Loss Functions Notes

2024-08-20

REGRESSION LOSS FUNCTIONS

1. Mean Squared Error (MSE)

- **Formula:** $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
    - Here, y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of data points. The differences between actual and predicted values are squared to emphasize larger errors more significantly than smaller ones.
- **When to Use It:** It's best for situations where you want to ensure that your model has very few large errors, making it suitable for regression tasks without outliers since the squaring will unduly increase the influence of outliers.

2. Mean Absolute Error (MAE)

- **Formula:** $\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
    - The absolute value $|y_i - \hat{y}_i|$ means you take the positive value of the difference, regardless of whether the prediction was too high or too low.
- **When to Use It:** Use MAE if you need a straightforward measure that gives an average level of error in your predictions. It's particularly useful when dealing with outliers because it doesn't square the errors, thus not exaggerating their impact.

3. Huber Loss

- **Formula:**

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

    - a is the error $y_i - \hat{y}_i$, and δ is a threshold parameter that you choose. For errors smaller than δ, the loss is quadratic, and for larger errors, it's linear.
- **When to Use It:** This is effective when you want to combine the benefits of MSE and MAE. It handles outliers like MAE (by being linear for large errors) but remains sensitive to small errors like MSE.

CLASSIFICATION LOSS FUNCTIONS

4. Binary Cross-Entropy

- **Formula:** $-[y\log(p) + (1-y)\log(1-p)]$
  - y is the actual class label (0 or 1), and p is the predicted probability of the class label being 1. This formula calculates the logarithm of the predicted probability, directly linking the prediction accuracy to the loss.
- **When to Use It:** It's used in binary classification tasks, especially suitable when the output of your model is a probability that needs to be as close as possible to the actual class label.

5. Categorical Cross-Entropy

- **Formula:** $-\sum_{c=1}^{M} y_{o,c}\log(p_{o,c})$
- **Use Case:** Ideal for multi-class classification tasks where each instance belongs exclusively to one class out of M classes.
- **Advantages:** Handles multiple classes efficiently, ensuring model outputs a probability distribution across all classes.
- **Drawbacks:** Requires the output layer to use a softmax function to ensure output probabilities sum to 1. The model must produce a full probability distribution for each class, which can be computationally intensive.

6. Hinge Loss

- **Formula:** $\max(0, 1 - y \times \hat{y})$
  - Here, y represents the actual class label, which should be +1 or -1 (not 0 or 1), and $\hat{y}$ is the predicted value, not necessarily bounded between +1 and -1.
- **When to Use It:** Commonly used in training classifiers, especially Support Vector Machines. Hinge loss is valuable when you want a large margin of error between classes, which helps to ensure that future data points are classified correctly even if they are not exactly on the correct side of the decision boundary. These loss functions play critical roles in the learning algorithm by guiding how the model weights should be adjusted to reduce errors between actual and predicted values. Each loss function has its particular strengths and is chosen based on the specific requirements and characteristics of the data you are working with.