**Phần 1\_3\_Các hàm mất mát (loss) phổ biến**

**Notes:**

* *Về format cấu trúc soạn như sau:*
  + *Lý thuyết…*
  + *Bộ code mấu/ ví dụ …*
  + *Ứng dụng (nếu có)...*
* *Mems làm nhớ note tên để mn dễ contact*

[**I. Nội dung chính**](#_b5fp7wid63hv) **1**

[**II. Nội dung biên soạn chi tiết**](#_bh3evd21p67) **1**

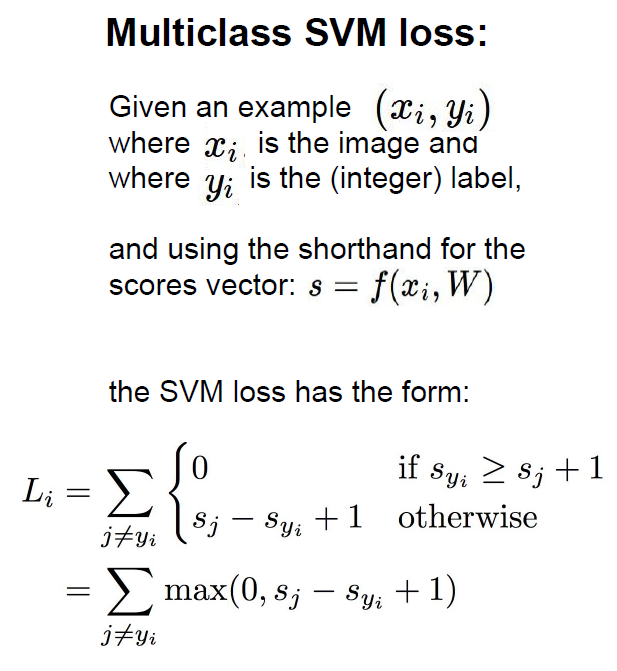
### **I. Nội dung chính**

Về dạng bài toán hay công việc (tasks): **phân loại**, **hồi quy**, **phát hiện**, **phân** **đoạn**, **sinh ảnh**. Cho mỗi dạng bài toán, cần hiểu rõ:

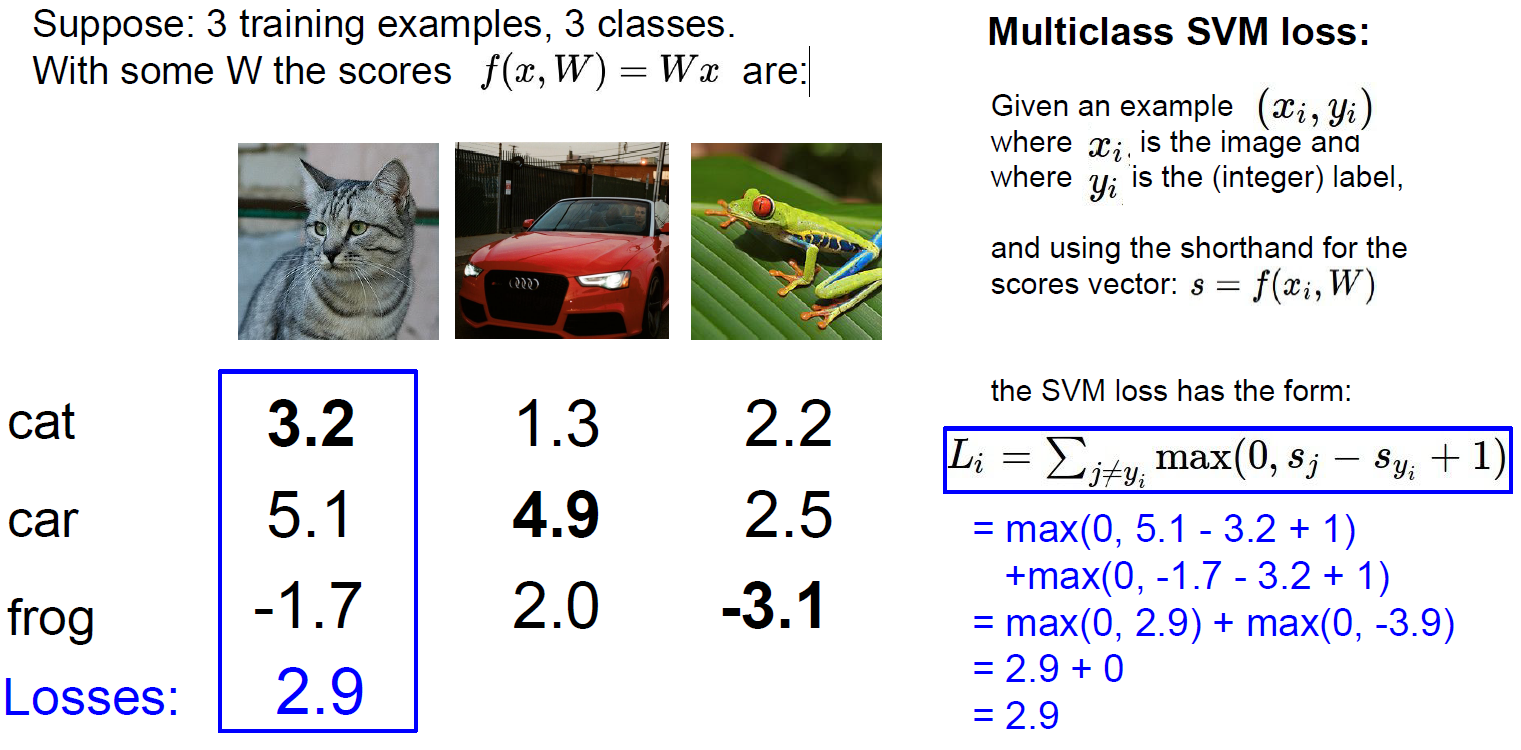
* (3) Các hàm mất mát (loss) phổ biến
  + - Ví dụ: cho hồi quy: **MSE (L2), MAE(L1), Distribution-Focal-Loss**
    - Ví dụ: cho phân loại: **binary-crossentropy, crossentropy và các** **biến thể (weighted/with-logits)**
    - Ví dụ: cho phân đoạn: **crossentropy, dice-loss, IoU-loss**
    - Ví dụ: cho phát hiện: I**oU-loss và các biến thể (Complete-IoU,** **Distance-IoU), Distribution-Focal-Loss**
    - Ví dụ: cho sinh ảnh: **KL-Divergence**

