Machine Learning Engineer Nanodegree

Capstone Proposal

Gil Akos
November 4th, 2016

Proposal // Financial Transaction Predictor

Domain Background

Managing your personal finances is a time-consuming and stress-inducing activity. With the rise of services and apps that mark the trend towards the unbundling of banks[1], the "complicated-ness" of personal finance has only increased and our ability as individuals to optimally manage our expanding number of accounts has only become more challenging. From a systemic viewpoint the complexity of banking is also significant - more than $14 trillion flows through the banking system in the United States every day[2]. Furthermore, the transactions that make up these flows are often inconsistently formatted or minimally labeled. Even large fintech aggregator apps do a poor job of munging, cleaning, and categorizing transaction data. Here are a few hilarious examples from my own Mint account (which I do manage with some regularity):

```
$9.99 PMG CLE AIRPRT 10/25 #000322122 PURCHASE 18930 BROOKPARK R CLEVELAND OH
```

becomes **PMG CLE AIRPRT** categorized as **Doctor**

I hope I didn't go to a discount Doctor at the Cleveland Airprt (mis-spelling intentional).

```
$29.05 BOGARTS SMOKEHOUSE 04/10
```

becomes **BOGARTS** categorized as **Personal Care**

I tried hard, but am having a difficult time rationalizing how delicious St. Louis Style Bar-B-Que helps me with my personal care.

As we can see in the two examples above, we would receive at a minimum four data points per transaction - Amount, Date, Category, and a Textual Description. Implictly we also have which account the transaction processes through, as well as day of the week, day of the month, and day of the year. Building a model using deep learning with the capability of predicting categorization and future cash flow would have value in that it would help us optimize the work our money does for us while reducing the stress of having to do so directly and explicitly[3]. The transaction data is key - but how many dimensions do we need, and can we infer labels i.e. categories when the data is lacking detail (as we can directly see in the second example above)?

Across the fintech industry, there are some relatively popular players in the aggregation, budgeting, and forecasting game. Mint[4] pulls data from as many different accounts that you connect and tries to clean and categorize the transactions (with relative success - see above), Level Money[5] offers up projections of your cash flow and spending plotted across the month (overly coarse in detail in my experience), and Plaid[6] offers an API that formats transaction data into digestible formats, even predicting some labels with a paired confidence value to indicate reliability (label filling only works for some transaction types). Outside of the industry activity around this problem, there are also plenty of articles on using Deep Learning using Long Short-Term Memory models for creating predictions with financial time series data.

I am particularly passionate about this problem becuase not only am I frustrated by the lack of smarts in the tools I use to manage my finances, but also this is a key ingredient to the next generation of technology we are building for my startup, Astra. Creating models that can accurately predict a missing key label given the one or two of the others would be beneficial to our progress as well as the broader machine learning field that seeks to use deep learning for financial predictions.

Problem Statement

This capstone project seeks to predict labels for an incompletely labeled financial transaction within a time series. Our time series will be composed of financial transaction data with labels that include amount, date, category, and a textual description, plus the implied labels mentioned above. Our model will seek to predict missing labels, given one or more labels with a comparative study of performance for which labels are the strongest predictors and which label is subsequently most accurately predicted. Similar studies have been developed using a combination of Restricted Boltzman Machines and Multi-Layer Perceptrons[7] for stock value predictions demonstrating error improvement compared to other machine (non-deep) learning

models and using Deep Recurrent Neural Networks to predict energy loads, both of which suggest Deep Learning techniques offer a more robust learning model than other techniques. Furthermore, a market predictor has been developed using TensorFlow and documented in detail by the Google Cloud Platform team[9] that can act as a technical and analytical bench mark.

Datasets and Inputs

For preliminary training and testing of the deep learning model used for this capstone, I will use my own financial transactions from the past five years as downloaded from my Mint account. For submission of the project, I will include an anonymized version. The format for the each transaction below with italic items representing the most important labels. During the development of the project, using more of these labels may prove advantageous.

- *Date (MM/DD/YYYY)*
- *Description (Clean Textual Description)*
- Original Description (Dense Raw Textual Description)
- *Amount ($)*
- Transaction Type (Credit/Debit)
- *Category*
- Account Name

After developing the initial model with my personal dataset, my ambition is to include functionality that will allow anyone with a Mint account to execute the learning program and observe the results.

