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： Homework 6		学号：201900130024
日期：2021.11.19	班级： 数据 19	姓名： 刘士渤
Email：liuburger@qq.com		

实验目的：

利用 RNN 生成句子

实验软件和硬件环境：

VScode JupyterNoteBook

联想拯救者 Y7000p

实验原理和方法：

RNN

实验步骤：（不要求罗列完整源代码）

1. 补全 NetworkVisualization-Pytorch.ipynb：
   - compute_saliency_maps：

     提示给出了一种新的方法：

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

观察样例，发现这个句式能够提取以 y 为下标的 s 的每一个向量中的元素：

```
tensor([[-0.7335, -0.3691,  0.1478,  0.3098, -0.2637],
        [-0.2382, -0.9886,  1.1639, -0.1064,  0.2294],
        [-0.0168, -0.2740, -0.8286, -0.5712,  1.2797],
        [-0.3749, -0.9427, -0.0254, -1.4340, -0.6948]])
tensor([1, 2, 1, 3])
tensor([-0.3691,  1.1639, -0.2740, -1.4340])
```

然后补全代码：

因为要得到 Tensor 张量所以要用[1.0]而不是 1.0；

题目中提示要取梯度的绝对值、并取 3 个输入 channel 的最大值。

```
#前向
scores=model(X)
scores=scores.gather(1,y.view(-1,1)).squeeze()
#后向
scores.backward(torch.FloatTensor([1.0]*scores.shape[0]))
#saliency
saliency,_=torch.max(X.grad.data.abs(),dim=1)
```

- make_fooling_image：
  通过 X_fooling 得到 scores，并得到取得最大值的下标；
  只有当 index=target_y 的时候才停止循环，否则用梯度上升最大化目标类；
  dX 的计算方法：

```
dX = learning_rate * g / ||g||_2
```

  torch.norm 默认是 L2 范数。
  在 grad 更新时，每一次运算后都需要将上一次的梯度记录清空，即.grad.data.zero_()

```
while True:
    scores=model(X_fooling)
    _,idx=torch.max(scores,1)
    if idx!=target_y:
        scores[:,target_y].backward()
        dX=learning_rate*X_fooling.grad.data/torch.norm(X_fooling.grad.data)
        X_fooling.data+=dX.data
        X_fooling.grad.data.zero_()
    else:
        break
```

- create_class_visualization：
  I 是 image，y 是目标类，sy(I)是卷积网络对 y 类图像 I 的分数，R(I)是 I 的正则项；

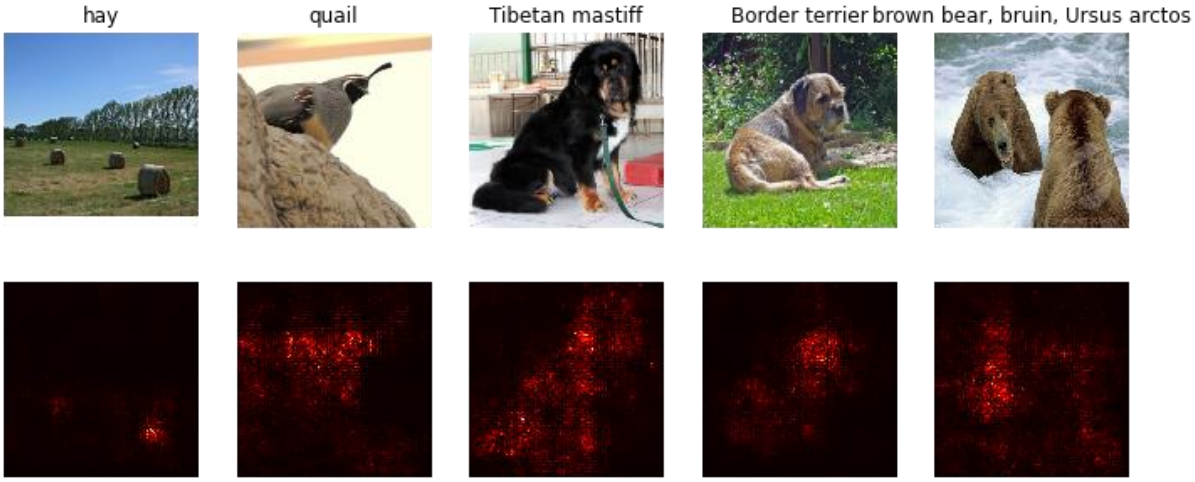
$$I^* = \arg\max_I (s_y(I) - R(I)) \qquad R(I) = \lambda \|I\|_2^2$$

