International Journal of Combinatorial Optimization Problems and Informatics, 14(3), Sep-Dec 2023, 42-52. ISSN: 2007-1558.
doi.org/10.61467/2007.1558.2023.v14i3.406

www.editada.org

Development of a Methodology for Dynamic Motor Control with Machine Learning

David Aguilar Robles, Ma. del Carmen Santiago Díaz, Ana Claudia Zenteno Vázquez, Yeiny Romero Hernández, Judith Pérez Marcial, Gustavo T. Rubín Linares

Benemérita Universidad Autónoma de Puebla
{marycarmen.santiago, ana.zenteno, yeiny.romero, judith.perez, gustavo.rubin
}@correo.buap.mx

Abstract. In robotics, the challenge of achieving precision in the positional control of various motor systems necessitates the real-time evaluation of multiple variables. This is essential to prevent instability in the robot caused by processing time. This work proposes a methodology grounded in the automatic learning of motor systems. The goal is to generate control actions preemptively, ahead of their measurement and processing. To this end, a second-order system is proposed, along with its response to disturbances such as step signals.
Keywords: Machine learning, dynamic control, intelligent assistants.

Article Info
Received 11 Sep, 2023
Accepted 11 Dec, 2023

1 Introduction

Artificial intelligence (AI) refers to systems or machines that imitate human intelligence to perform tasks and can iteratively improve based on the information they collect. AI manifests itself in various forms. Some examples are:

- Chatbots use AI to understand customer problems faster and provide more efficient responses.
- Intelligent assistants use AI to analyze critical information from large free-text data sets to improve scheduling.
- Recommendation engines can provide automated suggestions for TV shows based on users' viewing habits.

AI has become an umbrella term for applications that perform complex tasks that previously required human input, such as communicating with customers online or playing chess. The term is often used interchangeably with its subfields, which include machine learning and deep learning. However, there are certain differences. For example, machine learning focuses on creating systems that learn or improve their performance based on the data they consume. It is important to keep in mind that, although all machine learning is AI, not all AI is machine learning (Schmid, Gerharz, Groll, & Pauly, 2022).

1.1 Machine learning

Machine learning is a branch of artificial intelligence that allows machines to learn without being programmed for this specific purpose. An essential skill to create systems that are not only intelligent, but autonomous and capable of identifying patterns in data to turn them into predictions (Brownlee, 2022).

Types of machine learning

- Classical machine learning is often classified by how an algorithm learns to be more accurate in its predictions. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. The type of algorithm data scientists choose to use depends on the type of data they want to predict (Morante, 2018).

- Supervised learning: In this type of machine learning, data scientists provide algorithms with labeled training data and define the variables they want the algorithm to evaluate for correlations. Both the input and output of the algorithm are specified.
- Unsupervised learning: This type of machine learning involves algorithms that are trained on unlabeled data. The algorithm scans through data sets looking for any meaningful connections. The data with which the algorithms are trained, as well as the predictions or recommendations they generate, are predetermined.
- Semi-supervised learning: This machine learning approach involves a combination of the above two types. Data scientists can feed an algorithm mostly labeled training data, but the model is free to explore the data itself and develop its own understanding of the data set.
- Reinforcement learning: Data scientists often use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative signals as it figures out how to complete a task. But for the most part, the algorithm decides for itself what steps to take along the way.

Multivariate data occur in a variety of disciplines, for example, in biomedical research, social sciences, or econometrics. The data is said to be multivariate if the response consists not only of one variable, but of $d \geq 2$ output variables, say $Y \in R$. So, we are often interested in finding a functional relationship between the output Y and some variable characteristics $X \in R$, that is, we want to perform a multivariate regression analysis. Unlike univariate multiple regression ($d=1$), which also includes multiple characteristics of X, multivariable regression attempts to specify the relationship of several outcome variables with X simultaneously. The goal of such multivariate analyzes is that consideration of possible dependencies between results can lead to procedures with better power (in case of inference) or precision (in case of prediction) compared to separate univariate analyzes (Schmid, Gerharz, Groll, & Pauly, 2022).

2 Methodology

Regression is a predictive modeling task that involves predicting a numerical output given some input. It is different from classification tasks that involve predicting a class label. Typically, a regression task involves predicting a single numerical value. Although, some tasks require predicting more than one numerical value. These tasks are known as multiple output regression or multiple output regression for short.

In multiple output regression, two or more outputs are required for each input sample and the outputs are required simultaneously. The assumption is that the outputs are a function of the inputs (Brownlee, 2022).

