

Deep Learning
Prof. Prabir Kumar Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 51
Face Recognition

Hello welcome back to the NPTEL online certification course on Deep Learning. For our previous classes, we have talked about various deep neural network architectures. We have also talked about the algorithms for training the neural networks. And, we have also talked about the algorithms of how to make or how to train the neural network more efficiently or how to make the learning process faster as well as more accurate.

Now, given this background of the architecture of deep neural networks and the training of deep neural networks; now it is time to look at some of the use cases or where the neural networks or deep neural networks particularly the convolutional neural networks have been used in real life practice. So, today we will talk about one of the applications, which has been published in one of the literature that is the application of face recognition or face verification.

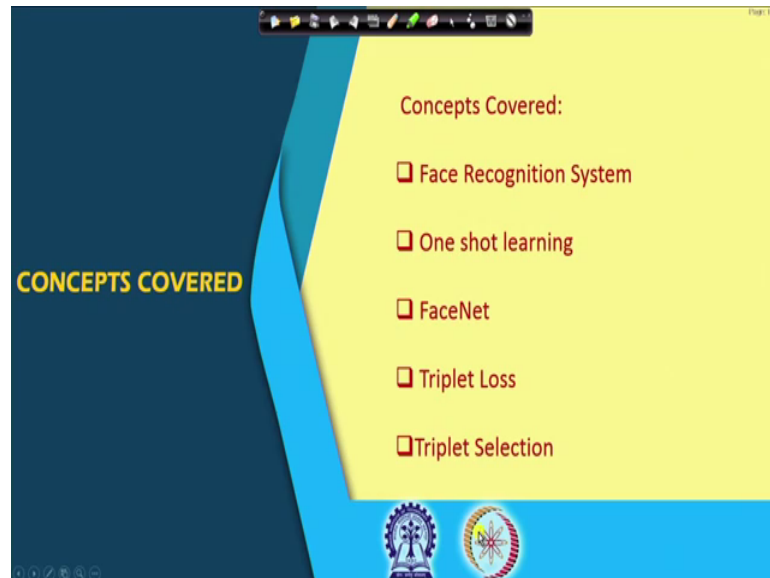
The network known as face net after discussing about face net, we will talk about other applications of the neural networks. Say for example, use of neural network for segmentation purpose because so far what we have seen is the neural networks which have been used for the recognition purpose or classification purpose that is how we have seen that the neural networks are being trained.

Now, the classification problem where entire image is classified belonging to different classes say for example, image of a dog or image of a bird. The segmentation problem is something like classifying every pixel within the image to belong to different classes. And when you collect all those pixels belonging to a particular class, that becomes a segment.

So, moving from image level classification to pixel level classification we will talk about after today's discussion. And we will see that what are the architectural as well as algorithmic modifications that, you need in those deep neural network architectures. So,

that you can take decisions not only at the image level, but you can take decisions at a finer that is at the pixel level. So, we will talk about that in our future lectures.

(Refer Slide Time: 03:10)



So, today we will talk about a face recognition system. You will see that how the face recognition system is important in today's digital world. Then we will talk about a learning mechanism, which is known as one short learning which is very very applicable in face recognition systems.

Then a particular network which employs face recognition is face net we will discuss about that. And, for training of face net the corresponding loss function, which is known as triplet loss we will discuss about this triplet loss. And, we will also talk about how to select the proper triplets for training of the face net right.

(Refer Slide Time: 03:58)

The slide features a yellow background with a dark blue curved shape on the right side. At the top, there is a navigation bar with various icons. The title 'Face Recognition System' is in a large, dark font. Below the title, a paragraph states: 'Face recognition system has become an integral part of our modern day to day life. Various applications of face recognition system are:'. A list of five applications follows, each preceded by a square checkbox: 'Payments', 'Access and security', 'Criminal identification', 'Advertising', and 'Healthcare'. At the bottom left, there are two circular logos. At the bottom right, a video inset shows a man with glasses and a mustache, wearing a light-colored shirt, speaking.

So, let us see that what is the importance of a face recognition system? In today's world face recognition has become very very important. Say for example, think of a situation that you are going to and put ATM for withdrawing your money from your own bank account. So, currently what we do is, while we are doing the money we swipe our ATM card.

Then we are prompted to enter the pin which is a secret code that only the owner of the account should know but do not it be better to improve upon the security. So, that in the ATM system suppose or in the tailoring machine. We have camera that will take image of your face when you stand in front of the ATM machine. And, then your face image will be matched with the image which is already there in the database of the bank.

