Predicting Auction Price of Vehicle License Plate with Deep Recurrent Neural Network

Vinci Chow

The Chinese University of Hong Kong

October 28, 2017

Abstract

In Chinese societies, superstition is of paramount importance, and vehicle license plates with desirable numbers can fetch very high prices in auctions. Unlike other valuable items, license plates are not allocated an estimated price before auction. I propose that the task of predicting plate prices can be viewed as a natural language processing (NLP) task, as the value depends on the meaning of each individual character on the plate and its semantics. I construct a deep recurrent neural network (RNN) to predict the prices of vehicle license plates in Hong Kong, based on the characters on a plate. I demonstrate the importance of having a deep network and of retraining. Evaluated on 13 years of historical auction prices, the deep RNN's predictions can explain over 80 percent of price variations, outperforming previous models by a significant margin.

Keywords: price predictions; expert system; recurrent neural networks; deep learning; natural language processing

1 Introduction

Chinese societies place great importance on numerological superstition. Numbers such as 8 (representing prosperity) and 9 (longevity) are often used solely because of the desirable qualities they represent. For example, the Beijing Olympic opening ceremony occurred on 2008/8/8 at 8 p.m., the Bank of China (Hong Kong) opened on 1988/8/8, and the Hong Kong dollar is linked to the U.S. dollar at a rate of around 7.8.[1]

License plates represent a very public display of numbers that people can own, and can therefore unsurprisingly fetch an enormous amount of money. Governments have not overlooked this, and plates of value are often auctioned off to generate public revenue. Unlike the auctioning of other valuable items, however, license plates generally do not come with a price estimate, which has been shown to be a significant factor affecting the sale price [1, 2]. The large number of character combinations and of plates per auction makes it difficult to provide reasonable estimates.

This study proposes that the task of predicting a license plate's price based on its characters can be viewed as a natural language processing (NLP) task. Whereas in the West numbers can be

[1] In one of his newspaper column, Steven N.S. Cheung, a leading economist in Hong Kong, recalls that in a exchange of letters regarding the establishment of the linked exchange rate, Sir John Henry Bremridge, who played a key role in the establishment of the link in his role as the financial secretary of Hong Kong from 1981 to 1986, told him that "I knew people in Hong Kong like the character '8', don't you think I was a genius?"

desirable (such as 7) or undesirable (such as 13) in their own right for various reasons, in Chinese societies numbers derive their superstitious value from the characters they rhyme with. As the Chinese language is logosyllabic and analytic, combinations of numbers can stand for sound-alike phrases. Combinations of numbers that rhyme with phrases that have positive connotations are thus desirable. For example, "168," which rhythms with "all the way to prosperity" in Chinese, is the URL of a major Chinese business portal (http://www.168.com). Looking at the historical data analyzed in this study, license plates with the number 168 fetched an average price of US$10,094 and as much as $113,462 in one instance. Combinations of numbers that rhyme with phrases possessing negative connotations are equally undesirable. Plates with the number 888 are generally highly sought after, selling for an average of $4,105 in the data, but adding a 5 (rhymes with "no") in front drastically lowers the average to $342.

As these examples demonstrate, the value of a certain combination of characters depends on both the meaning of each individual character and the broader semantics. The task at hand is thus closely related to sentiment analysis and machine translation, both of which have advanced significantly in recent years.

Using a deep recurrent neural network (RNN), I demonstrate that a good estimate of a license plate's price can be obtained. The predictions from this study's deep RNN were significantly more accurate than previous attempts to model license plate prices, and are able to explain over 80 percent of price variations. There are two immediate applications of the findings in this paper: first, an accurate prediction model facilitates arbitrage, allowing one to detect underpriced plates that can potentially fetch for a higher price in the active second-hand market. Second, the feature vectors extracted from the last recurrent layer of the model can be used to construct a search engine for historical plate prices. among other uses, the search engine can provide highly-informative justification for the predicted price of any given plate.

In a more general sense, this study demonstrates the value of deep networks and NLP in making accurate price predictions, which is of practical importance in many industries and has led to a huge volume of research. As detailed in the following review, studies to date have relied on small, shallow networks. The use of text data is also rare, despite the large amount of business text data available. By demonstrating how a deep network can be trained to predict prices from sequential data, this study provides an approach that may improve prediction accuracy in many industrial applications.

