# Training a Neural Network to Predict House Rents Using Artifical Intelligence and Deep Learning

Chung-Hsing Chao*
Department of Intelligent Vehicles and Energy, Minth University of Science and Technology, Hsinchu, Taiwan, China
davidee@mitust.edu.tw
*Corresponding author

2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA) | 979-8-3503-1467-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIPCA59209.2023.10257703

***Abstract*—We are training neural networks to predict house rents using artificial intelligence and deep learning, which is being used in the real estate and financial industries. Real estate agents, financial institutions, and real estate developers can benefit from rent forecasting, which is an important application scenario. Many real estate websites now use machine learning models to predict rents to help tenants and landlords find the right rental price. Machine learning models are also being used by some real estate companies and financial institutions to determine rents more accurately. Training neural networks to predict rent is still a relatively new field that needs more research and experimentation to improve its performance and accuracy.**

***Keywords—Artificial intelligence, deep learning, and rental housing prices***

## I. Introduction

Rent market transparency can be improved by using neural networks to predict rent. Consumers can better understand the rent market by using a neural network model, which allows them to make better rent-related decisions; landlords and tenants can make smarter decisions; rental market efficiency can be improved: A more accurate forecast of rents will help the rental market allocate resources more efficiently, which will boost its efficiency. Applying neural network technology to predict rents could help the real estate industry respond better to market changes, thereby promoting its growth.

Market monopolies can also result from neural networks: some large real estate developers and financial institutions may use them to predict rents to increase profits. In the absence of information asymmetry, some tenants and landlords may miss out on opportunities or be treated unfairly. A neural network may infiltrate personal information since it requires a large amount of data to train. Using neural network rent prediction techniques widely may result in some real estate-related workers losing their jobs, particularly those who work in rent evaluation.

The improved regression model developed by Chen, Hu, Nian and Yang (2020) [1] works well. Ghosh, Jana, and Abedin (2023) [2] proposes a deep learning approach for predicting rental prices in online accommodation marketplaces, such as Airbnb. Oshodi, Ohiomah, Odubiyi, Aigbavboa, and Thwala (2019) [3] present a neural network approach for predicting rent prices. Lazcano, Herrera, and Monge (2023) [4] proposes a recurrent neural network (RNN) approach for rent forecasting, which takes into account the temporal nature of rental prices. Zulkifley, Rahman, Ubaidullah, Ibrahim (2020) [5] compares the performance of different machine learning algorithms, including neural networks, for predicting house price prediction.

Through this machine learning pipeline, we can use Python and machine learning libraries such as TensorFlow to train a neural network to predict rent. Important steps include data collection, data preprocessing, dataset segmentation, model building, model training, model tuning, model testing, and model deployment. This pipeline requires some machine learning basics, such as neural network model selection and hyperparameter tuning, but it can provide an efficient approach to the rent prediction problem. In this study, training a neural network to predict house rent can be applied to many other domains and can help us build effective machine learning models.

## II. Analysis

Training a neural network to predict rent uses the following steps:

1) Data collection: download Taiwan housing rent data as shown in Figure 1. This data collection is collected from "591 Housing Trading Network" and the location is "Zhongshan District, Taipei City". There are 4 characteristics in "Area", "Floor", "Is it possible to run mess", "Is it possible to keep pets", and there are corresponding "House-rent prices". Total data: 689, the number of characteristics:4, tag: rent price, 591 housing transaction network address: https://www.591.com.tw/.

2) Data pre-processing: converting the collected data into a usable data format for the neural network. This includes steps such as data cleaning, data transformation, and data scaling. For example, print(data [1], label [1]); print (data [5], label [5]) [9,3,1,0] [16999.] [6,5,0,0] [9000.] From the above results, it is indoor 9 pings, located on the 3rd floor, can run a mess but cannot keep pets, and the monthly rent is NT$16,999 yuan; while 6 pings, located on the 5th floor, cannot run a mess, and cannot keep pets, The monthly rent is NT$9000 yuan.

