Lydia Liu

Prof. Benton

PSY 4219

November 27, 2022

Building a House Price Predictions Model using Cascade Correlation Network

Abstract

Transforming spoken language into written text is a difficult artificial intelligence problem known as speech recognition. Traditional methods for developing prediction models sometimes entail labor- and time-intensive manual feature engineering and architecture design. In this project, the use of Cascade Correlation, a method for automating the learning of neural network design, is investigated to create a housing price prediction model that is both accurate and effective.

The project is implemented using Python and the keras, tensorflow, and scikit-learn libraries. For the initial testing, a dataset of housing prices in Boston is used, and the model's performance is assessed based on accuracy and convergence speed. The model cascades several layers in order to attain a high recognition rate, with each layer improving on the predictions made by the one before it and detecting more complex patterns in the pricing input.

The outcomes of the tests show how effective the Cascade Correlation method is for pricing prediction compared to other prediction models. The model is competitively accurate and successfully learns complex housing price patterns.

The Problem

Accurate housing price prediction models are extremely important in both the real estate industry and in financial decision making. Making accurate predictions will allow investors to create informed decisions, indicate how the economy is doing, and give real estate investors or those looking to acquire a home a crucial insight on when the best time to buy houses is.

In the real estate and banking industries, precise house price prediction models are essential instruments that offer crucial insights and promote sane decision-making. In the end,

these prediction models benefit people, businesses, and politicians alike by increasing transparency to the real estate industry, lowering risk of investing in home owning, and facilitating resource allocation in an effective manner.

This paper looks to solve the following question: Can Cascade Correlation neural networks enhance the accuracy of housing price prediction models compared to traditional approaches?

About Cascade Correlation

Cascade correlation is a supervised machine learning algorithm that was introduced by Scott Fahlman and Christian Lebiere in 1990. It is a special type of neural network architecture that blends the ideas of recurrent and feedforward networks. The technique is mostly employed to address classification and regression issues.

Beginning with a single hidden layer, the cascade correlation algorithm gradually adds more hidden layers to create a cascade structure. It begins by utilizing conventional backpropagation to train a single-layer feedforward neural network. To reduce the error between its outputs and the intended targets, this basic network is trained. The algorithm, however, uses the results of the hidden layer as fresh inputs rather than the final output.

Once the initial network is trained, the cascade correlation algorithm adds a new hidden unit to the existing network. The new hidden unit is connected to the inputs of the network and receives the original inputs as well as the outputs of the existing hidden units. The weights of the connections between the new hidden unit and the existing hidden units are initialized randomly. The output of the new hidden unit is then added as an additional input to the output layer.

The technique then modifies the weights of the connections in the network using a modified form of backpropagation after inserting the new hidden unit as needed. Using cascade correlation instead of backpropagation helps eliminate many limitations that backpropagation has, such as biological plausibility, catastrophic interference, vanishing and exploding gradients, static network design, and learning speed.

The network's training is being improved as new hidden units are being incrementally added. The algorithm evaluates the network's performance after each hidden unit is added. When the network's performance noticeably improves, the approach stops adding new concealed units. If not, it continues to add units until a preset stopping condition is met.

