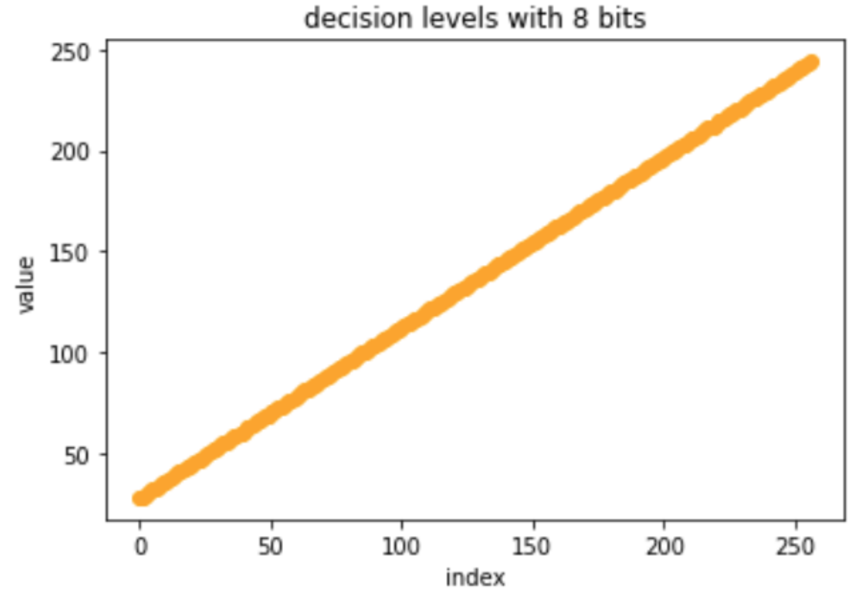

decision levels with 3 bits
250
200
150
100
50
value
0
1
2
3
4
5
6
7
8
index

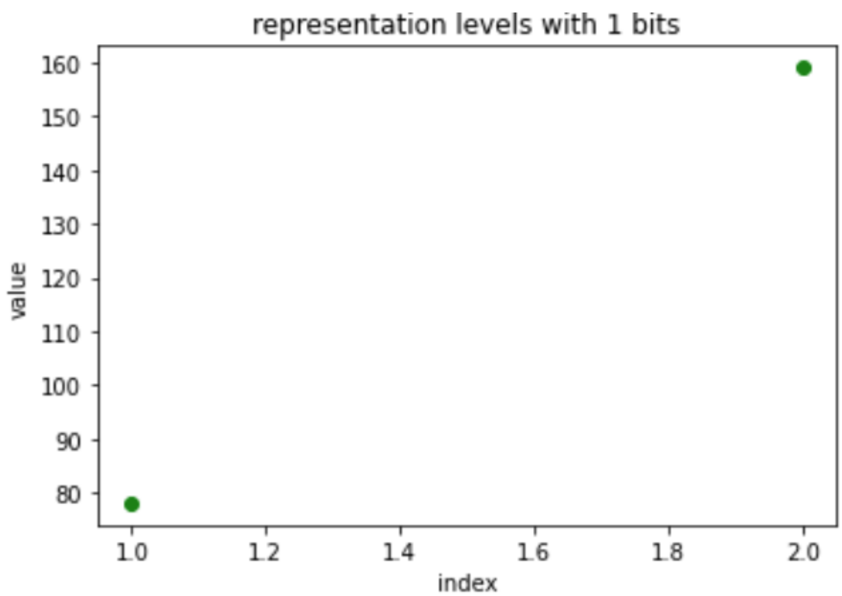

decision levels with 4 bits
250
200
150
100
50
0
value
0
2
4
6
8
10
12
14
16
index

