Taming Binarized Neural Networks and Mixed-Integer Programs

Johannes Aspman\textsuperscript{1}, Georgios Korpas\textsuperscript{1,2}, Jakub Marecek\textsuperscript{1}

\textsuperscript{1}Department of Computer Science, Czech Technical University, Prague, Czechia
\textsuperscript{2}HSBC Lab, Innovation & Ventures, HSBC Holdings, London, United Kingdom
johannes.aspman@cvut.cz, georgios.korpas@hsbc.com, jakub.marecek@cvut.cz

Abstract

There has been a great deal of recent interest in binarized neural networks, especially because of their explainability. At the same time, automatic differentiation algorithms such as backpropagation fail for binarized neural networks, which limits their applicability. We show that binarized neural networks admit a tame representation by reformulating the problem of training binarized neural networks as a subadditive dual of a mixed-integer program, which we show to have nice properties. This makes it possible to use the framework of Bolte et al. for implicit differentiation, which offers the possibility for practical implementation of backpropagation in the context of binarized neural networks.

This approach could also be used for a broader class of mixed-integer programs, beyond the training of binarized neural networks, as encountered in symbolic approaches to AI and beyond.

Introduction

There has been a great deal of recent interest in binarized neural networks (BNNs) (Hubara et al. 2016; Courbariaux et al. 2016; Yuan and Agaian 2021), due to their impressive statistical performance (Rastegari et al. 2016, e.g.), the ease of distributing the computation (Hubara et al. 2016, e.g.), and especially their explainability. This latter property, which is rather rarely encountered in other types of neural networks, stems precisely from the binary representation of the outputs of activation functions of the network, which can be seen as logical rules. This explainability is increasingly mandated by regulation of artificial intelligence, including the General Data Protection Regulation and the AI Act in the European Union, and the Blueprint for an AI Bill of Rights pioneered by the Office of Science and Technology Policy of the White House. The training of BNNs typically utilizes the Straight-Through-Estimator (STE) (Courbariaux, Bengio, and David 2015; Courbariaux et al. 2016; Rastegari et al. 2016; Zhou et al. 2016; Lin, Zhao, and Pan 2017; Bulat and Tzimiropoulos 2017; Cai et al. 2017; Xiang, Qian, and Yu 2017), where the weight updates in backpropagation unfortunately (Alizadeh et al. 2018) do not correspond to subgradients of the forward paths. This can lead to poor stationary points (Yin et al. 2018), and thus poor explanations.

Here, we draw a new relationship between binarized neural networks and so-called tame geometry (van den Dries 1998) to address this challenge. We introduce a certain reformulation of the training of BNN, which allows us to make use of the results of implicit differentiation and non-smooth optimization when training the BNNs (Davis et al. 2020; Bolte and Pauwels 2021; Bolte et al. 2021, 2022) and, eventually, to obtain weight updates in the backpropagation that do correspond to subgradients of the forward paths in common software frameworks built around automated differentiation, such as TensorFlow or PyTorch.

This builds on a long history of work on tame topology and o-minimal structures (Grothendieck 1997; van den Dries 1998; Kurdyka 1998; Kurdyka, Mostowski, and Parusinski 2000; Fornasiero and Servi 2008; Fornasiero 2010; Kawakami et al. 2012; Fornasiero 2013; Fujita 2023, e.g.), long studied in topology, logic, and functional analysis.

Our reformulation proceeds as follows: In theory, the training of BNNs can be cast as a mixed-integer program (MIP). We formulate its sub-additive dual, wherein we leverage the insight that conic MIPs admit a strong dual in terms of non-decreasing subadditive functions. We show that this dual problem is tame, or definable in an o-minimal structure. This, in turn, makes it possible for the use of powerful methods from non-smooth optimization when training the BNN, such as a certain generalized derivative of (Bolte and Pauwels 2021) that comes equipped with a chain rule. Thus, one can use backpropagation, as usual in training of neural networks.

In the process, we establish a broader class of nice MIPs that admit such a tame reformulation. A MIP is nice if its feasible set is compact, and the graph of the objective function has only a finite number of non-differentiable points. This class could be of independent interest, as it may contain a number of other problems, such as learning causal graphs (Chen, Dash, and Gao 2021), optimal decision trees (Nemecek, Pevny, and Marecek 2023; Nemecek et al. 2023), or certain problems in symbolic regression (Austel et al. 2020; Kim, Leyffer, and Balaprakash 2023). We hope that this could bring closer symbolic approaches, which can often be cast as MIPs, and approaches based on neural networks and backpropagation.
Background

Let us start by introducing the relevant background material. We begin by introducing the relevant notions of BNNs, MIPs, and their subadditive dual. We discuss how the BNN can be recast as a MIP, and thus, by strong duality, how training the BNN relates to a maximization problem over a set of subadditive functions. Our main goal is to link the BNN with its tameness. We begin by introducing the relevant notions of BNNs, then proceed to introduce the relevant background material. Let us start by introducing the relevant background material. We begin by introducing the relevant BNNs, MIPs, and their subadditive dual. We discuss how the BNN can be recast as a MIP, and thus, by strong duality, how training the BNN relates to a maximization problem over a set of subadditive functions. Our main goal is to link the BNN with its tameness.

