Introduction:

This Report shows the procedure followed and varying effects on the grid by varying the spread parameter sigma at different Epochs.

Sigma(0)={1,10,30,50,70} which decays exponentially.

alpha(0)=0.8, decays exponentially.

Epochs={20,40,100,1000}

Reference Note:

- Epochs=20
- Epochs=40
- Epochs=100
- Epochs=1000

Procedure followed:

Note: ipynb contains the initial cells where the saved model is loaded. Output can be obtained simply by placing imgs.pkl in place and running the code below the first markdown.

- **Color input taken:**

Colors	0	1	2	3
Red	[255,0,0]	[210,0,0]	[160,0,0]	[130,0,0]
Green	[0,204,0]	[102,204,0]	[0,255,0]	[0,102,51]
Blue	[0,76,153]	[0,0,204]	[0,0,255]	[76,0,153]
Yellow	[255,255,0]	[192,186,0]	[255,248,60]	[232,226,44]
Teal	[0,102,102]	[19,146,146]	[16,168,168]	[34,195,195]
Pink	[255,0,255]	[255,153,255]	[204,0,204]	[255,61,243]

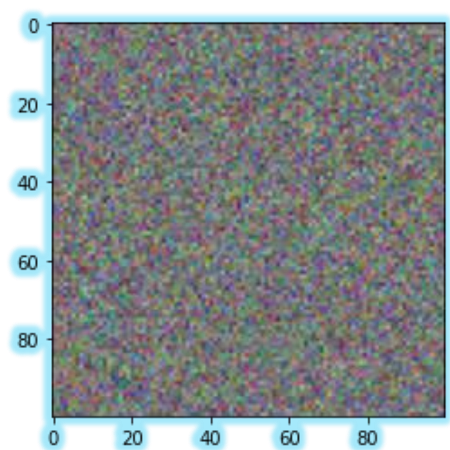
- **Algorithm:**
  - **Dataset generation**: (24,3) dimensional dataset is created set to above values and normalized to {0,1}.
  - **Weights Initialization**: Randomly initialized (100,100,3) for each input initially. These weights will suffice for every incoming input.

- **Distance function:** To calculate the distance between the input vector and the weight vector mapping to each 100,100 neuron in feature map.
- **Index function:** To get the index with the minimum distance.
- **Gen_image:** To generate the image at given number of epochs
- **Train:**Update weights at weights[j,k,:] ex: (1,1,:) between the input and a particular neuron and return the images at given set of epochs.
- **Som:** calls train and takes epochs,sigma.

- <u>**Note**</u>:
  - **Sigma can be changed to required value at the markdowns specifying sigma after the function (SOM).**
  - **Model is saved using pickle and loaded at the beginning as well as the end.**
  - **Imgs.pkl file is uploaded along with the ipynb.**

ANALYSIS

Figure of the original grid with randomly initialized weights.



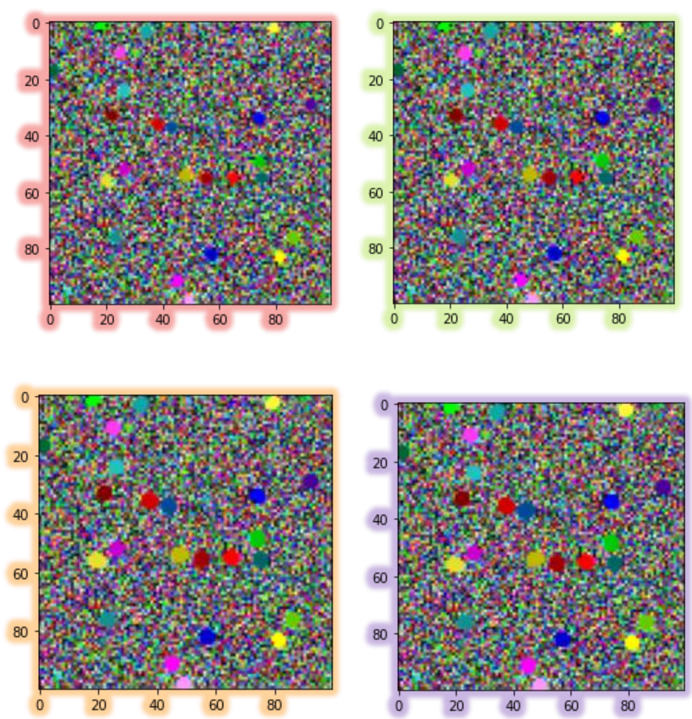
- Initially the weights are randomly initialized and no update have been made so far by minimizing the distance between the input set and the weights.
- **Result:** All the colors are distorted over the map as the weights.

