Neural Network Repair using Formal Methods

Mayank Deora
Ph.D. student
Indian Statistical Institute Kolkata

Agenda

This presentation is based on the following research papers:

- Guy Katz et. al. Minimal Modifications of Deep Neural Networks using Verification (presented at the 23rd Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning LPAR 2020)
- Guy Katz et. al. Minimal Multi-Layer Modifications of Deep Neural Networks (https://arxiv.org/abs/2110.09929)

Problem statement

Minimal Repair of Deep Neural Networks (DNN)

Input

- DNN **N** (A faulty one)
- A set of specified input-output pairs (x_i, y_i) on which N fails

Problem statement

Minimal Repair of Deep Neural Networks (DNN)

Input

- DNN **N** (A faulty one)
- A set of specified input-output pairs (x_i, y_i) on which N fails

Goal

- Modify DNN **N** such that:
  - Modification to N is as small as possible
  - For each input x_i, we get the desired output y_i

Problem statement

Minimal Repair of Deep Neural Networks (DNN)

Input

- DNN **N** (A faulty one)
- A set of specified input-output pairs (x_i, y_i) on which N fails

Goal

- Modify DNN **N** such that:
  - Modification to N is as small as possible
  - For each input x_i, we get the desired output y_i

Constraints

- Cannot change the topology of N
- Cannot change activation function
- Can only alter weights

Related Work

- Semantic Repair (modifying the classification rules)
- Retraining with the counter-examples
- Modifying topology / activation functions

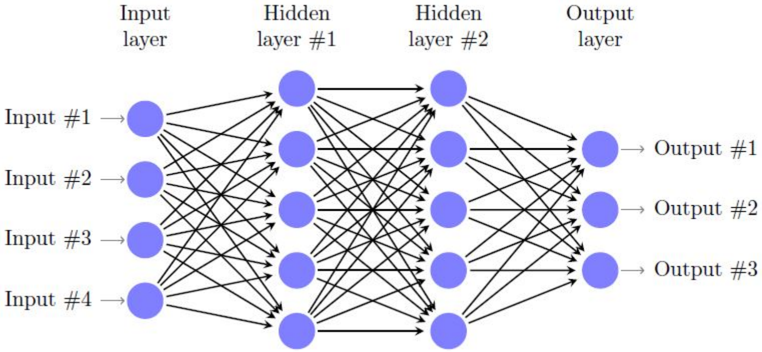
Novelty of this work: Model edits restricted to weights only

Applications of Neural Network Repair

Example

- In Machine Learning as as Service(MLaaS), clients often modify and fine-tune the DNN according to their needs.
  - Instead of modifying and retraining, minimal modification can be done.
- For low resource devices, retraining may be difficult
- Watermark analysis: For a DNN, preserve a specific set of tests for which the output classification should not change, even after retraining

