BLG 561 E FALL 2021

Deep Learning

16.11.2021

Gözde ÜNAL

Designing Cost/Losses for Style Transfer:

$$J_{overall}(G) = ?$$

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

picture you took

Content C

Style S eg. Van Gogh

G

G : Noise

$$\hat{G} = \arg\min_{G} J_{overall}(G).$$

$\hat{G}$ your picture in the style of Van Gogh

1 Content Loss:

- pretrained ConvNet (VGG)

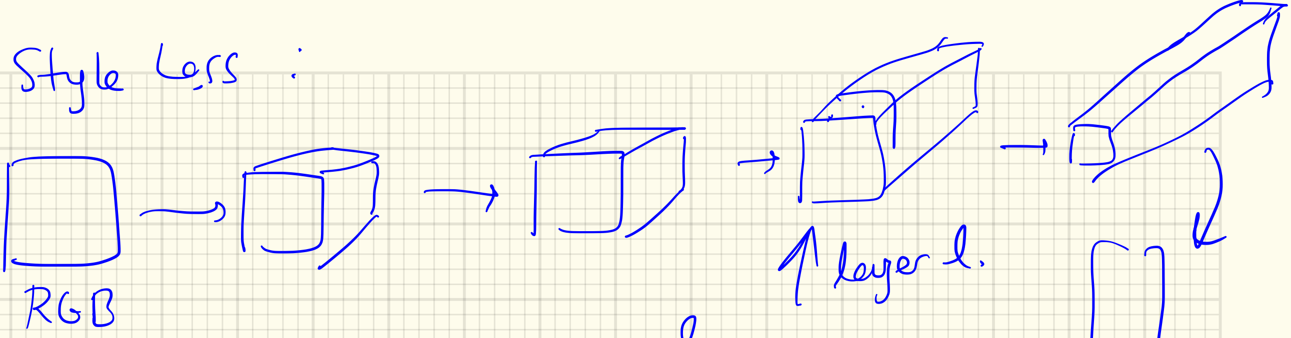
Let $a^{[l],C}$, $a^{[l],G}$: l: typically middle layers

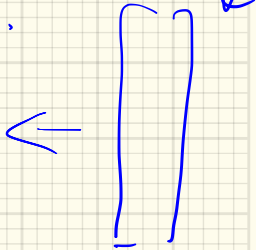
similar content

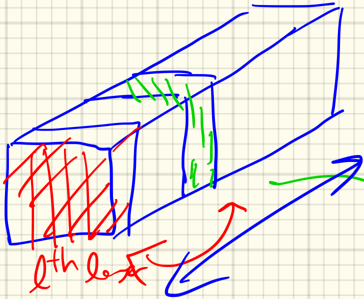
$$J_{content}(C, G) = \sum_{l = k_1, k_2 \dots} \left\| a^{[l],C} - a^{[l],G} \right\|_2^2$$

feature loss (perceptual loss)

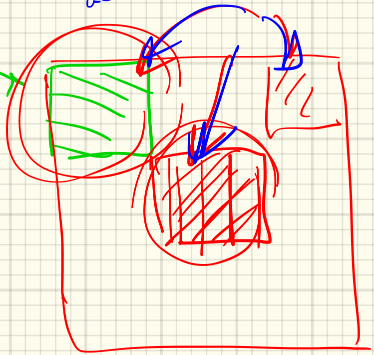
② Style Loss :







look at correlations of activations across channel.



Assumption: Certain set of activations coexist to produce a certain style

Style (Gram) matrix

$$G^{[l],s} = \begin{bmatrix} & & \\ & G^{[l]}_{kk'} & \\ & & \end{bmatrix}$$

$$G^{[l]}_{kk'} = \sum_{i}\sum_{j} \underbrace{a^{[l],s}_{i,j,k} \, a^{[l],s}_{i,j,k'}}$$

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

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

hyperparameter set $l = 1, \dots, \#layers$.

