Birla Institute of Technology and Science, Pilani
K K Birla Goa Campus

Modelling Neutron Star Parameters Using Artificial Neural Networks

(June 27, 2021)

A Study-Oriented Project

(Second Semester 2020 - 2021)

By

Ameya Thete

(2018B5A70885G)

Project Mentors

Dr. Kinjal Banerjee

Dr. Tuhin Malik

Table of Contents

List of Figures

List of Tables

Abstract

Neutron stars are studied by looking at the equation of state (EoS) the nuclear matter. The EoS is characterised by certain intercorrelated nuclear matter parameters (NMPs) from which the properties of stars (such as mass, radius and tidal deformability) can be derived. These NMPs are usually extracted from the correlations with specific finite nuclei observables, and the NMPs so obtained constrain the properties of neutron stars. However, demanding different correlations among NMPs and then deriving star properties is a computationally time-consuming process. In this project, we develop a machine learning framework based on artificial neural networks that can predict neutron star properties for a given NMP distribution (specifically a MVGD of NMPs). We also investigate the role of intercorrelations among NMPs and study their effects on the model and neutron star properties.

1 Introduction

Neutron stars are the cores of erstwhile supergiant stars resulting from supernova explosions, followed by a gravitational collapse of the cores. They are composed almost entirely of neutrons and are some of the densest objects in the universe. Studying these stars offers an insight into different theories of nuclear physics and physics beyond the Standard Model. To understand the theory of neutron stars, one needs the theory of matter at extreme conditions, *i.e.*, the theory of infinite nuclear matter equation of state (EoS), where the EoS is the energy (or pressure) as a function of density.

1.1 EoS Parameterisation

For a given baryonic density ρ $(= \rho_n + \rho_p)$, where ρ_n and ρ_p are the neutron density and proton densities respectively, and asymmetry $\delta = (\rho_n - \rho_p)/\rho$, the energy per nucleon $e(\rho, \delta)$, can be decomposed into a symmetric component $e(\rho, 0)$ and a density-dependent symmetric energy coefficient $S(\rho)$:

$$e(\rho, \delta) = e(\rho, 0) + S(\rho)\delta^2 \tag{1.1}$$

Now, consider the Taylor expansion of this energy functional around the saturation density ρ_0. This gives

$$e(\rho, 0) = e(\rho_0) + \frac{K_0}{2}x^2 + \frac{Q_0}{6}x^3 + \mathcal{O}(x^4) \tag{1.2}$$

$$S(\rho) = J_0 + L_0 x + \frac{K_{sym,0}}{2}x^2 + \mathcal{O}(x^3) \tag{1.3}$$

where $x = (\rho - \rho_0)/3\rho_0$ and $J_0 = S(\rho_0)$ is the symmetry energy at saturation density. The Taylor coefficients K_0, Q_0, L_0 and $K_{sym,0}$ are the incompressibility, the skewness coefficient, the symmetry energy slope and the curvature, respectively, all evaluated at the saturation density [1]. These quantities are the key nuclear matter parameters (NMPs) that describe any EoS. Hence, an EoS can be imagined as a point in the seven-dimensional parameter space of NMPs characterised by a multi-variate Gaussian distribution with a mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, with the parameters being $e_0, \rho_0, K_0, Q_0, J_0, L_0$, and $K_{sym,0}$. Thus, the i^{th} EoS can be written as

$$\text{EoS}_i = \{e_0, \rho_0, K_0, Q_0, J_0, L_0, K_{sym,0}\} \equiv \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{1.4}$$

where

$$\boldsymbol{\mu}^T = \begin{pmatrix} \bar{e}_0 & \bar{\rho}_0 & \bar{K}_0 & \bar{Q}_0 & \bar{J}_0 & \bar{L}_0 & \bar{K}_{sym,0} \end{pmatrix} \tag{1.5}$$

and

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma^2_{e_0} & Cov(e_0, \rho_0) & \dots & Cov(e_0, K_{sym,0}) \\ Cov(\rho_0, e_0) & \sigma^2_{\rho_0} & \dots & Cov(\rho_0, K_{sym,0}) \\ \vdots & \ddots & & \vdots \\ Cov(K_{sym,0}, e_0) & Cov(K_{sym,0}, \rho_0) & \dots & \sigma^2_{K_{sym,0}} \end{pmatrix} \tag{1.6}$$

where $Cov(X, Y)$ is the covariance between X and Y and σ^2_Z is the variance of Z.

Hence, once a NMP distribution is fixed, a equation of state can be calculated by using the Taylor expansion for the functional, or using a mean field approach. Once an EoS is determined, it can be used to calculate

various neutron star properties such as its maximum mass, radius and tidal deformability by solving the neutron star structure equations [2].

Correlations between nuclear matter parameters and neutron star properties have been explored using several models. These studies, however, show considerable model dependence since different models with similar values of the nuclear matter parameters can possibly result in different EoSs. Futhermore, given a large number of NMPs, calculating (even computationally) the EoSs and subsequently the star properties is a computationally expensive process. Looking at and demanding different correlations between NMPs to observe resulting properties requires repeating this time consuming process. Therefore, the task of this project is to develop a machine learning model that, once trained, can predict neutron star properties for a given NMP distribution while also maintaining physically relevant correlations between the target variables and input parameters.

