Discovering the Laws of Physics with Artificial Intelligence

Iwona Kotlarska
Andrej Jakovljević
Marijana Davitkova

Mentor: Pedro Zuidberg Dos Martires

Abstract

Artificial intelligence is a new and revolutionary aspect of computer science that is being rapidly developed and implemented in the real world. It is the ability of a computer or a machine to mimic human behavior and intelligence through a certain algorithm. The way artificial intelligence works is, it upgrades itself based on previously obtained data and information using various methods and approaches, such as statistics, probability theory, clustering, and networks. The goal of our project is to find mathematical laws of physics that describe a physical process. The way our project functions is similar to evolution. Firstly, in order to describe a physical process, a population of formulas is randomly produced, which are ranked and compared and reduced to the better portion. That portion is mutated: new and better formulas are obtained. However, in order for all this to function, it requires a set of data, according to which the algorithm should sync its work. The method deployed in our project is a supervised learning method, namely *symbolic regression*. Our project produced the wanted answer, and even exceeded our expectations. It could be improved if it were run on a faster and better computer. Furthermore, this project opens the doors to further extend the application of the developed algorithm to fields of science other than physics.

1 Introduction

Ever since the earliest days of humankind, people have been trying to observe nature around them. They had searched for patterns and laws to explain the processes occurring daily, and thus science was born. At first, science was done with the help of human senses, sight, touch, smell, hearing, taste. Later on, tools were invented to ease the work and yield greater precision, and people used theories, experiments and simulations to satisfy their thirst for knowledge. Nevertheless, as the pace of technological advancement has taken up, more and more complex challenges have arisen, and the need for a new mean of doing science has appeared. That need is met by artificial intelligence.

What we wanted to achieve with this project is to recreate a simpler example of artificial intelligence, a program meant to rediscover the laws of physics, more specifically the basic equations of motion, from a set of data provided to it.

Artificial intelligence Artificial intelligence (AI) is an area of computer science that deals with giving machines the ability to seem like they have human intelligence, or put more simply, the power of a machine to copy intelligent human behavior. It is the ability of a machine to upgrade its knowledge on its own.

Symbolic regression Symbolic regression is an algorithm used in the programming of artificial intelligence that is inspired by natural selection in evolution. The computer is given a population of functions, which are compared, ranked and reduced, something like survival of the fittest. The remaining functions are crossed over and mutated. This process is repeated until the wanted formula is discovered.

2 Methodology

In order to see our project through, we had to split the whole procedure in pieces, and tackle every problem independently. For faster and simpler conduct, we split the work between us.

2.1 Data

We obtained the data to fuel our algorithm from two different sources

- **Simularions:** We wrote computational simulations of physical systems that reproduce data obtained in actual physical experiments.
- **Experiments:** We build a simple pendulum experiment and recorded the data by filming the experiment

In our project, we fed data to the computer that was obtained by our computer simulation. The simulation produces data that corresponds to data obtainable in the real world. We had two kinds of simulations, with and without an actual physical experiment. For the one with physical experimentation, we made a pendulum, which we recorded and processed. The other simulation was with generating random data by a program which feeds it to the main algorithm.

2.2 Algorithm

After we had obtained the data, we needed an algorithm to process the data. We found the Symbolic Regression [3] to be a neat solution, and we implemented it, so that it develops our equations into clean formulas fitting the previously obtained data. The algorithm was split into several functions that fulfilled their purpose and gave us the desired result.

3 Symbolic Regression

Symbolic regression is our algorithm, which does all the work in finding the proper equation. As we have mentioned before, it can be compared to evolution, as it functions similarly to it. Symbolic regression is made up of 7 steps.

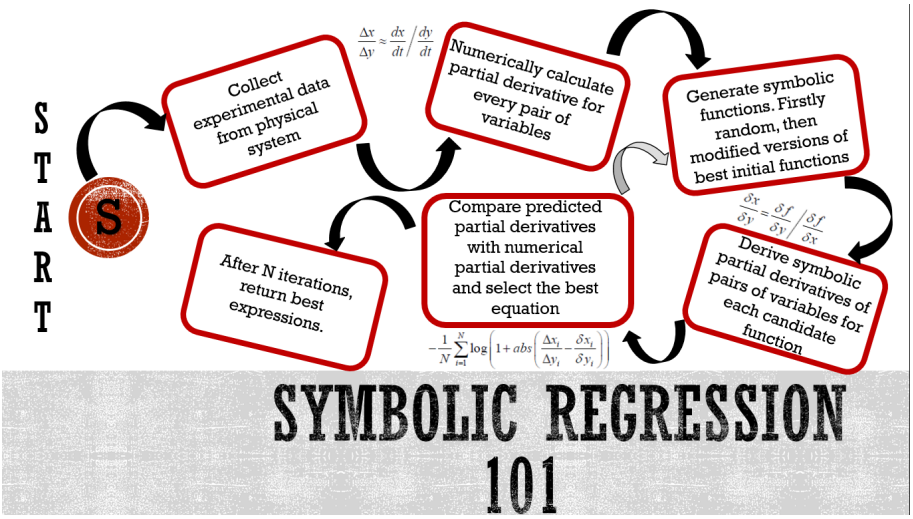

Figure 1: Diagrammatic representation of the steps in symbolic regression.

