Incorporating Domain Knowledge in Neural Networks

Zhongqi Ma

Mentors: Maria Han Veiga, Dao Nguyen

August 2022

1 Abstract

A common scenario when making value predictions using machine learning models is that there are some fixed traits that the particular datasets would follow. For example, in physics it is common knowledge that density can not be negative, and in math a parabola is symmetric with respect to some x value. In this paper, we introduce a way to incorporate known knowledge regarding the dataset into neural network models to make better predictions, and showcase the differences in results that it can cause. We focus on a parabolic model and the inclusion of domain knowledge is achieved through changing the loss function. In reality we expect there to be multiple other approaches to including domain knowledge that would yield potential improvements to the neural network model

2 Introduction

Neural network models has become one of the most common and powerful tools used in pattern detection and value prediction. However, as neural network models have become one of the most common data prediction methods, one of its biggest limitations is also magnified. Neural networks rely heavily on the known training data that is given in their process of value prediction. The problem that arises with this dependence is that if the known training data fails to show a significant feature within the model generating the dataset, the neural network will also fail to learn that feature. For example, if the model behaves like a periodic function, then the neural network not understand its periodic behavior if the training dataset does not have enough data points indicating this behavior. In situations like this, if we are able to incorporate the known knowledge that the model behaves periodically into the neural network, then the network could be able to make much more accurate predictions without the training dataset having to show its periodic trait. In the paper [2] the authors introduced a set of techniques that can be used individually or in combination with each other to incorporated domain knowledge in neural networks. In our research, we seek to determine a viable way to incorporate such domain knowledge into a neural network that seeks to prediction data points generated by a parabola

3 Data generating model and datasets

To test the results and changes induced by the inclusion of domain knowledge in neural network models, we must first find a data generating model that contains a distinct trait that we can include in our neural network. For simplicity, we have chosen our data generating model to be the simple parabola $y = x^2$.

This data generating model not only has a very distinct symmetric trait, but also is easy to generate data points with and situations where the symmetric trait is not obvious within a set of training data can also be easily simulated.

To efficiently test the behaviors of our neural network model, we have decided to generated four different datasets that each represent an interval of significant interest. Our training dataset consists of 100 data points generated from the interval between $x = -4$ to $x = 2$, this interval is chosen as it contains a partial indication of the curvature behavior of the parabolic generating model. Our first testing dataset contains 20 data points inside the same interval as the training range. This set of data should be predicted with the lowest amount of accuracy, even without the inclusion of domain knowledge. Our second testing dataset contains 20 data points inside the interval between $x = 2$ to $x = 4$, which is the interval where the data points are symmetric with respect to the training data points between $x = -4$ to $x = -2$. We expect to see significant improvement in prediction accuracy with the inclusion of domain knowledge on the prediction for this dataset. Our last testing dataset contains 20 data points inside the interval between $x = 4$ to $x = 8$, since this interval is outside the symmetric range as our training dataset only starts from $x = -4$, we are uncertain on the behaviors of the neural network predictions with and without the inclusion of domain knowledge.

4 Neural Network without Domain Knowledge

We first created a neural network model with Mean Squared Error as the loss function to determine the base behavior of neural network model predictions on a dataset generated by the most common parabola: $y = x^2$.

Mean squared error loss is simply taking the square of the distance between the predicted data point and the actual data point, and using that as the loss for that particular prediction. We ran five trials with a standard neural network built by us and recorded the mean and standard deviation of the error resulting from prediction using MSE as our loss function.

Mean Squared Error

ReLU Activation Function and MSE Loss	Training Data	Dataset within Training Range	Dataset within Symmetry Range	Dataset outside Symmetry Range
Trial 1	0.1158	0.0702	8.1261	521.7676
Trial 2	0.0507	0.0504	15.1092	671.7506
Trial 3	0.0151	0.0181	12.6636	613.8778
Trial 4	0.0191	0.0137	6.5264	479.0822
Trial 5	0.6909	1.0834	88.8461	1417.0863
Mean	0.17832	0.24716	26.25428	740.7129
Standard Deviation	0.289363805615008	0.468054145372093	35.1584821962923	385.587728414508

Figure 1: A table indicating the recorded errors from the five trials with MSE as the loss function

As shown in figure 1, the neural network model without the inclusion of domain knowledge seems to behave well for the testing dataset within training range, having a mean error of 0.24716 and a standard deviation of approximately 0.468.

