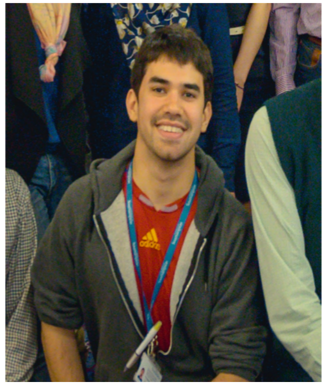
IX Brazilian Symposium on Computing Systems Engineering 2019

Incremental Bounded Model Checking of Artificial Neural Networks in CUDA

Luiz H. Sena, Iury V Bessa, Lucas C. Cordeiro, Mikhail R. Gadelha and Edjard Mota



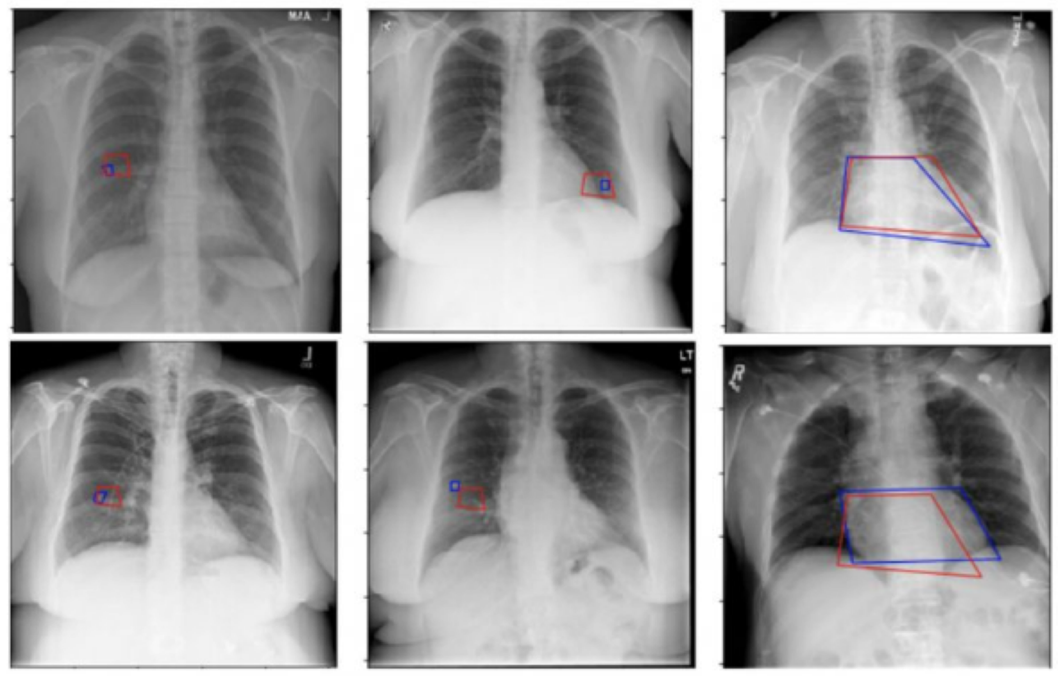



Summary

- Introduction
- Preliminaries
- Incremental BMC of ANNs in CUDA
  - Verification of Covering Methods
  - Verification of Adversarial Cases
- Experimental Evaluation
- Conclusion

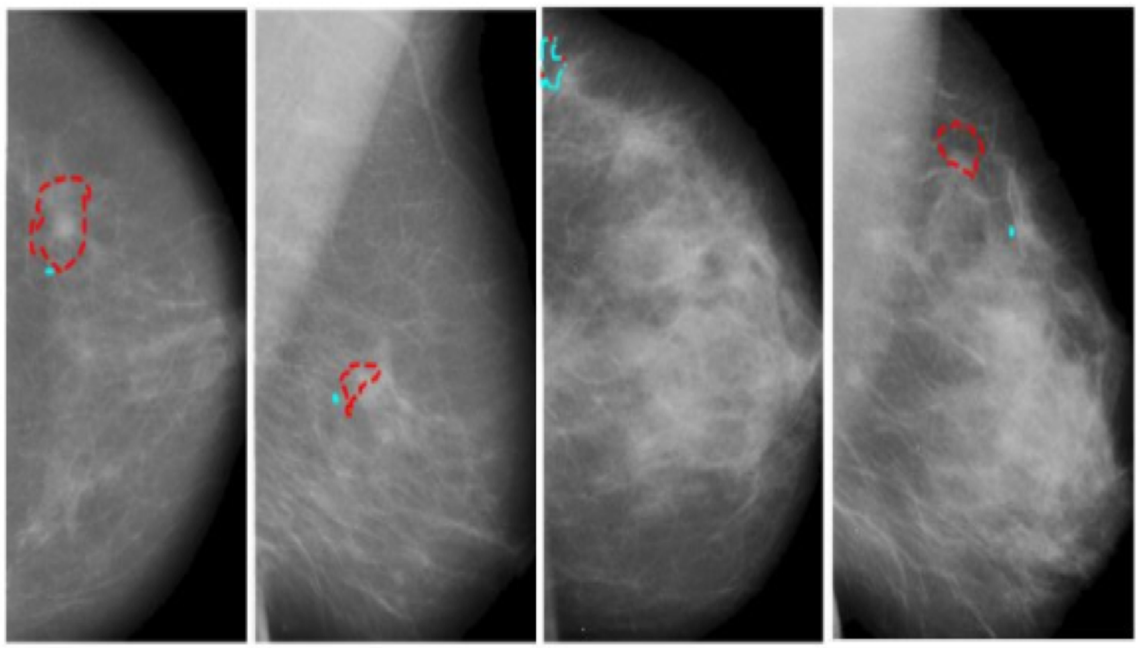
AI in Safety Critical Systems

- Marking regions to examine;



Red quadrilateral is marked by ANN and blue quadrilateral is marked by the radiologist.

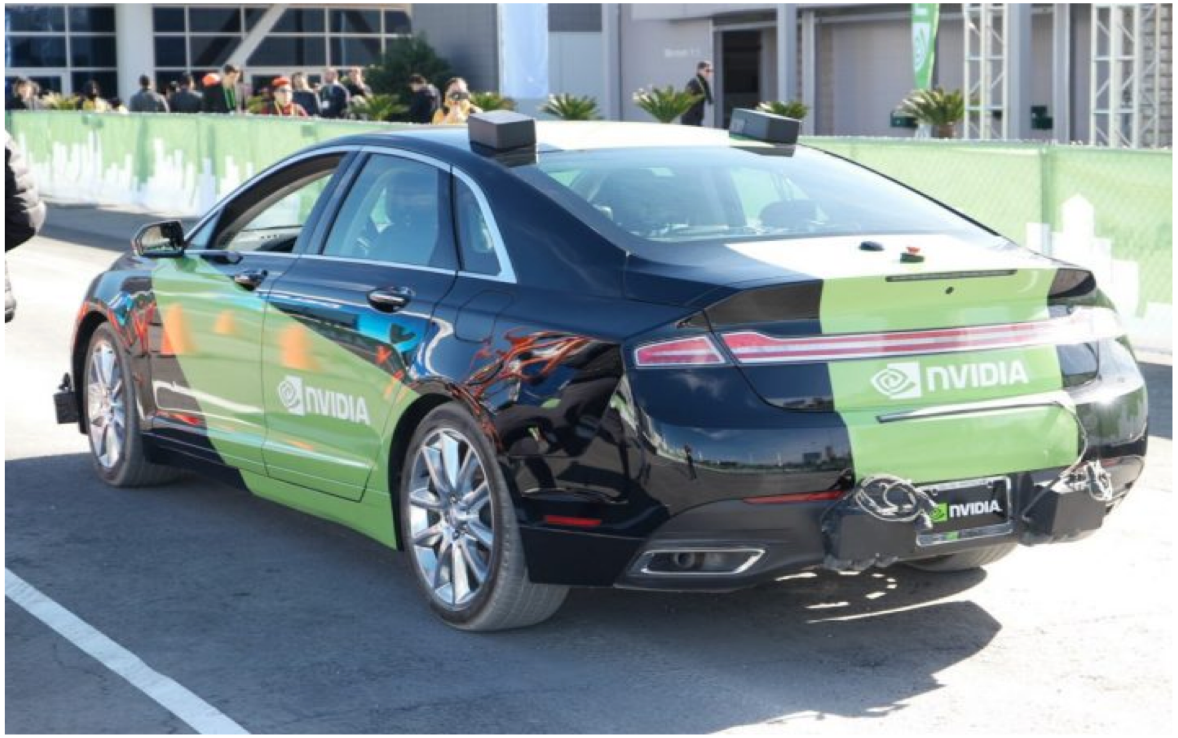
- Mass detection and localization.

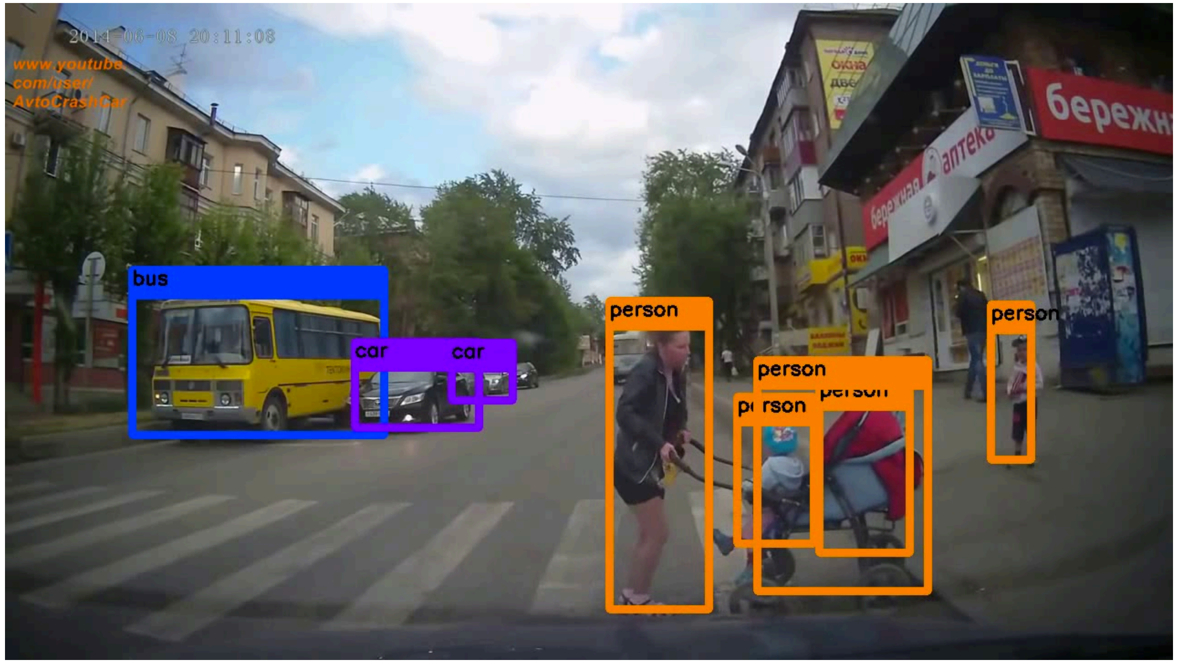


Red contours denote ground truth and cyan contours are false positive.

Self-driving car

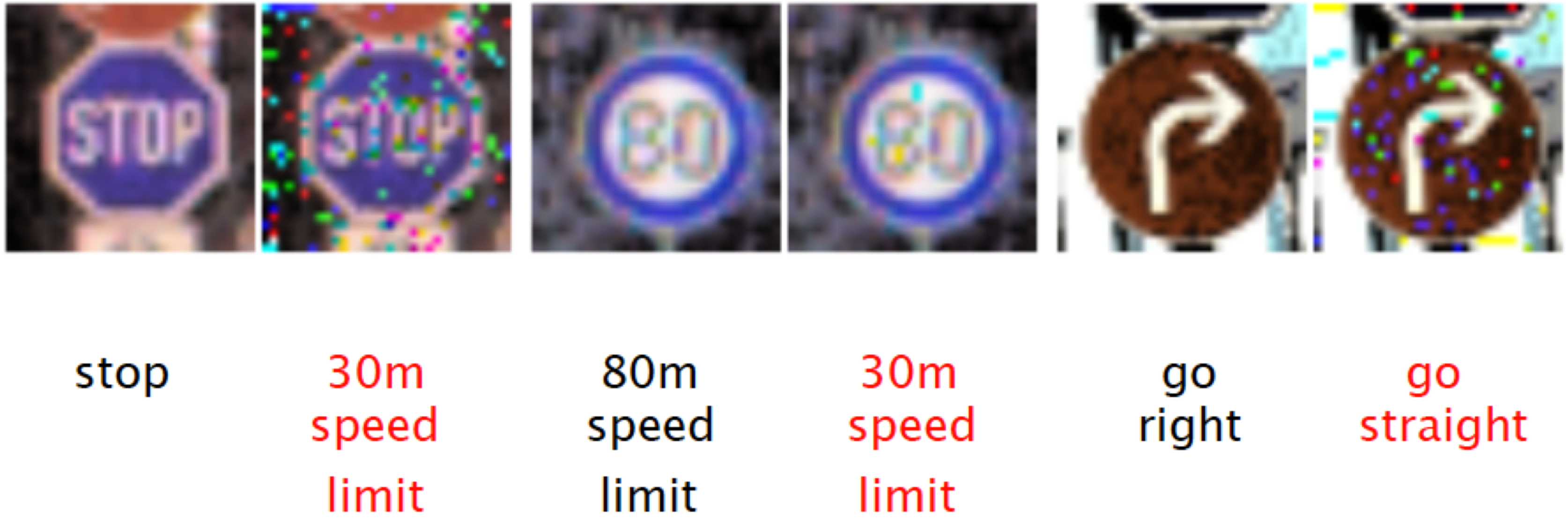
- Recognizing traffic signs and objects;





Adversarial Cases

- Adversarial Cases are not simple ANN errors, but particular cases where the correct label can be easily classified by humans.



Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam

A woman crossing Mill Avenue at its intersection with Curry Road in Tempe, Ariz., on Monday. A pedestrian was struck and killed by a self-driving Uber vehicle at the intersection a night earlier.
Caitlin O'Hara for The New York Times

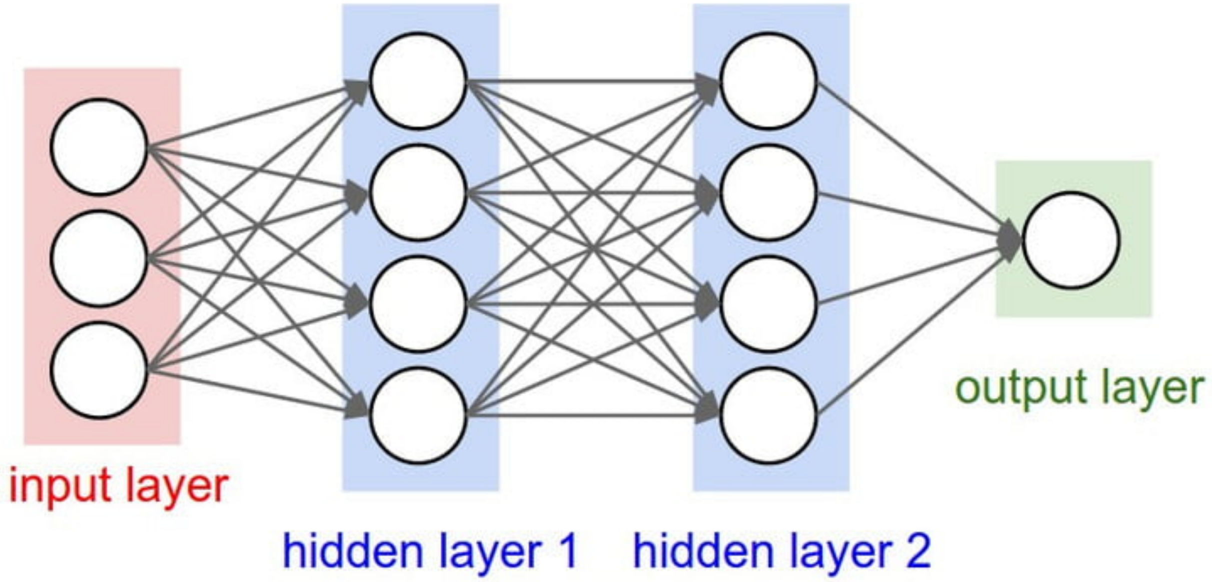
Objectives

- Showing how unsafe an ANN through:

- Generated adversarial cases
- How adversarial is a set of images for the ANN neurons w.r.t. covering methods.

