UNIVERSIDADE FEDERAL DE SÃO PAULO
INSTITUTO DE CIÊNCIA E TECNOLOGIA
BACHARELADO EM CIÊNCIA E TECNOLOGIA

Projeto 02:

Implementação *Self Organizing Maps*

Nome: Willian Dihanster Gomes de Oliveira **RA:** 112269

SÃO JOSÉ DOS CAMPOS
2018

Implementação

A implementação dessa rede foi baseada no pseudocódigo disponibilizado pelo professor nos slides de aula, sendo traduzido para Python 3. Sendo assim, o grid possui dimensões *nxmxp,* sendo *n* e *m* definido por parâmetros, já o *p* é definido pelo número de atributos.

Bases de Dados

- **Iris Dataset**

  O *Iris Dataset* é uma das bases mais famosas utilizadas na literatura. Ela é composta por dados de 3 flores diferentes, com 4 atributos, além disso, possui 150 instâncias.

- **Cores RGB**

  Este é um conjunto de dados gerado manualmente. Isto é, foram gerados 100 exemplos com 3 atributos cada (com valores entre 0 e 255 aleatoriamente), que representam os componentes R, G e B.

- ***Jain's Toy Problem***

  É um *dataset* muito utilizado para clusterização, que possui dois clusters, conforme a Figura 1.

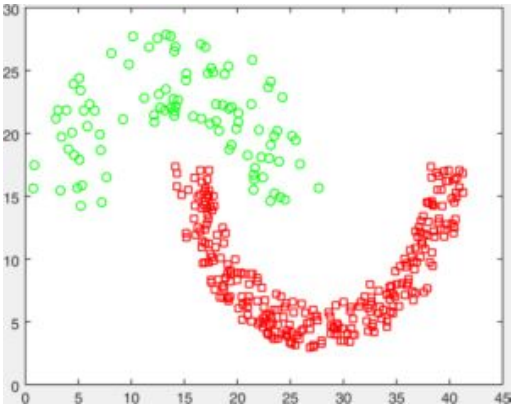

Figura 1: Demonstração do *dataset Jain's Toy Problem*

Configurações e Parâmetros Iniciais

- Primeiramente, o conjunto de dados utilizado para o experimento é pré-processado para retirar seu atributo de classe/cluster (para os *dataset* que possuem informação de classe ou de cluster a qual pertence), além disso, os dados são normalizados.
- Os pesos da redes são gerados aleatoriamente entre 0 e 1.
- A métrica da distância utilizada foi a distância euclidiana.
- Máximo de Épocas (max.épocas) = 500
- *Learning Rate* inicial η_0 = 0.3
- Sigma inicial σ_0 = 10
- Constante de Tempo $\tau = max.épocas/log(\sigma_0)$
- O tamanho do *grid* foi variado.

Resultados e Discussões

- **Conjunto de Dados 1: *Iris Dataset***

Com os parâmetros descritos acima e com um grid de 10x10 e 15x15, respectivamente, na Figura 1 a seguir.

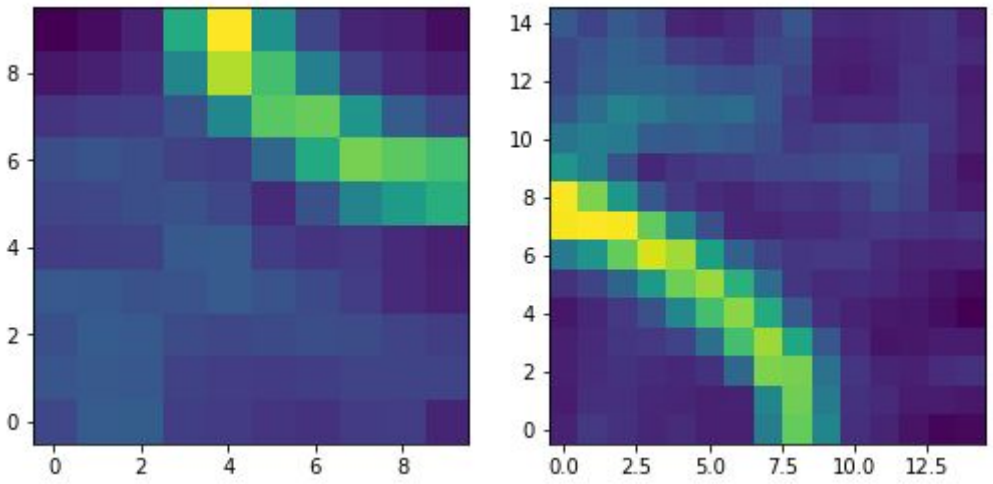

Figura 2: Plot da Umatrix para o *dataset* Iris.

Analisando ambas os gráficos da Figura 1, pode-se notar que para ambos tamanho do grid, houve uma boa separação dos clusters, conforme o esperado para o Iris, isto é, uma região bem separada (Cluster 1), e dois clusters mais misturados (Clusters 2 e 3), separados pela região de fronteira em tons de amarelo e verde.

- **Conjunto de Dados 2: Cores RGB**

Para esse *dataset,* é interessante a visualização que se pode ter de que o método funciona, como mostra a Figura 3, onde é possível visualizar a rede 15x15 antes do treinamento e depois do treinamento, respectivamente.