3) Data segmentation: training set, validation set, test set: Before training the neural network, the data is usually divided into "training set", "verification set" and "test set". Training set is like the sample questions written by students when studying; Verification set is just Like exercises; test sets like final exams. When the neural network is trained, it will only see the data of the training set. During the training process, the verification set will be used to simulate the test. After the training is completed, it can be tested with the test set to see how well it has learned. The Internet is not a student who can only memorize answers, but really can solve problems. The difference between the verification set and the test

set is that the neural network will use the verification set to see the current effect immediately after each training cycle, while the test set will only see the results after all the training cycles are over. Using the verification set allows us to ensure that the training direction of the neural network is correct, and once a deviation occurs, the training can be stopped immediately.

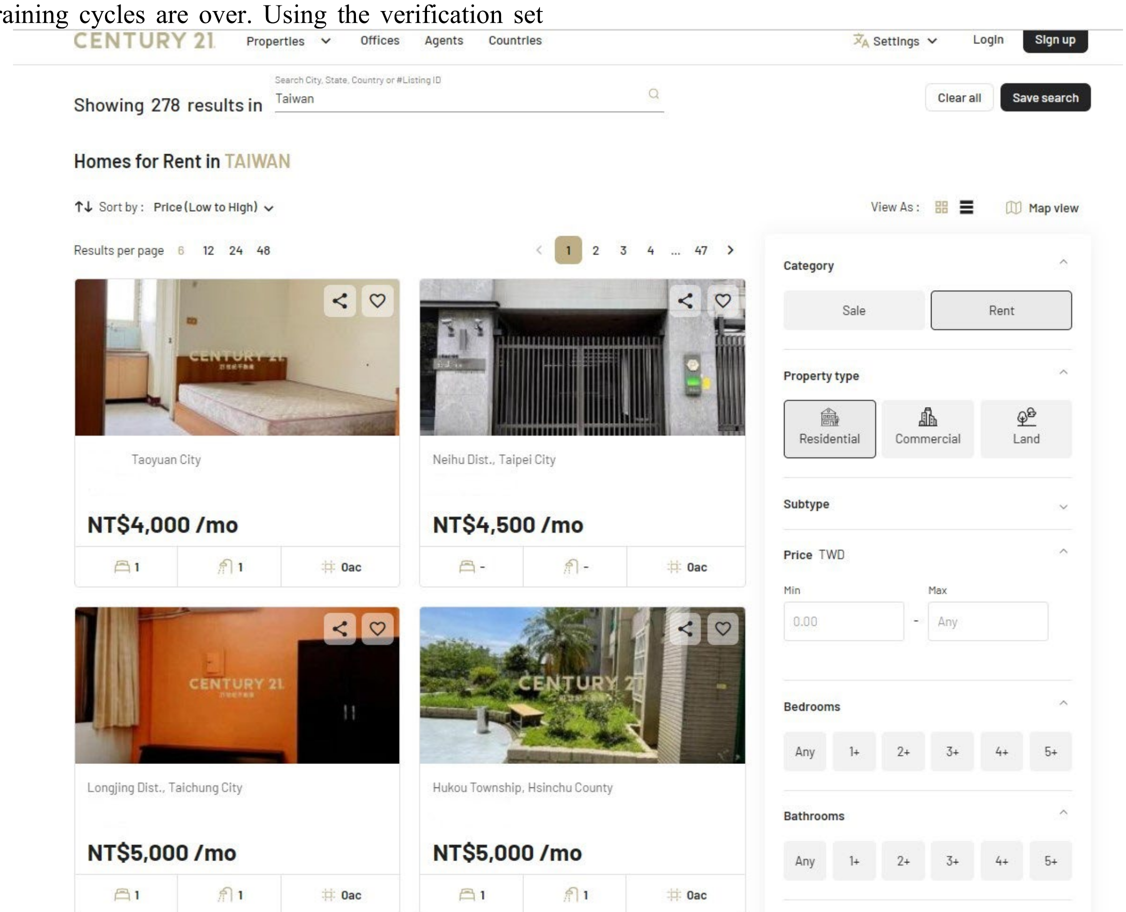


Figure 1: 591 Housing Trading Network.

