

The Australian National University
2600 ACT | Canberra | Australia



Australian
National
University

School of Computing

College of Engineering, Computing
and Cybernetics (CECC)

Unleashing the power of Machine Learning in Geodynamics

— 12 pt Honours project (S2 2023)

A thesis submitted for the degree
Bachelor of Advanced Computing (Honours)

By:
Xuzeng He

Supervisors:
Dr. Rhys P. Hawkins
Dr. Alberto F. Martin Huertas
Dr. Siavash Ghelichkhan

September 2023

Declaration:

I declare that this work:

- upholds the principles of academic integrity, as defined in the [University Academic Misconduct Rules](#);
- is original, except where collaboration (for example group work) has been authorised in writing by the course convener in the class summary and/or Wattle site;
- is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener;
- gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used;
- in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling.

September, Xuzeng He

Abstract

Table of Contents

1	Introduction	1
2	Background	2
2.1	Neural networks: Fully connected, convolutional and long short-term memory	2
2.2	Related works for solving geoid and mantle convection using Neural networks	2
3	Geoid prediction	3
3.1	Dataset of Geoid prediction	3
3.2	Fully connected Neural Network (FNN) for Prediction	4
4	Mantle Convection Simulation	8
4.1	Dataset of mantle convection simulation	8
4.2	Compression of temperature fields	8
4.3	Fully Connected Neural Network (FNN) for Prediction	8
4.4	Long short-term memory (LSTM) for Prediction	8
5	Concluding Remarks	9
5.1	Conclusion	9
5.2	Future Work	9
	Bibliography	10

Introduction

Introduction (actually more like background) to Geoid problem and Mantle Convection
In this project I explore the use of machine learning as a low-cost solution to both geoid problem and mantle convection problem...

Background

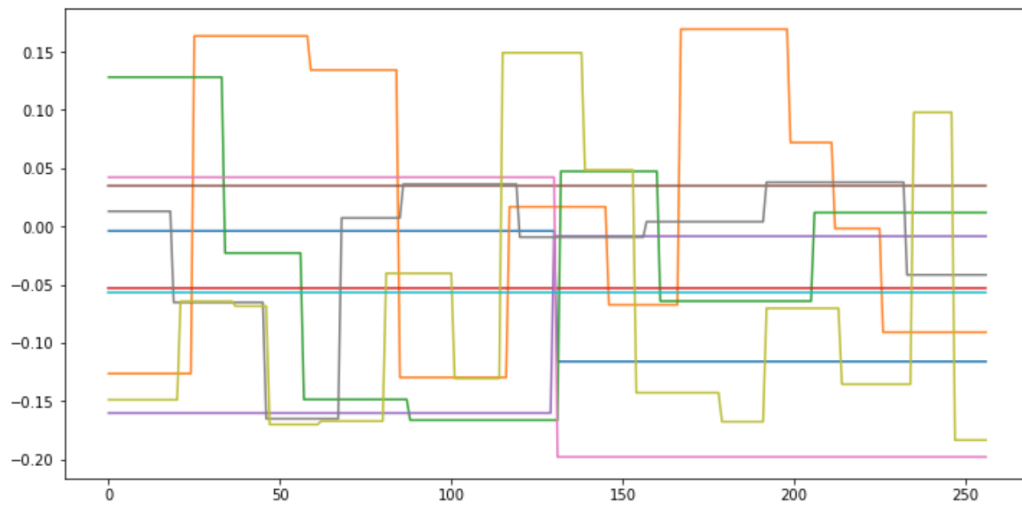
- 2.1 Neural networks: Fully connected, convolutional and long short-term memory
- 2.2 Related works for solving geoid and mantle convection using Neural networks

