Deep Learning in Insurance Risk Quantification: A Comparative Study

Eugene Kwak

September 3, 2018

---

1. Definition

Project Overview

Balancing losses and premiums is the crux of managing a successful insurance business. This balancing act consists of being able to quantify and understand these financial losses (risk) that will be incurred in order to set premiums at rates that are both profitable and attractive to consumers. Here I focus on methods for quantifying risk. Insurers invest heavily on teams of analysts, domain experts and actuaries to perform this task. Countless studies are also regularly published exploring machine learning approaches to solve this problem and evaluate commercial solutions (see comparative analysis of risk models in healthcare by the Society of Actuaries from 2016)[1]. In short, risk prediction is and always has been one of the biggest challenges in insurance.

With the advent of machine learning and big data in commercial insurance, companies are eagerly seeking new ways to apply novel techniques to improve their business, but are often challenged with the complexities that come with supporting these technologies. Traditional methods used in practice include actuarial sciences and mostly linear models. More robust techniques like ensembling are recently being utilized. However, the industry is still behind in terms of state of the art methods like deep learning and the technologies to support them.

The goal of this project is to [1] determine the feasibility of deep learning in this task and [2] to do a proof of concept to make a case for insurers to invest in deep learning technologies. Allstate ran a Kaggle competition in 2016 and provided an anonymized dataset well-suited for this task[2].

Problem Statement

Can deep neural networks provide a viable solution for risk modeling in insurance claims?

Predictive models for assessing future risk are commercially available across nearly all insurance domains from automobile to healthcare. For example, Equifax offers its Insight Score[3] product while companies like Verscend sells its DxCG[4] product for healthcare insurance. In general, these solutions do not utilize deep learning for predicting future risk. The upfront investments needed simpy do not justify the marginal increase in performance deep learning often provides. Additionally, many domains have a strong preference for model interpretability, which is often why linear models are so well received.

I will attempt to make a case for deep learning through this proof of concept. Three models will be trained and evaluated.

1. Benchmark Model: A benchmark ensemble model will be trained as a proxy for a competitive market solution that a deep learning model will need to beat.

2. Deep Learning Model (CPU): A deep neural network will be trained to compare against the benchmark model. This model will also be used to benchmark the cost-benefits of technologies optimal for deep learning (GPU's).
3. Deep Learning Model (GPU): The same deep neural network architecture will be trained on a GPU cluster for comparison against the CPU trained model.

A successful deep learning model will [1] beat the benchmark model in validation or cross-validation performance, [2] be comparible to the solutions provided via Kaggle submissions for Allstate's competition, and [3] be more computationally efficient on a GPU cluster. I will prototype a fully-functional machine learning platform that can integrate with an organization's database technology stack.

Metrics

The problem statement is broken into three comparisons as outlined above. Each comparison will be based on a different set of metrics specific to the specific goals.

Benchmark Model Comparison One of the most common metrics to compare solutions and models is the humble coefficient of determination (R^2).

Table 5.2.2: R-Squared and MAE of Ensemble Models

	Concurrent Models		Prospective Models	
	R-Squared	MAE	R-Squared	MAE
Best-Fit Single Model	55.4%	57.9%	24.8%	91.8%
Mean Ensemble	55.7%	63.1%	24.8%	92.5%
Weighted Ensemble	58.2%	57.6%	26.4%	90.3%

Benchmark metric r2

Eq 1.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

GPU vs CPU comparison metric fit time mae r2

In this section, you will need to clearly define the metrics or calculations you will use to measure performance of a model or result in your project. These calculations and metrics should be justified

based on the characteristics of the problem and problem domain. Questions to ask yourself when writing this section:

Are the metrics you've chosen to measure the performance of your models clearly discussed and defined? Have you provided reasonable justification for the metrics chosen based on the problem and solution?

2. Analysis

Data Exploration

In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section:

If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader? If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed? If a dataset is not present for this problem, has discussion been made about the input space or input data for your problem? Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)

Exploratory Visualization

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

Have you visualized a relevant characteristic or feature about the dataset or input data? Is the visualization thoroughly analyzed and discussed? If a plot is provided, are the axes, title, and datum clearly defined? Algorithms and Techniques In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

Are the algorithms you will use, including any default variables/parameters in the project clearly defined? Are the techniques to be used thoroughly discussed and justified? Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen? Benchmark In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

Has some result or value been provided that acts as a benchmark for measuring performance? Is it clear how this result or value was obtained (whether by data or by hypothesis)?