The paper is organized as follows: Section 2 describes Hong Kong license plate auctions, followed by a review of related studies in Section 3. Section 4 details the model, which is tested in Section 5. Section 7 explores enhancements to the model, including the effect of retraining over time. Section 8 concludes the paper, and includes a discussion of further developments that have potential practical uses.

2 License Plate Auctions in Hong Kong

License plates have been sold through government auctions in Hong Kong since 1973, and restrictions are placed on the reselling of plates. Between 1997 and 2009, 3,812 plates were auctioned per year, on average.

Traditional plates, which were the only type available before September 2006, consist of either a two-letter prefix or no prefix, followed by up to four digits (e.g., AB 1, LZ 3360, or 168). Traditional plates can be divided into the mutually exclusive categories of special plates and ordinary plates. Special plates are defined by a set of legal rules and include the most desirable plates.[2] Ordinary plates are issued by the government when a new vehicle is registered. If the

[2] A detailed description of the rules is available on the government's official auction web-

vehicle owner does not want the assigned plate, she can return the plate and bid for another in an auction. The owner can also reserve any unassigned plate for auction. Only ordinary plates can be resold.

In addition to traditional plates, personalized plates allow vehicle owners to propose the string of characters used. These plates must then be purchased from auctions. The data used in this study do not include this type of plate.

Auctions are open to the public and held on weekends twice a month by the Transport Department. The number of plates to be auctioned ranged from 90 per day in the early years to 280 per day in later years, and the list of plates available is announced to the public well in advance. The English oral ascending auction format is used, with payment settled on the spot, either by debit card or check.

3 Related Studies

Most relevant to the current study is the limited literature on the modeling price of license plates, which uses hedonic regressions with a larger number of handcrafted features [3, 4, 5]. These highly ad-hoc models rely on handcrafted features, so they adapt poorly to new data, particularly if they include combinations of characters not previously seen. In contrast, the deep RNN considered in this study learns the value of each combination of characters from its auction price, without the involvement of any handcrafted features.

The literature on using neural networks to make price predictions is very extensive and covers areas such as stock prices [6, 7, 8, 9], commodity prices [10, 11, 12], real estate prices [13, 14, 15], electricity prices [16, 17], movie revenues [18, 19, 20, 21], automobile prices [22] and food prices [23]. Most studies focus on numeric data and use small, shallow networks, typically using a single hidden layer of fewer than 20 neurons. The focus of this study is very different: predicting prices from combinations of alphanumeric characters. Due to the complexity of this task, the networks used are much larger (up to 1,024 hidden units per layer) and deeper (up to 9 layers).

The approach is closely related to sentiment analysis. A particularly relevant line of research is the use of Twitter feeds to predict stock price movements [24, 25, 26], although the current study has significant differences. A single model is used in this study to generate predictions from character combinations, rather than treating sentiment analysis and price prediction as two distinct tasks, and the actual price level is predicted rather than just the direction of price movement. This end-to-end approach is feasible because the causal relationship between sentiment and price is much stronger for license plates than for stocks.

Deep RNNs have been shown to perform very well in tasks that involve sequential data, such as machine translation [27, 28, 29, 30] and classification based on text description [31], and are therefore used in this study. Predicting the price of a license plate is relatively simple: the model only needs to predict a single value based on a string of up to six characters. This simplicity makes training feasible on the relatively small volume of license plate auction data used in the study, compared with datasets more commonly used in training deep RNN.

site: `http://www.td.gov.hk/en/public_services/auction_of_vehicle_registration_marks/how_to_obtain_your_favourite_vehicle_registration/schedule/index.html`.

4 Modeling License Plate Price with a Deep Recurrent Neural Network

The input from each sample is an array of characters (e.g., [“X,” “Y,” “1,” “2,” “8”]), padded to the same length with a special character. Each character s_t is converted by a lookup table to a vector representation h_0^t, known as *character embedding*. The dimension of the character embedding is a hyperparameter while the values are learned through training. The embedding is fed into the neural network sequentially, denoted by the time step t.

The neural network consists of multiple bidirectional recurrent layers, followed by one or more fully connected layers [32]. The bidirectionality allows the network to access hidden states from both the previous and next time steps, improving its ability to understand each character in context. The network also uses batch normalization, which has been shown to speed up convergence [33].