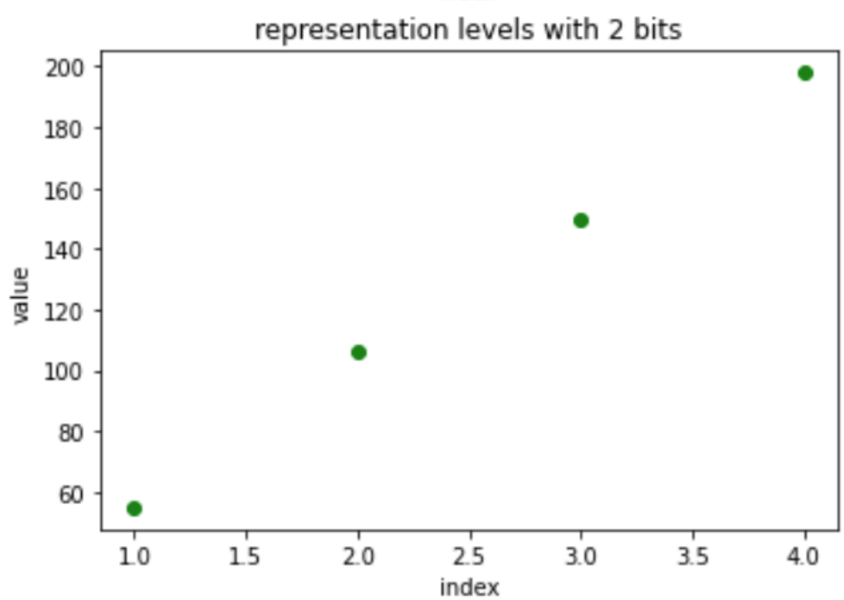

decision levels with 5 bits
250
200
150
100
50
0
value
0
5
10
15
20
25
30
index

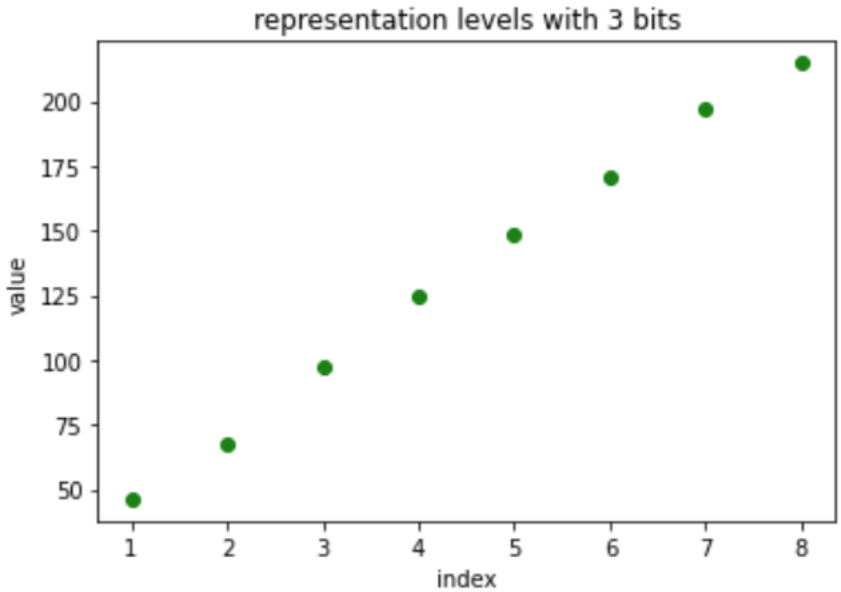

decision levels with 6 bits
value
index
250
200
150
100
50
0
0
10
20
30
40
50
60

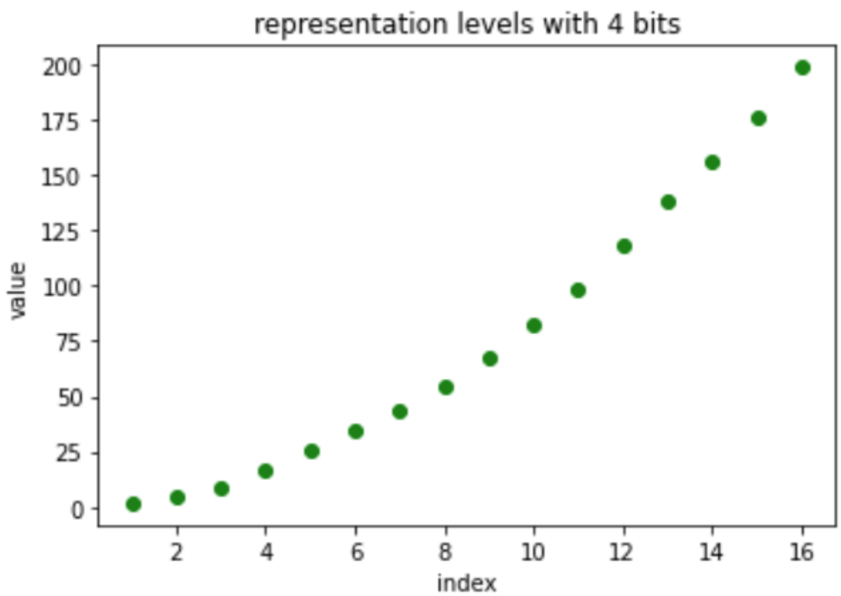

decision levels with 7 bits
value
index
250
200
150
100
50
0
20
40
60
80
100
120

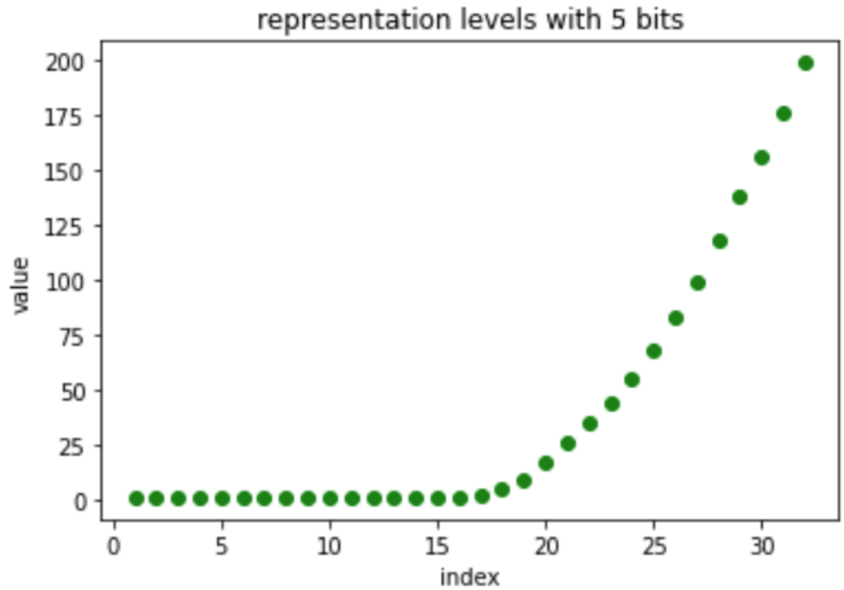

