Web Dashboard to Create Machine Learning Model for different types of Optimizer

-by Akash Saurabh

Guide and Mentor: Dr. Ramana Pidaparti, Dr. Ramviyas Parasuraman, Sanjay Sarma

The University Of Georgia , Athens, GA, USA

Abstract

We have seen in today's world business intelligence is one of the key areas of research and development when it comes to using large sets of legal and trusted datasets. Due to the complexity and volume of corporate and research data, we are bound to develop various tools to have a human intractable meaning giving visual representations for a common person to make some meaning out of it. These 'meanings' vary depending on the use case for corporate or research.

Following this path I wanted to develop an auto-model generating web hostable artificial neural network(ANN) driven tool that can help users to enter datasets and get predictive outputs that are visually observable with the help of graph charts and results. Our tool has ANN layers which will be explained in detail in further sections. The second interesting addition in the future to this project can be the development of a feedback algorithm (IANN) which can help prediction by adding a derivative approach. This part however only left for future consideration and not included in this project due to the time complexity involved.

In this project, we are making an open-source program framework on a python-based web simulator and providing a sample online dashboard to create the artificial neural network model and calculate the accuracy of artificial neural networks with different types of optimizers.

In this project, users can upload the data to the dashboard and create a model, test the model for accuracy, and save it in different formats. I provided an installation manual to the user so that they can use it for further development purposes.

A neural network inverse mapping approach was used in a structural integrity redesign problem to achieve the desired input of the uploaded dataset. Also, through the inverse-mapping procedure, the input of the uploaded dataset was obtained for two trained neural-network models. The results obtained through the inverse-mapping procedure are compared with actual panel configurations and environments and experimental data and if they provide good accuracy our model is working fine.

Key Terms — artificial neural networks, Particle Swarm Optimize.

Introduction

As discussed we will now see how an abstract concept of data visualization is broken down into discrete logical codes. The business intelligence details are just outputs of well-designed code blocks which we can vary in complexity. One such approach is ANN which stands for the artificial neural network. We divide our project implementation into three broad parts namely input features, ANN defined logic and output prediction. The ANN further divides into the input layer, hidden layer, and output layer.

When analyzing complex physical systems, a common problem is that the system parameters of interest cannot be measured directly. For many of these systems, scientists have developed sophisticated theories on how measurable quantities y arise from the hidden parameters x. We will call such mappings the forward process. However, the inverse process is required to infer the hidden states of a system from measurements. Unfortunately, the inverse is often both intractable and ill-posed since crucial information is lost in the forward process.

This project makes it possible to quantify uncertainty, reveal multi-modal distributions, and identify degenerate and unrecoverable parameters – all highly relevant for applications in many fields of science and processing of big datasets in corporations.

To fulfill the objective of achieving the input parameters of uploaded data corresponding to the desired reliability, we develop an ANN model that generates the predicted data. In general, we use the Jupyter notebook for working on our latest Machine Learning model. It could be a sci-kit-learn Linear Regression model, a Keras Convolutional Neural Network. In this paper, we integrate it with a python dash web front end, where users can upload a file to test and train the model. We created a dashboard that shows model accuracy and other related graphs for analysis purposes. The dashboard has three functions: first is to visualize the dataset, the second is to create the model and the last step is to check the output curves and accuracy.

The workflow online server looks like this:

1. Train and tune the model using the flexibility and convenience of python. Save the final predicted data.
2. In our case, we are using Heroku to implement the flask on the server-side and setup the HTTP endpoints for the functions.
3. We design the frontend using python dash which is a python dashboard library. Design the UI and define the API calls that will communicate with your back end.

The other purpose of this project is to use this for development purposes too. Like if any wants to check their optimizer they can use our code to implement dashboard and its functionality, like:

1. A developer can add his optimizer code to the project with the easy installation step which we provide in the GitHub account.
2. Developers can use the given project dashboard for multiple purpose, like, using this in their own project to add optimization method in Neural network training.
3. The user can create and do data visualization.
4. Sample website can be reached by clicking on the below link:

(Click on this link: https://easydashnew.herokuapp.com).