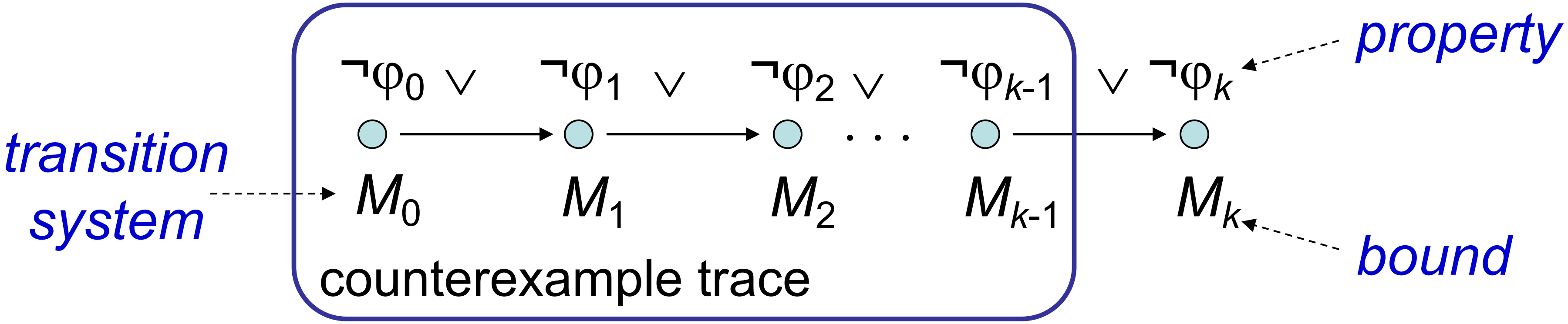
Artificial Neural Networks

- Artificial Neural Networks (ANNs) are versatile systems capable of generalizing and responding to unexpected inputs/patterns;
- ANNs are based in mathematical operations and learning algorithms.



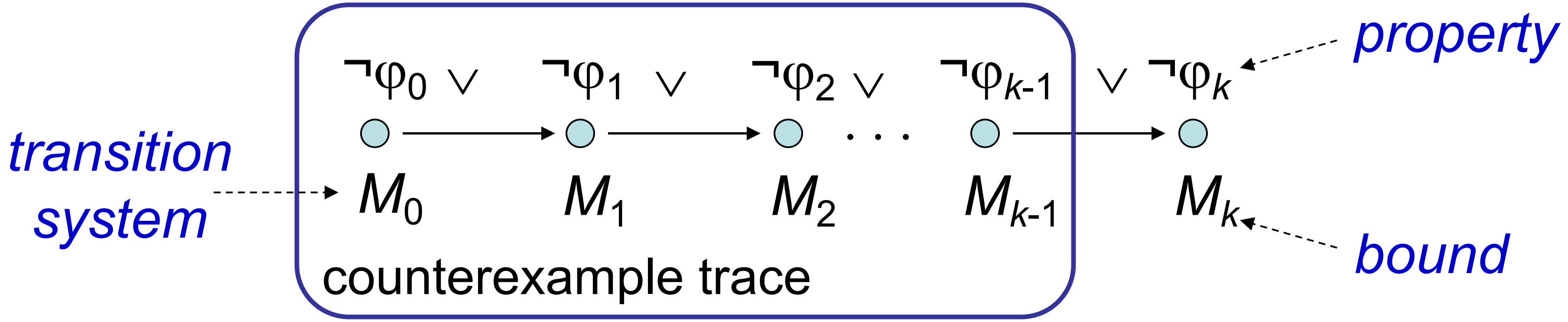
Bounded Model Checking (BMC)

Basic idea: check negation of given property up to given depth



Bounded Model Checking (BMC)

Basic idea: check negation of given property up to given depth

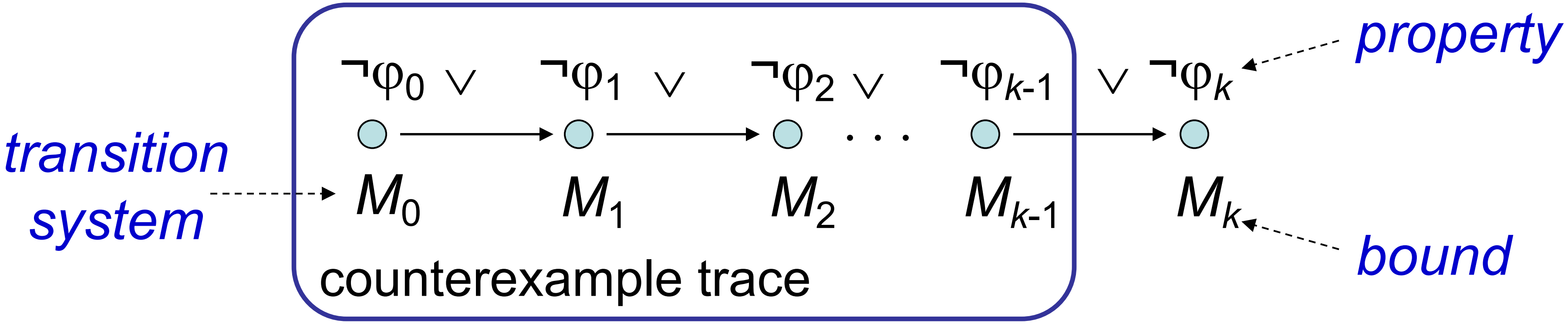


- Transition system M unrolled k times
  - for programs: loops, recursion, …
- Translated into verification condition ψ such that

  ψ **satisfiable iff** φ **has counterexample of max. depth** $\boldsymbol{k}$

Bounded Model Checking (BMC)

Basic idea: check negation of given property up to given depth

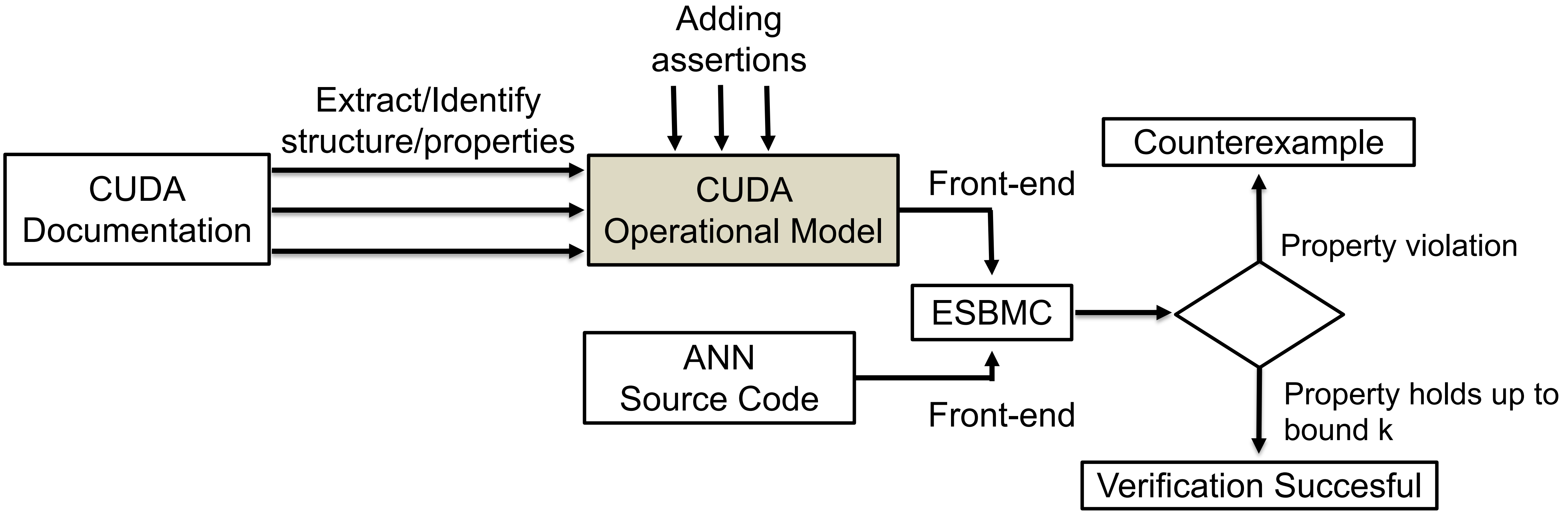


- Transition system M unrolled k times
  - for programs: loops, recursion, …
- Translated into verification condition ψ such that

ψ **satisfiable iff φ has counterexample of max. depth k**

BMC has been applied successfully to verify HW and SW

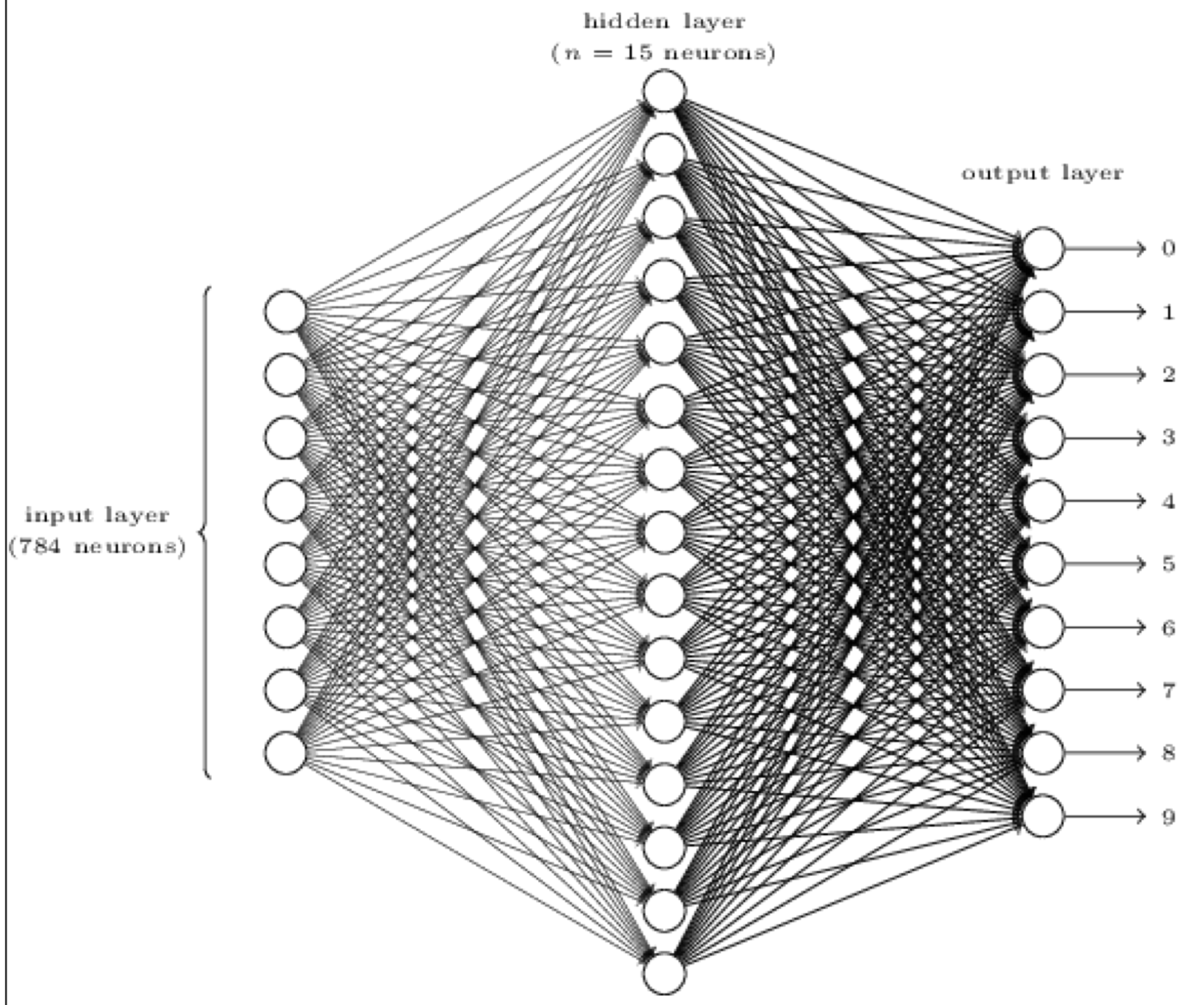
CUDA Operational Model



ANNs in CUDA

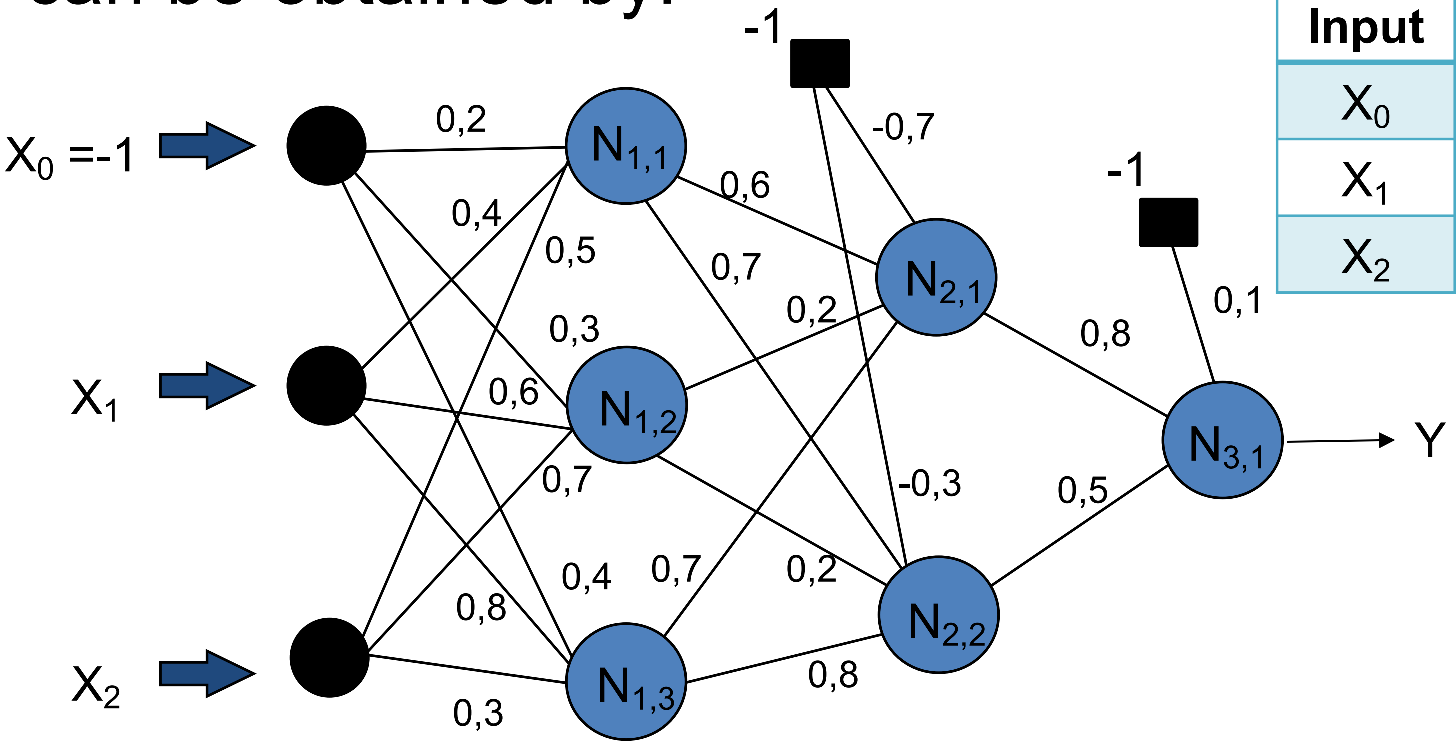

```
void feedForward(){
        cublasSgemm(cublasHandle, CUBLAS_OP_T, CUBLAS_OP_N,
        Layer.Outputs, batchSize, Layer.Inputs,
        1,
        Layer.MatrixBias, Layer.Inputs,
        data, Layer.Inputs,
        0,
        LayerResults,Layer.Outputs);

        cublasSgemm(cublasHandle, CUBLAS_OP_N, CUBLAS_OP_N,
        Layer.Outputs, batchSize, 1,
        0,
        BiasVector, Layer.Outputs,
        onevec, 1,
        1,
        LayerResults, Layer.Outputs);
        __ESBMC_assert((LayerResults[CorrecLabel] > P) || (
           LayerResults[WrongLabel] < P),"The correct label
           became a wrong label")
```