So, this is a sort of authentication or security that you are withdrawing money from your own account through the ATM, in absence of any persons being from the bank being present over there. So, right now there are two different modalities which are mostly used. One is authentication based on the fingerprint in addition to that the authentication based on face recognition can also be employed.

Think of another application that is access and security. So, there are many areas or many offices which are highly secured that is only the authorized persons are allowed to enter those offices because there may be some strictly confidential documents or strictly

confidential information that might be in from available in that machine in that particular office.

So, those offices are not accessible to any person only the authorized person can access those offices and if we want to automate this particular process. So, before you have an access to the office there might be a camera taking your face image. And, then try to verify with that with an image which is already available in the database; that whether your face image is same as the image of an authorised person, which is already authorized by the authority and only when it is authenticated.

That is this it is the same person then only you are given entry to that particular office. So, this is another application of face recognition or face based authentication; commonly known as biometric authentication. Similarly criminal identification particularly criminal identification from cloud, in many cases if you visit a police station; we will find that photographs of wanted persons are tested on the notice board.

In some cases it is also written that most were most wanted. So, if there is any notorious criminal who is being whom the police is trying to find out another fellow is hiding somewhere. So, can we identify that person if the person moves along with a mob in a mob right? So, you take the face images from the mob and try to identify whether that person is present within the mob or not.

Similarly there are many such applications, the applications of face recognition of face authentication can be in advertising industry. It can be in healthcare applications and the applications are many more. So, you find that face recognition has become a very very important aspect in today's digital world.

(Refer Slide Time: 08:01)

Face Recognition System

Challenges:

- ❑ Different illumination condition.
- ❑ Different Pose and orientation of image.
- ❑ Other variational conditions.
- ❑ Limited Dataset for training.

Image Source: Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815-823. 2015.

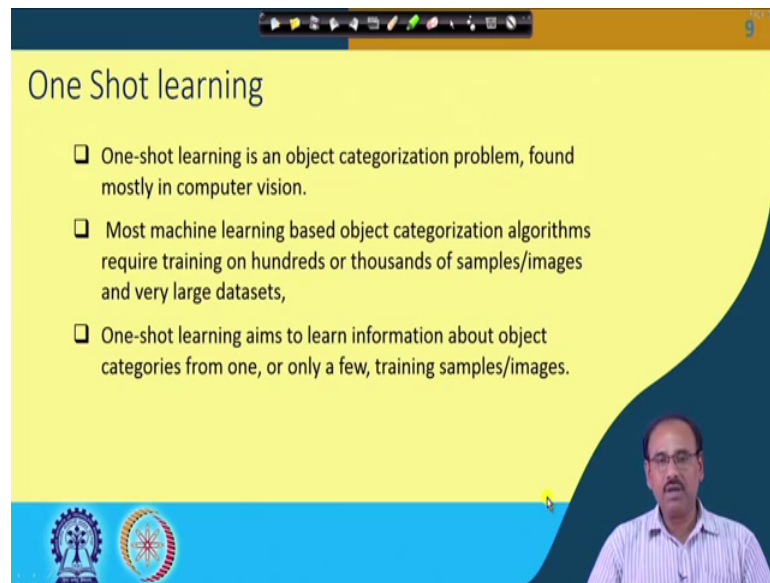
However, though it is very very important the face recognition task is not very simple in fact, there are many challenges. So, some of the challenges are as given in these photographs. You find that all these face images are photographs of the same person of course, these photographs are taken from this particular paper which is cited at the bottom.

You find that though these face images are for the same person, but it is very difficult to identify very difficult to match that their face images of the same person. The reason being when the images that are captured the illumination conditions may be different. So, the images might have been captured under different illumination conditions. Maybe the person was in different pose or the face was in different orientation.

In addition to that there the face may be in different scale as well not only orientation and pose even the scale or the magnification may be different. There may be very such various such other variational conditions, which makes the images of the same person or the face images of the same person to appear to be different. So, these are the challenges to make your face recognition system work or even to design a face recognition system.

And of course, one of the most important challenges the availability of a limited data set for training of the system. So, that the system can identify the system can recognize face in persons from their face images with high accuracy.

(Refer Slide Time: 09:52)



One Shot learning

- ❑ One-shot learning is an object categorization problem, found mostly in computer vision.
- ❑ Most machine learning based object categorization algorithms require training on hundreds or thousands of samples/images and very large datasets,
- ❑ One-shot learning aims to learn information about object categories from one, or only a few, training samples/images.

