CS230

Forecasting Financial Performance of Companies For Stock Valuation

Eric Zhang
Department of Computer Science
Stanford University
zhyzhang@stanford.edu

Abstract

Reliable knowledge of the future financial performance of companies is extremely useful in the investment research process and can aid investors in making better investment decisions. In this project, I built recurrent neural network models with gated recurrent units (GRU) to forecast the future financial performance of companies based on information on their historical financial statements. I found the GRU models offered improved forecasting capabilities and significantly outperformed naive forecasting methods on longer time horizons.

1 Introduction

Investing in the stock market is a popular and often profitable activity enjoyed by both amateurs and professionals. Investors make use of a variety of strategies that differ in holding period and riskiness. Among these, one frequently used investment strategy is value investing, where investors buy stocks that are trading for less than what they perceive to be its intrinsic value and make a profit when the stock eventually reaches its fair value. Such valuation is done through fundamental analysis: investors look at a company's financial statements and future growth prospects to determine the correct value for the share. The caveats of such analysis are that it is prone to judgmental errors and involves the challenging task of finding trends within a huge amount of numerical data. To tackle this problem, I built a deep learning model in this project that can analyze historical financial data of public companies and project future financial performance of the company (for a one to three year time horizon). The projected fundamentals can then be used to derive a fair value for the stock, which can help inform our investment decisions. It is important to note that the future financial performance of a company and its stock value are influenced by a wide variety of factors, such as corporate decisions, growth or decline of the particular industry, market sentiment, and other unexpected real-world events. Many of these factors cannot be captured within a deep learning model, so it is extremely difficult for the model to predict a company's future with perfect accuracy. It is likely that a seasoned investor who has a thorough understanding of the market and insights about real-world events will make better investment decisions than a model. However, the advantage of using such a model is that it can quickly give the investor a good gauge of where the company and its stock are heading without having to go through the laborious process of combing through a company's dense financial reports, thus speeding up the investment research process. Furthermore, with the wealth of data within a company's historical financial data, a deep learning model may be able find hidden patterns or indicators of a company's future financial performance that the human eye cannot see.

2 Related Work

Deep learning applied to the stock market is a popular area of research. Most deep learning research in the stock market focuses on using the financial time series to predict short term price changes [1]. On the other hand, research on using financial statement information to predict stock prices is relatively more limited, but still substantial [2]. For example, Nguyen [3] compared the use of gradient boosted regression trees to forecast stock earnings from financial statements with actual predictions from financial analysts and found that although machine learning models are unable to outperform professional analysts, a combination of machine learning and analyst still provide the best overall performance. [4] developed long short-term memory networks (LSTM) and temporal convolution networks (TCNs) to predict future earnings per share based on quarterly financial reporting data and found that both models outperform naive forecasts. Notably, however, most research in this area has aimed to directly predict stock prices or earnings per share and largely ignored other financial fundamentals indicative of company performance. Although such an approach can give a more direct picture of potential investment earnings, Alberg and Lipton [5] found that projecting future financials to derive stock prices has better outcomes than predicting stock prices directly. To my knowledge, the use of deep learning models to predict fundamental data on financial statements, rather than stock prices or earnings per share, has not been explored significantly. This is where this project aims to focus on.

3 Dataset and Features

The data used for this project was taken from the Compustat database through the Wharton Research Data Services (WRDS). Compustat is one of the largest database of financial and market information of companies around the world. For this project, I took the quarterly financial information of all publicly traded US companies within the past 20 years. This gave me a total of 27550 companies. For each company, I split the financial information into seven year intervals: the first four years were used as inputs to the model, and the final three years made up the output labels. Following this, I removed from the dataset entries with missing or invalid information and randomly sorted them into training and testing sets with a 90/10 train/test split. This gave me a training set size of 17291 and test set size of 1921.

