arXiv:1607.02470v1 [q-fin.ST] 8 Jul 2016

Deep Learning for Mortgage Risk

Justin A. Sirignano, Apaar Sadhwani, and Kay Giesecke*

September 15, 2015; this version: July 7, 2016†

Abstract

This paper analyzes multi-period mortgage risk at loan and pool levels using an unprecedented dataset of over 120 million prime and subprime mortgages originated across the United States between 1995 and 2014, which includes the individual characteristics of each loan, monthly updates on loan performance over the life of a loan, and a number of time-varying economic variables at the zip code level. We develop, estimate, and test dynamic machine learning models for mortgage prepayment, delinquency, and foreclosure which capture loan-to-loan correlation due to geographic proximity and exposure to common risk factors. The basic building block is a deep neural network which addresses the nonlinear relationship between the explanatory variables and loan performance. Our likelihood estimators, which are based on 3.5 billion borrower-month observations, indicate that mortgage risk is strongly influenced by local economic factors such as zip-code level foreclosure rates. The out-of-sample predictive performance of our deep learning model is a significant improvement over linear models such as logistic regression. Model parameters are estimated using GPU parallel computing due to the computational challenges associated with the large amount of data. The deep learning model's superior accuracy compared to linear models directly translates into improved performance for investors. Portfolios constructed with the deep learning model have lower prepayment and delinquency rates than portfolios chosen with a logistic regression.

*Sirignano (jasirign@illinois.edu) is from the University of Illinois at Urbana-Champaign. Sadhwani (apaars@stanford.edu) and Giesecke (giesecke@stanford.edu) are from Stanford University.

†The authors gratefully acknowledge support from the National Science Foundation through Methodology, Measurement, and Statistics Grant No. 1325031 as well as from the Amazon Web Services in Education Grant award. We are very grateful to Michael Ohlrogge, Andreas Eckner, Jason Su, and Ian Goodfellow for comments. Chris Palmer, Richard Stanton, and Amit Seru, our discussants at the Macro Financial Modeling Winter 2016 Meeting, provided insightful comments on this work, for which we are very grateful. We are also grateful for comments from the participants of the Macro Financial Modeling Winter 2016 Meeting, the 7th General AMaMeF and Swissquote Conference in Lausanne, the Risk Seminar at the Federal Reserve Bank of San Francisco, the Western Conference on Mathematical Finance at UT Austin, the Machine Learning in Finance Conference at Columbia University, Northwestern University, Imperial College London, Georgia State University, the Consortium of Data Analytics in Risk Symposium at UC Berkeley, and seminar participants at Morgan Stanley, J.P. Morgan, Payoff, Bank of England, and Winton Capital Management. We are also grateful to Powerlytics, Inc. for providing access to income data.

1 Introduction

This paper develops and tests a multi-period deep learning model for mortgage delinquency and prepayment risk at loan and pool levels. The analysis is based on an unprecedented dataset of over 120 million prime and subprime mortgages originated across the United States between 1995 and 2014, which includes the individual characteristics of each loan as well as monthly updates on loan performance. The breadth and granularity of the data allows us to accurately measure the influence on mortgage risk of the local economic conditions that borrowers face. For example, the zip-code level foreclosure rate is a significant predictor of risk. We also find evidence of highly nonlinear relationships between the explanatory variables and the behavior of mortgage risk. Capturing these nonlinear relationships significantly improves out-of-sample predictive performance, especially for prepayment.

In our model, a mortgage can be in one of several states, including current, 30 days past due, 60 days past due, 90+ days past due, foreclosure, REO (real estate owned), and paid off. The probability transition function between the states is specified by a neural network model, whose input is a vector of explanatory variables at the beginning of a period and whose output is the conditional probability distribution for the mortgage's state at the end of a period. In a neural network, the relation between explanatory variables and state transition probabilities is described by an interconnected set of input, output, and "hidden" nodes. The input nodes represent the explanatory variables while the output nodes represent the state transition probabilities. The hidden nodes connect the input and output nodes, and transform linear combinations of input variables in a nonlinear fashion. They capture the highly nonlinear relationships we observe between many of the explanatory variables and state transition rates. In particular, we study *deep* neural networks, which have multiple layers of hidden nodes. Our use of deep neural networks enables us to extract richer and more complex nonlinear features from the mortgage data than shallow or one-layer networks. In our out-of-sample tests of predictive performance, deep networks outperform shallow networks, which in turn outperform baseline models which only consider linear dependencies on covariates (i.e., logistic regression). Linear models of mortgage delinquency and prepayment are standard in the empirical literature.

The massive size of our loan performance dataset allows us to examine the influence on mortgage risk of a broad set of static and time-varying risk factors. We consider a wide range of loan-level characteristics at origination (e.g., credit score, loan-to-value ratio, product type and features, and many others), as well as a number of variables describing loan performance (e.g., number of times delinquent in past year). We also consider several time-varying factors that describe the economic conditions a borrower faces, including both local variables such as housing prices, average incomes, and foreclosure rates at the zip code level, as well as national-level variables such as mortgage rates. Our model takes account of the significant volatility of these factors over the sample period. It allows us to incorporate the stochastic dynamics of the variables when estimating state transition probabilities over several future periods. The dependence of state transition probabilities on common or correlated factors captures the loan-to-loan correlation observed in the data. Addressing this correlation allows for the analysis of delinquency and prepayment risk at the pool level. Prepayment risk is important because prepayments lead to early cashflows that might have to be reinvested at lower interest rates.

We use maximum likelihood methods to fit our deep learning model. In total, we have over 3.5 billion mortgage-month observations, each characterized by nearly 300 explanatory variables (almost 2 terabytes of data). We use a GPU parallel computing approach to address the significant computational challenges posed by the size of the dataset. The fitting algorithms run on a cluster of Amazon Web Services (AWS) nodes. We maximize the likelihood using minibatch gradient descent on a sequence of blocks of data. In order to avoid overfitting, we use ℓ^2 regularization, dropout, choose an optimal model complexity via cross-validation, and build ensembles of deep learning models. As a result, we find no evidence of overfitting even though our nonlinear model is fairly rich.

The fitted model yields insight into the key variables influencing mortgage delinquency and prepayment. In particular, we find local economic factors such as unemployment rates in the mortgage's county, as well as housing prices and lagged foreclosure rates in the mortgage's zip code, to have significant explanatory power. These factors reflect the local economic conditions a borrower faces. Local economic data is often unavailable or infrequently reported; the lagged foreclosure rates at the zip code level offer a new approach for incorporating geographic risk factors in loan-level models. The significance of foreclosure rates is consistent with a contagion effect where a mortgage becomes more likely to default when nearby mortgages default. Recent empirical work documents such an effect.[1]

Out-of-sample tests over a roughly 2 year period document the performance of our deep learning model when predicting mortgage state transitions at both the loan and pool levels over multiple future horizons. We consider several performance measures, including the negative log-likelihood (which is equivalent to the cross-entropy error) and the receiver operating characteristic (ROC) curve. Pool-level performance is measured using a portfolio ranking metric and pool-level prediction error. We find that the neural network strongly outperforms a baseline logistic regression model, indicating the importance of incorporating nonlinear relationships between state transition rates and risk factors. In particular, the deep learning model is significantly more accurate for predicting prepayment. The deep learning model's area under the ROC curve (AUC) is over 10% greater than the logistic regression's AUC for predicting transitions to paid off. The deep learning model's superior accuracy directly translates into improved profit and loss for an investor or lender. Portfolios constructed using the deep learning model outperform portfolios chosen via the logistic regression model, with a 50% reduction in prepayments over a 1 year out-of-sample period. The neural network portfolio has a 46% smaller loss than the logistic regression portfolio at a 1 year time horizon.

This paper's results show that significant practical gains can be achieved using deep learning loan-level models. We anticipate several types of applications for our work, including (i) the analysis by a mortgage lender or originator of the delinquency and prepayment risk of a borrower over several future periods, (ii) the determination by lenders and their regulators

[1]See Agarwal, Ambrose & Yildirim (2015), Anenberg & Kung (2014), Campbell, Gigli & Pathak (2011), Goodstein, Hanouna, Ramirez & Stahel (2011), Harding, Rosenblatt & Yao (2009), Lin, Rosenblatt & Yao (2009), Towe & Lawley (2013), and others.

of how much capital lenders need to fund their mortgage portfolios, (iii) the stress testing of mortgage portfolios, (iv) the analysis by a mortgage servicer of the appropriate service actions given a mortgage's state, (v) the determination by rating agencies of credit ratings for mortgage backed securities, and (vi) the study of the interaction between local economic conditions and mortgage risk.

The remainder of the introduction discusses the related literature. Section 2 discusses our dataset and performs some exploratory analyses. Section 3 develops the modeling framework and specifies the neural network model. Section 4 provides details on likelihood estimation and our computational approach. Section 5 discusses the empirical results. Section 6 offers concluding remarks.

1.1 Related Literature

There is substantial empirical literature on mortgage delinquency and prepayment risk. In early work, von Furstenberg (1969) and von Furstenberg (1970) establish the influence on home mortgage default rates of variables such as income, loan age, and loan-to-value ratio. Gau (1978) and Vandell (1978) examine additional variables. Commercial mortgage default was first studied by Magee (1968) and Vandell (1992). Curley & Guttentag (1974) is an early study of prepayment rates. Green & Shoven (1986) examine the influence of interest rates on prepayments. Campbell & Dietrich (1983) and Schwartz & Torous (1993) analyze the influence on default and prepayment of several loan-level and macro-economic variables, recognizing the "competing" nature of these events. Schwartz & Torous (1989) pioneered the use of empirical pool-level prepayment models for the pricing of agency mortgage-backed securities. More recently, Stanton & Wallace (2011) use empirical models of default and prepayment to price private-label MBS. Chernov, Dunn & Longstaff (2016) estimate market-implied risk-neutral prepayment rates using agency MBS prices, and relate them to various explanatory variables. Gorovoy & Linetsky (2007) propose a model for the risk-neutral dynamics of prepayment rates to develop closed-form values for mortgages.

Our loan-level modeling approach represents a significant departure from earlier formulations. It is based on an unprecedented dataset of 120 million prime and subprime mortgages observed over the period 1995–2014. Prior work examines much smaller samples, on the order of thousands to hundreds of thousands of loans, focusing on specific geographic regions, time periods, loan products, or borrower profiles. Our dataset covers about 70 percent of all mortgages originated across the United States between 1995 and 2014. It includes all product types, including fixed-rate, adjustable-rate, hybrid, balloon, and other types of loans, and tracks the performance of these loans during several economic cycles. With samples spread across over 30,000 zip codes, we are in a position to study the joint influence on mortgage state transitions of a broad set of risk factors describing borrower, product, and performance characteristics, as well as local and national economic conditions. We show that *local* conditions, for instance local unemployment rates, housing prices, and foreclosure rates, have a significant influence on mortgage risk. This influence can be measured accurately only if there is a sufficient number of samples at the local (i.e., zip code) level.

Besides yielding insights into the local drivers of mortgage delinquency and prepayment risk, the breadth of our dataset allows us to distinguish between multiple mortgage states. Earlier work typically only considers transitions from current to default or prepayment.[2] However, this ignores important transitions between other delinquency states such as 30 days behind payment, 60 days behind payment, 90+ days behind payment, and foreclosure. Transitions between these states are frequent; see Tables 8–10. Many mortgages enter foreclosure but eventually return to current. Similarly, many mortgages are consistently behind payment but do not ever enter foreclosure. During these periods of delinquency, the lender or servicer suffers a disruption to the cashflow from the mortgage. Such delays in cashflow pose a significant risk for the lender/servicer and these risk exposures need to be considered. Our approach models the transitions between the various states of delinquency, conditional on a vector of loan-level as well as local and national economic factors (which themselves might be random and time-varying). Once fitted, our model can be used by a servicer to optimally choose among alternative responses to delinquencies. Our dynamic model can also be used for multi-period risk analysis and selection of loan portfolios; see Sirignano & Giesecke (2015) and Sirignano, Tsoukalas & Giesecke (2016). The model gives the distribution of the total cashflow from the loan portfolio and the distribution of the loans in the different states of delinquencies. Accounting for all of the possible states of delinquency allows for more accurate modeling of the timing of portfolio cashflows.

There are several statistical advantages to using our large dataset rather than a small sample as in prior work, justifying the effort to overcome the computational challenges that such big datasets generate. Mortgage data is very high-dimensional and therefore large amounts of data are required to accurately fit models. Larger datasets allow richer, nonlinear models to be estimated without overfitting. Secondly, the additional data allows one to model infrequent transitions. As shown in Tables 8–10, several state transitions are relatively rare. For example, a transition from 90+ days past due to REO for a prime borrower has only a 0.3% probability. Finally, local economic conditions are an important factor influencing mortgage risk. Local economic conditions can be inferred via local lagged foreclosure and prepayment rates. However, large amounts of data are required to accurately measure lagged foreclosure and prepayment rates at a granular geographic level (e.g., zip code level).

Machine learning has been extremely successful in applications to static classification problems (e.g., image classification, speech recognition, and natural language processing). Several papers have also used machine learning to predict defaults of consumer loans (e.g., Khandani, Kim & Lo (2010), Butaru, Chen, Clark, Das, Lo & Siddique (2015)), mortgages (e.g., Episcopos, Pericli & Hu (1998), Feldman & Gross (2005), Fitzpatrick & Mues (2016)), and other types of loans. However, the models developed in these papers are static and give a result for a single time horizon (e.g., a year). In our paper, a machine learning classifier is embedded in a dynamic framework. We propose a dynamic model for mortgage transitions where the transition probability function is given by a machine learning model (e.g., logistic regression or a neural network). Unlike a static model, this formulation generates forecasts for multiple future time periods that incorporate the time-series dynamics of the risk factors of those periods.

[2] "Default" usually means a severe delinquency (such as 60 days or more late).

Status quo models in both industry and the academic literature use models which are link functions of a linear combination of the explanatory variables. Standard examples are logistic regression (e.g., Campbell & Dietrich (1983) and others) and the Cox proportional hazard model (e.g., Deng (1997), Deng, Quigley & Van Order (2000), Moody's Analytics (2013), Palmer (2013), Schwartz & Torous (1993), Stanton & Wallace (2011), and many others).[3] In contrast, our model has a highly nonlinear architecture with neural networks as its fundamental building block. The highly nonlinear structure of our model allows it to capture the nonlinear relationships between loan features and loan behavior we observe in the data (see Section 2.4). In out-of-sample tests and using multiple performance metrics, our neural network model outperforms a simpler model based upon the standard logistic regression. Shallow neural networks (in a static framework) have previously been used for predicting financial default, see Episcopos et al. (1998), West (2000), Baesens (2005), Atiya (2001), and others. These previous papers differ significantly from this paper. These papers fit their neural networks to only a few thousand samples while we have over 3.5 billion samples. This allows us to fit a much more complex neural network with a richer architecture, which proves to be essential in accurately capturing the unique risk characteristics of a variety of different types of mortgages. We use deep neural networks, which have been the object of extensive research in machine learning over the past decade and have enjoyed great success across a number of classification problems. To further improve accuracy, we build an ensemble of neural networks. Mortgage performance data is extremely noisy compared to standard classification problems, and the ensemble proves to be key in reducing the variance of the prediction and improving out-of-sample results.

This is one of the first papers exploring the use of deep learning methods in finance. In an early application of neural networks to finance, Hutchinson, Lo & Poggio (1994) use shallow neural nets to price options. The application of deep learning to finance has only been explored very recently. Sirignano (2016) develops a deep neural network architecture which takes advantage of the specific structure of limit order books. Sirignano (2016)'s deep learning approach outperforms status quo models for limit order books. Dixon, Klabjan & Bang (2016) predict the direction of financial market movements using a deep neural network. Heaton, Polson & Witte (2016) outline some other financial applications of deep learning.

Our paper takes advantage of many of the methods developed in the deep learning literature over the past decade. Deep learning has achieved major success across a number of fields. Deep convolution networks have reached state-of-the-art performance for image classification; see Krizhevsky, Sutskever & Hinton (2012), Ciresan, Meier & Schmidhuber (2012), Simonyan & Zisserman (2014), Ciresan, Meier, Gambardella & Schmidhuber (2011), Larochelle, Bengio, Louradour & Lamblin (2009), Ciresan, Meier, Gambardella & Schmidhuber (2010), Karpathy, Toderici, Shetty, Leung, Sukthankar & Fei-Fei (2014), and Glorot, Bordes & Bengio (2011). Similarly, impressive results have been achieved in natural language processing and speech recognition; see Hinton, Deng, Yu, Dahl, Mohamed, Jaitly, Senior, Vanhoucke, Nguyen, Sainath & Kingsbury (2012), Dahl, Yu, Deng & Acero (2012), Graves,

[3] Linear models are also standard in the empirical literature examining corporate credit risk. See Duffie, Saita & Wang (2007) and many others.

Mohamed & Hinton (2013), Collobert & Weston (2008), Dahl et al. (2012), and Wang, Wu, Coates & Ng (2012). Recently, deep neural networks have also been used for reinforcement learning; see Mnih, Kavukcuoglu, Silver, Rusu, Veness, Bellemare, Graves, Riedmiller, Fidjeland, Ostrovski & Petersen (2015), Guo, Singh, Lee, Lewis & Wang (2014), Hasselt, Guez & Silver (2015), and Lillicrap, Hunt, Pritzel, Heess, Erez, Tassa, Silver & Wierstra (2015).

2 The Data

Our dataset includes data for over 120 million mortgages as well as local and national economic factors. The mortgage dataset includes highly-detailed characteristics for each mortgage and month-by-month loan performance. We complement this dataset with extensive local economic data such as zip code level housing prices, average incomes, and unemployment rates.

