Style Cost Function

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

English

Help us translate!

0:00
In the last video, you saw how to define the content cost function for the neural style transfer. Next, let's take a look at the style cost function. So, what is the style of an image mean? Let's say you have an input image like this, they used to seeing a convnet like that, compute features that there's different layers. And let's say you've chosen some layer L, maybe that layer to define the measure of the style of an image. What we need to do is define the style as the correlation between activations across different channels in this layer L activation. So here's what I mean by that. Let's say you take that layer L activation. So this is going to be nh by nw by nc block of activations, and we're going to ask how correlated are the activations across different channels. So to explain what I mean by this may be slightly cryptic phrase, let's take this block of activations and let me shade the different channels by a different colors. So in this below example, we have say five channels and which is why I have five shades of color here. In practice, of course, in neural network we usually have a lot more channels than five, but using just five makes it drawing easier. But to capture the style of an image, what you're going to do is the following. Let's look at the first two channels. Let's see for the red channel and the yellow channel and say how correlated are activations in these first two channels. So, for example, in the lower right hand corner, you have some activation in the first channel and some activation in the second channel. So that gives you a pair of numbers. And what you do is look at different positions across this block of activations and just look at those two pairs of numbers, one in the first channel, the red channel, one in the yellow channel, the second channel. And you just look at these two pairs of numbers and see when you look across all of these positions, all of these nh by nw positions, how correlated are these two numbers. So, why does this capture style? Let's look another example. Here's one of the visualizations from the earlier video. This comes from again the paper by Matthew Zeiler and Rob Fergus that I have reference earlier. And let's say for the sake of arguments, that the red

neuron corresponds to, and let's say for the sake of arguments, that the red channel corresponds to this neurons so we're trying to figure out if there's this little vertical texture in a particular position in the nh and let's say that this second channel, this yellow second channel corresponds to this neuron, which is vaguely looking for orange colored patches. What does it mean for these two channels to be highly correlated? Well, if they're highly correlated what that means is whatever part of the image has this type of subtle vertical texture, that part of the image will probably have these orange-ish tint. And what does it mean for them to be uncorrelated? Well, it means that whenever there is this vertical texture, it's probably won't have that orange-ish tint. And so the correlation tells you which of these high level texture components tend to occur or not occur together in part of an image and that's 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 other things as well, how often they occur and 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 occurs or doesn't occur with this orange-ish tint and this gives you a measure of how similar is the style of the generated image to the style of the input style image. So let's now formalize this intuition. So what you can to do is given an image computes something called a style matrix, which will measure all those correlations we talks about on the last slide. So, more formally, let's let a superscript l, subscript i, j,k denote the activation at position i,j,k in hidden layer l. So i indexes into the height, j indexes into the width, and k indexes across the different channels. So, in the previous slide, we had five channels that k will index across those five channels. So what the style matrix will do is you're going to compute a matrix clauses G superscript square bracketed l. This is going to be an nc by nc dimensional matrix, so it'd be a square matrix. Remember you have nc channels and so you have an nc by nc dimensional matrix in order to measure how correlated each pair of them is. So particular G, l, k, k prime will measure how correlated are the activations in channel k compared to the activations in channel k prime. Well here, k and k prime will range from 1 through nc, the number of channels they're all up in that layer. So more formally, the way you compute G, l and I'm just going to write down the formula for computing one elements. So the k, k prime elements of this. This is going to be sum of a i, sum of a j, of deactivation and that layer i, j, k times the activation at i, j, k prime. So, here, remember i and j index across to a different positions in the block, indexes over the height and width. So i is the sum from one to nh and j is a sum from one to nw and k here and k prime index over the channel so k and k prime range from one to the total number of channels in that layer of the neural network. So all this is doing is summing over the different positions that the image over the height and width and just multiplying the activations together of the channels k and k prime and that's the definition of G,k,k prime. And you do this for every value of k and k prime to compute this matrix G, also called the style matrix. And so notice that if both of these activations tend to be lashed together, then G, k, k prime will be large, whereas if they are uncorrelated then g,k, k prime might be small. And technically, I've been using the term correlation to convey intuition but this is actually the unnormalized cross of the areas because we're not subtracting out the mean and this is just multiplied by these elements directly. So this is how you compute the style of an image. And you'd

actually do this for both the style image s,n for the generated image G. So just to distinguish that this is the style image, maybe let me add a round bracket S there, just to denote that this is the style image for the image S and those are the activations on the image S. And what you do is then compute the same thing for the generated image. So it's really the same thing summarized sum of a j, a, i, j, k, l, a, i, j,k,l and the summation indices are the same. Let's follow this and you want to just denote this is for the generated image, I'll just put the round brackets G there. So, now, you have two matrices they capture what is the style with the image s and what is the style of the image G. And, by the way, we've been using the alphabet capital G to denote these matrices. In linear algebra, these are also called the grand matrix of these in called grand matrices but in this video, I'm just going to use the term style matrix because this term grand matrix that most of these using capital G to denote these matrices. Finally, the cost function, the style cost function. If you're doing this on layer l between s and G, you can now define that to be just the difference between these two matrices, G l, G square and these are matrices. So just take it from the previous one. This is just the sum of squares of the element wise differences between these two matrices and just divides this out this is going to be sum over k, sum over k prime of these differences of s, k, k prime minus G l, G, k, k prime and then the sum of square of the elements. The authors actually used this for the normalization constants two times of nh, nw, in that layer, nc in that layer and I'll square this and you can put this up here as well. But a normalization constant doesn't matter that much because this causes multiplied by some hyperparameter b anyway. So just to finish up, this is the style cost function defined using layer l and as you saw on the previous slide, this is basically the Frobenius norm between the two star matrices computed on the image s and on the image G Frobenius on squared and never by the just low normalization constants, which isn't that important. And, finally, it turns out that you get more visually pleasing results if you use the style cost function from multiple different layers. So, the overall style cost function, you can define as sum over all the different layers of the style cost function for that layer. We should define the book weighted by some set of parameters, by some set of additional hyperparameters, which we'll denote as lambda l here. So what it does is allows you to use different layers in a neural network. Well of the early ones, which measure relatively simpler low level features like edges as well as some later layers, which measure high level features and cause a neural network to take both low level and high level correlations into account when computing style. And, in the following exercise, you gain more intuition about what might be reasonable choices for this type of parameter lambda as well. And so just to wrap this up, you can now define the overall cost function as alpha times the content cost between c and G plus beta times the style cost between s and G and then just create in the sense or a more sophisticated optimization algorithm if you want in order to try to find an image G that normalize, that tries to minimize this cost function j of G. And if you do that, you can generate pretty good looking neural artistic and if you do that you'll be able to generate some pretty nice novel artwork. So that's it for neural style transfer and I hope you have fun implementing it in this week's printing exercise. Before wrapping up this week, there's just one last thing I want to share of you, which is how to do convolutions over 1D or 3D data rather than over only 2D images. Let's go into the last video.

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt
Lecture slides	pptx

Would you like to help us translate the transcript and subtitles into additional languages?