An Approach to Mixed Dataset Clustering and Validation with ART-2 Artificial Neural Network Model

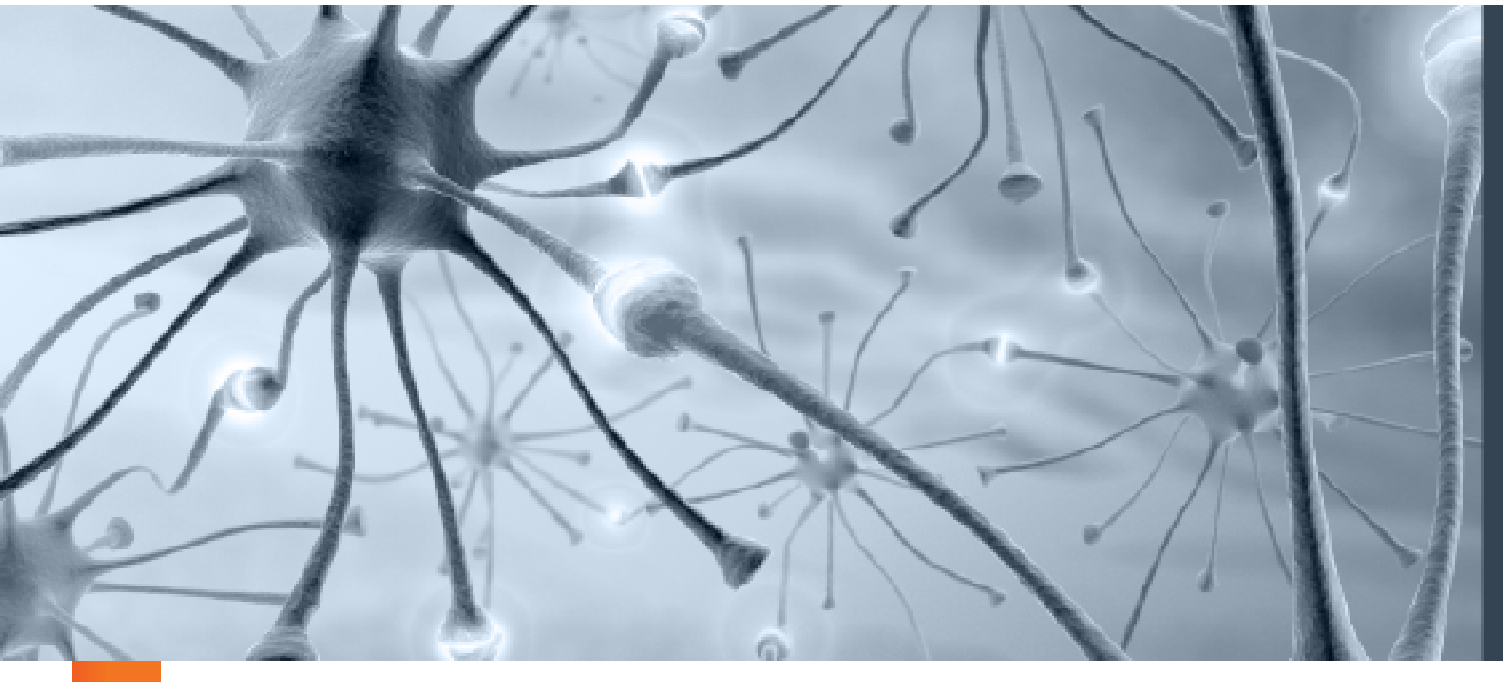
happiest minds
The Mindful IT Company
Born Digital . Born Agile

Abstract

Among many data clustering approaches available today, mixed data set of numeric and category data poses a significant challenge due to difficulty of an appropriate choice and employment of distance/similarity functions for clustering and its verification. Unsupervised learning models for artificial neural network offers an alternate means for data clustering and analysis. The objective of this study is to highlight an approach and its associated considerations for mixed data set clustering with Adaptive Resonance Theory 2 (ART-2) artificial neural network model and subsequent validation of the clusters with dimensionality reduction using Autoencoder neural network model.

Copyright Information

Contents



Introduction

Data mining application and processes utilize cluster analysis very widely to partition large data sets and to discern useful information, in terms of patterns which subsequently find use in many areas of analysis and decision making. Cluster analysis or clustering is used in many domains and industries and currently many different techniques are available to perform clustering of data.

Conventional clustering methods, algorithms and measures available today are more focused on clustering of data based on one type of predominant data attribute, which typically is either numerical or categorical. However, in real life scenarios, there exists a vast number of data sets which are essentially mixed data sets of numerical and categorical data. The primary challenge to clustering of mixed data sets is the presence of many data attributes of numeric and categorical data types and the need to consider these attributes together to arrive at a meaningful separation of data.

Another aspect of many current clustering methods is the requirement for a distance/similarity measure that is essentially used in the clustering process. In a mixed data sets, while the numeric attributes exhibit continuous characteristics, the category attributes display a discontinuous and unordered behaviour. To handle this challenge, typically, for numeric portion of the data set a distance measure is used and for the categorical portion a similarity measure. While this is a viable approach, this has the potential to distort the date element evaluation process.

Another approach to address mixed data set is to transform the entire dataset into a completely numeric data set with some standard/custom transformation method(s) and use the final transformed dataset as the target dataset. This also has the potential to distort the data characteristics.

While there are quite a few techniques to cluster mixed data sets with use of standard clustering algorithms and distance/similarity measures e.g. with employment of Gower Coefficient [1] alongside k-mean based algorithms, in this paper an approach is discussed towards clustering of mixed dataset based on unsupervised learning of artificial neural networks.

In this paper, unsupervised learning of an ART-2 (Adaptive Resonance Theory 2) [2] network has been employed to classify a mixed dataset of a selected NSE Nifty stock data at different fidelity and post classification and cluster identification the dimensionality of the mixed dataset has been reduced with an Autoencoder [3] network for visual representation and intuitive validation of clusters. The data and the representation are presented and described and from the visual representation of the clusters inferences have been derived and presented.

