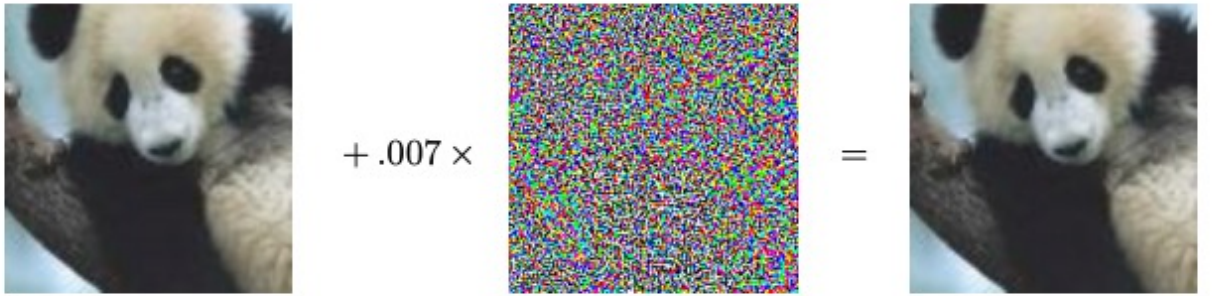
# 딥러닝 모델 속이기 – 적대적 공격 & 텐서보드

# 오늘의 학습내용

- 딥러닝 모델 속이기
- FGSM(Fast Gradient Signed Method)
- PyTorch를 이용해서 FGSM 구현하기
- 텐서보드(TensorBoard)로 시각화하기

# 딥러닝 모델을 속이기 위한 적대적 공격(Adversarial Attack)

- FGSM(Fast Gradient Signed Method) 방법을 이용해서 딥러닝 모델을 속일 수 있습니다.
- panda : 판다
- nematode : 선충
- gibbon : 긴팔원숭이



## 적대적 공격의 종류

- 적대적 공격(Adversarial Attack)의 목적은 상황에 따라 다를 수 있지만 공통적인 목표는 입력데이터에 작은 변화를 주어 의도적으로 잘못된 예측 결과를 만들어내게 하는 것입니다.
- 화이트 박스(whitebox) 공격 모델은 공격자가 모델에 대해 아키텍처, 입력, 출력, 가중치를 포함한 모든 것을 알고 있고 접근할 수 있다고 가정합니다.
- 블랙박스(blackbox) 공격 모델은 공격자가 모델의 입력과 출력에 대해서만 접근 가능하고 모델의 가중치와 아키텍처에 관한 내용은 모른다고 가정합니다.
- FSGM은 화이트박스 공격 모델 중 한가지 방법론입니다.

# FGSM(Fast Gradient Signed Method)

- FGSM(Fast Gradient Signed Method) : FSGM은 신경망의 기울기(Gradient)를 이용해서 적대적 샘플을 생성하는 기법입니다.
- 입력 이미지에 대한 손실 함수의 기울기(Gradient)를 계산하여 손실이 최대화되는 이미지를 생성합니다.
- 이렇게 딥러닝 모델을 속이기 위한 이미지를 적대적 이미지(Adversarial Image)라고 합니다.
- 적대적 이미지를 생성하는 수식은 다음과 같습니다.

$$adv_x = x + \epsilon * \text{sign}(\nabla_x J(\theta, x, y))$$

$adv_x$ : 적대적 이미지

$x$ : 원본 입력 이미지

$y$ : 원본 입력 레이블

$\epsilon$ : 왜곡의 양을 조절할 수 있는 계수(coefficient)

$\theta$ : 모델의 파라미터

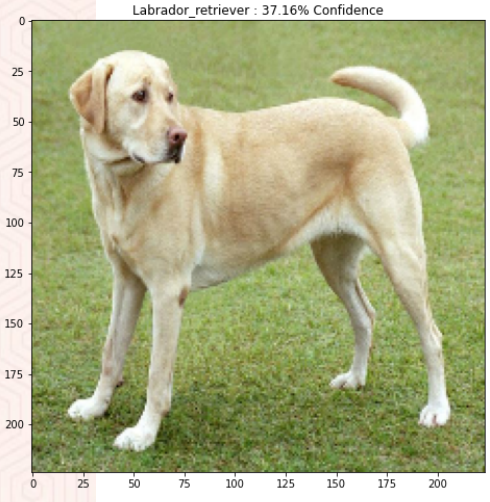
$J$ : 손실 함수

# FGSM(Fast Gradient Signed Method)

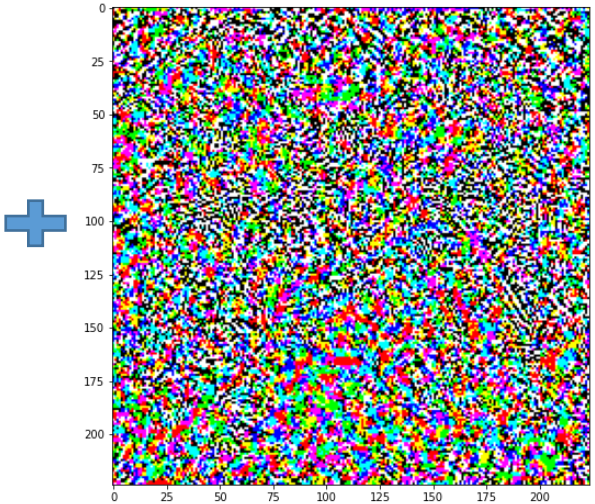
- FGSM의 목적은 손실을 최대화하는 이미지를 생성하는 것입니다.
- 따라서 적대적 이미지는 각 픽셀의 손실에 대한 기여도를 기울기(Gradient)로 계산한 후 손실에 대한 기여도를 원본 이미지에 더함으로써 딥러닝 모델이 잘못된 예측을 수행하는 적대적 이미지를 만들 수 있습니다.

