Application of Deep Learning to Credit Risk Modeling

Guanting Chen
ICME
Stanford University
guanting@stanford.edu

Shiqi Yang
Department of Management Science and Engineering
Stanford University
sqyang@stanford.edu

Abstract

Deep Learning has become a robust machine learning tool in recent years, and models based on deep learning has been applied to various fields. In our project, we applied RNN (recurrent neural network) to the prediction of small business loans from 1990 to 2014. Our goal is to model the credit risk based on intrinsic features of the loan and indicators of the macroeconomic behavior. Here we train and compare LSTM (long short term memory) and GRU (gated recurrent unit) models on several sets of loan data from different periods. We used ROC curve and AUC as the metric to tune hyper-parameters and to measure the performance of our prediction. It turns out that our model performs well in predicting short-term risk(2 years).

1 Introduction

Small Business are backbones of the US economy. According to the report from SBA (*Small Business Administration*), there are over 22 million small business in the US, defined as generally less than 500 employers for most manufacturing industries and annual receipts around $7.5 million for many non-manufacturing industries. However, the credit risk of small business loans is not well measured. Most internal data of small business are not disclosed to be used to model credit risk, but risk assessment is essential to make a particular investment succeed and maintain a safe management system. For example, banks need to assess risks to underwrite loans; investors try to minimize risk while expecting high return; regulators wish to understand the behavior of small business during economic cycles. Given the importance of small business to the wide economy, we aim to predict the loan status (survival or default) based on risk factors, and identify if some exogenous factors (such as unemployment rate, GDP growth and so on) have significant impact on small business.

The input to our algorithm is a set of intrinsic features of loan and indicators of macroeconomic behavior. The dataset consists of categorical data, numeric data and timestamps. We then use LSTM and GRU models to output a predicted loan status. We label default as 1 and survival as 0.

2 Related work

There is a substantial empirical literature on credit delinquency risk. In early work, von Furstenberg (1969) establishes the influence on home mortgage default rates of variables such as income, loan age, and loan-to-value ratio. Gau (1978), Vandell (1978), Webb (1982), and others examine additional variables. Commercial mortgage default is studied by Titman & Torous (1989) and Vandell (1992). However, the approach they used is based on probabilistic and econometric modeling(like a Hazard

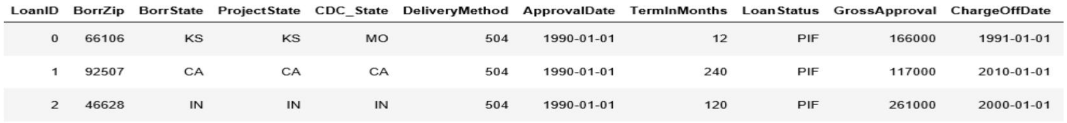
CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

rate model or Cox model). They have to specify a model first and then do the market calibration to fit the model, but our deep learning model have the capacity of catching these behaviors automatically. More deep learning related work has been done by several other users, Brown, Goetzmann & Kumar (1998) use neural networks to predict stock markets. Swanson & White (1997) propose the use of neural networks for macroeconomic forecasting, and Elliott & Timmermann (2008) discuss other applications in economic forecasting. More recent applications of deep learning in financial economics include Sirignano (2016), who use DNN to model transition probability of mortgage risk. To our knowledge, there are few published empirical results in the application of RNN in financial economics. And in this project we are exploring several RNN model to forecast credit default event.

3 Dataset and Features

3.1 Raw Dataset and Features

We obtained the original dataset with idiosyncratic features of small business loan from SBA. The intrinsic features include: *Charge-off Date, Gross Approval Amount, Term of the Loan, Zip code/ State of the borrower/project/CDC, Delivery Method, Business Type, Industry Type/Naics code*. We also collected data which could explain macroeconomic behavior. The exogenous features include: *monthly S&P500 Index, monthly 10 year TED spread, monthly State Unemployment Rate, monthly interest rate*.The time frame for the data set ranges from 1/1/1990 to 1/31/2014.



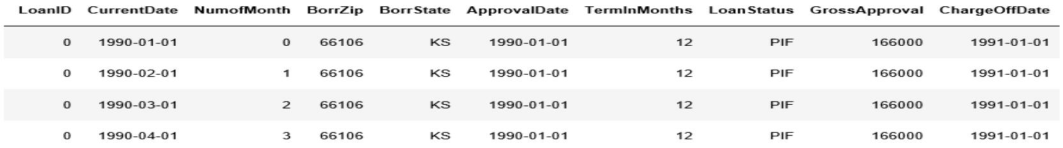
LoanID	BorrZip	BorrState	ProjectState	CDC_State	DeliveryMethod	ApprovalDate	TermInMonths	LoanStatus	GrossApproval	ChargeOffDate
0	66106	KS	KS	MO	504	1990-01-01	12	PIF	166000	1991-01-01
1	92507	CA	CA	CA	504	1990-01-01	240	PIF	117000	2010-01-01
2	46628	IN	IN	IN	504	1990-01-01	120	PIF	261000	2000-01-01

