

Modelling and trading commodities with a new deep belief network

Andreas Karathanasopoulos*

University of Dubai, Dubai

Received: 4 August 2016

Revised: 1 May 2017

Accepted: 18 May 2017

Abstract

The scope of this project is to study a novel methodology in the task of forecasting and trading the crack spread modelled index. More specifically in this research we are expanding the earlier work carried out by Karathanasopoulos et al. (2016c) and Dunis et al. (2005) who model the Crack Spread with traditional neural networks. In this research paper we provide for first time a more advanced approach to non-linear modelling and trading the 'Crack'. The selected trading period covers 4500 trading days and the proposed model is a deep belief network (DBN). To model, test and evaluate the crack spread we use an expansive universe of 500 inputs correlated with the main index. Moreover we have used for reasons of comparison a radial basis function combined with partial swarm optimizer and two linear models such as random walk theorem and buy and hold strategy.

Keywords: spread trading; deep belief networks; PSO RBF neural networks

JEL Classification Codes: Q02, C15, C53

1. Introduction

In this paper, we propose a deep belief network DBN for short-term prediction of the crack spread index. The contribution of this paper it's coming in two different places. In first place it is the first time that deep beliefs networks have been used in the area of oil predictions. Secondly it is first time that 500 inputs have been used for the training of deep beliefs networks. The basic idea behind this analysis is to apply 3 different oil commodities in the equation 1 and try to predict one day ahead the Crack Spread. All the three commodities under our research are crude oil, gasoline and heating oil and all of them are traded in the New York Mercantile Exchange

* E-mail: andreas.kara@hotmail.com.

Citation: Karathanasopoulos, A. (2017) Modelling and trading commodities with a deep belief network, *Economics and Business Letters*, 6(2), 28-34.

(NYMEX). After running both artificial intelligent models we reveal that neural networks offer interesting results and seems that outperform the traditional linear models.

Related now to the Crack Spread it's quite straightforward and easy to be calculated. As previously mentioned we are using three commodity variables. The input variable is the crude oil (CL) which is denominated in US dollars per barrel while the outputs consist of gasoline (RBOB) and heating oil (HO) which both are denominated in US cents per gallon. In terms of creating the equation and applying our three variables we need to convert them in the same units. Because the quantity of a crude oil contract is coming in 1,000 barrels per contract and both the gasoline and heating oil amount are coming to 42,000 gallons per contract, the latter two needs to be multiplied by 0.42. After these adjustments the equation is taking the below form.

$$3 : 2 : 1 \text{ CRACK SPREAD } S_t = \frac{((2 \times \text{RBOB} \times 0.42) + (1 \times \text{HO} \times 0.42)) - (3 \times \text{CL})}{3} \quad (1)$$

The final form of equation 1 is the below where we are converting the variables of daily prices to returns. Butterworth and Holmes (2002), Dunis et al. (2006) and Karathanasopoulos et al. (2016b) have used the same way to forecast the crack spread.

$$\Delta S_t = \left[\left(\frac{(P_{\text{RBOB}(t)} - P_{\text{RBOB}(t-1)})}{(P_{\text{RBOB}(t-1)})} + \frac{(P_{\text{HO}(t)} - P_{\text{HO}(t-1)})}{(P_{\text{HO}(t-1)})} \right) - \left(\frac{(P_{\text{CL}(t)} - P_{\text{CL}(t-1)})}{(P_{\text{CL}(t-1)})} \right) \right] \quad (2)$$

2. Literature review

Deep belief networks DBNs have been proposed by Hinton et al (2006). The DBN is a layer based model where the bottom layer is observable and the other multiple hidden layers are created by stacking multiple restricted Boltzmann machines RBMs on top of each other. Hinton and Salakhutdinov (2006) and Kang and Choi (2011) have used the DBNs and restricted Boltzmann machines RBMs to solve many problems such as forecasting electricity prices, Lee et al (2009) present a convolutional deep belief network in the use of measuring the realistic image sizes. Liu (2010) proposed a classification approach based on deep belief networks that outperform traditional support vector machines. In the meantime Zhou et al, (2010) have used successfully a discriminative deep belief network to address the image classification problem with limited data and seems that deep beliefs networks outperform all the other algorithms in their research. Finally Chao et al (2011). did a preliminary use of deep beliefs networks in the area of time series forecasting applying them in exchange rate predictions. Deep beliefs show an excellent performance beating the other generic feedforward neural networks. To the best of our knowledge, this is the first time that a deep belief network has been used in oil predictions.

