Neural Network Modeling on Portfolio Management

James Liao, ID: 490851

Washington University in St. Louis- MS in Business Analytics (Fintech)

Abstract:

Neural Network has applied on various industries to solve the predictive problem for years. Its flexibility and ability to solve both regression and classification problem makes scientists easier to adjust and control parameters for model's optimization; the speed of its algorithm is fast and efficient that it enables scientists' input even more data points, to reach the best training outcome. In this article, I am going to present portfolio management using machine learning tools - Neural network with risk parity optimization method to improve portfolio performance. First, I will select top 15 stocks in the given lists from the cleansed database, then build the global market indicators, which serve as the input variables for the model. After generating simulated return with the lowest MSE, I will apply risk parity method and compare the return, volatility, and Sharpe ratio with those of equal weighted method.

Keywords: machine learning; neural network; risk parity; portfolio; back propagating algorithm; optimization; simulation-based optimization; global indexes

---

1. Introduction

Neural Network method is used for statistical analysis and data modeling which is perceived as an alternative to linear or non-linear regression, and cluster analysis. (Cheng & Titterington 1994). Thus, this methodology has broadly applied in forecasting and modeling. The purpose of neural network is to learn and recognize the patterns in the dataset, generating similar pattern for value forecasting. Its algorithm concepts derived from function of animal's brain, with complicated interconnected neurons passing messages- as model receives data at each stage of input and response with optimized parameters.

In recent years, various machine learning tools are applied on financial fields especially in asset allocation area. Both machine learning methodology and traditional optimal portfolio theory are utilized synchronically in pragmatic portfolio management. Markowitz 1952 pioneered the investment theory by creating a Mean-Variance model, which generate an efficient frontier to adjust portfolios with maximal return or minimal risk. This theory laid a solid foundation for the asset management and open the gate for mathematical optimization for portfolio in the following years. However, it has three main disadvantages: 1. Hard to estimate ex ante 2. Misestimation of returns has great impact on the composition of portfolio 3. The historical risk control strategies may not be extrapolated into the future (Daniel, Winfried, et al 2017). On the contrary, Risk parity improves the portfolio with the stable or even higher return under same risk level as mean -variance method does (Asness, Frazinii, et al.), because it considers equal risk contribution for the portfolio rather than omitting high risk asset, in which mean- variance framework usually results, when targeting low risk tolerance investment environment. Hence, in this article, we will apply risk parity method as the main method for calculating asset allocation weight.

Stock market index is always one of the best indicators to monitor portfolio and adjust investment strategies. It tracks equities and help investors and traders analyze them, comparing the markets efficiency among different countries. It also serves as barometer that shows the characteristics and condition of the markets, reflecting majority of investors' sentiment to the existing trading environment and the cash flow variation in the stock markets. After normalization, indexes from main financial countries have visible effects to the stocks market worldwide (Y.Shapira & D.Y.Kenett et al 2009). Hence, it is our goal to find whether the global indices with lag period can predict the stock return in the US markets.

In this article, I will first utilize risk parity method select the top 15 stocks with relatively stable volatility from the given lists, then apply neural network model to ingest global indices data and historical stocks data, generating the estimated return with lowest MSE. The data will split to training data and out sample testing data. To prevent over-fitting problem, the hidden layers and notches selection is abided by the common rules of thumb. Finally, I will make performance comparison between neural network equal-risk portfolio and equal-weighted portfolio during the date 2021/3/1~3/12.

2. Methodology and Mathematical Definition

The data source is Yahoo Finance. After connecting its API and get the price database, we first transform them into data frame for further data manipulation.