Each recurrent layer is implemented as follows:

$$h_l^t = \left[h_{l^-}^t : h_{l^+}^t\right] \tag{1}$$

$$h_{l^-}^t = f(B_l(W_{l^-} h_{l-1}^t + U_{l^-} h_{l^-}^{t-1})) \tag{2}$$

$$h_{l^+}^t = f(B_l(W_{l^+} h_{l-1}^t + U_{l^+} h_{l^+}^{t+1})) \tag{3}$$

$$B_l(x) = \gamma_l \hat{x} + \beta_l \tag{4}$$

where f is the rectified-linear unit, h_{l-1}^t is the vector of activations from the previous layer at the same time step t, h_l^{t-1} represents the activations from the current layer at the previous time step $t-1$, and h_l^{t+1} represents the activations from the current layer at the next time step $t+1$. B is the BatchNorm transformation, and $\hat{x}$ is the within-mini-batch-standardized version of x.[3] W, U, γ and β are weights learnt by the network through training.

The fully connected layers are implemented as

$$h_l = f(b_l + W_l h_{l-1}) \tag{5}$$

except for the last layer, which is implemented as

$$y = b_l + W_l h_{l-1} \tag{6}$$

b_l is a bias vector learnt from training. The outputs from all time steps in the final recurrent layer are added together before being fed into the first fully connected layer.

To prevent overfitting, dropout is applied after every layer except the last [34].

The model's hyperparameters include the dimension of character embeddings, number of recurrent layers, number of fully connected layers, number of hidden units in each layer, and dropout rate. These parameters must be selected ahead of training.

5 Experiment

5.1 Data

The data used are the Hong Kong license plate auction results from January 1997 to July 2010, obtained from the HKSAR government. The data contain 52,926 auction entries, each consisting

[3] Specifically, $\hat{x}_i = \frac{x_i - \bar{x}_i}{\sqrt{\sigma_{x_i}^2 + \epsilon}}$, where $\bar{x}_i$ and $\sigma_{x_i}^2$ are the mean and variance of x within each mini-batch. ϵ is a small positive constant that is added to improve numerical stability, set to 0.0001 for all layers.

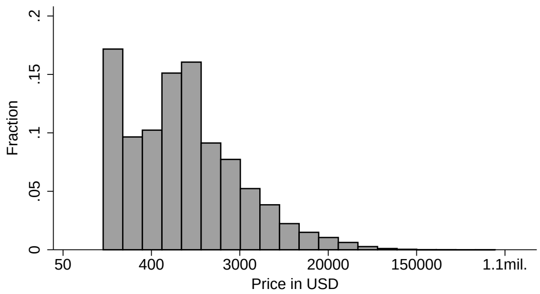

Figure 1: Distribution of Plate Prices

of i. the characters on the plate, ii. the sale price (or a specific symbol if the plate was unsold), and iii. the auction date.

Figure 1 plots the distribution of prices within the data. The figure shows that the prices are highly skewed: while the median sale price is $641, the mean sale price is $2,073. The most expensive plate in the data is "12," which was sold for $910,256 in February 2005. To compensate for this skewness, log prices were used in training and inference.

Ordinary plates start at a reserve price of HK$1,000 ($128.2), with $5,000 ($644.4) for special plates. The reserve prices mean that not every plate is sold, and 5.1 percent of the plates in the data were unsold. As these plates did not possess a price, we followed previous studies in dropping them from the dataset, leaving 50,698 entries available for the experiment.

The finalized data were randomly divided into three: training was conducted with 64 percent of the data, validation was conducted with 16 percent, and the remaining 20 percent served as the test set.

5.2 Training

I conducted a grid search to investigate the properties of different combinations of hyperparameters, varying the dimension of character embeddings $(12, 24, 48, 96, 128, 256)$, the number of recurrent layers $(1, 3, 5, 7, 9)$, the number of fully connected layers $(1, 3)$, the number of hidden units in each layer $(64, 128, 256, 512, 1024, 2048)$ and the dropout rate $(0, .05, .1)$. A total of 1080 sets of hyperparameters were investigated.

Training was repeated under each set of hyperparameters 30 times with different initializations. During each training session, a network was trained for 40 epochs under mean-squared error. An Adagrad optimizer with a learning rate of 0.001 and a gradient clip of 15 was used throughout [35]. After training was completed, the best state based on the validation error was reloaded for inference.

Training was conducted with a pair of NVIDIA GTX 1080s. To fully use the GPUs, a large mini-batch size of 2,048 was used.[4] The median training time on a single GPU ranged from 8 seconds for a 2-layer, 64-hidden-unit network with an embedding dimension of 12, to 1 minute 57 seconds for an 8-layer, 1,024-hidden-unit network with an embedding dimension of 24, and to 7 minutes 50 seconds for a 12-layer 2,048-hidden-unit network with an embedding dimension of 256.

