SELF-ORGANIZING MAPS APPLIED TO MANUFACTURING PROCESS CONTROL

Kishan Jainandunsing, PhD
Senzo Labs

Abstract

The Self-Organizing Map or SOM is a powerful machine learning tool for analysis and visualization of high-dimensional sets of unlabeled data. Its power lies in its capability to map correlations between data points in a grid, producing information-rich 2-D or 3-D visualizations. This mapping inherently produces clusters of data points with distinct statistical properties, while at the same time allowing blending from one cluster into another where data points at the edges of neighboring clusters may exhibit a blend of the statistical properties from their own and from neighboring clusters. The SOM can be applied to a wide range of different problems. In this white paper we look at the underlying principles of the SOM as a machine learning tool and illustrate its application in manufacturing process control.

Introduction

The Self-Organizing Map was introduced by the Finnish professor Teuven Kohonen in 1995. It is sometimes also referred to as the Kohonen map or network. Although the SOM is a learning network of nodes, it differs significantly from an artificial neural network (ANN) because the underlying learning mechanism is different than that of an ANN. The SOM uses a competitive learning approach as opposed to the back-propagation gradient descent learning approach of an ANN. Since the SOM acts on unlabeled data, it belongs to the category of unsupervised machine learning algorithms. Furthermore, because the SOM organizes the data in groups with same or similar statistical properties, it is inherently a data clustering tool.

The SOM has found successful application in a wide range of areas, such as process monitoring and control, fault diagnosis, resource allocation and optimization, image analysis, pattern recognition, phoneme recognition, data mining and knowledge discovery.

Intuitive Understanding of the SOM Algorithm

During iterative training the SOM forms a malleable net that folds onto the data set. This net tends to approximate the probability density of the data set. In a high-dimensional data set, each data point is a vector $\boldsymbol{x}$ of dimensionality much larger than 1.

In each training step a data point is randomly pulled from the data set. Its distance (or similarity) is computed to each vector in a set of weight vectors. These weight vectors are associated with the nodes in the Kohonen net and are modeling the probability density of the data set. Initially the modeling is very rough, but it improves with each iteration as the procedure steps through the entire data set $\{\boldsymbol{x}^{(0)}, \dots, \boldsymbol{x}^{(n)}\}$ and updates the set of weight vectors in each step. Figure 1 shows an example of a Kohonen network organized in a 2-D grid of nine nodes and the same network redrawn as an ANN with the weight vectors for each node.

The node with its weight vector nearest in distance to the data vector $\boldsymbol{x}^{(k)}$ becomes the best matching unit, or BMU, of the data vector $\boldsymbol{x}^{(k)}$. The distances are calculated as Euclidean distances, $||\boldsymbol{x}^{(k)} - \boldsymbol{w}_i||$, where $\boldsymbol{w}_i$ is the weight vector associated with node i. Note that a BMU may have more than one data vector mapped to it, but not the other way around.

Next the weight vectors of the entire net are updated, causing the BMU to move even closer to the data vector that it was already closest to. This step can be thought of as a reinforcement step.

