**Contrastive Learning**

* Research for what is best for Japanese Handwriting Leaflets

**Contrastive Learning: A Foundation for Similarity-Based Representation**

**Contrastive Learning** is a powerful machine learning paradigm designed to learn meaningful representations by comparing samples, rather than by classifying them into predefined categories. The core idea is to train models to understand the concept of *similarity* and *dissimilarity* based on relationships between data points.

In contrastive learning, the model is not directly trained to assign a class label to an input. Instead, it learns to encode data into a vector space where *similar instances are embedded close together*, while *dissimilar ones are pushed farther apart*. This is particularly useful in scenarios where class labels are ambiguous or unavailable, such as clustering writer styles in historical handwriting.

There are typically two types of sample structures used:

* **Pairs:** Consist of a "positive pair" (similar inputs) or a "negative pair" (dissimilar inputs).
* **Triplets:** Consist of an anchor, a similar input (positive), and a dissimilar one (negative).

Contrastive learning is widely used in **self-supervised learning**, **face verification**, **signature authentication**, and **handwriting analysis**, where the key is to learn a robust feature space without requiring exhaustive labeling.

## Siamese Network: Learning Binary Similarity

A **Siamese Network** is a type of neural network architecture that operates on pairs of inputs to determine whether they are similar. It is particularly effective when used in conjunction with **contrastive loss**, a loss function that directly optimizes the distance between learned embeddings.

### Architecture:

The Siamese Network consists of **two identical neural network branches** (usually CNNs for image input), which share the same weights. Each branch processes one input image and produces an embedding vector. These embeddings are then compared using a distance metric such as **Euclidean distance** or **cosine similarity**.

### Contrastive Loss Function:

The model is trained using the following contrastive loss function:



Where:

* DDD is the distance between the two embeddings,
* y=1y = 1y=1 for a similar (positive) pair, and y=0y = 0y=0 for a dissimilar (negative) pair,
* margin is a tunable parameter that sets a minimum distance for dissimilar pairs.

The goal is to minimize the distance between embeddings of the same writer while maintaining a minimum distance between embeddings of different writers.

### Application:

In our Japanese handwriting analysis project, the Siamese network is used to **verify whether two character images come from the same writer**. This binary similarity setup is helpful for tasks such as:

* Writer authentication
* Duplicate detection
* Verification modules for OCR systems

## Triplet Network: Learning Relative Similarity

While Siamese networks focus on binary similarity, **Triplet Networks** extend contrastive learning by focusing on **relative similarity**. Instead of comparing just two samples, triplet networks train on **three** samples simultaneously:

* **Anchor (A)**: The reference image
* **Positive (P)**: An image of the same class (e.g., same writer)
* **Negative (N)**: An image of a different class (e.g., different writer)

### Triplet Loss Function:

The training objective is to ensure that the distance between the anchor and positive is smaller than the distance between the anchor and negative by at least a specified margin:



Here:

* f(x)f(x)f(x) is the embedding of image xxx,
* ∥⋅∥\| \cdot \|∥⋅∥ is typically Euclidean distance,
* The margin ensures that the negative example is sufficiently farther than the positive.

This relative comparison makes triplet loss particularly effective for **fine-grained distinction tasks**, where even subtle differences in handwriting need to be captured.

### Benefits:

* Learns a richer, more nuanced embedding space
* Better for clustering, retrieval, and ranking tasks
* Naturally generalizes to unseen classes (e.g., new writers)

### Application in Our Project:

In our project, the triplet network is used to **learn stylistic differences between writers** with higher precision. This allows us to:

* Group similar handwriting styles (unsupervised clustering)
* Improve writer identification in unlabeled datasets
* Build stronger foundations for downstream tasks like authorship attribution or historical style analysis

