Analysis and Prediction of Lorenz Attractor Dynamics Using VAE-RNN

-Apoorva Chavali

Abstract

In the recent era, deep learning models have gained significant popularity due to their ability to learn and replicate complex trends, making them suitable for addressing real-world problems. This project aims to bridge the gap between the non-linear dynamics of real-world phenomena and the capabilities of deep learning models. The Lorenz attractor, a hallmark of chaotic systems, serves as a challenging benchmark for modelling and predicting such non-linear dynamics. This study employs a Variational Autoencoder-Recurrent Neural Network (VAE-RNN) to capture the underlying structure of the Lorenz attractor and predict its future states. This project demonstrates that by combining the representational power of VAEs with the temporal modelling capability of RNNs, it is possible to learn the dynamics of chaotic systems, enabling both accurate short-term predictions and meaningful latent space representations. This approach highlights the potential of deep learning in unravelling the complexities of non-linear systems and offers a promising direction for applying such models to real-world chaotic phenomena.

Introduction

The Lorenz attractor was first studied by first studied by mathematician and meteorologist Edward Lorenz. It has the shape of a butterfly and infinitesimally small differences in initial conditions result in vastly different trajectories over time. This sensitivity makes long-term prediction impossible, as any uncertainty in the initial state quickly amplifies. Lorenz attractor explores new paths indefinitely within its constraints. The Lorenz attractor arises from deterministic equations, meaning its behaviour is governed entirely by the system's rules without any randomness. However, due to its sensitivity to initial conditions and complex dynamics, predicting its long-term behaviour is practically impossible.

These features collectively make the Lorenz attractor a quintessential example of chaos in a deterministic system, offering insights into weather modelling, fluid dynamics, and other real-world chaotic phenomena.

To address this challenge of sensitivity in initial conditions, some approaches focus on capturing the overall structure of the attractor in state space rather than precise time series predictions. VAE RNN’s approach focuses on capturing the overall structure of the attractor in state space, the VAE-RNN can mitigate issues related to the system's sensitivity to initial conditions and chaotic nature.

Data Preparation:

The input data is the **normalized trajectory of the Lorenz attractor** in 3D (x, y, z) space to have zero mean and unit variance. The trajectory is generated by solving the Lorenz system of equations using numerical integration (solve_ivp). This data represents the chaotic evolution of the system over time.

Raw Data

- Shape: (N,3) where:
  - N: Total number of time steps (e.g., 10,000).
  - 3: Features (x, y, z) at each time step.

Before being passed to the encoder:

- Normalised = (Data – Mean) / standard deviation
- **Why?**
  - Prevents features with larger magnitudes from dominating.
  - Improves training stability and convergence.

VAE Architecture

Now we have established that chaotic systems like the Lorenz attractor exhibit high sensitivity to initial conditions and non-linear dynamics. A VAE models these complexities by encoding inputs into a **probabilistic latent space**. Why is this helpful? Latent space reflects the continuous nature of the attractor, capturing its geometry and dynamics compactly. By sampling from the latent space, the model can generate new trajectories consistent with the learned dynamics and **explicitly models this uncertainty**, using distributions (mean and variance) instead of fixed values.

VAEs excel at learning compressed representations while preserving enough information to reconstruct the input data. This property is crucial for reconstructing the chaotic trajectories of the Lorenz attractor and predicting future states, which requires both understanding the global system structure and preserving fine details.

So why not use an LSTM alone? LSTMs are powerful for sequence modelling because they maintain a memory of past states through their gating mechanisms. But they alone lack an explicit latent representation to encode the global structure of the system. They focus more on local patterns and can struggle with generalization in high-variability or highly non-linear data like chaotic systems.

Experimental results

The loss decreases steadily across the epochs and reached 0.0139 by epoch 20, which indicates the model is effectively reconstructing the sequences while regularizing the latent space. Fig 1 shows that similar points in the original trajectory are mapped to nearby locations in the latent space, preserving the attractor's dual-lobe behaviour.