4) Data normalization: The attributes and ranges of each feature value are different. For example, the range of "squares" in the data is between 4 and 26, and "can you keep pets" is only 0 (no) and 1 (yes). 1 (1-0) is already the biggest change for "Keeping pets", but for "Number of pings", the change of 1 is only a difference of 4 pings to 5 pings. Currently, the greater the value, the greater the impact on the weight. The larger the numbers, directly bringing them into the neural network will result in poor training results. Therefore, before bringing it into the neural network, it is necessary to "normalize" the data first, that is, let each feature use the same "measurement standard". There are more than one normalization methods. Here, we choose to subtract the average from the data, and then divide it by the standard deviation. We call this operation process "Standardization". After normalization, each data is based on 0, and the standard deviation is used as the unit. The "mean" and "standard deviation" used by regularization only include the training set, instead of using all the data. This is because the "validation set" and "test set" other than the training set will test the model later, if the validation set and the test set are also used to calculate the average and standard deviation at this time, it will cause data leakage (will The information of the unknown sample is leaked to the model, so that the false impression that the prediction result is good in the subsequent test). In addition to eigenvalues, labels also need to be normalized, because normalization can not only solve the problem of different units, but also make the value smaller, which is more conducive to the training of neural networks. As mentioned earlier, there is more than one way of normalization, and the label is divided by the maximum value of the label (to facilitate subsequent restoration).

5) Building a Model - Sequence Model: Keras is a Python language deep learning development tool. It can easily and quickly construct a neural network (see Figure 2). The neural network is built layer by layer like building blocks. It supports image recognition, text, sound, and other content. You can choose (ex: CNN, RNN, etc), the code does not need to make any changes when changing the hardware environment (ex: CPU, GPU). Sequential models can be like building blocks, where neural networks are stacked layer by layer to build a linearly stacked model. As for the way of stacking neural layers, in addition to using the add() method of the model, it can also be changed to directly put each neural layer in a sequence as its parameter when constructing the model object, so the 2 created by the following program. Create a model sequence model and add the first layer to include the input layer information. The activation function of Rectified Linear Unit, ReLU, commonly used in the second hidden layer. The activation function of ReLU is defined as shown in Figure 3.

$$f(x) = \max(0, x) \quad (1)$$

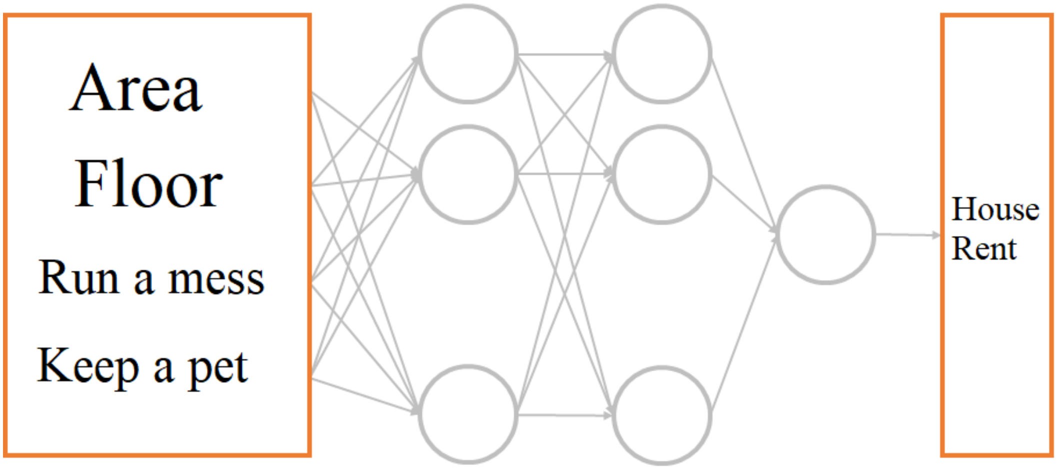


Figure 2: Establishing Correspondence Function Between Two Groups of Data-Regression Problem.

