Face verification vs. face recognition

→ Verification
- Input image, name/ID
- Output whether the input image is that of the claimed person

1:1 99% 99.9

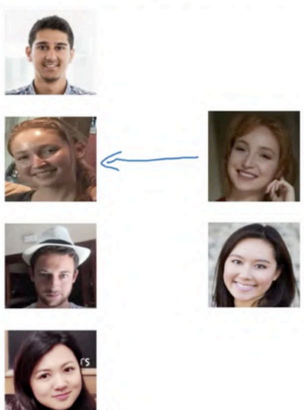
Harder

→ 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")

1:K

K=100 ←

One-shot learning



Learning from one example to recognize the person again

→ CNN → o

softmax

(6)

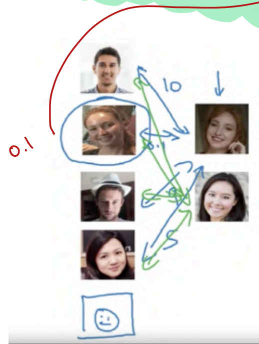
Learning a "similarity" function

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

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

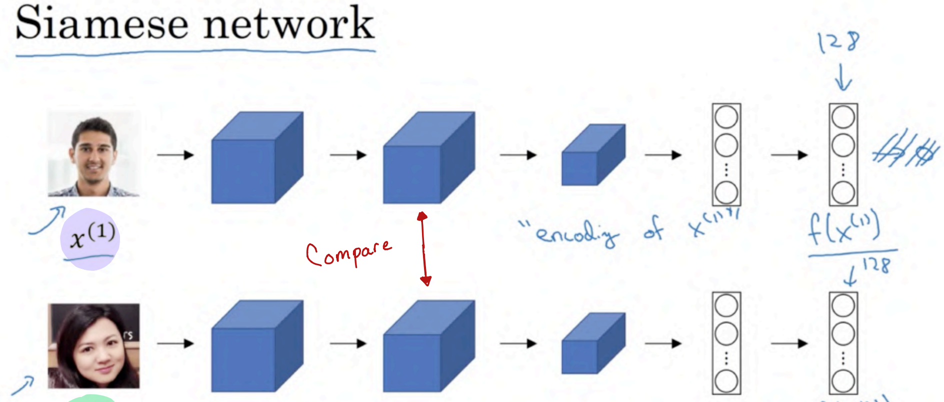
$> \tau$ "different"

} Verification.



d(img1, img2)

Siamese network

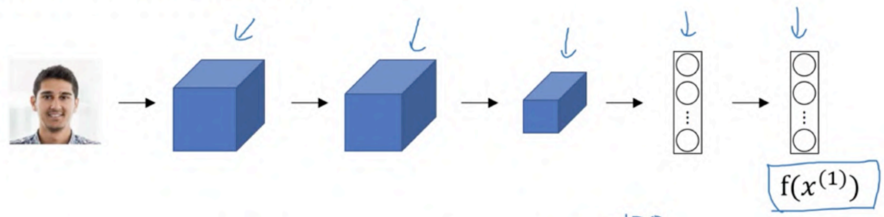


$$\rightarrow d(x^{(1)}, x^{(2)}) = \| f(x^{(1)}) - f(x^{(2)}) \|_2^2$$

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

Andrew Ng

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

Learning Objective

Triplet Loss

$d(A,P) = 0.5$ want small

Gap we want between 0.2

$d(A,N) = \cancel{0.51}$ 0.7

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

$$\boxed{\underbrace{\| f(A) - f(P) \|^2}_{0} - \underbrace{\| f(A) - f(N) \|^2}_{0} + \underbrace{\alpha}_{\text{margin}} \leq 0}$$

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

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

Andrew Ng

Loss function

distance of positive - distance of negative + α margin

Given 3 images A, P, N:

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

$\leq 0 \quad > 0$

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

A, P

Training set: 10k pictures of 1k persons

[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.

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

Won't learn much!

Train on "hard"

Choose triplets that're "hard" to train on.

