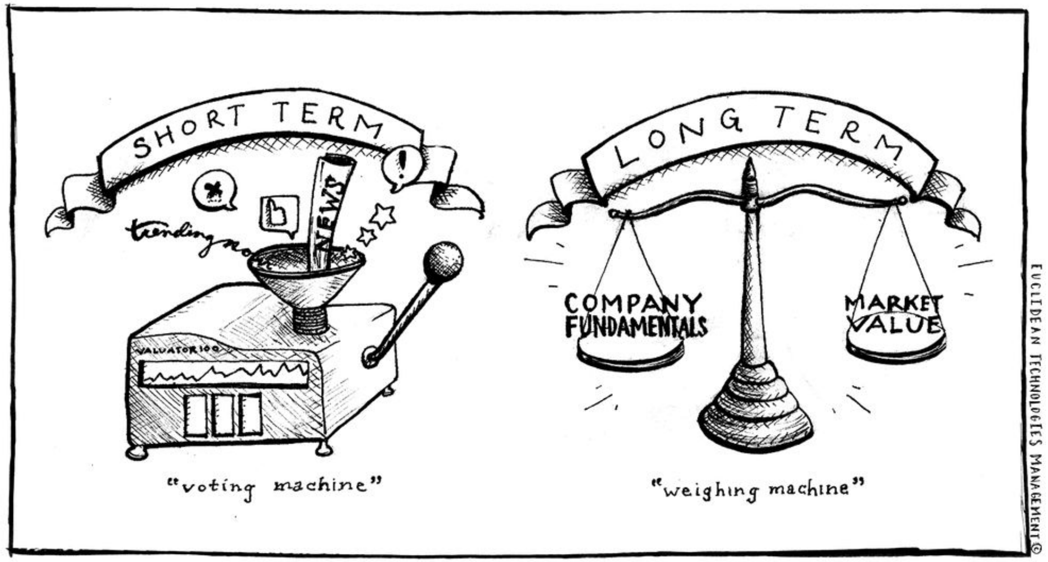
#ml-papers June 2019

Deep Learning and Long Term Investing

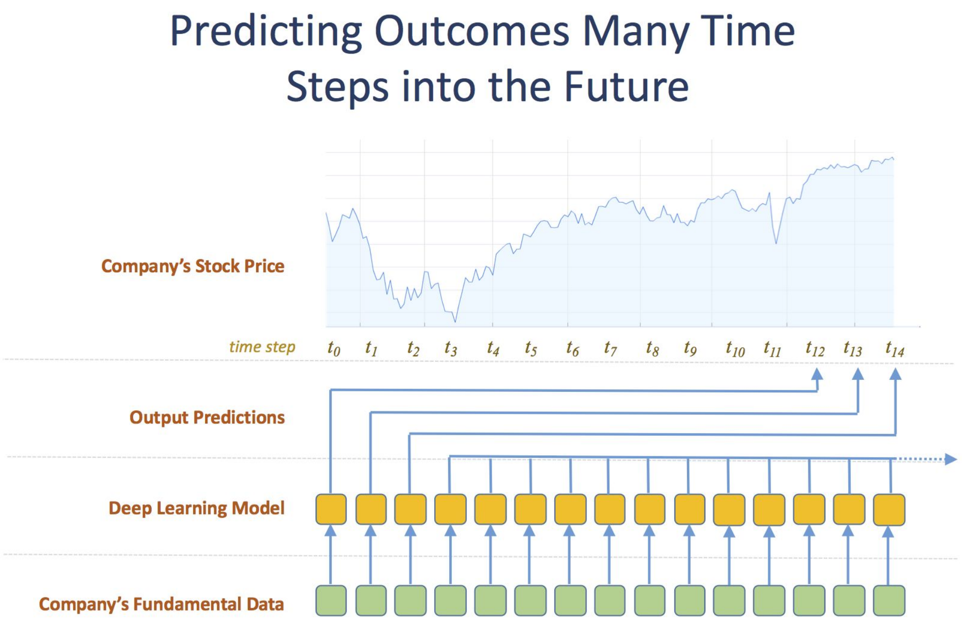
Why DL for long-term investing?

