Carnegie Mellon University

11-785 Introduction to Deep Learning

- Spring 2023 -

Recitation 7: CNN Verification

Prakruthi Pradeep
Arjun Chauhan

Objectives

1. What is Verification?
2. The Verification Task
3. Verification optimized approaches
4. Tips and run-through of HW2P2

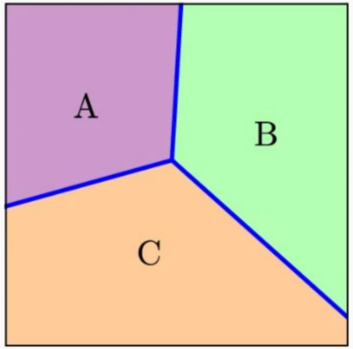
The Verification Problem

- A matching operation: Match the given sample to the closest sample from a reference of N other samples
- Case in point: Determine if the person in a given “query” picture is also present in a given gallery of images or not, with no reference to their identity?
- Can also be a 1 to 1 task, where we want to verify if the two embeddings are similar (belong to the same class)

Closed Set & Open Set

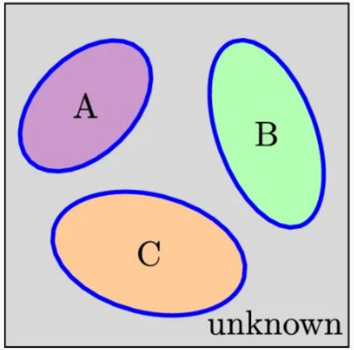
Closed Set

- K known classes are present during training and testing
- Learns decision boundaries that divide the feature space into K parts



Open set

- K known classes during training but K known and U unknown classes present during inference
- Tight decision boundary around the K classes are learned

