The BeMi Stardust: a Structured Ensemble of Binarized Neural Networks

LION17 - The 2023 Learning and Intelligent OptimizatioN Conference

Ambrogio Maria Bernardelli[1], Stefano Gualandi[1], Hoong Chuin Lau[2], Simone Milanesi[1]

[1]Department of Mathematics - University of Pavia
[2]School of Information Systems - Singapore Management University

2023-06-07, Nice

UNIVERSITÀ DI PAVIA

Università della Svizzera italiana

Overview

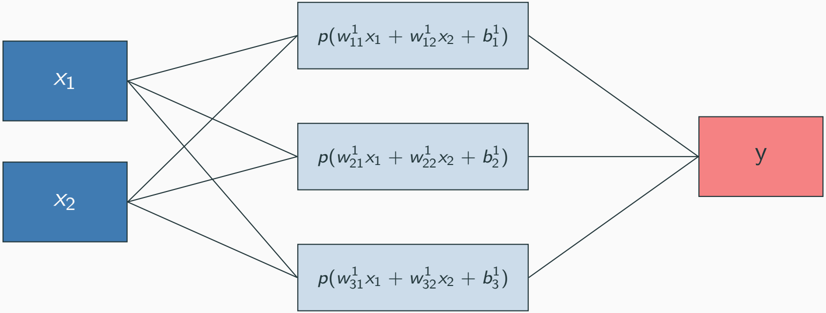
Introduction

Neural Networks - NNs

$$y = p \circ T_L \circ p \circ T_{L-1} \circ \cdots \circ p \circ T_2 \circ p \circ T_1(x)$$

where $T_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$, $T_\ell(x) = W^\ell x + b^\ell \;\; \forall \ell \in \{1, \dots, L\}$

$p : \mathbb{R} \to \mathbb{R}$ non-linear, applied component-wise.



Binarized Neural Networks - BNNs

BNNs are getting increasing attention thanks to their compactness and versatility.

In this kind of NN, every neuron $j \in N_l$ is connected to every neuron $i \in N_{l-1}$ by a weight $w_{ilj} \in \{-1, 0, 1\}$. Given a value x for input neurons, the preactivation $a_{lj}(x)$ of neuron $j \in N_l$ and the activation $p_j(x)$ are, respectively,

$$a_{lj}(x) = \sum_{i \in N_{l-1}} w_{ilj} \cdot p_{(l-1)i}(x) \qquad \text{and} \qquad p_{lj}(x) = \begin{cases} x_j & \text{if } l = 0, \\ +1 & \text{if } l > 0, a_{lj}(x) \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

Binarized Neural Networks - BNNs

BNNs are getting increasing attention thanks to their compactness and versatility.
In this kind of NN, every neuron $j \in N_l$ is connected to every neuron $i \in N_{l-1}$ by a weight $w_{ilj} \in \{-1, 0, 1\}$. Given a value x for input neurons, the preactivation $a_{lj}(x)$ of neuron $j \in N_l$ and the activation $p_j(x)$ are, respectively,

$$a_{lj}(x) = \sum_{i \in N_{l-1}} w_{ilj} \cdot p_{(l-1)i}(x) \quad \text{and} \quad p_{lj}(x) = \begin{cases} x_j & \text{if } l = 0, \\ +1 & \text{if } l > 0, a_{lj}(x) \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

Recent works[1] show that this kind of networks are hard to train with GD-based algorithms in a context of few-shot learning. Instead, MILP approaches are being researched.

[1] Toro Icarte, R., Illanes, L., Castro, M.P., Cire, A.A., McIlraith, S.A. and Beck, J.C.: Training binarized neural networks using MIP and CP. In: Proceedings of CP'19. vol 11802, pp. 401–417. Springer (2019).

Mathematical Models

Models

We use three different approaches during the training:

Models

We use three different approaches during the training:

(M-M) Max-Margin: a way of finding robust BNNs by maximizing the margins of their neurons. Intuitively, neurons with larger margins requires bigger changes on their inputs and weights to change their activation values;[1]

[1] Toro Icarte, R., Illanes, L., Castro, M.P., Cire, A.A., McIlraith, S.A. and Beck, J.C.: Training binarized neural networks using MIP and CP. In: Proceedings of CP'19. vol 11802, pp. 401–417. Springer (2019).

Models

We use three different approaches during the training:

(`M-M`) Max-Margin: a way of finding robust BNNs by maximizing the margins of their neurons. Intuitively, neurons with larger margins requires bigger changes on their inputs and weights to change their activation values;[1]

(`M-W`) Min-Weight: a way of finding simple BNNs by minimizing the number of connections;[1]

[1] Toro Icarte, R., Illanes, L., Castro, M.P., Cire, A.A., McIlraith, S.A. and Beck, J.C.: Training binarized neural networks using MIP and CP. In: Proceedings of CP'19. vol 11802, pp. 401–417. Springer (2019).

Models

We use three different approaches during the training:

(M-M) Max-Margin: a way of finding robust BNNs by maximizing the margins of their neurons. Intuitively, neurons with larger margins requires bigger changes on their inputs and weights to change their activation values;[1]

(M-W) Min-Weight: a way of finding simple BNNs by minimizing the number of connections;[1]

(S-M) Sat-Margin: a way of finding BNNs by maximizing the number of correct predictions. At the same time each correctly predicted sample is confidently predicted.[2]

[1] Toro Icarte, R., Illanes, L., Castro, M.P., Cire, A.A., McIlraith, S.A. and Beck, J.C.: Training binarized neural networks using MIP and CP. In: Proceedings of CP'19. vol 11802, pp. 401–417. Springer (2019).

[2] Thorbjarnarson, T., Yorke-Smith, N.: On Training Neural Networks with Mixed Integer Programming. arXiv preprint arXiv:2009.03825 (2020).

An insight: Min-Weight (0-margin)

The MILP training of a BNN consists of finding a parameter configuration that satisfies a set of linear(izable) constraints - weak inequalities or equalities - and minimizes an objective function. This function encodes our beliefs into the network architecture.