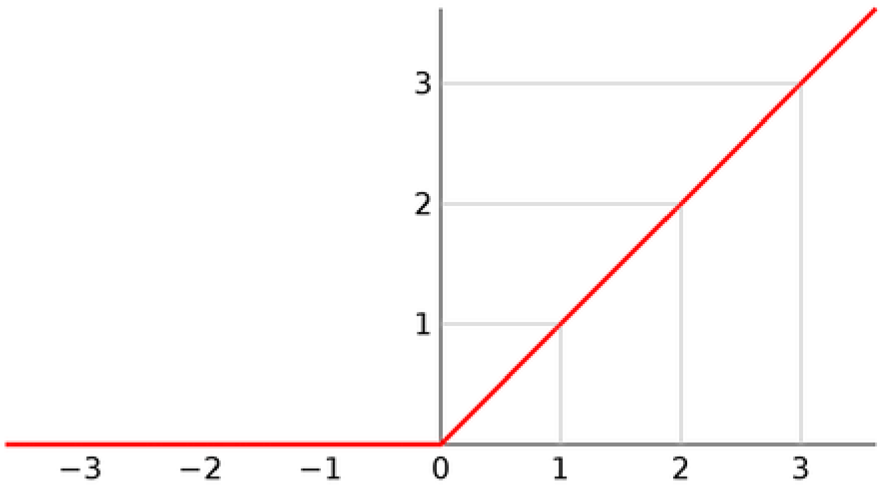


Figure 3: The activation function of "ReLU" (Rectified Linear Unit).

6) Compile model: Compiling the model refers to telling the model how to train, so you must define: loss function, metrics evaluation criteria, optimizer. Keras is TensorFlow's high-level API for building and training deep learning models. This API can be used for rapid prototyping, cutting-edge research and production environments. It and the loss function are usually inverse indicators of each other, that is, the better the evaluation effect, the smaller the loss value. So Keras also allows us to directly use the loss function as the metrics function if it is interpreted in reverse. The results returned by metrics will only be used to evaluate the performance, not for training because the optimizer will only use the loss value returned by the loss function for optimization, which has nothing to do with the evaluation performance. Optimizer is a method that can speed up the training of neural networks, and even make the training effect better. It will transform the gradient, and then pass the recalculated update value to the weight and bias value. Adam (Adaptive Momentum Estimation) optimization is a stochastic gradient descent method that is based on adaptive estimation of first order and second-order moments, which can be said to be a combination of the two Optimizers of Momentum and AdaGrad (Adaptive Gradient). Adam keeps the exponentially decaying average of past gradients like Momentum and the squared decaying average of past gradients like as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)(\partial L_t / \partial W_t) \tag{2}$$

$$v_t = \beta_2 m_{t-1} + (1 - \beta_2)(\partial L_t / \partial W_t)^2 \tag{3}$$

It also does "deviation correction" for the parameters, so that each learning rate will have a definite range, which will make the updating of parameters and bias correction for $m_t$ and $v_t$. This will make the parameter update more stable.

$$\widehat{m}_t = m_t / (1 - \beta_1^t) \tag{4}$$

$$\hat{v}_t = v_t / (1 - \beta_2^t) \tag{5}$$

Adam Weight update equation as follows:

$$W \leftarrow W - \eta[\hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)] \tag{6}$$

7) Training model: After the model is compiled, you can execute fit() for training. ...The following explains several common parameters: At the end of each training cycle, this data will be used to calculate the verification loss and evaluate the results. To verify the performance of the model on unseen data. But please note that this information will not be used for training.

8) Outcome assessment: Loss Curve - It is the purpose of the neural network to make the loss value smaller, so displaying the curve of the loss value can more clearly confirm whether the neural network is moving towards the goal. The MAE (Mean Absolute Error) of regression losses is defined as. The average of absolute differences between the actual and predicted value.