2-1. Database information

Table (1) Database information

Period	2017/01/01 ~2021/3/12, daily interval
Stocks (total 36 stocks)	UPS, UNP, CSX, NSC, FDX, DAL, KEX, UAL, CHRW, ODFL, KNX, ALK, HUBG, ATSG, EXPD, SAIA, LSTR, SAVE, JBHT, SKYW, MRTN, SRI, ALGT, RAIL, CSS, CPA, STNG, NM, EGLE, TK, DSX, GMT, JBLU, KSU, OSG, and R

Market Indices (total 17 indices)	X000001SS, X399001SZ, HIS, N225, KS11, NZ50, JKSE, BSESN, GDAXI, FCHI, SSMI, DJI, GSPC, IXIC, VIX, BVSP, AORD

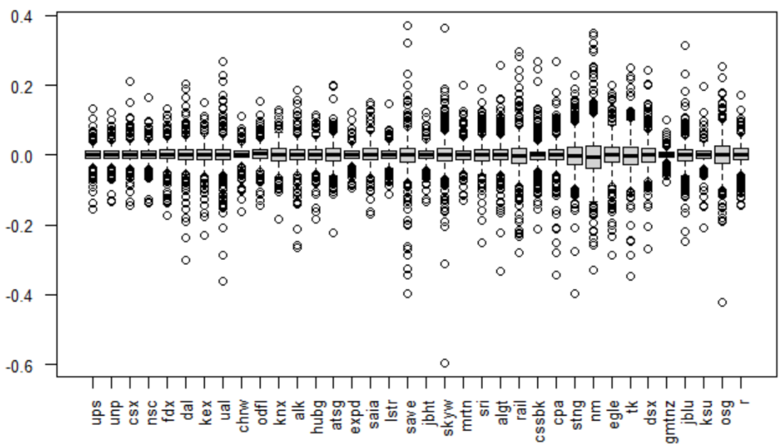
(See appendix (1) for detailed information of indices)

2-2. Data Cleansing

First, the raw data includes NULL value, which stands for the markets closing days. However, it is not appropriate to drop all NULL value in one time since it may delete the markets opening days of other countries. For instance, from the perspective of EST Time zone, Japan markets start on Sunday, when NYSE is still closed. If we drop the NULL value based on the closing time of listing stocks, we may miss the change of price of N225 to predict precisely and authentically. The following is the main concept of data cleansing: For the markets closing days, the price will be updated with the latest value of “last week”, rolling over all rows in the database. Hence, the percentage will not change during the time market closed days and can serve as one of the predictors in a single row. After adjustment, the date of tickers is lagged for one day to fulfill forward prediction on future stock return by using indicators yesterday. Finally, I transform database into log return basis, which is the common method when calculating return continuously. After combining the data, cleansed database includes 1522 rows and 53 columns.

2-3. Descriptive Analytics

Chart (1) Boxplot of tickers



Based on Chart (1), we can clearly find that the log return of almost all stocks has high volatility. Among all, the stock NM has highest standard deviation 0.077, and GMTNZ has the lowest standard deviation 0.01. Chart (2) shows the volatility of listed stocks.

Chart (2) Standard Deviation of listed stocks

```
        nm       rail       save        osg         tk       stng       skyw       egle 
0.07737545 0.05902036 0.05810566 0.05625105 0.05484453 0.05408351 0.05386115 0.05026065 
       ual        cpa        dsx       jblu       algt        alk      cssbk        dal 
0.04708157 0.04491335 0.04269187 0.04200238 0.04133273 0.03936029 0.03901135 0.03826076 
       sri       atsg       saia          r        kex       hubg        knx       mrtn 
0.03811146 0.03470263 0.03405370 0.03403972 0.03378858 0.03008035 0.03003886 0.03001751 
       fdx        csx       odfl        ksu        nsc       jbht        unp        ups 
0.02929816 0.02734237 0.02687664 0.02652302 0.02511658 0.02432365 0.02331630 0.02298598 
      lstr       chrw       expd      gmtnz 
0.02205200 0.02197436 0.02068853 0.01434110 
```

It is obvious that the listed stocks have extreme high volatility which will negatively affect the portfolio performance at the end. To maximum profit under same risk tolerance, or hold a stable return with lower risk, it is necessary to adjust selected elements using risk parity methodology in the beginning and pick those relatively unfluctuating and control maximum drawdown.

2-4 Risk Parity Framework

Risk Parity approach first appear in the *All-Weather* asset allocation strategy released by Bridgewater Associates. Unlike Modern Portfolio Theory (MPT), risk parity allocates assets by equalizing risk contributions without considering historical performance. This method usually found to lead an excess return under same level of portfolio volatility based on CAPM approach (Maillard, S., et al. 2010) (Chaves, et al. 2012).

