MODELING AGENCY CMBS PREPAYMENTS USING INDEPENDENTLY RECURRENT NEURAL NETWORKS

PROJECT SUMMARY

Róbert Ferenc Dóró

Contents

1. Project Background

While nowadays residential borrowers exercise their implied call options more efficiently than they used to due to the activity of so-called „fast" mortgage servicers (Quicken Loans, Flagstar Bank etc.), commercial borrowers are still more efficient. In addition, while a residential mortgage pool consists of hundreds of loans, multifamily mortgage pools consist of a single loan in many cases, making the investment opportunity less diversified. Therefore, lenders require commercial borrowers to agree to very strict prepayment provision terms in order to offer investors more stable cash flow streams and limit exposure to negative convexity. Such provisions make the possibility of prepaying or refinancing the loan less attractive as the penalties imposed on the prepaid amount can exceed the dollar gain if the loan was refinanced.

Agency CMBS products are either guaranteed by the U.S. government or by one of the Government Sponsored Enterprises. The guarantee ensures that investors virtually do not bear any credit risk and that any default event will be a prepayment event from the investor's perspective. As a result, by allowing to extend the range of potential prepayment sources with default-related events, an investor is only interested in pricing the implied call option via a prepayment model.

It is well documented that recurrent neural networks (RNN) that have memory cells can easily outperform basic RNNs. The Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) cells have gained popularity among researchers to predict prepayments. However, the application of LSTM cells is not the only solution that prevents gradients from exploding or vanishing. The Independently Recurrent Neural Network (IndRNN) was first proposed by Shuai et al. (2019). The novel approach is based on a regularization technique that does not let the gradients to deviate from a predefined range. The motivation for this variant was to address the gradient decay problem from which the LSTM and GRU cells with hyperbolic tangent and sigmoid functions suffer. In addition to reducing the computational complexity to train the model, the regular RNN structures in many cases are hard to interpret as their layers are fully connected. The IndRNN model used in this project aim to predict the status of a commercial mortgage loan using publicly available loan-specific and macroeconomic variables.

2. The Data

U.S. agencies publish monthly loan performance data to their websites and are freely accessible to the public. I have selected the Federal National Mortgage Association's loan performance database. The loan-specific attributes that are available in the data tape were sufficient for the purpose of this analysis. The loans have been acquired by Fannie Mae on or after January 1, 2000 through December 31, 2019. Most of the loans are part of Fannie Mae's Delegated Underwriting & Servicing (DUS) Program.

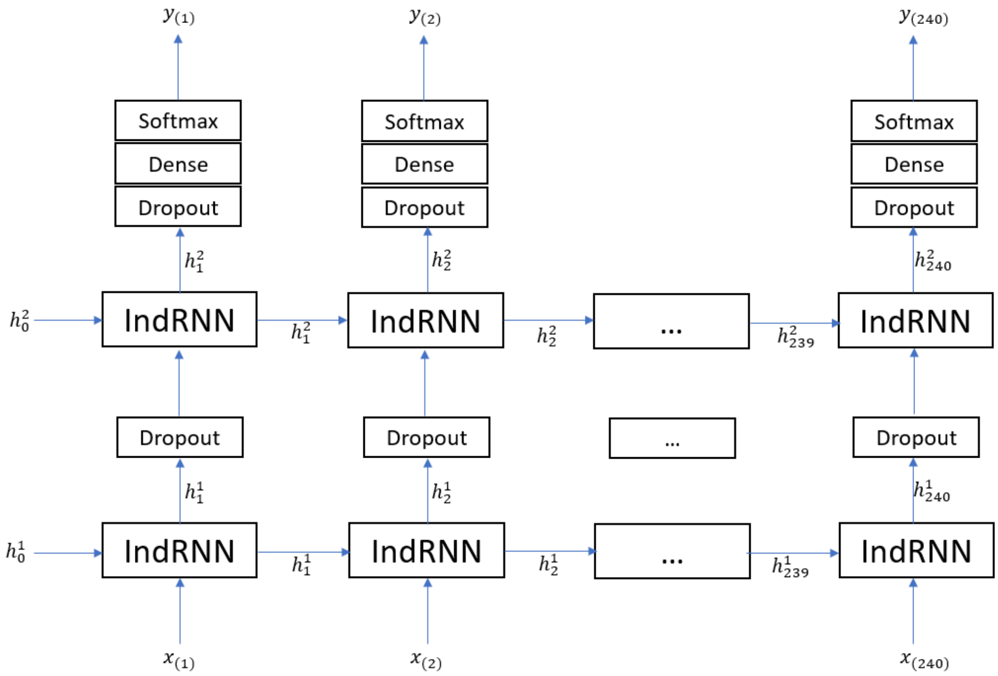
In order to capture borrower behavior due to changes in market environment, I have selected three macroeconomic variables as well. Monthly state level house price index data was downloaded from the Federal Housing Finance Agency's website. Freddie Mac's Primary Mortgage Market Survey mortgage rate series was used as a benchmark for mortgage rate changes. Metropolitan statistical area level house price index data was collected from the U.S. Bureau of Labor Statistics' website. The categorical features were one-hot encoded, the numerical features were standardized. Given that for most of the loans the original term is less than 240 months, I decided to zero pad the data up to 240 months. The target feature values were pre-processed and mapped to one of the following categories: Live, Prepaid, Not Known. The dataset was split into training, validation and test sets.

