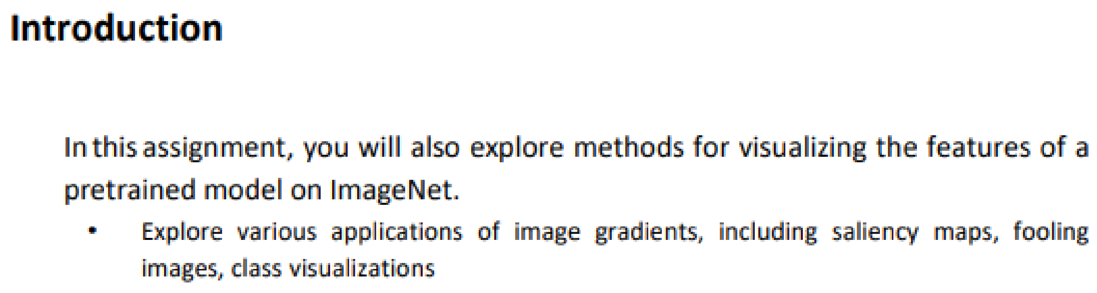
计算机科学与技术学院神经网络与深度学习课程实验报告

实验题目：Network Visualization (PyTorch)		学号：201900301174
日期：2021-11-20	班级： 智能 19	姓名： 韩旭
Email：hanx@mail.sdu.edu.cn		

实验目的：



Introduction

In this assignment, you will also explore methods for visualizing the features of a pretrained model on ImageNet.

- Explore various applications of image gradients, including saliency maps, fooling images, class visualizations

实验软件和硬件环境：

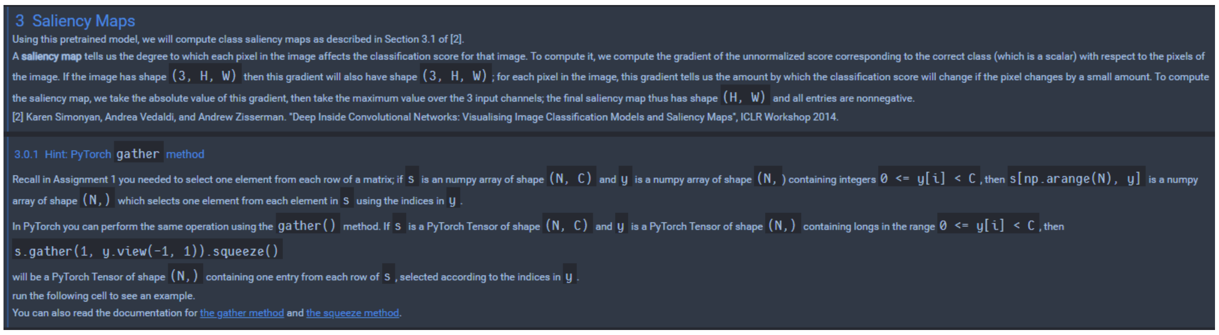
Termius

Jupyter notebook

RTX3090

Xeon Gold 6226R

实验原理和方法：



3 Saliency Maps

Using this pretrained model, we will compute class saliency maps as described in Section 3.1 of [2].

A **saliency map** tells us the degree to which each pixel in the image affects the classification score for that image. To compute it, we compute the gradient of the unnormalized score corresponding to the correct class (which is a scalar) with respect to the pixels of the image. If the image has shape `(3, H, W)` then this gradient will also have shape `(3, H, W)`; for each pixel in the image, this gradient tells us the amount by which the classification score will change if the pixel changes by a small amount. To compute the saliency map, we take the absolute value of this gradient, then take the maximum value over the 3 input channels; the final saliency map thus has shape `(H, W)` and all entries are nonnegative.

[2] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

3.0.1 Hint: PyTorch `gather` method

Recall in Assignment 1 you needed to select one element from each row of a matrix; if `s` is an numpy array of shape `(N, C)` and `y` is a numpy array of shape `(N, )` containing integers `0 <= y[i] < C`, then `s[np.arange(N), y]` is a numpy array of shape `(N,)` which selects one element from each element in `s` using the indices in `y`.

