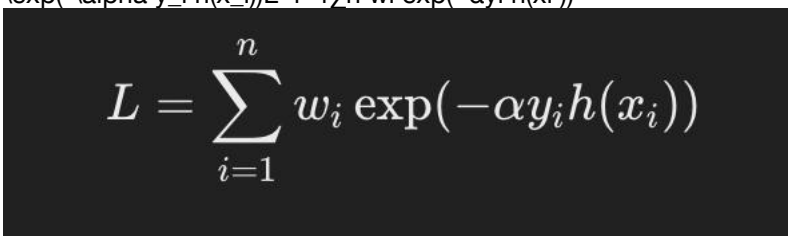
2. Math Intro

- **Mathematical Principles:** AdaBoost (Adaptive Boosting) is an ensemble learning algorithm that combines multiple weak classifiers to form a strong classifier. It adjusts weights iteratively, emphasizing misclassified samples to improve subsequent classifiers.

Key Equation:

The exponential loss function: L=∑i=1nwiexp (−αyih(xi))L = \sum_{i=1}^{n} w_i \exp(-\alpha y_i h(x_i))L=i=1∑n wi exp(−αyi h(xi))



$$L = \sum_{i=1}^{n} w_i \exp(-\alpha y_i h(x_i))$$

- Where:
  - wiw_iwi : sample weights
  - yiy_iyi : true labels
  - h(xi)h(x_i)h(xi): predictions by weak classifiers
  - α\alphaα: weight of each weak classifier.

3. Numerical Technique

- **Code Implementation:**
  - The AdaBoost algorithm is implemented using Python in an object-oriented manner.
  - Key methods include:
    - `train`: Iteratively trains weak classifiers and adjusts weights.
    - `predict`: Aggregates predictions from all weak classifiers.
    - `DecisionStump`: A simple weak classifier for binary classification tasks.
- **Testing Configuration:**
  - **Train Size:** 80
  - **Test Size:** 20
  - **Number of Estimators:** 50

4. Previous Work

- Freund and Schapire (1997): Introduced the AdaBoost algorithm and its theoretical foundation in "A decision-theoretic generalization of on-line learning."
- Pedregosa et al. (2011): Implementation of AdaBoost as part of the Scikit-learn machine learning library.

References

- Freund, Y., & Schapire, R. E. (1997). "A decision-theoretic generalization of on-line learning and an application to boosting." Journal of Computer and System Sciences, 55(1), 119-139.
- Pedregosa, F., et al. (2011). "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, 12, 2825-2830.

Summary

- AdaBoost successfully combines multiple weak classifiers to create a strong classifier.
- Misclassified samples are emphasized through iterative weight adjustments.
- Our implementation achieves a test accuracy of consistently above 80%, demonstrating its effectiveness.
- Future work could explore real-world datasets and more complex weak classifiers.