Special Application

: Face Recognition & Neural Style Transfer

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")

One-shot learning

학습 데이터 $S = \{(x_i, y_i) \mid i = 1, 2, ...\}$를 학습하고 새로운 데이터 x'가 주어졌을 때 x'에 대해 예측한 라벨 y'이 S에 있는 y_i 중 어떤 것과 같은지 찾는다.

BUT 머신러닝 모델은 트레이닝 샘플 하나로 테스트 확인하는 것을 어려워한다.

Learning from one example to recognize the person again.

새로운 데이터가 들어왔다고 다시 학습시키는 것이 아니라 similarity function을 사용한다.

"similarity" function

$d(img1, img2)$ = degree of difference between images

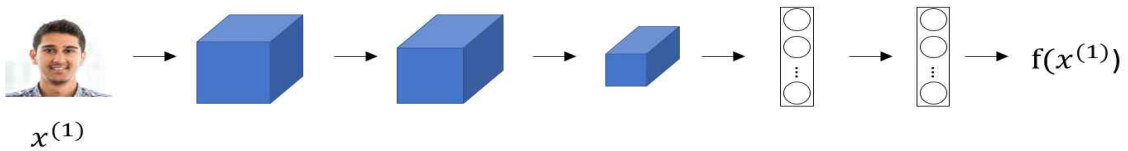
If $d(img1, img2) \leq \tau$: "same"

If $d(img1, img2) \leq \tau$: "different"

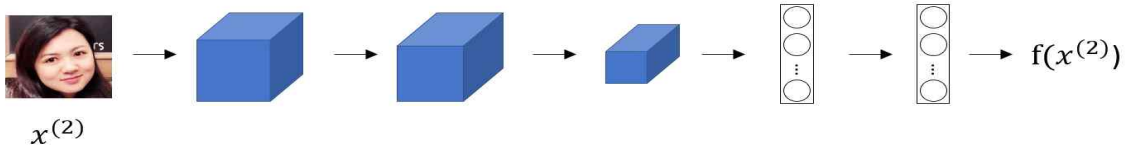
} verfication

Siamese network

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



$x^{(1)}$



$x^{(2)}$

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

Goal of learning

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

Learn parameters so that:

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

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

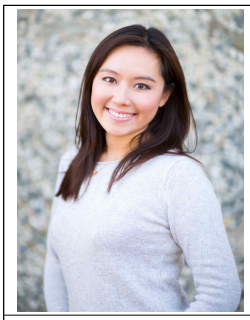
Triplet loss

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

얼굴 이미지에 대한 좋은 인코딩을 얻기 위해서 parameters을 훈련시키는 방법 중 하나는 triplet loss function에 gradient descent을 정의하고 적용하는 것이다.

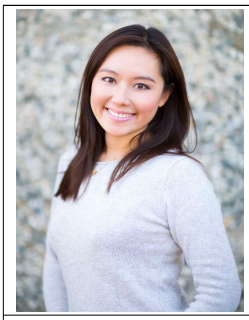
triplet의 의미: 3가지 이미지(anchor/ positive/ negative image)를 동시에 본다.

Learning Objective

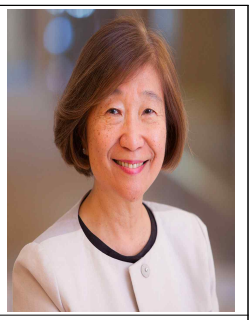


Anchor

Positive



Anchor



Negative

$\Rightarrow d(A, P) - d(A, N) + \alpha \leq 0$

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

α(margin)을 넣는 이유: 모든 이미지에 대한 f가 항상 영벡터와 같다면 자명하게 식을 만족한다. 하지만 신경망이 모든 인코딩에 대해 0을 반환하지는 않으므로 모든 인코딩이 서로 같지는 않다. 그리고 신경망이 자명한 값을 반환하는 다른 방법은 모든 이미지에 대한 인코딩이 다른 이미지와 같은 것이다. 이 경우에는 $N-N$이 된다.

자명하게 되는 것을 방지하기 위해 목적을 수정해서 0 보다 작거나 같은 것이 아니라 0 보다 충분히 작은 값을 되도록 해야 한다. 구체적으로 $-\alpha$보다 작아야 한다. 이것으로 자명해를 갖지 못하도록 한다. 관례상 우변에 $-\alpha$로 적는 대신 좌변에 $+\alpha$를 쓴다.