2.1 Loan Features and Performance Data

The mortgage data was acquired from CoreLogic, who collects the data from mortgage originators and servicers. It is the most complete mortgage dataset available from any data vendor. It covers roughly 70 percent of all mortgages originated in the US and contains mortgages from over 30,000 zip codes across the US. The mortgages' origination dates range from January 1995 to June 2014. The dataset includes 25 million subprime and 93 million prime mortgages. We adopt CoreLogic's designation of loans as subprime vs. prime. These designations are based on their categorizations by the originators and servicers who provide the mortgage data to CoreLogic. Loan characteristics such as FICO score, documentation status, and product features such as negative amortization are often used in practice to distinguish between subprime and prime loans. Our approach reflects the way that these loans are viewed by the key economic actors in the mortgage market. The loan data is divided into (1) loan features at origination and (2) performance data, which we describe below.

2.1.1 Loan Features at Origination

Each mortgage has a number of detailed features at origination, such as borrower FICO score, original loan-to-value (LTV) ratio, original debt-to-income (DTI) ratio, original balance, original interest rate, product type, type of property, prepayment penalties, zip code, state, and many more. Many of the variables are categorical with many categories (some with up to 20 categories).[4] A complete list of the features we consider and their possible values is provided in Table 2. Tables 3 through 5 provide summary statistics for FICO score, original LTV ratio, original interest rate, and original balance. The median FICO score of subprime

[4]We encode a categorical variable as a vector with length equal to the number of categories. The vector's elements are all 0's except for one element which equals 1 to indicate the value of the categorical variable for that sample.

borrowers is 630, while that of prime borrowers is 730. The median interest rate of subprime loans is 7.8 percent, while that for prime loans is 5.8 percent.

2.1.2 Performance Data

Month-by-month performance for each mortgage is reported between 1995 and 2014. This includes how many days behind payment the mortgage currently is, the current interest rate, current balance, whether the mortgage is real estate owned (REO), is in foreclosure, or has paid off. Table 6 provides the full list of performance features we consider.

2.1.3 Types of mortgage products

The dataset covers various mortgage products including, for example, fixed rate mortgages, adjustable rate mortgages (ARMs), hybrid mortgages, and balloon mortgages. A fixed rate mortgage has constant interest and principal payments over the lifetime of the mortgage. An ARM has interest payments which fluctuate with some other index interest rate (such as the Treasury rate) plus some fixed margin. A hybrid mortgage has a period with a fixed rate followed by a period with an adjustable rate. Hybrid mortgages can also have other features such as interest rate caps. A balloon mortgage only partially amortizes; a portion of the loan principal is due at maturity. Table 1 lists the fraction of mortgages in each product category. The vast majority of the prime mortgages are fixed-rate (86%), followed by ARMs (9%) and hybrids (4%). 48% of the subprime mortgages are fixed-rate, 29% are ARMs, 6% are balloon products, and 9% are hybrids.

2.1.4 Missing data and errors

A fraction of the mortgages have missing data, including missing data for some of the key features such as FICO, LTV ratio, original interest rate, and original balance. The missing data is a result of reporting errors by the originator or servicer, or incomplete information provided by the borrower at the time of origination. Key features that are missing are more likely to be the result of reporting errors. For instance, original balance and original interest rate are required details in any mortgage contract. If they are missing, it must be due to a reporting error and not the borrower failing to provide this information. Similarly, FICO score and LTV ratio are almost universally available for mortgage borrowers. Other features, however, may simply not have been provided by the borrower at the time of origination. An example is debt-to-income ratio, which is often not available for subprime borrowers. We take the following approach towards missing data in the CoreLogic dataset. We insist that any sample must have at least FICO, LTV ratio, original interest rate, and original balance. Samples missing one of these variables are removed, leaving us with roughly 73 million mortgages (60% of the prime loans and 70% of the subprime loans). Missing data for other features, which are more likely to be due to incomplete information provided by the borrower, are typically encoded as an additional indicator variable (1 if it's missing, 0 if it's not missing). This approach eliminates the need to remove the corresponding samples, and allows us to measure the implications of missing features.

There are also certain events reported in the CoreLogic dataset which are errors. For instance, monthly mortgage transitions from current to 60 days delinquent or from 30 days delinquent to 90+ days delinquent are not possible. Errors of this type are very infrequent in the dataset and we remove those samples where such errors occur. Mortgages can also have their "servicing released" or their state may be reported as "unknown", "status no longer provided", or "invalid data". "Servicing released" means the servicer which previously reported the data to CoreLogic for that particular mortgage no longer services that mortgage and therefore no longer reports data for it. The mortgage state being "unknown", "status no longer provided", or "invalid data" could be due to a range of clerical/software errors. Whenever a mortgage is in any of these states or has its "servicing released", we exclude any subsequent monthly data from our sample.

Every monthly observation from each of the 73 million loans constitutes a data sample. After cleaning the data, there are roughly 3.5 billion monthly observations remaining. 90% of the samples are for prime mortgages and the remaining are for subprime. The samples cover the period January, 1995 to May, 2014. Each sample (i.e., monthly observation) has 294 explanatory variables as well as the outcome for that month (i.e., if the loan is current, 30 days delinquent, 60 days delinquent, etc.). Of the 294 explanatory variables, 233 are loan-level feature and performance variables, 25 are indicators for missing features, and 36 are economic variables which are introduced next.

2.2 Local and National Economic Factors

We complement the loan-level data described above by data for local and national economic factors which may influence loan performance. Table 7 lists the factors we consider. We use a mortgage's zip code to match a mortgage with local factors such as the monthly housing price in that zip code. Housing prices are obtained from Zillow and the Federal Housing Administration (FHA). Zillow housing prices are at the five-digit zip code level and cover roughly 10,000 zip codes. In order to cover less populated areas not covered by the Zillow dataset, we also include FHA housing prices which cover all three-digit zip codes. The monthly national mortgage rate, which is the primary driver of prepayments and obtained from Freddie Mac, is also included as a factor.[5] Unemployment rates at the county level for each year and state unemployment rates for each month are obtained from the Bureau of Labor Statistics.[6] Our data also includes the yearly median income in each zip code, which was acquired from the data provider Powerlytics. Moreover, we include a dummy variable for the current year. Finally, the granular geographic data is used to construct the lagged default and prepayment rates in each zip code across the US, using the historical data for all 73 million mortgages. The inclusion of these rates allows us to capture the contagion effect where defaults of mortgages increase the likelihood of default for nearby surviving mortgages. In light of the mortgage meltdown, such a feedback mechanism has

[5] The monthly national mortgage rate used in this paper is an average of 30 year fixed rates for first-lien prime conventional conforming home purchase mortgages with a loan-to-value of 80 percent.

[6] We match counties and zip codes in order to associate each mortgage with a particular county.

been supported by several recent empirical papers; see Agarwal et al. (2015), Anenberg & Kung (2014), Campbell et al. (2011), Goodstein et al. (2011), Harding et al. (2009), Lin et al. (2009), Towe & Lawley (2013), and others.

2.3 Mortgage States and Transitions

Mortgages can transition between 7 states: current, 30 days delinquent, 60 days delinquent, 90+ days delinquent, foreclosed, REO, and paid off. X days delinquent simply means the mortgage borrower is X days behind on their payments to the lender. We use the standard established by the Mortgage Bankers Association of America for determining the state of delinquency. A mortgage is determined to be 1 month delinquent if no payment has been made by the last day of the month and the payment was due on the first day of the month. REO stands for real estate owned property. When a foreclosed mortgage does not sell at auction, the lender or servicer will assume ownership of the property. Paid off can occur from a mortgage prepaying, maturing (this is very rare since the mortgages in the dataset are almost entirely originated in the 2000s), a shortsale, or a foreclosed mortgage being sold at auction to a third party (this is again rare in comparison to prepayments, which form the bulk of the paid off events in the dataset).[7]

The state transition matrix for the monthly transitions between states are given in Tables 8, 9, and 10 for subprime, prime, and all mortgages, respectively. The state transition matrix records the empirical frequency of the different types of transitions between states. For the calculation of these transition matrices and the remainder of the paper, REO and paid off are treated as absorbing states.[8] That is, we stop tracking the mortgage after the first time it enters REO or paid off. The transition matrices highlight that mortgages frequently transition back and forth between current and various delinquency states. Disruptions in cashflow to the lender or servicer are common due to the mortgage being behind payment. Similarly, even loans that are extremely delinquent may return to current; the transition from foreclosed back to current is actually a relatively frequent occurrence.[9] A foreclosure could get cured via paying the outstanding balance, there could be a pre-auction sale that covers all or some of the amount outstanding, or there could be a sale at the foreclosure auction that covers all or some of the amount outstanding. Any of these will register as a foreclosure to paid off transition. Mortgages can also transition directly from current, 30 days delinquent, 60 days delinquent, or 90+ days delinquent to REO via a "deed in lieu of foreclosure".[10]

[7] A foreclosed loan sold at auction may or may not be sold for a loss. The CoreLogic dataset makes no distinction between the two events.

[8] In some states in the USA there are laws that allow the mortgage borrower to reclaim their mortgage even after it has entered REO. However, such events are exceedingly rare.

[9] Many servicers follow a "dual path servicing approach" where they foreclose on the borrower as a threat in order to force the borrower to become current on payments.

[10] A "deed in lieu of foreclosure" is when the loan is in default and the borrower gives ownership of the property directly to the lender, thereby forgoing foreclosure.

2.4 Nonlinear Effects

The relationships between state transition rates and explanatory variables (i.e., loan-level features and economic factors) are often highly nonlinear. For instance, Figure 1 shows the paid off rate versus the FICO score. Figure 2 shows the paid off rate versus the "incentive to prepay", initial interest rate minus national mortgage rate.[11] When the mortgage interest rate is above the available mortgage rate, there is an incentive to prepay. Figure 3 shows the paid off rate versus the time since origination. Several spikes in the prepayment rate occur at $1, 2$, and 3 years due to ARM and hybrid mortgages having rate resets. These spikes can occur due to the expiration of prepayment penalties. Many of the subprime mortgages also started with low teaser rates and would later jump to higher rates; borrowers would refinance to avoid these rate jumps. Figure 4 plots the paid off rate versus the LTV ratio. Each of these plots displays significant nonlinear relationships between the loan covariate and the paid off rate. In Section 3 below, we propose a state transition model that can capture these relationships. As results later in the paper show, modeling nonlinear relationships can in particular lead to significantly improved predictions for prepayments.

3 Model Framework

We propose a dynamic model for the performance of a pool of mortgage loans over time. We adopt a discrete-time formulation for periods $0, 1, \ldots, T$ (e.g., months).[12] We enumerate the possible mortgage states (current, 30 days delinquent, etc.), and let $\mathcal{U} \subset \mathbb{N}$ denote the set of these states. The variable $U_t^n \in \mathcal{U}$ prescribes the state of the n-th mortgage at time t after origination. A mortgage will transition between the various states over its lifetime. For instance, a trajectory of the state process might be:

$$U_0^n = 1 \text{ (current)}, \quad U_1^n = 2 \text{ (30 days late)}, \quad U_2^n = 1 \text{ (current)}, \quad U_3^n = 5 \text{ (paid off)}.$$

We allow the dynamics of the state process to be influenced by a vector of explanatory variables $X_t^n \in \mathbb{R}^{d_X}$ which includes the mortgage state U_t^n. In our empirical implementation, X_t^n represents the original and contemporary loan-level features in Tables 2 and 6, and the contemporary local and national economic factors in Table 7.[13] We specify a probability transition function $h_\theta : \mathcal{U} \times \mathbb{R}^{d_X} \to [0, 1]$ satisfying

$$\mathbb{P}[U_t^n = u \,|\, \mathcal{F}_{t-1}] = h_\theta(u, X_{t-1}^n), \qquad u \in \mathcal{U}, \tag{1}$$

where θ is a parameter to be estimated. Equation (1) gives the marginal conditional probability for the transition of the n-th mortgage from its state U_{t-1}^n at time $t-1$ to state u at time t given the explanatory variables X_{t-1}^n. The family of conditional probabilities give

[11] A more accurate proxy for the incentive to prepay would be the *current interest rate* minus the mortgage rate. However, a large portion of the mortgages in the dataset are missing the current interest rate so the initial interest rate was used instead to achieve greater coverage.

[12] We fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and an information filtration $(\mathcal{F}_t)_{t=0,1,\ldots,T}$.

[13] As usual, categorial variables are encoded in terms of indicator functions (dummy variables).

a conditional transition probability matrix, which is the conditional counterpart of the empirical transition matrix reported in Table 10. Note that the conditional probabilities will be correlated across loans if X_{t-1}^n includes variables that are common to several loans. This formulation allows us to capture loan-to-loan correlation due to geographic proximity and common economic factors.

We propose to model the transition function h_θ by a *neural network*. Let g denote the standard softmax function:

$$g(z) = \left(\frac{e^{z_1}}{\sum_{k=1}^K e^{z_k}}, \ldots, \frac{e^{z_K}}{\sum_{k=1}^K e^{z_k}} \right), \quad z = (z_1, \ldots, z_K) \in \mathbb{R}^K, \tag{2}$$

where $K = |\mathcal{U}|$.[14] The vector output of the function g is a probability distribution on $\mathcal{U}$. The specification $h_\theta(u, x) = (g(Wx + b))_u$, where $W \in \mathbb{R}^K \times \mathbb{R}^{d_X}$, $b \in \mathbb{R}^K$, and V_u is the u-th element of the vector V, gives a logistic regression model. Here, the link function g takes a *linear* function of the covariates x as its input. The output $h_\theta(u, x)$ varies only in the constant direction given by W. A standard approach to achieve a more complex model with greater flexibility is to replace x in the specification with a nonlinear function of x. Let $\phi : \mathbb{R}^{d_X} \to \mathbb{R}^{d_\phi}$ and set $h_\theta(u, x) = (g(W\phi(x) + b))_u$, where $W \in \mathbb{R}^K \times \mathbb{R}^{d_\phi}$ and $b \in \mathbb{R}^{d_\phi}$. This is a logistic regression of the basis functions $\phi = (\phi_1, \ldots, \phi_{d_\phi})$. For instance, polynomials, step functions, or splines could be chosen as the basis functions. A neural network has a similar specification, except that the basis functions are not fixed ahead of time. Instead, these functions are chosen via learning a parameterized function ϕ_θ using the data. Thus, a neural network can be seen as a logistic regression in the nonlinear transformation ϕ_θ, where ϕ_θ must also be learned using the data.

Define the nonlinear transformation $\phi_\theta(x)$ as $h_{\theta,L-1}(x)$. A multi-layer neural network repeatedly passes linear combinations of learned basis functions through simple nonlinear link functions to produce a highly nonlinear function. Formally, the output $h_{\theta,l} : \mathbb{R}^{d_X} \to \mathbb{R}^{d_l}$ of the l-th layer of the neural network is:

$$h_{\theta,l}(x) = g_l(W_l^\top h_{\theta,l-1}(x) + b_l), \tag{3}$$

where $W_l \in \mathbb{R}^{d_l} \times \mathbb{R}^{d_{l-1}}$, $b_l \in \mathbb{R}^{d_l}$, and $h_{\theta,0}(x) = x$. For $l = 1, \ldots, L-1$, the nonlinear transformation $g_l(z) = (\sigma(z_1), \ldots, \sigma(z_{d_l}))$ for $z = (z_1, \ldots, z_{d_l}) \in \mathbb{R}^{d_l}$ and $g_L(z)$ is given by the softmax function $g(z)$ defined in (2). Note that $d_L = K = |\mathcal{U}|$. The function $\sigma : \mathbb{R} \to \mathbb{R}$ is a simple nonlinear link function; typical choices are sigmoidal functions, tanh, and rectified linear units (i.e., $\max(x, 0)$). The final output of the neural network is given by:

$$h_\theta(u, x) = (h_{\theta,L}(x))_u = (g(W_L^\top h_{\theta,L-1}(x) + b_L))_u. \tag{4}$$

The parameter specifying the neural network is

$$\theta = (W_1, \ldots, W_L, b_1, \ldots, b_L), \tag{5}$$

[14]Certain transitions are not allowed in the dataset (e.g., current to 60 days delinquent). Although such a transition is theoretically allowed in the formulation (2), the transition probabilities of transitions which do not occur in the dataset will be driven to zero during training.

where L is the number of layers in the neural network. At each layer l, the output $h_{\theta,l}(x)$ is a simple nonlinear link function g_l of a linear combination of the nonlinear basis functions $h_{\theta,l-1}(x)$, where the nonlinear basis function $h_{\theta,l-1}(x)$ must be learned from data via the parameter θ. The output $h_{\theta,l}(x)$ from the l-th layer of the neural network becomes the basis function for the $(l+1)$-th layer.

The layers between the input at layer $l = 0$ and the output at layer $l = L$ are referred to as the hidden layers. Thus, the neural network h_θ has $L - 1$ hidden layers. A neural network with zero hidden layers ($L = 1$) is a logistic regression model. More hidden layers allow for the neural network to fit more complex patterns. Each subsequent layer extracts increasingly nonlinear features from the data. Early layers pick up simpler features while later layers will build upon these simple features to produce more complex features. The l-th layer has d_l outputs where each output is an affine transformation of the output of layer $l - 1$ followed by an application of the nonlinear function σ. This composition of functions is called a hidden unit, or simply, a unit, since it is the fundamental building block of neural networks. The number of units in the l-th layer is d_l and the complexity of any layer (and the complexity of the features it can extract) increases with the number of units in that layer. Thus, increased complexity can be achieved by increasing either the number of units or the number of layers. Given enough units, a neural network can approximate arbitrarily well continuous functions on compact sets (Hornik 1991). This of course includes approximating arbitrarily well interactions such as the product and division of features. The advantage of more layers (as opposed to simply adding more units to existing layers) is that the later layers learn features of greater complexity by utilizing features of the lower layers as their inputs. Neural networks with three or more hidden layers are considered *deep neural networks*. The number of layers and the number of neurons in each layer, along with other hyperparameters of the neural network, are chosen by the standard approach of cross-validation. Section 4.4 provides the details.