For many Machine Learning algorithms used in Data Science to work better, the input variables to the algorithm must be normalized. Normalizing means, in this case, compressing or extending the values of the variable so that they are in a defined range. However, a poor application of normalization, or a careless choice of normalization method can ruin your data, and with it your analysis (Morante, 2018).

There are normalization techniques that compress the input data between empirical limits (the maximum and minimum of the variable). This means that, if there is noise, it will be amplified (Morante, 2018).

To see the effect of normalizing the data and having noise in the signal that will be measured in the future directly from the robot's sensors, it was decided to do four case studies:

1. Implement the algorithms with the unnormalized and noise-free data set.
2. Implement the algorithms with the unnormalized and noisy data set.
3. Implement the algorithms with the normalized and noise-free data set.
4. Implement the algorithms with the unnormalized and noisy data set.

3 Results

Next, we see the results, first we will take a look at the algorithms with the plant receiving unit steps and then receiving sine inputs. Plant with sinus inputs

Table 1. Database parameters.

Simulation time	Sampling	Number of step entries
10 s	100	50

Table 2. Plant with sinus entrances

	with noise	noiseless
Normalized	Linear regression method MSE: 214263771407263412925759488.0000 k-Nearest Neighbors for multi-output regression MSE: 27.1421 Decision Tree for multi-output regression MSE: 89.9494 Multi-output neural network MSE: 2935.1144	Linear regression method MSE: 5.1602 k-Nearest Neighbors for multi-output regression MSE: 27.1174 Decision Tree for multi-output regression MSE: 141.6744 Multi-output neural network MSE: 2604.0644
Not normalized	Linear regression method MSE: 473034037975336687658598 4.0000 k-Nearest Neighbors for multi-output regression MSE: 133.0015 Decision Tree for multi-output regression MSE: 174.3104 Multi-output neural network MSE: 8.8519	Linear regression method MSE: 125.3017 k-Nearest Neighbors for multi-output regression MSE: 132.9581 Decision Tree for multi-output regression MSE: 174.3605 Multi-output neural network MSE: 5.4136

Here the prediction is better when the data has no noise, and is unnormalized. The neural network also improves its behavior when the data is not normalized. From the table we can see that the mean square error (MSE) decreases when the data does not contain noise. Therefore, it is recommended that the data obtained from the sensors undergo signal treatment to reduce signal noise. On the other hand, it cannot yet be concluded that normalizing the data affects the prediction of the models, since this can also be affected by the amount of information with which the models are trained. Therefore, it remains to generate a more extensive database to see if the results improve.

3.1 Comparison of the results of the three methodologies

Table 3. Comparison of the neural network with step input in the three implementations.

	Noisy and normalized	Noisy and unnormalized	Noiseless and normalized	No noise and no normalization
Multi-output neural network (1)	20215,8381	0,4571	18726,291	0,1747
Multi-output neural network (2)	2494159,312	0,5066	163.150	0,0981
Multi-output neural network (3)	284.105	0,0776	276104,4268	38912,2673

Table 4. Comparison of the neural network with step input in the three implementations.

	Noisy and normalized	Noisy and unnormalized	Noiseless and normalized	No noise and no normalization
Multi-output neural network (1)	2935,1144	8,8519	2604,0644	5,4136
Multi-output neural network (2)	285180,3809	1478,1502	163.150	1105,6508
Multi-output neural network (3)	7.225	17,5880	13993,253	13,2334

Now the MO models are trained with three different inputs in the same dataset, the inputs will be unit step, sine and impulse. The parameters of the neural network are also varied to see which parameters improve the prediction.

First implementation

Simulation time	Sampling	Number of step entries
10 s	100	500

For this first implementation, a dataset was generated with the entries organized randomly.

