# Finding Koopman operator using Neural Network

EE 695: Data Driven System theory

Name: Rajavelu E Roll Number: 234102308

## 1 Koopman Operator

1. The Koopman operator is a linear operator that maps functions of the system state to functions of the system state at a later time.

2. The Koopman operator works by transforming the dynamics of a nonlinear system into a linear system.The Koopman operator can be used to identify the key modes of variation in a nonlinear system. This makes it easier to study and control the system.

## 2 System of Non linear functions

Choosing system of 4 non-linear functions as $-e^{x_1^2},\ {x_2}^2 - 1 + x_1, sin(x_3) - x_2, x_1^3 - 1$

$$\frac{dx_1}{dt} = -e^{x_1^2}$$
$$\frac{dx_2}{dt} = {x_2}^2 - 1 + x_1$$
$$\frac{dx_3}{dt} = sin(x_3) - x_2$$
$$\frac{dx_4}{dt} = x_1^3 - 1$$

## 3 Finding solution to the system of ODEs

The solution to ordinary differential equations (ODEs) is computed by odeint function from the scipy.integrate module.

```
odeint(func, y0, t, .....)

Parameters
func: the non linear differential equation
y_0  : Initial value of Y
t   : the time space for which we want the curve(basically the range of x)
```

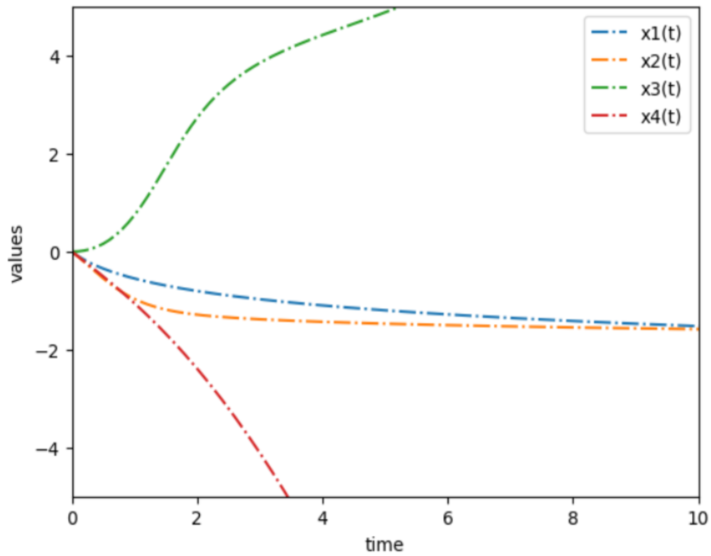



Figure 1: Solutions of system of ODEs

# 4 Methodology

Converting a non-linear system into a linear system using Koopman Operator involves finding an appropriate Koopman Operator that linearizes the system. These are the following steps to achieve this:

- When dealing with nonlinear functions in ordinary differential equations (ODEs), we need to identify a suitable function g(x) that accurately represents the system's dynamics. Since g(x) is often unknown, we can approximate it using a neural network with a reduced dimensionality.
- The subsequent block should adhere to linearity, and the layer weights of this block in the trained model represent the Koopman operator. This ensures that the neural network effectively captures the linear dynamics of the system, which can be further utilized for analysis and control purposes.
- To balance the output and input dimensions of the model, we need to approximate another function, $g^{-1}(x)$ which serves as an upscaling mechanism. This function effectively inverts the dimensionality reduction performed by the initial neural network block.
- Now we can compute the mean square error between the predicted and the true outputs in each iteration and do backpropagation to update the weights.

# 5 Neural Network Model Description

- The model comprises three primary blocks: "g" "Koopman" and "g inverse" each playing a distinct role in achieving the desired system representation and behavior.
- The g block comprises a cascade of three hidden layers, each with a decreasing

number of neurons. This sequential reduction in dimensionality serves to compress the input data while preserving the essential features of the system's dynamics.

