Long & Short Classifications of Crude Oil Futures

Conly Sharp

May 2, 2024

Outline

Introduction

Goal:
To construct a convolutional neural network which can accurately classify a subset of a given securities historical closing price into either a "long" or "short" position based on the last thirty days of trading data.

Introduction

Goal:
To construct a convolutional neural network which can accurately classify a subset of a given securities historical closing price into either a "long" or "short" position based on the last thirty days of trading data.

We'll specifically look at the closing price of West Texas Intermediate Crude Oil. The value of this security is the price you would pay at the end of a futures contract expiry for the specified number of crude oil barrels listed in the contract.

Goal:
To construct a convolutional neural network which can accurately classify a subset of a given securities historical closing price into either a "long" or "short" position based on the last thirty days of trading data.

We'll specifically look at the closing price of West Texas Intermediate Crude Oil. The value of this security is the price you would pay at the end of a futures contract expiry for the specified number of crude oil barrels listed in the contract.

Using subsets of thirty days, we convert each subset into *Gramian Angular Field* images, which preserve the temporal connection of the data and encodes it into a two-dimensional image.

Gramian Angular Fields

What are they?

Gramian Angular Fields

What are they?

Gramian Angular Fields refers to a new technique for encoding time series data into images, first introduced by Oates & Wang in 2015 [1].

Gramian Angular Fields

What are they?

Gramian Angular Fields refers to a new technique for encoding time series data into images, first introduced by Oates & Wang in 2015 [1].

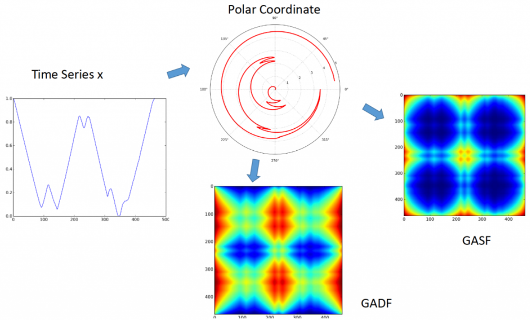


Figure: The process of converting a time series into a GAF. [1]

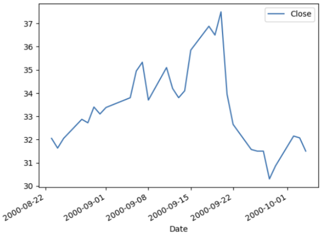
Gramian Angular Fields

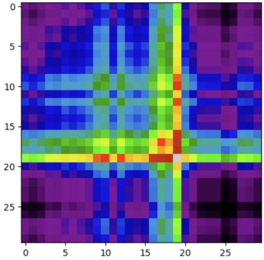
Essentially, we convert our standard time series data from cartesian coordinates to polar coordinates.

A streamlined summary would be:

1. Scale the values to (-1,1) or (0,1).
2. Convert every value to a respective angle using arccos(x)
3. Convert every time point to a radius by partitioning the time series into N subintervals, and representing each timestamp as $(\frac{i}{N})$, where $i \in$
4. Once each cartesian pair has a polar coordinate representation, use the trigonometric sum (or difference) to associate each pair with an inner product to construct the Gram matrix
5. Construct a Gram matrix (Hermitian matrix of inner products) from each point.

This is the time series from August 23rd, 2000 to October 4th, 2000 (30 trading days) of WTI Crude Oil and its respective Gramian Angular Summation Field.





Data

Data is sourced from Yahoo Finance, *yfinance* package. We take the commodity's closing price history from August 23, 2000 - October 15, 2022.

Date	Open	High	Low	Close	Adj Close	Volume
2000-08-23 00:00:00-04:00	31.950001	32.799999	31.950001	32.049999	32.049999	79385
2000-08-24 00:00:00-04:00	31.900000	32.240002	31.400000	31.629999	31.629999	72978
2000-08-25 00:00:00-04:00	31.700001	32.099998	31.320000	32.049999	32.049999	44601
2000-08-28 00:00:00-04:00	32.040001	32.919998	31.860001	32.869999	32.869999	46770
2000-08-29 00:00:00-04:00	32.820000	33.029999	32.560001	32.720001	32.720001	49131
...	...	...	...	...	...	...
2022-10-10 00:00:00-04:00	93.480003	93.639999	90.540001	91.129997	91.129997	289508
2022-10-11 00:00:00-04:00	91.139999	91.349998	87.910004	89.349998	89.349998	333158
2022-10-12 00:00:00-04:00	88.669998	90.070000	86.279999	87.269997	87.269997	332271
2022-10-13 00:00:00-04:00	87.080002	89.660004	85.559998	89.110001	89.110001	298032
2022-10-14 00:00:00-04:00	89.269997	89.730003	85.199997	85.610001	85.610001	237026

5562 rows × 6 columns

Figure: Dataframe for WTI Crude Oil from Yahoo Finance.

Data

A 80/20 split was used, corresponding to the training interval being from 2000-2018 and testing from 2018-2022.

Data

A 80/20 split was used, corresponding to the training interval being from 2000-2018 and testing from 2018-2022.

Training and testing data is split into subsets of 30 days. Every sequence of 30 days increments up one closing price day. Then, every sequence of 30 days is transformed to a GASF.

Data

Next, every GASF is labeled as either *long* or *short*.