For comparison reasons in this research paper we are using a hybrid neural network architecture, named radial basis function combined with partial swarm optimizers to forecast the price of the crack spread. Nekoukar and Beheshti (2010) and Shen et al. (2011) state that this hybrid methodology is superior to meta-heuristic methods that are widely used in the literature as it eradicates the risk of being trapped in local optima and the final solution is assured to be optimal for a subset of the training set. For literature review reasons its worth mentioning that Kennedy and Eberhart, (1995) have created the PSO optimizers and Ding et al (2005) have used it for optimizing the structure of the radial basis function network. In this paper we are using the same hybrid model that Sermpinis et al. (2013) and Karathanasopoulos et al. (2013, 2015, 2016a, 2016b) have used successfully in different forecasting tasks.

3. Data

As we have described in the first part, in the prediction of the crack spread we will use three different future contracts such as Crude oil (WTI), Unleaded Gasoline (RBOB) and Heating oil. In terms of training both neural networks we will divide the data set into training and validation sub sets. Below we are presenting how we are dividing the whole data set. In terms of optimizing the data set we used around 100 different data combinations. The one presenting in the table 1 is the only one that gives the best performance for both nonlinear models.

Table 1. Dataset

<i>Name of Period</i>	<i>Number of trading days</i>	<i>Beginning</i>	<i>End</i>
Full Sample	4500	1/1/2000	31/12/2016
Training Period	3200	1/1/2000	1/1/2011
Validation Period	1300	2/1/2012	31/12/2016

4. Trading signals and strategy

Trading signals are generated based on directional forecasts of commodity prices. When a model forecasts a negative return, then a short position (sale) is assumed at the close of each day and when the model forecasts a positive return a long position (purchase) is executed. Profits and losses are computed using daily data.

5. Forecasting models

5.1. Random walk model

The first benchmark we employ is the random walk forecast of log crude oil prices. The random walk model is a widely used benchmark in the crude oil prediction literature, Bauimester and Kilian (2012), Alquist et al. (2013) and Karathanasopoulos et al. (2014). Let Y_t denote the log price of oil on day t . According to the random walk model, tomorrow's price of crude oil is equal to today's price. That is:

$$\hat{Y}_{t+1} = Y_t, \quad (3)$$

where \hat{Y}_{t+1} is the predicted log oil price on day $t+1$. Consistent with the existing literature, we start by evaluating the predictive and trading performance of the random walk model.

5.2. Buy-and-hold strategy

This simple strategy assumes that the investor opens a long position and keep the position open till the moment his investment grows, instead of selling it.

5.3. Radial basis functions - Particle swarm optimization

A radial basis function neural network (RBF) is a feedforward neural network where hidden units do not implement an activation function, but a radial basis function. In other words, they have the same structure as MLPs but the hidden nodes have the Gaussian function instead of the sigmoid. Particle swarm optimization (PSO) is a stochastic optimization technique introduced by Eberhart and Kennedy (1995). It based on the social behaviour of of birds within a flock. In the first stage, a random population of individuals (birds) is generated. Each individual in the population is called particle. All particles are evaluated through a fitness

function and have an initial fitness price. Each particle flies through the problem space with a specific velocity that directs the flying. The aim of all particles is to find the optimal solution. This is achieved through a number of iterations. The maximum number of iterations is 350 and on average the optimal solution is coming after 100 iterations.

5.4. Deep beliefs networks

Deep belief network is a new probabilistic feed-forward forecasting tool with input layer and output layer (see figure 1).

