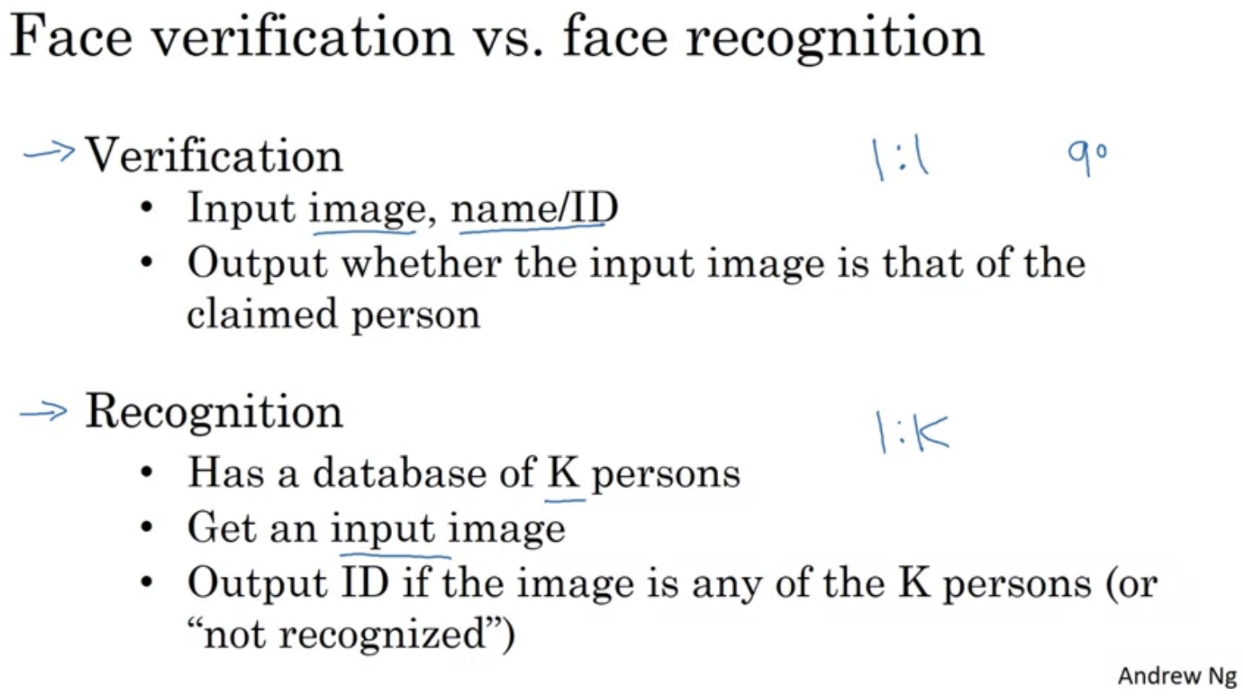
Convolutional Neural Networks DeepLearning AI

Face Recognition

Alec Dewulf

What is Face Recognition?

Liveness detection is used to predict whether the image is human or not human. It can be implemented with supervised learning.



Face verification vs. face recognition

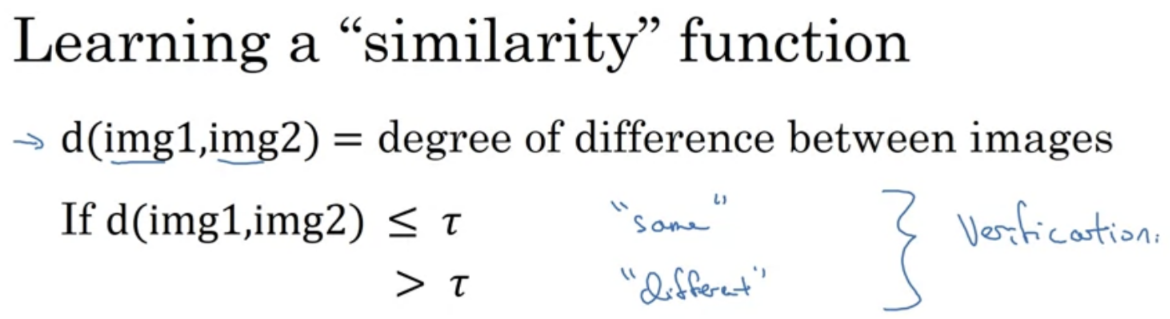
→ Verification
- Input image, name/ID
- Output whether the input image is that of the claimed person

