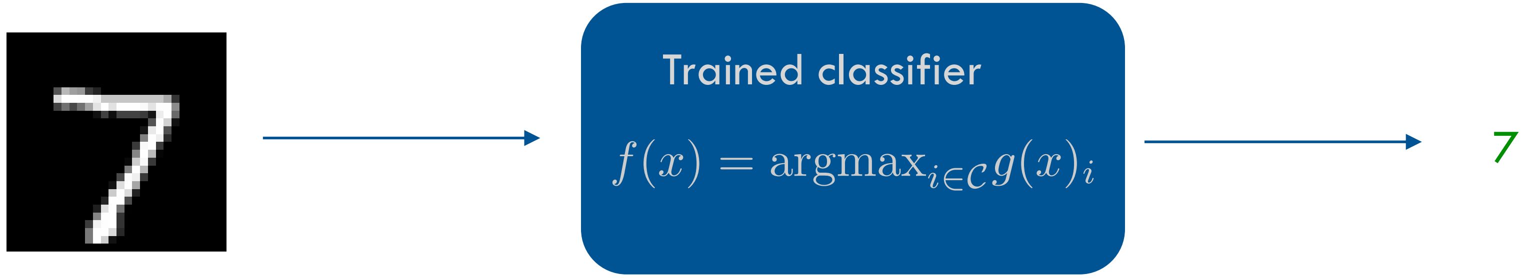
Generating Adversarial Examples using Gradient Descent

Arjun Bhagoji

University of Chicago

Standard Classification



- Classifier chooses class with highest 'score'

- Examples of possible classifiers:
    - Logistic Regression
    - Support Vector Machines
    - Neural Networks

- **Q**: Can we modify this input so the trained classifier provides the wrong output?

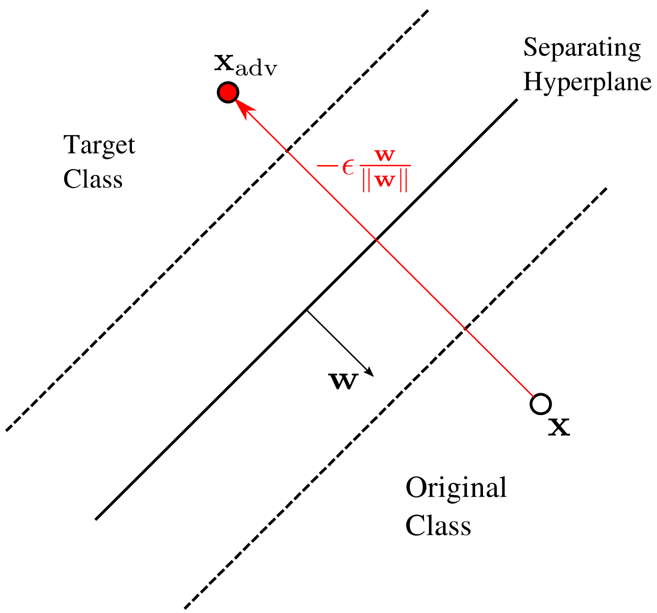
Adversarial Examples

Find the smallest perturbation δ such that:

$$f(x+\delta) \neq f(x)$$

$x+\delta$ → **Trained classifier** $f(x) = \text{argmax}_{i \in \mathcal{C}}\, g(x)_i$ → 3

