DeepLearning

Ananya Rao

October 2023

Supervised Learning

There are two types of methods used in supervised learning.

Regression

Linear Regression

Uni-variate Linear Regression

We define a function f(linear in this case) with parameters w and b. This function is trained using data called the training set. To determine the extent of fitting of this function, we define a cost function J(w,b).

Model

$$f(x_i) = wx_i + b$$

Parameters

$$w, b$$

Cost Function(Mean Squared Error)

$$J(w,b) = \frac{1}{2n}\sum_{i=1}^{n}(f(x_i) - y_i)^2$$

Objective is to minimize

$$J(w,b) \forall w, b$$

Gradient descent algorithm is widely used to calculate the optimal w and b for the cost function.

$$w = w - \alpha\frac{\partial J(w,b)}{\partial w}$$

$$b = b - \alpha\frac{\partial J(w,b)}{\partial b}$$

where α is the **learning rate**.

if the learning rate is too small...

- The algorithm will work but will be very slow.

if the learning rate is too big...

- The algorithm might not achieve the minimum ever. It may converge or diverge.

Multiple Linear Regression

In such regression, there are multiple parameters which affect the result so we choose a model that accounts for this dependence.

Model

$$f(\vec{x}) = \vec{w} \cdot \vec{x} + b = w_1x_1 + w_2x_2 + ... + w_nx_n + b$$

Gradient Descent in detail

In a multiple regression model, different parameters multiplied with their coefficients can take different ranges of values. Significantly differing ranges might result in a narrow cost function graph which will make the algorithm slower.
To deal with this, we scale the features so that all of them lie in comparable ranges such that our cost function graph becomes more like a circle.

Feature Re-scaling

- We divide parameters by their range difference such that their scaled values vary from -1 to 1 at max.
- One way is to use mean

$$x_{scaled} = \frac{x - \mu}{x_{max} - x_{min}}$$

- Another method is to use Normal Distribution.

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

Learning Curve
Plot of the cost function(with modified w and b as input at each iteration) against the number of iterations.
For a good learning rate, the graph will decrease and eventually become flat. For a bad learning rate, the graph might show strange behaviour like increasing-decreasing, or like a parabola.

Choosing a good learning rate
If the learning curve is unusual, the α should be decreased.
Smaller α leads to the expected result but very slowly, so alpha should be increased gradually until anomalies kick in.

Logistic Regression

Unlike Linear Regression, we use logistic regression in those cases where the label can take values from a finite set. To achieve this, we use a sigmoid function to model our problem.

Model

$$f(\vec{x}) = \frac{1}{1+e^{-z}}, 0 < g(z) < 1$$

where $z = \vec{w} \cdot \vec{x} + b$. *Note: z can be a polynomial as well.*
We can interpret this model as the probability that the given input has the label "1". Therefore, we define a threshold and if the model takes a value less than the threshold, we assign value "0" to it and "1" otherwise.
Decision boundary - The curve obtained on setting $f(\vec{x})$ = threshold. $f(\vec{x})$ takes a value greater than the threshold for the values at one side of this curve and less than the threshold for the values at the other side.
We get complex boundaries when z is a polynomial, which is based on requirements.

Cost Function
We will make some modifications to the cost function defined for Linear Regression. First of all, we will define *Loss Function*. *Loss Function* is the loss caused by each data value.
In linear regression, $L(f(\vec{x}), y) = (f(x) - y)^2$. In logistic regression, the loss function is defined as

$$L(f(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -log(f(\vec{x}^{(i)})) & \text{if } y^{(i)} = 1 \\ -log(1 - f(\vec{x}^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

We can write it in a simpler way

$$L(f(\vec{x}^{(i)}), y^{(i)}) = -y^{(1)} log(f(\vec{x}^{(i)})) - (1 - y^{(1)} log(1 - f(\vec{x}^{(i)})))$$

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{n} [L(f(\vec{x}^{(i)}), y^{(i)})]$$

This particular cost function is preferred for logistic regression based on statistical grounds. It is estimated using maximum likelihood estimation.

Minimizing the Cost Function - GDM
The same algorithm is applied here as well. Well, the derivative functions look the same for both the regressions in terms of $f(\vec{x})$ but actually they are different as $f(\vec{x})$ are very different for both methods.

Overfitting It is possible that the chosen model with estimated parameters overfit/underfit the training set. Both of them show bias in predicting values of new incoming data.

- **Underfit(high bias)** - does not fit the training set well.
- **Overfit(high variance)** - fits the training set extremely well

In Machine Learning, we want to define a model which is neither underfitting nor overfitting

Classification