Context-aware Stock Market Movement Prediction: a MAML and Incremental Learning Approach

Stanford University

Irina A. Marton

Objective

Formulate an algorithm capable of adapting to a diverse range of stock market conditions and predict directional movements in stock prices.

The approach entails meta-training a stacked LSTM model utilizing MAML[4] and following foundational training with incremental learning, systematically applied at predetermined intervals. This model is then evaluated against the regression time series prediction model "Stockbot"[1] and "ContextLSTM" model in terms of ROI (return on investment).

Background Information

Training Dataset & Labelling

The dataset encompasses stock market data from January 1st, 2011, to June 30th, 2023, covering 216 companies across 10 different industries. All data is normalized using the scikit-learn MinMaxScaler and labelled according to the desired trading strategy with the help of the labelling algorithm designed by Wu et al. (2020) [5].

Stockbot Model

An LSTM-based algorithm initially developed by Mohanty et al. (2022) [1], classifies stock data time series into buy, sell, or hold after making price predictions, with the help of a control bot. Slight modifications were made to the control bot in order to incorporate the desired trading strategy for this experiment.

Classification Model: ContextLSTM

With the use of TensorFlow, and keeping the same model architecture of the "Stockbot", the classification model has a "context-aware" input and outputs a buy, sell or hold label. The labels are generated according to a trading strategy of buy or sell when the price has a 2,5% change from the price at time *t* and hold otherwise.

MAML Meta-Trained Model: MetaContext

While ContextLSTM treats all stocks as a single task, the MetaContext model keeps the model architecture of ContextLSTM, but treats each company's stock as a different classification task. Formulated as a 1-shot 3-way classification problem, the input consists of 3 support examples and 1 query from the same stock ticker.

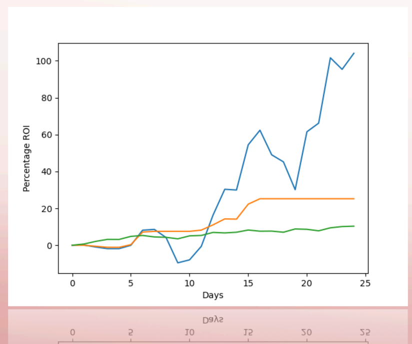
Context-awareness

Inspired by the Few-Shot Classification input, through simple feature engineering, the "context-aware" input was created. It incorporates the first time series of the specified window size, concatenated with its corresponding label, and then concatenated with subsequent time series of the same widow size for which the label needs to be predicted. By changing the input, the classification model has had an increase from 48% to 80% accuracy.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | + | L | + | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |

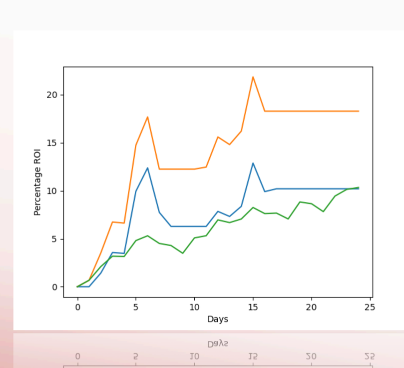
Time-series 1 + Label 1 + Time-series 2

Without context:



48% accuracy

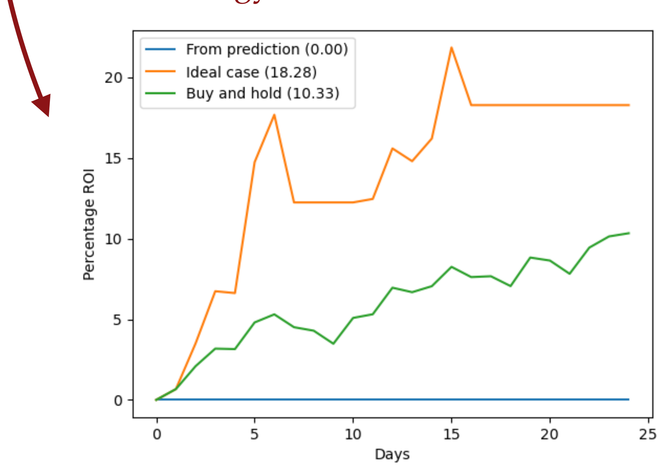
With context:

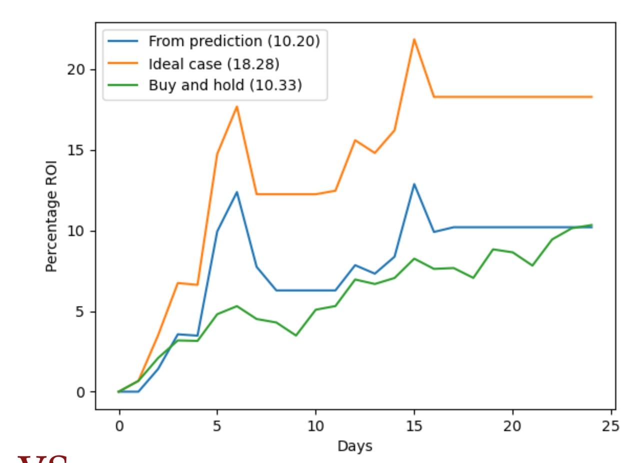


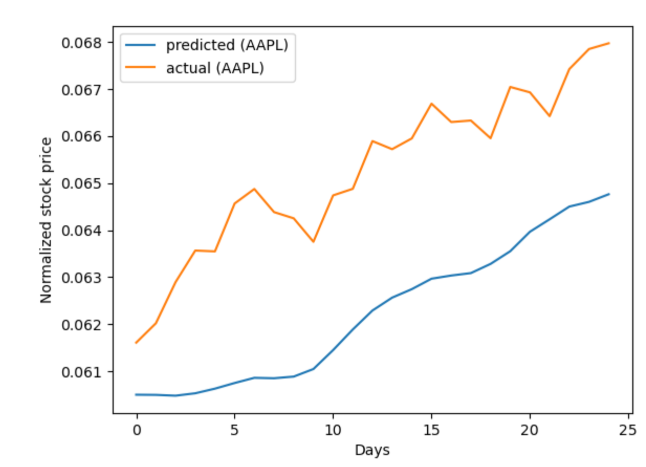
80% accuracy

Stockbot vs. ContextLSTM

After training, both Stockbot and ContextLSTM with inputs containing 21 features each, the models were asked to make predictions for the next 25 days for Apple Inc., a company that was withheld from the training dataset. Even though "Stockbot" was able to generalize and predict the overall trend, its predictions were not accurate enough to trigger any actions apart from hold and it failed to implement the desired trading strategy. In contrast, the classification model achieved an 80% accuracy and was almost successful in surpassing the "Buy and hold" strategy.







VS.

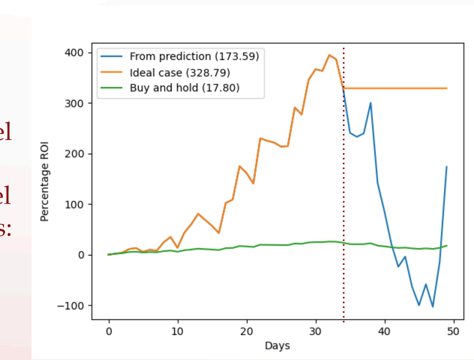
	Precision	Recall	f1-score	Support
0	0.85	0.85	0.85	13
1	0.67	0.67	0.67	6
2	0.83	0.83	0.83	7
Accuracy			0.80	25
Macro avg	0.78	0.78	0.78	25
Weighted avg	0.80	0.80	0.80	25

MetaContext Training Results

Inspired by the Omniglot classification MAML meta-training experiment from [8], the MetaContext model input was constructed in a similar fashion. Two experiments were conducted regarding the task composition. After randomly sampling a task (ticker), in **experiment A**, the query example is then selected at random from the task's dataset. After removing the query example, examples with the labels buy, sell, and hold are **selected at random** from the remaining dataset. In **experiment B**, after randomly selecting the query example, the support examples are selected as the **most recent 3 entries** in the dataset.

$TASK_A$: X_{buy} (Support 1) X_{sell} (Support 2) X_{hold} (Support 3) $X_?$ (Query) $TASK_B$: X_{t-3} (Support 1) X_{t-2} (Support 2) X_{t-1} (Support 3) X_t (Query)

Task A Model & Task B Model predictions:



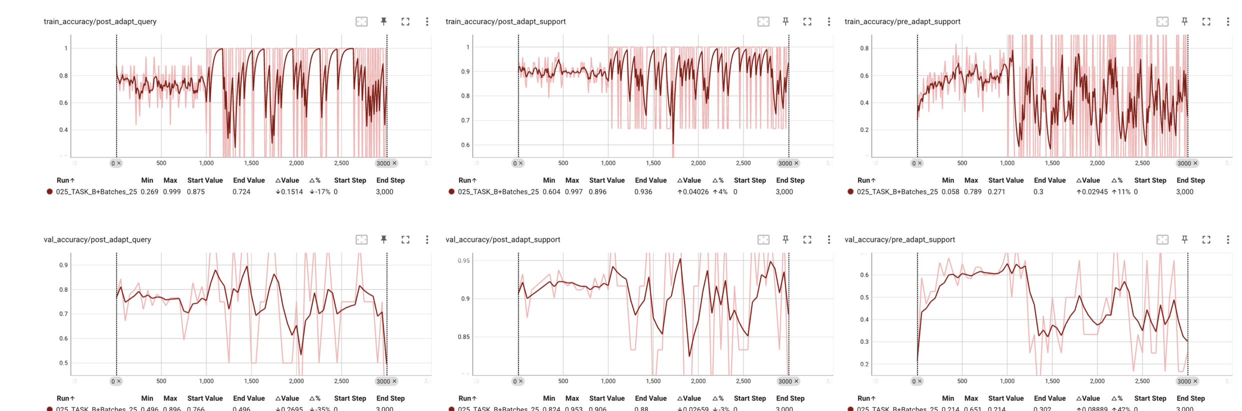
Task A & Task B MetaContext Model Classification Report

