How Does an Adversarial Attack Work?

Remember training is directed by a loss function.

$$\ell(h_\theta(x), y)$$

This loss function penalizes the difference between the ground truth labels and the network predictions.

$$\underset{\theta}{\text{minimize}} \frac{1}{m} \sum_{i=1}^{m} \ell(h_\theta(x_i), y_i)$$

Stochastic Gradient Descent

$$\theta := \theta - \frac{\alpha}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_\theta \ell(h_\theta(x_i), y_i)$$

When we apply Stochastic gradient descent over a batch B with α step size.

Key Insight

$$\theta := \boxed{\theta} - \frac{\alpha}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_\theta \ell(h_\theta(x_i), y_i)$$

We aren’t limited to differentiate the loss with respect to θ. e can also compute the gradient of the loss with respect to the input x.

Key Insight

$$\theta := \theta - \frac{\alpha}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_\theta \ell(h_\theta(\boxed{x_i}), y_i)$$

We aren't just limited to differentiate the loss with respect to θ. e can also compute the gradient of the loss with respect to the input x.

Modified Optimization

$$\underset{\hat{x}}{\text{maximize}}\, \ell(h_\theta(\hat{x}), y)$$

Therefore we **adjust the image** to maximize the loss. The optimal x is the adversarial example we are looking for.

Set the Optimization Constraints

$$\underset{\delta \in \Delta}{\text{maximize}}\, \ell(h_\theta(x+\delta), y)$$

It’s not particularly impressive that we can “fool” the classifier into misclassifying images. Instead need to ensure that x is close to our original input x.

Norm for the Adversarial Noise

$$\Delta = \{\delta : ||\delta||_\infty \leq \epsilon\}$$

$$||z||_\infty = \max_i |z_i|$$

Δ represents al the allowable perturbations. δ is the adversarial noise we add into the original image
A common way to limit delta is the perturbation is the infinite norm.