Binarized Neural Networks

There is some ambiguity in the literature as to what constitutes a binarized neural network (BNN). We will follow (Bah and Kurtz 2020) and refer to a BNN as a neural network where the activation functions take values in the binary set \{0, 1\}. A BNN is characterized by a vector \(L = (L_0, \ldots, L_n)\) with \(|L| = n\) layers where each layer contains \(L_i \in \mathbb{N}_{>0}\) neurons \(x_i^\ell\), see Fig. 1. We allow the input layer \(x_i^{(0)}\) to take any real values, \(x_i^{(0)} \in \mathbb{R}\), while due to binarized activations, the following layers will have \(x_i^{(j \geq 0)} \in \{0, 1\}\). The neuron \(x_i^{\ell}\) in the layer \(\ell\) is connected with the neuron \(x_j^{(\ell+1)}\) in the layer \(\ell+1\) via a weight coefficient matrix \(w_i^j \in \mathbb{R}^{L_\ell \times L_{\ell+1}}\). Consider an input vector \(x = (x_1^{(0)}, \ldots, x_{L_0}^{(0)})\). The preactivation function of the BNN is given as

\[
a_j^{(\ell+1)}(x) = \sum_{i \in L_{\ell+1}} w_{ij}^\ell \sigma_j^{(\ell)}(x),
\]

where \(\sigma_j^{(\ell)}(x)\) is the activation function at layer \(\ell\) with

\[
\sigma_j^{(\ell)}(x) = \begin{cases} x & \text{if } \ell = 0, \\ 1 & \text{if } \ell > 0 \text{ and } \sigma_j^{(\ell)}(x) \geq \lambda\ell, \\ 0 & \text{otherwise}, 
\end{cases}
\]

where \(\lambda\ell \in \mathbb{R}\) is a learnable parameter. Note again that the activation functions of all the neurons in the network of our BNN are constrained in the set \{0, 1\} except for the input layer neurons. This set can be mapped to \{-1, 1\} by a re-definition \(\tilde{\sigma}_j^{(\ell)} = 2\sigma_j^{(\ell)} - 1\).

The BNN can be viewed as a weight assignment \(w = \{w_1, \ldots, w_{L_n}\}\) for a function

\[
f_w : \mathbb{R}^{L_0} \rightarrow \{0, 1\}^{L_n},
\]

\[
x^{(0)} \mapsto \hat{y},
\]

where \(\hat{y} = x^{(L_n)}\) is the vector of output layer neurons. BNNs are trained by finding an optimal weight assignment \(W\) that fits and generalizes a training set \(S = \{(x_1, y_1), \ldots, (x_m, y_m)\}\). The traditional approaches of backpropagation and gradient descent methods in usual deep learning architectures cannot be used directly for training BNNs. For optimizers to work as in standard neural network architectures, real-valued weights are required, so, in practice, when binarized weights and/or activation functions are utilized, one still uses real-valued weights for the optimization step. Another problem is related to the use of deterministic functions (2) or stochastic functions (Hubara et al. 2016) for binarization, which “flattens the gradient” during backpropagation. A common solution to these problems is to use the Saturated STE (Straight Through Estimator) (Bengio, Léonard, and Courville 2013) (see also (Yin et al. 2018)). Other possible solutions include the Expectation BackPropagation (EBP) algorithm (Soudry, Hubara, and Meir 2014) which is a popular approach to training multilayer neural networks with discretized weights, and Quantized BackPropagation (QBP) (Hubara, Hoffer, and Soudry 2018). Ref. (Alizadeh et al. 2018) presents a comprehensive practical survey on the training approaches for BNNs. In this article, we suggest that BNNs can be efficiently trained using nonsmooth implicit differentiation (Bolte et al. 2021).

Mixed-Integer Programming

A mixed-integer linear program (MILP) is an optimization problem of the form

\[
\begin{align*}
\max & \quad cx + hy \\
\text{s.t.} & \quad Ax + G\tilde{y} \geq b \\
& \quad x \in \mathbb{Z}_{\geq 0} \\
& \quad y \in \mathbb{R}_{\geq 0}.
\end{align*}
\]

As illustrated in Figure 2, the feasible set is a subset of the intersection of a polyhedron with the integral grid.

