LOSS FUNCTIONS

Dr. Umarani Jayaraman

What is loss function?

- It provides a measure of how well the model is performing on training data (that includes validation data) with respect to its objective.
- A loss function, in the context of deep learning and optimization, is a measure of how well a model's predictions match the true values of the target variable in the dataset.

Various Loss Function

- Regression Loss Functions
  - Squared Error Loss
  - Absolute Error Loss
  - Huber Loss
- Binary Classification Loss Functions
  - Binary Cross-Entropy
  - Hinge Loss
- Multi-class Classification Loss Functions
  - Multi-class Cross Entropy Loss
  - Kullback Leibler Divergence Loss

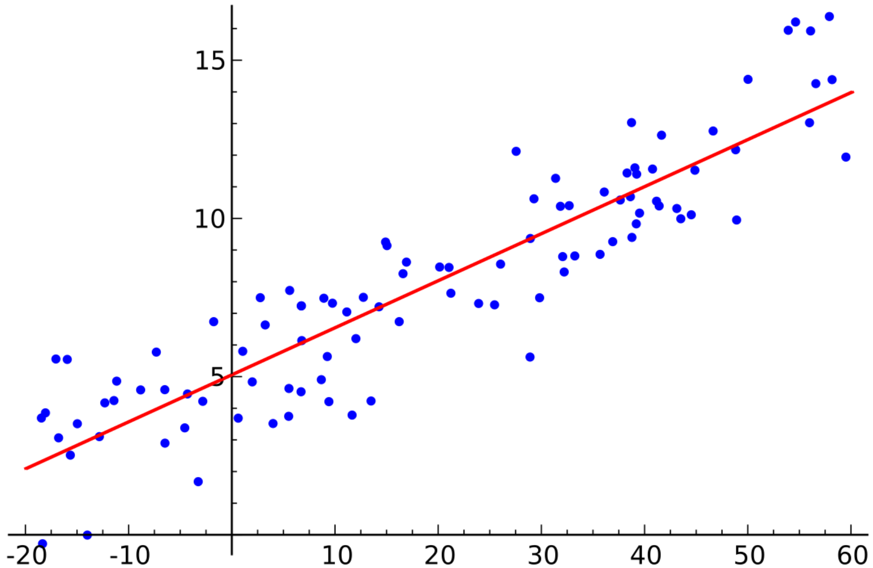
Various Loss Function

- 1. Regression
    1. MSE(Mean Squared Error)
    2. MAE(Mean Absolute Error)
    3. Hubber loss
- 2. Classification
    1. Binary cross-entropy
    2. Categorical cross-entropy
- 3. Auto-Encoder
    1. KL Divergence
- 4. GAN
    1. Discriminator loss
    2. Minmax GAN loss
- 5. Object detection
    1. Focal loss
- 6. Word embeddings
    1. Triplet loss

Regression Loss

- **Mean Squared Error/Squared loss/ L2 loss**

$$MSE = \frac{1}{N} \sum_{i}^{N} (Yi - \hat{Y}i)^2$$



Regression Loss

- **Mean Absolute Error/ L1 loss**

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |Y_i - \hat{Y}_i|$$

Classification Loss-
Binary Cross Entropy Loss

- **Let us start by understanding the term 'entropy'.**
- Generally, we use entropy to indicate disorder or uncertainty.
- It is measured for a random variable X with probability distribution p(X):

$$S = \begin{cases} -\int p(x).\log p(x).dx, & \text{if } x \text{ is continuous} \\ -\sum_x p(x).\log p(x), & \text{if } x \text{ is discrete} \end{cases}$$

Binary Cross Entropy Loss

- The negative sign is used to make the overall quantity positive.
- A greater value of entropy for a probability distribution indicates a greater uncertainty in the distribution.
- Likewise, a smaller value indicates a more certain distribution.

Binary Cross Entropy Loss

- This makes binary cross-entropy suitable as a loss function – **you want to minimize its value.**
- We use **binary cross-entropy** loss for classification models which output a probability *p*.