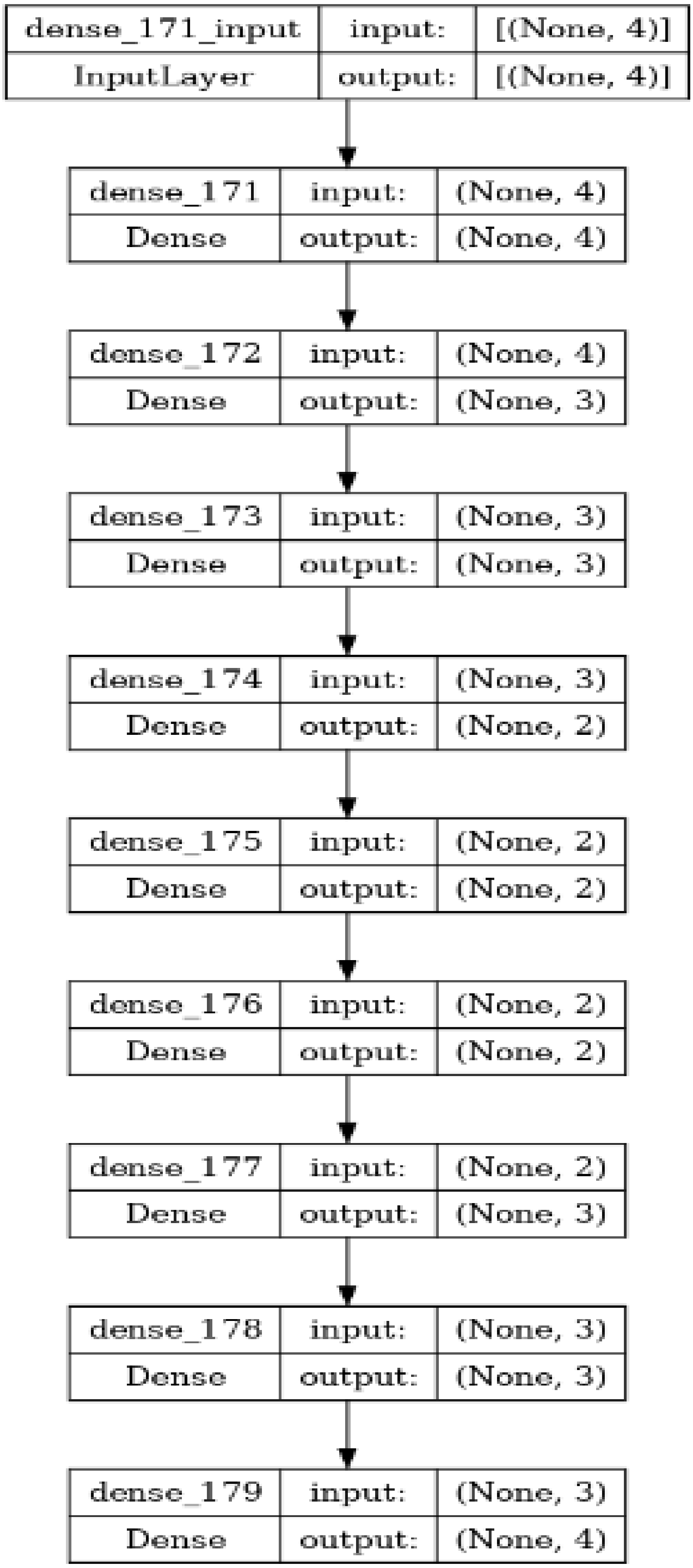
- The Koopman block is implemented using a single layer with an activation function set to 'linear.' This choice of activation function ensures that the block preserves the linearity introduced by the Koopman operator, enabling the model to effectively capture the linearized dynamics of the system.

- The g inverse block plays a crucial role in maintaining consistency in the model's input and output dimensions. By reversing the order of the layers in the g block, the g inverse block effectively inverts the dimensionality reduction process, ensuring that the model's output can be directly compared to the input data.