Deep Neural Network: Introduction

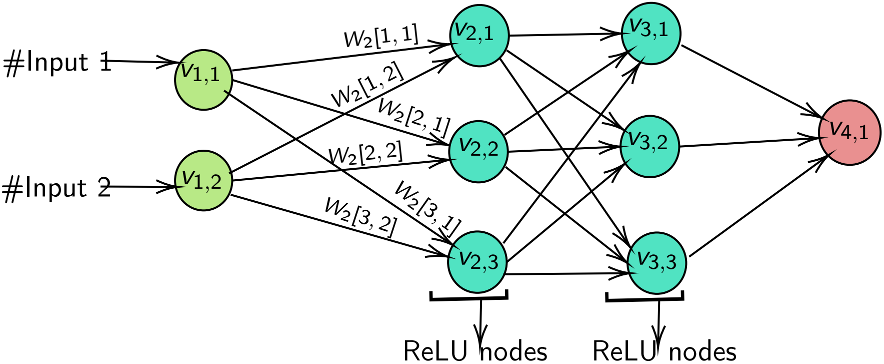


Deep Neural Network with 4 layers

Deep Neural Networks

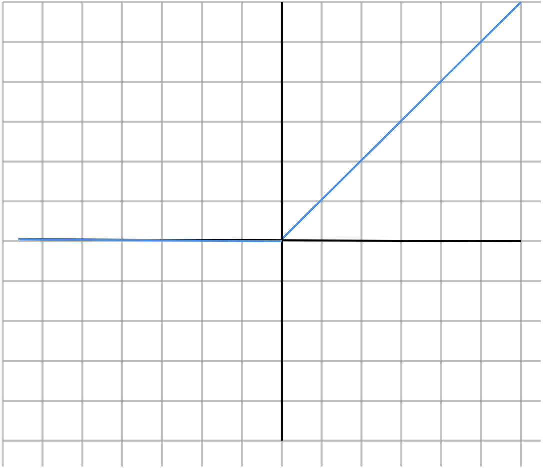
Notations

- $v_{i,j}$ represents the j^{th} neuron of the i^{th} layer.
- s_i represents the number of neurons in the i^{th} layer.
- Each layer $2 \leq i \leq n$ has a weight matrix W_i of size $s_i \times s_{i-1}$.
- $W_i[j,k]$ represents the weight on the edge from $v_{i-1,k}$ to $v_{i,j}$.



Notations

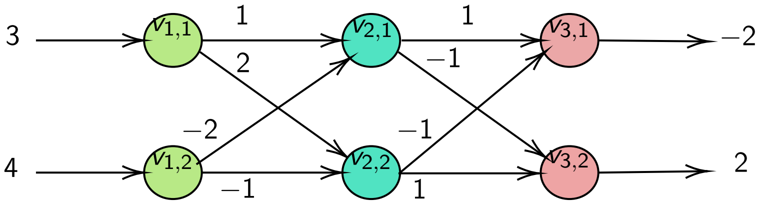
$\text{ReLU}(x) = max\{0, x\}$



Notations

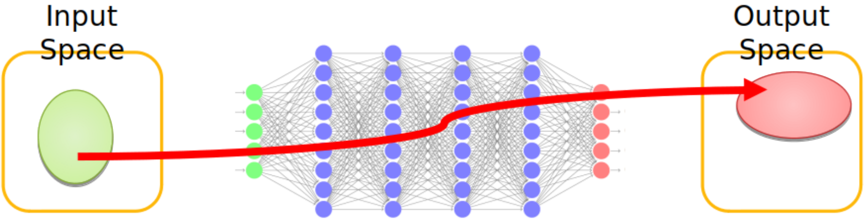
$\text{ReLU}(x) = max\{0, x\}$

$$v_{2,1} = ReLU(3 \times 1 + 4 \times (-2)) = ReLU(-5) = 0$$



$$v_{2,2} = ReLU(3 \times 2 + 4 \times (-1)) = ReLU(2) = 2$$

Verification of Deep Neural Networks



Definition

For a Neural Network $\overline{N} : \overline{x} \rightarrow \overline{y}$, an input property $P(\overline{x})$ and an output property $Q(\overline{y})$, does there exist an input $\overline{x_0}$, with output $\overline{y_0} = N(\overline{x_0})$ such that $\overline{x_0}$ satisfies P and $\overline{y_0}$ satisfies Q?

Both $\overline{x}$ and $\overline{y}$ are vectors over reals.

Verification of Deep Neural Networks

Definition

For a Neural Network $\overline{N} : \overline{x} \to \overline{y}$, an input property $P(\overline{x})$ and an output property $Q(\overline{y})$, does there exist an input $\overline{x_0}$, with output $\overline{y_0} = N(\overline{x_0})$ such that $\overline{x_0}$ satisfies P and $\overline{y_0}$ satisfies Q?

- $P(\overline{x})$ characterizes the inputs, we are checking.

Verification of Deep Neural Networks

Definition

For a Neural Network $\overline{N} : \overline{x} \rightarrow \overline{y}$, an input property $P(\overline{x})$ and an output property $Q(\overline{y})$, does there exist an input $\overline{x_0}$, with output $\overline{y_0} = N(\overline{x_0})$ such that $\overline{x_0}$ satisfies P and $\overline{y_0}$ satisfies Q?

