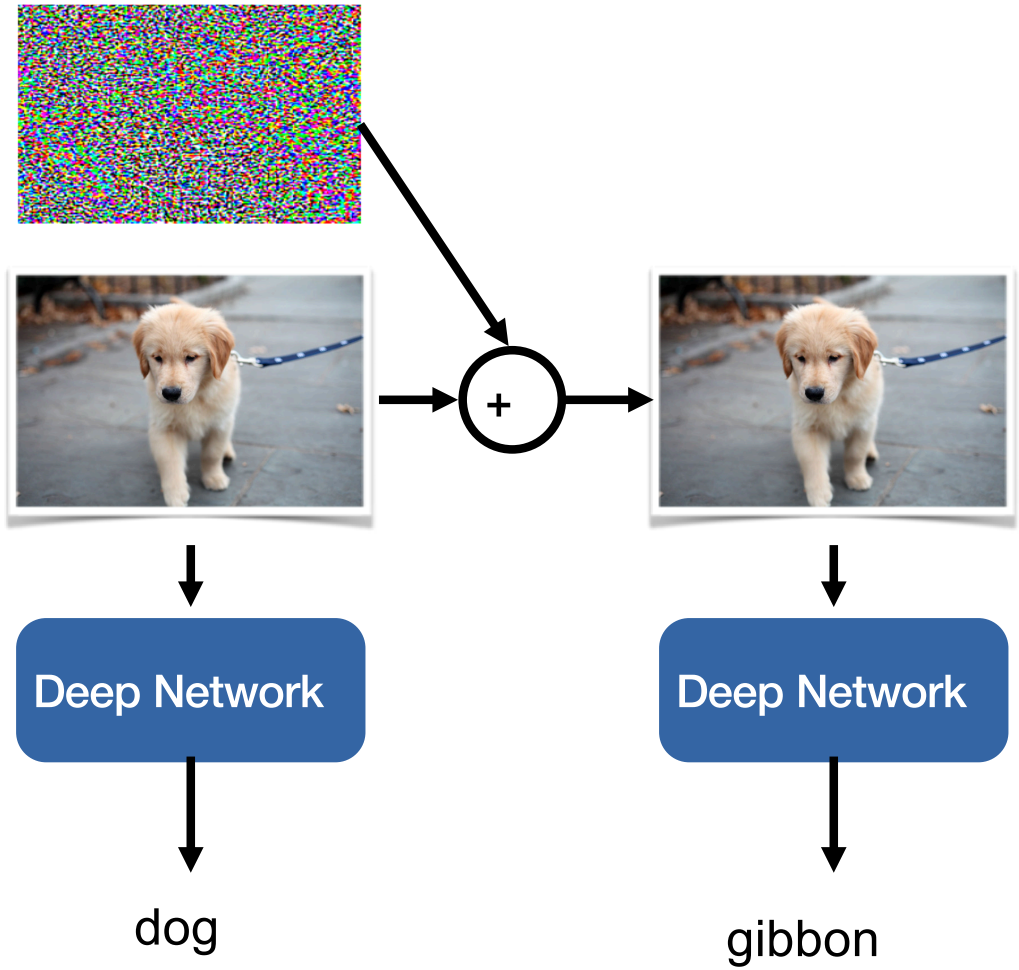
Fooling deep networks

Adversarial perturbations

Fooling a deep network

- Image + noise = wrong prediction



- Intriguing properties of neural networks, Szegedy et al., arXiv 2013
- Explaining and Harnessing Adversarial Examples , Goodfellow et al., ICLR 2015

Fast gradient sign

Assume networks are locally linear

For input $\mathbf{x}$

Find ϵ

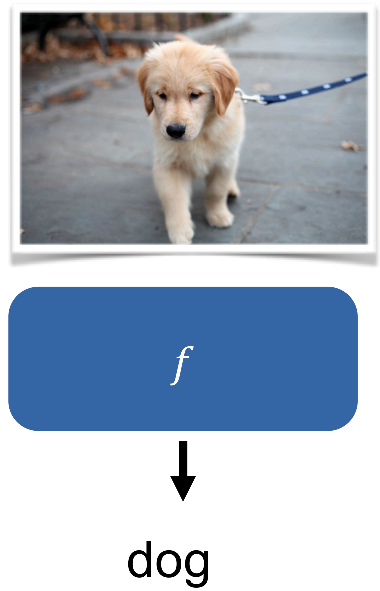
Such that $f(\mathbf{x} + \epsilon) \neq f(\mathbf{x})$