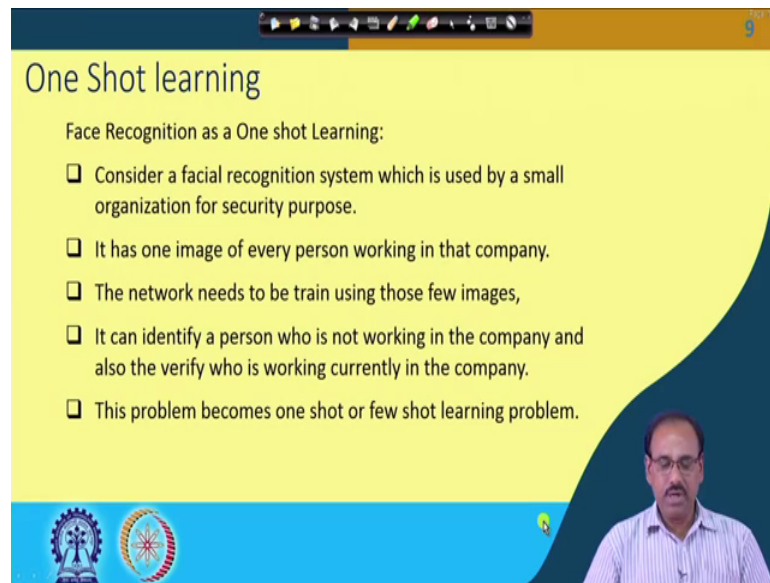
The slide features a yellow background with a blue footer. In the footer, there are two circular logos on the left and a video inset of a man in a striped shirt on the right. A navigation bar is visible at the top of the slide.

So, these are the different challenges. So, given this, the kind of training mechanism or the learning mechanism that has to be employed is what is known as one shot learning or one short training. So, what is this one shot learning? This one shot learning is a very very its a problem or categorization problem, which is found mostly in computer vision applications.

As we have seen before that while training of deep neural networks or training of machine learning algorithms. That that deep learning algorithms are data hungry; that means, you need millions and millions of data for training of the neural network. But that may not be true, when we think of designing a face authentication system or a face recognition system.

So, one short learning actually tries to learn or tries to learn the face images or learn about the object categories from one or maybe very few training samples which are available. Now why the samples are very few? What is the problem in getting large number of images that we use in other machine learning algorithms?

(Refer Slide Time: 11:15)



One Shot learning

Face Recognition as a One shot Learning:

- ❑ Consider a facial recognition system which is used by a small organization for security purpose.
- ❑ It has one image of every person working in that company.
- ❑ The network needs to be train using those few images,
- ❑ It can identify a person who is not working in the company and also the verify who is working currently in the company.
- ❑ This problem becomes one shot or few shot learning problem.

The slide features a yellow background with a dark blue curved shape on the right side. At the bottom left, there are two circular logos. At the bottom right, there is a small video inset of a man with glasses and a mustache, wearing a light-colored shirt, speaking.

You think of a situation that a face recognition system or face based authentication system which is used by very small organization for security purpose. And that particular organization has a small number of employees working in that particular organization. So, what the organization may try to do is. At the beginning when a person joins that organization.

It may try to take, which you all have faced that when you apply for something the organization asks for your passport photograph right. When you apply for whether for admission or opening a bank account or applying for some examination or something apply for identity card you are asked to provide your passport size photograph.

So, here again in the same manner when an employee joins that organization the organization may take a very nice photograph for that person. So, you find that as the organization is small, the number of such photographs, which will be used for training purpose. Of course, is very very small not many photographs of a particular person. So, the network or the authentication system the recognition system has to be trained using only those few photographs right or maybe some more photographs that can be taken at random.

So, using that it is expected that, the system will be able to identify persons who are employees of that organization. Who will have maybe free access to all different facilities of that organization. Or it will also be able to identify the persons who are not

employees of that organization, but may be customers or even in some cases service providers.

So, there are different categories of people may be visiting that particular office and the face based authentication system. Should be able to identify, whether the person is an employee or the person is a customer or the person is a service provider or which category is. And here you find that because the number of images that you have are very few for training, the network or the authentication system.

So, the normal machine learning algorithms cannot be applied in this case. Rather this particular problem of learning becomes a one shot or maybe we can call it a few shot learning problem; where the learning has to be done using very few images which are available.