3. Methodology

(approx. 3-5 pages)

Data Preprocessing

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented? Based on the Data Exploration section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected? If no preprocessing is needed, has it been made clear why? Implementation In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

Is it made clear how the algorithms and techniques were implemented with the given datasets or input data? Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution? Was there any part of the coding process (e.g., writing complicated functions) that should be documented? Refinement In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

Has an initial solution been found and clearly reported? Is the process of improvement clearly documented, such as what techniques were used? Are intermediate and final solutions clearly reported as the process is improved?

4. Results

(approx. 2-3 pages)

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate? Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data? Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results? Can results

found from the model be trusted? Justification In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

Are the final results found stronger than the benchmark result reported earlier? Have you thoroughly analyzed and discussed the final solution? Is the final solution significant enough to have solved the problem?

5. Conclusion

(approx. 1-2 pages)

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

Have you visualized a relevant or important quality about the problem, dataset, input data, or results? Is the visualization thoroughly analyzed and discussed? If a plot is provided, are the axes, title, and datum clearly defined?

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

Have you thoroughly summarized the entire process you used for this project? Were there any interesting aspects of the project? Were there any difficult aspects of the project? Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

Are there further improvements that could be made on the algorithms or techniques you used in this project? Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how? If you used your final solution as the new benchmark, do you think an even better solution exists? Before submitting, ask yourself. . .

Does the project report you've written follow a well-organized structure similar to that of the project template? Is each section (particularly Analysis and Methodology) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification? Would the intended audience of your project be able to understand your analysis, methods, and results? Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes? Are all the resources used for this project correctly cited and referenced? Is the code that implements your solution easily readable and properly commented? Does the code execute without error and produce results similar to those reported?

Sources

[1] https://www.soa.org/research-reports/2016/2016-accuracy-claims-based-risk-scoring-models/

[2] https://www.kaggle.com/c/allstate-claims-severity

[3] https://www.equifax.com/business/insight-score-insurance/

[4] https://www.verscend.com/solutions/performance-analytics/dxcg-intelligence

---

2. Problem Statement

Can deep neural networks be a viable solution for an automated risk assessment tool in insurance claims?

Risk prediction often includes cost as a dependent variable making this a regression problem. I believe that makes this investigation a great extension to the Deep Learning section of this Nanodegree as the program focused more on classification tasks.

In this project, I will explore the viability of deep neural networks in risk prediction within the domain of auto insurance. I will use the problem proposed by Allstate in a Kaggle competition[2]. There are plenty of solutions applied that have performed very well using traditional machine learning techniques. Even linear regression tends to perform quite well at this task. My examination will cover two comparisons. The first will be to compare a deep neural network against a reasonably performant and sophisticated benchmark model. The second will be to compare against the median performance attained from Kaggle submissions, which evaluated models on the mean absolute error (MAE). The median MAE is **1125.5644**.

The two performance metrics I will use are R^2 and MAE.

A proper deep learning architecture requires substantial investment in both hardware and talent. If I find reasonable performance above the above mentioned comparisons, then I believe it establishes the groundwork for building a case to support such an investment.

3. Datasets and Input

For their Kaggle competition, Allstate provided both train and test data. The model is to be trained on the train data and used to make predictions on the test data for submission to Kaggle. All data are completely anonymized by Allstate.

The train data consists of 200,867 claims and 130 predictors. The dependent variable "loss" is a measure of cost for the claim. Of the predictors, 116 are categorical and 14 are continuous. The 14 continuous variables have been scaled with a mean of 0.5 and standard deviation of 0.2. It is unclear as to whether any of the features provided were dates. Claims typically are transactional, so the lack of a clear time variable slightly limits the breadth of the analysis. However, the unit of analysis appears to be at the claim level which is often standard practice for these sorts of analyses and thus appropriate for this study.

Continuous variables are prefixed with "cont" and categorical variables are prefixed with "cat". **Figure 1** shows a correlation heatmap of the continuous variables including the target variable (loss).

Figure 1: Correlation matrix of continuous variables

4. Solution Statement

Since the problem comes in two parts, then so should the solution. Again, the goal is to create a deep learning regression model to determine whether it is a viable solution.

1. Can I beat a benchmark model such as an ensemble of regression models with deep learning? In this case, I first establish a benchmark model using traditional machine learning approaches that performs reasonably well. I then build a competing deep learning model and try to beat it. This step only uses the train data.
2. Can a quickly developed deep learning model be competitive against other solutions? Here I use the supplied test data to make predictions using the deep learning model and submit my output to Kaggle for scoring on their platform. My goal here is to determine if I can build a competitive model with limited time and resources. Competitive is defined arbitrarily as being better than the median of all results from the Kaggle competition.