Dataset Description & Clustering Challenges

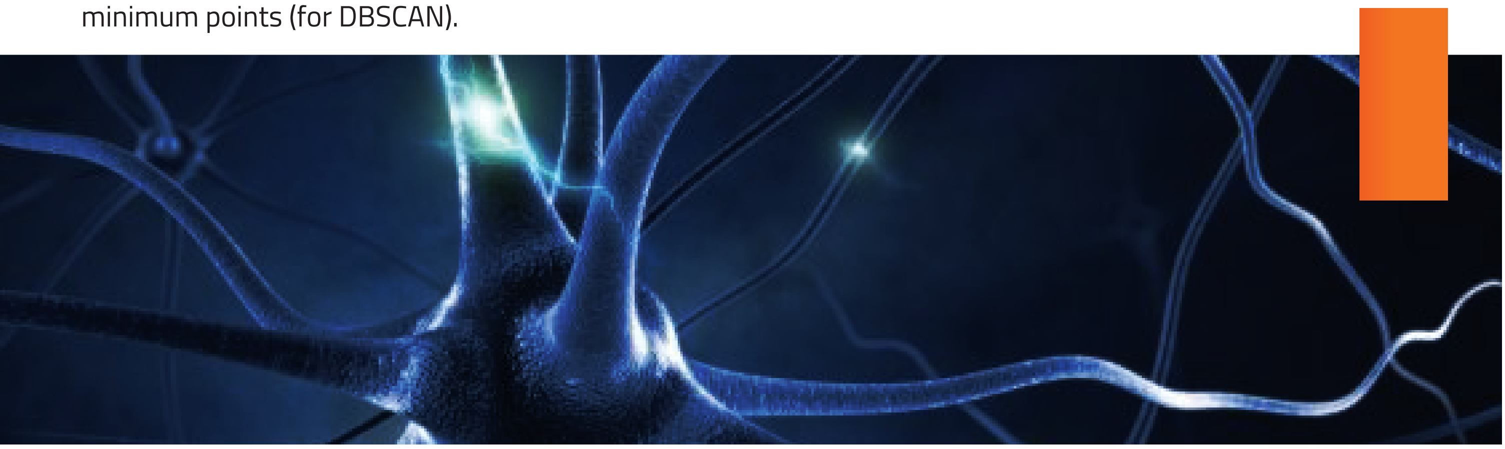
In technical analysis of stock/securities signals are often derived from comparing one trade day's price points (open, high, low and close) with the preceding trade day's price points. Signals for trading (buy, sell or hold) are derived with these signals as inputs into the security analysis process among other things. Technical analysts and traders for example place importance in metrics like higher highs, higher lows, lower highs and lower lows [4]. The dataset under consideration is derived from daily open, high, low and close stock price of a selected stock from NSE Nifty 50 index. Each data vector in the dataset consists of nineteen (19) components derived from comparison of two consecutive trade day's open, high, low and close prices.

The absolute values of the comparisons constitute the numerical data attributes of a data vector and their corresponding signs or directions (+ve or -ve/up or down) constitute the categorical part of a data vector. The absolute values are taken as is where as the directional components are expressed as either '0' (+ve/up) or '1' (-ve/down)1. The vector attributes are detailed below in Table 1 - Dataset DescriptionIt is evident from the dataset that each data point vector is a mixed attribute set of numeric and categorical data, to cluster this dataset successfully, the current popular clustering algorithms e.g. k-mean (centroid based) and DBSCAN (density based)2 would require either an appropriate distance function (for k-mean based algorithms) or density definition e.g. ε radius and minimum points (for DBSCAN).

A mixed dataset is a collection of data points where each data point have at least one or more attributes belonging to different statistical data types. Though data types in any combination in any data point vector may constitute a mixed dataset, however in practice most mixed datasets constitute of numerical and category data. The dataset(s) used in this paper is described below.

Challenges of clustering a mixed dataset are many and the below dataset being a mixed dataset, it is difficult to select and determine an appropriate distance function to use directly with the dataset (though there are some distance functions available for mixed dataset [1]) as distance functions typically tend to be more appropriate for specific datasets and generalization to other datasets becomes more difficult. In addition, centroid based clustering methods give better results if the initial number of cluster centers are appropriately chosen and the estimate of initial number of clusters are approximately known [11]. Density based clustering methods on the other hand have some notion of density e.g. in DBSCAN in takes the shape of ε parameter, the radius from each core point and minPts, the minimum number of points required in the neighborhood. This, for some datasets, which have large density variation become difficult to determine and as the number of dimensions increase, the difficulty becomes more pronounced.

In addition, the dataset being drawn from the stock market will tend to exhibit some random walk as is inherent in all market driven data, which makes the attempt to cluster the dataset little more difficult. The below dataset then provides for a good case of a mixed dataset that has an element randomness and noise associated with it and would provide a good proving ground for ART-2 based clustering attempt.



1 Though the direction/+ve or -ve sign derived from the comparison is expressed as an integer, it is nonetheless treated as a distinct category attribute and is not considered as numerical attribute. The integers '0' and '1' merely serve as labels which can be processed as part of a pattern by artificial neural networks

2 k-mean based algorithms and DBSCAN are mentioned here since they are very popular, powerful, well understood and widely used algorithms

Table 1 - Dataset Description

Attribute Name	Vector Position	Derivation	Data Type	Description
RANGE_COMPARE_PREVIOUS	0	ABS(((HP-LP)-(HC-LC))/ (HP-LP))	Numerical	Absolute value of range (high price-low price) comparison of a trade day to its preceding trade day
RANGE_COMPARE_PREVIOUS_SIGN	1	The corresponding sign/direction of the above	Categorical	The sign/direction of the comparison expressed in either '0' (+ve) or '1' (-ve) label
OPEN_COMPARE_PREVIOUS	2	ABS((OP-OC)/OP)	Numerical	Absolute value of open price comparison of a trade day to its preceding trade day
OPEN_COMPARE_PREVIOUS_SIGN	3	The corresponding sign/direction of the above	Categorical	The sign/direction of the comparison expressed in either '0' (+ve) or '1' (-ve) label
HIGH_COMPARE_PREVIOUS	4	ABS((HP-HC)/HP)	Numerical	Absolute value of range (high price-low price) comparison of a trade day to its preceding trade day
HIGH_COMPARE_PREVIOUS_SIGN	5	The corresponding sign/direction of the above	Categorical	The sign/direction of the comparison expressed in either '0' (+ve) or '1' (-ve) label
LOW_COMPARE_PREVIOUS	6	ABS((LP-LC)/LP)	Numerical	Absolute value of range (high price-low price) comparison of a trade day to its preceding trade day
LOW_COMPARE_PREVIOUS_SIGN	7	The corresponding sign/direction of the above	Categorical	The sign/direction of the comparison expressed in either '0' (+ve) or '1' (-ve) label
CLOSE_COMPARE_PREVIOUS	8	ABS((CP-CC)/CP)	Numerical	Absolute value of range (high price-low price) comparison of a trade day to its preceding trade day
CLOSE_COMPARE_PREVIOUS_SIGN	9	The corresponding sign/direction of the above	Categorical	The sign/direction of the comparison expressed in either '0' (+ve) or '1' (-ve) label
CLOSE_TO_OPEN_PREVIOUS	10	ABS((CC-OP)/OP)	Numerical	Absolute value of range (high price-low price) comparison of a trade day to its preceding trade day
CLOSE_TO_OPEN_PREVIOUS_SIGN	11	The corresponding sign/direction of the above	Categorical	The sign/direction of the comparison expressed in either '0' (+ve) or '1' (-ve) label
CLOSE_TO_OPEN_CURRENT	12	ABS((CC-OC)/CC)	Numerical	Comparison of difference between close price and open price w.r.t. close price of a trade day
CLOSE_TO_OPEN_CURRENT_SIGN	13	The corresponding sign/direction of the above	Categorical	The sign/direction of the comparison expressed in either '0' (+ve) or '1' (-ve) label
HIGH_TO_CLOSE_CURRENT	14	ABS((HC-CC)/(HC-LC))	Numerical	Comparison of difference between high price and close price w.r.t. range (high-low) of a trade day
HIGH_TO_CLOSE_CURRENT_SIGN	15	The corresponding sign/direction of the above	Categorical	The sign/direction of the comparison expressed in either '0' (+ve) or '1' (-ve) label
CLOSE_TO_LOW_CURRENT	16	ABS((CC-LC)/(HC-LC))	Numerical	Comparison of difference between close price and low price w.r.t. range (high-low) of a trade day
CLOSE_TO_LOW_CURRENT_SIGN	17	The corresponding sign/direction of the above	Categorical	The sign/direction of the comparison expressed in either '0' (+ve) or '1' (-ve) label
BLACK	18	The colour of the candlestick	Categorical	The colour of the candle on a candlestick graph for a trade day expressed as '0' - if the candle is not black or '1' - if the candle is black