(1) is a dynamic model and therefore gives transition probabilities between the states over multiple periods (2 month, 6 month, 1 year, etc.). The transition probability matrix for 1-month ahead transitions is specified by the transition function in (1). The transition probability matrix for t-months ahead is simply the expectation of the product of the transition probability matrices at months $0, 1, \ldots, t-1$. Note that the transition probability matrices at months $t = 1, 2, \ldots, t-1$ are random due to their dependence on the random covariates X_t^n. To compute these expectations, a time-series model for X_t^n needs to be formulated and Monte Carlo samples from X_t^n need to be generated. This approach is implemented in Section 5. An alternate assumption, which is advantageous for reducing the computational burden and can be accurate for shorter time horizons, is that the economic covariates in X_t^n are frozen at $t = 0$. That is, only the state of the mortgage and deterministic elements of X_t^n (e.g., the balance of a fixed rate mortgage and time to maturity) are allowed to evolve over time. Then, the transition probability matrix for a horizon $t > 1$ is the product of the *deterministic* transition probability matrices at months $0, 1, \ldots, t-1$. This computation is extremely fast because there is no need for any Monte Carlo samples. We implement this approach for predictions at a 1 year time horizon in Section 5 and the method performs well

out-of-sample. This is not surprising since many economic covariates do not vary significantly over a year long period.

An alternative approach to the dynamic model (1) is to train a model for each different time horizon. Many static models could be trained for each of the time horizons (1 month, 2 months, 6 months, 1 year, 1.5 years, 2 years, etc.). Training so many models is computationally expensive and the two approaches mentioned above do not incur this cost.

Our formulation captures loan-to-loan correlation due to geographic proximity and common economic factors. Pool-level quantities, such as the distribution of the prepayment rate for a given pool, can also be computed via standard Monte Carlo simulation. The cashflow from a pool is the sum of the cashflows from the individual loans. Thus, one simply needs to simulate all of the individual loans as described above and then aggregate the individual cashflows. Large portfolios can be rapidly simulated using methods from Sirignano & Giesecke (2015). Fast optimal selection of mortgage portfolios can be performed using methods from Sirignano et al. (2016). If the economic covariates are frozen at time $t = 0$, the pool-level distribution can be approximated in closed-form via a Poisson approximation or the central limit theorem. Such approximations are accurate (for the distribtion where covariates are frozen) even for pools with only a few hundred loans.

It is important to note that we have chosen a specific architecture for the neural network h_θ; there are alternative neural network architectures that could have been selected. For instance, one could individually model transitions from each particular initial state with a neural network; such an approach would require fitting K different neural networks. Another approach would be to have separate models for each product (fixed-rate vs. ARM, etc.) or borrower class (prime vs. subprime). Clearly, our neural network architecture is more parsimonious, which is a desirable characteristic. However, in addition to parsimony, there is a statistical motive for our architecture. Neural networks learn via their hidden layers recognizing, and abstracting, nonlinear features from the data (i.e., nonlinear functions of the initial input). Different transitions may strongly depend upon the same nonlinear features. Similarly, different types of products are likely to depend on some of the same nonlinear features. For instance, it is likely that there are many similar factors driving the transitions current $\to$ paid off and 30 days delinquent $\to$ paid off. In our neural network architecture, all transitions are modeled by the same neural network, which has the advantage that more data can be used to better estimate the nonlinear factors which drive multiple types of transitions.

A typical approach in industry is to use a logistic regression with handcrafted (potentially nonlinear) features. A "handcrafted feature" is some transformation of the original explanatory variables motivated by empirical observation.[15] The neural network has some advantages over such an approach. A neural network model eliminates the need to handcraft features. Given sufficient data (which is in abundance for mortgages), a neural network can learn any continuous function arbitrarily well. With the large number of explanatory variables, checking every possible nonlinear combination of features is challenging if not im-

[15]Development of handcrafted features require significant domain expertise and are generally tightly-guarded by the users. This makes it difficult to directly compare our neural network against such proprietary models.

possible. Therefore, it is likely that important relationships will be missed by the handcrafted approach whereas the neural network will identify these relationships. The neural network could also be used to identify potential nonlinear relationships which could be used by an analyst to develop handcrafted nonlinear features.

Finally, a logistic regression with standard basis function (e.g., polynomials) could also be used. The advantage of a neural network is that it typically has a much sparser representation. A logistic regression with basis functions will need many basis functions with many nonzero coefficients. Deep neural networks typically need exponentially fewer units than shallow neural networks or logistic regressions with basis functions; see Bengio & LeCun (2007) and Montufar, Pascanu, Cho & Bengio (2014).

4 Likelihood Estimation

This section discusses the estimation of the parameter (5) specifying the neural network by the method of maximum likelihood.

4.1 Estimators

We wish to estimate the parameter θ in (5) given observations of $X_t = (X_t^1, \dots, X_t^N)$ at each time $t = 0, \dots, T$ where N is the number of mortgages which are observed. We let $X_t^n = (U_t^n, L_t^N, V_t^n)$, where $U_t^n \in \mathcal{U}$ is the state of the n-th mortgage, and the vector L_t^n includes the lagged default and prepayment rates in the zip code of the n-th mortgage.[16] The vector $V_t^n \in \mathbb{R}^{d_Y}$ includes the remaining contemporary local and national economic factors in Table 7, as well as the original and contemporary loan-level features in Tables 2 and 6. We assume that the variables V_t^n are exogenous in the sense that the law of $V = (V_0, \dots, V_T)$, where $V_t = (V_t^1, \dots, V_t^N)$ does not depend on the parameter θ specifying the law of the observed mortgage states $U = (U_0, \dots, U_T)$, where $U_t = (U_t^1, \dots, U_t^N)$. Therefore, the likelihood problem for V can be treated separately from that for U.

Although the model framework (1) is a dynamic model where the function h_θ may assume a very complicated form, the likelihood problem for the observed states U is very tractable. The likelihood of U depends only on the observed value of V and is independent of V's exact form or parameterization since V is exogenous. Letting $L = (L_0, \dots, L_T)$ and writing informally, the log-likelihood function for θ given V is

$$\mathcal{L}_{T,N}(\theta) = \log \mathbb{P}_\theta\big[U, L|V\big] = \log \mathbb{P}_\theta\big[L|U, V\big]\mathbb{P}_\theta\big[U|V\big] = \log \mathbb{P}_\theta\big[U|V\big],$$

where we use the fact that $\mathbb{P}_\theta\big[L|U, V\big] = 1$ since $L_t = (L_t^1, \dots, L_t^N)$ is a deterministic function of $U_0, \dots, U_t$. Under the standard assumption that the variables $U_t^1, \dots, U_t^N$ are conditionally

[16] In general, L_t^n could include any variables describing the aggregate lagged performance of the mortgages.

independent given X_{t-1}, we have

$$\begin{aligned}\mathcal{L}_{T,N}(\theta) &= \sum_{t=1}^{T} \log \mathbb{P}_\theta\big[U_t \big| U_{t-1}, V_{t-1}\big] \\ &= \sum_{t=1}^{T}\sum_{n=1}^{N} \log \mathbb{P}_\theta\big[U_t^n \big| U_{t-1}^n, V_{t-1}^n\big] \\ &= \sum_{n=1}^{N}\sum_{t=1}^{T} \log h_\theta(U_t^n, X_{t-1}^n).\end{aligned}$$

Note that the above expression implicitly assumes that every mortgage is originated at time $t = 0$. The modification for the case where mortgages have different origination dates is straightforward.

A maximum likelihood estimator (MLE) $\theta_{T,N}$ for the parameter θ satisfies

$$\theta_{T,N} \in \arg\max_{\theta\in\Theta} \mathcal{L}_{T,N}(\theta). \tag{6}$$

The asymptotic properties of the MLEs have been studied before. Under certain conditions, the estimators are consistent and asymptotically normal; see White (1989*a*) and White (1989*b*). Identifiability of neural networks has also been studied before in Sussmann (1992) and Albertini & Sontag (1993).

4.2 Addressing Overfitting

Neural networks tend to be low-bias, high-variance models. We use several methods to address overfitting, including regularization, dropout, and ensemble modeling. A standard ℓ^2 regularization term is included in the objective function in addition to the log-likelihood $\mathcal{L}_{T,N}(\theta)$. The ℓ^2 term represents the sum of the squares of the parameters, thereby giving preference to models with smaller parameters.

Secondly, we use dropout in each of the layers. Dropout is a widely-used technique in deep learning that has proven to be very successful; see Srivastava, Hinton, Krizhevsky & Sutskever (2014). During fitting, hidden units are randomly removed from the network. This prevents complex "fictitious" relationships forming between different neurons since neuron i cannot depend upon neuron j being present.

Finally, we also build an ensemble of neural networks. This simply means that we fit a set of randomly initialized neural networks on datasets obtained by bootstrapping from the original datasets. Typically, each neural network reaches a different local minimum due to each being trained with a different random initial starts and random sequence of bootstrapped samples. Variance (i.e., overfitting) of an individual neural network's prediction can be reduced by taking the prediction as the average of the neural networks' predictions. The averaged prediction, or ensemble prediction, has lower variance since the idiosyncratic variance for each neural network is averaged out.

4.3 Computational Implementation

There are significant computational hurdles to training models due to the large size of our dataset as well as due to the size of the deep neural networks. The dataset includes the individual characteristics of each loan as well as monthly updates on loan performance. We include 294 features for each mortgage. Since our models are dynamic, there is a sample for each month of data. In total, we train over roughly 3.5 billion samples, which is almost 2 terabytes of data, and only a fraction of the dataset can be loaded into (RAM) memory at any one time. Further, many of the deep neural networks contain tens of thousands of free parameters. Estimating these parameters in order to fit the model requires computing the gradients using backpropagation, which is a time and memory intensive procedure. Fitting just one such model on our data using typical computing resources (e.g., using MATLAB or R on a desktop with conventional CPU) would require weeks of training time, which makes fitting and iterating through models impractically slow. In contrast, we train up to 10 models simultaneously in a span of few days. This is made possible by using several tools that harness both optimized hardware as well as computational tricks, which we describe in the remainder of the section.

While training, every data sample undergoes the same series of transformations through the layers of the neural network, which makes the procedure very amenable to parallelization. Accelerated training can be achieved by employing Graphics Processing Units (GPU) which enable performing several thousand simple operations, such as matrix multiplication, simultaneously. We harness the power of GPUs, which provide more than a 10x speedup over CPU, to address the problem of a large dataset. Moreover, to iterate faster it is important to be able to train multiple models simultaneously. Therefore, we set up a cluster computing environment where each model is trained independently on individual nodes (powered by GPUs) and all nodes have access to a central data server. This avoids the need for replicating data on individual nodes and enables efficient training. We achieve this practically by using Amazon Web Services (AWS), which is a cloud computing platform that allows flexible scaling of compute resources. In our implementation, we use up to 10 single-GPU nodes, where each GPU contains 1,536 CUDA cores and 4 GB of memory. The bandwidth of the central data server allows up to 15 nodes to fetch data simultaneously to train their models.

There are several other software optimizations that help make the training faster. We use a specialized deep learning library Torch, which has been developed by Facebook and Google and uses the Lua programming language. Such specialized libraries optimize the commonly used operations for neural networks and have fast routines written in C that speed up training by an order of magnitude. Further, we use single precision floating point operations (instead of double precision) throughout our code. This has no practical effect on the parameter estimates and it halves the memory requirements and leads to substantial speed up in the computations.

Gradient descent for fitting models is impractical due to the size of the data. We use the standard machine learning method of minibatch gradient descent with momentum; see Ngiam, Coates, Lahiri, Prochnow, Le & Ng (2011) for a discussion of minibatch gradient descent for deep neural networks. In minibatch gradient descent, gradient steps are sequentially taken using subsets of the dataset. A block of the data is loaded into memory and the gradient of the objective function is calculated on this block of data. A step is then taken in the direction of the minibatch gradient with the step size determined by the "learning rate". Another block of data is then loaded into memory and the process repeats. The size of this block of data, referred to as the batch size, and the learning rate are optimized in order for fast (but stable) convergence of the model parameters.

In order that the minibatch gradients are unbiased, blocks of data must be drawn at random from the entire dataset. If gradients are biased, training may not converge and accuracy may be lost. A typical issue with mortgage and other types of loan data is that it is not stored randomly, but instead split into categories such as geographic region, time period, and loan type. Due to the size of the data, randomly scrambling the data can be computationally challenging.[17]

4.4 Hyperparameter Selection

Neural networks have a number of hyperparameters which need to be chosen. The standard approach to choosing these hyperparameters is to cross-validate them via a validation set. We train neural networks with different hyperparameters on the training set and compare the log-likelihood on the validation set. In particular, we cross-validate the number of layers and number of neurons per layer. The more layers and more neurons, the more complex the neural network is and the better able it is to fit complex relationships. However, with more complexity, there is also a higher chance of overfitting. We also cross-validate the size of the ℓ^2 penalty, the learning rate schedule, batch size, and the type of nonlinearity via a sparse grid search. For each grid point, we train a neural network and record its validation error. We then choose the hyperparameters at the grid point with the lowest validation error.

Several learning rate schedules were tested. The learning rate schedule is critical for training the neural network. If the learning rate is too high, there can be large oscillations that may drive the estimates away from the optimal value. If the learning rate is too low, the neural network will learn very slowly. The chosen learning rate schedule is:

$$\text{Learning rate} \quad = \quad \frac{LR_0}{1 + t/800}, \tag{7}$$

[17]The original CoreLogic dataset needs to be reorganized for model fitting. The original dataset provided by CoreLogic is divided into static data (origination features) and dynamic data (monthly loan performance). The static data itself is divided into separate geographic regions (e.g., Pacific, Northeast, Southeast). The dynamic data is divided into geographic regions and then into months. In order to create a training sample, one has to match the static data for a loan with all of the monthly updates for that loan in the dynamic data files. In addition, one has to randomly order the training samples such that there is no bias towards a particular origination time or geographic location. Matching static data with the dynamic data via a search through these different subsets is impractical due to the size of the dataset. In order join the static and dynamic data, we create a hash table whose keys are the loan IDs and whose values are the destination folders $1, \ldots, L$ (randomly chosen). This hash table is used to randomly distribute the loans to the folders. Secondly, we use another hash table to match static data with dynamic data with these destination folders in order to avoid a search.

where the initial learning rate $LR_0 = 0.1$ and t is the epoch number. The half-life (i.e., the number of epochs until the learning rate is reduced by half) is 800. Each epoch contains approximately 1.5 million training samples. The batchsize is $4,000$, meaning approximately 375 gradient steps are taken per epoch.

Cross-validation leads to the choice of 5 hidden layers, with 200 units in the first hidden layer and 140 units in each subsequent one. The rectified linear unit (ReLU) nonlinearity (i.e., $\sigma(x) = \max(0, x)$) was found to yield better performance and faster convergence than the sigmoid nonlinearity ($\sigma(x) = 1/(1 + e^{-x})$). Besides this optimal choice of hyperparameters from the validation set, we also test some other configurations out-of-sample for comparison purposes. This includes shallower neural networks (1 and 3 hidden layers) as well as deeper neural networks (7 or more hidden layers).

5 Empirical Results

This section reports the out-of-sample performance for the models. Several performance metrics are considered for both loan-level and pool-level model performance. The neural network significantly outperforms the logistic regression according to each of these metrics. In particular, the out-of-sample negative log-likelihood for the neural network is significantly lower than for the logistic regression. Since the log-likelihood was the objective function for training and also directly measures how well the model fits the empirical distribution, it is the best metric for comparing model performance. However, the log-likelihood is not very interpretable, so we consider several other metrics to further compare model performance. These metrics include Receiver Operating Characteristic curves and pool-level accuracy. We also develop a "loan-ranking metric" which has a very natural interpretation as an investment portfolio. Section 5.1 reports the out-of-sample loan-level performance for the models. Section 5.2 reports the out-of-sample pool-level performance for the models. Section 5.3 provides interpretation for which features are most important.

The training set includes all the data before May 1, 2012. The validation set, which is used for the selection of hyperparameters (see Section 4.4), is May 1, 2012 until October 31, 2012. Once the hyperparameters are chosen, the model is re-fitted on the combined training and validation sets. The final trained model is then tested out-of-sample on the test set, which is from November 1, 2012 until May 31, 2014. All covariates are normalized by their means and standard deviations (which are calculated using data only from the training set).

5.1 Out-of-sample loan-level performance

Model parameters were trained to minimize the negative log-likelihood. Table 14 reports the in-sample and out-of-sample negative log-likelihood for the different models. We take logistic regression as the baseline model. Neural networks with 1, 3, 5, and 7 hidden layers are tested. The neural networks strongly outperform logistic regression, reducing the negative log-likelihood by roughly 8%. Deep neural networks outperform the shallow neural network (with only 1 hidden layer), reducing the negative log-likelihood by roughly 1%. The ensemble of deep neural networks performs the best, reducing the error of any single deep neural network by roughly 1%. The ensemble of deep neural networks is composed of eight 5-layer neural networks. Figure 5 shows the out-of-sample negative log-likelihood versus the the number of neural networks in the ensemble. Including just eight neural networks in the ensemble significantly improves out-of-sample performance. Although larger ensembles lead to marginal improvements in performance, the computational cost (which increases linearly with the number of neural networks used) may not justify using larger ensembles in practice. Henceforth, we only consider ensembles of eight independently trained neural networks. The neural networks especially outperform the logistic regression for predicting the transition to paid off (largely due to prepayments). Prepayment is known to have nonlinear dependencies (see Section 2.4). The transition to paid off is also the most difficult transition to predict.

