What is a Siamese Network?

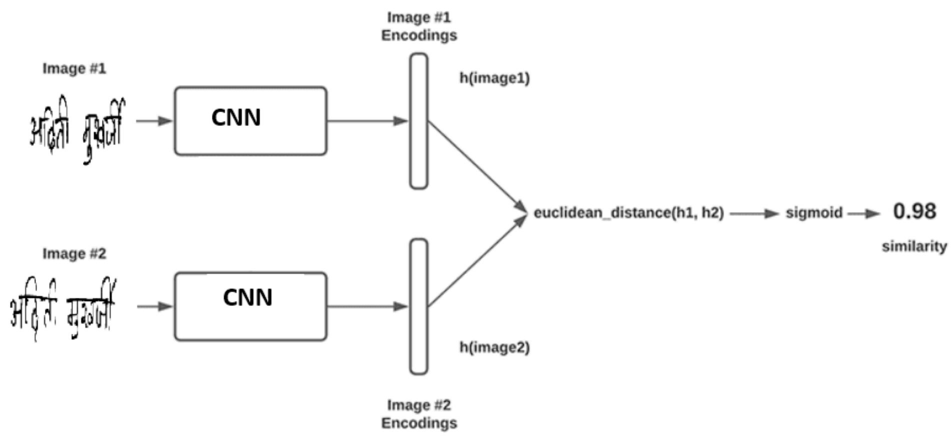
- A Siamese Network is a type of neural network architecture that uses two or more identical sub-networks to process input data.

- These sub-networks have the same architecture and parameters and are used to extract features from the input data.

- The outputs of the sub-networks are then compared to perform a specific task, such as similarity measurement or verification.

- The Siamese Network is commonly used in tasks such as face recognition and signature verification.

Architecture of Siamese Network



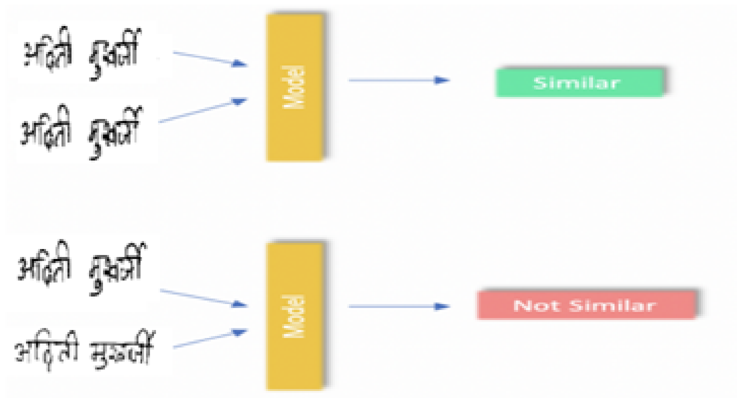
Contrastive Loss

- The contrastive loss function takes as input two feature representations (embeddings) of a pair of data samples, such as two images, and a binary label indicating whether the pair is similar or dissimilar.

- The contrastive loss then computes the Euclidean distance between the embeddings and applies a penalty based on the label.

- If the pair is similar, the loss function encourages the embeddings to be close together. If the pair is dissimilar, the loss function encourages the embeddings to be far apart.

- The contrastive loss helps the Siamese Network learn a discriminative embedding space, where similar inputs are close together and dissimilar inputs are far apart. This enables the network to perform tasks such as similarity measurement, one-shot learning, and verification more accurately.

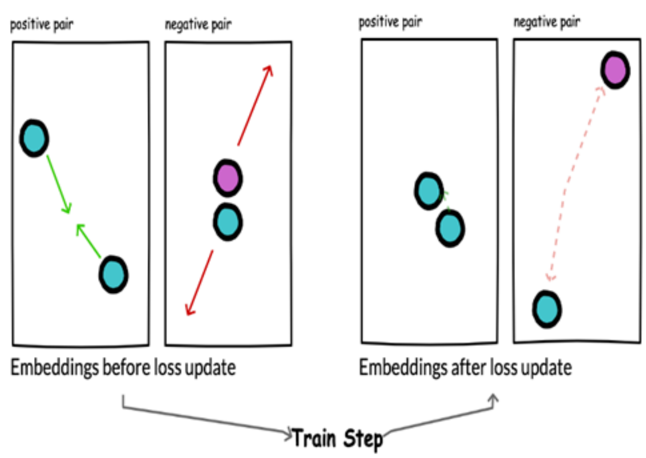


Contrastive Loss Equation

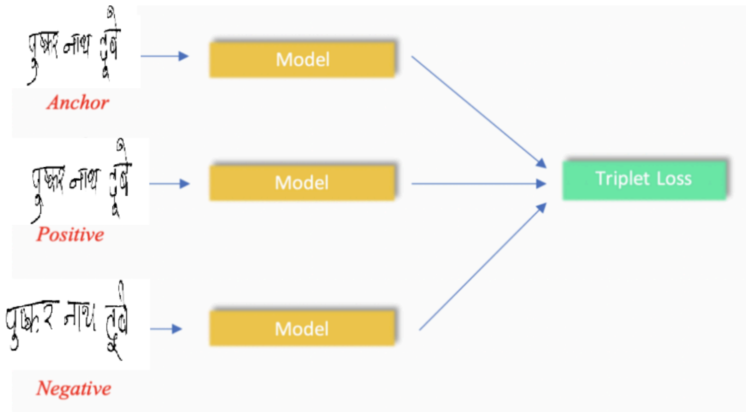
$$Y * D^2 + (1 - Y) * max(margin - D, 0)^2$$