In PyTorch you can perform the same operation using the `gather()` method. If `s` is a PyTorch Tensor of shape `(N, C)` and `y` is a PyTorch Tensor of shape `(N,)` containing longs in the range `0 <= y[i] < C`, then

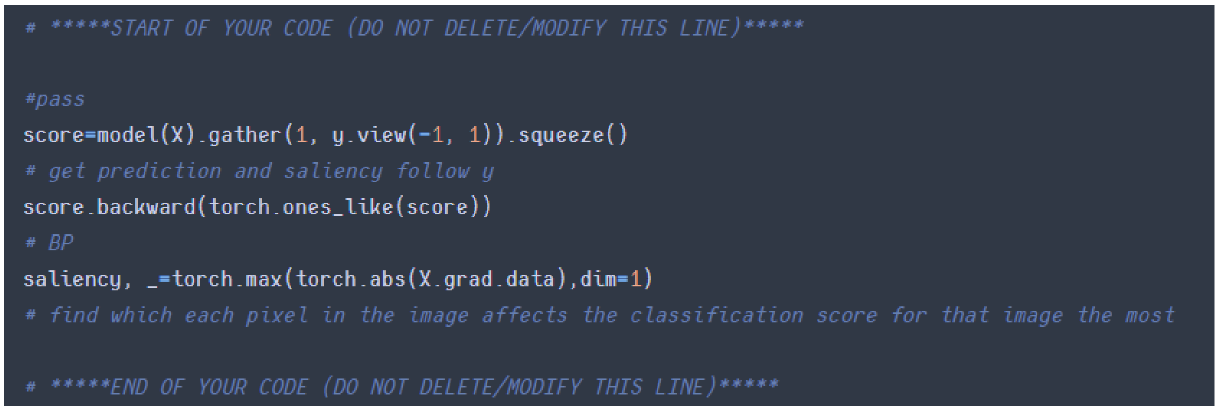
`s.gather(1, y.view(-1, 1)).squeeze()`

will be a PyTorch Tensor of shape `(N,)` containing one entry from each row of `s`, selected according to the indices in `y`.

run the following cell to see an example.

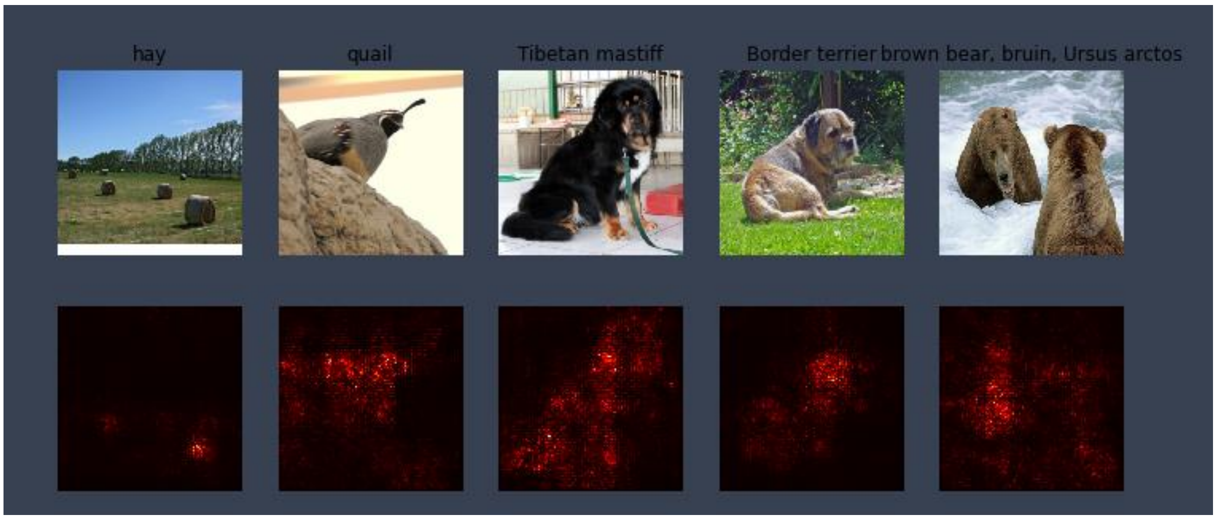
You can also read the documentation for the gather method and the squeeze method.