However, when we use the same neural network model to predict the testing dataset within symmetry range and outside symmetry range, there is a significant increase in the errors for both the testing datasets, which will be much more apparent later when we compare the results with the predictions generated by a neural network with a customized loss function to include domain knowledge.

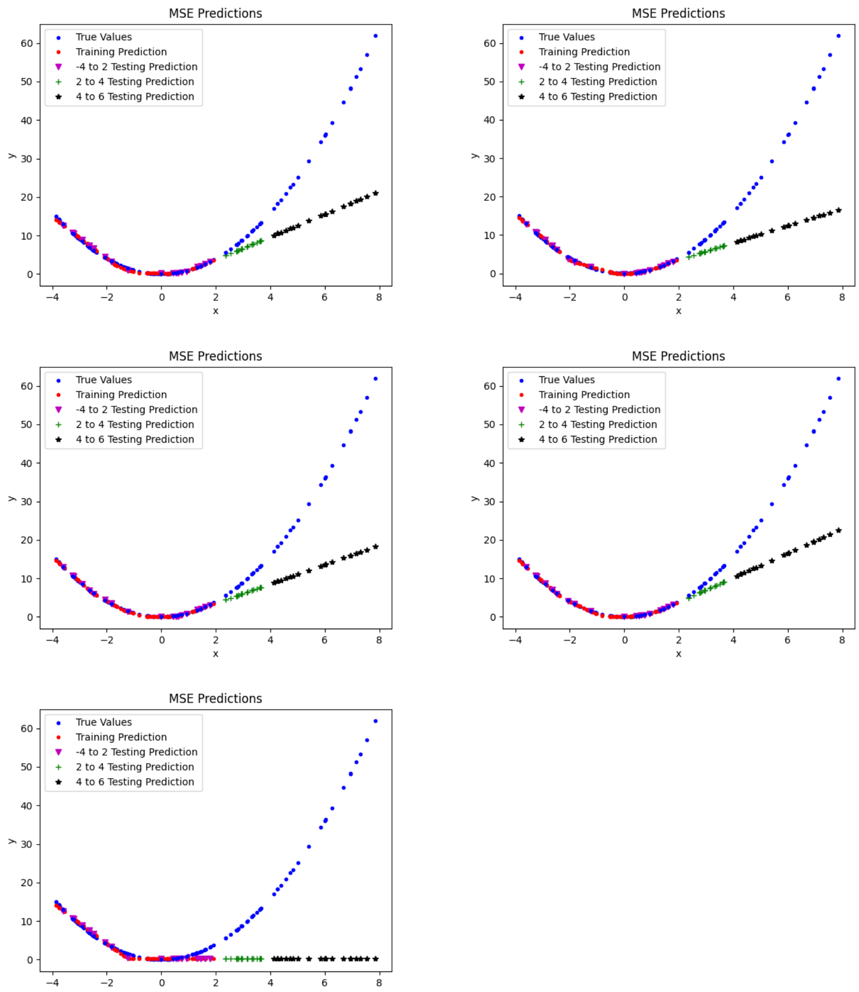

Figure 2: Behaviors of trials 1 to 5 predicted with a neural network model and MSE loss function

As indicated by the graphs of trials 1 to 5, we can clearly see that the predictions from the neural network with a MSE loss function behaves well for the testing dataset within the training interval, but starts to fall off in accuracy when the testing interval enters the symmetry range.

5 Neural Network with Domain Knowledge

5.1 Determining the loss function

Tirtharaj Dash, Sharad Chitlangia, AdityaAhuja and Ashwin Srinivasan introduced in [2] a set of techniques that can be used to include domain knowledge in deep neural networks. From their paper we have decided that for our neural network it would be best to customize a loss function tailor to include the domain knowledge that the data points should be symmetric with respect to the x-axis. Gaining insights from a similar idea employed by Arka Daw, Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar in [1] for lake temperature modeling, we apply that idea of adding a penalization based on each datapoint's deviation from the prediction of their reflection across the x-axis:

$$Loss(x, y) = (y_{true} - y_{pred})^2 + (pred(-x) - y_{pred})^2$$

With this loss function, we have the standard MSE loss plus an extra component. The second part of this loss function essentially takes the squared value of the difference between the predicted value of $-x$ and the predicted value of x. This means that if the difference is small (i.e between 0 and 1), then the square lowers the value and we penalize less as there is not much of a deviation. However if there is a huge deviation and the difference is large, we penalize it more as the square increases the value of a large amount

5.2 Prediction Behavior

To test the prediction behavior of the neural network with our customized loss function that aims to include domain knowledge by the symmetric trait by penalizing deviations from symmetry, we simulated 10 trials with the same training and testing datasets mentioned in section 3.

