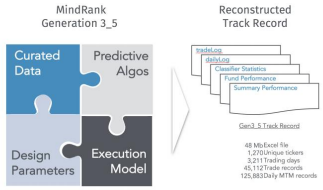
MINDRANK: DL TO SYSTEMATICALLY TRADE A HEDGED PORTFOLIO OF U.S. EQUITIES

CS230
Kevin M. Lalande
klalande@stanford.edu

Background

Santé Capital ("SC") manages a Long/Short equity hedge fund using MindRank, a systematic ML-driven quantitative trading strategy to identify mispriced securities. MindRank is comprised of four components: a) data pipeline, b) predictive algorithms, c) portfolio design parameters, and d) trade execution model. Upgrade SC's predictive algorithms, originally implemented in Matlab circa 2014-2015, to modern DeepNN and RecurrentNN architectures using Python, TensorFlow and Keras.

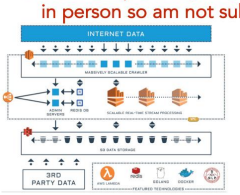


Prediction Problem

Much published research on DL applied to financial markets focuses on predicting 1-day ahead direction of 1-to-few indexes using 10s—100s of easily obtainable features. MindRank attempts a more challenging application: predict 4-6 week directions for 2,500 tickers using a rich set of 4,500 features with fifteen years of historically accurate data. Classify stocks into 5 categories based on predicted price movement. Class 5 (Long) expected to increase by more than a LockGain threshold (+15%) without first decreasing by more than a StopLoss (-10%). Class 2 (Short) is opposite.

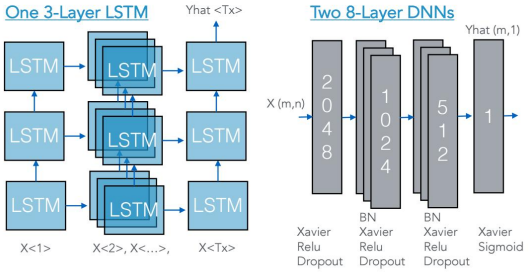
Data Pipeline

SC crawls 10 million URLs to ingest and NLP ~1,000 Gb of unstructured text data each day. This is then combined with structured data to create a dataset containing 4,525 features for each of the ~2,500 publicly traded company on the NYSE and Nasdaq with a market cap >$200M in each of the 155 monthly periods from 200401 to 201612. Features include macroeconomic indicators, fundamental & technical metrics, analyst & investor sentiment, and stock & index price-volume. Financial time series data is de-noised using a discrete Wavelet transform. All other features are normalized for mean and variance. LSTM data is daily with a batch size of 1 and is pre-trained on m=1,000 prior days split 90/10 Train/Dev set. DNN is monthly with a batch size of 128 on m=~2,500 tickers split 90/10. Test is day- or month- ahead predict.



Note: I am planning to attend today's poster session in person so am not submitting a video overview

Predictive Algorithms



Input:	3,200 days, 90-day trailing SP500, R2000, VIX, RC2AR5 X.shape(1,Tx=7,540)	155 YearMonths X.shape (2500 tickers, 4525 features)
Output:	Rolling Class 2 Arrival Rate-5 Yhat ∈ ℝ (0-1.0) predict % of total tickers Class 2	DNN #1: Class 2 or Not Class 2 DNN #2: Class 5 or Not Class 5 Yhat.shape(m,1) ∈ (0,1)
Loss Fn:	MSE	Binary cross entropy
Learning:	RMSprop, mini=1, ep=10	SGD α=0.05, mini=128, epochs=20

Results

Performance is measured by Precision, Recall, F1, Advantage over random guessing, and is benchmarked against existing Gen3.5 algos over 155 months. Real-world impact is evaluated by feeding predictions into a Reconstruct Trackrecord module to calculate financial metrics of IRR, volatility, and Sharpe.

	CS230 Project		MindRank 3_5	
	Class 2	Class 5	Class 2	Class 5
Actual	115,826	120,532	118,718	122,582
Predicted	158,506	159,610	139,532	118,995
True Positive	60,766	57,815	58,613	46,989
Incidence	28.5%	29.6%	28.4%	29.3%
Advantage	34.7%	22.3%	47.9%	34.7%
Train	80.8%	78.6%	na	na
Dev	69.5%	66.5%	na	na
Test	62.3%	59.3%	na	na
IRR	19.3%		23.2%	
Volatility	9.9%		10.9%	
Sharpe Ratio	1.25		1.43	

Discussion

- A fundamental challenge to this type of prediction is that Class prior probabilities fluctuate widely from month to month. E.g.: Class 2 μ = P but σ = 0.7P.
- This makes it difficult for DNNs without temporal architecture to backpropagate useful gradient updates from one period to the next.
- One key idea herein is to train an RNN to predict forward Rolling Class 2 Arrival Rate 5-day Average from daily price/vol info, and then use that to adjust DNN class_weight in each successive month.
- Better performance was achieved by training two separate binary classification DNNs rather than a single multi-task DNN with a 5-way softmax.
- Bayes error is difficult to estimate in this application, but the models appear to train and generalize well. Hyperparameters were tuned empirically with DOE.
- This system has achieved a substantial Advantage over random guessing (22-35%), although it has not yet achieved parity with MindRank's existing algos.