  由 img 计算 scores；
  scores[:,target_y]即是 sy(img)，l2_reg 是 λ 的角色；
  同样用梯度上升最大化 scores。
  在 grad 更新时，每一次运算后都需要将上一次的梯度记录清空，即.grad.data.zero_()

```
scores=model(img)
score=scores[:,target_y]-l2_reg*torch.norm(img)
score.backward()
dX=learning_rate*img.grad.data/torch.norm(img.grad.data)
img.data+=dX
img.grad.data.zero_()
```

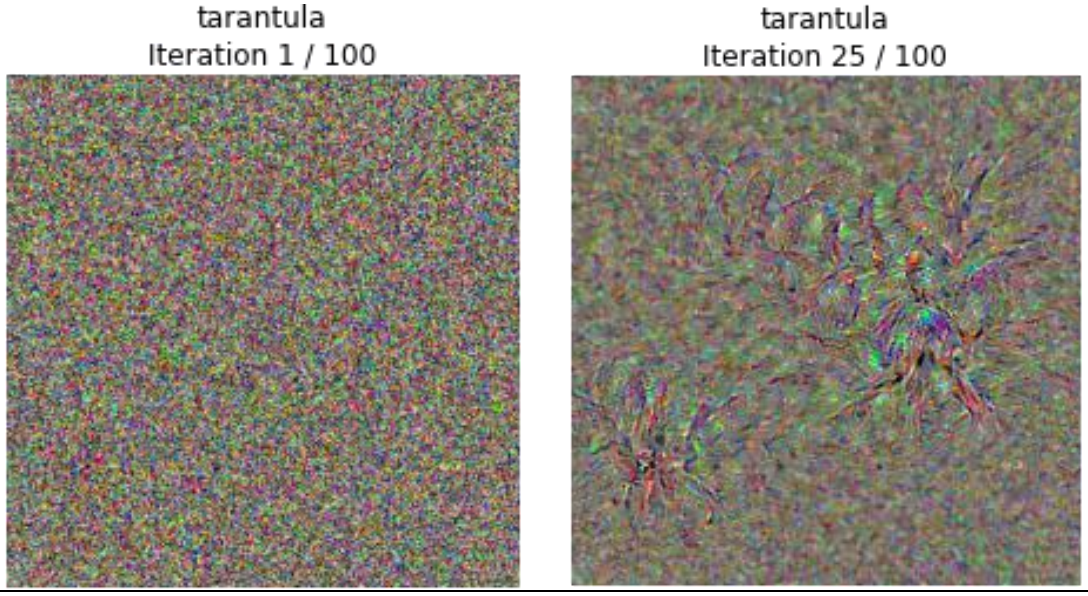
结论分析与体会：

1. 得到的 saliency maps：

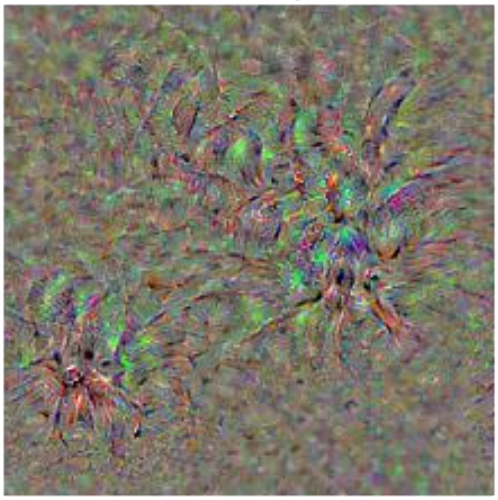