The basic idea behind this new model is the use a layer by-layer unsupervised learning method in order to pre-train the initial values of the weights in the neural network. A layer-by-layer unsupervised training procedure implies that each layer captures the features of the previous one, and transferring them to next one. In this research paper each pair of layers is pre trained by using the restricted Boltzmann machines. Restricted Boltzmann machines are composed from 2 different layers connected together. One layer has visible nodes/neurons and the other hidden nodes/neurons (see figure 2).

Figure 1. Deep belief network structure

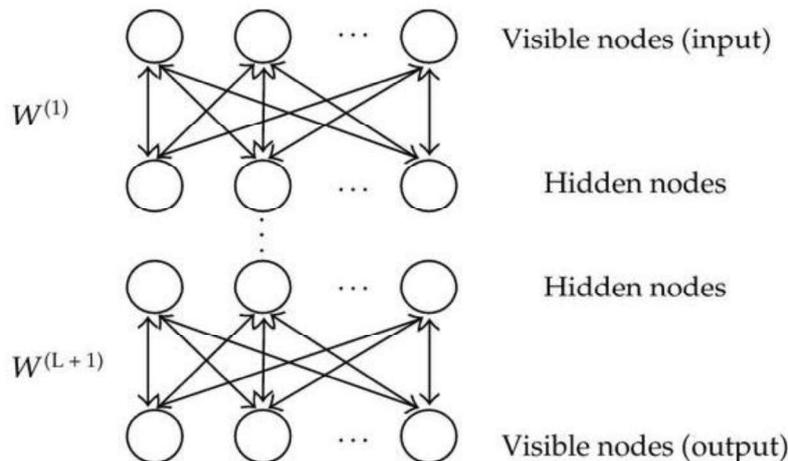
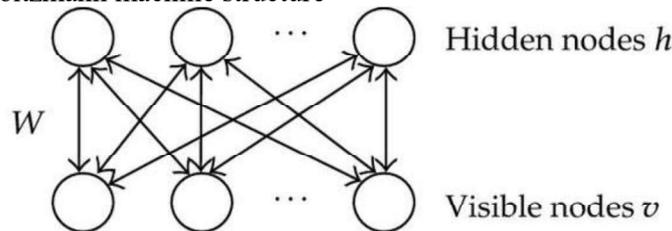


Figure 2. Restricted Boltzmann machine structure



The nodes on each layer have no connections between them only with units of other layers. All these connections are symmetric and bidirectional. As i mentioned before the restricted Boltzmann machines have been used in a numerous tasks but in our experiment are providing the learning training for the structure of the deep belief network. In few words restricted Boltzmann machines are a special type of generative energy based models that can learn a probability distribution over its set of inputs. Further to that the standard type of restricted Boltzmann machine has binary valued hidden and visible nodes.

In the below equation v_i and h_j represent the states of visible node i and hidden node j . For binary state nodes, where is, v_i and $h_j \in \{0, 1\}$, the state of h_j is set to be 1 with probabilities:

$$p_{h_j} = p(h_j = 1|v) = \sigma\left(b_j + \sum_i w_{ij}v_i\right) \quad (4)$$

where $\sigma(\cdot)$ is the logistic sigmoid function, b_j is the bias of j , v_i is the binary state, w_{ij} is the weight between v_i and h_j .

After choosing binary state nodes for the hidden units, we set the state of v_i to be 1 with probability:

$$p_{v_i} = p(v_i = 1|h) = \sigma\left(b_i + \sum_j w_{ij}h_j\right) \quad (5)$$

In terms of training the network firstly a training sample is presented to the visible nodes, and the $\{v_i\}$ is obtained. Then the hidden nodes state that $\{h_j\}$ are sampled according to probabilities in equation 4. This process is repeated till the moment we will update the visible and then the hidden nodes, and the developed states v_i and h_j are obtained. Hence the weights are coming in the below form:

$$\Delta w_{ij} = \eta\left(\langle v_i h_j \rangle - \langle v'_i h'_j \rangle\right) \quad (6)$$

where η is the learning rate, and $\langle \cdot \rangle$ refers to the expectation of the training data.

