Predicting Cryptocurrencies Prices with Neural Networks

Emad Almasri
Computer Engineering Department,
Istanbul University
Istanbul, Turkey
emadmasri7@gmail.com

Emel Arslan
Computer Engineering Department
Istanbul University
Istanbul, Turkey
earslan@istanbul.edu.tr

***Abstract*—A cryptocurrency is a digital asset designed to work as a medium of exchange that uses cryptography to secure its transactions, to control the creation of additional units, and to verify the transfer of assets.**

Cryptocurrencies are a type of digital currencies, alternative currencies and virtual currencies. Cryptocurrencies use decentralized control as opposed to centralized electronic money and central banking systems. The decentralized control of each cryptocurrency works through a blockchain, which is a public transaction database, functioning as a distributed ledger.

Neural Networks field has many techniques to perform predictions. They are widely used to predict the future values of stock exchange indicators variables. In this paper we will try to use Artificial Neural Network to predict cryptocurrencies close prices, and we'll study the difference in price change with the normal stock exchanges.

Keywords*—Cryptocurrency, Bitcoin, Neural Networks, Feed-Forward Artificial Neural Network, Resilient Algorithm, Back Propagation Learning Algorithm.

I. INTRODUCTION

The concept of cryptocurrency was based on different papers appeared in the last few decades es. The earliest ideas of applying cryptography to cash came from David Chaum in 1983[20]. In October 31, 2008 Satoshi Nakamoto published white paper titled Bitcoin: A Peer-to-Peer Electronic Cash System via "The Cryptography Mailing List" [9]. In his paper he introduced his motivation saying "What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party." In January 3, 2009 Satoshi released Bitcoin source code and software client to the world [9].

The idea of prediction stock exchange prices appeared a long time ago. Many statistical and algorithmic methods were developed to achieve that goal. Stock market prediction is one of the most difficult tasks of time series analysis since the financial markets are influenced by many external social-psychological and economic factors [6]. Since the early 1990's, ANNs have become the most popular machine learning technique used as alternative to standard statistical models in financial time series analysis and prediction [7]. In this paper we will use a feed forward ANN with Back Propagation Learning Algorithm to train the network. The data used in the research to train the network are collected used Crypto Compare API to cryptocurrencies different indicators, for two prediction trials, predicting day close prices and hour close prices using Encog Framework for machine learning.

II. THE CRYPTOCURRENCY, BITCOIN AS A STUDY EXAMPLE

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin [10] (figure 1).

Bitcoin is a peer-to-peer network that maintains a public decentralized ledger of digital math-based assets known as bitcoins. The integrity of this ledger is backed and secured by a subnetwork of computers (miners) who audit and archive its transactions for a reward [9]. A payee can verify the signatures to verify the chain of ownership. Their ownership cannot be changed within the ledger without instructions from their current owner that have been cryptographically authenticated (digital signatures) by a majority of nodes on the Bitcoin network. In essence, "sending a bitcoin" is sending instructions to the network to make a change of custody in the public ledger [9].

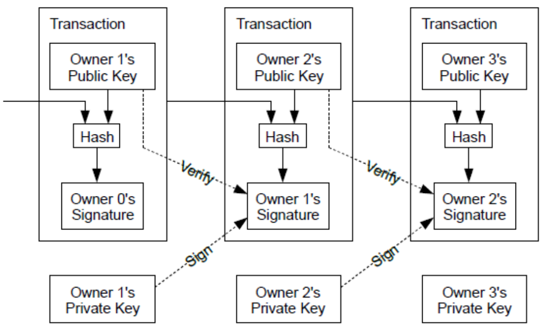

Fig. 1. Figure 1: Satoshi Nakamoto diagram in his paper "Bitcoin: A Peer-to-Peer Electronic Cash System".

