Why do we learn a function

�

(

�

�

�

1

,

�

�

�

2

)

d(img1,img2) for face verification? (Select all that apply.)

Given how few images we have per person, we need to apply transfer learning.

This allows us to learn to recognize a new person given just a single image of that person.

This allows us to learn to predict a person’s identity using a softmax output unit, where the number of classes equals the number of persons in the database plus 1 (for the final “not in database” class).

We need to solve a one-shot learning problem

For the statement "Why do we learn a function d(img1, img2) for face verification?" The correct justifications are:

This allows us to learn to recognize a new person given just a single image of that person: True. By learning a function that can measure the difference between two images, we can determine whether a new image is of a previously seen person by comparing it to a single existing image of that person.

We need to solve a one-shot learning problem: True. One-shot learning is a problem of learning from minimal data, such as a single example. In the context of face verification, one-shot learning is needed because we often have only one or a few images of each individual and we want to verify whether a new image is of the same individual. The function d(img1, img2) measures the distance or dissimilarity between two images, thus enabling one-shot learning.

The other options are not true in this context:

Given how few images we have per person, we need to apply transfer learning: False. While transfer learning can be very useful in the context of small datasets, it's not specifically related to the function d(img1, img2) for face verification.

This allows us to learn to predict a person’s identity using a softmax output unit, where the number of classes equals the number of persons in the database plus 1 (for the final “not in database” class): False. This statement is more related to a classification approach to face recognition, rather than a verification approach. The function d(img1, img2) doesn't involve softmax units or class prediction; it measures the distance or dissimilarity between two images.

Q

Question 3

You want to build a system that receives a person's face picture and determines if the person is inside a workgroup. You have pictures of all the faces of the people currently in the workgroup, but some members might leave, and some new members might be added. To train a system to solve this problem using the triplet loss you must collect pictures of different faces from only the current members of the team. True/False?

False. When building a face recognition system using triplet loss, you should ideally have images of many different faces, not just those of current team members. While having images of current team members is certainly important for verifying their identities, in order to learn a robust embedding space (the goal of the triplet loss approach), the model should see a variety of faces.

If your model only ever sees faces of the current team members, it may not generalize well to new team members when they are added. Furthermore, the model might struggle with correctly identifying that a person is not a team member if it hasn't been exposed to a sufficient variety of faces during training.

Thus, for a robust face recognition system, it's important to have a diverse training set that captures as much of the variability in human faces as possible. This includes variability in age, ethnicity, gender, lighting conditions, facial expressions, accessories (like glasses or hats), etc.

Triplet loss:

max

⁡

(

∥

�

(

�

)

−

�

(

�

)

∥

2

−

∥

�

(

�

)

−

�

(

�

)

∥

2

+

�

,

0

)

max(∥f(A)−f(P)∥

2

−∥f(A)−f(N)∥

2

+α,0)

is larger in which of the following cases?

When the encoding of A is closer to the encoding of P than to the encoding of N.

When

A

=

P

�=�

and

A

=

N

�=�

.

When the encoding of A is closer to the encoding of N than to the encoding of P.

The triplet loss function will be larger in the case where the encoding of A (Anchor) is closer to the encoding of N (Negative) than to the encoding of P (Positive).

In other words, the loss will be large when a negative example (a different identity) is closer to the anchor than a positive example (the same identity). The purpose of the triplet loss function is to ensure that positive examples (images of the same person) are closer in the encoding/embedding space than negative examples (images of different people). So, it penalizes situations where this is not the case

Q. You train a ConvNet on a dataset with cats, dogs, birds, and other types of animals. You try to find a filter that strongly responds to horizontal edges. You are more likely to find this filter in layer 6 of the network than in layer 1. True/False?

1 point

ChatGPT

False.

Earlier layers in a Convolutional Neural Network (ConvNet) typically detect low-level features such as edges, while later layers generally detect more high-level features (like shapes or specific object parts). Therefore, a filter that strongly responds to horizontal edges is more likely to be found in layer 1 of the network than in layer 6.