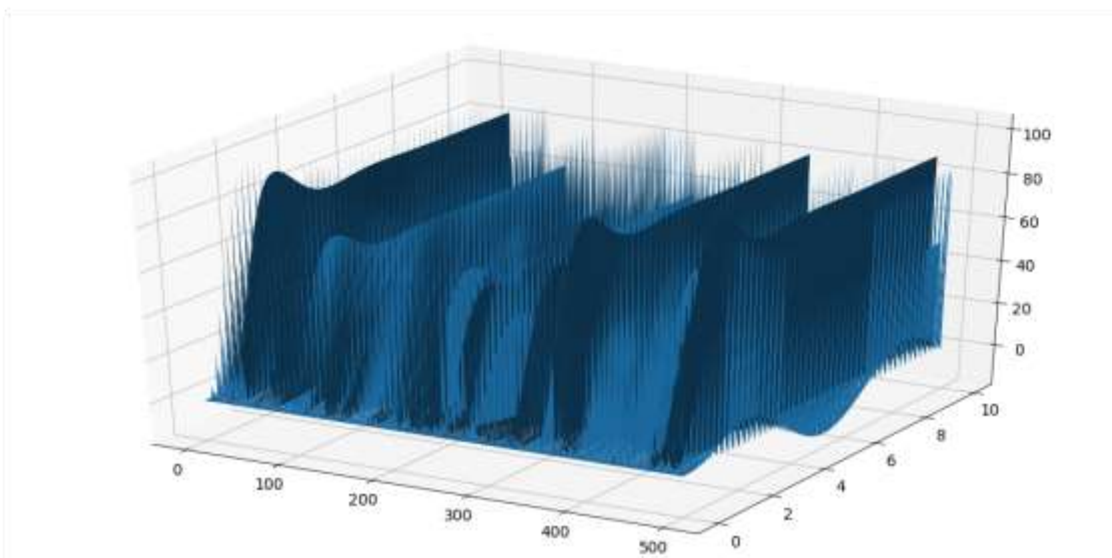


Fig. 1. Data set visualization for First implementation.

To create the neural network model that best predicted the results, trial and error was used. Below is the model that best fitted the data.

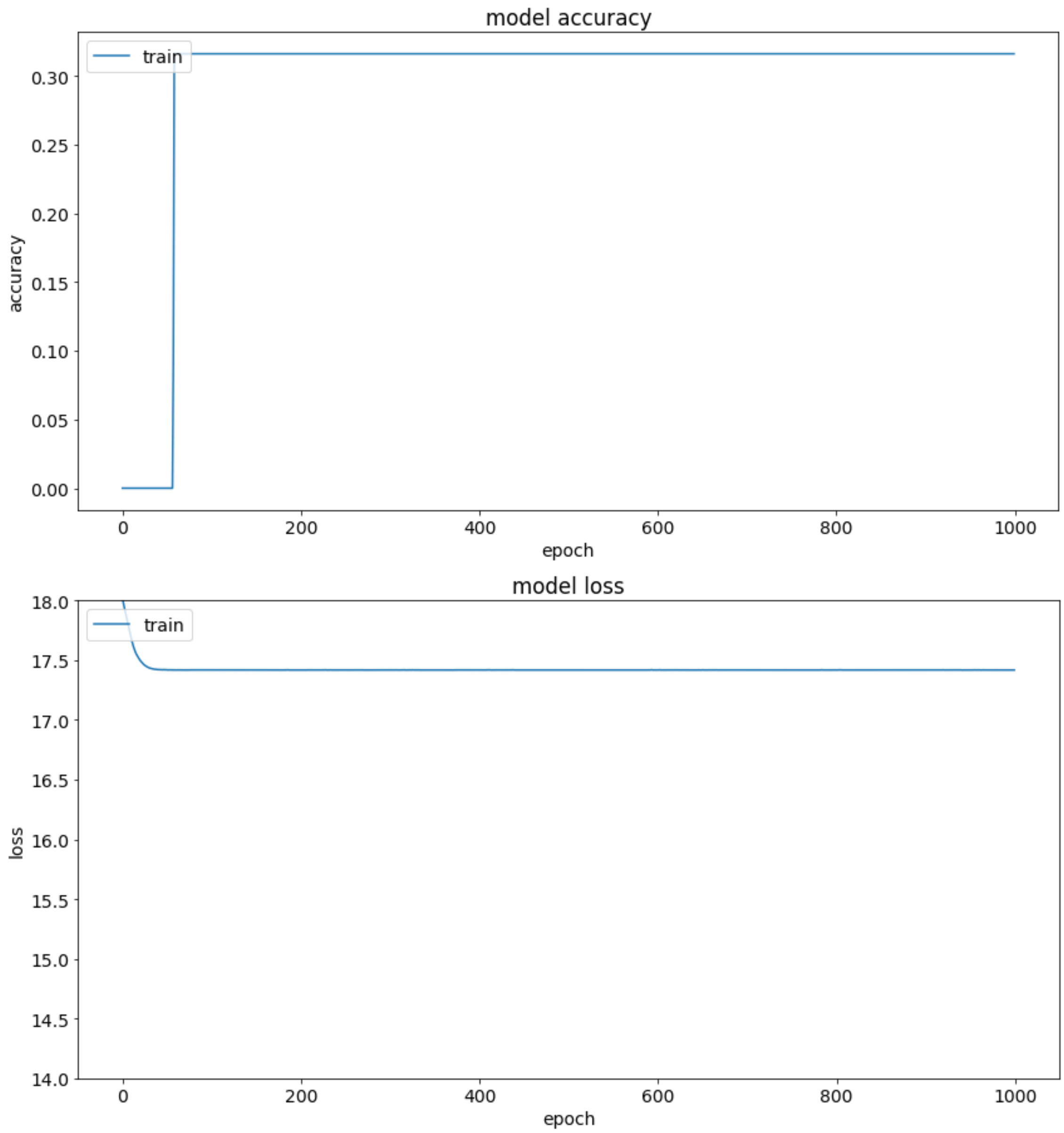


Fig. 2. Model that best fitted the data.

