UNIVERSITY OF CAMBRIDGE

Discovering Symbolic Models from Deep Learning with Inductive Biases - Reproduction Report

Executive Summary

Vishal Jain

Department of Physics

MPhil in Data Intensive Science

Supervisor: Dr Miles Cranmer

A dissertation presented for the degree of
Master of Philosophy

University of Cambridge
July 2, 2024

1 Introduction

The aim of this report is to reproduce the key results of the paper *Discovering Symbolic Models from Deep Learning with Inductive Biases* by *Cranmer et al.* [2]. The main contributions of the paper are two fold - first the presentation of a systematic framework whereby inductive biases can be employed to distill low dimensional symbolic approximations of learned neural networks. The second is showing its successful application in 3 contexts: rediscovering force laws, rediscovering Hamiltonians and discovering a new equation for a nontrivial cosmology problem. The scope of this report is limited to the reproducibility of the experiments relating to the rediscovery of force laws. However, in doing so, the wider contribution of the framework is validated.

2 Motivation

Why would it be desirable to approximate a high dimensional neural network with a low dimensional symbolic model? The answer lies in the scientific method itself and the unreasonable effectiveness of mathematics in the natural sciences. When using neural networks as a tool for science, the lack of interpretability is a major drawback. Science is not just about making predictions, but also about understanding the underlying mechanisms that govern the phenomena being studied. Neural networks are famously known to be black boxes, with the decision-making process being opaque to the user. If they are to be used for science, there needs to be some distillation of what the network has learned in a form that is interpretable to humans. Symbolic models are a natural choice for this task, as they provide a compact and interpretable representation of the underlying mechanisms due to their low dimensionality. Further, they are likely to generalise better to out of distribution data than the neural network itself, indeed this was the shown to be the case in the findings presented by the original paper. Choosing to describe the world in terms of closed form low dimensional mathematical equations seems to be a very good inductive bias.

3 Symbolic Distillation Framework

The general framework prescribed by the paper for symbolic distillation is as follows:

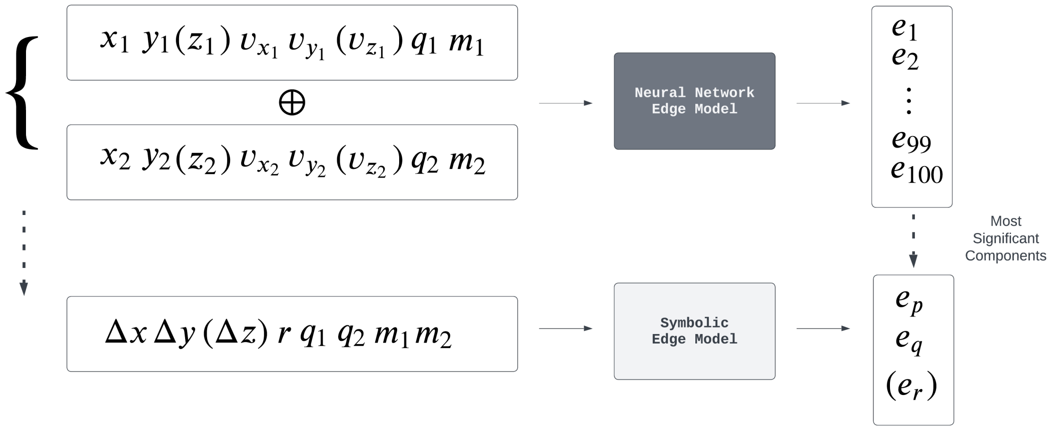
- Design a neural network with an architecture that has a seperable internal structure and an inductive bias that is well suited to modelling the underlying problem.
- Train the model with regularisation techniques that encourage the network to learn a low-dimensional representation of the data.
- Replace a learned component of the neural network with a symbolic model that approximates that component's output.
- Retrain the neural network and repeat the process until all components of the neural network have been replaced with low dimensional symbolic models.

4 Method

To perform the symbolic distilation, a technique known as symbolic regression is employed. Symbolic regression is a type of regression analysis that searches the space of mathematical expressions to find the best fitting equation to the data. The symbolic regression is performed using the open-source library `PySR` [1]. This work involves using a type of neural network architecture known as a graph neural network (GNN) to predict the instantaneous accelerations of particles simulated under a variety of force laws in a 2D or 3D space. The GNN operates through a scheme of message passing, involving two main learned components: the edge model (ϕ^e) and the node model (ϕ^v).