i.e. the networks predicts something different

Has to put some constraints on perturbation

Optimal attack with $\| \epsilon \|_\infty \leq c$ if function is linear

- $\epsilon = \text{sign}\big(\nabla_{\mathbf{x}} \ell(f(\mathbf{x}), y)\big)$



Projected gradient descent

Networks are not linear

Optimize for the attack using gradient descent

Assume networks are locally linear

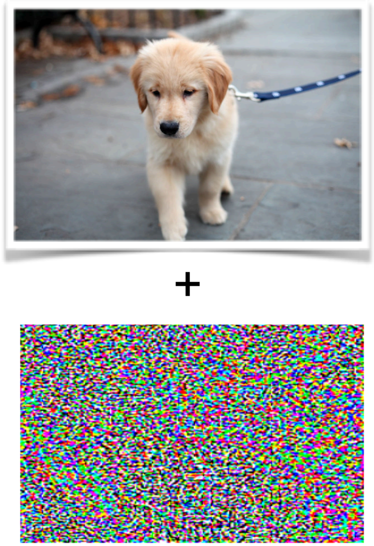
For input $\mathbf{x}$

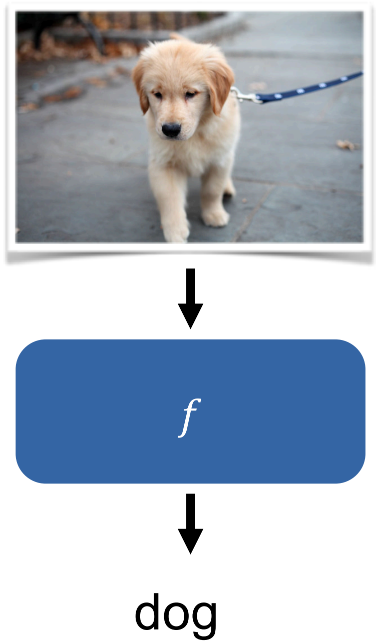
Find ϵ

Such that $f(\mathbf{x} + \epsilon) \neq f(\mathbf{x})$

(i.e. predicts different class)

- $\text{maximize}_{\epsilon} \ell(f(\mathbf{x} + \epsilon), y)$
- s.t. $\| \varepsilon \|_{\infty} < c$





Towards Deep Learning Models Resistant to Adversarial Attacks, Madry et al., ICLR 2018