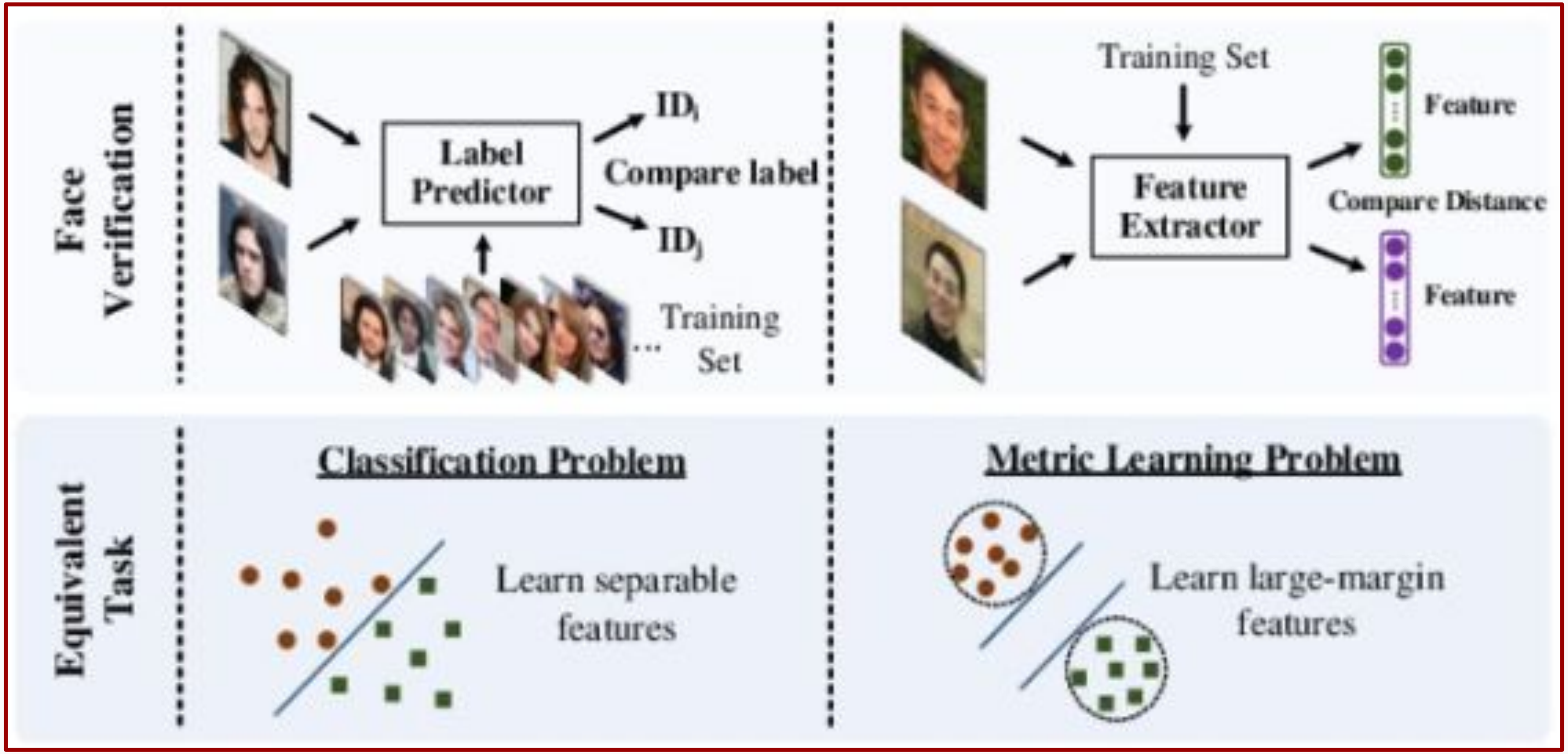


Closed Set or Open Set

Classification → **Closed Set**

Verification → **Open Set**

Closed Set vs. Open Set (Verification)



Closed Set **Open Set**

Problem Statement

Definition	This is an open set problem, where the subjects in the test data may not have been seen during training at all.
Task	Task: determine if the person in a given “query” picture is also present in a given gallery of images or not, with no reference to their identity