Finally, I also trained recreations of models from previous studies as well as a series of fully-connected networks and character n-gram models for comparison. Given that the maximum

[4] I also experimented with smaller batch sizes of 64 and 512. By keeping the training time constant, the smaller batch size resulted in worse performance, due to the reduction in epochs.

Table 1: Model Performance

Configuration	Train RMSE	Valid RMSE	Test RMSE	Train R^2	Valid R^2	Test R^2
RNN						
1024-24-7-1-.05	.5217	.5712	.5714	.8385	.8063	.8052
1024-48-7-1-.05	.5176	.5737	.5701	.8409	.8046	.8061
Woo et al. (2008)	.7127	.7109	.7110	.6984	.7000	.6983
Ng et al. (2010)	.7284	.7294	.7277	.6850	.6842	.6840
MLP 1024-24-8-.05	.7222	.7725	.7643	.6885	.6502	.6545
MLP 2048-24-8-.05	.5884	.7254	.7204	.7932	.6915	.6930
unigram kNN-10	.8945	1.004	.9997	.5221	.4086	.4088
(1-4)-gram kNN-10	.9034	1.012	1.013	.5125	.3996	.3931
Combined	.5054	.5551	.5527	.8484	.8171	.8177
Combined + Extra	.4874	.5296	.5298	.8590	.8335	.8325

Configuration of RNN is reported in the format of [Hidden Units]-[Embed. Dimension]-[Recurrent Layers]-[Fully Connected Layers]-[Dropout Rate]. Configuration of MLP is reported in the same format except there is no recurrent layer. Numbers for RNN, MLP and Ensemble models are the medians from 30 runs.

length of a plate is six characters, for the n-gram models I focused on $n \leq 4$, and in each case calculated a predicted price based on the median and mean of k closest neighbors from the training data, where $k = 1, 3, 5, 10, 20$.

5.3 Model Performance

Table 1 reports the summary statistics for the best two sets of parameters out of the 1080 sets specified in section 5.2, based on the median validation RMSE. The performance levels of these models were quite close, with both able to explain more than 80 percent of the variation in prices. As a comparison, *Woo et al. (2008)* and *Ng et al. (2010)*, which represent recreations of the regression models in [4] and [5], respectively, were capable of explaining only 70 percent of the variation at most.[5]

The importance of having recurrent layers can be seen from the inferior performance of the fully-connected network (MLP) with the same embedded dimension, number of layers and neurons as the best RNN model. This model was only capable of explaining less than 66 percent of the variation in prices. Even with double the neurons a MLP was only capable of explaining 69 percent of the variation.

In the interest of space, I include only two best-performing n-gram models based on median prices of neighbors. Both models were significantly inferior to RNN and hedonic regressions, being able to explain only 40 percent of the variation in prices. For unigram, the best validation performance was achieved when $k = 10$. For $n > 2$, models with unlimited features have very poor performance, as they generate a large number of features that rarely appear in the data. Restricting the number of features based on occurances and allowing a range of n within a single model improve performance, but never surpassing the performance of the simple unigram. The performance of using median price and using mean price are very close, with a difference smaller than 0.05 in all cases.

[5] To make the comparison meaningful, the recreations contained only features based on the characters on a plate. Extra features such as date and price level are examined in Part 7.1.

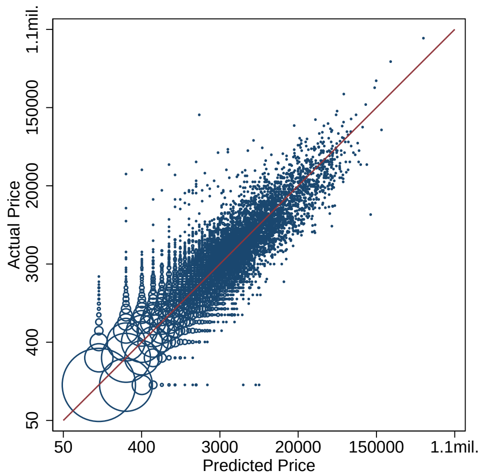

Figure 2: Actual vs Predicted Price. Plates are grouped by their predicted price and actual price, in bins of HK$1,000 ($128.2). The size of the circle represent the number of plates in a given bin.