Global adversarial attacks

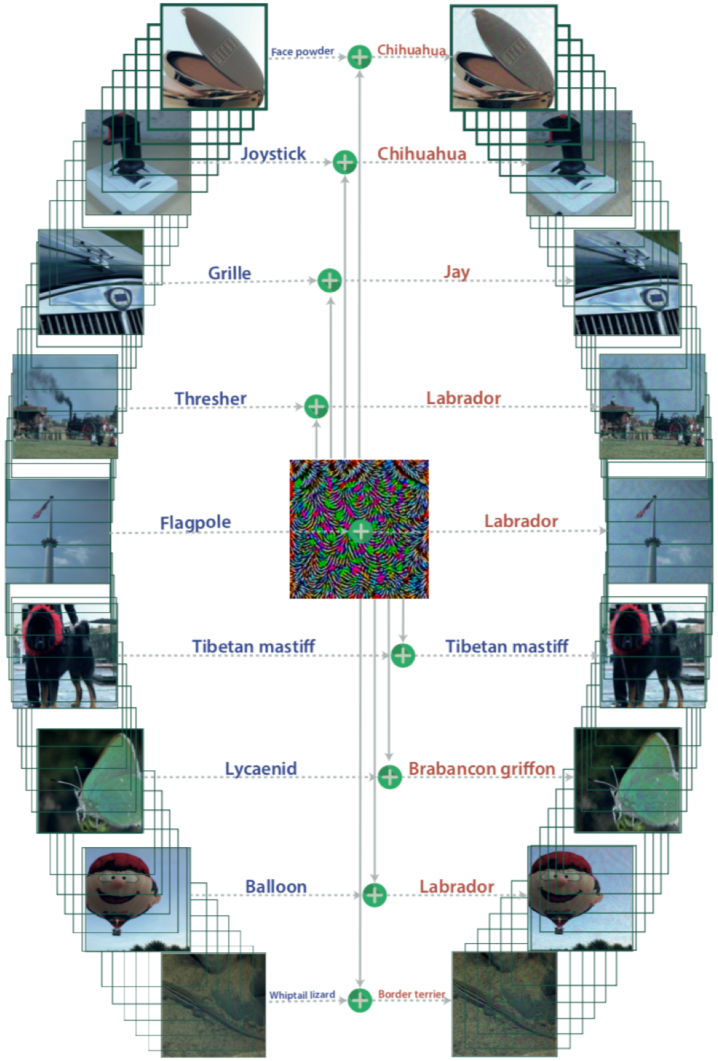
Attacks all possible inputs at once

- PGD on entire dataset

Attack not input specific

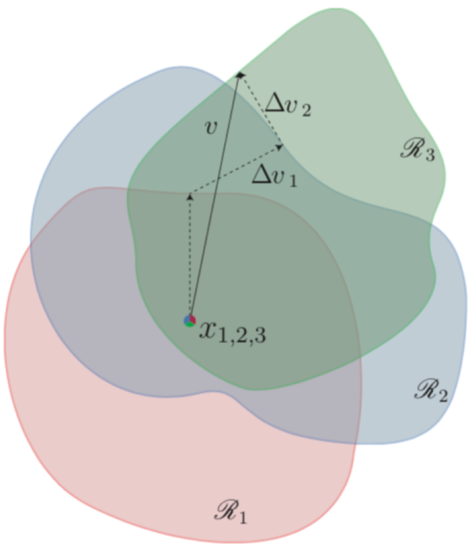
Attack transfers between architectures

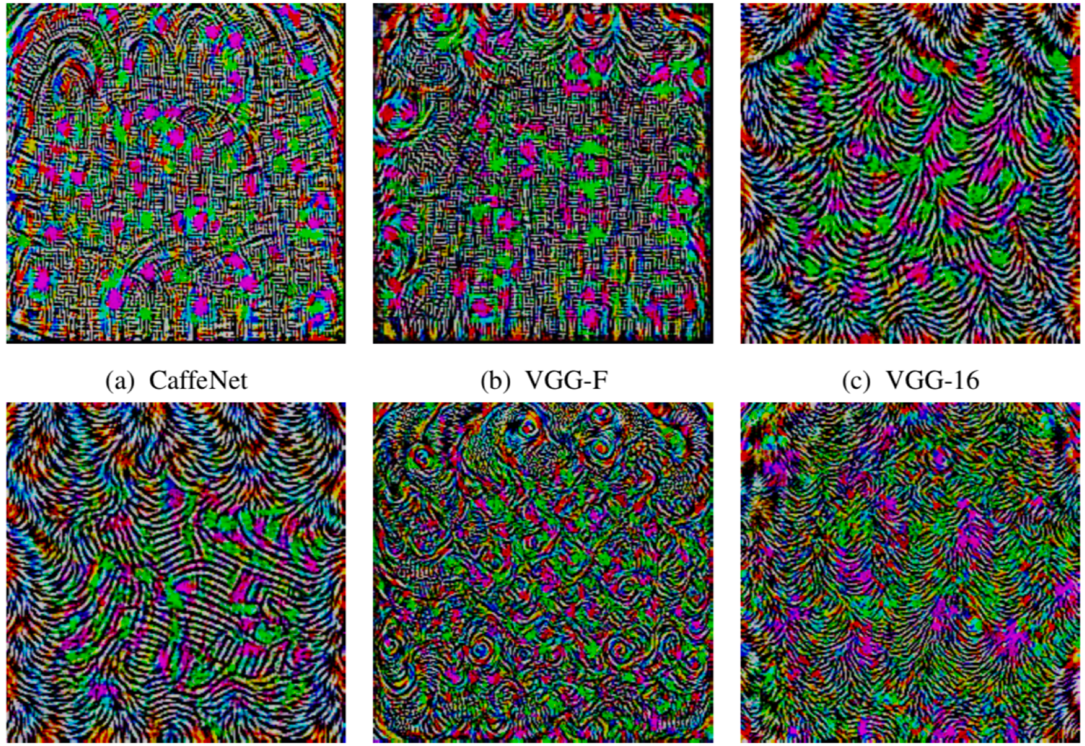
- Dataset specific?