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

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

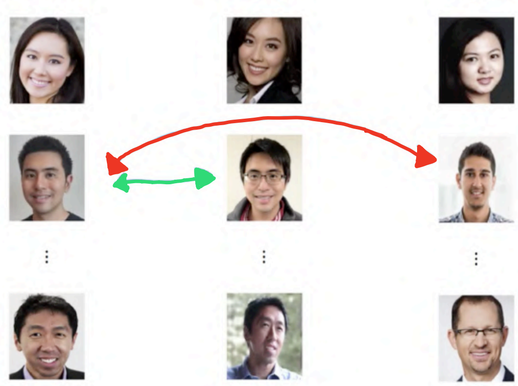
Face Net

Deep Face

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

Training set using triplet loss

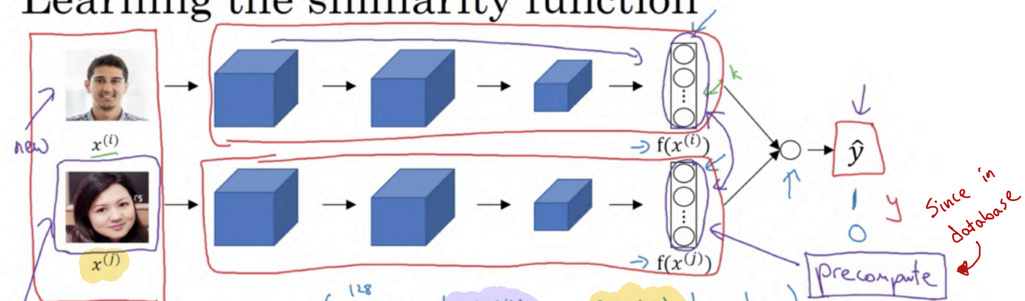
Anchor Positive Negative



J

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

Learning the similarity function



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

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

$$\frac{\left(f(x^{(i)})_k - f(x^{(j)})_k\right)^2}{f(x^{(i)})_k + f(x^{(j)})_k} \qquad \chi^2$$

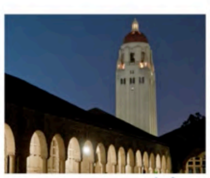
[Taigman et. al., 2014. DeepFace closing the gap to human level performance] Andrew Ng

Face verification supervised learning

x	y	
	1	"Same"
	0	"different"
	0	
	1	

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

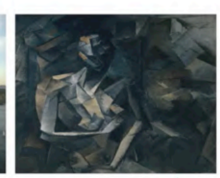
Neural style transfer


Content (C)


Style (S)


Generated image (G)

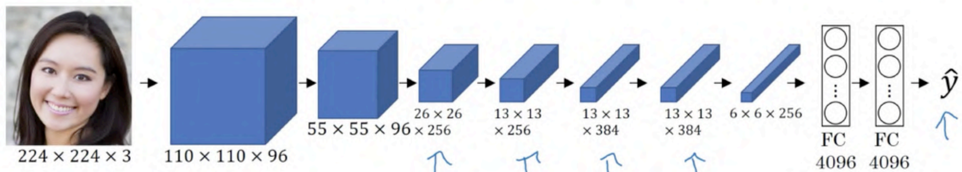

Content (C)


Style (S)


Generated image (G)

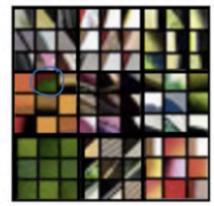
[Images generated by Justin Johnson] Andrew Ng

Visualizing what a deep network is learning



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

Repeat for other units.



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

Andrew Ng

Visualizing deep layers: Layer 1

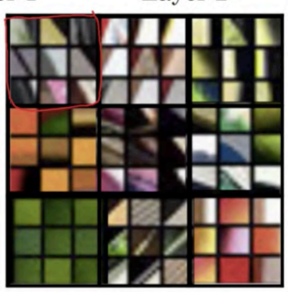
Layer 1

Layer 2

Layer 3

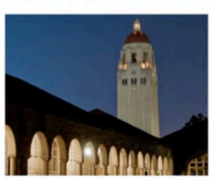
Layer 4

Layer 5



Andrew Ng

Neural style transfer cost function



Content C

Style S

Generated image G

Measures how similar the style of image C is to image G

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

[Gatys et al., 2015. A neural algorithm of artistic style. Images on slide generated by Justin Johnson]

Andrew Ng

Find the generated image G

1. Initiate G randomly

   G: $100 \times 100 \times 3$

   RGB

2. Use gradient descent to minimize $J(G)$

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

[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng

Content cost function

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

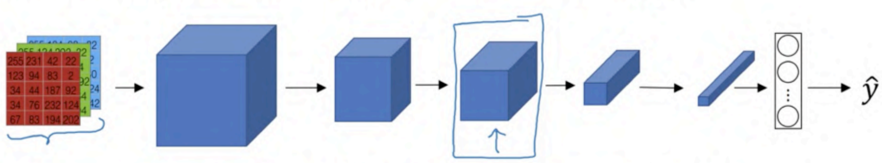
- Say you use hidden layer l to compute content cost.
- Use pre-trained ConvNet. (E.g., VGG network)
- Let $a^{[l](C)}$ and $a^{[l](G)}$ be the activation of layer l on the images
- If $a^{[l](C)}$ and $a^{[l](G)}$ are similar, both images have similar content

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

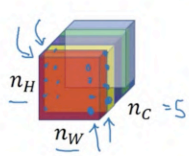
[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng

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.



