Style Transfer Report

The idea over here is to input two images (Content image and Style image), and use the contour lines from the content image and the texture and the colors from the style image, to get a mixed image. The content, style and mixed (initially noise) images are sent through the neural networks. The different loss functions at different layers are calculated and the corresponding weights are generated. There are two types of loss function employed, viz. content loss and style loss. Gradient of these combined loss functions are calculated and used to update the mixed image. This is repeated a number of times.

The necessary files are imported. A VGG-16 model is used instead of the original VGG-19 model. It is imported using **import vgg16.** The vgg data is then downloaded using vgg16.maybe_download() in a specified directory. The image is loaded and converted into a floating point array by the function load_image(). The pixels are checked to be between 0 and 255 and the image is saved as a jpeg file by save_image(). The function that plots the large image is plot_image_big(). For plotting the content, style and mixed image the function plot_images() is used. The loss function for the content image is create_content_loss() which uses the mean square error function mean_squared_error(). It is the Mean Squared Error of the feature activations in the given layers in the model, between the content-image and the mixed-image.

The Gram-matrix is essentially just a matrix of dot-products for the vectors of the feature activations of a style-layer. If an entry in the Gram-matrix has a value close to zero then it means the two features in the given layer do not activate simultaneously for the given style-image and vice versa. The function used to calculate the Gram-matrix is gram_matrix(). The loss function for the style image which uses the gram_matrix() function is create_style_loss(). The main optimization algorithm for the Style-Transfer which basically does gradient descent on the loss-functions is style_transfer(). The loss function for denoising the mixed image is create_denoise_loss() which can be minimized so as to suppress some of the noise in the image.

The images are then uploaded, the content and style ids are specified and then the style transfer is performed.

Here are the transitions from zero iteration to 60 iterations.

