Machine learning as a tool to obtain new black hole solutions

CS 490, RnD Project

Liza Dahiya

IIT Bombay

10th December, 2021

Aims & Objective

The aim of the research project is explore and use various neural networks libraries to solve various differential equations in black hole and general relativity to obtain solutions satisfying various constraints.

What is `neurodiffeq`?

Understanding `neurodiffeq`

`neurodiffeq` is a Python package for solving differential equations, It models loss for the following objective functions:

$$\min(\mathcal{L}\tilde{u} - f)^2 \tag{1}$$

Here, $\tilde{u}(x, t) = u_0(x) + (1 - e^{-(t-t_0)})u_N(x, t)$,

where $u_0(x)$ is the initial conditions imposed on the differential equation and $u_N(x)$ is the solution of differential equation modelled by the neural network.

Schwarzschild Equations

Framing the equations

Aim: Solve differential equations to yield α and β of schwarzschild solutions.

$$ds^2 = -e^{2\alpha(t,r)}\partial t^2 + e^{2\beta(t,r)}\partial r^2 + r^2\partial\Omega^2 \tag{2}$$

Ricci Tensor equations R_{00} and R_{11} were solved satisfying IVP constraints given by **Minkowski space**, which states that when $r \to \infty$, the value of α and β tends to 0.

$$\partial s^2 = -\partial t^2 + \partial r^2 + r^2\partial\Omega^2 \tag{3}$$

The ricci tensors to be solved by `neurodiffeq` are given as:

$$R_{00} = \frac{e^{2(\alpha-\beta)}}{\delta^2}[\partial_\rho^2\alpha + (\partial_\rho\alpha)^2 - \partial_\rho\alpha\partial_\rho\beta + \frac{2}{\rho}\partial_\rho\alpha]$$

$$R_{11} = -\frac{1}{\delta^2}[\partial_\rho^2\alpha + (\partial_\rho\alpha)^2 - \partial_\rho\alpha\partial_\rho\beta - \frac{2}{\rho}\partial_\rho\beta]$$

Solution Obtained

The plots for α and β obtained after feeding the network equations R_{00} and R_{11} is shown in the following figure:

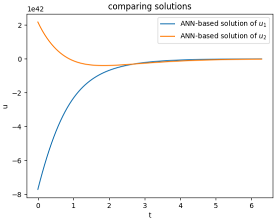

Figure: Plot showing variation of α and β

Hyperparameter Tunning

For choosing the IVP value of r (or δ), we run the model with values of r ranging from 100 to 2, and model the value of training loss.

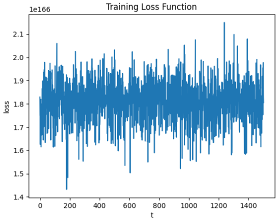

Figure: Observed training loss at $r \to \infty$ (set 100)

Solving Schwarzschild in a Single Variable

Framing the Equation

When solving the two variable equation, the relation between in α and β was obtained as follows. We used this relation and substituted in the original equation to be solved.

$$\boxed{\beta = -\alpha}$$

After substitution, we obtained the Ricci tensor R_{00} as follows which we solved using `neurodiffeq` with the IVP constraints as in the previous case.

$$R_{00} = \frac{e^{4\alpha}}{\delta^2}[\partial_\rho^2\alpha + 2(\partial_\rho\alpha)^2 + \frac{2}{\rho}\partial_\rho\alpha] \quad (4)$$

Using Bernaulli Equations

To simplify the equation above, we perform these series of subsitution to convert the equations into generic known form to be solved easily.

$$v = \frac{\partial \alpha}{\partial \rho}$$

$$u = \frac{1}{v}$$

$$\frac{\partial v}{\partial \rho} = \frac{\partial^2 \alpha}{\partial^2 \rho}$$

Using Bernaulli Equations

The equation obtained after all substituion corresponds to the famous **Bernoulli equation** which has a formulated solution and hence we solve this first degree ODE equation instead of the orginal one.

$$\frac{\partial u}{\partial \rho} - \frac{2u}{\rho} = 2$$

Solution Obtained

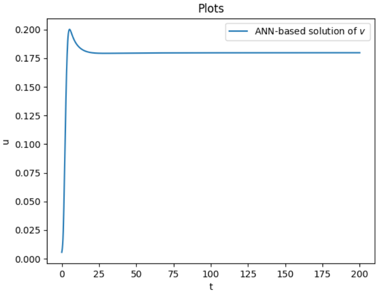

Figure: $v = \frac{1}{u}$ where u is solution of the Bernoulli equation

Infinitely degenerate Ricci-flat solutions in $f(R)$ gravity

Original Equations

Aim: To obtain the following solution of $A(r)$ which is generated when $\delta(r)$ in equated to 0 in the equation of T_1 or T_2.

$$A(r) = 1 + C_2 r^2 - \frac{C_3}{r^2} \tag{5}$$

where $C_2 = \frac{\alpha_0}{12\alpha_1} > 0$ and C_3 is the constant of integration.

We'll use T_2 in our solution and apply $\delta(r) = 0$ in the equation of T_2 below:

$$T_2[A(r), \delta(r)] = r^2 A(r) \left[\left(\frac{\Phi(r)}{r}\right)' + \left[\left(\frac{4+r}{2r}\right) + \frac{3}{2}[\ln{(A(r))}]' \right] \left(\frac{\Phi(r)}{r}\right) \right]$$
$$- \frac{r^2 A(r)}{2} \left([\ln A(r)]'^2 + \left(\frac{4+r}{r}\right)[\ln{(A(r))}]' - \frac{4}{r}\left[\frac{1}{r} + 2\right] \right) - 2\left(1 + \frac{\alpha_0 r^2}{2\alpha_1}\right)$$

Constraints and Relations

We will use the following relation and substitution respectively and apply these in the above equation of T_2 to simplfy the final equation to be used by `neurodiffeq`.

$$\Phi(r) = r(\delta'(r) + [\ln A(r)]') \quad \text{as} \quad \delta(r) = 0 = \delta'(r)$$
$$= r[\ln A(r)]'$$

$$B(r) = \ln(A(r))$$

Final Equation solved by `neurodiffeq`

After all the substitution and simplification, we obtain the final equation as given below.

$$\left[B''(r) + (B'(r))^2 + \frac{2}{r}\left(\frac{1}{r} + 2\right)\right] - 2e^{-B(r)}\left(\frac{1}{r^2} + \frac{\alpha_0}{2\alpha_1}\right) = 0 \tag{6}$$

In this equation, we put $\alpha_0 = 1$ and $\alpha_1 = 1$. The IVP for the equation would be $r \to 10$, then $A(r) = 100$ and $A'(r) = 10$.

Work Action Plan

Work to be continued. . .

- Obtain solutions for Ricci Tensors by hyperparameter tuning
- Understand cases in which the solutions blow up (ex: using log and exp).
- Making results more robust than they're currently
- Address the equations that gave different results on reruning the model everytime.