Although we can see that the precision is low and the deviation between the predictions made by the neural network and the real values of the observations used during learning is high.

Second implementation

Simulation time	Sampling	Number of step entries
10 s	250	600

On this occasion the dataset was ordered as follows, a third of the data set are step inputs with random magnitude, another third of the set are sine inputs with random magnitude and the rest are impulse inputs with random magnitude.

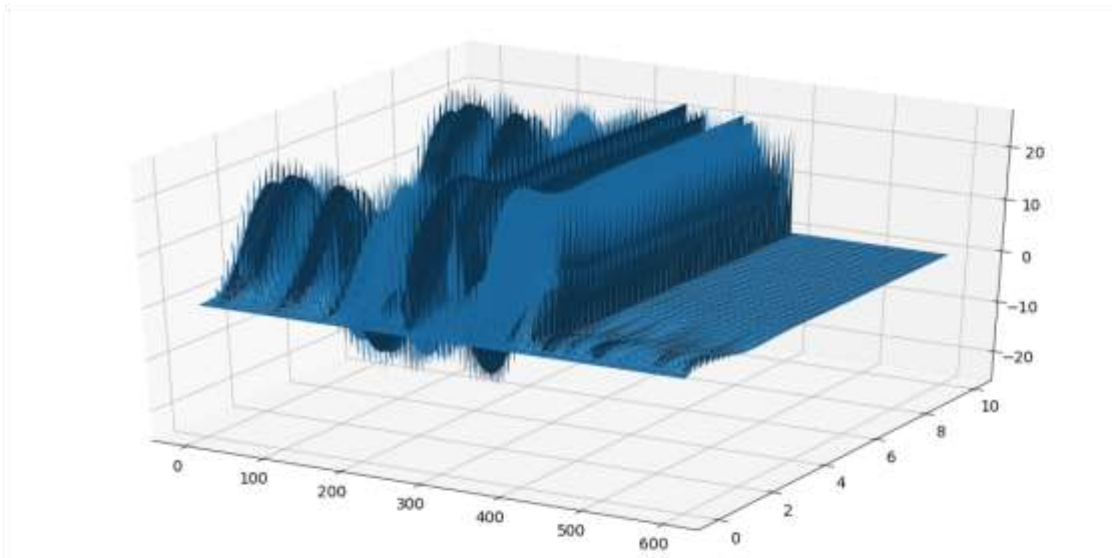
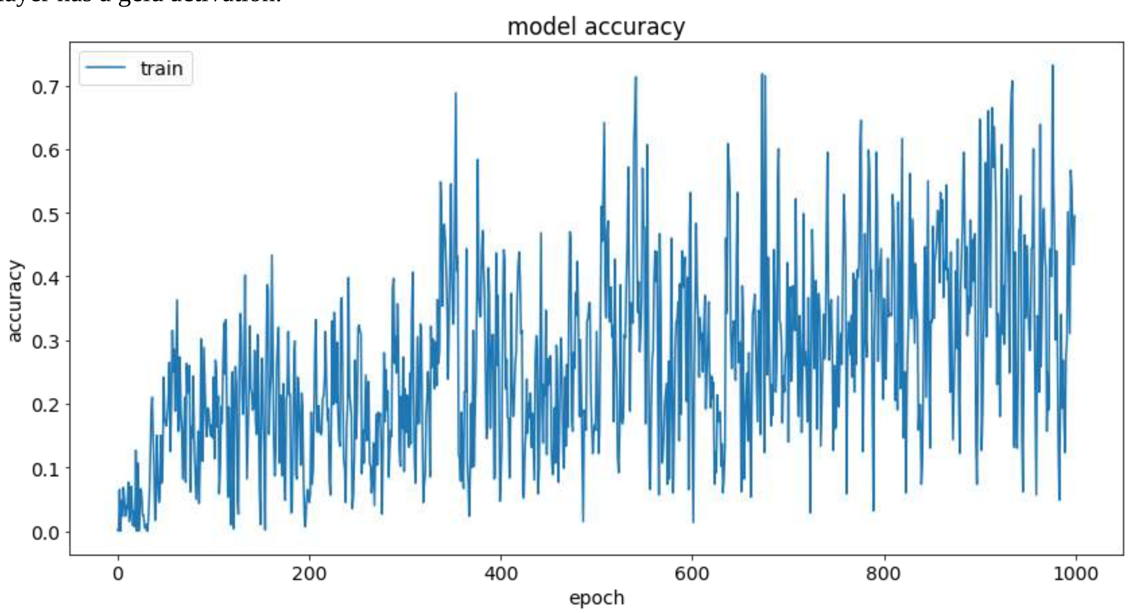


Fig. 3. Data set visualization for second implementation.