Mean Squared Error

ReLU Activation Function and Symmetry Loss	Training Data	Dataset within Training Range	Dataset within Symmetry Range	Dataset outside Symmetry Range
Trial 1	0.0216	0.0177	0.0623	151.4078
Trial 2	0.2179	0.1417	0.2832	215.6361
Trial 3	0.5009	0.3910	0.4705	268.8645
Trial 4	0.1464	0.1046	0.2731	217.9301
Trial 5	0.2165	0.1458	0.2770	216.7129
Trial 6	0.3477	0.3059	0.3813	255.1532
Trial 7	0.1503	0.1640	0.3314	230.1162
Trial 8	0.2617	0.1992	0.3307	233.6589
Trial 9	0.4056	0.3330	0.4161	256.3202
Trial 10	0.0873	0.0515	0.1642	189.3226
Mean	0.23559	0.18544	0.29898	223.51225
Standard Deviation	0.14766005590921	0.122603581604381	0.119049464976165	34.6222691932118

Figure 3: Behaviors of trials 1 to 10 predicted with a neural network model and custom loss function for domain knowledge inclusion

As shown in figure 3, we see that the results for prediction of the data points within the training range does not see much change as it has already been very accurate with the MSE loss neural network. However, we can clearly see that the prediction for the data points within the symmetry range ($x = 2$ to $x = 4$ has seen a drastic improvement in terms of accuracy. Referring back to figure 1, the mean and standard deviation of the error for dataset within symmetry range predictions were approximately 26.25 and 35.16, where as the neural network predictions with domain inclusion for the same dataset had a mean of approximately 0.299 and a standard deviation of approximately 0.119

Another huge improvement we can see is for the predictions of the dataset outside symmetry range. Referring back to figure 1, the mean and standard deviation of the error for dataset outside symmetry range predictions were approximately 740.71 and 380.59, where as the neural network predictions with domain inclusion for the same dataset had a mean of approximately 223.51 and a standard deviation of approximately 34.62

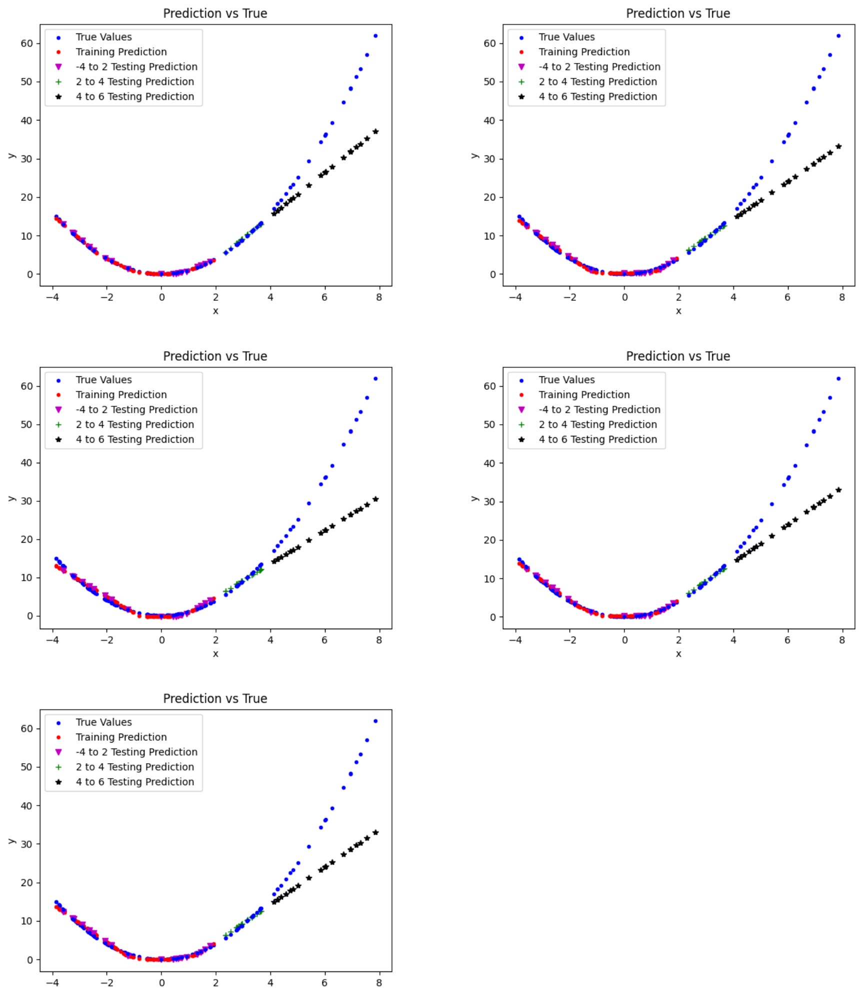

Figure 4: Behaviors of trials 1 to 5 predicted with a neural network model and customized loss function for domain knowledge inclusion