III. FEED FORWARD ARTIFICIAL NEURAL NETWORK

A. Artificial Neural Network:

An artificial neural network (ANN), usually called neural network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. In an artificial neural network, simple artificial nodes, variously called "neurons" (Figure 2), are connected together to form a network of nodes mimicking the biological neural networks. The artificial neuron receives one or more inputs (representing the one or more dendrites) sums them to produce an output (representing a biological neuron's axon). Usually the sums of each node are weighted, and the sum is passed through a non-linear function known as an activation function or transfer function. The transfer functions usually have a sigmoid shape, but they may also take the form of other non-linear functions, piecewise linear functions, or step functions [13][14][15][16].

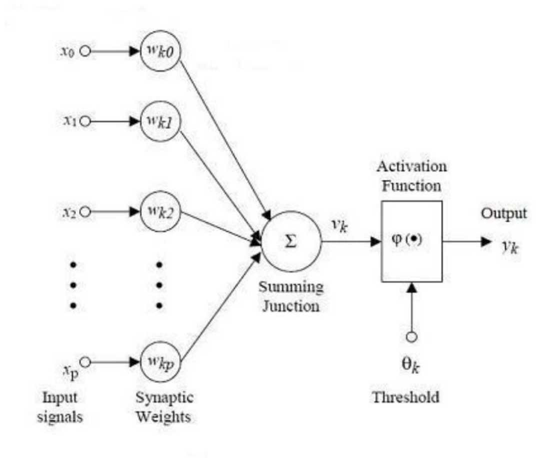

Figure 3: Neuron's components in the Artificial Neural Network. [14]

B. Feed Forward Neural Network:

In our model we use a feed forward neural with hidden layers. in feed forward network neurons are only connected forward. Each layer of the neural network contains connections the next layer without back connections network (Figure 3). Typically, the network consists of a set of sensory units or neurons that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes [14]. The input signal propagates through the network in a forward direction, on a layer by layer basis. Each neuron receives the input signal from the nodes in layer before as a vector:

$$x_i = [x_1\ x_2\ \ \dots.\ x_n]^T$$

So, the net input will be calculated by:

$$u_j = \sum_{i=1}^{n} w_{ji}.x_i$$

Where W_{ji} is the weight matrix from i neuron to j neuron.

And the output to the next layer is calculated like this:

$$y_j = f(u_j) = f\left(\sum_{i=1}^{n} w_{ji}.x_i \right)$$

Where $f(u_j)$ is the activation function.

C. Data Normalization:

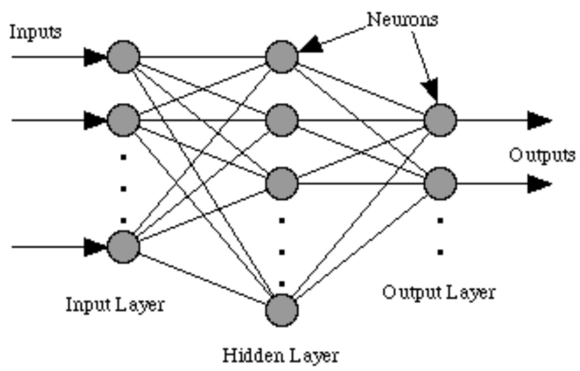

Figure 2: The Feed Forward Neural Network with Hidden Layer. [14]

As the collected data has different values with different scales, it is necessary to adjust and normalize the values at the beginning of the modelling for improving the network training step. The data normalization range is chosen to be [0,1] and the equation for data normalization is given by:

$$X^*_{k,n} = \frac{X_{k,n} - \min(X_k)}{\max(X_k) - \min(X_k)}$$

where X_k is the data series, X^* is the normalized data, $X_{k,n}$ is the original value [11][12].

III. Learning Algorithm, Resilient Back Propagation Algorithm

A. Learning Algorithm:

The property that is of primary significance for a neural network is the ability of the network to learn from its environment. Learning involves adjustments to the synaptic connections that exist between the neurons. Ideally the network becomes more knowledgeable about its environment after each iteration of the learning process. Learning is process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place. A set of well-defined rules for the solution of a learning problem is called a learning algorithm. Learning algorithms differ from each other in the way in which the adjustment to a synaptic weight of a neuron is formulated. Another factor to be considered is the manner in which a neural network relates to its environment. In this context we speak of a learning paradigm that refers to a model of the environment in which the neural network operates [14].

