NLP for Credit Default Probability Prediction

By LSTM Neural Network

Niyu Jia

Sponsored by Bank of China

Contact Information:
Questrom School of Business
Boston University
595 Commonwealth Ave, Boston, MA

Phone: +1 (617) 407 0802
Email: nyjia@bu.edu

Abstract

Credit default probability prediction is important for financial institutions to do their risk-return trade-off. In this project, an innovative way to detect credit risk is proposed. Company senior mangers' resumes become the key information to do company credit risk analysis. With the mature structure of Natural Language Processing model, it can be interpreted as a multi-classification model with word embedding inputs and credit scoring rank outputs. The neural network achieved lowest cross-entropy loss of 0.732 in training data set and lowest cross-entropy loss of 0.974 in validation set after taking over-fitting into consideration.

Data Exploration

Playing with data is always the first step. We gathered over 25,000 samples with up to 10 resumes information in each sample. Besides, the Distance to Default (DD) of each company is calculated by KMV method. These DDs are approximately normally distributed centered at DD=2.16 after removing the empty and extreme values.

As for the resumes, we combined the sentences and then split them into words,merging them into a list. Now the basic structure of out input has been built.

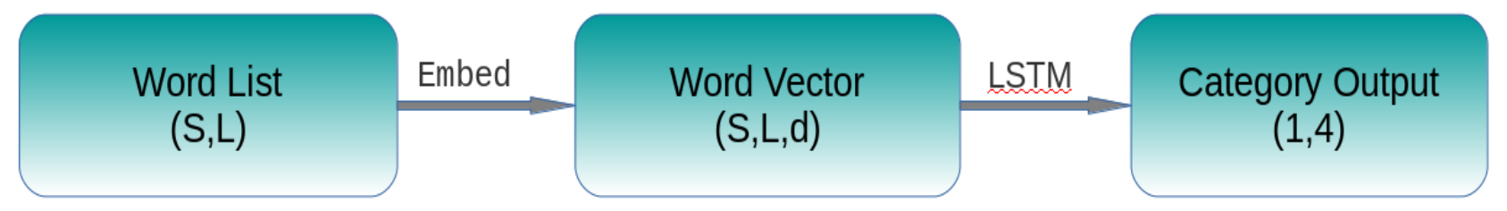
Data Wrangling

- Delete the words which are in the stop-words corpus
- Normalize the DD into [0,1]
- Divide company credit into 4 ranks according to the normalized DD and predetermined threshold; denoted by 4-dimension dummy variable
- Embed the words into vectors in terms of the pretrained model issued by Baidu
- Other data source included the numeric data form bank of china internal database about the financial statistics of each company

Model Construction

Due to the sample size is relatively large, we choose the deep learning RNN model. First of all, we used the pre-trained word2vec model to transfer word list to word vector with same 3D dimension(sample size,length,vector dimension).The network requires same dimension for each sample therefore we truncated the long sample and padded the short sample.

Now we want some method to output only a vector of dimension (1,4) to determine which category this sample belongs to. The deficiency is apparent because of the gradient vanishing. Long-Short Term Memory neural network (LSTM) solves this problem by creating a connection between the forget gate activation and the gradients computation. Of course in the RNN timestep we will mask these padded inputs.


Word List (S,L)
Embed
Word Vector (S,L,d)
LSTM
Category Output (1,4)

s: sample size
l: unified length of each sample
d: dimension of each word vector

Network Structure

A brief structure of neural network is provided as below.