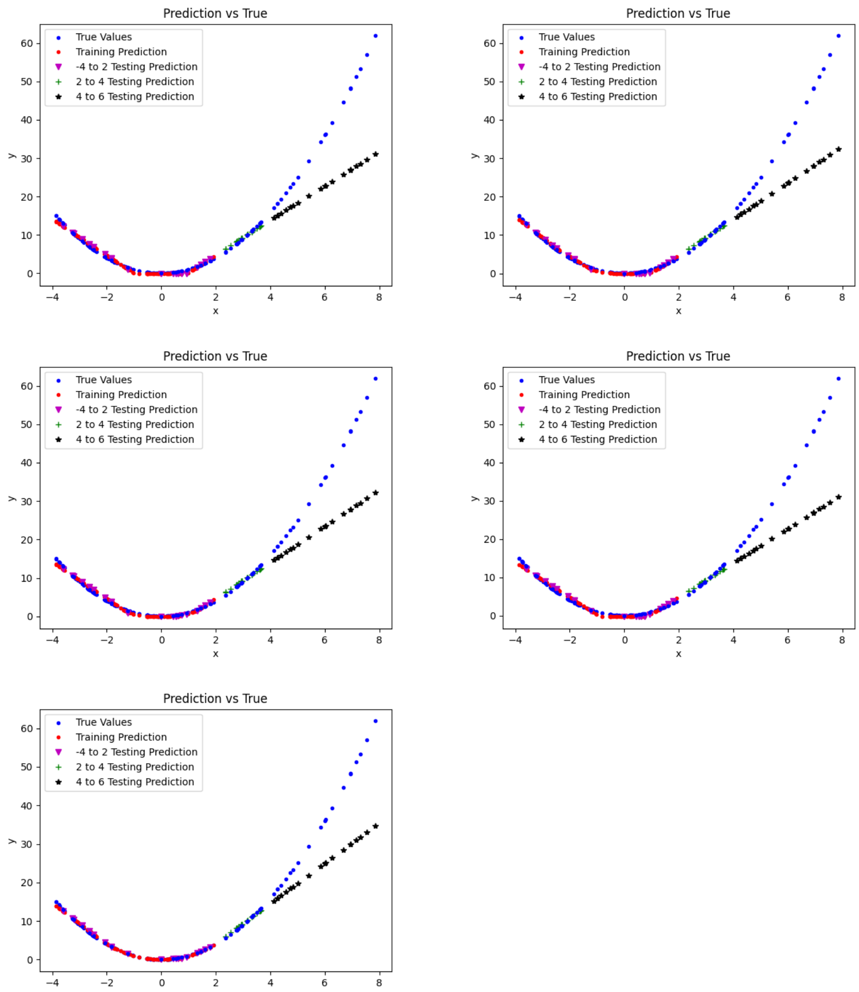

Figure 5: Behaviors of trials 5 to 10 predicted with a neural network model and customized loss function for domain knowledge inclusion

Comparing the trials with the customized loss function in figure 4 and figure 5 with the trials with MSE loss function in figure 2, we can see a clear improvement in the fit between the prediction values and the true values starting from $x = 2$. This indicates that the customized loss function that we have created for domain knowledge inclusion has improved the overall accuracy of our model prediction, especially for data points in the intervals that follow the traits in which we tried to include in our neural network.

6 Future Research

In reality, datasets often do not come as being perfectly aligned with the model. The most common issue with datasets is that noise exists. Our next goal is to determine a way to deal with noise that comes with the datasets that we have domain knowledge on. If we are able to prove that domain knowledge inclusion can improve models that try to value predict with a noisy training dataset, then it can be extended to use for many scientific fields with patterned data that are not always behaving exactly like the model suggests

References

[1] A. Daw, A. Karpatne, W. Watkins, J. Read, V. Kumar, Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling.

[2] T. Dash, S. Chitlangia, A.Ahuja, et al. A review of some techniques for inclusion of domain-knowledge into deep neural networks. Sci Rep 12, 1040 (2022). https://doi.org/10.1038/s41598-021-04590-0