Training Binarized NN with MaxSAT

Knowledge and Representation Learning

Anna Putina[1]

[1]Università degli Studi di Padova

February, 2025

Table of Contents

Università degli Studi di Padova

Table of Contents

Università degli Studi di Padova

Binarized Neural Networks

- Neural networks with binary ({-1,1}) inputs, outputs, and weights.
- The goal is to find the set of weights that maximize the correct predictions of the net.

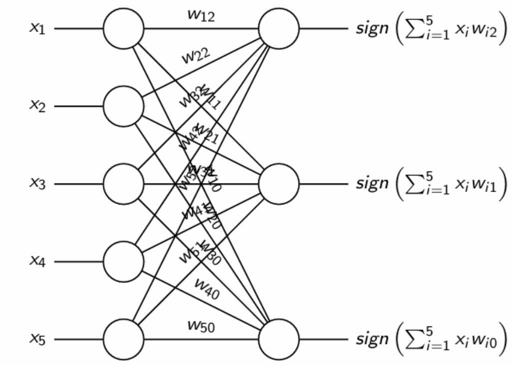

Figure 1. Binarized NN Structure

UNIVERSITÀ DEGLI STUDI DI PADOVA

Objectives

Fit a binarized neural network using MaxSAT:

1. Implement a method to encode a binarized neural network layer with m inputs and n outputs in Max-SAT.
2. Use this method to encode an entire BNN with multiple stacked layers in Max-SAT.
3. Define a training approach for the BNN using Max-SAT.
4. Generate train/test data for three binary functions (5 inputs, 5 outputs) and evaluate different network configurations.

UNIVERSITÀ DEGLI STUDI DI PADOVA

Table of Contents

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dataset Creation

- The dataset consists of all possible binary input combinations of size N=5, where each input is either -1 or 1.
- The outputs are generated using different logical functions, allowing for varying levels of complexity.

UNIVERSITÀ DEGLI STUDI DI PADOVA

Logical Functions

- Three different logical functions are defined. Each function uses basic logical operations (AND, OR, XOR, NOT).
- Nested conditions that simulate real-world nonlinear relationships. The functions are designed to test the ability of the BNN to approximate logical structures.

Logical Functions Example

$$\text{logical\_function\_1}(x) = \{\ (x_3, \quad x_1 \vee x_2, \quad x_0, \quad \neg x_3, \quad x_3 \wedge x_4)\ \}$$

$$\text{logical\_function\_2}(x) = \left\{ \begin{array}{c} (x_0 \wedge x_1, \quad x_2 \vee x_3, \\ (x_2 \wedge \neg x_3) \vee (\neg x_2 \wedge x_3), \quad \neg x_3, \\ (x_4 \wedge x_0) \vee x_1) \end{array} \right\}$$

$$\text{logical\_function\_3}(x) = \left\{ \begin{array}{c} ((x_0 \wedge x_1) \vee ((x_2 \wedge \neg x_3) \vee (\neg x_2 \wedge x_3))), \\ (x_1 \vee x_2) \wedge (\neg x_3 \vee x_4), \\ x_0 \vee (x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3), \\ (x_3 \wedge x_4) \vee (\neg x_0 \wedge \neg x_1) \vee (x_0 \wedge x_1), \\ ((x_2 \wedge x_3) \vee (x_4 \wedge \neg x_0) \vee (\neg x_4 \wedge x_0)) \wedge x_1 \end{array} \right\}$$

UNIVERSITÀ DEGLI STUDI DI PADOVA

Table of Contents

Università degli Studi di Padova

Indexing Scheme

- A single-layer BNN consists of:
  - N input neurons
  - M output neurons
  - A weight matrix W of shape (N, M)
- A multi-layer BNN consists of:
  - N input neurons
  - H hidden neurons
  - M output neurons
  - Two weight matrices:
    - W_1 of shape (N, H) (input to hidden)
    - W_2 of shape (H, M) (hidden to output)
- We assign unique indices to neurons and weights to avoid overlap in MaxSAT.

Università degli Studi di Padova

General SAT Indexing Formula

SAT Indexing Function

$$\text{SAT\_index}(i, n, N, \text{offset}) = \text{offset} + i \cdot N + n + 1$$

- i - Current index (e.g., training sample).
- n - Neuron or weight index in a layer.
- N - Number of elements in the layer.
- **offset** - Used to distinguish different sections.

UNIVERSITÀ DEGLI STUDI DI PADOVA

Encoding Input Neurons

Input Encoding Formula

$$x_{\text{enc}}(i, n) = \text{SAT\_index}(i, n, N, 0) = i \cdot N + n + 1$$

- N - Number of input neurons.
- n - Index of an input neuron.
- i - Index of a training sample.

Università degli Studi di Padova

SAT Index for First Layer Weights

Formula

$$w_{1\text{enc}}(i, h) = \text{SAT\_index}(n, h, H, \text{train\_size} \times N) =$$

$$\text{train\_size} \times N + (n \times H) + h + 1$$

- train_size - Number of training samples.
- N - Number of input neurons.
- H - Number of hidden neurons.
- n - Index of an input neuron.
- h - Index of a hidden neuron.