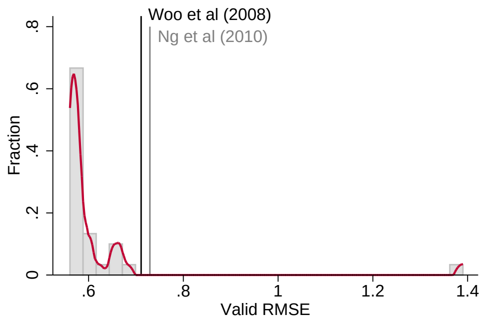

Figure 3: Performance Fluctuations. The histogram represents the best model's validation RMSE distribution. The red line is the kernel density estimate of the distribution. The two vertical lines indicate the validation RMSE of the comparison models.

Figure 2 plots the relationship between predicted price and actual price from a representative run of the best model, grouped in bins of HK$1,000 ($128.2). The model performed well for a wide range of prices, with bins tightly clustered along the 45-degree line. It consistently underestimated the price of the most expensive plates, however, suggesting that the buyers of these plates had placed on them exceptional value that the model could not capture.

Figure 4 plots the variation in performance as the hyperparameters deviate from the best-performing model. The effectiveness of deep networks has been previously noted and is also demonstrated here. Performance improved significantly with a hidden unit count and a recurrent layer count, leveling off around 512 hidden units and 7 recurrent layers. Under the hyperparameters of the best model, the median RMSE of a 1-layer model was 35 percent higher than that of a 7-layer model, while that of a 64-units-per-layer model was 11 times that of a 1,024-units-per-layer model. However, there appears to be no benefit in stacking fully connected layers: the model with three fully connected layers had a median RSME 24 percent higher than that of the one-layer version.

Performance peaked out relatively early with the dimensionality of character embedding, which is not surprising given there were only 33 possible characters.[6] Unlike the hidden unit count or the recurrent layer count, there is a clear sweet spot for the dimensionality of embedding, and performance worsens rapidly as the dimensionality increases beyond that point.

A small amount of dropout was necessary to achieve good performance. Without dropout, the model was much more likely to converge to local maxima, resulting in poor fit in many cases.

[6] The alphabets "I," "O" and "Q" are not used to avoid confusion with "1" and "0".

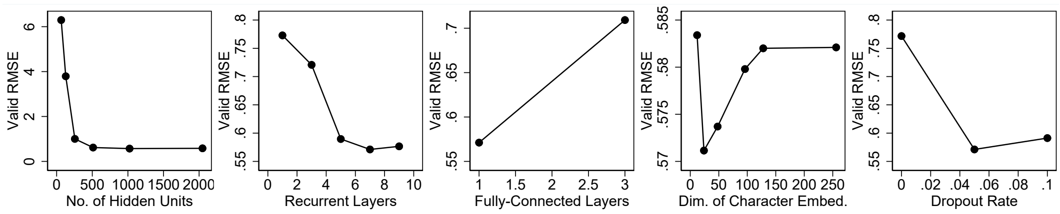

Figure 4: Hyperparameters' Effect on Model Performance. Each point represents the median from 30 runs.

5.4 Model Stability

Unlike hedonic regressions, which give the same predictions and achieve the same performance in every run, a neural network is susceptible to fluctuations due to convergence to local maxima. These fluctuations can be smoothed out by combining the predictions of multiple runs of the same model, although the number of runs necessary to achieve good results is then a practical concern.

Figure 3 plots the kernel density estimate of validation RMSEs for the best model's 30 training runs. The errors are tightly clustered, with only one run producing particularly inaccurate predictions. Excluding that run, the standard deviation of validation RMSE was only 0.034, which suggests that in practice several runs should suffice.

6 Explaining the Predictions

Compared to models such as regression and n-gram it is relatively hard to understand the rationale behind a RNN model's prediction, given the large number of parameters involved and the complexity of the their interaction. If the RNN model is to be deployed in the field, it would need to be able to explain its prediction in order to convince human users to adopt it in practice. One way to do so is to extract a feature vector for each plate by summing up the output of the last recurrent layer over time. This feature vector is of the same size as the number of neurons in the last layer, which can be fed into a standard k-nearest-neighbor model to provide a "rationale" for the model's prediction.

To demonstrate this procedure, I use the best RNN model in Table 1 to generate feature vectors for all training samples. These samples are used to setup a k-NN model. When the user submit a query, a price prediction is made with the RNN model, while a number of examples are provided by the k-NN model as rationale.