Loss function

Given 3 images A, P, N:

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

Cost

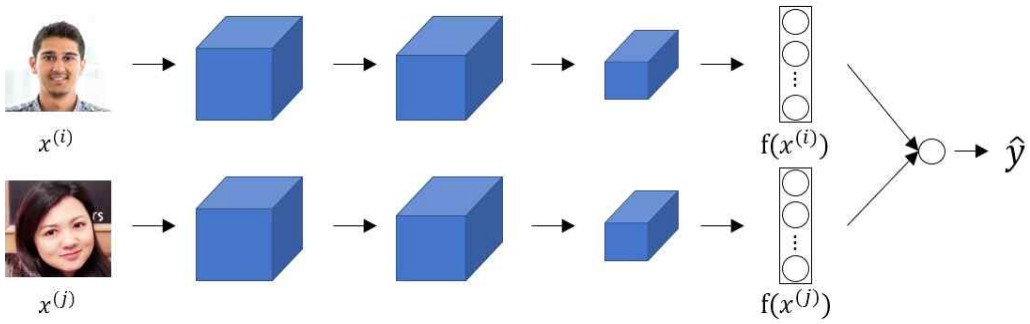
$$J=\sum_{i=1}^{m}L(A^{(i)},P^{(i)},N^{(i)})$$

※ Choosing the triplets A, P, N

During training if A, P, N are chosen radomly, $d(A,P)-d(A,N)+\alpha\leq 0$ is easily satisfied.

∴ Chose triplets that're "hard" to train on. $d(A,P)\approx d(A,N)$

Face verification and binary classification

New	
database	

여기서 database의 $f(x^{(j)})$는 precompute 되어있다.

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

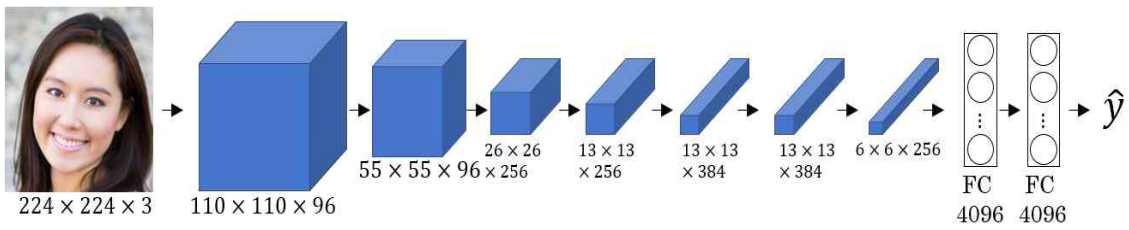
$$\chi^2=\left|f(x^{(i)})_k-f(x^{(j)})_k\right|=\frac{\left(f(x^{(i)})_k-f(x^{(j)})_k\right)^2}{f(x^{(i)})_k+f(x^{(j)})_k}$$: 카이 제곱 유사도

Neural Style Transfer

	+		=	
Content(C)		Style(S)		Generated image(G)

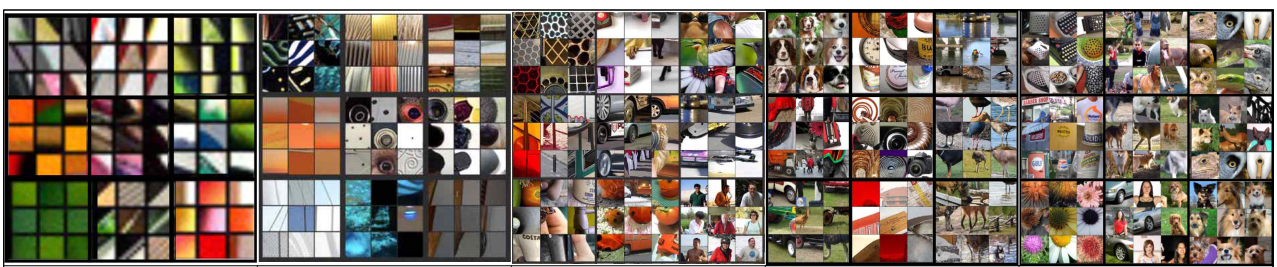
Visualizing what a deep network is learning

[Zeiler and Fergus., 2013, Visualizing and understanding convolutional networks]



Pick a unit in layer 1.
Find the nine image patches that maximize the unit's activation.
Repeat for other units

신경망 훈련을 일시 중지하고 특정 유닛의 activation를 최대화하는 이미지가 무엇인지 파악한다. layer 1의 hidden units은 신경망의 상대적으로 작은 부분만 본다.
시각화하면 unit's activation을 나타내는 데 작은 이미지 patches만 나타난다. 특정 유닛이 모든 이미지를 보기 때문이다.
9개의 서로 다른 대표 뉴런이고 9개의 이미지 패치 각각에 대해 최대로 activation 시킨다. 그래서 layer 1의 hidden을 training 시킨다. 모서리나 특정 명암 색과 같은 상대적으로 단순한 feature를 찾는다.
더 깊은 layer의 hidden unit으로 시각화하면 이미지의 더 큰 영역을 볼 것이다. 그러면 각 픽셀은 신경망의 나중 layer의 output에 영향을 줄 것이다.