B. Supervised Learning:

Training is the means by which the weights and threshold values of a neural network are adjusted to give desirable outputs, thus making the network adjust the response to the value which best fits the training data. Propagation Training is a form of supervised training, where the expected output is given to the training algorithm [8].

The learning rule is provided with a set of training set of proper network behavior:

$$\{x_1, d_1\}, \{x_2, d_2\}, \ldots., \{x_n, d_n\}$$

x_n: Input to the network.
d_n: Corresponding correct desired output.

As the inputs are applied to the network, the network outputs are compared with the desired outputs.
The weight vector: $w_i = [w_{i1} \quad w_{i2} \quad \ldots \quad w_{in}]$
The learning signal: $r = f(w_i, x, d_i)$

The increment of the weight vector w_i, (Δw_i) produced by the learning step at time t according to the general rule is:

$$\Delta w_i(t) = c.r[w_i(t), x(t), d_i(t)].x(t)$$

The learning rate $c > 0$.
The adapted weight vector:

$$w_i(t+1) = w_i(t) + \Delta w_i(t)$$

In discrete time:

$$w_i{}^{k+1} = w_i{}^k + c.r[w_i{}^k, x^k, d_i{}^k].x^k$$
$$w_i{}^{k+1} = w_i{}^k + \Delta w^k$$

[14].

C. Resilient Back Propagation:

Resilient back propagation (Rprop), an algorithm that can be used to train a neural network, is similar to the more common (regular) back-propagation. But it has two main advantages over back propagation: First, training with Rprop is often faster than training with back propagation. Second, Rprop doesn't require to specify any free parameter values, as opposed to back propagation which needs values for the learning rate (and usually an optional momentum term). The main disadvantage of Rprop is that it's a more complex algorithm to implement than back propagation.

The Algorithm:

```
while epoch < maxEpochs loop
  calculate gradient over all training
     items
  for each weight (and bias) loop
    if prev and curr partials have same
    sign
      increase the previously used delta
      update weight using  new delta
    else if prev and curr partials have
     different signs
      decrease the previously used delta
      revert weight to prev value
    end if
    prev delta = new delta
    prev gradient = curr gradient
  end-for
  ++epoch
end-while
return curr weights and bias values
```

Rprop is based on a mathematical concept called the gradient (figure 4). The idea here is there must be some measure of error (there are several), and that the value of the error will change as the value of one weight changes, that the values of the other weights and biases the same. A gradient is made up of several "partial derivatives." A partial derivative for a weight can be thought of as the slope of the tangent line (the

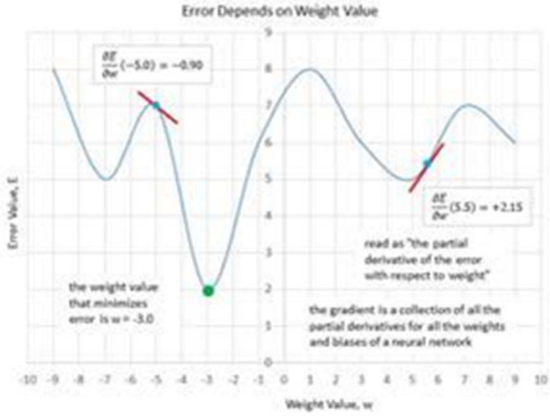

Figure 4: Learning in Rprop [17]

slope, not the tangent line itself) to the error function for some value of the weight. The sign of the slope/partial derivative indicates which direction to go in order to get to a smaller error. A negative slope means go in the positive weight direction, and vice versa. The steepness (magnitude) of the slope indicates how rapidly the error is changing and gives a hint at how far to move to get to a smaller error [17].

III. RESEARCH METHOD AND ENCOG MACHINE LEARNING FRAMEWORK

A. Main Research Method:

To examine the model specified before to predict cryptocurrencies close prices, we built an application contains data collectors and a neural network model based on Encog Machine Learning Framework. We performed tests on different cryptocurrencies on hour/day close prices.