→ Recognition
- Has a database of K persons
- Get an input image
- Output ID if the image is any of the K persons (or "not recognized")

Andrew Ng

One-shot Learning

The one-shot problem is being able to recognize an image given a single example of that that image (i.e a person's face).



Learning a “similarity” function

→ $d(\text{img1},\text{img2})$ = degree of difference between images

If $d(\text{img1},\text{img2}) \le \tau$ “same”

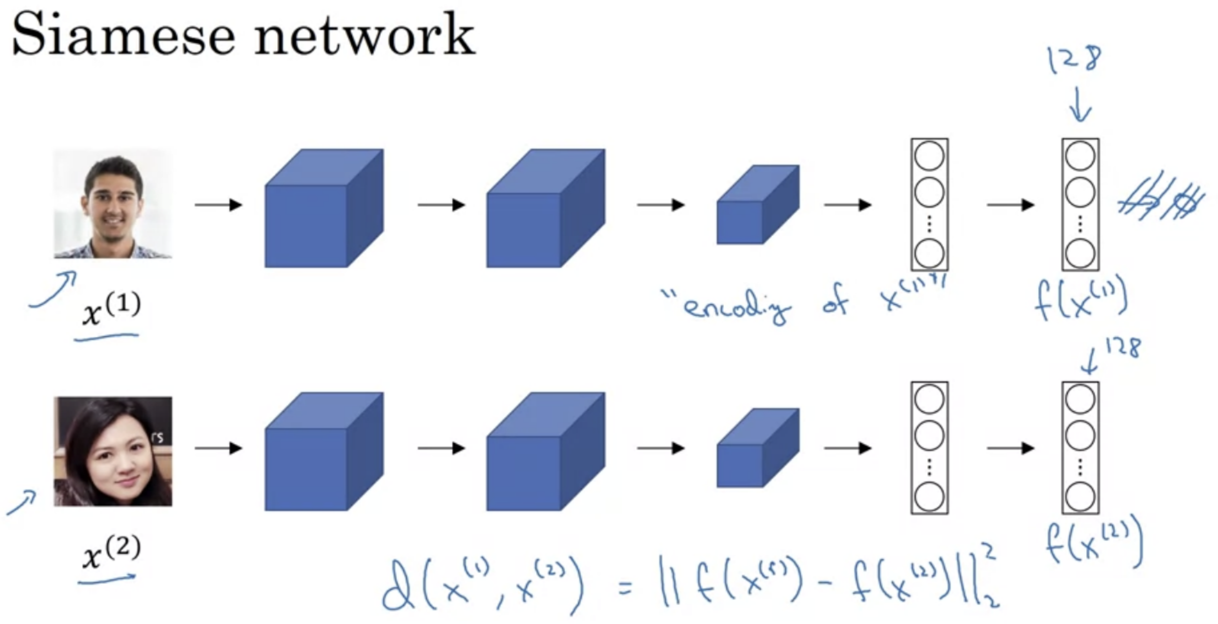
$> \tau$ “different”

} Verification

If the distance between two images is greater than some bound tau, we define them as different. If we are using this for user recognition, hopefully the d function will output a number that is much smaller than the others for the correct class.

Siamese Network

Think of the first part of the network as converting the image into a vector encoding. We simply take out the softmax layer.

