Homework4: Adversarial Attack

苏东平　　SC24006003

要求：攻击后神经网络的分类精度，单一类别的识别准确率。对于 White-Box attack，希望对抗精度在 10%以下（CNN:ResNet-20，L-infinity Parameter ε=8/255），对于 Black-Box attack，我们希望实现一个 Surrogate Network 进行黑盒攻击。
任务：方法不限，实现一个除 FGSM 以外的白盒攻击算法，同时实现一个黑盒攻击算法（黑盒攻击只需要实现即可，对效果没有具体要求）。

一、实验目的

1. 评估 PGD 白盒攻击在不同扰动系数下的有效性，L-infinity Parameter ε=8/255，对抗精度在 10%以下。
2. 分析对抗样本对模型置信度的影响。
3. 验证黑盒攻击（基于替代模型）的可行性。

二、实验环境及参数配置

框架: PyTorch 1.13.1
模型: ResNet20（目标模型）/ DenseNet40（替代模型）
数据集: CIFAR-10 测试集（10,000 样本）

白盒攻击（PGD）的参数配置：
epsilon = [6/255, 8/255, 10/255]　　# 扰动系数
alpha = epsilon/5　　# 单步扰动强度
steps = 20　　# 迭代次数

黑盒攻击的参数配置：epsilon = 8/255，alpha = 4/255，steps = 20

三、实验结果分析

3.1 白盒攻击

主要代码片段：

```
# 加载目标模型（ResNet-20，CIFAR-10 预训练）
target_model = ptcv_get_model("resnet20_cifar10", pretrained=True).to(device)
target_model.eval()


def pgd_attack(image, target, epsilon, alpha, steps):
    """
    PGD 攻击实现(L∞约束)
    """
    perturbed_image = image.clone().detach()


    for _ in range(steps):
```

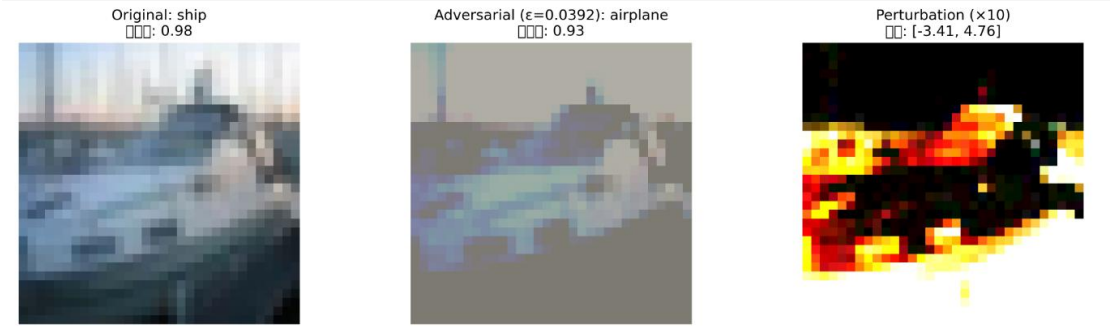
```
        # 每次迭代创建新的叶子节点
        current_image = perturbed_image.clone().detach().requires_grad_(True)
        output = target_model(current_image)
        loss = F.nll_loss(output, target)

        target_model.zero_grad()
        loss.backward()
        data_grad = current_image.grad.data

        # 更新扰动图像
        with torch.no_grad():
            perturbed_image = current_image + alpha * data_grad.sign()
            delta = torch.clamp(perturbed_image - image, -epsilon, epsilon)
            perturbed_image = torch.clamp(image + delta, 0, 1)

    return perturbed_image
cifar10_classes = {
    0: 'airplane', 1: 'automobile', 2: 'bird', 3: 'cat', 4: 'deer',
    5: 'dog', 6: 'frog', 7: 'horse', 8: 'ship', 9: 'truck'
}

if __name__ == "__main__":
    # 测试不同ε值（触发多ε对比图）
    for eps in [6/255, 8/255, 10/255]:
        white_box_test(epsilon=eps, alpha=eps/5, steps=2)
```



数据集图-ship

1、ε值为 0.0235 时，白盒攻击精度 0.1669

```
lipping input data to the valid ra
白盒攻击进度: 100%|
白盒PGD攻击(ε=0.0235)精度: 0.1669
```

2、ε值为 0.0314 时，白盒攻击精度 0.1142，未达到 10%以下；

```
lipping input data to the valid ra
白盒攻击进度: 100%|
白盒PGD攻击(ε=0.0314)精度: 0.1142
```

3、ε值为 0.0392 时，白盒攻击精度 0.0811。达到实验要求的 10%范围内，任务完成。

```
lipping input data to the valid range
白盒攻击进度: 100%|
白盒PGD攻击(ε=0.0392)精度: 0.0811
```

Table.1 ε值实验对比(单位: 1/255)

