03 Loss Function

- In machine learning, a **loss function** (also known as a cost function or objective function) quantifies how well a model's predictions match the actual target values
- It measures the discrepancy between the predicted values and the true values
- The goal of training a machine learning model is to minimize the loss function, thereby improving the model's accuracy
- The optimization algorithm (like gradient descent) uses the loss function to update the model's parameters, aiming to minimize the loss

Types of Loss Functions

Different types of loss functions are used depending on the problem (regression or classification) and the specific characteristics of the model

01 Loss Functions for Regression

1. **Mean Squared Error (MSE)**
   $\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$
    - MSE calculates the average of the squared differences between the predicted and actual values
    - It's the most commonly used loss function for regression problems
    - MSE penalizes larger errors more heavily than smaller ones due to the squaring operation

```
from sklearn.metrics import mean_squared_error
```

```
def error(X,y,w):
        n = X.shape[0]
        e = 0
        for i in range(n):
                y_i = model(X[i],w)
                e = e + (y[i] - y_i)**2
        return e/(2*m)
```

2. **Mean Absolute Error (MAE)**
   $\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|$
    - MAE calculates the average of the absolute differences between the predicted and actual values

- It's less sensitive to outliers than MSE
- MAE gives equal weight to all errors

3. **Huber Loss**
   - Huber loss is a combination of MSE and MAE
   - It is quadratic for small errors and linear for large errors
   - It's robust to outliers while still being sensitive to small errors
   - Balances sensitivity to outliers and small errors

02 Loss Functions for Classification

1. **Binary Cross-Entropy (Log Loss)**
   $\text{Log Loss} = -\frac{1}{m}\sum_{i=1}^{m}[y_i\log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$
   - Used for binary classification, log loss penalizes incorrect predictions
   - It measures the difference between the predicted probability and the actual label (0/1)
   - Highly sensitive to confident wrong predictions
   - Standard choice for logistic regression and binary classification tasks
2. **Categorical Cross-Entropy**
   $\text{Categorical Cross-Entropy} = -\sum_{i=1}^{m}\sum_{j=1}^{k} y_{ij}\log(\hat{y}_{ij})$
   - Used for multi-class classification problems
   - It calculates the cross-entropy loss between the predicted probability distribution and the true distribution
   - Generalization of binary cross-entropy for multiple classes
3. **Hinge Loss**
   $L(y, \hat{y}) = \max(0, 1 - y \cdot \hat{y})$
   - Used primarily for training Support Vector Machines (SVMs)
   - Hinge loss penalizes predictions that are on the wrong side of the decision boundary or within a margin
   - Encourages a large margin between classes
4. **Kullback-Leibler (KL) Divergence**
   $D_{KL}(P||Q) = \sum_i P(i)\log\left(\frac{P(i)}{Q(i)}\right)$
   - KL Divergence measures how one probability distribution diverges from a second, expected probability distribution
   - It's often used in models like Variational Autoencoders (VAEs)
   - Sensitive to changes in distribution