Table 2 illustrate the outcome of this procedure with three examples. The model was asked to predict the price of three plates, ranging from low to high value. The predicted prices are listed in the *Prediction* section, while the *Historical Examples* section lists for each query the top four entries returned by the k-NN model. Notice how the procedure focused on the numeric part for the low-value plate and the alphabetical part for the middle-value plate, reflecting the value of having identical digits and identical alphabets respectively. The procedure was also able to inform the user that a plate has been sold before. Finally, the examples provided for the high-value plate show why it is hard to obtain an accurate prediction for such plates, as the historical prices for similar plates are also highly variable.

Table 2: Explaining Predictions with Automated Selection of Historical Examples

	Plate	Price	Plate	Price	Plate	Price
Prediction	LZ3360	1000	MM293	5000	13	2182000
Historical Examples	HC3360	1000	MM293	5000	178	195000
	BG3360	3000	MM203	5000	138	1100000
	HV3360	3000	MM923	9000	12	7100000
	EC4360	1000	MM296	4000	198	500000

The plates listed in the *Prediction* section are user queries and the prices are predictions. The plates and their corresponding prices listed in the *Historical Examples* section are historical data from the training sample.

7 Performance Enhancements

7.1 Ensemble Models

Combining several models is known to improve prediction accuracy. Two combinations are considered in this section: a combination of the preceding neural network and [4], and a combination of these two models plus features not related to the characters on plates. In each case, the combination was conducted through linear regression, with the prediction of each model acting as features. The two models were thus implemented as follows:

$$y = \alpha + \delta_1 y_{rnn} + \delta_2 y_{woo} \tag{7}$$

$$y = \alpha + \delta_1 y_{rnn} + \delta_2 y_{woo} + \sum_i \nu_i x_i \tag{8}$$

where y_{rnn} is the prediction of the neural network, y_{woo} the prediction of [4]'s regression model with only the license-plate-specific features, and x_i a series of additional features, including the year and month of the auction, whether it was an afternoon session, the plate's position within the session's ordering, the existence of a prefix, the number of digits, a log of the local market stock index, and a log of the consumer price index. α, δ and ν were estimated by linear regression on the training data.

Both models performed better than the neural network alone. The simple combined model shown (*Combined* in Table 1) improved performance by slightly more than 1 percent as measured by R^2, while the additional features (*Combined + Extra*) improved performance by another 1.6 percentage points.

Overall, the ensemble models improved accuracy, but only by a small amount, suggesting that the neural network was successful in explaining most of the variation in prices.

7.2 Retraining Over Time

Over time, a model could conceivably become obsolete if, for example, taste or the economic environment changed. In this section, I investigate the effect of periodically retraining the model.

To retain sufficient samples for subsequent testing and retraining, the dataset was roughly divided into two. Initial training was conducted with the first 8 years of data, which contained 25,990 samples. In the subsequent 5 years, retraining was conducted yearly, monthly, or never. The best RNN-only model and the *Combined + Extra* model were used, with the sample size kept constant in each retraining. The process was repeated 30 times as before.

Figure 5 plots the median RMSE and R^2, evaluated monthly. For the RNN model with no retraining prediction, accuracy dropped rapidly by both measures. RMSE dropped an average

