Neural Network Approximation based on Hausdorff distance of Tropical Zonotopes

Panagiotis Misiakos, Georgios Smyrnis, Georgios Retsinas, Petros Maragos

National Technical University of Athens
School of Electrical and Computer Engineering

International Conference on Learning Representations (ICLR) 2022
Our work: Neural Networks and Tropical Geometry

Contributions

✓ **Novel** bound on neural network approximation.
✓ **2 new** algorithms for neural network compression.
Preface: Tropical Mathematics

Tropical Algebra

Tropical Geometry

Tropical Algebra

R_{\text{max}} = \mathbb{R} \cup \{-\infty\}

a \lor b = \max(a, b)

a + b = a + b

- Replaces classical operations of addition and multiplication with max and +, respectively.

Tropical Polynomials

f(x) = \max_{i \in [n]} \{a_i x + b_i\}

- Expressive for ReLU networks.

Tropical Geometry

Newton Polytopes

\text{Newt}(f) = \text{conv}\{a_i : i \in [n]\}

\text{ENewt}(f) = \text{conv}\{(a_i, b_i) : i \in [n]\}

- They provide geometric interpretation for tropical polynomials.
Tropical Algebra

✓ Tropical Semiring $\mathbb{R}_{\text{max}} = \mathbb{R} \cup \{-\infty\}$

\[
\begin{align*}
    a \lor b &= \max(a, b) \\
    a + b &= a + b
\end{align*}
\]
- Replaces classical operations of addition and multiplication with max and +, respectively.

Tropical Geometry

Newton Polytopes

\[
\text{Newt}(f) = \text{conv}\left\{a_i : i \in [n]\right\}
\]
\[
\text{ENewt}(f) = \text{conv}\left\{(a_i, b_i) : i \in [n]\right\}
\]
- They provide geometric interpretation for tropical polynomials.
Tropical Algebra

✓ Tropical Semiring $\mathbb{R}_{\text{max}} = \mathbb{R} \cup \{-\infty\}$

\[ a \vee b = \max(a, b) \]
\[ a + b = a + b \]

- Replaces classical operations of addition and multiplication with max and +, respectively.

✓ Tropical Polynomials

\[ f(x) = \max_{i \in [n]} \{a_i^T x + b_i\} \]

- Expressive for ReLU networks.

Tropical Geometry

- Newton Polytopes

\[ \text{Newt}(f) = \text{conv}\left\{a_i : i \in [n]\right\} \]
\[ \text{ENewt}(f) = \text{conv}\left\{(a_i, b_i) : i \in [n]\right\} \]

- They provide geometric interpretation for tropical polynomials.
Preface: Tropical Mathematics

Tropical Algebra

✓ **Tropical Semiring** \( \mathbb{R}_{\text{max}} = \mathbb{R} \cup \{-\infty\} \)

\[
 a \lor b = \max(a, b) \\
 a + b = a + b
\]

- Replaces classical operations of addition and multiplication with max and +, respectively.

✓ **Tropical Polynomials**

\[
 f(x) = \max_{i \in [n]} \{a_i^T x + b_i\}
\]

- Expressive for ReLU networks.

Tropical Geometry

✓ **Newton Polytopes**

\[
 \text{Newt}(f) = \text{conv}\{a_i : i \in [n]\} \\
 \text{ENewt}(f) = \text{conv}\{(a_i, b_i) : i \in [n]\}
\]

- They provide geometric interpretation for tropical polynomials.
Linear Regions and the Newton Polytope

✓ 1–1 mapping: between linear regions and vertices. [1]
✓ The upper envelope determines the tropical polynomial and vice versa

\[ f, g \in \mathbb{R}_{\max}[x] : \quad f = g \iff UF(ENewt(f)) = UF(ENewt(g)) \]

Idea: What if we relax the previous equality?