decision levels with 8 bits
value
index
250
200
150
100
50
0
50
100
150
200
250

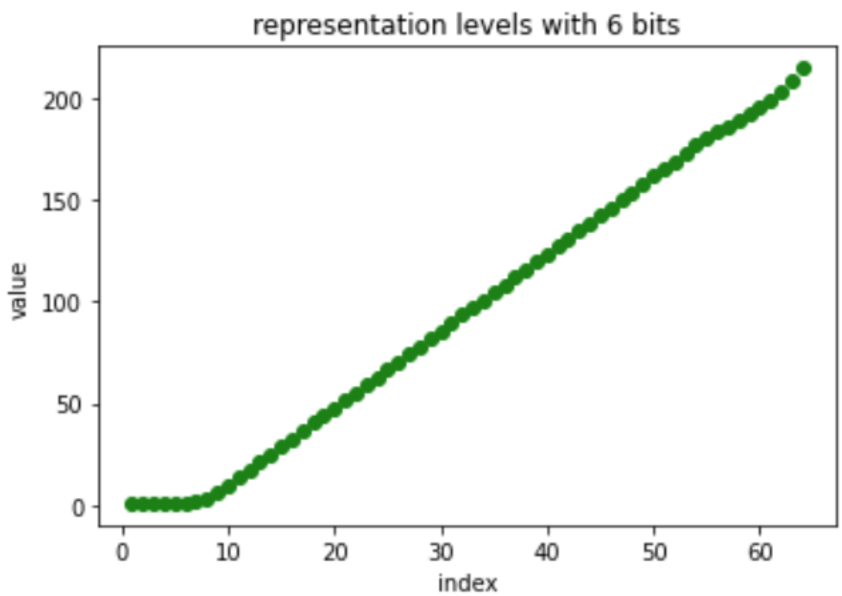

representation levels with 1 bits
value
index
160
150
140
130
120
110
100
90
80
1.0
1.2
1.4
1.6
1.8
2.0

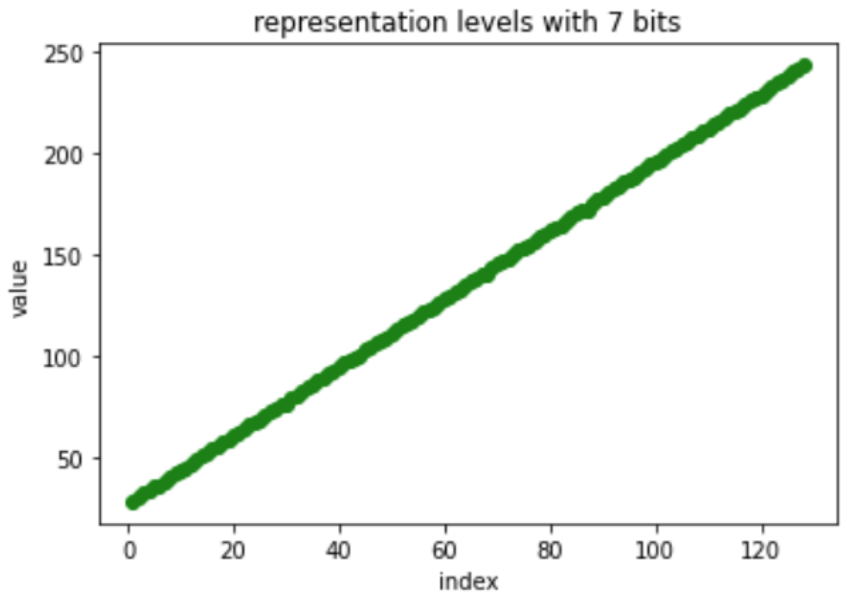

representation levels with 2 bits
value
index
200
180
160
140
120
100
80
60
1.0
1.5
2.0
2.5
3.0
3.5
4.0