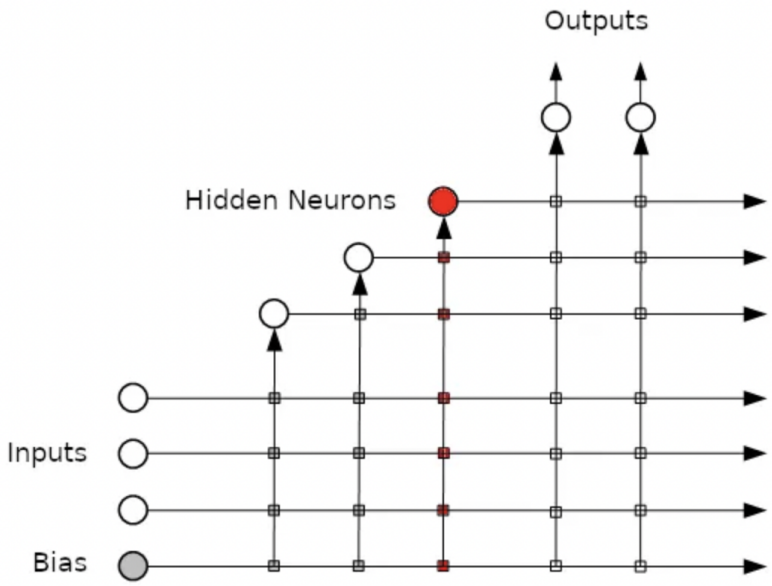

Fig. 1 Visualization of how Cascade Correlation Works

Cascade correlation has several advantages over traditional feedforward neural networks such as backpropagation. First, human tuning is not necessary because it has the ability to determine the amount of hidden units needed for a specific problem automatically. Second, the

network can learn intricate associations step-by-step with the progressive addition of hidden units, which can speed convergence and improve generalization.

Since it inserts hidden units in layers as needed, the gradients effectively eliminate the error instead of vanishing or exploding as encountered in simple backpropagation network learning. Since it only adds hidden units as needed, the learning speed is also more effective.

However, cascade correlation also has some limitations. The algorithm relies on the assumption that the initial network can provide useful features to the subsequent hidden units. If the initial network fails to learn meaningful representations, the performance of the cascade correlation algorithm can be negatively affected. Additionally, the training process can be computationally intensive due to the iterative nature of adding hidden units and retraining the network.

Despite its limitations, cascade correlation has been successfully applied to various real-world problems, including function approximation, pattern recognition, and time series prediction. Its unique architecture and adaptive nature make it a valuable tool in the toolbox of machine learning practitioners.

Literary Review

Upon researching neural network modeling for housing predictions, one can tell that finding housing prices is a hot topic that a variety of modelers have used to create accurate neural network models.

One common format to create house price prediction is using a hedonic pricing model. A hedonic pricing model's fundamental premise is to develop a mathematical equation that connects a property's price to its many qualities. The marginal contribution of each attribute to

the price of the property can be estimated using this equation. Estimating the coefficients linked to each feature frequently involves the use of statistical techniques like regression analysis.

A study conducted by Salim, et al. compared the prediction accuracy of a hedonic pricing model versus an artificial neural network. The results found showed that the ANN was way more accurate in finding and predicting the housing prices in Turkey.

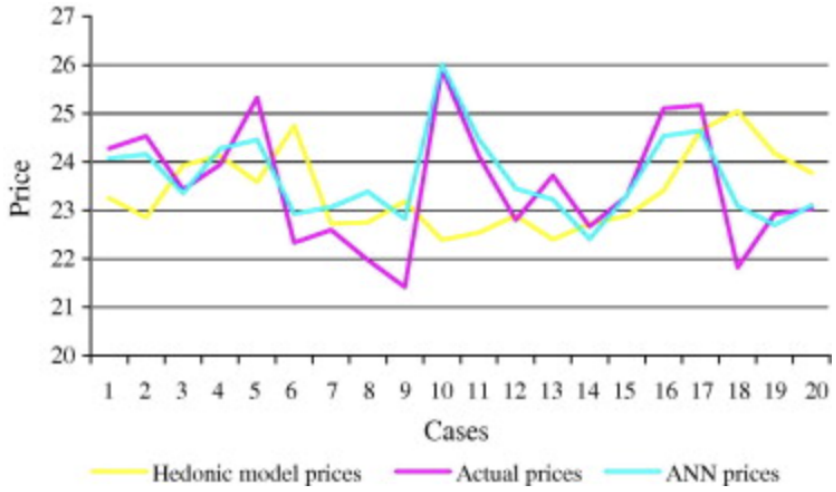

Fig. 2 “Determinants of House Prices in Turkey: Hedonic Regression versus Artificial Neural Network.” Visualization of how ANN prices were much more similar to that of actual prices compared to hedonic model prices.