Related Work

In this[1] paper they tried to predict the relevance of the input parameters in the input process by using the knowledge stored in the NNs. They rank order the input parameters according to the irrelative importance in the input mechanism by using the sensitivity of the outputs caused by the changes in inputs through NN inverse mapping. The inverse-mapping procedure was investigated for two NNs.

The procedure of inverse mapping using NNs has been used to solve engineering problems in various fields. For example, in remote sensing, the geophysical parameters are obtained from measurements made by sensors, [2] using iterative inversion of neural networks. In robotics, the inverse kinematics problem, to compute the positions of joints that yield a given location for the arm, is solved using inverse mapping with neural networks [3][4].

Inverse mapping of Artificial Neural Network[5]. Artificial Neural Networks are statistical models for establishing non-linear relations between input and output data. Hence, for a given set of observations with Y inputs and X outputs, an ANN can be trained to map a relationship between Y and X. Inverse mapping of neural networks(IANN) is a procedure to obtain the right input parameter values for a pre-trained ANN for a set of known outputs. An optimizer is applied for correcting the input of the network iteratively till the output converges with the targets with minimal errors.

This is widely used in many engineering applications, especially in manipulator inverse kinematics[6–8], sensor measurements[9], and structural integrity analysis [1]. In our current work, we used a different type of optimization algorithm in the feedback loop, which corrects and estimates the setting values given as inputs to the ANNs for the desired output (true observations).

The work mentioned in [1], is implemented and tested in MATLAB, and MATLAB is not open source and they cannot create or provide a library for the implemented code. We proposed a user-friendly open-source solution to the user for a better user experience and to increase development opportunities. We provide a graphical user interface. An online dashboard for an Informatica model. Our proposed solution help peoples of engineering and STEM background to create, train, test machine learning model and apply forward mapping on that machine learning model. It also provides different types of optimization algorithm to choose from while making the model. Since the code of this project is also available for continuous development purposes. It also provides a development opportunity for those who want to test their optimization algorithm.

Proposed idea/method/solution/framework

The tool that we propose here will be capable of uploading the datasets and produce a predictive visual dashboard that has graph separated areas for better visualization. Each of these parts of the dashboard will be dedicated to one certain use case for visualization. The aim is to make this tool available online as a web-hosted tool application. I deployed a sample of this dashboard in the Heroku free environment. Since the Heroku environment supports 500 Mb of free space. Since TensorFlow and PyTorch need more space to train the model it is not feasible to deploy the complete dashboard in the free environment. So I deployed a sample dashboard with supporting results.
(Click on this link:
https://easydashnew.herokuapp.com)

Architecture.

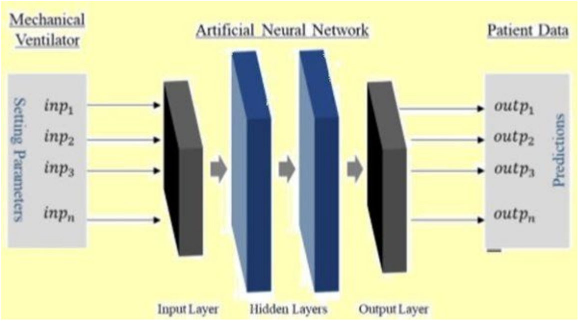


Figure 1: Dashboard Architecture

Proposed idea. In this project, we proposed a user-friendly graphical user interface to create and test models and provide a module in python which is easily accessible by GitHub. In this project, we are using neural networks to implement ANN input parameters. The artificial neural network is used to create the model. In many fields, we are required to use ANN like in the case of house rate prediction in which we can give input of all the parameters related to a house and then predict the rate of the house. So in this scenario, we can calculate the house rate by applying ANN.

Method/Solution/Framework. In this project we are creating artificial neural networks; the artificial neural network is to get output parameters of the uploaded dataset.