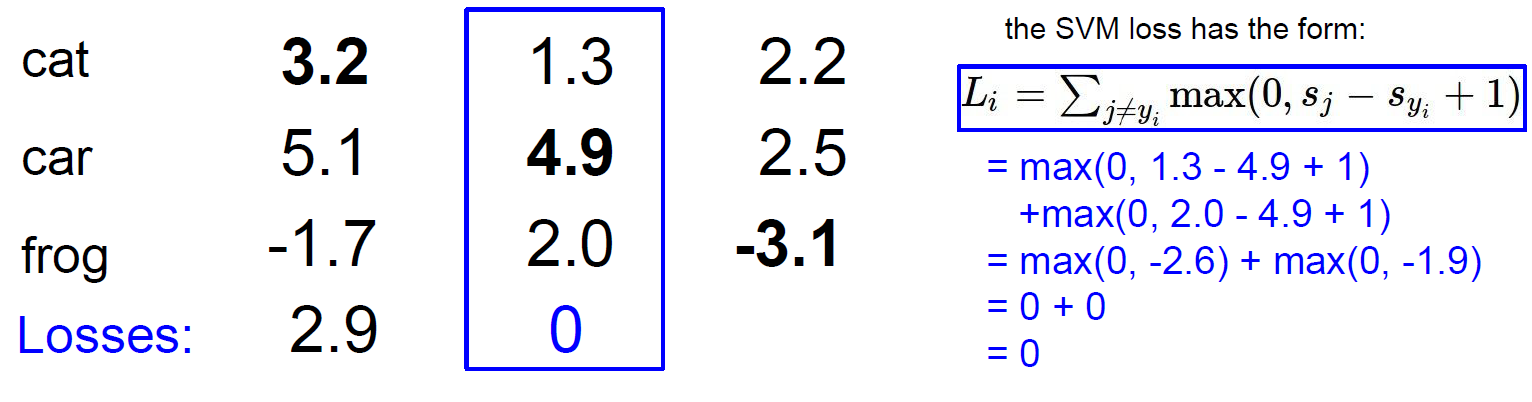
### **II. Nội dung biên soạn chi tiết**

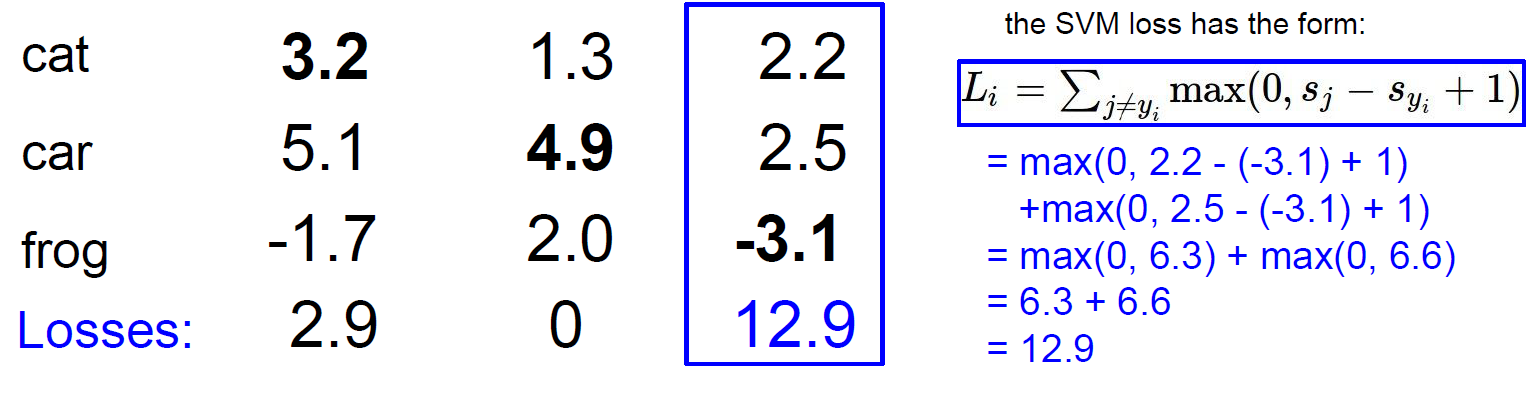
1. Hàm Multiclass SVM Loss, hay Hinge Loss, được sử dụng trong các mô hình phân loại đa lớp như Support Vector Machine (SVM) để đo lường độ lỗi trong quá trình huấn luyện.



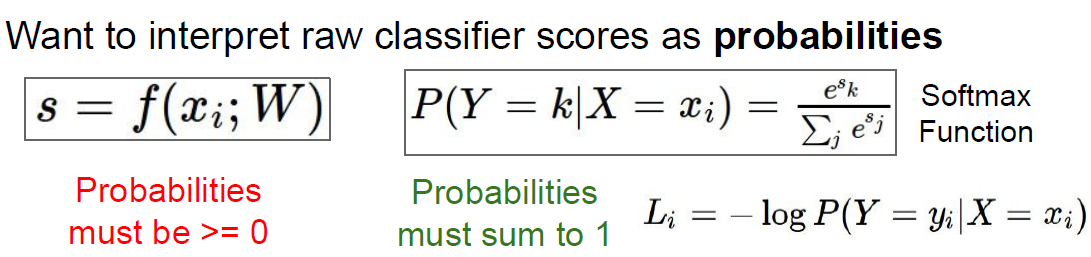
Ví dụ:

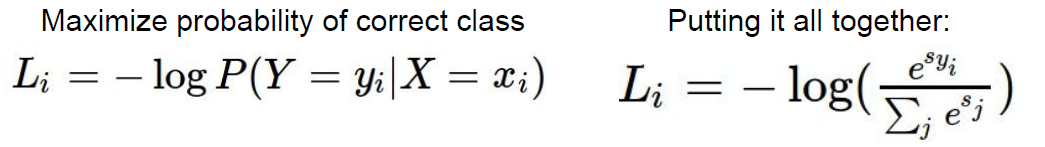




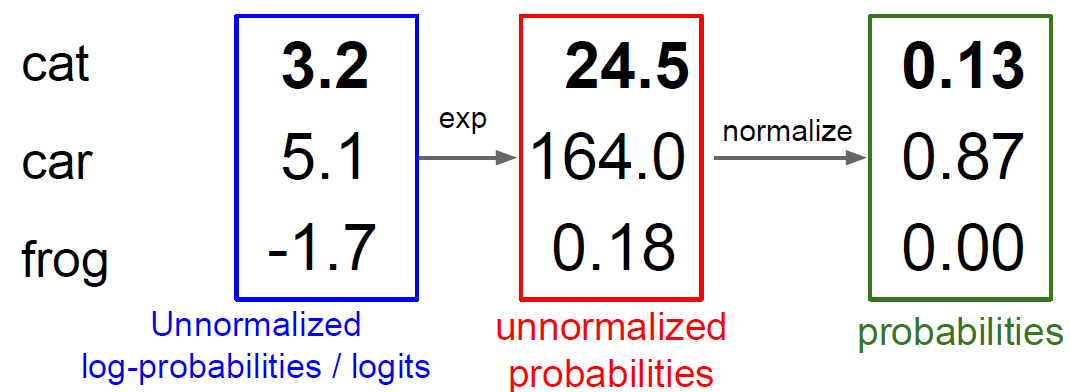


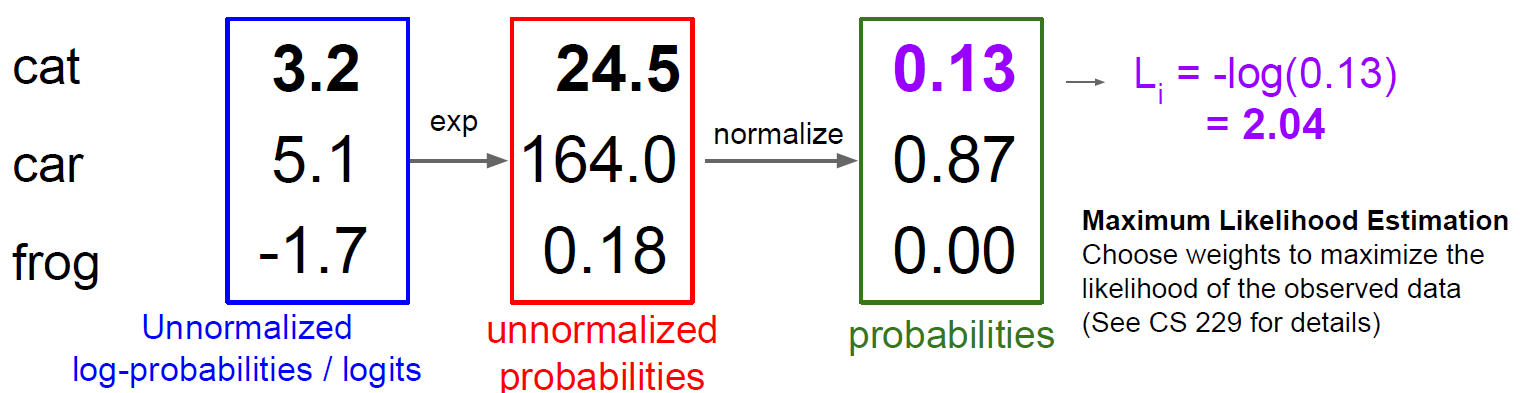
1. Hàm Softmax Classifier là một hàm phân loại thường được sử dụng trong các mô hình học máy để dự đoán xác suất thuộc về các lớp trong bài toán phân loại đa lớp. Nó chuyển đổi các "điểm số" (scores) đầu ra của một mô hình thành các xác suất dự đoán cho mỗi lớp.

