1D CNN Non linear Regression model

Sonal Yadav
Master in Computer science
Lakehead University
Thunder Bay, ON
syadav3@lakeheadu.ca

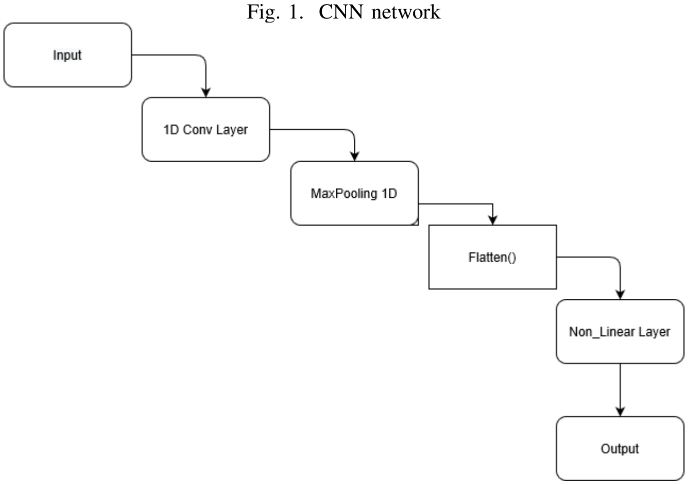
***Abstract*—In this project a 1D convolution neural network is proposed for predicting the housing values on the housing dataset. CNN is a feed forward network with alternating convolutional and subsampling layers. The main aim of the proposed model is to generate a high precision nonlinear regression model. The model is tested on various parameters and the accuracy for the dataset is calculated using the mean square error and R2 score. The report covers the various training parameters responsible in 1D CNN network and also the factors like kernel size and the CPU utilization time. It also highlights the detailed steps towards building the nonlinear regression model along with the varying results with combination of different training parameter. The model also aims at using the various CNN layers which in turn will help in achieving the desired accuracy.**

***Index Terms*—CNN, L1 error, R2 score, RELU**

I. Introduction

Convolutional Neural Networks (CNN) are feed forward neural network with alternating convolutional and subsampling layers. CNN represents a feature function which are applied on words or data to extract high level features. They use multiple layer perceptron which requires minimal preprocessing. Convolution Neural Network is basically used for 2D data classification such as images. A CNN is usually a feature map that corresponds to the features of the input. In text analysis the words in a sentence or a document or sequence of characters. CNN works well with text data also even though its one form of data and features can be extracted. The convolutional layer is followed by a max pooling layer which helps in extracting maximum features and reducing the dimension of the input to that layer. The CNN network basically involves two main operations which are convulsion and pooling. The output of these operations are fed to the fully connected layer which works the same as any traditional neural network. Regression analysis is a form of predictive analysis technique which helps in building a relation between the target and the predictor values. It helps us to compare effect of variables on different scales which help building a good predictor model. Linear regression establishes relation between dependent variable and independent variables using the straight line. Logistic regression is used when dependent variable is a binary value and is represented in a non linear form. In this report we focus on generating a non linear regression model which establishes the relation between the housing median value and the independent variables.

Fig. 1. CNN network



II. Background

The Convolution neural network architecture commonly used 2D CNN as it has many hidden layers and has ability to learn complex data objects and patterns. They are exclusively used for images and videos. This led to introduction of 1D CNN network architecture, as it has a relatively small architecture which require simple array operations. It is easier to train and test the data using 1D CNN model. Along with less computation time, 1DCNN has low hardware requirement and is best suited for real time and low-cost applications. The configuration of 1DCNN is formed using various important parameters: 1. The hidden CNN layers and neurons 2. Kernel 3. Max pooling layer 4. Activation functions The major advantage of 1DCNN is that it results in low computational complexity, since the convolution is linearly weighted.