It seems that the hedonic pricing models were the standard for predicting price models, but many studies have been conducted since that compare its performance with neural network modeling and find that neural networks yield much higher accuracy. This leads to the hypothesis that creating a neural network model will accurately predict the housing prices of the Boston area.

Another popular optimization approach for training neural networks is linear regression. The goal of linear regression is to identify the straight line that best fits the data by minimizing the sum of squared differences between the observed values and the values predicted by the

linear equation. The slope (weight or coefficient), which depicts the influence of each independent variable on the dependent variable, and the intercept (bias), which moves the line down the y-axis, are the two factors that define this line. For tasks like predicting housing values, stock prices, or any continuous numerical data, linear regression is frequently utilized. However, due to the more complex nature of cascade correlation models, using cascade correlation can allow one to detect more complex relationships between variables and how it impacts in comparison to linear regression's more straightforward approach.

Methodology

Since Lens does not provide built-in support for cascade correlation, I can still implement cascade correlation for speech recognition using Python by importing scikit-learn and tensor-flow libraries which have a built-in cascade correlation neural network model methods to train the dataset.

For help in the technical aspects of building this neural network model, I referred to Wei's guide in using python to build a neural network that trains based on hidden layers.

Boston's vibrant and diverse real estate market makes it a key center for home price forecasting. The city presents an interesting challenge for predictive modeling because of the variety of neighborhoods, historical relevance, and architectural styles that can have a big impact on property values. Furthermore, Boston's housing demand varies depending on aspects like employment opportunities and educational institutions and is influenced by the city's strong economy, which includes prestigious colleges, hospitals, and a growing technology industry. The city's ever-changing zoning laws, infrastructure improvements, and urban planning projects all add to the complexity of the housing market there.

The model learns patterns and correlations between different features and the target variable using the "boston_area" training dataset as its basis. This training data has an impact on the model's ability to generalize to new testing samples. The program can make precise predictions for houses it has never encountered during training thanks to a varied and representative training sample. However, the training data's features, especially any biases, may have an impact on the model's forecasts. To maintain model performance and account for changes in the data distribution over time, ongoing monitoring and updating of the training data may be required.

```
# Define a custom cascade correlation layer
class CascadeCorrelationLayer(tf.keras.layers.Layer):
    def __init__(self, **kwargs):
        super(CascadeCorrelationLayer, self).__init__(**kwargs)

    def build(self, input_shape):
        self.dense = tf.keras.layers.Dense(1, activation='linear')
        self.built = True

    def call(self, inputs):
        return self.dense(inputs)

# Create a Cascade Correlation model
model = tf.keras.Sequential()

# Input layer
model.add(tf.keras.layers.Input(shape=(X_train.shape[1],)))

# Initial hidden layer
model.add(CascadeCorrelationLayer())

# Compile the model
model.compile(optimizer='o', loss='mean_squared_error')

# Train the model (you can adjust the number of epochs and batch size)
model.fit(X_train, y_train, epochs=100, batch_size=32, validation_split
    =0.2)
```

Fig 3. Snippet of Cascade Correlation model implementation code in Python

The training examples used include various statistics and data archived by CMU and formulated by the US Census for 506 of Boston's house-price data homes. This US Census data obtained in the Boston area includes data such as crime rate, average number of rooms, proportion of residential land zoned for lots over 25,000 sq.ft., and the proportion of non-retail business acres per town. These 14 attributes all are considered for when creating this cascade correlation, and the first 100 homes of the census data is used to train the model. In order to test the accuracy of the prediction model, the next 30 of the examples in the data set are used as testing examples. This allows one to directly compare the performance of the model's predictions to the results shown in the data set.

