Briefing about the experiment

The aim of this project was to give an insight of the main ideas behind how SOMs are mapping the colors from their three-dimensional components - red, green and blue, into two dimensions.

Although a color is rendered by the computer using values for each component (red, green and blue) from 0 to 255, the input vectors have been adjusted so that each component has a value between 0 and 1. (This is to match the range of the values used for the node's weights).

We can choose to use the training set shown in Fig. 1 or a training set made up of random colors by uncommenting the line `"//#define RANDOM_TRAINING_SETS"` found in `"constants.h"` file.

It's obvious that different values for the number of iterations and the initial learning rate would affect the algorithm.

For my experiment I considered the number of epochs desired for the training to be 1000, while the value of the learning rate at the start of training is 0.1.

The initial configuration in the experiments has not been considered a determinant factor for the results. The initial weights tended only to produce, in general, symmetric or rotated versions of an equivalent map, with the relation among final values neighbors' units being maintained. The visual inspection of the produced maps, represented by the "colors" of the weights, lead to this conclusion. The term "colors" was used in the sense that each cluster unit has a $3D$ vector for weight representation, with each vector component related to a chromaticity component (r, g, and b, respectively); so, such information was used to visualize the network array on a color display.

The number of samples used to generate a feature map can range from approximately 800 to 15000, with coherent results. Fewer samples lead to uncomplete maps, in a sense that one more sample could make change the current weight configuration significantly. Using more than 15000 samples for the network training would produce configuration slightly different from the current state to the next. In this situation the neighborhood would be already reduced to include only the winning cluster unit and the gain factor could be reduced to have almost zero influence on the winning unit. This described situation was attained by parameters settings actually used, such as the number of nodes, decreasing factor for neighborhood size and gain factor, among others. These settings yielded good results. On the other hand, no extensively experiments were considered with lower decreasing ratios for neighborhood size and

gain factor in a way that the influence of both neighborhood and gain factor could still be significant above 20000 or even more samples.

Further Work

In future, it would be interesting to experiment with different decay functions for the learning rate, neighborhood radius and neighborhood influence. There are large speed gains to be made there.