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.

Tropical Algebra

Tropical Geometry

Tropical Algebra

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

$$a \vee b = \max(a, b)$$

$$a + b = a + b$$

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

Tropical Geometry

Tropical Algebra

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

$$a \vee b = \max(a, b)$$
$$a + b = a + b$$

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

✓ **Tropical Polynomials**

$$f(\boldsymbol{x}) = \max_{i \in [n]} \{\boldsymbol{a}_i^T \boldsymbol{x} + b_i\}$$

- Expressive for ReLU networks.

Tropical Geometry

Preface: Tropical Mathematics

Tropical Algebra

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

$$a \vee b = \max(a, b)$$
$$a + b = a + b$$

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

✓ **Tropical Polynomials**

$$f(\boldsymbol{x}) = \max_{i \in [n]} \{\boldsymbol{a}_i^T \boldsymbol{x} + b_i\}$$

- Expressive for ReLU networks.

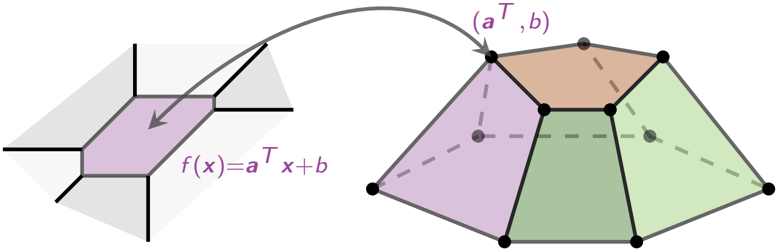
Tropical Geometry

✓ **Newton Polytopes**

$$\text{Newt}(f) = \text{conv}\{\boldsymbol{a}_i : i \in [n]\}$$
$$\text{ENewt}(f) = \text{conv}\{(\boldsymbol{a}_i, b_i) : i \in [n]\}$$

- 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

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

[1] Charisopoulos, V., Maragos, P. A tropical approach to neural networks with piecewise linear activations. *arXiv preprint arXiv:1805.08749*, 2018

Idea: What if we relax the previous equality?

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

Idea: What if we relax the previous equality?

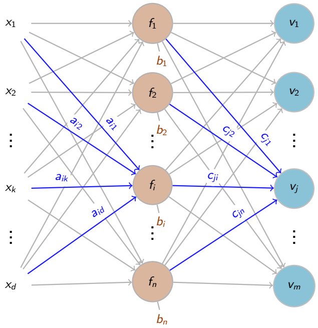
Question: Would $\text{ENewt}\,(f) \approx \text{ENewt}\,(g)$ imply $f \approx g$?

Proposition

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

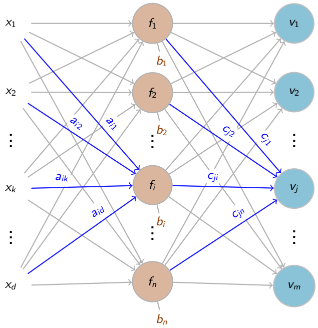
$$\max_{x \in \mathcal{B}} |p(\boldsymbol{x}) - \tilde{p}(\boldsymbol{x})| \leq \rho \cdot \mathcal{H}\left(P, \tilde{P}\right)$$

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



ReLU neural network with 1 hidden layer

[2] L. Zhang, G. Naitzat, L.-H. Lim. "Tropical Geometry of Deep Neural Networks." in *International Conference on Machine Learning*, pages 5824–5832. 2018.
[3] P. Maragos, V. Charisopoulos and E. Theodosis, "Tropical Geometry and Machine Learning," in *Proceedings of the IEEE*, vol. 109, no. 5, pp. 728-755, May 2021, doi: 10.1109/JPROC.2021.3065238.

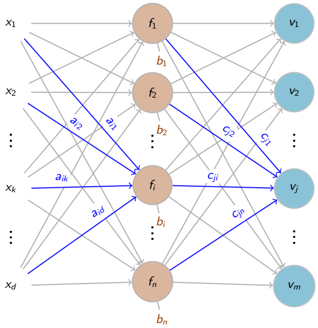


ReLU neural network with 1 hidden layer

✓ i−th hidden layer node.

$$f_i(\boldsymbol{x}) = \max\left(\boldsymbol{a}_i^T \boldsymbol{x} + b_i, 0\right)$$

[2] L. Zhang, G. Naitzat, L.-H. Lim. "Tropical Geometry of Deep Neural Networks." in *International Conference on Machine Learning*, pages 5824–5832. 2018.
[3] P. Maragos, V. Charisopoulos and E. Theodosis, "Tropical Geometry and Machine Learning," in *Proceedings of the IEEE*, vol. 109, no. 5, pp. 728-755, May 2021, doi: 10.1109/JPROC.2021.3065238.



ReLU neural network with 1 hidden layer

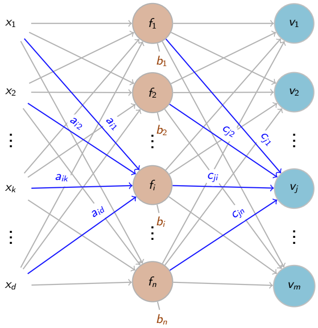
- ✓ $i-$th hidden layer node.

$$f_i(\boldsymbol{x}) = \max\left(\boldsymbol{a}_i^T \boldsymbol{x} + b_i, 0\right)$$

- ✓ $j-$th output node.

$$v_j(\boldsymbol{x}) = p_j(\boldsymbol{x}) - q_j(\boldsymbol{x})$$

[2] L. Zhang, G. Naitzat, L.-H. Lim. "Tropical Geometry of Deep Neural Networks." in *International Conference on Machine Learning*, pages 5824–5832. 2018.
[3] P. Maragos, V. Charisopoulos and E. Theodosis, "Tropical Geometry and Machine Learning," in *Proceedings of the IEEE*, vol. 109, no. 5, pp. 728-755, May 2021, doi: 10.1109/JPROC.2021.3065238.