The first layer is the convolution layer where the convolution of the input image with the kernel is performed. All the convoluted values are then stored in a feature map, the values filled in the feature map is obtained by moving the kernel over the input image by a specific stride value. After this step is the pooling layer or subsampling of data. This operation consists of applying some operation on the feature map and extracting some specific value for each region. The most commonly used pooling technique is max pooling which selects the maximum values in the input feature map of each step. Padding is another term which is commonly used in CNN network, in order to minimize the data loss by using the kernel. It refers to the number of pixels added to any image which is being processed by the kernel. After each convolution layer, there is a activation layer RELU, which trains the data much faster and alleviated the problem of vanishing gradient descent. The RELU layer just changes the negative values to zero and keeps the positive value as it is. After all the feature extraction done in the convolution and pooling process there comes the full connected layers for the last step where the network is trained for classification

III. Proposed model

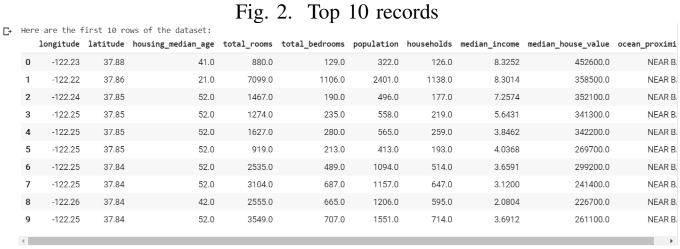
The model proposed in this project a non linear regression model based on 1D CNN, which has been trained over different parameters in order to find the best possible solution. There are various parameters which are kept in mind while building this model: Modular coding: Modular programming is the process of subdividing a computer program into separate sub-programs. It enables to work and debug the pieces of the program independently. Over/under fitting issue: Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. Underfitting refers to a model that can neither model the training data or generalize to a new one. Vanishing gradient issue: It is situation which commonly occurs in any neural network. It is mainly described where a deep multilayer feed-forward network or a recurrent neural network is unable to propagate useful gradient information from the output end of the model back to the layers near the input end of the model. Number of trainable parameters: The CNN has two main kind of parameters that is weight and bias. Kernel size: The kernel size refers to the dimension of the filter, which must be a small odd number. Inference time: It is the time taken by the optimized GPU code to execute. L1 error: The L1 mean squared error is the most popular metric used for regression problems. It aims at finding the average squared error between predicted values and actual values. R2 score: This is another statistical measure of how close the data is to the regression line. The metric is used for determination for multiple regression. It gives the percentage of response variable variation that is explained by linear model. The higher the R-squared, the better the model fits the data.

The following section describes the in-detail model, which aims at increasing the accuracy whilst keeping the parameters intact. In the model proposed the housing dataset is used which has 8 features as the input and the goal is to find the median value of the housing and build a non regression model.

1.The initial steps involve importing all the necessary libraries that will help in building the 1DCNN. The dataset used for this experiment is the housing datasets of the California housing dataset which consists of 8 columns, with each column having its own importance.

2.The first 10 columns of the dataset are read and displayed along with sub plotting of each feature on the graph is plotted.