Recasting a BNN as a MIP

The interactions and relations between BNNs and MILPs have been studied in recent literature. For example, in Ref. (Icarte et al. 2019) BNNs with
weights restricted to \([-1, 1]\) are trained by a hybrid method based on constraint programming and mixed-integer programming. Generally, BNNs with activation functions taking values in a binary set and with arbitrary weights can be reformulated as a MIP (Bah and Kurtz 2020). However, the precise form of the corresponding MIP depends on the nature of the loss function. Generally, a loss function for a BNN with \(n\) layers is a map \(\mathcal{L} :\{0, 1\} \times \mathbb{R}^{L_n} \to \mathbb{R}\), which allows the BNN to be represented as

\[
\min \sum_{i=1}^{m} \mathcal{L}(y_i, \hat{y}_i) \quad \text{(BNN-MINLP)}
\]

s.t. \(\hat{y}^i = a^L \left( w^{(L)} a^{(L-1)} \left( \ldots a^{(1)} \left( w^{(1)} x_i \right) \ldots \right) \right)
\]

\[
w^{(\ell)} \in \mathbb{R}^{L_{\ell} \times L_{\ell-1}}, \quad \forall \ell,
\]

\[
\lambda_{\ell} \in \mathbb{R}, \quad \forall \ell,
\]

\[
\hat{y} \in \{0, 1\}^m.
\]

The loss function \(\mathcal{L}\) can be chosen in different ways; for example the 0-1 loss function \(\mathcal{L}(\hat{y}, y) = I_{\hat{y} \neq y}\), where \(I\) is the indicator function, and the square loss \(\mathcal{L}(\hat{y}, y) = \|\hat{y} - y\|^2\). The following result will then be essential for us (Bah and Kurtz 2020, Thm. 2):

**Theorem 1 (MILP formulation)** (BNN-MINLP) is equivalent to the following mixed-integer linear program:

\[
\min \sum_{i=1}^{m} \mathcal{L}(y_i, u^{(L)}) \quad \text{(BNN-MILP)}
\]

s.t. \(w^{(1)} x_i \leq M_1 u^{(1)} + \lambda_1\)

\[
w^{(1)} x_i = M_1 \left( u^{(1)} - 1 \right) + \lambda_1
\]

\[
\sum_{i=1}^{d_{L-1}} s_{(k)}^{(i)} < M_k u^{(k)} + \lambda_k, \quad \forall k \in [L] \setminus \{1\}
\]

\[
\sum_{i=1}^{d_{L-1}} s_{(k)}^{(i)} \geq M_k \left( u^{(k)} - 1 \right) + \lambda_k, \quad \forall k \in [L] \setminus \{1\}
\]

\[
s_{(i,j)}^{(k)} \leq u_{j}^{(k)} - \lambda_{k}^{(i)}, \quad s_{(i,j)}^{(k)} \geq u_{j}^{(k)} - \lambda_{k}^{(i)}, \quad \forall k \in [L] \setminus \{1\}, i \in [d_{k-1}], j \in [d_k]
\]

\[
s_{(i,j)}^{(k)} \leq u_{j}^{(k)} + \left( 1 - u_{j}^{(k)} \right), \quad \forall k \in [L] \setminus \{1\}, i \in [d_{k-1}], j \in [d_k]
\]

\[
s_{(i,j)}^{(k)} \geq u_{j}^{(k)} - \left( 1 - u_{j}^{(k)} \right), \quad \forall k \in [L] \setminus \{1\}, i \in [d_{k-1}], j \in [d_k]
\]

\[
W^k \in [-1, 1]^{d_x \times d_y}, \quad \forall k \in [L]
\]

\[
\lambda_{k} \in [-1, 1], \quad \forall k \in [L]
\]

\[
u^{(k)} \in \{0, 1\}^{d_{k}}, \quad \forall k \in [L], i \in [m]
\]

\[
s_{(i,j)}^{(k)} \in [-1, 1], \quad \forall i \in [m], k \in [L] \setminus \{1\}, l \in [d_{k-1}],
\]

where \(x := x^{(0)}, u^{(\ell)} := x^{(\ell)}, for 0 < \ell \leq L, M_1 := (n+1), \|x\| < r\) a Euclidean norm bound, \(n\) the dimension of \(x\), and \(M_\ell := (d_{\ell-1} + 1)\). The new variables \(s_{(i,j)}^{(k)} \in [-1, 1]\) have been added to linearize the products \(w^{(\ell)} \hat{y}^{(\ell-1)}\) that would otherwise appear. Finally, we have rescaled the weights \(w\) and parameters \(\lambda\) to lie in \([-1, 1]\), without loss of generality. See also Lemma 1 in (Bah and Kurtz 2020).

This gives the first step in our aim to link the theory of tame geometry, or o-minimality, to BNNs. The next step is to look at the dual problem of this MILP.

