Loss Functions in Deep Learning

Mayank Pratap Singh

August 5, 2024

1 Introduction to Loss Functions

A loss function measures how well a neural network's prediction matches the true data. It's a single number that represents the cost associated with the prediction errors. The goal of training a neural network is to minimize this loss.

1.1 Loss Function vs Cost Function

While a loss function computes the error for a single training example, a cost function is the average error over the entire training set.

2 Regression Loss Functions

2.1 Mean Squared Error (MSE)

MSE is the average of the squared differences between the predicted and actual values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Advantages:

- Easy to interpret.
- Differentiable, allowing the use of gradient descent.
- Single local minimum.

Disadvantages:

- Not robust to outliers.

Example: For $y_i = 6.3$ and $\hat{y}_i = 6.1$,

$$\text{MSE} = (6.3 - 6.1)^2 = 0.04$$

2.2 Mean Absolute Error (MAE)

MAE is the average of the absolute differences between the predicted and actual values.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Advantages:

- Intuitive and easy to understand.
- Robust to outliers.

Disadvantages:

- Not differentiable at zero.

Example: For $y_i = 6.3$ and $\hat{y}_i = 6.1$,

$$\text{MAE} = |6.3 - 6.1| = 0.2$$

2.3 Huber Loss

Huber loss is a combination of MSE and MAE, less sensitive to outliers than MSE.

$$L_\delta(a) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \delta \\ \delta|y - \hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

Example: For $y_i = 6.3$ and $\hat{y}_i = 6.1$ with $\delta = 1$,

$$L_\delta = \frac{1}{2}(6.3 - 6.1)^2 = 0.02$$

3 Classification Loss Functions

3.1 Binary Cross Entropy

Used for binary classification problems.

$$\text{BCE} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Example: For $y = 1$ and $\hat{y} = 0.8$,

$$\text{BCE} = -[1 \log(0.8) + (1 - 1) \log(1 - 0.8)] = 0.223$$

3.2 Categorical Cross Entropy

Used for multi-class classification.

$$L = -\sum_{j=1}^{k} y_j \log(\hat{y}_j)$$

Example: For $y = [1, 0, 0]$ and $\hat{y} = [0.7, 0.2, 0.1]$,

$$L = -[1 \log(0.7) + 0 \log(0.2) + 0 \log(0.1)] = 0.357$$

3.3 Sparse Categorical Cross Entropy

Similar to Categorical Cross Entropy but without one-hot encoding.

$$L = -\log(\hat{y}[y])$$

Example: For $y = 0$ and $\hat{y} = [0.7, 0.2, 0.1]$,

$$L = -\log(0.7) = 0.357$$

4 Other Loss Functions

4.1 KL Divergence for Autoencoders

$$D_{KL}(P \| Q) = \sum_{i} P(i) \log \frac{P(i)}{Q(i)}$$

Example: For distributions $P = [0.4, 0.6]$ and $Q = [0.5, 0.5]$,

$$D_{KL} = 0.4 \log \frac{0.4}{0.5} + 0.6 \log \frac{0.6}{0.5} = 0.018$$

4.2 Discriminator Loss for GANs

$$L_D = -\frac{1}{2}[\log(D(x)) + \log(1 - D(G(z)))]$$

Example: For $D(x) = 0.9$ and $D(G(z)) = 0.1$,

$$L_D = -\frac{1}{2}[\log(0.9) + \log(1 - 0.1)] = 0.105$$

4.3 Focal Loss for Object Detection

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t)$$

Example: For $p_t = 0.9$, $\alpha_t = 0.25$, and $\gamma = 2$,

$$FL = -0.25(1 - 0.9)^2 \log(0.9) = 0.001$$

4.4 Embedding Triplet Loss

$$L = \max(d(a, p) - d(a, n) + \alpha, 0)$$

Example: For $d(a, p) = 0.5$, $d(a, n) = 1.0$, and $\alpha = 0.2$,

$$L = \max(0.5 - 1.0 + 0.2, 0) = 0$$

5 Creating Custom Loss Functions

Custom loss functions can be created by defining the desired behavior mathematically and implementing it in a deep learning framework.

Example: Suppose we want to penalize large errors more severely. We could use a loss function like:

$$L = \sum_{i=1}^{n} \log(1 + (y_i - \hat{y}_i)^2)$$

Explanation: This loss function grows logarithmically with the square of the error, reducing the impact of very large errors compared to MSE.

6 Conclusion

Each loss function has its own strengths and weaknesses, and the choice depends on the specific problem and data characteristics. Understanding the intuition and mathematics behind these loss functions is crucial for effectively applying them in deep learning.