2. 看起来很小的 difference 会让模型将 hay 识别成 stingray：

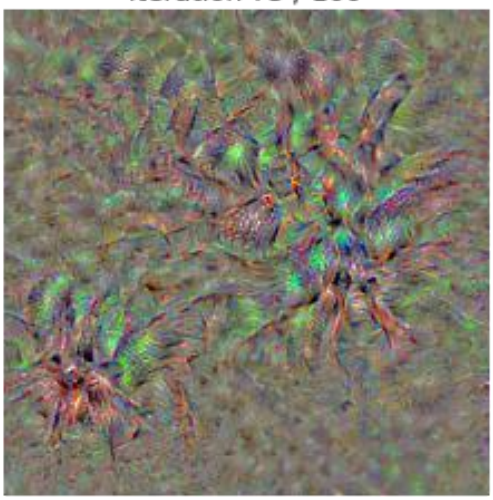
3. Tarantula 的迭代过程：



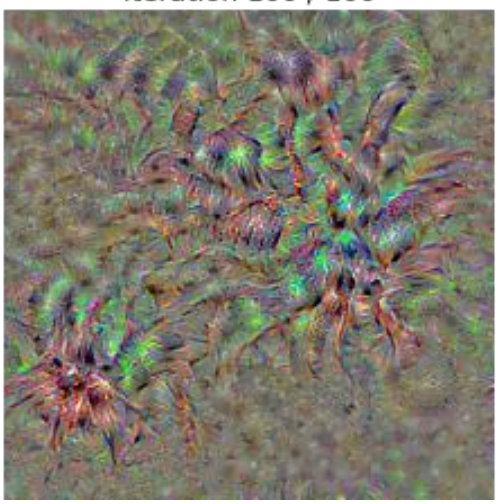
tarantula
Iteration 50 / 100



tarantula
Iteration 75 / 100

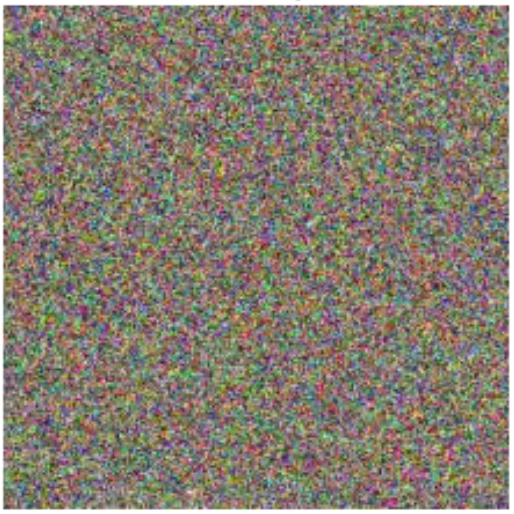


tarantula
Iteration 100 / 100

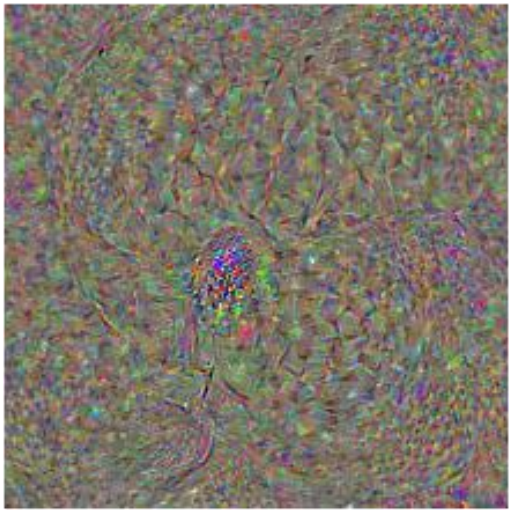


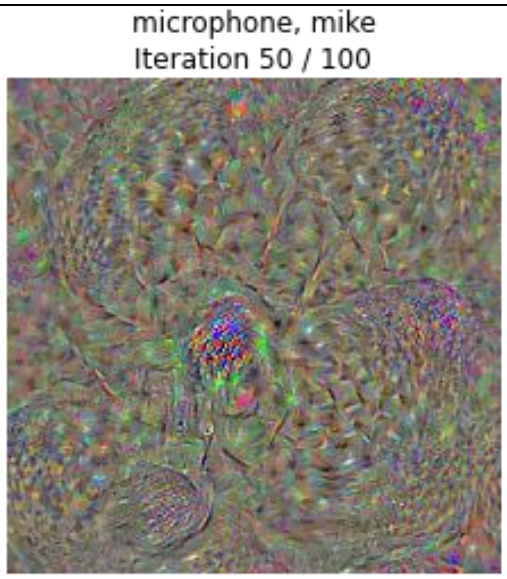
4. microphone，mike 的迭代过程（随机的）：