# 적대적 이미지 생성

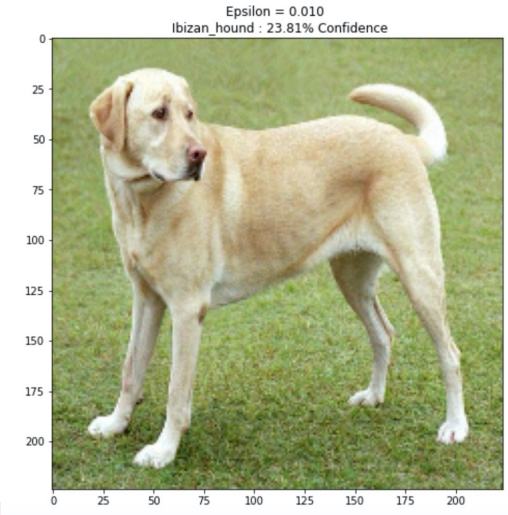
❖ 원본 이미지에 기울기(Gradient)를 더해서 적대적 이미지를 생성할 수 있습니다.




Input



Gradient




Adversarial Image

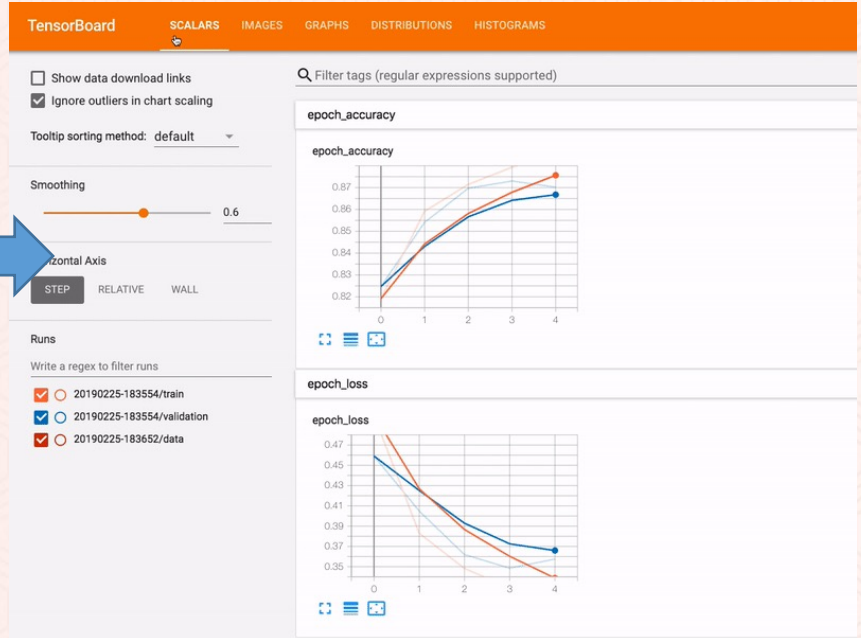
# 파이토치(PyTorch)를 이용한 적대적 공격 구현

- 파이토치(PyTorch)를 이용해서 적대적 공격 알고리즘을 구현해봅시다.
- 7강_딥러닝_모델_속이기_적대적_공격.ipynb
- https://colab.research.google.com/drive/1wAuhCqPZIEE0pH3wkwrhr9zLD04h-xfV?usp=sharing

# 학습과정 Visualization의 필요성과 TensorBoard

- ❖ 터미널 로그 등을 이용해서 학습 과정을 모니터링 할 경우, 한눈에 학습 과정의 문제점을 파악하기 쉽지 않습니다.
- ❖ 따라서 PyTorch에서는 학습과정 시각화를 위해 TensorBoard라는 기능을 제공합니다.

```
WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versions/3.7/lib/python
3.7/site-packages/tensorflow/python/util/deprecation.py:574: calling map_fn_v2 (from
 tensorflow.python.ops.map_fn) with dtype is deprecated and will be removed in a fut
ure version.
Instructions for updating:
Use fn_output_signature instead
반복(Epoch): 100, 트레이닝 데이터 정확도: 0.920000
반복(Epoch): 200, 트레이닝 데이터 정확도: 0.980000
반복(Epoch): 300, 트레이닝 데이터 정확도: 0.920000
반복(Epoch): 400, 트레이닝 데이터 정확도: 0.940000
반복(Epoch): 500, 트레이닝 데이터 정확도: 0.980000
반복(Epoch): 600, 트레이닝 데이터 정확도: 0.960000
```



# 텐서보드 시각화하기

- 파이토치(PyTorch)를 이용해서 학습과정을 시각화하는 과정을 실습해봅시다.
- 7강_텐서보드로_시각화하기.ipynb
- https://colab.research.google.com/drive/1KUjhRxsRyIZHIdLA6nCvR6VsSL1n-VVu?usp=sharing