Feature extraction model

Feature embedding

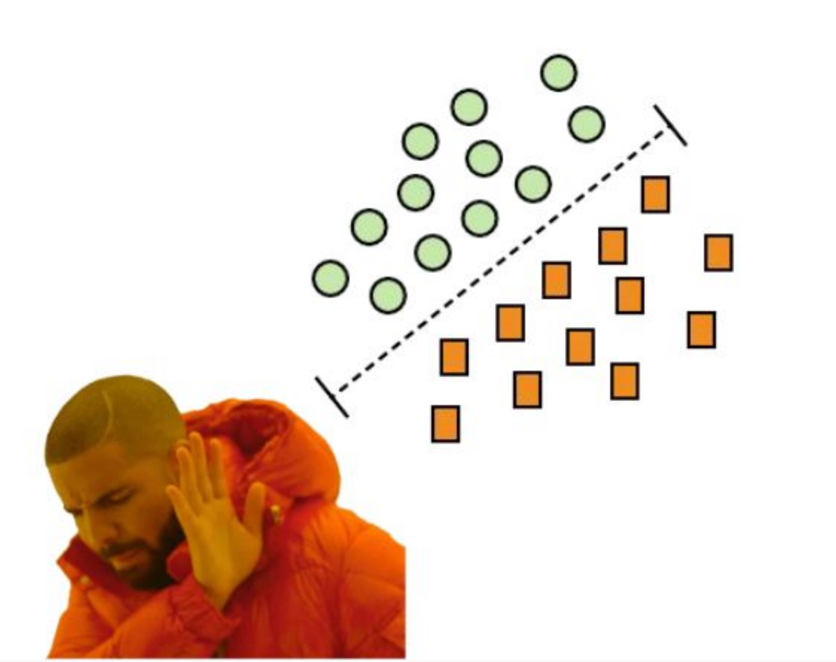
Feature extraction model

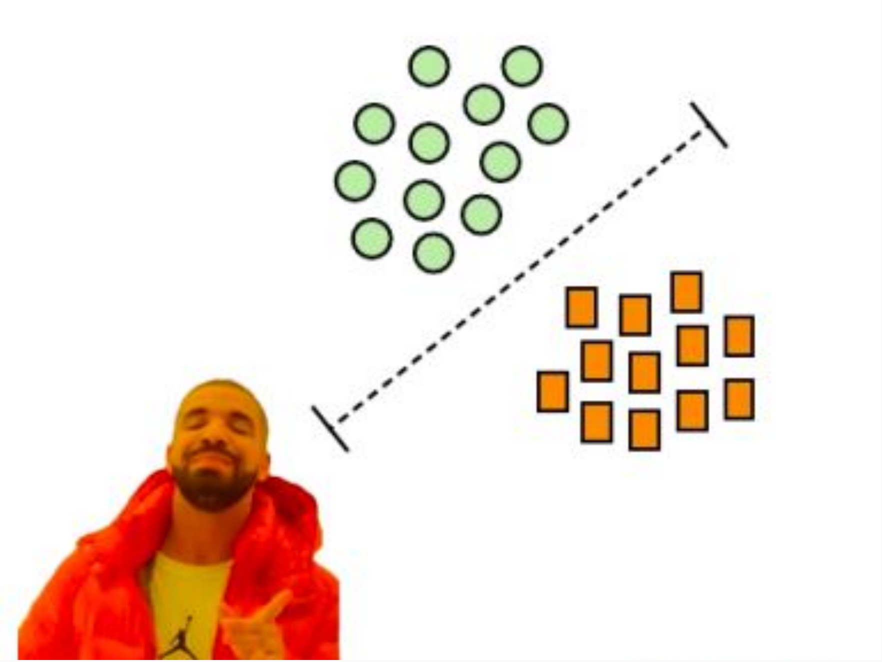
Feature embedding

Cosine Similarity

A similarity score from -1 to 1

What we need: **Discriminative Features**





How can we get discriminative features?

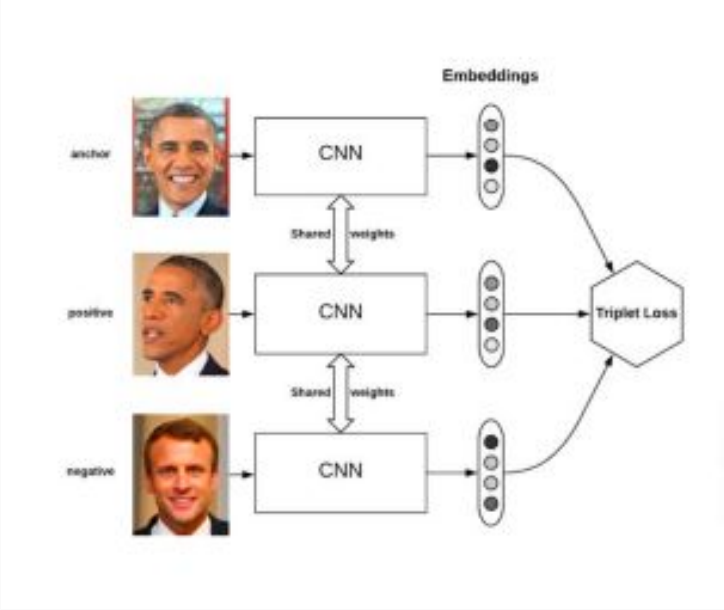
Use a loss function which correlates to your evaluation criteria !

Train your network to generate discriminative embeddings

Enter Contrastive Losses

Training for Verification - Metric Learning

Task	Determine if the person in a given “query” picture is also present in a given gallery of images or not, with no reference to their identity