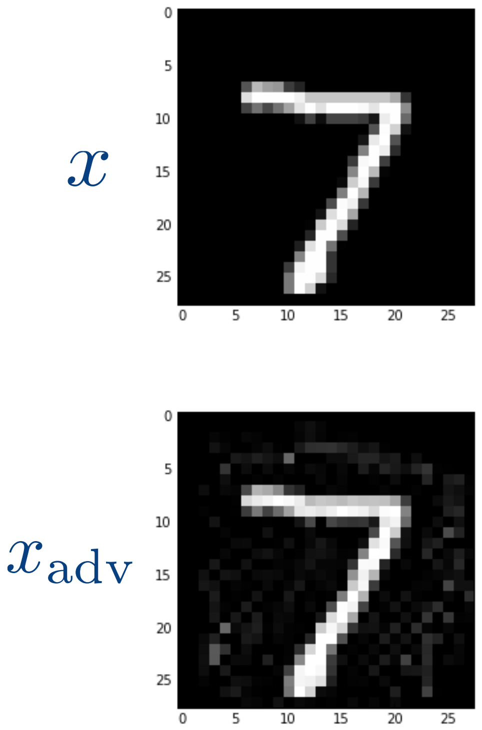
Adv. Examples for Linear Classifiers



$$x_{\text{adv}} = x - \epsilon \frac{\mathbf{w}}{\|\mathbf{w}\|_2}$$

$$\epsilon \in [0, \infty)$$

Attack on two-class Linear Classifier

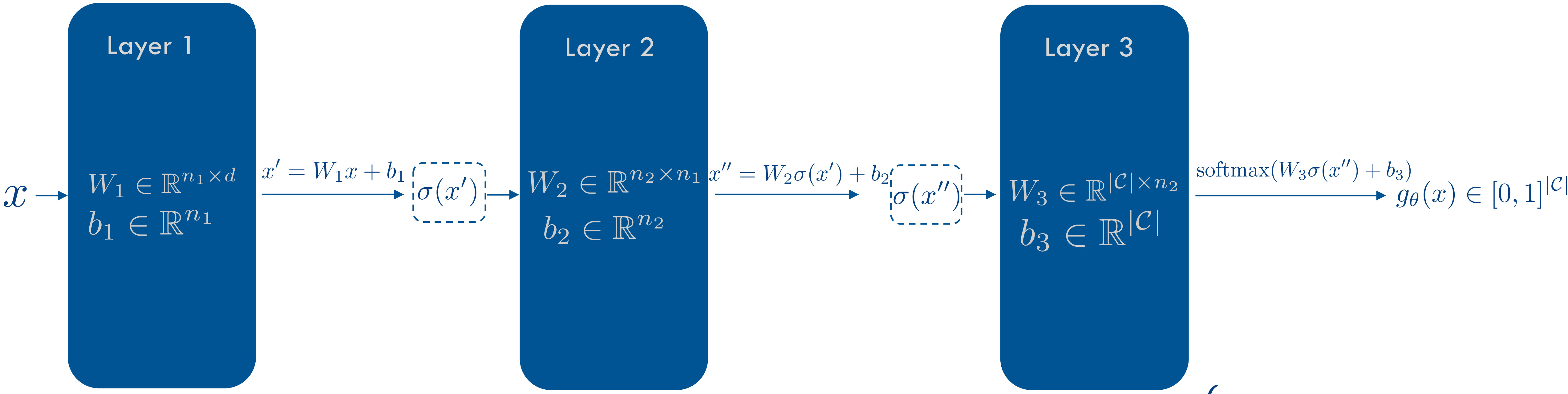


Adversarial image with $\epsilon = 2.0$.

Leads to 100% misclassification on test set.

ϵ is the *amount of perturbation* or *adversarial budget*

Overview: Neural Networks



$\sigma(\cdot)$ is an element-wise non-linear function, e.g. Rectified Linear Unit

$$\sigma(a) = \begin{cases} 0 \ , a < 0, \\ a \ , a \geq 0 \end{cases}$$

$$\mathrm{softmax}(z) = \frac{e^{z_i}}{\sum_i e^{z_i}}$$

$$J(g_{\{W_1, b_1, W_2, b_2, W_3, b_3\}}) = J(g_\theta) = \mathrm{loss}(g_\theta(x), y)$$

Remember: Stochastic Gradient Descent

Each update step to the **parameters** to minimize loss is:

$$\begin{aligned}\theta_t &= \theta_{t-1} - \alpha \nabla_\theta J(g_\theta) \\ &= \theta_{t-1} - \alpha \nabla_\theta \text{loss}(g_\theta(x), y)\end{aligned}$$

Q: Can we use the same idea to maximize loss over an input?