- Stock price should approach fundamental value in the long-term
- Sequenced input data = use RNN
- Automate “factor engineering”
- Lots of text to analyze
  - SEC filings
  - blog posts
  - social media
  - earnings transcripts
- Short-term allows testing with greater frequency, but many exogenous factors
  - Over the long-term, fundamentals play primary role in determining value



Setup

- Predict 1-year horizon
  - Balance between having sufficient data and removing noise from non-fundamental factors
- Monthly predictions
  - *The model is asked, at each time step (month), to make a prediction about what will happen to the stock's price 12 time steps (one year) in the future*
- **Target:** *if the change in price for a stock is greater than the median change in price for all stocks, we assign it an outcome of +1. Otherwise, the outcome is -1*
  - Does not predict degree of outperformance, but this lower bar makes learning easier and can still use it to achieve good performance



The Data Set (from S&P's Compustat db)

- Includes any stocks that have been public for at least 36 months and traded between January 1970 and December 2015 (45 year total span)
- For each month, there are approximately 1,300 to 5,000 companies. The entire dataset represents approximately 10,000 companies.
- Non-US-based companies, companies in the financial sector, and companies with market capitalization below 100 million US dollars (date adjusted) are excluded

Income Statement & Other Items	Balance Sheet
Sales (Revenue)	Cash & Cash Equivalents
Cost of Goods Sold	Receivables
Sales, General, and Admin Expenses	Current Assets
Operating Income After Depreciation	Property Plant & Equipment
Net-Income	Other Assets (Incl. Goodwill)
Capital Expenditures	Total Assets
Dividends	Debt in Current Liabilities
Common Shares Outstanding	Accounts Payable
Price per Share	Current Liabilities
	Long-Term Debt
	Other Liabilities
	Total Liabilities
	Minority Interest
	Preferred Stock
	Shareholders' Equity

Input Features

Momentum

- Calculate the trailing 1-, 3-, 6-, and 9-month stock price change adjusted for splits, then find the percentile ranking, among all companies within the same month

Valuation

- Book to Market = (Shareholder's Equity)/(Market Cap)
- Earnings Yield = the reciprocal of P/E ratio
- Use the respective relative percentile rankings AND the raw values

Input Features (cont.)

Normalized fundamental features

- Normalize data by dividing items by the L2 norm of each fundamental item

Year-over-year changes in fundamental features

- Year-over-year "log" change in value — i.e., log[v(t)/v(t-1)] — for balance sheet and income statement items that do not take on a negative value.
  - We use logarithms here to ensure that outlier changes (very large changes) don't have a disproportionate impact on the factor values

Input Features (cont.)

Missing value indicator features

- Reasons for missing values: an unreported item, a fiscal year change that prevents the creation of a trailing twelve-month sequence, a data collection issue, or a division by zero when a ratio is computed.
- Every input feature has a corresponding binary indicator feature that is equal to 1 if the feature's value is missing for a given company-month; otherwise it is equal to 0
  - If the value is missing but it is not missing in the prior time-step (for the same company), then the prior time step value is pulled forward into the current time-step. However, the missing value indicator field value is still set to 1. If no value can be pulled forward, the missing value is set to zero.

Part 3 - Overview

1) Intro to traditional model for long term investing a.k.a "factor" models

2) Intro to deep learning

3) Similarities & Differences between the 2 techniques

Part 3 - Factor Models

- When back-testing factor models we care about by how much they outperform the market but also about the **statistical significance** of the back-test (measured as a t-score)
- We run into a familiar problem when we're testing > 100 factors.. increased FPR
- They cite this paper which proposes adjusting statistical significance by the # of hypotheses being tested.. we might be able to apply something similar to Wisdom one day?

Part 3 - Factor Model Example

1) sort companies on Book Equity (amount available for distribution to shareholders) / Market Equity (how much investors think the company is worth), split into high, middle & low (30%, 40%, 30%).
    i) Companies in the high group are desirable, low group = "growth" because they are expensive and presumably there's a reason for that.. this seems to correspond with an intuition of liquidity
2) HML (high minus low) is a strategy where you invest in "high" group stocks (value stocks) and short low (growth) stocks, HML returns (called value returns) are used to benchmark other strategies

Part 3 - Deep Learning

- DL 101: function approximation, training loop & convergence.. not too much time spent on this author says to treat it as a black box for now
- Like other supervised ML, but with less feature engineering :) We can dump raw signals into a NN with dropout and let it learn efficient representations.
- Incredibly obvious advantage: **less time spent researching new factors**
- RNNs can model a companies evolution over time and recognize distributional shifts
- target: "probability that the company (represented by the input) will have a total return (price change plus dividends reinvested) greater than the median total return of all stocks"

Part 3 - Factor Models vs Deep Learning

- They can both be used for the same thing
- Authors tie out of sample prediction with multiple comparison bias
- Machine learning is an "automation of the scientific method" (cite)
- No real conclusion made..? It's basically just repeating earlier sections