# Trimmed Neural Networks

**Mastane Achab** FIRSTNAME.ACHAB@M4X.ORG

**Massil Achab** FIRSTNAME.ACHAB@M4X.ORG

*Editions Achab, Tizi Ouzou, Algeria*

## 1. Introduction

### 1.1 Motivation

We introduce a new type of artificial neural network (ANN): the trimmed neural network (TNN) model. As most ANNs, a TNN is an alternating sequence of linear and nonlinear vectorial operators. Recall that in usual ANN models, nonlinear functions are independently applied on each entry of each layer (see e.g. LeCun et al. (2015)). In contrast, we design TNNs' nonlinearities as functions of the whole layer: indeed, they are based on sorting all the layer's entries. In particular, we focus on the trimming operation which consists in summing all entries but a certain fraction of the smallest/largest ones. We show that TNNs enjoy convexity properties useful in various statistical learning contexts.

### 1.2 Formal Definition

A trimmed neural network (TNN) $\mathcal{N} : \mathbb{R}^d \to \mathbb{R}$ with $L+1$ layers (or equivalently $L$ hidden layers) is characterized by $L$ matrices $\Omega_i \in \mathbb{R}^{p_i \times q_{i-1}}$ (with convention $q_0 = d$), $L$ trimming operators $\mathcal{T}_i : \mathbb{R}^{p_i} \to \mathbb{R}^{q_i}$ $(1 \leq i \leq L)$ and the last layer's coefficients $\omega \in \mathbb{R}^{q_L}$. We define $\mathcal{T}_i$ as follows: for all $v \in \mathbb{R}^{p_i}$, $\mathcal{T}_i v = v' = (v'_1, \ldots, v'_{q_i})$ with

$$\forall 1 \leq j \leq q_i, \quad v'_j = \underset{a \in [0,1]^{p_i}, ||a||_1 / p_i = \tau_{i,j}}{\text{mix(i)}} a^\intercal v,$$

where $\text{mix} = (\text{mix}(1), \ldots, \text{mix}(L)) \in \{\min, \max\}^L$ and $\tau_{i,j} \in [0,1]$. If $\text{mix(i)} = \min$ (resp. $\text{mix(i)} = \max$), then $v'_j$ is the sum of the $\tau_{i,j} p_i$ smallest (resp. largest) components of $v$. Observe that $\mathcal{T}_i$ is a nonlinear mapping as soon as $\{\tau_{i,j}, 1 \leq j \leq q_i\} \nsubseteq \{0,1\}$ ; and $v \mapsto v'_j$ is concave (resp. convex) in the case $\text{mix(i)} = \min$ (resp. $\text{mix(i)} = \max$). Then,

$$\mathcal{N} = \omega^\intercal \mathcal{T}_L \Omega_L \ldots \mathcal{T}_1 \Omega_1.$$

A TNN is said to be a min-TNN (resp. max-TNN), denoted by $\mathcal{N}_{\min}$ (resp. $\mathcal{N}_{\max}$), if $\text{mix} = (\min, \ldots, \min)$ (resp. $\text{mix} = (\max, \ldots, \max)$). We denote by $\overline{\mathcal{N}} = \mathcal{N}_{\min} + \mathcal{N}_{\max}$ the sum of min/max-TNNs.

**Remark 1** *We point out that the subsequent analysis of TNNs straightforwardly extends to more general trimming operators:* $\mathcal{T}_i v = (v'_1, \ldots, v'_{q_i})$ *with* $v'_j = \text{mix(i)}_{a \in A_{i,j}} a^\intercal v$, *where* $A_{i,j}$ *is a compact subset of* $\mathbb{R}^{p_i}_+$ *for* $1 \leq j \leq q_i$.

### 1.3 Convexity Properties

Min and max-TNNs satisfy the following convexity properties.

**Proposition 2** *Let* $\mathcal{N}_{\min} = \omega^\intercal \mathcal{T}_L \Omega_L \dots \mathcal{T}_1 \Omega_1$ *and* $\mathcal{N}_{\max} = \omega'^\intercal \mathcal{T}'_{L'} \Omega'_{L'} \dots \mathcal{T}'_1 \Omega'_1$ *be min/max-TNNs with respective number of hidden layers* $L$ *and* $L'$*. Assume nonnegative matrices* $\Omega_i \in \mathbb{R}_+^{p_i \times q_{i-1}}$ *and* $\Omega'_j \in \mathbb{R}_+^{p'_j \times q'_{j-1}}$ *for all* $i \in \{2, \dots, L\}$ *and* $j \in \{2, \dots, L'\}$ *and nonnegative last layers coefficients* $\omega \in \mathbb{R}_+^{q_L}$ *and* $\omega' \in \mathbb{R}_+^{q'_{L'}}$*. Let* $x \in \mathbb{R}^d$*,* $i \in \{1, \dots, L\}$ *and* $j \in \{1, \dots, L'\}$*. Then,*

