

# Homework 3 - Option 1

UCLA MAE 263F, Fall 2024  
Mechanics of Flexible Structures and Soft Robots

He Kai Lim<sup>1</sup>

**Abstract—**This report strives to provide all deliverables for Homework 3 in partial fulfilment of the requirements for UCLA MAE 263F, Fall 2024 (Mechanics of Flexible Structures and Soft Robots)

## I. INTRODUCTION

Fits are important to perform good science, and machine-learned iterative methods are a useful method to create these linear fits when the data is unintuitive. There are two main types of fits: linear and nonlinear. Both need to be carefully examined to understand the methods of implementing machine learning methods in creating these fits.

## II. PROBLEM 1 - LINEAR FITS

Here, we want to fit data to a linear equation:

$$y = mx + b \quad (1)$$

### A. Theory

We use the Mean Squared Error (MSE) loss function:

$$\text{Loss}(m, b) = \frac{1}{N} \sum_{i=1}^N (y_i - (m \cdot x_i + b))^2 \quad (2)$$

The gradients of the loss function with respect to  $m$  and  $b$  are:

$$\frac{\partial \text{Loss}}{\partial m} = -\frac{2}{N} \sum_{i=1}^N x_i \cdot (y_i - (m \cdot x_i + b)) \quad (3)$$

$$\frac{\partial \text{Loss}}{\partial b} = -\frac{2}{N} \sum_{i=1}^N (y_i - (m \cdot x_i + b)) \quad (4)$$

Using gradient descent, we update the hyperparameters  $m$  and  $b$  as follows:

$$m \leftarrow m - \eta \frac{\partial \text{Loss}}{\partial m}, b \leftarrow b - \eta \frac{\partial \text{Loss}}{\partial b} \quad (5)$$

where  $\eta$  is the learning rate.

### B. Results: 1a

Compare the predicted values with the actual data. After fitting the model, plot the predicted values  $y_{pred}$  versus the actual data  $y$ . Include the plot in your report.

These are the results of fitting the model for 10000 epochs at a learning rate of 0.001.

<sup>1</sup>He Kai Lim is with the Department of Mechanical and Aerospace Engineering, University of California Los Angeles, Los Angeles, CA 90095 USA limhekai@ucla.edu

TABLE I  
LINEAR FIT: NOMINAL VALUES OF  $m_{fit}$  AND  $b_{fit}$

$m_{fit}$	$b_{fit}$
-0.011781642379859762	0.06320409591699154

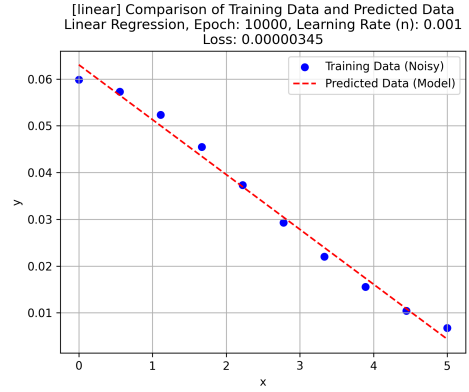


Fig. 1. Predicted values  $y_{pred}$  versus actual data  $y$  with original values

### C. Results: 1b

Problem 1b: Experiment with the learning rate and number of epochs.

Here I manually tried different values of epochs and learning rate ( $\eta$ ) and found the set that minimized loss with ensured convergence

TABLE II  
LINEAR FIT: TABLE OF HYPERPARAMETERS

Epoch	Learning Rate ( $\eta$ )	Loss	Convergence	Figure
10000	0.001	0.00000391	Yes	1
1000	0.001	0.02600305	No	2
10000	0.01	0.00000343	Yes	3
1000	0.01	0.00000343	Yes	4

Changing the learning rate effects the number of epochs required for convergence. A smaller learning rate requires more epochs before convergence. Furthermore, a larger learning rate also affects the precision of the fit, meaning that a minimum loss might not be found because of the larger "steps" from the learning rate.

Changing the number of epochs affects the final loss, assuming loss is not yet minimized. This means there is a minimum number of epochs required to get the optimum fit, and thereafter, higher numbers of epochs are wasted a computation resource.