Universal adversarial perturbations, Moosavi-Dezfooli et al., CVPR 2017

Universal Perturbations





(a) CaffeNet (b) VGG-F (c) VGG-16

(d) VGG-19 (e) GoogLeNet (f) ResNet-152

	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	**93.7%**	71.8%	48.4%	42.1%	42.1%	47.4 %
CaffeNet	74.0%	**93.3%**	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	**78.9%**	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	**78.3%**	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	**77.8%**	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	**84.0%**

Universal adversarial perturbations, Moosavi-Dezfooli et al., CVPR 2017

Defense

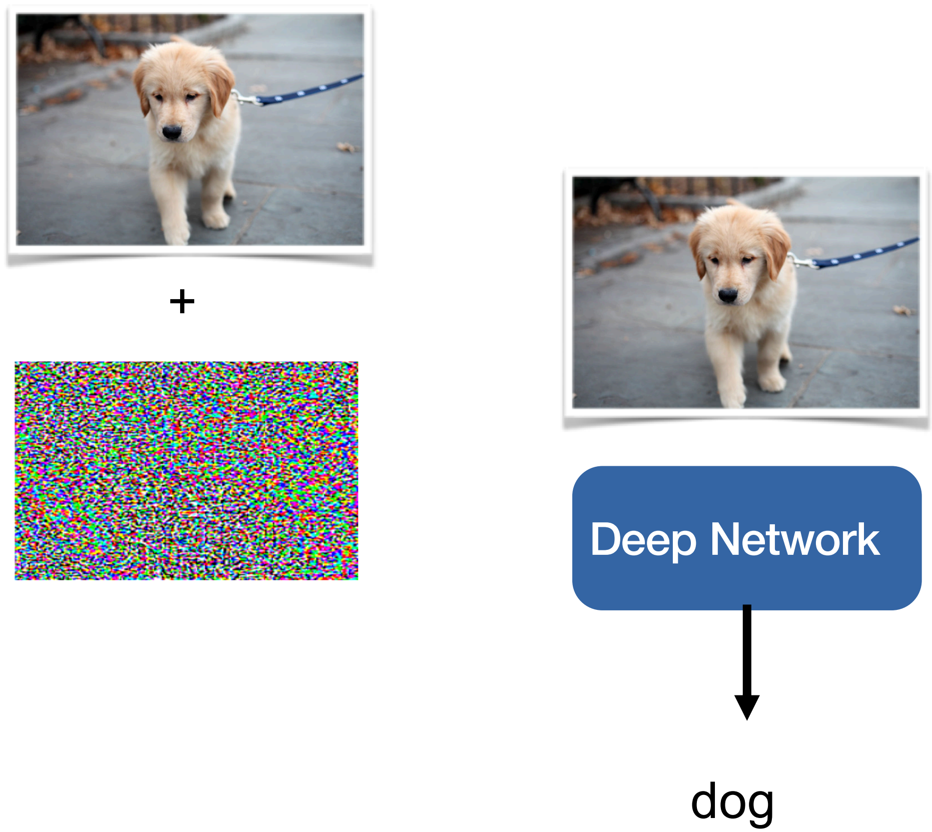
Show network attacked
images during training

for each iteration

- Construct mini-batch
- Perturb mini-batch
- Forward / backward
  - Original
  - Perturbed

