Tutorial: Gradient Descent for Exponential Function Data Fitting

Gradient descent is a powerful optimization technique used for curve fitting problems. In this tutorial, we illustrate how to use gradient descent clearly and explicitly to fit an exponential model to data.

1 Problem Formulation

We want to fit an exponential model of the form:

$$y = ae^{bx}$$

to a given set of data points $\{(x_i, y_i)\}_{i=1}^N$. The parameters a and b are unknown and must be estimated from data.

2 Formulating the Objective Function

We use the Mean Squared Error (MSE) as our objective function:

$$L(a, b) = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - ae^{bx_i}\right)^2$$

The goal is to find parameters a, b that minimize $L(a, b)$.

3 Compute the Gradient

We apply gradient descent, which requires computing the partial derivatives of the loss function $L(a, b)$ with respect to parameters a and b:

$$\nabla L(a, b) = \begin{bmatrix} \frac{\partial L}{\partial a} \\ \frac{\partial L}{\partial b} \end{bmatrix}$$

Compute these explicitly:

$$\frac{\partial L}{\partial a} = \frac{1}{N}\sum_{i=1}^{N} 2\left(y_i - ae^{bx_i}\right)\left(-e^{bx_i}\right)$$

$$= -\frac{2}{N}\sum_{i=1}^{N}(y_i - ae^{bx_i})e^{bx_i}$$

$$\frac{\partial L}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} 2(y_i - ae^{bx_i})(-ax_i e^{bx_i})$$

$$= -\frac{2a}{N}\sum_{i=1}^{N}(y_i - ae^{bx_i})x_i e^{bx_i}$$

Therefore, the gradient vector is:

$$\nabla L(a,b) = \begin{bmatrix} -\frac{2}{N}\sum_{i=1}^{N}(y_i - ae^{bx_i})e^{bx_i} \\ -\frac{2a}{N}\sum_{i=1}^{N}(y_i - ae^{bx_i})x_i e^{bx_i} \end{bmatrix}$$

4 Gradient Descent Update Rule

We use these gradients to iteratively update parameters a and b:

$$a^{(k+1)} = a^{(k)} - \alpha\frac{\partial L}{\partial a}(a^{(k)}, b^{(k)})$$

$$b^{(k+1)} = b^{(k)} - \alpha\frac{\partial L}{\partial b}(a^{(k)}, b^{(k)})$$

where α is the learning rate.

5 Numerical Example (Explicit)

Consider fitting the following data points ($N = 3$):

$$(0, 1.05), \quad (1, 2.85), \quad (2, 7.90)$$

We start with initial guesses $a^{(0)} = 1$, $b^{(0)} = 0.5$ and choose a learning rate $\alpha = 0.01$.

Iteration 1:

Compute terms first:

For $a = 1$, $b = 0.5$:

$$ae^{bx_1} = 1 \cdot e^{0.5 \cdot 0} = 1.000, \quad y_1 - ae^{bx_1} = 1.05 - 1 = 0.05$$

$$ae^{bx_2} = 1 \cdot e^{0.5 \cdot 1} = 1.6487, \quad y_2 - ae^{bx_2} = 2.85 - 1.6487 = 1.2013$$

$$ae^{bx_3} = 1 \cdot e^{0.5 \cdot 2} = 2.7183, \quad y_3 - ae^{bx_3} = 7.90 - 2.7183 = 5.1817$$

Then gradients are:

$$\frac{\partial L}{\partial a} = -\frac{2}{3}\left[0.05 \cdot 1.000 + 1.2013 \cdot 1.6487 + 5.1817 \cdot 2.7183\right] \approx -11.074$$

$$\frac{\partial L}{\partial b} = -\frac{2 \cdot 1}{3}\left[0.05 \cdot 0 \cdot 1.000 + 1.2013 \cdot 1 \cdot 1.6487 + 5.1817 \cdot 2 \cdot 2.7183\right] \approx -20.060$$

Update parameters:

$$a^{(1)} = 1 - 0.01(-11.074) = 1.1107$$

$$b^{(1)} = 0.5 - 0.01(-20.060) = 0.7006$$

Repeat these iterations until convergence.

6 Convergence Criteria

Gradient descent stops when the magnitude of the gradient vector is less than a threshold ϵ, e.g.:

$$\|\nabla L(a, b)\| < 10^{-6}$$

7 Interpretation and Practical Considerations

- Exponential fitting is nonlinear; thus gradient descent convergence can depend strongly on initial parameter guesses.
- Learning rate (α) choice is crucial. Too large can cause divergence; too small slows convergence.
- It may help to experiment with different initial guesses or normalize the data.

8 Summary

Gradient descent for exponential fitting involves clear steps:

1. Define exponential model and loss function explicitly.

2. Compute gradients carefully.
3. Iteratively update parameters using gradient descent.
4. Repeat until convergence criteria are satisfied.

Practicing this method with varied datasets and initial values will build intuition and understanding of nonlinear optimization techniques.