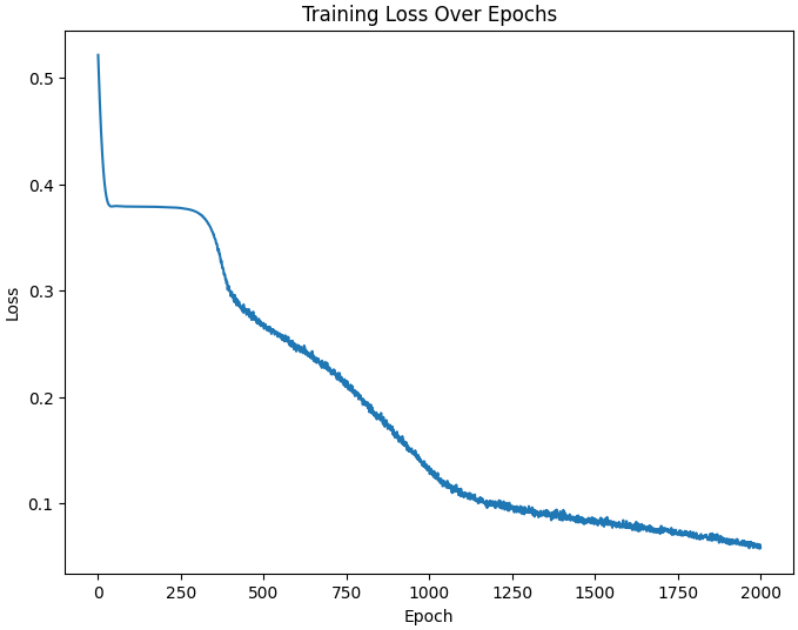
PDE using PINNs

This model is trained using a specific method, in which the exact solution of the PDE problem at boundaries must be given to the model.

1) User inputs:
   First of all, the PDE problem is entered by user in the specific format explained in the notebook. Then, the domain for x and t, which are the independent variables is given, along with the exact solution of the PDE in the boundaries. The solution must be given in a python code format to later be converted into a function called “sol”.
2) Model:
   The model consists of 5 hidden layers followed by sigmoid activation function. Adam optimizer is used in this task.
   - Loss function: The total loss for training this model is comprised of two parts. One for PDE and the other is the error in boundary conditions, combined together they make up the total loss. Also, Mean Squared Error was used for computing the error.
   - PDE Loss: This loss is computed by getting the difference of the u predicted by model for a certain point (x, t) and the actual value for this point. Using this loss to train the model, it will learn effectively to predict more accurately.
   - Boundary loss: I’ve set 300 boundary points (x, 0) and the u for these points is easily calculated using the exact solution provided by the user. Including this loss in training will ensure that the model will satisfy the condition for boundary points.

3) Training result:
The graph below tracks the loss during training and show its value for each epoch:



This plot shows the values of u predicted by model with their corresponding input pair (x, t)