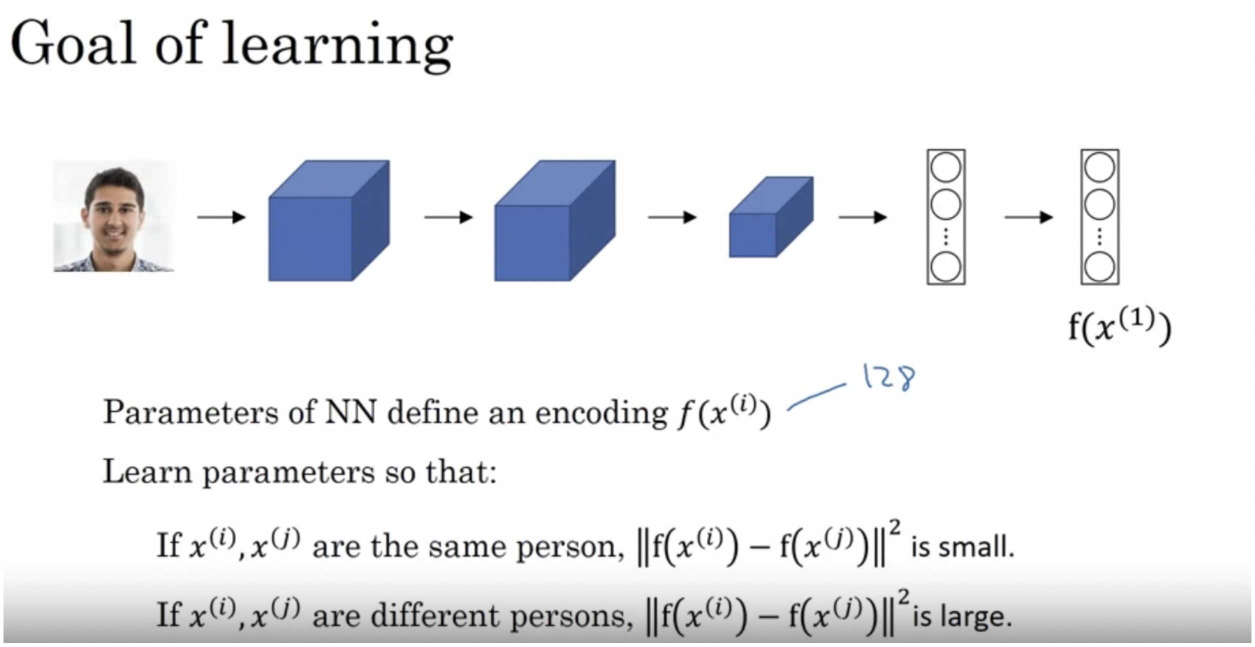


Siamese network

$$d(x^{(1)}, x^{(2)}) = \| f(x^{(1)}) - f(x^{(2)}) \|_2^2$$

You can compute the distance between the images as the squared norm of the output vectors.

We want to learn parameters that minimize this distance function when the images we are comparing contain the same image.



Goal of learning

Parameters of NN define an encoding $f(x^{(i)})$ — 128

Learn parameters so that:

If $x^{(i)}, x^{(j)}$ are the same person, $\left\|f(x^{(i)}) - f(x^{(j)})\right\|^2$ is small.