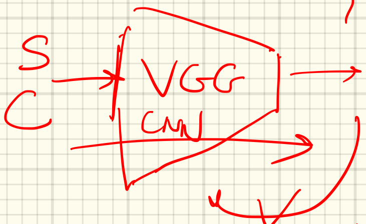
Overall Loss in Style Transfer:

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

$\min J$

G

$\{a^{l,S}\}_{l=1}^{\#layers}, \{a^{l,C}\}$



X no training

One-Shot Learning / Metric Learning

Use CNNs / NNs,

Input Image 1 → NN (Siamese)
Input Image 2 →

→ Learn a "similarity"
how similar are the two input images.

Say you have a Face Recognition System, say you have 10 employees:

→ Classical approach:

Input images of a person

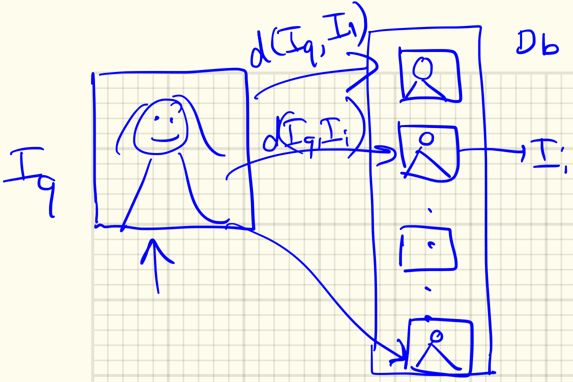
Stack CNN ConvNet + FC Layers + softmax → p_1, p_2, ⋮, p_{10}

Cons
1) To train such a system, you require many different images of each of the 10 persons
2) When a new employee joins, you have to retrain!

Instead → One-Shot Classification

NN learns:
$$\begin{cases} d(I_1, I_2) < \tau & \text{for the "same" person} \\ d(I_1, I_2) > \tau & \text{for different persons} \end{cases}$$

Similarity fn.

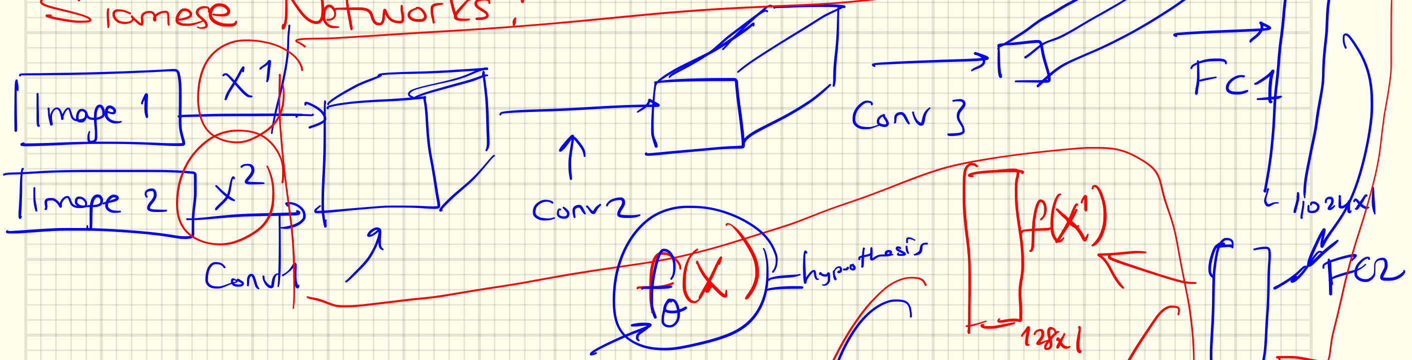


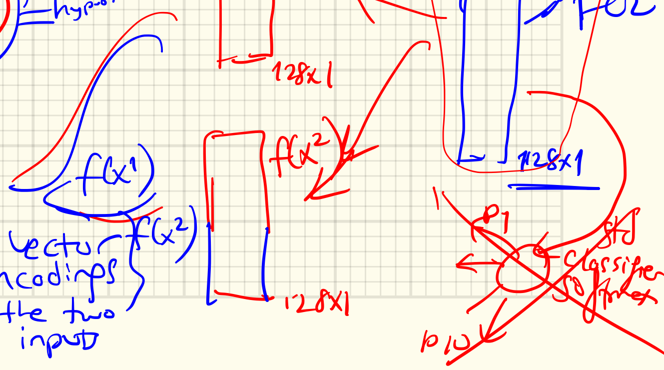
For the matching picture, we want

$$d(I_q, I_j) \approx \text{small number.}$$

goal.

