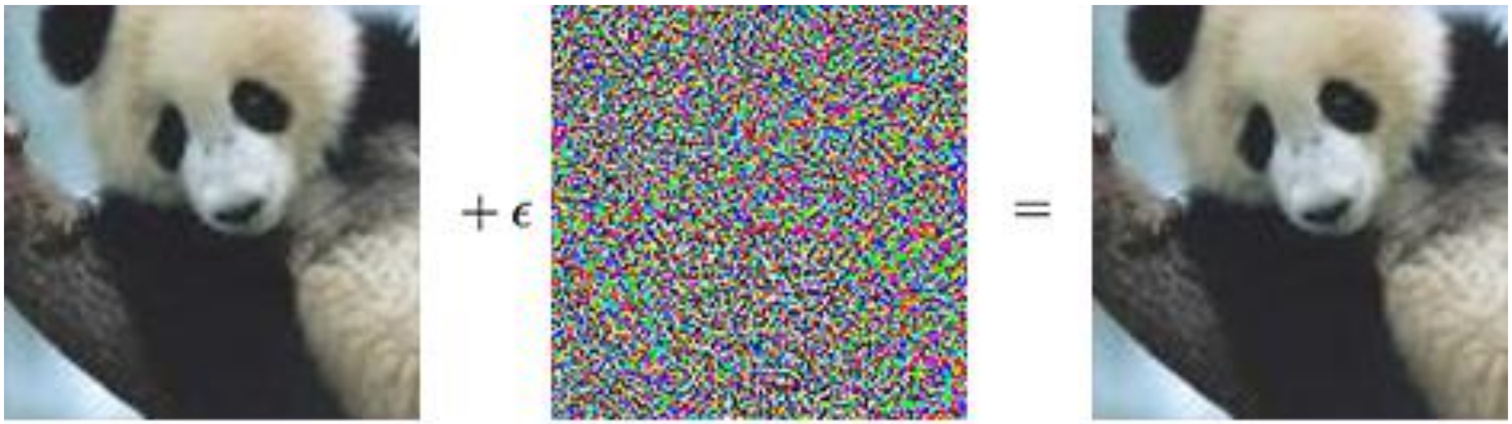
Adversarial Examples

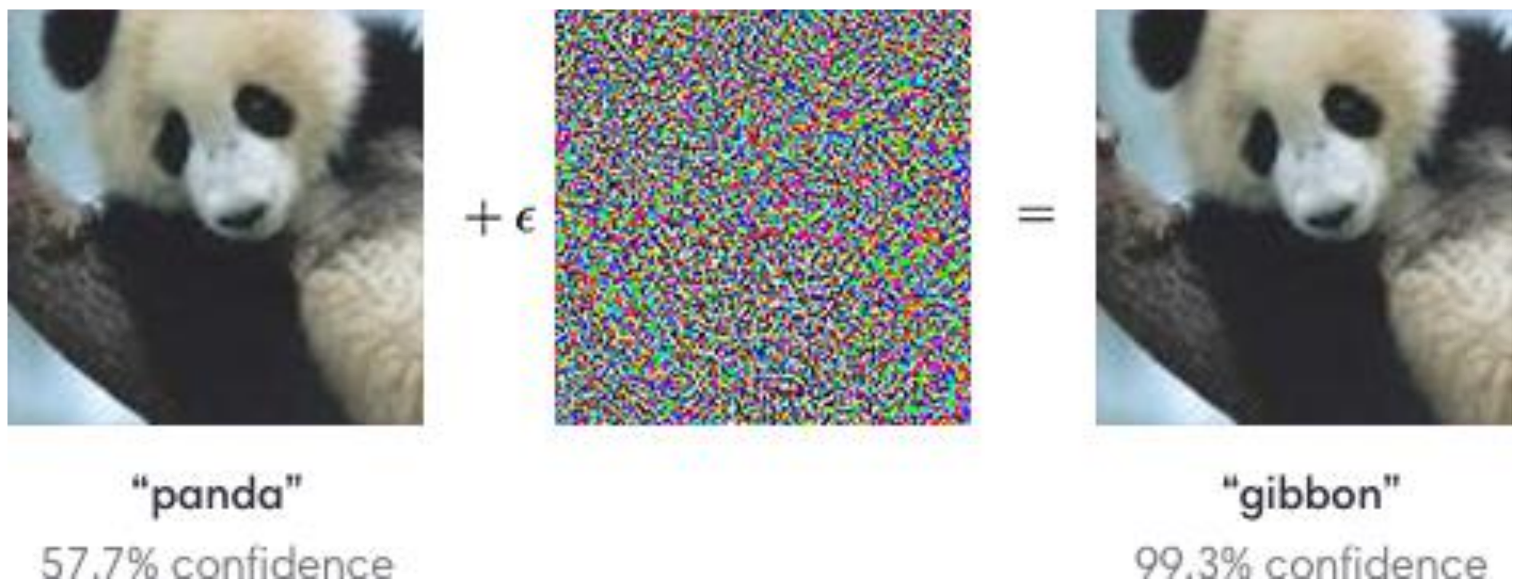


Question: What are these pictures of?