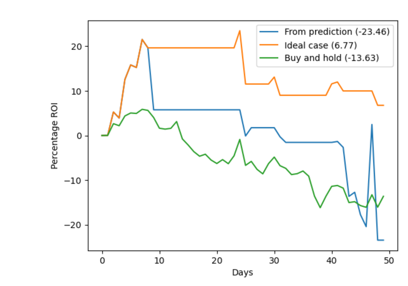
	Precision	Recall	f1-score	Support
0	0.70	1.00	0.82	35
1	0.00	0.00	0.00	15
Accuracy			0.70	50
Macro avg	0.35	0.50	0.41	50
Weighted avg	0.49	0.70	0.58	50

ALL RESULTS ARE PRELIMINARY

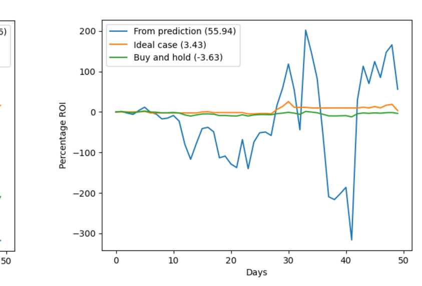
Incremental Learning Results

As time moves forward, the market conditions evolve, the data distribution changes and we start to encounter concept drift. Therefore, in order to counter it, an incremental learning strategy was implemented. The remaining most recent 250 entries represent 1 year of trading days, after observing the initial training results, the decision was made to split the 250 test days from each of the 215 training tickers into batches of 25 as such: 1st batch, entries 1-25; 2nd batch, entries 26-50; …; 10th batch, entries 225-250.

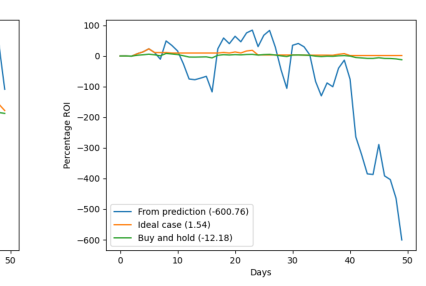




Predictions after batch 1



Predictions after batch 2



Predictions after batch 3

After:	Batch 1	Batch 2	Batch 3
Accuracy	0.78	0.42	0.40
Precision Weighted avg	0.81	0.18	0.16
Recall Weighted avg	0.78	0.42	0.40
F1-score Weighted avg	0.79	0.25	0.23

ALL RESULTS ARE PRELIMINARY

Conclusion & Future work

The proposed algorithm shows potential, however, it is hard to say at this point as it needs extensive hyperparameter optimization and additional modifications to the MAML algorithm as described in "How to train your MAML"[6] in order to stabilize the training process and improve generalization performance.
After initial meta-training the model is able to accurately predict the next 30+ days, however, the incremental learning results obtained so far indicate that online learning might not be sufficient to maintain a high accuracy in predictions.

Future work:

- Hyperparameter tuning for initial training
- Hyperparameter tuning for incremental learning
- MAML algorithm optimisation
- Further experiments with other task constructions
- Further experiments with other "context-aware" input formats

References

[1] Mohanty, S., Vijay, A., & Gopakumar, N. (2022). Stockbot: Using LSTMs to predict stock prices. July 2022.
[2] Zou, J., Zhao, Q., Jiao, Y., Cao, H., Liu, Y., Yan, Q., Abbasnejad, E., Liu, L., & Shi, J. Q. (2022). Stock market prediction via deep learning techniques: A survey. arXiv.
[3] Murphy, N. J. (2019). An Online Learning Algorithm for Technical Trading. Dissertation, University of Cape Town.
[4] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning- Volume 70 (pp. 1126–1135). JMLR.org.
[5] Wu, D., Wang, X., Su, J., Tang, B., & Wu, S. (2020). A Labeling Method for Financial Time Series Prediction Based on Trends. September 2020.
[6] Antoniou, A., Edwards, H., & Storkey, A. (2018). How to train your MAML. arXiv preprint arXiv:1810.09502.
[7] Lemos Neto, A. C., Coelho, R. A., & de Castro, C. L. (2020). An Incremental Learning approach using Long Short-Term Memory Neural Networks. Congresso Brasileiro de Automática, December 2020.
[8] Finn, C. (2023). Homework 2. CS330: Deep Multi-Task and Meta Learning. Stanford University. Retrieved from https://cs330.stanford.edu/materials/CS330_Homework_2.pdf.