**Subadditive dual** In the context of MIPs, the notion of duality is much more involved than in convex optimization (Güzelsoy, Ralphs, and Cochran 2010). Only recently (Kocuk and Morán 2019; Morán R, Dey, and Vielma 2012), it is emerging that subadditive duals (Jerolmack 1978; Johnson 1980, 1974, 1979; Güzelsoy and Ralphs 2007; Wolsey and Nemhauser 1999) can be used to establish strong duality for MIPs. To introduce the subadditive dual, we use the modern language of (Morán R, Dey, and Vielma 2012; Kocuk and Morán 2019):

**Definition 1 (Regular cone)** A cone \(K \subseteq \mathbb{R}^m\) is called regular if it is closed, convex, pointed and full-dimensional.

If \(x - y \in K\), we write \(x \succeq_K y\) and similarly, if \(x \in \text{int}(K)\) we write \(x \succ_K 0\).

**Definition 2 (Subadditive and non-decreasing functions)**

A function \(f : \mathbb{R}^m \to \mathbb{R}\) is called:

- subadditive if \(f(x+y) \leq f(x) + f(y)\) for all \(x, y \in \mathbb{R}^m\);
- non-decreasing with respect to a regular cone \(K \subseteq \mathbb{R}^m\) if \(x \succeq_K y \implies f(x) \geq f(y)\).

The set of subadditive functions that are non-decreasing with respect to a regular cone \(K \subseteq \mathbb{R}^m\) is denoted \(\mathcal{F}_K\) and for \(f \in \mathcal{F}_K\) we further define \(f(x) := \limsup_{\delta \to 0^+} \frac{f(x + \delta)}{\delta}\). Note that this is the upper \(x\)-directional derivative of \(f\) at zero.

Let us start by stating the relation between subadditive functions and MIPs. To this end, we consider a generic conic MIP,

\[
z^* := \inf c^T x + d^T y, \\
\text{s.t. } Ax + Gy \succeq_K b, \\
x \in \mathbb{Z}^{n_1}, \\
y \in \mathbb{R}^{n_2}. \\
(6)
\]

Note that problem (6) is a generalization of the primal form of a MILP, as in Thm. 1, which is recovered by setting \(K = \mathbb{R}^{n_1}_+\). We define the subadditive dual problem of (6) as

\[
\rho^* := \sup f(b), \\
\text{s.t. } f(A^j) = -f(-A^j) = c_j, \\
\text{for } j = 1, \ldots, n_1, \\
f(G^k) = -f(-G^k) = d_k, \\
\text{for } k = 1, \ldots, n_2, \\
(7)
\]

\[
f(0) = 0, \\
f \in \mathcal{F}_K,
\]

where \(A^j\) and \(G^k\) denotes the \(j\)th column of the matrices \(A\) and \(G\), respectively, and \(c_j, d_k\) are the components of the corresponding vectors from the primal MIP.

In general, the subadditive dual (7) is a weak dual to the primal conic MIP (6), where any dual feasible solution provides a lower bound for the optimal value of the primal.
(Zălinescu 2011; Ben-Tal and Nemirovski 2001; Morán R. Dey, and Vielma 2012). Under the assumptions of feasibility, strong duality holds:

**Theorem 2 (Thm. 3 of (Kocuk and Morán 2019))** If the primal conic MIP (6) and the subadditive dual (7) are both feasible, then (7) is a strong dual of (6). Furthermore, if the primal problem is feasible, then the subadditive dual is feasible if and only if the conic dual of the continuous relaxation of (6) is feasible.

That is: Theorem 2 provides a sufficient condition for the subadditive dual to be equivalent to (6). A sufficient condition for the dual feasibility is that the conic MIP has a bounded feasible region.

**Properties of subadditive functions** To show our main result, Theorem 4, we will need to introduce some structural properties of subadditive functions. These are discussed in detail in (Rosenbaum 1950; Matkowski and Świątkowski 1993; Bingham and Ostaszewski 2008). For example, if \( f, g \) are two non-decreasing subadditive functions on \( \mathbb{R}^m \), then the following hold:

- \( f + g \) is subadditive;
- the composition \( g \circ f \) is subadditive;
- if further \( f \) is non-negative and \( g \) positive on the positive quadrant \( \mathbb{R}^+_m \) then \( f(x)g(x) \) is subadditive on \( \mathbb{R}^+_m \).

Let us note that, when we set \( K = \mathbb{R}^+_m \) in (7) we have that \( f(x) \) is non-negative on \( \mathbb{R}^m \) due to the combination of being non-decreasing, subadditive and having the condition \( f(0) = 0 \).

Following (Bingham and Ostaszewski 2008), we define properties NT (as in “no trumps”) and WNT (for “weak no trumps”), see also Def. 1 and 2 of (Bingham and Ostaszewski 2008):

**Definition 3 (NT)** For a family \( \{ A_k \}_{k \in \mathbb{N}} \) of subsets of \( \mathbb{R}^n \) we say that NT(\( A_k \)) holds, if for every bounded/convergent sequence \( \{ a_j \} \) in \( \mathbb{R}^n \) some \( A_k \) contains a translate of a subsequence of \( \{ a_j \} \).