In the case of `M-W`, we want a network that is as light as possible (while maintaining acceptable accuracy).

$$
\begin{aligned}
\min_{w,c,u} \quad & \sum_{l \in \mathcal{L}} \sum_{i \in N_{l-1}} \sum_{j \in N_l} v_{ilj} \\
\text{s.t.} \quad & \sum_{i \in N_{L-1}} c^k_{iLj} \geq 0 && \text{if } y^k_j = 1, \\
& \sum_{i \in N_{L-1}} c^k_{iLj} \leq -0 - \epsilon && \text{if } y^k_j = -1, \\
& u^k_{lj} = 1 \implies \sum_{i \in N_{l-1}} c^k_{ilj} \geq 0, \\
& u^k_{lj} = 0 \implies \sum_{i \in N_{l-1}} c^k_{ilj} \leq -0 - \epsilon, \\
& v_{ilj} = |w_{ilj}|, \\
& c^k_{i1j} = x^k_i w_{i1j}, \quad c^k_{ilj} = (2u^k_{lj} - 1) w_{ilj}, \\
& w_{ilj} \in \{-1, 0, 1\}, \; u^k_{lj}, v_{ilj} \in \{0, 1\}, \\
& c^k_{i1j} \in [-\mathfrak{b}, \mathfrak{b}], \; c^k_{ilj} \in \{-1, 0, 1\}.
\end{aligned}
$$

An insight: Max-Margint and Sat-Margin

$$
\begin{aligned}
\max_{w,c,u,m} \quad & \sum_{l \in \mathcal{L}} \sum_{j \in N_l} m_{lj} \\
\text{s.t.} \quad & \sum_{i \in N_{L-1}} c_{iLj}^k \geq m_{Lj} && \text{if } y_j^k = 1, \\
& \sum_{i \in N_{L-1}} c_{iLj}^k \leq -m_{Lj} - \epsilon && \text{if } y_j^k = -1, \\
& u_{lj}^k = 1 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \geq m_{lj}, \\
& u_{lj}^k = 0 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \leq -m_{lj} - \epsilon, \\
& c_{i1j}^k = x_i^k w_{i1j}, \quad c_{ilj}^k = (2u_{lj}^k - 1) w_{ilj}, \\
& w_{ilj} \in \{-1, 0, 1\}, \ u_{lj}^k \in \{0, 1\}, \\
& c_{i1j}^k \in [-\mathfrak{b}, \mathfrak{b}], \ c_{ilj}^k \in \{-1, 0, 1\}, \\
& m_{lj} \geq 0.
\end{aligned}
$$

An insight: Max-Margint and Sat-Margin

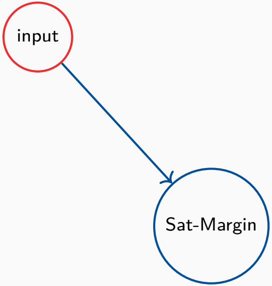
$$
\begin{aligned}
\max_{w,c,u,m} \quad & \sum_{l\in\mathcal{L}} \sum_{j\in N_l} m_{lj} \\
\text{s.t.} \quad & \sum_{i\in N_{L-1}} c^k_{iLj} \geq m_{Lj} && \text{if } y^k_j = 1, \\
& \sum_{i\in N_{L-1}} c^k_{iLj} \leq -m_{Lj} - \epsilon && \text{if } y^k_j = -1, \\
& u^k_{lj} = 1 \implies \sum_{i\in N_{l-1}} c^k_{ilj} \geq m_{lj}, \\
& u^k_{lj} = 0 \implies \sum_{i\in N_{l-1}} c^k_{ilj} \leq -m_{lj} - \epsilon, \\
& c^k_{i1j} = x^k_i w_{i1j}, \quad c^k_{ilj} = (2u^k_{lj} - 1) w_{ilj}, \\
& w_{ilj} \in \{-1, 0, 1\}, \; u^k_{lj} \in \{0, 1\}, \\
& c^k_{i1j} \in [-\mathfrak{b}, \mathfrak{b}], \; c^k_{ilj} \in \{-1, 0, 1\}, \\
& m_{lj} \geq 0.
\end{aligned}
$$

$$
\begin{aligned}
\max_{w,c,u,q,\hat{y}} \quad & \sum_{k\in T} \sum_{j\in N_L} q^k_j \\
\text{s.t.} \quad & q^k_j = 1 \implies \hat{y}^k_j \cdot y^k_j \geq \frac{1}{2}, \\
& q^k_j = 0 \implies \hat{y}^k_j \cdot y^k_j \leq \frac{1}{2} - \epsilon, \\
& \hat{y}^k_j = \frac{2}{N_{L-1}+1} \sum_{i\in N_{L-1}} c^k_{iLj}, \\
& u^k_{lj} = 1 \implies \sum_{i\in N_{l-1}} c^k_{ilj} \geq 0, \\
& u^k_{lj} = 0 \implies \sum_{i\in N_{l-1}} c^k_{ilj} \leq -\epsilon, \\
& c^k_{i1j} = x^k_i w_{i1j}, \quad c^k_{ilj} = (2u^k_{lj} - 1) w_{ilj}, \\
& w_{ilj} \in \{-1, 0, 1\}, \; u^k_{lj}, q^k_j \in \{0, 1\}, \\
& c^k_{i1j} \in [-\mathfrak{b}, \mathfrak{b}], \; c^k_{ilj} \in \{-1, 0, 1\}.
\end{aligned}
$$

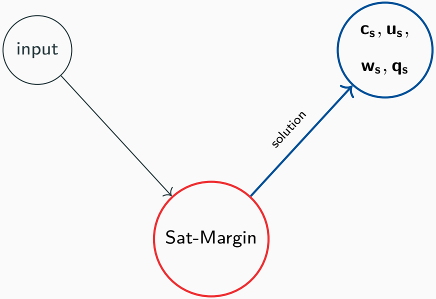
Methodology

Our model: Lexicographic multi-objective function

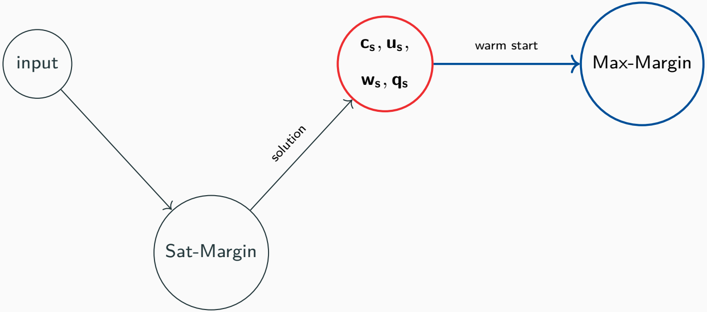
Our model: Lexicographic multi-objective function



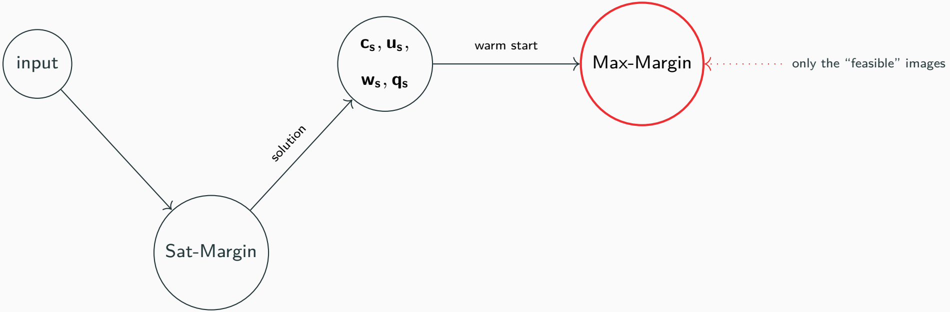
Our model: Lexicographic multi-objective function