代码：



```python
# *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

#pass
score=model(X).gather(1, y.view(-1, 1)).squeeze()
# get prediction and saliency follow y
score.backward(torch.ones_like(score))
# BP
saliency, _=torch.max(torch.abs(X.grad.data),dim=1)
# find which each pixel in the image affects the classification score for that image the most

# *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
```

可视化：

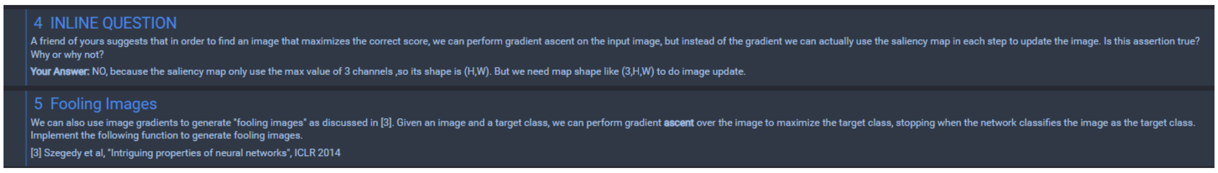


可以看到每张图里面的显著部分，跟我们预想的差不多，都是图像的主要分类内容。

INLINE QUESTION

A friend of yours suggests that in order to find an image that maximizes the correct score, we can perform gradient ascent on the input image, but instead of the gradient we can actually use the saliency map in each step to update the image. Is this assertion true? Why or why not?

Your Answer: NO, because the saliency map only use the max value of 3 channels ,so its shape is (H,W). But we need map shape like (3,H,W) to do image update.



4 INLINE QUESTION

A friend of yours suggests that in order to find an image that maximizes the correct score, we can perform gradient ascent on the input image, but instead of the gradient we can actually use the saliency map in each step to update the image. Is this assertion true? Why or why not?

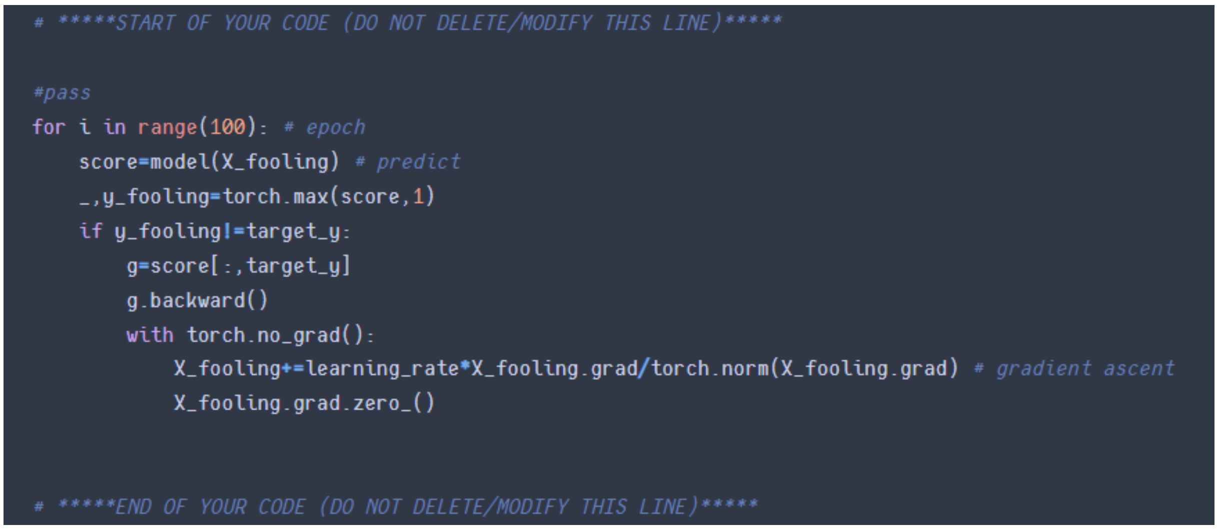
Your Answer: NO, because the saliency map only use the max value of 3 channels ,so its shape is (H,W). But we need map shape like (3,H,W) to do image update.