We next consider Receiver Operating Characteristic (ROC) curve and Area Under Curve (AUC), which are standard measures of predictive performance for a binary classifier. Assume that the binary classifier generates an estimate of the probability that the input sample belongs to the positive class. The ROC curve plots the true positive rate versus the false positive rate as the discriminative threshold is varied between 0 and 1. The AUC is the area under the ROC curve and a higher value shows an improved ability of the classifier to discern between the two classes. Alternatively, the AUC can be interpreted as the probability that the model generates a larger value for a sample randomly chosen from the positive class than for a sample randomly chosen from the negative class. Table 15 reports the in-sample and out-of-sample AUC for the ROC curve for the state of the mortgage (paid off, current, 30 days delinquent, 60 days delinquent, 90+ days delinquent, foreclosure, and REO) at a 1-month horizon. Specifically, we define the AUC for state u to be the AUC for the two-way classification of whether the mortgage is in state u or not in state u at a 1-month horizon. The neural networks outperform logistic regression for all states, with the most pronounced improvements observed for paid off, current, and 30 days delinquent. The AUCs for neural networks are more than 10% greater for paid off. Deep neural networks outperform the shallow neural network. The deep neural networks have roughly 1.5% greater AUC for paid off than the shallow neural network. The ensemble of deep neural networks outperforms a single deep neural network with a 1% greater AUC for paid off.

We now examine performance at a more granular level. The previous AUCs were only for the terminal outcome of the mortgage. Model performance for each possible type of state transition can be measured via the AUCs for state transitions. Specifically, the AUC for event $u \to v$ is the AUC for the two-way classification of whether the mortgage is in state v or not in state v at a 1-month horizon conditional on the mortgage currently being in state u. The state transition AUC can be useful for determining model performance for transitions of interest. For instance, a servicer may be interested in whether a mortgage which is 30 days delinquent will become even more delinquent (60 days delinquent) or return to current.

Figures 6, 7, 8, 9, and 10 report the out-of-sample AUCs for state transitions for the logistic regression, 1-layer neural network, 3-layer neural network, 5-layer neural network, and ensemble of neural networks. They give a complete picture of model performance in different states and for different types of transitions. The 1-layer neural network outperforms logistic regression, the deep neural network outperforms the 1-layer neural network, and the ensemble of deep neural networks outperforms the single deep neural network. We highlight that the AUC for the transition current $\to$ 30 days delinquent is surprisingly high across all models (.87-.89), indicating the models are able to identify early financial distress in the mortgages. The full ROC curves for the transitions current $\to$ current and current $\to$ paid off are provided in Figures 11 and 12, respectively. In the ROC curves, the neural networks strictly dominate the logistic regression. The deep neural networks strictly dominate the shallow neural network and the ensemble strictly dominates all other models.

The use of dropout significantly improves out-of-sample performance for neural networks with greater depths. Figure 13 shows that dropout provides effective regularization and allows fitting of neural networks with larger numbers of layers. Figure 13 also indicates the behavior of the test loss as a function of the depth of the neural network. Deeper neural networks do not always yield better performance due to higher model capacity; there are several other factors at play, such as over-fitting and difficulty in training the model, that affect the out-of-sample performance. We observe that the best neural networks for this problem have between 3 and 5 hidden layers. See Goodfellow, Bengio & Courville (2016) for an excellent discussion of the optimal depth and number of hidden units.

5.2 Out-of-sample pool-level performance

An investor desires a loan portfolio with uninterrupted cashflow. Delinquency often produces a loss of cashflow while prepayments lead to early cashflows that might have to be reinvested at lower interest rates. Uninterrupted cashflows require loans which are both unlikely to be delinquent and unlikely to prepay. This is equivalent to a portfolio of loans which are highly likely to remain current. An example is a bank which originates loans and retains some loans on their balance sheet. Another example is an investment fund or structurer who constructs a loan portfolio. Given an available pool of loans to select a portfolio from, an investor can rank those loans by the likelihood that they remain current according to their model of choice. For a portfolio of size N, the investor will choose the N loans with the highest probabilities of remaining current.

The above approach can also be used as a metric to measure model performance, and we refer to it as a "loan ranking metric." We form two portfolios of N loans from an available pool of 100,000 mortgages (randomly chosen from the entire dataset). The first portfolio is chosen using the neural network and the second portfolio is chosen using the logistic regression. A portfolio is chosen by ranking the available loans according to their probability of being current as predicted by the model and then choosing the N loans mostly likely to be current.[18] Figures 14 and 15 show how the 5-layer neural network portfolio and logistic regression portfolio perform out-of-sample over 1 month and 1 year time horizons, respectively. The neural network portfolio significantly outperforms the logistic regression portfolio in terms

[18]The same approach can be used to rank loans according to other criteria. For instance, if one wanted to account for both the interest rate and the risk of the loan, the expected return for each loan could be calculated for each model. Then, the loans could be ranked according to their expected returns.

of delinquency and prepayment rates. Table 11 reports the percent of the 5-layer neural network and logistic regression portfolios with size $N = 20,000$ in each state (REO, paid off, etc.) at a 1 month time horizon. Table 12 reports the percent of the 5-layer neural network and logistic regression portfolios with size $N = 20,000$ in each state (REO, paid off, etc.) at a 1 year time horizon. At a 1 year time horizon, the neural network portfolio has a significantly lower prepayment and foreclosure rate than the logistic regression portfolio. The neural network portfolio also has a larger current rate than the logistic regression portfolio. The 1 year transition probabilities are produced using the method described in Section 3 where the time-varying covariates (e.g., unemployment rates) are frozen.

The outperformance of the neural network portfolio in Table 12 directly translates into improved profit and loss for a lender or MBS investor. We conservatively assume that prepayment results in a loss of 5% of notional, foreclosure and REO produce losses of 40% of notional, and m months delinquent leads a loss of $\frac{m}{360} \times 100\%$ of notional. The neural network portfolio has a 46% smaller loss than the logistic regression portfolio at a 1 year time horizon.

An important application of loan-level models is predicting delinquency and prepayment rates at the *pool-level*. For instance, pool-level predictions are necessary for the risk analysis of mortgage-backed securities and other mortgage portfolios held by financial institutions. The expected delinquency rates, prepayment rate, and return of a mortgage pool is easily calculated by simply summing the expectations of the individual mortgages. We compare the pool-level predictions of the logistic regression model and the 5-layer neural network for 2,000 pools created from 2 million mortgages in our test dataset. Each pool contains 1,000 mortgages. Pools are created by rank ordering the loans according to some metric (e.g., the interest rate) and then sequentially placing the loans in pools of size 1,000.[19] This produces pools with varying levels of risk. Four cases are examined. We create pools by rank ordering according to FICO score, interest rate, LTV ratio, and the logistic regression's predicted probability that the loan is current. Figures 16, 17, 18, and 19 compare the logistic regression pool-level prediction with the neural network pool-level prediction for a 1 year time horizon. The 1 year transition probabilities and the pool-level prediction are produced using the method described in Section 3 where the time-varying covariates are frozen. The diagonal line represents a perfect prediction. The neural network prediction is much more accurate than the logistic regression prediction.

The logistic regression's pool-level predictions in Figures 16, 17, 18, and 19 appear to be systematically biased upwards (i.e., above the diagonal line). This systematic bias is due to the predictions for all pools depending upon the same covariates. For instance, if one prediction is biased upwards due to the realized value of the national mortgage rate, it is likely that all pool predictions will be biased upwards since they all depend upon the same realization of the national mortgage rate.

One can also generate the pool-level distribution by simulating the time-varying covariates forward in time, as described in Section 3. Fitting and simulating all the time-varying covariates can be computationally challenging (there is a large number of zip code level vari-

[19] The loans with the top thousand highest interest rates are placed in the first pool, loans with the 1001-2000th highest interest rates are placed in the second pool, etc.

ables), so it may be advantageous to freeze some of the variables and only simulate the most important ones. The variables to be simulated could be chosen using the variable ranking methods described in Section 5.3 below. This approach where time-varying variables are simulated forward in time, unlike the approach where all the variables are frozen, accounts for loan-to-loan correlation due to common exposure to the national mortgage rate. Modeling this correlation is important for the risk analysis of mortgage portfolios.

Figures 20 and 21 show the pool-level distribution for the 5-layer neural network and logistic regression for several different pools. Each pool has 10,000 mortgages and the time horizon is 1 year. The national mortgage rate was simulated forward and all other time-varying covariates were frozen. The national mortgage rate was fitted to historical mortgage rate data using an autoregressive model.[20] The simulated trajectories of the national mortgage rate using the autoregressive model are shown in Figure 22. Note that the actual observed prepayments in Figures 20 and 21 falls in the center of the neural network-produced distribution while it falls in the tail of the logistic regression-produced distribution. Thus, in these cases, the neural network-produced distribution more accurately captured the out-of-sample outcome. To confirm the superior performance of the neural network, we consider 50 test portfolios with 10,000 mortgages each obtained by slicing a pool of 500,000 mortgages. Table 13 reports average statistics (over the 50 test portfolios) for the distributions produced by the 5-layer neural network and the logistic regression distribution, and gives a sense of the out-of-sample prediction error associated with the two models. The neural network-produced distribution tends to have less variance with the mean of the distribution closer to the observed number of prepayments, thereby predicting the prepayments more accurately.

5.3 Interpretation of Neural Network

Neural networks are complicated functions of the covariates. The role and importance of the various covariates is not immediately obvious as it is in logistic regression. We propose and implement several methods to interpret the fitted neural network.

Table 16 rank orders the covariates according to the neural network's sensitivity to the covariate. This provides information on which covariates are most important. The sensitivity of the neural network to a particular covariate is measured by the magnitude of the derivative of the neural network's output with respect to the covariate (averaged over the data). The derivative is over a representative sample drawn from the dataset. Specifically, we calculate the sensitivity (with respect to j-th input) of the neural network's prediction for a transition from state u to v as:

$$\text{Sensitivity}(u, v, j) = \frac{1}{|M_u|} \sum_{(n,t)\in M_u} \left| \frac{\partial h_\theta(v, x)}{\partial x_j} \right|_{x=X_t^n} \tag{8}$$

where $M_u = \{(n, t) : U_t^n = u, 1 \leq t \leq T, 1 \leq n \leq N\}$ and x_j is the j-th element of x. Here

[20] We use an autoregressive model of order 4. The parameters are $[0.6687, 1.3514, -0.5131, 0.2410, -0.0838]$. The order of the autoregressive model (i.e., the lag) was chosen using the partial autocorrelation plot.

X_t^n and U_t^n are the vector of explanatory variables and the state of mortgage n at time t, respectively. Note that the quantity in (8) aggregates over only the relevant mortgages and times, namely those in M_u: a mortgage n at time t is relevant for computing this sensitivity if it is in state u then. This allows computing the probability that this mortgage attains state v at time $t+1$, which in turn facilitates computing the sensitivity for transition from state u to v. It is important to aggregate over the dataset in order to average the sensitivities computed over precisely those pockets of the high dimensional space $\mathcal{X}$ that are observed in our data (here $\mathcal{X}$ is the space in which X_t^n reside). This approach is preferable to both, either computing the sensitivity at a single representative point $\bar{X} \in \mathcal{X}$, or aggregating over randomly sampled points in $\mathcal{X}$. Table 16 reports results for the case of u = current and v = paid off. The analysis for Table 16 was performed using the 5-layer neural network.

Table 17 rank orders the covariates according to how important each covariate is to the neural network's prediction. Specifically, the importance of a covariate is measured by how much the negative log-likelihood increases when that covariate is removed as an explanatory variable. Covariates that are very important, and whose information is not also largely contained in the other remaining covariates, will produce large increases in the negative log-likelihood if they are removed. The analysis for Table 17 was performed using the 5-layer neural network.

Both Tables 16 and 17 show the importance of local economic variables such as lagged default and prepayment rates, state unemployment, and zip code level housing prices. The results imply that mortgage behavior can vary depending upon the geographic region, and even within the same state. The local dependence also provides insight into the correlation structure of mortgage portfolios. Correlations (at least in the model setting of this paper) are driven by the dependence of mortgages on economic covariates. Two mortgages will be more correlated if they strongly depend upon the same economic covariates. The covariate ranking results indicate that mortgages depend upon local economic covariates, which means two mortgages in the same geographic location will depend upon the same economic covariates. Consequently, correlation between mortgages will be higher if they are geographically close.

6 Conclusion

We have built deep neural networks for modeling mortgage delinquency and prepayment risk using a dataset of over 120 million prime and subprime mortgages. The deep neural network outperforms shallow neural networks as well as a logistic regression model according to multiple metrics. We have embedded the neural network within a dynamic model, which allows for prediction at all monthly time horizons as well as simulation of mortgage pool cash-flows. Cloud computing for data management along with GPUs for accelerated training were used due to the significant computational cost of training our models. The neural network's out-of-sample performance was improved via using methods from deep learning as well as model ensembles. Finally, we develop tools to interpret the neural network by evaluating the importance of each covariate. Along with well-known covariates in the mortgage literature, we find that local economic variables are important. Rating agencies, lenders, servicers, and

MBS investors could all potentially benefit from the proposed modeling approach.

The deep learning approach developed here complements previous work on fast simulation of large loan portfolios in Sirignano & Giesecke (2015) and fast optimal selection of loan portfolios in Sirignano et al. (2016). Simulation and optimization of mortgage portfolios is computationally challenging due to the large sizes typical in practice (e.g., an MBS can have tens of thousands of mortgages). Given the estimated neural network, investors can use the computational methods developed in Sirignano & Giesecke (2015) to rapidly simulate mortgage portfolios for risk analysis purposes. The computational methods in Sirignano et al. (2016) can be used to rapidly select optimal loan portfolios given the underlying neural network model.

References

Agarwal, S., B. Ambrose & Y. Yildirim (2015), 'The subprime virus', *Real Estate Economics* **43**(4), 891–915.

Albertini, F. & E. Sontag (1993), 'For neural networks, function determines form', *Neural Networks* **6**, 975–990.

Anenberg, E. & E. Kung (2014), 'Estimates of the size and source of price declines due to nearby foreclosures', *American Economic Review* **104**(8), 2527–51.

Atiya, A. (2001), 'Bankruptcy prediction for credit risk using neural networks: A survey and new results', *IEEE Transactions on Neural Networks* **12**(4).

Baesens, B. (2005), 'Neural network survival analysis for personal loan data', *Journal of the Operational Research Society* **56**(9), 1089–1098.

Bengio, Y. & Y. LeCun (2007), 'Scaling learning algorithms towards AI', *Large-scale kernel machines* **34**(5).

Butaru, F., Q. Chen, B. Clark, S. Das, A. Lo & A. Siddique (2015), Risk and risk management in the credit card industry. Working Paper, MIT.

Campbell, J., S. Gigli & P. Pathak (2011), 'Forced sales and house prices', *American Economic Review* **101**(5), 2108–2131.

Campbell, T. & J. Dietrich (1983), 'The determinants of default on insured conventional residential mortgage loans', *Journal of Finance* **38**, 1569–1581.

Chernov, M., B. Dunn & F. Longstaff (2016), Macroeconomic-driven prepayment risk and the valuation of mortgage-backed securities. Working Paper, UCLA.

Ciresan, D., U. Meier & J. Schmidhuber (2012), 'Multi-column deep neural networks for image classification', *IEEE Conference on Computer Vision and Pattern Recognition* pp. 3642–3649.

Ciresan, D., U. Meier, L. Gambardella & J. Schmidhuber (2010), 'Deep, big, simple neural networks for handwritten digit recognition', *Neural Computation* **22**(12), 3207–3220.

Ciresan, D., U. Meier, L. Gambardella & J. Schmidhuber (2011), 'Convolutional neural network committees for handwritten character classification', *International Conference on Document Analysis and Recognition* pp. 1135–1139.

Collobert, R. & J. Weston (2008), 'A unified architecture for natural language processing: Deep neural networks with multitask learning', *Proceedings of the 25th International on Machine Learning* pp. 160–167.

Curley, A. & J. Guttentag (1974), 'The yield on insured residential mortgages', *Explorations in Economic Research* **1**, 114–161.

Dahl, G., D. Yu, L. Deng & A. Acero (2012), 'Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition', *IEEE Transactions on Audio, Speech, and Language Processing* **20**(1), 30–42.

Deng, Y. (1997), 'Mortgage termination: An empirical hazard model with a stochastic term structure', *The Journal of Real Estate Finance and Economics* **14**(3), 309–331.

Deng, Y., J. Quigley & R. Van Order (2000), 'Mortgage terminations, heterogeneity, and the exercise of mortgage options', *Econometrica* **68**(2), 275–307.

Dixon, M., D. Klabjan & J.H. Bang (2016), Classification-based financial markets prediction using deep neural networks. Working Paper, Illinois Institute of Technology and Northwestern University.

Duffie, D., L. Saita & K. Wang (2007), 'Multi-period corporate default prediction with stochastic covariates', *Journal of Financial Economics* **83**, 635665.

Episcopos, A., A. Pericli & J. Hu (1998), 'Commercial mortgage default: A comparison of logit with radial basis function networks', *Journal of Real Estate Finance and Economics* **17**(2), 163–178.

Feldman, D. & S. Gross (2005), 'Mortgage default: classification trees analysis', *The Journal of Real Estate Finance and Economics* **30**(4), 369–396.

Fitzpatrick, T. & C. Mues (2016), 'An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market', *European Journal of Operational Research* **249**(2), 427–439.

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

Glorot, X., A. Bordes & Y. Bengio (2011), 'Deep sparse rectifier neural networks', *International Conference on Artificial Intelligence and Statistics* pp. 315–323.

Goodfellow, I., Y. Bengio & A. Courville (2016), Deep learning. Book in preparation for MIT Press.
URL: *http://www.deeplearningbook.org*

Goodstein, R., P. Hanouna, C. Ramirez & C. Stahel (2011), Contagion effects in strategic mortgage defaults. GMU Working Paper in Economics No. 13-07.

Gorovoy, V. & V. Linetsky (2007), 'Intensity-based valuation of residential mortgages: an analytically tractable model', *Mathematical Finance* **17**(4), 541–573.

Graves, A., A. Mohamed & G. Hinton (2013), 'Speech recognition with deep recurrent neural networks', *IEEE International Conference on Acoustics, Speech, and Signal Processing* pp. 6645–6649.

Green, J. & J. Shoven (1986), 'The effects of interest rates on mortgage prepayments', *Journal of Money, Credit and Banking* **18**, 41–59.