I took the following quarterly indicators as input,

Cash and short-term investments; Cost of goods sold; Long term debt; Depreciation and amortization; Earnings per share; Goodwill; Intangible assets; Inventories; Current liabilities; Long-term liabilities; Accounts receivable; Net income; Stockholder equities; Operating income; Property, plant, and equipment; Total revenue; Research and development expenses; Selling, general and administrative expenses; Income taxes

For the outputs, the model forecasts three indicators for the three years after the input time frame – earnings before interests and taxes (EBIT), total revenue, and cost of goods sold (COGS). I decided to focus on forecasting just three indicators since it makes the model more tractable. Although it is ideal for the model to have predictions for every indicator, it is not necessary to do so since there is a lot of redundant information between the indicators. It is entirely possible to derive a stock value using just a few important ones. In particular, EBIT is probably the most important metric on the financial statement: it is frequently used by financial analysts to gauge a company's investment potential and allows us to derive a fair value for the stock using the EBIT/EV multiple.

4 Model

Since we are looking at a continuous period of financial data instead of a single point in time, recurrent neural networks are well-suited for this task. In this project, I tested both LSTM and GRU cells for the recurrent layers using a variety of hyperparameters. I also experimented with adding dense layers at the back of the recurrent models. Through my preliminary testing, the GRU models outperformed the LSTM models in both accuracy and training speed, so I proceeded with GRU for further optimization.

GRU is the newest form of recurrent neural networks that, like LSTM, is able to capture long range connections and tackle the vanishing gradient problem in standard recurrent neural networks with its gating mechanism. However, the GRU has fewer gates than LSTM, resulting in a simpler network with fewer parameters to train but similar and sometimes better performance [6]. The improved performance of GRU over LSTM in this project may be attributed to the fact that there are fewer time steps (only 16), which plays to the strength of GRU's simpler architecture. One modification made in this project to the standard GRU cell is to replace the standard tanh activation units with rectified linear units (ReLU). ReLU is sometimes used over tanh in the GRU because it is fast to execute, can induce sparseness, and may improve performance [7]. It is worth noting that GRU has been rarely used in deep learning research in finance, so this could be a direction of further research in the field.

In my testing, I also found that using three separate models for each output (EBIT, revenue, COGS) gives the most consistent performance. Therefore, I trained models and experimented with different architectures and hyperparameters on each output variable separately. After many rounds of iteration, I found that the use of two recurrent layers followed by two dense layers give the best performance. The model architecture for the GRU network is summarized in figure 1. The input consists of 16 time steps of quarterly financial data (corresponding to four full years), with each time step having the 19 different features listed in the previous section. The output is the three years forecast of EBIT, revenue, and COGS of a given company.

The optimal hyperparameter values which provide the best performance are summarized in table 1. A learning rate of 0.0002 and 200 epochs is needed for reliable convergence. A batch size of 256 maximizes performance while providing good training speed. The two GRU layers each have 128 hidden units while the following two dense layers have 128 units and 64 units respectively. Through my testing, I also found that while having two layers of GRU significantly improved performance from just having one layer, the addition of a third layer only marginally improved performance. Therefore, the two layer model is chosen as it is simpler and can be trained faster.