Legend: OC - Current day's open price | OP - Previous day's open price | HC - Current day's high price | HP - Previous day's high price | LC - Current day's low price LP - Previous day's low price | CC - Current day's close price | CP - Previous day's close price

An example is provided below with data of two consecutive trade day for Cipla Ltd. (ISIN - INE059A01026)	Date - 09/01/2012	Open - 335.45	High - 346.5	Low - 333.25	Close - 344.65
	Date - 10/01/2012	Open - 345	High - 349.2	Low - 343.5	Close - 345.8

RANGE_COMPARE_PREVIOUS - 0.569811321 RANGE_COMPARE_PREVIOUS_SIGN - 1	HIGH_COMPARE_PREVIOUS - 0.007792208 HIGH_COMPARE_PREVIOUS_SIGN - 0	CLOSE_COMPARE_PREVIOUS - 0.003336718 CLOSE_COMPARE_PREVIOUS_SIGN - 0
OPEN_COMPARE_PREVIOUS - 0.02846922 OPEN_COMPARE_PREVIOUS_SIGN - 0	LOW_COMPARE_PREVIOUS - 0.030757689 LOW_COMPARE_PREVIOUS_SIGN - 0	CLOSE_TO_OPEN_PREVIOUS - 0.030854077 CLOSE_TO_OPEN_PREVIOUS_SIGN - 0
CLOSE_TO_OPEN_CURRENT - 0.002313476 CLOSE_TO_OPEN_CURRENT_SIGN - 0	HIGH_TO_CLOSE_CURRENT - 0.596491228 HIGH_TO_CLOSE_CURRENT_SIGN - 0	CLOSE_TO_LOW_CURRENT - 0.403508772 CLOSE_TO_LOW_CURRENT_SIGN - 0 BLACK - 0

In the above dataset description, the category variables are expressed as simple sign/direction of up/down; however, this has been done merely to simplify the dataset to illustrate the clustering process, in real life many more categories could be defined on the numeric variables viz. up, down, no change, sideways, up with strong bias, down with strong bias, sideways with weak bias etc.

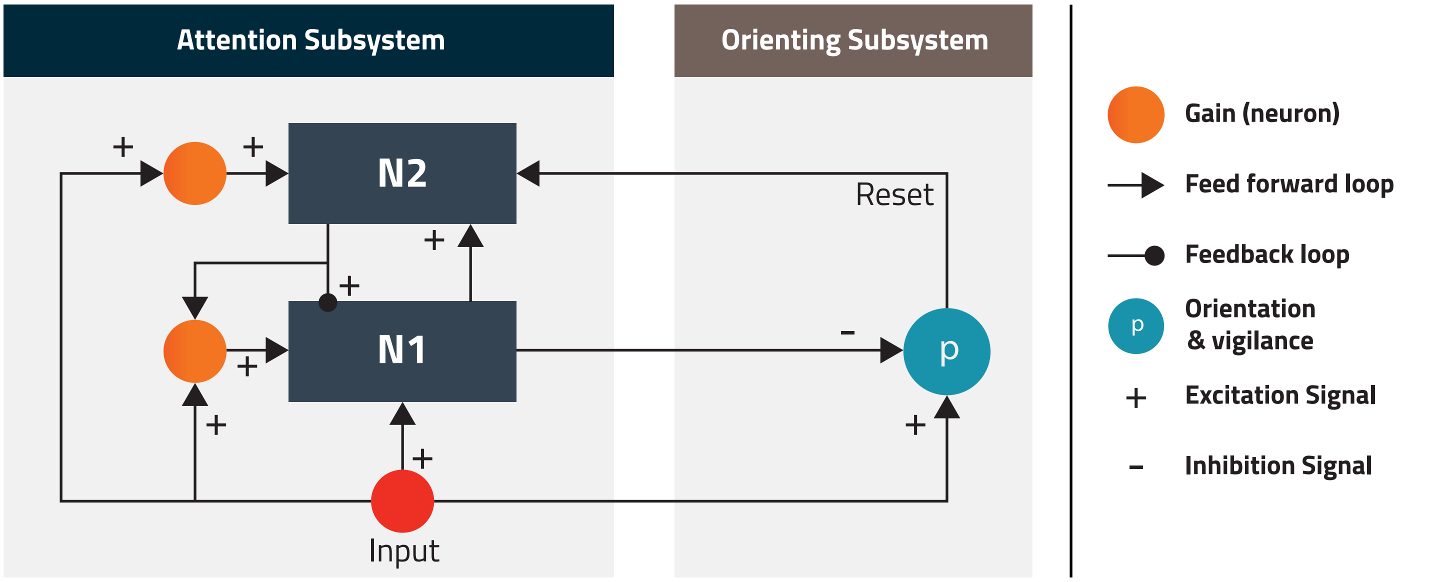
ART/ART-2 Neural Network Model

ART (Adaptive Resonance Theory)/ART-2 neural network models work on the principal that identification of objects/vectors into classes occur from two sources of information viz. top down retained knowledge (long term memory)and expectation and bottom up inputs and information (short term memory). The comparison of the long term and short term memory brings forward a classification or categorization of data vectors. As long as this comparison does not exceed a defined threshold (termed as Vigilance Parameter) the input is considered member of a defined class in long term memory. However, if the comparison exceeds the Vigilance Parameter, the object/vector is considered to be of a class that has not been encountered previously and the vector properties are learnt into the long term memory. By having the concept of a long and short term memory, ART family of networks can retain knowledge over time and increase the knowledge base. ART family networks offer a framework to retain old knowledge while gaining new knowledge i.e. it addresses the plasticity/stability problem faced in learning systems [6].

The primary difference between the ART-1 model and ART-2 model is that ART-1 model could handle only binary inputs, whereas ART-2 has been extended out of ART-1 to support continuous inputs.

