deeplearning.ai

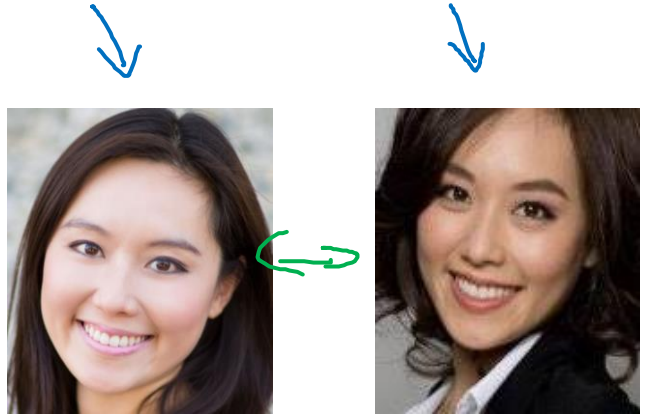
Face recognition

Triplet loss

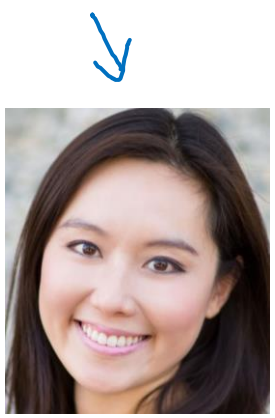
One way to learn the parameters of the neural network so that it gives you a good encoding for your pictures of faces is to define and apply gradient descent on the triplet loss function

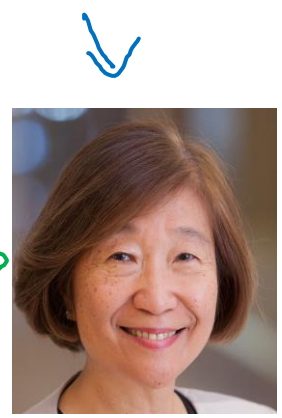
Learning Objective

So that's what having a margin parameter here does, which it pushes the anchor positive pair and the anchor negative pair further away from each other.



3 images at a time.





Anchor Positive Anchor Negative

A P A N

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

Want: $\|f(A) - f(P)\|^2 + \alpha \leq \|f(A) - f(N)\|^2$, $\alpha = 0.2$

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

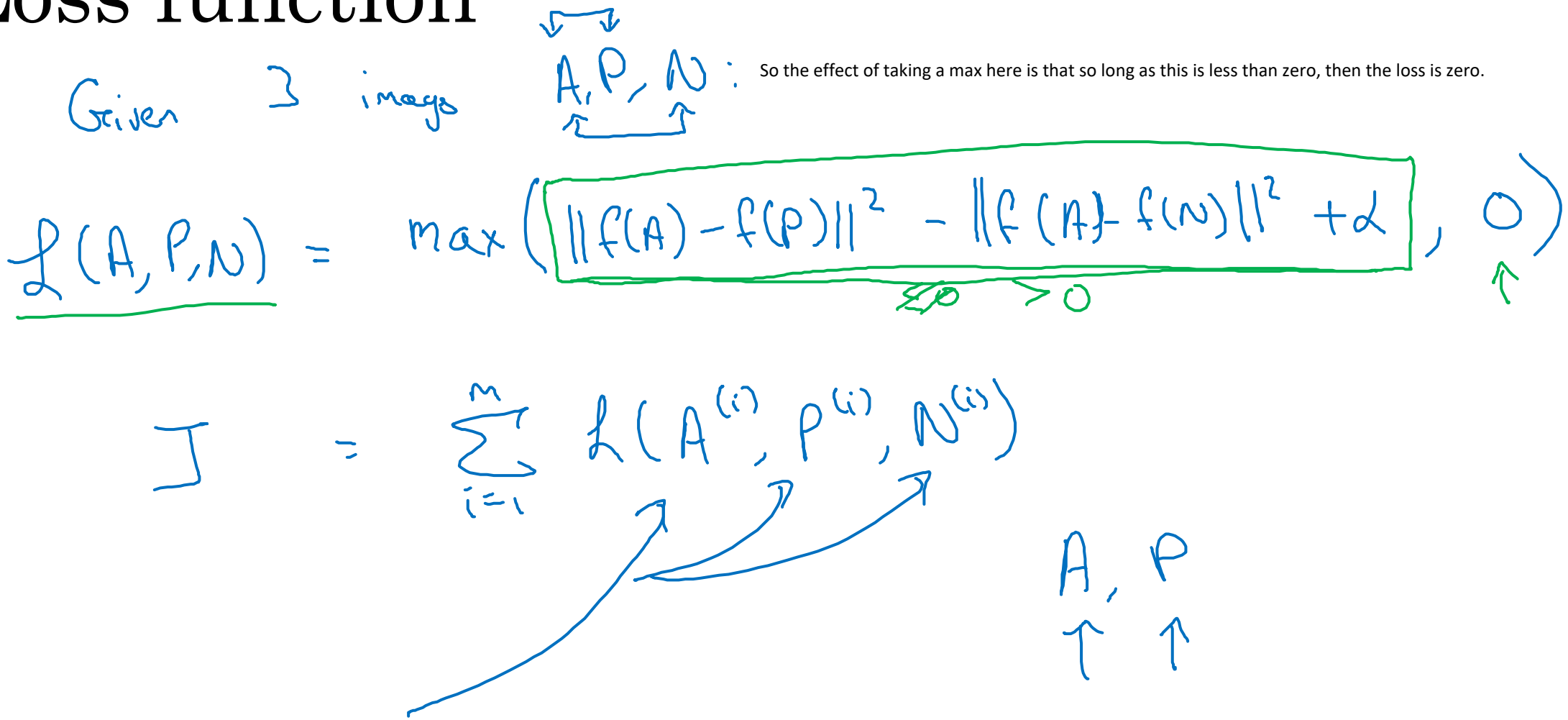
$$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha \leq 0$$

α = margin

$f(\text{img}) = \vec{0}$

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

Loss function



So the effect of taking a max here is that so long as this is less than zero, then the loss is zero.