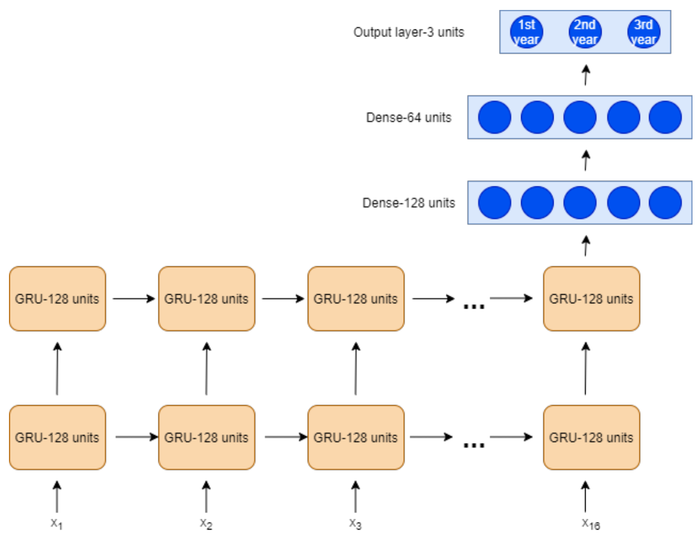

Figure 1: Model architecture

Hyperparameters	Optimal value
Learning rate	0.0002
Number of epochs	200
Batch size	256
Number of hidden units in GRU layers	128
Number of hidden units in first dense layer	128
Number of hidden units in second dense layer	64

Table 1: Optimal hyperparameter values

I chose mean absolute error (MAE), one of the most frequently used evaluation metric in financial forecasting, as the loss function for forecasting EBIT. The alternative is mean squared error (MSE), but using MAE tended to result in more consistent models because of the significant number of outliers in the dataset. MAE is evaluated by

$$MAE = \sum_{i=1}^{m} \frac{|Y_i - F_i|}{m}$$

where Y_i is the actual value and F_i is the predicted value of a given year.

For forecasting revenue and COGS, I used mean absolute percentage error (MAPE) as the loss function. Because revenue and COGS are always positive, using MAPE is appropriate in this situation and ensures that the model does not neglect companies with smaller absolute values. Compared to both MAE and MSE, MAPE resulted in the best overall performance. It is evaluated as follows

$$MAPE = \frac{100}{m} \sum_{i=1}^{m} \frac{|Y_i - F_i|}{Y_i}$$

5 Results and Discussion

Using the architecture described above, training the model was able to result in good convergence. An example of the loss function curve for forecasting EBIT is shown in figure 2.

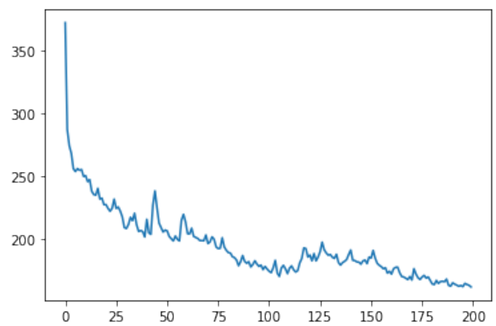

Figure 2: Loss curve for the EBIT model

To evaluate the model, I compared the model's predictions with predictions from a naive forecasting method. The naive forecast assumes that the financial performance of the company always remains the same as the final year of the input time frame. The MAEs of the model's predictions and the naive forecast are shown in table 2.

Model	Dataset	MAE (first year)	MAE (second year)	MAE (third year)
GRU	Train	119	163	195
GRU	Test	118	169	198
Naive	Train	132	195	236
Naive	Test	122	200	242

Table 2: MAE of three-year predictions of GRU model and naive forecast on training and test set

Unsurprisingly, the absolute errors of both the model forecast and naive methods increase as the forecast time horizon increases, indicating that it is more challenging to predict further into the future. However, the GRU model is able to outperform the naive forecast at every forecast horizon. It is also notable that the model displays low variance as the MAE for the model is comparable between the training and test sets.

We can also compute a better metric of forecast accuracy, the mean absolute scaled error (MASE). MASE is a metric developed by Hyndman and Koehler [8] that offers numerous advantages over other simpler metrics like MSE or MAE. It is computed by scaling the out-of-sample MAE by the in-sample one-step naive forecast

$$MASE = \frac{\frac{1}{j}\sum_j |e_j|}{\frac{1}{T-1}\sum_{t=2}^{T} |Y_t - Y_{t-1}|}$$

A value below 1 indicates an improvement over the naive random walk model, while a value above 1 indicates a larger error than the naive model. The MASE of the GRU model is shown in table 3. The results indicate that although the GRU model is only able to provide a slight improvement over the naive model in the first year, this advantage becomes more apparent in the second and third years of forecast. An improvement of around 15 to 20% compared the naive model is consistent with findings by [4].

