Supplementary Material for Visualizing Deep Neural Network by Alternately Image Blurring and Deblurring

Feng Wang[a], Haijun Liu[a], Jian Cheng[a,*]

[a]*School of Electronic Engineering, University of Electronic Science and Technology of China*

1. The necessity of weight decay while visualizing VGGNet

In the main text of our paper, we mentioned that when generating the [300, 300] images for AlexNet and GoogLeNet, no constraint is added to the optimization procedure, i.e. only image jittering, image blurring and deblurring are applied. However, we found that some pixels of VGGNet's visualization would be extremely large during training. Through the local pixel sharing methods such as image resizing and blurring, the big value will also spread to pollute the neighbour pixels(Figure. 4). To suppress the big value, we follow [2] to add weight decay during training.

2. Details of Automatically Learning Rate Tuning

In the experiment section of our main paper, we introduced a method to get the best learning rate automatically. The method is based on an observation that the most visually satisfying image always corresponds to the highest activation. This phenomenon cannot be generalized to the first layer, where the activation is a linear function of the input image's magnitude. But in higher layers, the activation is a highly distorted function w.r.t. the input. It is a mixture of functions that are either direct or inverse proportion to the input image's magnitude. So either too large or too small pixel scale may lead to lower activation.

In this supplement material, we will show some empirical evidence of the observation. We choose two neurons in two layers with different depth, one from the mid-level Inception-4a layer, another from the class-level neuron of GoogLeNet model. Then we train the visualization directly by maximising the chosen neurons without any constraints or regularizations. The training procedure is seen to be converged when the least-square slope of the activation is below a certain small number, such as 1e-3. As we can see from the figures, there do exsits a learning rate which corresponds to the highest activation that creates the best visually acceptable images. Lower learning rate

*Corresponding author. Tel/Fax: +86-028-61830064
Email address: justus.cheng@gmail.com (Jian Cheng)

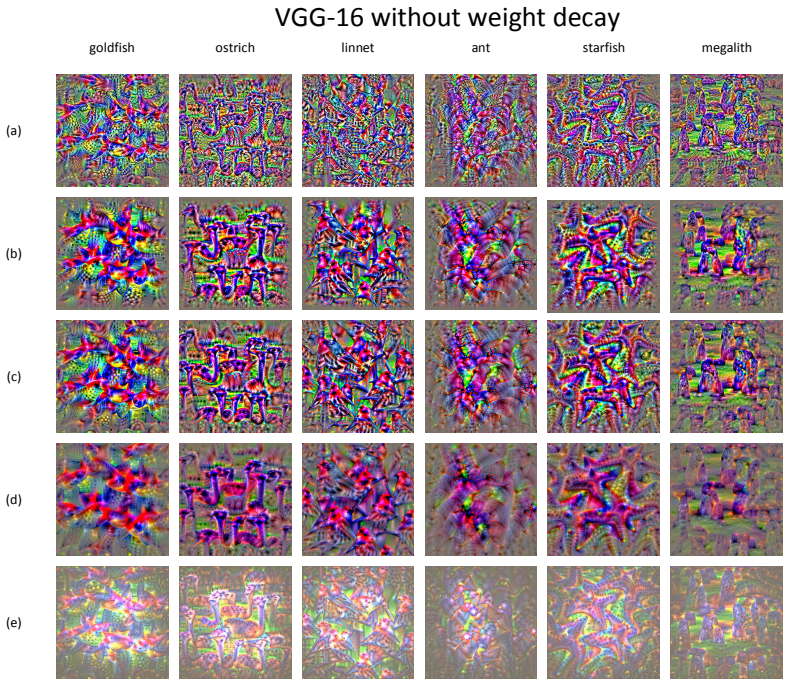

Figure 1: VGG-16 visualization without weight decay constraint during training. Compared these images with the version in the Figure 12 of the main text, b-d rows are polluted by extreme values in different degrees. Image blurring suffers the most from the extreme value problem. Best viewed in color, zoomed in.