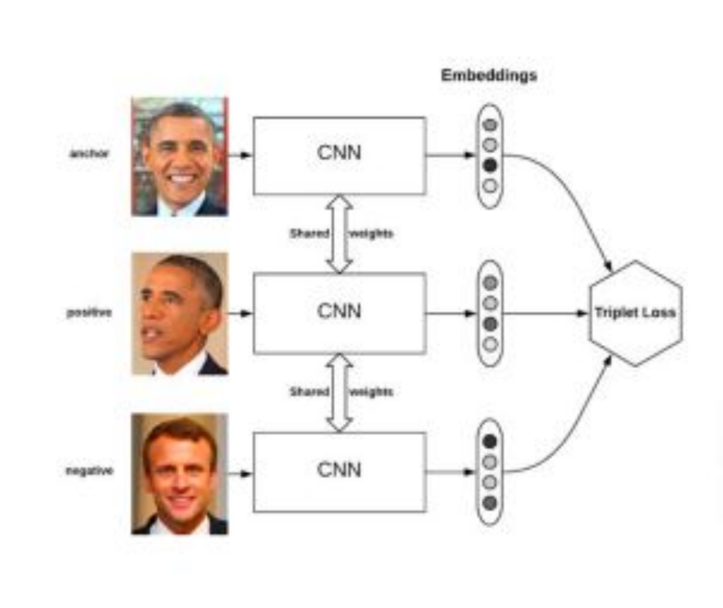


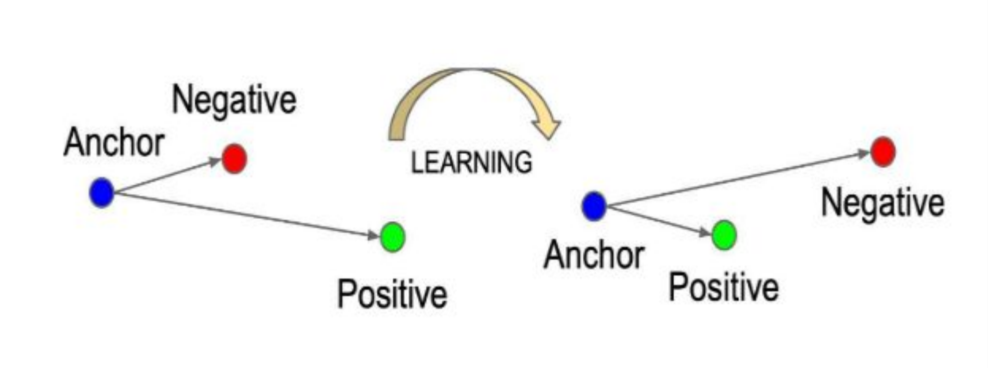
- Feature extraction (CNNs)
- Feature embeddings (creates important features which are separable)
- Contrastive Loss

$$\mathcal{L}_{\text{triplet}}(\mathbf{x},\mathbf{x}^+,\mathbf{x}^-) = \sum_{\mathbf{x}\in\mathcal{X}} \max\left(0, \|f(\mathbf{x}) - f(\mathbf{x}^+)\|_2^2 - \|f(\mathbf{x}) - f(\mathbf{x}^-)\|_2^2 + \epsilon\right)$$

Training for Verification - Metric Learning

Task	Determine if the person in a given “query” picture is also present in a given gallery of images or not, with no reference to their identity





$$\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^{+}, \mathbf{x}^{-}) = \sum_{\mathbf{x} \in \mathcal{X}} \max\left(0, \|f(\mathbf{x}) - f(\mathbf{x}^{+})\|_2^2 - \|f(\mathbf{x}) - f(\mathbf{x}^{-})\|_2^2 + \epsilon\right)$$

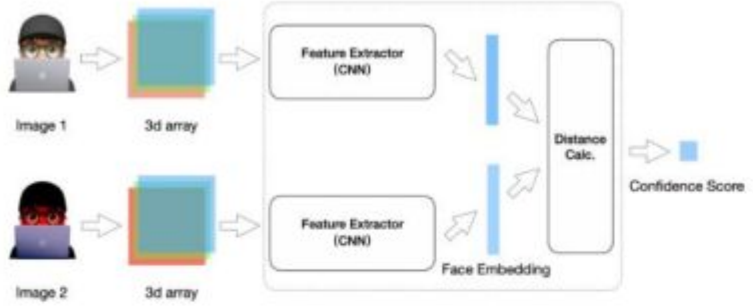
Examples of Contrastive Losses

1. Centre Loss
2. Triplet Loss
3. Sphere Face (Angular Softmax)
4. CosFace Loss
5. ArcFace Loss

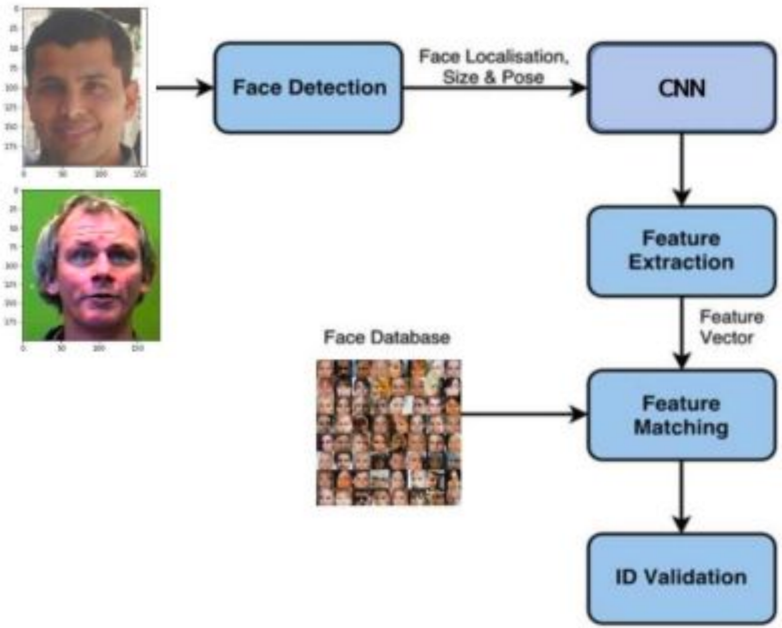
Problem Statement

Definition	This is an open set problem, where the subjects in the test data may not have been seen during training at all.
Task	Task: determine if the person in a given “query” picture is also present in a given gallery of images or not, with no reference to their identity