The continuous restricted Boltzmann has been built from Chao et al. (2011) and Chen and Murray (2002) where the inputs nodes are represented as s_i , and the output nodes as s_j :

$$s_j = \phi_j\left(\sum_i w_{ij}s_i + \sigma \cdot N_j(0, |1)\right) \quad (7)$$

where $\phi_j(x)$ is a sigmoid function, $N_j(0, 1)$ represents a Gaussian unit, σ is a constant, and a_j is a “noise-control” parameter which controls the slope of the sigmoid function.

Instead of building the DBN we are using as many RBMs as the number of hidden layers In order to run the DBN model to forecast the Crack spread forecast we are using two hidden layers. Finally, in terms of constructing the DBN we will use the backpropagation learning algorithm in terms of getting better optimal outputs.

6. Empirical results

In tables 2 and 3 we are presenting the statistical and empirical performance in the out-of-sample period for all the models. The measures we used are the Mean Absolute Error (MAE), the mean absolute percentage error (MAPE) the Root Mean Squared Error (RMSE), the annualized returns and the Sharpe ratio.

In this section we present the results of the proposed methodologies applied to the crack spread in the relevant out-of-sample period. The trading rule that we follow is simple. We go or stay long when the forecasted returns are above zero and go or stay short when the forecasted returns are below zero.

After taking consideration the above results its straightforward that the new feed forward deep belief network outperforms the RBF-PSO and the other two linear models. In the meantime, these findings are not coming from a single run of the models but from the average of 100 runs. It's important to minimize the error of any stochastic model with some Monte Carlo simulation.

Table 2. Out of sample statistical performance for Crack Spread

<i>Forecast</i>	<i>Random walk</i>	<i>Buy and Hold</i>	<i>RBF-PSO</i>	<i>Deep belief Network</i>
MAE	0.0230	0.0205	0.0190	0.0134
MAPE	209.66%	200.30%	190.33%	183.44%
RMSE	0.0190	0.0179	0.0167	0.0125

Table 3. Out of sample trading performance results for Crack Spread (including cost)

<i>Forecast</i>	<i>Random walk</i>	<i>Buy and Hold</i>	<i>RBF-PSO</i>	<i>Deep belief Network</i>
Annualised returns	7.28%	9.56%	15.87%	16.45%
Sharpe ratio	0.56	0.67	1.27	1.78

7. Concluding remarks

In this research paper, we propose a deep belief network DBN for oil predictions. More specifically the forecasting deep belief network is used to forecast the crack spread modelled index. Its first time in the whole literature that such model has been used in the area of data series analysis. Moreover it's quite important to mention that the number of inputs exceeds the 500 series. The evaluation of the forecasted returns has been measured through statistical and empirical analysis. After benchmarking the new methodology with a random walk model, a buy and hold strategy and a radial basis functions model combined with partial swarm optimizer we are coming to a clear result that DBNs are performing successfully in the real world and seem to be a huge challenging tool for managers and traders.

References

- Alquist, R., Kilian, L. And Vigfusson, R.J. (2013) Forecasting the Price of Oil, *Handbook of Economic Forecasting*, Elsevier.
- Baumeister, C. and Kilian, L. (2012) Real-Time Analysis of Oil Price Risks Using Forecast Scenarios, *Staff Working Papers*, 12, 1, Bank of Canada.
- Butterworth, D. and Holmes, P. (2002) Inter-Market Spread Trading: Evidence from UK Index Futures Markets, *Applied Financial Economics*, 12(11), 783-791.
- Chao, J., Shen, F. and Zhao, J. (2011) Forecasting exchange rate with deep belief networks, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '11)*, 1259–1266, San Jose, Calif, USA,
- Chen, H. and Murray, A.F. (2002) A continuous restricted Boltzmann machine with hardware-amenable learning algorithm, *Proceedings of the 12th International Conference on Artificial Neural Networks*, 358–363.
- Ding, H., Xiao, Y. and Yue, J. (2005) Adaptive Training of Radial Basis Function Networks Using Particle Swarm Optimization Algorithm, *Lecture Notes in Computer Science*, 3610, 119- 128.
- Dunis, C.L., Laws, J. and Evans, B. (2005) Modelling and Trading The Gasoline Crack Spread: A Non-Linear Story, *Derivatives Use, Trading & Regulation*, 12 (1-2), 126-145.
- Dunis, C.L, Laws, J. And Evans, B. (2006) Modelling and Trading the Soybean-Oil Crush Spread with Recurrent and Higher Order Networks: A Comparative Analysis, *Neural Network World*, 13(3/6), 193-213.