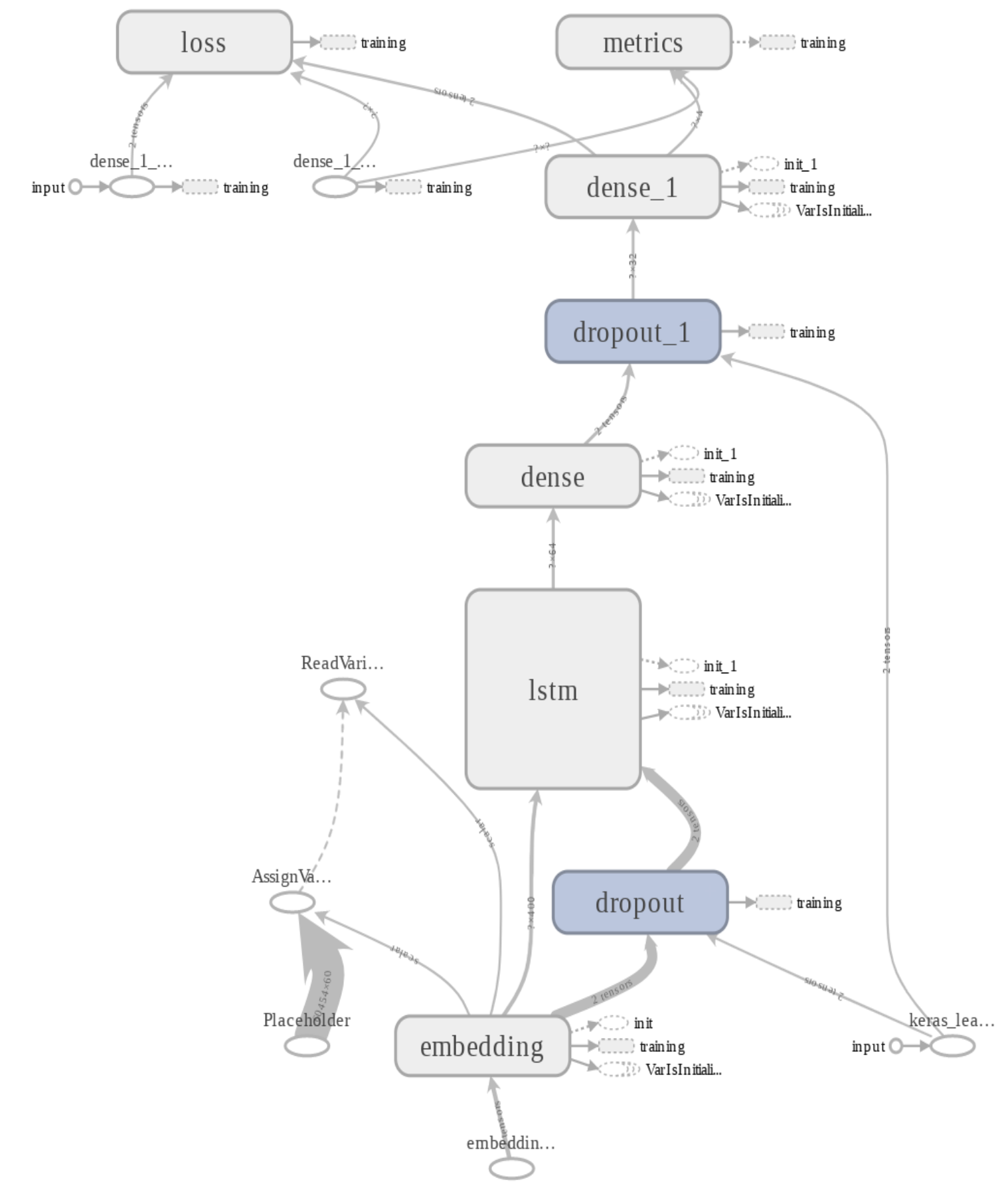


Figure 1: Network Structure

For the LSTM, we can summarize it as following:

$$
\begin{aligned}
i_t &= \sigma(W^i x_t + U^i h_{t-1}) && \textit{Input gate}\\
f_t &= \sigma(W^f x_t + U^f h_{t-1}) && \textit{Forget gate}\\
o_t &= \sigma(W^o x_t + U^o h_{t-1}) && \textit{Exposure gate}\\
n_t &= tanh(W^c x_t + U^c h_{t-1}) && \textit{New memory cell}\\
c_t &= f_t c_{t-1} + i_t n_t && \textit{Final memory cell}\\
h_t &= o_t \cdot tanh(c_t) && \textit{Hidden state}
\end{aligned}
$$