CSC411/2515: Machine Learning and Data Mining, Winter 2018

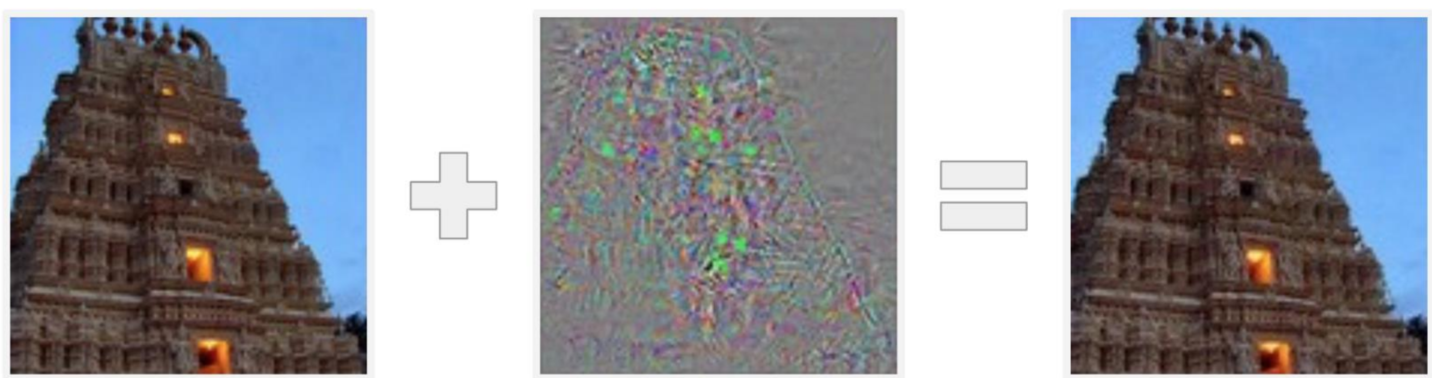
Michael Guerzhoy and Lisa Zhang

Adversarial Examples



The adversary

- Suppose your friend built a neural network image classifier $f_\theta(x)$, and you want to break it.
- Idea: find a perturbation direction ϵ to an image x that was correctly classified, so that $f_\theta(x + \epsilon)$ is wrongly classified



Original image Temple (97%)

Perturbations

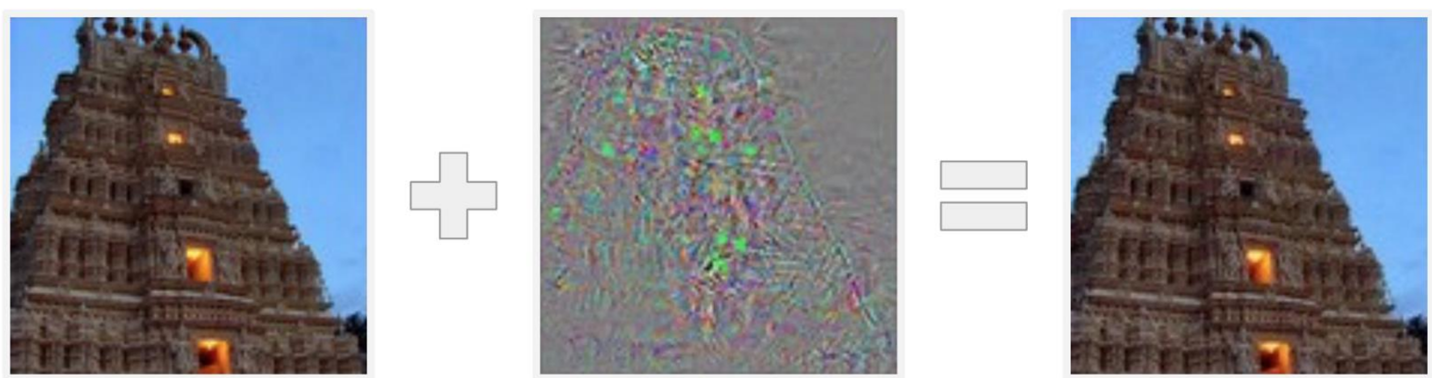
Adversarial example Ostrich (98%)

The adversary

- What ϵ should we use? If we know the architecture and weights of f_θ, then we can perform gradient descent on ϵ to optimize:

$$\arg\min_{\epsilon} \log p(f_\theta(x+\epsilon) = correct\ class)$$

- Then rescale ϵ to be small (imperceptible)



Original image Temple (97%) | **Perturbations** | **Adversarial example** Ostrich (98%)

Adversarial Attacks

- Non-targeted Adversarial Attack
  $$\arg\min_{\epsilon} \log p(f_\theta(x+\epsilon) = correct\ class)$$
- Targeted Adversarial Attack
  $$\arg\max_{\epsilon} \log p(f_\theta(x+\epsilon) = specific\ class)$$
- White-box Adversarial Attack
  - Can access the network f_θ
  - Can compute gradients of f_θ as above
- Black-box Adversarial Attack
  - Cannot compute gradients of f_θ