Verification of Covering Methods

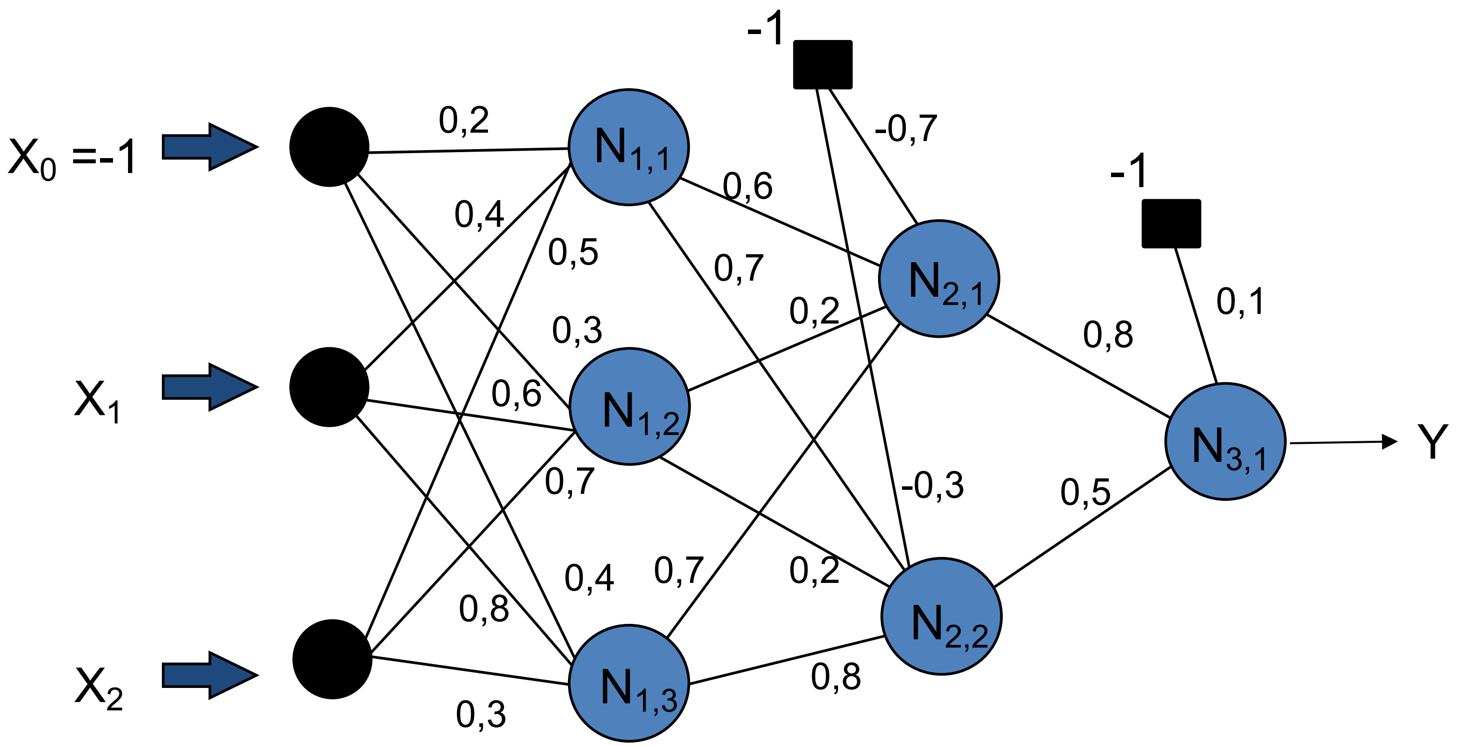
- For a ANN with linear activation function the neuron activation potencial $v_{n,k}$, where n is the neuron index and k is the layer, can be obtained by:



Input	Weight	v
X_0	W^1_{n0}	$X_0{*}W^L_{n0}+X_1{*}W^L_{n1}+X_2{*}W^L_{n2}$
X_1	W^1_{n1}	
X_2	W^1_{n2}	

Verification of Covering Methods

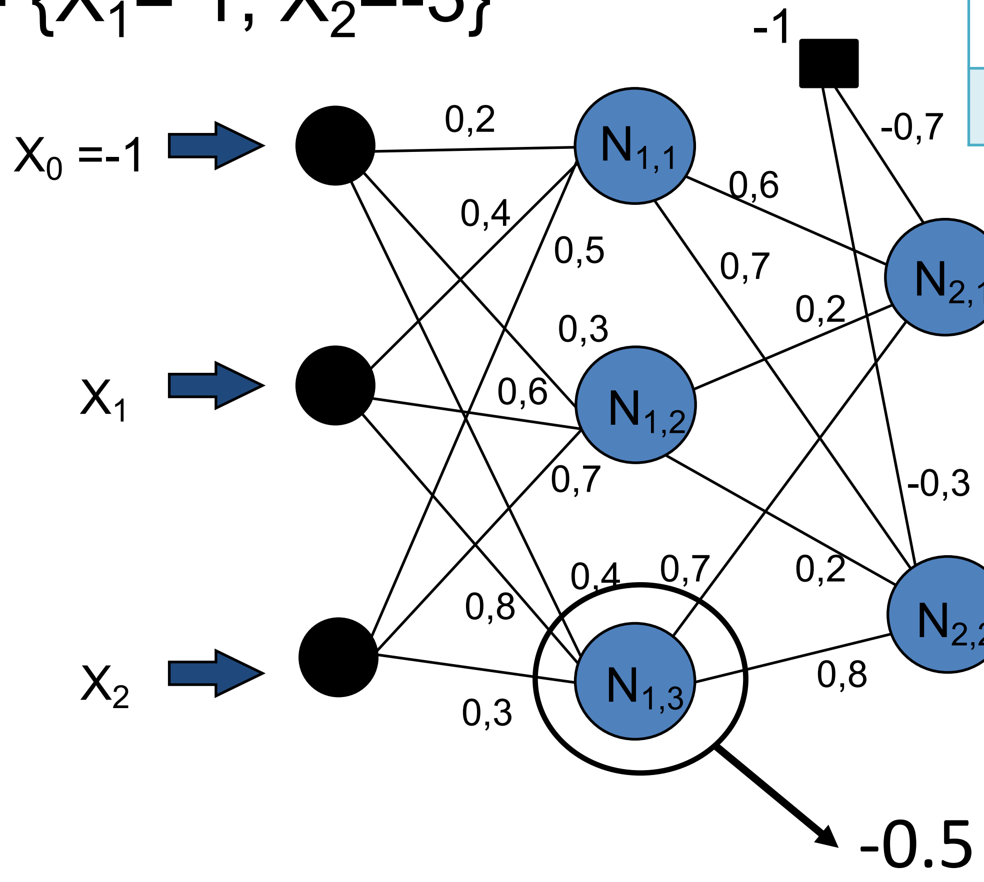
- Sign change (sc) occurs when the activation potencial of a neuron changes with respect to two different inputs.



Verification of Covering Methods

- Sign change (sc):
  - $X_A = \{X_1 = 1; X_2 = -3\}$

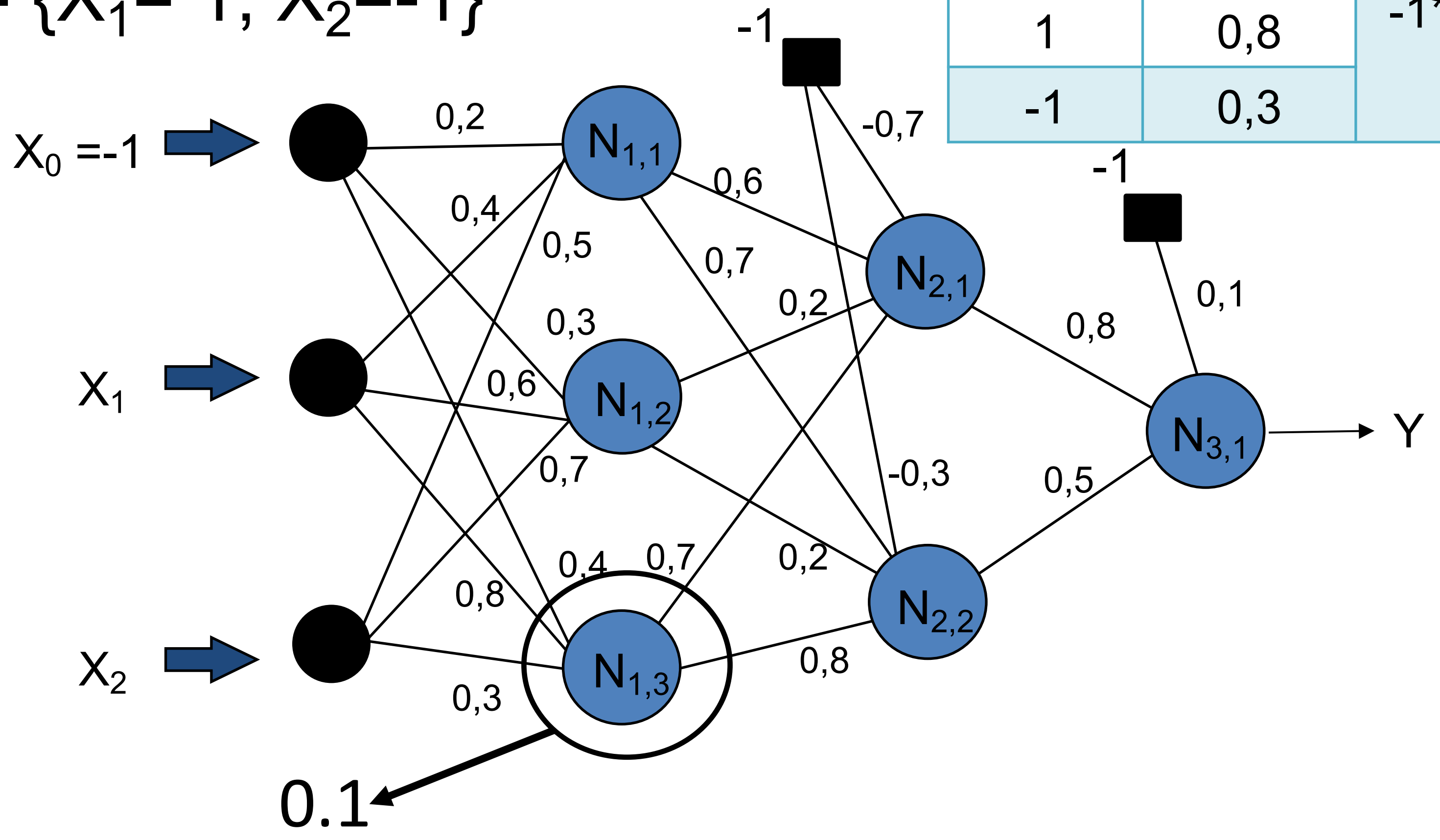
Input	Weight	$v_{1,3}$
-1	0,4	-1*0,4 + 1*0,8 -3*0.3 = -0,5
1	0,8	
-3	0,3	



Verification of Covering Methods

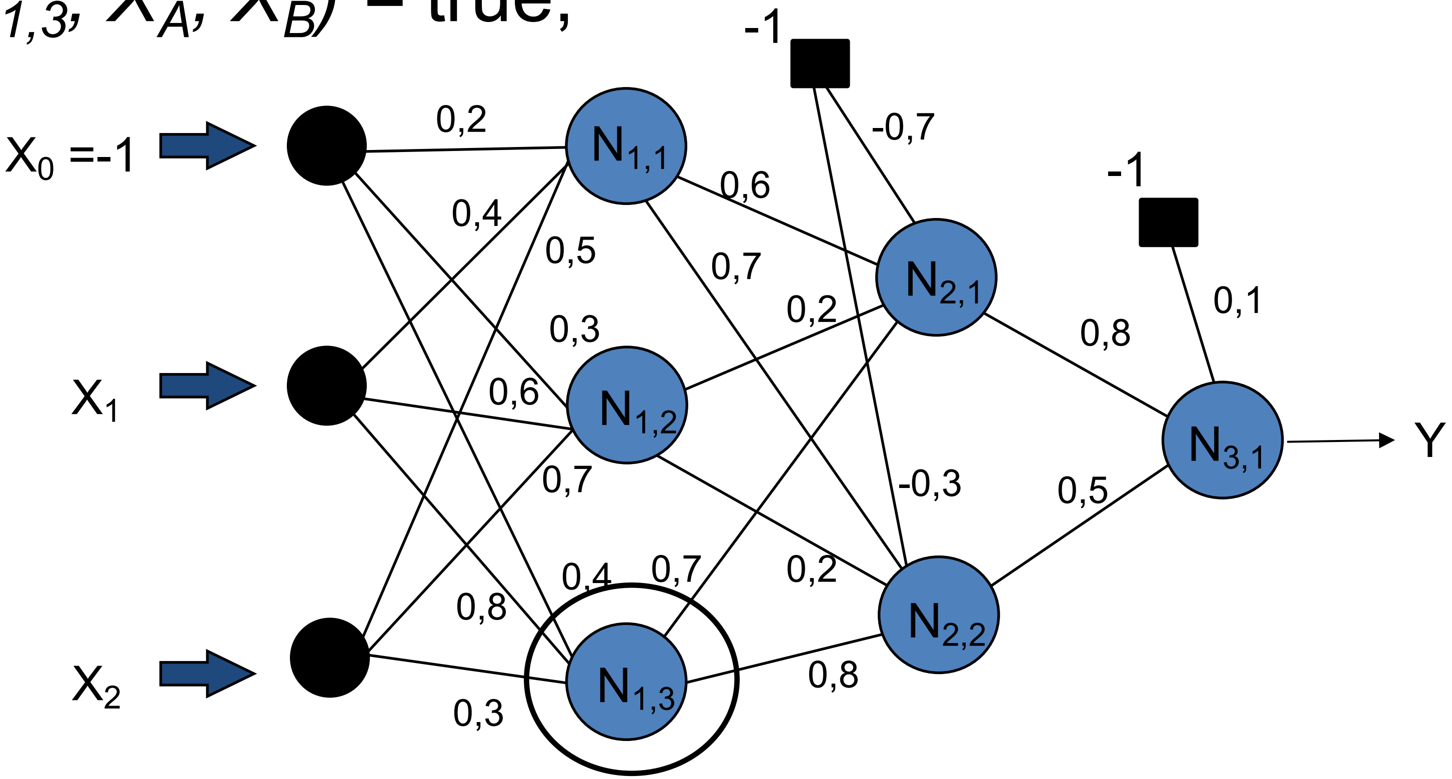
- Sign change (sc):
  - $X_B = \{X_1 = 1; X_2 = -1\}$

