Predicting Bitcoin Using Radian Basis Neural Network

Setareh Roshan
Computer Department
Shahid Rajaee Teacher Training University
Tehran, Iran
setare.roshan1996@gmail.com

***Abstract*— Nowadays predicting bitcoin, and stock market changes is very challenging. There are lots of methods for predicting stock markets. In this article I used RBF, also compared two different clustering method's (k-means and SOM) performance on RBF. Best accuracy achieved using SOM clustering method and it was 80%, but I recommend that we should use k-means because it's much faster (best performance 65%).**

Keywords—bitcoin, k-means, SOM, RBF, neural network, machine learning, unsupervised learning, self organizing map

I. Introduction

Bitcoin is a cryptocurrency invented in 2008 by an unknown person or group of people using the name Satoshi Nakamoto. Bitcoins are created as a reward for a process known as mining. They can be exchanged for other currencies, products, and services,[15] but the real-world value of the coins is extremely volatile.

In this study I will focus on radial basis function (RBF). RBF is a popular kernel function used in various kernelized learning algorithms. Radial Basis Function Networks. Radial basis function (RBF) networks are a commonly used type of artificial neural network for function approximation problems. Depending on the case, it is typically observed that the RBF network required less time to reach the end of training compared to MLP.

I want to know how good RBF works with different clustering methods such as k-means, and SOM. Each one of these methods clusters the train samples in different ways (see Methods). At last, I hope to achieve an acceptable accuracy in advisable time, also in today world bitcoins are extremely valuable, so in this article I will predict the bitcoin market changes.

II. Methods

A. Clustering

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same groups (called a cluster) are more similar (in some sense) than those in other groups (clusters). It can be achieved by various algorithms that in this article I only use two of.

1) K-Means

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. k-means clustering minimizes within-cluster variances using Eq. (1).

$$||a-b|| = \sqrt{\sum_{i}^{n}(a_i - b_i)^2} \quad (1)$$

First, I started with k random values, that their dimensions are as many as input sample's dimensions. This k values can be selected from the training set. In each epoch, I've updated the k values, which they Ire our cluster's centers. I defined a new center for each cluster by averaging the amounts of its members. This process continued until the centers remained unchanged (I considered a threshold: 0.05).

I used Elbow method which finds the maximum distance in each cluster I've found the optimal k.

2) Self Organizing Map

The Kohonen SOM is an unsupervised neural network commonly used for high-dimensional data clustering. It has three layers, an input layer (an n dimensional space), Weight layer, and Kohenon layer which is a computational layer that consists of processing units organized in a 2D structure (or 1D string-like structure) [1].

SOM's distinct property is that they can map high-dimensional input vectors onto spaces with fewer dimensions.

SOM has four stages. First, I initialized weight vectors values randomly. Second, for each neuron I use Eq. (2) to find its best match neuron in Weight layer. Third, a topological neighborhood (using Eq. (2)) of excited neurons emerges around the winning node.

$$T_{j,I}(x) = \exp\left(\frac{-S_{j,I}^2(x)}{2\sigma^2}\right) \quad (2)$$

$S_{j,I}$ is the lateral distance between two neurons j, and I, I(x) represents the wining neuron, and σ is the neighborhood size, the radius of a neighborhood in SOM must decrease over time, and usually, this accomplished by the exponential decay function Eq. (3).

$$\sigma(t) = \sigma_0 \exp(-t/\lambda) \quad (3)$$

The network updates its neurons' weight vectors. All excited neurons adjust their weight vectors values to

become more similar to input patterns. The winning unit's weight vectors are moved closest to the input, and I adjust the weight vectors of the units in its neighborhood too, but to a lesser extent. The farther a unit is from the best matching unit, the less it's modified using Eq. (4).

$$\Delta w_{j,i} = \eta(t) * T_{j,I}(x)(t)(x_i - w_{j,I})$$

(4)

B. Radial Basis Function

The training of the RBF model is terminated once the calculated error reached the desired values or number of training iterations already was completed. An RBF network with a specific number of nodes in its hidden layer is chosen. A Gaussian function Eq. (5) is used as the transfer function in computational units [2].

