Project Report

SGTB Khalsa College, University of Delhi

Team No. : 18

Pawanpreet Kaur(2020PHY1092)

Preetpal Singh(2020PHY1140)

Anjali(2020PHY1164)

Paper Title: Computational Physics

Submitted to: Dr Mamta and Dr Ramo Chote

B.Sc(H) Physics Sem III

Aim

Simulate charged particles to generate large amount of data for their dynamical states(position, force) and static attributes(charge) with Graph Neural Networks(GNNs).

Finding physics symbolic expressions from generated simulated data with machine learning technique Symbolic Regression.

Motivation

As we know that Machine Learning(ML) is an emerging field and its hard for physicists to survive without ML in future. In our project we're using state of the art GNNs to simulate data for physics problem. Here, we're using simplest model but these simulations provenly extended to complex physics problem for which experimental setup is quite expensive. In 2nd part of project deriving symbolic expressions from experimental results do the same job for which physicists are known.

What we already done

For 2nd part of project, We have setup Programming setup of base AI-Feynman Library and derive following equations

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}}$$

equations for several example data sets in the form of a table of numbers, whose rows are of the form $\{x_1, x_2, \ldots, y\}$, where $y = f(x_1, x_2, \ldots)$ as numerical value.

Integration of Physics and Programming

In this project, we are aimed to produce a large data set of charged particles, their position and force between these particles with GNNs. Technically, GNNs are made of nodes and edges in which nodes represent statistical properties(chrage) and edges dynamical properties(force, distance). So, we will programme charge particles, required properties as nodes and edges and simulate them using neural network to generate data.

Then we will figure out how physics principles(Dimensional Analysis, Symmetry, multiplicative separability etc.) as described in figure [1] implemented at different levels for our generated data in process of deriving equation of electrical force which is expected result as a flow chart given in figure [2] for gravitational law of forces.

The expected final result is equation of Electrical Force or some other equations which give similar results.

Team Contribution

As it is quite expected that the student who come with idea should have strong intuition than others. So, Preetpal will be leading initially to find resources, programming, having active presence in base community. However, Once the intuition is build of how to approach things then all will equally contribute in programming, results analysis. Pawanpreet and Anjali will be leading in final report writing to balance initial contribution of Preetpal. On the other hand, there will be flexibility to constructively intervene each other's work to utilize time, energy and efforts for collective purposes.

Technical Overview of project

Part I

Our project is based on these [5] [1], [2], [4] research papers. Our model will takes graphs as input, performs object- and relation-centric reasoning in a way that is analogous to a simulation, and is implemented using deep neural networks(GNNs). We can use GNNs to various physical domains: n-body problems, rigid-body collision, and non-rigid dynamics. Previous results show they can be trained to accurately simulate the physical trajectories.

The behavior of our model will be similar in spirit to a physical simulation engine which will generates sequences of states by repeatedly applying rules that approximate the effects of physical interactions and dynamics on objects over time. The interaction rules are relation-centric, operating on two or more objects that are interacting, and the dynamics rules are object-centric, operating on individual objects and the aggregated effects of the interactions they participate in.

Here we will focus on binary relations, which means one interaction term per relation, but another option is to have the interactions correspond to n-th order relations by combining n senders in each function. These possibilities are beyond the scope of this work, but are interesting future directions.

We will train, validate and simulate test sets and add noise to some of the data's input positions. We haven't specified above parameters yet to perform simulation.

Part II

Generic functions $f(x_1, \ldots, x_n)$ are extremely complicated and symbolic regression to discover. However, functions appearing in physics and other scientific applications often have some of the following simplifying properties to make them them easier to discover[3]:

1. Units: f and the variables upon which it depends have known physical units.
2. Low-order polynomial: f is a polynomial of low degree.
3. Compositionality: f is a composition of a small set of elementary functions, each typically taking no more than two arguments.
4. Smoothness: f is continuous and analytic in its domain. It will enables approximating f using a feed-forward neural network with a smooth activation function.
5. Symmetry: f exhibits translational, rotational, or scaling symmetry with respect to some of its variables.
6. Separability: f can be written as a sum or product of two parts with no variables in common. It enables the independent variables to be partiotioned into two disjoint sets and the problem to be transformed into two simpler ones, each involving the variables from one of these sets.