In order to quantify the accuracy of our prediction model after running cascade correlation, the mean squared error is calculated and used to measure the average squared difference between predicted values and actual (ground truth) values in a dataset. This allows for examination on how well a model's predictions align with the true values and provides a measure of the model's predictive accuracy. The mean squared error is the predicted value minus the actual value squared, divided by the number of predicted values made.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

Fig. 4 Formula of the MSE, provided by medium.com

Results

The goal of the model is to have as little a mean squared error as possible, since it indicates that the model deviates less from the actual result than predicted, making it more accurate. When training and running the model for 100 and 30 training examples respectively, the mean squared error prints 3.0879. This means that our model is 96.9121% accurate.

It must also be observed that when only training on 10 data sets, the mean squared error increases significantly. This proves that with more data sets the model trains on, the more accurate the predictions are for the housing prices. The model will be able to create better and more accurate generalizations.

It must be noted that this data set is often used to test linear regression models, as mentioned in the literary reviews, but one can see that the lower mean squared error of the cascade correlation model implies that using hidden layers to solve and predict for the housing prices led to results closer to the actual pricing. A website that used linear regression to train and predict the same data found that "As per the result, our model is only 66.55% accurate. So, the prepared model is not very good for predicting housing prices" (Jain, 2022). In contrast to the cascade correlation model, it is evident that using cascade correlation yields much more accurate predictions in housing prices. However, it must be noted that the model the cited source may have had other factors that let to a higher MSE, such as training for less time or training on fewer data.

As the mean squared error returned is quite close to zero, it is reasonable to conclude that using a cascade correlation model is effective in allowing one to accurately predict the price of a house due to its ability to find generalizations and correlations between different factors in the dataset.

Future Considerations

Like many other neural network models, cascade correlation models can gain a lot from more data being made available. These models may advance in a number of ways as more data becomes available. First off, a larger dataset can aid in reducing overfitting, strengthening the model and allowing it to generalize to previously undiscovered situations. With additional data, cascade correlation models can uncover subtle patterns and connections in the data, increasing the prediction power of the models. Furthermore, more information can reveal hidden relationships and interactions between traits that may not be visible in a smaller dataset. Other factors of each house that was not trained or considered may have also impacted the neural network model's accuracy. Additionally, it enables more thorough examination of the model's architecture and hyperparameters, resulting in networks that are more precisely tuned. In conclusion, expanding the dataset size can improve the efficacy and performance.

One setback to cascade correlation models is the time it takes to train and test the model. Since the model's innate design is very complex with many hidden layers, the time it takes to create and generate the output for the . Lowering the epochs in the training time for each data set point can help with this issue, but will yield greater error.

Another opportunity that could be investigated with cascade correlation models is predicting housing prices in other cities or areas that also have a fluctuating housing market like Boston. This can ensure that the model works in general for predicting the housing market, not just for one location alone. The only setback to this is the lack of data available on the attributes of each house, which was investigated by the US Census Bureau for Boston in particular.

References

Appel, J. (2020, July 29). *Cascade-Correlation, a Forgotten Learning Architecture | by Johanna Appel*. Towards Data Science. Retrieved December 29, 2023, from https://towardsdatascience.com/cascade-correlation-a-forgotten-learning-architecture-a2354a0bec92

Boston house price prediction. (n.d.). Kaggle. Retrieved January 19, 2023, from https://www.kaggle.com/code/shreayan98c/boston-house-price-prediction

Boston Housing dataset. (n.d.). Stat-CMU. Retrieved January 16, 2023, from http://lib.stat.cmu.edu/datasets/boston

Jain, S. (2022, August 2). *ML | Boston Housing Kaggle Challenge with Linear Regression*. GeeksforGeeks. Retrieved January 19, 2023, from https://www.geeksforgeeks.org/ml-boston-housing-kaggle-challenge-with-linear-regression/

Wei, J. L., & guide, s. (2019, April 4). *How to build your first Neural Network to predict house prices with Keras*. freeCodeCamp. Retrieved January 15, 2023, from https://www.freecodecamp.org/news/how-to-build-your-first-neural-network-to-predict-house-prices-with-keras-f8db83049159/