

locally weighted linear regression in detail with a solved example having mesh network with input layer, hidden layer and output layer

ChatGPT

Locally Weighted Linear Regression (LWLR) is a non-parametric regression technique that fits a linear regression model to the data points, giving more weight to the points in the vicinity of the query point (where the prediction is being made). The idea is to provide more relevance to the nearby data points in making predictions, allowing the model to adapt to the local behavior of the data.

The weighted linear regression model is given by:

$$y = X\beta + \epsilon$$

where:

- y is the response variable.
- X is the matrix of input features.
- β is the vector of coefficients.
- ϵ is the error term.



In LWLR, a weight function $w(i)$ is introduced, which assigns weights to the data points based on their proximity to the query point. The model parameters are estimated by minimizing the weighted sum of squared residuals:

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n w(i) \cdot (y_i - X_i \beta)^2$$

The weight function $w(i)$ typically assigns higher weights to points closer to the query point and lower weights to points farther away. Common choices for the weight function include Gaussian and bi-square functions.

Now, let's consider an example related to a mesh network with an input layer, a hidden layer, and an output layer. In a neural network context, we will use LWLR to make predictions for a specific data point in the input layer.

Let's assume the input layer has two features (x_1 and x_2), the hidden layer has three neurons, and the output layer has one neuron.

Let's go through the steps of Locally Weighted Linear Regression (LWLR) for the provided data and query point:

1. **Input Data:**

Input Layer (features)	Output Layer (response)
(x_{1i}, x_{2i})	y_i
(2, 3)	8
(1, 4)	5
(3, 1)	12

2. **Query Point:**

- Query Point: $(x_{1q}, x_{2q}) = (2, 2)$.

3. **Weight Calculation:**

- Choose a weight function, such as the Gaussian weight function:

$$w_i = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_q\|^2}{2\tau^2}\right)$$

where \mathbf{x}_i is the feature vector of a training point, \mathbf{x}_q is the feature vector of the query point, and τ is a bandwidth parameter.



- Calculate the weights for each training point based on the chosen weight function and

For the given example, let's assume $\tau = 1$:

- Weight for the first training point (2, 3):

$$w_1 = \exp\left(-\frac{(2-2)^2 + (3-2)^2}{2 \cdot 1^2}\right) = \exp(-1) \approx 0.3679$$

- Weight for the second training point (1, 4):

$$w_2 = \exp\left(-\frac{(1-2)^2 + (4-2)^2}{2 \cdot 1^2}\right) = \exp(-5) \approx 0.0067$$

- Weight for the third training point (3, 1):

$$w_3 = \exp\left(-\frac{(3-2)^2 + (1-2)^2}{2 \cdot 1^2}\right) = \exp(-2) \approx 0.1353$$

4. Weighted Linear Regression:

- The weighted linear regression model is given by:

$$\hat{y}_q = \mathbf{X}_q \hat{\beta}_q$$

- The weighted sum of squared residuals is minimized to find $\hat{\beta}_q$:

$$\hat{\beta}_q = \arg \min_{\beta_q} \sum_{i=1}^n w_i \cdot (y_i - \mathbf{X}_i \beta_q)^2$$

For simplicity, let's assume a simple linear regression model ($y = \beta_0 + \beta_1 x$).

- For the first training point (2, 3):

$$\mathbf{X}_1 = [1, 2], \quad \hat{\beta}_{1q} = \arg \min_{\beta_{1q}} w_1 \cdot (8 - (\beta_{0q} + \beta_{1q} \cdot 2))^2$$

- For the second training point (1, 4):

$$\mathbf{X}_2 = [1, 1], \quad \hat{\beta}_{2q} = \arg \min_{\beta_{2q}} w_2 \cdot (5 - (\beta_{0q} + \beta_{2q} \cdot 1))^2$$

- For the third training point (3, 1):

$$\mathbf{X}_3 = [1, 3], \quad \hat{\beta}_{3q} = \arg \min_{\beta_{3q}} w_3 \cdot (12 - (\beta_{0q} + \beta_{3q} \cdot 3))^2$$

Let's find the values of $\hat{\beta}_{0q}$ and $\hat{\beta}_{1q}$ for each training point using the provided data

For the first training point (2, 3):

- Input features: $\mathbf{X}_1 = [1, 2]$
- Weighted linear regression minimization problem:

$$\hat{\beta}_{1q} = \arg \min_{\beta_{1q}} \exp(-1) \cdot (64 - 32(\beta_{0q} + \beta_{1q}) + 4\beta_{1q}^2)$$

- Simplifying and solving:

$$\hat{\beta}_{1q} = \frac{4 \cdot \exp(-1)}{1 + 2 \cdot \exp(-1)}$$

For the second training point (1, 4):

- Input features: $\mathbf{X}_2 = [1, 1]$
- Weighted linear regression minimization problem:

$$\hat{\beta}_{2q} = \arg \min_{\beta_{2q}} \exp(-5) \cdot (25 - 10(\beta_{0q} + \beta_{2q}) + \beta_{2q}^2)$$

- Simplifying and solving:

$$\hat{\beta}_{2q} = \frac{\exp(-5)}{1 + \exp(-5)}$$

For the third training point (3, 1):

- Input features: $\mathbf{X}_3 = [1, 3]$
- Weighted linear regression minimization problem:
$$\hat{\beta}_{3q} = \arg \min_{\beta_{3q}} \exp(-2) \cdot (144 - 72(\beta_{0q} + \beta_{3q}) + 9\beta_{3q}^2)$$
- Simplifying and solving:
$$\hat{\beta}_{3q} = \frac{9 \cdot \exp(-2)}{1 + 3 \cdot \exp(-2)}$$

These are the values of $\hat{\beta}_{1q}$, $\hat{\beta}_{2q}$, and $\hat{\beta}_{3q}$ specific to the query point. The detailed calculations involve algebraic manipulations and solving equations, and the final results are obtained by simplifying and solving these equations.