For this problem you will have to find the similarity between each unknown identity and all the known identities, and then predict the one with the highest similarity



Face Verification as image retrieval

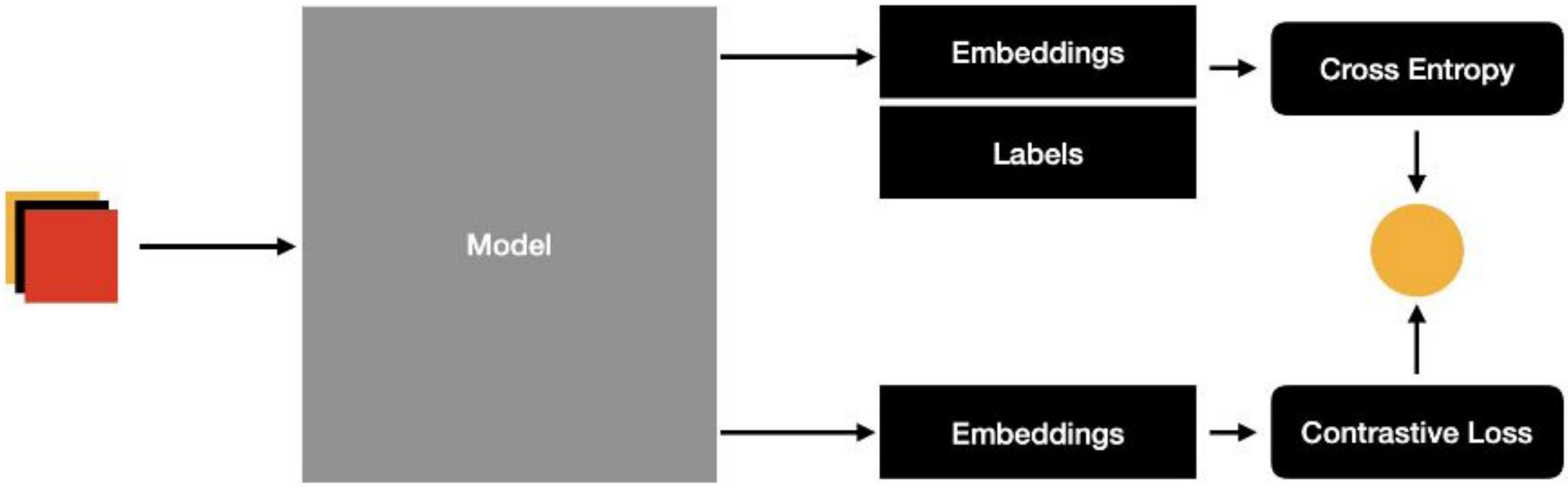


Verification Optimized Approach

1. Joint Loss Optimization

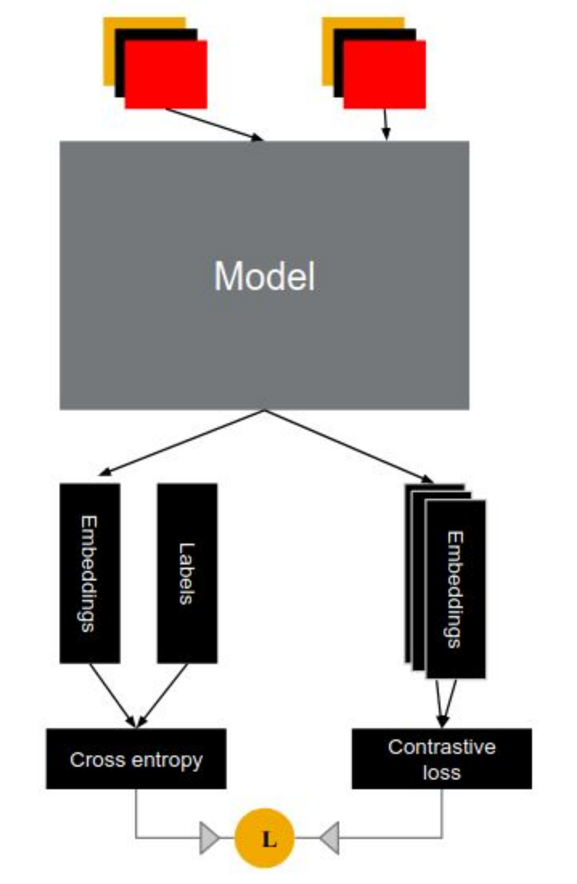
2. Sequential Layer Integration

Verification Optimized Approach - Joint Loss Optim.

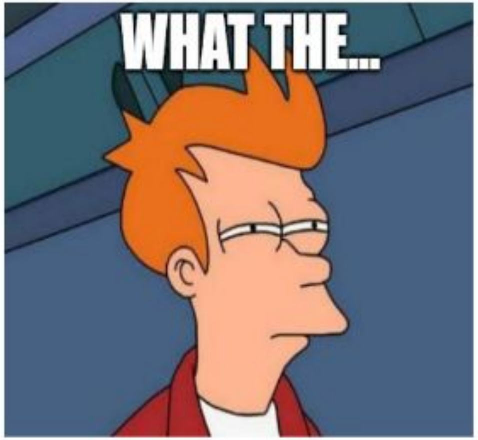