- $P(\overline{x})$ characterizes the inputs, we are checking.
- $Q(\overline{y})$ characterizes the undesired behavior for those inputs.

Verification of Deep Neural Networks

Definition

For a Neural Network $\overline{N} : \overline{x} \to \overline{y}$, an input property $P(\overline{x})$ and an output property $Q(\overline{y})$, does there exist an input $\overline{x_0}$, with output $\overline{y_0} = N(\overline{x_0})$ such that $\overline{x_0}$ satisfies P and $\overline{y_0}$ satisfies Q?

- $P(\overline{x})$ characterizes the inputs, we are checking.
- $Q(\overline{y})$ characterizes the undesired behavior for those inputs, we are checking.
- Negative answer (UNSAT) means property Q holds.

Verification of Deep Neural Networks

Definition

For a Neural Network $\overline{N} : \overline{x} \rightarrow \overline{y}$, an input property $P(\overline{x})$ and an output property $Q(\overline{y})$, does there exist an input $\overline{x_0}$, with output $\overline{y_0} = N(\overline{x_0})$ such that $\overline{x_0}$ satisfies P and $\overline{y_0}$ satisfies Q?

- $P(\overline{x})$ characterizes the inputs, we are checking.
- $Q(\overline{y})$ characterizes the undesired behavior for those inputs, we are checking.
- Negative answer (UNSAT) means property Q holds.
- Positive answer (SAT) includes a counterexample, demonstrating an input x satisfying P for which Q holds.

DNN Repair Problem

Let N denote a DNN, let X denote a set of fixed input points $X = \{x_1, ..., x_n\}$

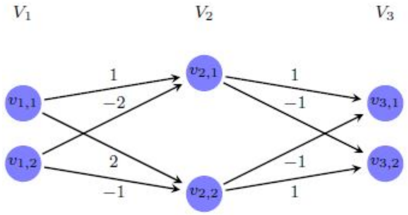
Let Q denote a predicate over the classifications $N(x_1), ..., N(x_n)$ of the points X.

Definition

The DNN repair problem is to find a new DNN N', such that $Q(N'(x_1), ..., N'(x_n))$ holds, such that the distance between N and N' is at most some $\delta > 0$.

DNN Repair Problem

- In following DNN N, for input $V_1 = \langle 3, 4 \rangle$, output will be $V_3 = \langle -2, 2 \rangle$, i.e. $v_{3,2} > v_{3,1}$



- We need $Q(N'(\langle 3, 4 \rangle)) = v_{3,1} \geq v_{3,2}$

Note:

Recall that hidden layer nodes are ReLU nodes

Distance between 2 DNNs

Let N^1 and N^2 denote two DNNs with identical topology i.e. the same number of layers ($n^1 = n^2$), and the same number of neurons in every pair of matching layers ($s_i^1 = s_i^2$ for all $1 \leq i \leq n^1$).

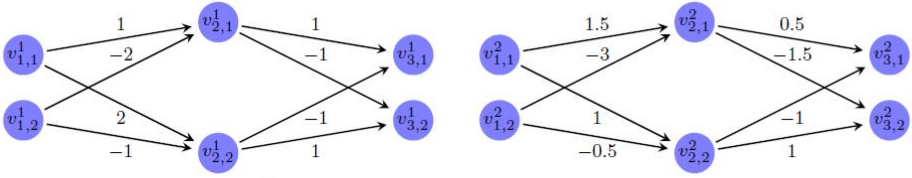
Definition

We define an L-distance between N^1 and N^2 denoted $||N^1 - N^2||_L$ as:

$$||N^1 - N^2||_L = \left(\sum_{i=2}^{n^1} \sum_{j=1}^{s_i^1} \sum_{k=1}^{s_{i-1}^1} |W_i^1[j,k] - W_i^2[j,k]|^L \right)^{1/L}$$