Edge Model (ϕ^e):

- The edge model is applied to each edge k in the graph.
- It takes as input the feature vectors of the two nodes connected by the edge, $\mathbf{x}_i, \mathbf{x}_j$, which contain their positions, velocities, masses, and charges, and outputs a 100 dimensional edge message $\mathbf{e}_k$.
- The edge model is distiled by recording its inputs, the concatenated feature vectors and the output, the edge message. For symbolic distilation, the outputs are the d most significant components of the edge message. Where significance is defined by the standard deviation of the component across the sampled edge messages and d refers to the problem's dimensionality (2D or 3D).



$x_1\ y_1 (z_1)\ v_{x_1}\ v_{y_1}\ (v_{z_1})\ q_1\ m_1$
$\oplus$
$x_2\ y_2 (z_2)\ v_{x_2}\ v_{y_2}\ (v_{z_2})\ q_2\ m_2$
Neural Network Edge Model
e_1 e_2 $\vdots$ e_{99} e_{100}
Most Significant Components
$\Delta x\ \Delta y\ (\Delta z)\ r\ q_1\ q_2\ m_1 m_2$
Symbolic Edge Model
e_p e_q (e_r)

Figure 1: Symbolic distillation of the edge model.

Node Model (ϕ^v):

- Once every edge has an associated edge message, the node model is applied to each node in the graph.
- It takes as input the feature vector of the node, $\mathbf{x_i}$, and the aggregated edge message $\bar{\mathbf{e_k}}$, where aggregation is performed by summing over all edge messages associated with inbound edges.
- The output of the node model is the instantaneous acceleration of the node $\mathbf{a}$.
- The node model is distiled by recording the input node's feature vectors along with the aggregated edge messages. For symbolic distillation, only the d most significant components of the aggregated edge message are used.

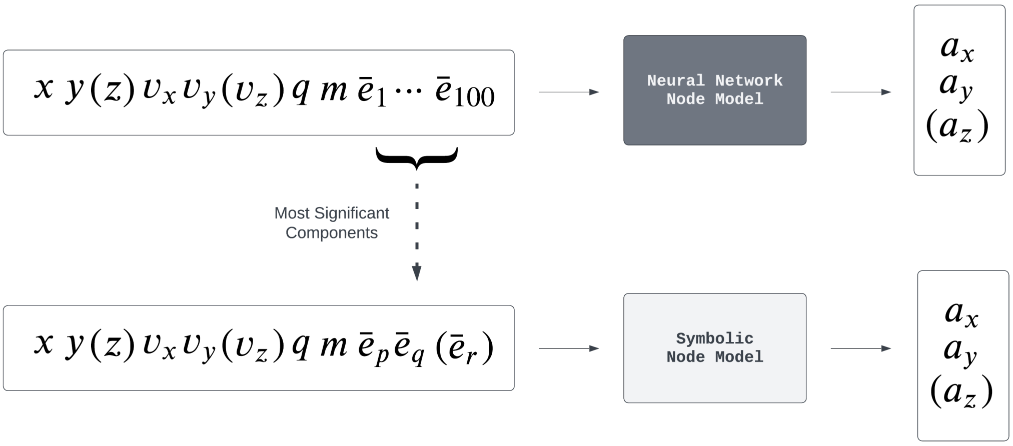

$x\ y(z)\ v_x v_y (v_z)\ q\ m\ \bar{e}_1 \cdots \bar{e}_{100}$
Neural Network Node Model
a_x
a_y
(a_z)
Most Significant Components
$x\ y(z)\ v_x v_y (v_z)\ q\ m\ \bar{e}_p \bar{e}_q\ (\bar{e}_r)$
Symbolic Node Model
a_x
a_y
(a_z)

Figure 2: Symbolic distillation of the node model.

The action of the edge function can be interpreted as the application of the force law and the node function as the application of Newton's second law. The goal will be to recover the underlying force law by performing symbolic regression on the edge model. Figure 3 provides a visual summary of the graph neural network architecture, illustrating how it can model the dynamics of particles using seperable components that can be distiled into symbolic models.