Consider a portfolio on N assets, where asset of x_i is w_i , and the covariance matrix of assets $X = (x_{1,} x_2 \ldots x_n)$ is $\sum$, the volatility of the portfolio is $w^t X$, which is:

$$\sigma(w) = \sqrt{\sum w \times w'}$$

Since $\sigma(w)$ is homogeneous of degree1 in w, it follows from Euler's theorem that:

$$\sigma(w) = \sum_{i=1}^{N} \sigma_i(w), \text{ where } \sigma_i(w) = w_i \partial_{w_i} \sigma(w) = \frac{w_i \times (\sum w)_i}{\sqrt{\sum w \times w'}}$$

Thus $\sigma(w)$ can be viewed as the contribution of asset i in the portfolio to the overall risk. Equal risk contribution means $\sigma_i(w)$ are the same among asset i, which can be calculated as:

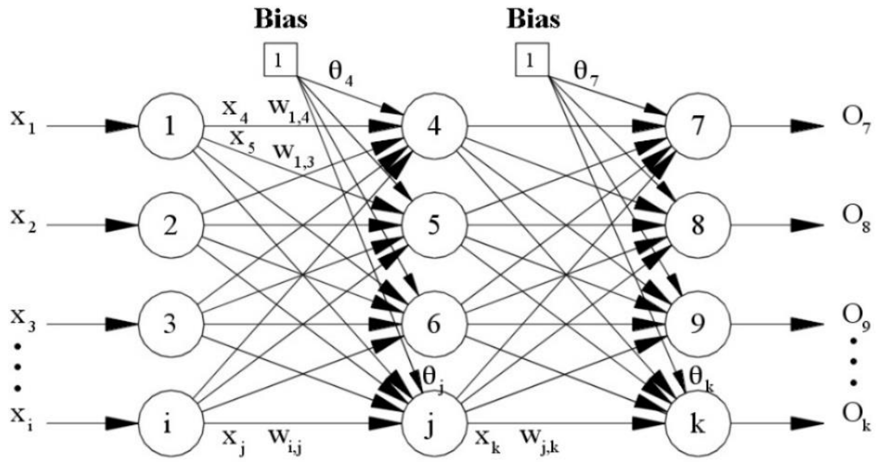
$$\frac{\sigma(w)}{N \times (\sum w)_i}$$

2-5 Artificial Neural Network Framework

Artificial Neural Network was first introduced by Warren and Walter (1943) with based algorithms “Threshold logic”. Threholding is used to avoid the effects of noise and for improving the signal to noise ratio (James. J, 2014). It was mainly used in pattern recognition and classification. Later, Werbo (1975) released the algorithm “Backpropagation” that enable training for multi-layer networks. This method was then popularized by Rumelhart, Hinton, and Williams (1986) through their work on propagating neural networks.

The main concept of back-propagation method is to repeatedly adjusts weights and biases of the connection between input and nodes within layers, to minimize difference between actual and estimated output values by gradient descending approach. The gradient descending process is also called “The Chain Rule”, for it keep adjusting backwardly and forwardly, which affects the parameters afterwards so that it can reassure the parameters are optimized and efficient.

Chart (3) Overview of Neural Networks-Back Propagation concept



Chart(3) is the high level overview of the neural network process. We can further recognize its rules based on the following parameters definition:

Input Layer:

$$x_i = a_i,\ \ \text{i} = 1{,}2{,}3\ldots.i$$

The first set of activations a equals to the input values x_i

Hidden Layer:

$$H_{in} = w_{in} \times x_{in} + b, \qquad i = 1{,}2{,}4..i, \qquad n = 1{,}2{,}3 \ldots number\ of\ layer(j)$$

$$a_{in} = f(H_{in})$$

Output Layer:

$$O_k = w_{in}a_{in},\ \text{i= 1,2,3}\dots k\ ,\ \text{k = 1,2,3}\dots number\ of\ output(k)$$

The final part of neural network is generating predicted value O_k

Back-propagation function apply gradient descending function to minimize weights and biased, which can be written as:

$$\frac{\partial Gradient\ descending\ function\ C}{\partial x} = [\frac{\partial\ C}{\partial x_i}\ , i = 1,2,3\ \dots.]$$

3. Empirical Exploration and Out-of-Sample Test Result

3-1 Stocks selection and neural network construction

In this step, we extract the stocks with lowest risk using risk parity approach, and the top 15 holding will be serving as the selected target and added in the neural network model.