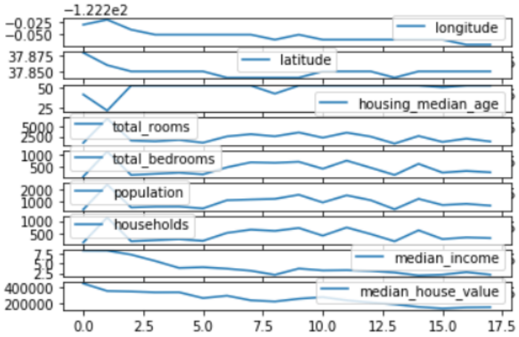
Fig. 2. Top 10 records



```
Here are the first 10 rows of the dataset:
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximi
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR B.
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR B.
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR B.
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR B.
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR B.
5	-122.25	37.85	52.0	919.0	213.0	413.0	193.0	4.0368	269700.0	NEAR B.
6	-122.25	37.84	52.0	2535.0	489.0	1094.0	514.0	3.6591	299200.0	NEAR B.
7	-122.25	37.84	52.0	3104.0	687.0	1157.0	647.0	3.1200	241400.0	NEAR B.
8	-122.26	37.84	42.0	2555.0	665.0	1206.0	595.0	2.0804	226700.0	NEAR B.
9	-122.25	37.84	52.0	3549.0	707.0	1551.0	714.0	3.6912	261100.0	NEAR B.

Fig. 3. Subplot of Columns



3.The next step involves the splitting of dataset into train and test, with a ratio if 70 percent training data and 30 percent testing data.

4.All the data is converted to an array using the numpy function in python. The following imports like pytorch and maxpooling layer along with the CNN layer is done for the further processing on the data. Along with these the function to flatten the data and the activation function relu is also imported. The main non linear regressor is sequential in nature.

5.In the model we have used 2 convultion layer for the d=function with the input data. It is followed by max pooling layer of size 1.

6.The kernel used on the input is also of size 1.The iteration of the operation is divided into batches of size 64.

7.The average loss and average score of the train dataset is calculated in the following code. epochs of 100 is used to train the data completely and generate the best result.

8.The final accuracy of the dataset is displayed in form of L1 score and R squared loss.

Experimented Results

The proposed non linear regression model is presented by calculating the L1 score and R-squared error. The aim of the model is to increase the R- squared error, to best fit the model along the given parameters. There are various parameters tweaking done in the code in order to obtain a suitable result which addresses all the factors of any non linear model.

There are various different combinations used with the parameters to display the result:

1.One convolution layer with one kernel and one max pooling layer and 64 as batch input over 100 epochs

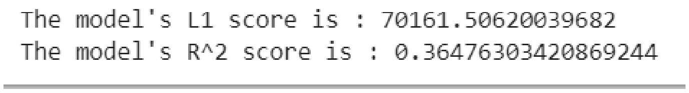


Fig. 4. Result 1

```
The model's L1 score is : 70161.50620039682
The model's R^2 score is : 0.36476303420869244
```

2.One convolution layer with one kernel and one max pooling layer and 100 as batch input over 100 epochs.

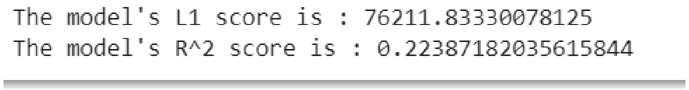


Fig. 5. Result 2

```
The model's L1 score is : 76211.83330078125
The model's R^2 score is : 0.22387182035615844
```

3.2 convolution layer with one kernel and one max pooling layer and 100 as batch input over 100 epochs.

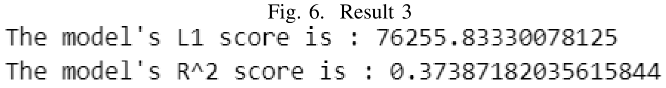


Fig. 6. Result 3

```
The model's L1 score is : 76255.83330078125
The model's R^2 score is : 0.37387182035615844
```

Conclusion

1D CNN surpasses all the traditional and conventional approaches. The model proposed in this paper is basically a 1D CNN and solution is non linear. 1DCNN is majorly used in applications where labeled data for training is low and real time application is desired. 1DCNN are easier to train and use less computation power while offering minimum computational complexity. The accuracy of any CNN model is measured by calculating the L1 error and R-squared error , where the R-squared score has to be high so that the model best fits.There are also many applications of CNN model in the real world like Image recognition, video analysis , drug discovery and time series forecasting.

References

[1] Kiranyaz, Serkan, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. ”1D convolutional neural networks and applications: A survey.” arXiv preprint arXiv:1905.03554 (2019).

[2] Redmon, Joseph, and Anelia Angelova. ”Real-time grasp detection using convolutional neural networks.” In 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1316-1322. IEEE, 2015.

[3] Sumit Saha, “A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.