



deeplearning.ai

# Face recognition

---

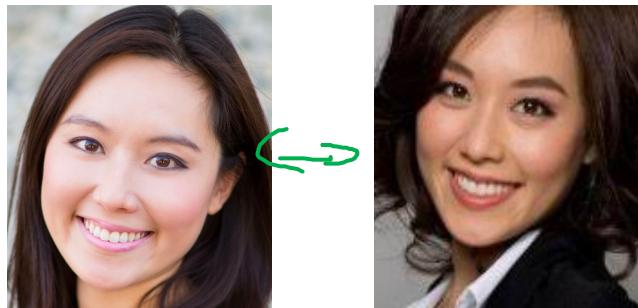
## Triplet loss

NN가 얼굴 사진에 대해 good encoding을 만들도록 NN의 파라미터를 학습하는 한가지 방법은 triplet loss function에 gradient descent를 적용하는 것이다.

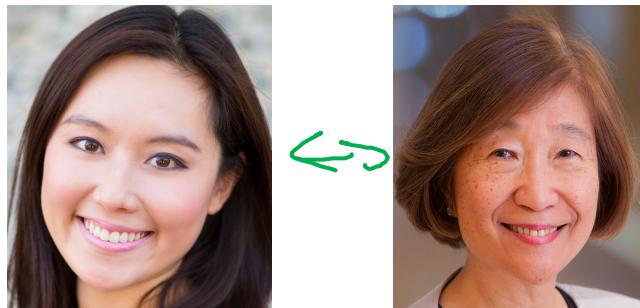
triplet loss function을 이용해 NN의 파라미터를 학습하기 위해서는 여러 이미지를 동시에 봐야한다.

# Learning Objective

예를 들어 this pair of images가 주어졌을 때 encoding이 유사하기를 원한다.  
왜냐하면 동일인이니까.



반면 얘들의 encoding은 다르기를 원한다.



triplet loss라는 이름이 붙은 이유는 항상 앵커 이미지 하나에 대해서 positive image는 거리가 가깝게, negative image는 거리를 멀게 만들기를 원하기 때문이다.  
(항상 동시에 이미지 3장을 처리한다)

$$\begin{array}{c} \text{Anchor} \\ \hline A \end{array} \quad \begin{array}{c} \text{Positive} \\ \hline P \end{array}$$

$d(A, P) = 0.5$

고로 이 margin(gap)은  $d(A, P)$ 는 push down 해주고  $d(A, N)$ 은 push up 해줘서 그 둘 간을 훨씬 벌려주는 효과가 되는 것

Want :  $\|f(A) - f(P)\|^2 + \lambda \leq 0$

what you want is...  
encoding이 다음과 같은 속성을 가지도록  
NN의 파라미터가 학습되기를 원한다.

$$\begin{array}{c} \text{Anchor} \\ \hline A \end{array} \quad \begin{array}{c} \text{Negative} \\ \hline N \end{array}$$

$d(A, N) = 0.7$

$$\|f(A) - f(N)\|^2 \leq d(A, N)$$

위 식을 다시 쓰면

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

Margin

예를 들어 이 경우, 0.51은 positive pair의 거리 0.5보다 크긴 하지만 이걸로 충분치는 않다고  $d(A, N)$ 이  $d(A, P)$  보다 좀 더 많이 크기를 원한다고 말해주는 효과가 있다. 적어도 0.7은 돼야지...

모든 이미지에 대해서 output을 0으로 만들도록  $f$ 가 학습이 되면 왼쪽 식을 항상 만족하게 되는데 이건 우리가 원하는게 아니다.

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

objective를 약간 수정하면 된다. 0보다 작거나 같기만 하면 되는게 아니라 0보다 약간 작은 수 보다 작아야 한다고 주장

Andrew Ng

triplet loss function을 formalize해보자

# Loss function

Given 3 images



$$L(A, P, N)$$

triplet of images로 정의되는 위 single example에 대한 loss는 이렇게 정의한다.

$$= \max \left( \left[ \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha \right], 0 \right)$$

우리가 원하는건 요 green box가 0 보다 작거나 같아지는 것이므로 loss function을 정의하기 위해 0과 함께 max를 취한다.

요게 0 보다 작으면 loss는 0이 되고 0 보다 크면 positive loss를 가질 것이다.

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

overall cost function for NN

sum over a training set

A,  
↑

P  
↑

training set을 위한 triplet을 구성하려면 same person에 대한 image pair가 필요함. 고로 동일인에 대한 여러 장의 이미지가 있는 데이터셋이 필요하다.

10k 이미지로 triplets를 구성한 후 위 cost function에 대한 gradient descent를 사용해 learning algorithm을 학습시킨다.

Training set: 10k pictures of 1k persons

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

10k pictures of 1k persons 상황이므로 인당 평균 10개 정도의 이미지를 갖는다. 만약 인당 한 장의 이미지만 있다면 train이 불가능할 것이다.

하지만 일단 시스템을 train하고 나면 one-shot learning problem에 적용할 수 있다.

Andrew Ng

# Choosing the triplets A,P,N



one of the problems is...

During training, if A,P,N are chosen randomly,

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

왜냐하면 랜덤하게 선택된 (A,N)은  
(A,P)랑은 많이 다를 것이기 때문이다.

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

this constraint

그러면 요 term이  $d(A, P)$ 에 비해 margin값 보다 훨씬 클 가능성이 높아지고  
결과적으로 NN은 이로부터 많이 학습하지 못하게 될 것이다.

그러면 어떻게 해야되나? 이렇게 해야지

Choose triplets that're “hard” to train on.

이 constraint를 만족하되  
 $d(A, P)$ 과  $d(A, N)$ 가 상당히  
가깝도록 triplet을 선택한다.

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

그러면 learning algorithm은  $d(A, N)$ 은 push up하고,  $d(A, P)$   
는 push down 하도록 추가적인 노력을 기울여야 한다.

Face Net  
Deep Face

computational efficiency를 높여주는 효과가 있다. 만약 랜덤으로  
선택하면 너무 많은 triplet들이 너무 쉬운 example일 것이고 그러면  
gradient descent가 거의 아무것도 하지 않게 된다.

# Training set using triplet loss

Anchor



Positive



Negative



:

:

:



이런 training set을 가지고 cost function  $J$ 를 최소화하도록 gradient descent를 사용한다.

그러면 적절하게(동일인의 두 이미지에 대한  $d$ 는 작아지고 타인의 두 이미지에 대한  $d$ 는 커지도록) encoding이 생성되도록 학습되는 방향으로 NN의 모든 파라미터를 back propagating하는 효과가 있을 것이다.

$J$

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