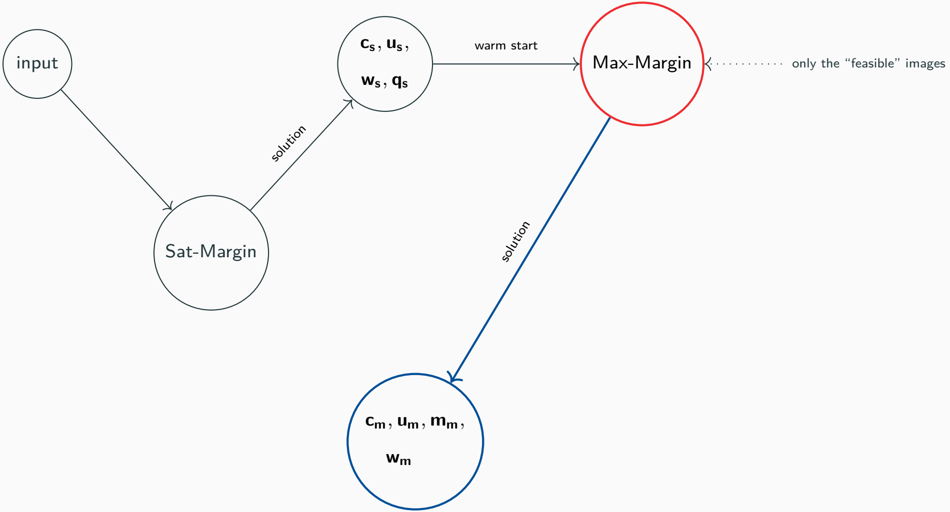
Our model: Lexicographic multi-objective function



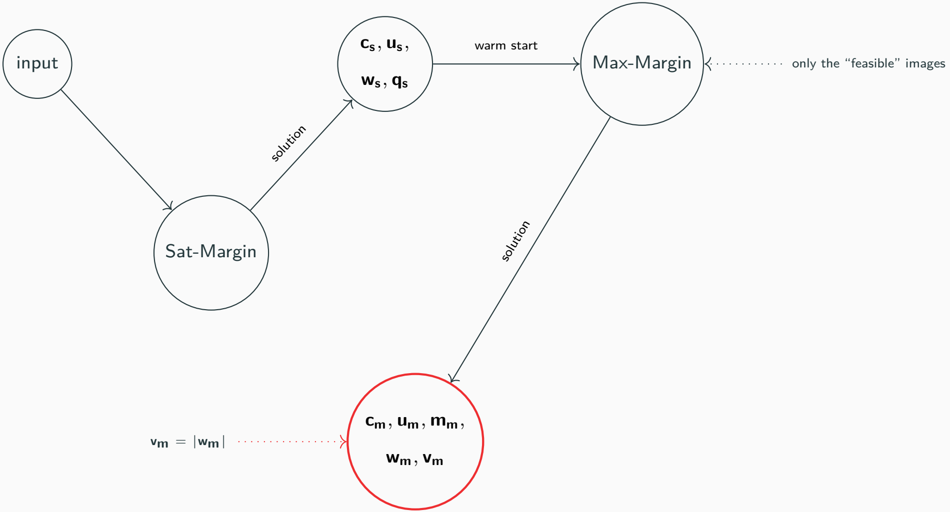
Our model: Lexicographic multi-objective function



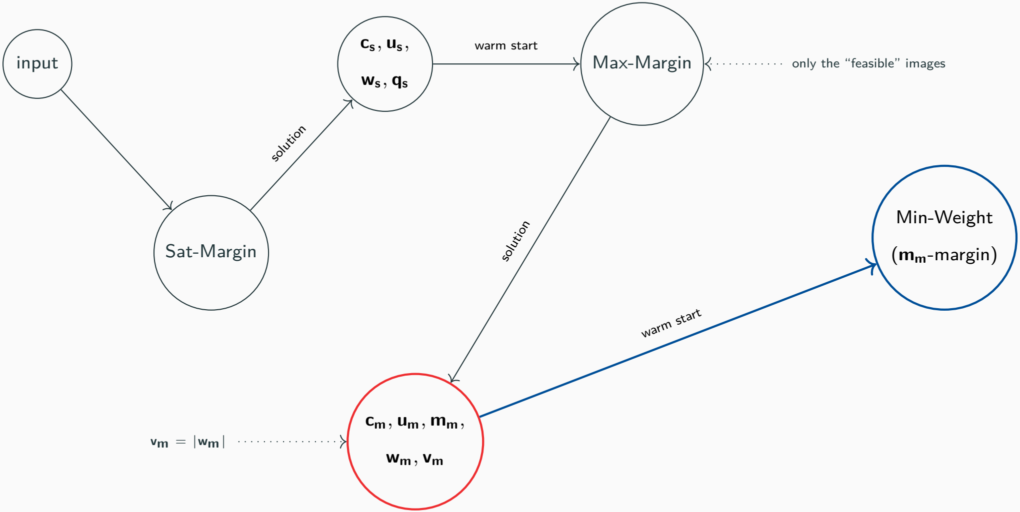
Our model: Lexicographic multi-objective function



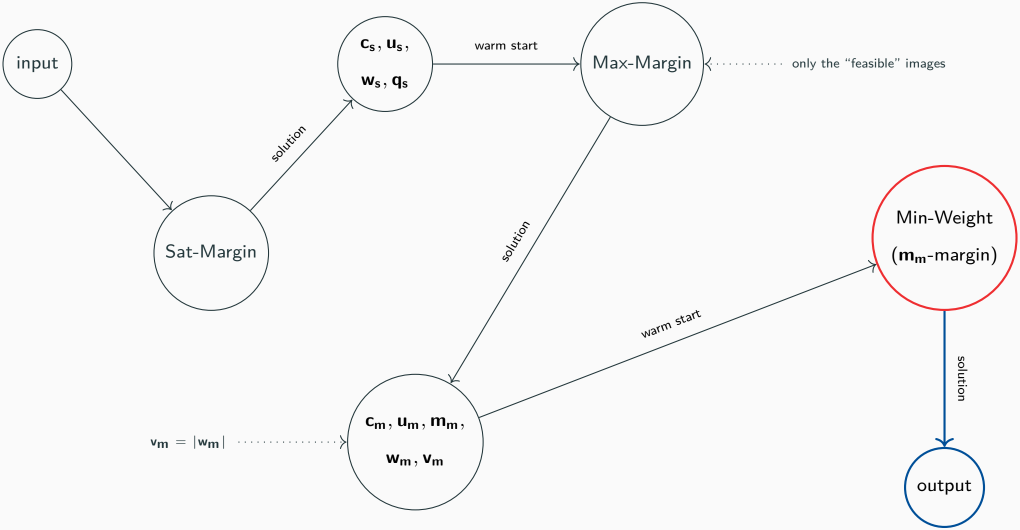
Our model: Lexicographic multi-objective function



Our model: Lexicographic multi-objective function



Our model: Lexicographic multi-objective function



Structured ensemble

Suppose we have a labelling problem and that our set of labels is $\mathcal{I}$.