Chart (4) Weight of Listed Equities-unadjusted

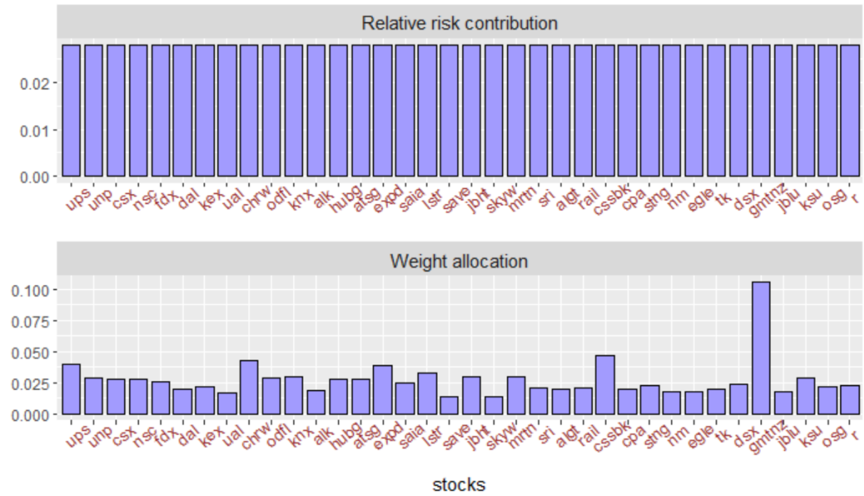


Table (2) Weight of Listed Equities

gmtnz	cssbk	chrw	ups	expd	lstr	jbht	knx
0.10558947	0.04733052	0.04268280	0.03945983	0.03922717	0.03323873	0.03026670	0.03002655
mrtn	unp	odfl	ksu	nsc	atsg	csx	hubg
0.02942160	0.02908303	0.02882649	0.02853414	0.02831871	0.02820883	0.02788862	0.02742109
fdx	saia	dsx	stng	r	osg	kex	sri
0.02574442	0.02467764	0.02433686	0.02336588	0.02259836	0.02173860	0.02152919	0.02133792
rail	cpa	dal	algt	tk	alk	nm	jblu
0.02128664	0.02036691	0.02017252	0.02006324	0.01997579	0.01919577	0.01797283	0.01785432
egle	ual	skyw	save				
0.01764441	0.01699248	0.01397816	0.01364380				

From Chart(4) and Table(2), the top15 selected ticker with largest holding proportion is: gmtz, cssbk, chrw, ups, expd, lstr, jbht, knx, mrtn, unp, odfl, ksu, ncs, atsg, csx. After extracting stocks, it is the process for building neural network model. The training data is 80% of the data start at 2017/01/01, while the 20% left of the data is the testing data. The number of hidden layers is followed by the common rules of thumb (Jeff Heaton, 2017):

1. The number of hidden neurons should be less than 2/3 size of the input layer plus size of the output layer.
2. The number of hidden neurons should be less than twice the size of the input layer.
3. The number of hidden neurons should be between the size of input and output layer.

since we have 16 input variables, the ideal hidden neurons number falls either 15 or 16 based on requirements above. For accuracy, I will select the hidden layer that fit the lowest mean square error (MSE), which regarded as the optimal layer number. According to the Table (3), 16 is the optimal number for hidden layers. Chart (5) displays the graph of logical route in this neural network model.