Data that will be used

1. Data table: A table of numbers generated by GNNs, whose rows are of the form $\{x_1, x_2, \ldots, y\}$, where $y = f(x_1, x_2, \ldots)$; the challenge is to discover the correct analytic expression for the mystery function f.

2. Unit table: A table specifying the physical units of the input and output variables as 6D vectors.

3. Equation: The analytic expression for the mystery function f, for answer checking.

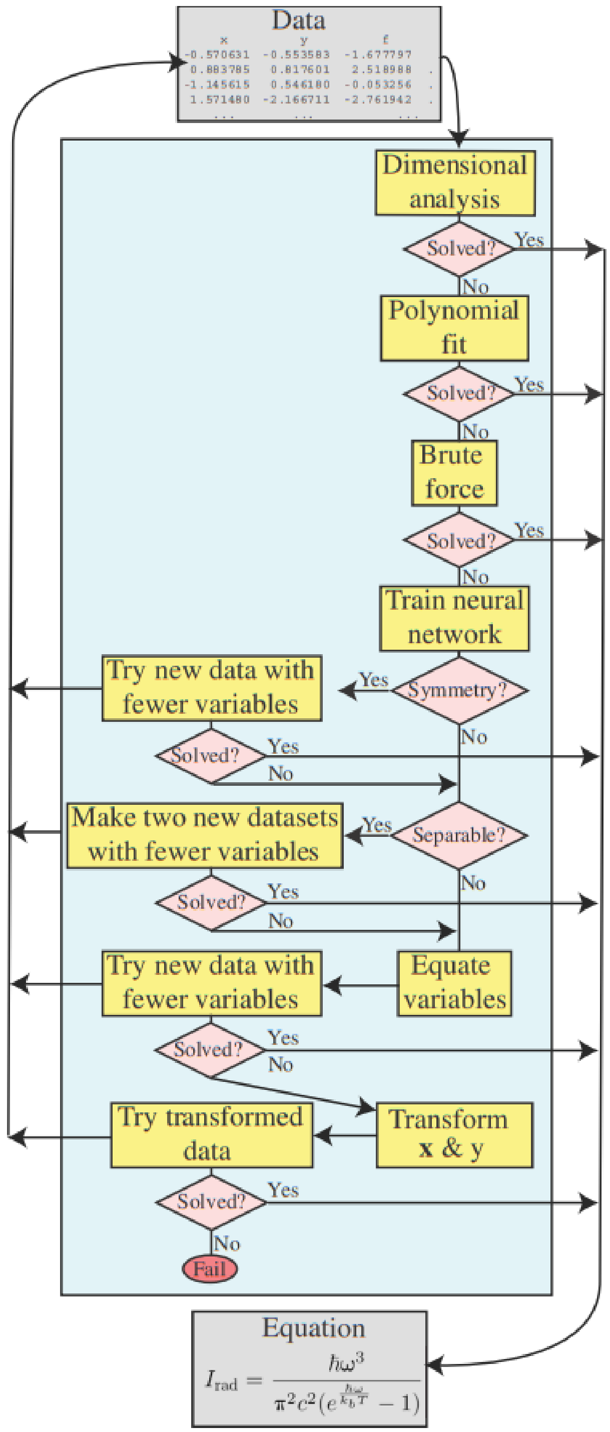

Figure 1: **Schematic illustration of AI Feynman algorithm:** It is iterative How this algorithm approach the given dataset to extract analytic expressions from experimental/simulated results based on physics principles.