Training set: 10k pictures of 1k persons

If you had just one picture of each person, then you can't actually train this system but of course after having trained the system, you can then apply it to your one shot learning problem where for your face recognition system maybe you have only a single picture of someone you might be trying to recognize. but for your training set you do need to make sure you have multiple images of the same person. At least some person for your training set so that you can have pairs of anchor and positive images.

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

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.

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

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

So in that case the learning algorithm has to try extra hard to take this thing on the right and try to push it up and take this thing on the left and try to push it down too. So at least there is at least a margin of alpha between the left side and right side.

And the effect of choosing these triplets is that it increase the computational efficiency of your learning algorithm.

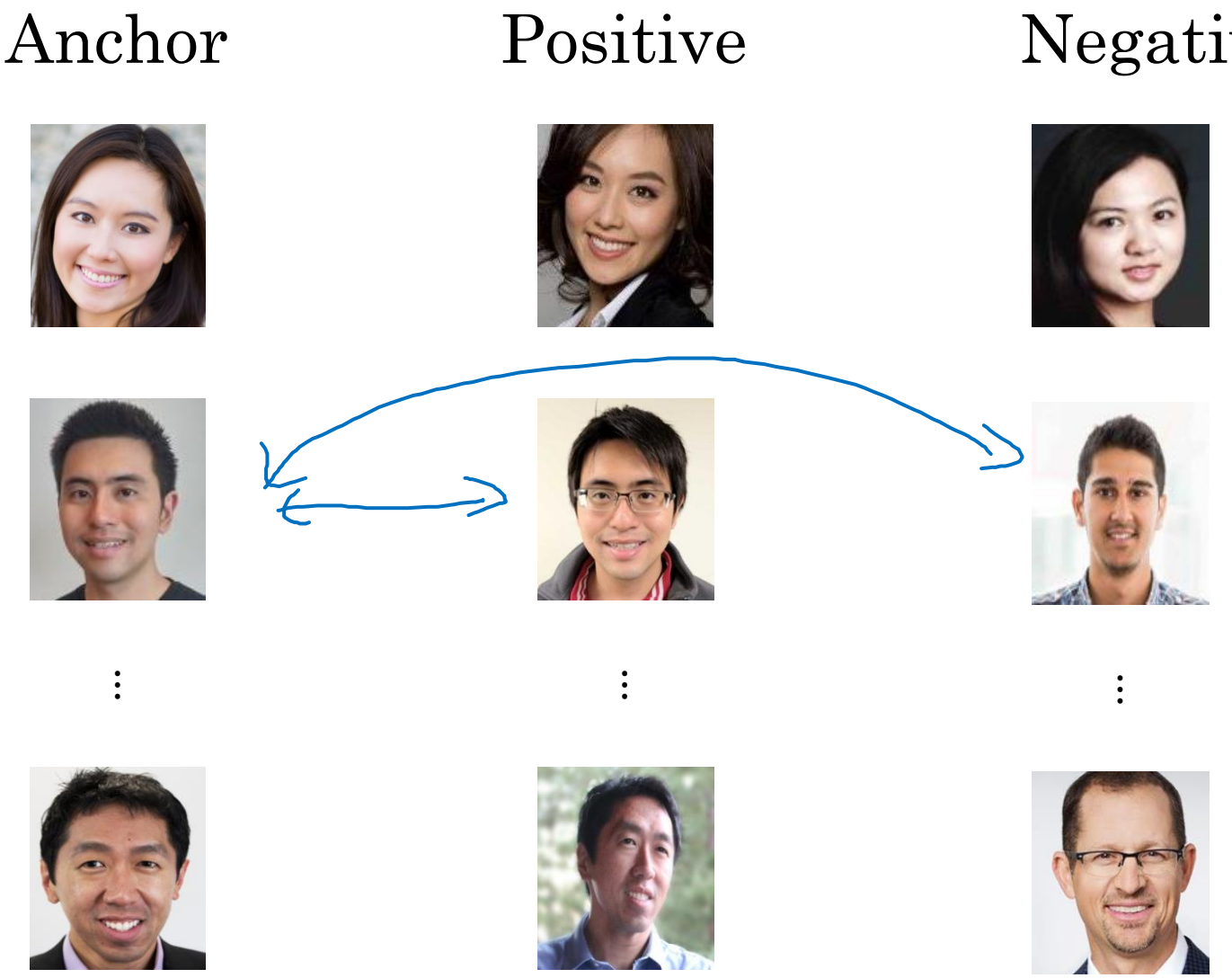
Face Net

Deep Face

It's only by choosing hard triplets that the gradient descent procedure has to do some work to try to push these quantities further away from those quantities.

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

Training set using triplet loss



J

$d(x^{(i)}, x^{(j)})$