The same process was repeated to create the neural network model, compared to the previous network the output layer has a gelu activation.



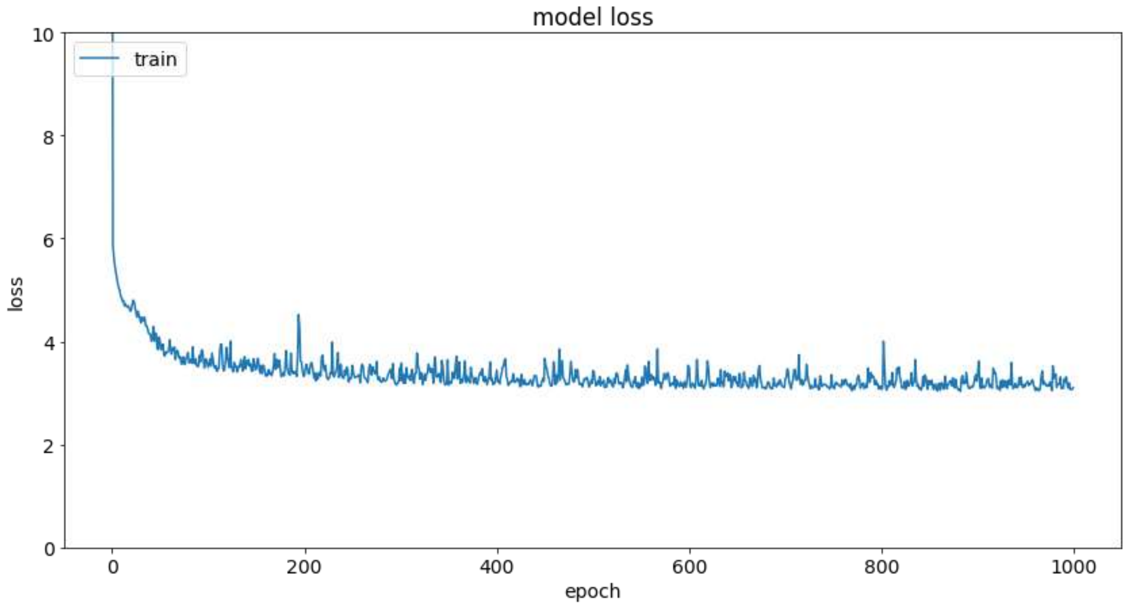


Fig. 4. Model that best fitted the data.

With these adjustments we can see that both the prediction and the deviation between the predicted and real results improved, it was reduced from 17.5 to 3.8 approximately.

Third implementation

Simulation time	Sampling	Number of step entries
10 s	250	600

For this dataset, the neural network was also divided into three equal parts, a third for each input, only this time its magnitude is not random, but grows linearly.