For developing the neural network we use python TensorFlow, PyTorch, sci-kit learn, pandas, NumPy libraries in the web dashboard this part takes user input like to upload a dataset, number of hidden layers, type of activation function, and the type of optimizer. This one is created by python TensorFlow, PyTorch, NumPy, pandas, sci-kit learn. The web dashboard is using the python-dash framework. It is hosted on the Heroku platform. The whole installation manual is available on GitHub for future improvement.

Experimental verification/validation

We get our inspiration from the business intelligence tools like Tableau, Splunk, and Team Foundation. These tools input large data sets and process them. Custom query languages are part of such tools. What we got inspired from is the flexibility of Splunk UI, the robust structure of Tableau, and the sharing capabilities of the Team Foundation.

The end tool will be a strong enough tool capable of inputting models, data sets, and produced output in form of different media types like charts and metrics. The final implementation however will be limited specifically by the use cases particular to this project. The whole system is a web browser-based so we don't have the hassle of installing any application to do any kind of visualization.

The whole setup is python based. To host the dashboard I install the necessary libraries like python, pip, TensorFlow, PyTorch NumPy pandas, sci-kit learn, python dash, git, etc. The web dashboard is designed by python dash and on the server, we are using flask server. We are using "ventilator data" as the baseline to calculate the accuracy and desired charts.

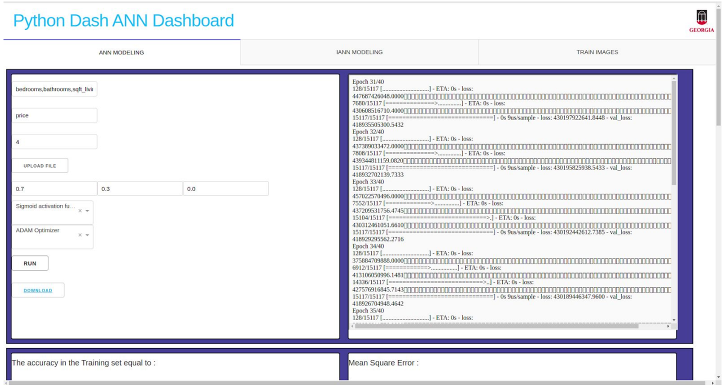
Results

In these days of COVID-19, where ventilators and their high degree of parameter control are at high priority for patients. This requires a high level of expertise even for clinicians. I take the data from [1] which is the ventilation parameter control and calculate the accuracy and mean square error using Tensorflow and PyTorch artificial neural networks. The accuracy generated by Tensorflow for the [1] dataset is 60% and the accuracy generated by the Pytorch model is 73%.

The final tool is hosted on web and is accessible from below URL:

https://annmodel.herokuapp.com/

Some sample views are as below:



bedrooms,bathrooms,sqft_livi

price

4

0.7 | 0.3 | 0.0

Sigmoid activation fu...