(Refer Slide Time: 13:55)

FaceNet

FaceNet learns an embedding function $f(x)$; $\|f(x)\|_2 = 1$

$$f: \underline{x} \in R^{M \times N} \rightarrow R^d; \quad d < M \times N$$

Take two images \underline{x}_i and \underline{x}_j

$$\|f(\underline{x}_i) - f(\underline{x}_j)\|^2$$

Small if \underline{x}_i and \underline{x}_j are same person
Large otherwise

So, what does this face net actually do or the face net the face recognition system the face net actually learns an embedding function? So, $f(x)$ where x is an input image or an input face image. So, from this face image this face net learns an embedding function $f(x)$ with the constant that mod of $f(x)$ should be equal to one that is for normalization purpose.

Or in other words we can write that the face net learns a function f ; which maps your input face image x of course, here we assume that this input face image x is of size M by

N ; so M by N number of pixels. So, it learns a mapping or learns an embedding of an input facing as x to R d dimensional feature vector. So, it maps x which belongs to R M by N to R d .

And if d is less than M by N then what we are gaining is an image of size M by N is being represented or is being embedded which is known as embedding. It is being embedded into a d dimensional space. So, you remember that in our earlier discussions of this particular course, we have talked about the feature vectors. That is we said that given an image if I can represent that image by a feature vector of dimension d .

Then basically what we are doing is we are mapping that image. Or transforming that image into a vector in a d dimensional space and exactly that is what is being done here. So, in traditional machine learning applications that d dimensional vector has to be decided that what will be the components of these those d dimensional vectors.

Some components may represent intensity, some component may represent colour, some component may represent the texture, some component may represent shape of that particular region and so on. So, it has to be handcrafted and the advantage we said that moment I transform an image into a vector in a d dimensional vector. That means, I am representing that image by a point in a d dimensional space. And exactly the same thing is done over here.

But here this embedding function, which is f that embeds an input image to a d dimensional vector or it represents that input image embeds it in this input image to our d dimensional space as a vector this is not handcrafted. But the machine or the deep neural network has to learn this embedding function f ; through by making use of the training data which is made available to the system.

And the approaches again as we have seen before that we have to define a loss function and the gradient descent on. On that loss function in a back propagation way, will train the neural network in such a way, that this deep neural network will learn this embedding function f right.

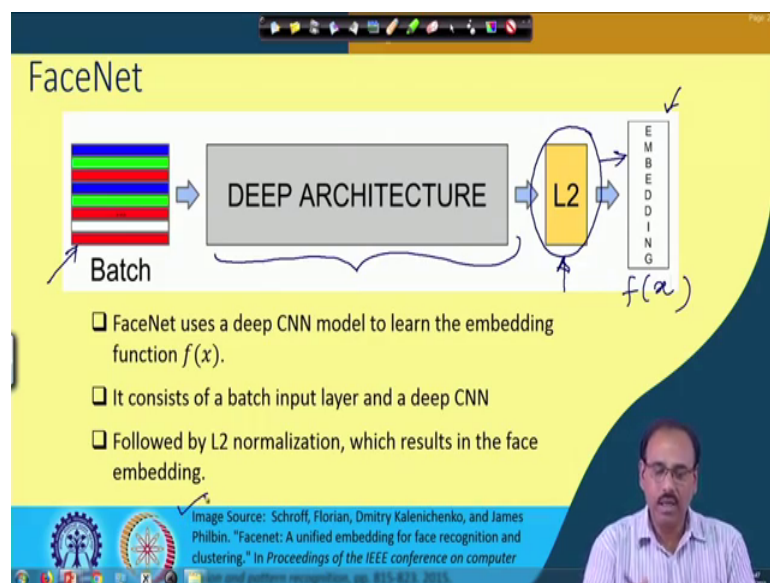
So, the advantage that we said even we have said earlier that if I take two images say x_i and x_j . So, x_i is an image x_j is an image both of them are embedded in that d dimensional space may be an ingredients space through this embedding function f . So, f

of x_i is the embedding of image x_i and f of x_j is the embedding of image x_j . So, x_i minus x_j mod of that square this indicates the squared Euclidean distance between these two embeddings $f(x_i)$ and $f(x_j)$.

Now if x_i and x_j these two images are images of the same person may be in different orientation may be in different pose may be under different illumination whatever. But what is expected is if x_i and x_j are the images of the same person. Then the distance between $f(x_i)$ and $f(x_j)$ should be very small or in other words mod of $f(x_i)$ minus $f(x_j)$ square which is actually the squared Euclidean distance between $f(x_i)$ and $f(x_j)$.