Structured ensemble

Suppose we have a labelling problem and that our set of labels is $\mathcal{I}$.

We set a parameter $1 < p \leq n = |\mathcal{I}|$.

Structured ensemble

Suppose we have a labelling problem and that our set of labels is $\mathcal{I}$.

We set a parameter $1 < p \leq n = |\mathcal{I}|$.

We train one network $\mathcal{N}_{\mathcal{J}}$ for every $\mathcal{J} \in \mathcal{P}(\mathcal{I})_p$.

Structured ensemble

Suppose we have a labelling problem and that our set of labels is $\mathcal{I}$.

- We set a parameter $1 < p \leq n = |\mathcal{I}|$.
- We train one network $\mathcal{N}_{\mathcal{J}}$ for every $\mathcal{J} \in \mathcal{P}(\mathcal{I})_p$.
- When testing an input, we feed it to our list of networks $(\mathcal{N}_{\mathcal{J}})_{\mathcal{J} \in \mathcal{P}(\mathcal{I})_p}$ and we obtain a list of labels $(\mathfrak{e}_{\mathcal{J}})_{\mathcal{J} \in \mathcal{P}(\mathcal{I})_p}$.

Structured ensemble

Suppose we have a labelling problem and that our set of labels is $\mathcal{I}$.

- We set a parameter $1 < p \leq n = |\mathcal{I}|$.
- We train one network $\mathcal{N}_{\mathcal{J}}$ for every $\mathcal{J} \in \mathcal{P}(\mathcal{I})_p$.
- When testing an input, we feed it to our list of networks $(\mathcal{N}_{\mathcal{J}})_{\mathcal{J} \in \mathcal{P}(\mathcal{I})_p}$ and we obtain a list of labels $(\mathfrak{e}_{\mathcal{J}})_{\mathcal{J} \in \mathcal{P}(\mathcal{I})_p}$.
- We then apply a majority voting system.

Structured ensemble

Suppose we have a labelling problem and that our set of labels is $\mathcal{I}$.

- We set a parameter $1 < p \leq n = |\mathcal{I}|$.
- We train one network $\mathcal{N}_{\mathcal{J}}$ for every $\mathcal{J} \in \mathcal{P}(\mathcal{I})_p$.
- When testing an input, we feed it to our list of networks $(\mathcal{N}_{\mathcal{J}})_{\mathcal{J} \in \mathcal{P}(\mathcal{I})_p}$ and we obtain a list of labels $(\mathfrak{e}_{\mathcal{J}})_{\mathcal{J} \in \mathcal{P}(\mathcal{I})_p}$.
- We then apply a majority voting system.

For the sake of simplicty, suppose $\mathcal{I} = \{0, 1, \ldots, 9\}$ and $p = 2$. So every $\mathcal{J}$ is a set of type $\{i, j\}$, $i, j \in \{0, 1, \ldots, 9\}$, $i \neq j$. We denote with $\mathfrak{e}_{\{i,j\}}$ the output of the network $\mathcal{N}_{\{i,j\}}$.