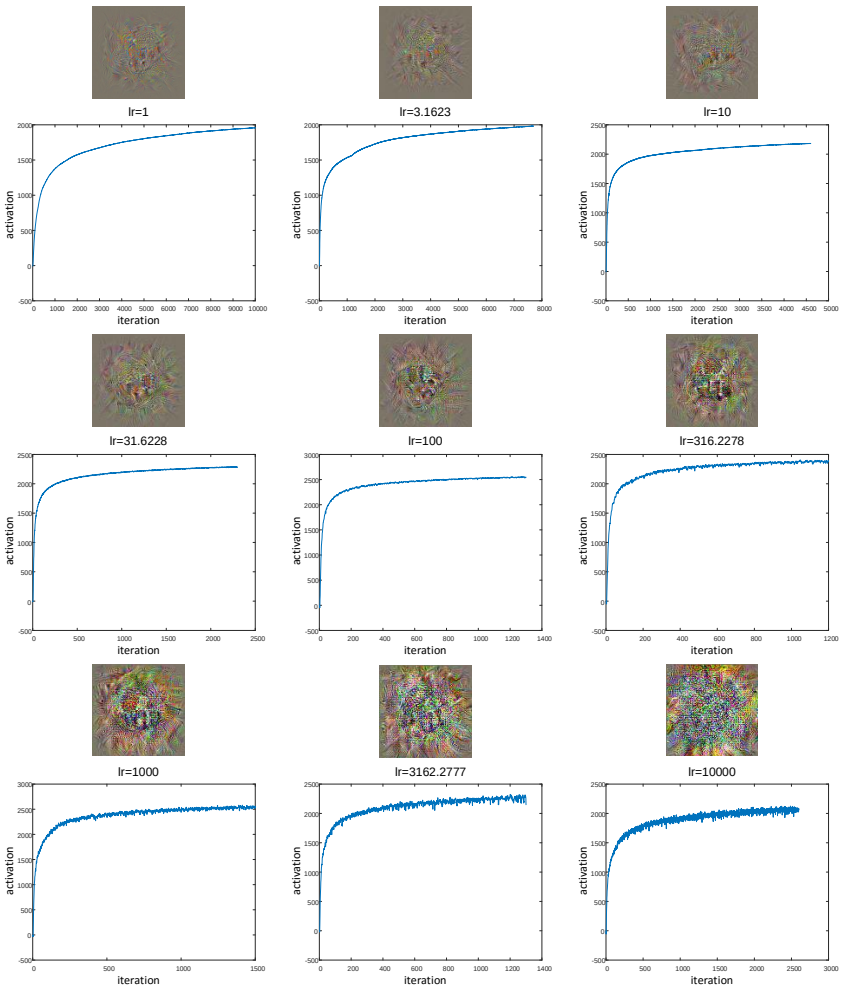

Figure 2: The activations during training of one chosen neuron in Inception-4a layer using different learning rate.

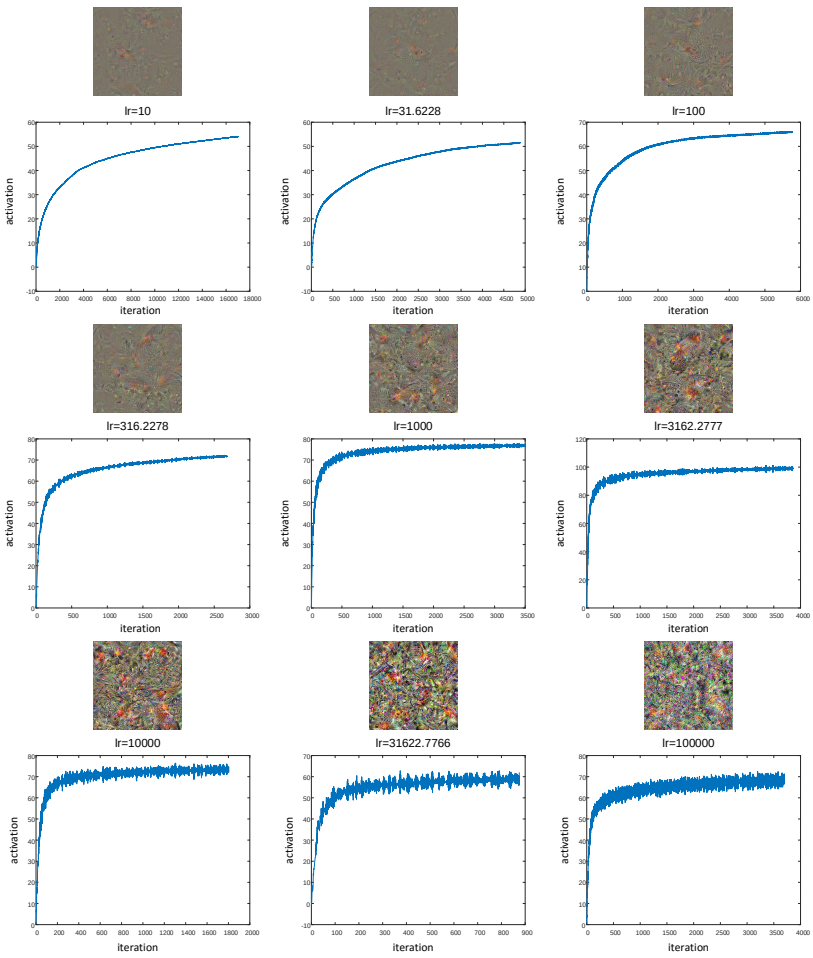

Figure 3: The activations during training of one chosen neuron in the classification layer using different learning rate.