That should be small if x_i and x_j are images of the same person but if x_i and x_j are images of different persons. Then we want we what we want is that this squared Euclidean distance between $f(x_i)$ and $f(x_j)$ should be large. And that is how you measure the similarity whether the two images are similar or not.

(Refer Slide Time: 19:29)



So, given this now what we what this face met the architecture is something like this. It has an input bachelor where you feed in the training data or even the inference data the test data. The face net is actually a deep convolution neural network architecture which is a deep neural network. And, then I have an L 2 normalization layer because you find that we had put a constant that this embedding function $f(x)$ should be such your embedding should be such that mod of $f(x)$ should be equal to 1.

And that is what is achieved through this normal L 2 normalization layer L 2. And after this what we get is the embedding of your input in a x. So, this is where I get f of x. So, this phase net it uses a deep convolutional neural network model to learn the embedding function f x. And it consists of an input bachelor and followed by a deep convolution layer, which is followed by an L 2 normalization layer or normalization operation.

And finally, we get an embedding of the input image into our d dimensional a d dimensional Euclidean space. So, here basically what I am getting is our d dimensional vector the vector which is embedding of your input image. So, this is how the face net works. And once your embedded vector of the embedding is complete, then your decision is taken in this embedding domain only that is in the d dimensional space.

I do not have to take decision in the original image space, where the images are of size M by N. So, this work which we are going to discuss today was presented in this particular paper face net are unified embedding for the face recognition and clustering as has been given over here. So, let us see how this network actually works.

(Refer Slide Time: 21:52)

Training/Triplet Loss

Embeddings

anchor → CNN → Embedding

positive → CNN → Embedding

negative → CNN → Embedding

Shared weights

Triplet Loss

- Minimize **triplet loss function** :-loss function using **three** images
- An anchor image A, a positive image P (same person as the anchor), and a negative image N (different person than the anchor).
- Distance $d(f(A), f(P))$ must be less than or equal to the distance $d(f(A), f(N))$

Image Source: <https://omoindrot.github.io/triplet-loss>

So, the first and foremost thing is of course, we have to train this network. And for training of this network what we need is, we have to define a loss function. And that training has to be done by and gradient descent approach our gradient descent algorithm will try to minimize the loss function. So, the loss function which is defined in this face net is what is known as a triplet loss and that is used for the training purpose.

So, the triplet loss when you define a triplet loss function the triplet loss function actually makes use of three images one is anchor image. So, what you say it is the particular example that I have taken here for a small organization where we can have passport size photographs. You can ask the employees to give their passport size photographs right.

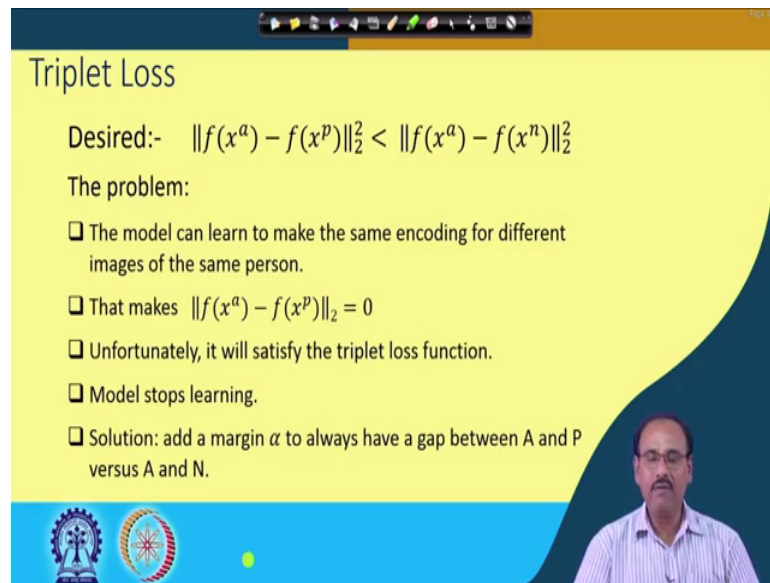
And those images can act as anchor images. And then I need two more images one of the and so, those are anchor images just say A. So, I call them as anchor images A right. And I need two more images that is another image of the same person, but which is different from the anchor image. And we call them as image P or if this is a positive image as this is image of the same person.