Majority Voting - Example 1

$\boxed{\text{input}}$

Majority Voting - Example 1

$\mathcal{N}_{\{0,1\}}$

$\mathcal{N}_{\{0,2\}}$

$\vdots$

input

$\mathcal{N}_{\{0,9\}}$

$\mathcal{N}_{\{1,2\}}$

$\vdots$

$\mathcal{N}_{\{8,9\}}$

Majority Voting - Example 1

input

$\mathcal{N}_{\{0,1\}}$

$\mathcal{N}_{\{0,2\}}$

$\vdots$

$\mathcal{N}_{\{0,9\}}$

$\mathcal{N}_{\{1,2\}}$

$\vdots$

$\mathcal{N}_{\{8,9\}}$

0

1

2

3

4

5

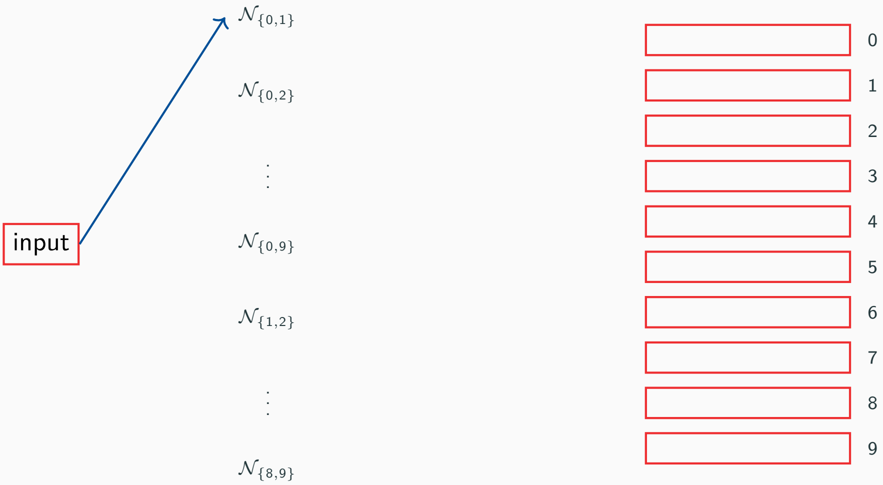
6

7

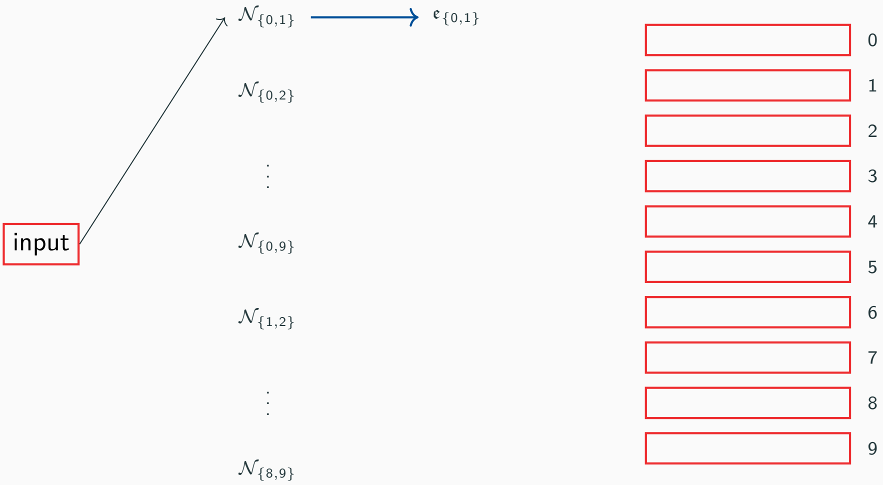
8

9

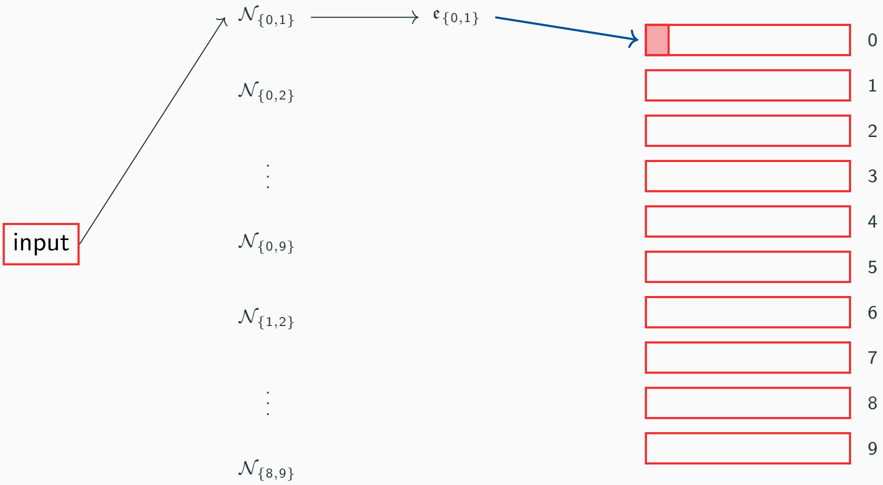
Majority Voting - Example 1



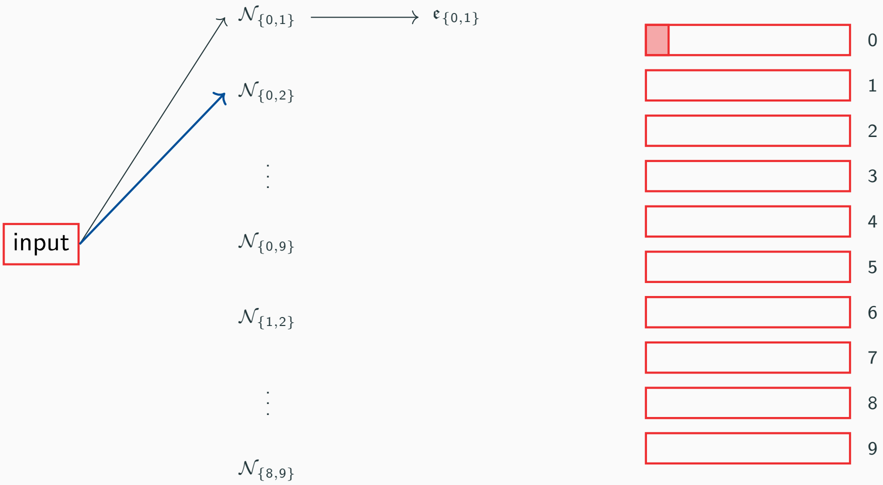
Majority Voting - Example 1