Probability that the element belongs to class 1 (or positive class) = p
Then, the probability that the element belongs to class 0 (or negative class) = 1 - p

The probability

Binary Cross Entropy Loss

- Then, the cross-entropy loss for output label **y** (can take values 0 and 1) and predicted probability **p** is defined as:

$$-\left[y \log(p) + (1-y)\log(1-p)\right]$$

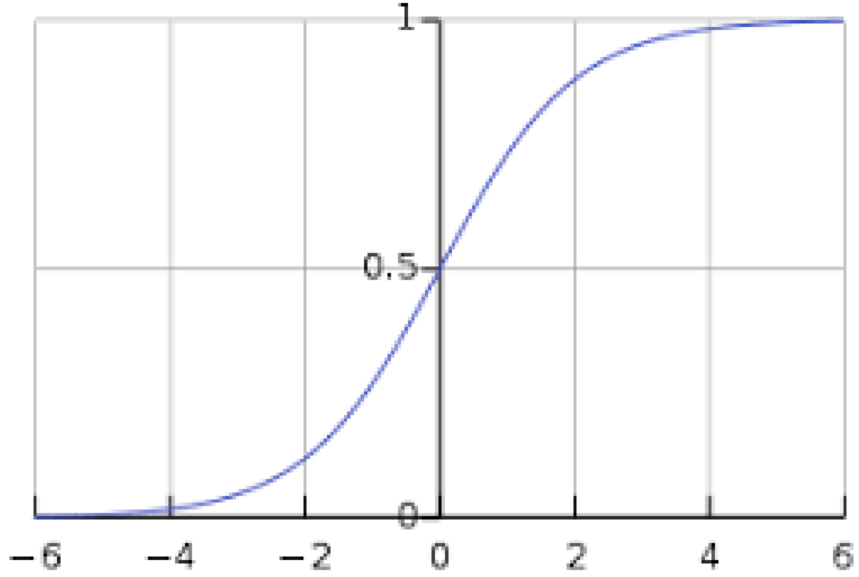
Binary Cross Entropy Loss

- This is also called **Log-Loss.**
- To calculate the probability **p**, we can use the sigmoid function. Here, z is a function of our input features:

$$S(z) = \frac{1}{1 + e^{-z}}$$

Binary Cross Entropy Loss

- The range of the sigmoid function is [0, 1] which makes it suitable for calculating probability.



Binary Cross Entropy Loss

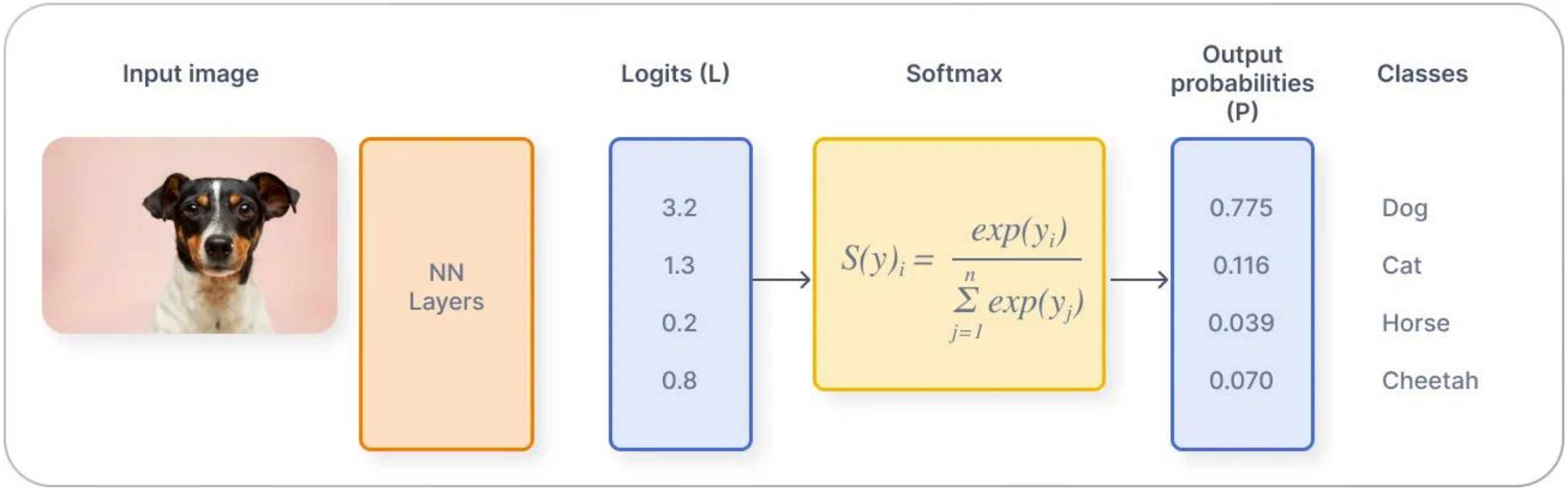
```
def update_weights_BCE(m1, m2, b, X1, X2, Y, learning_rate):
    m1_deriv = 0
    m2_deriv = 0
    b_deriv = 0
    N = len(X1)
    for i in range(N):
        s = 1 / (1 / (1 + math.exp(-m1*X1[i] - m2*X2[i] - b)))

        # Calculate partial derivatives
        m1_deriv += -X1[i] * (s - Y[i])
        m2_deriv += -X2[i] * (s - Y[i])
        b_deriv += -(s - Y[i])

    # We subtract because the derivatives point in direction of steepest ascent
    m1 -= (m1_deriv / float(N)) * learning_rate
    m2 -= (m2_deriv / float(N)) * learning_rate
    b -= (b_deriv / float(N)) * learning_rate

    return m1, m2, b
```