And I need image of another person not the same person which we call as negative image N. Then once I have this then if I compute the distance between $f A$ and $f P$ that is $f A$ is the embedding of the anchor image A. $f p$ is the embedding of the positive image P that is A and P are for the same from the same person. And $f N$ is the embedding of the negative image that is image of not the same person as that other.

And it is not of the same person so, that is what is my negative image. So, what you do is you compute the distance between $f A$ and $f P$, which is $d f A, d f P$ and you compute the distance between $f A$ and $f N$ that is the distance between the anchor image and the positive image.

And, the distance between the anchor image and the negative image. So; obviously, in this case what is desired is that the distance between anchor image and the positive image should be less than the distance between the anchor image and the negative image right.

(Refer Slide Time: 24:42)



Triplet Loss

Desired:- $\|f(x^a) - f(x^p)\|_2^2 < \|f(x^a) - f(x^n)\|_2^2$

The problem:

- The model can learn to make the same encoding for different images of the same person.
- That makes $\|f(x^a) - f(x^p)\|_2 = 0$
- Unfortunately, it will satisfy the triplet loss function.
- Model stops learning.
- Solution: add a margin α to always have a gap between A and P versus A and N.

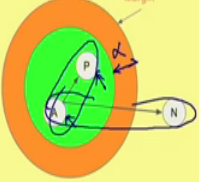
Or in other words what we desire is that mod of $f(x^a) - f(x^p)$ is the anchoring is minus $f(x^p)$ square that is the squared Euclidean distance must be less than $f(x^a) - f(x^n)$ square right. And, that is quite obvious because x^a and x^p are for the same person, x^a and x^n therefore different persons. Now, the problem is the model can learn to make the same encoding for different images belonging to the same person.

And if it is so, then if x^a minus, $f(x^p)$ because it is the same in embedding so, that will be equal to 0. And, if it is 0, then the first condition which is desired that is $f(x^a) - f(x^p)$ squared less than $f(x^a) - f(x^n)$ squared that will be true always. And, if it is always true; that means, there will be no further learning as there will be no triplet loss function triplet loss will always be 0 so, the model stops learn. The solution to this is by putting a margin say alpha. So, what this alpha does is, it maintains a gap between A and P and A and N.

(Refer Slide Time: 25:59)

Triplet Loss

Thus : $\|f(x^a) - f(x^p)\|_2^2 + \alpha < \|f(x^a) - f(x^n)\|_2^2$



The loss that is being minimized is then

$$L = \sum_{i=1}^N [\|f(x^a) - f(x^p)\|_2^2 - \|f(x^a) - f(x^n)\|_2^2 + \alpha]$$

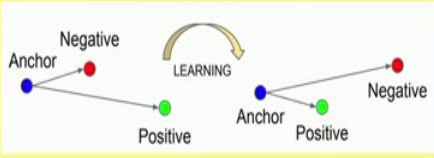
Image Source : <https://medium.com/@ahmdtaha/facenet-a-unified-embedding-for-face-recognition-and-clustering>

So, what we have a triplet loss function must be something like this. That f of x^a minus f of x^p squared plus α must be less than f of x^a minus f of x^n squared right. So, diagrammatically I can put it this way that this is my anchor, this is my positive sample, the distance between these two is this. And the distance between anchor and negatives sample is this ok. And this is what is my margin, that is what is α right.

So, I want that the distance between these two must differ at least by α . And that leads to a loss function, which is given by this that L equal to f of x^a minus f of x^p squared minus f of x^a minus f of x^n squared plus α take the summation over all possible triplets are x^a , x^p and x^n . And you try to minimize this loss while training of your neural network or the face net. So, this is what this is how the face net is trained.


(Refer Slide Time: 27:15)

Triplet Loss



- The Triplet Loss minimizes the distance between an anchor and a positive.
- Maximizes the distance between the anchor and a negative.
- Compact clusters of embedding of same person.
- Pictures of the same person become close to each other.
- Pictures of different persons are far from each other.

