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

Homework 3

Due Date: 11:59 PM, 12/10/2024

Submission Instructions: Submit your solution by uploading all files (PDF report and code) to your GitHub repository. Then, submit only the repository URL on the course website (BruinLearn).

1. 50%

Generate 10 data points using the code below.

```
# Seed for reproducibility
np.random.seed(42)

# Generate 10 random x values within a range
x_generated = np.linspace(0, 5, 10)

# Parameters for the function (can use the previously fitted values or set randomly)
n_true = 0.06
a_true = 0.25
m_true = 0.57
b_true = 0.11

# Generate corresponding y values based on the function with added noise
noise = 0.001 * np.random.normal(0, 0.1, size=x_generated.shape) # Add Gaussian noise
y_generated = n_true * np.exp(-a_true * (m_true * x_generated + b_true) ** 2) + noise

# Display the generated x and y arrays
x_generated, y_generated
```

In this problem, we want to fit the data to the linear equation:

$$y = mx + b$$

using gradient descent and backpropagation.

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

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

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

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

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

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

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

where η is the learning rate.

For this assignment, use the following hyperparameters:

- **Epochs:** 10000
- **Learning Rate:** 0.001

Using the provided code and the above equations, perform gradient descent and backpropagation to fit the data to the linear model.

The expected output is the optimized values of m (slope) and b (intercept) that minimize the loss. These values can be compared to the true data for evaluation.

Include the following information in your report.

(a) **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.

(b) **Experiment with the learning rate and number of epochs:** Try different values for the learning rate (η) and the number of epochs to observe their effect on the model's performance. Find the best combination that minimizes the loss while ensuring the model converges. In your report, describe the effect of varying the learning rate and number of epochs. Discuss how these changes impact the convergence of gradient descent and the final loss.

2. 50%

In this problem, we want to use the same data as Problem 1 and fit the data to the nonlinear equation:

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

using gradient descent and backpropagation.

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

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

The gradients of the loss function with respect to n, a, m, and b are:

$$\frac{\partial \text{Loss}}{\partial n} = -\frac{2}{N} \sum_{i=1}^{N} (y_i - n \cdot \exp(-a \cdot y_{\text{int}})) \cdot \exp(-a \cdot y_{\text{int}})$$

$$\frac{\partial \text{Loss}}{\partial a} = \frac{2}{N} \sum_{i=1}^{N} (y_i - n \cdot \exp(-a \cdot y_{\text{int}})) \cdot n \cdot \exp(-a \cdot y_{\text{int}}) \cdot (-y_{\text{int}})$$

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

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

Using gradient descent, we update the parameters n, a, m, and b as follows:

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

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

where η is the learning rate.

For this assignment, use the following hyperparameters:

- **Epochs:** 10000
- **Learning Rate:** 0.001

Using the provided code and the above equations, perform gradient descent and backpropagation to fit the data to the nonlinear model.

The expected output is the optimized values of n, a, m, and b that minimize the loss. These values can be compared to the true data for evaluation.

Include the following information in your report.

(a) **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.

(b) **Experiment with the learning rate and number of epochs:** Try different values for the learning rate (η) and the number of epochs to observe their effect on the model's performance. Find the best combination that minimizes the loss while ensuring the model converges. In your report, describe the effect of varying the learning rate and number of epochs. Discuss how these changes impact the convergence of gradient descent and the final loss.