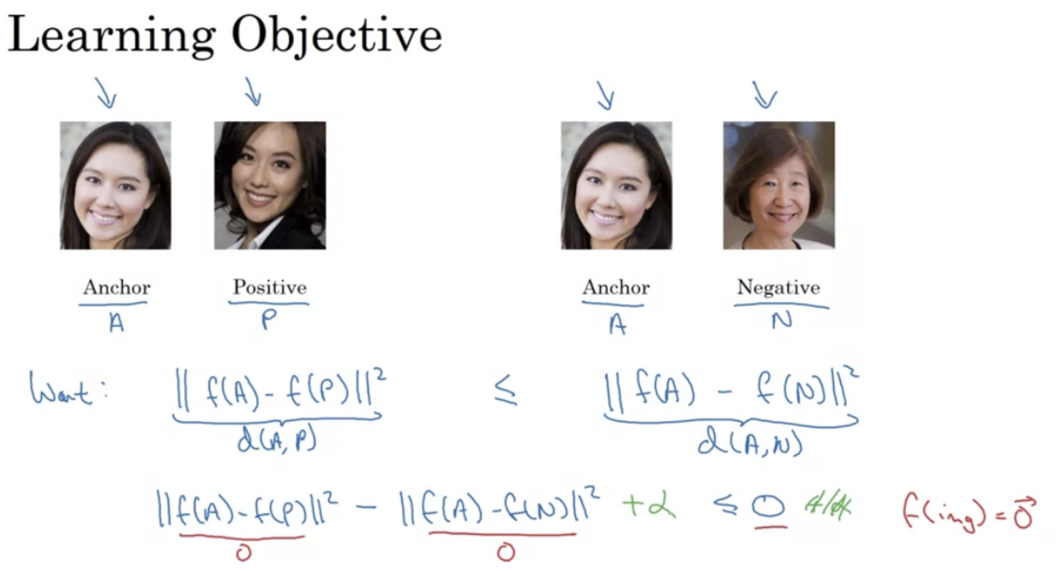
If $x^{(i)}, x^{(j)}$ are different persons, $\left\|f(x^{(i)}) - f(x^{(j)})\right\|^2$ is large.

You can use backpropagation to vary these parameters and minimize this error.

Triplet Loss

This is a loss function for a Siamese style network that outputs a vector that we use to determine distance from an image.

Triplet loss looks at three images at one time: positive, negative, and anchor images. The positive image is another correct class and the negative is a negative class.

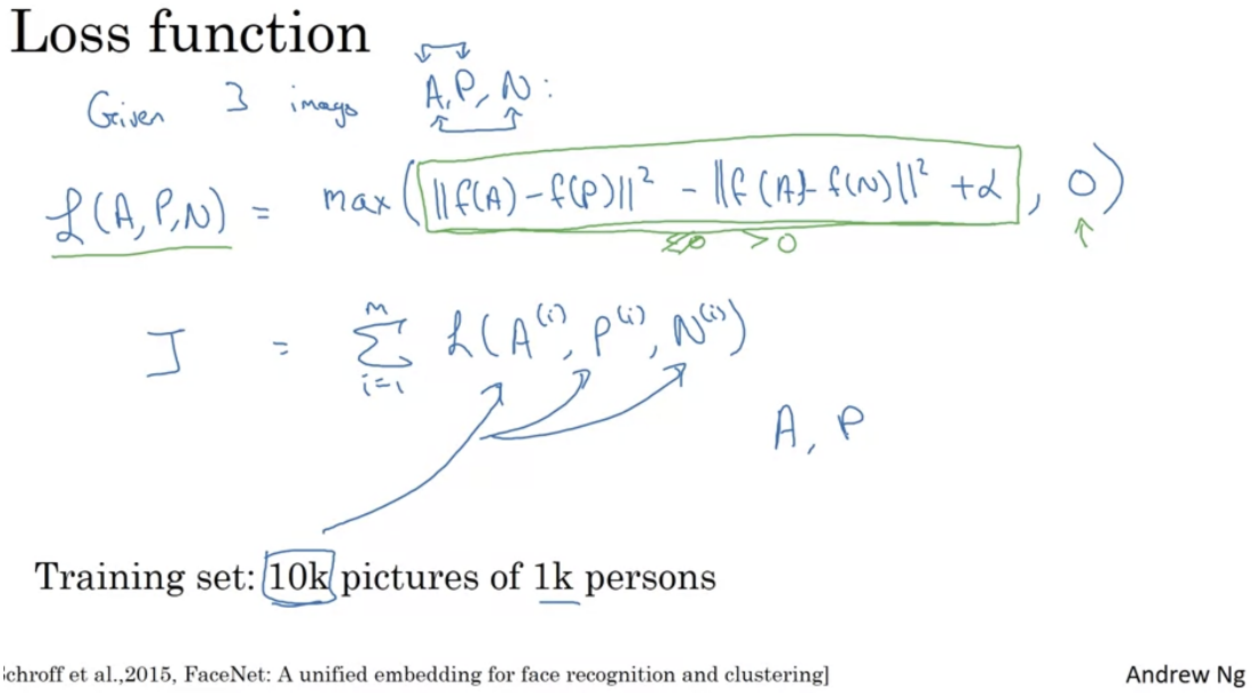


Learning Objective

Want: $\underbrace{\|f(A) - f(P)\|^2}_{d(A,P)} \le \underbrace{\|f(A) - f(N)\|^2}_{d(A,N)}$