Basic Structure of ART Networks

Figure 1 - Basic ART Network Structure [5]



The basic structure of an ART family (ATR-1/ART-2/ART-2A/ART-3) of network is provided above (Figure 1 - Basic ART Network Structure), it consists of two primary subsystem termed as "Attention Subsystem" and "Orientation Subsystem". The attention subsystem represent the Long Term Memory (LTM) and Short Term Memory (STM) of the system, which is essentially used to classify inputs. The purpose of orientation subsystem is to stabilize the processing in STM and for learning in LTM. [5]

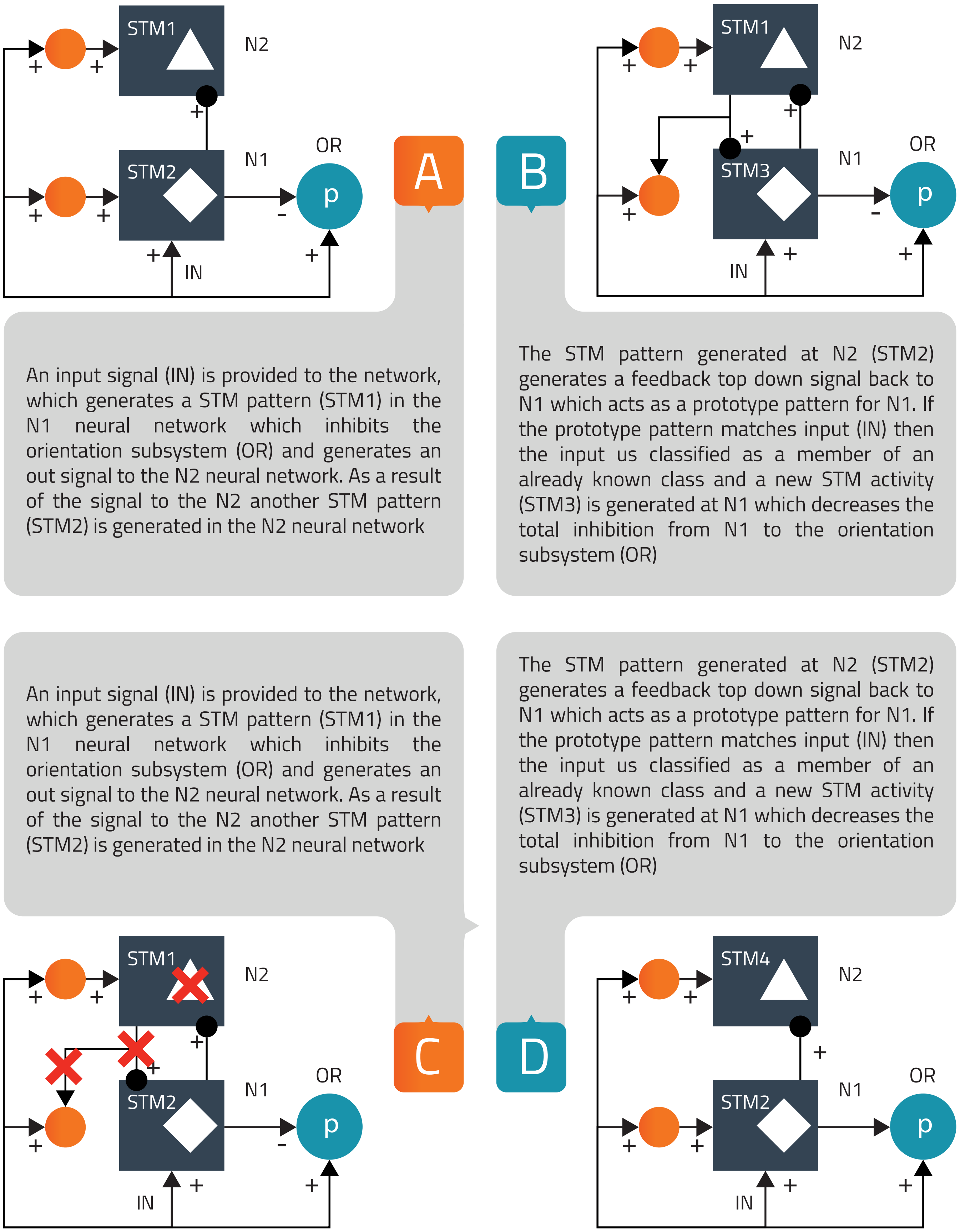
The attention subsystem consists of two independent neural networks (N1 and N2), where the networks themselves may contain multiple layers of neurons. N1 and N2 are connected to each other with feed forward and feedback connections containing weights. These weights on the connections between N1 and N2 constitute the LTM of the ART network. The short term memory on the other hand are the weights of the individual neurons and the pattern of activity that is generated in the N1 and N2 neural networks as input is processed. [5]

The N1 network receives signal from three sources, the input vector (the bottom up signal), N2 (the top down signal) and the gain control signal. At any point two out of the three inputs must be active for N1 to generate activity. [5]

ART Network Vector Classification process

In an ART network the classification process for a data vector happens as follows:

Figure 2 - ART Network Classification Process



An input signal (IN) is provided to the network, which generates a STM pattern (STM1) in the N1 neural network which inhibits the orientation subsystem (OR) and generates an out signal to the N2 neural network. As a result of the signal to the N2 another STM pattern (STM2) is generated in the N2 neural network

The STM pattern generated at N2 (STM2) generates a feedback top down signal back to N1 which acts as a prototype pattern for N1. If the prototype pattern matches input (IN) then the input us classified as a member of an already known class and a new STM activity (STM3) is generated at N1 which decreases the total inhibition from N1 to the orientation subsystem (OR)

An input signal (IN) is provided to the network, which generates a STM pattern (STM1) in the N1 neural network which inhibits the orientation subsystem (OR) and generates an out signal to the N2 neural network. As a result of the signal to the N2 another STM pattern (STM2) is generated in the N2 neural network

The STM pattern generated at N2 (STM2) generates a feedback top down signal back to N1 which acts as a prototype pattern for N1. If the prototype pattern matches input (IN) then the input us classified as a member of an already known class and a new STM activity (STM3) is generated at N1 which decreases the total inhibition from N1 to the orientation subsystem (OR)

(Adapted from Carpenter et al., 1991 [2] and David Weenink, 1997 [5], refer references section)

Data Clustering Process with ART-2 Network

For the purpose of classification of the dataset mentioned above, trading prices of Cipla Ltd. has been considered from 26th October 2010 to 11th December 2015, in this interval Cipla stock has not undergone any split or bonus (last bonus on 11th February 2006 and last split on 23rd March 2004), as any split or bonus in the interim period would have required an extra step of adjusting the trade prices for split and bonus.

The Open, High, Low and Close prices for the stock was collected from NSE [http://nseindia.com/products/content/equities/equities/eq_security.htm] . The data for the date range has been split into two portions one of 300 data points, 'Dataset A' (26/10/2010 to 07/01/2012) and another of 974 data points, 'Dataset B' (09/01/2012 to 11/12/2015)

Dataset A has been used to identify clusters and to train the Autoencoder for dimensionality reduction. For an ART-2 network there is no difference between a training and a classification set as the network is adaptive, it learns while classifying data points and discovers new clusters/classes as it classifies more and more data points. For an Autoencoder, however, a dataset is needed for training and Dataset A was also used for this purpose.

For clustering, the ART-2 network implementation in Java of PWR-APW open source library has been used [7]. The dataset A was classified at three (3) fidelity levels with different Vigilance Parameter (ρ) values set on the ART-2 implementation. Each data element was individually classified with an output result of an integer, designating the cluster the data point has been classified to. This cluster identifier was associated with each data point.

Post classification of Dataset A with ART-2 network, an implementation of Autoencoder [3] with Encog Machine Learning Framework [8] has been used on Dataset A to reduce the dimension of the 19 element data point vectors to 2 element representation. Dataset A served as the training dataset for the Autoencoder.

After Dataset A was classified, the same ART-2 classifier as used to classify Dataset B and the cluster identifiers thus obtained were associated with the corresponding elements of Dataset B. Further to this the Autoencoder trained above was used to reduce the dimensionality of Dataset B and the dataset was plotted with the reduced dimension.