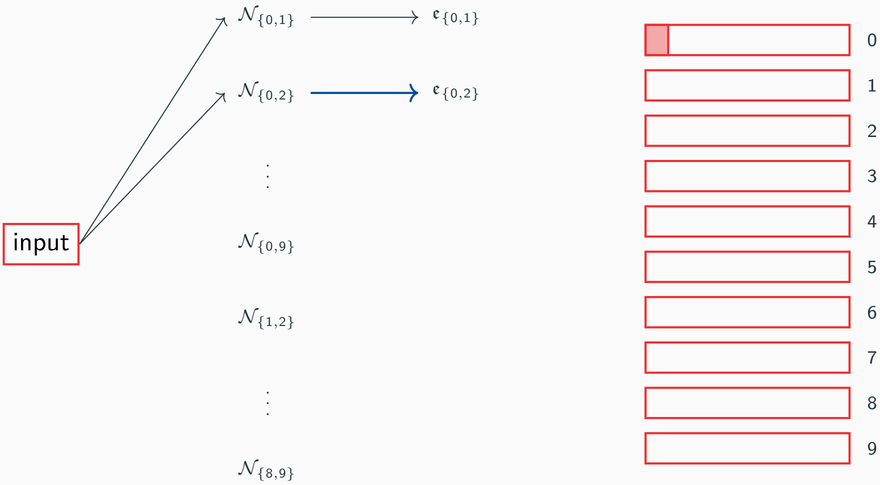
Majority Voting - Example 1



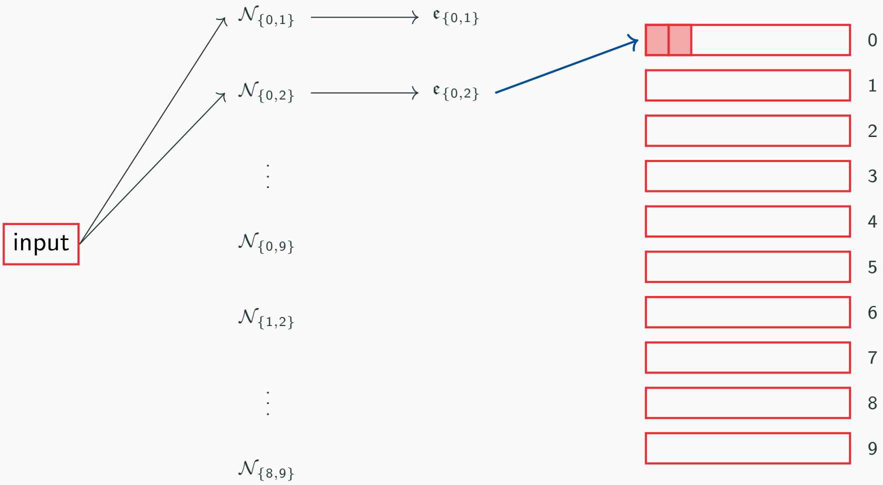
Majority Voting - Example 1



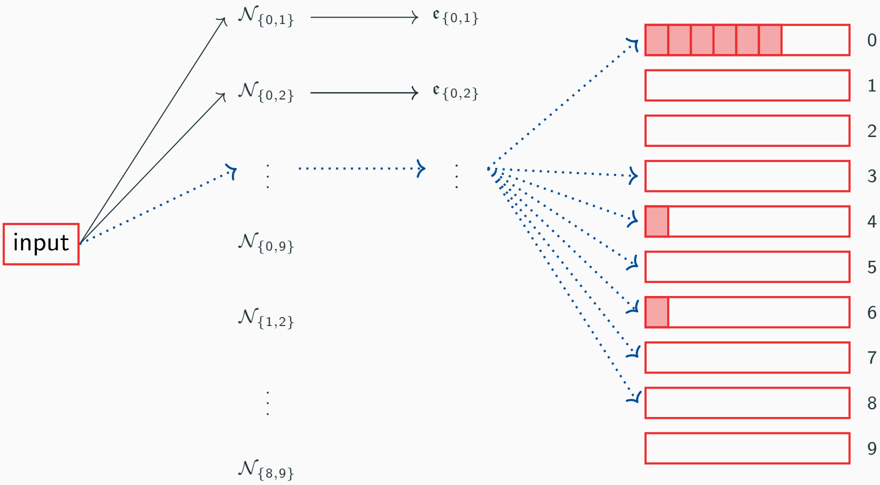
Majority Voting - Example 1



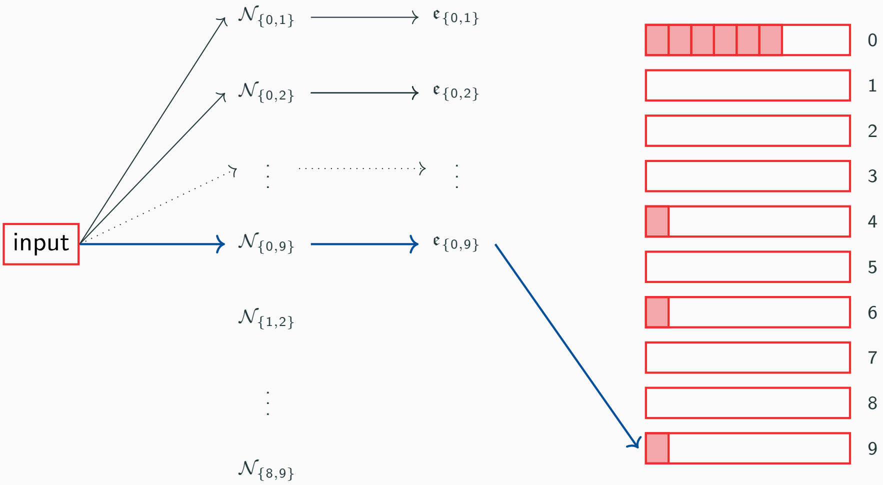
Majority Voting - Example 1



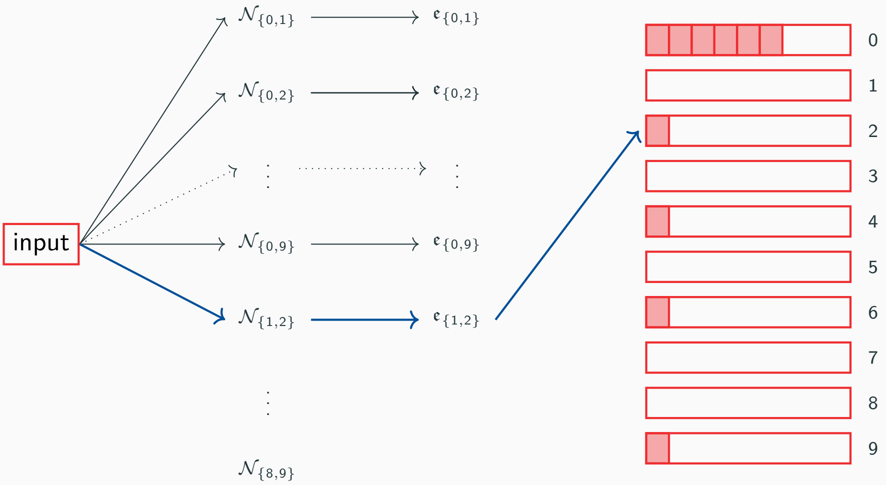
Majority Voting - Example 1



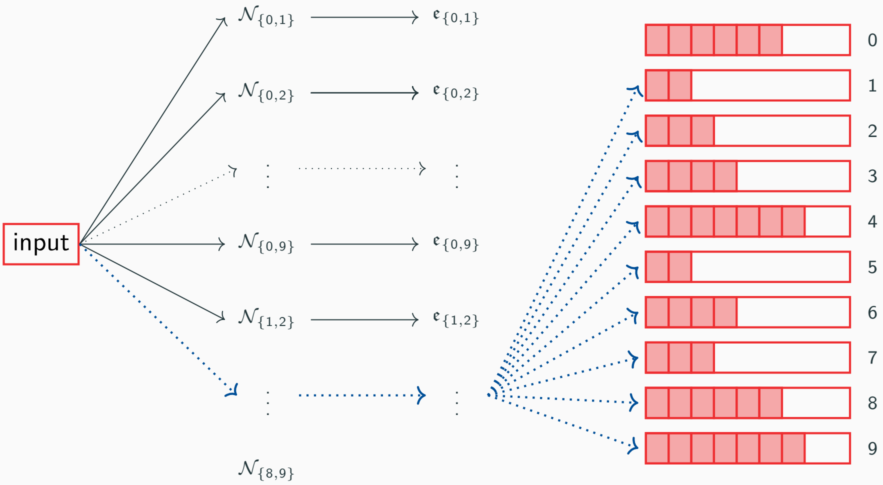
Majority Voting - Example 1



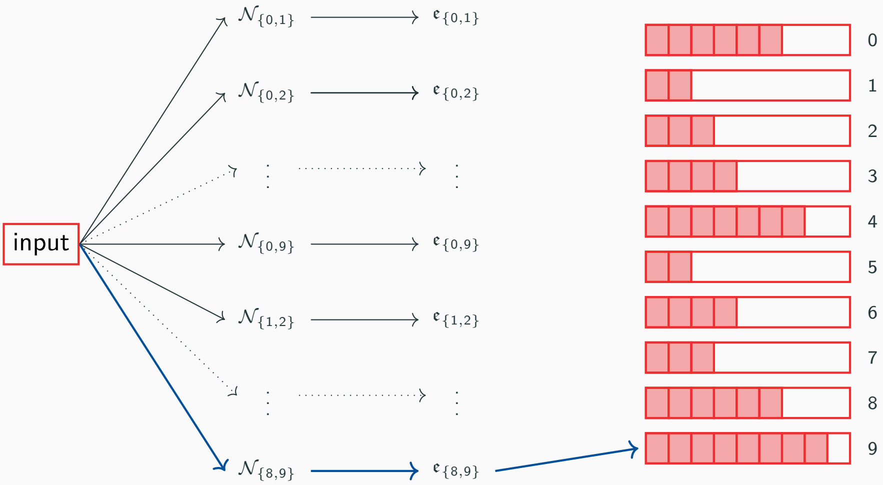
Majority Voting - Example 1



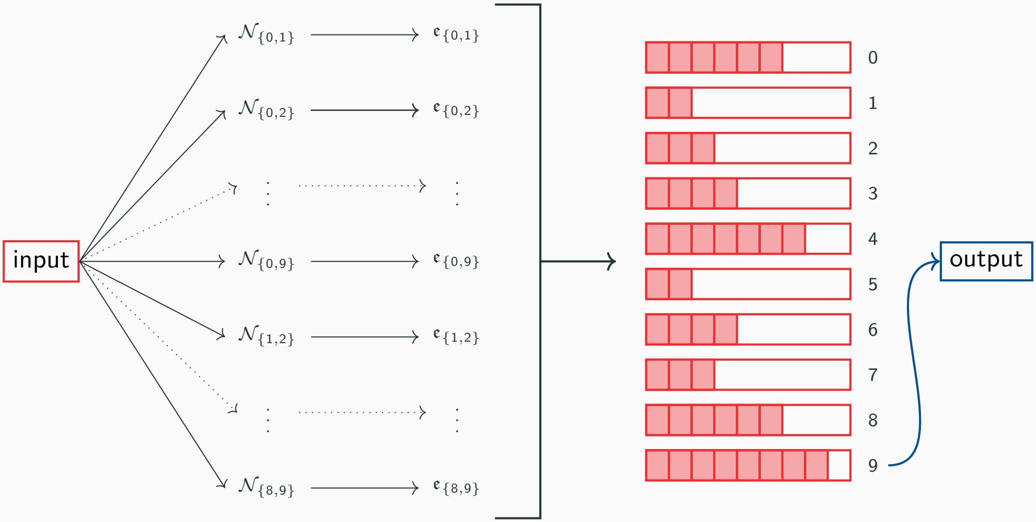
Majority Voting - Example 1



Majority Voting - Example 1



Majority Voting - Example 1



What about ex-aequo?