THIS SPACE IS INTENTIONALLY LEFT BLANK

2 Nuclear Matter Parameter Data

We obtained three separate datasets, each corresponding to a different distribution of NMPs. These datasets were labelled `Case I`, `Case II`, and `Case III` respectively. All datapoints were obtained by sampling a multi-variate Gaussian distribution, and the mean and standard deviations for all parameters are listed in Table 1. `Case I` corresponds to a completely independent distribution of parameters, with the correlations among different NMPs being zero. In `Case II`, the correlation between L_0 and $K_{sym,0}$ is set to 0.8 while the remaining correlations are not fixed manually. `Case III` is similar to `Case II`, but here the values of the other NMPs are fixed to their central (mean) values. For all cases, the distribution of NMPs are filtered such that the resulting EoSs satisfy the causality condition (the speed of sound does not exceed the speed of light) and yield the maximum mass of a neutron star above $1.8M_\odot$. This results in there being about 3000 observations for each case.

Parameter (P)	μ_P	σ_P
e_0	-16.0	0.25
ρ_0	0.16	0.005
K_0	230.0	20
Q_0	-300.0	100
J_0	32.0	3
L_0	60.0	20
$K_{sym,0}$	-100.0	100

Table 1: The mean μ_P and standard deviation σ_P of the MVGD. All quantities in MeV except ρ_0, which is in fm^{-3}. EoSs sampled for three different cases using three different distributions for P [3].

The reason for focusing on the $L_0 - K_{sym,0}$ correlations is because the tidal deformability of neutron stars (one of our target variables) is mainly sensitive to this correlation [3]. These three distributions will allow us to unravel how existing correlations among the NMPs might affect the correlations between the neutron star properties and the NMPs.

Figure 1 shows the 7×7 matrix for the correlation coefficients between the NMPs for `Case II` data. It should be emphasised that all seven NMPs can be independently varied within the models. These correlations among the NMPs are the reflection of constraints imposed by finite nuclei. The strong $L_0 - K_{sym,0}$ correlation which we are particularly interested in has also been observed in many other nuclear models.

Each dataset also contains the target NS properties that have to be predicted. These are the maximum NS mass, the maximum NS radius (R_{max}), the radius for $M = 1.4M_\odot$ ($R_{1.4}$), and the tidal deformabilities (Λ_M for $M = 1.0, 1.4, 1.8\ M_\odot$). A neural network is trained on this data to predict NS properties. The results are then compared with existing models to verify if correlations are preserved and further studies are conducted to observe the effects of changing NMP correlations on the output sector.

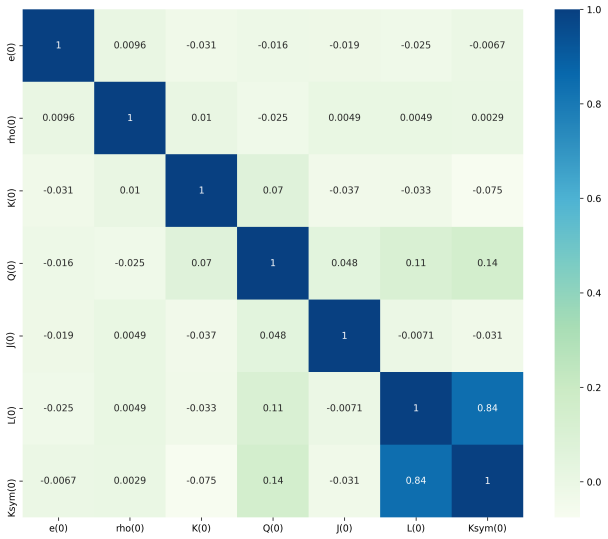

Figure 1: The correlations among various NMPs for `Case II` data. The only significant correlation is between L_0 and $K_{sym,0}$