Siamese Networks:





$$d(x_1, x_2) = \left\| f_\theta(x_1) - f_\theta(x_2) \right\|_2^2$$

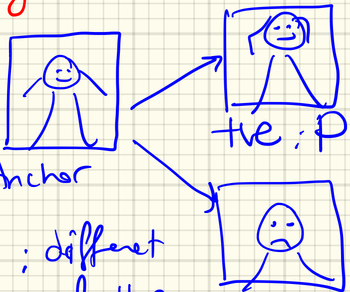
Goal: Learn parameters θ s.t.

If $x^{(i)}$ & $x^{(j)}$ are of the "same" person $\quad \|f_\theta(x^i) - f_\theta(x^j)\|_2^2$ "small"

" " " " "different" " $\quad \|f_\theta(x^i) - f_\theta(x^j)\|_2^2$ "large"

What is the loss?

eg. Triplet Loss (FaceNet, 2015, Shroff)



A: Anchor

+ve: P

-ve N

A, P: different pictures of the same person

Triplet loss minimizes:

$$\min \mathcal{L}(A, P, N)$$

$$\mathcal{L} = \max\Big(0, \underbrace{\|f_\theta(A) - f_\theta(P)\|^2}_{\text{small}} - \|f_\theta(A) - f_\theta(N)\|^2 + \alpha\Big)$$

α: margin

hyperparameter

as long as this part is -ve loss $\to 0$

α: margin parameter to avoid learning trivial $f(x) = 0$

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

- Say you have 30K pictures, choose/sample your triplets;
You need multiple pictures of the same person in the training set to form (A, P) pairs.

→ Choosing triplets is tricky! when: $d(A, N) \gg d(A, P)$

$$\|f(A) - f(N)\|^2 \gg \|f(A) - f(P)\|^2$$

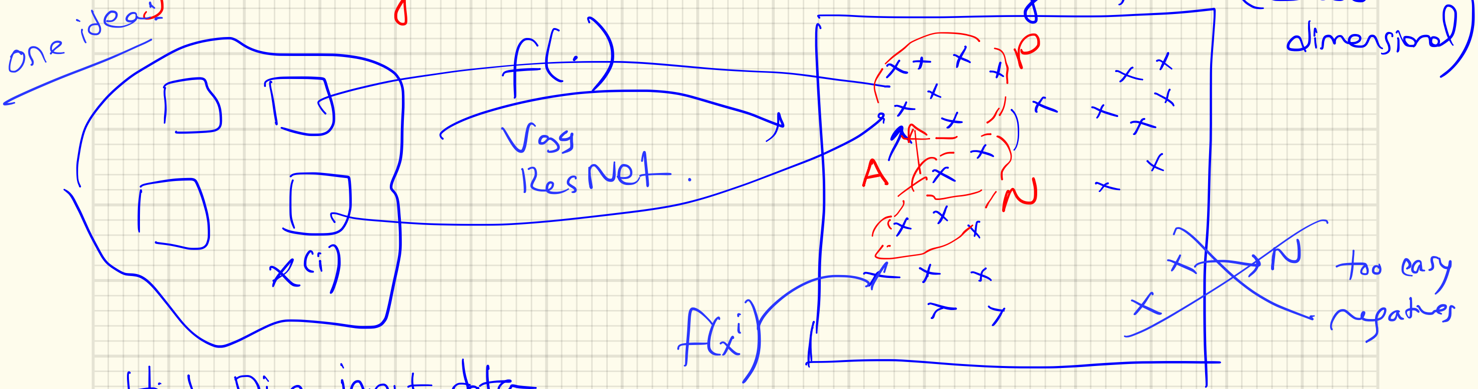
$$\underbrace{d(A, P) + \alpha}_{\leq} \leq d(A, N) \text{ easily satisfied.}$$

→ Choose triplets that are hard to train on!