$$hidden\ layer\ output_j = \exp\left(-\frac{||sample - mean_{cluster_i}||}{2\sigma^2_{cluster_i}}\right)$$

(5)

For each sample in training set I compute its hidden layer output then I will compute output like I did in MLP. In computing hidden layer output, I don't have weights. In contrast in output layer, I have Weights (random initial Weights). For computing output, I use Eq. (6):

$$output_z = 1/(1 + \exp(-hidden\ layer\ output_j * weight_{z,j}))$$

(6)

For each $output_z$ I should multiply all hidden layer output that it's connected to $output_z$ with their weights and perform a sigmoid function on them. At last, I will compute an error (for each iteration) then with this error I will be able to update the Weights (Eq. (7)).

$$new\ weight_{z,k} = old\ weight_{z,j} + \eta * error_z * (hidden\ layer\ output_j - output_z)$$

(7)

Which η is learning rate.

III. Experimental Results

As this is new data, I used a histogram for visualizing the distribution of data (Fig 1), and as it's obvious the distribution of data is Gaussian.

In k-mean clustering, I considered 0.001 as if centers haven't changed much, so algorithm will automatically terminate. I should find an optimal k (number of clusters) using elbow method (Fig 2). Next values after k=30 haven't changed much, so optimal k is 30. It's quite interesting because as you can see in Fig 5 our data (after performing PCA, and converting seventy dimensional data to two) have only 3 clusters.

In SOM clustering I use a network to cluster our data (see Methods). In Fig 4,in epoch 100, you can see I have almost 4 clusters (these figures show weights and their changes through epochs. Fig 3 shows initial weights.), as the epoch raises these 4 will merge into 3 clusters Fig 5.

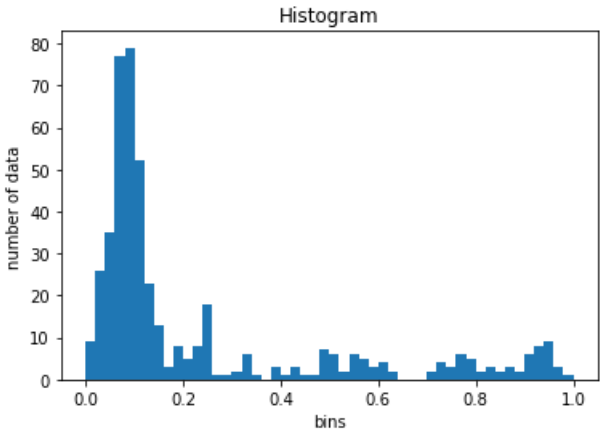

Fig 1 ***Data Histogram****. This a histogram of the bitcoin data. Y-axis represents numbers of data in a bin, and x-axis shows bins. In this diagram I chose 50 for bin.*

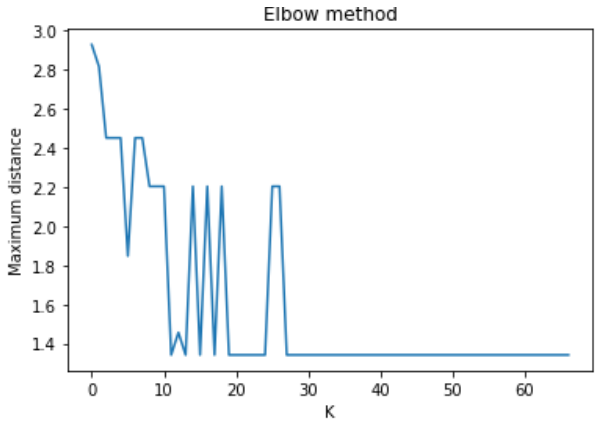

Fig 2 ***Elbow Method****. This diagram represents elbow method on K-Means clustering. Horizontal line shows number of clusters (k) and vertical line shows the maximum distance in each k.*