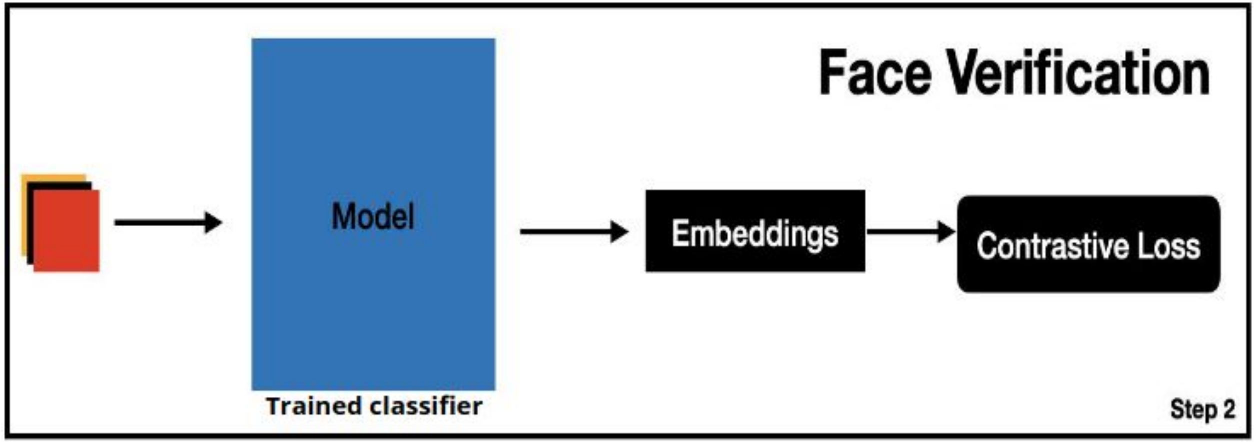
Verification Optimized Approach - Joint Loss Optim.

```
logits= model(x, return_feats=False)  # → 7_000 features

# get anchor, positive and negative from TripletDataset
anchor_emb = model(anchor, return_feats=True)
positive_emb = model(positive, return_feats=True)
negative_emb = model(negative, return_feats=True)

loss_1 = CrossEntropyLoss(logits, targets)
loss_2 = TripletLoss(anchor_emb, positive_emb, negative_emb)

L = w1*loss_1 + w2*loss_2 # e.g w1 = 0.7 and w2 = 0.3
```