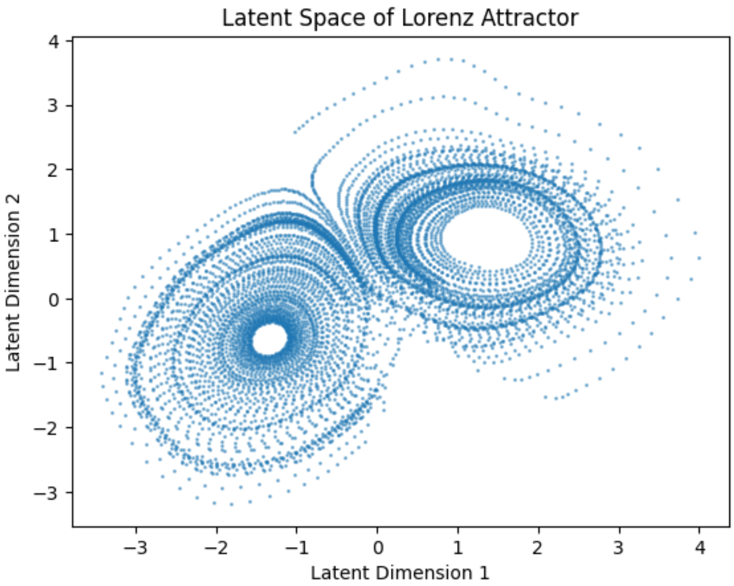

Fig 1

Fig2, Fig 3, Fig 4 show the accurate preservation of the Lorenz attractor's structure in the climatology plots.

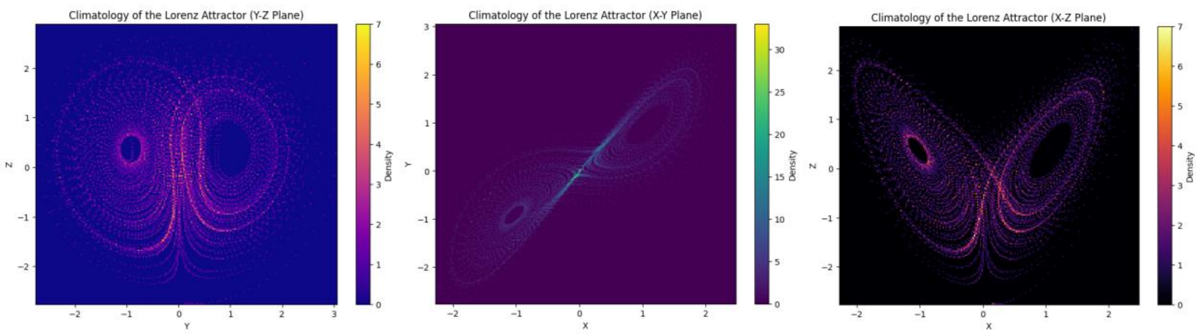

Fig 2,3,4

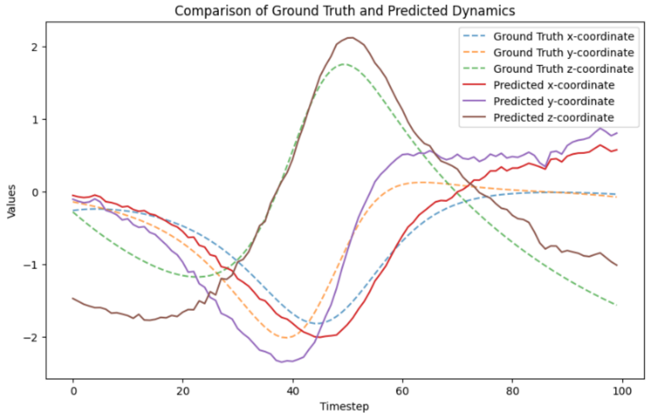

Fig 5

In fig 5, we can observe that at the beginning of the prediction (first few timesteps), the **predicted values closely match the ground truth** for all three coordinates. This is because the model uses the last sequence of the actual data as the initial input, which provides a strong starting point. As time progresses, the predictions start to diverge from the ground truth. This divergence becomes more prominent after 40-50 timesteps. This is because of the attractor's sensitivity to initial conditions. To quantify this, Lyapunov's exponents were calculated. We obtained a value of 0.4002 for the largest Lyapunov's exponent which confirms that the model is correctly capturing the dynamics. (Fig 6). Later, predicting far into the future without corrections accumulates small errors, leading to divergence. Despite divergence, the models successfully captures the general trends and oscillatory nature of the attractor. Instead of predicting 100 steps at once, reintroducing intermediate ground truth sequences every 30-50 steps may mitigate error accumulation.