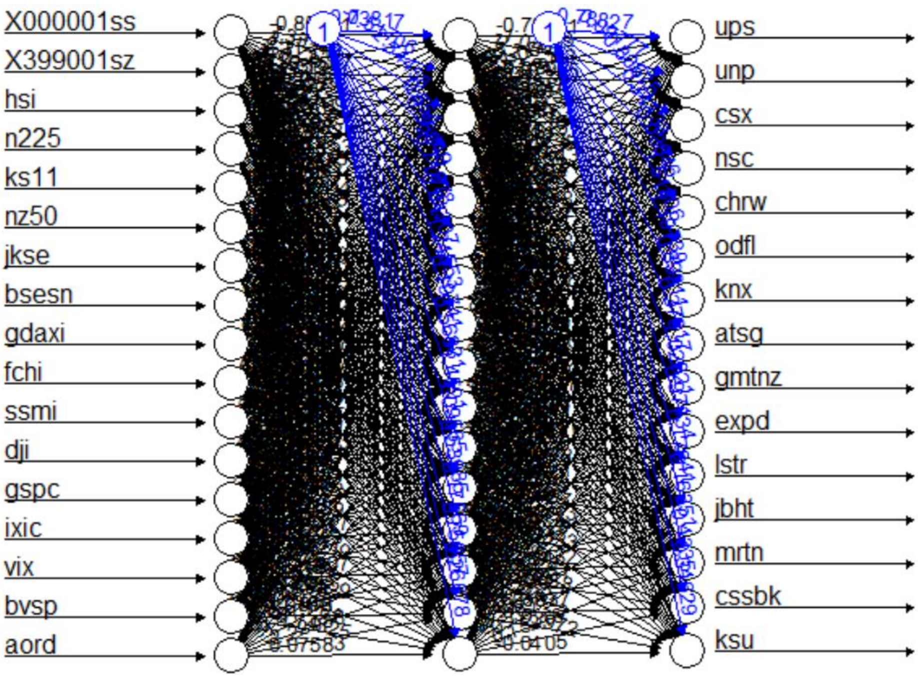
$$\text{MSE function} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widetilde{y_i})^2$$

where y_i is the true value and $\widetilde{y_i}$ is the predicted value.

Table (3) MSE for Training Data

Number of hidden layers	MSE
15	0.0007563
16	0.0007291
17	0.0007888
18	0.0007922

Chart (5) Neural Network Model Construction



3-2 Dynamic Model Results & Comparison

Table (6) is the results of MSE from training data and testing data. Although it is obvious training data has lower MSE, testing data has similar and acceptable range of MSE. Thus, the model's construction is efficient.

Table (6) Sample Test Result

	MSE
Train	0.0007291
Test	0.0007775

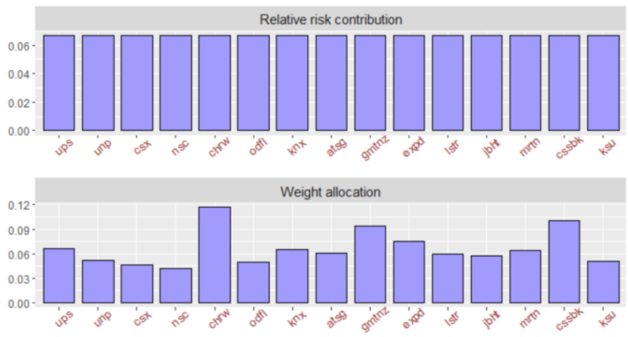
Table (7) shows the weight changes during the period 3/8~3/12. Among all, ticker "chrw" has the largest proportion, while "nsc" has least percentage holding. Based on the Chart(6), the last proportion of "gmtnz" holding is still large as it is before the portfolio being adjusted by selecting certain stocks.

Table (7) Dynamic Weight of Each Stocks during 3/8-3/12

date <chr>	ups <dbl>	unp <dbl>	csx <dbl>	nsc <dbl>	chrw <dbl>	odfl <dbl>	knx <dbl>
0308	0.06620526	0.05231672	0.04663026	0.04218810	0.1162048	0.05030327	0.06472763
0309	0.06497074	0.05334080	0.04709524	0.04241915	0.1145360	0.05005315	0.06239799
0310	0.06423787	0.05400803	0.04678883	0.04247979	0.1156114	0.05061854	0.06189010
0311	0.06427312	0.05404118	0.04625142	0.04301092	0.1155667	0.05170658	0.06236270
0312	0.06385528	0.05282793	0.04510168	0.04283721	0.1173419	0.05176797	0.06805364