Guo, X., S. Singh, H. Lee, R. Lewis & X. Wang (2014), 'Deep learning for real-time atari game play using offline monte-carlo tree search planning', *Advances in Neural Information Processing Systems* pp. 3338–3346.

Harding, J., Eric Rosenblatt & V. Yao (2009), 'The contagion effect of foreclosed properties', *Journal of Urban Economics* **66**, 164–178.

Hasselt, H., A. Guez & D. Silver (2015), Deep reinforcement learning with double q-learning. Google DeepMind.

Heaton, J. B., N. G. Polson & J. H. Witte (2016), Deep learning in finance. Working Paper, University of Chicago.

Hinton, G., L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath & B. Kingsbury (2012), 'Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups', *Signal Processing Magazine, IEEE* **29**(6), 82–97.

Hornik, K. (1991), 'Approximation capabilities of multilayer feedforward networks', *Neural Networks* **4**(2), 251–257.

Hutchinson, J., A. Lo & T. Poggio (1994), 'A nonparametric approach to pricing and hedging derivative securities via learning networks', *Journal of Finance* **49**, 851–889.

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar & L. Fei-Fei (2014), 'Large-scale video classification with convolutional neural networks', *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* pp. 1725–1732.

Khandani, A., A. Kim & A. Lo (2010), 'Consumer credit-risk models via machine-learning algorithms', *Journal of Banking and Finance* **34**(11), 2767–2787.

Krizhevsky, A., I. Sutskever & G. Hinton (2012), 'Imagenet classification with deep convolutional neural networks', *Advances in Neural Information Processing Systems* pp. 1097–1105.

Larochelle, H., Y. Bengio, J. Louradour & P. Lamblin (2009), 'Exploring strategies for training deep neural networks', *The Journal of Machine Learning* **10**, 1–40.

Lillicrap, T., J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver & D. Wierstra (2015), Continuous control with deep reinforcement learning. Google DeepMind.

Lin, Z., E. Rosenblatt & V. Yao (2009), 'Spillover effects of foreclosures on neighborhood property values', *Journal of Real Estate Finance and Economics* **38**(4), 387–407.

Magee, M. (1968), 'Statistical prediction of mortgage risk', *Land Economics* **44**, 461–469.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski & S. Petersen (2015), 'Human-level control through deep reinforcement learning', *Nature* **518**(7540), 529–533.

Montufar, G., R. Pascanu, K. Cho & Y. Bengio (2014), 'On the number of linear regions of deep neural networks', *Advances in Neural Information Processing Systems* pp. 2924–2932.

Moody's Analytics (2013), Portfolio analyzer. Moody's Investors Service.

Ngiam, J., A. Coates, A. Lahiri, B. Prochnow, Q. Le & A. Ng (2011), 'On optimization methods for deep learning', *International Conference on Machine Learning (ICML-11)* pp. 265–272.

Palmer, C. (2013), Why did so many subprime borrowers default during the crisis: Loose credit or plummeting prices? Working Paper, University of California, Berkeley.

Schwartz, E. & W. Torous (1989), 'Prepayment and the valuation of mortgage-backed securities', *The Journal of Finance* **44**(2), 375–392.

Schwartz, E. & W. Torous (1993), 'Mortgage prepayment and default decisions: A poisson regression approach', *Journal of the American Real Estate and Urban Economics Association* **21**(4), 431–449.

Simonyan, K. & A. Zisserman (2014), Very deep convolutional networks for large-scale image recognition. Working Paper, Oxford University.

Sirignano, J. (2016), Deep learning for limit order books. Working Paper, Imperial College London.

Sirignano, J., G. Tsoukalas & K. Giesecke (2016), Large-scale loan portfolio selection. *Operations Research,* forthcoming.

Sirignano, J. & K. Giesecke (2015), Risk analysis for large pools of loans. Working Paper, Stanford University.

Srivastava, N., G. Hinton, A. Krizhevsky & I. Sutskever (2014), 'Dropout: A simple way to prevent neural networks from overfitting', *The Journal of Machine Learning Research* **15**(1), 1929–1958.

Stanton, R. & N. Wallace (2011), 'The bear's lair: Index credit default swaps and the subprime mortgage crisis', *Review of Financial Studies* **24**(10), 3250–3280.

Sussmann, H. (1992), 'Uniqueness of the weights for minimal feedforward nets with a given input-output map', *Neural Networks* **5**(4), 589–593.

Towe, C. & C. Lawley (2013), 'The contagion effect of neighboring foreclosures', *American Economic Journal: Economic Policy* **5**(2), 313–35.

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

Vandell, Kerry D. (1978), 'Default risk under alternative mortgage instruments', *The Journal of Finance* **33**(5), 1279–1296.

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.

von Furstenberg, G. (1970), 'Risk structures and the distribution of benefits within the fha home mortgage insurance program', *Journal of Money Credit and Banking* **2**, 303–322.

Wang, T., D. Wu, A. Coates & A. Ng (2012), 'End-to-end text recognition with convolutional neural networks', *21st IEEE International Conference on Pattern Recognition* pp. 3304–3308.

West, D. (2000), 'Neural network credit scoring models', *Computers and Operations Research* **27**, 1131–1152.

White, Halbert (1989*a*), 'Learning in artificial neural networks: A statistical perspective', *Neural Computation* **1**(4), 425–464.

White, Halbert (1989*b*), 'Some asymptotic results for learning in single hidden-layer feedforward network models', *Journal of the American Statistical Association* **84**(408), 1003–1013.

Product type	Percent of Total	Percent of Subprime	Percent of Prime
Fixed Rate	80.6 %	48 %	86.3 %
ARM	11.7 %	29 %	8.7 %
GPM (graduated payment)	.01 %	0 %	.01 %
Balloon Unknown	.9 %	1 %	.9 %
Balloon 5	.03 %	0 %	.03 %
Balloon 7	.03 %	.004 %	.04 %
Balloon 10	.004 %	.006 %	.004 %
Balloon 15/30	.2 %	1.07 %	.005 %
ARM Balloon	.2 %	1.3 %	.01 %
Balloon Other	.7 %	3.3 %	.26 %
Two Step Unknown	.02 %	0 %	.02 %
Two Step 10/20	.003 %	0 %	.003 %
GPARM	.002 %	0 %	.002 %
Hybrid 2/1	1 %	6.7 %	0 %
Hybrid 3/1	.63 %	2.2 %	.35 %
Hybrid 5/1	1.9 %	.22 %	2.2 %
Hybrid 7/1	.5 %	.005 %	.64 %
Hybrid 10/1	.24 %	.02 %	.28 %
Hybrid Other	.02 %	.02 %	.02 %
Other	.7 %	4.3 %	.01 %
Invalid data	.18 %	.6 %	.11 %

Table 1: Types of mortgages for full dataset, subprime subset, and prime subset.

Feature	**Values**
FICO score	Continuous
Original debt-to-income ratio	Continuous
Original loan-to-value ratio	Continuous
Original interest rate	Continuous
Original balance	Continuous
Original term of loan	Continuous
Original sale price	Continuous
Buydown flag	True, False
Negative amortization flag	True, False
Occupancy Type	Owner occupied, second home, non-owner occupied or investment property, other
Prepayment penalty flag	True, False
Product type	See Table 1
Loan purpose	Purchase, Refinance Cash-out, Refinance No Cash Out, Second mortgage, Refinance Cash Out Unkown, Construction Loan, Debt Consolidation Loan, Home Improvement Loan, Education Loan, Medical Loan, Vehicle Purchase, Reverse Mortgage, Other
Documentation	Full documentation, Low or minimal documentation, No asset or income verification, Other
Lien type	1st Position, 2nd Position, 3rd Position, 4th Position, Other
Channel	Retail Branch, Wholesale Bulk, Mortgage Broker, Realtor Originated, Relocation Corporate, Relocation Mortgage Broker, Builder, Direct Mail, Other Direct, Internet, Other Retail, Mortgage Banker, Correspondent, Other
Loan type	Conventional Loan, VA Loan, FHA Loan, Other Government Loan, Affordable Housing Loan, Pledged Asset Loan, Other
Number of units	1,2,3,4,5
Appraised value $<$ sale price?	True, False
Initial Investor Code	Portfolio Held, Securitized Other, GNMA/Ginnie Mae, GSE
Interest Only Flag	True, False
Original interest rate $-$ national mortgage rate at origination	Continuous
Margin for ARM mortgages	Continuous
Periodic rate cap	Continuous
Periodic rate floor	Continuous
Periodic pay cap	Continuous
Periodic pay floor	Continuous
Lifetime rate cap	Continuous
Lifetime rate floor	Continuous
Rate reset frequency	1,2,3, ... (months)
Pay reset frequency	1,2,3, ... (months)
First rate reset period	1,2,3, ... (months since origination)
Convertible flag	True, False
Pool insurance flag	True, False
Alt-A flag	True, False
Prime flag	True, False
Subprime flag	True, False
Geographic state	CA, FL, NY, MA, etc.
Vintage (origination year)	1995, 1996, ..., 2014

Table 2: Loan-level features at origination (from CoreLogic).

Feature	Mean	Median	Min	Max	25% Quantile	75% Quantile
FICO	634	630	300	900	580	680
Original LTV	74	80	0	200	68	90
Original interest rate	7.8	7.8	0	30	6.3	9.6
Original balance	160,197	124,000	7	318,750	68,850	210,000

Table 3: Summary statistics for some mortgage features in subprime data.

Feature	Mean	Median	Min	Max	25% Quantile	75% Quantile
FICO	720	730	300	900	679	772
Original LTV	74	79	0	200	63	90
Original interest rate	5.8	5.8	0	20.6	4.9	6.6
Original balance	190,614	151,353	1	6,450,000	98,679	238,000

Table 4: Summary statistics for some mortgage features in prime data.

Feature	Mean	Median	Min	Max	25% Quantile	75% Quantile
FICO	707	718	300	900	660	767
Original LTV	74	79	0	200	63	90
Original interest rate	6	5.95	0	30	4.9	6.9
Original balance	186,202	148,500	1	6,450,000	94,000	234,000

Table 5: Summary statistics for some mortgage features in full dataset (prime and subprime).

Feature	**Values**
Current status	Current, 30 days delinquent, 60 days delinquent, 90+ days delinquent, Foreclosed, REO, paid off
Number of days delinquent	Continuous
Current interest rate	Continuous
Current interest rate − national mortgage rate	Continuous
Time since origination	Continuous
Current balance	Continuous
Scheduled principal payment	Continuous
Scheduled principal + interest payment	Continuous
Number of months the mortgage's interest been less than the national mortgage rate and the mortgage did not prepay	Continuous
Number of occurrences of current in past 12 months	0-12
Number of occurrences of 30 days delinquent in past 12 months	0-12
Number of occurrences of 60 days delinquent in past 12 months	0-12
Number of occurrences of 90+ days delinquent in past 12 months	0-12
Number of occurrences of Foreclosed in past 12 months	0-12

Table 6: Loan-level performance features (from CoreLogic).

Feature	**Values**
Monthly zip code median house price per square feet (Zillow)	Continuous
Monthly zip code average house price (Zillow)	Continuous
Monthly three-digit zip code average house price (FHA)	Continuous
Monthly state unemployment rate (BLS)	Continuous
Yearly county unemployment rate (BLS)	Continuous
National mortgage rate (Freddie Mac)	Continuous
Median income in same zip code (Powerlytics)	Continuous
Total number of prime mortgages in same zip code (CoreLogic)	Continuous
Lagged subprime default rate in same zip code (CoreLogic)	Continuous
Lagged prime default rate in same zip code (CoreLogic)	Continuous
Lagged prime paid off rate in same zip code (CoreLogic)	Continuous
Current year	1999, ..., 2014

Table 7: Local and national economic risk factors. Data sources in parentheses. "Default rate" is taken to be the states foreclosure or REO.

Current State/Next State	Current	30 days	60 days	90+ days	Foreclosure	REO	Paid Off
Current	93	4.7	0	0	.01	.002	2
30 days	30	45	23	0	.2	.004	2
60 days	11	16	35	32	5	.01	1.5
90+ days	4	1	2	82	9	.3	2.2
Foreclosure	2	.4	.3	6.5	85	4	1.4
REO	0	0	0	0	0	100	0
Paid off	0	0	0	0	0	0	100

Table 8: Monthly transition matrix for subprime data. Transition probabilities are given in percentages.

Current State/Next State	Current	30 days	60 days	90+ days	Foreclosure	REO	Paid Off
Current	97.1	1.3	0	0	.001	.0002	1.57
30 days	34.6	44.4	19	0	.004	.003	1.82
60 days	12	16.8	34.5	34	1.6	.009	1.1
90+ days	4.1	1.4	2.6	80.2	10	.3	1.3
Foreclosure	1.9	.3	.1	6.8	87	2.5	1.3
REO	0	0	0	0	0	100	0
Paid off	0	0	0	0	0	0	100

Table 9: Monthly transition matrix for prime data. Transition probabilities are given in percentages.

Current State/Next State	Current	30 days	60 days	90+ days	Foreclosure	REO	Paid Off
Current	96.7	1.6	0	0	.002	.0004	1.61
30 days	34.2	44.5	19.3	0	.02	.003	1.84
60 days	12	16.7	34.5	33.8	1.9	.009	1.1
90+ days	4.1	1.4	2.5	80.4	9.9	.3	1.3
Foreclosure	1.9	.3	.1	6.8	86.8	2.6	1.3
REO	0	0	0	0	0	100	0
Paid off	0	0	0	0	0	0	100

Table 10: Monthly transition matrix for full dataset (both subprime and prime). Transition probabilities are given in percentages.

State/Portfolio	Logistic regression	5-Layer Neural Network
REO	0.00	0.00
Paid off	0.83	0.36
Current	98.71	99.25
30 dd	0.46	0.39
60 dd	0.00	0.00
90+ dd	0.00	0.00
Foreclosure	0.01	0.00

Table 11: Percent of portfolio which is in each state at a 1 month time horizon. Portfolios have size $N = 20,000$ and are chosen from an available pool of $100,000$ mortgages.

State/Portfolio	Logistic regression	5-Layer Neural Network
REO	0.03	0.02
Paid off	8.14	4.06
Current	89.09	93.28
30 dd	1.54	1.60
60 dd	0.36	0.36
90+ dd	0.54	0.49
Foreclosure	0.30	0.19

Table 12: Percent of portfolio which is in each state at a 1 year time horizon. Portfolios have size $N = 20,000$ and are chosen from an available pool of $100,000$ mortgages.

	5-Layer Neural Network	Logistic Regression
Avg. Actual Prepayments	1723.8	1723.8
Avg. Predicted Prepayments	1456.9	2853.8
Avg. Absolute Gap	278.5	1186.0
Avg. Standardized Gap	1.9	2.4

Table 13: Comparison of out-of-sample pool-level distribution from the 5-layer neural network and logistic regression. The table reports averages for 50 random test portfolios. "Avg. Predicted Prepayments" is the average across 50 test portfolios of the mean of the forecast distribution. The "Avg. Absolute Gap" is the absolute difference between the predicted and the actual number of prepayments. The "Avg. Standardized Gap" is the difference between the predicted and the actual number of prepayments measured in multiples of the forecast standard deviation.

Model	In-sample Neg. Log-Likelihood	Out-of-sample Neg. Log-likelihood
Logit	.1838	.1805
1 hidden layer NN	.1684	.1685
3 hidden layer NN	.1657	.1670
5 hidden layer NN	.1686	.1670
7 hidden layer NN	.1654	.1673
Ensemble	.1640	.1654

Table 14: In-sample and out-of-sample negative log-likelihood for different models. Ensemble is composed of eight 5-layer neural networks.

	Model	Current	30dd	60dd	90+ dd	F	REO	Paid off
Out-of-sample	Logit	.92719	.93206	.99069	.99670	.99781	.98980	.63521
	1 hidden layer NN	.94142	.94081	.99155	.99690	.99798	.99113	.73764
	3 hidden layer NN	.94211	.94117	.99168	.99691	.99799	.99187	.74250
	5 hidden layer NN	.94254	.94140	.99170	.99691	.99799	.99205	.74679
	7 hidden layer NN	.94052	.94109	.99169	.9969	.99798	.99187	.73336
	Ensemble	.94423	.94200	.99181	.99696	.99802	.99251	.75814
In-sample	Logit	.92453	.90937	.98956	.99713	.99735	.99287	.75218
	1 hidden layer NN	.94254	.92015	.99064	.99737	.99780	.99396	.86145
	3 hidden layer NN	.94580	.92204	.99085	.99743	.99796	.99460	.87878
	5 hidden layer NN	.94313	.92162	.99079	.99741	.99789	.99406	.86318
	7 hidden layer NN	.94616	.92221	.99082	.99742	.99789	.99425	.88075
	Ensemble	.94748	.92348	.99100	.99747	.99803	.99514	.88726

Table 15: Out-of-sample AUC for different models. "F" stands for foreclosure. The AUC for event u is the AUC for the two-way classification of whether u or another event $u' \neq u$ occurs. "dd" stands for days delinquent.

Feature	Ave. Absolute Gradient
Current outstanding balance	.1707
Original loan balance	.0731
Original interest rate	.0603
FICO score	.0589
Current interest rate - national mortgage rate	.0538
Time since origination	.0460
Lagged prime prepayment rate in same zip code	.0392
Scheduled interest and principal due	.0334
Current interest rate - original interest rate	.0320
Lagged prime default rate in same zip code	.0289
State unemployment rate	.0288
Lagged default rate for subprime mortgages in same zip code ($\leq$ 100)	.0285
Current delinquency status = current	.0264
Zillow zip code housing price change since origination	.0241
Original interest rate - national mortgage rate	.0230
Lien type = unknown	.0218
Number of times 60dd in last 12 months	.0191
Pool insurance flag = missing	.0188
Original appraised value	.0185
Current delinquency status = 90+ dd	.0171
Original term of the loan	.0169
Number of times 90+ dd in last 12 months	.0169
Total days delinquent $\geq$ 160	.0167
Scheduled principle due = missing	.0163
Number of times current in last 12 months	.0158
ARM periodic rate cap	.0155
Current delinquency status = foreclosure	.0148
Lagged default rate for subprime mortgages in same zip code ($\geq$ 1000)	.0140
LTV ratio	.0137
Zillow zip code median house price change since origination	.0135
$\vdots$	$\vdots$

Table 16: Covariate ranking by average absolute gradient for transition current $\rightarrow$ paid off. Performed using 5-layer neural network.

