Cost Functions

Finds deviation between actual and predicted → evaluation of model

3 main types:
1. Regression Cost Function
2. Binary Classification Cost Function
3. Multi-Class Classification Cost Function

1. Regression Cost Function - Predicting continuous values

Variables: y - actual value, y' - predicted, n - # of points

1. Mean Error (ME)

$$ME = \frac{1}{n} \cdot \sum_{i=0}^{n} (y - y')$$

- positives and negatives can cancel each other out ⇒ not good

2. Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \cdot \sum_{i=0}^{n} |y - y'|$$

- MAE addresses pos/neg issues, but doesn't penalize outliers or show dramatic changes.

3. Mean Squared Error (MSE)

$$MSE = \frac{1}{n} \cdot \sum_{i=0}^{n} (y - y')^2$$

- MSE addresses pos/neg issues, and also exxagerates any error

2/3: Classification Cost Functions

Ex: Classifying Happy, Sad, Mad Faces

"Actual" Values:
- Happy [1, 0, 0] or
- Sad [0, 1, 0] or
- Mad [0, 0, 1]

ML model provides probability distribution: $[x_1, x_2, x_3]$

Cross Entropy Cost Function:

A = matrix of prob. distribution
B = matrix of actual value

$$C = -B^{T} \log(A) = -\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \log x_1 \\ \log x_2 \\ \log x_3 \end{bmatrix}$$

→ transverse

$$= -[\log x_1 + 0 + 0] = -\log x_1$$

Binary Cross Entropy - only 2 options

- Actual output, y, is either 1 or 0
- Predicted output is p, $0 \le p \le 1$

$$BCE = -\left(\underbrace{y \cdot \log(p)}_{} + \underbrace{(1-y) \cdot \log(1-p)}_{}\right)$$

calculates both cases $\nearrow y \rightarrow y=1$
$\searrow 1-y \rightarrow y=0$

Gradient Descent (G.D)

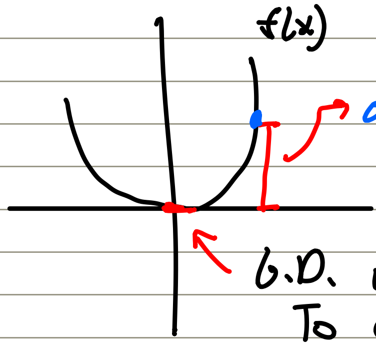
Ex on MSE:

$$E = \frac{1}{n}\sum_{i=0}^{n}(y-y')^2 = \frac{1}{n}\sum_{i=0}^{n}(y-(mx+b))^2 \longrightarrow$$ only variables to change:
- "m"
- "b"

since n is constant

Let $f(x) = x^2$

Ex: current MSE = blue point



current MSE

G.D. wants to reach here

To get there, consider 2 factors:
1. Direction to move
2. Magnitude of move

} must change m and b in error function

2 variables we can change are m, b

$$\frac{\partial E}{\partial m} = \frac{-2}{n}\sum_{i=0}^{n}(y-mx-b)x$$

$$\frac{\partial E}{\partial b} = \frac{-2}{n}\sum_{i=0}^{n}(y-mx-b)$$

$$m_{new} = m_{curr} \boxed{-}\boxed{L}\cdot\boxed{\frac{\partial E}{\partial m}}$$

$$b_{new} = b_{curr} \boxed{-}\boxed{L}\cdot\boxed{\frac{\partial E}{\partial b}}$$

→ direction is opposite of gradient
→ size of "step" to take