B. Research Data:

The data used in the research are collected using CryptoCompare API, which provides practical API to get Cryptocurrencies data with helpful variable to get suitable data, e.g. sample amount, hour/day/year-based observations [19]. The first set of collected data for training the model belongs to 400 observations from 3-2-2017 to 10 -3-2018 based on daily close price and volume values, and the predictions are made on the close prices of 31 days between March-April 2018. The other set of data belongs to 200 observations based on hour close prices from 27-3-2018 to 3-4 -2018 and the predictions are performed on hour close prices from 4-4-2018 to 11-4-2018.

C. Encog Machine Learning Framework:

Encog is an advanced machine learning framework that supports a variety of advanced algorithms, as well as support classes to normalize and process data. Machine learning algorithms such as Support Vector Machines, Neural Networks, Bayesian Networks, Hidden Markov Models, Genetic Programming and Genetic Algorithms are supported. [18].

D. Chosen Currencies Variables and Parameters:

We chose 3 cryptocurrencies for the test, Bitcoin, Bitcoin Cash and Dash. After performing the component analysis on the data that we have, we chose close price and market volume as input.

After creating the basic network, the basic layers, input and output units and two hidden layers are being added using Encog Framework as specified above. The model is being trained with the first set of data through 1000 epochs, with the advantage that Rprop Algorithm doesn't need extra parameter for learning, and using a sentinel value to avoid going high error values.

IV. Experimental Results

A. *Frist Trail, 31 Days Close Prices:*

The first train data set belongs to 400 observations from 3-2-2017 to 10-3-2018 based on daily close price and market volume values. The predictions are performed on the close prices days between from 10-3-2018 to 11-4-2018. Figures 5-7.

B. *Second Trail, 150 hours Close Prices:*

The other train data set belongs to 200 observations based on hour close prices and market volume values from 27-3-2018 to 3-4-2018. The predictions are performed on hour close prices from 4-4-2018 to 11-4-2018. Figures 8-10.

Figure5: Bitcoin day close prices 10-3-2018 to 11-4-2018, Real values (blue), predictions (green).

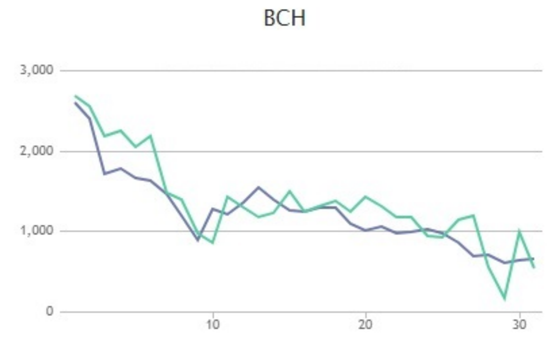

Figure 6: Bitcoin Cash day close prices 10-3-2018 to 11-4-2018, Real values (blue), predictions (green).

Figure 7: Dash day close prices 10-3-2018 to 11-4-2018. Real values (blue), predictions (green).

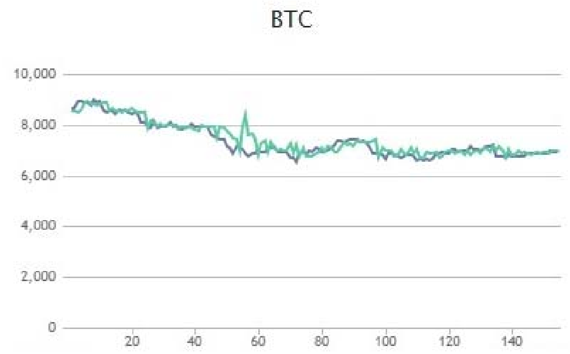

Figure 8: Bitcoin hour close prices 4-4-2018 to 11-4-2018. Real values (blue), predictions (green).