Layer 1	Layer 2	Layer 3	Layer 4	Layer 5

Cost function

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

두 hyperparameters α와 β를 사용하여 $J_{Content}$와 J_{Style} 간 상대적 가중치를 구한다.

두 hyperparameters를 사용하여 가중치의 상대적 비용을 구체화하는 것은 사실 불필요하다. hyperparameter 하나로 충분하다. 하지만 원저자는 두 hyperparameters를 사용해서 그 관례를 지킨다.

Find the generated image G

1. Initiate G randomly
2. Use gradient descent to minimize J(G)

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

1. Content cost function

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

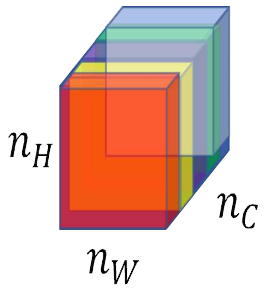
- Say you use hidden layer l to compute content cost.
- Use pre-trained ConvNet.
- Let $a^{[l](C)}$ and $a^{[l](G)}$ be the activation of layer l on the images.
- If $a^{[l](C)}$ and $a^{[l](G)}$ are similar, both images have similar content.

l이 매우 작은, 예로 들어 1이라면 G와 C가 매우 비슷한 픽셀 값을 가진다. 반대로 l이 매우 deep하면 결과는 마치 'C에 개가 있으면 G 어딘가에 개가 있는지 확인해야 해' 와 같다. 그래서 l은 너무 얕지도 깊지도 않는 layer에서 선택해야 한다.

So, just be clear on using this notation as if both of these have been unrolled into vectors, so then, this becomes the square root of the L_2 norm between $a^{[l](C)}$ and $a^{[l](G)}$, after you've unrolled them both into vectors. There's really just the element-wise sum of squared differences between these two activation. But it's really just the element-wise sum of squares of differences between the activations in layer l, between the images in C and G.

2. Style cost function

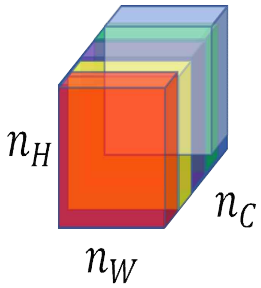
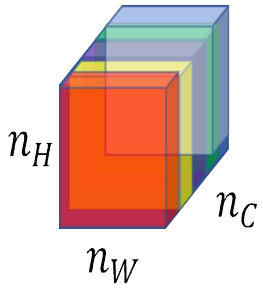
Meaning of the "style" of an image



Say you are using layer l's activation to measure "style".
Define style as correlation
between activations across channels.

⇒ How correlated are the activations
across different channels?

Intuition about style of an image

Style image	Generated Image	Correlated?
		

correlation(상관 관계): 높은 레벨의 texture 구성 요소들이 이미지 부분에서 함께 발생할 수도 있고 아닐 수도 있다는 뜻

높은 레벨의 feature들(수직 texture나 주황빛 혹은 다른 물체들)이 이런 한 이미지상의 다른 부분에 있을 때 얼마나 자주 같이 발생하고 얼마나 자주 같이 발생하지 않는지를 측정하는 것이다.

Style image에 대한 채널 간 correlation 정도와 Generated image에 대한 채널 간 correlation 정도를 측정하여 비교한다.

Style Matrix

Let $a^{[l]}_{i,j,k}$ = activation at (i,j,k), $i \in H$, $j \in W$, $k \in C$

\# $G^{[l]}$ is $n_C^{[l]} \times n_C^{[l]}$ and $G^{[l]}_{kk'}$, $k = 1, 2, \cdots, n_C^{[l]}$

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

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

$$J_{Style}^{[l]}(S, G) = \frac{1}{\left(2 n_H^{[l]} n_W^{[l]} n_C^{[l]}\right)^2} \parallel G^{[l](S)} - G^{[l](G)} \parallel_F^2$$
$$= \frac{1}{\left(2 n_H^{[l]} n_W^{[l]} n_C^{[l]}\right)^2} \sum_k \sum_{k'} \left(G_{kk'}^{[l](S)} - G_{kk'}^{[l](G)}\right)^2$$

Style cost function

$$J_{Style}(S, G) = \sum_l \lambda^{[l]} J_{Style}^{[l]}(S, G)$$

여러 layers의 style cost를 사용하면 시각적으로 더 좋은 결과를 얻을 수 있다.
전체 style cost function은 모든 layers의 style cost의 합에 추가적인 hyper parameter를 곱한다.

신경망에서 다른 layer를 사용하는 것이다. 얕은 layer에서 모서리와 같은 비교적 간단한 낮은 레벨의 features를 측정한다. 깊은 layer에서는 높은 수준의 features를 측정한다. 스타일 계산할 때 신경망이 낮은 레벨과 높은 레벨의 correlation을 모두 고려한다.

1D and 3D generalizations of models

1D: sound, sentence, ...

3D: CT, MRI, ...