3.1 Collecting data

The algorithm references the data produced in the simulations/experiments.

3.2 Calculating the derivatives

We numerically calculate partial derivative for every pair of variables, in order to use them in the creation of combinations during the following stage, the generating of the symbolic functions.

$$\frac{\Delta x}{\Delta y} \approx \frac{dx}{dt} \Big/ \frac{dy}{dt} \tag{1}$$

where t is time, and x and y arbitrary variables.

3.3 Generating symbolic functions

Since the algorithm has to start from somewhere, it initially guesses the equations, therefore, in the first generation, the functions are random, and create the population of equations. As the algorithm loops, and new generations are formed, the newly generated functions are not random, but modified versions of the best initial functions. The

latter process is made possible by mutation and crossing over. Below we show different examples of functions from the population:

$$15 \times x(t) + 75 \times v(t)^3 + 0.45 \times a(t) - 12.85 = 0 \tag{2}$$

$$3.45 \times sin(x(t)) + a(t)^2 = 0 \tag{3}$$

$$45 \times x(t) + 12.54 \times sin^2(v(t)) + 0.8 \times a(t) = 0 \tag{4}$$

3.4 Deriving symbolic partial derivatives

In order to compare the functions to one another, we calculate the symbolic derivatives of pairs of variables for each candidate function.

$$\frac{\delta x}{\delta y} = \frac{\delta f}{\delta y} \Big/ \frac{\delta f}{\delta x} \tag{5}$$

3.5 Comparing of functions

Next, the algorithm compares the predicted partial derivatives with the numerical partial derivatives, and selects the best equations using a certain cost function C.

$$C = -\frac{1}{N} \sum_i log \left[1 + abs \left(\frac{\Delta x_i}{\Delta y_i} - \frac{\delta x_i}{\delta y_i} \right) \right] \tag{6}$$

Furthermore, we added additional penalties for lack of variables, zero derivatives, and the size of the function. The final C function contained those as well:

$$\begin{aligned} C = & -\frac{1}{N} \sum_i log \left[ ]1 + abs \left(\frac{\Delta x_i}{\Delta y_i} - \frac{\delta x_i}{\delta y_i} \right) \right] \\ & + \text{penalty(zero derivatives)} \\ & + \text{penalty(lack of variables)} \\ & + \text{penalty(size)} \end{aligned} \tag{7}$$

After the comparison, we reduced the function population by half using natural selection, i.e. only the better half according to the C function remains.

3.6 Creating new generations

This is the part of the algorithm that loops between the creation and comparing of the functions, going back to the third step of symbolic regression: generating symbolic functions. Since it already has initial functions, the algorithm modifies them through mutation (changing of one variable into another), or crossing over (combining elements of two functions).