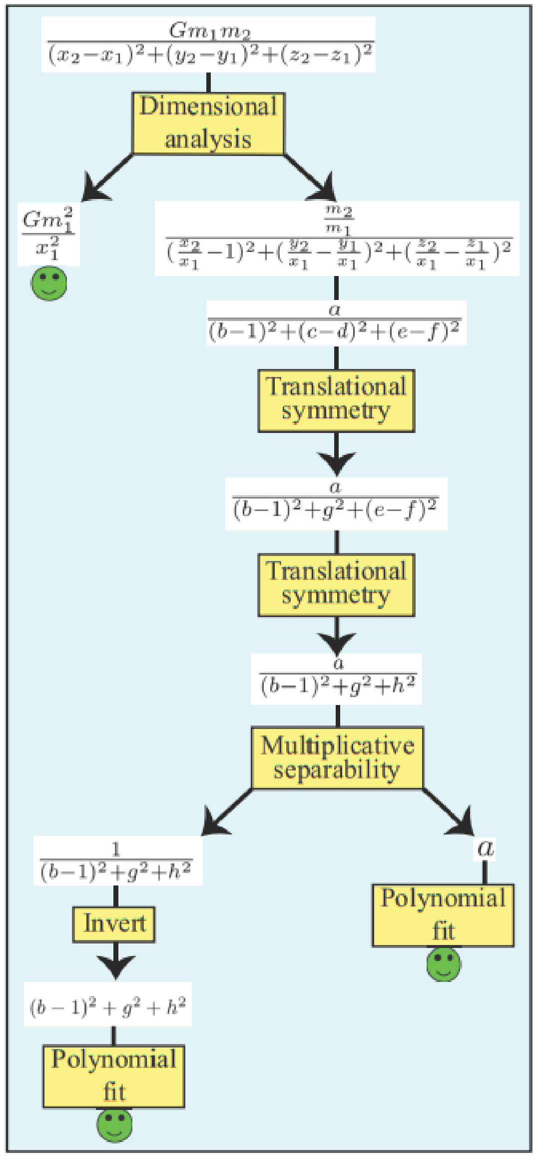

Figure 2: **How AI Feynman algorithm discovered mystery Equation:** Given a mystery table with many examples of the gravitational force F together with the nine independent variables $G, m_1, m_2, x_1, \ldots, z_2$, this table was recursively transformed into simpler ones until the correct equation was found. First, dimensional analysis generated a table of six dimensionless independent variables $a = m_2/m_1 \ldots i\ f = z_1/x_1$ and the dimensionless dependent variable $\mathcal{F} \equiv F \div Gm_1^2/x_1^2$. Then, a neural network was trained to fit this function, which revealed two translational symmetries (each eliminating one variable, by defining $\boldsymbol{g} \equiv \boldsymbol{c} - \boldsymbol{d}$ and $h \equiv \boldsymbol{e} - \boldsymbol{f}$) as well as multiplicative separability, enabling the factorization $F(a, b, g, h) = G(a)H(b, g, h)$, thus splitting the problem into two simpler ones. Both G and H then were solved by polynomial fitting, the latter after applying one of a series of simple transformations (in this case, inversion). For many other mysteries, the final step was instead solved using brute-force symbolic search as described in the text.

Team Members

Partner A
Name: Pawanpreet Kaur
Roll No. : 2020PHY1092
Partner B
Name: Preetpal Singh
Roll No. : 2020PHY1140
Partner C
Name: Anjali
Roll No. : 2020PHY1164

List of Programming tools will be used

- Python, Fortran (Programming Language)
- Gnuplot(Plotting)
- Latex(Final Report Writing)

References

[1] *[1612.00222] Interaction Networks for Learning about Objects, Relations and Physics.* URL: https://arxiv.org/abs/1612.00222 (visited on 09/17/2021).

[2] *[2006.11287] Discovering Symbolic Models from Deep Learning with Inductive Biases.* URL: https://arxiv.org/abs/2006.11287 (visited on 09/17/2021).

[3] *AI Feynman: A physics-inspired method for symbolic regression.* URL: https://www.science.org/doi/10.1126/sciadv.aay2631 (visited on 09/17/2021).

[4] *Graph Neural Networks in Particle Physics – arXiv Vanity.* URL: https://www.arxiv-vanity.com/papers/2007.13681/ (visited on 09/17/2021).

[5] Alvaro Sanchez-Gonzalez et al. "Learning to Simulate Complex Physics with Graph Networks". In: *arXiv:2002.09405 [physics, stat]* (Sept. 2020). arXiv: 2002.09405. URL: http://arxiv.org/abs/2002.09405 (visited on 09/17/2021).