Figure 1: Original Dataset

3.2 Data Preprocessing

We first merged macro economic data with loan data by corresponding months and states. We removed states such as Puerto Rico and Virgin Islands because there is not enough economic information about the unincorporated U.S. territory. We then encoded all categorical features into one-hot to better fit our model, labeled each loan with a unique Loan ID, and normalized all numeric variables. We will discuss about feature selections in later section.

The original data only specified the starting date and the loan term. To better capture the time series effect, we discretized our time-series data into quarters (instead of months since we have many long-term loans). We would like to estimate the probability of default as time goes on. After processing the data, the dataset becomes :



LoanID	CurrentDate	NumofMonth	BorrZip	BorrState	ApprovalDate	TermInMonths	LoanStatus	GrossApproval	ChargeOffDate
0	1990-01-01	0	66106	KS	1990-01-01	12	PIF	166000	1991-01-01
0	1990-02-01	1	66106	KS	1990-01-01	12	PIF	166000	1991-01-01
0	1990-03-01	2	66106	KS	1990-01-01	12	PIF	166000	1991-01-01
0	1990-04-01	3	66106	KS	1990-01-01	12	PIF	166000	1991-01-01

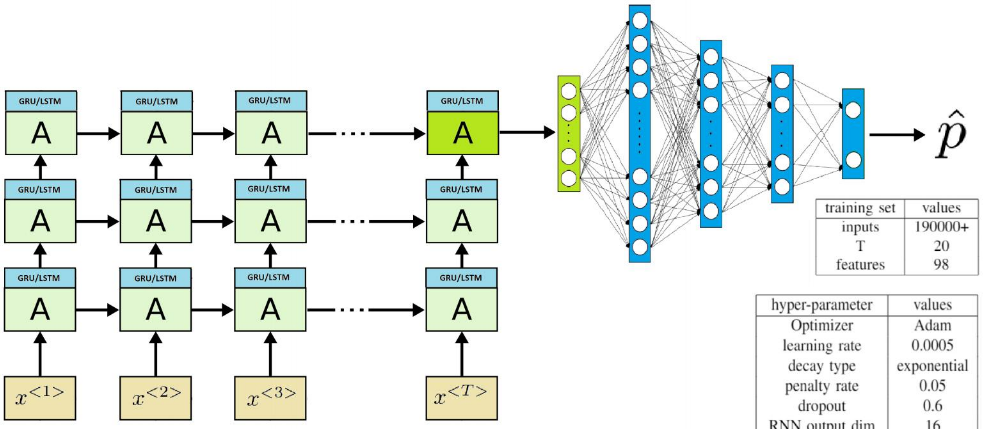
Figure 2: Processed Dataset **features are not all displayed*

3.3 Training/Validation/Test sets

To verify the robustness of our approach given a certain period, we fit our model in two nearby periods. One training set uses data from 01/01/2002 to 10/01/2007 to predict default in 2009. Another training set uses data from 01/01/2003 to 10/01/2008 to predict default in 2010. We take 5% of the training data to be our dev set. To avoid look ahead bias, we shift the data two years later as our test sets. So the dimension of our data is of (m, T, n), where m is the number of loans appeared in the corresponding time period, and in our training set it is approximately $190000+$. T stands for the length of time, here we have 4 quarters per year so $T = 20$. Because some of the loans won't have

T periods, we also have to do the paddings here and mask the padded value in training. Lastly, we have to fix n, since our training set and test set data come from different time periods, sometimes the feature dimension don't match, we have to manually delete some categories out.

4 Methods



training set	values
inputs	190000+
T	20
features	98

hyper-parameter	values
Optimizer	Adam
learning rate	0.0005
decay type	exponential
penalty rate	0.05
dropout	0.6
RNN output dim	16
dense layer 1 dim	16
dense layer 2 dim	8

We used a RNN model combined with a DNN model, and we define our cost in a way such that the final output can be interpreted as a reasonable default probability. One thing to notice is that, sometimes when the number of default is very small, say 0.1%, if we use our default cross entropy loss, the algorithm will tend to predict all loans as surviving, so we have to add weight to our False Negative(predict survival, actually defaulted) predictions. Therefore, we define our cost to be

$$\text{Cost} = \sum_{i=1}^{m} \omega y^i \log \hat{y}^i + (1 - y^i) \log (1 - \hat{y}^i) + \frac{1}{2}\lambda \sum_{\text{all weights}} ||W||_2^2$$

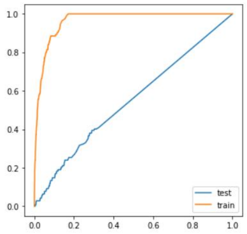
5 Experiments/Results/Discussion

5.1 Metric

We use ROC curve and AUC to measure the performance of our model. Usually for forecasting the future, a AUC around 0.8 is considered to be a good one. We did not use accuracy because there was only a small portion of loans at default. Accuracy could still be very high if we predict all loans to be surviving.