for $L \neq \infty$

Distance between 2 DNNs



(a) N^1 (b) N^2

Distance between N^1 and N^2

$$||N^1 - N^2||_L = \left(\sum_{i=2}^{n^1} \sum_{j=1}^{s_i^1} \sum_{k=1}^{s_{i-1}^1} |W_i^1[j,k] - W_i^2[j,k]|^L \right)^{1/L}$$

$$||N^1 - N^2||_1 = (|-0.5| + |1| + |1| + |-0.5| + |0.5| + |0.5| + |0| + |0|)$$

$$||N^1 - N^2||_\infty = max\{|-0.5|, |1|, |1|, |-0.5|, |0.5|, |0.5|, |0|, |0|\}$$

DNN Minimal Repair Problem

Input

- DNN **N** (A faulty one)
- Input points $X = \{x_1, ..., x_n\}$, and desired predicates $Q(N'(x_1), ..., N'(x_n))$

Goal

- Modify DNN **N** such that:
  - Distance between N and modified N' ($||N - N'||_L$) is the least.
  - As mentioned before, can only modify weights.

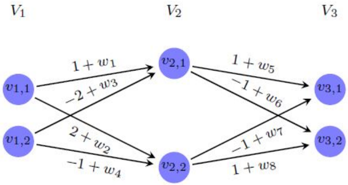
Minimal Repair as an Optimization Problem

Minimal Repair as an Optimization Problem

$$Min \, ||N - N'||_L$$

$$subject\ to : Q$$

How to generate the repaired network?



Recall

- To get N', we can only modify the weights.
- In this case, we have a single testcase to fix: input <3,4> on which the current N produces $v_{3,2} > v_{3,1}$ while Q specifies $v_{3,1} \geq v_{3,2}$.

An Optimization Problem

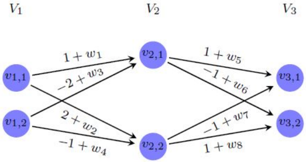
Minimal Repair as an Optimization Problem

$$Min\,||N - N'||_L$$

$$sub\ to:\ v_{3,1} \geq v_{3,2}$$

$v_{3,1} = (1 + w_5)\text{ReLU}(3(1 + w_1) + 4(-2 + w_3)) + (-1 + w_7)\text{ReLU}(3(2 + w_2) + 4(-1 + w_4))$
$v_{3,2} = (-1 + w_6)\text{ReLU}(3(1 + w_1) + 4(-2 + w_3)) + (1 + w_8)\text{ReLU}(3(2 + w_2) + 4(-1 + w_4))$

Due to ReLU functions and multiplication of w variables, it is non-linear and high-dimensional.



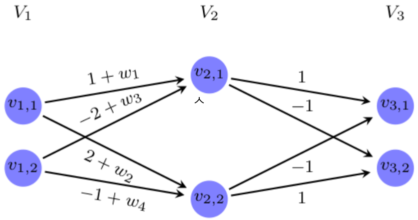
An Optimization Problem

- Due to ReLU functions and multiplication of w variables, it is a non-linear optimization problem.
- Larger networks make the constraints more complex.

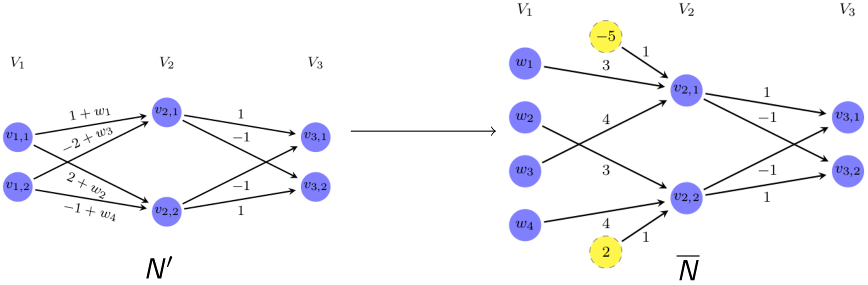
Problem!!