5 Fooling Images

We can also use image gradients to generate "fooling images" as discussed in [3]. Given an image and a target class, we can perform gradient **ascent** over the image to maximize the target class, stopping when the network classifies the image as the target class. Implement the following function to generate fooling images.

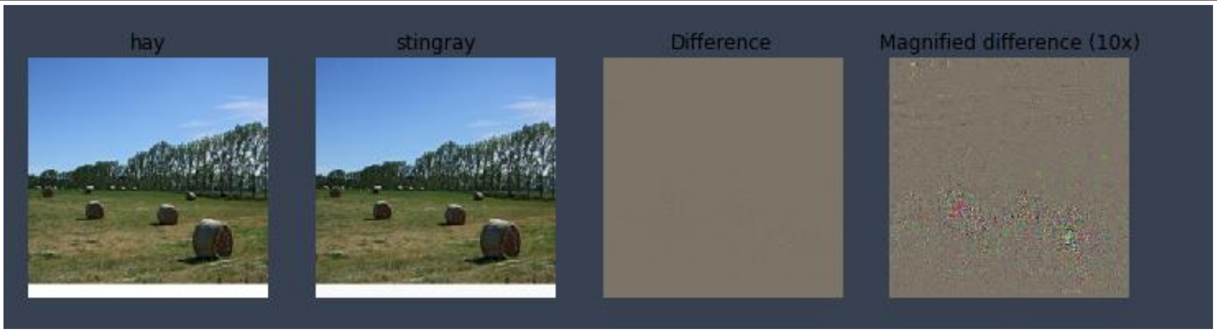
[3] Szegedy et al, "Intriguing properties of neural networks", ICLR 2014

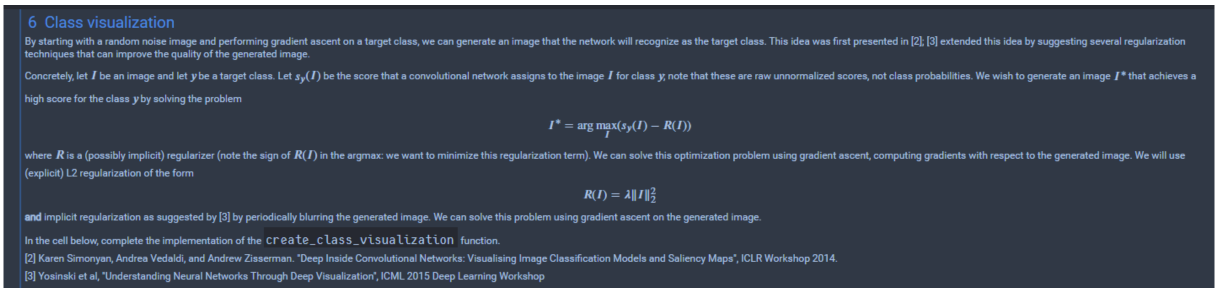
代码：



```
# *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****


#pass
for i in range(100): # epoch
    score=model(X_fooling) # predict
    _,y_fooling=torch.max(score,1)
    if y_fooling!=target_y:
        g=score[:,target_y]
        g.backward()
        with torch.no_grad():
            X_fooling+=learning_rate*X_fooling.grad/torch.norm(X_fooling.grad) # gradient ascent
            X_fooling.grad.zero_()



# *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
```

结果可视化：





6 Class visualization

By starting with a random noise image and performing gradient ascent on a target class, we can generate an image that the network will recognize as the target class. This idea was first presented in [2]; [3] extended this idea by suggesting several regularization techniques that can improve the quality of the generated image.

Concretely, let I be an image and let y be a target class. Let $s_y(I)$ be the score that a convolutional network assigns to the image I for class y; note that these are raw unnormalized scores, not class probabilities. We wish to generate an image I^* that achieves a high score for the class y by solving the problem

$$I^* = \arg\max_I (s_y(I) - R(I))$$

where R is a (possibly implicit) regularizer (note the sign of $R(I)$ in the argmax: we want to minimize this regularization term). We can solve this optimization problem using gradient ascent, computing gradients with respect to the generated image. We will use (explicit) L2 regularization of the form