5.2 Baseline model performance

We firstly trained a baseline model using raw data(without discretization, without time series data such as SP500, 10-year interest rate etc.). The original training set only contains around 50,000 samples, we divided it into two parts: 80% training set and 20% validation set. It turns out in the training set the model is doing well, however, it lacks the ability to predict future.



Left is the performance of our naive baseline model. On the training set the AUC is 0.95, while on the test set the AUC is so low that it completely lack the power of predicting future. This justify our method to discretize our data and adding time dependent features in order to move to a RNN model.

5.3 RNN model performance

We train our model in period I (from 01/01/2002 to 10/01/2007)) and II(from 01/01/2003 to 10/01/2008), and our RNN model can achieve much better performance.

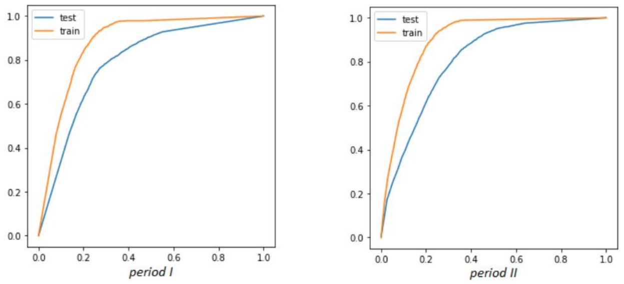
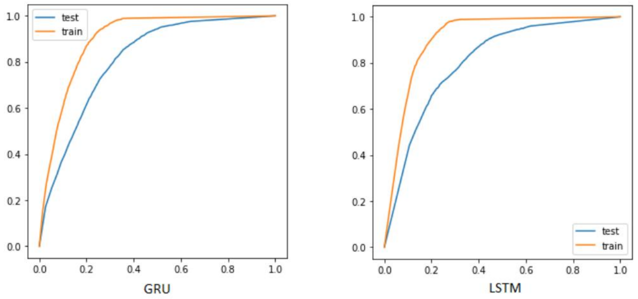


Figure 3: period I test AUC = 0.76, period II test AUC = 0.75

As we can see from the plot, our method is pretty robust in terms of training set performance and test set performance, and it is achieving a good AUC in test set. We also compared a 3-layer GRU model with a 3-layer LSTM model, and it turns out that GRU model have slightly better performance than the LSTM model, below are some comparison between the performance of these two models.



Comparison on the cost function for GRU and LSTM model, notice that GRU performs better in the test set.

Period	Train		Dev		Test	
	LSTM	GRU	LSTM	GRU	LSTM	GRU
I	0.606	0.732	0.581	0.693	0.926	0.903
II	0.681	0.746	0.702	0.755	1.002	0.948

5.4 Hyper-parameter selection

Hyper-parameter tuning is a time-consuming process for our project. It turns out that the most important parameter is the regularization coefficient λ. We can see in figure 4 that a good selection of λ could boost model performance in terms of AUC from 0.6 to 0.76.

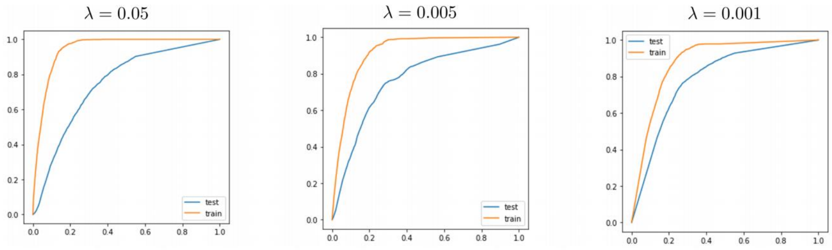
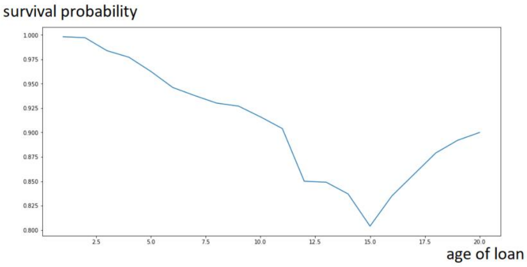


Figure 4: The effect of regularization parameter

Besides the learning rate, the weight w we added to the cost function also matters, from a grid search we find $w = 50$ will yield the desirable result. We choose our learning rate $\alpha = 0.005$ with exponential learning rate decay. For this cost function, learning is not hard and the Adam optimization algorithm will almost always decrease the cost function as iteration goes on, so choosing learning rate can be a manual process: if it is too slow then scale it by 5 or 10 and vice versa. For other hyper-parameters, please refer to the table in section 4.
In terms of feature selection, we performed the cross-validation method(leave on out), we measure the performance based on the ROC and AUC, and it turns out that the important variables are gross approval amount, loan Age and business type, while the unimportant features are SP500 and TED spread.