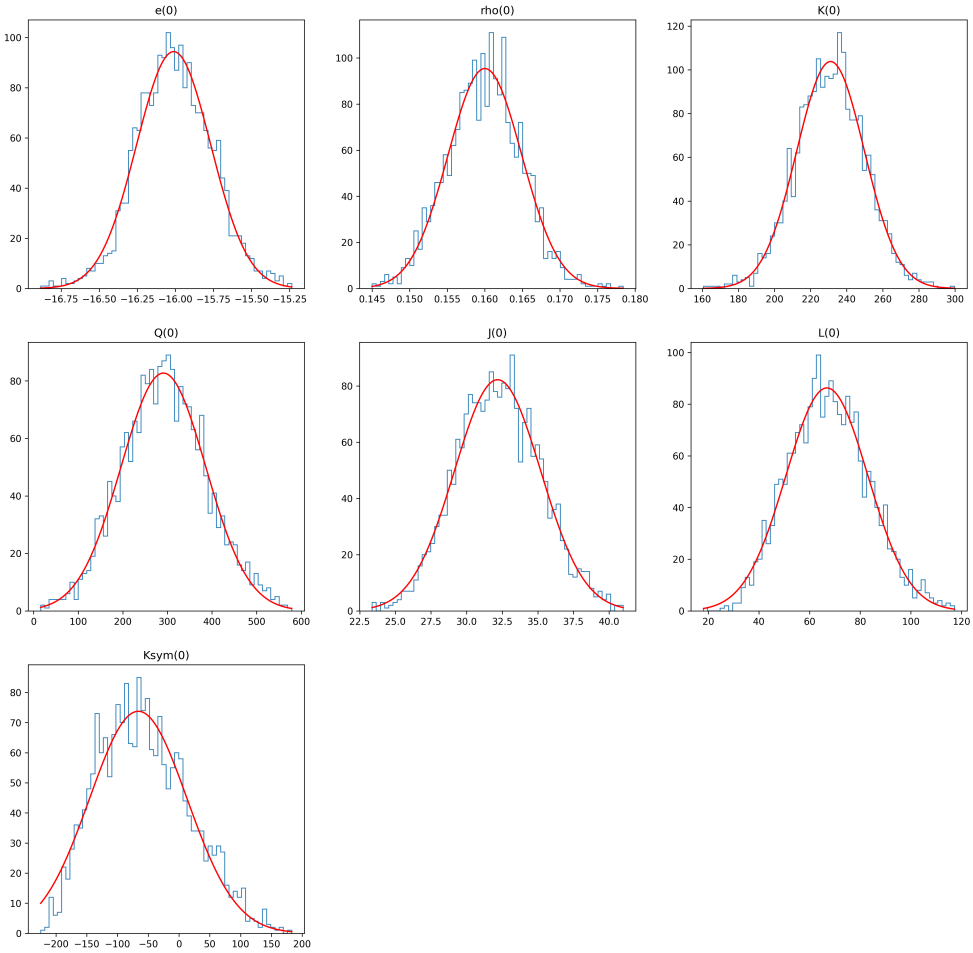

Figure 2: The distribution of features in `Case II` data. The red Gaussian fit reflects the fact that the data is sampled from a MVGD.

3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a biologically-inspired mathematical model for statistical learning composed of a interconnected units called neurons. A neuron receives a signal then processes it and can signal neurons connected to it. The signal at each connection, called an edge, is a real number and the output of each neuron is computed by a non-linear function of the sum of its inputs. Neurons and edges typically also have associated weights that can be altered as the network "learns" the task at hand. Hence, an ANN is capable of modelling and representing complex mathematical relationships.

3.1 Structure

3.1.1 Neuron

A neuron with a label j receiveing an input $p_j(t)$ from predecessor neurons consists of:

1. an activation $a_j(t)$, the neuron's state;
2. an optional threshold $\theta_j(t)$;
3. an activation function f that computes the new activation at time $t+1$ from $a_j(t)$, $\theta_j(t)$ and $p_j(t)$ such that

$$a_j(t+1) = f(a_j(t), p_j(t), \theta_j) \tag{3.1}$$

3.1.2 Propagation function

The propagation function computes the input $p_j(t)$ to the neuron j from the outputs $o_i(t)$ and has the form

$$p_j(t) = \sum_i o_i(t) w_{ij} \tag{3.2}$$

where w_{ij} are the weights of the connections.

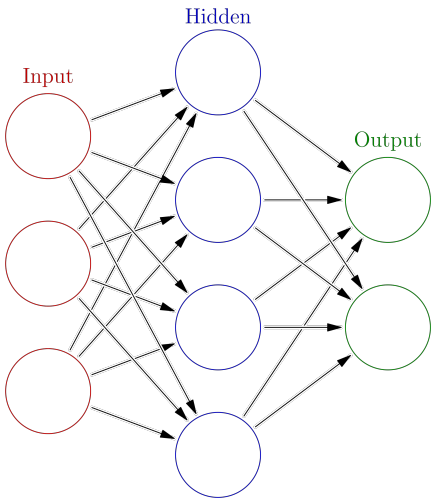

Figure 3: A typical neural network consists of multiple layers of neurons. A neuron is represented by a circle and the arrows represent connections between them. [4]

3.2 Backpropagation

Backpropagation is a method used to modify the connection weights to compensate for each error found during learning. The error amount is effectively divided among all the connections. Technically, the algorithm calculates the gradient of a cost function associated with a given state with respect to the weights. The weight updates are then performed via stochastic gradient descent.

4 Methodology

The dataset was preprocessed before being fed to a neural network architecture as training data. While the dataset was clean and had no notable outliers that could affect the performance of the network, the features had variable ranges and needed to be scaled. Standardisation of data is also a common requirement for estimators as they tend to perform poorly if the features are not more or less standard normally distributed. Therefore the dataset was scaled to a standard normal distribution (mean 0 and unit standard deviation).

A total of three models were created by training a separate model on each case data to observe the effect of correlations among the NMPs on NS properties. Hence, each case data was divided into a 70:20:10 split corresponding to training, testing and validation sets. The purpose of a validation set is to monitor the performance of the network during training and can provide insights into possible overfitting scenarios. Since one of the goals of the project was to develop a model-independent estimator for NS properties, we also wanted to study if a model trained on a certain distribution and correlations would be able to also accurately make predictions if it was fed data from another, related distribution but with different correlations among the inputs. Consequently, each trained model was also evaluated on case data it was not trained on.