$$MAE = \sum_{i=1}^{n} |y_i - x_i|/n \tag{7}$$

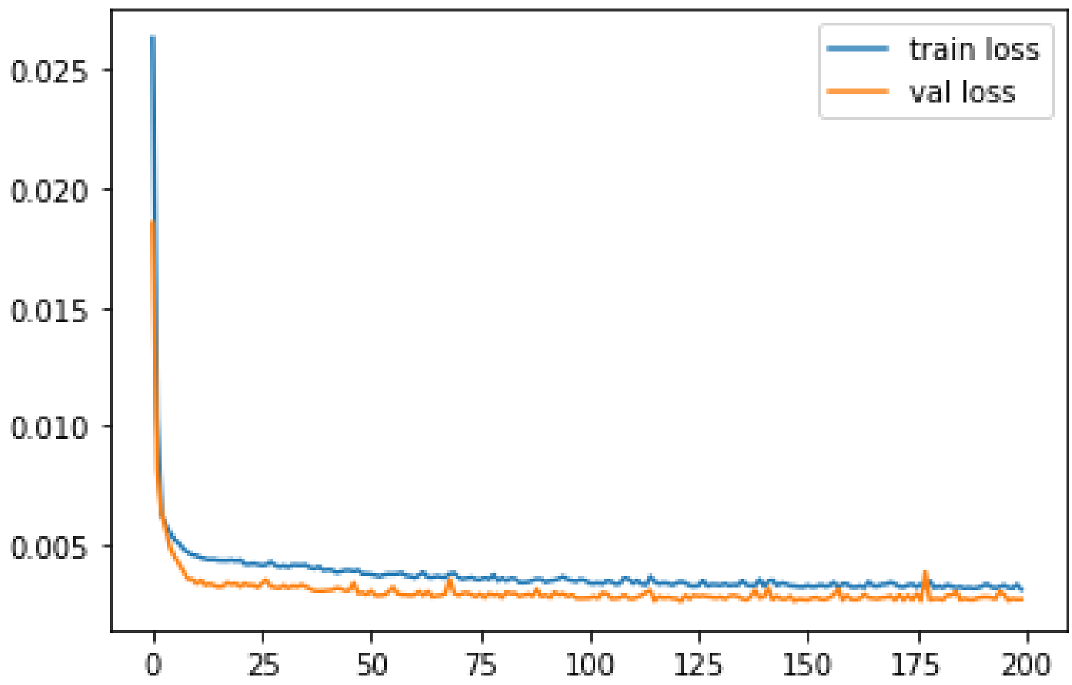


Figure 4: Train loss and validation loss.

## III. Result

The purpose of the neural network is to make the loss value of the research data smaller. Figure 4 shows the curve of the loss value, including the loss value of the training data and the loss value of the verification data. It can be seen from the figure that the loss value of the verification data is lower than the loss value of the training data, which means that the neural network model is confirmed to be moving towards the goal.

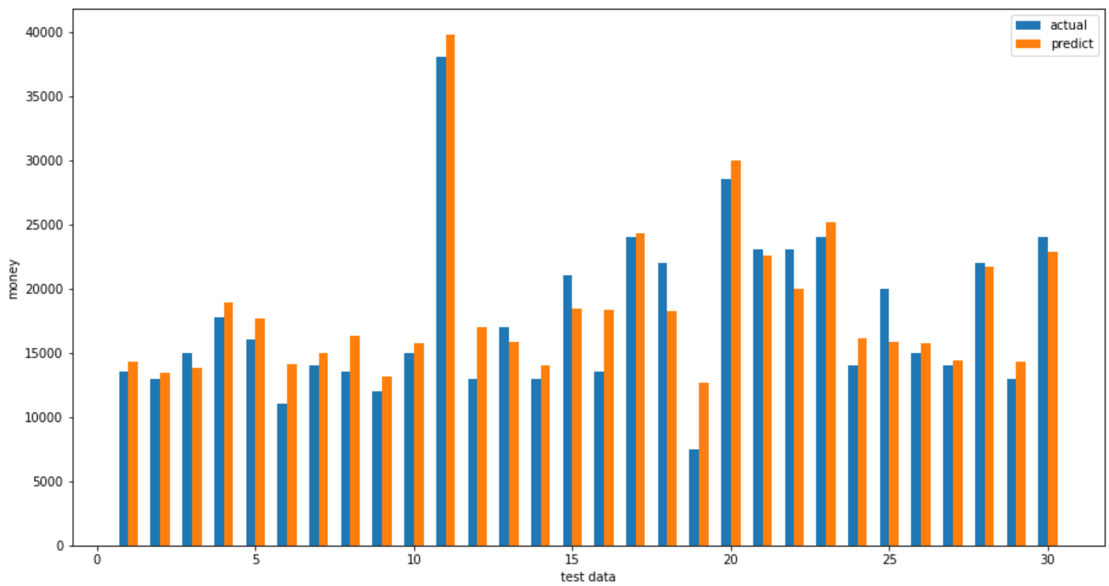


Figure 5: Comparison Chart.