**Definition 4 (WNT)** Let \( f : \mathbb{R}^m \to \mathbb{R} \). We call \( f \) a WNT-function, or \( f \in \text{WNT} \), if NT(\( \{ F^j \}_{j \in \mathbb{N}} \)) holds, where \( F^j := \{ x \in \mathbb{R}^m : |f(x)| < j \} \).

We have the following theorem by Csiszár and Erdős (Csiszár and Erdos 1964), nicely explained in (Bingham and Ostaszewski 2009):

**Theorem 3 (NT theorem, (Csiszár and Erdos 1964))** If \( T \) is an interval and \( T = \bigcup_{j \in \mathbb{N}} T_j \) with each \( T_j \) measurable/Baire, then NT(\( \{ T_k : k \in \mathbb{N} \} \)) holds.

Here, Baire refers to the functions having “the Baire property”, or the set being open modulo some meager set. Note that this is not necessarily related to being definably Baire as in Def. 7.

The following properties are shown in (Bingham and Ostaszewski 2008):

- If \( f \) is subadditive and locally bounded above at a point, then it is locally bounded at every point.
- If \( f \in \text{WNT} \) is subadditive, then it is locally bounded.

- If \( f \in \text{WNT} \) is subadditive and \( \inf_{t<0} f(tx)/t \) is finite for all \( x \), then \( f \) is Lipschitz.

**Tame topology** The subject of tame topology goes back to Grothendieck and his famous “Esquisse d’un programme” (Grothendieck 1997). Grothendieck claimed that modern topology was riddled with false problems, which he ascribed to the fact that much of modern progress had been made by analysts. What he proposed was the invention of a geometrical version of topology, lacking these artificial problems from the onset. Subsequently, tame topology has been linked to model-theoretic notions of o-minimal structures, which promise to be good candidates for Grothendieck’s dream. o-minimal structures are a generalisation of the (semi-)algebraic sets, or the sets of polynomial equations (and inequalities). As such, they provide us with a large class of sets and functions that are in general non-smooth and non-convex, while capturing most (if not all) of the popular settings used in modern neural networks and machine learning (Davis et al. 2020).

An o-minimal structure over \( \mathbb{R} \) is a collection of subsets of \( \mathbb{R}^m \) that satisfies certain finiteness properties, such as closure under boolean operations, closure under projections and fibrations. Formally,

**Definition 5 (o-minimal structure)** An o-minimal structure on \( \mathbb{R} \) is a sequence \( S = (S_m)_{m \in \mathbb{N}} \) such that for each \( m \geq 1 \):

1) \( S_m \) is a boolean algebra of subsets of \( \mathbb{R}^m \);
2) if \( A \in S_m \), then \( \mathbb{R} \times A \) and \( A \times \mathbb{R} \) belongs to \( S_{m+1} \);
3) \( S_m \) contains all diagonals, for example \( \{(x_1, \ldots, x_m) \in \mathbb{R}^m : x_1 = \ldots = x_m \} \in S_m \);
4) if \( A \in S_{m+1} \), then \( \pi(A) \in S_m \);
5) the sets in \( S_1 \) are exactly the finite unions of intervals and points.

Typically, we refer to a set included in an o-minimal structure as being definable in that structure, and similarly, a function, \( f : \mathbb{R}^m \to \mathbb{R} \), is called definable in an o-minimal structure whenever its corresponding graph, \( \Gamma(f) = \{(x,y) : |f(x) - y| \leq \mathbb{R}^{m \times n} \} \), is definable. A set, or function, is called tame to indicate that it is definable in some o-minimal structure, without specific reference to which structure.

The moderate sounding definition of o-minimal structures turns out to include many non-trivial examples. First of all, by construction, semialgebraic sets form an o-minimal structure, denoted \( \mathbb{R}^{\text{semi alg}} \). If this was the only example of an o-minimal structure, it would not have been a very interesting construction. The research in o-minimal structures really took off in the middle of the nineties, after Wilkie (Wilkie 1996) proved that we can add the graph of the real exponential function, \( x \mapsto e^x \), to \( \mathbb{R}^{\text{semi alg}} \) to again find an o-minimal structure, denoted \( \mathbb{R}^{\text{exp}} \). As a result, the sigmoid function, which is a prevalent activation function in numerous neural networks, can be considered tame. Another important structure is found by including the set of restricted real-analytic functions, where the domain of an analytic function is restricted to lie in a finite subset of the original domain in
a particular way. This gives rise to an o-minimal structure denoted $\mathbb{R}_{\text{an}}$ (van den Dries and Miller 1994). A classical example of this would be the function $\sin(x)$, where we restrict $x$ to lie in a finite interval $x \in [0, \alpha] \subset \mathbb{R}$ for some $\alpha < \infty$. Note that without this restriction on the argument, $\sin(x)$ is not tame. Furthermore, we can construct a very important o-minimal structure by combining $\mathbb{R}_{\text{an}}$ with $\mathbb{R}_{\text{an}}$. This gives the structure denoted $\mathbb{R}_{\text{an,exp}}$ (van den Dries and Miller 1994). It is important to note that the fact that $\mathbb{R}_{\text{an,exp}}$ is an o-minimal structure is a non-trivial result. In general, it does not hold that the combination of two o-minimal structures gives another o-minimal structure.