Geoid prediction

3.1 Dataset of Geoid prediction

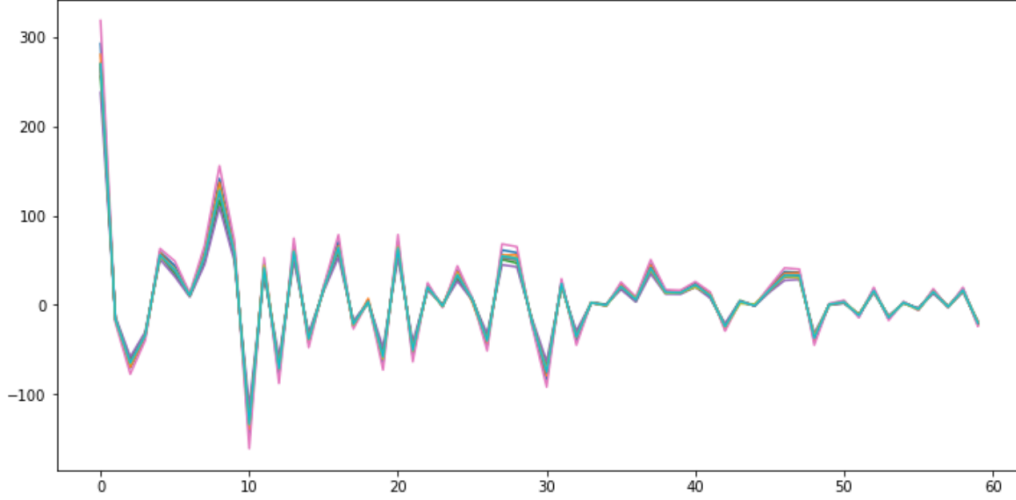
The data set consists of 1000 pairs of input and output. In the data set, input is a vector with 257 values representing a geoid model and output is a vector with 60 values representing a set of parameters. The following graphs plot 10 pairs of model input and output in separate figures to observe the patterns.

Every 100th input in the data set



3 Geoid prediction

Every 100th input in the data set



From the second plot we can observe that the outputs in the data set can be seen as some "curves" with the same patterns but different altitude. In this case, each of the 60 values in the output data are normalised to be between 0 and 1 using their maximum and minimum values separately before feeding into the neural network. This is because each of the parameters in an output vector can be seen as equal and we want to prevent the neural network from spending most of its effort learning the parameter with a higher range. Hence, one can expect a higher accuracy when the parameters in the output data are standardized to a same range.

After the output data is normalised using a scaler, the entire data set is randomly divided in a ratio of 8:1:1: 80 per cent of the data set is used for training, 10 per cent of the data for testing accuracy and the remaining 10 per cent to perform validation during training and prevent overfitting. For a data set with 1000 samples, this result in a train-test-validation split of 800-100-100.

3.2 Fully connected Neural Network (FNN) for Prediction

To test FNNs with different architectures (e.g. different number of hidden layers and neurons per hidden layer) or other hyperparameters (e.g. optimizer), a systematic testing method is applied. This method mainly consists of three files: one file to store all the different set of FNN architectures and hyperparameters in a text format, another file to fetch all these combinations of architectures and hyperparameters line by line, build them as FNN models and train these models, and the last file for testing and visualisation of the trained models by specifying the path of the trained model. The training file and the testing file are both in the format of a Jupyter Notebook.

The trained FNN architecture (in the format of a light-weight file) along with another

3 Geoid prediction

text file contains the training loss and validation loss during training will be stored in a specified path for further testing and visualisation. The name of these two files uniquely defines each experiment by including the values of hyperparameters to generate the model in the file names. These files are also put in separate folders with the folder name associated with commit IDs to handle tracking of the process during the research in an educated or extensible way.

In this way, one can open the same testing Jupyter Notebook in different browser tabs, and then visualize simultaneously different models in different tabs using a cell in which the paths to the FNN file and its training data are specified.

The systematic testing capability is implemented here to ensure traceability. In other words, as different values of the hyperparameters are tested, I would like to be able to record the results (e.g., the trained network and the training data) so that I don't have to repeat them again or rely on my memory to compare the the performance of different structures.

