Neural Networks and Deep Learning Assignment

Student Name

Institution Affiliation

Course

Tutor Name

Date

Neural Networks and Deep Learning Assignment

Probability p as a function of x:

Any real-valued inputcan be mapped to the range of 0 to 1 by the logistic function. The probability p that the output f(x) is 1 (i.e., the positive class) for an input vector (x) and weight vector (w) is as follows: [p = \frac{1}{1 + e^{-z}}] where the linear combination of weights and features is represented by (z = w^T x).

Derivative of log p with respect to each weight (w_i):

The derivative of (\log p) for each weight (w_i) is going to be computed as follows:

[\frac{\partial}{\partial w_i} \left(\log \frac{1}{1 + e^{-z}}\right) = \frac{\partial}{\partial w_i}]

The chain rule gives us: [\frac{\partial \log p}{\partial w_i} = \frac{1}{p} \frac{\partial p}{\partial w_i} = \frac{1}{p} p (1 - p) x_i = (1 - p) x_i]

Derivative of log(1-p) with respect to each weight (w_i):

we are going to determine the derivative of (\log(1-p)) for every weight (w_i):

[\frac{\partial}{\partial w_i} \left(\log \frac{e^{-z}}{1 + e^{-z}}\right) \frac{\partial \log(1-p)}{\partial w_i}]

Again using the chain rule: [\frac{\partial\log(1-p)}]{\partial (1-p)} \frac{\partial w_i} = \frac{1}{1-p}{\partial w_i} = -\frac{1}{1-p} p (1 - p) x_i = -p x_i]

Learning rule for minimizing negative-log-likelihood loss:

It is as follows: [L(w) = -\sum_{i=1}^{N} \left(y_i \log p_i + (1 - y_i) \log (1 - p_i)\right)] where the true label (0 or 1) for the (i)-th case is denoted by (y_i).

The loss gradient with relation to weights (w_i) is going to be expressed as: [$\frac{\partial L}{\partial w_i} = -(y_i - p_i) x_i$]

For updating weights, the learning rule (gradient descent) is as follows: [$w_i \leftarrow w_i + \alpha (y_i - p_i) x_i$] When learning rate (α) is expressed.

Resemblance to other learning rules:

The gradient descent update utilized in other machine learning methods, such as neural networks and linear regression, is similar to the learning rule for reducing negative-log-likelihood loss.

It seeks to modify weights according to the discrepancy between true labels and expected probability.

The model is encouraged to improve its predictions by the negative-log-likelihood loss, which reduces the difference between expected and actual outcomes.