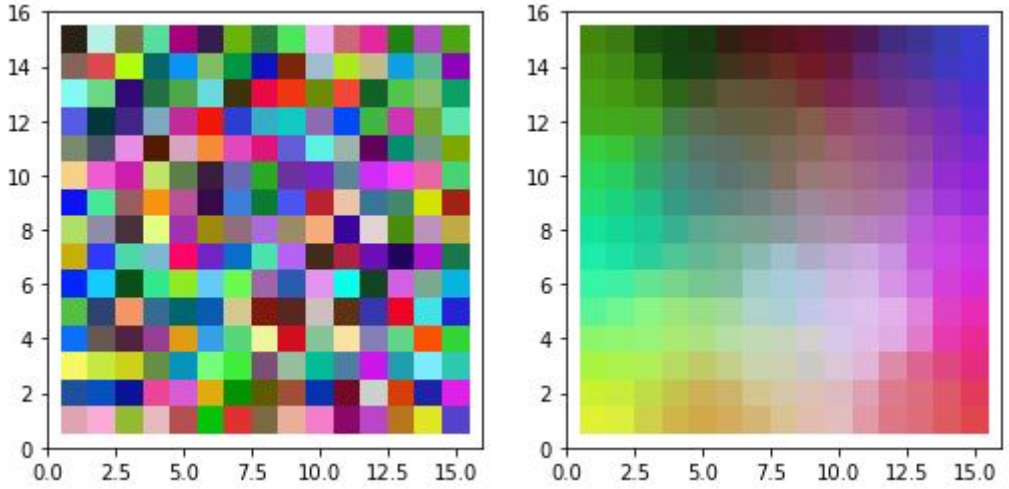

Figura 3: Rede para a Cores RGB antes e depois do treinamento.

Pode-se notar que a rede conseguiu agrupar bem as cores parecidas em comparação a base original. Assim, confirmando a eficiência da Rede SOM. Podemos ver também na Figura 4, a umatrix da rede.

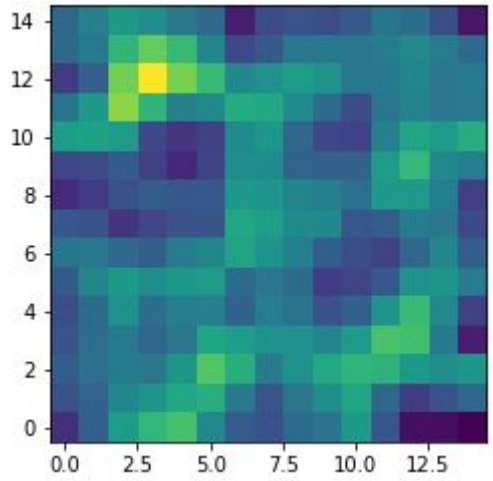

Figura 4: Umatrix geradas para o *dataset* de Cores RGB.

Analisando o resultado da umatrix, pode-se notar a presença de vários clusters, como o esperado para as cores rgb.

- **Conjunto de Dados 3: *Jain's Toys Problem***

Para o conjunto de dados 3, temos os resultados da Figura 5, para a variação no número de grids, para 10x10 e 15x15.

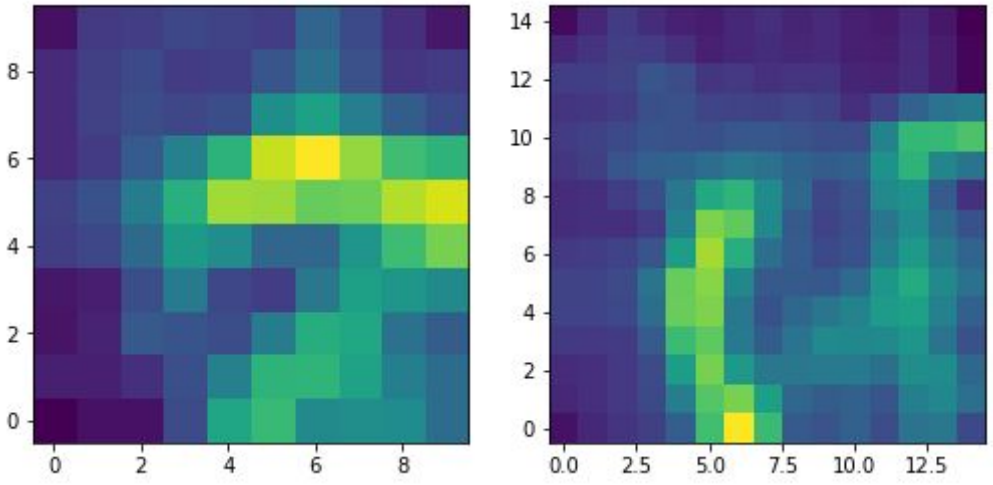

Figura 5: Umatrix geradas para o *dataset Jain's Toys Problem.*

Com a variação no tamanho do grid, houve um leve diferença, nos grids gerados. O grid de 15x15 pode fazer mais sentido, levando em conta que parece que houve uma maior separação entre dois clusters, com uma fronteira no meio "incompleta", o que pode nos levar a acreditar que seja os pontos do *dataset* que quase se intersectam, ou seja, estão muito perto, levando a um confusão.

Conclusões

Sendo assim, pode-se concluir a eficiência das redes SOM - *Self Organizing Maps* como um ferramenta de visualização, organizando dimensionalmente dados complexos em clusters.