Face Recognition

Wednesday, September 9, 2020 11:54 AM

① What is face recognition

- Identifies a face from photo or a video
- Liveness detection: Use video to tell if the person is real or an image. It can be learned by supervised learning
- Face verification
  Given an image and ID or name, output if the image is same as the claimed person.
- Face Recognition
  It has a database of K persons.
  Given an input image, output the ID from the database or if not recognised
- A face verification system can be used as a face recognition system if the accuracy is high (~99.9%)
- Recognition is harder than verification

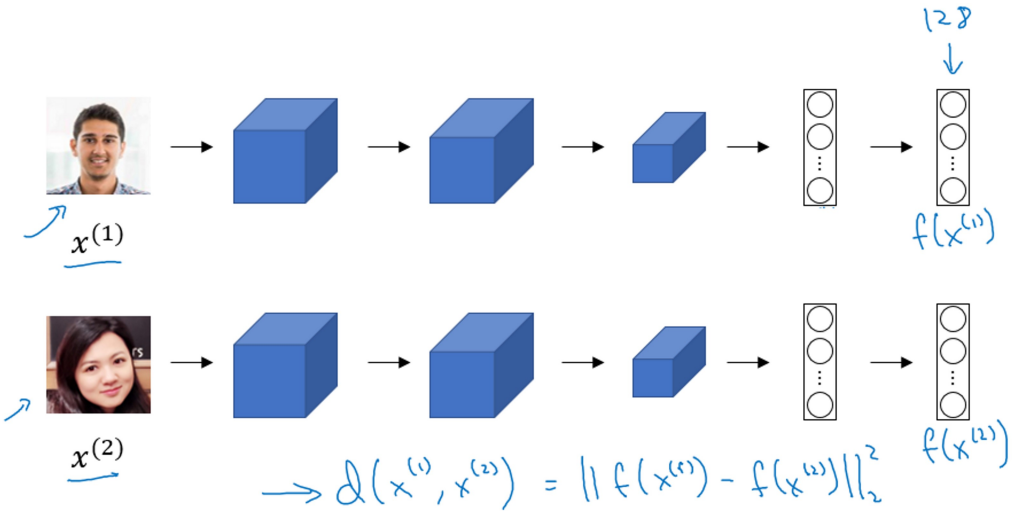
② One Shot Learning

- Learn from just one training example to recognize the person again
- Learning a "similarity" function
  - → d(img1, img2) = degree of difference between images
  - → If $d(img1, img2) \leq \tau$ "same", $> \tau$ "different" } verification
    where τ is a hyperparameter
  - → This robust to new images since adding a new output to a softmax function every time a new employee joins is not feasible

③ Siamese Network

- Used to implement similarity function
- A siamese network takes multiple

- Used to implement similarity function
- A siamese network takes multiple inputs with two or more networks with the same architecture and parameters



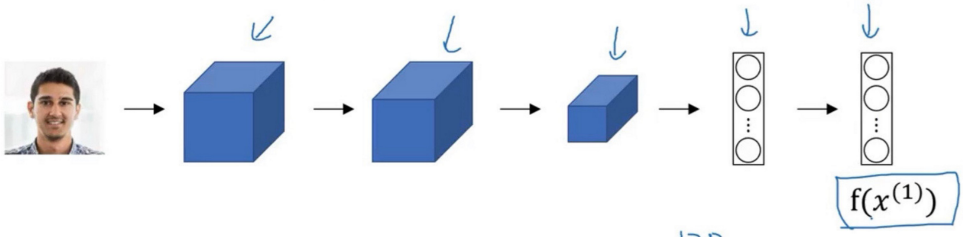
→ It takes an image and outputs an encoding of it in the form of a vector (eg: shape (128,))

→ Loss function is

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

→ If $x^{(1)}$ & $x^{(2)}$ are the same person, $\delta(x^{(1)}, x^{(2)})$ will be low or else, it will be high

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, $\|f(x^{(i)}) - f(x^{(j)})\|^2$ is small.