5. Benchmark Model

The benchmark model will be an ensemble of various simple regression models. Although a simple linear regression could serve as a suitable benchmark for this task, I believe that a good comparison will be against a stronger candidate. Ensembling has been known to perform quite well in real world settings, including insurance. With the democratization of machine learning technologies and tools, it is getting easier for companies to apply more advanced methods like ensembling. The ensemble will be kept relatively simple and include 3 models: OLS regression, decision tree, and an elastic net. The individual model outputs will be stacked in order to create a single prediction.

The benchmark model will be self-contained within its own fitted pipeline. The pipeline will include a preliminary feature selection step using feature importance in order to reduce dimensionality. The goal is to simulate a real world data science model development and deployment process as much as possible.

6. Evaluation Metrics

I will use two metrics to evaluate my model. The first is the R^2 score to compare the deep learning model against the benchmark model. R^2 is a commonly reported metric particularly to senior management in business. The second metric I will use is the mean absolute error, which is simply the average of the absolute values of errors. Allstate used MAE for their Kaggle competition and using MAE allows for a direct comparison against the solutions offered by other data enthusiasts. I

will use MAE only once on the test dataset for the Kaggle submission once I am satisfied with the final deep learning model.

7. Project Design

A key benefit of using this Kaggle competition is that the data has been anonymized and is relatively clean so more effort can be focused on the application of techniques. This is a luxury that is often not available in real world data science projects where the majority of the time is often spent in data cleansing. Although this is a significant departure from a real world data science exercise, data cleansing is not the focus of this project. I will assume this was already done.

That said, I will follow the old adage of "trust, but verify" and perform basic validations to ensure cleanliness of the data. The following outlines the proposed workflow of the study.

1. **Exploratory Data Analysis**
   Adhering to the "trust, but verify" philosophy, this step will first examine the dataset for common issues such as missing data and row index uniqueness. Then, I will move forward taking a closer look at the feature space to understand correlations and distributions for the categorical and continuous features. I will also look at the distribution of the target variable to check for outliers.

2. **Feature Engineering**
   This step includes feature engineering such as encoding categorical variables to binaries. Given the level of anonymity of the data provided, my ability to sensibly engineer new features is rather limited. Thus, only simple transformations will be considered.

3. **Feature Selection**
   Given the relatively high dimensionality of the data provided, intuition dictates that some form of dimensionality reduction or feature selection will be needed. I will apply a few common techniques, such as removing variables with low or no variance and highly correlated variables. Once the obvious are removed, I will fit a tree-based model to allow an algorithm to rank order features by importance.

4. **Build Benchmark Model Pipeline**
   As mentioned, the benchmark model will be an ensemble of simple regression models. Here I will use cross validation and search a hyperparameter space with varying numbers of predictors to determine an optimal model. Once all the simpler models are tuned and fit, a final ensemble model will be fit to the underlying model outputs to complete the benchmark model. The ensemble's cross-validation R^2 score and MAE will be calculated.

5. **Build Deep Learning Model Pipeline**
   The proposed deep learning model will be built using Keras on a dual GPU personal computer. I will try various network architectures this way to [1] learn more about deep neural networks in regression tasks, [2] use this project as a live test of a deep learning machine I am building, and [3] attempt a comparison of a full-blown deep learning architecture against a traditional CPU-based machine learning implementation (the benchmark model). The third point is to help determine if investing in deep learning is worth it.

6. **Compare Deep Learning Model to Benchmark Model**
   The deep learning model will then be compared against the benchmark model based on the final R^2 score achieved during learning. The reason for R^2 is that it is a very intuitive measure for comparing models and is often reported in business settings. The MAE will also be observed for both models as supplemental information.

7. **Predict with Test Data**
   Up to this point, the test data has not been touched in any capacity beyond downloading it from Kaggle. A suitable candidate for a deep learning model will be used to make predictions on this test set.

8. **Submit Predictions to Kaggle**
   The predictions will be formatted for submission to the Kaggle competition site. Kaggle has the actual true values of the test data labels, so this will allow me to get a measure of final generalizability of my model. More importantly, I can directly compare how my model does against the competition. My goal is to build a competitive model, here defined as surpassing the median ranking MAE of 1125.5644 achieved during the competition. This will provide what I need to answer the ultimate question of this investigation.
   Can deep neural networks be a viable solution for an automated risk assessment tool in insurance claims?

Sources

[1] https://www.soa.org/research-reports/2016/2016-accuracy-claims-based-risk-scoring-models/

[2] https://www.kaggle.com/c/allstate-claims-severity