ADVERSARIAL MACHINE LEARNING

ATTACK MODEL : JSMA ALGORITHM

21

Algorithm 1 Crafting adversarial samples

$\mathbf{x}$ is the benign sample, ℓ is the target network output, $\mathcal{F}$ is the function learned by the network during training, Υ is the maximum distortion, and θ is the change made to features.

Input: $\mathbf{x}$, ℓ, $\mathcal{F}$, Υ, θ

$\mathbf{x}^* \leftarrow \mathbf{x}$

$\Gamma = \{1 \dots |\mathbf{x}|\}$

while $\mathcal{F}(\mathbf{x}^*) \neq l$ and $||\delta_{\mathbf{x}}|| < \Upsilon$ **do**

    Compute forward derivative $\nabla\mathcal{P}(\mathbf{x}^*)$

    $S =$ `saliency_map` $(\nabla\mathcal{P}(\mathbf{x}^*), \Gamma, l)$

    Modify $\mathbf{x}^*_{i_{max}}$ by θ s.t. $i_{max} = \arg\max_i S(\mathbf{x}, l)[i]$

    $\delta_{\mathbf{x}} \leftarrow \mathbf{x}^* - \mathbf{x}$

end while

return $\mathbf{x}^*$

ATTACK MODELS: CARLINI WAGNER ATTACKS

- Nicholas Carlini and David Wagner proposer three attack models with L_0, L_2 and L_∞ distance
- L2 attack is most optimal and broke all the existing defences
- The defined the problem as following:

$$\begin{aligned} \text{minimize } & \mathcal{D}(x, x+\delta) \\ \text{such that } & \mathcal{C}(x+\delta) = \ell \\ & x+\delta \in [0,1]^n \end{aligned}$$

$\mathcal{D}$ can be L_0, L_2, L_∞ distance

Carlini, N., & Wagner, D. (2017, May). Towards evaluating the robustness of neural networks.

ATTACK MODEL : JSMA ALGORITHM

Algorithm 1 **Crafting adversarial samples**

$\mathbf{x}$ is the benign sample, ℓ is the target network output, $\mathcal{F}$ is the function learned by the network during training, Υ is the maximum distortion, and θ is the change made to features.

Input: $\mathbf{x}$, ℓ, $\mathcal{F}$, Υ, θ

1: $\mathbf{x}^* \leftarrow \mathbf{x}$
2: $\Gamma = \{1 \ldots |\mathbf{x}|\}$
3: **while** $\mathcal{F}(\mathbf{x}^*) \neq l$ and $||\delta_{\mathbf{x}}|| < \Upsilon$ **do**
4: Compute forward derivative $\nabla\mathcal{P}(\mathbf{x}^*)$
5: $S = \texttt{saliency\_map}\left(\nabla\mathcal{P}(\mathbf{x}^*), \Gamma, l\right)$
6: Modify $\mathbf{x}^*_{i_{max}}$ by θ s.t. $i_{max} = \arg\max_i S(\mathbf{x}, l)[i]$
7: $\delta_{\mathbf{x}} \leftarrow \mathbf{x}^* - \mathbf{x}$
8: **end while**
9: **return** $\mathbf{x}^*$