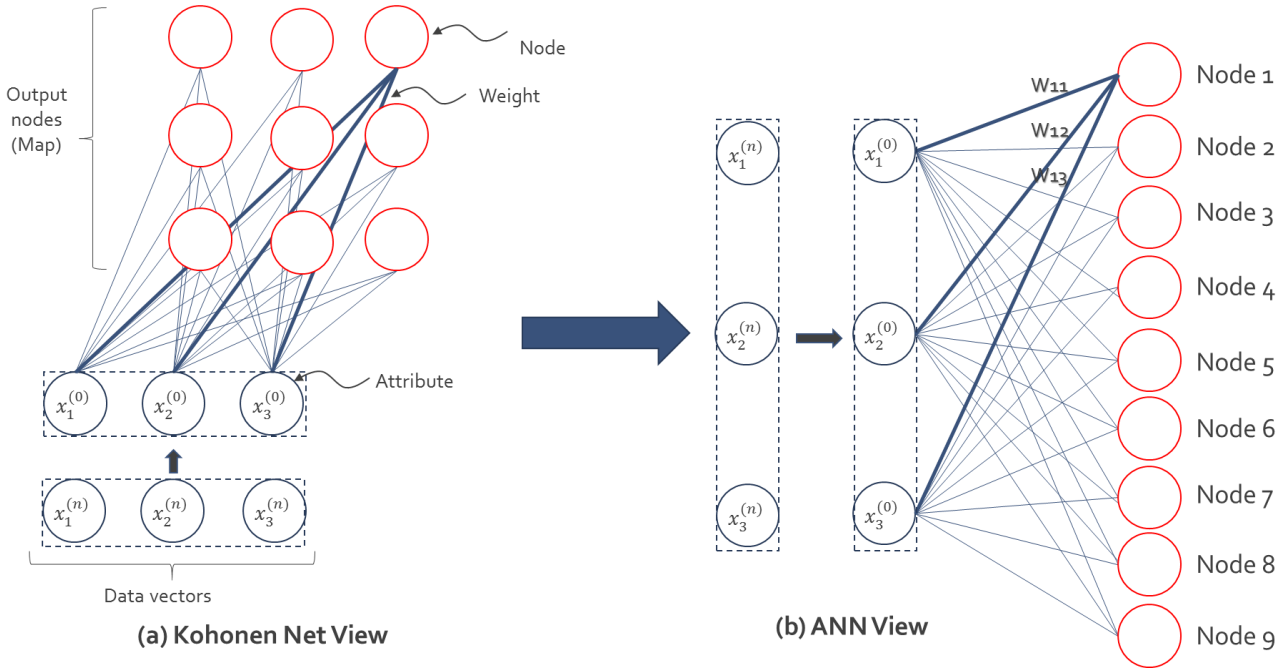

Figure 1: Kohonen net and equivalent ANN view

Pulling at the BMU causes all nodes to move since the nodes in the net are all connected. However, the farther away a node is from the BMU the smaller the displacement of the node is since the update or reinforcement of the weight vectors of nodes is weaker the farther away they are from the BMU. This is illustrated in Figure 2. The orange colored nodes receive strong weight updates, while the purple colored ones receive a weaker weight update. The strength of the weight update is usually chosen to decay exponentially with the distance from the BMU.

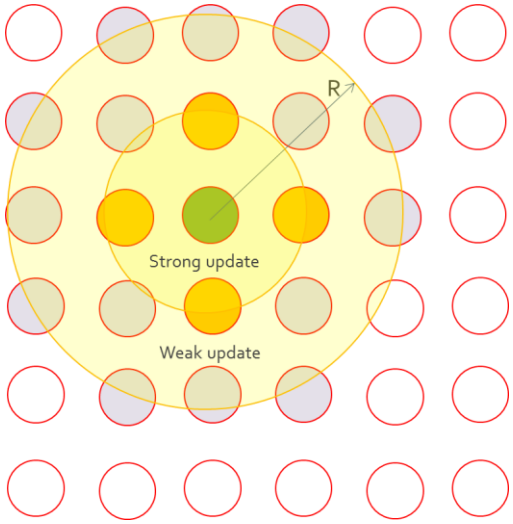

Figure 2: Weight updates for the BMU and nodes in its neighborhood

A single iteration through the entire data set is called an epoch. It may take several epochs to get to a state where the updates of the weight vectors have become insignificant. The procedure is illustrated graphically in Figure 3.

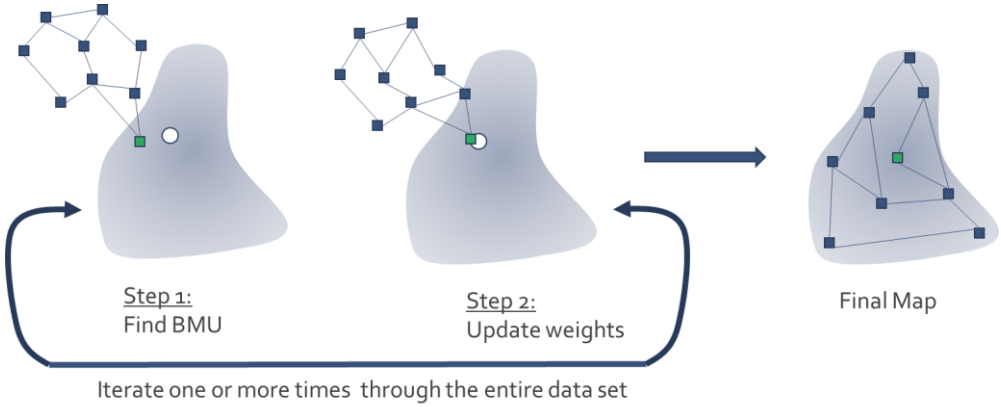

Figure 3: Graphical illustration of the iterative steps of the SOM procedure

By mapping the weight vectors to a color scale the result is a "heat map" which provides a visualization of the probability distribution of the data set. The larger the number of nodes is chosen, the higher the resolution of the probability distribution is. See Figure 4 for an example where countries have been categorized by quality of life, as measured by several contributing factors, such as health, nutrition, education, income, etc.