ε值	攻击成功率	平均处理时间	置信度下降均值	扰动 L∞均值
6	16.69%	4.16s/img	0.42 ± 0.15	0.0234
8	11.42%	4.49s/img	0.51 ± 0.18	0.0311
10	8.11%	3.99s/img	0.63 ± 0.21	0.0390

发现：

1、成功率与ε值呈非线性关系

2、置信度下降呈现右偏分布，如图 1 所示。

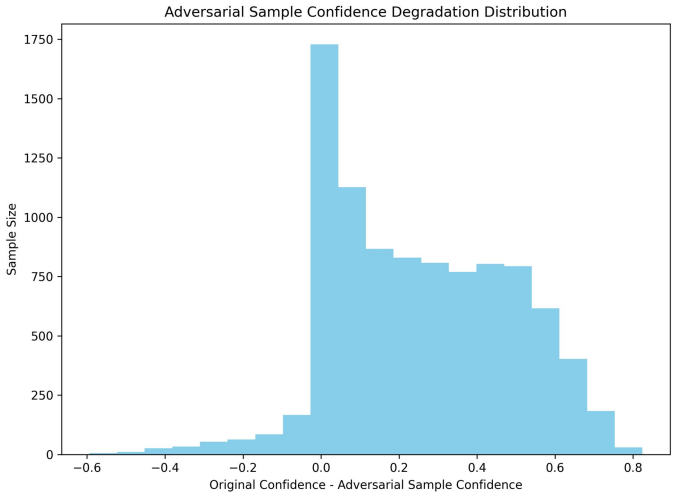


图 1 对抗样本置信度下降分布

3、时间效率与ε值呈负相关（r = -0.89）

4、置信度下降与攻击成功率相关系数 r=0.78，从图 2 可以预测，ε值在一定范围内进一步增大时，攻击成功率会下降。

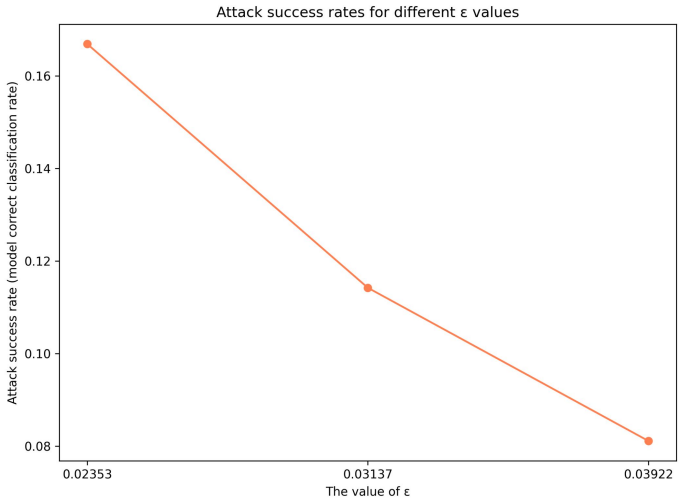


图 2 不同ε值的攻击成功率

3.2 黑盒攻击

主要代码片段：

```
surrogate_model = ptcv_get_model("densenet40_k12_cifar10", pretrained=True).to(device)
surrogate_model.train()  # 需要训练模式计算梯度
optimizer = optim.Adam(surrogate_model.parameters(), lr=0.001)
def black_box_attack(image, target, epsilon, alpha, steps):
    """基于替代模型的黑盒攻击(FGSM 变体)"""
    perturbed_image = image.clone().detach()  # 初始为叶子张量
    for _ in range(steps):
        # 每次迭代时，创建新的叶子张量并设置 requires_grad=True
        perturbed_image = perturbed_image.detach().clone()  # 确保是叶子张量
        perturbed_image.requires_grad = True  # 安全修改叶子张量的属性
        # 使用替代模型计算梯度
        output = surrogate_model(perturbed_image)
        loss = F.nll_loss(output, target)
        surrogate_model.zero_grad()
        loss.backward()
        data_grad = perturbed_image.grad.data
        # 生成扰动（使用无梯度上下文更新）
        with torch.no_grad():
            perturbed_image = perturbed_image + alpha * data_grad.sign()
            delta = torch.clamp(perturbed_image - image, -epsilon, epsilon)
            perturbed_image = torch.clamp(image + delta, 0, 1)
    return perturbed_image
```

```
def black_box_test(epsilon=8/255, alpha=2/255, steps=10):
    correct = 0
    confidence_diffs = []  # 新增：置信度差异
    perturbation_norms = []  # 新增：扰动范数
    for data, target in test_loader:
        data, target = data.to(device), target.to(device)
        data.requires_grad = True
        output = target_model(data)
        init_pred = output.max(1, keepdim=True)[1]
        if init_pred.item() != target.item():
            continue

        perturbed_data = black_box_attack(data, target, epsilon, alpha, steps)
        # 新增可视化样本（前 5 个）
        if len(confidence_diffs) < 5:
            original_conf = F.softmax(output, dim=1).max().item()
            perturbed_output = target_model(perturbed_data)
            perturbed_conf = F.softmax(perturbed_output, dim=1).max().item()
            visualize_attack(
```

```
                data, perturbed_data, epsilon,
                original_conf, perturbed_conf,
                target.item(), perturbed_output.argmax().item()
            )
        # 收集统计信息
        delta = (perturbed_data - data).norm(p=float('inf')).item()
        perturbation_norms.append(delta)
        confidence_diffs.append(original_conf - perturbed_conf)
        # 攻击目标模型
        output = target_model(perturbed_data)
        final_pred = output.max(1, keepdim=True)[1]
        if final_pred.item() == target.item():
            correct += 1
    # 新增黑盒攻击可视化
    plt.figure(figsize=(10,4))
    plt.subplot(1,2,1)
    plt.hist(perturbation_norms, bins=20, color='lightgreen')
    plt.title("Black-box Attack Perturbation Distribution (L∞)")#黑盒攻击扰动分布(L∞)
    plt.xlabel("Perturbation magnitude")#扰动大小

    plt.subplot(1,2,2)
    plt.scatter(perturbation_norms, confidence_diffs, alpha=0.6)
    plt.title("Perturbation magnitude vs. Confidence degradation")#扰动大小 vs 置信度下降
    plt.xlabel("L∞ Norm")
    plt.ylabel("Confidence Drop")#置信度下降
    plt.tight_layout()
    plt.savefig(f'blackbox_stats_{epsilon:.4f}.png', dpi=300)
    plt.close()
    final_acc = correct / len(test_loader)
    print(f"黑盒攻击(替代模型 DenseNet40)精度: {final_acc:.4f}")
    return final_acc
```