$\|\underbrace{f(A) - f(P)}_{0}\|^2 - \|\underbrace{f(A) - f(N)}_{0}\|^2 + \alpha \le \underline{0}$ [illegible] $f(img) = \vec{0}$

We want to ensure the network doesn't just set all the econdings equal to each other or to zero (which would lower the cost but not be helpful. This is what the alpha parameter, called "margin" accomplishes.



Loss function

Given 3 imags A, P, N:

$$\mathcal{L}(A,P,N) = \max\left(\boxed{\|f(A)-f(P)\|^2 - \|f(A)-f(N)\|^2 + \alpha}, 0\right)$$

$\cancel{\leq 0}$ >0

$$J = \sum_{i=1}^{m} \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$$

A, P

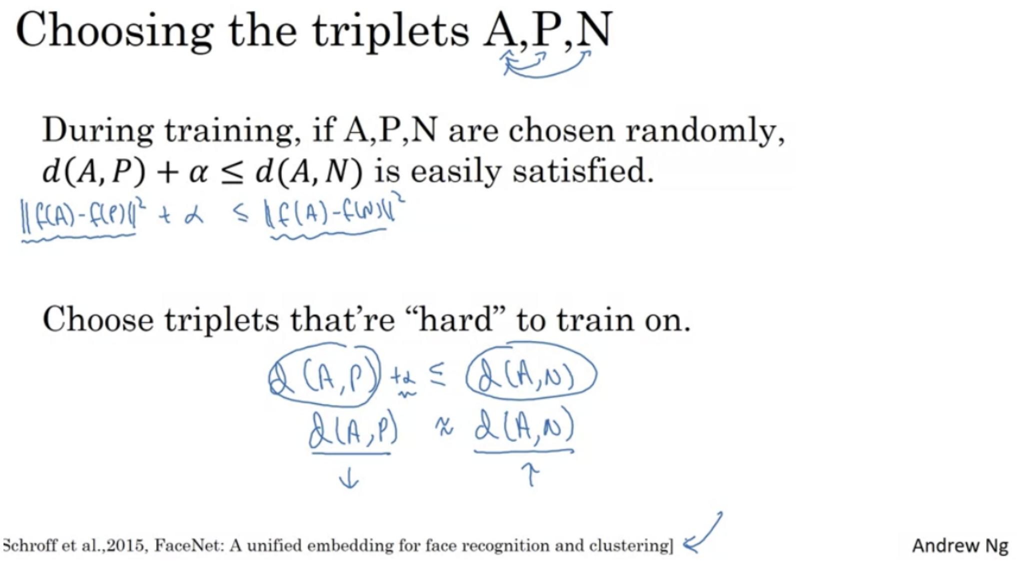
Training set: 10k pictures of 1k persons

chroff et al.,2015, FaceNet: A unified embedding for face recognition and clustering]

Andrew Ng

You still need multiple pictures of the same person (for the Pis).

You want to choose image triplets such that the distance from the anchor and the positive is close to the distance from the anchor to the negative. You want the negative to be similar to the image we are trying to classify.