(See appendix (2) for full information of weight of stocks)

Chart(6) Weight of Selected Stocks at 3/12

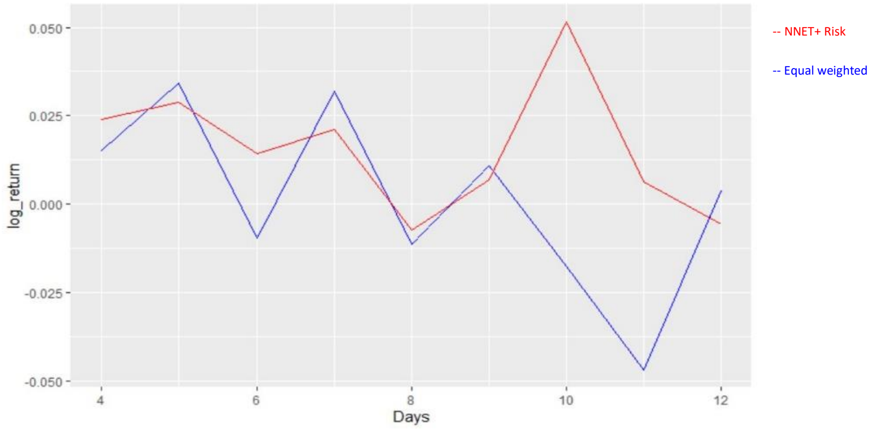


From the table (8), we can find the overall performance between 2021/3/1~3/12 of neural network-risk parity portfolio and equal weighted portfolio. For the mean log return, neural network-risk parity method outweighs the latter for about 13% while standard deviation is 0.3% lower. It is obvious that the risk parity method combined with neural network method improves the portfolio and gains excess return under similar volatility as that of equal-weighted portfolio. The Sharpe ratio also 4% higher than unadjusted portfolio. Based on the result, we can make a conclusion that using risk parity and neural network approach can be the better strategy comparing with equal-weighted investment. Chart (7) is the trend of log return of both portfolio strategies during the date 2021/3/1~3/12. We can notice that the trend of neural network-risk parity model (red line) has lower volatility and higher return contrasting the trend of equal-weighted approach (blue line).

Table (8) Portfolio performance (3/1-3/12)

	Mean log return	Standard deviation	Sharpe Ratio
NNet-risk parity	0.1404	0.0191	0.6855
Equal-weighted	0.0105	0.0250	0.1326

Chart(7) Trend of two portfolios



4. Conclusions and Future Works

In this paper, we apply machine learning-Neural Network-back propagating approach and Risk Parity method to build a global indices model that forecast future stock return and optimize the portfolio dynamically based on balanced stocks allocation. From the result, we can find that the adjusted portfolio strategy with the aid of neural network algorithms performs better than equal-weighted portfolio strategies, with higher mean log return, lower volatility, and higher Sharpe ratio. However, there is still some work can be improved or finished in the future research. For example, the hidden layers selection can be decided by more quantitative approach rather than by the thumb rules. In the future study, the goal may focus on whether it is possible to find the ideal input and output number for neural network approach under different time, whether other common modern portfolio theory fit the machine learning methods well. Moreover, the liquidity adjusted value at risk (LVaR) method can be applied as well to further calculate tail risk and liquidity risk of the portfolio using machine learning tools. As a result, it could be a better asset allocation method in the fluctuating financial markets if more detailed but critical factors are considered.