Post the processing detailed above, the plots for clustered Dataset A data points, in 3 fidelity levels, was plotted for an intuitive understanding of the clusters and the impact of Vigilance Parameter on the clustering process. The Dataset B clustered data points were plotted to make an intuitive validation of the clusters by looking at the cluster stability across the two datasets; the above results are presented, observation made and inferences drawn in Results, Observations & Inferences section.

Autoencoder Neural Network Model

Autoencoders are a family of neural network models/architecture which is focused towards transformation of a specific representation of a dataset (e.g. raw high dimensional data) into another representation (e.g. low dimensional representation). The input to an Autoencoder is transformed to an intermediate representation, termed as a code and this code in turn is decoded and provided back as the output. The transformation process of an Autoencoder is not linear but non-linear and the transformation of the intermediate code to its final output representation with the decoder is more of a predictive behaviour than a simple linear transformation.

The primary motivation behind Autoencoders is to produce a low dimensional representation of a high dimensional data space. There are many methods available for reduction of dimensionality of data and one popular method is Principal Component Analysis (PCA) [13]. PCA attempts to find the vector components that account for most variability in a dataset and thus provide a reduced set of vector components. However, PCA is more suited for numerical datasets than categorical datasets or mixed dataset. Another recent and emerging method towards determining significant variation providing vector component is Factor Analysis of Mixed data (FAMD) [14], however, FDMA is in active research phase of development.

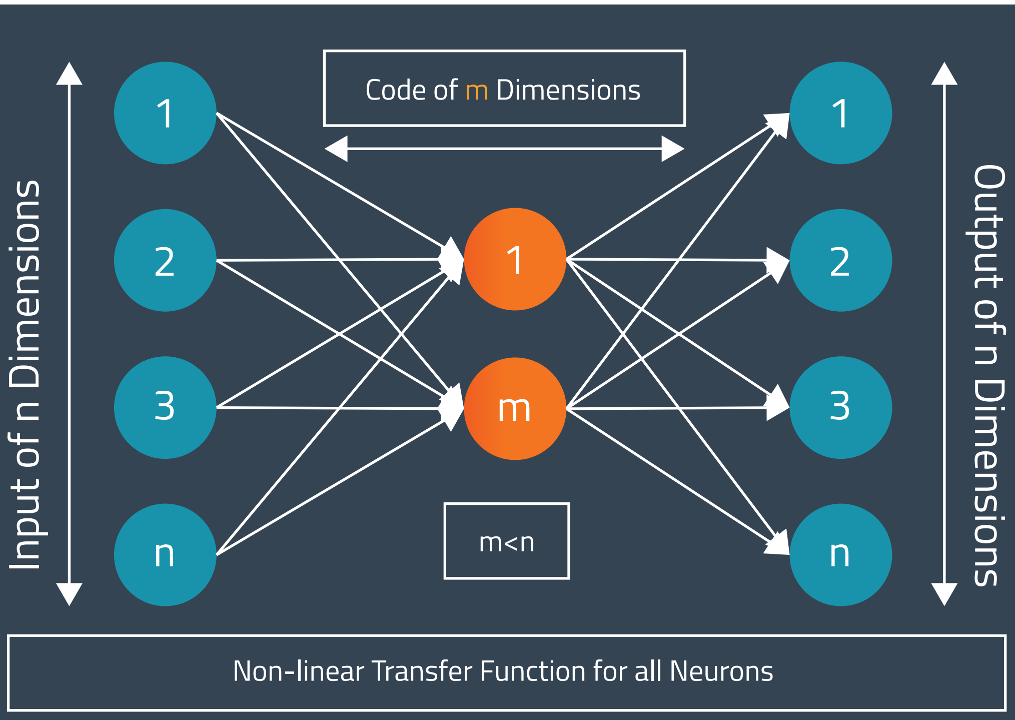
The current dimensionality methods, however, focus on finding significant components of the data vector that provide for maximum variability in a dataset and partially discount the other dimensions, which may leads to a higher dimensional dataset for visualization (e.g. reduction of dimension from 19 to 10) and loss of certain amount of information. Autoencoders, however, try to encode and decode the entire dataset at a particular intermediate dimension (e.g. reducing 19 dimensions into 2 dimensions), which also has a drawback of introducing some error into the encoding/decoding process. However, the ability of Autoencoders to reduce higher dimensional datasets to lower dimensional datasets is quite convenient for visualization and other analysis.

Another aspect of Autoencoders is the ability to map and represent non-linear relationships in data. On the other hand, many of the current methods, PCA for example focus more on linear relationships. Autoencoders by leveraging non-linear transfer functions e.g. Sigmoid or other functions can capture non-linear relationships and reduce the dataset in a non-linear fashion. Autoencoders with linear transfer functions replicate the behaviour of PCA.

Figure 3 - Basic Autoencoder Structure, provides a simple view of an Autoencoder, which is a three layer neural network where the input and output layers contain equal number of neurons representing each dimension of a data vector (n). The middle (hidden) layer contains the number of neurons to which the data vector needs to be reduced (m), where m < n. The input to the input layer is encoded as activations of the middle layer and the activations are in turn decoded to reproduce the input vector.

The reduced dimension dataset obtained from an Autoencoder is considered to be a lossy compression of the input dataset and the objective of an Autoencoder is to fit the training data appropriately and therefore if the training sample does not adequately represent the larger dataset, then encoding error may be high.

Figure 3
Basic Autoencoder Structure



Autoencoder Dimensionality Reduction Process

As indicated in Data Clustering Process with ART-2 Network section, as part of the larger data clustering process, dimensionality of the dataset was reduced with an Autoencoder after clustering and the results are plotted for a more intuitive identification and validation of the clusters. This section details the Autoencoder implementation used for the dimensionality reduction process.

The Autoencoder is constructed with Encog Machine Learning Framework [8] as a three layer perceptron. The input layer neuron count reflects the dimension of the data vectors and fixed at 19 (refer Table 1 - Dataset Description). Since plotting of the clustered data required two dimensions, the middle layer neuron count (code generation layer) was fixed at 2 and the output layer count reflected the input neuron count to reconstruct the data vector and was fixed at 19.