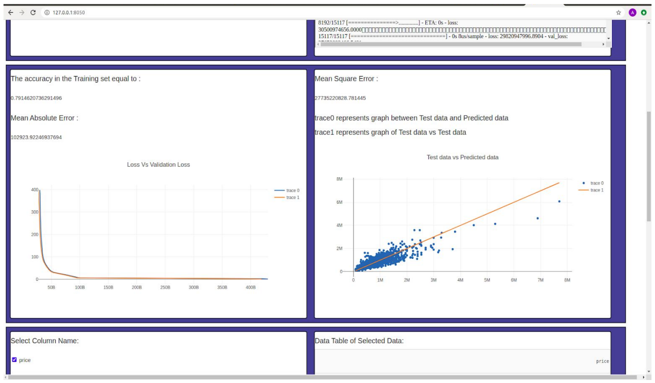
ADAM Optimizer

```
Epoch 31/40
128/15117 [..............................] - ETA: 0s - loss:
447687426048.0000□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
7680/15117 [==============>...............] - ETA: 0s - loss:
430608516710.4000□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
15117/15117 [==============================] - 0s 9us/sample - loss: 430197922641.8448 - val_loss:
418935505300.5432
Epoch 32/40
128/15117 [..............................] - ETA: 0s - loss:
437389033472.0000□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
7808/15117 [==============>...............] - ETA: 0s - loss:
439344811159.0820□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
15117/15117 [==============================] - 0s 9us/sample - loss: 430195825938.5433 - val_loss:
418932702139.7333
Epoch 33/40
128/15117 [..............................] - ETA: 0s - loss:
457022570496.0000□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
7552/15117 [=============>................] - ETA: 0s - loss:
437209531756.4745□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
15104/15117 [============================>.] - ETA: 0s - loss:
430312461051.6610□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
15117/15117 [==============================] - 0s 9us/sample - loss: 430192442612.7385 - val_loss:
418929295562.2716
Epoch 34/40
128/15117 [..............................] - ETA: 0s - loss:
375884709888.0000□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
6912/15117 [============>.................] - ETA: 0s - loss:
413106050996.1481□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
14336/15117 [===========================>..] - ETA: 0s - loss:
427576916845.7143□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
15117/15117 [==============================] - 0s 9us/sample - loss: 430189446347.9600 - val_loss:
418926704948.4642
Epoch 35/40
128/15117 [..............................] - ETA: 0s - loss:
```

The accuracy in the Training set equal to :

Mean Square Error :

Figure 2: Dashboard UI view
(User Input)



```
8192/15117 [===============>.............] - ETA: 0s - loss:
30500974656.0000
15117/15117 [==============================] - 0s 8us/sample - loss: 29820947996.8904 - val_loss:
```

The accuracy in the Training set equal to :

0.7914620736291496

Mean Absolute Error :

102923.92246937694

Mean Square Error :

27735220828.781445

trace0 represents graph between Test data and Predicted data

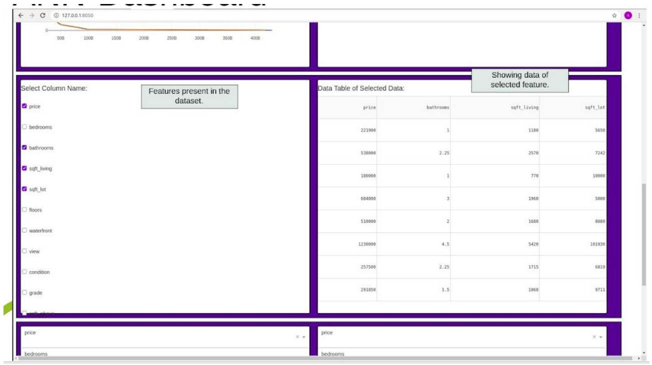
trace1 represents graph of Test data vs Test data

Select Column Name:

☑ price

Data Table of Selected Data:

Figure 3: Dashboard UI view
(Graphical Result)



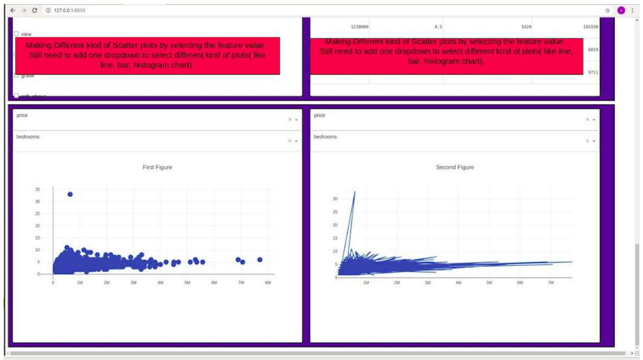
Select Column Name:

☑ price
☐ bedrooms
☑ bathrooms
☑ sqft_living
☑ sqft_lot
☐ floors
☐ waterfront
☐ view
☐ condition
☐ grade

Data Table of Selected Data:

price	bathrooms	sqft_living	sqft_lot
221900	1	1180	5650
538000	2.25	2570	7242
180000	1	770	10000
604000	3	1960	5000
510000	2	1680	8080
1230000	4.5	5420	101930
257500	2.25	1715	6819
291850	1.5	1060	9711

Figure 4: Dashboard UI view
(Data Processing)



Making Different kind of Scatter plots by selecting the feature value. Still need to add one dropdown to select different kind of plots(like line, bar, histogram chart).