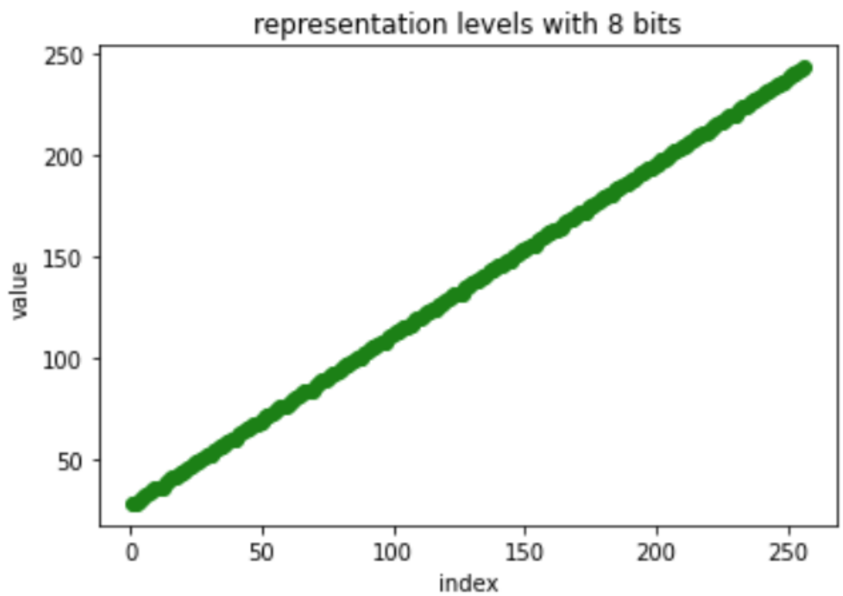

representation levels with 3 bits
value
index


representation levels with 4 bits
value
index

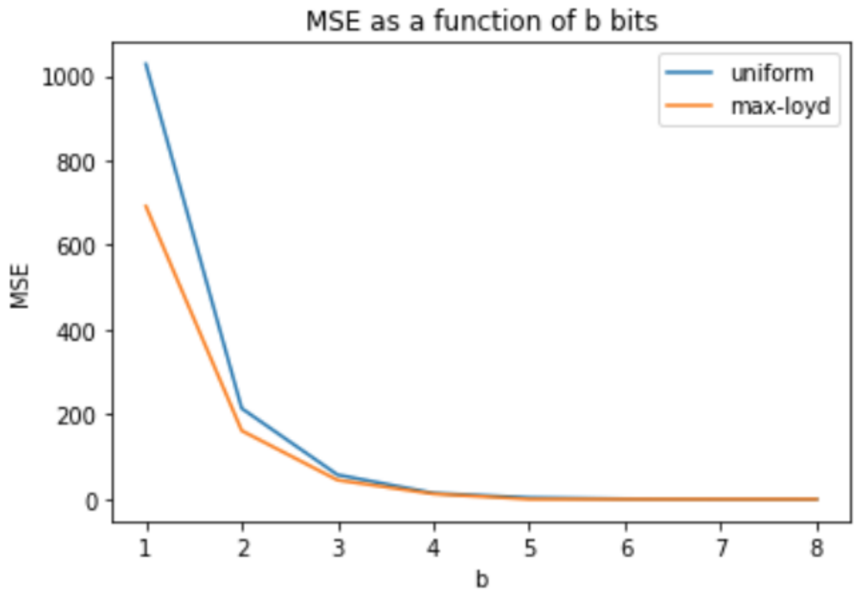

representation levels with 5 bits
value
index
0
25
50
75
100
125
150
175
200
0
5
10
15
20
25
30


representation levels with 6 bits
value
index
0
50
100
150
200
0
10
20
30
40
50
60


representation levels with 7 bits
value
index
250
200
150
100
50
0
20
40
60
80
100
120

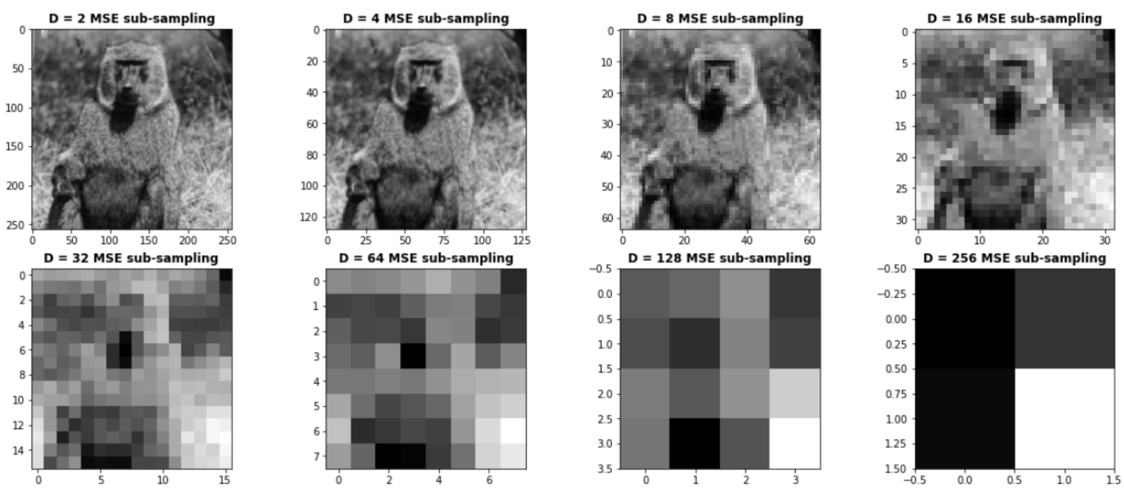

representation levels with 8 bits
value
index
250
200
150
100
50
0
50
100
150
200
250

iii.