Input	Weight	$v_{1,3}$
-1	0,4	-1*0,4 + 1*0,8 -1*0.3 = 0,1
1	0,8	
-1	0,3	



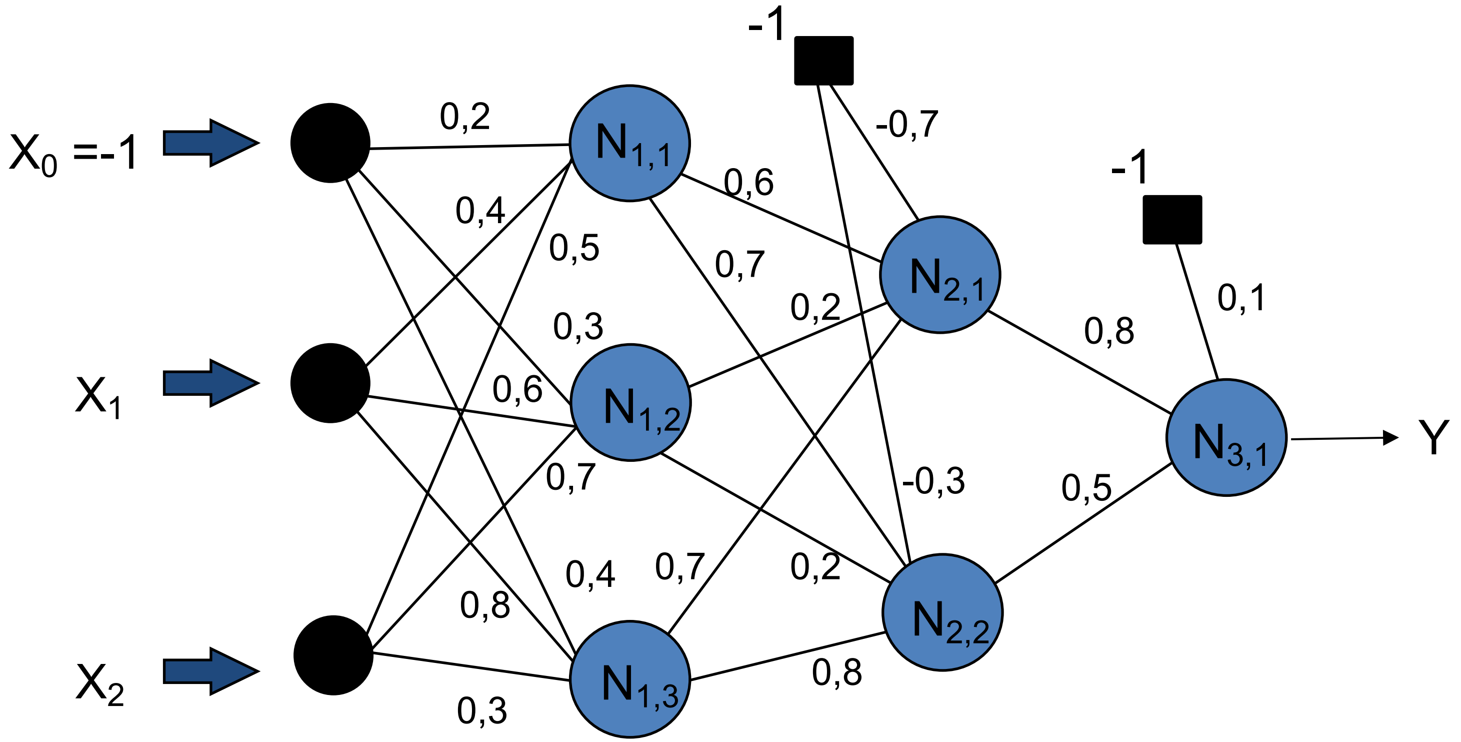
Verification of Covering Methods

- Sign change (sc):
  - $sc(n_{1,3}, X_A, X_B)$ = true;



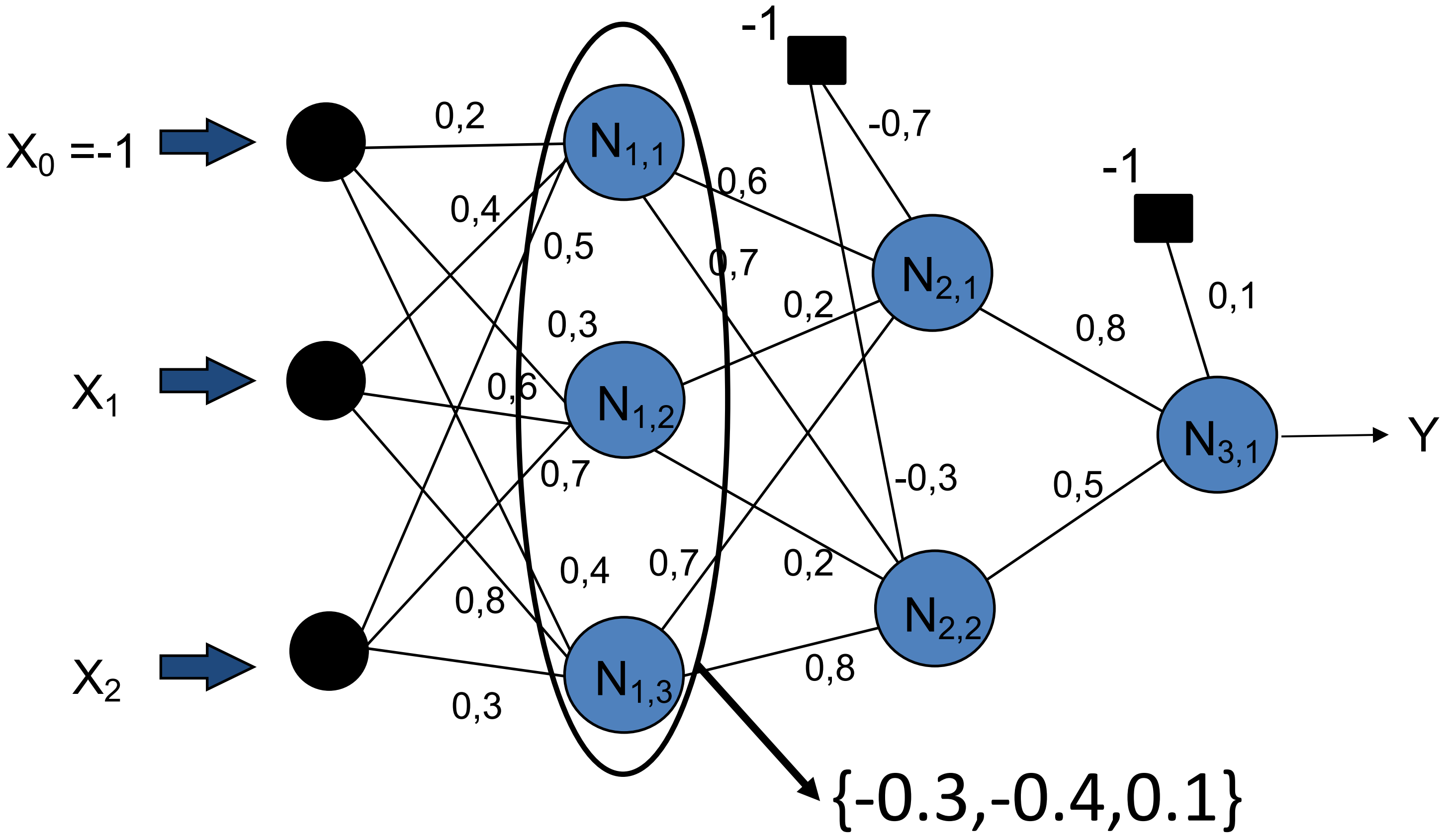
Verification of Covering Methods

- Value change (vc) occurs when there is no sign change in every neuron of a layer, but there is atleast one who has a value change with respect to a metric h.



Verification of Covering Methods

- Value change (vc):
  - $X_A = \{X_1 = 1; X_2 = -1\}$



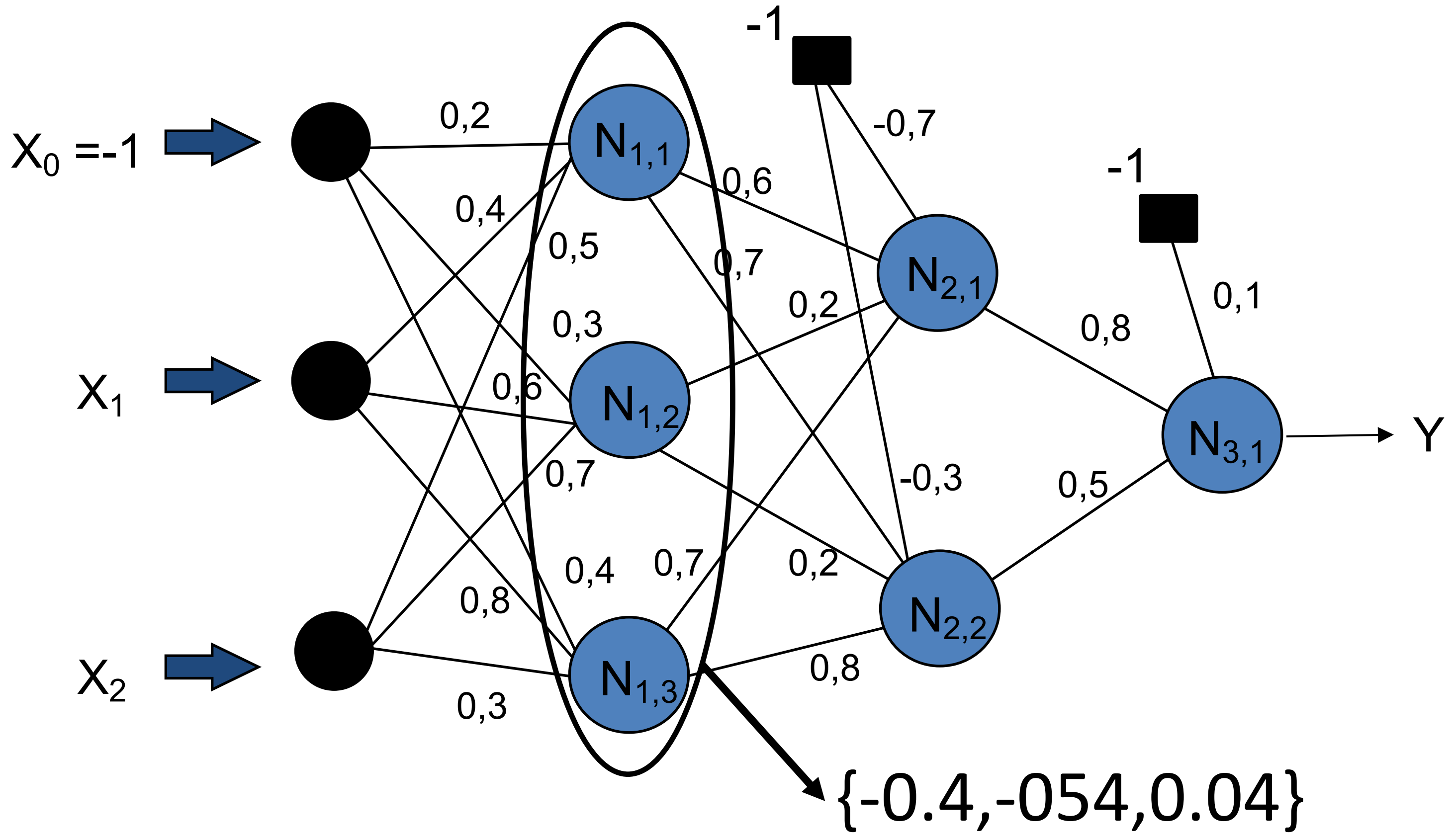
Input	Weight	$v_{1,1}$
-1	0,2	-0,3
1	0,4	
-1	0,5	

Input	Weight	$v_{1,2}$
-1	0,3	-0,4
1	0,6	
-1	0,7	

Input	Weight	$v_{1,3}$
-1	0,4	0,1
1	0,8	
-1	0,3	

Verification of Covering Methods

- Value change (vc):
  - $X_B = \{X_1 = 1; X_2 = -1.2\}$



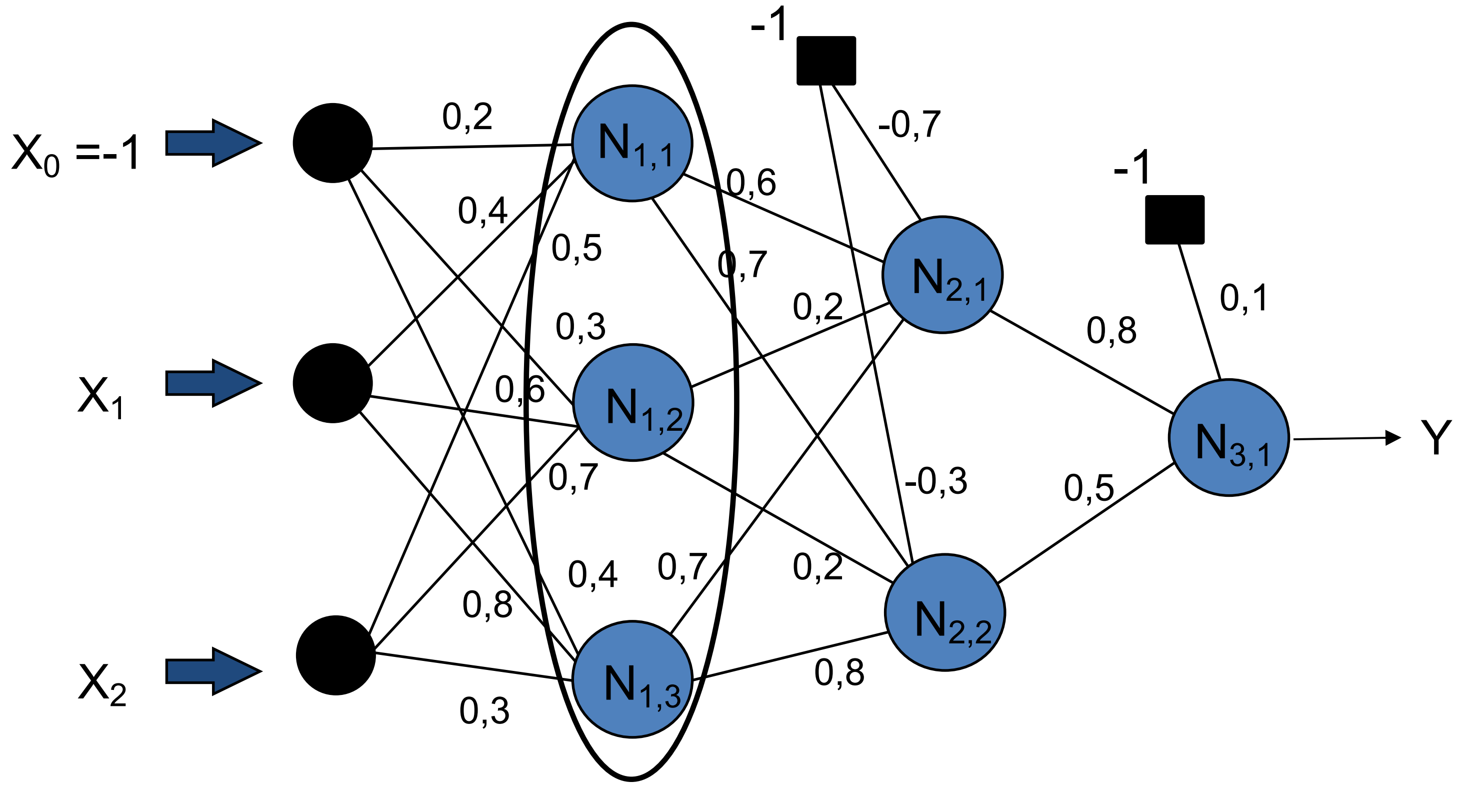
Input	Weight	$v_{1,1}$
-1	0,2	-0,4
1	0,4	
-1,2	0,5	