Each row representing Epochs={1,10,30,50,70}, Deacy is :

sigma=1	sigma=10	sigma=30	sigma=50	sigma=70
0.98	9.8	29.4	49	68.6
0.96	9.6	28.8	48	67.2
0.90	9.0	27	45	63
0.36	3.6	10.8	18	25.2

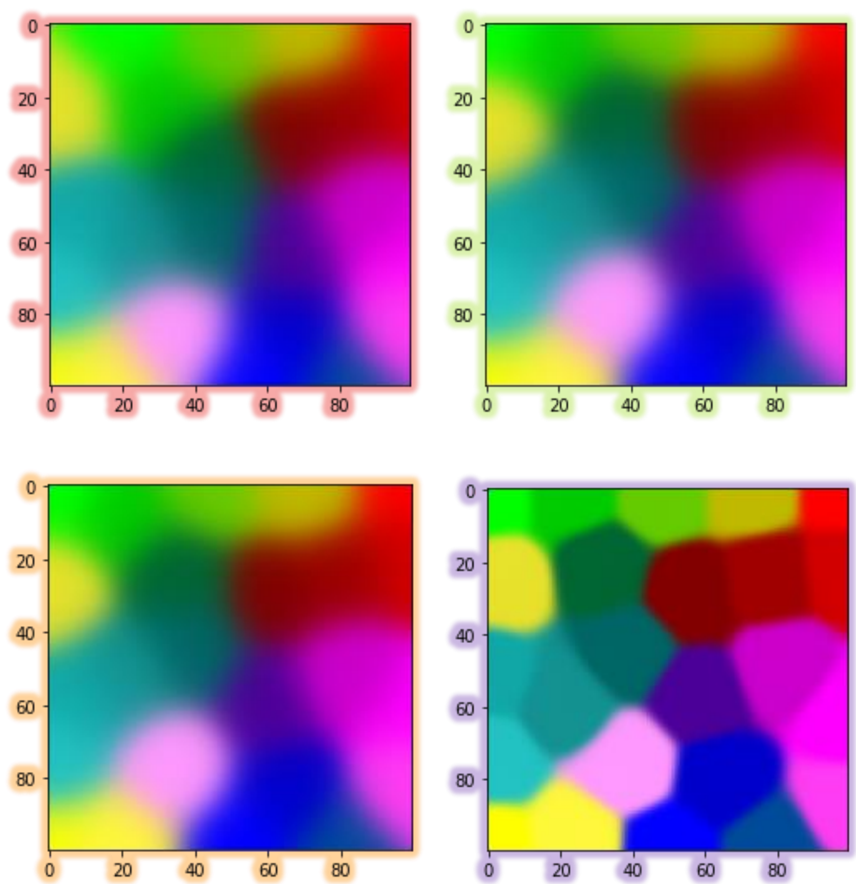
1. SOM at sigma=1: {radius of different neighbors on SOM.}

- **It can be seen that sigma decays from 0.98 to 0.36 between 20 and 1000 epochs as the k increases.**
- **Neighborhood around the winning candidate decreases at each k as sigma decays and the convergence occurs in general.**
- **In case of sigma=1, Neighborhood is very small initially and consideration of many neurons in the map are not taken up and from there it shrinks more to the winning node which generally takes all. Thus, for each input all the weights may not get updated as required ranging from epoch 20 to 1000 as sigma shrinks which is seen in the images below.**