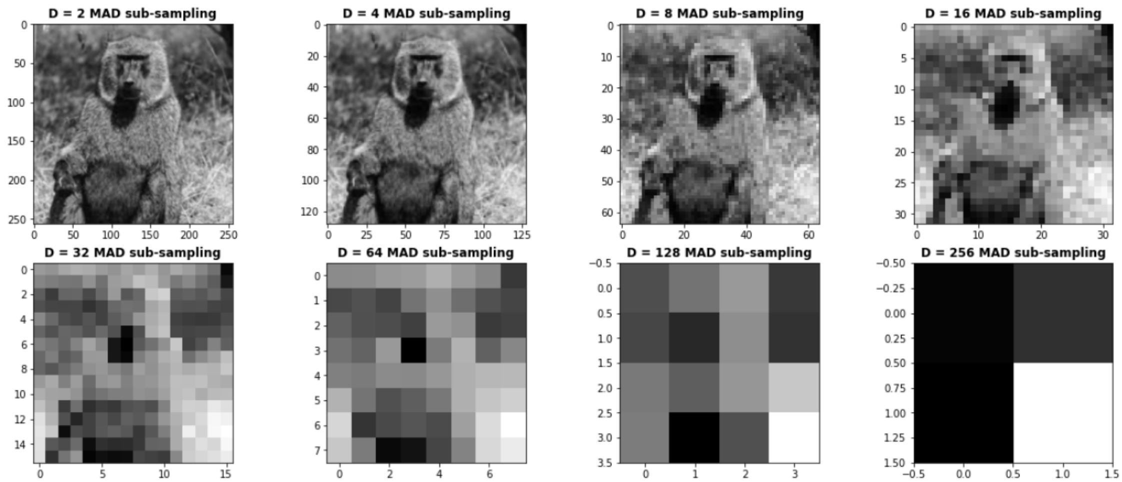


2. Sub sampling and reconstruction:

(a) First, we resized our image and changed its dimensions to 512x512. Then, we refered $D = N_x = N_y = 2^b$, for $b \in \{1, ..., 8\}$. We obtained the following results for the requested $\left\{\tilde{J}_i\right\}_{i=1}^{i=8}$