Input	Weight	$v_{1,2}$
-1	0,3	-0,54
1	0,6	
-1,2	0,7	

Input	Weight	$v_{1,3}$
-1	0,4	0,04
1	0,8	
-1,2	0,3	

Verification of Covering Methods

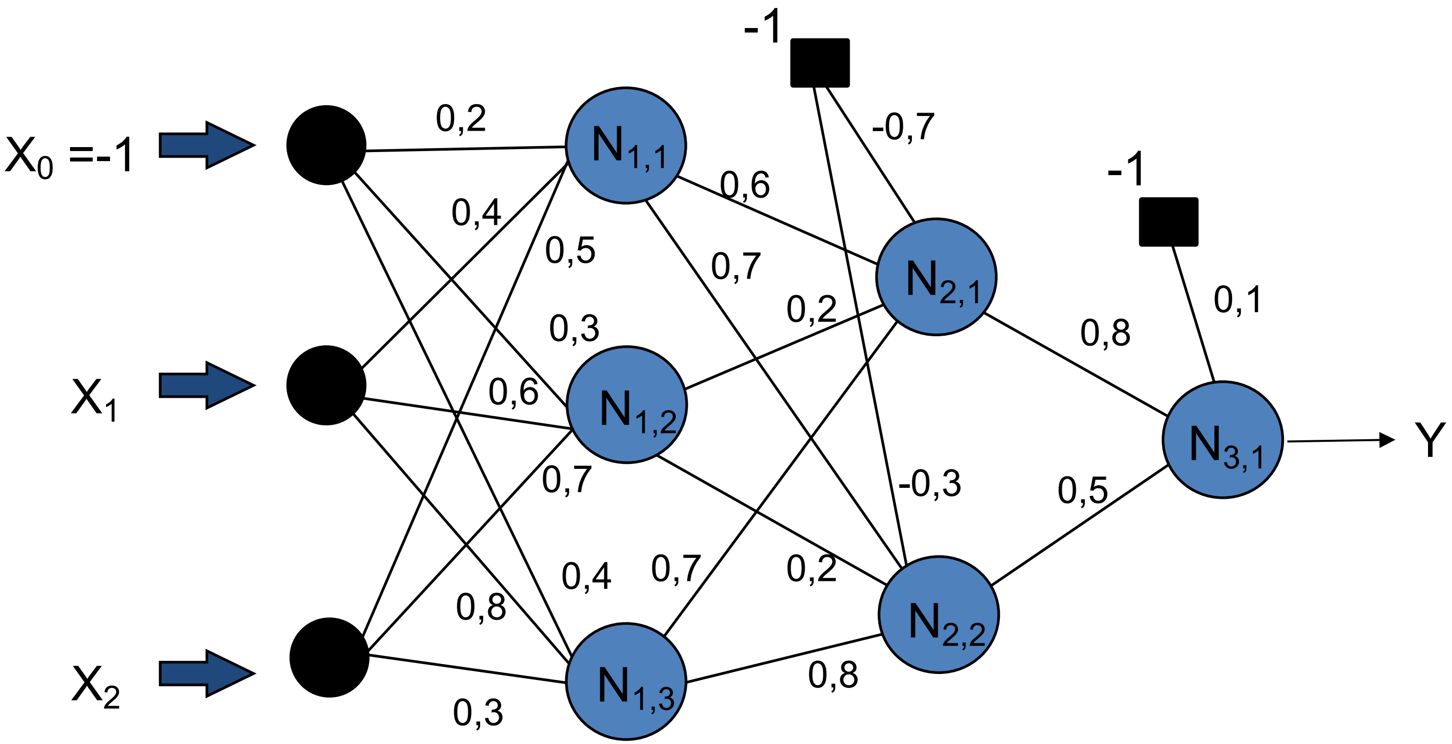
- Value change (vc): Suposing the boolean h(x,y) = x – y > 0.13. Then, $vc(h, n_{1,2}, X_A, X_B)$ = true;



$V^1(X_1)$	$V^1(X_2)$	$h(v_{n,l}(X_1), v_{n,l}(X_2))$
-0,3	-0,4	**False**
-0,4	-0,54	**True**
0,1	0,04	**False**

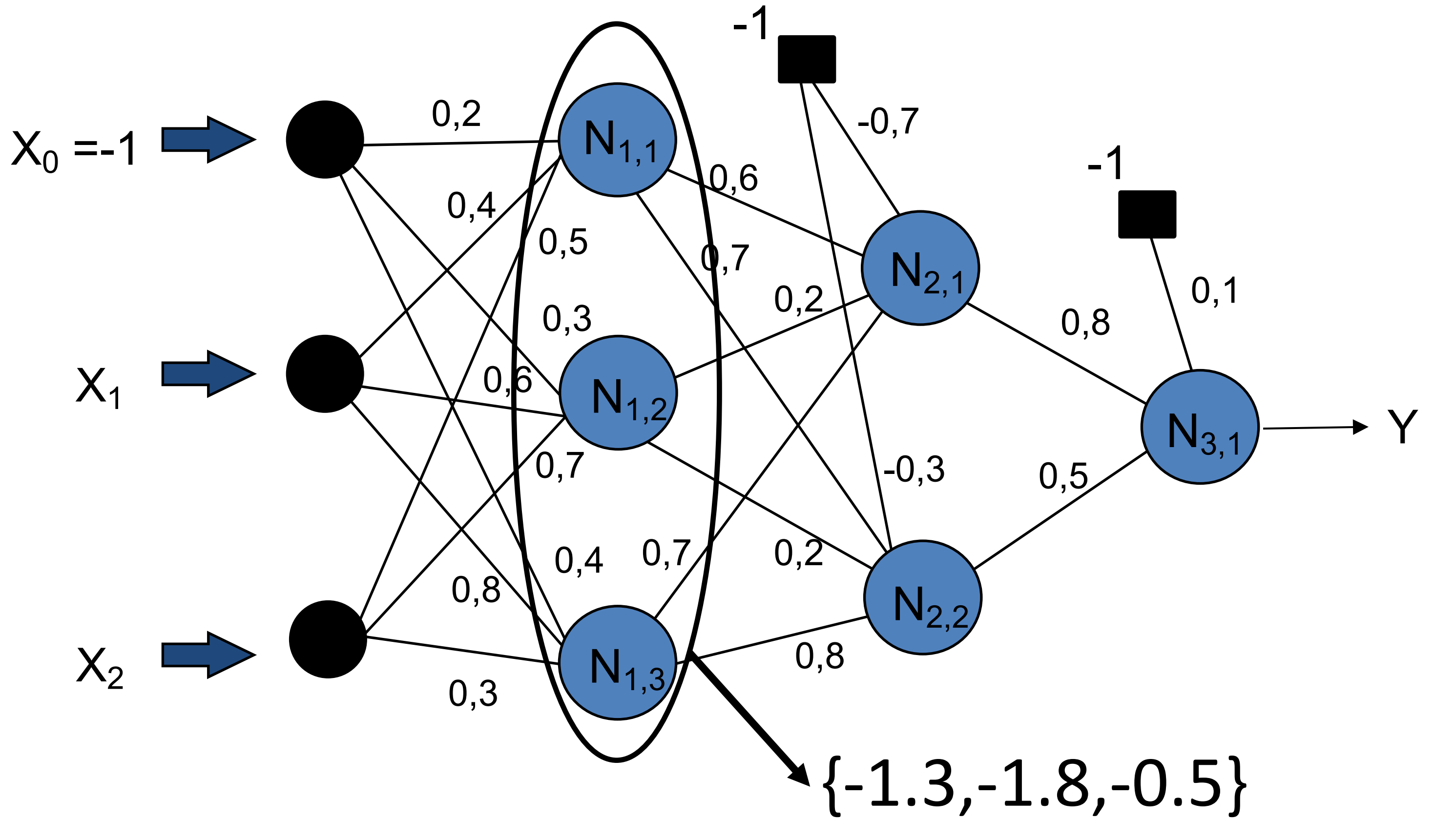
Verification of Covering Methods

- Distance change (dc) occurs when there is no sign change in every neuron of a layer, but all neurons shows a value change with respect to a metric g.



Verification of Covering Methods

- Distance change (dc):
  - $X_A = \{X_1 = 1; X_2 = -3\}$



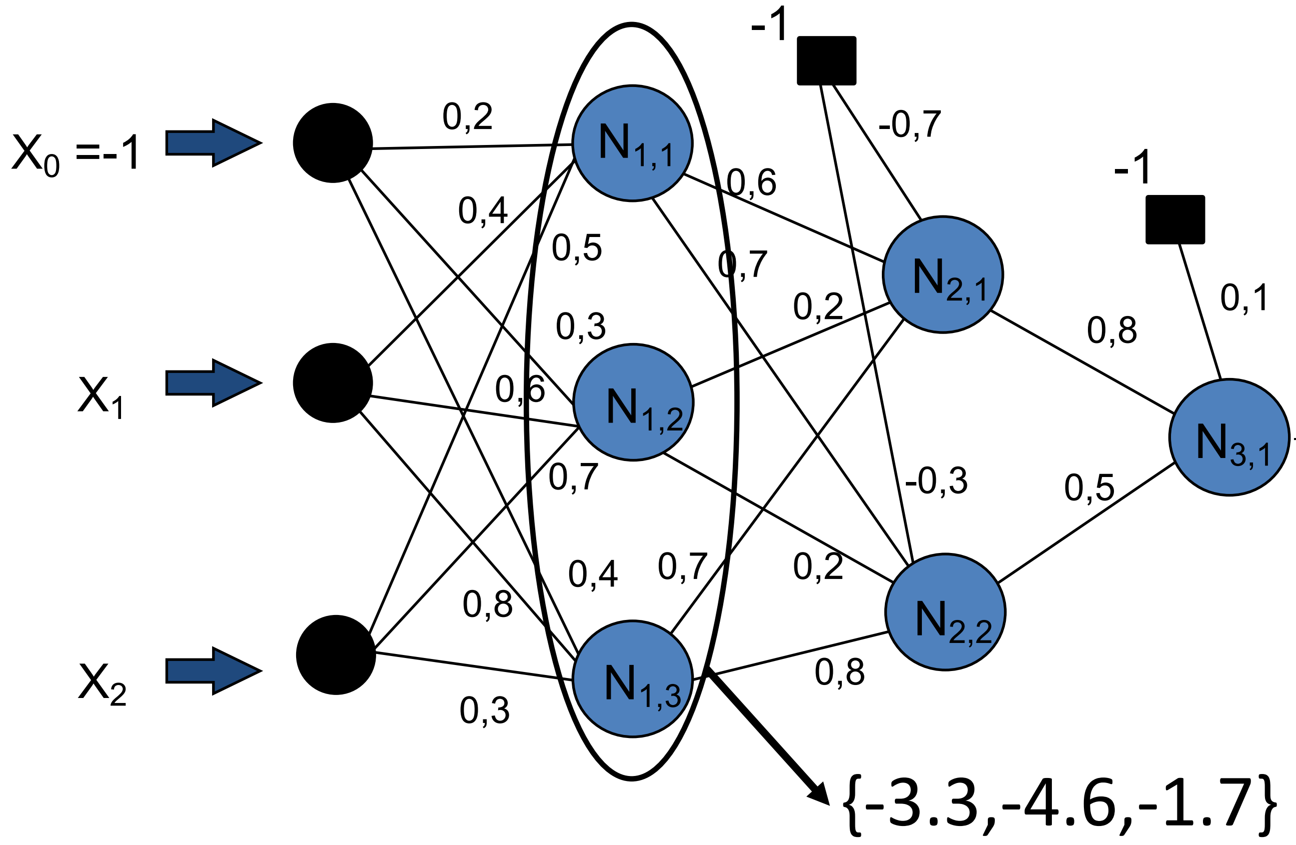
Input	Weight	$v_{1,1}$
-1	0,2	-1,3
1	0,4	
-3	0,5	

Input	Weight	$v_{1,2}$
-1	0,3	-1,8
1	0,6	
-3	0,7	

Input	Weight	$v_{1,3}$
-1	0,4	-0,5
1	0,8	
-3	0,3	

Verification of Covering Methods

- Distance change (dc):
  - $X_B = \{X_1 = 1; X_2 = -7\}$



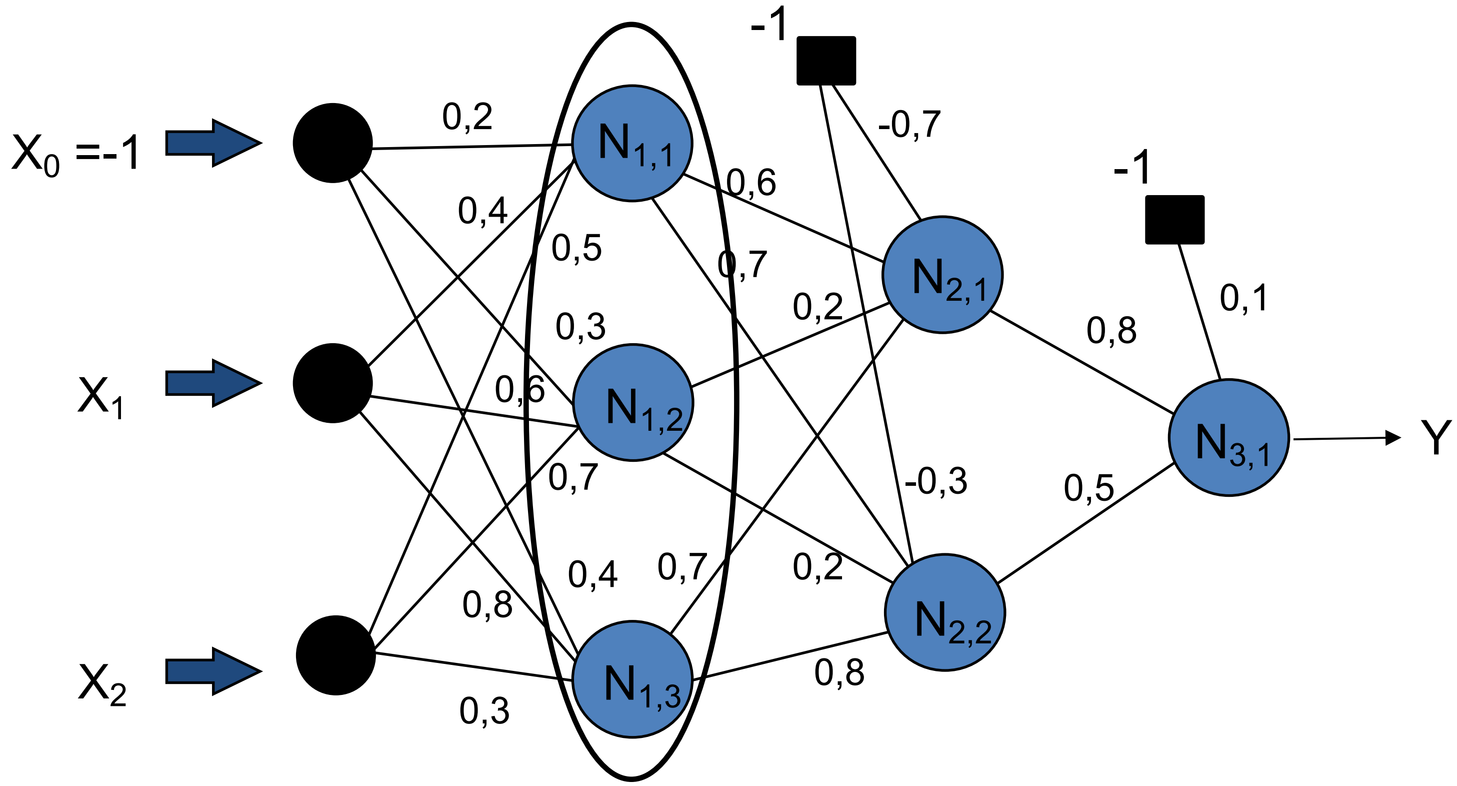
Input	Weight	$v_{1,1}$
-1	0,2	-3.3
1	0,4	
-7	0,5	