Black-box Attacks

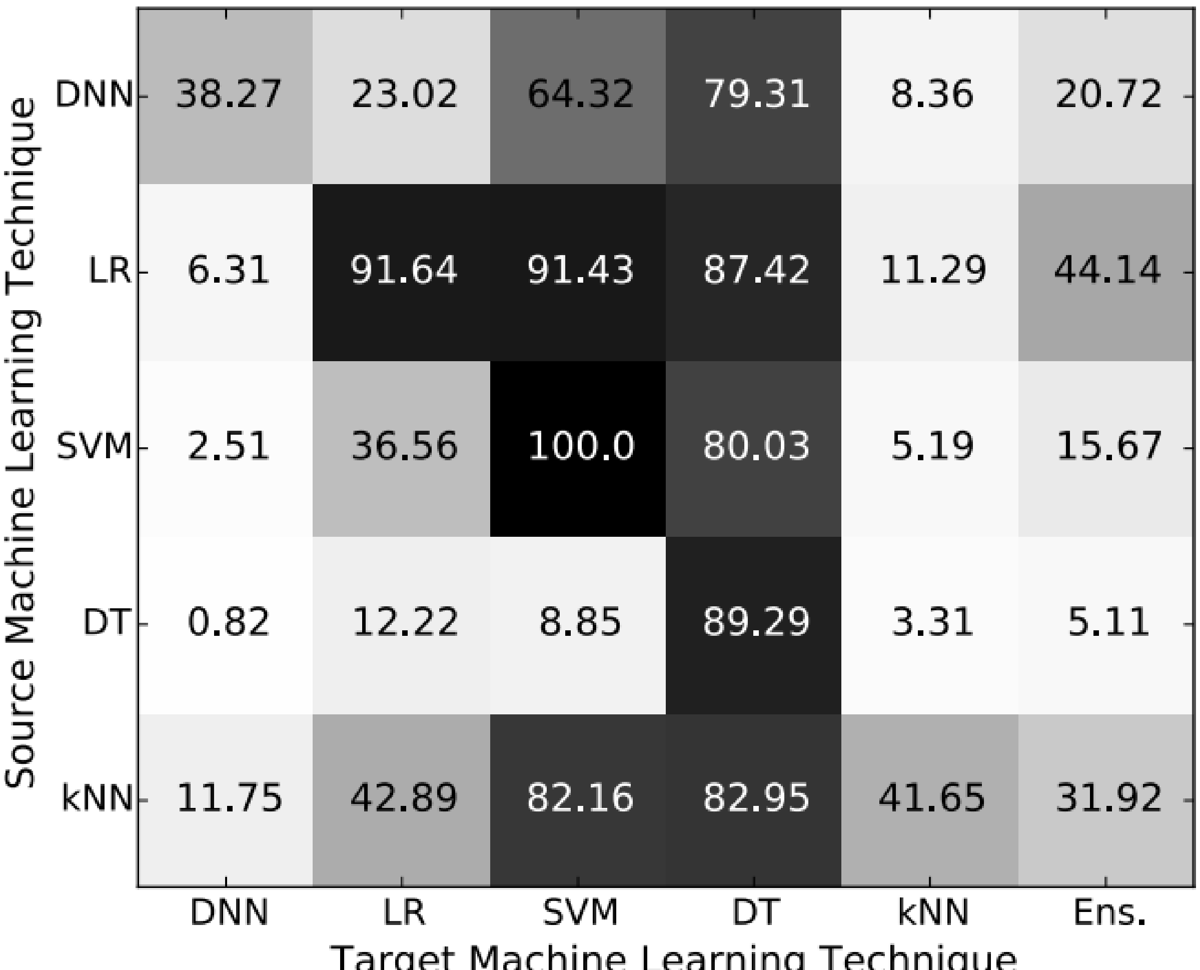
- Target model with unknown weights, machine learning algorithm, training set; maybe non-differentiable
- Substitute model mimicking target model with known, differentiable function
- Generate adversarial example
- Moral: adversarial attacks often **transfer**!

Transferable Attacks

- "Adversarial examples that affect one model often affect another model, even if the two models have different architectures or were trained on different training sets, so long as both models were trained to perform the same task"

Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples (McDaniel & Goodfellow, 2016)

Transfers cross-technique



Transferability matrix: $cell(i, j)$ is the percentage of adversarial samples crafted to mislead a classifier learned using machine learning technique *i* that are misclassified by a classifier trained with technique *j*.

Failed Defenses

- Generative pre-training
- Adding noise at test time
- Ensembles
- Weight decay
- Adding noise at training time
- Adding adversarial noise at training time
- Dropout
- …

Adversarial Attacks

- Printed Object: https://openai-public.s3-us-west-2.amazonaws.com/blog/2017-07/robust-adversarial-examples/iphone.mp4
- 3D Printed Objects https://www.youtube.com/watch?v=piYnd_wYlT8