Combinatorial Attacks on Binarized Neural Networks

Elias B. Khalil | Amrita Gupta | Bistra Dilkina

Problem Setting

- Adversarial Attack: a small perturbation to an input that fools ML model into predicting incorrect output (STOP → YIELD)
- Attack-to-Protect: designing better algorithms for generating adversarial perturbations helps train robust ML models
- ML model – Binarized Neural Network (BNN):
  - Weight: +1 / -1
  - Activation: \( \text{sign}(x) \in \{-1, +1\} \)
- Why care about BNNs?
  - Fast inference / Small size
  - Great for low-power devices, smartphones

This work: Effective combinatorial methods for attacking BNNs

Gradient Attacks Fail on BNN

True BNN

Approximated BNN

Final layer activations for inputs to a small BNN with two output classes (\( o_1 \) and \( o_2 \)) as a single input dimension (\( x_1 \)) is varied. The relative activations of the two classes differ significantly between the true BNN (left) and an approximation of the BNN (right) used to enable the gradient-based attack FGSM.

Experimental Evaluation

- MNIST Handwritten Digit Recognition dataset (100 test images)
- FGSM: Standard gradient-based attack
- Run each method for 180 seconds

How often does each attack fool the BNN?

- BNNS (x-axis): depth x width
- Flip rate (y-axis): fraction of images for which prediction changes
- Higher is better: IProp much better than FGSM; MIP does not scale well

Does IProp find good solutions early?

- Runtime (x-axis): 180 seconds time limit
- Objective value (y-axis)
- Higher is better: IProp outperforms FGSM quickly

Can IProp benefit from FGSM?

- Axes similar to bar chart
- Higher is better: IProp benefits from warmstarting with FGSM

Global Integer Program

Max

\[
\text{difference in outputs between target (Yield) and prediction (Stop)}
\]

subject to

\[
a_{l, j} = \sum_{j'=1}^{r} w_{l, j', j} \cdot (x_{j'} + p_{j'}) \quad \forall j \in [r]
\]

forward pass in BNN

\[
a_{l, j} \leq h_{l, j} \cdot h_{l-1, j}
\]

\[
a_{l, j} \geq h_{l, j} \cdot (1 - h_{l, j})
\]

\[
p_{j} \in [-\epsilon, \epsilon]
\]

perturbation

\[
h_{l, j} \in [0, 1]
\]

\[
a_{l, j} \in [L_{l, j}, U_{l, j}]
\]

Mixed-binary variables

Maximize

\[
\text{difference in outputs between target (Yield) and prediction (Stop)}
\]

subject to

\[
a_{l, j} = \sum_{j'=1}^{r} w_{l, j', j} \cdot (x_{j'} + p_{j'}) \quad \forall j \in [r]
\]

Forward pass in BNN

\[
a_{l, j} \leq h_{l, j} \cdot h_{l-1, j}
\]

\[
a_{l, j} \geq h_{l, j} \cdot (1 - h_{l, j})
\]

\[
p_{j} \in [-\epsilon, \epsilon]
\]

\[
h_{l, j} \in [0, 1]
\]

\[
a_{l, j} \in [L_{l, j}, U_{l, j}]
\]

IProp: Integer Target Propagation

Target

Find +1 / -1 assignment that produces Target

Layer 2

Layer 1

+1

-1

-1

Stop

+1

-2

YIELD

-2

YIELD

+2

STOP

-1

STOP

FIND +1 / -1 assignment that produces Target

(Input) Layer 0

-1

+1

-1

+1

-1

STOP

YIELD