$$R(I) = \lambda \|I\|_2^2$$

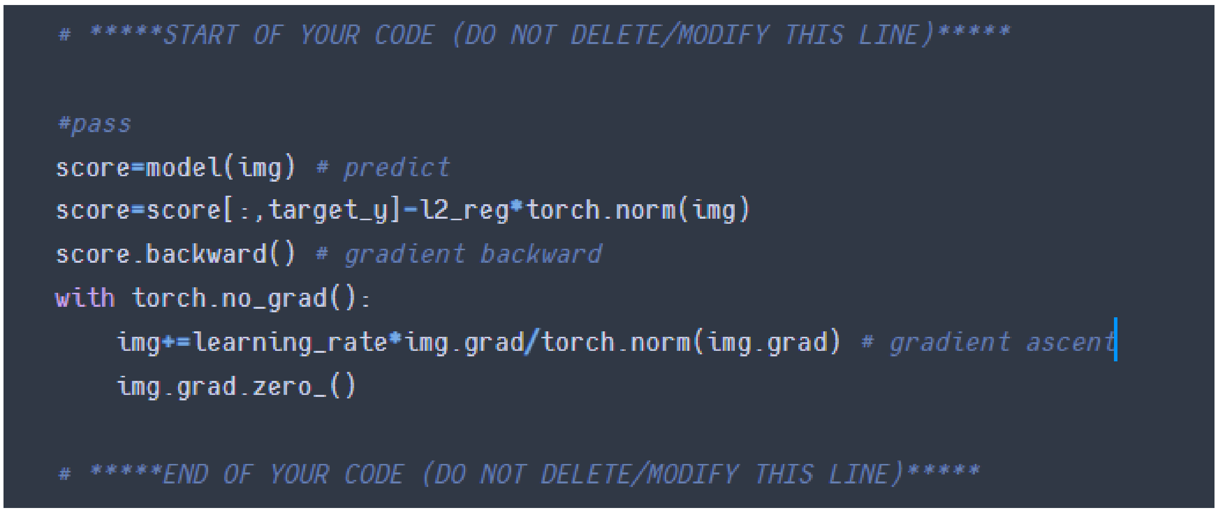
and implicit regularization as suggested by [3] by periodically blurring the generated image. We can solve this problem using gradient ascent on the generated image.

In the cell below, complete the implementation of the `create_class_visualization` function.

[2] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

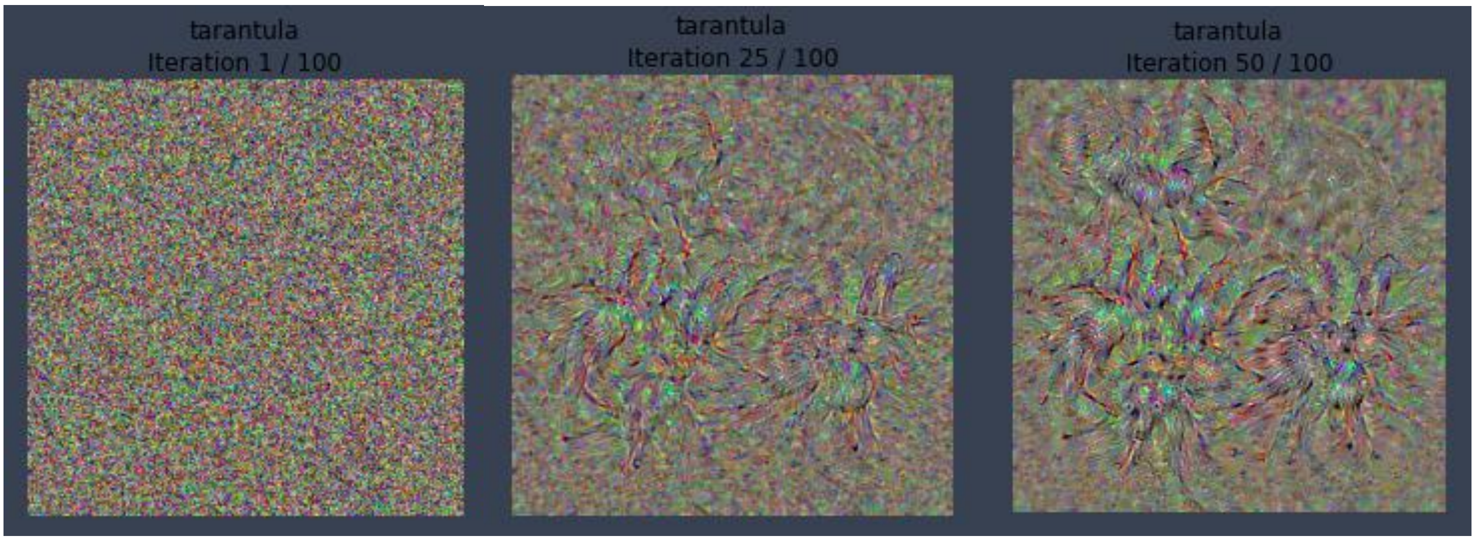
[3] Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML 2015 Deep Learning Workshop