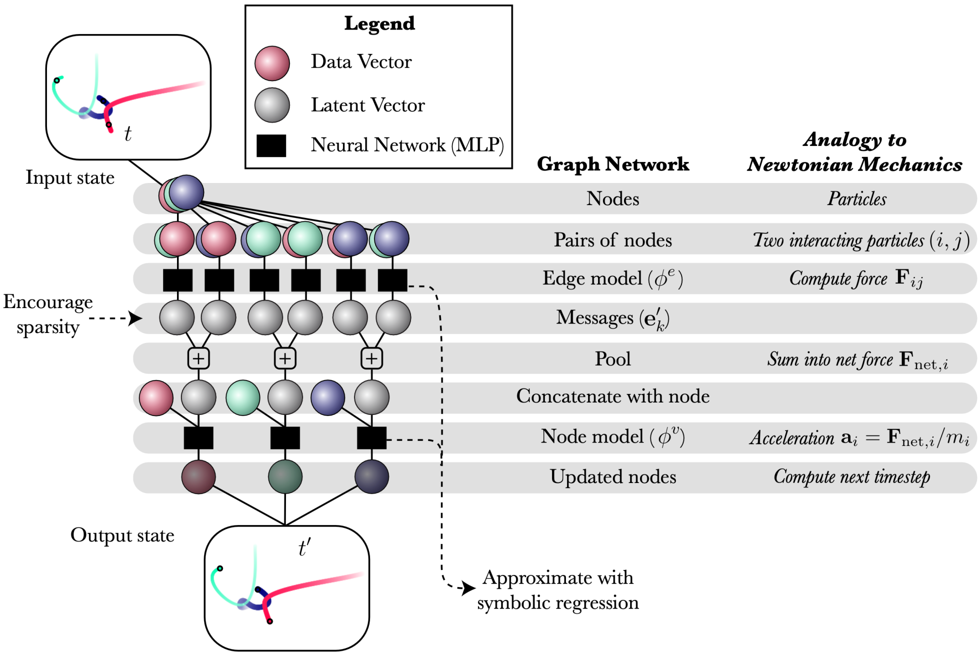

Legend
Data Vector
Latent Vector
Neural Network (MLP)
t
Input state
Graph Network
Analogy to Newtonian Mechanics
Nodes
Particles
Pairs of nodes
Two interacting particles (i, j)
Edge model (ϕ^e)
Compute force $\mathbf{F}_{ij}$
Encourage sparsity
Messages ($\mathbf{e}'_k$)
Pool
Sum into net force $\mathbf{F}_{\text{net},i}$
Concatenate with node
Node model (ϕ^v)
Acceleration $\mathbf{a}_i = \mathbf{F}_{\text{net},i}/m_i$
Updated nodes
Compute next timestep
Output state
t'
Approximate with symbolic regression

Figure 3: Computational graph of the graph neural network architecture. Figure taken from the original paper [2].

This process is repeated for each force law using 4 different training strategies to investigate how to best train the neural network to facilitate symbolic distillation. The four training strategies are: standard, bottleneck, L$_1$, and KL. The standard strategy uses a 100-dimensional edge message with a loss measured by mean absolute error, serving as the baseline. The KL strategy samples edge messages from a 100-dimensional normal distribution, adding a Kullback-

Leibler divergence penalty to the loss. The L_1 strategy adds the L1 norm of the edge message as a penalty to the standard loss. Finally, the bottleneck strategy reduces edge message dimensions to align with the problem's dimensionality (2 or 3).

5 Results

Sim.	Standard	Bottleneck	L1	KL
Charge-3	×	✓	×	×
Charge-2	×	✓	×	×
r^{-2}-3	×	✓	✓	×
r^{-2}-2	×	✓	✓	×
r^{-1}-3	×	✓	✓	✓
r^{-1}-2	×	✓	✓	✓
Spring-3	×	✓	✓	✓
Spring-2	×	✓	✓	✓

Table 1: Original

Sim.	Standard	Bottleneck	L1	KL
Charge-3	✓	✓	✓	×
Charge-2	×	✓	×	×
r^{-2}-3	✓	✓	✓	×
r^{-2}-2	×	✓	×	×
r^{-1}-3	✓	✓	✓	✓
r^{-1}-2	✓	✓	✓	✓
Spring-3	✓	✓	✓	×
Spring-2	✓	✓	✓	×

Table 2: Reproduced

Table 3: Comparison of the symbolic distillation results for the rediscovery of various force laws under different training strategies.

.

6 Conclusion

This reproduction study was able to reproduce most of the force laws shown in the original paper and also found that one of the training strategies, namely the standard strategy, was more successful than originally reported. Future work would be to repeat the analysis for different random seeds. This would help give an understanding of what is natural variation in the results due to the stochastic nature of the training process versus what is a systematic discrepancy between the reproduced study and the original findings.

References

[1] Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression.jl, 2023.

[2] Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases, 2020.