For every $k \in \{0, 1, \ldots, 9\}$ we define

$$C_k = \{\{i, j\} \in \mathcal{P}(\{0, 1, \ldots, 9\})_2 \mid \mathfrak{e}_{\{i,j\}} = k\}$$

and we say that a label k is a dominant label if $|C_k| \geq |C_l|$ for every $l \in \{0, 1, \ldots, 9\}$.

What about ex-aequo?

For every $k \in \{0, 1, \dots, 9\}$ we define

$$C_k = \{\{i, j\} \in \mathcal{P}(\{0, 1, \dots, 9\})_2 \mid \mathfrak{e}_{\{i,j\}} = k\}$$

and we say that a label k is a dominant label if $|C_k| \geq |C_l|$ for every $l \in \{0, 1, \dots, 9\}$. Then we can have three possible outcomes:

What about ex-aequo?

For every $k \in \{0, 1, \dots, 9\}$ we define

$$C_k = \{\{i, j\} \in \mathcal{P}(\{0, 1, \dots, 9\})_2 \mid \mathfrak{e}_{\{i,j\}} = k\}$$

and we say that a label k is a dominant label if $|C_k| \geq |C_l|$ for every $l \in \{0, 1, \dots, 9\}$. Then we can have three possible outcomes:

(a) there exists one $k \in \{0, 1, \dots, 9\}$ such that $|C_k| > |C_l|$ for every $l \in \{0, 1, \dots, 9\} \setminus \{k\}$ (there exists exactly one dominant label) $\implies$ our input is labelled as k;

What about ex-aequo?

For every $k \in \{0, 1, \ldots, 9\}$ we define

$$C_k = \{\{i, j\} \in \mathcal{P}(\{0, 1, \ldots, 9\})_2 \mid \mathfrak{e}_{\{i,j\}} = k\}$$

and we say that a label k is a dominant label if $|C_k| \geq |C_l|$ for every $l \in \{0, 1, \ldots, 9\}$. Then we can have three possible outcomes:

(a) there exists one $k \in \{0, 1, \ldots, 9\}$ such that $|C_k| > |C_l|$ for every $l \in \{0, 1, \ldots, 9\} \setminus \{k\}$ (there exists exactly one dominant label) $\implies$ our input is labelled as k;

(b) there exist $k_1, k_2 \in \{0, 1, \ldots, 9\}, k_1 \neq k_2$, such that $|C_{k_1}| = |C_{k_2}| > |C_l|$ for every $l \in \{0, 1, \ldots, 9\} \setminus \{k_1, k_2\}$ (there exist exactly two dominant labels) $\implies$ our input is labelled as $\mathfrak{e}_{\{k_1, k_2\}}$;

What about ex-aequo?

For every $k \in \{0, 1, \ldots, 9\}$ we define

$$C_k = \{\{i,j\} \in \mathcal{P}(\{0, 1, \ldots, 9\})_2 \mid \mathfrak{e}_{\{i,j\}} = k\}$$

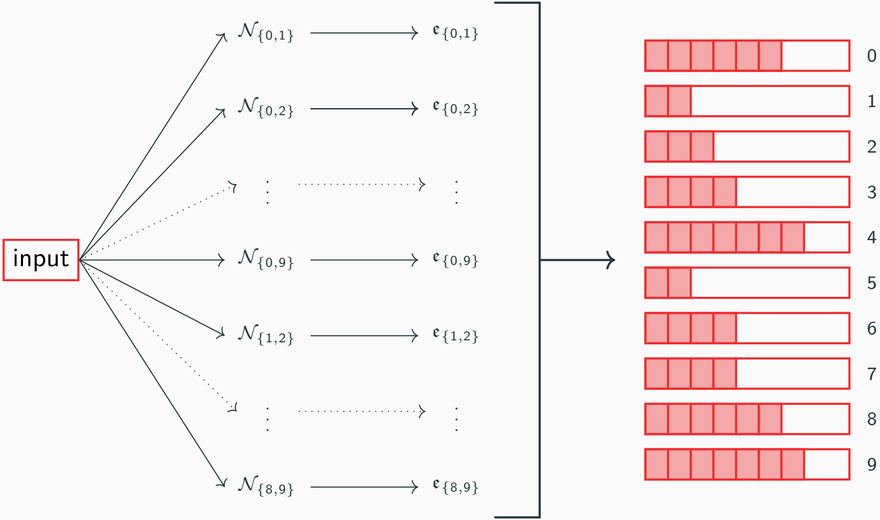
and we say that a label k is a dominant label if $|C_k| \geq |C_l|$ for every $l \in \{0, 1, \ldots, 9\}$. Then we can have three possible outcomes:

(a) there exists one $k \in \{0, 1, \ldots, 9\}$ such that $|C_k| > |C_l|$ for every $l \in \{0, 1, \ldots, 9\} \setminus \{k\}$ (there exists exactly one dominant label) $\implies$ our input is labelled as k;

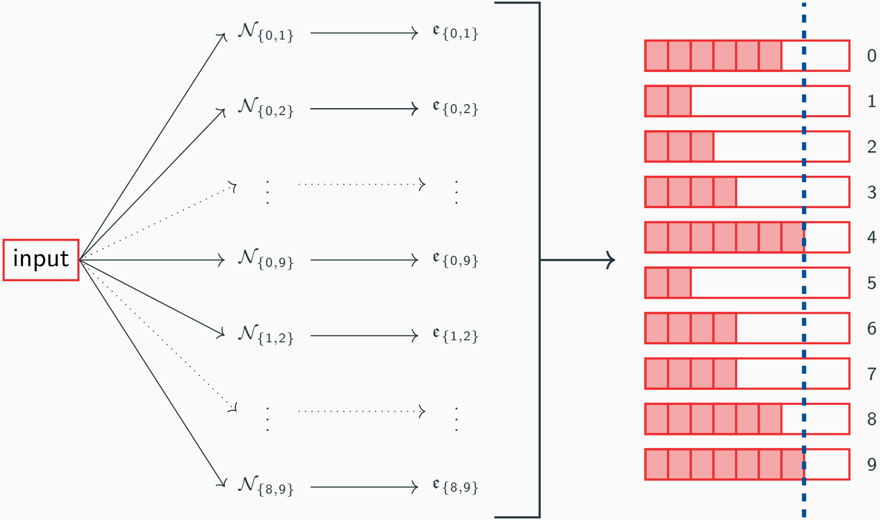
(b) there exist $k_1, k_2 \in \{0, 1, \ldots, 9\}$, $k_1 \neq k_2$, such that $|C_{k_1}| = |C_{k_2}| > |C_l|$ for every $l \in \{0, 1, \ldots, 9\} \setminus \{k_1, k_2\}$ (there exist exactly two dominant labels) $\implies$ our input is labelled as $\mathfrak{e}_{\{k_1,k_2\}}$;

(c) there exist three or more dominant labels $\implies$ our input is labelled as -1.