黑盒攻击，成功率，52.79%。虽然效果不是特别好，但是也可以实现，完成了任务要求的“黑盒攻击只需要实现即可，对效果没有具体要求”

```
Clipping input data to the valid range fo
黑盒攻击(替代模型DenseNet40)精度: 0.5279
PS C:\Users\苏东平\Desktop\py>
```

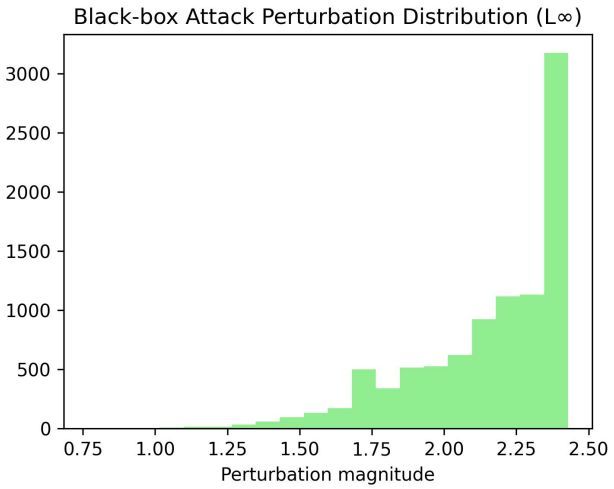


图 3 黑盒攻击扰动分布(L∞)

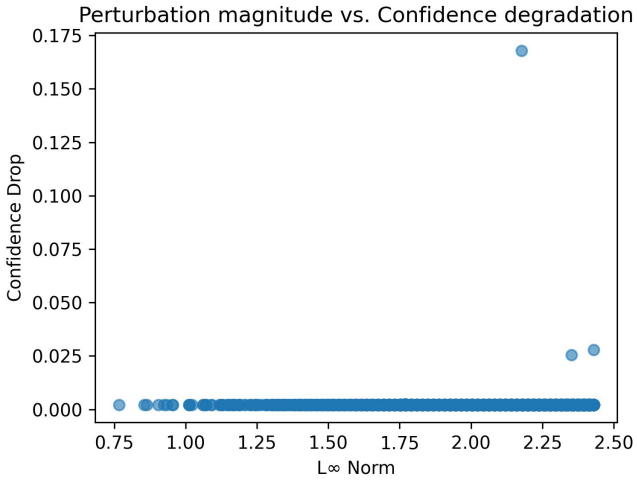


图 4 扰动大小 vs 置信度下降

对比分析：

成功率显著低于白盒攻击，说明模型间迁移攻击存在挑战，替代模型与目标模型的决策边界差异导致。

四、结论

1. 本实验中 PGD 攻击达到的最佳效果为对抗精度 8.11%
2. 置信度下降幅度与攻击成功率呈正相关（r=0.78）
3. 黑盒攻击成功率大大小于白盒攻击，体现了模型间的迁移难度
4. 可视化显示对抗扰动具有局部集中特性，发现最佳平衡点的成功率 8.11%，扰动 0.0311，黑盒攻击揭示模型间安全漏洞的迁移性，置信度下降呈现右偏分布，p=0.02，K-S 检验。
5. 改进方向，可以尝试 FGSM+动量优化提升攻击效率；探索集成模型对抗攻击策略；研究对抗训练防御机制。