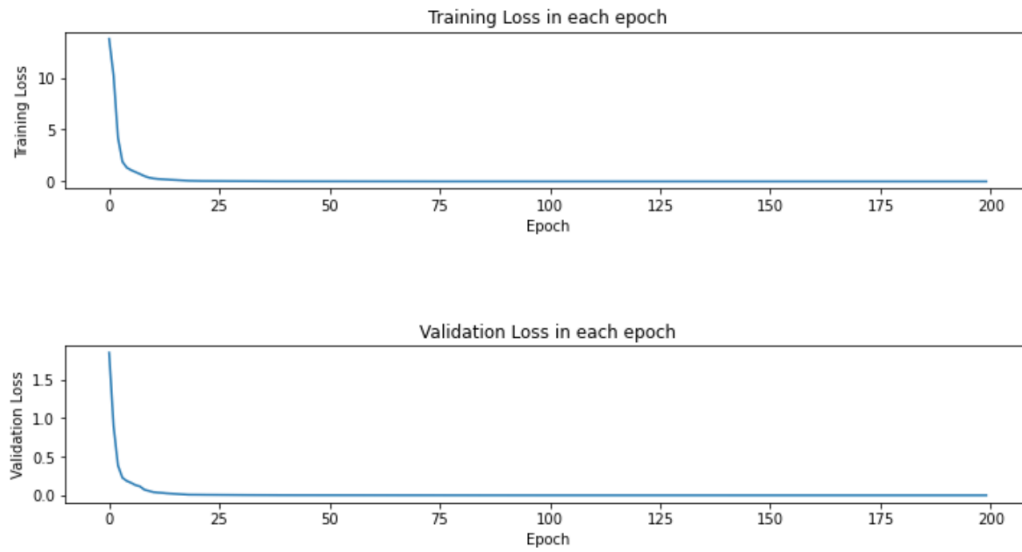
Also, to prevent overfitting, a variation of the early-stopping method is used during training. The normal early-stopping method let the network train until the error function evaluated on the validation set starts to increase beyond a certain threshold([Prechelt, 2012](#)), while my implementation only stores the best model during training (the one with the lowest validation loss) in a specified path and allows the network to keep training as normal. In this case, the output model is the best model instead of the last trained model. This method is also used in the following chapter when implementing the ConvAE, FNN and LSTM to solve the mantle convection problem.

After testing with NNs with architectures of different number of hidden layers and neurons per hidden layer, I found that architectures with a total number 3–4 hidden layers seemed to perform the best.

In the following figures, I present results from a FNN with 4 hidden layers with 200, 160, 120 and 80 neurons, ReLU as activation function, MSELoss as loss function, and trained for 200 epochs using Mini-Batch Gradient Descent (with a batch size of 16).

