Gradient descent: Takeaways

Concepts

- It's important to scale or normalize your data before using the data in machine learning algorithms, as features on different scales could produce bias into machine learning algorithms.
- Gradient descent is a general method that can be used to estimate coefficients of nearly any model. Gradient descent minimizes the residuals in the estimated model by updating its gradient.
- Cost functions measure the difference between a model's predictions and its corresponding observations with the coefficients as parameters. For our model $h_\theta(x) = \theta_1 x + \theta_0$, the cost function is defined as:
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$
- Gradient descent relies on finding the direction of the largest gradient where a gradient is the "slope" of a multivariable function. To find the gradient, we can take the partial derivative in terms of each parameter in the cost function.
- A partial derivative represents the slope of a multivariable function in terms of a single variable. Mathematically, a partial derivative is denoted using ∂.
    - For example, $\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$ is read as the partial derivative of $J(\theta_0, \theta_1)$ in terms of θ_0.
- Partial derivatives of the cost function:
    - In terms of θ_0: $\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$.
    - In terms of θ_1: $\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) * x_i$.
- Gradient descent is a widely used method to find the optimal parameters.
- Gradient descent algorithm for two variables:
    - Repeat until convergence { $\theta_1 := \theta_1 - \alpha * \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}$ $\theta_0 := \theta_0 - \alpha * \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$ }.
    - In the algorithm above, θ_n is the current value of our coefficient, α is the learning rate, $\frac{\partial J(\theta_0)}{\partial \theta_1}$ is the partial derivative of our cost function in terms of θ_0, and $\frac{\partial J(\theta_1)}{\partial \theta_1}$ is the partial derivative of our cost function in terms of θ_1.
- In general, learning rates can be between 0.0001 and 1. If the rate is too large, the algorithm will overshoot the minimum; if it's too small, it will take many iterations to converge.

Resources

- Gradient descent
- Partial derivative