Once a suitable model out of the three was selected, we generated new data by fixing all but one NMP correlations and generated three sets of datasets for $L_0 - K_{sym,0}$ correlations of 0.3, 0.6, and 0.9. The best model was evaluated on these datasets to demonstrate its ability to effectively capture input-output correlations and the fact that the predictions were physically correct. As a next step, we repeated the generation of three datasets, but this time we fixed all NMP correlations and instead changed the standard deviations (errors) of the NMPs to 0.3, 0.6 and 0.9 times the original error. The model was once again evaluated on this set of data to study if the error in the NMPs had any effect on the input-output correlations.

4.1 Model Architecture

We adopted a neural network architecture consisting of two hidden dense layers consisting of fifteen neurons each and a final output layer consisting of 6 neurons for the target properties. Figure 4 provides a visualisation of the model architecture. We used the rectified linear unit, or ReLU activation function on each neuron in the hidden layer (typically no activation is applied on the output layer unless the output has to be constrained to a certain domain). While ReLU can lead to some neurons dying during traning, we did not find any appreciable change in the model performance by changing the activation function to a LeakyReLU function. We also did not choose a large learning rate ($\eta = 0.01$), hence the dying ReLU problem was mitigated [5].

The weights of the network were initialised randomly using the Glorot initialisation technique. Trying out the He initialisation technique did not yield any change in performance and hence, we continued with Glorot.

Furthermore, we also experimented with Dropout but it did not positively affect the performance of the network. This exact architeture was used to create three neural networks which were trained on invidual case datasets and then evaluated on the same, as well as other case datasets.

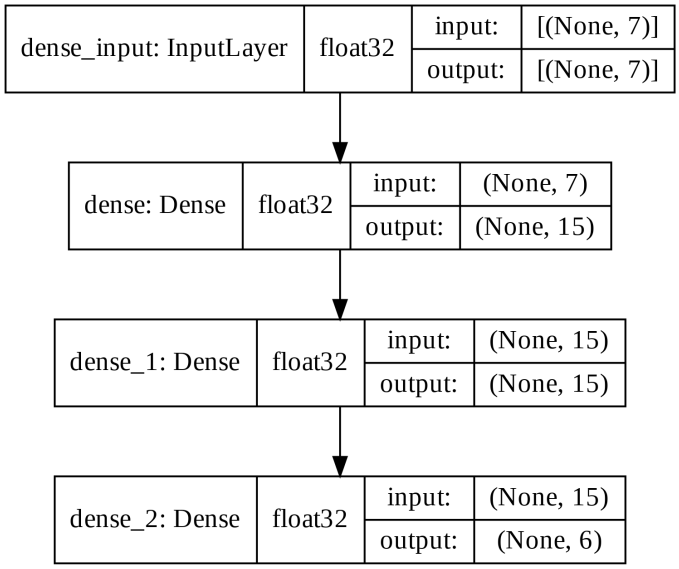

Figure 4: Neural network architecture of the model implemented. The model consists of 3 dense layers in total, two hidden and one output. `InputLayer` is a layer that is implemented to accept the inputs for each feature to the network, it has no other significant use.

4.2 Evaluation Metrics

Each model was trained to minimise the mean square error between the predicted and actual neutron star properties. Given n data points, for a feature $y^{(i)}$, and a prediction $\hat{y}^{(i)}$, the mean squared error is defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2 \tag{4.1}$$

We also looked at the correlations between Λ_M and L_0 as well as Λ_M and $K_{sym,0}$ for $M = 1.0, 1.4, 1.8$ on the predictions and then compared them to established values in literature. The theoretical correlations are listed in Table 2.

THIS SPACE IS INTENTIONALLY LEFT BLANK

		$\Lambda_{1.0}$	$\Lambda_{1.4}$	$\Lambda_{1.8}$
`Case I`	L_0	0.82	0.56	0.22
	$K_{sym,0}$	0.26	0.58	0.71
`Case II`	L_0	0.90	0.83	0.70
	$K_{sym,0}$	0.84	0.86	0.80
`Case III`	L_0	0.96	0.91	0.82
	$K_{sym,0}$	0.92	0.97	0.98

Table 2: The values of the correlation coefficients for $\Lambda_{1.0,1.4,1.8}$ with L_0 and $K_{sym,0}$[3]

5 Results

Each model was trained for a period of 40 epochs with a batch size of 16. The optimiser used was Adam. The performance of all three trained models is tabulated below.

Model	Trained on	Final Training MSE	Final Validation MSE
Model 1	`Case I`	0.0039	0.0046
Model 2	`Case II`	0.0347	0.0270
Model 3	`Case III`	0.0681	0.370

Table 3: The performance of all three trained models. While Model 1 perfomed the best on its own testing set, it performed very poorly on other case data and was not able to accurately capture correlations.

We find that all three models performed rather similarly on their own testing sets, but when they were tested on other case data, that is, case data on which they were not trained on, Models 1 and 3 performed rather poorly. They were also unable to accurately predict the correlations expected between Λ_M and L_0 and $K_{sym,0}$. The RMS values for the predictions of Model 2 are tabulated in Table 4. We find that the model is able to accurately predict all NS properties and the RMS is well within the error range for the outputs. Therefore, we chose to go ahead with Model 2 for further studies.

The next task was to study the correlations predicted by Model 2. These have been tabulated in Table 5.

