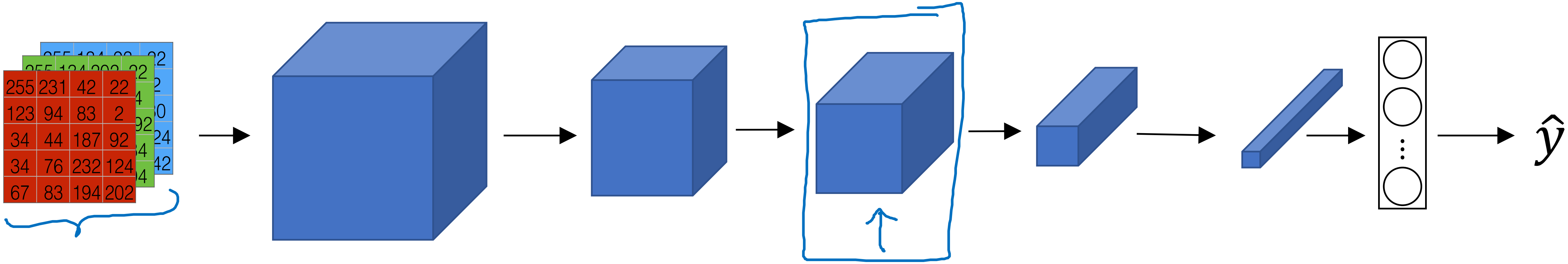
deeplearning.ai

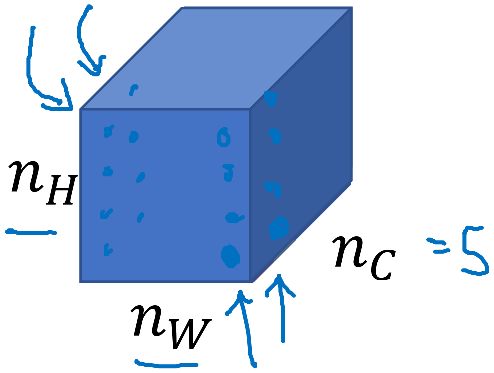
Neural Style Transfer

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.

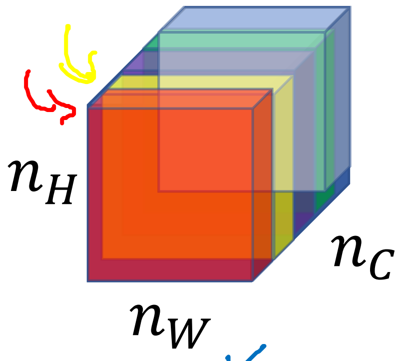


How correlated are the activations across different channels?

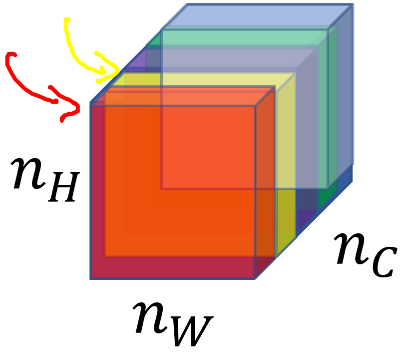
[Gatys et al., 2015. A neural algorithm of artistic style]

Intuition about style of an image

Style image

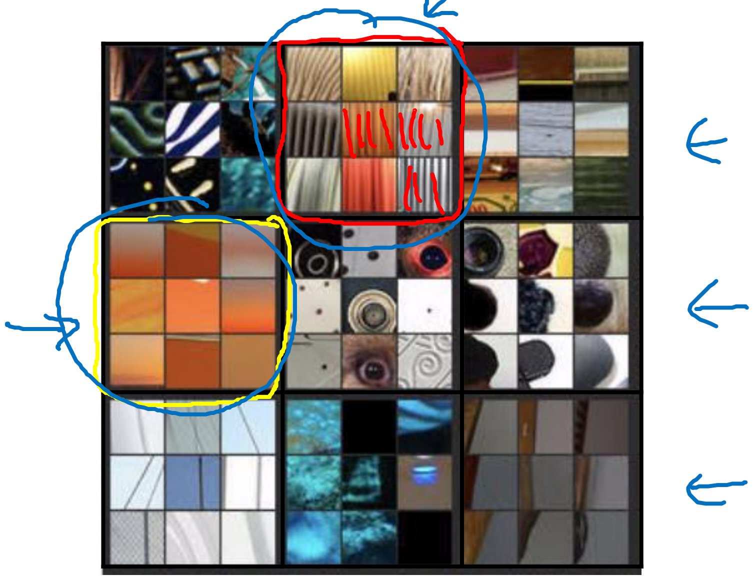


Generated Image



Correlated?

Uncorrelated



And so, the correlation tells you which of the high-level texture components tend to occur or not occur together in part of an image. And the degree of correlation that gives you one way of measuring how often these different high level features such as vertical texture or this orange tint or another thing as well, how often they occur together and don't occur together in different parts of an image.

And so, if we use the degree of correlation between channels as a measure of the style, then what you can do is measure the degree to which in your generated image, this first channel is correlated or uncorrelated with the second channel.
And that will tell you in the generated image, how often this type of vertical texture occur or doesn't occur with this orangeish tint and this gives a measure of how similar is the style of the generate image to the style of the input style image.

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

Style matrix

So what you're going to do is given an image, computed something called a style matrix, which will measure all those correlations we talk about on the last slide.

H W C

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

n_c

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

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

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

Technically, this is the unnormalized cross-covariance, because we're not subtracting the mean, and we just multiplying out these things

"Gram matrix"

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

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

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

β

But the normalization constant doesn't add to that much, because this cost is multiplied by some hyperparameter b anyway.

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

Andrew Ng

Style cost function

decisional normalization constant, which isn't that important

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

basically a Frobenius norm

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

It turns out that you get more visually pleasing result if you use the style cost function from multiple different layers.

So the overall cost function:

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

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

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

Andrew Ng