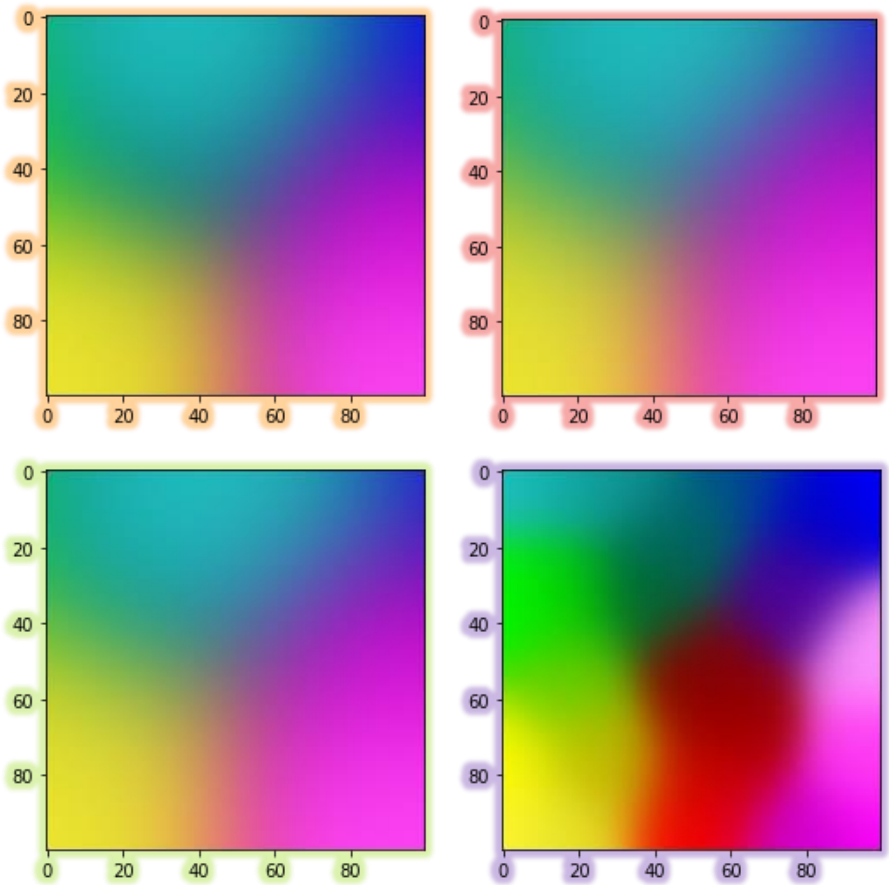
2. SOM at sigma=10:

- It can be seen from the table that sigma decays from 9.8 to 3.6 as the epochs proceed from 20 to 1000.
- In case of sigma=10, Neighborhood is not very small initially covering good number of neurons and it reduces to minimum at 1000 epochs when sigma is 3.6 and convergence occurs which takes decent number of neurons in consideration at each weight update. This gives us promising results.



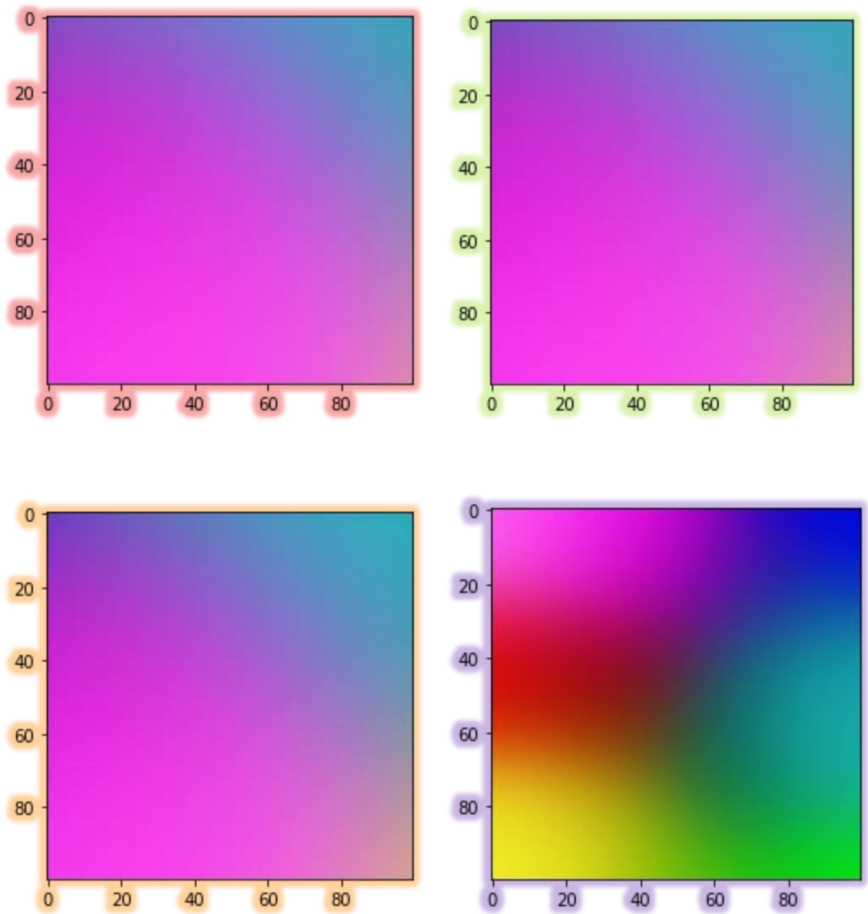
3. SOM at sigma=30:

- Sigma decays from 29.4 to 10.8 as the epochs proceed from 20 to 1000.
- Thus, sigma reaches minimum of 10.8 and it covered well number of neighbors in the beginning for updates on weights and converges well with large clusters of RGB values given the value of sigma=10.8 at 1000th epoch.