NS Property	RMSE
NS Mass	0.0243
R_{max}	0.0879
$R_{1.4}$	0.1940
$\Lambda_{1.0}$	123.8
$\Lambda_{1.4}$	19.23
$\Lambda_{1.8}$	5.344

Table 4: The RMS values of predicted NS properties by Model 2.

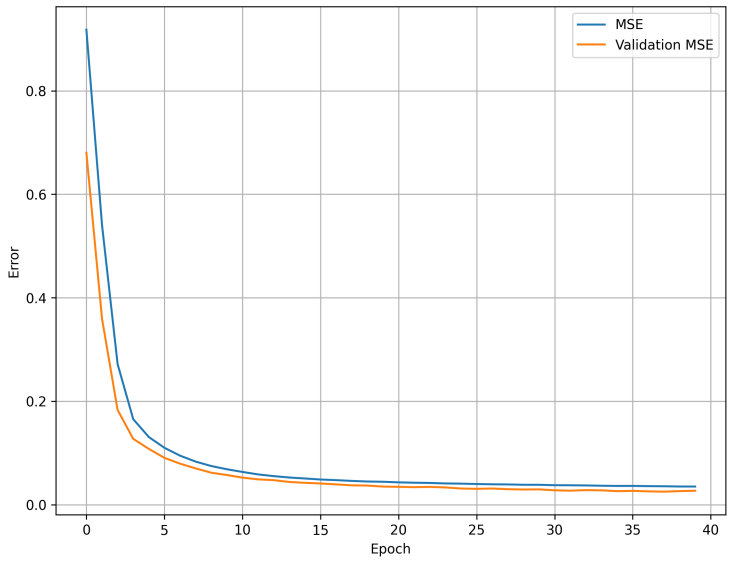

Figure 5: Training and Validation loss curves for Model 2.

	$\Lambda_{1.0}$	$\Lambda_{1.4}$	$\Lambda_{1.8}$
L_0	0.8996	0.8202	0.6870
$K_{sym,0}$	0.8440	0.8503	0.7953

Table 5: The values of the correlation coefficients for predicted $\Lambda_{1.0,1.4,1.8}$ with L_0 and $K_{sym,0}$

Comparing these values with the `Case II` row in Table 2 shows that the model is able to capture the correlations between the input sector and output sector very accurately.

5.1 Effect of $L_0 - K_{sym,0}$ correlations on the output sector

To study how the correlations among the NMPs would affect the correlations between the input and out sectors, we generated three new datasets. These datasets were obtained by manually setting the $L_0 - K_{sym,0}$ correlations to $0.3, 0.6$, and 0.9. The best model from the previous study,that is Model 2 (which was trained on `Case II` data) was then used to predict NS properties for these three datasets. Confidence ellipses in the planes of $\Lambda_M - L_0$ and $\Lambda_M - K_{sym,0}$ with $M = 1.0, 1.4$, and $1.8 M_\odot$ were plotted to observe the results.

From Figure 6, we can observe that as the $L_0 - K_{sym,0}$ correlation gets stronger, the $\Lambda_M - K_{sym,0}$ and $\Lambda_M - L_0$ correlations get stronger as well. This indicates that the model is consistent with theoretical observations and is able to accurately capture the correlations among NMPs and NS properties.