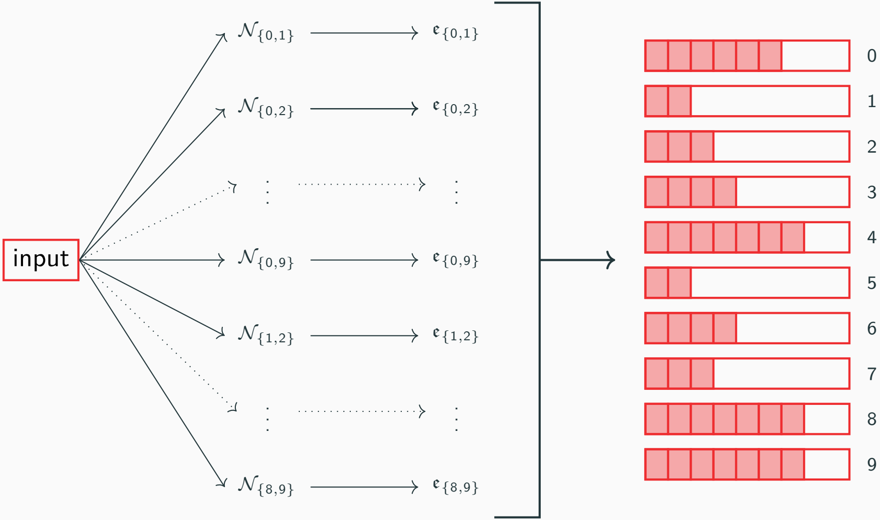
Ensemble majority voting - Example 2



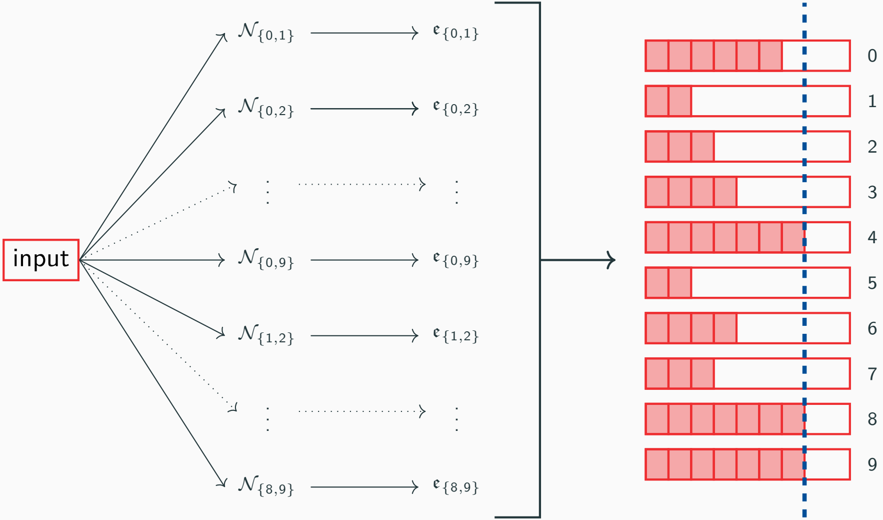
Ensemble majority voting - Example 2



Ensemble majority voting - Example 3



Ensemble majority voting - Example 3



Computational Analysis

Comparison with literature

From few-shot to small dataset regime

Confusion matrix

Actual \ Predicted	0	1	2	3	4	5	6	7	8	9	-1
0	448	0	0	2	1	17	6	0	5	1	20
1	0	487	2	4	0	1	2	0	2	1	1
2	12	58	335	19	2	8	17	2	25	2	16
3	0	9	11	406	0	19	2	12	24	5	12
4	0	1	14	0	382	1	19	6	4	66	7
5	3	5	1	35	2	400	7	3	25	8	11
6	17	3	15	1	9	23	418	2	3	0	9
7	1	20	26	14	7	1	1	383	6	35	6
8	8	13	9	31	7	32	4	5	367	12	12
9	7	5	5	6	43	0	0	15	8	401	9

Networks architecture: [784, 10, 3, 1];
Time limit for each network: 290s + 290s + 20s;

Training images per digit: 40;
Tested images: 5000.

The role of Min-Weight

Dataset	Layers	Images per class	Model S-M time (s)	Gap (%) mean	Gap (%) max	Links (%) (M-M)	Links (%) (M-W)
MNIST	784,4,4,1	10	2.99	17.37	28.25	49.25	27.14
		20	5.90	19.74	24.06	52.95	30.84
		30	10.65	20.07	26.42	56.90	30.88
		40	15.92	18.50	23.89	58.70	29.42
	784,10,3,1	10	6.88	6.28	9.67	49.46	23.96
		20	17.02	7.05	8.42	53.25	26.65
		30	25.84	7.38	15.88	57.21	25.02
		40	44.20	9.90	74.16	59.08	24.22
F-MNIST	784,4,4,1	10	7.66	17.21	25.92	86.38	56.54
		20	14.60	22.35	28.00	93.18	57.54
		30	26.10	19.78	29.53	92.56	58.78
		40	39.90	22.71	75.03	93.13	64.61
	784,10,3,1	10	13.83	6.14	8.98	86.65	53.72
		20	26.80	7.84	9.59	93.57	51.03
		30	38.48	7.18	16.09	92.90	52.50
		40	64.52	12.10	55.19	93.57	55.67

Conclusions

Final remarks and future perspectives

A way of combining MILP literature approaches to preserve feasibility, robustness, and simplicity;

a structured ensemble of BNNs that can be trained in parallel;

a majority voting system based on the structure of the ensemble.

Final remarks and future perspectives

A way of combining MILP literature approaches to preserve feasibility, robustness, and simplicity;

a structured ensemble of BNNs that can be trained in parallel;

a majority voting system based on the structure of the ensemble.

Exploit the BeMi structure on Integer-valued NNs;

improve the training data selection by using a k-medoids approach;

formulate an alternative model to improve the solver performances.

That's all Folks!

Any Questions?

You can also send me an e-mail at

ambrogiomaria.bernardelli01@universitadipavia.it