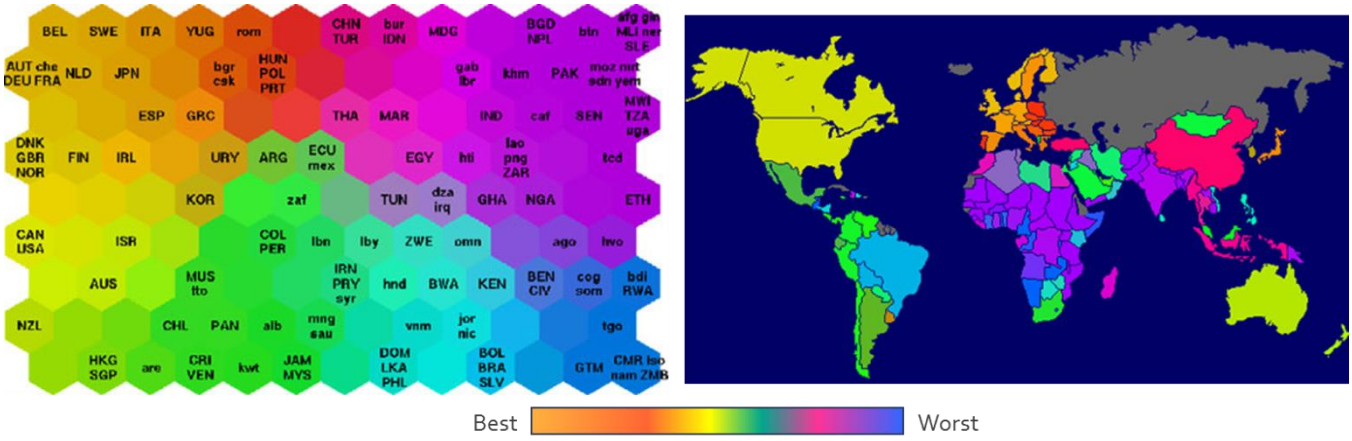

Figure 4: SOM heat map example

The resulting map reveals how the data is separated in clusters based on the underlying probability density function of the data set. The cluster boundaries are not binary, but are blending, since the probability density function is a continuous function.

Training the SOM

In order to train the SOM the data set is split into a training set and a test set. The training set is used to generate the weight vectors and BMUs. Once trained, data vectors from the test set with known properties are used as input to the Kohonen net. The result of each output node is computed as the Euclidean distance between the data vector and the weight vector of the output node, i.e., $||\boldsymbol{x} - \boldsymbol{w_i}||$ as we saw before in determining the BMUs.

The node with the smallest Euclidean distance is chosen as the best representation or prediction $\hat{\boldsymbol{x}}$ of the data vector. The SOM is trained properly if the test set yields predictions within a pre-set error bound, i.e., $\left|\left|\boldsymbol{x}-\hat{\boldsymbol{x}}\right|\right| \leq \varepsilon$.

SOM Industrial Application Categories

Industrial applications of the SOM can be divided in two main categories:

- Process monitoring.
- Process modeling.

In process monitoring we can distinguish two important sub-categories:

- *Process analysis*. In this use case the SOM is trained on a data set which originates from a known good process, since the objective is to understand and gain insight in the dynamics of a normal operating process. The SOM facilitates understanding of how several process variables interact and are correlated with each other through visual inspection of the map.
- *Anomaly/fault detection*. In this use case the SOM is trained on a data set that is representative of normal process operation, since the objective is to detect deviation or trend towards anomalous operation. Once trained, the SOM is fed real-time process data. Any deviation from normal operation triggers an alarm in order to undertake immediate corrective action, while any perceived drift towards the bounds of normal operation triggers scheduling of corrective action. The first case corresponds to the case where $\left|\left|\boldsymbol{x}-\hat{\boldsymbol{x}}\right|\right| > \varepsilon$, whereas the second case corresponds to $\left|\left|\boldsymbol{x}-\hat{\boldsymbol{x}}\right|\right|$ showing a trend line towards ε from which the date and time that $\left|\left|\boldsymbol{x}-\hat{\boldsymbol{x}}\right|\right|$ will cross ε can be extrapolated.