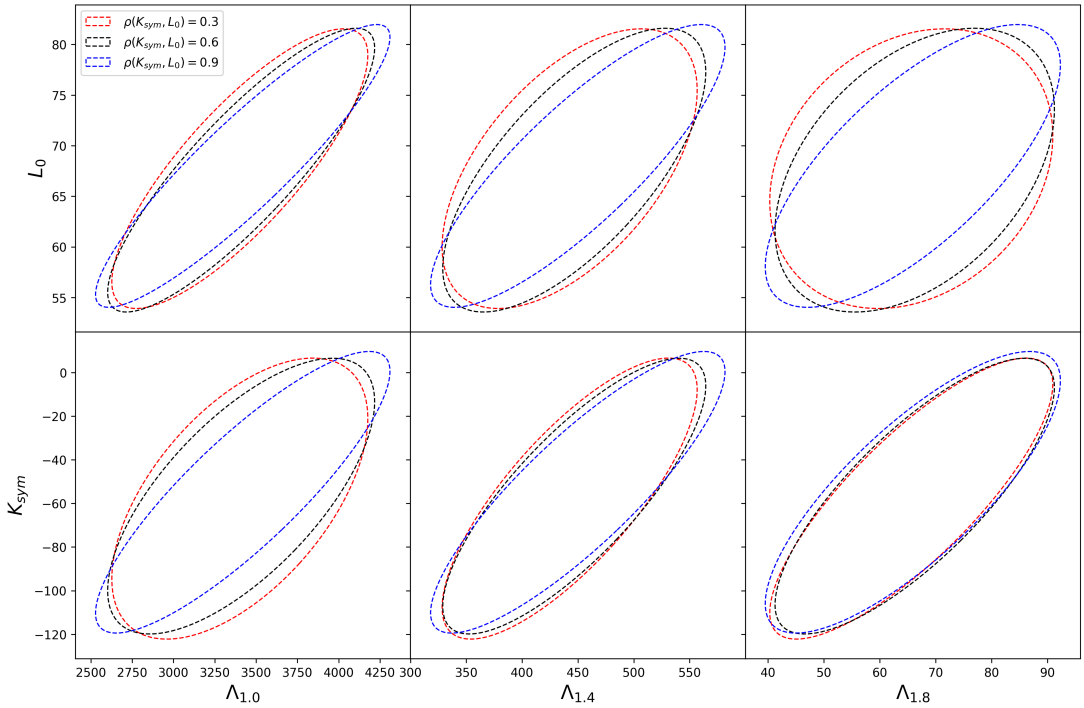

Figure 6: The 1σ confidence ellipses in the planes of $\Lambda_M - L_0$ (top) and $\Lambda_M - K_{sym,0}$ (bottom) with $M = 1.0, 1.4,$ and $1.8 M_\odot$ obtained for the datasets with $L_0 - K_{sym,0}$ correlation set to $0.3, 0.6$ and 0.9.

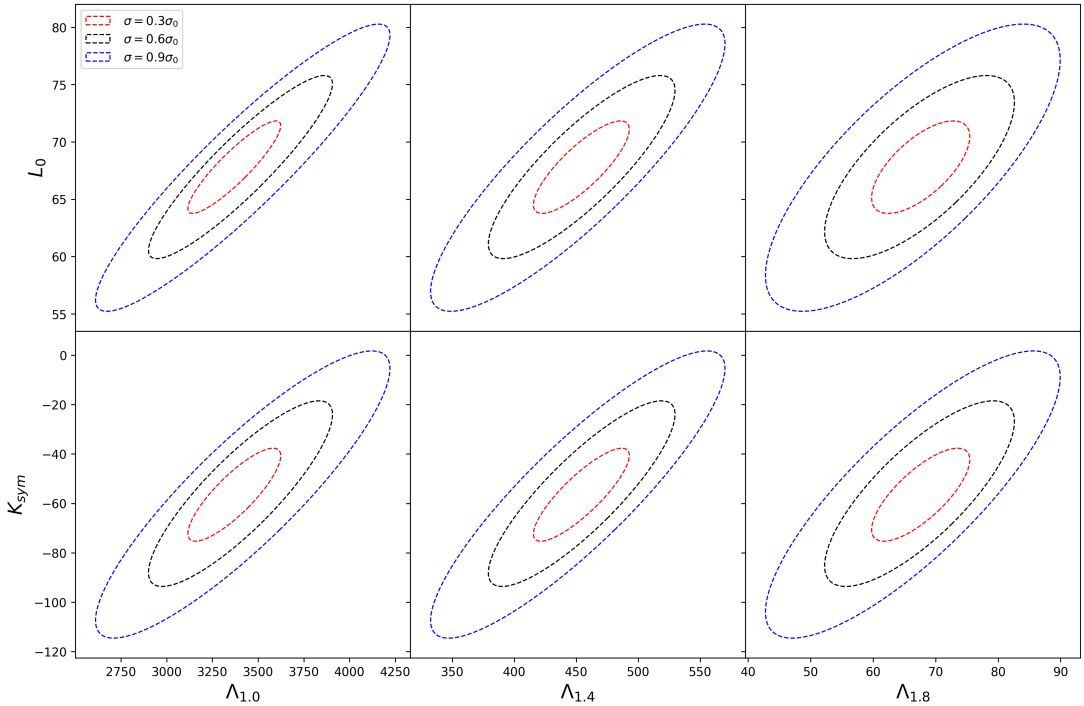

Figure 7: The 1σ confidence ellipses in the planes of $\Lambda_M - L_0$ (top) and $\Lambda_M - K_{sym,0}$ (bottom) with $M = 1.0, 1.4,$ and $1.8 M_\odot$ obtained for the datasets with standard deviations (errors) set to $0.3, 0.6$ and 0.9 times the original.

5.2 Effect of NMP errors on the output sector

Next, to study how the errors (or standard deviations) of the NMPs might affect the model and the input-output correlations, we generated three datasets by fixing the intercorrelations between NMPs, but instead varying the individual errors for the parameters as a function of the original standard deviation, σ_0. The best model from the previous study,that is Model 2 (which was trained on `Case II` data) was then used to predict NS properties for these three datasets. Confidence ellipses in the planes of $\Lambda_M - L_0$ and $\Lambda_M - K_{sym,0}$ with $M = 1.0, 1.4$, and $1.8 M_\odot$ were plotted to observe the results.

Figure 7 shows the results we obtained. Here, we find that the standard deviation has very little effect on the input-output correlations.

5.3 Variation of tidal deformability with NS Radius

The tidal deformability can be expressed in terms of a dimensionless quadrupole tidal Love number k_2 as

$$\Lambda = \frac{2}{3} k_2 R^5 \tag{5.1}$$

where R is the radius of the neutron star. It is evident that the tidal deformability depends on the radius and the Love number, and hence it is expected that $\Lambda_{1.4}$ is strongly correlated with $R_{1.4}$. However, k_2 is also moderately correlated with $R_{1.4}$ [6], which ensures the persistence of a strong correlation. Figure 8 shows the scatter plot of $\Lambda_{1.4}$ versus $R_{1.4}$ and a curve is drawn to fit the data.