**Question:** Would $ENewt(f) \approx ENewt(g)$ imply $f \approx g$?
Idea: What if we relax the previous equality?

**Question:** Would $\text{ENewt}(f) \approx \text{ENewt}(g)$ imply $f \approx g$?

**Proposition**

Let $p, \tilde{p} \in \mathbb{R}_{max}[x]$ be two tropical polynomials and let $P = \text{ENewt}(p), \tilde{P} = \text{ENewt}(\tilde{p})$. Then,

$$\max_{x \in B} |p(x) - \tilde{p}(x)| \leq \rho \cdot \mathcal{H}(P, \tilde{P})$$

where $B = \{x \in \mathbb{R}^d : \|x\| \leq r\}$ is the hypersphere of radius $r$, and $\rho = \sqrt{r^2 + 1}$. 
Tropical Geometry of Neural Networks

ReLU neural network with 1 hidden layer

ReLU neural network with 1 hidden layer

✓ $i$–th hidden layer node.

\[ f_i(x) = \max (a_i^T x + b_i, 0) \]
Tropical Geometry of Neural Networks

ReLU neural network with 1 hidden layer

✓ $i$–th hidden layer node.

$$f_i(x) = \max(a_i^T x + b_i, 0)$$

✓ $j$–th output node.

$$v_j(x) = p_j(x) - q_j(x)$$


Tropical Geometry of Neural Networks

**Tropical Geometry**

- ENewt \( f_i \) is linear segment with endpoints 0 and \((a_i^T, b_i)\).

- \( P_j = ENewt(p_j), Q_j = ENewt(q_j) \) are Minkowski sums of segments ⇔ zonotopes [2,3].

- \((a_i^T, b_i)\) are called **generators**.

**ReLU neural network with 1 hidden layer**

- \( i \)-th hidden layer node.

\[
 f_i(x) = \max \left( a_i^T x + b_i, 0 \right) 
\]

- \( j \)-th output node.

\[
 v_j(x) = p_j(x) - q_j(x) 
\]


Neural Network Approximation

Theorem

Let \( v, \tilde{v} \in \mathbb{R}_{\max}[x] \) be two neural networks with 1 hidden layer and \( \tilde{P}_j, \tilde{Q}_j \) denote the positive and negative zonotopes of \( \tilde{v} \). The following bound applies.

\[
\max_{x \in B} \| v(x) - \tilde{v}(x) \|_1 \leq \rho \cdot \left( \sum_{j=1}^{m} \mathcal{H}(P_j, \tilde{P}_j) + \mathcal{H}(Q_j, \tilde{Q}_j) \right)
\]

✓ Geometrical approximation problem.
✓ **Goal**: approximate the zonotopes.
Compression Algorithms I. Zonotope K-means

(a) Original network

✓ Applies only to networks with one output neuron.

Zonotope K-means

1. Split zonotope generators into positive and negative.
2. Apply K-means to each generating set.
3. Construct final network.

✓ Applies only to networks with one output neuron.
Compression Algorithms I. Zonotope K-means

(a) Original network

(b) Original zonotopes

✓ Applies only to networks with one output neuron.

Zonotope K-means
1. Split zonotope generators into positive and negative.
Compression Algorithms I. Zonotope K-means

(a) Original network
(b) Original zonotopes
(c) Resulting zonotopes.

✓ Applies only to networks with one output neuron.

Zonotope K-means
1. Split zonotope generators into positive and negative.
2. Apply K-means to each generating set.
Compression Algorithms I. Zonotope K-means

(a) Original network
(b) Original zonotopes
(c) Resulting zonotopes.
(d) Compressed network.

✓ Applies only to networks with one output neuron.

Zonotope K-means
1. Split zonotope generators into positive and negative.
2. Apply K-means to each generating set.
3. Construct final network.
Compression Algorithms II. Neural Path K-means

- Zonotope K-means doesn’t generalize directly to multiple output nodes.
- Zonotope K-means doesn’t generalize directly to multiple output nodes.

**Neural Path K-means**
Zonotope K-means doesn’t generalize directly to multiple output nodes.

**Neural Path K-means**

1. For each node form the vector of weights of incident edges.
Neural Path K-means

1. For each node form the vector of weights of incident edges.
2. Execute K-means to these vectors.

- Zonotope K-means doesn't generalize directly to multiple output nodes.
Zonotope K-means doesn’t generalize directly to multiple output nodes.