Yet a 3rd sub-category of process monitoring is possible, which is *fault identification*. However, this is in general much harder to do, because it means that we must have a priori knowledge of all possible anomalies/faults and then train the SOM with a data set that is representative of normal operation and faulty operation. Using the SOM for fault identification is therefore only recommended if one has such a priori knowledge.

In process modeling the objective is to be able to do predictions based on the probability density function of the SOM. We can distinguish three important sub-categories in process modeling:

- *Non-linear regression modeling*. The regression of $\boldsymbol{y}$ on $\boldsymbol{x}$ is generally defined as $\hat{\boldsymbol{y}} = E(\boldsymbol{y}|\boldsymbol{x})$, i.e., the expectation of output $\boldsymbol{y}$ given input $\boldsymbol{x}$. It can be shown that the weight vectors of the nodes in the Kohonen net represent local averages of the training data. The regression performed by the SOM on an input $\boldsymbol{x}$ is then equivalent to searching for the BMU closest to the input $\boldsymbol{x}$. I.e.,

  $$k = arg\min_i ||\boldsymbol{x}-\boldsymbol{w_i}||, \text{ and } \hat{\boldsymbol{y}} = \boldsymbol{w_k}.$$

  From the above equation is follows that the BMU for the input $\boldsymbol{x}$ is node k.

  The accuracy of the regression is dependent on the number of nodes in the Kohonen net. A SOM with a large number of nodes provides a dense quantization of the data space, whereas a SOM with a small set of nodes provides a coarse quantization of the data space. Although a dense quantization provides a higher accuracy, if too dense it can exhibit overfitting of the data space by the SOM, meaning that the SOM models the training data set extremely well, but produces large regression errors on data outside of the training data set. In contrast, a coarse quantization, if too coarse can exhibit a high bias, meaning that the SOM not only poorly models the data space, but also produces large prediction errors on the training set as well as on data outside of the training set.

- *Local linear modeling*. By dividing the input data set into disjoint sets, the SOM can be made to model the data set piece wise linearly, increasing its accuracy in making predictions. Indeed, consider the

Voronoi set V_i of node i, given by the set of data vectors $\{\boldsymbol{x_1}, \dots, \boldsymbol{x_n}\}$ for which the weight vector $\boldsymbol{w_i}$ of node i is closest. I.e.,

$$V_i = \{\boldsymbol{x}\big|\big|\boldsymbol{m_i} - \boldsymbol{x}\big|\big| < \Big|\big|\boldsymbol{m_j} - \boldsymbol{x}\big|\Big|, \forall j \neq i\}$$

Using the above formula, a series of input data vectors is automatically divided into disjoint clusters, where each cluster contains data vectors for which the corresponding weight vector is the best prediction for the data vectors in the cluster.

- *Sensitivity analysis*. The non-linear regression model can be exploited to investigate how small changes made in process parameters influences the process. Generally speaking the physical limitations of the process limit the range of values defined by the regression model. If the BMU is not influenced by making a small change to one or more process parameters, then the process model is robust. Tracking the change of BMU due to the change in parameters reveals the non-linear relationship between the parameters.

Conclusions

The Self-Organizing-Map or Kohonen network is a useful unsupervised clustering tool for the analysis and monitoring of industrial processes. Its ability to model a hyper-dimensional probability distribution of the stochastic properties of process parameters and the non-linear dependencies between them, combined with the ability to visualize such complexity in a two or three dimensional space makes the SOM a powerful machine learning tool in the arsenal of Industry 4.0 big data analytics tools. The fact that it uses an unsupervised learning algorithm makes it so much more useful in detecting process anomalies since it is not necessary to possess a priori knowledge of anomalous process parameter value ranges, but instead we can focus only on the normal operational range of the process parameters.

The SOM doesn't require the construction of an explicit analytical process model to analyze performance and sensitivity. Instead, the SOM relies solely on big data analytics to extract a probability distribution function that exposes the complex relationships between large numbers of variables that may include manufacturing process variables beyond engineering, such as those that are financial (investments), environmental, socio-economical, etc. in nature. Process control operators can use the SOM as a visual aid in learning how to adjust process variables to steer the process towards the desired operation region and keep it within that region. Likewise others in the corporation outside of engineering, such as finance and operations, can use the SOM in the same way as a visual aid to study how variables beyond process engineering are correlated with each other and what the effect that changes in these variables have on the performance of the process vs. the desired region of operation of the process.