Restricted Repair

To overcome the problem, repair is restricted to weight modifications in <u>only</u> a single layer.



N'



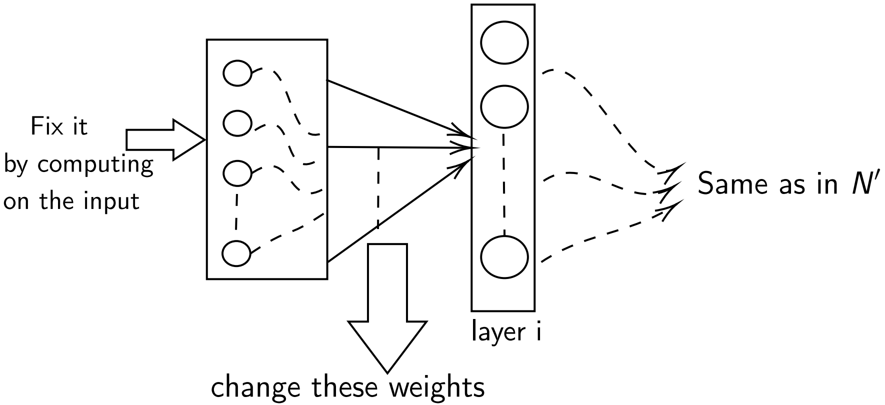
$$X = \{\langle 3, 4\rangle\}$$

$$v_{3,1} = \mathrm{ReLU}(3w_1 + 4w_3 - 5) - \mathrm{ReLU}(3w_2 + 4w_4 + 2)$$

$$v_{3,2} = -\mathrm{ReLU}(3w_1 + 4w_3 - 5) + \mathrm{ReLU}(3w_2 + 4w_4 + 2)$$

Note: The two networks are equivalent.

Modification to single layer



Construction of $\overline{N}$

Single layer modification

For some fixed point $x \in X$, for modified DNN N'

$$V_k = V_{k'} \;\; \forall k(1 \leq k \leq i-1)$$

$$V_i' = \text{ReLU}(W_i' V_{i-1}') = \text{ReLU}((W_i + W_\epsilon) V_{i-1}') \tag{1}$$

$$\Leftrightarrow$$

DNN $\overline{N}$ (having input W_ϵ and next(second) layer computes the ReLU function in equation 1, which is fed into layers $,i+1, ..., n$ of N').

DNN Repair reduced to DNN verification

Summary till now

- We started with the faulty network N and an input output pair $(3, 4)$
- We chose a layer in N and added all possible edge weight modifications in that layer to get N'
- From N', we got a transformed network $\overline{N}$

From Repair on N to verification on $\overline{N}$

For $x = \langle 3, 4 \rangle$, and $\delta > 0$, consider the DNN Verification query with the following predicates on $\overline{N}$:

$$P = \bigwedge_{i=1}^{4} -\delta \leq w_i \leq \delta \qquad Q = v_{3,1} \geq v_{3,2}$$

DNN Repair reduced to DNN verification

$x = \langle 3, 4 \rangle$, $\delta > 0$, DNN Verification query on $\overline{N}$:

$$P = \bigwedge_{i=1}^{4} -\delta \leq w_i \leq \delta \qquad Q = v_{3,1} \geq v_{3,2}$$

The above DNN Verification is SAT $\Leftrightarrow$ DNN Modification to N' is SAT (In L_∞ norm)

Modification to N' is SAT implies that Repair with δ is possible.

DNN Repair reduced to DNN verification

Proof.