3. Framework

The LSTM network was used as a baseline model to evaluate the IndRNN model's performance. In an LSTM cell, four fully connected layers are created to control their recurrent gates. In contrast, in an IndRNN cell there is only a single layer that eliminates the necessity of long- and short-term states. Dropping 75% of the trainable parameters results in a computationally less complex task. Furthermore, each neuron is completely independent from the rest of the neurons at the same depth. The connections between the neurons can be established by stacking multiple layers (Shuai et al., 2019).

The initial hidden (both long-term and short-term) states were set to zero. First, the input layer passes through data to the first LSTM/IndRNN layer. The following dropout layer drops some of the activations and lets the rest to flow through. Depending on the number of layers used to train the model, LSTM/IndRNN and dropout layer pairs are stacked on top of previous pairs. The last dropout layer is followed by a dense layer with three neurons for the output labels. The three output labels are mutually exclusive. Therefore, I have selected the softmax function to produce the final outputs. Weighted categorical cross entropy was selected as the loss function in order not to overtrain the model on Live and Not Known labels by favoring false positives for prepayments/defaults over missing a true positive. The weights were selected based on the inverse proportion of the labels' occurrence so as to address the issue of imbalanced data. The network is a many-to-many RNN as we are interested in all the interim output states of the model, not just the last one. Figure 1 shows the complete unrolled IndRNN network.

Figure 1: IndRNN network unrolled through time



4. Results/Conclusion

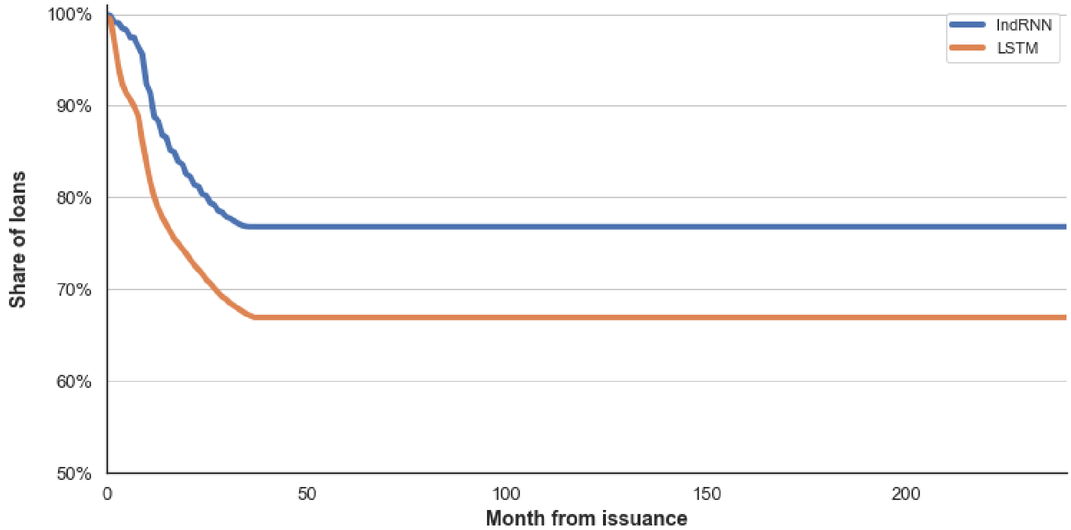
The models were implemented using Tensorflow and Keras in python. Table 1 shows the performance of the selected models on the training, the validation and the test sets. The accuracy is the mean accuracy rate across all predictions. Note, that the weights of the loss function do not represent the inverse proportion of the rate of occurrence in the test set as there are only 36 months of known information about the loans. Since we are only interested in the generalization ability of the models in relation to the other, this is an acceptable compromise.

Table 1: Performance metrics of the selected models

	Metric	LSTM	IndRNN
Training	loss	0.0013	0.0014
	accuracy	0.9564	0.9565
Validation	loss	0.0013	0.0013
	accuracy	0.9705	0.9664
Test	loss	0.0002	0.0001
	accuracy	0.9924	0.9934

The LSTM model performs slightly better (97.05%) than the IndRNN model (96.64%) on the validation set, whereas the IndRNN model outperforms (99.34%) the LSTM model (99.24%) on the test set. An alternative categorical accuracy measure is shown in Figure 14: the proportion of loans in each month that have been labeled correctly from the very first month. The IndRNN model performs significantly better than the LSTM model using this type of accuracy. By the end of the 36th month, the LSTM still labels 67% of the loans correctly, whereas the IndRNN scored around 77%.

Figure 14: Proportion of loans that are correctly labeled from origination



In this project I have analyzed two alternative models using a single framework to model long-term agency CMBS prepayments. LSTM/GRU networks are regarded as the go-to RNN models for similar tasks. The application of IndRNN challenges this view and aims to extend the list of RNNs with memory cells which can effectively tackle the issue of prepayment forecasting. The results indicate that IndRNN models can compete against LSTM cells in modeling prepayments and are even capable of outperforming them. Therefore, it is worth testing IndRNN networks as a potential candidate model to serve as a replacement for traditional prepayment models.

References

Shuai Li, W. L. (2019). *Deep Independently Recurrent Neural Network (IndRNN).* arXiv.