After obtaining the housing rent prediction model, it is necessary to use it to predict the rent as shown in Figure 5. The blue is the actual rent, the orange is the predicted rent, the x-axis is the number of data, and the y-axis is the price (unit: Taiwan dollar). The prediction includes 30 test data brought into the model, and the result of "predicted rent" should be compared with the actual rent, and the two are subtracted to obtain the error distribution map in Figure 6. The blue in the comparison chart (Figure 5) is the actual rent, the orange is the predicted rent, the X-axis is the number of data, and the Y-axis is the price (Unit: NT$). It is not easy to see whether the forecast is good or bad. Therefore, the following uses the error distribution diagram in Figure 6 to analyze.

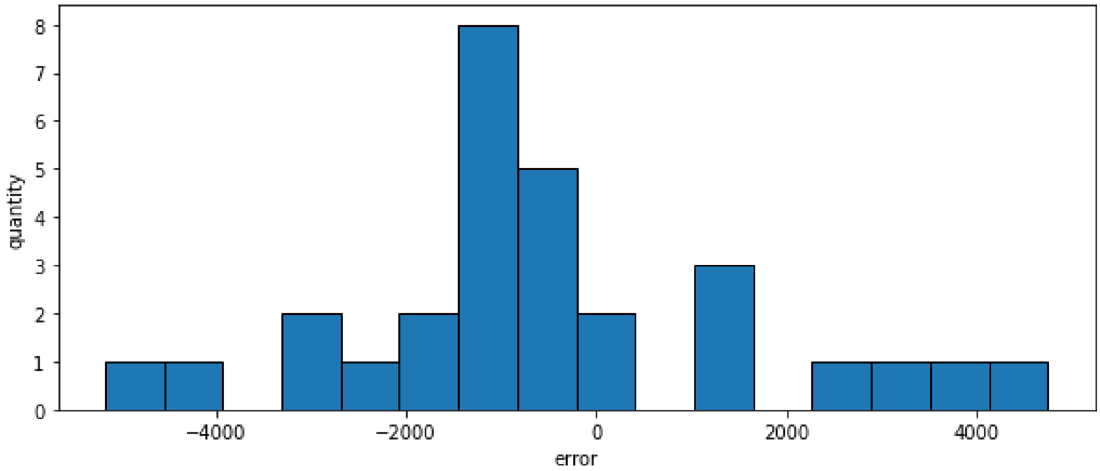


Figure 6: Error distribution map.

The actual rent minus the predicted rent gets the "error value" and divides all the error value range into 15 equal parts and obtains the amount of data for each equal part. The x-axis is the error range (unit: Taiwan dollar), and the y-axis is the number of data items (unit: number of houses). From the error distribution diagram, it can be clearly seen that the error digits are between -NT$2,000~+NT$2,000, and only a few data have relatively large errors.

## IV. CONCLUSIONS

In this paper, we use the 591-rent trading network to obtain five types of data: the number of square feet, floor, availability of a restaurant, availability of pets and price, to build a neural network model, and use the first four inputs to predict the fifth rent price. In this experiment, only 4 features using a regression prediction model can be used to predict the rent very accurately.

## REFERENCES


[1] Chen, D. Hu, F. Nian, G. Yang, T. (2020) Deep Residual Learning for Nonlinear Regression, Entropy, 22, 193-217

[2] Ghosh, I. Jana, R.K. Abedin, M.Z. (2023), "An ensemble machine learning framework for Airbnb rental price modeling without using amenity-driven features", International Journal of Contemporary Hospitality Management, Vol. ahead-of-print No. ahead-of-print.

[3] Oshodi,O. Ohiomah, I. Odubiyi,T. Aigbavboa,C. Thwala, W. (2019) Forecasting Rental Values of Residential Properties: A Neural Network Model Approach, 11th International Conference (CITC-11) September 9-11, 2019, London, UK.

[4] Lazcano, A. Herrera, P.J. Monge, M. (2023) A Combined Model Based on Recurrent Neural Networks and Graph Convolutional Networks for Financial Time Series Forecasting, Mathematics, 11, 1, 224-245.

[5] Zulkifley, N.H. Rahman, S.A. Ubaidullah, N.H. Ibrahim, I. (2020) House Price Prediction using a Machine Learning Model: A Survey of Literature, I.J. Modern Education and Computer Science, 6, 46-54.