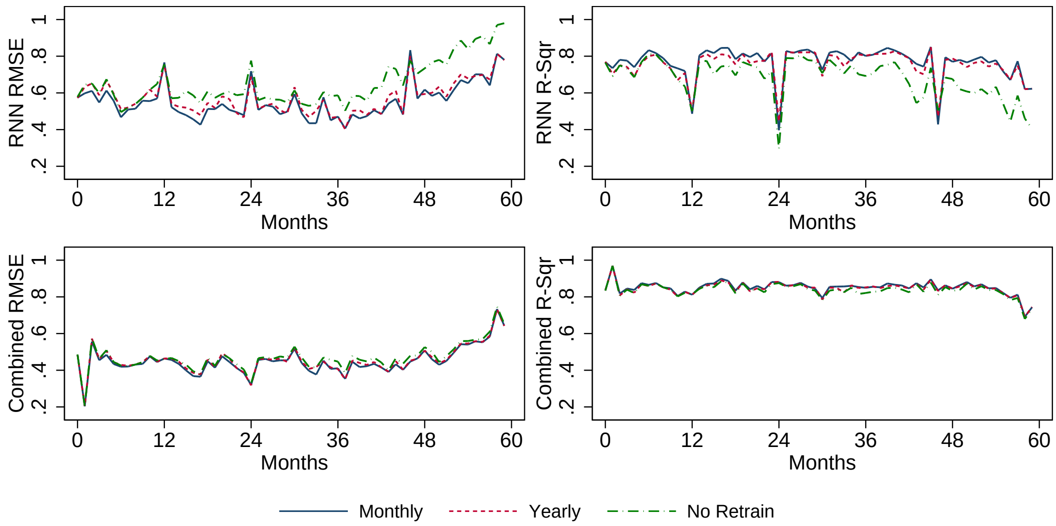

Figure 5: Impact of Retraining Frequency. The first two diagrams plot the monthly performance of the best RNN model, while the last two diagrams plot the same for the *Combined + Extra* model.

of 0.7 percent per month, while R^2 dropped 0.3 percent per month. Yearly retraining was significantly better, with a 12.1 percent lower RMSE and a 7.2 percent higher R^2. The additional benefit of monthly retraining was, however, much smaller. Compared with the yearly retraining, there was only a 3.3 percent reduction in the RMSE and a 1.5 percent increase in the explanatory power. The differences were statistically significant.[7]

For the ensemble model, the performance between different retraining frequencies was very close, with a less than 4 percent difference in the RMSE and a less than 1 percent difference in R^2 when going from no retraining to monthly retraining. Nevertheless, the differences remained statistically significant, as retraining every month did improve accuracy. The performance of the ensemble model was also considerably more stable than the RNN alone, with only half of the volatility at every retraining frequency. The primary reason behind this difference was the RNN's inability to account for extreme prices, particularly at month 12, 24 and 46. Auctions in these three months had an unusually high number of valuable plates, resulting in average sold prices that were 15 to 20 times higher than the overall average. The ensemble model was able to predict these extreme prices because [4] handcrafted features specifically for these valuable plates.

These results suggest that while there is a clear benefit in periodical retraining, this benefit diminishes rapidly beyond a certain threshold. Moreover, while deep RNN generally outperforms handcrafted features, the latter could be used to capture outliers.

[7] Wilcoxon Sign-Rank Tests:
RNN yearly retraining = RNN no retraining: $z = -6.257$, $p = 0.000$
RNN monthly retraining = RNN yearly retraining: $z = -4.923$, $p = 0.000$
Combined yearly retraining = Combined no retraining: $z = -6.062$, $p = 0.000$
Combined monthly retraining = Combined yearly retraining: $z = -5.319$, $p = 0.000$

8 Concluding Remarks

This study demonstrates that a deep recurrent neural network can provide good estimates of license plate prices, with significantly higher accuracy than other models. The deep RNN is capable of learning the prices from the raw characters on the plates, while other models must rely on handcrafted features. With modern hardware, it takes only a few minutes to train the best-performing model described previously, so it is feasible to implement a system in which the model is constantly retrained for accuracy.

Notwithstanding this good performance, several areas can be further improved.

First, the ensemble models in this study were constructed with linear regression. The use of another neural network instead of linear regression may better capture the high-order interactions between the characters on the plate and the other features, thereby further improving the performance of the model.

Second, while the model outputs only a single price estimate, auctions that provide estimates typically give both a high and a low estimate. As there is only one realized price to train on for each sample, designing a model that can output a meaningful range of estimates is a non-trivial task; a range that is too narrow will often be violated, while one that is too broad will be of no practical use.

Finally, the performance of the model on personalized plates has yet to be studied. Personalized plates contain owner-submitted sequences of characters and so may have vastly more complex meanings. Exactly how the model should be designed—for example, whether there should be separate models for different types of plates, or whether pre-training on another text corpus could help—remains to be studied.

9 Acknowledgments

I would like to thank Travis Ng for providing the license plate data used in this study.

References

[1] O. Ashenfelter, "How auctions work for wine and art," *Journal of Economic Perspectives*, vol. 3, pp. 23–36, September 1989.

[2] P. R. Milgrom and R. J. Weber, "A theory of auctions and competitive bidding," *Econometrica*, vol. 50, no. 5, pp. 1089–1122, 1982.

[3] C.-K. Woo and R. H. Kwok, "Vanity, superstition and auction price," *Economics Letters*, vol. 44, no. 4, pp. 389 – 395, 1994.

[4] C.-K. Woo, I. Horowitz, S. Luk, and A. Lai, "Willingness to pay and nuanced cultural cues: Evidence from hong kongs license-plate auction market," *Journal of Economic Psychology*, vol. 29, no. 1, pp. 35 – 53, 2008.