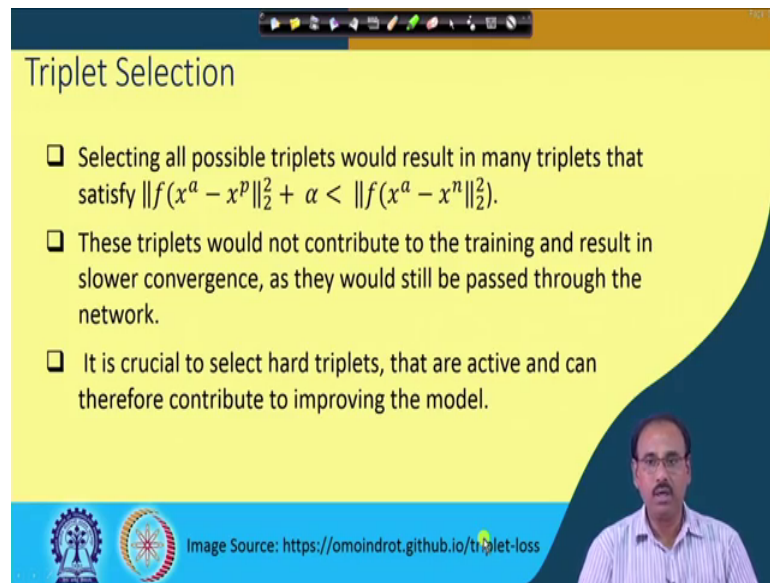
Image Source: Schroff, Florian, Dmitry Kuznetsov, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815-823. 2015.



So, once the face net is trained actually what this face net tries to do is. It tries to make because I want that the distance of the anchor from the positive sample should be low and the distance of the anchor from the negative sample should be high. So, face net actually tries to make compact clusters of the images belonging to the same person.

Whereas, the clusters of the images belonging to different persons should be wide a part that is what is being tried by the face net. So, images belonging to the same person forms a compact cluster; so every pair of images within the same cluster the distance between every such pair should be very very small. So, this is how triplet loss is used for training of the face net.

(Refer Slide Time: 28:04)



Triplet Selection

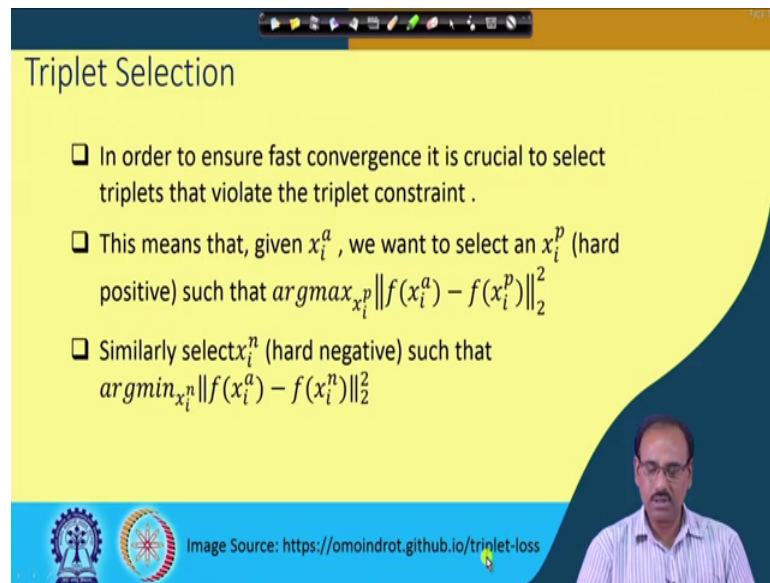
- ❑ Selecting all possible triplets would result in many triplets that satisfy $\|f(x^a - x^p)\|_2^2 + \alpha < \|f(x^a - x^n)\|_2^2$.
- ❑ These triplets would not contribute to the training and result in slower convergence, as they would still be passed through the network.
- ❑ It is crucial to select hard triplets, that are active and can therefore contribute to improving the model.

Image Source: <https://omoindrot.github.io/triplet-loss>

Now, the problem that becomes is that how to select this triplet. Because, if we have a database of large number of images selecting of all possible triplets becomes very very difficult I mean it is extremely time consuming. So, what you try to do is you form the mini batches and you select the triplets within the mini batches. Or your computation time will be low because selection of the triplets is very very important.

And there what we have to do is, we have to try to find out the hard positive triplets and the hard negative triplets. That means, we should try to find out the pair of hard positive triplets or a positive triplet whose distance for the anchor is very large. And, I have to find out a negative triplet whose distance from the anchor is small ok. And these are the pairs which actually gives you a faster learning of this network. So, selection of the triplet is also very very important.