Making Different kind of Scatter plots by selecting the feature value. Still need to add one dropdown to select different kind of plots(like line, bar, histogram chart).

Figure 5: Dashboard UI view
(Data Visualization)

I develop the ANN model using python, TensorFlow, and PyTorch. Python dash is deployed in Heroku cloud server. The connection is stabilized for taking input from the user.

Conclusions

We build a tool that is robust with the input of various data sets. The user can input the model and respective data set to start the processing. The data set is inputted in CSV, excel file formats. Then the tool creates an ANN model as per user preference of optimizer. What is crucial here to note is that we allow one type of dataset per model inputted at a time. It is even advised that data is presorted and arranged with as much uniformity as possible preprocessing to yield the best and optimal results. The tool is developed in the python framework which gives us the flexibility to use and exploit many python libraries that are available in the python community.

Future Work.

The future work on this project could be the development of a feedback algorithm (IANN) which can help prediction by adding a derivative approach that is common in gradient calculus. We know that the main objective of data analysis and synthesis is to develop an insight into the predictability of future patterns or models yield by datasets that we are considering.

Acknowledgments

I would like to express my special thanks of gratitude to my Master's Report Guide Dr.

Ramana Pidaparti and Dr. Ramviyas Parasuraman and special gratitude to Sanjay Sarma who gave me the golden opportunity to do this wonderful project on the topic of web dashboard to create machine learning model. This project also helped me in doing a lot of research and I came to know about so many new things. I am thankful to them. I am specifically showing gratitude towards Sanjay and Dr. Pidaparti. They helped me to understand the project and its requirement and how IANN, PSO, and GPSO algorithm works and most importantly for the patient they show. This project can be used as a starting point to create the software as a service(SaaS) for ML.

Reference

1. Pidaparti, R.; Jayanti, S.; Palakal, M.; Mukhopadhyay, S. Structural integrity redesign through303 neural-network inverse mapping. AIAA journal 2003, 41, 119–124
2. Davis, D.T., Chen,Z., Tsang, L., Hwang, J.N., and Chang, A.T.C., "Retrieval of Snow Parameters by Iterative Inversion of a Neural Network," IEEE Transactions on Geoscience and Remote Sensing ,Vol.31, No.4, 1993, pp.842–852.
3. DeMers, D., and Kreutz-Delgado, K., "Canonical Parameterization of Excess Motor Degrees of Freedom with Self Organizing Maps, "IEEE Transactionson Neural Networks, Vol.7, 1996, pp.43–55.
4. DeMers, D., and Kreutz-Delgado, K., "Inverse Kinematics of Dexterous Manipulators, "Neural Systems for Robotics, edited by O. Omvidarand P.vander Smagt, Academic Press, NewYork, 1997, pp.75–116.
5. Sanjay Sarma O V, , Amie Koenig, and Ramana M Pidaparti, "Mechanical Ventilator Parameters Estimation through Informatics Models"
6. DeMers, D.; Kreutz-Delgado, K. Canonical parameterization of excess motor degrees of freedom with self-organizing maps. IEEE transactions on neural networks 1996, 7, 43–55
7. Tejomurtula, S.; Kak, S. Inverse kinematics in robotics using neural networks. InformationSciences1999, 116, 147–164.
8. 14. DeMers,D.; Kreutz-Delgado, K. Inverse kinematics of dexterous manipulators. In Neural Systems for Robotics;299 Elsevier, 1997; pp. 75–116
9. Davis, D.T.; Chen, Z.; Tsang, L.; Hwang, J.N.; Chang, A.T. Retrieval of snow parameters by iterative301 inversion of a neural network. IEEE Transactions on Geoscience and Remote Sensing 1993, 31, 842–852