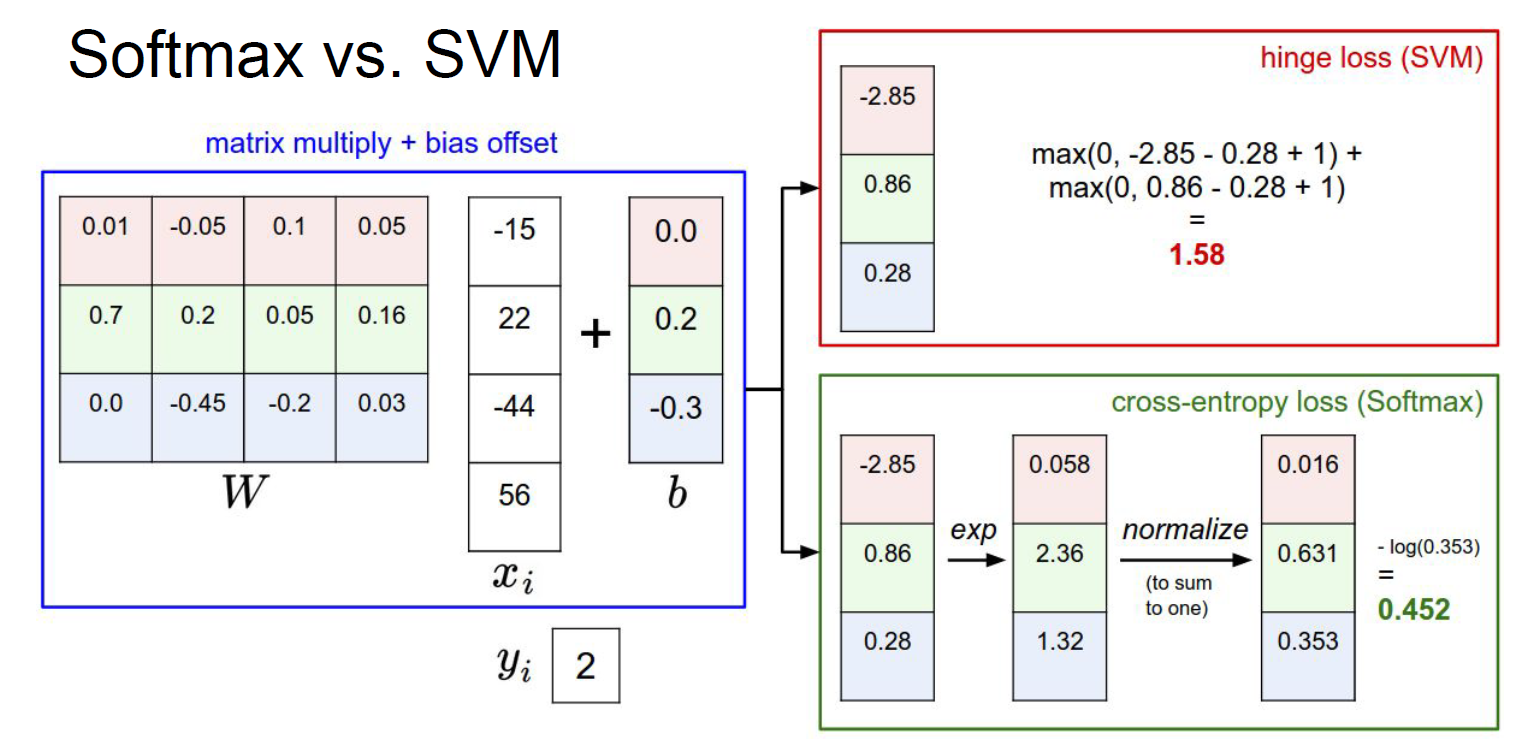




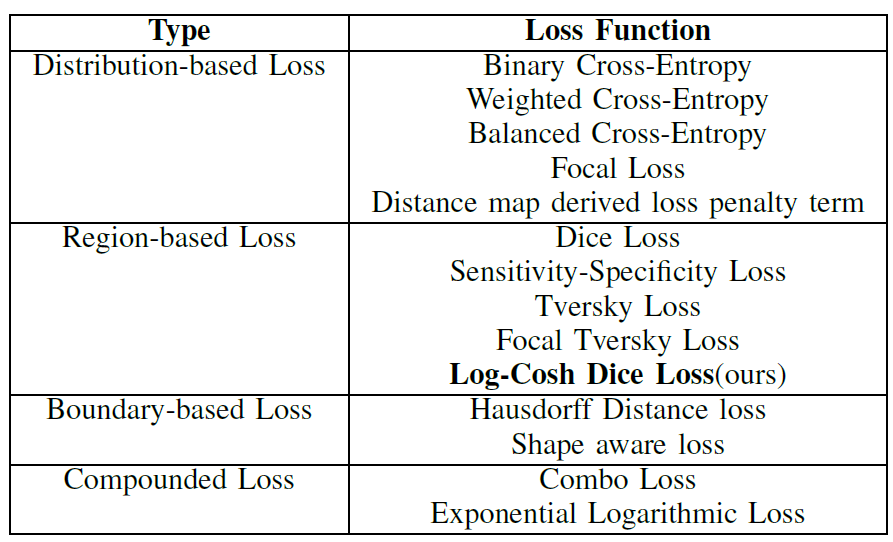
Ví dụ:

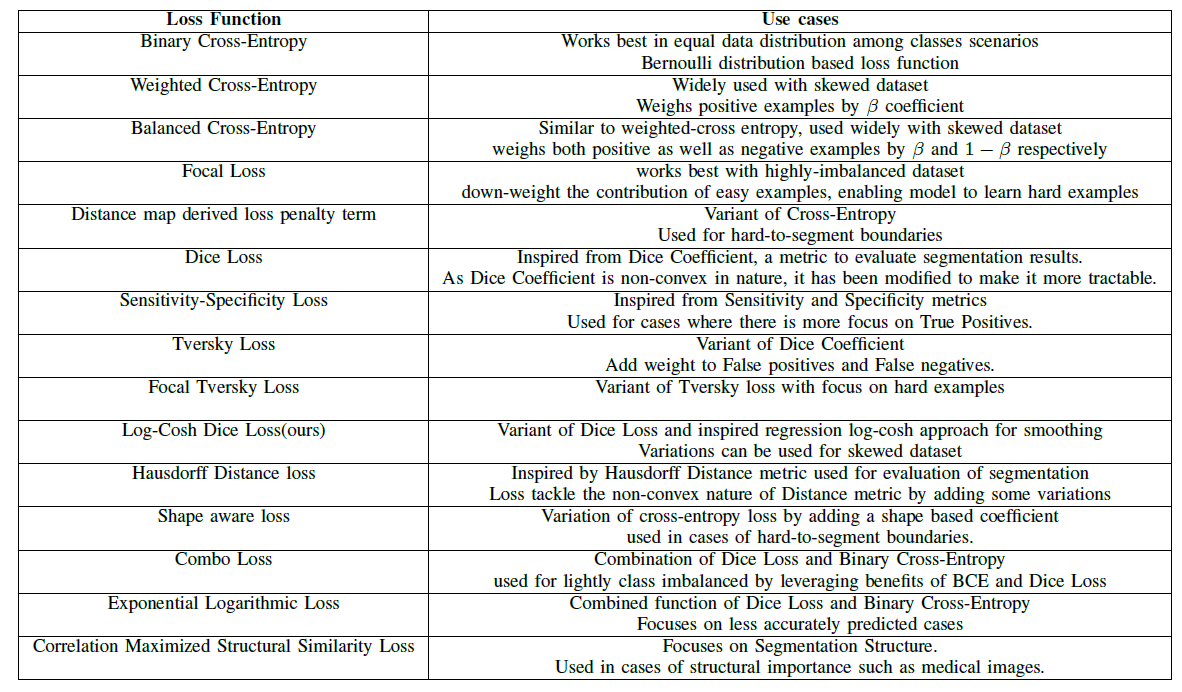




**

1. Một số hàm loss khác sử dụng trong segmentation:





**MSE (L2) - Mean Squared Error: Formula: MSE = (1/n) \* Σ(yi - ŷi)^2** where yi is the actual value, ŷi is the predicted value, and n is the number of samples.

Use case: MSE is commonly used in regression problems to measure the average of the squares of the errors or deviations, which gives more weight to larger errors. It is used to evaluate the performance of regression models.