Feature	Test Loss
State unemployment rate	1.160
Current delinquency status = current	.353
Current outstanding balance	.303
Original interest rate	.233
FICO score	.204
Current delinquency status = foreclosure	.195
Number of times 30dd in last 12 months	.179
Current delinquency status = 30 dd	.177
Current delinquency status = 90+ dd	.176
Number of interest only months = missing	.176
Number of times current in last 12 months	.175
Original loan balance	.175
Current delinquency status = 60 dd	.173
Debt-to-income ratio = missing	.172
Total days delinquent $\geq$ 160	.171
Current year < 1999	.171
Vintage year < 1995	.171
ARM margin = missing	.171
Is the loan a prime mortgage?	.171
Average zip code income = missing	.171
Lien type = first lien	.171
Scheduled principle due = missing	.170
Original interest rate - national mortgage rate	.170
LTV ratio	.169
Burnout (number of months where it was optimal to prepay but did not)	.168
Time since origination	.168
Current interest rate - national mortgage rate	.168
Number of times 60dd in last 12 months	.168
Number of times foreclosure in last 12 months	.168
$\vdots$	$\vdots$

Table 17: Covariate ranking by leave-one-out. Test loss when no features are dropped is .167. Performed using 5-layer neural network.

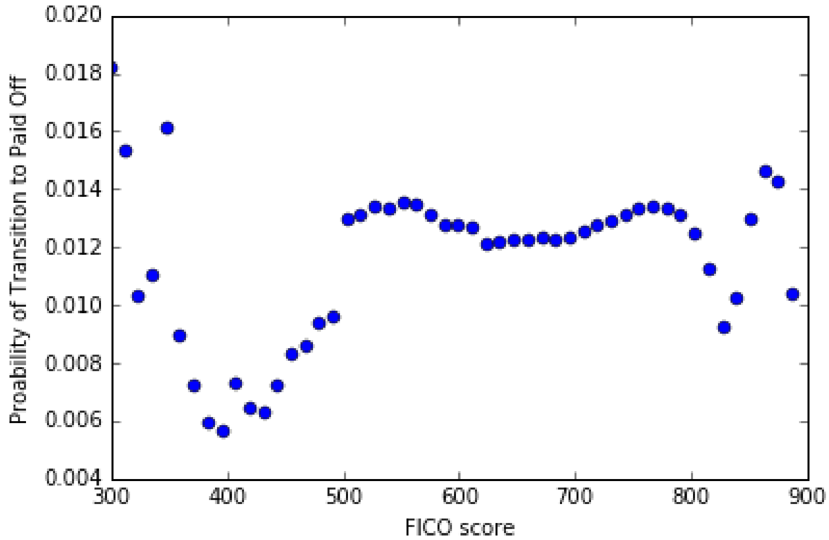

Figure 1: Prepayment rate versus FICO score. The figure shows that the prepayment rate, which is one of the model outputs, has a significant nonlinear relationship with the FICO score of the borrower, an explanatory variable input to the model. The propensity to prepay is less for borrowers with lower FICO scores but it plateaus once the score crosses a threshold of about 500 points. This reinforces the need for a model family that is capable of learning nonlinear functions of the data.

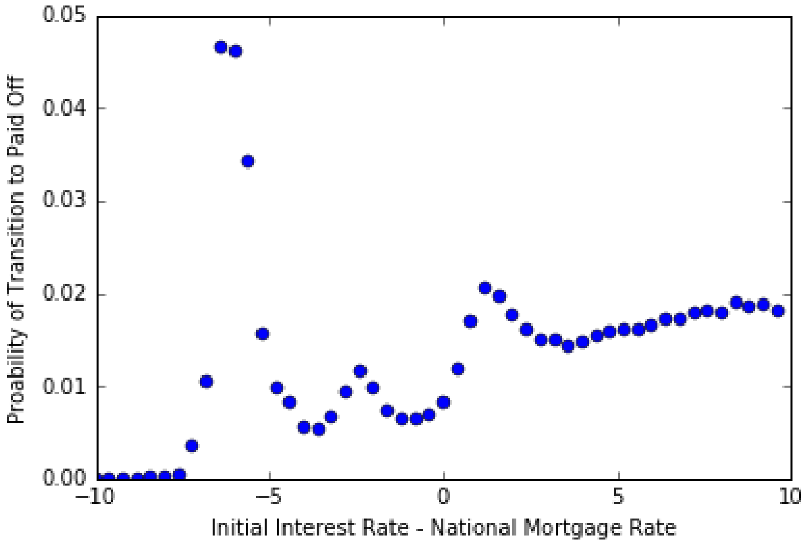

Figure 2: Prepayment rate versus incentive to prepay. The figure shows that the prepayment rate, which is one of the model outputs, has a significant nonlinear relationship with the incentive to prepay. Here, the incentive to prepay is measured as the difference between the national mortgage rate and the initial interest rate of the mortgage. A higher interest rate on the loan (as compared to the national mortgage rate) should encourage the borrower to seek better terms by refinancing the loan, implying that an upward trend should be observed in the graph. The observed data, however, point to more complicated underlying mechanisms, such as the presence of prepayment penalties or the lack of refinancing options due to other factors such as low FICO scores. This reinforces the need for a model family that is capable of learning such nonlinear interactions.

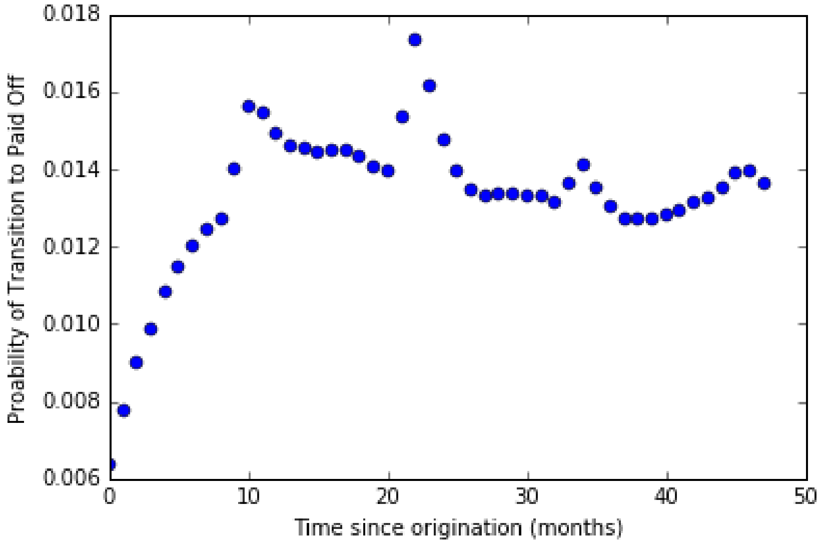

Figure 3: Prepayment rate versus time since origination. The figure shows that the prepayment rate, which is one of the model outputs, has a significant nonlinear relationship with the age of the mortgage. The age of the mortgage, measured as the time since its origination, is one of the most important factors affecting refinancing opportunities. For new mortgages, there is little refinancing, but as the borrower's credibility improves (assuming no delinquency), her incentive to refinance increases. This leads to spikes in prepayment at 12, 24 and 36 months, which is when contracts typically allow for refinancing the loan. Capturing such underlying mechanisms from the data requires a model family that is capable of learning such nonlinear interactions.

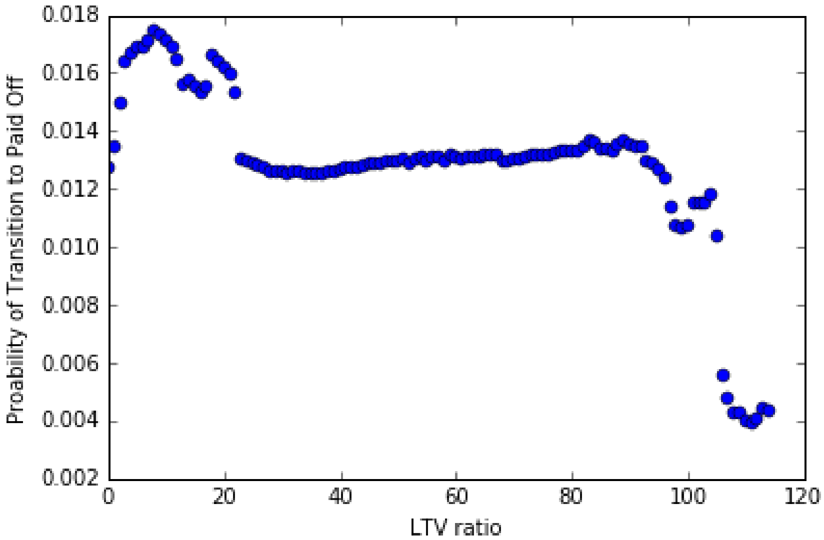

Figure 4: Prepayment rate versus LTV ratio at origination. The figure shows that the prepayment rate, which is one of the model output, has a significant nonlinear relationship with the loan-to-value (LTV) ratio. One should expect this curve to have a downward slope since a loan with high LTV will have lesser opportunities to refinance due to a large loan amount relative to the value of the asset. This trend is observed in the data, albeit with significant nonlinearity as seen in the figure. Capturing such trends requires a model family which is capable of learning nonlinear functions of the data.

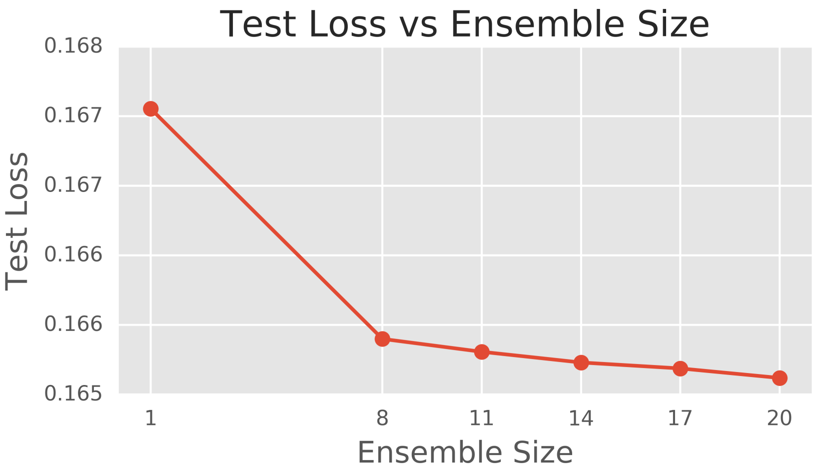

Figure 5: Out-of-sample negative log-likelihood versus number of neural networks in the ensemble. The figure shows the improvement in the out-of-sample negative log-likelihood, an indicator of the performance of the ensemble, as the number of independently trained models in the ensemble are increased. Note that each model in an ensemble is a 5-layer neural network that is trained with bootstrapped data and random initialization chosen independently of that for other models. The predictions from all models within an ensemble are averaged to produce a low-variance estimate of transition probabilities, so the computational effort increases linearly with the ensemble size. The figure shows that the gains beyond an ensemble size of 8 are marginal and may not justify using bigger ensembles due to the computational burden.

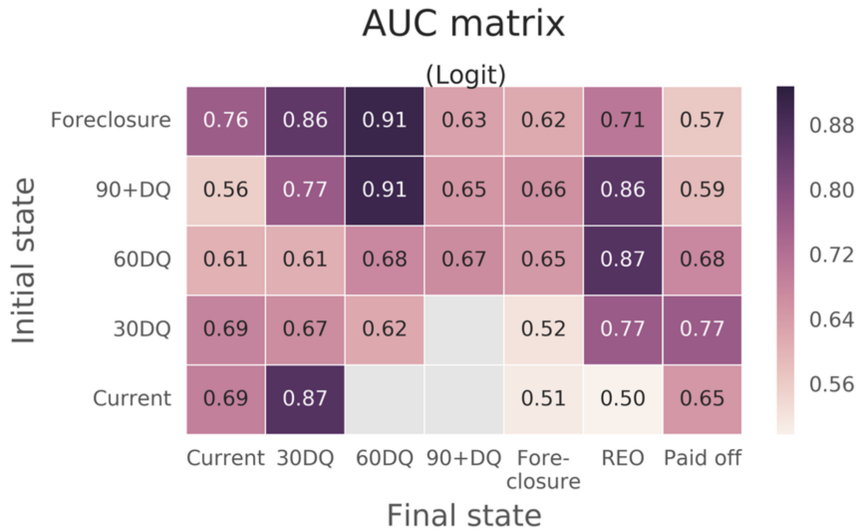

Figure 6: Out-of-sample AUCs for the logistic regression model. The AUC matrix above offers a more granular view into model performance. For mortgages in state u in the current month, the AUC for event $u \to v$ is the AUC for the two-way classification of whether the mortgage will be in state v or not next month. A higher value, depicted by a darker color, indicates better performance. We see marked improvement in the AUC values in going from the logistic regression model to the deeper networks, especially for transitions to foreclosure and paid off as well as for the transitions from the delinquent states to current.

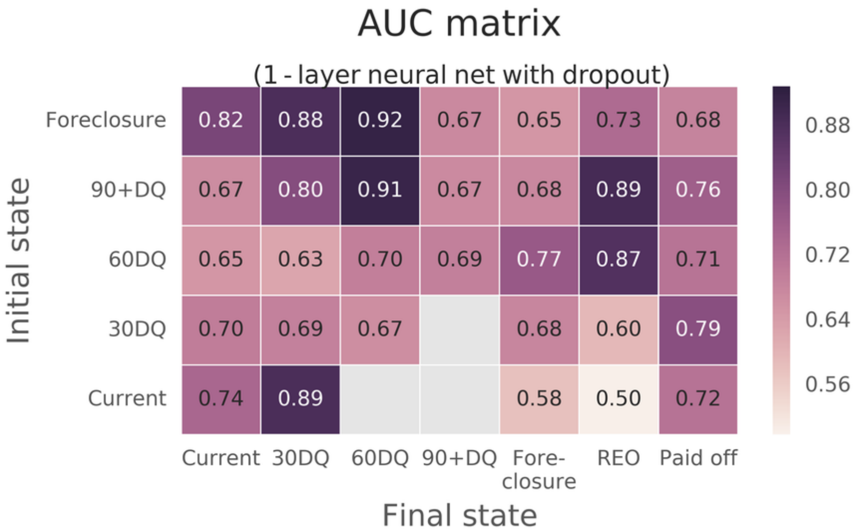

Figure 7: Out-of-sample AUCs for the 1-layer neural network. The AUC matrix above offers a more granular view into model performance. For mortgages in state u in the current month, the AUC for event $u \to v$ is the AUC for the two-way classification of whether the mortgage will be in state v or not next month. A higher value, depicted by a darker color, indicates better performance. We see marked improvement in the AUC values in going from the logistic regression model to the deeper networks, especially for transitions to foreclosure and paid off as well as for the transitions from the delinquent states to current.

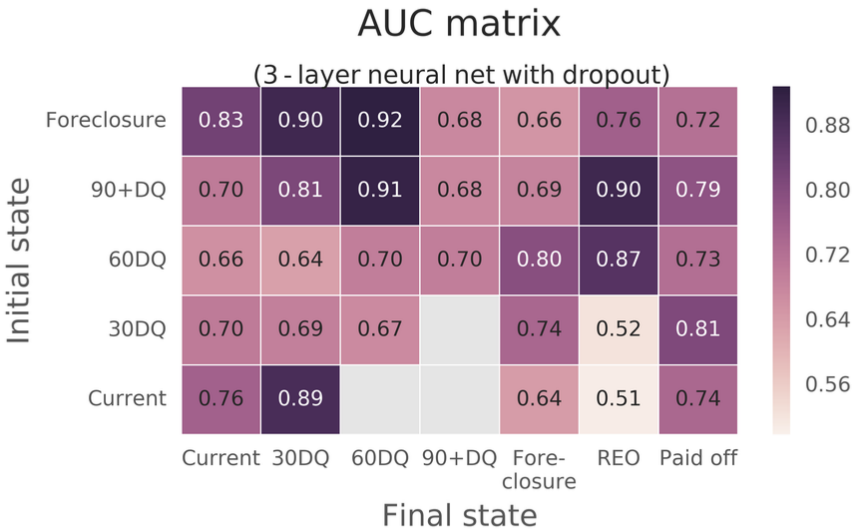

Figure 8: Out-of-sample AUCs for the 3-layer neural network. The AUC matrix above offers a more granular view into model performance. For mortgages in state u in the current month, the AUC for event $u \to v$ is the AUC for the two-way classification of whether the mortgage will be in state v or not next month. A higher value, depicted by a darker color, indicates better performance. We see marked improvement in the AUC values in going from the logistic regression model to the deeper networks, especially for transitions to foreclosure and paid off as well as for the transitions from the delinquent states to current.