- Hinton, G.E. and Salakhutdinov, R.R. (2006) Reducing the dimensionality of data with neural networks, *Science*, 313(5786), 504–507.
- Hinton, G.E., Osindero, S. and Teh, Y.W. (2006) A fast learning algorithm for deep belief nets, *Neural Computation*, 18(7), 1527–1554,
- Kang, Y. And Choi, S. (2011) Restricted deep belief networks for multi-view learning, *Lecture Notes in Computer Science*, 6912, 130–145.
- Karathanasopoulos, A., Dunis, C. and Khalil, S. (2016a) Modelling, forecasting and trading with a new sliding window approach: the crack spread example, *Quantitative Finance*, 16(12), 1875-1886.
- Karathanasopoulos, A., Dunis, C., Likothanassis, S., Sermpinis, G. and Theofilatos, K. (2013) A Hybrid Genetic Algorithm-Support Vector Machine Approach in the Task of Forecasting and Trading, *Journal of Asset Management*, 14, 52-71.
- Karathanasopoulos, A., Mitra, S., Skindilias, K. and Lo, C.C. (2016b) Modelling and Trading the English and German Stock Markets with Novelty Optimization Techniques, *Journal of Forecasting*, in press.
- Karathanasopoulos, A., Sermpinis, G., Laws, J. and Dunis, C. (2014) Modelling and Trading the Greek Stock Market with Gene Expression and Genetic Programming Algorithms, *Journal of Forecasting*, 33(8), 596-610.
- Karathanasopoulos, A., Sermpinis, G., Stasinakis, C. and Theofilatos, K. (2015) Forecasting US Unemployment with Radial Basis Neural Networks, Kalman Filters and Support Vector Regressions, *Computational Economics*, 1-19.
- Karathanasopoulos, A. (2016c) Modelling and trading the English stock market with novelty optimization techniques, *Economics and Business Letters*, 5(2), 50-57.
- Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization, *Proceedings of the IEEE International Conference on Neural Networks*, 4, 1942-1948.
- Lee, H., Grosse, R., Ranganath, R. and Ng, A.Y. (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, *Proceedings of the 26th International Conference On Machine Learning*, 609–616.
- Liu, T. (2010) A novel text classification approach based on deep belief network, *Lecture Notes in Computer Science*, 6443, 314–321,
- Nekoukar, V. and Beheshti, H. (2010) A Local Linear Radial Basis Function Neural Network for Financial Time-Series Forecasting, *Applied Intelligence*, 33(3), 352-356.
- Sermpinis, G., Theofilatos, K., Karathanasopoulos, A., Georgopoulos, E. and Dunis, C. (2013) Forecasting Foreign Exchange Rates with Adaptive Neural Networks Using Radial-Basis Functions and Particle Swarm Optimization, *European Journal of Operational Research*, 225(3), 528-540.
- Shen, W., Guo, X., Wu, C. and Wu, D. (2011) Forecasting Stock Indices Using Radial Basis Function Neural Networks Optimized by Artificial Fish Swarm Algorithm, *Knowledge-Based Systems*, 24(3), 378-385.
- Zhou, S., Chen, Q. and Wang, X. (2010) Discriminative Deep Belief networks for image classification, *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, 1561–1564.