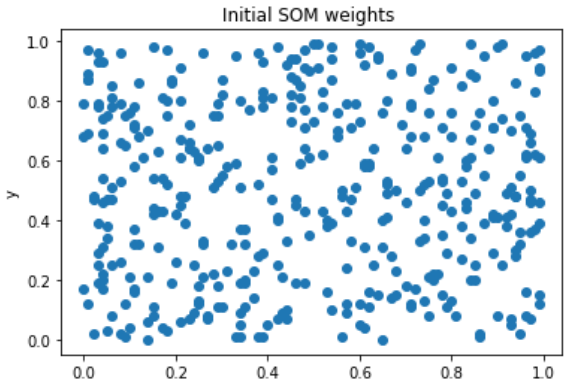

Fig 3 ***SOM Initial Weights****. This diagram shows SOM's random initial Weights. The data is two dimensional as it's obvious in this scatter diagram.*

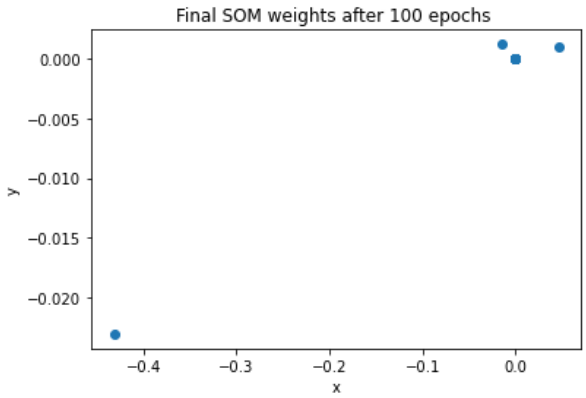

Fig 4 ***SOM Weights****. In 100 epochs, SOM's Weights updated. Figure shows that data can be divided to four clusters.*

Fig 6 shows two clustering algorithms on RBF network. According to this data SOM had better performance.

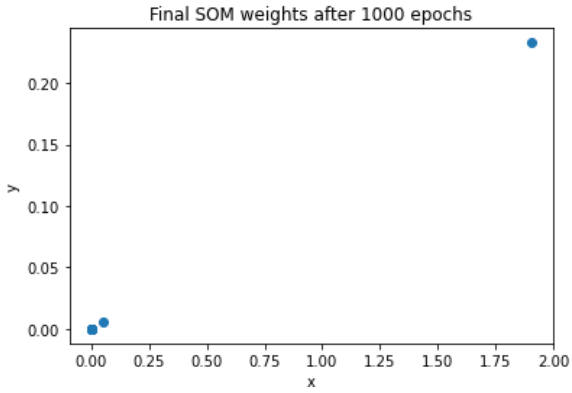

Fig 5 ***SOM Weights****. Shows SOM's Weights in 1000 epochs. There are only two clusters.*

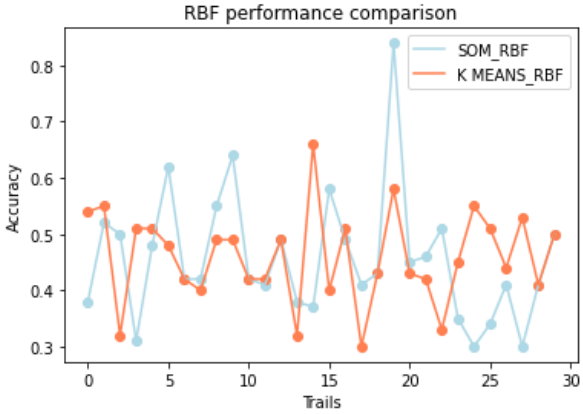

Fig 6 ***RBF Performance.*** *Shows RBF performance. the light blue diagram shows RBF performance with SOM clustering method, and the coral diagram shows it with K-means method.*

IV. Conclusion

Although SOM had better performance, I prefer K-means because it's much faster. In future work I'll use MLP for predicting stock changes.

V. References

[1]. Kohonen, Teuvo; Honkela, Timo. (2007). Kohonen Network. *Scholarpedia*.

[2]. Buhmann, Martin Dietrich. (2003). Radial basis functions : theory and implementations. *Cambridge University Press.*

[3]. Rokach, Lior, and Oded Maimon. (2005). Clustering methods. *Data mining and knowledge discovery handbook. Springer US*, pp. 321-352.