We thus see that o-minimal structures capture a very large class of, generally, non-smooth non-convex functions. More importantly, they include all classes of functions widely used in modern machine learning applications. The great benefit of this class is that they are still nice enough such that we can have some control over their behaviour and prove convergence to optimal points (Bolte, Daniilidis, and Lewis 2009; Davis et al. 2020; Bolte and Pauwels 2021; Aravanis et al. 2022; Josz 2023).

Perhaps the most fundamental results regarding o-minimal structures are the monotonicity and cell decomposition theorems. The former states that any tame function of one variable can be divided into a finite union of open intervals, and points, such that it is continuous and either constant or strictly monotone on each interval. The cell decomposition theorem generalizes this to higher dimensions by introducing the concept of a cell, which is the analogue of the interval or point in one dimension. The theorem then states that any tame function or set can be decomposed into a finite union of definable cells. A related notion is that of a stratification of a set. Generally, a stratification is a way of partitioning a set into a collection of submanifolds called strata. There exist many different types of stratifications, characterized by how the different strata are joined together. Two important such conditions are given by the Whitney and Verdier stratifications. Both of these are applicable to tame sets (Loi 1996; Lê Loi 1998). These results are at the core of many of the strong results on tame functions in non-smooth optimization.

**Locally o-minimal structures** There exists a few variants of weakenings of the o-minimal structures. One such example is what is called a locally o-minimal structure (Fornasiero and Servi 2008; Fornasiero 2010; Kawakami et al. 2012; Fornasiero 2013; Fujita 2023).

**Definition 6 (Locally o-minimal structure)** A definably complete structure $\mathbb{K}$ extending an ordered field is locally o-minimal if, for every definable function $f : \mathbb{K} \to \mathbb{K}$, the sign of $f$ is eventually constant.

Here, definably complete means that every definable subset of $\mathbb{K}$ has a supremum in $\mathbb{K} \cup \{\pm \infty\}$ and $X \subseteq \mathbb{K}$ is nowhere dense if $\text{Int}(X)$ is empty. Every o-minimal expansion of an ordered field is a definably complete structure (but the converse is not true). Note also that every o-minimal structure is locally o-minimal (Fornasiero 2010). Locally o-minimal structures satisfy a property called definably Baire:

**Definition 7 (Definably Baire, from (Fornasiero 2013))** A definably complete structure $\mathbb{K}$ extending an ordered field is definably Baire if $\mathbb{K}$ is not the union of a definable increasing family of nowhere dense subsets.

Finally, when we work with structures expanding $\langle \mathbb{R}, +, \cdot, < \rangle$ we have that local o-minimality implies o-minimality (Kawakami et al. 2012), while the same is generically not true when we do not have multiplication.

**Non-smooth differentiation** Bolte et al., (Bolte and Pauwels 2021), introduced a generalized derivative, called a conservative set-valued field, for non-smooth functions. The main idea behind this construction is that the conservative fields come equipped with a chain rule. Namely, given a locally Lipschitz function $f : \mathbb{R}^m \to \mathbb{R}$, we say that $D : \mathbb{R}^m \Rightarrow \mathbb{R}^m$ is a conservative field for $f$ if and only if the function $t \mapsto f(x(t))$ satisfies

$$\frac{d}{dt} f(x(t)) = (v, \dot{x}(t)), \quad \forall v \in D(x(t)),$$

for any absolutely continuous curve $x : [0, 1] \to \mathbb{R}^m$ and for almost all $t \in [0, 1]$. Having a chain rule is key for applications to backpropagation algorithms and automatic differentiation in machine learning.

Automatic differentiation for non-smooth elementary functions is subtle and even the well-known Clarke generalized gradient is known to introduce complications in this setting. Having a derivative flexible enough to include automatic differentiation was therefore indeed the main motivation behind the work of Bolte et al. In many ways, we can see the conservative fields as a generalization of the Clarke derivatives.

The conservative fields provide a flexible calculus for non-smooth differentiation that is applicable to many machine learning situations. In (Bolte et al. 2021), a non-smooth implicit differentiation using the conservative Jacobians is developed. This can be seen as a form of automatic subdifferentiation (backpropagation). The automatic subdifferentiation is an automated application of the chain rule, made available through the use of the conservative fields. It amounts to calculating the conservative Jacobians of the underlying functions. This "conservative subgradient descent" is given by picking an initial value for the parameters, captured by a vector $v_0$ followed by performing the following update in steps

$$v_{k+1} = v_k + \alpha_k g_k,$$

$$g_k \in J(v_k),$$

with $(\alpha_k)_{k \in \mathbb{N}}$ a sequences of step-sizes and $J(v_k)$ the conservative Jacobian (Bolte et al. 2021).