Input	Weight	$v_{1,2}$
-1	0,3	-4,6
1	0,6	
-7	0,7	

Input	Weight	$v_{1,3}$
-1	0,4	-1,7
1	0,8	
-7	0,3	

Verification of Covering Methods

- Distance change (dc): Suposing the boolean $g(I_k)$ = $EuclidianDistance(I_k) > 0.1$. Then, $dc(g, I_1, X_A, X_B)$ = true;



$V^1(X_1)$	$V^1(X_2)$	$g(I_1)$
-1,3	-3,3	**True**
-1,8	-4,6	
-0,5	-1,7	

Verification of Covering Methods

- Covering Methods are used to mesure how adversarial can be a pair or a set of inputs to the ANN neurons. They are sectioned in four methods:

- SS-Cover:

  $sc(n_{k,i}, x_1, x_2);$

  $\neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k;$

  $sc(n_{k+1,j}, x_1, x_2);$

- DS-Cover:

  $dc(g, k, x_1, x_2);$

  $sc(n_{k+1,j}, x_1, x_2);$

- SV-Cover:

  $sc(n_{k,i}, x_1, x_2);$

  $\neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k;$

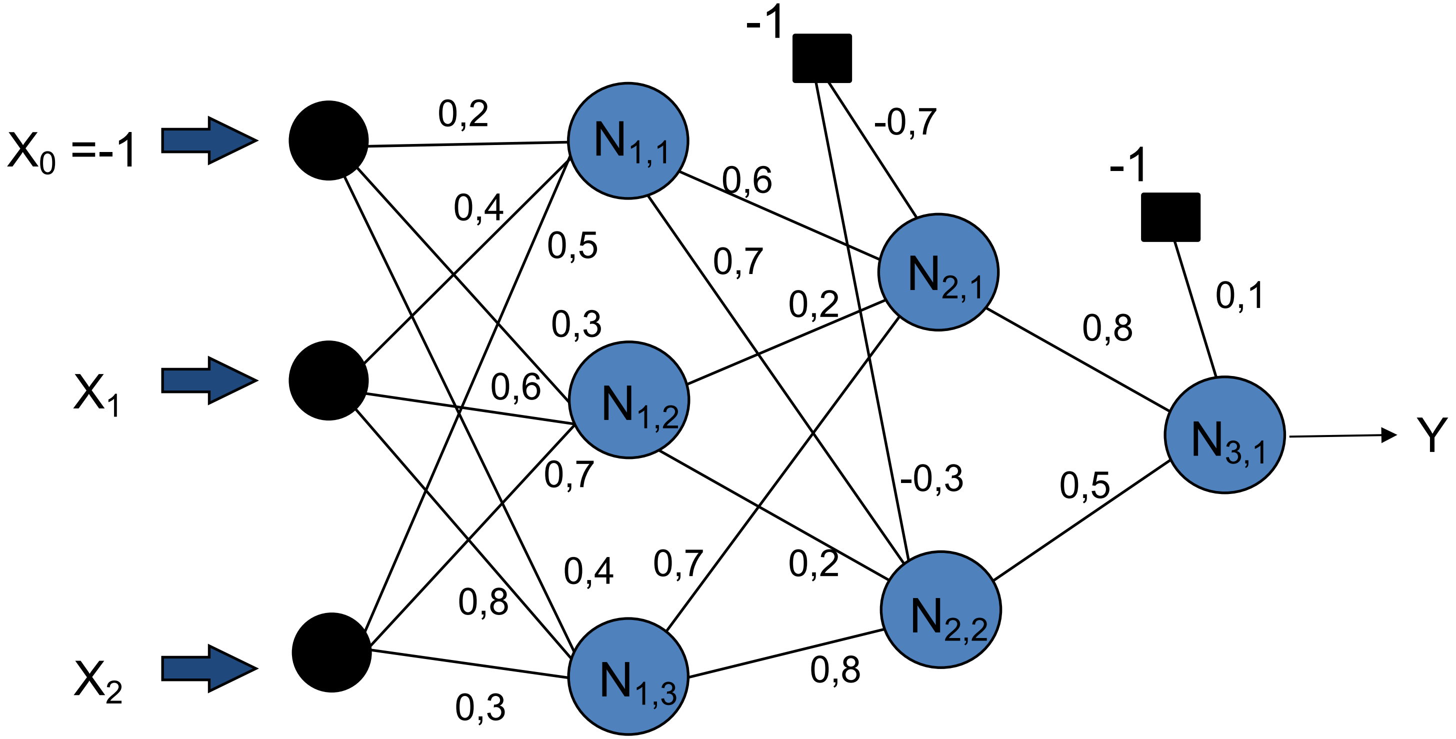
  $vc(h, n_{k+1,j}, x_1, x_2);$

- DV-Cover:

  $dc(g, k, x_1, x_2);$

  $vc(h, n_{k+1,j}, x_1, x_2);$

Verification of Covering Methods



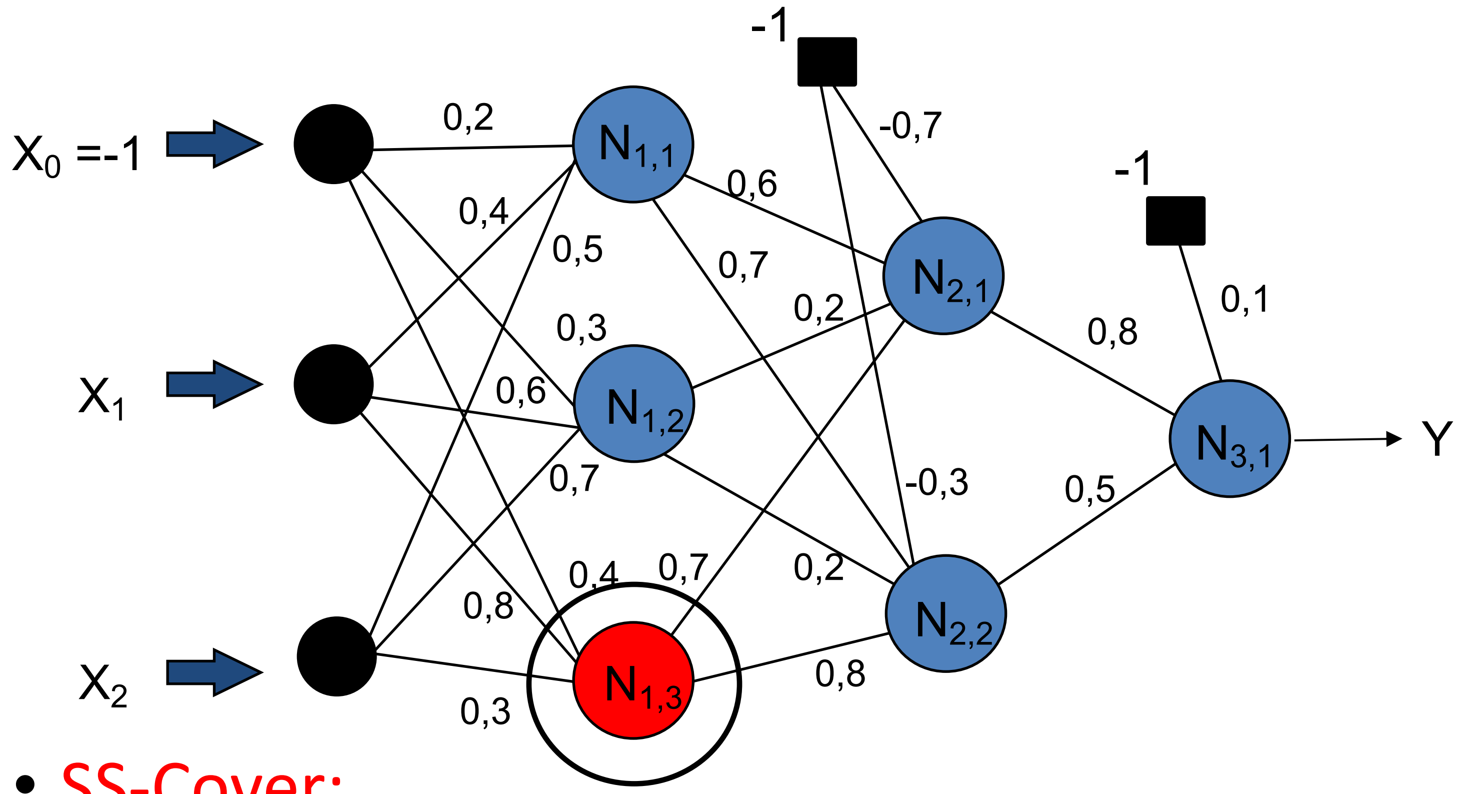
- $X_A = \{1, -1\}$
- $X_B = \{1, -3\}$

	X_A	X_B	SC
$N_{1,1}$	-1,3	-0,3	F
$N_{1,2}$	-1,8	-0,4	F
$N_{1,3}$	-0,5	0,1	T
$N_{2,1}$	-0.79	0,51	T
$N_{2,2}$	-1,37	0,09	T
$N_{3,1}$	-1,41	0,35	T

- SS-Cover:

$$sc(n_{k,i}, x_1, x_2);$$
$$\neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k;$$
$$sc(n_{k+1,j}, x_1, x_2);$$

Verification of Covering Methods



Layer1	X_A	X_B	SC
$N_{1,1}$	-1,3	-0,3	F
$N_{1,2}$	-1,8	-0,4	F
$N_{1,3}$	-0,5	0,1	T

Layer2	X_A	X_B	SC
$N_{2,1}$	-0.79	0,51	T
$N_{2,2}$	-1,37	0,09	T

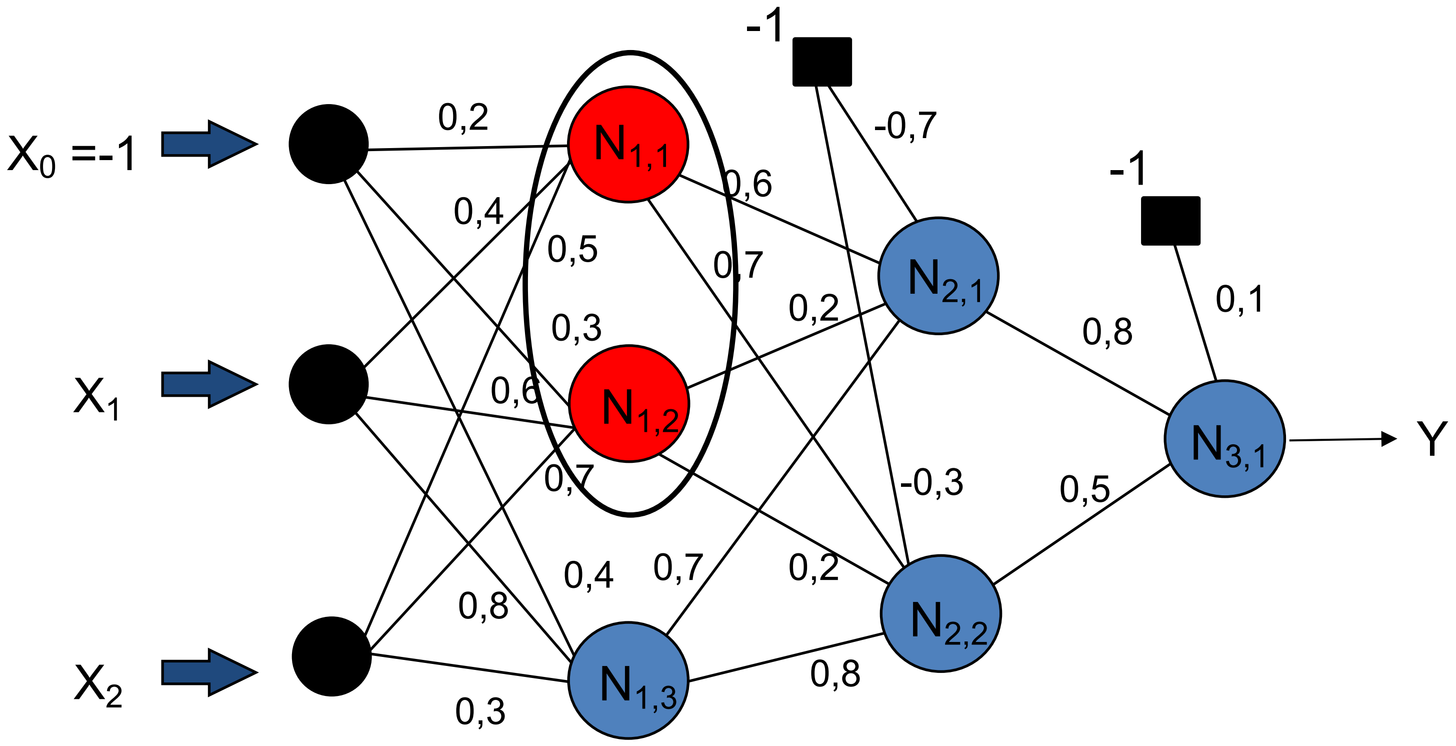
- SS-Cover:

  $sc(n_{k,i}, x_1, x_2)$;

  $\neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k$;

  $sc(n_{k+1,j}, x_1, x_2)$;

Verification of Covering Methods



Layer1	X_A	X_B	SC
$N_{1,1}$	-1,3	-0,3	F
$N_{1,2}$	-1,8	-0,4	F
$N_{1,3}$	-0,5	0,1	T