Solution Statement

Using a Keras[10]implementation of the TensorFlow[11] library, the solution to this project will be a Recurrant Neural Net with a Long Short-Term Memory model capable of learning from time series data. In order to include the Description and Category labels, they will need to be converted into vectors[12] and merged[13] with the Amount values. The measures of performance will mean squared difference for both predicted Amount values as well as vectorized Categories which will give an indication of proximity to the correct categorization.

Benchmark Model

This project will use three models as benchmarks for the various elements as a part of a Deep Learning approach to the problem - Deep Learning for Multivariate Financial Time Series[7] for an

overall error rate comparison, Deep Learning for Multivariate Financial Time Series[8] for the improvement approach compared to other machine learning model types, and Machine Learning with Financial Time Series Data[9] for a technical approach. All use Deep Learning for Financial Time Series applications.

Evaluation Metrics

For preliminary testing before creating the composite model of all labels, I will establish some base line values using Support Vector Machines using two labels at a time, building the combinations of parameters incrementally. The resulting values will provide benchmarks for Recurrant Neural Networks that composite the same label combinations. This will validate the approach and give an indication at how well each of the labels contributes to a predictive model. Also, as an overall benchmark for the final composite model, the error values documented of approximately 40% from the Multivariate Financial Time Series study[7] will give an indication as to whether the system as applied to personal finance is valuable in comparison to being applied to the stock market.

Project Design

This project will be implemented through the TensorFlow library, first using Support Vector Machines to identify classification accuracy benchmarks, then using Recurrant Neural Networks to incrementally build the complexity of a Long Short-Term Memory model to accommodate the Time Series characteristic of our financial transaction data.

1. Prepare Data Set

- Confirm correct labels on Mint
- Download CSV
- Anonymize personal data
- Define a function to convert labels to vectors

2. Develop Bench Marks Study

- Load the dataset from CSV file
- Run TensorFlow Support Vector Machine combination studies (all combinations of 2 and 3 of 4 labels) using a train, test, validate ratio of 70, 15, 15
- Document and Visualize Results

3. Develop Deep Learning Study

- Load the dataset
- Run TensorFlow Recurrant Neural Net (Long Short-Term Memory) combination studies (all combinations of 2 and 3 of 4 labels) using a train, test, validate ratio of 70, 15, 15
- Document and visualize results

4. Develop Merged Deep Learning Study

- Load the dataset
- Run TensorFlow Merged Recurrant Neural Net (Long Short-Term Memory) combination studies (all combinations of 2 and 3 of 4 labels) using a train, test, validate ratio of 70, 15, 15
- Document and visualize results

5. Develop Merged Deep Learning Study

- Analyze results of above three studies
- Develop detailed visualizations of comparison between bench marks, studies, and final results
- Develop final report

6. Bonus - Develop Generic Mint Onboarding

- Experiment with Mint API[14] for any Mint user to access the functionality
- Develop simple function for querying predictions based on one or more labels

[Footnotes]

1. TechCrumnch // What's next for personal financial services?

   > If 2015 was the year of the great "Bank Unbundling," with new companies dissecting the consumer banking experience to offer specialized services, it was also a year that saw the emergence of a new landscape of financial influencers taking a seat at the table.

2. Federal Reserve Bank of New York // Intraday Liquidity Flows

> On a typical day, more than $14 trillion of dollar-denominated payments is routed through the banking system.

3. [TechCrunch // AI can make your money work for you] (https://techcrunch.com/2016/09/08/ai-can-make-your-money-work-for-you/)

> Did you know that extra cash in your checking account is a missed opportunity? Every day, it loses value to inflation. To generate better returns, you could keep the bare minimum in your checking account and invest the rest. However, unexpected expenses can drain your account suddenly. Without extra cushioning in your checking account, you risk getting slapped with bank fees or credit card debt that quickly cancel out any gains from your investments. It feels like you can't win. Either you're missing out on capital gains, or you're playing limbo with your account balance. AI will make this struggle a thing of the past. Advances in AI will create a robo-accountant that knows your spending better than you do. By analyzing your purchase history, it will constantly move money between your checking, savings, investments and credit cards. This way, your checking account's balance is always in the narrow "sweet spot:" high enough to avoid fees, but not so high that you miss out on investment yield.

4. Mint
5. Level Money
6. Plaid
7. Deep Learning for Multivariate Financial Time Series
8. Deep Learning for Multivariate Financial Time Series
9. Machine Learning with Financial Time Series Data
10. Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras
11. TensorFlow Tutorial for Time Series Prediction
12. Plaid Categories - See ID Field
13. Two merged LSTM encoders for classification over two parallel sequences
14. Unofficial Mint API