Multi-class cross-entropy/categorical cross-entropy

Multi-class Classification

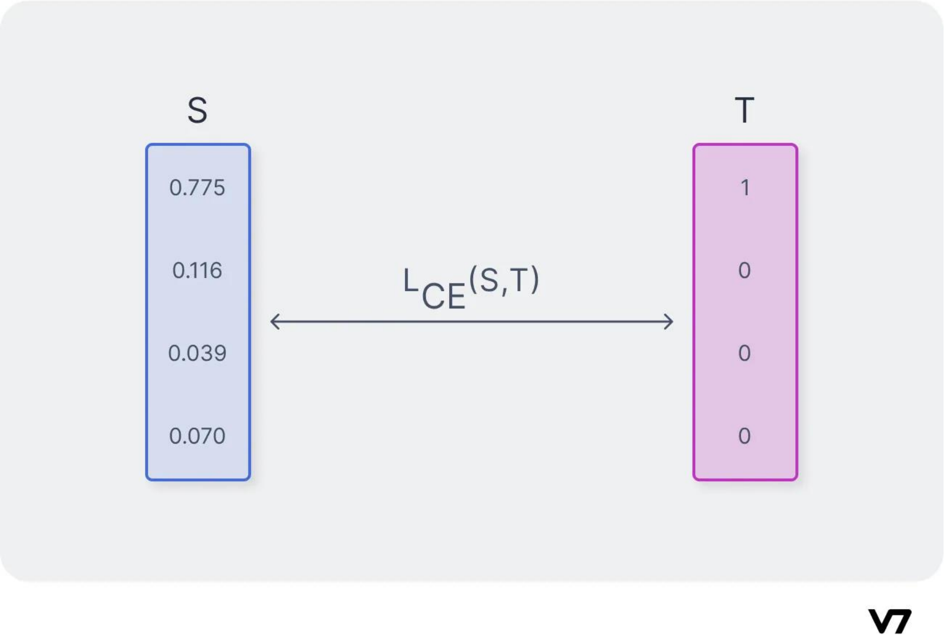
Class			
Class#	1	2	3
Class Label	0	1	2
One-hot encoding	[1, 0, 0]	[0, 1, 0]	[0, 0, 1]