Choosing the triplets A,P,N

During training, if A,P,N are chosen randomly,
$d(A,P) + \alpha \leq d(A,N)$ is easily satisfied.

$$\|f(A)-f(P)\|^2 + \alpha \leq \|f(A)-f(N)\|^2$$

Choose triplets that're "hard" to train on.

$$d(A,P) + \alpha \leq d(A,N)$$

$$\underset{\downarrow}{d(A,P)} \approx \underset{\uparrow}{d(A,N)}$$

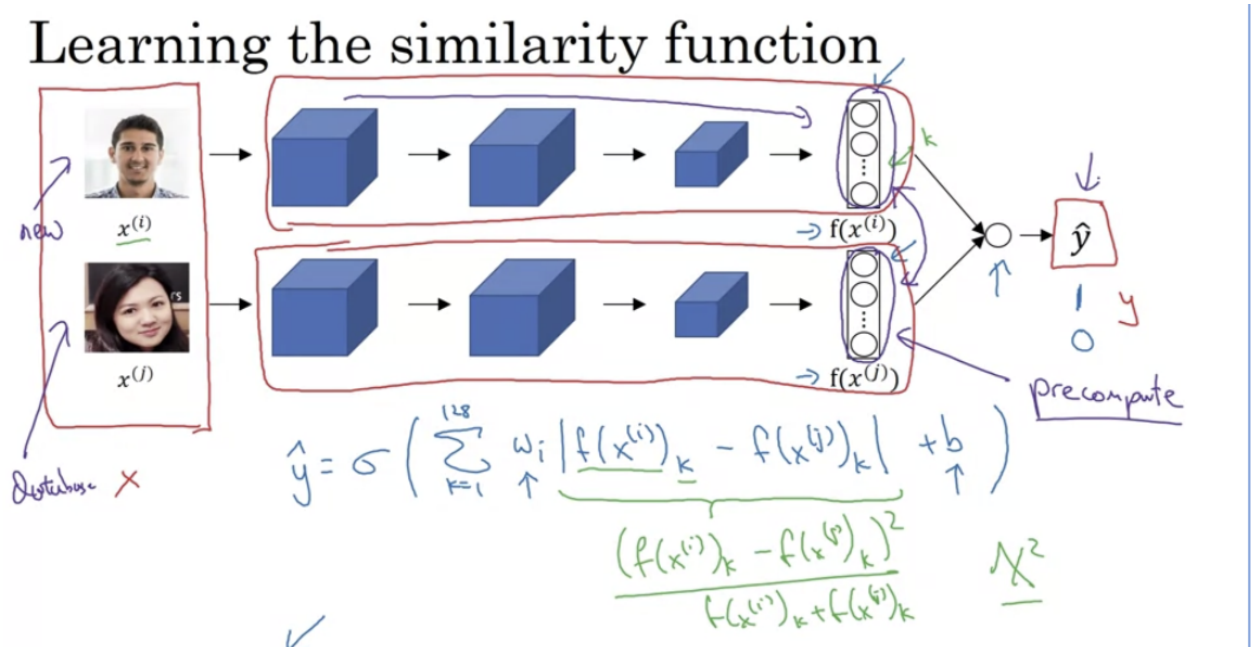
Schroff et al.,2015, FaceNet: A unified embedding for face recognition and clustering]

Andrew Ng

You can use this cost function to train networks to output a good encoding for faces in a face recognition system. Very large datasets (~1 million or ~10 million) are used for this in industry. Luckily there are pretrained models online.

Face Verification and Binary Classification

This is an alternate way to train a network for facial recognition. We can treat it as a simple binary problem (i.e. output a 1 if it's the face we're looking for and 0 otherwise).



Learning the similarity function

new
$x^{(i)}$
$x^{(j)}$
Database ×
k
$f(x^{(i)})$
$f(x^{(j)})$
$\hat{y}$
1
0
y
precompute
$\hat{y} = \sigma\left(\sum_{k=1}^{128} w_i \left| f(x^{(i)})_k - f(x^{(j)})_k \right| + b\right)$
$\frac{(f(x^{(i)})_k - f(x^{(j)})_k)^2}{f(x^{(i)})_k + f(x^{(j)})_k}$
χ^2

You should precompute the definition encoding and store it for each employee. You can then compare the output from the current image (where an employee is looking at a camera) with the stored encoding. This way you don't need to store images or run an image through the model twice every time.