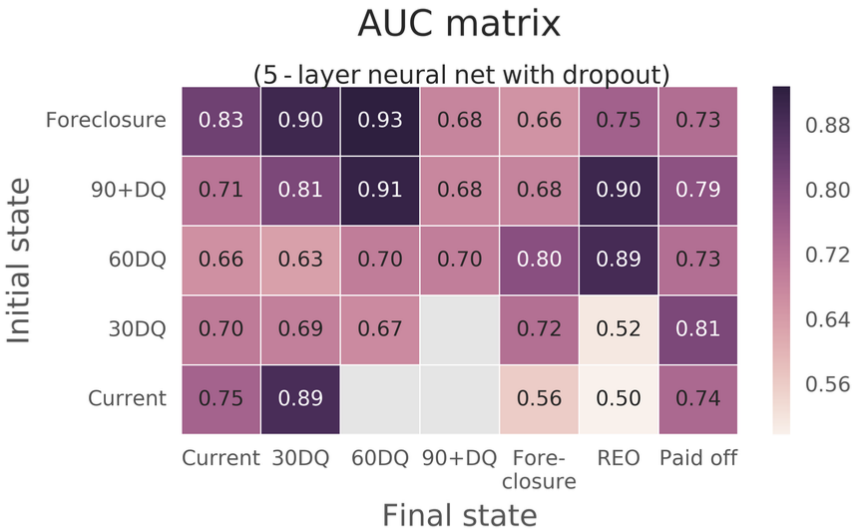

Figure 9: Out-of-sample AUCs for the 5-layer neural network. The AUC matrix above offers a more granular view into model performance. For mortgages in state u in the current month, the AUC for event $u \to v$ is the AUC for the two-way classification of whether the mortgage will be in state v or not next month. A higher value, depicted by a darker color, indicates better performance. We see marked improvement in the AUC values in going from the logistic regression model to the deeper networks, especially for transitions to foreclosure and paid off as well as for the transitions from the delinquent states to current.

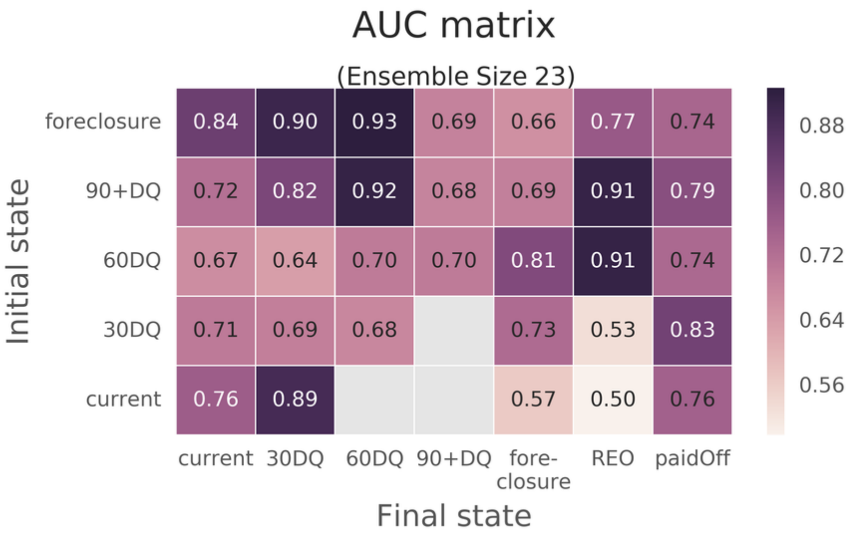

Figure 10: Out-of-sample AUCs for an ensemble of independently trained 5-layer neural networks. The AUC matrix above offers a more granular view into model performance. For mortgages in state u in the current month, the AUC for event $u \to v$ is the AUC for the two-way classification of whether the mortgage will be in state v or not next month. A higher value, depicted by a darker color, indicates better performance. In going from the 5-layer neural network to their ensemble, *every* transition in the matrix sees an improvement in prediction.

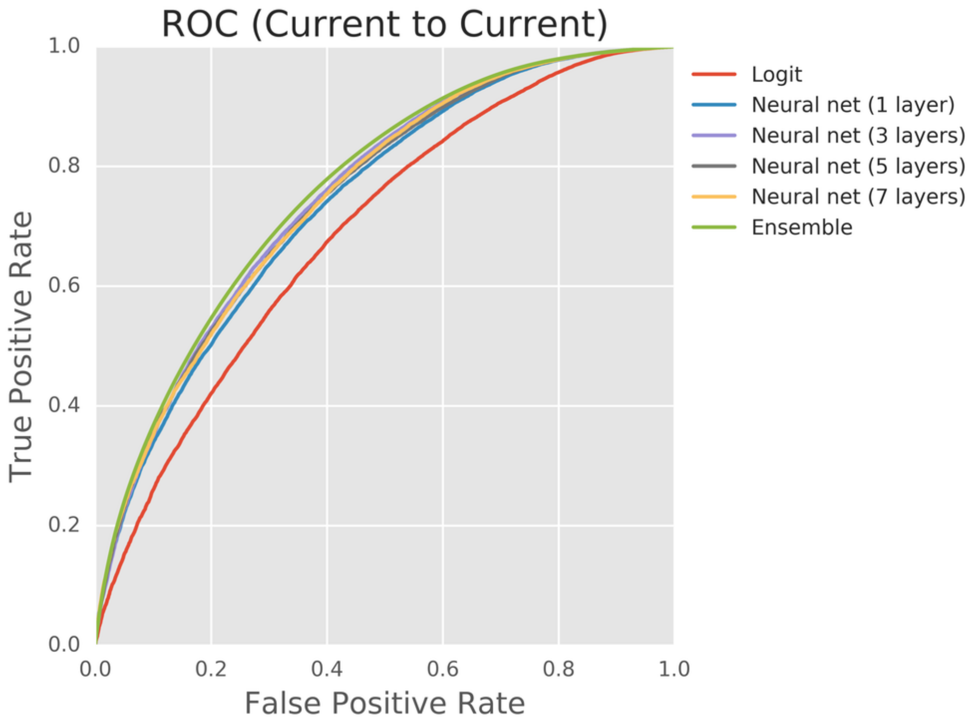

Figure 11: Out-of-sample ROC curves for various models for the transition current $\rightarrow$ current. Observe that the ROC curve corresponding to the ensemble dominates the curves for the neural networks, which in turn dominate the curve for the logistic regression model. This implies that for those mortgages that are presently in the current state, predicting whether the state next month would be current or not is best predicted by the ensemble, followed by the neural networks, and then by the logistic regression model. Further, the gap between the curves for the logistic regression model and those for the deep neural networks indicates the significant gain in predictive power due to the modeling of more complex nonlinear interactions obtained by adding multiple hidden layers to the model.

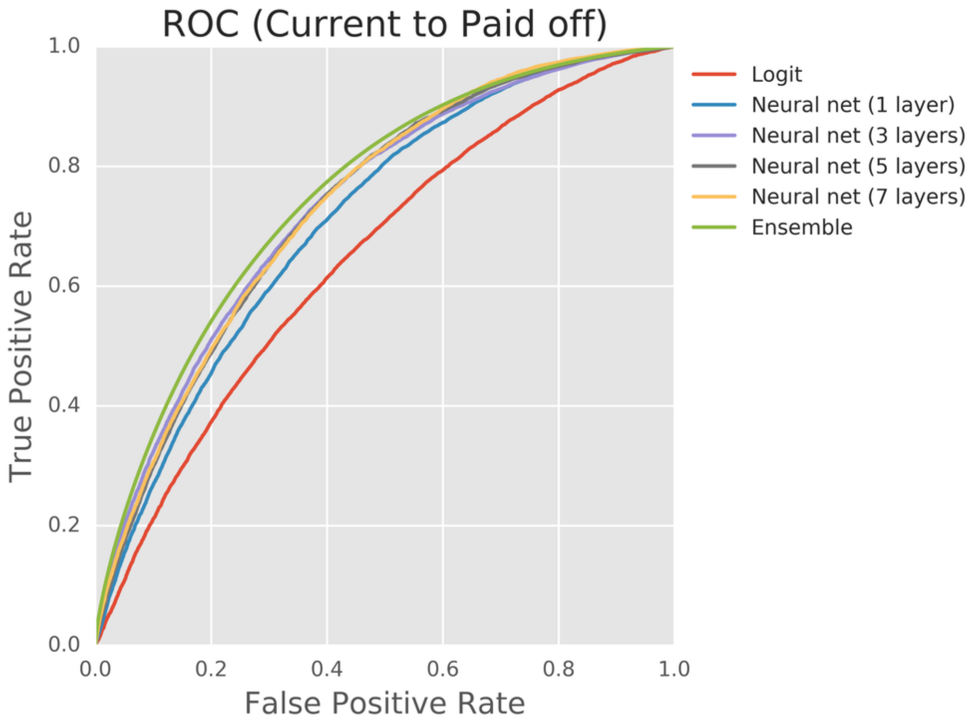

Figure 12: Out-of-sample ROC curve for the transition current $\rightarrow$ paid off. Observe that the ROC curve corresponding to the ensemble dominates the curves for the neural networks, which in turn dominate the curve for the logistic regression model. This implies that for those mortgages that are presently in the current state, predicting whether the state next month would be paid off or not is best predicted by the ensemble, followed by the neural networks, and then by the logistic regression model. Further, the gap between the curves for the logistic regression model and those for the deep neural networks indicates the significant gain in predictive power due to the modeling of more complex nonlinear interactions obtained by adding multiple hidden layers to the model.

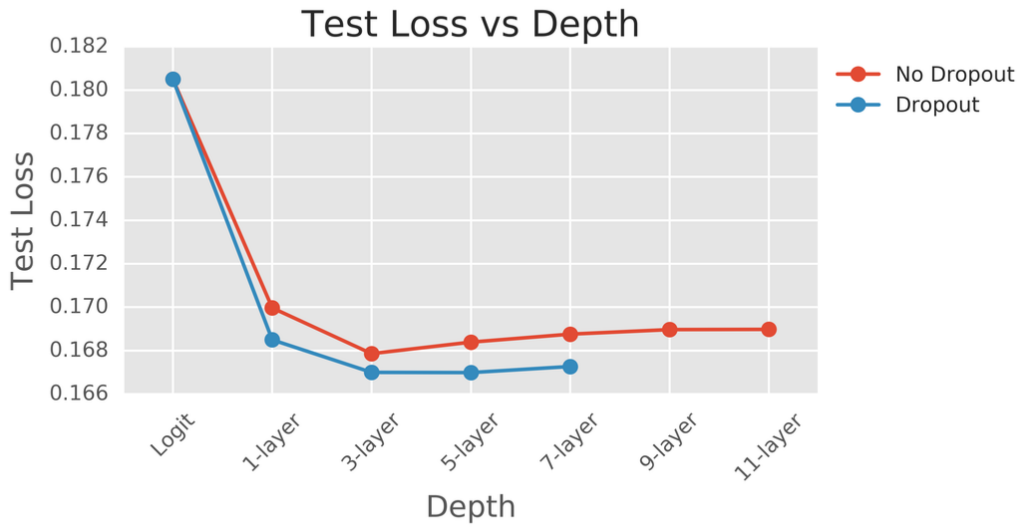

Figure 13: Out-of-sample negative log-likelihood versus number of hidden layers, for models trained with and without the use of dropout. In general, the use of dropout is important for regularization, and given a fixed architecture, the network trained with dropout shows better out-of-sample performance than the one trained without dropout. Another important observation is that it is not generally true that deeper neural networks always yield better performance due to higher model capacity; there are several other factors at play, such as over-fitting and difficulty in training the model, that affect the out-of-sample performance. We observe that the best neural networks for this problem have between 3 and 5 hidden layers.

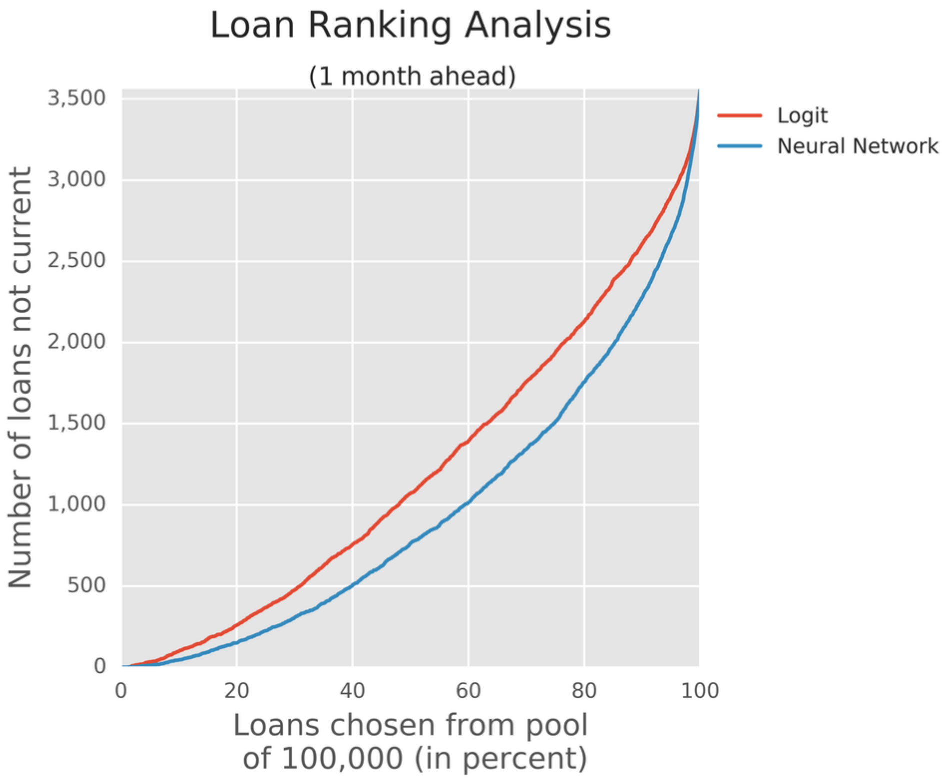

Figure 14: Comparison of out-of-sample performance for 5-layer neural network portfolio and logistic regression portfolio for 1-month ahead horizon. From a random pool of $100,000$ loans, we suppose that an investor is interested in buying a fixed number of loans N, so as to maximize the number of loans (out of these N) that remain current after one month. This requires ranking the loans on their probability of remaining current in the next month and then selecting the top N loans. This ranking is obtained for two models, 5-layer neural network and logistic regression model, and the number of loans selected, N, is varied from 0 to $100,000$. The figure shows for each portfolio size N (expressed as percent of the pool size) on the x-axis the corresponding number of loans that are not current in the subsequent month on the y-axis. The portfolio constructed using the 5-layer neural network yields superior performance for all portfolio sizes. Note that the curves intersect at the end points by design, since the portfolios selected for $K = 0$ (no loans) and $K = 100,000$ (entire pool of loans) are identical.

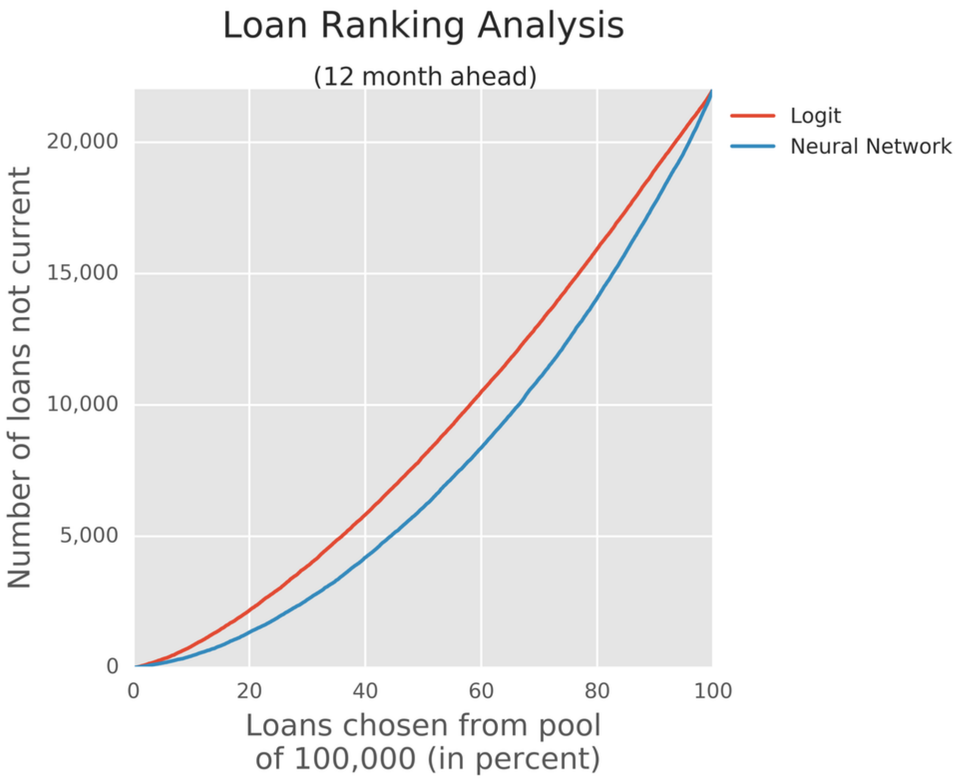

Figure 15: Comparison of out-of-sample performance for 5-layer neural network portfolio and logistic regression portfolio for 1-year ahead horizon. From a random pool of $100,000$ loans, we suppose that an investor is interested in buying a fixed number of loans N, so as to maximize the number of loans (out of these N) that remain current after 12 months. This requires ranking the loans on their probability of remaining current after 12 months and then selecting the top N loans; this probability is computed by using the method in Section 3 where the time-varying covariates are frozen. This ranking is obtained for two models, 5-layer neural network and logistic regression model, and the number of loans selected, N, is varied from 0 to $100,000$. The figure shows for each portfolio size N (expressed as percent of the pool size) on the x-axis the corresponding number of loans that are not current after 12 months on the y-axis. The portfolio constructed using the 5-layer neural network yields superior performance for all portfolio sizes. Note that the curves intersect at the end points by design, since the portfolios selected for $K = 0$ (no loans) and $K = 100,000$ (entire pool of loans) are identical.