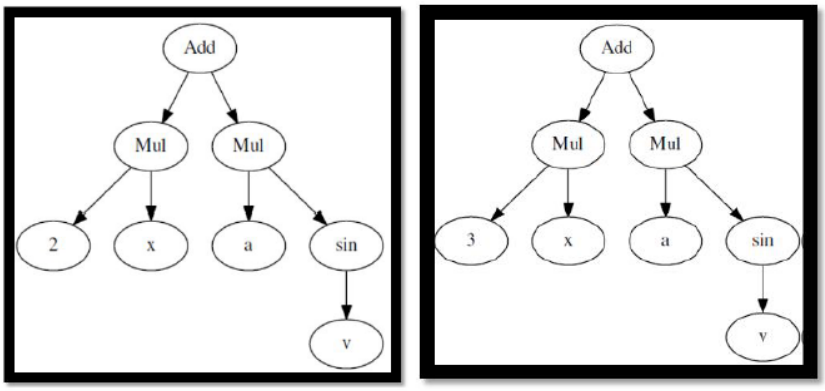

Figure 2: Process of mutation

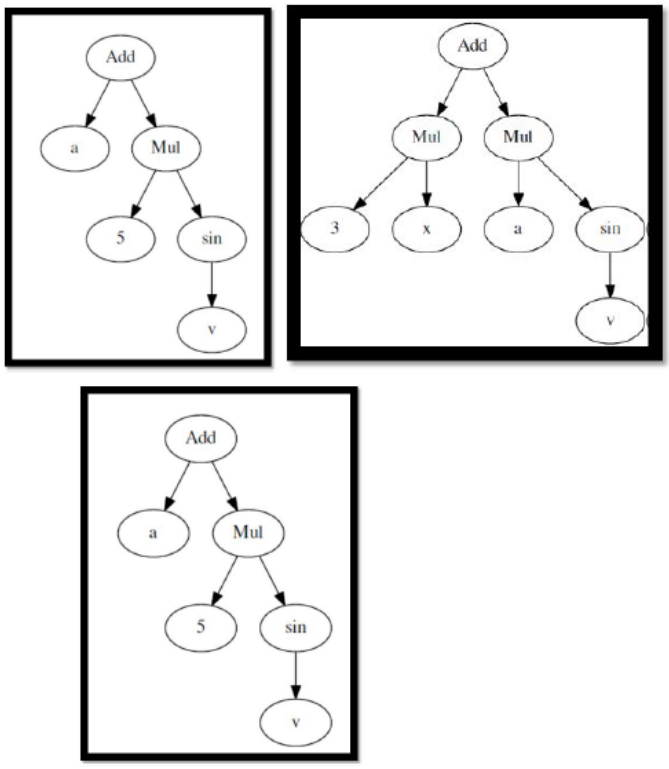

Figure 3: Process of crossing over

3.7 Result

After N iterations, the algorithm breaks out of the loop, and prints out its ”best” solution, again according to the C function. The accuracy of the entire code depends on the number of iterations, N.

4 Simulation

4.1 Deriving the equation of motion for the pendulum

In order to compare the functions, we needed their derivatives. Firstly, we have to observe Newton's second law of motion.

$$F = m \times a \tag{8}$$

Next, we calculate the acceleration due to gravity, which will be a function of θ.

$$a = -g \times sin\theta \tag{9}$$

Furthermore, we calculate the arc length of the pendulum.

$$arcL = L\theta \tag{10}$$

The latter redefines the acceleration formula.

$$\frac{d^2\theta}{dt^2} \times L = a \tag{11}$$

When we combine the acceleration formulas, we get the simple harmonic motion of a pendulum.

$$\frac{d^2\theta}{dt^2} + \frac{gsin\theta}{L} = 0 \tag{12}$$

Afterwards, we solve and integrate the previous equation, and take its root, which gives us our derivative.

$$T = 4\sqrt{\frac{L}{g}}\frac{1}{\sqrt{2}}\int_0^{\theta_0} \frac{1}{cos\theta - cos\theta_0} d\theta \tag{13}$$

Since this equation cannot be solved in terms of elementary functions, we used the small angle approximation.

$$\theta \approx sin\theta \tag{14}$$

We generated a data sample using the Sympy library in Python, which simulates a pendulum with a small-angle approximation by the following equations

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} \times \theta \tag{15}$$

Where

- θ - angle of the pendulum
- g - gravitational acceleration on Earth
- l - length of the string of the pendulum
- t - time

4.2 Solving differential equations

We solved the equation of motion for the pendulum in two ways: 1) numerically , using the Euler method and without the small angle approximation 2) symbolically and with the small angle approximation. The output of the algorithm for both sets of data can be compared (figure 4).

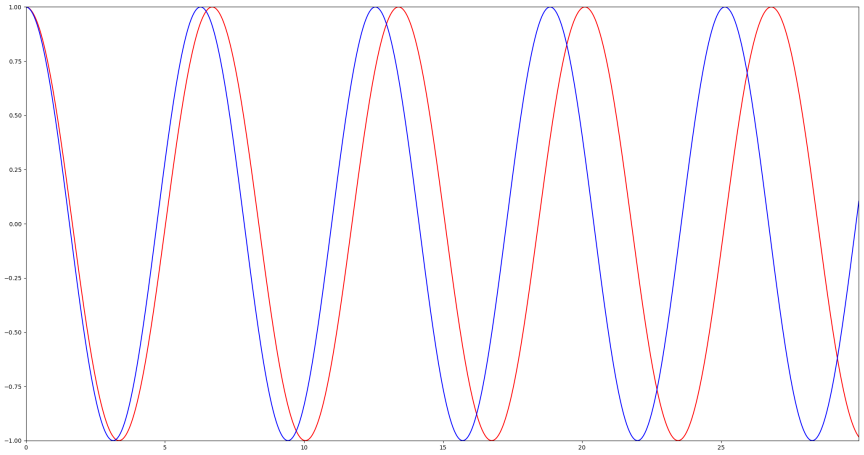

Figure 4: Plot of angle over time (blue line - without a small angle approximation, red line - with small angle approximation)

5 Experiment

An experiment was done with a pendulum to see if the algorithm can discover the equation of motion of the pendulum even in real-life situations, with friction and human errors. The setup of the experiment is shown in figure 5. The ball was hung from the wall covered in black nylon and painted in white, so that a computer visions program could easily recognize it.