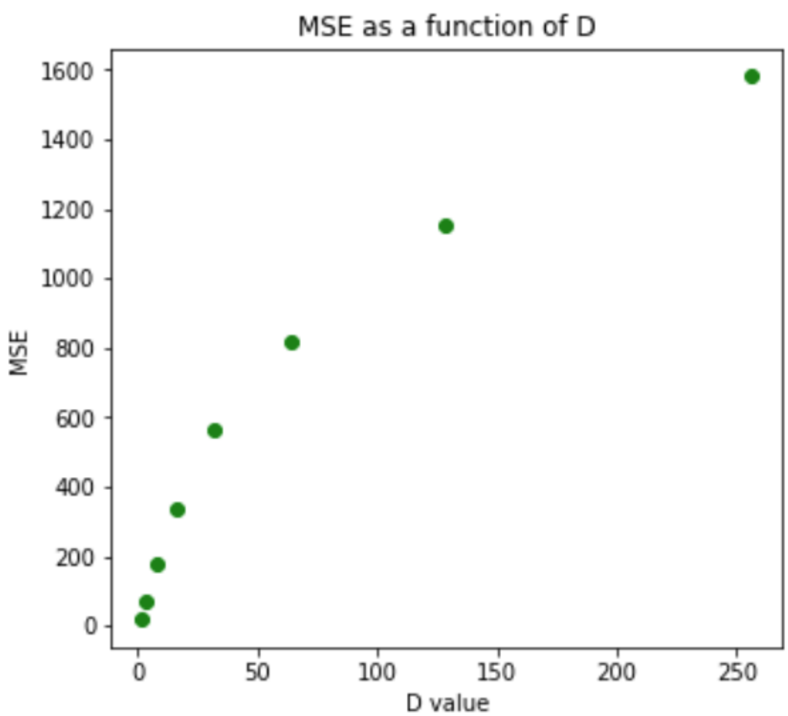


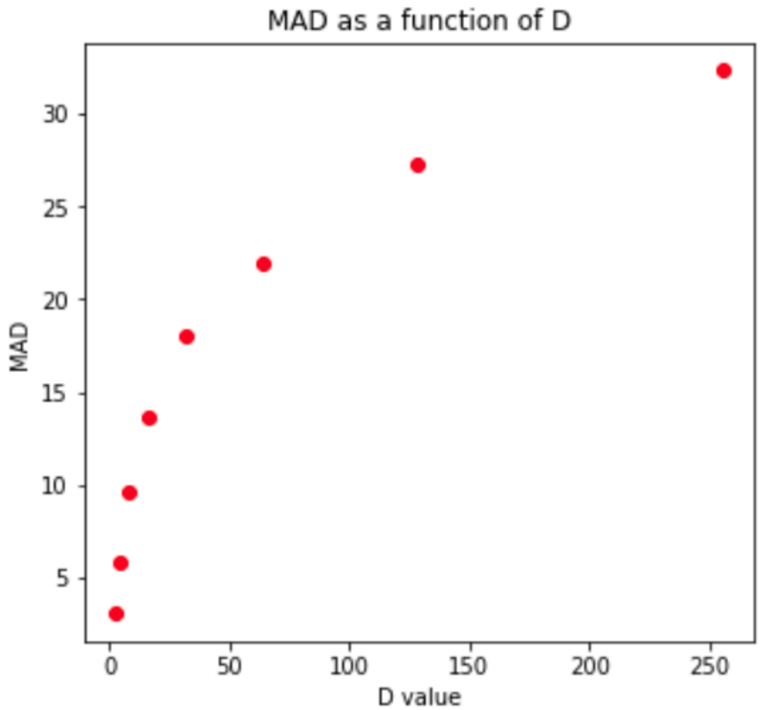
Next, We got the following results for the requested $\left\{\hat{J}_i\right\}_{i=1}^{i=8}$:



Observing the two sets of the sub-sampled images, there are no big differences between them. In addition, the noticable differences can be seen on images with higher D value, where every rectangular grid sample region is bigger, and in contrast - these differences are hardly noticed on images with lower D value, as also the image is getting sharper and clearer.

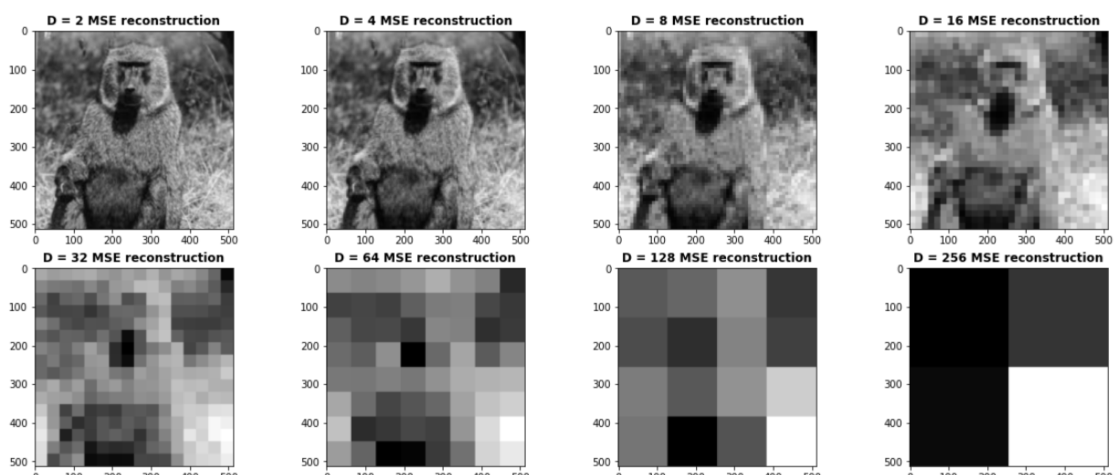
(b) Comparing MSE and MAD of the sub-sampled images, as a function of the factor D, yielded the following graphs:





The results shown above indicate that the integer sub-sampling factor D affects the clarity of the image - for D=2/4/8 it is still possible to identify the contents of the image, for D=16 it becomes harder to do so, and for D=32/64/128/256 we cannot identify its contents. The aforementioned conclusion is relevent for both MSE and MAD senses.

(c) After implementing the standard reconstruction, we received the new sets of images $\left\{\tilde{K}_i\right\}_{i=1}^{i=8}$ and $\left\{\hat{K}_i\right\}_{i=1}^{i=8}$:



3. We'll rewrite the pseudo code from the dry question 1.f for the new problem that is now considered over the space: $[0,1]x[0,1]$.