This gives a formal mathematical model for propagating derivatives which can be applied to guarantee local convergence of mini-batch stochastic gradient descent with backpropagation for a large number of machine learning problems. In particular, and of great importance for us, these results hold for locally Lipschitz tame functions.

Next, we will show that the subadditive dual of the MIP formulation of the BNN (BNN-MINLP) is locally Lipschitz and tame. This will allow us to use the machinery of (Bolte
and Pauwels 2021; Bolte et al. 2021) discussed above when training the BNN.

**Main Result**

We will now present the main result of the paper. To do so, we will restrict to a certain subset of conic MIPs, which we call nice:

**Definition 8** Let us consider a conic MIP (6). Under the following conditions:

(A) the conic MIP is feasible, 
(B) the conic dual of the continuous relaxation of (6) is feasible, 
(C) the graph of the objective function has a finite number of non-differentiable points,

we call the conic MIP nice.

For example:

**Proposition 1** The conic MIP of Theorem 1 is nice.

**Proof.** The feasible set is a product of \( \{0,1\}^L \) and the set \( S \). For any value in \( \{0,1\}^L \), we obtain a finite value within \( S \). The feasible set is then compact. Theorem 5 of (Kocuk and Moran 2019) then tells us that condition (B) of Definition 8 is satisfied. The objective function is a finite sum of loss functions for the original BNN, and as such it has a finite number of non-differentiable points, satisfying condition (C). ■

**Theorem 4** For a nice conic MIP (6), there exists an equivalent reformulation that is definable in an o-minimal structure.

**Proof.** Let us consider the subadditive dual (7) of the nice conic MIP (6). When the conic dual of the continuous relaxation is feasible, this dual is equivalent by Theorem 2. Furthermore, this dual is locally o-minimal by considering the No-Trumps theorem (Theorem 3) together with the fact that \( f(x) \) is non-decreasing and subadditive. By (Kawakami et al. 2012, Remark 22), a compact subset of a locally o-minimal structure is o-minimal. When we consider that the continuous relaxation of the mixed-integer set is bounded (cf. Property (B) of Definition 8 together with Thm. 5 of (Kocuk and Moran 2019)), we obtain o-minimality. ■

**Corollary 1** Training BNNs allows for implicit differentiation and chain rule.

**Proof.** This follows from Proposition 1 and Theorem 4 together with the work of (Bolte and Pauwels 2021; Bolte et al. 2021) discussed above, when one realizes that the subadditive dual is locally Lipschitz. Lipschitzianity is from (Bingham and Ostaszewski 2008): If \( f \in \text{WNT} \) is subadditive and \( \inf_{t<0} f(tx)/t \) is finite for all \( x \), then \( f \) is Lipschitz. ■

This corollary thus provides us with a practical way of training the BNNs, by utilizing the results of (Bolte and Pauwels 2021; Bolte et al. 2021) to optimize over the subadditive dual of the corresponding MILP.

Let us finally note that, in general, non-decreasing subadditive functions are not tame. A counterexample is given by the Cantor staircase function (Dobos 1996). This means that in general, the subadditive dual of a conic MIP need not fall under the tame setting and some additional property (“constraint qualification”) is necessary for our main result.

**An Example**

To make the above discussion more clear, we present a simple example outlining how the training of a BNN could make use of the implicit differentiation (Bolte et al. 2021). To this end, we consider three final layers of a BNN inspired by Example 1 of (Guzelsoy and Ralphs 2007), illustrated in Fig. 3, where there are a number of binary weights given by the final layer of (BNN-MILP). \( u_i^{(L)}, i = 1, \ldots, n \), to be learned. Here, of course \( w(L) = \hat{y} = x^{(L,n)} = a(l(w(L))a(l^{(-1)})\ldots a(l(1))a(l(1)x)\ldots) \). We split this vector into two, by introducing an \( m \in \mathbb{N} \) such that \( 1 < m < n \). The pen-ultimate two layers yield a bi-variate continuous valued output layer \( (Y_1, Y_2) \). Instead of the usual empirical risk, we consider an objective function involving weighted difference from values of the dependent variable in the training data (assumed to be zero), as well as one of the weights in the pen-ultimate layer, for the sake of a more interesting illustration:

\[
\min_{X,Y} 2(Y_1 - 0) + (Y_2 - 0) + \frac{1}{2} X_1,
\]

s.t.

\[
X_1 - \frac{3}{2} X_2 + Y_1 - Y_2 = b,
\]

\[
X_1 = \sum_{i=1}^{m} u_i^{(L)},
\]

\[
X_2 = \sum_{i=m+1}^{n} u_i^{(L)},
\]

\[
u_i^{(L)} \in \{0,1\}, X_1, X_2 \in \mathbb{Z}_+, Y_1, Y_2 \in \mathbb{R}_+.
\]

We note that our theory does cover the case of the usual empirical risk with square-loss function, but the illustrations would be more involved due to the non-linearity in the square loss.