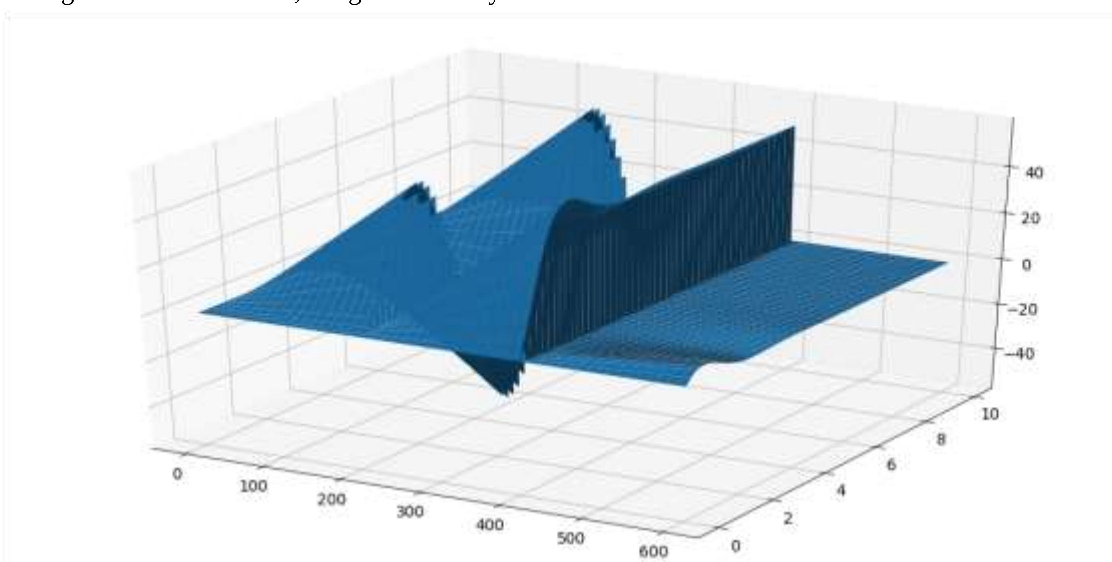


Fig. 5. Data set visualization for third implementation.

For this neural network, one layer was increased, but the number of neurons was reduced from 1750 to 150.

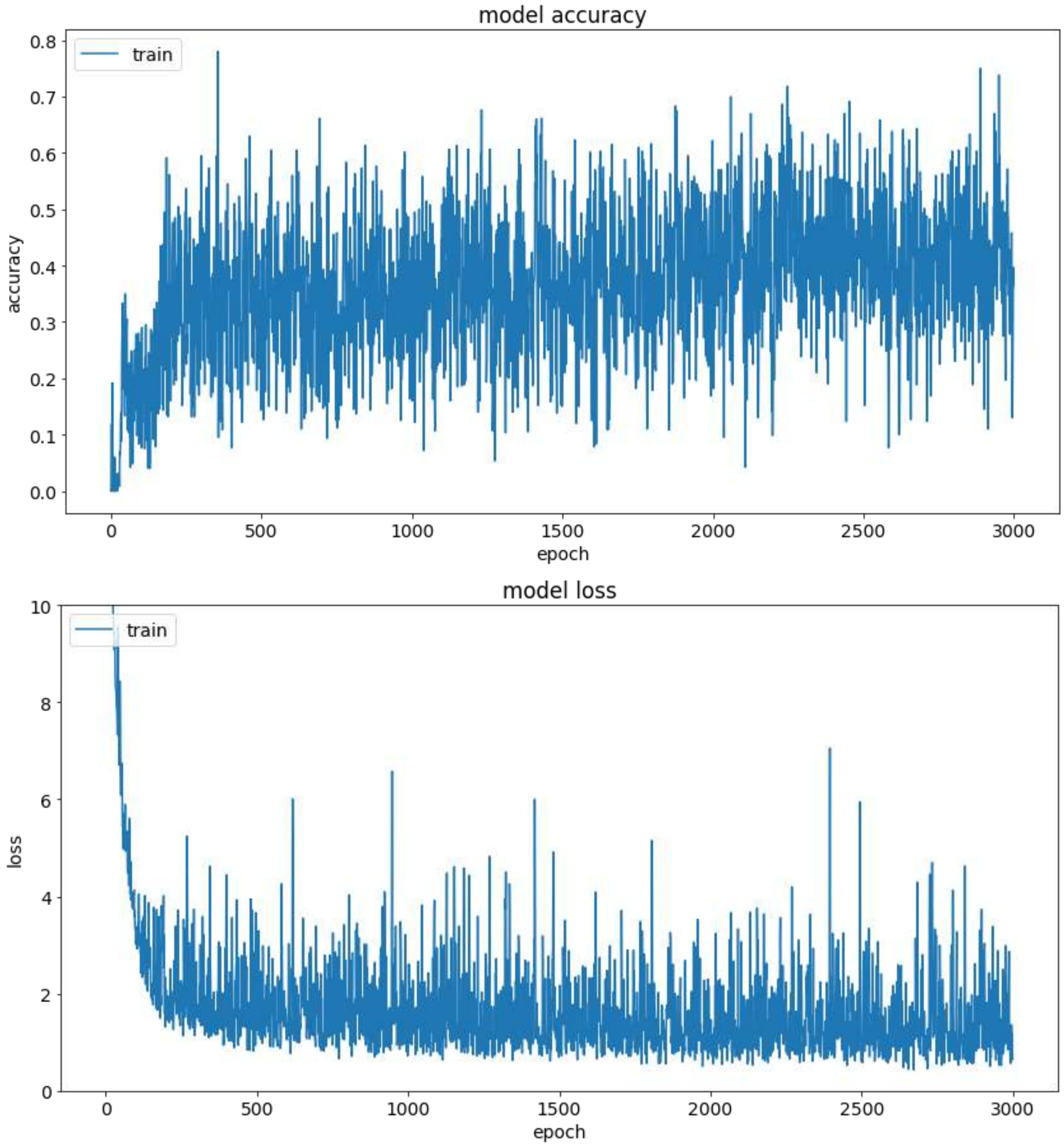


Fig. 6. Model that best fitted the data.