Parameter Issues

- Generally when we use cross-entropy as loss function in classification model and therefore we need a function maximize the entropy which is exactly the exponential family. In bi-classification cases we use sigmoid but here for multi-classification we use the multi-dimension version, the softmax.
- Dropout layers are implemented since they efficiently prevent the over-fitting
- Use mini-batch to accelerate the algorithm
- Why ReLu? ReLU is used because it does not saturate; the gradient is always high (equal to 1) if the neuron activates. As long as it is not a dead neuron, successive updates are fairly effective. ReLU is also very quick to evaluate.
- Why adam? Generally speaking, Adam optimizer achieved great performance with its iterations of both first and second momentum.

Results

Neural Network Outcome

Following we briefly show the loss of the training set and validation set with tensorboard.

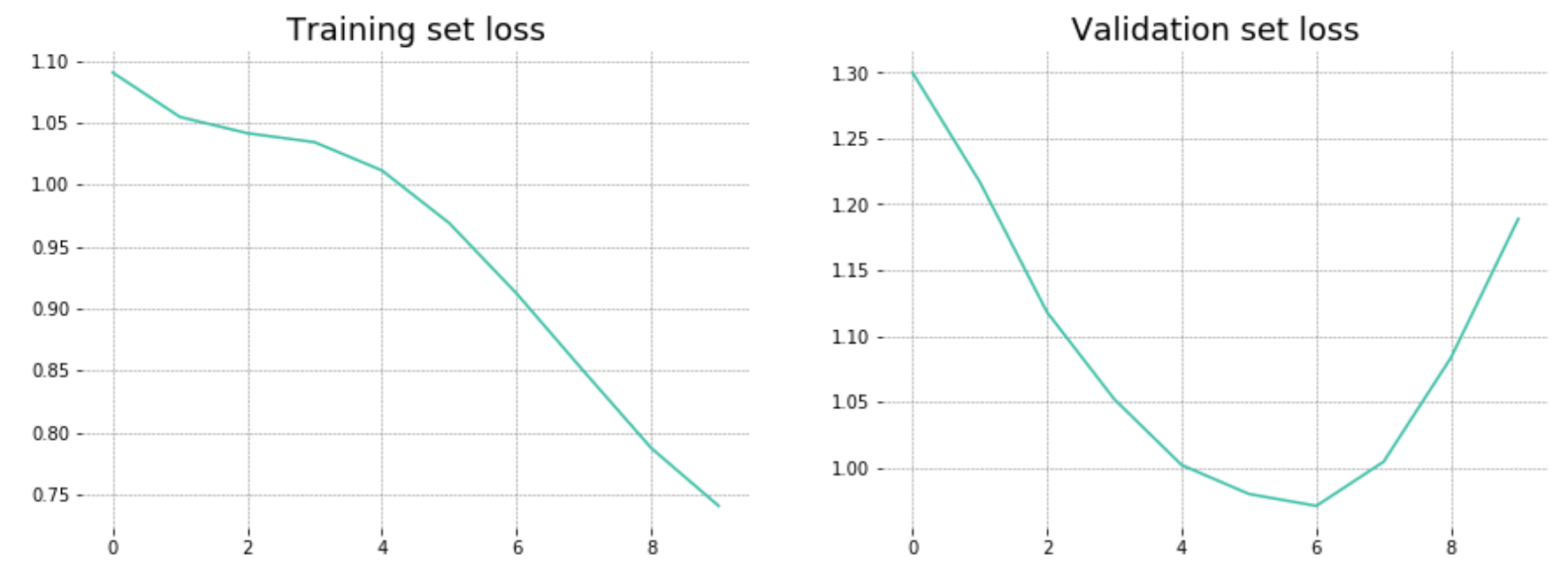
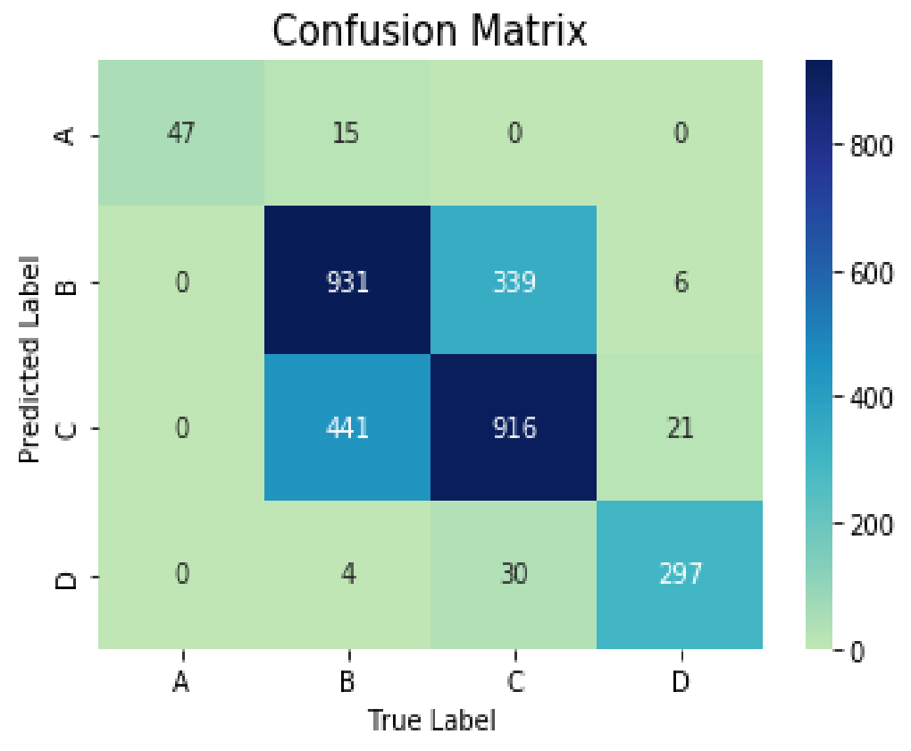