($\Rightarrow$) Given that DNN verification problem is SAT.

$$||N - N'||_\infty = max\{|w_1|, |w_2|, |w_3|, |w_4|\}$$

Since, $\forall\ 1 \leq i \leq 4, |w_i| \leq \delta$,

$$||N - N'||_\infty \leq \delta$$

($\Leftarrow$) If DNN modification problem is SAT, then

$$||N - N'||_\infty = max\{|w_1|, |w_2|, |w_3|, |w_4|\} \leq \delta$$

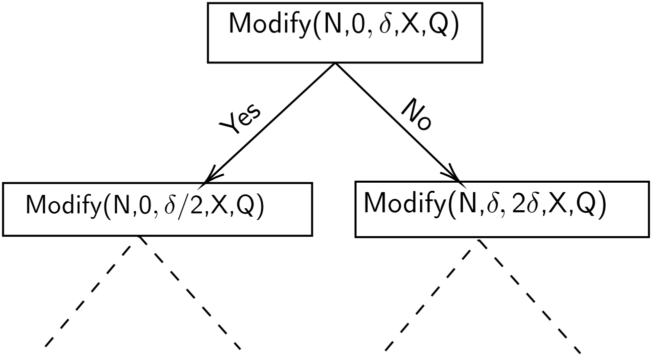
$$\implies \bigwedge_{i=1}^{4} -\delta \leq w_i \leq \delta$$

$\square$

DNN Minimal Repair Problem

Consider the decision variant of the DNN Repair problem: given N, Q, X and δ, does there exist any N' such that $||N' - N||_L < \delta$.

We can solve the minimal repair problem using binary search on δ.



DNN Minimal Repair reduced to DNN Verification

Reduction

Using Binary search on δ:
DNN minimal modification problem $\Rightarrow$ DNN modification problem $\Rightarrow$ DNN verification problem

DNN Minimal Repair reduced to DNN Verification

Modification for multiple points

For multiple points in X:

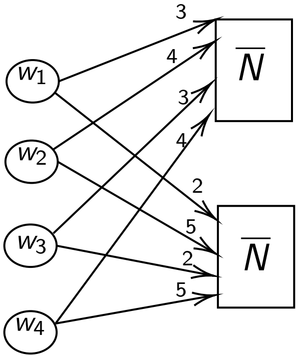
- Duplicate the construction of $\overline{N}$ for each $x \in X$.
- Consider it as a big DNN.
- Pass it to the underlying verification engine.

DNN Minimal Repair reduced to DNN Verification

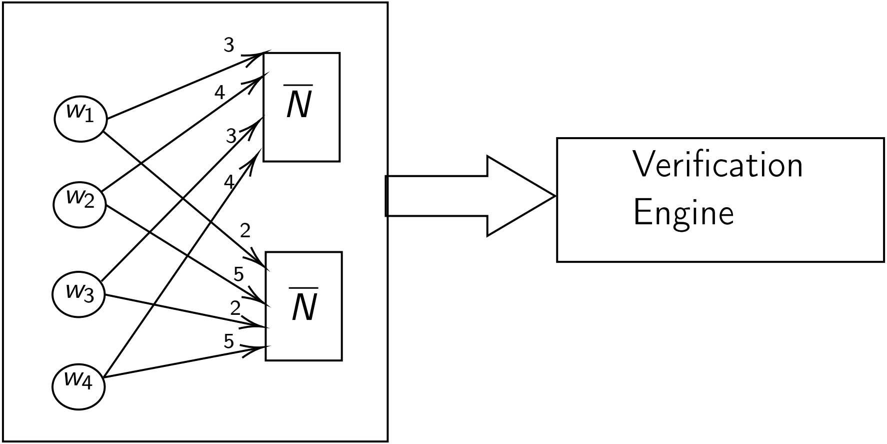
For multiple points in X:

- Duplicate the construction of $\overline{N}$ for each $x \in X$.
- Consider it as a big DNN.
- Pass it to the underlying verification engine.

Example: $X = \{\langle 3,4\rangle\langle 2,5\rangle\}$



- Modification for multiple points example:



DNN Verification for modifying the output layer

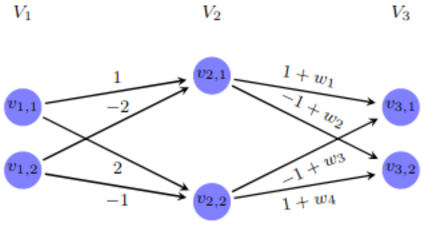
- DNN verification with RELU is NP-complete [Reference: Reluplex paper by Guy Katz et. al].
- Existence of multiple points in the set X makes the problem even more difficult.