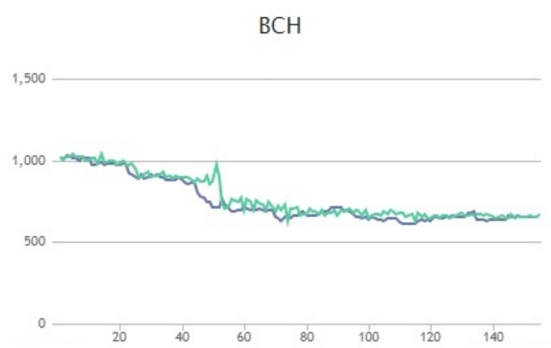

Figure 5: Bitcoin Cash hour close prices 4-4-2018 to 11-4-2018. Real values (blue), predictions (green).

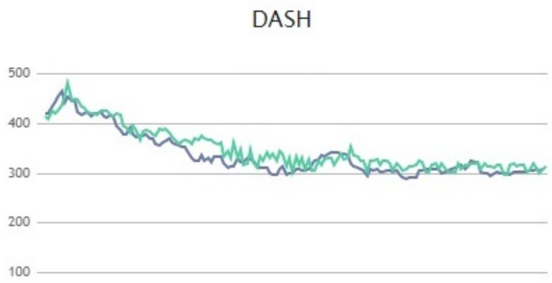

Figure 6: Dash hour close prices 4-4-2018 to 11-4-2018. Real values (blue), predictions (green).

C. Conclusions:

The task of predicting cryptocurrencies close prices seems to be more complex and harder than normal stock exchange predictions.

The main reason beyond the complex prediction task is that cryptocurrencies seem to be influenced by more external social-psychological factors. The different nature, popularity and new usage culture in the internet of the cryptocurrencies. And other complex factor playing important roles in changing the prices that don't exist in normal markets.

The success rate of the predictions varies between 75% to 97.3% for Bitcoin in day close price prediction (Figure 5). But seems to have less variation in other coin (Figures 6-7), which tell us that a specific cryptocurrency may need special model or strategies to get good results.

The hour close-price-based predictions seem to have more stable results, due to the hour based more observable and less variating nature of change in the prices (Figures 8-10).

From the results we suggest using more complex models to predict price changes for the cryptocurrencies.

We suggest also making use of other technologies to analyze web-based news and data related to cryptocurrencies to get better predictions. Data Mining or other technologies may be useful for this goal.

References

1. Andy Greenberg. "Crypto Currency". Forbes.com. (2011).
2. Schuettel, Patrick. The Concise Fintech Compendium. (2017).
3. McDonnell, Patrick. "What Is the Difference Between Bitcoin, Forex, and Gold". NewsBTC. (2015).
4. Allison, Ian. "If Banks Want Benefits of Blockchains, They Must Go Permissionless". NewsBTC. (2015).
5. Matteo D'Agnolo. "All you need to know about Bitcoin". timesofindia-economictimes. (2015).
6. D.N. Gujarati, Basic econometrics. McGraw-Hill, New York. (2003).
7. Schoeneburg, E., Stock Price Prediction Using Neural Networks: A Project Report. Neurocomputing, 1990.
8. Ciumac Sergiu, Financial Predictor via Neural Network. 2011.
9. Winklevoss Tyler and Winklevoss Cameron, Bitcoin, The Internet Money.
10. Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System" 2008.
11. Hakob Grigoryan, Stock Market Prediction using Artificial Neural Networks. Case Study of TAL1T, Nasdaq OMX Baltic Stock.
12. Manna Majumder , MD Anwar Hussian. Forecasting Of Indian Stock Market Index Using Artificial Neural Network
13. Jacek M. Zurada, Introduction to Artificial Neural Systems, PWS Publishing Company. 1995.
14. Simon Haykin, Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Company, 1994.
15. Mohamad H. Hassoun, Foundamentals of Artificial Neural Networks. The MIT Press, 1995.
16. Laurene Fausett, Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, Prentice Hall International, Inc., 1994.
17. James McCaffrey. How to Use Resilient Back Propagation to Train Neural Networks, visualstudiomagazine.com. (2015).
18. Encog Machine Learning, Frameworkhttp://www.heatonresearch.com/encog/
19. CryptoCompare API, https://www.cryptocompare.com/api/
20. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder Bitcoin and Cryptocurrency Technologies. (2016).