Layer2	X_A	X_B	SC
$N_{2,1}$	-0.79	0,51	T
$N_{2,2}$	-1,37	0,09	T

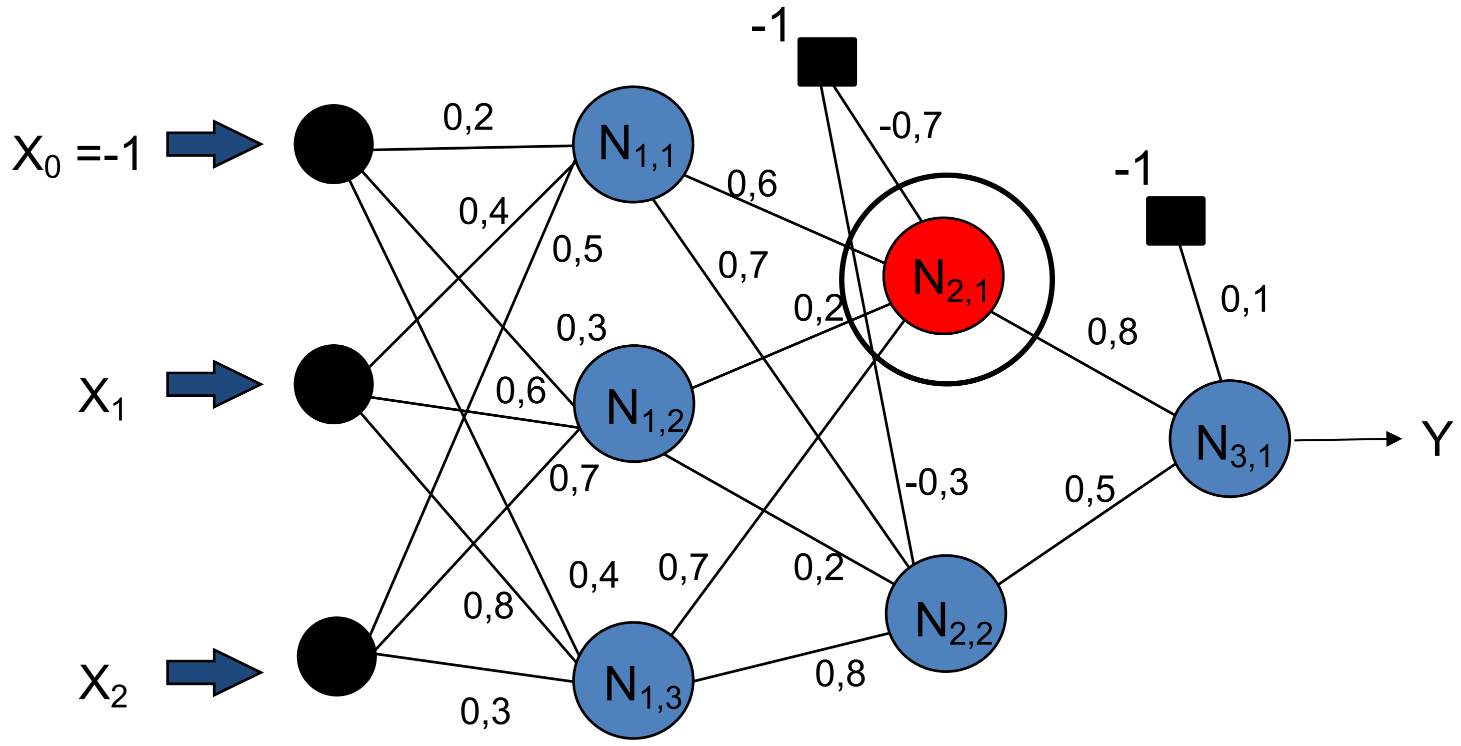
- SS-Cover:

  $sc(n_{k,i}, x_1, x_2);$

  $\neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k;$

  $sc(n_{k+1,j}, x_1, x_2);$

Verification of Covering Methods



Layer1	X_A	X_B	SC
$N_{1,1}$	-1,3	-0,3	F
$N_{1,2}$	-1,8	-0,4	F
$N_{1,3}$	-0,5	0,1	T

Layer2	X_A	X_B	SC
$N_{2,1}$	-0.79	0,51	T
$N_{2,2}$	-1,37	0,09	T

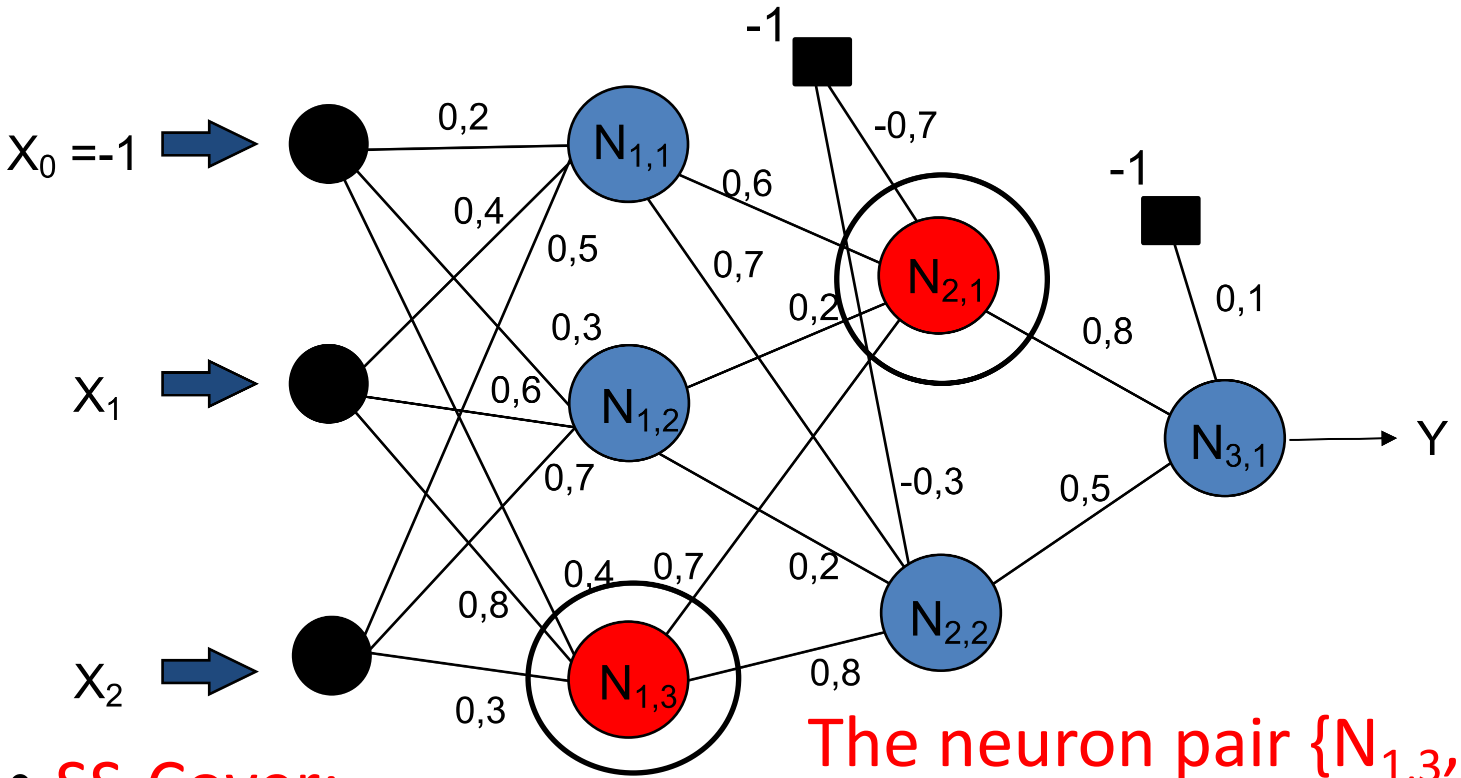
- SS-Cover:

  $sc(n_{k,i}, x_1, x_2)$;

  $\neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k$;

  $sc(n_{k+1,j}, x_1, x_2)$;

Verification of Covering Methods



Layer1	X_A	X_B	SC
$N_{1,1}$	-1,3	-0,3	F
$N_{1,2}$	-1,8	-0,4	F
$N_{1,3}$	-0,5	0,1	T

Layer2	X_A	X_B	SC
$N_{2,1}$	-0.79	0,51	T
$N_{2,2}$	-1,37	0,09	T

The neuron pair $\{N_{1,3}, N_{2,1}\}$ is SS-covered by X_A and X_B

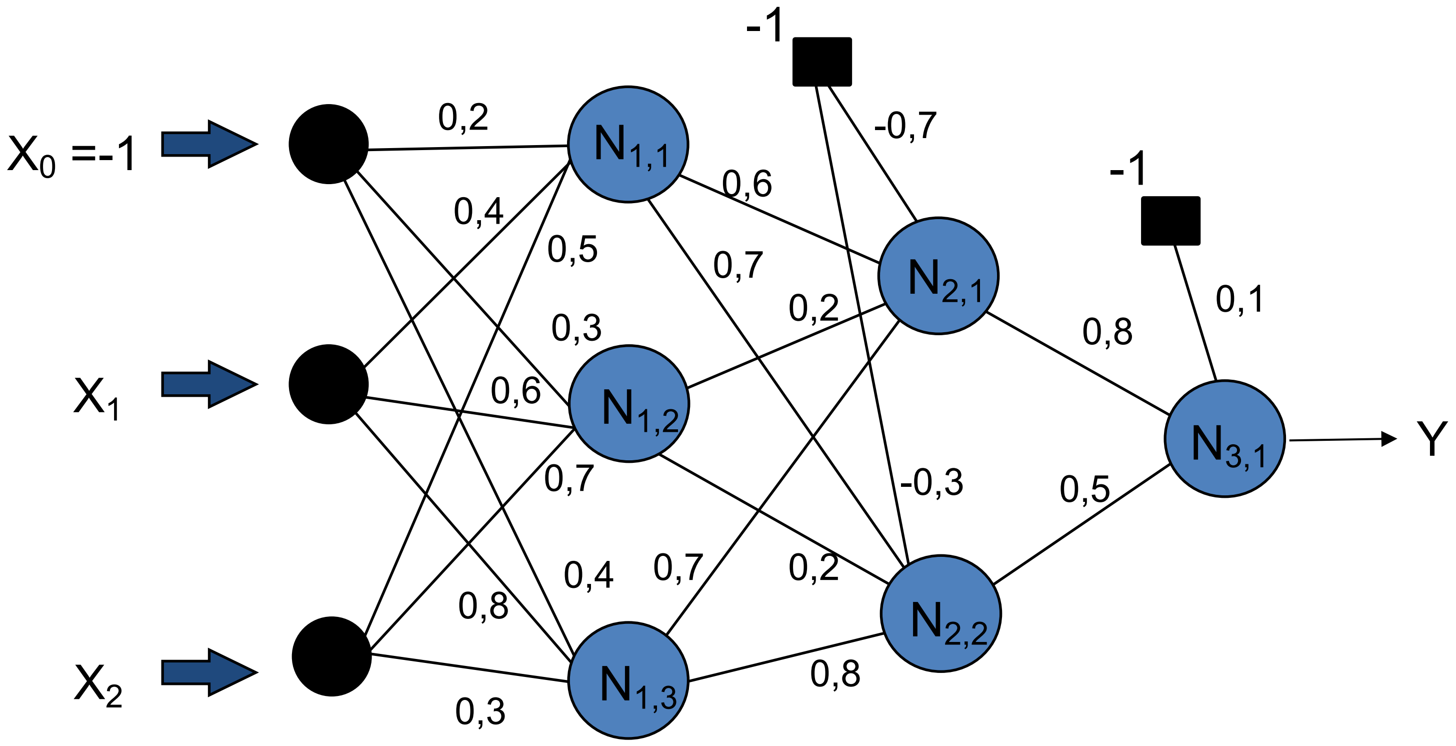
- SS-Cover:

  $sc(n_{k,i}, x_1, x_2)$;

  $\neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k$;

  $sc(n_{k+1,j}, x_1, x_2)$;

Verification of Covering Methods



- $X_A = \{1, -3\}$
- $X_B = \{1, -7\}$

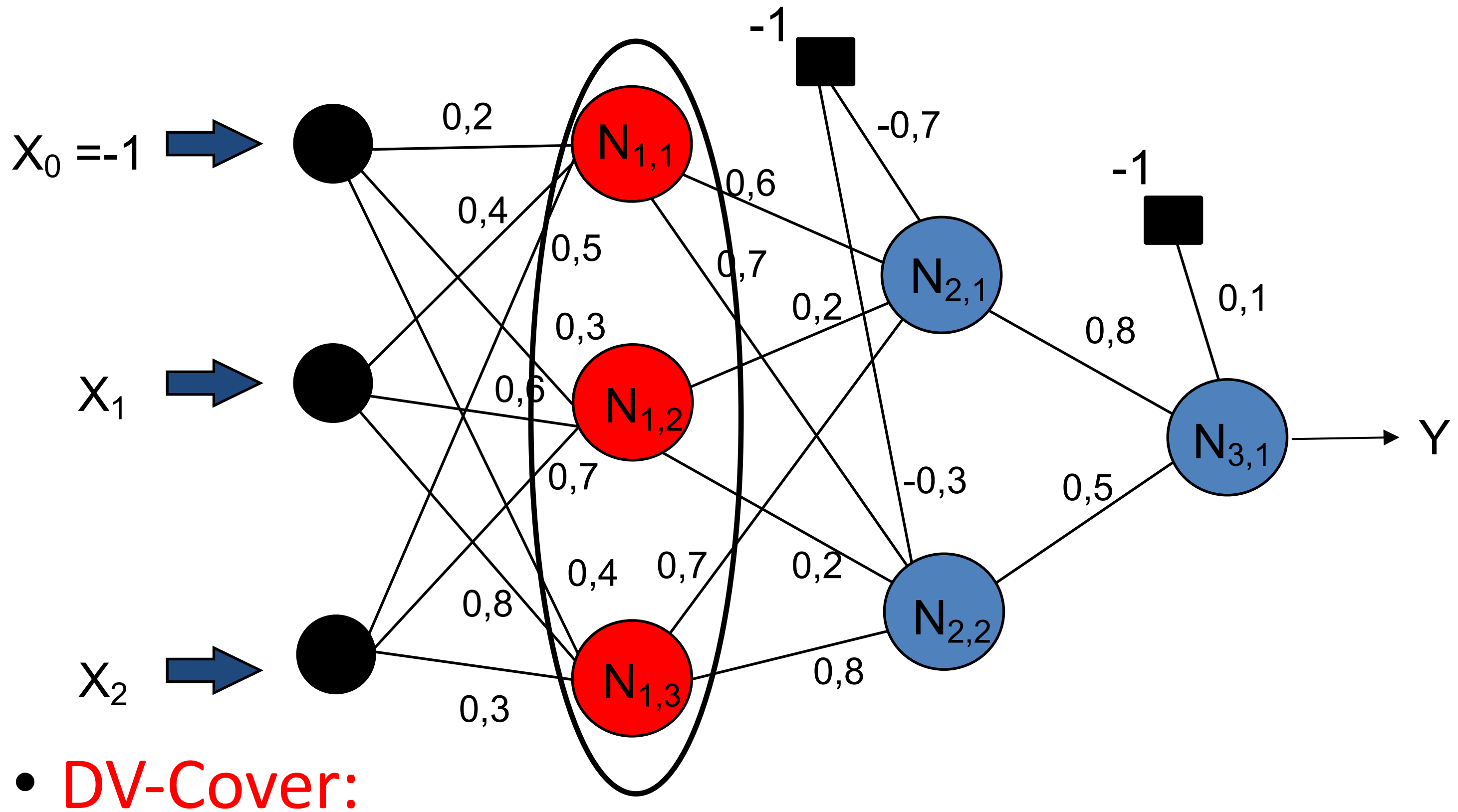
	X_A	X_B	SC
$N_{1,1}$	-1,3	-3,3	F
$N_{1,2}$	-1,8	-4,6	F
$N_{1,3}$	-0,5	-1,7	F
$N_{2,1}$	-0.79	-3.39	F
$N_{2,2}$	-1,37	-4,29	F
$N_{3,1}$	-1,41	-4,95	F