**MAE (L1) - Mean Absolute Error: Formula: MAE = (1/n) \* Σ|yi - ŷi|** where yi is the actual value, ŷi is the predicted value, and n is the number of samples.

Use case: MAE is also used in regression problems to measure the average of the absolute errors, which gives equal weight to all errors. It is less sensitive to outliers compared to MSE and is often used when the distribution of the data is skewed.

**Distribution-Focal-Loss: Formula: DFL = -Σ(1 - p)^γ \* log(p)** where p is the predicted probability, and γ is the focusing parameter. Note that when γ=0 we get the standard cross-entropy loss.

Use case: Distribution-Focal-Loss is a loss function designed for imbalanced classification problems, where the distribution of classes is skewed. It helps to address the issue of class imbalance by giving more weight to misclassified examples from the minority class. This loss function is commonly used in tasks such as object detection and semantic segmentation.

**Binary Cross-Entropy: BCE = -Σ(y \* log(p) + (1 - y) \* log(1 - p))** where y is the actual label (0 or 1), p is the predicted probability, and the summation is over all samples.

Use case: Binary Cross-Entropy is commonly used in binary classification problems, where the output is a probability of belonging to one of the two classes. It penalizes the model more for confidently incorrect predictions and is widely used in tasks such as spam detection and medical diagnosis.

**Cross-Entropy: CE = -ΣΣ(y \* log(p))** where y is the one-hot encoded actual label, p is the predicted probability distribution over all classes, and the double summation is over all samples and classes.

Use case: Cross-Entropy is used in multi-class classification problems to measure the difference between the predicted probability distribution and the actual distribution of the classes. It is commonly used in tasks such as image classification and natural language processing.

**Weighted Cross-Entropy: WCE = -ΣΣ(w \* y \* log(p))** where w is the weight associated with each class, y is the one-hot encoded actual label, and p is the predicted probability distribution over all classes.

Use case: Weighted Cross-Entropy is used when the classes in the dataset are imbalanced, and certain classes are more important than others. It allows assigning different weights to different classes based on their importance, and is commonly used in scenarios such as medical diagnosis and fraud detection.

**Cross-Entropy with Logits: CEWL = -ΣΣ(y \* log(sigmoid(logits)) + (1 - y) \* log(1 - sigmoid(logits)))** where y is the one-hot encoded actual label, logits are the raw predictions before applying the sigmoid function, and the double summation is over all samples and classes.

Use case: Cross-Entropy with Logits is used when the model's output is not a probability distribution but raw scores (logits). It is commonly used in neural networks for multi-class classification tasks.

**Dice Loss: L\_dice = 1 - (2 \* |A ∩ B|) / (|A| + |B|)**

where:

A is the ground truth segmentation mask

B is the predicted segmentation mask

|A ∩ B| is the intersection of A and B

|A| and |B| are the total number of pixels or voxels in A and B, respectively

Use case: Dice loss is commonly used in medical image segmentation tasks, such as tumor segmentation, where the class imbalance between the background and the object of interest is significant. It measures the overlap between the predicted and ground truth masks, penalizing the model for poor overlap.

**Intersection over Union (IoU) Loss: 1-IoU = 1-|A ∩ B|/|A ∪ B|**

where:

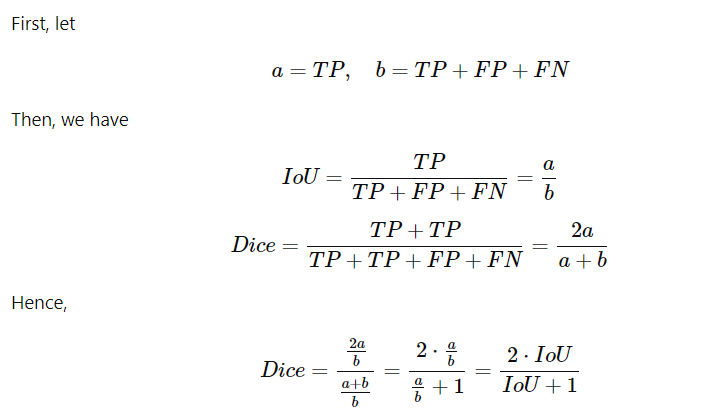
A is the ground truth segmentation mask

B is the predicted segmentation mask

|A ∩ B| is the intersection of A and B

|A ∪ B| is the union of A and B

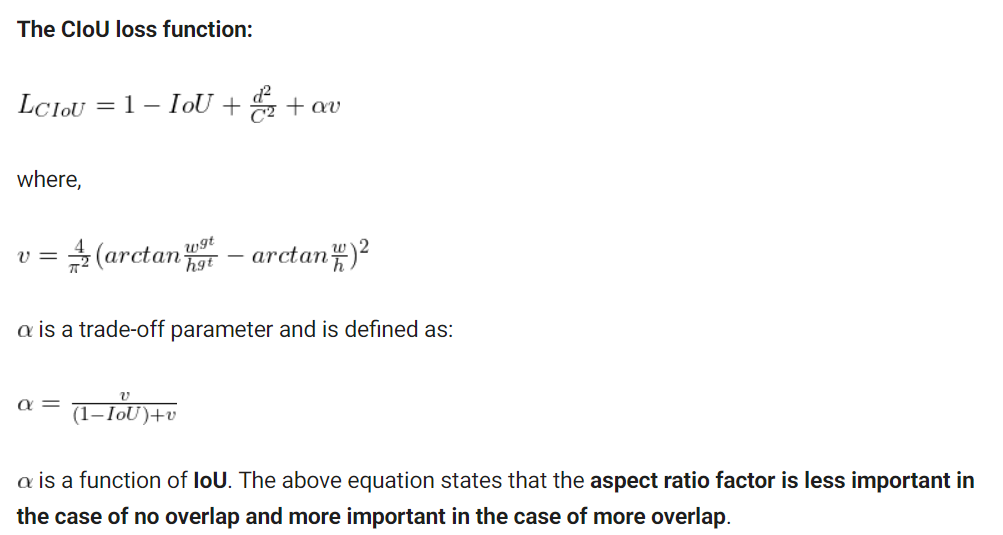
Use case: IoU loss is also commonly used in image segmentation tasks, particularly in evaluating the spatial overlap between the predicted and ground truth masks. It encourages the model to produce accurate and spatially aligned segmentations by penalizing the model for poor overlap.



**Generalized-IoU Loss: G-IoU = 1 - IoU + |C\(A ∪ B)|/C** where C is the area of the smallest rectangle enclose 2 both bounding boxes.

**Distance-IoU Loss: D-IoU = 1 - IoU + d^2 / C^2** where d is the distance between the predicted bounding box and the ground truth bounding box, and C is the area of the smallest rectangle enclose 2 both bounding boxes.

Loss for DIoU loss function is much better than GIoU loss function. It is because it does not depend on the orientation of the anchor box to ground truth.

**Complete-IoU loss: **

KL divergence, also known as Kullback-Leibler divergence, is a measure of how one probability distribution diverges from a second, expected probability distribution. In the context of image generation, KL divergence is often used as a loss function in variational autoencoders (VAEs) to train the model to generate realistic images while also learning a latent space representation.

**KL divergence formula: KL(P || Q) = ∑ P(x) \* log(P(x) / Q(x))**

Where P and Q are the probability distributions, and x represents the data points.

Use cases:

In the context of image generation, KL divergence is used in variational autoencoders (VAEs) to regularize the latent space representation. VAEs consist of an encoder network that maps input images to a latent space and a decoder network that generates images from the latent space. The KL divergence term in the VAE loss function encourages the latent space to follow a specific prior distribution, such as a standard normal distribution, which helps in generating diverse and realistic images while ensuring that the latent space is well-structured and continuous.

During training, the VAE aims to minimize the reconstruction error (typically measured using pixel-wise loss) while also minimizing the KL divergence between the learned latent distribution and the desired prior distribution. This dual objective allows the VAE to learn a latent space representation that captures meaningful features of the input images and enables the generation of new, realistic images by sampling from the learned latent space.

Overall, KL divergence plays a crucial role in guiding the learning process of VAEs for image generation, balancing the trade-off between reconstruction accuracy and the structure of the latent space.