An interesting observation that helps

Changes limited to only the weights towards output layer will render the problem easier to solve.

- No ReLU nodes at the output layer.

DNN Verification for modifying the output layer



Using the L_∞ norm, and by changing only the output layer, the minimal modification problem can be encoded as a linear program.

DNN Verification for modifying the output layer

$$Min: \ \delta$$

$$Sub\,to\ \delta \geq 0$$

$$-\delta \leq w_1 \leq \delta$$

$$-\delta \leq w_2 \leq \delta$$

$$-\delta \leq w_3 \leq \delta$$

$$-\delta \leq w_4 \leq \delta$$

$$v_{3,1} = 0(1 + w_1) + 2(-1 + w_3)$$

$$v_{3,2} = 0(-1 + w_2) + 2(1 + w_4)$$

$$v_{3,1} \geq v_{3,2}$$

DNN Verification for modifying the output layer

But using the L_1 norm, Linear Program is not possible to due to absolute values.

Watermark Resilience

For a DNN N, a watermark x is an input point, with a specific label l.

A set of watermarks ,$X = \{x_1, ..., x_n\}$ is called $\delta-$ resilient if for every DNN N' such that $||N - N'|| \leq \delta$, it holds that $N(x_i) = N'(x_i)$ for all $1 \leq i \leq n$.

Applications

Watermark Resilience

A set of watermarks $X = \{x_1, ..., x_n\}$ is called $\delta-$ resilient if for every DNN N' such that $||N - N'|| \leq \delta$,

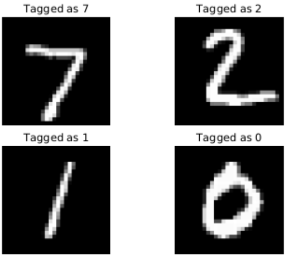
$$N(x_i) = N'(x_i) \;\; \forall 1 \leq i \leq n$$

.

Solving DNN minimal modification problem serves two purposes:

- Measuring the resilience of the Watermark set X.
- Comparing different watermarking schemes (for generating the set X) for a DNN on the basis of resilience.

Evaluation: Watermark Resilience



(a) Standard MNIST inputs.

(b) Watermark inputs.

Implemented as 3M-DNN tool.

3M-DNN is comprised of two logical levels:

- Search level: Heuristic search through possible changes to the values computed by the separation layers.
- Single-layer modification level: modifying the sub-networks

Minimal Multi-Layer Modifications of Deep Neural Networks

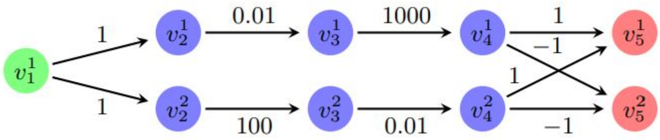
Goal is same as that in the minimal modification problem.

- Split the original DNN into multiple subnetworks.
- On input to first subnetwork, outputs of in between subnetworks are searched (using search heuristics).
- Apply single layer modification to the subnetworks
- Best cost and corresponding best change is found

Note:

It is an *Anytime* algorithm (more time gives better performance).

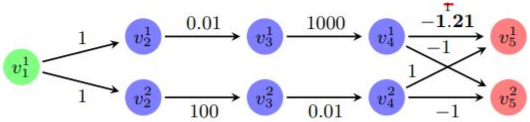
Example



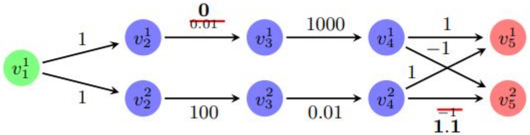
- On input 1, $v_5^1 > v_5^2$
- we want $v_5^2 \geq v_5^1$

Example

- we want $v_5^2 \geq v_5^1$



Vs



Thank You