ATTENTIVE NEURAL MODELS FOR ALGORITHMIC TRADING

Video Link: https://youtu.be/SmpI-3CjYPw
Will Geoghegan
willgeo@stanford.edu

INTRODUCTION

MOTIVATION:

- Stock market price movements are highly stochastic
- By incorporating **attention mechanisms** we can condition outputs on most relevant information
- We formulate price prediction as a **meta-learning** problem; different assets respond slightly differently to the same conditions

INPUTS/OUTPUTS:

- 20-day sequence of **trade data** and 8-quarter sequence of **income statements**
- Prediction recommending whether to take a long or short position for the asset

APPROACH:

- Baseline model: parallel feed-forward encoding MLPs on each sequence, concatenated into another dense layer, followed by a softmax output
- Experimental model: parallel **Simple Neural Attentive Meta-Learner modules** replace encoding MLPs
- Binary classification weighted by magnitude of sample return (i.e., **weighted cross entropy loss**)
- Compared to market performance using backtested model return

RESULTS:

- Experimental model **outperforms market returns** over withheld test period from 2018-7-2 to 2019-3-11

DATA

SOURCE:

- Sourced from **Quandl's** Core U.S. Equities and Fund Prices dataset

PROCESSING:

- Raw: 14M daily price and volume indicators for 7000 U.S. stocks, 197K quarterly income statements
- Processed: model takes **dual inputs** of 20×23 technical features and 8×210 fundamental features for each date/stock combination

SPLIT:

- **957K** training examples, **20K** validation, **17K** test
- Training set corresponds to data from 2010-8-13 to 2018-7-1; validation/test randomly split from 2018-7-2 to 2019-3-11

METHODS

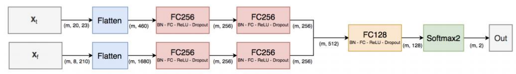

FIGURE 1: The baseline model. Each fully-connected block consists of BatchNorm, FC, ReLU, and Dropout layers.

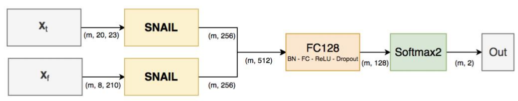

FIGURE 2: The experimental model. Both inputs are passed through a SNAIL module (see Fig 3) before being concatenated with one another.

$$\mathcal{L}_{weighted}(\mathbf{y}, \mathbf{b}, \hat{\mathbf{y}}) = -\frac{1}{m}\sum_{i=1}^{m} | y_i | b_i \log \hat{y}_i$$

EQN 1: Weighted cross entropy loss

$$R(x, y, f) = \frac{1}{S}\sum_{s \in S}\prod_{t \in T} 1 + f(x_{st}, y_{st})$$

EQN 2: Backtested model return

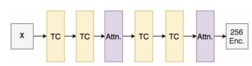

FIG. 3: SNAIL module overview. 1D temporal convolution blocks and causal attention layers. See (1) for details.

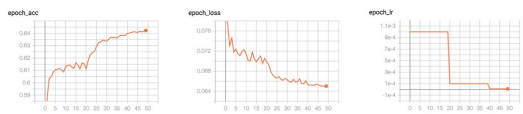

FIGURE 4: Hyperparameter tuning. The effect of decaying the learning rate (right) over 50 training epochs can be seen in the accuracy (left) and loss (center) of the baseline model.

RESULTS

Model	Training Set		Test Set (all)		Test (S&P500)	
	Acc.	Return	Acc.	Return	Acc.	Return
Market	—	183.68%	—	5.53%	—	3.49%
Baseline	**64.22%**	3717.28%	60.44%	19.52%	**55.10%**	**5.78%**
SNAIL	62.50%	**5652.70%**	**62.77%**	**22.67%**	51.75%	5.55%

TABLE 1: Accuracy and return metrics of both models compared to the Dow Jones Industrial Average. Both neural models generalize well to the unseen testing time range.

RESULTS, CONT.

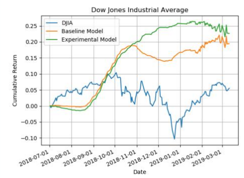

FIG. 5: Test set performance of both models against DJIA (8 month period)

PERFORMANCE:

- Not only does the SNAIL-based model outperform the others on the test set, from Fig. 5 it seems more robust than the baseline to volatile periods like the last few months of 2018.

ANALYSIS:

- Some overfitting but still able to generalize to unseen data
- Likely caused by shared global features like market volatility
- Samples from same day have same global features, highly correlated returns

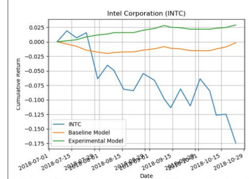

FIG. 6: Performance on INTC over test set (20 samples)

CONCLUSION

SUMMARY:

- Modeling stock price prediction as **meta-learning with an attention mechanism** allows us to outperform market on broad range of U.S. stocks
- Incorporating both technical and fundamental features allows us to learn a robust, general function on these inputs
- **Return-weighted cross entropy loss** allows model to learn effective trading policy

FUTURE WORK:

- Backtesting algorithm is very simple; **weighting capital allocation by confidence** should improve returns
- Efficiency improvements to data pipeline
- Principal component analysis **(PCA)** on fundamental features
- Generalize from prediction at t trading days in the future to a sequence-to-sequence model
- Real-time, **intraday** trading

REFERENCES

(1) Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. Meta-learning with temporal convolutions. CoRR, abs/1707.03141, 2017.

(2) Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, 2017.