**Neural Path K-means**

1. For each node form the vector of weights of incident edges.
2. Execute K-means to these vectors.
3. Construct reduced network.
Theoretical Evaluation

**Zonotope K-means Bound**

\[
\frac{1}{\rho} \cdot \max_{x \in B} |v(x) - \tilde{v}(x)| \leq K \cdot \delta_{\text{max}} + \left(1 - \frac{1}{N_{\text{max}}}\right) \sum_{i=1}^{n} |c_i| \| (a_i^T, b_i) \|
\]

**Neural Path K-means Bound**

\[
\frac{1}{\rho} \cdot \max_{x \in B} \|v(x) - \tilde{v}(x)\|_1 \leq \sqrt{m}K\delta_{\text{max}}^2 + \sqrt{m} \left(1 - \frac{1}{N_{\text{max}}}\right) \sum_{i=1}^{n} \| C_{:,i} \| \| (a_i^T, b_i) \| + \\
\frac{\sqrt{m}\delta_{\text{max}}}{N_{\text{min}}} \sum_{i=1}^{n} \left( \| (a_i^T, b_i) \| + \| C_{:,i} \| \right) + \sum_{j=1}^{m} \sum_{i \in N_j} |c_{ji}| \| (a_i^T, b_i) \|
\]

✓ Bounds represent distances of zonotope vertices from K-means centers.
✓ Approximation is better when \( K \approx n \). Both bounds become 0 when \( K = n \).
Experimental Evaluation I: Comparison with tropical techniques.

✓ Binary classification tasks.

<table>
<thead>
<tr>
<th>Percentage of Remaining Neurons</th>
<th>MNIST 3/5</th>
<th>MNIST 4/9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Smyrnis et al., 2020</td>
<td>Smyrnis et al., 2020</td>
</tr>
<tr>
<td>100% (Original)</td>
<td>99.18 ± 0.27</td>
<td>99.53 ± 0.09</td>
</tr>
<tr>
<td>1%</td>
<td>99.11 ± 0.36</td>
<td>99.01 ± 0.09</td>
</tr>
<tr>
<td>0.3%</td>
<td>99.18 ± 0.36</td>
<td>98.81 ± 0.09</td>
</tr>
</tbody>
</table>

✓ Multiclass classification tasks.

<table>
<thead>
<tr>
<th>Percentage of Remaining Neurons</th>
<th>MNIST</th>
<th>Fashion-MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Smyrnis and Maragos, 2020</td>
<td>Smyrnis and Maragos, 2020</td>
</tr>
<tr>
<td>100% (Original)</td>
<td>98.60 ± 0.03</td>
<td>88.66 ± 0.54</td>
</tr>
<tr>
<td>10%</td>
<td>93.48 ± 2.57</td>
<td>80.43 ± 3.27</td>
</tr>
<tr>
<td>5%</td>
<td>92.93 ± 2.59</td>
<td>—</td>
</tr>
</tbody>
</table>

Smyrnis et al., 2020 Zonotope K-means Neural Path K-means

Smyrnis and Maragos, 2020 Neural Path K-means
Experimental Evaluation II: Comparison with Thinet and baselines.

(a) LeNet5, MNIST
- ✓ 1 hidden layer with 84 neurons.

(b) LeNet5, F-MNIST

(c) deepNN, MNIST

(d) deepNN, F-MNIST
- ✓ 3 hidden layers.
Experimental Evaluation III: Larger datasets

(a) CIFAR-VGG, CIFAR10
CIFAR-VGG
✓ 1 hidden layer of size 512.

(b) CIFAR-VGG, CIFAR100
AlexNet
✓ 2 hidden layers of size 512.
Thank you!

Neural Network Approximation based on Hausdorff distance of Tropical Zonotopes

Panagiotis Misiakos, Georgios Smyrnis, Georgios Retsinas, Petros Maragos