As seen in the graphs, the results improved, but the prediction is less stable.

Fourth implementation

Simulation time	**Sampling**	**Number of step entries**
10 s	250	300

For this implementation, the dataset was organized in the same way as the previous implementation, but the number of entries was reduced to 300.

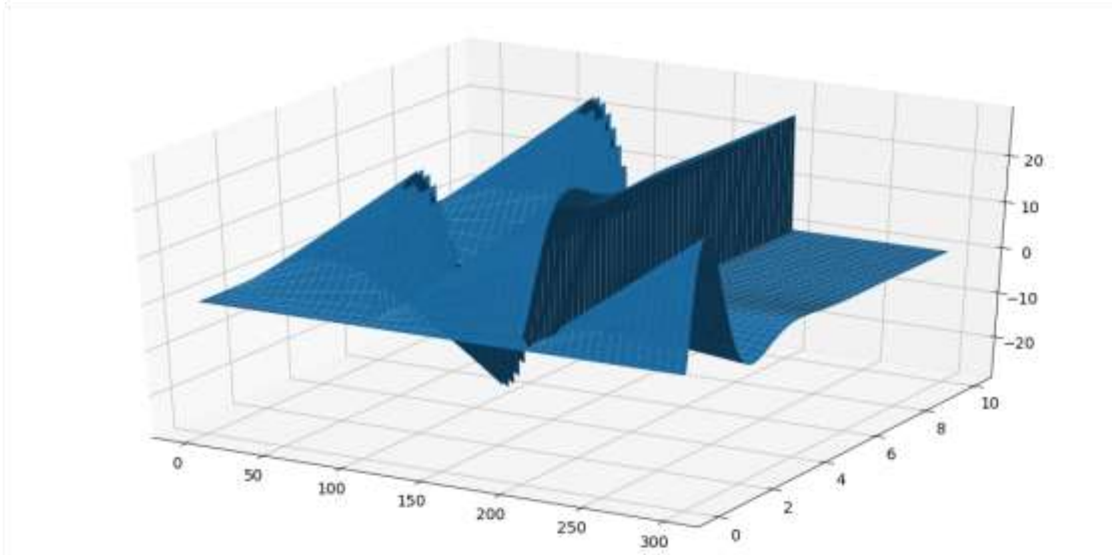
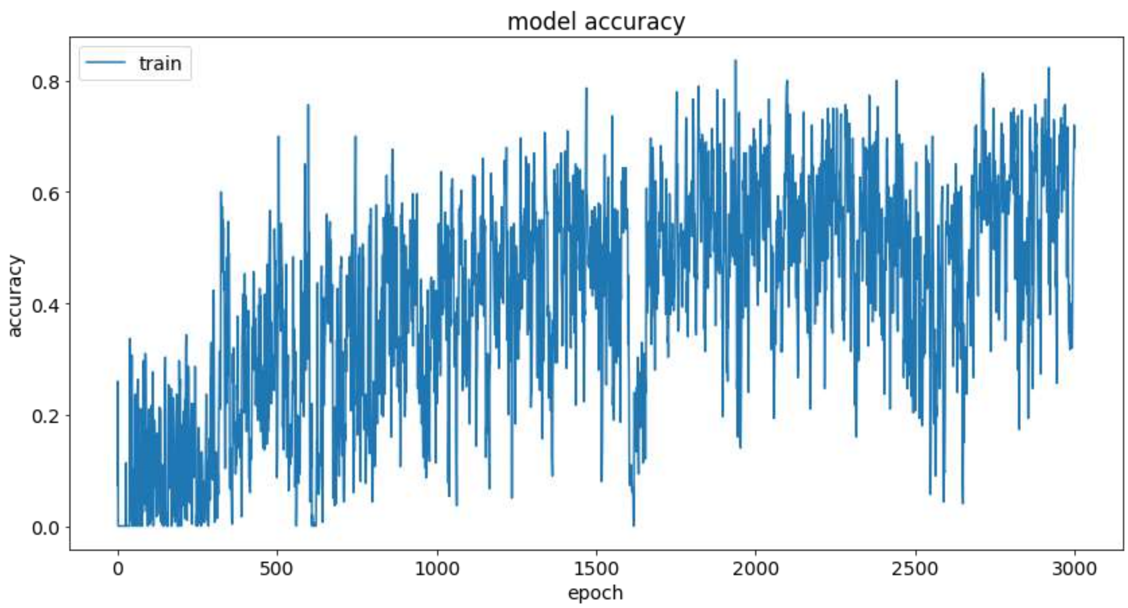


Fig. 7. Data set visualization for fourth implementation.

