# Generalised Decrease Schedule of Learning Parameters for SOMs

Prashant Kalvapalle, Shoeb Mohammed, Ragib Mostofa

**ProjectProposal**

## 1 Overview

In the realm of unsupervised learning, self-organizing maps (or SOMs) are easily one of the most widely utilized and highly regarded neural network paradigms. First proposed by Finnish professor Teuvo Kohonen in the 1980s, SOMs have routinely succeeded in outperforming non-neural machine learning algorithms, such as the k-means, when applied to clustering problems of high significance, even in fields other than computer science. As a result, there is no debate concerning the high propensity towards SOMs for unsupervised learning.

However, there does exist several aspects of the traditional SOM algorithm that could benefit from improvements. Examples include the learning speed and neighborhood radii of the self-organizing map, where the decrease schedule of these learning parameters is often used with predefined values that have no relation to the input data. Hence, the SOM is not able to perform at its fastest when applied. Our project aims to determine a solution to this problem - to optimize the convergence speed of self-organizing maps by generalising the decrease schedule of learning parameters such as the learning rate and the neighborhood radius with respect to the input data space.

## 2 Method

In their paper about the convergence of SOMs, Yin and Allison have proven that the neurons of an SOM and the data points of an input space should exhibit the same distribution characteristics, assuming that the SOM size and training period are sufficiently large. Using this idea and further assuming that each of the features of the input space are drawn from a Gaussian distribution and are independent of each other, Lutz Hamel has proposed a statistically efficient and computationally cheap metric to measure the map embedding accuracy, which is defined as follows.

$$ea = \frac{1}{d}\sum_{i=1}^{d} \rho_i,$$

where

$$\rho_i = \begin{cases} 1 & \text{if feature i is embedded} \\ 0 & \text{otherwise} \end{cases}$$

and $d$ is the number of features in the input space. Whether or not a feature is embedded is determined using z-scores, hypothesis testing and two-sample tests. In essence, the embedding accuracy is simply the fraction of features that have been successfully embedded by the SOM and hence, this metric ranges from 0 to 1 with 1 being the desired value. Moreover, the metric above can be improved by using Bayesian estimates of a features relative importance to weight that particular feature's $\rho_i$ term.

While the embedding accuracy is a good metric to evaluate convergence, we are able to achieve a better picture by coupling it with another metric that measures topological quality. Addressing this issue, Lutz Hamel describes the topographic error in his paper as,

$$te = \frac{1}{n}\sum_{i=1}^{n} err(x_i)$$

with

$$err(x_i) = \begin{cases} 1 & \text{if } bmu(x_i) \text{ and } 2bmu(x_i) \text{ are not neighbors} \\ 0 & \text{otherwise} \end{cases}$$

for training data $x_1, x_2, ..., x_n$ where $bmu(x_i)$ and $2bmu(x_i)$ are the best matching and the 2nd best matching unit for the input pattern $x_i$ on the map, respectively. Hence, topographic accuracy is defined as,

$$ta = 1 - te$$

The goal of our course project is to utilize these two measures simultaneously to derive a generalised decrease schedule for the training of a SOM. Notice that since the topographic accuracy is measured based on the input patterns, the decrease schedule we propose will have taken into account the distribution of the input data. We also plan to test the extent of the assumption involving the Gaussian distribution of the independent features in the input data space. We plan to achieve this by applying our decrease schedules on SOMs that are trained on data with features exhibiting varying degrees of normality. Furthermore, since past research has shown that smooth exponential decay of the learning rate fails to significantly boost the convergence speed, we will refrain from using such models. Hence, the decrease schedules we employ are expected to be stepwise in shape.

# 3 Data Description

In order to provide empirical evidence that suggests that our algorithm indeed decreases the number of learn counts required for a self-organizing map to converge, we plan on testing our algorithm on two different data sets and reporting the relevant convergence statistics. The data sets we currently plan on using are Fisher's Iris Flower data set and Human Activity Recognition Using Smartphones Data Set.

## 3.1 Fisher's Iris Flower Data Set

For our first data set, we intend to use Fisher's Iris Flower Data Set which contains data that is reasonably simple. Important information concerning this data set has been presented on page 3.

## 3.2 Human Activity Recognition Using Smartphones Data Set

For the second data set, we plan on using the Human Activity Recognition Using Smartphones Data Set. This is data set is considerably more complex than the Iris data set we had proposed above. Important information concerning this data set has been presented on page 4.

Table 1: Data Description of Fisher's Flower Iris Data Set

| | |
|---|---|
| **Origin of Data** | |
| Source | UCI Machne Learnng Repository website |
| URL | http://archive.ics.uci.edu/ml/datasets/Iris |
| **Data Characteristics** | |
| Background | Multivariate data containing different classes of the Iris flower plant and introduced by Ronald Fisher in 1936 |
| Associated problem | Classification |
| Attribute characteristics | Real numbers |
| Number of data points | 150 |
| Number of attributes | 4 |
| Number of classes | 3 |
| Number of labeled data points | 150 |
| Number of labeled samples per class | 50 |
| **List of input/output features** | |
| **Input Features** | |
| Sepal Length | [min, max] = [0,1], length of sepals, in cm, of the different classes of iris flowers |
| Sepal Width | [min, max] = [0,1], width of sepals, in cm, of the different classes of iris flowers |
| Petal Length | [min, max] = [0,1], length of petals, in cm, of the different classes of iris flowers |
| Petal Width | [min, max] = [0,1], width of petals, in cm, of the different classes of iris flowers |
| **Output Classes** | |
| Class Iris Setosa | |
| Class Iris Versicolour | |
| Class Iris Virginica | |
| **Input/Output Encoding** | |
| Input Encoding | No encoding is required since inputs are already numerical |
| Output Encoding | The three classes are encoded by assigning them 3 different IDs. Class Setosa is encoded as 1, class Versicolor is encoded as 2 and class Virginica is encoded as 3. |

Table 2: Data Description of Human Activity Recognition Using Smartphones Data Set

| | |
|---|---|
| **Origin of Data** | |
| Source | UCI Machne Learnng Repository website |
| URL | $http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones$ |
| **Data Characteristics** | |
| Background | Multivariate and time-series data of signals emitted by smartphones worn by subjects performing 6 classes of activities |
| Associated problem | Classification and Clustering |
| Attribute characteristics | Real numbers |
| Number of data points | 10299 |
| Number of attributes | 561 |
| Number of classes | 6 |
| Number of labeled data points | 10299 |
| Number of labeled samples per class | 1715 |
| **List of input/output features** | |
| **Input Features** | There is a total of 561 features, which include important statistics of the inertial and acceleration signals (as a function of time and frequency as domains) received by the smartphone. |
| **Output Classes** | WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING |
| **Input/Output Encoding** | |
| Input Encoding | No encoding is required since inputs are already numerical |
| Output Encoding | The six classes are given an ID or a single number ranging from 1 to 6. WALKING is 1, WALKING_UPSTAIRS is 2, WALKING_DOWNSTAIRS is 3, SITTING is 4, STANDING is 5 and LAYING is 6. |