Single Gradient Ascent Step

Untargeted Adversarial Example:

$$x_{\text{adv}} = x + \epsilon \frac{\nabla_x \text{loss}(g_\theta(x), y)}{\|\nabla_x \text{loss}(g_\theta(x), y)\|}$$

Intuition: Move the point in the direction of the gradient, this locally increases the loss the most

Q: How do we move the example such that it is classified in some specific target class ? T

Targeted Adversarial Example:

$$x_{\text{adv}} = x - \epsilon \frac{\nabla_x \text{loss}(g_\theta(x), T)}{\|\nabla_x \text{loss}(g_\theta(x), T)\|}$$

Multi-step Projected Gradient Descent

Multi-step, untargeted adversarial example:

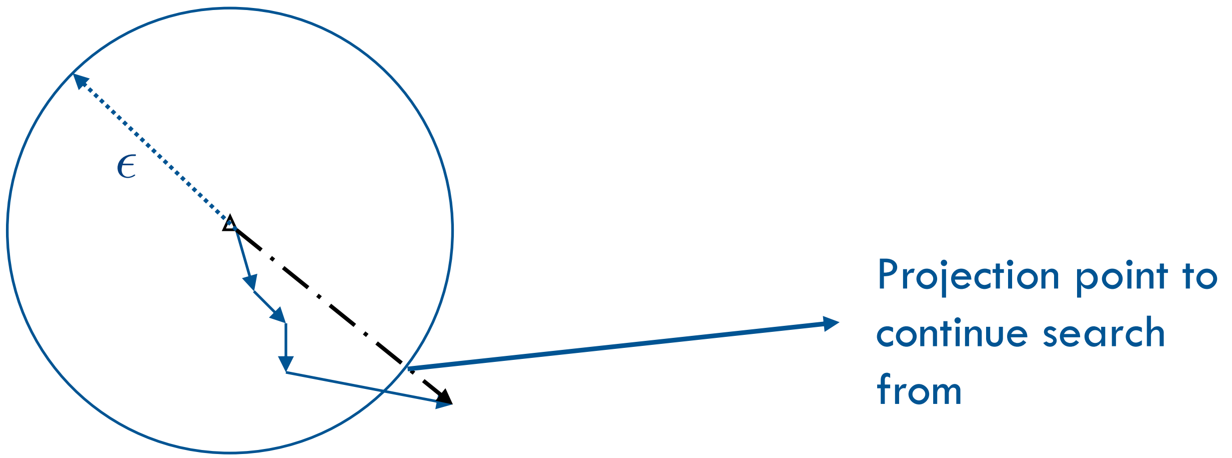
$$x^t_{\text{adv}} = P_{x,\epsilon}\left(x^{t-1}_{\text{adv}} + \alpha \frac{\nabla_{x^{t-1}_{\text{adv}}} \text{loss}(g_\theta(x^{t-1}_{\text{adv}}), y)}{\|\nabla_{x^{t-1}_{\text{adv}}} \text{loss}(g_\theta(x^{t-1}_{\text{adv}}), y)\|}\right)$$

Multi-step, targeted adversarial example:

$$x^t_{\text{adv}} = P_{x,\epsilon}\left(x^{t-1}_{\text{adv}} - \alpha \frac{\nabla_{x^{t-1}_{\text{adv}}} \text{loss}(g_\theta(x^{t-1}_{\text{adv}}), T)}{\|\nabla_{x^{t-1}_{\text{adv}}} \text{loss}(g_\theta(x^{t-1}_{\text{adv}}), T)\|}\right)$$

$P_{x,\epsilon}$ projects the adversarial example back onto an ϵ- ball around the original input

Intuition: Search for empirically best perturbation using small steps



Advanced: Robust Training

For e in [1,E]

for samples i in [1,N]

1. Compute adversarial example $x_{i,\text{adv}}^T$ for current state θ_{t-1}
2. Compute adversarial loss $\text{loss}(g_{\theta_{t-1}}(x_{i,\text{adv}}^T), y)$
3. Compute gradient $\nabla_\theta \text{loss}(g_{\theta_{t-1}}(x_{i,\text{adv}}^T), y)$
4. Update gradient $\theta_t = \theta_{t-1} - \alpha \nabla_\theta \text{loss}(g_{\theta_{t-1}}(x_{i,\text{adv}}^T), y)$