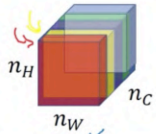
How correlated are the activations across different channels?

[Gatys et al., 2015. A neural algorithm of artistic style]

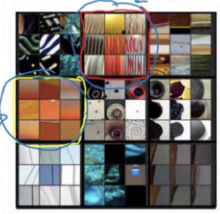
Andrew Ng

Intuition about style of an image

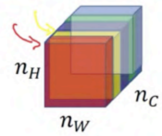
Style image



Correlated?
Uncorrelated



Generated Image



[Gatys et al., 2015. A neural algorithm of artistic style] Andrew Ng

Style matrix

Let $a^{[l]}_{i,j,k}$ = activation at (i, j, k). $G^{[l]}$ is $n_c^{[l]} \times n_c^{[l]}$

H W C

n_c

$G^{[l]}_{kk'}$

$k = 1, \ldots, n_c^{[l]}$

Channels in layer

Measure correlation

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

Unnormalized cross of the errors

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

"Gram matrix"

difference

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

$$= \frac{1}{(2 n_H^{[l]} n_W^{[l]} n_C^{[l]})^2} \sum_k \sum_{k'} \left(G^{[l](S)}_{kk'} - G^{[l](G)}_{kk'} \right)^2$$

[Gatys et al., 2015. A neural algorithm of artistic style] Andrew Ng

Style cost function

$$\left\| G^{[l](S)} - G^{[l](G)} \right\|_F^2$$

$$J^{[l]}_{style}(S, G) = \frac{1}{\left(2 n_H^{[l]} n_W^{[l]} n_C^{[l]}\right)^2} \sum_k \sum_{k'} (G^{[l](S)}_{kk'} - G^{[l](G)}_{kk'})$$

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

So the formula should be:

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

Also at 11:08 the style cost function formula should be the squared difference:

$$(G_S - G_G)^2$$

instead of just the difference:

$$(G_S - G_G).$$

The style cost function should be:

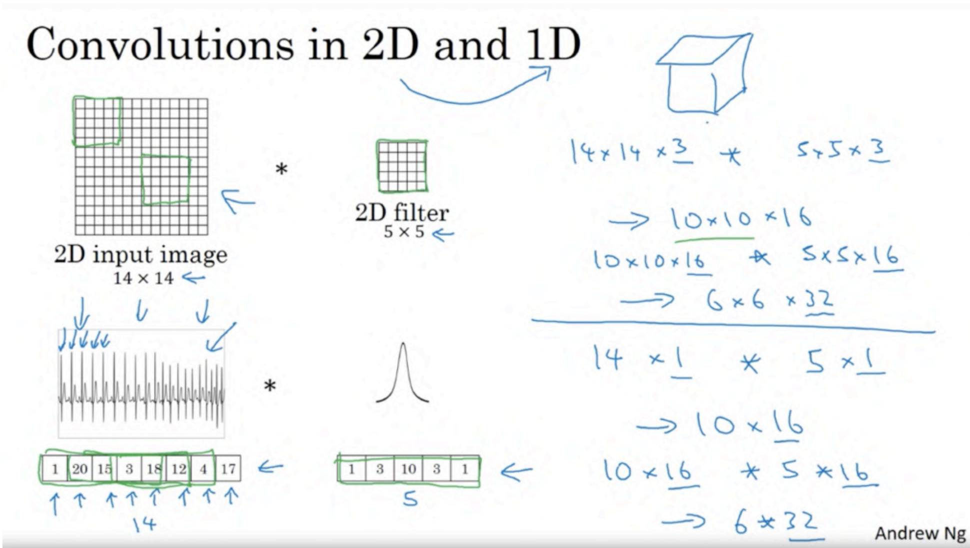
$$J^{[l]}_{style}(S, G) = \frac{1}{(2 n_H^{[l]} n_W^{[l]} n_C^{[l]})^2} \sum_k \sum_{k'} \left(G^{[l](S)}_{kk'} - G^{[l](G)}_{kk'} \right)^2$$

