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：NetworkVisualization		学号：201900130015
日期：2021.11.18	班级： 智能班	姓名：李德锋
Email：ldf2878945468@163.com		

实验目的：

探索在 ImageNet 上可视化预训练模型的特征的方法

探索图像梯度的各种应用，包括显著性地图，欺骗图像，类可视化

实验软件和硬件环境：

Intel(R) Core(TM) i7-8550U CPU

华为云

实验原理和方法：

探索图像生成三种技术：

Saliency Maps：显著性图能很快告诉我们图像的哪一部分影响了网络做出的分类决定；

Fooling Images：扰动一张输入图像，使之对人眼似乎一模一样，却能让预训练模型误分类；

Class Visualization：我们可以合成一张图像使特定类别的分类分值最大化，这能给我们一些感觉：网络模型在分类时到底在寻找什么

实验步骤：（不要求罗列完整源代码）

1. Saliency Maps

显着图告诉我们图像中的每个像素对该图像的分类分数的影响程度

如果图像具有形状 (3, H, W)，则此渐变也将具有形状 (3, H, W)；对于图像中的每个像素，这个梯度告诉我们如果像素变化很小，分类分数将变化的量。为了计算显着图，我们取这个梯度的绝对值，然后取 3 个输入通道的最大值；因此最终的显着图具有形状 (H, W) 并且所有条目都是非负的

先进行前向传播计算各类的得分

```
scores = model(X)
```

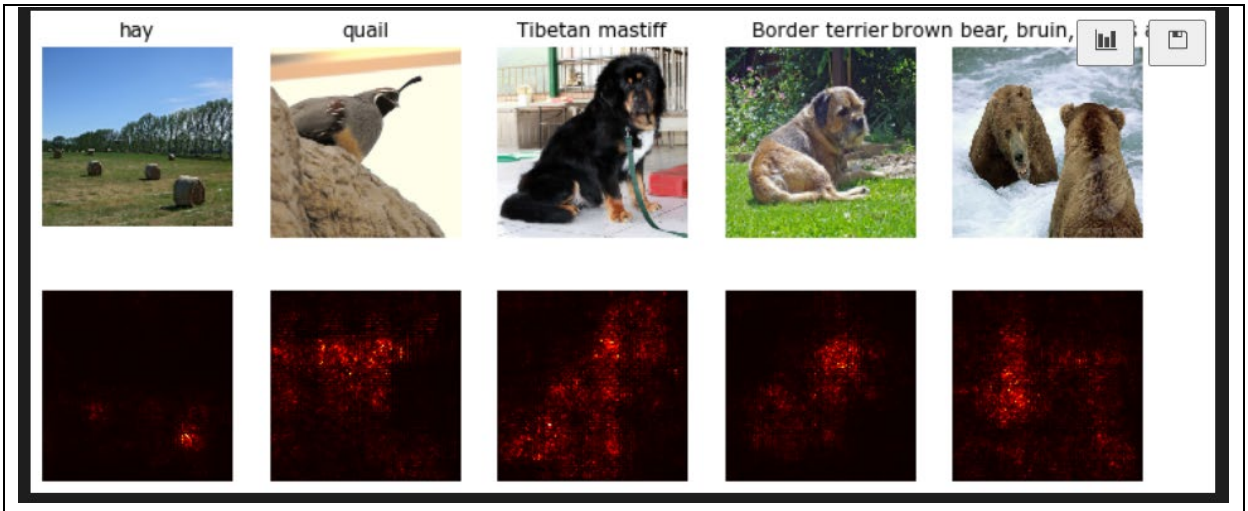
选择正确的分类来反向传播

```
correct_scores = scores.gather(1, y.view(-1, 1)).squeeze()
```

正确分类反向传播，计算通道值

```
correct_scores.backward(torch.ones(y.shape[0]))
saliency, _ = torch.max(torch.abs(X.grad), dim=1)
```

可视化：



2. Fooling Images

使用图像梯度来生成“愚弄图像”。给定图像和目标类，我们可以在图像上执行梯度上升以最大化目标类，当网络将图像分类为目标类时停止

当网络将图像分类为目标类时停止

```
scores = model(X_fooling)
if torch.argmax(scores) == target_y:
    break
```

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

```
target_score = scores[:,target_y]
target_score.backward()
g = X_fooling.grad.data
dX = learning_rate * g / torch.norm(g)
X_fooling.data += dX
X_fooling.grad.zero_()
```