(Refer Slide Time: 29:14)



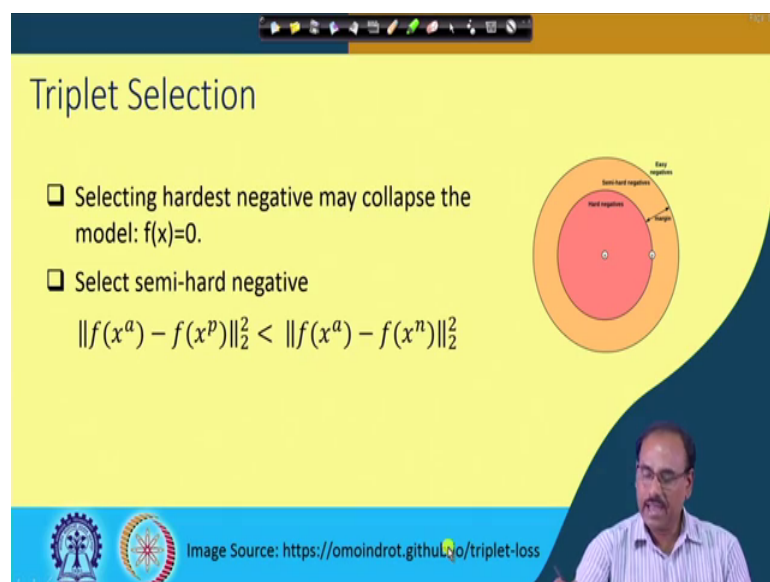
Triplet Selection

- ❑ In order to ensure fast convergence it is crucial to select triplets that violate the triplet constraint .
- ❑ This means that, given x_i^a , we want to select an x_i^p (hard positive) such that $\operatorname{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$
- ❑ Similarly select x_i^n (hard negative) such that $\operatorname{argmin}_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$

Image Source: <https://omoindrot.github.io/triplet-loss>

So, in many cases what you do is instead of trying to find out the hardest negative sample you try to find out a semi hard negative. Sample that just satisfies that distance between the anchor and the positive sample is just less than the distance between the anchor. And, the negatives are a negative sample and you forget about the margin that we discussed about earlier.

(Refer Slide Time: 29:48)



Triplet Selection

- ❑ Selecting hardest negative may collapse the model: $f(x)=0$.
- ❑ Select semi-hard negative

$$\|f(x^a) - f(x^p)\|_2^2 < \|f(x^a) - f(x^n)\|_2^2$$

Diagram illustrating triplet selection regions: Easy negatives (outermost), Semi-hard negative (middle), Hard negative (innermost), and Margin (innermost boundary).

Image Source: <https://omoindrot.github.io/triplet-loss>

So, while training I can try to look for a negative sample, which may be within this margin. So, these are the samples which are actually semi hard negative samples it may

not be hard negative samples because selection of those may be easier. So, once you have all this and your network is trained. Then for given any input image output becomes a vector which is an embedding of your input image to our d dimensional space.

(Refer Slide Time: 30:20)

Face Verification

- Pass the reference image and the query image through the embedding network.
- Use the distance between them for verification.

$$d(\text{reference}, \text{query}) = \|f(\text{reference}) - f(\text{query})\|_2^2$$

The diagram illustrates the process for both a reference image and a query image. Each image is processed through a series of three blue 3D blocks representing convolutional layers, which are then flattened into a single row of circles representing the embedding vector. The reference image's embedding is labeled $f(\text{reference_image})$ and the query image's embedding is labeled $f(\text{query_image})$.

Image Source : <https://www.coursera.org/learn/convolutional-neural-networks?specialization=deep-learning>

And then your recognition or verification can be done in that space only. Say for example, the verification is given an image or given two images I have to say whether that two images belong to the same person or not. So, what you do is, you give the image pass the image to the same network. And you get the embeddings of the two images if the distance between the two embeddings is less than certain threshold value.

I can always say that they belong to the same person. And if one of them is actually an anchor image I can verify that the other image is image of that person only. Similarly in case of recognition, I can make use of KNN classifier given an image. You get its embedding find out you select k number of embedding, which are the nearest to it and then you go for a fitting encoding mechanism.

So, within this k number of embedding whichever is in majority you recognize this image to be the image of that person which is the majority. So, that becomes your classification or recognition problem. And in the same manner, we can also find out that or we can also cross their face images in that embedding space that is all the embedded vectors which are very close to each other they form one cluster. And which are not close they become part of different aspects.

So, this is how the face net actually works. And as we have seen that face net has face recognition of a face; verification techniques has a wide applications in today's digital world. So, we stopped history recognition today. In our next class in our next lectures, we will talk about other applications of the deep neural network.

Thank you.