Data

Next, every GASF is labeled as either *long* or *short*.

Long if the 30th day < 31st day $\implies$ price increase.

Data

Next, every GASF is labeled as either *long* or *short*.

Long if the 30th day < 31st day $\implies$ price increase.

Short if the 31st day < 30th day $\implies$ price decrease.

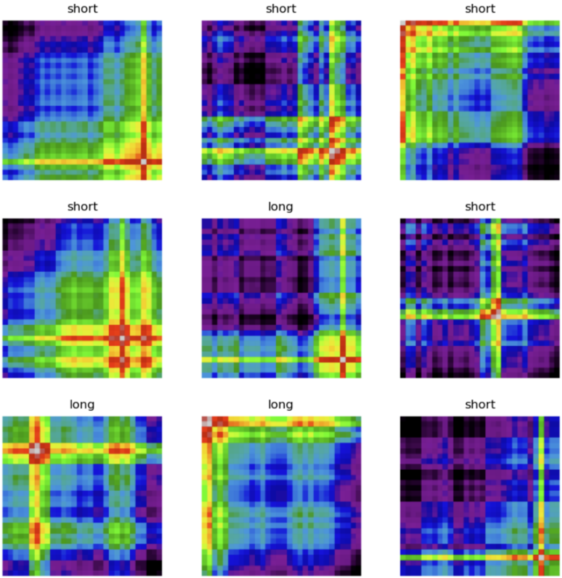


Figure: Batch of labeled GASFs.

Benchmark Test

The benchmarking was done on a simple convolutional network, with one convolutional layer and one dense layer with the relu activation function, Adam optimizer. Training accuracy finished at 99%!

Benchmark Test

The benchmarking was done on a simple convolutional network, with one convolutional layer and one dense layer with the relu activation function, Adam optimizer. Training accuracy finished at 99%!

Testing Classification Report:

	precision	recall	f1-score	support
0	0.53	0.43	0.48	594
1	0.44	0.54	0.48	488
accuracy			0.48	1082
macro avg	0.48	0.48	0.48	1082
weighted avg	0.49	0.48	0.48	1082

Figure: Classification report for benchmark test.

Hyperparameter Tuning

Hyperparameters were optimized, among these, those with the greatest effect on performance included switching to SGD with learning rate of 0.01, widening and deepening the architecture, and using Kaiming He weight initialization (normal or uniform), L1 regularization of 0.01 in the convolutional layers and dropout in the dense, as well as implementing Max Pooling and Batch Normalization.

Hyperparameter Tuning

Hyperparameters were optimized, among these, those with the greatest effect on performance included switching to SGD with learning rate of 0.01, widening and deepening the architecture, and using Kaiming He weight initialization (normal or uniform), L1 regularization of 0.01 in the convolutional layers and dropout in the dense, as well as implementing Max Pooling and Batch Normalization.

Testing Classification Report:

	precision	recall	f1-score	support
0	0.58	0.73	0.65	594
1	0.52	0.36	0.42	488
accuracy			0.56	1082
macro avg	0.55	0.54	0.54	1082
weighted avg	0.55	0.56	0.55	1082

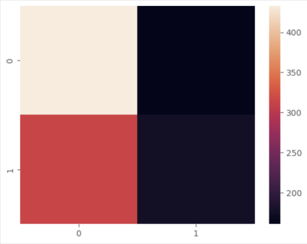


Figure: Final model's classification report and matrix.

Hyperparameter Tuning

The largest challenge when tuning the model was overfitting which was then followed by underfitting. The key struggle was finding a happy medium between the two.

In the end, the final model hovered around accuracy between 52-55 percent. This is a similar result to many in the literature [2].

Conclusions

To conclude, the model was able to classify Gramian Angular Field images into the correct label with with a reasonable degree of error that resembles much of the corresponding literature.

Conclusions

To conclude, the model was able to classify Gramian Angular Field images into the correct label with with a reasonable degree of error that resembles much of the corresponding literature.

However, there were some serious hindrances on this project. Mainly, the small amount of data. Better and more reliable results could be inferred by using a larger dataset. Losing the ability to augment data doesn't help, and perhaps a novel way of data augmentation could be developed (or already exists) that could be used to enrich GAF datasets.

Conclusions

Nevertheless, the use of Gramian Angular Fields is an interesting development that allows us to use neural network's computer vision techniques to make predictions on time series data. This is an exciting and novel way to tackle the perennial problem of forecasting the markets.

Bonus: Skin In The Game

The model predicted Crude Oil's closing price Monday (11/21) would be in the "short" position.

Bonus: Skin In The Game

The model predicted Crude Oil's closing price Monday (11/21) would be in the "short" position.

Let's use this prediction to make an investment decision.

Bonus: Skin In The Game

The model predicted Crude Oil's closing price Monday (11/21) would be in the "short" position.

Let's use this prediction to make an investment decision.

Short position $\implies$ price should be decreasing

Bonus: Skin In The Game

The model predicted Crude Oil's closing price Monday (11/21) would be in the "short" position.

Let's use this prediction to make an investment decision.

Short position $\implies$ price should be decreasing

Previous close: 80.17 USD.

Citations

[1] Oates & Wang: https://arxiv.org/abs/1506.00327

[2] Barra et al.: https://ieeexplore.ieee.org/document/9080613