Following the definition (7), the subadditive dual of (10)
is:

\[
\max_{f \in \mathcal{F}_{\mathbb{R}^+}} f(b),
\]

s.t. \( f(1) \leq \frac{1}{2}, \)
\[
f(-\frac{3}{2}) \leq 0,
\]
\[
\tilde{f}(1) \leq 2,
\]
\[
\tilde{f}(-1) \leq 1,
\]
\[
f(0) = 0.
\]

The subadditive dual problem is obviously an infinite-dimensional optimization problem over the whole space of subadditive functions \( \mathcal{F}_{\mathbb{R}^+} \). However, as shown in (Schrijver et al. 1980), the subadditive dual functions of MILPs are Chvátal functions, i.e., piecewise-linear. We can thus utilize this knowledge to finitely parametrize the space of relevant subadditive functions by the number of segments, slopes, and breakpoints of piecewise-linear subadditive functions. When we consider nice MILPs as in (8), we thus obtain a finite-dimensional problem. It is furthermore evident that we can approximate this problem by truncating in the number of segments of the piecewise-linear subadditive functions.

For the above example, we start with approximating \( f \) by a piecewise-linear function having two segments. By visual inspection of the behaviour of the value function \( f(b) \) (in solid lines) near the origin in Fig. 4, we see that we can approximate \( f(b) \) by

\[
\tilde{f}(b) := \begin{cases} 
2b, & b > 0, \\
-b, & b \leq 0,
\end{cases}
\]

(12)

based on the directional derivatives. This crude approximation is shown in Fig. 4 as dashed lines. A conservative field for this function is given by

\[
D\tilde{f}(b) = \begin{cases} 
2, & b > 0, \\
[-1, 2], & b = 0, \\
-1, & b < 0.
\end{cases}
\]

(13)

It is now clear that we can use the conservative fields of (Bolte and Pauwels 2021; Bolte et al. 2021) to train over this approximation of the piecewise-linear subadditive dual of the primal problem (10).

More generally, we can introduce slope variables \( s_1 \) and \( s_2 \), as well as a breaking point \( p \), to parametrize the two-segment approximation:

\[
\tilde{f}(b) := \begin{cases} 
s_1 b, & b > p, \\
s_2 b, & b \leq p,
\end{cases}
\]

(14)

and thus find the best two-segment approximation of the piecewise-linear subadditive dual of the primal problem (10), which in this case coincides with (12) above. Next, we can increase the precision of the approximation by introducing more and more segments of this approximating function, and optimize over the slopes and break points of the segments, and possibly also the number of segments. Following (Bertsimas and Dunn 2017), we have studied (Nemecek et al. 2023) formulations based on the optimal regression trees for piecewise regression.

\[
\text{Figure 4: The value function, } f(b), \text{ of the example (11) together with the simple approximation, } \tilde{f}(b), \text{ given by (12).}
\]

Conclusions and Limitations

We have introduced a link between binarized neural networks, and more broadly, nice conic MILPs, and tame geometry. This makes it possible to reuse pre-existing theory and practical implementations of automatic differentiation. Breaking new ground, we leave many questions open. The foremost question is related to the efficiency of algorithms for constructing the subadditive dual. Although Guzelsoy and Ralphs (Guzelsoy and Ralphs 2007, Section 4, Constructing Dual Functions) survey seven very different algorithms, their computational complexity and relative merits are not well understood. For any of those, an efficient implementation (in the sense of output-sensitive algorithm) would provide a solid foundation for further empirical experiments. Given the immense number of problems in symbolic AI, which can be cast as MILPs, and the excellent scalability of existing frameworks based on automatic differentiation, the importance of these questions cannot be understated.

Acknowledgements

The research of Johannes Aspman has been supported by European Union’s Horizon Europe research and innovation programme under grant agreement No. GA 101070568 (Human-compatible AI with guarantees). This work received funding from the National Centre for Energy II (TN02000025). Jakub would like to acknowledge most pleasant discussions with Jerome Bolte in Prague, the Czech Republic, and with Ted Ralphs in Monterrey, Mexico.

Disclaimer This paper was prepared for information purposes and is not a product of HSBC Bank Plc. or its affiliates. Neither HSBC Bank Plc. nor any of its affiliates make any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, but limited to, the completeness, accuracy, reliability of information contained herein and the potential legal, compliance, tax or accounting effects thereof.
References


Csiszár, I.; and Erdos, P. 1964. \limsup_{t\to\infty}(f(x + t) - f(x)). Magyar Tud Akad. Kut. Int. Közl. A, 9: 603–606.