A 20-layer neural network was used with 500 neurons per layer, in all of which relu activation was used.



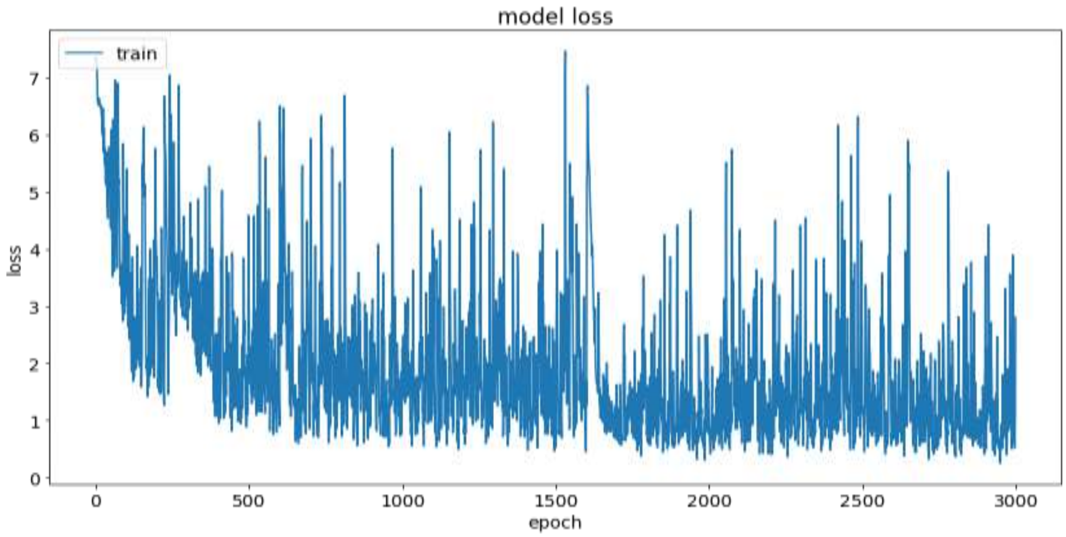


Fig. 7. Model that best fitted the data.

We see that there is improvement in the results, the deviation drops from 1.8 to 0.8 approximately. But the prediction is still unstable.

4. Conclusions

We can observe that the predictions improve in cases where the data are not normalized. It is also observed that the second methodology did not improve the predictions.

An important point to highlight is that the neural network shows better performance with the first methodology since it was specifically designed for that case. For the next two methodologies, the same neural network was used, but the data would improve if a distinct neural network were developed for each case.

To have a neural network capable of predicting any input, not just steps and sines, it is necessary to train the network with random inputs. This will be the next methodology to follow.

References

Alhanjouri, M. (2017). Speed control of DC motor using artificial neural network. *International Journal of Science and Research (IJSR)*, 7(3), 2140-2148.

Brownlee, J. (2022). How to Develop Multi-Output Regression Models with Python. *Machine Learning Mastery*. Retrieved from https://machinelearningmastery.com/multi-output-regression-models-with-python/

Chu, K. S. K., Chew, K. W., & Chang, Y. C. (2023). Fault-Diagnosis and Fault-Recovery System of Hall Sensors in Brushless DC Motor Based on Neural Networks. *Sensors*, 23(9), 4330.

Mamadapur, A., & Mahadev, G. U. (2019, August). Speed control of BLDC motor using neural network controller and PID controller. In 2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC) (pp. 146-151). IEEE.

Morante, S. (2018). Precauciones a la hora de normalizar datos en Data Science - Think Big Empresas. *Think Big*. Retrieved from https://empresas.blogthinkbig.com/precauciones-la-hora-de-normalizar/

Schmid, L., Gerharz, A., Groll, A., & Pauly, M. (2022). Machine Learning for Multi-Output Regression: When should a holistic multivariate approach be preferred over separate univariate ones? Retrieved from https://arxiv.org/abs/2201.05340

Shah, R., & Sands, T. (2021). Comparing methods of DC motor control for UUVs. *Applied Sciences*, 11(11), 4972.

Weerasooriya, S., & El-Sharkawi, M. A. (1991). Identification and control of a dc motor using back-propagation neural networks. *IEEE transactions on Energy Conversion*, 6(4), 663-669.