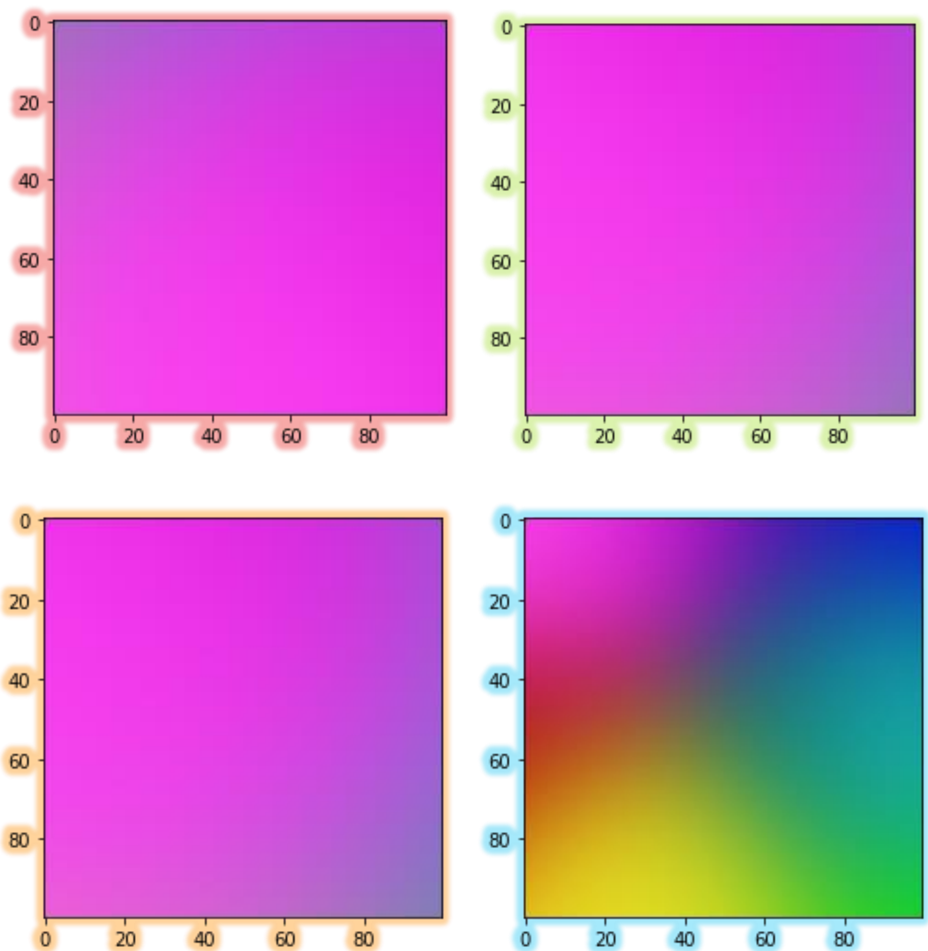
4. SOM at sigma=50:

- Here sigma decays from 49 to 18 within the given range of epochs {20,1000}.
- At sigma=49, large number of neighbors are getting updated on weights for each input which can be seen in the initial epochs with large values of sigma and it converges when the sigma is reduced to optimal i.e. 18 for this much initial spread.
- Colors can be seen separated at sigma=18 at 1000th epoch.



5. SOM at sigma=70:

- When taking sigma(0)=70, it decays exponentially from 68.6 to 25.2 over the period of k ranging form [20,1000].
- At such big value of sigma, much larger neighborhood is considered, and weights were updated accordingly which resulted in a single colored image as shown below for initial epochs [20,40,100]. Thus, when the epochs reached 1000 and the sigma decayed to 25.2, a much better map can be seen than the rest.



Extreme cases:

1. At sigma=1, small number of neighbors are considered for weight update initially and most are not taken into account which gives not so promising results in the map.

2. At sigma=70, very large number of neighbors are taken into account initially for which large number of weight updates are done until the sigma decays at large number of epochs. Thus till 100 epochs when plotted sigma decays to 63 and it still gives single colored 2d map.

Overall, it can be seen that the optimal value of the sigma here can be considered where even at few epochs and at large number of epochs, the optimal sigma resulted in optimal update of the weights for the neighborhood which in turn made the map converge towards the center. Initially the neighborhood can be taken as decently large but at the end with incremental epochs, as the sigma decays, the convergence should occur towards the center and winner will take all.