Figure 2: Train-Validation Loss Comparison

The loss of training set is decreasing monotonically but the loss in validation set rebound after several epochs. Different parameter tunings are tried and all of them show the rebound of loss value after 5-10 epochs. Therefore early stop the training will be beneficial.


Confusion Matrix

Predicted Label \ True Label	A	B	C	D
A	47	15	0	0
B	0	931	339	6
C	0	441	916	21
D	0	4	30	297

Figure 3: Confusion Matrix

For the testing test prediction, since the accuracy in multi-classification doesn't show enough information of the classifier quality, we now present a confusion matrix.

From the confusion matrix, the classifier can clearly identify the rank A and rank D company but may misclassify rank B and C with certain probability. It means that the information in the resume can simply distinguish some extreme bad and extreme good company according to the directors' experience (like university,past company and position, published paper etc.)

Another reason of the misclassification can come from the word split since Chinese words are not naturally separated by space.

Therefore, for the samples that are classified as rank B and rank C, other white-box models are provided to determine whether the bank should approve the application for these company.

Further Improvement

Since classification of B and C rank is less accurate, another bi-classification model can be implemented within these two categories. Moreover, since resumes are not providing efficient information to recognize rank B and rank C, we extract some numeric data(such as company age,average revenue,employment number etc.) form database to construct the model. Here we tried many binary models but only present the outcome of LDA(Linear Discriminant Analysis)

To sum up, LDA is finding a direction(w) to divide the data set into two part, which can achieve maximum between-class variance and minimum within-class variance. The optimization target can be written as:

$$J = \frac{w^T \Sigma_b w}{w^T \Sigma_w w} \tag{1}$$

$\Sigma_b : \; between - class\ cov\ matrix$
$\Sigma_w : \; within - class\ cov\ matrix$

Of course we can have multiple solutions when we try to maximize J therefore we need a restriction and the solution will become:

$$w = arg\ min\ (-w^T \Sigma_b w) \tag{2}$$

$$s.t\ \ w^T \Sigma_w w = 1 \tag{3}$$

For the result of this linear model, we use ROC and AUC to examine its quality.

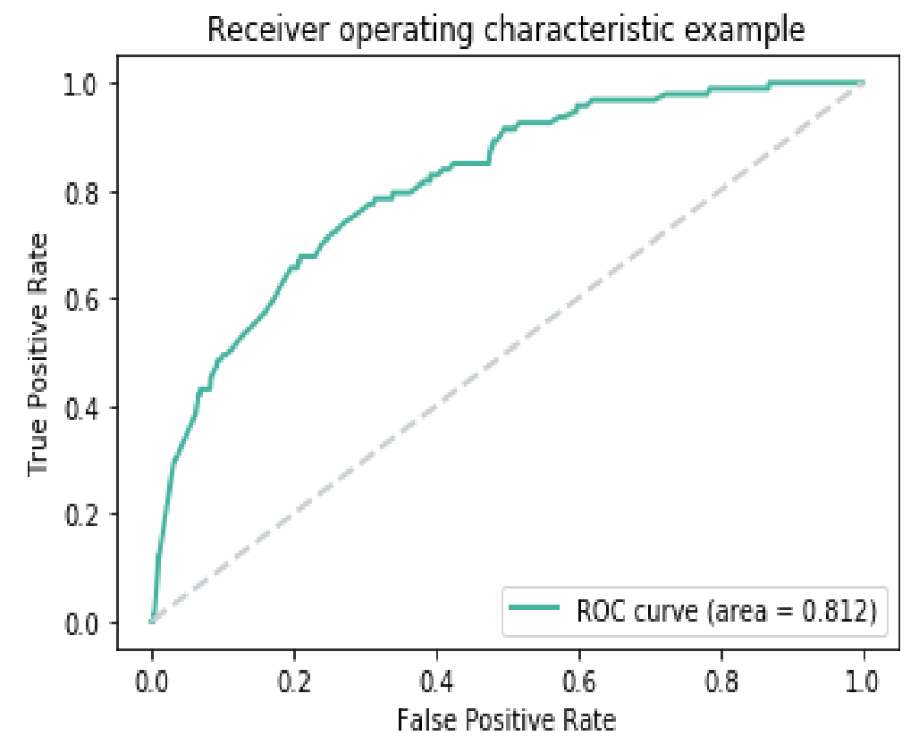


Figure 4: ROC Curve for LDA

Practical Concerns

In reality, accuracy is not the first thing that we concern. Instead, the time cost and model interpretability will be more important than the accuracy. The truth is that the majority of credit models are white-box linear models which are easy to explain and efficient to compute for extreme large data set.

Besides, an approval for a bad client will be much worse than a miss for a dozen of good clients which means the threshold in credit ranking should be adjusted in accordance to bank's risk tolerance.

Conclusion

- The resumes do contain some information about company credit quality which can be reflected on directors' major, working experience and working performance they described.
- Different methods are used to prevent over-fitting and the parameter tuning should be different with different data set.
- The classifier has relatively lower accuracy in classifying rank B and C but it classifies top rank and bottom rank with high accuracy which can avoid dramatic loss from credit default.
- The results can be further improved if we choose to do a bi-classification model within the rank B and rank C samples.
- Data,data and data. Improving data quality and refining the algorithm of computing distance to default will contribute to a more accurate classification.