- The Y value is our label. It will be 1 if the image pairs are of the same class, and it will be 0 if the image pairs are of a different class.
- The D variable is the Euclidean distance between the outputs of the sister network embeddings.
- The max function takes the largest value of 0 and the margin, m, minus the distance.

Embeddings before and after contrastive Loss



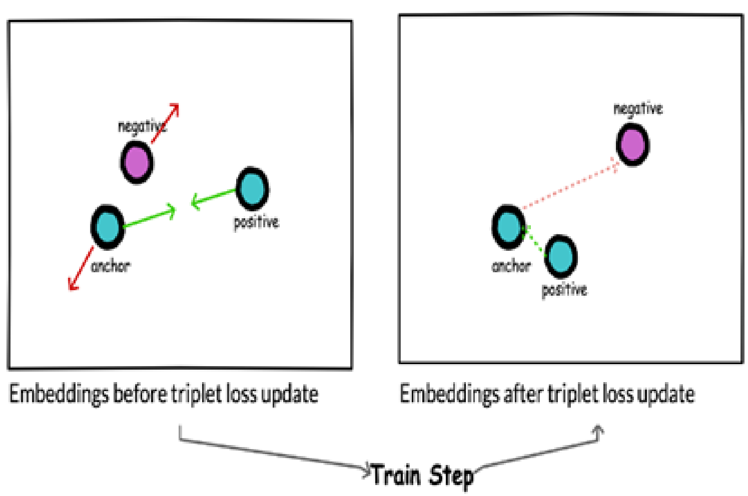
Triplet Loss



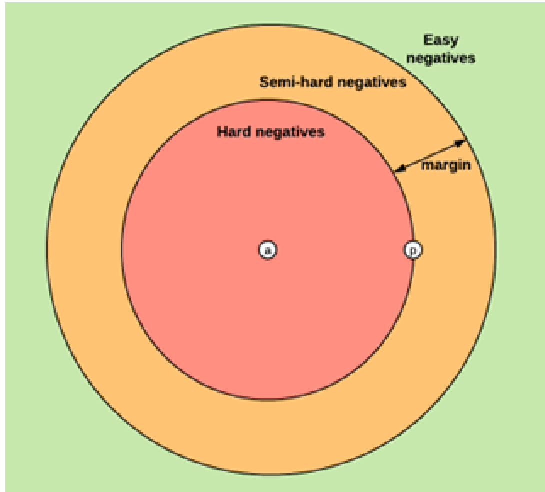
- A triplet consists of three data samples: an anchor, a positive sample, and a negative sample. The anchor and positive samples are similar, while the anchor and negative samples are dissimilar.

- The triplet loss function takes as input the feature representations (embeddings) of the anchor, positive, and negative samples and minimizes the difference between the distances of the anchor-positive and anchor-negative pairs.
- In other words, the loss function encourages the network to produce embeddings such that the distance between the anchor and positive sample is smaller than the distance between the anchor and negative sample by a margin.

Embeddings before and after triplet loss



Types of Triplet mining



1. **Easy triplets:** result when d(A,N)>d(A,P)+α. Here, the sampled anchor-to-negative distance is already large enough so loss is 0, and the network has nothing to learn from.

2. **Hard triplets:** result when d(A,N)<d(A,P). In this case, the anchor-to-negative distance is less than the anchor-to-positive distance, meaning high loss to backpropagation through the network.

3. **Semi-hard triplets:** result when d(A,P)<d(A,N)<d(A,P)+α. Semi-hard triplets occur when the negative example is more distant to the anchor than the positive example, but the distance is not greater than the margin. This, therefore, results in a positive loss (i.e., the negative is far … but not far enough.)

Lossless Triplet loss

$$\sum_{i=1}^{n}\left[-ln\left(-\frac{(f_i^a-f_i^p)^2}{\beta}\right)+1+\epsilon\right)-ln\left(-\frac{N-(f_i^a-f_i^n)^2}{\beta}\right)+1+\epsilon)]$$

- N is the number of dimensions (Number of output of your network)
- β is the scaling factor
- ϵ is the margin
- $(f_i^a - f_i^p)$ is the distance between anchor and positive
- $(f_i^a - f_i^n)$ is the distance between anchor and negative