Together, this means we want a learning rate that is just

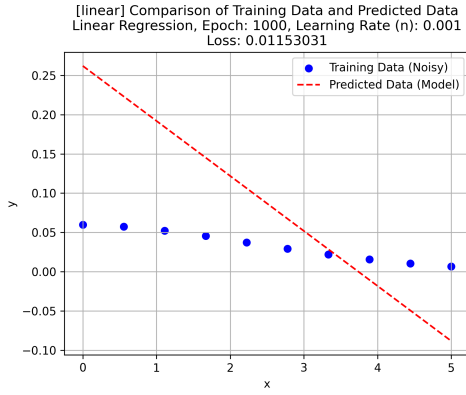


Fig. 2. Predicted values  $y_{pred}$  versus actual data  $y$  with no convergence

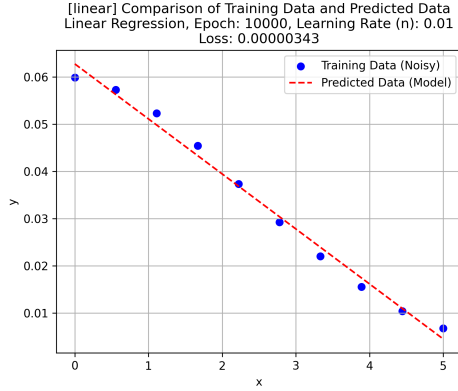


Fig. 3. Predicted values  $y_{pred}$  versus actual data  $y$  with minimum loss but excess epochs

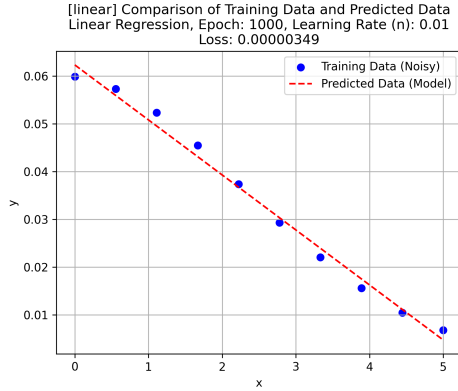


Fig. 4. Predicted values  $y_{pred}$  versus actual data  $y$  with minimum loss and optimized epochs

large enough to find the minimum loss fit, then we want to minimize the number of epochs within this fit. This was manually found and presented in Figure. 4.

### III. PROBLEM 2 - NONLINEAR FITS

#### A. Theory

Here, we want to fit the same data (from problem 1) onto the nonlinear equation:

$$y = n \cdot \exp(-a \cdot y_{int}), \text{ where } y_{int} = (m \cdot x + b)^2 \quad (6)$$

using gradient descent and backpropagation.

We use the Mean Squared Error (MSE) loss function:

$$\text{Loss}(n, a, m, b) = \frac{1}{N} \sum_{i=1}^N (y_i - n \cdot \exp(-a \cdot (m \cdot x_i + b)^2))^2 \quad (7)$$

with the following gradients of the loss function with respect to  $n$ ,  $a$ ,  $m$ , and  $b$  as:

$$\frac{\partial \text{Loss}}{\partial n} = -\frac{2}{N} \sum_{i=1}^N (y_i - n \cdot \exp(-a \cdot (m \cdot x_i + b)^2)) \cdot \exp(-a \cdot (m \cdot x_i + b)^2) \quad (8)$$

$$\frac{\partial \text{Loss}}{\partial a} = \frac{2}{N} \sum_{i=1}^N (y_i - n \cdot \exp(-a \cdot (m \cdot x_i + b)^2)) \cdot n \cdot \exp(-a \cdot (m \cdot x_i + b)^2) \cdot (- (m \cdot x_i + b)^2) \quad (9)$$

$$\frac{\partial \text{Loss}}{\partial m} = \frac{2}{N} \sum_{i=1}^N (y_i - n \cdot \exp(-a \cdot (m \cdot x_i + b)^2)) \cdot n \cdot \exp(-a \cdot (m \cdot x_i + b)^2) \cdot (-2a \cdot (m \cdot x_i + b) \cdot x_i) \quad (10)$$

$$\frac{\partial \text{Loss}}{\partial b} = \frac{2}{N} \sum_{i=1}^N (y_i - n \cdot \exp(-a \cdot (m \cdot x_i + b)^2)) \cdot n \cdot \exp(-a \cdot (m \cdot x_i + b)^2) \cdot (-2a \cdot (m \cdot x_i + b)) \quad (11)$$

and with a parameter update method of:

$$n \leftarrow n - \eta \frac{\partial \text{Loss}}{\partial n}, \quad a \leftarrow a - \eta \frac{\partial \text{Loss}}{\partial a} \quad (12)$$

$$m \leftarrow m - \eta \frac{\partial \text{Loss}}{\partial m}, \quad b \leftarrow b - \eta \frac{\partial \text{Loss}}{\partial b} \quad (13)$$

where  $\eta$  is the learning rate.

### B. Results: 2a

Compare the predicted values with the actual data. After fitting the model, plot the predicted values  $y_{pred}$  versus the actual data  $y$ . Include the plot in your report.

These are the results of fitting the model for 10000 epochs at a learning rate of 0.001.

