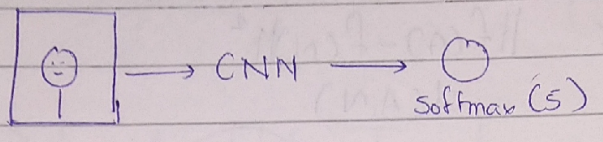
Week 4 Face Recognition System

face Verificatn

One shot Learning Problem

The model has to learn by looking at the picture once.



if you have 5 employees in your company

What if another one added to your company? You will need to change softmax o/p which is not possible

Learning a "similarity" function

d(img1, img2) = degree of difference btwn. images

if d(img1, img2) $\leq \tau$ "same"

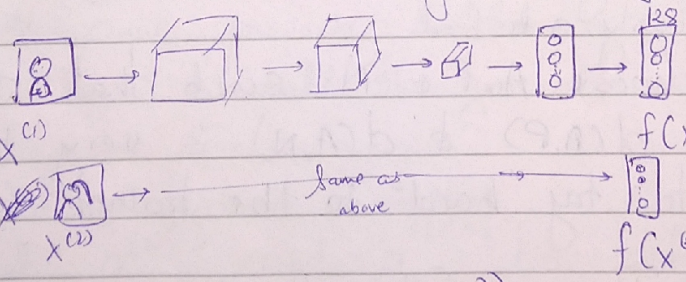
$> \tau$ "different"

This is called face Verificatn

To do face recognitn, you can compare 1 second image of person2 with all other images and o/p to model that it has the most similarity with person2 than with other persons

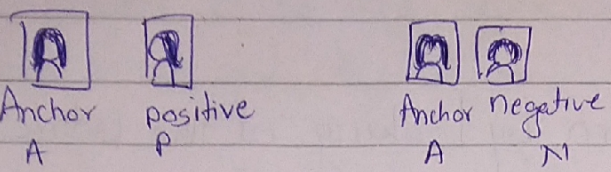
Thus, inputting fn d with pair of images, solves one shot learning problem.

To compare to images, ~~using~~ d uses siamese n/w.



Instead of $d(x^{(1)}, x^{(2)}) = \| f(x^{(1)}) - f(x^{(2)}) \|_2^2$ } if small → same, if ~~diff~~ large → diff persons

Learning Objective



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

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

α (generally written here)

Alpha is a hyperparameter which is added to not let all the encoding o/p a zero, and get a zero vector.

Triplet Loss funct^n

Given 3 images A, P, N

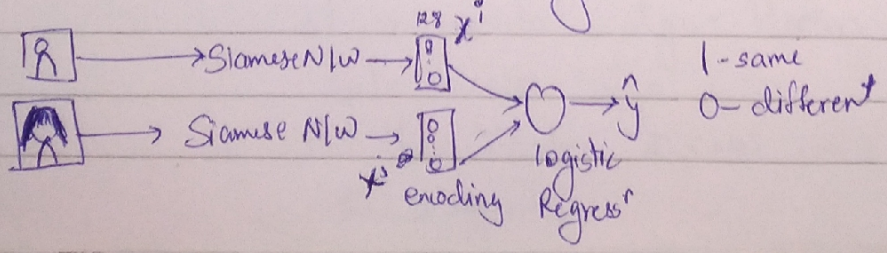
$$\mathcal{L}(A,P,N) = \max(\|f(A)-f(P)\|^2 - \|f(A)-f(N)\|^2 + \alpha, 0)$$

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

Training set: 10k Pictures of 1k Persons

~~If~~ For this you need pairs of Anchor & Positive images. For N, one can choose randomly. But if chosen randomly, the constraint of similarity can be easily satisfied.
So, one has to choose A, P & N such that the difference btwn $d(A,P)$ & $d(A,N)$ is very less, and the model has to try "hard" to be trained.

Face Verificat^n & Binary Classificat^n



This can serve as an alternative to triplet loss function

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

The sigmoid function of logistic regression

Neural Style Transfer

picture + drawing → picture in the texture of drawing

Content(C) Style(S) Generator (G)

Cost function

$$J(G) = \alpha J_{content}(C,G) + \beta J_{style}(S,G)$$

Finding Generator image G

(1) Initialize G randomly
G: 100X100X3

(2) Use Gradient descent to minimize $J(G)$

$$G = G - \frac{\partial}{\partial G} J(G)$$

Content Cost function

- Assume you use Hidden Layer l (of entire NN) to compute content cost.
- if l is small, the generated image will be similar to content image
- Use pretrained ConvNet
- Let $a^{[l](C)}$ and $a^{[l](G)}$ be activation of layer l on the images
- if $a^{[l](C)}$ & $a^{[l](G)}$ are similar, both images have similar content

So,

$$J_{content}(C,G) = \frac{1}{2} \left\| a^{[l](C)} - a^{[l](G)} \right\|^2$$

↑ maybe it was alpha

$1/4(n_H * n_W * n_C)$ gives correct o/p

Style Cost Function

One can define style as correlation between activations across layers
It checks whether a particular low level features comes with a high-level feature or not. This can be said for features of the same layer too (i.e their correlatn)

So, we make a style matrix containing correlations.
Let $a^{[l]}_{i,j,k}$ = activation at (i,j,k) $\quad l \to$ layer

$H \uparrow \quad \uparrow W \quad \uparrow C$

$G^{[l]}$ is $n_c^{[l]} \times n_c^{[l]}$

correlatn of channel k, with channel k', $\quad k, k' \in [1, n_c]$

$$G^{[l]}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a^{[l]}_{i,j,k} \; a^{[l]}_{i,j,k'}$$

if $a^{[l]}_{i,j,k}$ & $a^{[l]}_{i,j,k'}$ are highly correlated then value of $G^{[l]}_{kk'}$ will be large, and small otherwise

You compute this for style & Generated image

finally the cost function will be

$$J^{[l]}_{style}(S,G) = \frac{1}{(2 n_H^{[l]} n_W^{[l]} n_c^{[l]})^2} \left\| G^{[l](S)} - G^{[l](G)} \right\|_F^2$$

↑ normalizatn constants

This is called forbenius norm.

Now, instead of using this for a single layer we use it for multiple layers

$$J_{style}(S,G) = \sum_{l} \lambda^{[l]} J^{[l]}_{style}(S,G)$$

↑ hyper parameters