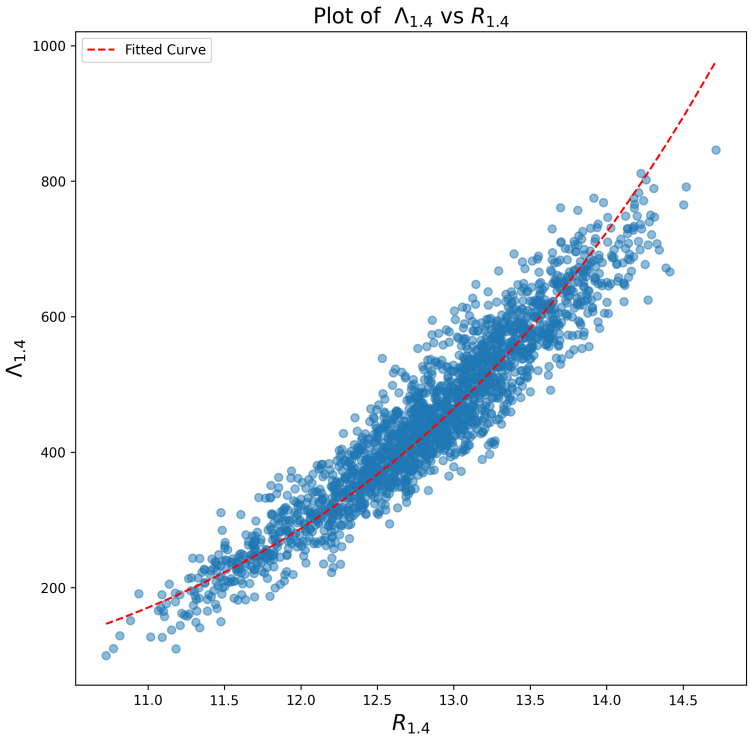

Figure 8: The variation of tidal deformability $\Lambda_{1.4}$ with the radius $R_{1.4}$ obtained for the model predictions. The red dotted line indicates a fitted curve.

The fitted curve is

$$\Lambda_{1.4} = 9.26 \times 10^{-5} (R_{1.4})^6 \tag{5.2}$$

This curve is in excellent agreement with the values in literature [6]. The exponent, though mass depenedent is close to six.

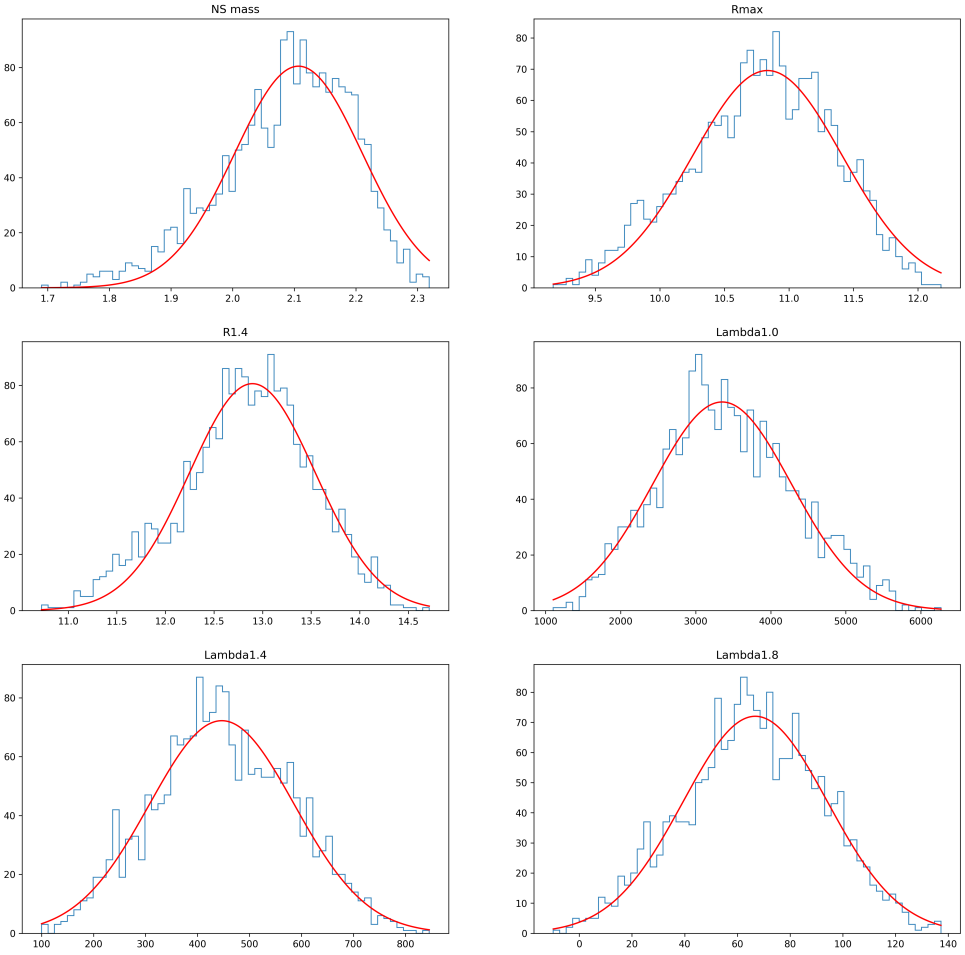

Figure 9: The parameter distribution for the output sector for the dataset generated for the $L_0 - K_{sym,0}$ correlation set to 0.8

6 The Reverse Problem

(Added 27 June 2021)

Next, we wished to study the reverse problem of predicting the EoS parameters given neutron star properties. To model this problem, we used techniques similar to the ones we used to model the forward problem. Here, we considered `Case II` data which contained both NMP and NS Properties. The task now was to construct a neural network that could generate NMPs for an EoS in the output sector in response to neutron star properties provided in the input sector.