	First year	Second year	Third year
MASE	0.966	0.845	0.818

Table 3: MASE of GRU model EBIT forecast

Similar results were observed in the forecasting of future revenue and COGS. Note that MAPE is used as the evaluation metric because revenue and COGS is always positive. The performance of the model and the naive forecast are shown in table 4.

Forecast	Model	Dataset	MAPE (first year)	MAPE (second year)	MAPE (third year)
Revenue	GRU	Train	16.7%	25.2%	32.2%
Revenue	GRU	Test	22.5%	27.6%	46.6%
Revenue	Naive	Train	26.0%	58.6%	77.9%
Revenue	Naive	Test	18.8%	37.1%	202.3%
COGS	GRU	Train	20.9%	27.9%	37.9%
COGS	GRU	Test	24.9%	30.0%	34.9%
COGS	Naive	Train	25.4%	39.0%	56.5%
COGS	Naive	Test	20.1%	35.0%	68.0%

Table 4: MAPE of three-year predictions of revenue and COGS by GRU models and naive forecast on training and test set

Again, we see that the GRU models have comparable performance with naive forecast in the first year, but greatly improves in the second and third year. Notably, the naive forecast had very large percentage errors in the third year, due to there being many outliers in the stock market – companies whose revenue rise precipitously, or whose revenue plummet. Interestingly, deep learning models are very good at capturing such outliers, which explains why large percentage errors are not seen for the GRU models. This phenomenon has also been observed in [3], where the models are able to vastly outperform analysts specifically on outlier stocks. These percentage errors are also consistent with [5], who reported forecast errors of 20 to 30% in their models.

6 Conclusion and Future Work

I developed recurrent neural network models using GRU cells to forecast future financial performance of companies. The models displayed significant improvement over naive forecast methods, especially in the two and three year forecast horizons. This project improves on the existing literature by focusing on predicting financial fundamentals rather than predicting stock prices directly. The current findings indicate that my model's performance is consistent with other models in the literature. Given the noisy nature of the stock market, using deep learning models to project fundamentals before deriving the stock prices may lead to more reliable outcomes and is an attractive area for future research. With more time, I would have liked to simulate my model on making actual investment decisions and comparing returns with existing models that directly predict stock earnings. Other future work include having more output features to give a more comprehensive set of indicators for financial performance. This project only used historical financial reporting data as input, which by itself is unable to capture many factors that affect future stock prices. In the future, we could incorporate other forms of data like financial time series or sentiment analysis into the model. Overall, the significant advantage the model provides over naive methods is a promising sign that there is actual valuable information contained within dense financial reporting data that deep learning models can utilize. The development of such models cannot replace investor experience and insight, but it can go a long way to make investing easier.

References

[1] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, Feb. 2020.

[2] C. Stoean, W. Paja, R. Stoean, and A. Sandita, "Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations," *PLOS ONE*, vol. 14, no. 10, Oct. 2019.

[3] J. Nguyen, "Comparing Machine Learning and Analyst's Predictions for Earnings," *Wharton Research Scholars*, vol. 189, May 2020.

[4] L. Eland, S. A. Tideman, K. Lopatta, and O. Kramer, "Earnings Prediction with Deep Learning," 2020. [Online]. Available: arXiv:q-fin/2006.03132

[5] J. Alberg and Z. C. Lipton, "Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals," 2018. [Online]. Available: arXiv:stat/1711.04837

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning*, Dec. 2014.

[7] Y. Jin, Z. Jia, P. Wang, Z. Sun, K. Wen, and J. Wang, "Quantitative Assessment on Truck-Related Road Risk for the Safety Control via Truck Flow Estimation of Various Types," *IEEE Access*, vol. 7, pp. 88799–88810, Jun. 2019.

[8] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, Nov. 2006.