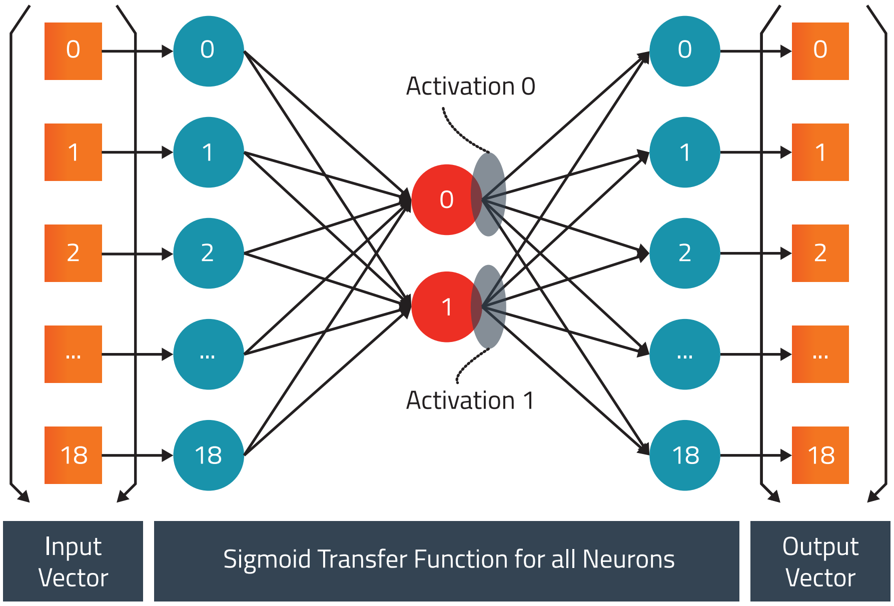
The below table provides the details of the Autoencoder network

Network Property	Property Value
Network Architecture	Feedforward Network
Learning Type	Supervised Learning
Layer Count	3
Neuron Connection	All Connected
Hidden Layer Count	1
Neuron Activation Function	Sigmoid
Presence of Bias Neuron	Layer 1, Layer 2 and Layer 3
Training Algorithm	Resilient Backpropagation (Rprop)
Weight Initialization	Random Initialization

The Autoencoder was trained with 'Dataset A' (refer Data Clustering Process with ART-2 Network) at error level of '5%'. During the training process it was observed that at any error level below '5%' the Resilient Backpropagation did not converge. During the training process, it was also observed that without the bias neurons, the learning algorithm convergence did not occur.

The training dataset contained 300 data points drawn from approximately five quarters of trading data for Cipla Ltd. and represented the population adequately. As Autoencoder code output, the activations of the neurons of the middle layer (Activation 0 and Activation 1) was taken (refer Figure 4 - Autoencoder).

Figure 4
Autoencoder



Results, Observations & Inferences

The data vectors from Dataset 'A' and Dataset 'B' was plotted after the clustering and dimensionality reduction operations for an intuitive understanding of the clusters and cluster stability. It was found that ART-2 network employed had clustered the data provided into distinguishable clusters of different shapes and characteristics. It was observed that the Vigilance Parameter (ρ) had a significant impact on the clustering operation; higher parameter values produced more distinct clusters, while lower values tended to produce more unresolved clusters, additionally; at higher parameter values more clusters were identified.

Figure 5 - Vigilance Parameter (ρ) = 0.9 (Dataset 'A') was obtained after plotting the Dataset 'A', which was clustered with Vigilance Parameter value set to 0.9 on the ART-2 network. From the plot, it is apparent that the network is able to cluster the data in different clusters, the cluster count being 5. Cluster 0, stands out quite distinctly, where as other clusters seems to have some resolution issues, especially Cluster 1, this is due to the fidelity of the ART-2 network.

The fidelity was adjusted by increasing the Vigilance Parameter (ρ), Figure 6 - Vigilance Parameter (ρ) = 0.93 (Dataset 'A'), was obtained by increasing the parameter value to 0.93. In this plot, clusters are more distinct, with Cluster 0 of earlier plot being fragmented to Cluster 9, Cluster 4 and Cluster 0 in this plot; the fidelity of Cluster 2 and Cluster 3 has also increased. Cluster 1 is seen to have fragmented and its sparseness reduced with emergence of Cluster 7. The cluster recognition in this plot is seen to have improved over Figure 5 - Vigilance Parameter (ρ) = 0.9 (Dataset 'A') with recognition of 10 clusters, however, it was observed that the clustering process could be still improved with higher Vigilance Parameter value.

Figure 7 - Vigilance Parameter (ρ) = 0.945 (Dataset 'A'), was obtained with Vigilance Parameter (ρ) increased to 0.945. At this fidelity of the ART-2 network, it was observed that the clusters became more distinct with emergence of total 13 clusters 3. The most distinct feature of this plot is fragmentation of Cluster 1 of earlier plots to more distinct clusters viz. Cluster 2 and Cluster 1. It will also be observed that a new cluster, Cluster 12 has emerged from the edges of earlier Cluster 1 and that the other clusters identified in Figure 6 - Vigilance Parameter (ρ) = 0.93 (Dataset 'A') have stabilized.

The below plots adequately portray the ability of ART-2 network to cluster the dataset into distinct clusters, due to the nature of the data, some randomness and noise is expected, however, overall the process has provided adequate clustering and it could be inferred that ART-2 networks could serve as an alternate means for clustering of mixed dataset alongside the more popular methods.

Figure 8 – Clustering Plot for Dataset B, is obtained by clustering Dataset B (refer Data Clustering Process with ART-2 Network section) and it is readily observed that the clusters identified with Dataset A has emerged in Dataset B as well, from which it can be inferred that the cluster identification is stable and the data does display a cluster tendency around the clusters identified.[4]

Another point of observation is the performance of Autoencoder on Dataset B, the Autoencoder was trained on Dataset A with adequate representation of data5 and it reproduced the reduced dimensional representation for Dataset B, the emergence of the same clusters and the stability of the clusters point to a stable Autoencoder. From this it can be inferred that Autoencoders can provide an alternate means for data dimension reduction alongside the current popular methods.

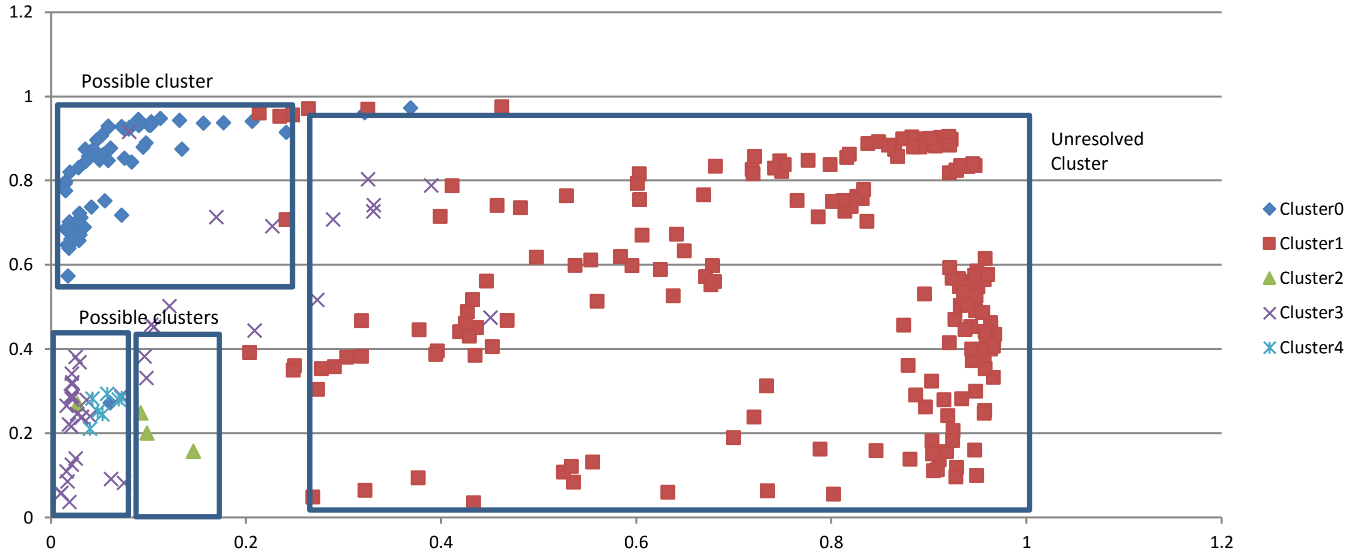
It is also observed that the resolution of the clusters increased with increase of Vigilance Parameter (ρ) alone and it can be inferred that Vigilance Parameter plays an important role in ART-2 network's ability to cluster data and this parameter is a sufficient point of adjustment for arriving at adequate cluster; it was also observed that in the above method, there is no requirement for any distance function or density based parameter to be provided.