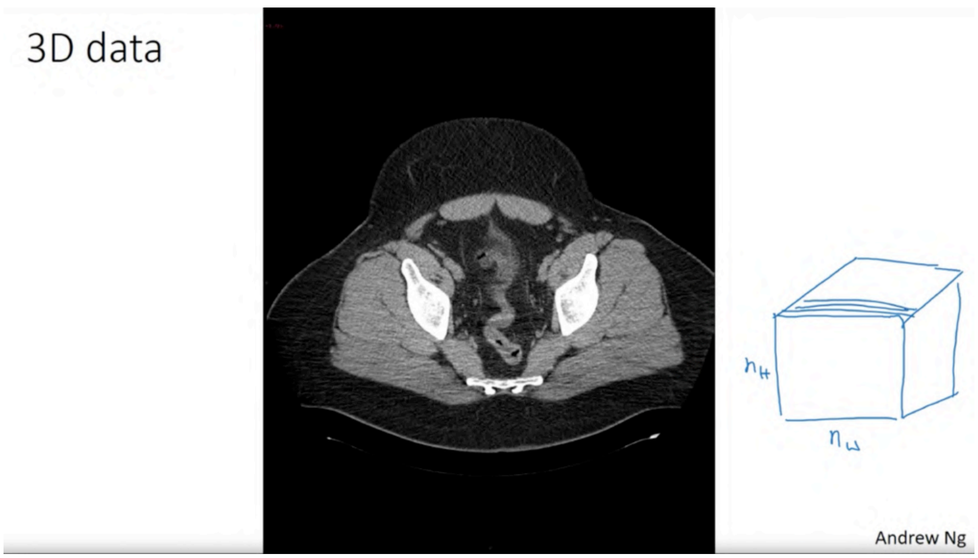
$$J_{style}(S, G) = \sum_l \lambda^{[l]} J^{[l]}_{style}(S, G)$$

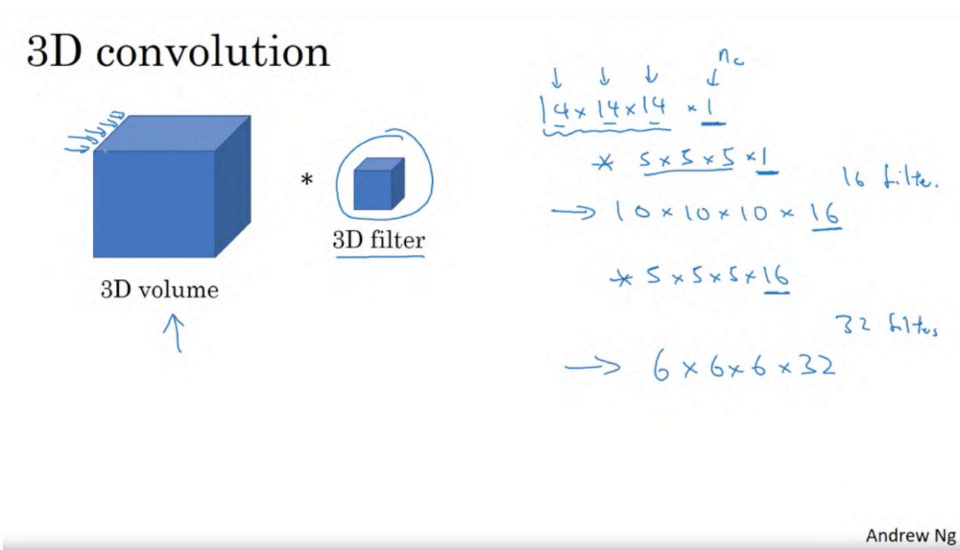
$$\underline{J(G)} = \alpha J_{content}(C, G) + \beta J_{style}(S, G)$$

G

[Gatys et al., 2015. A neural algorithm of artistic style] Andrew Ng


Convolutions in 2D and 1D
2D input image
14 × 14
*
2D filter
5 × 5
14×14×3 * 5×5×3
→ 10×10×16
10×10×16 * 5×5×16
→ 6×6×32
14×1 * 5×1
→ 10×16
10×16 * 5×16
→ 6×32
1 20 15 3 18 12 4 17
14
1 3 10 3 1
5
Andrew Ng


3D data
n_H
n_W
Andrew Ng


3D convolution
3D volume
*
3D filter
n_c
14×14×14×1
* 5×5×5×1
16 filte.
→ 10×10×10×16
* 5×5×5×16
32 filters
→ 6×6×6×32
Andrew Ng