Appendix (1)

Market Indices	Name	Country
X000001ss	SSE Composite Index	China
X399001sz	Shenzhen Component	China
hsi	HANG SEN INDEX	China
n225	Nikkei 225	Japan
ks11	KOSPI Composite Index	Korea
nz50	NZX50 Index	New Zealand
jkse	Jakarta Stock Exchange Index	Indonesia
bsesn	S&P BSE SEBSEX	India
gdaxi	DAX PERFORMAN CE-INDEX	German
fchi	CAC 40	French
ssmi	Swiss Market Index	Switzerland
dji	Dow Jones Industrial Average	USA
gspc	S & P 500	USA
ixic	Nasdaq Composite	USA
vix	VIX	USA
bvsp	IBOVESPA	Brazil
aord	ALL ORDINARIES	Australia

Appendix (2)

date <chr>	ups <dbl>	unp <dbl>	csx <dbl>	nsc <dbl>	chrw <dbl>	odfl <dbl>	knx <dbl>
0308	0.06620526	0.05231672	0.04663026	0.04218810	0.1162048	0.05030327	0.06472763
0309	0.06497074	0.05334080	0.04709524	0.04241915	0.1145360	0.05005315	0.06239799
0310	0.06423787	0.05400803	0.04678883	0.04247979	0.1156114	0.05061854	0.06189010
0311	0.06427312	0.05404118	0.04625142	0.04301092	0.1155667	0.05170658	0.06236270
0312	0.06385528	0.05282793	0.04510168	0.04283721	0.1173419	0.05176797	0.06805364

atsg <dbl>	gmtnz <dbl>	expd <dbl>	lstr <dbl>	jbht <dbl>	mrtn <dbl>	cssbk <dbl>
0.05917413	0.09416297	0.07429284	0.05889937	0.05717446	0.06519620	0.1020519
0.05823676	0.09565422	0.07408262	0.05758996	0.05757142	0.06562917	0.1055925
0.05633159	0.09608602	0.07422334	0.05841195	0.05833077	0.06599455	0.1041387
0.05642984	0.09365165	0.07617529	0.05967096	0.05792034	0.06493240	0.1030640
0.05521365	0.08774985	0.07489840	0.06088398	0.05885644	0.06951060	0.1008330

gmtnz <dbl>	expd <dbl>	lstr <dbl>	jbht <dbl>	mrtn <dbl>	cssbk <dbl>	ksu <dbl>
0.09416297	0.07429284	0.05889937	0.05717446	0.06519620	0.1020519	0.05047218
0.09565422	0.07408262	0.05758996	0.05757142	0.06562917	0.1055925	0.05083028
0.09608602	0.07422334	0.05841195	0.05833077	0.06599455	0.1041387	0.05084848
0.09365165	0.07617529	0.05967096	0.05792034	0.06493240	0.1030640	0.05094298
0.08774985	0.07489840	0.06088398	0.05885644	0.06951060	0.1008330	0.05026840

References

Haesen.D & Hallerbach.W et al 2017 Enhancing Risk Parity by Including Views. *The journal of Investing*

Jiayang Yu & Kuo-Chu Chang 2020. Neural Network Predictive Modeling Dynamic Portfolio Management- A Simulation-Based Portfolio Optimization Approach. *Journal of Risk and Financial Management*

YOUNES DJEHICHE 2018. A neural Networks Approach to Portfolio Case. *KTH VETENSKAP OCH KONST*

Obeidat. S & Shapiro.D et al 2018. Adaptive Portfolio Asset allocation Optimization with Deep Learning. *International Journal on Advances in Intelligent Systems*

James, J 2014. Threshnomics: An Introduction to Threshold Logic in Algorithms and Circuits. *Journal of Computer Science & Systems Biology*

Y. Shapira, D.Y.Kenett & E.Ben-Jacob 2009.The Index Cohesive effect on stock market correlations. *The European Physical Journal*

David.E.Rumelhart, Geoffrey E. Hinton & Ronald J. Williams 1986. Learning representations by back-propagating errors. *Nature*

Jeff Heaton 2017. The Number of Hidden Layers Research. *Hidden Research website*

Simeon Kostadinov 2018. Understanding Backpropagation Algorithm. *Towards Data Science website*

Cheng & Titterington 1994. Neural Networks: A Review from a Statistical Perspective.*Statist*

Asness, Frazinii, et al 2012. Leverage Aversion and Risk Parity. *Financial Analysts Journal.*

Chaves, et al. 2012. Risk Parity Portfolio vs Other Asset Allocation Heuristic Portfolios. *The journal of Investing*

Maillard.S et al 2010. The Properties of Equally Weighted Risk Contribution Portfolio. *The Journal of Portfolio Management*