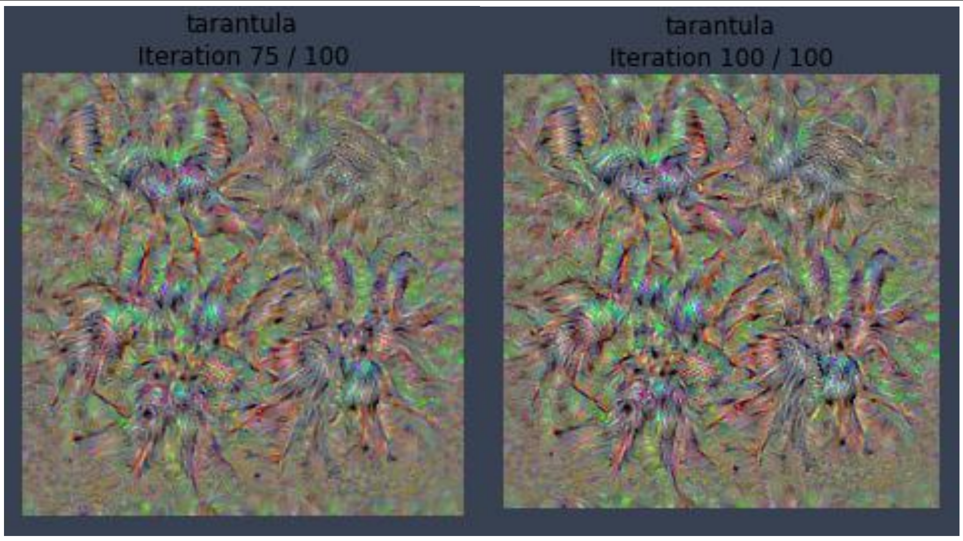
代码：

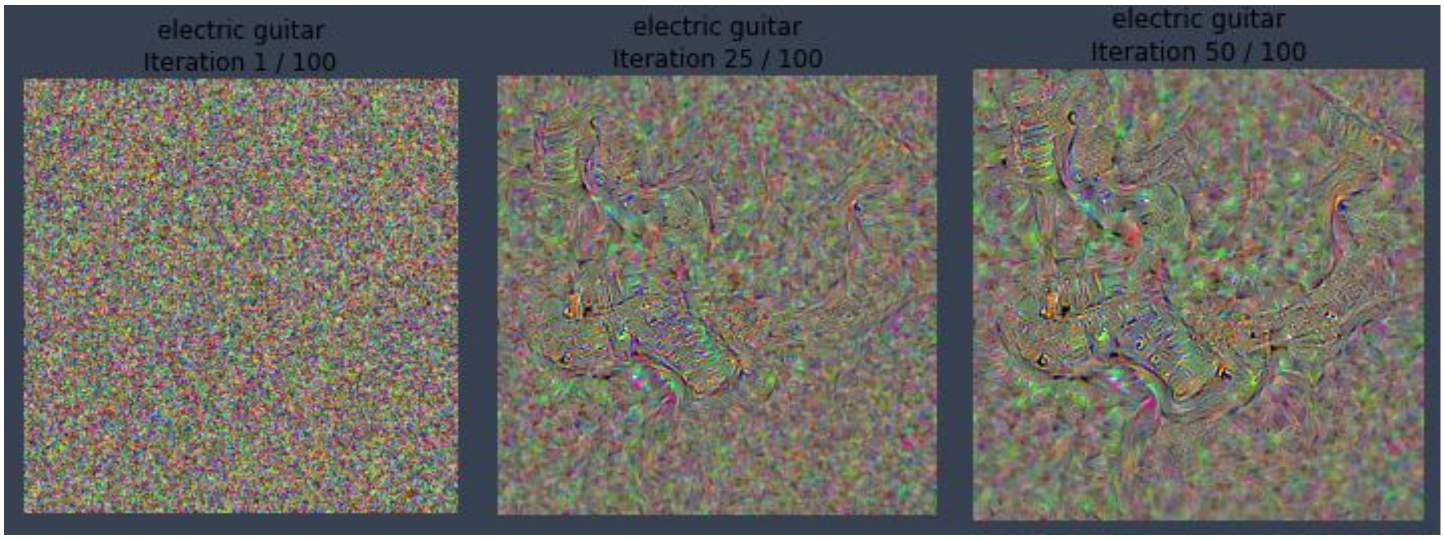


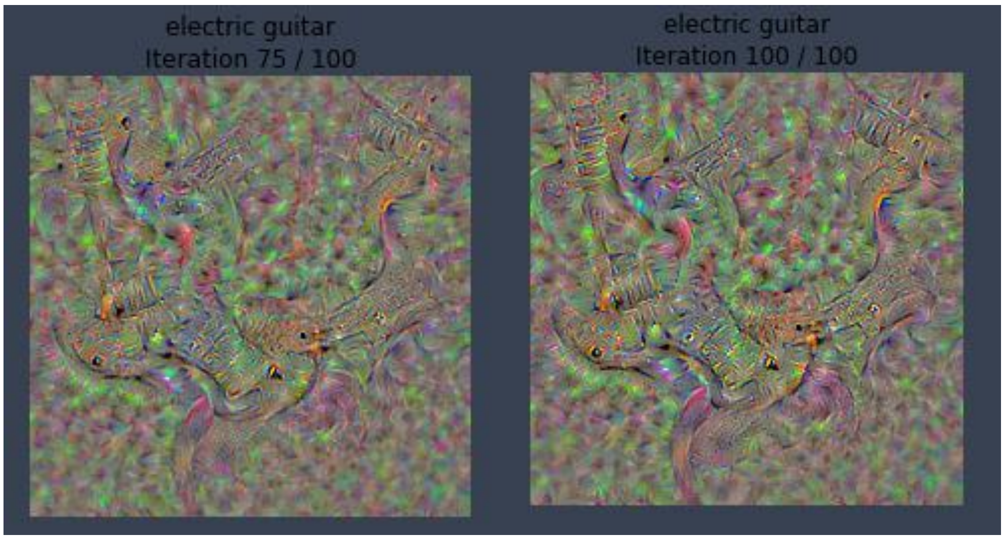
```python
# *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

#pass
score=model(img) # predict
score=score[:,target_y]-l2_reg*torch.norm(img)
score.backward() # gradient backward
with torch.no_grad():
    img+=learning_rate*img.grad/torch.norm(img.grad) # gradient ascent
    img.grad.zero_()

# *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
```

结果：




tarantula
Iteration 75 / 100
tarantula
Iteration 100 / 100


electric guitar
Iteration 1 / 100
electric guitar
Iteration 25 / 100
electric guitar
Iteration 50 / 100


electric guitar
Iteration 75 / 100
electric guitar
Iteration 100 / 100

结论分析与体会： 1， 既然有梯度下降也就有梯度上升。其实是下降还是上升，就是看我们的最后的目标是要减小还是增大，根据梯度本身的规律，要减小目标就用梯度下降，要增大目标就用梯度上升。我们这里要最大化正确类别分值，则要用梯度上升。 2， 只有 shape 是 torch.Size([N])的 tensor 才可以直接使用.backward()，否则还得在里面加上和要反向传播维度一样的 tensor，也就是一个权重向量，一般就是值为 1
就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟 1－3 道问答题： 1， 梯度更新的时候会出错？不要忘记清零，因为 tensor.grad 是会累加的。 2， Class visualization 的思路？ 3， 直接从随机噪声合成一张图片，再加上一些正则化手段，可以使生成的图片更加平滑（包括 L2 正则化和 jitter/blur）。