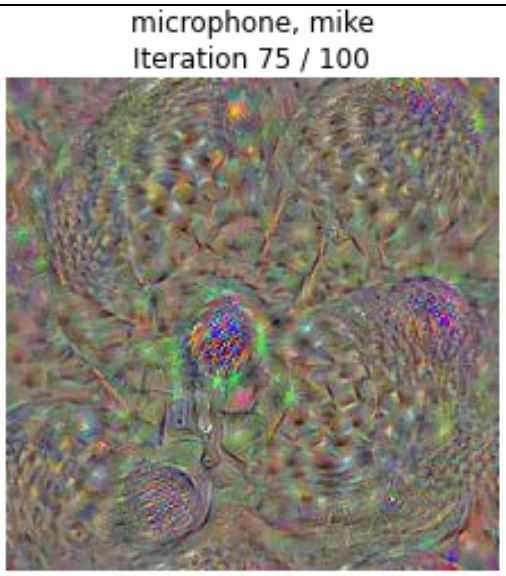
microphone, mike
Iteration 1 / 100

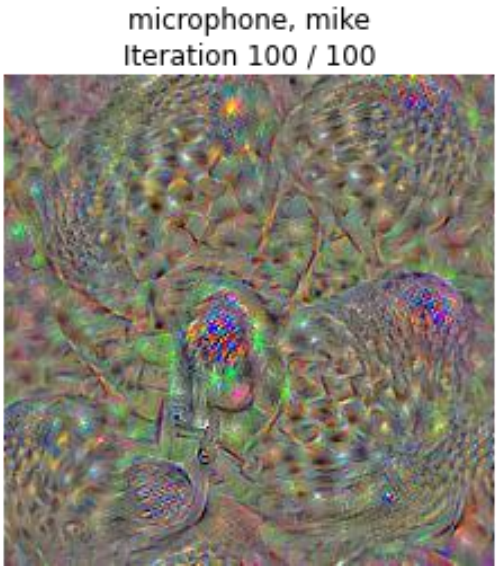


microphone, mike
Iteration 25 / 100



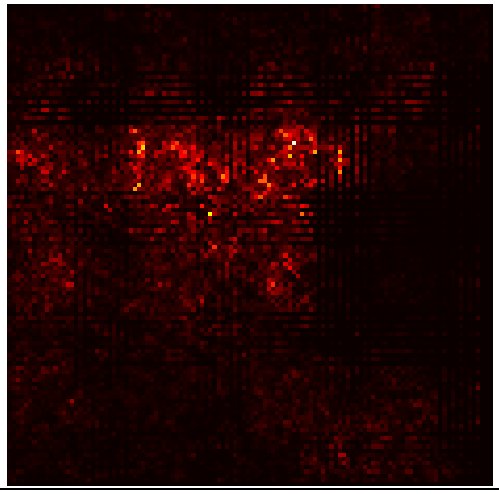






就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟 1－3 道问答题：

1. 由 saliency maps 的可视化可以得出这样一个问题：拿鹌鹑来说，模型学到的东西不完全是类别本身，而是包含了环境（如鹌鹑站立的岩石）；



这也就解释了为什么只是加了一点小小的噪音，模型就将干草识别成了黄貂鱼；

也就是说模型并没有学到某个类别的主要特征，而是将一些次要的特征甚至不属于该类别的特征当成了该类别的特征。