Neural Style Transfer

Cost function

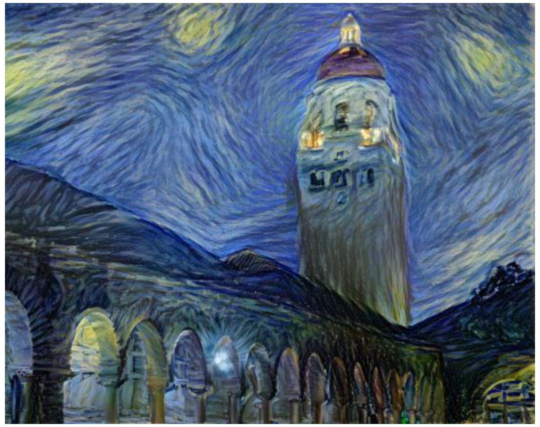
deeplearning.ai

To build a neural style transfer system, let's define a cost function for the generated image. What you see later is that it'll be by minimizing this cost function that you can generate the image that your want.

Neural style transfer cost function

Content C

Style S



Generated image G

$$J(G) = \alpha J_{content}(C, G)$$

$$+ \beta J_{style}(S, G)$$

Finally we weight these with two hyperparameters alpha and beta to specify the relative weighting between the content cost and the style cost.
It seems redundant to use two hyperparameters to specify the relatively cost of the weighting.
One hyperparameter seems it could be enough.
But the original authors of the neural style transfer algorithm used two different hyperparameter, and so I'm just going to follow their convention here.

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

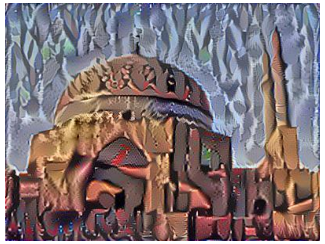
Find the generated image G

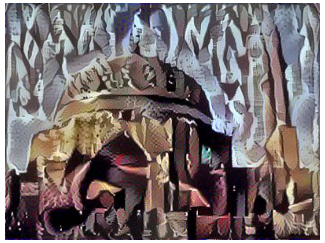
1. Initiate G randomly

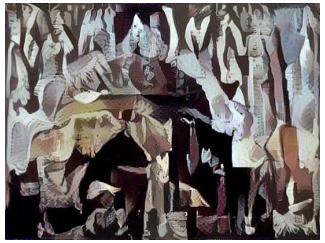
   $\underline{G}: \underline{100} \times \underline{100} \times \underset{\substack{\uparrow \\ \text{RGB}}}{3}$

2. Use gradient descent to minimize $\underline{J(G)}$

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







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