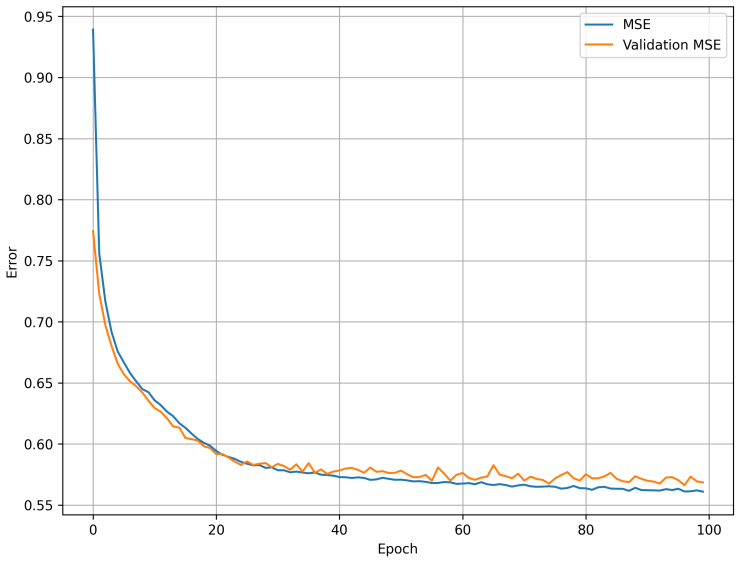

Figure 10: Training and Validation loss curves for the model trained on the reverse problem.

As described previously, the dataset was preprocessed before being fed to a neural network architecture as training data. While the dataset was clean and had no notable outliers that could affect the performance of the network, the features had variable ranges and needed to be scaled. Therefore the dataset was scaled to a standard normal distribution (mean 0 and unit standard deviation). The dataset was divided into a 70:20:10 split corresponding to training, testing and validation sets.

For the computational implementation we used the Python library Keras, with TensorFlow 2.4 as the back-end. The neural network archetecture is as follows: the network consists of two hidden dense layers having fifteen and twenty neurons, respectively. The final output layer consists of 7 neurons for the target NMPs. We used the rectified linear unit, or ReLU activation function on all layers except the output, which had of a linear activation function. The loss function used for the problem is the Mean Squared Error or MSE. The optimiser used for the network is Adam with its default parameters. To mitigate anh overfitting, we also implemented light weight decay/L^2 regularisation on the two hidden layers, with each term having a regularisation coefficient of 0.005.

The neural network was trained for 100 epochs with a batch size of 16. Figure 10 shows the variation of the loss function on the training and validation set as a function of epochs.
We observe that the model is satisfactorily able to predict NMPs. The RMS values for the predictions made by this model are tabulated in Table 6.

Nuclear Matter Parameter	RMSE
e_0	0.2436
ρ_0	0.0048
K_0	15.995
Q_0	73.490
J_0	2.261
L_0	5.573
$K_{sym,0}$	27.780

Table 6: The RMS values of predicted NMPs.

Next, we wanted to see if constraining values of the NS properties would have an effect on the distribution of the NMPs. We applied a filter on the NS Mass property in `Case II` to eliminate all data points which lied beyond a 1.5σ range around the mean. We then plotted the distribution of NMPs before and after the filter was applied.

We did not observe any appreciable change in the distribution of the NMPs after the filter was applied.

7 Conclusion

In this work we have shown that one can successfully model and estimate neutron star parameters using deep learning techniques and obtain results within a 0.001% margin. We have also showed that the neural-network based architecture is capable of predicting and retaining all major correlations in the input-output sector.

We have also shown that neural network based models are capable of satisfactorily generating Equations of State given neutron star properties as inputs.
Futher work needs to be done on the rigidity of such models and seeing if the model can reveal more correlations between the properties and the NMPs.

References

[1] I. Vidaña, C. Providência, A. Polls, and A. Rios, "Density dependence of the nuclear symmetry energy: A microscopic perspective," *Physical Review C*, vol. 80, Oct 2009.

[2] R. R. Silbar and S. Reddy, "Neutron stars for undergraduates," *American Journal of Physics*, vol. 72, p. 892–905, Jul 2004.

[3] T. Malik, B. K. Agrawal, C. Providência, and J. N. De, "Unveiling the correlations of tidal deformability with the nuclear symmetry energy parameters," *Physical Review C*, vol. 102, Nov 2020.

[4] Glosser.ca, "Derivative of file:artificial neural network.svg." Own work, CC BY-SA 3.0.

[5] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[6] T. Malik, N. Alam, M. Fortin, C. Providência, B. K. Agrawal, T. K. Jha, B. Kumar, and S. K. Patra, "Gw170817: Constraining the nuclear matter equation of state from the neutron star tidal deformability," *Physical Review C*, vol. 98, Sep 2018.