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

Andrew Ng

④ Triplet Loss

→ It is the loss function used in siamese network

→ Learning objective is to get difference between Anchor image and a positive

→ Learning objective is to get difference between Anchor image and a positive or negative image

→ The loss for a positive image should be less than loss from negative image

Learning Objective

$d(A,P) = 0.5$ $\quad$ $d(A,N) = 0.7$

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

$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha \leq 0$ $\quad$ $f(\text{img}) = \vec{0}$

α = margin

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

Andrew Ng

→ $\|f(A) - f(P)\|^2 <= \|f(A) - f(N)\|^2$

$\Rightarrow \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 <= 0$

But zero is a trivial solution
To prevent NN from choosing zero, we include α (a small number) as a hyperparameter (also called margin)

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

$\Rightarrow \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha \leq 0$

→ Loss Function

3 images A, P, N

$L(A, P, N) = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2, 0)$

if loss is negative, we truncate it to zero

Cost $\Rightarrow J = \sum_i (L(A[i], P[i], N[i]))$

we need multiple images of same person at least for training. To make sure we can get triplets from dataset
eg: 10k images for 1k persons

→ Choosing triplets A, P, N

eg. 10K images for 1K persons

→ Choosing triplets A, P, N

- If we choose randomly,

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

- we choose triplets that are harder to train on

⇒ when $d(A,P)$ is approximately equal to $d(A,N)$

$d(A,P) \approx d(A,N)$

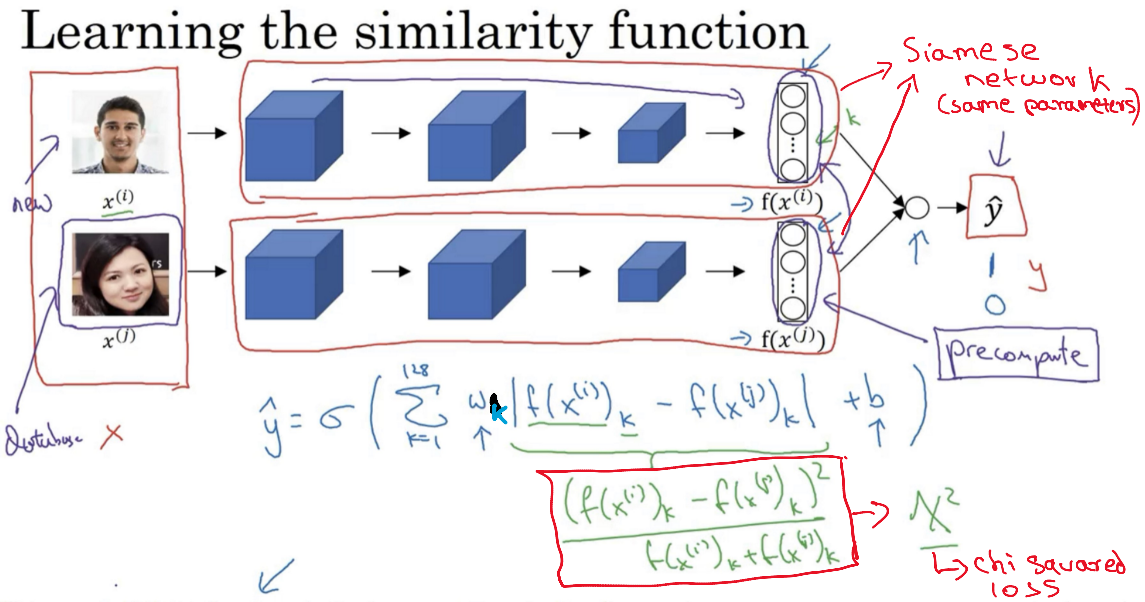
This increases learning significantly

→ Commercial models are trained on millions of images

→ Many models are available freely

⑤ Face Verification and Binary Classification

→ Face verification can also be approached as a binary classification problem



[Taigman et. al., 2014. DeepFace closing the gap to human level performance]

Andrew Ng

→ Final layer is a sigmoid layer

$$\hat{y} = \sigma\left(\sum_{k=1}^{128} w_k \left| f(x^{(i)})_k - f(x^{(j)})_k \right| + b\right)$$

→ database image encodings are precomputed

→ This version works as well as the triplet loss function