3 Geoid prediction

Training loss and Validation loss

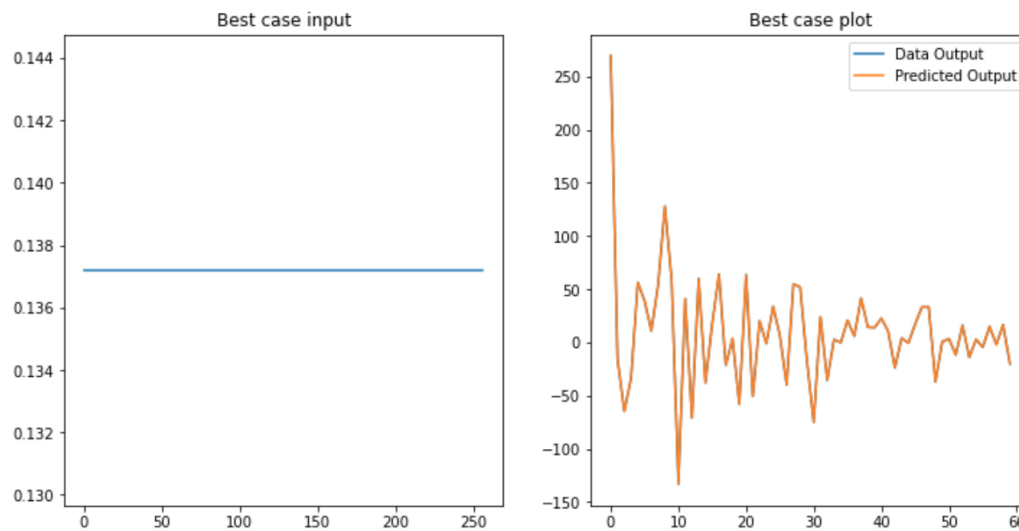


Overall testing result

Total loss for the model is 0.00023521817990547456, and accuracy is 100%
When the loss threshold is set to 0.01, accuracy is 100%
When the loss threshold is set to 0.001, accuracy is 99%
When the loss threshold is set to 0.0001, accuracy is 96%
When the loss threshold is set to 1e-05, accuracy is 50%

Best input and output

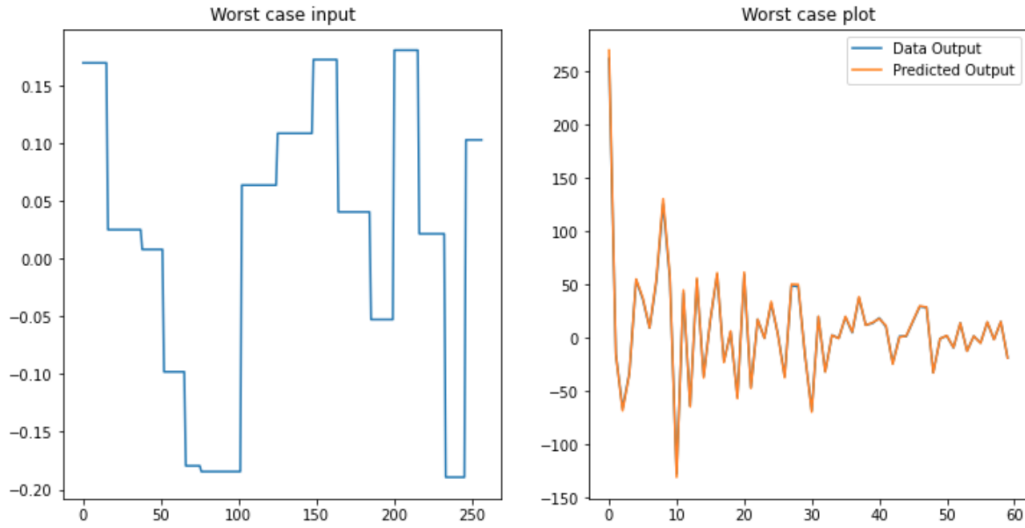
Best model has a error of 8.997102327635002e-07



3 Geoid prediction

Worst input and output

Worst model has a error of 0.0013982787387692296



On average, the prediction loss are low even when the loss value here is calculated using the normalised output data and no overfitting occurs. The accuracy of the prediction is nearly 100% when the threshold is set to be 10 times lower than the worst loss value. Overall, FNN is able to accurately solve the Geoid problem without numerous amount of data.

Mantle Convection Simulation

- 4.1 Dataset of mantle convection simulation
- 4.2 Compression of temperature fields
- 4.3 Fully Connected Neural Network (FNN) for Prediction
- 4.4 Long short-term memory (LSTM) for Prediction

Concluding Remarks

5.1 Conclusion

5.2 Future Work

Bibliography

PRECHELT, L., 2012. Early stopping — but when? In *Lecture Notes in Computer Science*, 53–67. Springer Berlin Heidelberg. doi:10.1007/978-3-642-35289-8_5. https://doi.org/10.1007/978-3-642-35289-8_5. [Cited on page 5.]