Learning the SIR Epidemic Model

> **GSoC 2026 Proposal Prototype for Human AI Foundation** > *Project:* Deducing deterministic ODEs of the SIR model from stochastic simulations using Machine Learning and Auto-differentiation.

Project Overview

The classic Susceptible-Infected-Removed (SIR) epidemic model is defined by a set of Ordinary Differential Equations (ODEs). However, real-world epidemics are often stochastic. This project aims to bridge that gap by using **Machine Learning** to deduce the deterministic form of the SIR model from a large number of synthetic, stochastically generated epidemics.

This repository contains the prototype built for the GSoC '26 proposal, demonstrating the end-to-end pipeline:

1. Generating stochastic SIR data (Gillespie/Binomial approximation).
2. Processing data through a PyTorch-based Neural Network.
3. Using Auto-Differentiation to extract dS/dt, dI/dt, and dR/dt.
4. Visualizing the comparison via a modern Web UI.

Tech Stack

- **Backend:** Python, FastAPI, Uvicorn
- **Machine Learning & Math:** PyTorch (for Auto-diff & PINNs), NumPy
- **Frontend:** HTML5, TailwindCSS, JavaScript
- **Data Visualization:** Chart.js

Project Structure

```
gsoc-sir-project/
├── backend/
│   ├── main.py            # FastAPI server & API endpoints
│   ├── simulator.py       # Stochastic SIR simulation logic
│   ├── ml_model.py        # PyTorch Neural Network & Auto-diff logic
│   └── requirements.txt   # Python dependencies
└── frontend/
    └── index.html         # Interactive Dashboard UI
```

Installation & Setup

Follow these steps to run the project on your local machine.

1. Clone the repository

```
git clone https://github.com/PritamHazra2708/SIR-Model-For-HumanAI.git
cd SIR-Model-For-HumanAI
cd gsoc-sir-project
```

2. Setup the Backend

Navigate to the backend directory and install the required dependencies:

```
cd backend
python -m venv venv
source venv/bin/activate  # On Windows use: venv\Scripts\activate
pip install fastapi uvicorn torch numpy pydantic
```

3. Run the FastAPI Server

Start the backend server using Uvicorn:

```
uvicorn main:app --reload
```

The API will be available at: `http://127.0.0.1:8000`

4. Run the Frontend

Simply open the `frontend/index.html` file in any modern web browser. No frontend build tools are required for this prototype.

🧠 Core Highlight: Auto-Differentiation

A major requirement of this project is using symbolic methods to approximate $S(t), I(t), R(t)$. In `backend/ml_model.py`, we utilize PyTorch's `autograd` engine to compute derivatives directly from the neural network's graph:

```
# Extracting exact derivatives learned by the model
dS_dt = torch.autograd.grad(S_pred, inputs, create_graph=True)
dI_dt = torch.autograd.grad(I_pred, inputs, create_graph=True)
dR_dt = torch.autograd.grad(R_pred, inputs, create_graph=True)
```

🤝 Contributing

This is an open-source initiative under the Human AI Foundation. Suggestions and pull requests are welcome!

📄 License

MIT License