Logistic Regression

1. Hypothesis (Prediction)

$$\boxed{\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}} \qquad z = mX + c$$

- $\hat{y}$ = predicted probability of class 1
- Sigmoid function maps any real number to [0, 1]

2. Loss Function - Cross Entropy

$$\boxed{L = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \right]}$$

- Measures how far predicted probabilities are from actual labels (0 or 1)
- Smaller loss → better classifier

3. Gradient Descent - Parameter Updates

- Computes derivatives w.r.t parameters:

$$\boxed{\frac{\partial L}{\partial m} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i) X_i} \quad \boxed{\frac{\partial L}{\partial c} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)}$$

- Update rules:

$$\boxed{m := m - \alpha \frac{\partial L}{\partial m}} \quad \boxed{c := c - \alpha \frac{\partial L}{\partial c}}$$

Where α is the learning rate.

4. Final Prediction

- Predict probability:

$$\boxed{\hat{y} = \frac{1}{1 + e^{-(mX+c)}}}$$

- Convert probability to class label:

$$\boxed{class = \begin{cases} 1 & \text{if } \hat{y} \geq 0.5 \\ 0 & \text{if } \hat{y} < 0.5 \end{cases}}$$

---

5. Key Points

- Sigmoid ensures output is between 0 and 1 → interpretable as probability
- Cross-entropy penalizes wrong predictions heavily.
- Gradient descent updates slope and intercept to minimize loss
- Foundation for neural networks (sigmoid + backprop)