Università degli Studi di Padova

SAT Index for Second Layer Weights

Formula

$$w_{2\text{enc}}(h, m) = \text{SAT\_index}(h, m, M, \text{train\_size} \times N + N \times H) =$$

$$\text{train\_size} \times N + (N \times H) + (h \times M) + m + 1$$

- train_size - Number of training samples.
- N - Number of input neurons.
- H - Number of hidden neurons.
- h - Index of a hidden neuron.
- M - Number of output neurons.
- m - Index of an output neuron.

Università degli Studi di Padova

Table of Contents

UNIVERSITÀ DEGLI STUDI DI PADOVA

Hard Constraints in MaxSAT

- Hard constraints are derived directly from the training data.
- Each input variable $x_i^{(n)}$ is assigned a unique SAT index.
- A constraint holds True when $x_i^{(j)} = 1$ and False when $x_i^{(j)} = -1$.

UNIVERSITÀ DEGLI STUDI DI PADOVA

Activation Function Transformation

- Activation fucntion is sign $\left(\sum x_i w_i\right)$
- We use use the majority rule:

$$\{\# x_i w_i > 0\} \quad > \quad \{\# x_i w_i < 0\}$$

- $x, w \in \{-1, 1\}$, thus $x_i w_i$ is equivalent to $x_i \equiv w_i$.

  "$x_i \equiv w_i$ for at least more than half the i's."
- This leads to the soft constraints: penalizing for misclassification.

UNIVERSITÀ DEGLI STUDI DI PADOVA

Activation Function as a logic formula

- Example for $i = 1, 2, 3$ for the postive output neuron.

$$(\neg x_1 \vee w_1) \wedge (\neg w_1 \vee x_1) \wedge (\neg x_2 \vee w_2) \wedge (\neg w_2 \vee x_2)$$

$$\vee$$

$$(\neg x_1 \vee w_1) \wedge (\neg w_1 \vee x_1) \wedge (\neg x_3 \vee w_3) \wedge (\neg w_3 \vee x_3)$$

$$\vee$$

$$(\neg x_2 \vee w_2) \wedge (\neg w_2 \vee x_2) \wedge (\neg x_3 \vee w_3) \wedge (\neg w_3 \vee x_3)$$

Università degli Studi di Padova

Activation Function in CNF

- After transformation to CNF:

$$(x_1 \vee x_2 \vee \neg w_1 \vee \neg w_2) \wedge (x_1 \vee \neg x_2 \vee \neg w_1 \vee w_2) \wedge$$
$$(\neg x_1 \vee x_2 \vee w_1 \vee \neg w_2) \wedge (\neg x_1 \vee \neg x_2 \vee w_1 \vee w_2)$$
$$\wedge$$
$$(x_1 \vee x_3 \vee \neg w_1 \vee \neg w_3) \wedge (x_1 \vee \neg x_3 \vee \neg w_1 \vee w_3) \wedge$$
$$(\neg x_1 \vee x_3 \vee w_1 \vee \neg w_3) \wedge (\neg x_1 \vee \neg x_3 \vee w_1 \vee w_3)$$
$$\wedge$$
$$(x_2 \vee x_3 \vee \neg w_2 \vee \neg w_3) \wedge (x_2 \vee \neg x_3 \vee \neg w_2 \vee w_3) \wedge$$
$$(\neg x_2 \vee x_3 \vee w_2 \vee \neg w_3) \wedge (\neg x_2 \vee \neg x_3 \vee w_2 \vee w_3)$$

UNIVERSITÀ DEGLI STUDI DI PADOVA

Table of Contents

UNIVERSITÀ DEGLI STUDI DI PADOVA

Computational Complexity with Hidden Layers

- Introducing a hidden layer significantly increases computational complexity.
- In a single-layer network, outputs are determined directly from input neurons and their corresponding weights.
- With a hidden layer, the output depends on:
  - The activation of hidden neurons, influenced by input values and weights w^1.
  - The combination of hidden neurons and their connections to output neurons via weights w^2.
- This leads to exponential growth in the number of constraints required for encoding in CNF.

UNIVERSITÀ DEGLI STUDI DI PADOVA

Computational Complexity Example

In the implementation case:

- With N = 5 input neurons, M = 5 output neurons and train_size = 22 training samples we have 8800 soft constraints.
- With N = 5 input neurons, H = 3 hidden neurons, M = 3 output neurons (the situation with 5 outputs is unfeasible for 15 Gb RAM computer) and train_size = 22 training samples we have 5068800 soft constraints.

UNIVERSITÀ DEGLI STUDI DI PADOVA

Table of Contents

UNIVERSITÀ DEGLI STUDI DI PADOVA

Experimental Results

- The dataset was split into 70% for training and 30% for testing.
- We compare a BNN without a hidden layer and a BNN with a hidden layer.
- The BNN without a hidden layer fails to handle even simple cases due to non-linearity of the logical functions.
- The BNN with a hidden layer achieves perfect training accuracy and high test accuracy but at a higher computational cost.
- Increasing the output neurons to 5 made training unfeasible, highlighting the scalability challenge of the MaxSAT approach.

Experimental Results

Model	Solver Cost	Training Accuracy	Test Accuracy
BNN Without Hidden Layer			
Function 1	27	0.75	0.54
Function 2	27	0.75	0.54
Function 3	26	0.76	0.52
BNN With Hidden Layer			
Function 1	2	1.00	0.80
Function 2	4	1.00	0.80
Function 3	1	1.00	0.80

Table 1. Comparison of BNN with and without a hidden layer

UNIVERSITÀ DEGLI STUDI DI PADOVA