Attacking “robust models”
Still works

- just harder

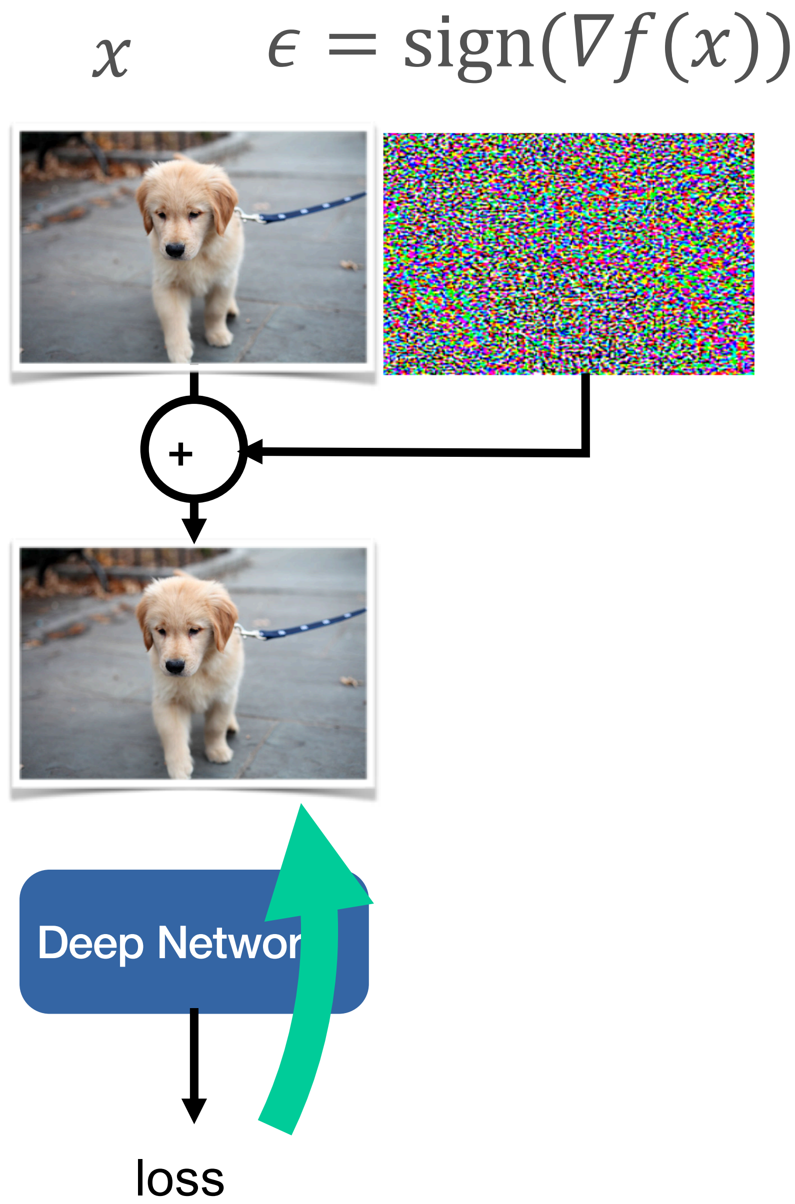


White box attacks

Attacker has access to model and gradients

- Fast gradient sign
- Projected gradient descent

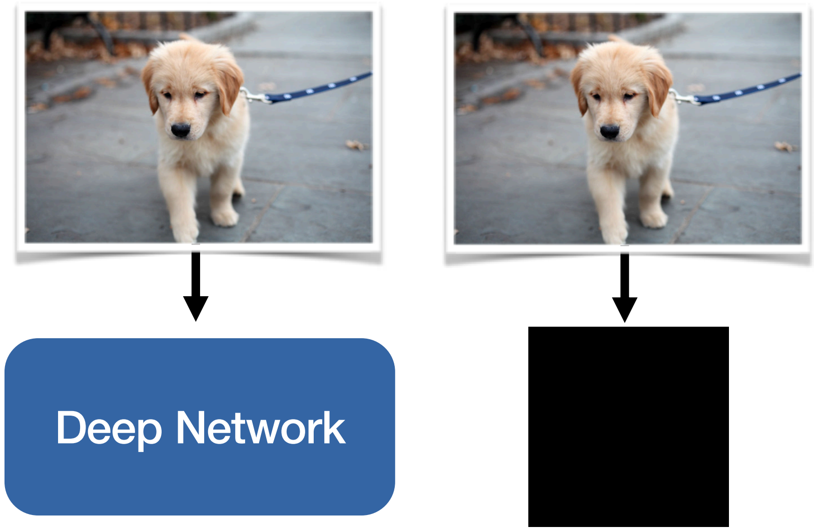
Can we defend against attacks if we do not allow backprop?



Back box attacks

Train network to imitate
black box network

- Attack new network
  - Attack black box
- If not successful
  - repeat



Practical Black-Box Attacks against Machine Learning, Papernot et al., arXiv 2016

What attacks should we worry about?

Random noise attacks don't matter (yet)

- Doing the wrong thing for real images does

Try a validation set

- No guarantees
- Might overfit to validation / test set
- Failures can be rare, but fatal