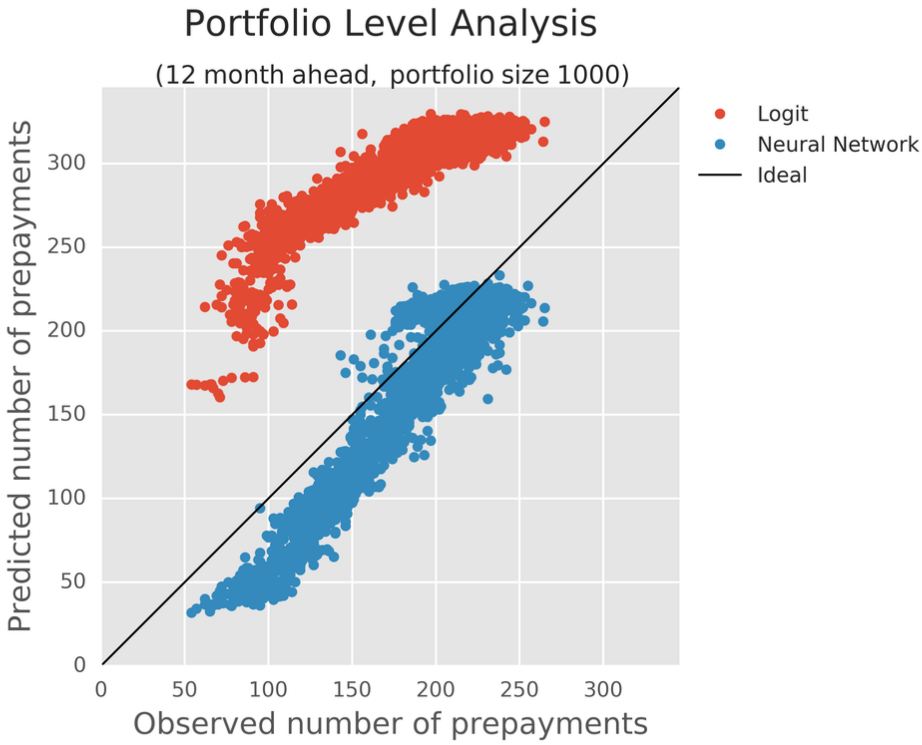

Figure 16: Comparison of out-of-sample pool-level predictions of the 5-layer neural network and the logistic regression model. A pool of 2 million mortgages is grouped into $2,000$ portfolios by ordering loans according to the borrowers' FICO score and then sequentially packaging every 1,000 loans into individual portfolios. For each such portfolio, the figure shows the observed number of prepayments in the next 12 months on the x-axis and the predicted number of prepayments in the next 12 months from the two models, the 5-layer neural network and the logistic regression model, on the y-axis. The $x = y$ line (in black) shows the ideal but hypothetical scenario under which the predicted and the observed number of prepayments coincide. It is seen that the predictions from the 5-layer neural network are much closer to this ideal line than those from the logistic regression model.

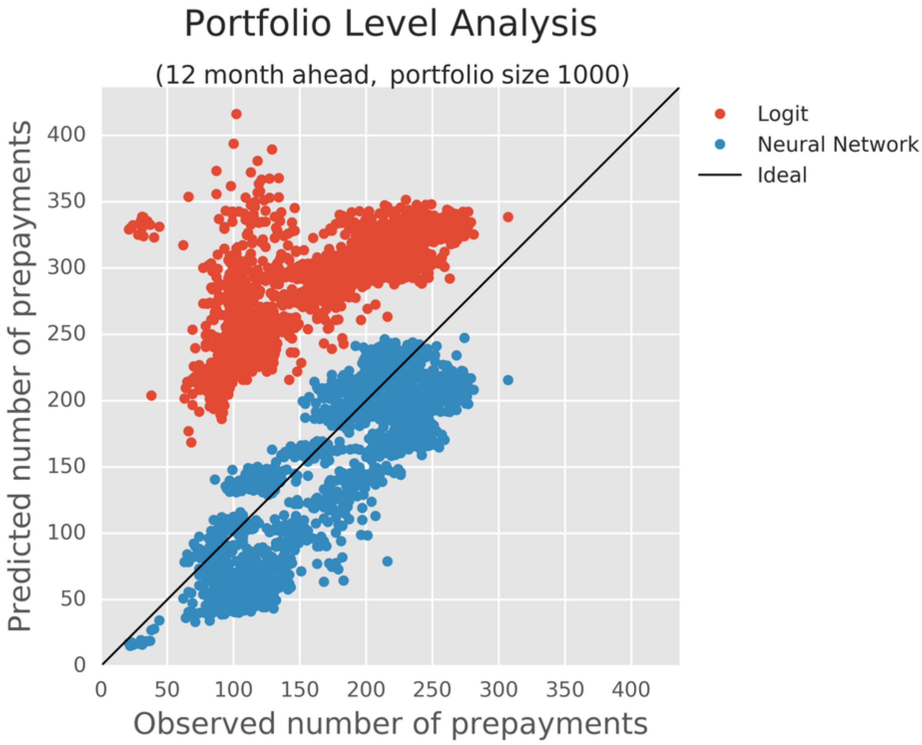

Figure 17: Comparison of out-of-sample pool-level predictions of the 5-layer neural network and the logistic regression model. A pool of 2 million mortgages is grouped into $2,000$ portfolios by ordering loans according to their initial interest rate and then sequentially packaging every 1,000 loans into individual portfolios. For each such portfolio, the figure shows the observed number of prepayments in the next 12 months on the x-axis and the predicted number of prepayments in the next 12 months from the two models, the 5-layer neural network and the logistic regression model, on the y-axis. The $x = y$ line (in black) shows the ideal but hypothetical scenario under which the predicted and the observed number of prepayments coincide. It is seen that the predictions from the 5-layer neural network are much closer to this ideal line than those from the logistic regression model.

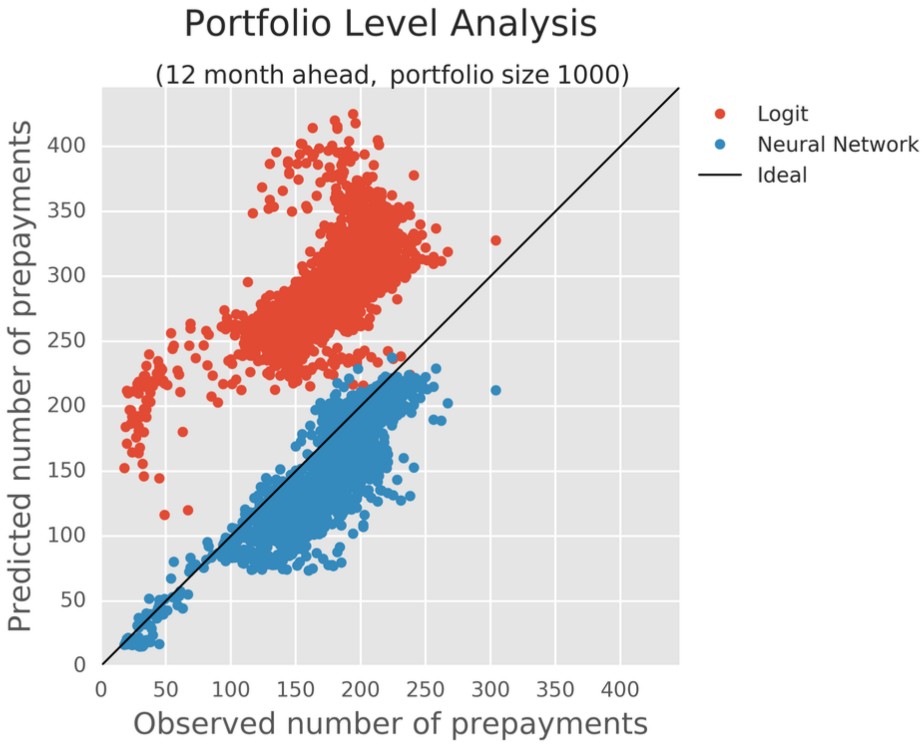

Figure 18: Comparison of out-of-sample pool-level predictions of the 5-layer neural network and the logistic regression model. A pool of 2 million mortgages is grouped into 2,000 portfolios by ordering loans according to their loan-to-value (LTV) ratio and then sequentially packaging every 1,000 loans into individual portfolios. For each such portfolio, the figure shows the observed number of prepayments in the next 12 months on the x-axis and the predicted number of prepayments in the next 12 months from the two models, the 5-layer neural network and the logistic regression model, on the y-axis. The $x = y$ line (in black) shows the ideal but hypothetical scenario under which the predicted and the observed number of prepayments coincide. It is seen that the predictions from the 5-layer neural network are much closer to this ideal line than those from the logistic regression model.

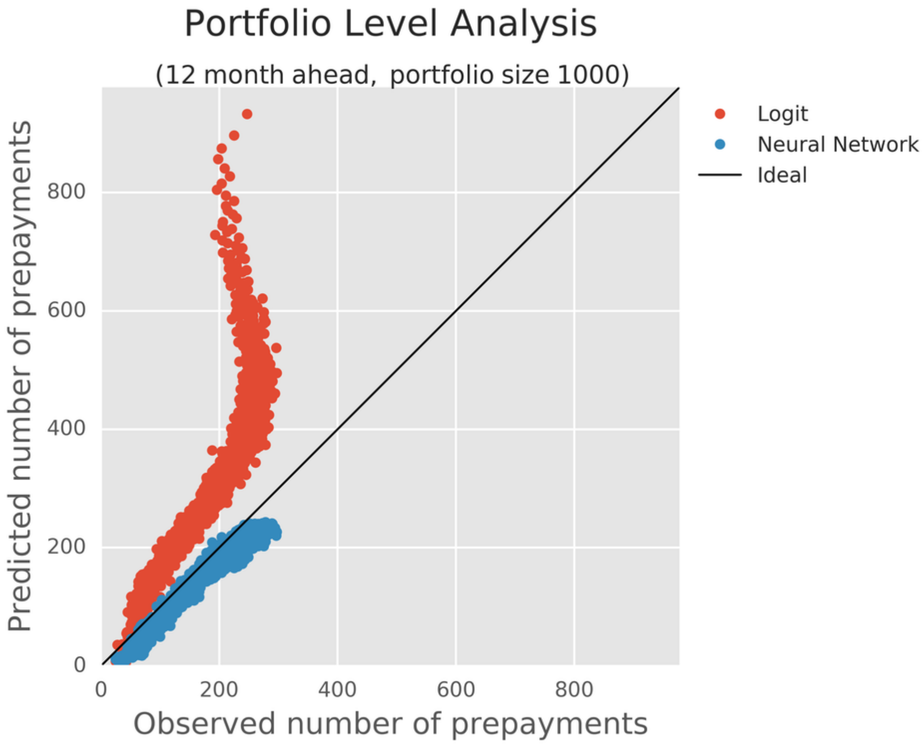

Figure 19: Comparison of out-of-sample pool-level predictions of the 5-layer neural network and the logistic regression model. A pool of 2 million mortgages is grouped into $2,000$ portfolios by ordering loans according to their probability of being current after 12 months and then sequentially packaging every 1,000 loans into individual portfolios. For each such portfolio, the figure shows the observed number of prepayments in the next 12 months on the x-axis and the predicted number of prepayments in the next 12 months from the two models, the 5-layer neural network and the logistic regression model, on the y-axis. The $x = y$ line (in black) shows the ideal but hypothetical scenario under which the predicted and the observed number of prepayments coincide. It is seen that the predictions from the 5-layer neural network are much closer to this ideal line than those from the logistic regression model. It is important to note here that the loans were ordered on their probability of being current after 12 months, where this probability is estimated using the logistic regression model. If the estimated probabilities were accurate, the portfolios so obtained would have large variations in quality with the observed number of prepayments covering the entire x-axis (as in previous plots) as well as the logistic regression model would see an increasing curve; however, neither of these trends is observed, implying that the predicted probabilities are inaccurate.

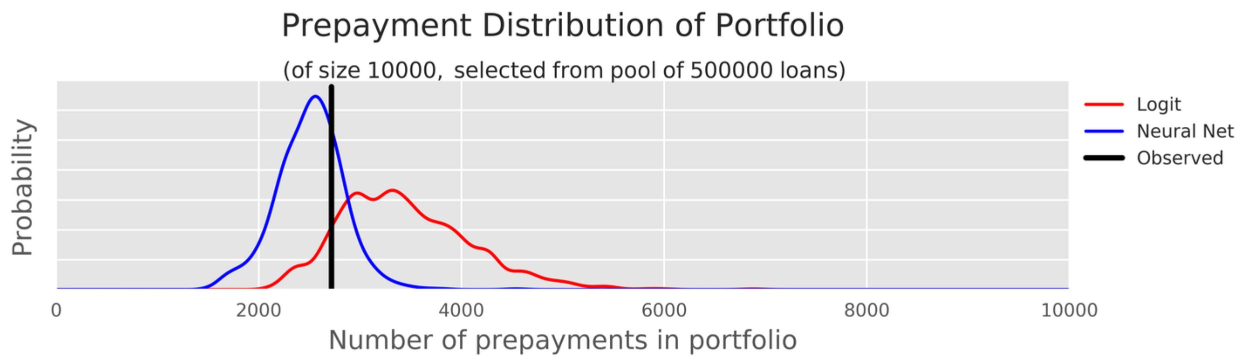

Figure 20: Comparison of out-of-sample pool-level distribution from the 5-layer neural network and the logistic regression model. The distribution of the number of prepayments at a 12-month horizon is obtained by simulating several trajectories for the time-varying covariates and then computing the transition probabilities for each loan for every trajectory; this approach is described in Section 3. For a wide range of portfolios, we observe that the gap between the mean of the distribution and the actual number of prepayments is lesser for the neural network model than for the logistic regression model.

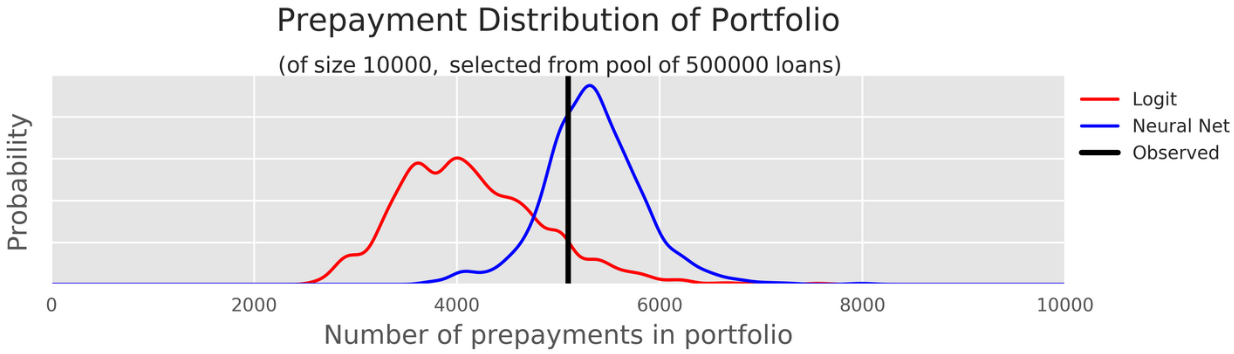

Figure 21: Comparison of out-of-sample pool-level distribution from the 5-layer neural network and the logistic regression model. The distribution of the number of prepayments at a 12-month horizon is obtained by simulating several trajectories for the time-varying covariates and then computing the transition probabilities for each loan for every trajectory; this approach is described in Section 3. For a wide range of portfolios, we observe that the gap between the mean of the distribution and the actual number of prepayments is lesser for the neural network model than for the logistic regression model.

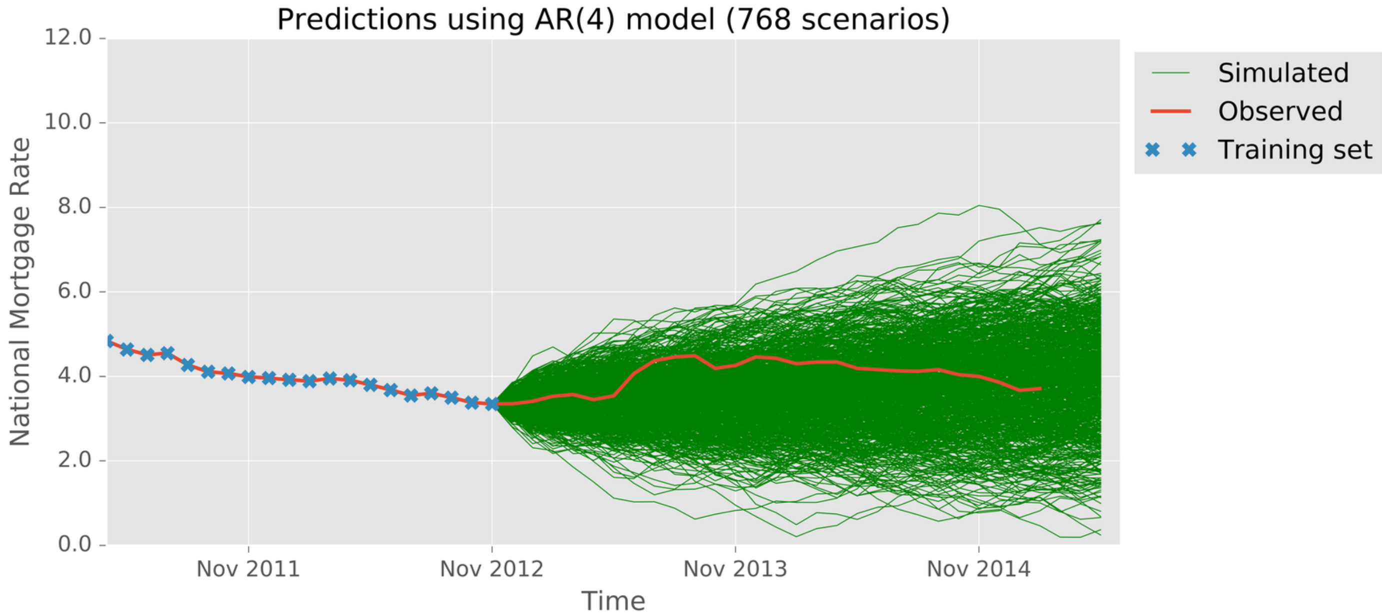

Figure 22: Simulated trajectories of the national mortgage rate. We use an AR(4) model to fit the national mortgage rate. The data used for fitting belongs to the train period, while the simulated trajectories extend over the test period. These simulated trajectories are used to obtain the distribution of the number of prepayments in a pool of loans, by harnessing the approach described in Section 3.