   (a) Our algorithm, modified:

      i. For each $i \in \{1, ..., N\}$ and $j \in \{1, ..., N\}$ we'll set the interval $I_{i,j} = \left[\frac{i-1}{N}, \frac{i}{N}\right) x \left[\frac{j-1}{N}, \frac{j}{N}\right)$ and do the following:

A. initialize $\hat{f}_i = 0$

B. Calculate $w'_{f_{ij},\hat{f_{ij}}}(x) = min\left\{\frac{1}{\varepsilon}, w_{f_{ij},\hat{f_{ij}}}(x)\right\}$

C. Set $\hat{f_{ij}}^{next} = \frac{\int_{\frac{i-1}{N}}^{\frac{i}{N}}\int_{\frac{j-1}{N}}^{\frac{j}{N}} f(x,y)w'_{f_{ij},\hat{f_{ij}}}(x)dx}{\int_{\frac{i-1}{N}}^{\frac{i}{N}}\int_{\frac{j-1}{N}}^{\frac{j}{N}} w'_{f_{ij},\hat{f_{ij}}}(x)dx}$

D. Use $\hat{f_{ij}}^{next}$ as $\hat{f_{ij}}$ and repeat the calculation from B until absolute value of $\varepsilon^p(f_{ij},\hat{f_{ij}}) - \varepsilon^p(f_{ij},\hat{f_{ij}}^{next})$ is bigger than a given delta .

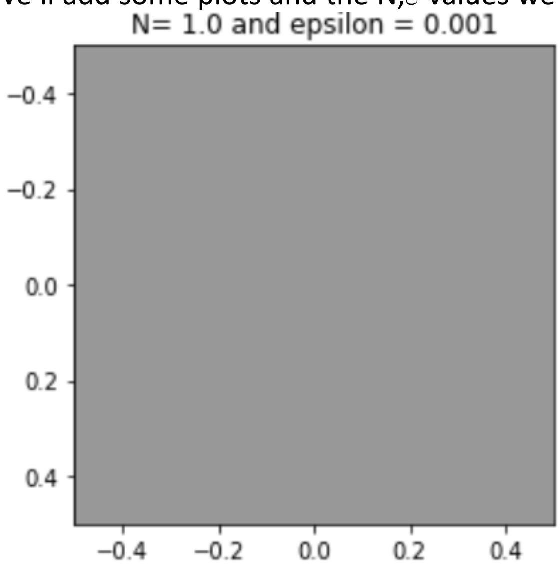
E. set $\hat{f} = \hat{f_{ij}}$

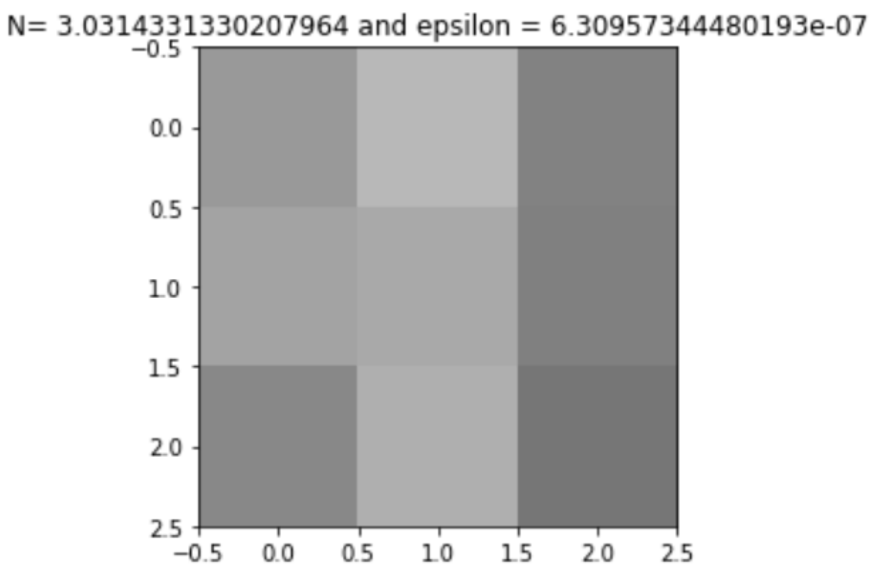
ii. return $\hat{f}(x)$

(b) We implemented the needed function in the code.

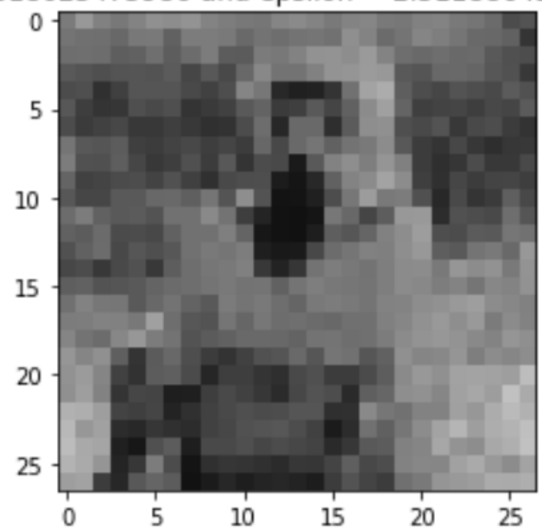
(c) Also this section is in the code.