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

$\Downarrow$

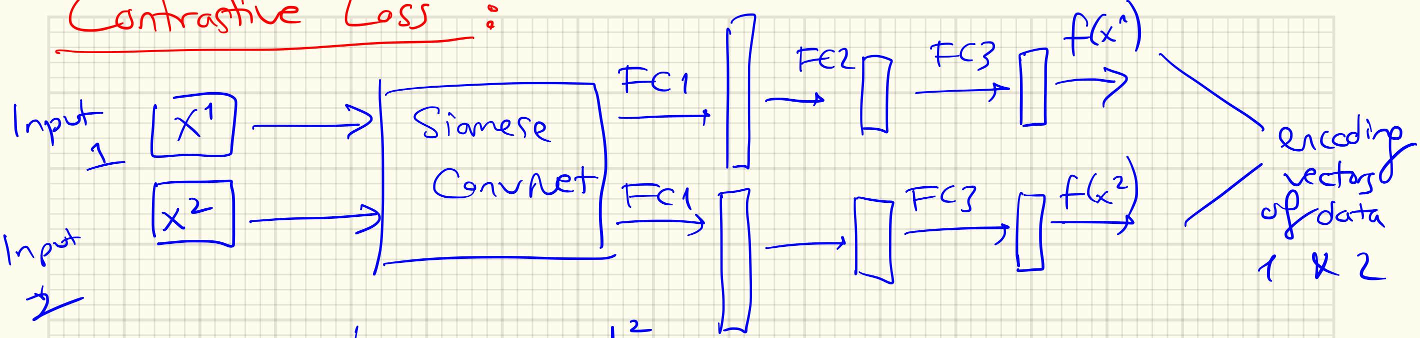
Hard Negative mining Problem!



High_Dim input data manifold

Cluster in a low-dim embedding space, then use L^2 norm to pick (A, P, N) triplets.
eg.

Contrastive Loss:



Contrastive Loss

$$\mathcal{L}(x^1, x^2) = y \cdot \overbrace{\|f(x^1) - f(x^2)\|_2^2}^{d^2} + (1-y)\cdot \max(0, \underbrace{\alpha}_{\text{margin}} - d)^2$$

Label:

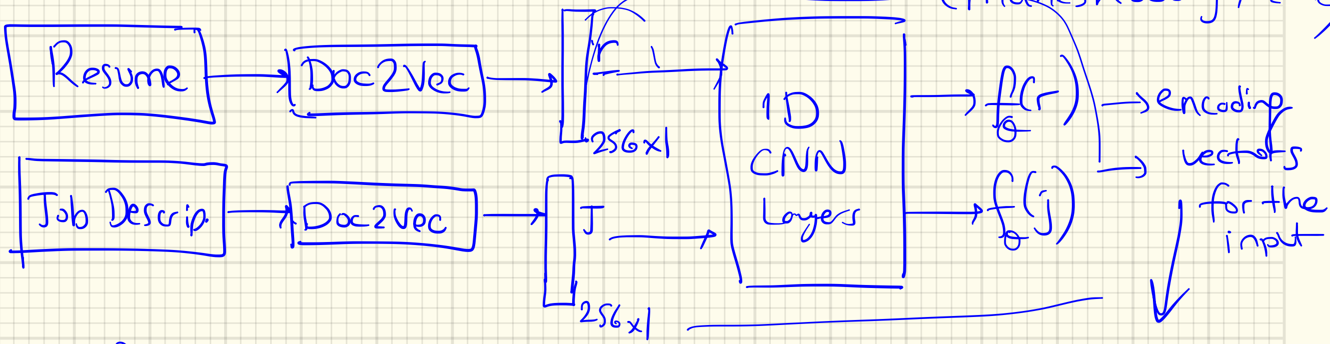
$y = 1$; if the two data are similar

$y = 0$; o/w

if $\underbrace{\|f(x^1) - f(x^2)\| > \alpha}$ $\longrightarrow$ 0 loss.

∴ If x^1 & x^2 are dissimilar, then their distance should be at least α, o/w a +ve loss is incurred.

Ex: Matching Resumes to Jobs via Deep Siamese Networks (Maheshwary, 2018)



(r, j) pairs : 1 match $y = 1$

0 no match $y = 0$

$$d = \| f_\theta(r) - f_\theta(j) \|_2$$

use contrastive loss

→ Many hyperparameters to tune !!