Categorical Cross Entropy Loss

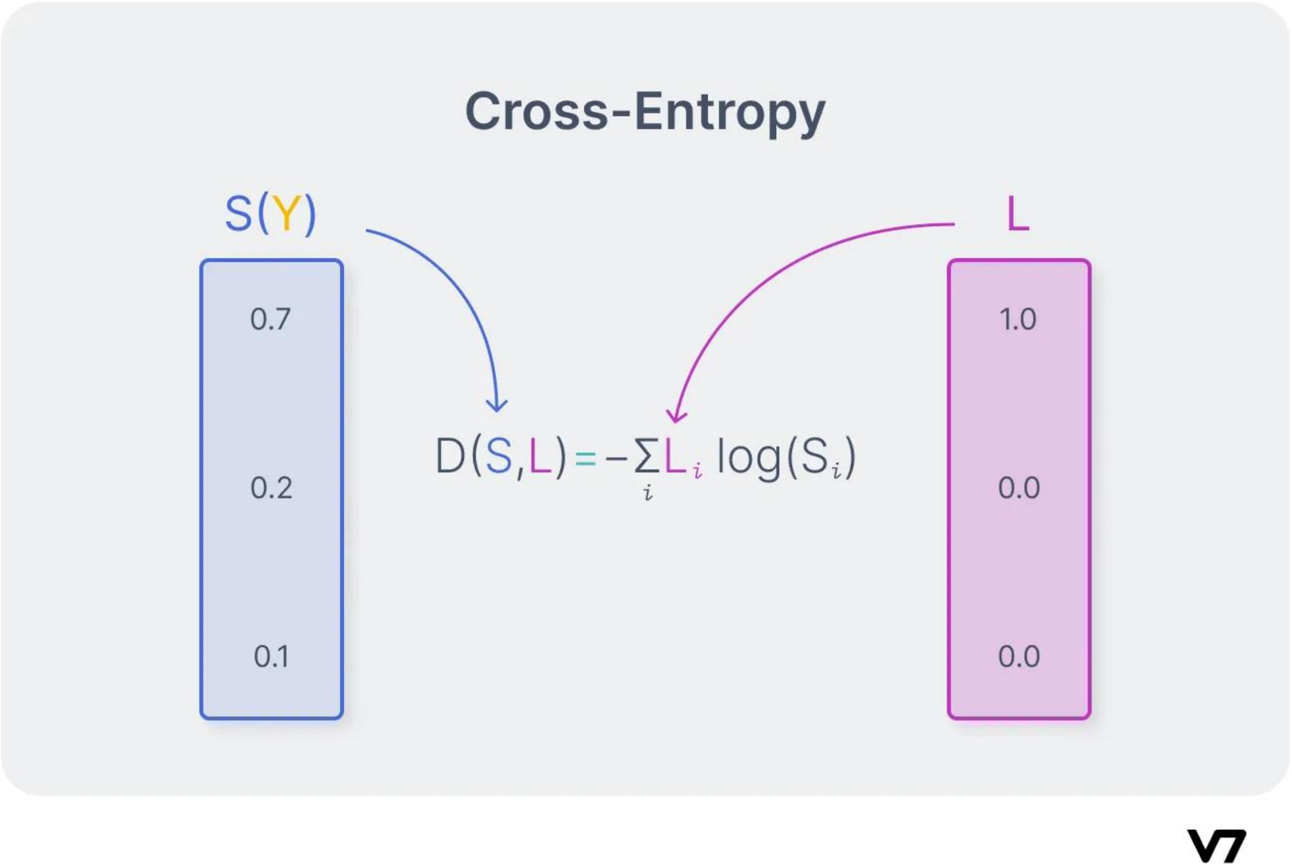


Categorical Cross Entropy Loss

- Softmax converts logits into probabilities. The purpose of cross-entropy is to take the output probabilities (P) and measure the distance from the truth values (as shown below).



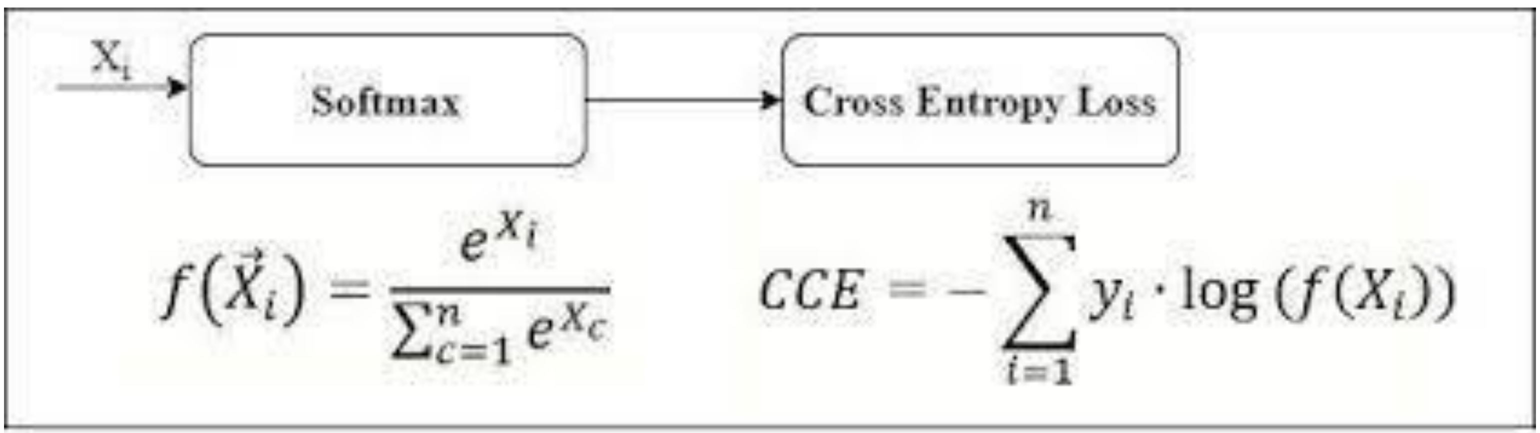
Categorical Cross Entropy Loss



Categorical Cross Entropy Loss

$$\textbf{Loss} = -\sum_{j=1}^{K} y_j \log(\hat{y}_j)$$

where k is number of classes in the data



Categorical Cross Entropy Loss

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^c e^{s_j}}\right)$$

Questions

- Why does cross-entropy is used most commonly as compared to MSE for classification problem?

Thank you