*(i)* $\Omega_i \mapsto \mathcal{N}_{\min}(x)$ *is concave on* $\mathbb{R}^{p_i \times q_{i-1}}$*,*

*(ii)* $\Omega'_j \mapsto \mathcal{N}_{\max}(x)$ *is convex on* $\mathbb{R}^{p'_j \times q'_{j-1}}$*.*

## 2. Learning TNNs

In this section, we analyse learning properties of sums of min/max-TNNs $\overline{\mathcal{N}} = \mathcal{N}_{\min} + \mathcal{N}_{\max}$.

### 2.1 Least Squares Regression

The least squares regression loss of $\overline{\mathcal{N}}$ for a pair $(x, y) \in \mathbb{R}^d \times \mathbb{R}$ is

$$\ell_{\mathrm{LS}}(\overline{\mathcal{N}}, (x, y)) = (\overline{\mathcal{N}}(x) - y)^2 = \{(\overline{\mathcal{N}}(x) - y)_-\}^2 + \{(\overline{\mathcal{N}}(x) - y)_+\}^2,$$

where for all $z \in \mathbb{R}$, $(z)_-$ (resp. $(z)_+$) denotes the negative (resp. positive) part of $z$. Notice that this decomposition of $\ell_{\mathrm{LS}}$ into negative/positive parts combined with Proposition 2 provides the following result.

**Proposition 3** *Consider the same notations and nonnegativity assumptions as in Proposition 2. Let* $(x, y) \in \mathbb{R}^d \times \mathbb{R}$*,* $i \in \{1, \dots, L\}$ *and* $j \in \{1, \dots, L'\}$*. Then,*

*(i)* $\Omega_i \mapsto \{(\overline{\mathcal{N}}(x) - y)_-\}^2$ *is convex on* $\mathbb{R}^{p_i \times q_{i-1}}$*,*

*(ii)* $\Omega'_j \mapsto \{(\overline{\mathcal{N}}(x) - y)_+\}^2$ *is convex on* $\mathbb{R}^{p'_j \times q'_{j-1}}$*.*

It follows from Proposition 3 that $\overline{\mathcal{N}}$ can be learned by coordinate descent (see e.g. Wright (2015)). More precisely, for $i \in \{1, \dots, L\}$ and $j \in \{1, \dots, L'\}$ the matrices $\Omega_i$ and $\Omega'_j$ can be learned by respectively minimizing the negative and positive part terms of the least squares error.

### 2.2 Classification

**Hinge Loss.** The hinge loss of $\overline{\mathcal{N}}$ for a pair $(x, y) \in \mathbb{R}^d \times \{-1, +1\}$ is

$$\ell_{\mathrm{H}}(\overline{\mathcal{N}}, (x, y)) = (1 - y\overline{\mathcal{N}}(x))_+ = \mathbb{I}\{y = -1\}(1 + \overline{\mathcal{N}}(x))_+ + \mathbb{I}\{y = +1\}(1 - \overline{\mathcal{N}}(x))_+.$$

This two-terms decomposition of $\ell_{\mathrm{H}}$ combined with Proposition 2 yields the following convexity properties.

**Proposition 4** *Consider the same notations and nonnegativity assumptions as in Proposition 2. Let* $(x, y) \in \mathbb{R}^d \times \{-1, +1\}$, $i \in \{1, \dots, L\}$ *and* $j \in \{1, \dots, L'\}$. *Then,*

*(i)* $\Omega_i \mapsto \mathbb{I}\{y = +1\}(1 - \overline{\mathcal{N}}(x))_+$ *is convex on* $\mathbb{R}^{p_i \times q_{i-1}}$,

*(ii)* $\Omega'_j \mapsto \mathbb{I}\{y = -1\}(1 + \overline{\mathcal{N}}(x))_+$ *is convex on* $\mathbb{R}^{p'_j \times q'_{j-1}}$.

From Proposition 4, $\overline{\mathcal{N}}$ can be learned by minimizing the positive (resp. negative) label term with respect to $\Omega_i$ (resp. $\Omega'_j$) for any $i \in \{1, \dots, L\}$ (resp. $j \in \{1, \dots, L'\}$).

## References


Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1): 3–34, 2015.