TABLE III

NONLINEAR FIT: NOMINAL VALUES OF  $n_{fit}$ ,  $a_{fit}$ ,  $m_{fit}$  AND  $b_{fit}$

$n_{fit}$	0.05814377819334344
$a_{fit}$	0.3232072211957599
$m_{fit}$	0.38542333824260566
$b_{fit}$	0.2709157627577561

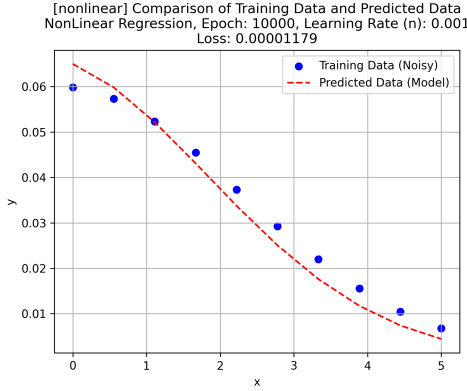


Fig. 5. Nonlinear predicted values  $y_{pred}$  versus actual data  $y$  with original values

### C. Results: 2b

Experiment with the learning rate and the number of epochs.

Here I manually tried different values of epochs and learning rate ( $\eta$ ) and found the set that minimized loss with ensured convergence

TABLE IV

NONLINEAR FIT: TABLE OF HYPERPARAMETERS

Epoch	Learning Rate ( $\eta$ )	Loss	Convergence	Figure
10000	0.001	0.00001179	Yes	5
1000	0.001	0.00005946	Yes	6
10000	0.01	0.00007575	Yes	7
50000	0.0001	0.00000775	Yes	8

Changing the learning rate effects the number of epochs required for convergence. A smaller learning rate requires more epochs before convergence. Furthermore, a larger learning rate also affects the precision of the fit, meaning that a minimum loss might not be found because of the larger "steps" from the learning rate.

Changing the number of epochs affects the final loss, assuming loss is not yet minimized. This means there is a minimum number of epochs required to get the optimum fit, and thereafter, higher numbers of epochs are wasted a computation resource.

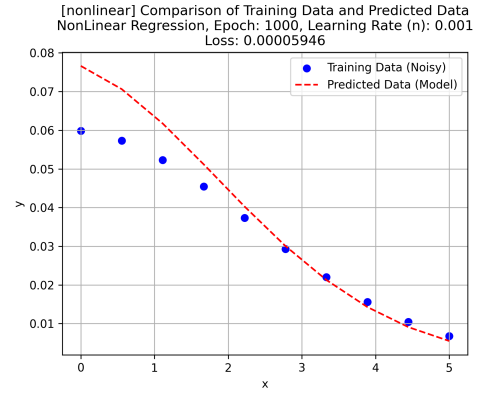


Fig. 6. Predicted values  $y_{pred}$  versus actual data  $y$  with no convergence

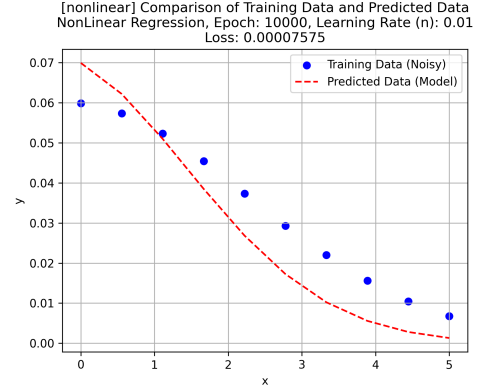


Fig. 7. Predicted values  $y_{pred}$  versus actual data  $y$  with minimum loss but excess epochs

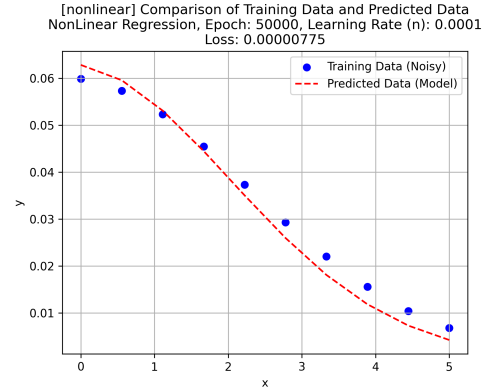


Fig. 8. Predicted values  $y_{pred}$  versus actual data  $y$  with minimum loss and optimized epochs

Together, this means we want a learning rate that is just large enough to find the minimum loss fit, then we want to minimize the number of epochs within this fit. This was manually found and presented in Figure. 8.

Importantly, we note the qualitative observation that the initial guess for  $n$ ,  $a$ ,  $m$ ,  $b$  also have nontrivial effect on the ability for the system to converge, and also on the minimum loss value. In the code submitted herein, there is logic to pick feasible initial values first before commencing

epoch iteration. This was required to mitigate the effects of randomness in initial guesses, and to enforce repeatability in the manual, heuristic search of the best hyperparameters.

#### ACKNOWLEDGMENT

Material used in this report are taken from the course UCLA MAE 263F, Fall 2024.

#### REFERENCES

- [1] M. Khalid Jawed, Singmin Lim, Discrete Simulation of Slender Structures, UCLA Fall 2024 MAE 263F