可视化原始图像、愚弄图像以及它们之间的差异

可以看到，用梯度上升的方式改变输入图像就能比较容易欺骗过分类器。某种程度上是因为分类器是数据驱动的，分类边界跟训练数据有关

3. Class Visualization

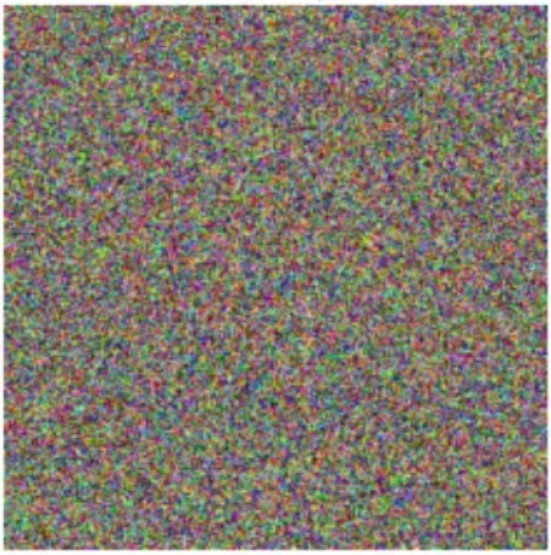
定义随机左右上下抖动函数

直接从随机噪声合成一张图片，再加上一些正则化手段

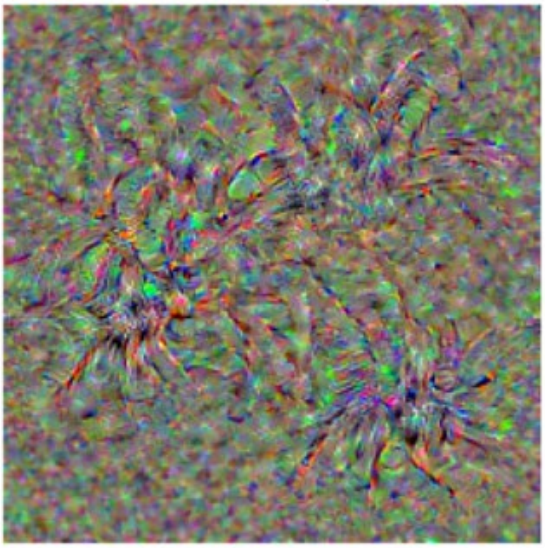
```
scores = model(img)
target_scores = scores[:,target_y]
target_scores.backward()
dX = img.grad.data + 2 *l2_reg * img.data
img.data += learning_rate * dX / torch.norm(dX)
img.grad.zero_()
```

可视化

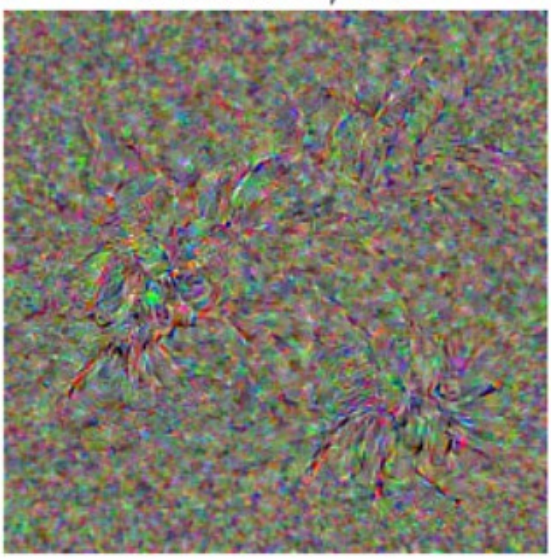
tarantula
Iteration 1 / 100



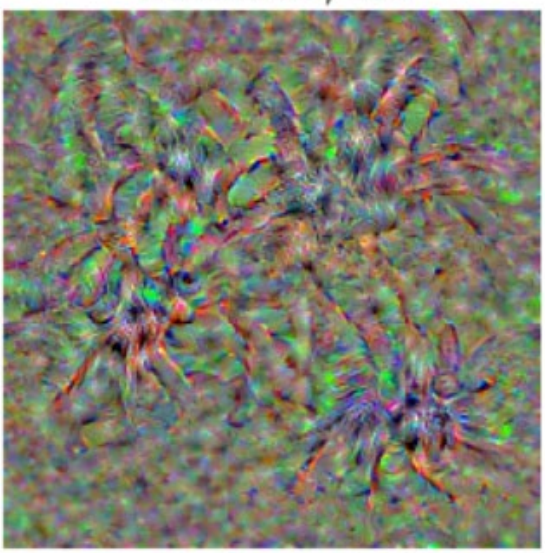
tarantula
Iteration 50 / 100

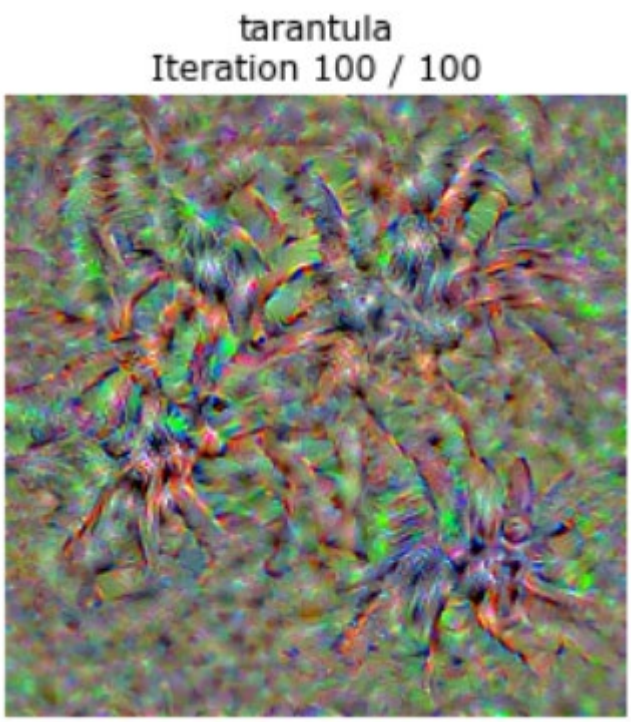


tarantula
Iteration 25 / 100



tarantula
Iteration 75 / 100





结论分析与体会：

学会了网络可视化的实现方式

学会了 3 个有关图像生成的技巧：显著图、欺骗图像、类可视化

加深了对 pytorch 的认识

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：

对于 pytorch 的 backward()函数认识不清楚，使用时报错

网络查询后了解用法