[5] T. Ng, T. Chong, and X. Du, "The value of superstitions," *Journal of Economic Psychology*, vol. 31, no. 3, pp. 293 – 309, 2010.

[6] N. Baba and M. Kozaki, "An intelligent forecasting system of stock price using neural networks," in *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, vol. 1, pp. 371–377 vol.1, Jun 1992.

[7] D. Olson and C. Mossman, "Neural network forecasts of canadian stock returns using accounting ratios," *International Journal of Forecasting*, vol. 19, no. 3, pp. 453 – 465, 2003.

[8] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10389 – 10397, 2011.

[9] F. A. de Oliveira, C. N. Nobre, and L. E. Zarate, "Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index v case study of petr4, petrobras, brazil," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7596 – 7606, 2013.

[10] N. Kohzadi, M. S. Boyd, B. Kermanshahi, and I. Kaastra, "A comparison of artificial neural network and time series models for forecasting commodity prices," *Neurocomputing*, vol. 10, no. 2, pp. 169 – 181, 1996. Financial Applications, Part I.

[11] W. Kristjanpoller and M. C. Minutolo, "Gold price volatility: A forecasting approach using the artificial neural networkvgarch model," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7245 – 7251, 2015.

[12] W. Kristjanpoller and M. C. Minutolo, "Forecasting volatility of oil price using an artificial neural network-garch model," *Expert Systems with Applications*, vol. 65, pp. 233 – 241, 2016.

[13] Q. Do and G. Grudnitski, "A neural network approach to residential property appraisal," *The Real Estate Appraiser*, p. 38V45, 1992.

[14] A. Evans, H. James, , and A. Collins, "Artificial neural networks: An application to residential valuation in the uk," *Journal of Property Valuation and Investment*, vol. 11, p. 195V204, 1992.

[15] E. Worzola, M. Lenk, and A. Silva, "An exploration of neural networks and its application to real estate valuation," *Journal of Real Estate Research*, p. 185V201, 1995.

[16] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International Journal of Forecasting*, vol. 30, no. 4, pp. 1030 – 1081, 2014.

[17] G. Dudek, "Multilayer perceptron for {GEFCom2014} probabilistic electricity price forecasting," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1057 – 1060, 2016.

[18] R. Sharda and D. Delen, "Predicting box-office success of motion pictures with neural networks," *Expert Systems with Applications*, vol. 30, no. 2, pp. 243 – 254, 2006.

[19] L. Yu, S. Wang, and K. K. Lai, "Forecasting crude oil price with an emd-based neural network ensemble learning paradigm," *Energy Economics*, vol. 30, no. 5, pp. 2623 – 2635, 2008.

[20] L. Zhang, J. Luo, and S. Yang, "Forecasting box office revenue of movies with {BP} neural network," *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 6580 – 6587, 2009.

[21] M. Ghiassi, D. Lio, and B. Moon, "Pre-production forecasting of movie revenues with a dynamic artificial neural network," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3176 – 3193, 2015.

[22] A. Iseri and B. Karlik, "An artificial neural networks approach on automobile pricing," *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 2155 – 2160, 2009.

[23] Z. Haofei, X. Guoping, Y. Fangting, and Y. Han, "A neural network model based on the multi-stage optimization approach for short-term food price forecasting in china," *Expert Systems with Applications*, vol. 33, no. 2, pp. 347 – 356, 2007.

[24] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1 – 8, 2011.

[25] L. Bing, K. C. C. Chan, and C. Ou, "Public sentiment analysis in twitter data for prediction of a company's stock price movements," in *2014 IEEE 11th International Conference on e-Business Engineering*, pp. 232–239, Nov 2014.

[26] V. S. Pagolu, K. N. R. Challa, G. Panda, and B. Majhi, "Sentiment analysis of twitter data for predicting stock market movements," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System*, 2016.

[27] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, Oct. 2014.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3104–3112, Curran Associates, Inc., 2014.

[29] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *CoRR*, vol. abs/1409.2329, 2014.

[30] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," *Proceedings of The 33rd International Conference on Machine Learning*, p. 173V182, 2016.

[31] J.-W. Ha, H. Pyo, and J. Kim, "Large-scale item categorization in e-commerce using multiple recurrent neural networks," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 107–115, ACM, 2016.

[32] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, Nov 1997.

[33] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio, "Batch normalized recurrent neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2657–2661, March 2016.

[34] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[35] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.