Content

Mixed

Style

Content

Mixed

Style

Content

Mixed

Style

Content

Mixed

Style

Content

Mixed

Style

Content

Mixed

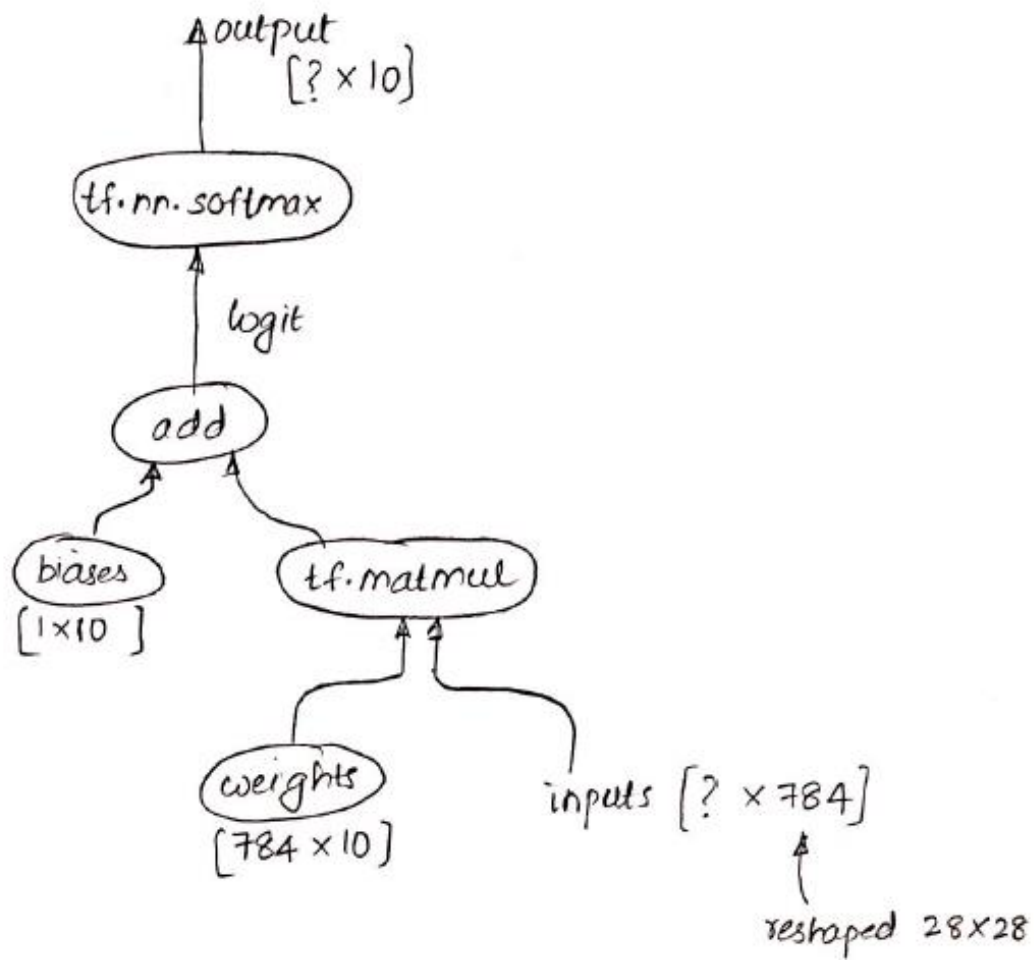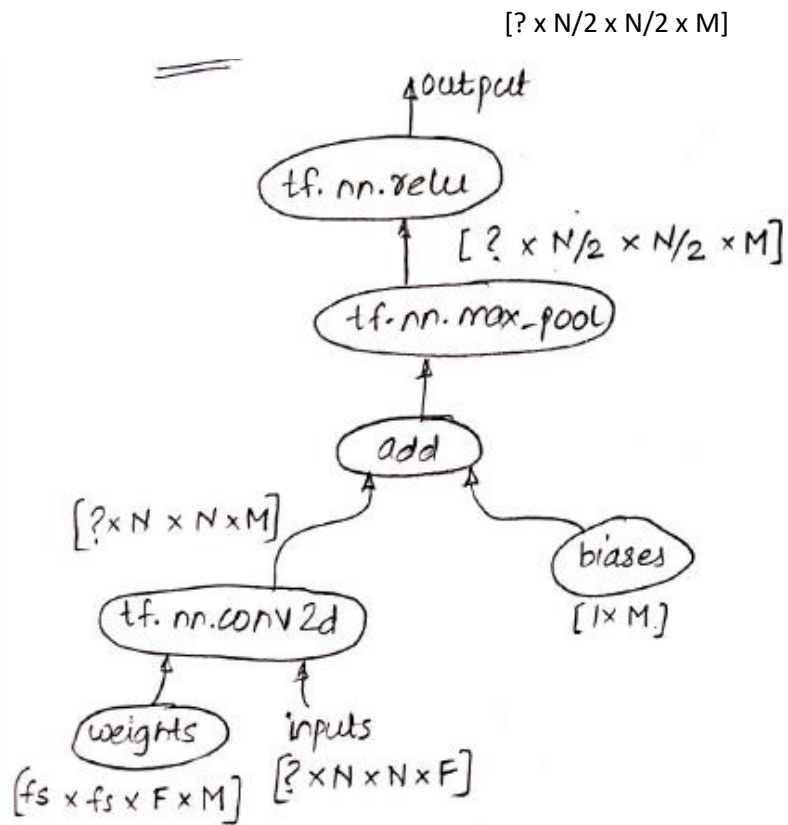Style

Content

Mixed

Style

Content

Mixed

Style

Graph for Part 1

output
$[? \times 10]$

tf.nn.softmax

logit

add

biases
$[1 \times 10]$

tf.matmul

weights
$[784 \times 10]$

inputs $[? \times 784]$

reshaped 28×28

Graph for Part 2

[? x N/2 x N/2 x M]

↑ output

(tf. nn. relu)

↑ [? x N/2 x N/2 x M]

(tf. nn. max_pool)

↑

(add)

[?x N x N x M]

(tf. nn. conv 2d)   (biases)

                          [I x M]

(weights)   inputs

[fs x fs x F x M]   [? x N x N x F]

	conv_1	conv_2
M	16	32 ⇒ [14x14x16]
F	1	16 ⇒ [7x7x36]