# Predicting Loan Grades Using Deep Neural Networks

Kaushal Alate (kalate), Bilguunzaya Battogtokh (bbatt99), Liang Ping Koh (lpkoh)

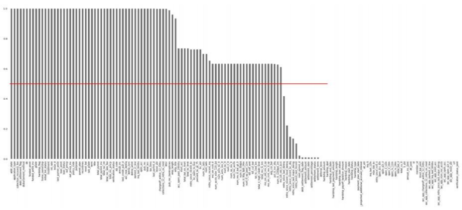
CS 230 (Deep Learning), Stanford University

## Introduction

- Lenders give each loan a grade to help them manage their portfolio's overall risk, set interest rates etc.
- Assigning loan grades can be time-consuming, expensive and difficult as there are many factors to be considered (like borrower's income, credit history etc.)
- Deep learning can enable automatic loan grade prediction or provide a verification tool for human-assigned grades, and we demonstrate that it can do so with 90.2% accuracy

## Data

- Loan data from LendingClub peer-to-peer online lending platform
- "Grade" feature (ground truth) is present
- Removed features/columns with less than 50% valid non-empty entries (figure below)
- Then removed incomplete rows, resulting in 63,207 loans for training and testing



## Features

- 102 features which are raw input data, including loan amount, borrower's income etc. These are appropriate: e.g. intuitively, as borrower's income rises, likelihood of loan being paid back rises (ceteris paribus)
- Each loan also has text "description", "title" and (borrower's) employment title / "emp_title"
- We converted each of these three text features to 50-dimensional embeddings using 1) pre-trained Word2Vec embeddings and 2) custom-trained (on this dataset) Word2Vec embeddings

## Models

***3 models:***

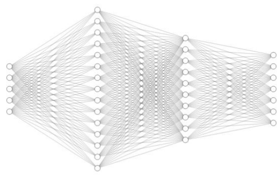
1. No text embeddings, 100 epochs
2. Text embedding is mean of every word's pre-trained Word2Vec embedding, 100 epochs
3. Text embedding is mean of every word's custom-trained Word2Vec embedding, 10 epochs (loss and accuracy plateaued after 10)
   - For model 3, we used L2 regularization to reduce observed overfitting

***Common choices for all models:***

- 100-dimensional dense hidden layer with ReLU and Dropout, followed by
- 50-dimensional dense hidden layer with ReLU and Dropout, and
- 7-dimensional output layer with Softmax
- Adam optimizer and categorical cross-entropy (CE) loss:

$$CE = -\sum_{i}^{C} t_i log(f(s)_i)$$

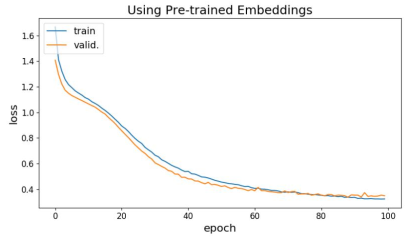
*A visual representation of our model*



## Results

| *Model* | *Train accuracy (train error)* | *Test accuracy (test error)* | *Train set size* | *Test set size* |
|---|---|---|---|---|
| No embeddings | 87.7% (0.3027) | 90.2% (0.2468) | 50570 | 12637 |
| Pre-trained Word2Vec embeddings | 87.8% (0.3069) | 86.6% (0.3366) | 50570 | 12637 |
| Custom-trained Word2Vec embeddings | 34.8% (1.586) | 35.3% (0.3528) | 50570 | 12637 |

*An example of a loss vs. epochs graph*




## Discussion

- Just using numerical & categorical input features led to 90% accuracy
- Using pre-trained text embeddings resulted in a slight drop in accuracy, and using custom-trained embeddings led to a very severe drop
- We were hoping for better accuracy with text embeddings, so we were surprised
- Not much data was available for custom-training embeddings (for every loan, "desc" feature was only a few sentences and "title" and "emp_title" were only a few words), which may explain why they failed to do well

## Future Work

- Accuracy (90.2%) seems high without text embeddings so it may be prudent to focus on tuning hyperparameters (such as the number of layers and neurons) further to get (closer) to 100 percent accuracy

## References

1. Jin, Yu, and Yudan Zhu. "A data-driven approach to predict default risk of loan for online Peer-to-Peer (P2P) lending." In 2015 Fifth International Conference on Communication Systems and Network Technologies, pp. 609-613. IEEE, 2015.
2. Lee, Heeyoung, Mihai Surdeanu, Bill MacCartney, and Dan Jurafsky. "On the Importance of Text Analysis for Stock Price Prediction." In LREC, pp. 1170-1175. 2014. [On a different financial topic but it informed our approach]
3. Serrano-Cinca, Carlos, Begoña Gutiérrez-Nieto, and Luz López-Palacios. "Determinants of default in P2P lending." PloS one 10, no. 10 (2015): e0139427.