Motivation:

- Origins
- Attack & defense models
- Constructing attacks
- Constructing defenses
- Connections to RL
- Data poisoning

Today: A simple attack.
Typically, f learned from training data

\[ S = \{(x_1,y_1), \ldots, (x_n,y_n)\} \]

\[ f = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i)) \]

Standard assumptions - ① there is some "natural" distribution
② Training data sampled i.i.d.

Easy way to fool/attack an ML system: just provide an "unnatural" data point as input,

ex: Imagenet (obscure)
Imperceptible perturbations.

Examples of imperceptible perturbations:

\[
\begin{align*}
\text{"panda"} & \quad + \epsilon \\
\text{57.7\% confidence} & \quad = \\
\text{"gibbon"} & \quad 99.3\% confidence
\end{align*}
\]

(Imagination: Natural images have pockets of adversarial examples.)

Q. What about "natural" examples?

A. 


https://www.labsix.org/physical-objects-that-fool-neural-nets/

**Conclusion**

- State of the art ML systems are particularly easy to fool ("adversarial")
- Even in benign (natural) settings, suggests fundamental limitations in our understanding of construction of ML systems.

**MODELS** for adversarial ML.

- **Test-time attacks**
  \[ x \rightarrow f \rightarrow y \]
  \( \{(x_i, y_i)\} \)
  \[ \text{Train-time attacks} \]
  \[ \text{Data poisoning} \]

**Setup**

- Full model \([\text{white box}]\)
- Architecture \([\text{grey box}]\)

What does the attacker know?
Training algorithm
- Query access (Black box)

(II) What can the attacker do?
- Perturb all features of the input?
- Only a few?
- etc.

(III) Given (I) & (II), design a defense.
- Typically, retrain \( f \).

Simple (linear) attack (white box)

Single neuron / perceptron / SVM

\[
\overline{x} = x + \eta \rightarrow \text{quantization/roundoff}
\]

\[
X \rightarrow x \rightarrow y
\]

\[
y = \text{sign}(w^T x)
\]
\[ \| \eta \|_\infty \leq \varepsilon \cdot (\text{Attacker's budget}) \]

\[ \langle \omega, \bar{x} \rangle = \langle \omega, x \rangle + \langle \omega, \eta \rangle \]

\[ \max_{\eta} \text{ when } \eta = \text{sgn}(x) \]

\[ \varepsilon \cdot \| w \|_1 \leq \varepsilon \cdot \sqrt{n} \| w \|_2. \]

Deep networks: Not linear, but can be linearized locally.

\[ \eta = \text{sgn} \left( \sum_{j} \log \left( \frac{W_j(x,y)}{1 - W_j(x,y)} \right) \right). \]

\[ \bar{x} = x + \varepsilon \text{sgn} \left( \sum_{j} W_j(x,y) \right). \]

"Fast gradient sign method" (FGSM)

[Goodfellow - Shlens - Szegedy, 2015].