**EX2 : Implementation of MLP - Backpropagation using Pytorch**

**ODD Batch:**

1. Train a neural network for the function y=x^4+x^5.

Constraints:

i)Use sigmoid activation function

ii)Consider the loss function as Mean Squared Error.

iii) weight updation : w.data = w.data - learning\_rate \* w.grad

1. Train a Neural Network with Stochastic Gradient Descent and Negative log likelihood loss function for XOR operation
2. Load the Lung image dataset using customized functions.
3. Create an MLP with backpropagation for classifying Vehicle images and save the model. (Note : EDA of the dataset should be included)

**EVEN Batch:**

1. Train a neural network for the function y=8x+7x^3+2.

Constraints:

i)No hidden Layers

ii)Consider the loss function as Mean Squared Error.

iii) weight updation : w.data = w.data - learning\_rate \* w.grad

1. Train a Neural Network with Stochastic Gradient Descent and L1 loss function for XOR function
2. Load the Tree dataset using customized functions.
3. Create an MLP with backpropagation for classifying handwritten digits and save the model. (Note : EDA of the dataset should be included)