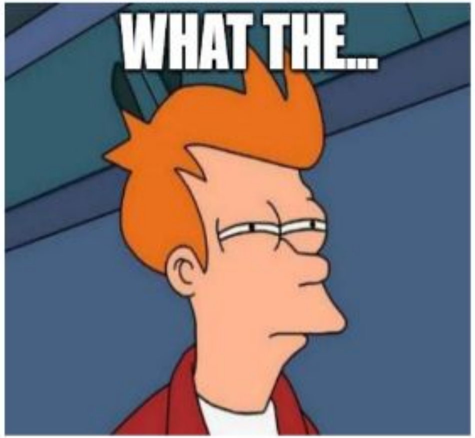


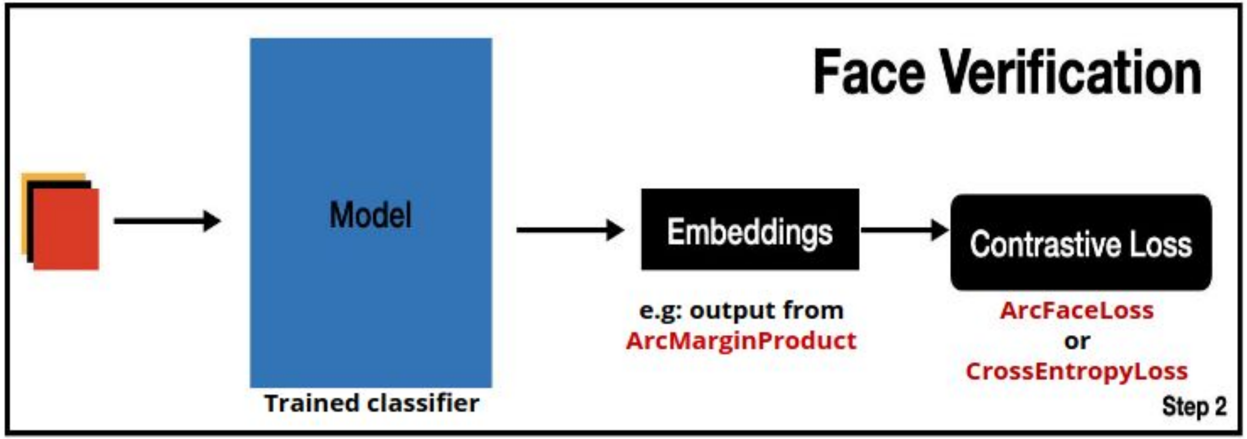
Verification Optimized Approach - Sequential





Verification Optimized Approach - Sequential





ArcMarginProduct would be an additional layer. Your features are an input to this layer

Verification Optimized Approach - Sequential

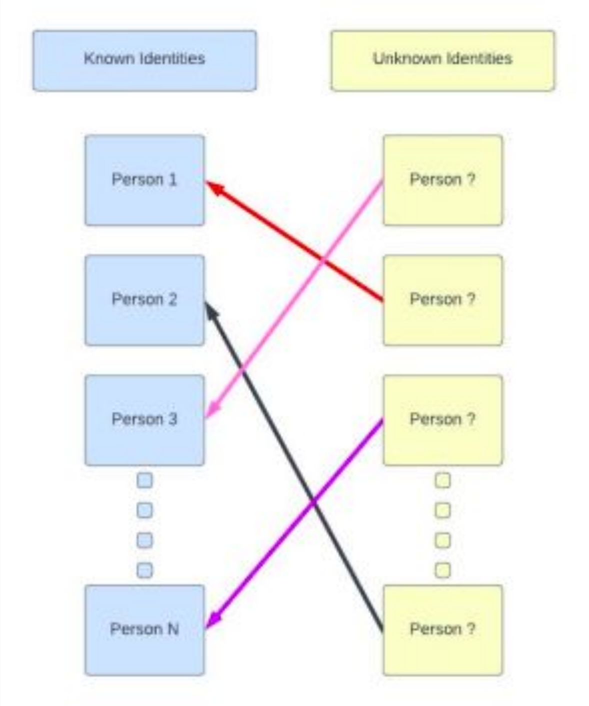
Ideally, ArcMarginProduct would be an additional layer to your classification network.

```
class VerificationNetwork(torch.nn.Module):
    def __init__(self, num_classes=7000):
        super().__init__()
        self.backbone = model.load_state_dict(torch.load(# TODO))
        self.arcFaceLayer = ArcMarginProduct(
                                embedding_size=# TODO,
                                n_classes=num_classes
        )

    def forward(self, x):
        feats = self.backbone(x, return_feats=True)
        out = self.arcFaceLayer(feats)
        return out
```

Dataset

- Subset of the VGGFace2 dataset.
- Images are downloaded from Google Image Search
- Large variations in pose, age, illumination, ethnicity, and profession
- 7,000 identities.
- Class-balanced, so each class has the equal number of training images, and all the images are resized to 224 x 224 pixels.
- **Aim**: Learn to classify images with the correct face identity from 7000 identities.