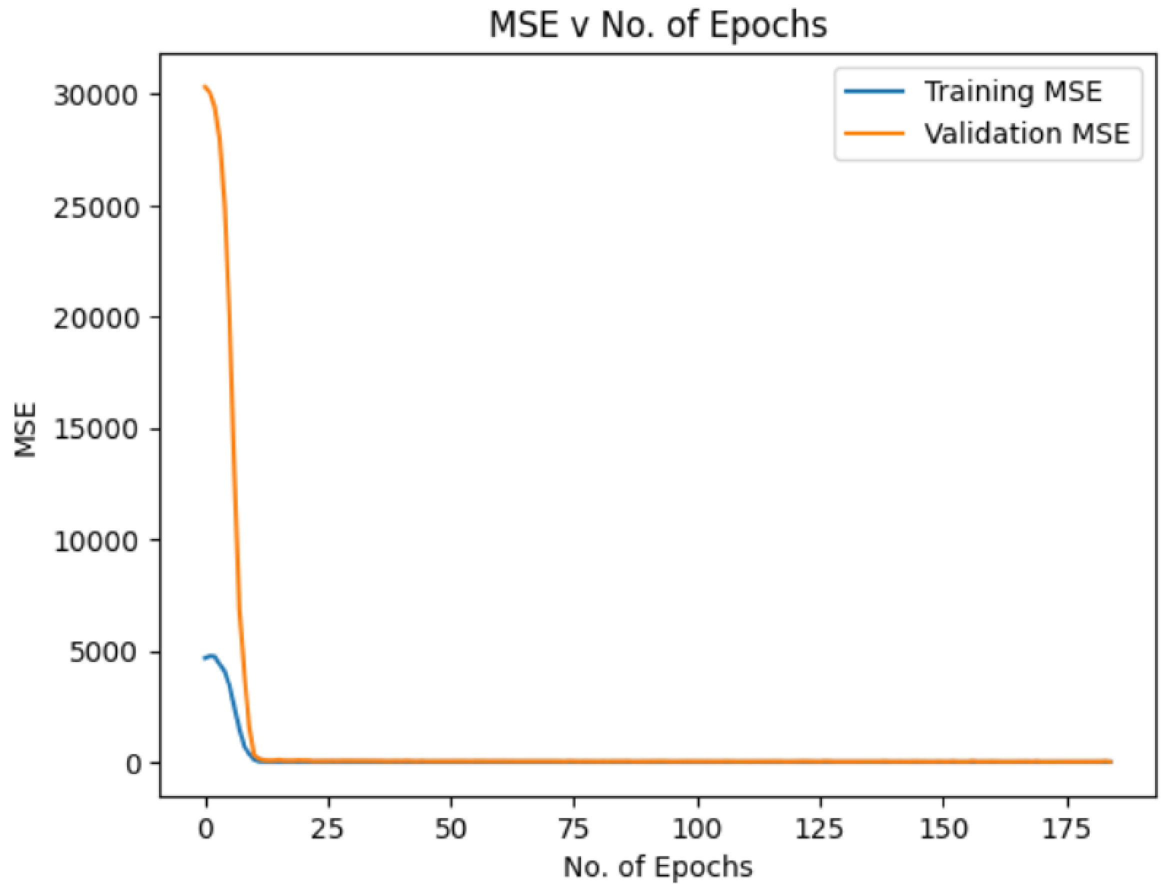
```
 5 values in Actual Output
[[   -2.13274621   -1.76882487   45.05415017 -225.12456642]
 [   -2.13414874   -1.76922474   45.32419183 -226.19562667]
 [   -2.13554734   -1.76962338   45.600405   -227.26860077]
 [   -2.13694204   -1.77002082   45.87536755 -228.34348608]
 [   -2.13833286   -1.77041705   46.14188352 -229.42027897]]
5 values in Predicted Output
[[  -2.0799336   -1.7095134   45.396206  -224.82726   ]
 [  -2.0826702   -1.7092793   45.58887   -225.91731   ]
 [  -2.0854154   -1.7090442   45.782463  -227.01266   ]
 [  -2.088177    -1.7088094   45.976665  -228.11145   ]
 [  -2.0909379   -1.70858     46.171143  -229.21178   ]]
```

Figure 2: Prediction v/s Actual Outputs

# 6 Neural Network Model Representation



# 7 Mean square Error vs Epochs




# 8 Results

- Trained model is giving fairly accurate predictions of the chosen non-linear functions.
- Koopman operator value obtained is:

```
Koopman Operator:
 [[-1.4666873   0.5093602 ]
 [-0.93931884  0.00586727]]
```