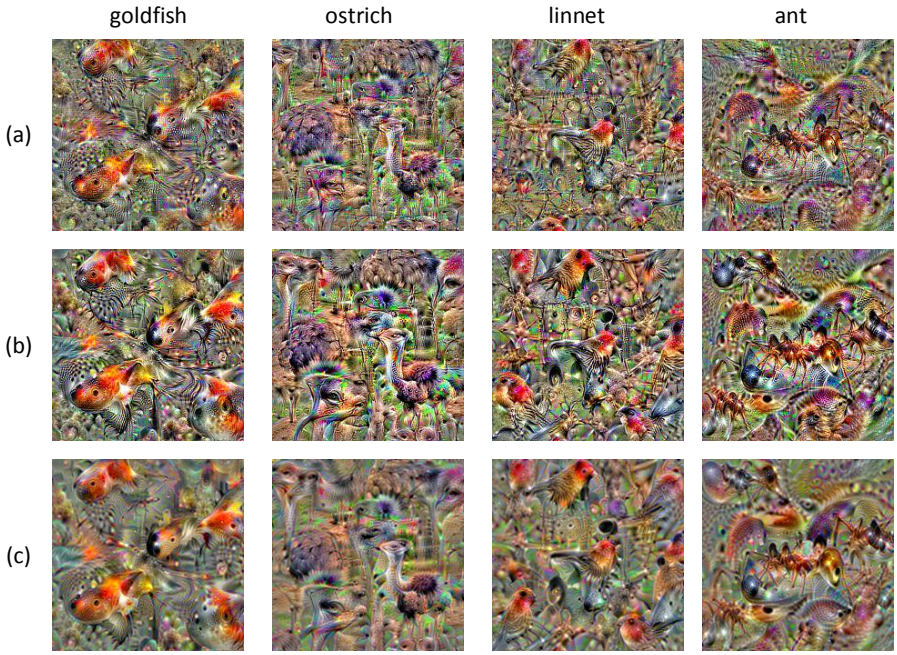

Figure 4: Effect of different interpolations. These images are created by following strategies: (a) translation only, (b) translation and scaling with "nearest" interpolation, (c) translation and scaling with "bilinear" interpolation.

creates images with lower magnitude, and we can hardly see the instance from the visualizations created by low learning rate. Higher learning rate creates images with higher magnitude and more noises. With the highest learning rate, the generated images have too much noises to be recognized by human eyes.

3. Image resizing with nearest interpolation

In the main text of our paper, we analysed why image jittering benefited to the image generation: the key factor is information sharing. Information sharing across different locations and scales will provide extract the patterns against translation and scaling variation. Information sharing in neighbour pixels will lead to more smooth images. To further prove this, we add another experiment, using nearest interpolation instead of the common bilinear interpolation method during image resizing. Image resizing with nearest interpolation has pixel sharing in the location of the back-propagated gradients, but do not have neighbour pixel sharing. In this way, the algorithm will produce more recognisable but sharpened images.

4. Other Details

Here we enumerate some less important instructions when exploiting our algorithm.

1. The initial image should not be zero nor gaussian while noise with big standard deviation. A good practice is to fill the image with values sampled from uniform gaussian distribution $N(0, 1)$.
2. In order to compute a visualization fast, it is important to calculate the receptive field of a specified neuron (channel in fully convolution network manner) before optimizing for the visualization image. This step can be done by setting a big input size, giving a 1 to the center of the gradient in the output side of the specified channel while other positions and channels to 0, then applying the back propagation algorithm once. Finally, the receptive field size can be obtained by counting the width and height of the non-zero values on the gradient.
3. When generating a specified neuron, don't send the gradient after the ReLU layer, or the it may not be passed through the ReLU's gradient,

$$\frac{dReLU(x)}{dx} = \begin{cases} 1 & if\ x > 0 \\ 0 & if\ x \leq 0 \end{cases} \tag{1}$$

   This step is often ignored when visualizing a concatenate layer in the Inception module[3] or the element-wise layer in a residual block[1]. When visualizing a residual block, it is necessary to train the residual part firstly, then finetune the training image by adding the identity path.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[2] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–9, 2015.