Your Verification Task



- Test set- Balanced set of 1000 known identities and 1000 unknown identities
- Task: find the similarity between each unknown identity and all the known identities, and then predict the one with the highest similarity

Homework 2 Part 2 Overview

- **Objective:** To solve an image-based face verification problem using a CNN
- **Scenario at hand:** Verifying faces in images.
- **Motivation:** Pictures of the faces have *indeterminacy of position* and CNN's are *position invariant*
- **Problem Type:** A open set problem, where the subjects in the test set may not been seen in the training set
- **Requirement:** Embeddings of two pictures of the same person are always closer than those from pictures of two different people.

Face Verification problems

- Face verification: Determining whether two face images are of the same person, without necessarily knowing who the person is.
- Larger class of problems:
  - Information retrieval: Google that allows a user to search for images using an image as the starting point.
  - Speaker verification: A speaker's voice contains personal traits of the speaker, so it is possible to automatically identify a speaker from his/her voice.
  - In each case, you are given an exemplar of a category of data (e.g., a face), and a "probe" instance, and you must determine if the two are from the same class.

Face Verification Approach: Use classification

- Can put images into categories/classes based on the closeness between their feature vectors (based on some metric)
- Classifying an image into one of several classes also requires the model to extract discriminative features from it
- Face images of persons with similar features (i.e features for which closeness is below a certain cutoff) will belong to the same class
- Perform multi-class face classification where the input to your system will be an image of a person's face, and the model will predict the class(out of a total of N classes) that image belongs.

Face Verification Approach: Use Contrastive Loss

- Crossing the high cutoff for classification does not imply high cutoff in face verification
- Using previous slide, we produce feature vectors that are ’separable’ by optimizing the network using cross-entropy loss.
- This doesn’t optimize for direct comparison of two instances to see if they belong to the same class. Distance between feature vectors belonging to the same class may be greater than the distance between feature vectors in different classes.
- Minimizing cross-entropy loss only aims to make the classes linearly separable in the embedding space
- Want the distance between the farthest points in each class to be less than the distance between points belonging to 2 different classes.
- Use Center loss, Sphere loss, Large-margin softmax loss, Large-margin Gaussian mixture loss etc., jointly with cross-entropy loss to get high-quality feature vectors.

Face Verification Procedure Overview

- The verification consists of two steps:
  - 1. Extracting the feature vectors from the images.
  - 2. Comparing the feature vectors using a similarity metric.
- Sample verification system:
  - 1. image1 ==⇒ feature extractor ==⇒ feature vector1
  - 2. image2 ==⇒ feature extractor ==⇒ feature vector2
  - 3. Feature vector 1, feature vector 2 ==⇒ similarity metric ==⇒ similarity score
- We have framed the problem a bit differently, as a one-to-many comparisons, where we compare one image to many images and then predict the image with the highest similarity
- We are going to compare each unknown identity with all the known identities and then predict the known identity with the highest similarity score.

Similarity metrics

- Assuming our CNN is competent to generate accurate face embeddings; we only need to find a proper distance metric to evaluate how close given face embeddings are.
- If two face embeddings are close in distance, they are more likely to be from the same person .
- Two prevalent distance metrics:
  - Cosine Similarity
  - Euclidean Distance
- Use a feature extractor that is good at extracting discriminative features from the images
- Use extracted feature vectors to compare the images. The comparison is done by using a similarity metric, which is a function that takes two feature vectors as input, and outputs a number that represents how similar the two feature

One-to-many comparisons

- Similarity metric, is a function that takes two feature vectors as input, and outputs a number that represents how similar the two feature
- You will be given a dataset of images, half of which are known identities and the other half of unknown identities.
- Set a threshold for the similarity score:
  - If maximum similarity score between an unknown identity and every known identity is below this threshold, say the unknown identity is not presented in the known set. For similarity scores above the threshold, predict the correct known identity.
- This is one-to-many comparison: can get an N x M matrix of similarities as well as the index of the highest similarity wrt each unknown identity.

Advanced Loss functions

- You are encouraged to try other advanced loss functions such as Center-loss, SphereFace, CosFace, ArcFace
- Alternatively, you can optimize the net using comparator losses, that optimize the network for the verification task. (triplet-loss, pair wise loss)
- We also encourage you to explore the interconnection between classification accuracy and verification performance