Algorithm: Poison instance generation

Input: target instance \( t \), base instance \( b \), learning rate \( \lambda \).

Initialize \( x \):

\[ x_0 \leftarrow b \]

Define:

\[ L_p(x) = \| f(x) - f(t) \|_2 \]

For \( i = 1 \) to \( \text{maxIters} \):

Forward step:

Backward step:

\[ x_i = \frac{\hat{x}_i - \lambda \beta b}{1 + \beta \lambda} \]

\[ \hat{x}_i = x_i - \lambda \nabla x L_p(x_i) \]

End for

Target train instances

Base train instances

Clean model:

Poisoned model:

Feature space visualization of unsuccessful single-shot poisoning attack

Ali Shafahi\textsuperscript{1}, W. Ronny Huang\textsuperscript{1}, Mahyar Najibi\textsuperscript{1}, Octavian Suciu\textsuperscript{1}, Christoph Studer\textsuperscript{2}, Tudor Dumitras\textsuperscript{1}, Tom Goldstein\textsuperscript{1}

\textsuperscript{1}University of Maryland and \textsuperscript{2}Cornell University

Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Nets

Introduction: Different types of Attacks

Evasion vs. Poisoning

- Modify images at test time
- Inject poisoned images at train time

Works for Different Settings

Artefacts Examples: manipulate the target

Transfer learning

Make poisons: collision attack method

\[ p = \text{argmin} \| f(x) - f(t) \|_2^2 + \beta \| x - b \|_2^2 \]

Close to target in feature space

Close to base in pixel space

Poison is near base in pixel space, collides with target in feature space

“One-shot kill” possible

Training data

Base

Poison

Target

Decision boundary

Decision boundary

“Target fishes”

“Clean-label” attacks: Poison is near base in pixel space, collides with target in feature space

How to attack in the end-to-end case?

“One-shot kill” attacks do not work here!

“IT’S TOUGH TO FOOL THE FEATURE EXTRACTORS!”

Transfer learning

End-to-end re-training

- Use pre-trained feature extractor
- Classification layers re-trained

“One-shot kill” attacks on InceptionV3

Transfer learning vs end-to-end

Make poisons: collision attack method

Clean-label: Poisons are labeled “correctly”

Performance only changes on a selected target

Multiple poisons required

FOOLING FEATURE EXTRACTORS REQUIRE:

- “Pre-trained net is used”
- “All layers are re-trained”

Successful poisoning in end-to-end: sucking out the target

HOW TO ATTACK IN THE END-TO-END CASE?

SUCCESS RATE DEPENDS ON \# OF POISONS AND OPACITY OF TARGET

Watermarking: overlay the target onto the poison

Makes it difficult to separate images!

Using multiple Poisons

Multiple poisons required

SUCCESS RATE OF VARIOUS EXPERIMENTS

Supervised security desk

Phishing/competitor email

Problem: feature layers learn to separate poison from target in feature space

Multiple Poisons

Successful poisoning in end-to-end: 60 poison dogs causing a bird to get misclassified

THREAT MODEL: CLEAN-LABEL ATTACKS

WHY POISON?

You can’t always control target!

Poison data can be placed on the web

Poison data can be sent/emailed to data collectors

Attacks are hard to detect

Clean-label: Poisons are labeled “correctly”

Performance only changes on a selected target

Attacks can be executed by outsider

Poison data can be placed on the web

Poison data can be sent/emailed to data collectors

Success rate depends on \# of poisons and opacity of target

Successful poisoning in end-to-end: sucking out the target

Fooling feature extractors require:

- “Pre-trained net is used”
- “All layers are re-trained”

Multiple poisons required

SUCCESS RATE OF VARIOUS EXPERIMENTS