[3] In this plot Cluster 3 and Cluster 6 are not plotted as the data points were too low, with Cluster 3 data count at 2 data points and Cluster 6 data count at 4 data points which corresponds to 1.33% of the dataset clustered

[4] The application of the clustering process on Dataset B produced 17 clusters vis-à-vis Dataset A, however, the data point count of the rest 4 clusters was very small, and these clusters could be result of noise and randomness in the data.

[5] 5 quarters of trading data of 300 data points, while the features to be recognized is 19. In general a training set of (feature set x 10) is considered a good training set, here 300 points have been used for training

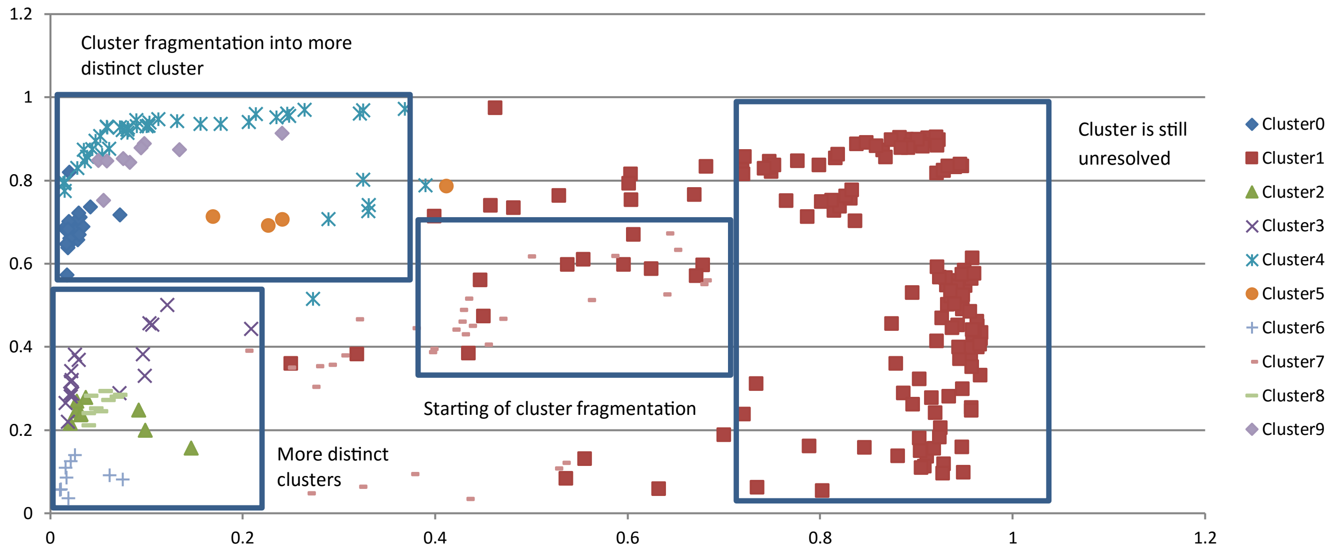
Figure 5 - Vigilance Parameter (ρ) = 0.9 (Dataset ‘A’)



Cluster membership count

Cluster 0 count = 65 | Cluster 1 count = 182 | Cluster 2 count = 4 | Cluster 3 count = 42 | Cluster 4 count = 7

Figure 6 - Vigilance Parameter (ρ) = 0.93 (Dataset ‘A’)

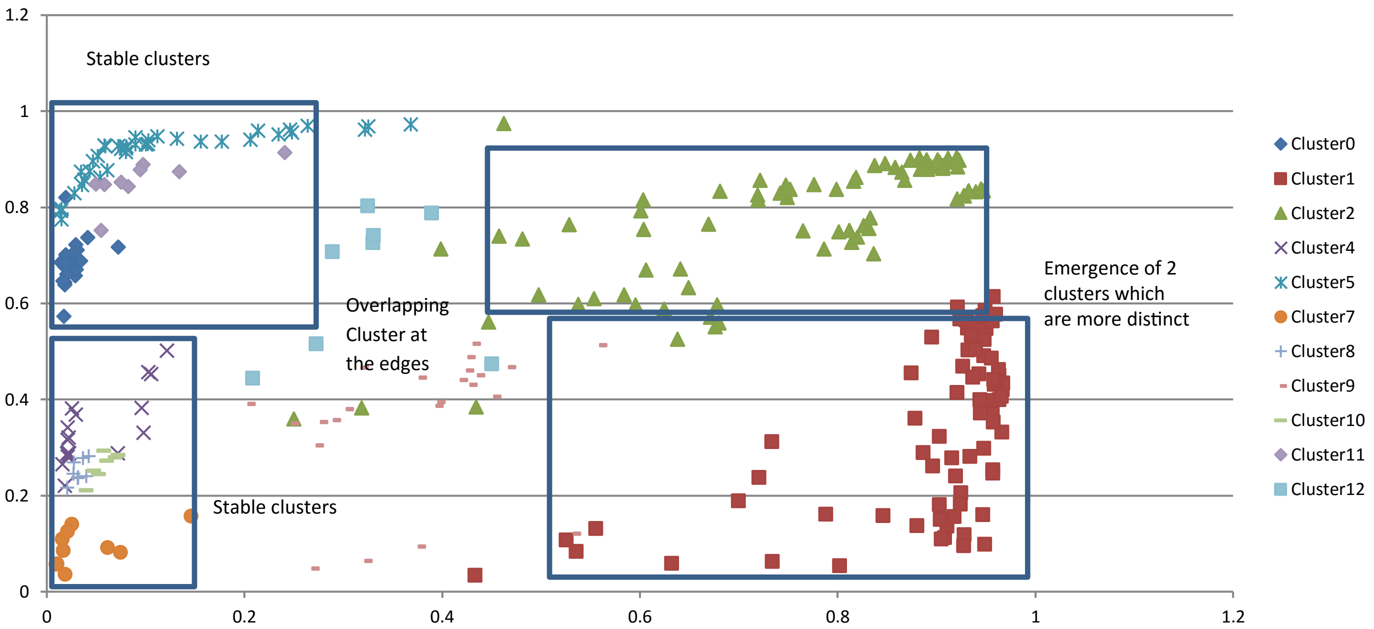


Cluster membership count

Cluster 0 count = 23 | Cluster 1 count = 144 | Cluster 2 count = 9 | Cluster 3 count = 17 | Cluster 4 count = 44

Cluster 5 count = 4 | Cluster 6 count = 9 | Cluster 7 count = 32 | Cluster 8 count = 9 | Cluster 9 count = 9