Figure 5: Picture which shows the setup of the experiment

After the pendulum was recorded with a mobile phone camera, the footage was transfered onto a computer and cut into frames using Matlab [2], with the exact position of the ball on each frame was found using image processing library sp2 in Python to find white circles in black background (figure 6)

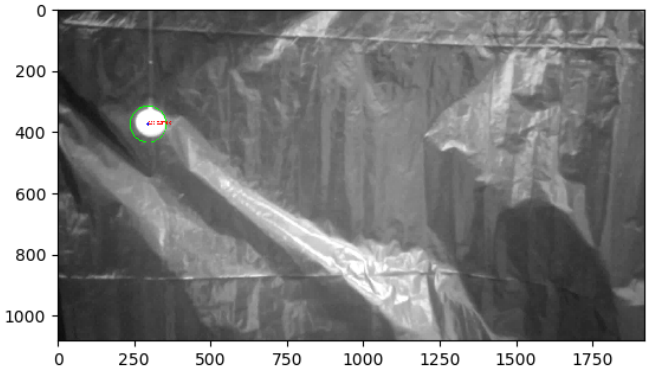

Figure 6: Output of the program that finds the ball and its center

Data of the position of the pendulum was then processed in Matlab. We plotted

the data and fit it using cubic interpolation [1]. Then, the first and second derivatives of the position were calculated numerically (representing velocity and acceleration, respectively, figure 7), using the following equations

$$v_i = \frac{x_{i+1} - x_i}{\Delta t} \tag{16}$$

$$a_i = \frac{v_{i+1} - v_i}{\Delta t} \tag{17}$$

Where a and v represent acceleration and velocity, while t is the time between two frames in the video.

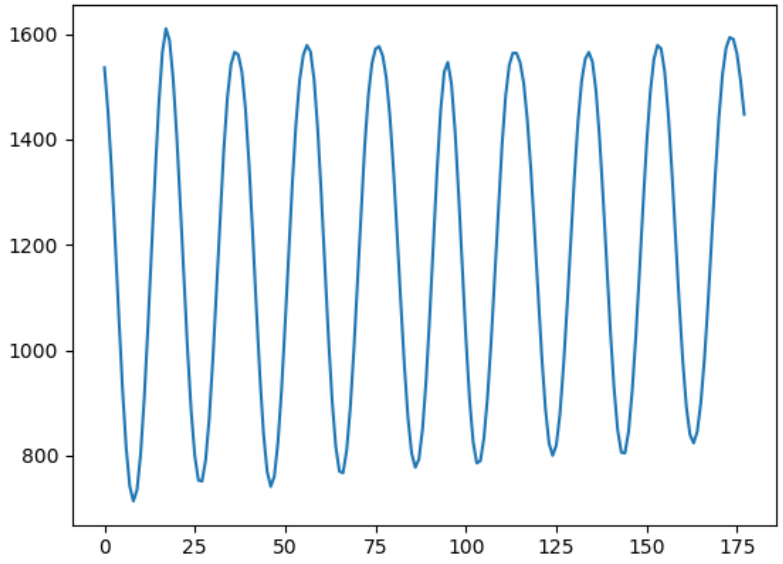

Figure 7: Fitted data of the experiment with the pendulum (position of the ball over time)

6 Results

6.1 Expected results

The predicted output was a simple formula that would explain the motion that we obtained through the aforementioned simulations. However, in order to obtain those results, we required a greater processing power and more time, which we did not have.

6.2 Final Results

The equation which we obtained at the end was not the correct formula but it was close enough. We may, hence, assume that given more running time, the formula might have been found. Below we give the equation of motion obtained by our algorithm.

$$8 \times a(x) + 4 \times v(x) + x = 0 \tag{18}$$

Therefore, our code works for one dimensional problems, and our next step would be to extend it to more dimensions.

7 Conclusion

What we achieved was done in limited time and with limited technology. Had we had some more time for this project, we could well have expanded it in order to tackle two dimensional problems too, and perhaps extend it to other fields of science as well. Another thing we would like to note is that the birth of artificial intelligence does not mean the end of human usefulness. Artificial intelligence is but a tool in humankind's long and rewarding stride towards pushing the frontiers of science.

References

[1] URL: https://nl.mathworks.com/help/matlab/ref/spline.html (visited on 08/17/2017).

[2] MATLAB. *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.

[3] Michael Schmidt and Hod Lipson. "Distilling Free-Form Natural Laws from Experimental Data". In: *Science* 324.5923 (2009), pp. 81–85.