(d) We'll add some plots and the N,ε values we used in them:

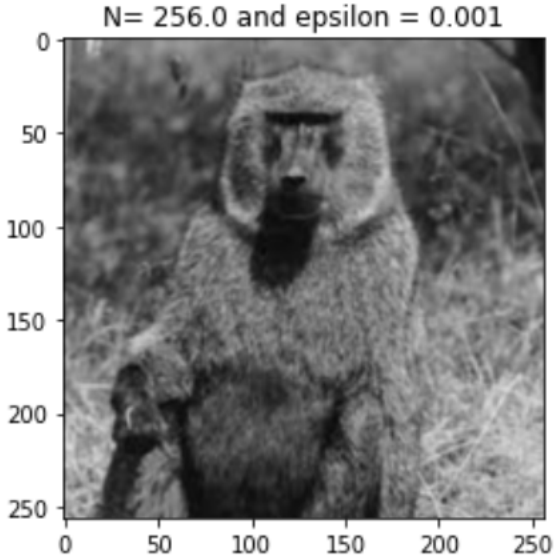




N= 27.857618025475986 and epsilon = 2.5118864315095822e-05



N= 256.0 and epsilon = 0.001

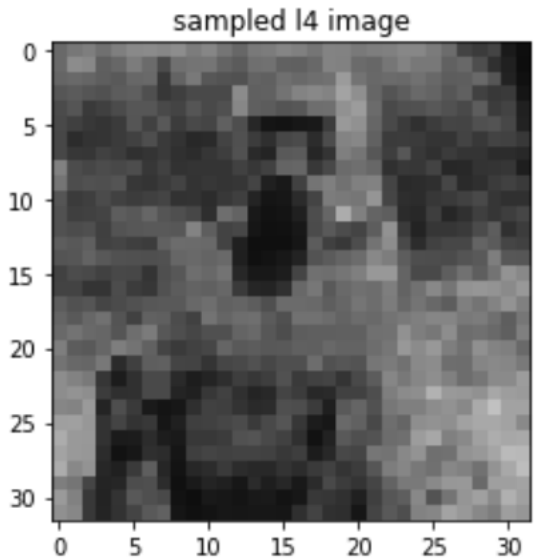


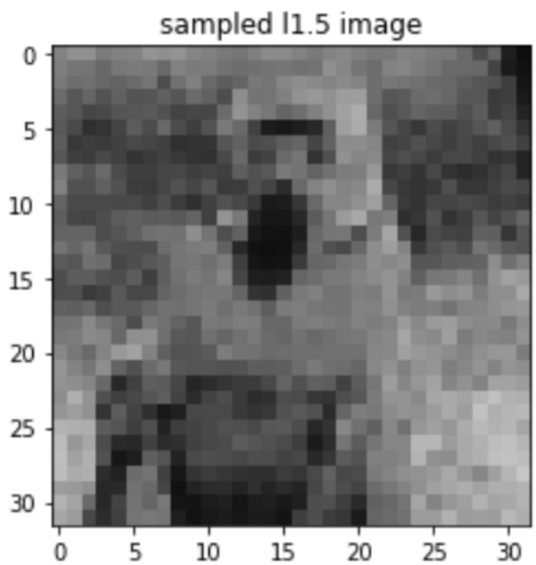
From further experiments we deduce thata when N grows the error decreases which is intuitive because the block size will accordingly get smaller therefore the sampling from it will be more accurate. For small N values we will get really bad results no matter what ε values we'll use.
On the other hand, when increasing ε, we get that $\frac{1}{\varepsilon}$ is smaller and when ε is smaller, $\frac{1}{\varepsilon}$ increases. In here we decrease the values of ε which means that in the step B of the algorithm above, $\frac{1}{\varepsilon}$ gets bigger each time, therefore it will be selected less than the other component $w_{f_{ij},\hat{f_{ij}}}(x)$. This is the wanted result, we want to choose the actual weight and not the estimated "bound" of $\frac{1}{\varepsilon}$.
We can also see in the image below

```
Printing the approximation l1 errors
[0.1409130744380978, 0.1409130744380978, 0.1409130744380978, 0.1409130744380978, 0.1409130744380978, 0.11059847066890466]
Printing the exact l1 errors
[0.11040843888825062, 0.07833815929936426, 0.05020901549096199, 0.029784812179266276, 0.011810018502026355]
```

That the exact l1 calculation using the median is obviously better, which is also intuitive because we don't itirate over intervals until we find a good approximation, we just use the exact calculation.

(e) Below are the picture above, once approximated with the L^4 problem solver and once with the $L^{1.5}$, we used in both $N = 2^5$ and $\varepsilon = 0.001$





and the errors are:

```
error for l4 approximation is 0.0005163381109501241
error for l1.5 approximation is 0.015713388368610092
```

And we can see that the approximation error for L^4 is smaller, which is intuitive because we raise the equation in the power of p, so if we raise number in the range $[0, 1]$ in the power of 4 they will be significantly smaller than in the power of 1.5.