5.5 Discussion

Considering the economic structure will change overtime, and our test set comes from the data 2 years later, it is impossible to completely reduce variance. So in terms of regularization effects we are doing a fair job. Another interesting fact our model learned is the "term structure". If we analyze the output sequence of our RNN (recall that we are training the model based only on the last state instead of the whole sequence), we will discover the following graph.



This is the output sequence of our RNN model, if we interpret it as the probability of survival during each time step, we will find that it shows a "term structure": as time increases the probability of surviving decreases, and it bounces back near the end of the loan. It is amazing that our RNN learned this automatically without deliberately modeling this behavior in our cost function.

6 Conclusion/Future Work

Our model turns out to perform well in predicting short-term risks. Start with a baseline DNN model and a bad outcome of test set, we could justify that, by discretizing time into quarters, RNN model is a better choice for our loan data with time sequence effect. Among two RNN models we trained, GRU performs slightly better than LSTM, and 3-layers architecture is better than 1-layer through our experiments. We also notice that tuning hyper-parameters, such as regularization coefficient λ, could significantly increase AUC value. We think the reason GRU works better than LSTM is that LSTM contains more hyper-parameters than GRU and training them become harder.

Although we got good ROC curve for predicting the future risk event. Our method have some drawbacks. For example, it performs poorly in terms of predicting the future in the sample period from 1990 to 2000, where there are much fewer default events happening (imbalanced data). It is possible that another RNN model with a finer tuning with the weight can handle this problem . Also, our model fails to predict the future risks more than 2 years. One way to model the prediction of default in the future is that, we can fit a autoregressive model to the time dependent variables (SP500, interest rate and TED spreads), then we can feed the simulated time dependent variable into our RNN to get a prediction for longer period. Also, in a risk management perspective, We can try to model the loss a pool-level loans based on the output of our RNN. If the probability is realistic, we can use inverse sampling to simulate the event of default, and therefore generate some risk measures, which we can compare the result with some traditional probability method of risk modeling.

7 Contributions

Both team members contribute to the project. To be more specific:

Guanting Chen: Build and compare the RNN model, process the data, tune the hyper-parameter, and wrote the report

Shiqi Yang: Built baseline model, discretized quarterly data, helped train RNN models, and wrote reports.

Link to code

https://drive.google.com/open?id=1EOUmlbA8tnE-rewJRKW_p-HhTSYTe9K_

References

[1] Sirignano, Justin. & Apaar Sadhwani. & Kay Giesecke. (2016). Deep learning for mortgage risk.

[2] von Furstenberg, G. (1969), ‘Default risk on fha-insured home mortgages as a function of the terms of financing: A quantitative analysis’, *Journal of Finance 24*, 459–477

[3] Gau, G. (1978), ‘A taxonomic model for the risk-rating of residential mortgages’, *The Journal of Business 51(4)*, 687–706

[4] Vandell, K. (1978), ‘Default risk under alternative mortgage instruments’, *The Journal of Finance 33(5)*, 1279–1296.

[5] Webb, Bruce G. (1982), ‘Borrower risk under alternative mortgage instruments’, *The Journal of Finance 37(1)*, 169–183.

[6] Titman, S. & W. Torous. (1989), ‘Valuing commercial mortgages: An empirical investigation of the contingent-claims approach to pricing risky debt’, *Journal of Finance 44(2)*, 345– 373.

[7] Vandell, K. (1992), ‘Predicting commercial mortgage foreclosure experience’,*Journal of the American Real Estate and Urban Economics Association 20* , 55–88.

[8] Brown, S. & W. Goetzmann & A. Kumar (1998), ‘The dow theory: William peter hamilton’s track record reconsidered’, *The Journal of Finance 53(4)*, 1311–1333.

[9] Swanson, N. & H. White (1997), ‘A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks’, *The Review of Economics and Statistics 79(4)*, 540–550.

[10] Elliott, G. & A. Timmermann (2008), ‘Economic forecasting’, *Journal of Economic Literature 46(1)*, 3–56.

[11] Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. " O’Reilly Media, Inc.".