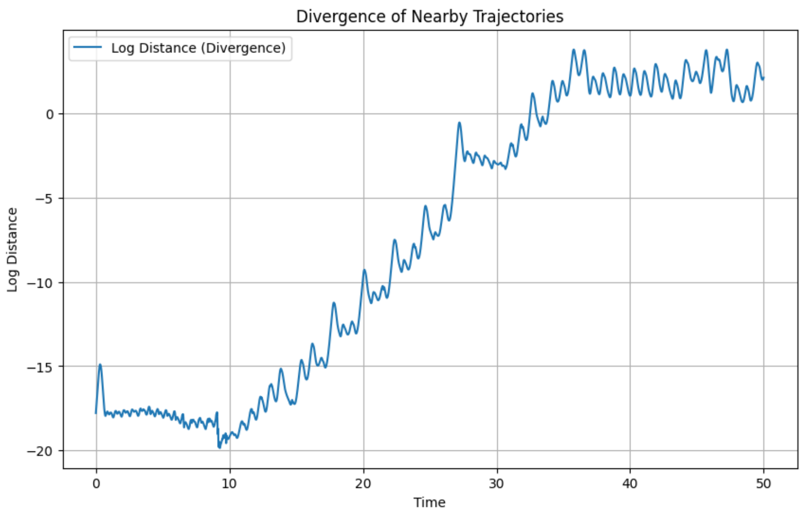

Fig 6

We are also calculating the number of timestamps it takes for the trajectory to jump from one wing to another. The time interval between these jumps is not constant but reflects the system's chaotic nature. The predicted jump align closely with the actual jumps, it shows that the model understands both the attractor's geometry and its time evolution. The **ground truth position after 32 timesteps** (purple point) also resides on the same wing as the predicted point, although there is a small positional error (Euclidean distance = 0.3714).

Model's o/p - The system will jump to the other wing in 71 timestamps.

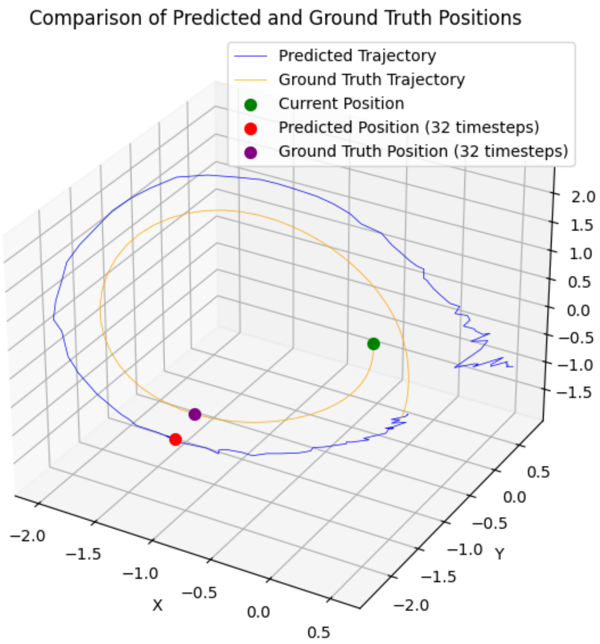

Fig 7

Fig 7 illustrates the comparison between the **predicted trajectory** and the **ground truth trajectory** of the Lorenz attractor in a 3D phase space. It highlights the **current position** (green point), the **predicted position after 32 timesteps** (red point), and the **ground truth position after 32 timesteps** (purple point). The blue line represents the predicted trajectory, while the orange line corresponds to the ground truth trajectory.

To show that the model is doing well for short term forecasts, Euclidean distance is calculated. At an intermediate prediction point (32 timesteps), the predicted position shows a low Euclidean error of 0.3714 compared to the ground truth.

Conclusion

This project demonstrates the feasibility and effectiveness of using VAE-RNNs for modelling chaotic systems, laying a strong foundation for future research. By addressing challenges in long-term predictions and extending the scope to real-world applications, this approach has the potential to revolutionize the way we analyse and predict complex dynamical systems.

References –

1. https://en.wikipedia.org/wiki/Lorenz_system
2. https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf
3. https://towardsdatascience.com/vae-for-time-series-1dc0fef4bffa
4. https://conservancy.umn.edu/server/api/core/bitstreams/472c6e3b-ab0e-463a-ac87-736ed452f1a2/content
5. https://conservancy.umn.edu/server/api/core/bitstreams/472c6e3b-ab0e-463a-ac87-736ed452f1a2/content