- DV-Cover:

  $dc(g, k, x_1, x_2);$

  $vc(h, n_{k+1,j}, x_1, x_2);$

Verification of Covering Methods



Layer1	X_A	X_B	SC
$N_{1,1}$	-1,3	-3,3	F
$N_{1,2}$	-1,8	-4,6	F
$N_{1,3}$	-0,5	-1,7	F

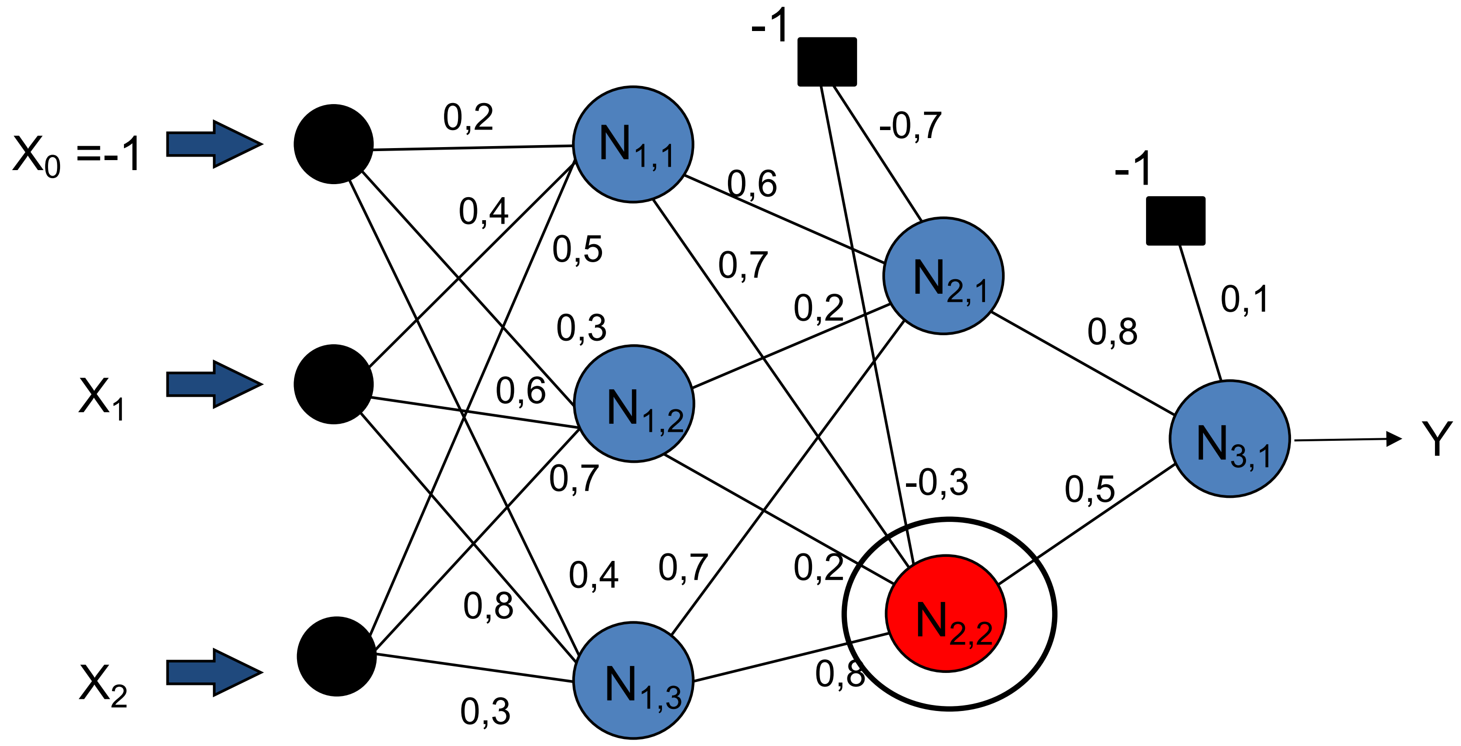
Layer2	X_A	X_B	SC
$N_{2,1}$	-0.79	-3.39	F
$N_{2,2}$	-1,37	-4,29	F

- DV-Cover:

  $dc(g, k, x_1, x_2);$

  $vc(h, n_{k+1,j}, x_1, x_2);$

Verification of Covering Methods



Layer1	X_A	X_B	SC
$N_{1,1}$	-1,3	-3,3	F
$N_{1,2}$	-1,8	-4,6	F
$N_{1,3}$	-0,5	-1,7	F

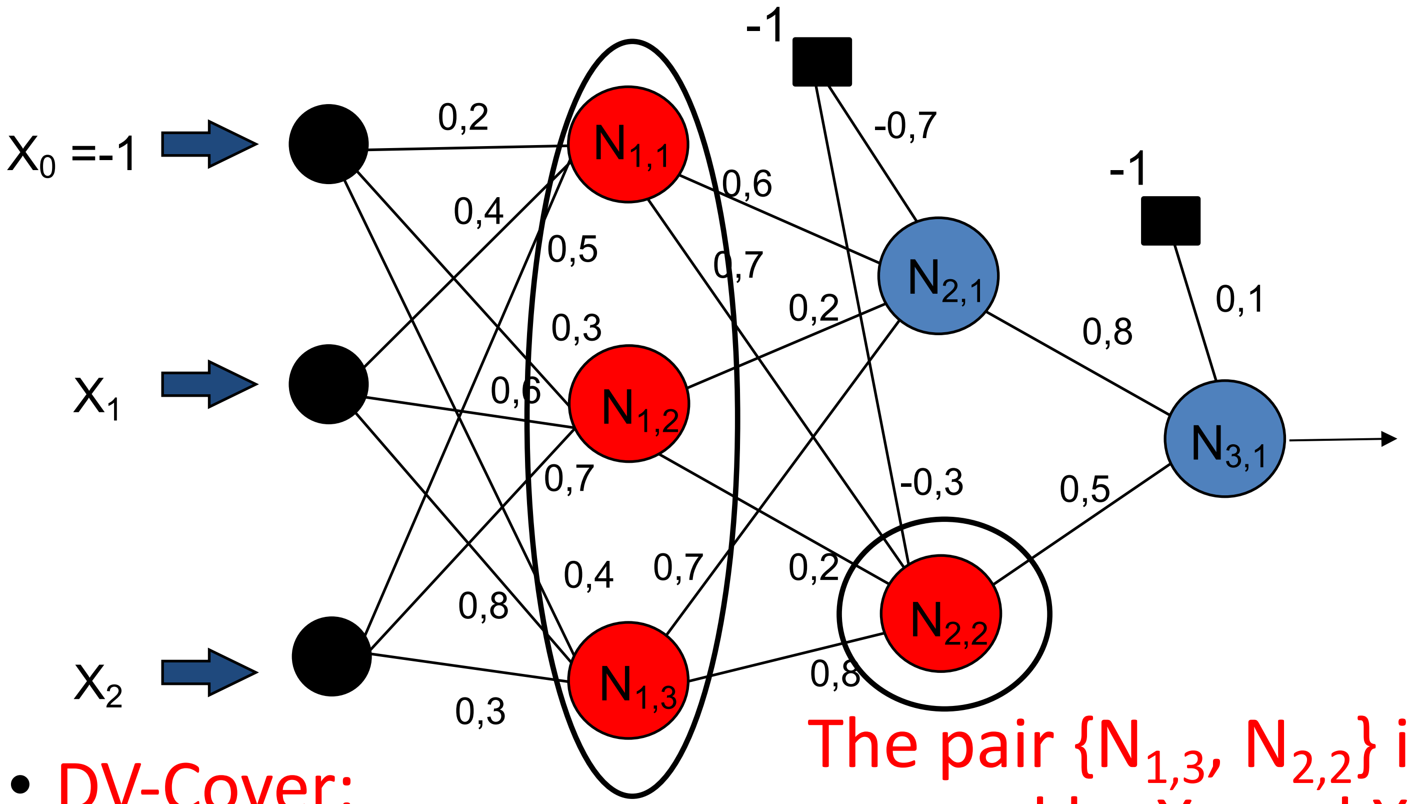
Layer2	X_A	X_B	SC
$N_{2,1}$	-0.79	-3.39	F
$N_{2,2}$	-1,37	-4,29	F

- DV-Cover:

  dc(g, k, x_1, x_2);

  vc(h, $n_{k+1,j}$, x_1, x_2);

Verification of Covering Methods



The pair $\{N_{1,3}, N_{2,2}\}$ is DV-covered by X_A and X_B

Layer1	X_A	X_B	SC
$N_{1,1}$	-1,3	-3,3	F
$N_{1,2}$	-1,8	-4,6	F
$N_{1,3}$	-0,5	-1,7	F

Layer2	X_A	X_B	SC
$N_{2,1}$	-0.79	-3.39	F
$N_{2,2}$	-1,37	-4,29	F

- DV-Cover:
  $dc(g, k, x_1, x_2)$;
  $vc(h, n_{k+1,j}, x_1, x_2)$;

Verification of Adversarial Cases

- Conditions for adversarial cases:

$$I^d \in \mathcal{D}^{m \times n},\ I \in \mathbb{M}^{m \times n},\ \mathcal{D}^{m \times n} \subseteq \mathbb{M}^{m \times n}$$

- Restriction:

$$R = \{I \in \mathbb{M}^{mn} \mid \delta(I^d, \mathcal{I}) \leq b\}$$

- Adversarial case definition:

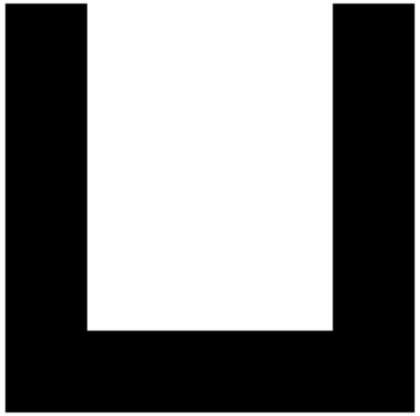
$$\mathcal{Y}^d = \{N(\mathcal{I})\ ,\ \forall\ \mathcal{I} \in \mathcal{R}\}$$

Verification of Adversarial Cases

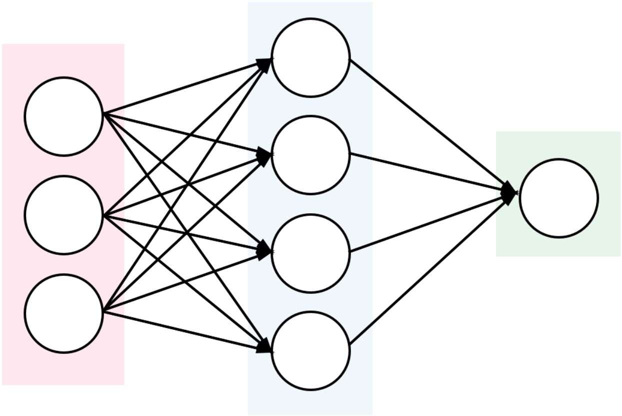
Non-Deterministic

$$R = \{I \in \mathbb{M}^{mn} \mid \delta(I^d, \mathcal{I}) \leq b\}$$

Assume the euclidian distance



Base image



ANN model

$$\mathcal{Y}^d = \{N(\mathcal{I})\ ,\ \forall\ \mathcal{I} \in \mathcal{R}\}$$

Checks if output corresponds to the base image label

Experimental Evaluation

Benchmark Description:
-ANN that solves the problem of vocalic recognition;
-Dataset composed by: 100 vocalics with noises; 100 characters;
-Backpropagation
-Cross-validation

Experimental Evaluation

- Experiments were conducted on a 8-core 3.40GHz Intel Core i7 with 24 GB of RAM and Linux OS.
- Frameworks versions:

    -CUDA v9.0, cuDNN v5.0, cuBLAS v10.1, and ESBMC-GPU v2.0.

Experimental Objectives

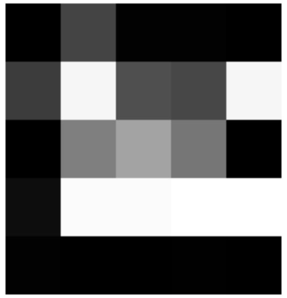
1. Evaluate the performance and correctness of our symbolic verification algorithms to check all four covering methods;
2. Evaluate the performance and correctness of our verification algorithm checkNN to verify adversarial cases obtained from changing input images and parameter proximity.

Results

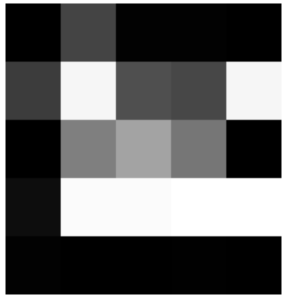
- The four properties specify that 80% of all neurons must be covered by a set of inputs.
- Verification time of the dataset w.r.t all 4 covering methods is around 20 minutes. It correctly verified the 4 properties.
- The verification of two inputs did not take more than a few seconds.

Results

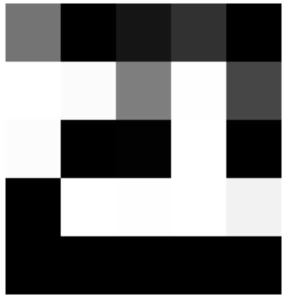
- Adversarial cases for label “E” missclassified as label “O”:



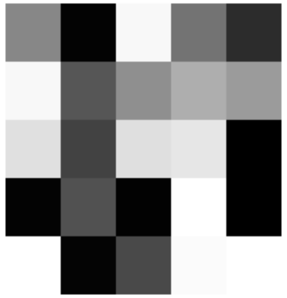
b = 0.5



b = 1.5



b = 2.5



b = 3.5

- Label “U” missclassified as label “O”:

b = 0.3

b = 0.5

b = 1.0

b = 1.5

Results

Benchmark	Image	λ	Verification Time (hours)
Ex1	Vocalic O	0.5	1
Ex2	Vocalic O	1.5	4
Ex3	Vocalic O	2.5	8
Ex4	Vocalic O	3.5	6
Ex5	Vocalic E	0.5	25
Ex6	Vocalic E	0.7	25
Ex7	Vocalic E	1.5	14
Ex8	Vocalic E	3.0	12
Ex9	Vocalic U	0.3	6
Ex10	Vocalic U	0.5	5
Ex11	Vocalic U	1.0	20
Ex12	Vocalic U	1.5	19
Ex13	Vocalic A	1.0	63
Ex14	Vocalic A	1.5	63

Conclusion

- Our verification method is able to find adversarial cases for different input images and proximity parameter values;
- Our approach exhaustively verifies all possible adversarial cases inside the proximity parameter b;
- The verification of covering methods is able to verify our dataset correctly. The dataset is not able to cover 80% of neuron w.r.t. a covering method;
- There is a high verification time in some benchmarks due to bitaccurate verification.

Future Work

- Support convolutional layers;
- Implement some techniques as invariant inference to prune the state space exploration;
- Investigate fault localization and repair techniques;

Questions?

Luiz Henrique Coelho Sena
lhcs@icomp.ufam.edu.br
Universidade Federal do Amazonas