

# Cryptocurrency Market Prediction using Neural Networks

Muhammad Obaidullah

mobaidullah@ryerson.ca

Department of Electrical and Computer Engineering  
Ryerson University, Toronto, Canada.

**Abstract**—Trading in cryptocurrency has gained momentum over the past months. Since it is a new market, currencies have very high volatility and it is crucial to detect buy and sell conditions quickly. We use popular Binance crypto exchange’s API to grab time-series price data and tweets from twitter using twitter API to train the Recurrent Neural Network to predict buy and sell condition for over 130 coins. These conditions can be used by the trader to manage large portfolio of many coins enabling diversification and reduced risk.

**Index Terms**—neural networks, cryptocurrency, API, REST, recurrent neural networks, market prediction, non-convex optimization, invex optimization, power allocation.

## I. INTRODUCTION

Two sets of time-series data are collected, one from binance and another from twitter. Twitter data is collected based on popular keywords that are found empirically to be related to the price of cryptocurrency on binance. They are sorted by time. Then they are placed in bins according to the time interval of each bin.

## II. INTERFACING WITH BINANCE

Binance is popular cryptocurrency trading market. In order to grab latest trade data from the exchange, http requests are made to a particular web address and their server responds in JavaScript Object Notation (JSON) formatted data. The response and request format is called an Application Programming Interface (API) and described in the API docs. The format of request and response which conforms to Representational State Transfer (REST) is called a REST-ful API.

```

1 # These API keys are linked to binance account
  mobaidullah786@gmail.com. Be free to change
  these to your account
binance_api_key = "LyYhwIIFK7I4BTAvoCRK8NbhQMgiZ5b
  DaojofJ9dPKtTd3I88gtd27YftqsSOOvb"
3 binance_secret = "7wtspFs3j0Dz4FFjvxbqkif41iakfk9v
  BNmMAGsOtQu7L3JyEiobHbBTrBN9dx7K"
5 def getTickerPrices():
  client = Client(binance_api_key, binance_secret)
  # get all symbol prices
7  return client.get_all_tickers()

```

## III. INTERFACING WITH TWITTER

Twitter is popular social media platform where users can share their thought of market and news networks can update followers on latest news. Users can search certain terms and get all tweets (posts) regarding the search term. Each tweet has a time associated with it. This property is useful to extract time-series data out of tweets. Tweepy python module is used to search for specific terms (@mentions) on twitter and then the results are sorted and put in hourly bins. A day will have 24 hours so a tweet will reside in any of those hour bins.

```

# These api keys are linked to my twitter account
  @OBNinja. Be free to change these for yourself
2 twitter_consumer_key = "UgZySRgdafLiribCS1ovdhIX"
  twitter_consumer_secret = "kNnbEXFwm4oIV2iuBWg9rP5mp
  TKizZ71HZ0AjYqUv3SIYGQkuK"
4 twitter_access_token_key = "96954945-OeqpocX5rfMH1vB
  PID6jpSG2kg7ZVy1kzkXj6Zncb"
  twitter_access_token_secret = "E0yIetCHJtn0rBBSeCPv
  qOuOHpQIgVkfmxYdXqzaimW0"
6
  auth = OAuthHandler(twitter_consumer_key,
  twitter_consumer_secret)
8  auth.set_access_token(twitter_access_token_key,
  twitter_access_token_secret)
10 api = tweepy.API(auth, wait_on_rate_limit=True)
12 def getTweetsWithTime(search_term):
  tweets = {}
  for status in tweepy.Cursor(api.search, q=
  search_term, geocode="
  43.653226,-79.383184,500km", tweet_mode='
  extended').items(1):
16  tweets[status.created_at] = status.full_text
  return tweets

```

## IV. GETTING THE RSS FEEDS

RSS URLs can be added so that the python module can scan for any new feeds and look for any special mentions of keywords. However, most RSS feed servers have a limitation that they store feeds for only 24 hours. Any previous day data of feeds is not available. One potential solution to this is to run a python script forever and collect RSS feeds and submit to a file so that the neural network can keep training and predicting. Once variables are found, old, RSS feeds can be deleted from the file.

```

# Function to fetch the rss feed and return the
  parsed RSS

```

```
2 def parseRSS(rss_url):  
    return feedparser.parse(rss_url)  
4  
# Function grabs the rss feed headlines (titles) and  
# returns them as a list  
6 def getHeadlines(rss_url):  
    headlines = []  
8    feed = parseRSS(rss_url)  
    for newsitem in feed['items']:  
10        headlines.append(newsitem['title'])  
    return headlines
```

## V. CONCLUSION

The resulting bins array from both sources is fed into the Recurrent neural network for training purposes. The eventual goal is to create a LSTM to continuously adapt, learn and predict the crypto market.