Figure 7 - Vigilance Parameter (ρ) = 0.945 (Dataset 'A')



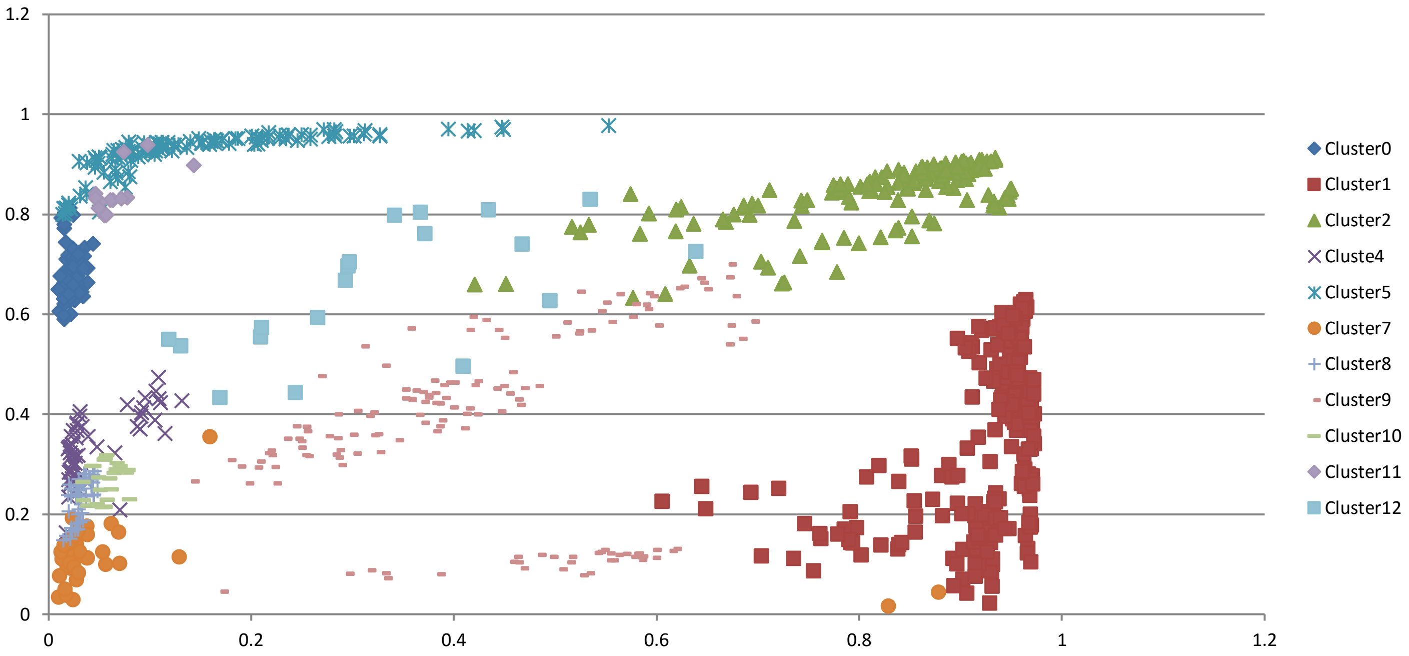
Cluster membership count

Cluster 0 count = 23 | Cluster 1 count = 75 |Cluster 2 count = 77 | Cluster 3 count = 2 | Cluster 4 count = 16 | Cluster 5 count = 38

Cluster 6 count = 4 | Cluster 7 count = 10 | Cluster 8 count = 8 | Cluster 9 count = 23 | Cluster 10 count = 7

Cluster 11 count = 9 |Cluster 12 count = 8

Figure 8 – Clustering Plot for Dataset B



Conclusion

The ability of artificial neural networks with unsupervised learning is already recognized as a means to cluster data in research literatures, this paper provided a practical approach, in terms of the combination of mechanisms to successfully use artificial neural networks to cluster a dataset of mixed data attributes where there is no requirement for a distance function to be calculated or density based parameters to be used. The ART-2 neural network model provides an adequate means for data clustering alongside the other popular methods and Autoencoders a viable means for dimensionality reduction of higher dimensional datasets.

The paper also identified a number of areas of further enquire, in terms of comparison of the approach outline with the more popular clustering approaches; the 'goodness' measure of the clusters identified and the semantic analysis of the clusters to identify its properties.

About the author

Sanjay Debnath, Sr. Architect
IMSS Innovation Office

Sanjay is a Mechanical Engineering graduate from Bangalore University and also holds a PGDBA in marketing and production management.
He has extensively worked on product architectures for a number of server related engines and components and helped build various Internet standards based products for different verticals.
He is currently engaged with Innovation Office of Infrastructure Management and Security Services vertical at Happiestminds Technologies Ltd. in the capacity of Sr. Architect and is leading cyber security related SaaS based product development initiatives with strong emphasis on machine learning and artificial intelligence.

References

[1] Michael Greenacre and Raul Primicerio, "Multivariate Analysis of Ecological Data", Chapter 5 Measures of distance between samples: non-Euclidean [http://www.econ.upf.edu/~michael/stanford/maeb4.pdf]
[2] Gail A. Carpenter, Stephen Grossberg, "ADAPTIVE RESONANCE THEORY" [cns.bu.edu/Profiles/Grossberg/CarGro2003HBTNN2.pdf]
[3] Autoencoder, https://en.wikipedia.org/wiki/Autoencoder
[4] Tyler Yell, " Buy the Higher Low and Sell the Lower High " [https://www4.dailyfx.com/forex/education/trading_tips/chart_of_the_day/2013/05/02/Buy_the_Higher_Low_and_Sell_the_Lower_High.html]
[5] David Weenink, Category ART: A variation on adaptive resonance theory neural net, Institute of Phonetic Sciences, Proceeding 21, 1997 [http://www.fon.hum.uva.nl/Proceedings/Proceedings21/DavidWeenink/DavidWeenink-Contents.html]
[6] Adaptive Resonance Theory, https://en.wikipedia.org/wiki/Adaptive_resonance_theory
[7] pwr-apw implementation of ART-2 Network [https://www.openhub.net/p/pwr-apw] [https://code.google.com/p/pwr-apw/]
[8] Encog Machine Learning Framework, [http://www.heatonresearch.com/encog/]
[9] Yoshua Bengio, Learning Deep Architectures for AI [http://www.iro.umontreal.ca/~bengioy/papers/ftml.pdf]
[10] G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507 [http://www.cs.toronto.edu/~rsalakhu/papers/science.pdf]
[11] D T Pham, S S Dimov, and C D Nguyen, Selection of K in K-means clustering [https://www.ee.columbia.edu/~dpwe/papers/PhamDN05-kmeans.pdf]
[12] Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu, A Density-Based Algorithmfor Discovering Clusters in LargeSpatial Databaseswith Noise [https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf]
[13] Principal component analysis [https://en.wikipedia.org/wiki/Principal_component_analysis]
[14] Jérôme Pagès, Factorial analysis of qualitative and quantitative data both mixed and structured according to a hierarchy [https://www.rocq.inria.fr/axis/modulad///sda11/HCSDA11-Pages.pdf]
[15] Rprop, [https://en.wikipedia.org/wiki/Rprop]