ReLU neural network with 1 hidden layer

✓ $i-$th hidden layer node.

$$f_i(\boldsymbol{x}) = \max\left(\boldsymbol{a}_i^T \boldsymbol{x} + b_i, 0\right)$$

✓ $j-$th output node.

$$v_j(\boldsymbol{x}) = p_j(\boldsymbol{x}) - q_j(\boldsymbol{x})$$

Tropical Geometry

✓ ENewt (f_i) is linear segment with endpoints $\mathbf{0}$ and $(\boldsymbol{a}_i^T, b_i)$.

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

✓ $(\boldsymbol{a}_i^T, b_i)$ are called **generators**.

[2] L. Zhang, G. Naitzat, L.-H. Lim. "Tropical Geometry of Deep Neural Networks." in *International Conference on Machine Learning*, pages 5824–5832. 2018.
[3] P. Maragos, V. Charisopoulos and E. Theodosis, "Tropical Geometry and Machine Learning," in *Proceedings of the IEEE*, vol. 109, no. 5, pp. 728-755, May 2021, doi: 10.1109/JPROC.2021.3065238.

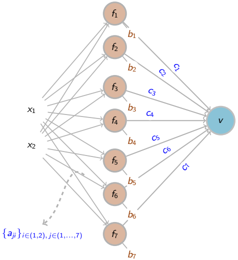
Neural Network Approximation

Theorem

Let $v, \tilde{v} \in \mathbb{R}_{max}[\mathbf{x}]$ 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.

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

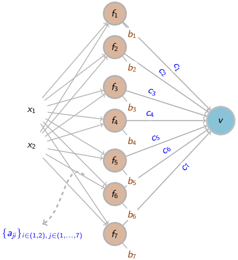
- ✓ Geometrical approximation problem.
- ✓ **Goal:** approximate the zonotopes.



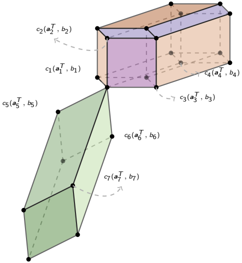
(a) Original network

✓ Applies only to networks with one output neuron.

Zonotope K-means



(a) Original network

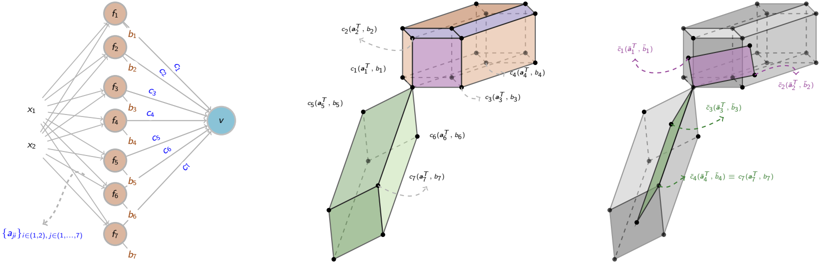


(b) Original zonotopes

✓ Applies only to networks with one output neuron.

Zonotope K-means

1. Split zonotope generators into positive and negative.

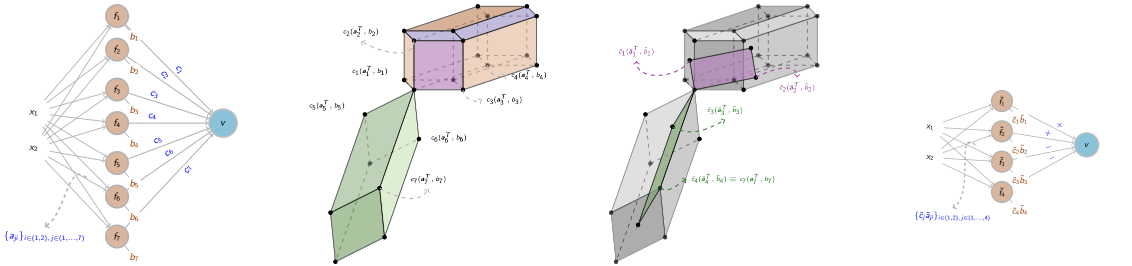


(a) Original network (b) Original zonotopes (c) Resulting zonotopes.

✓ 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.



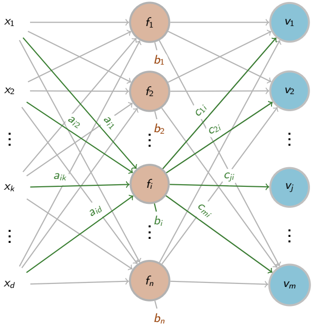
(a) Original network (b) Original zonotopes (c) Resulting zonotopes. (d) Compressed network.

✓ Applies only to networks with one output neuron.

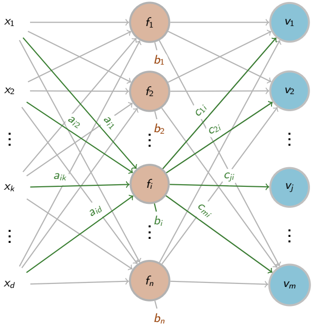
Zonotope K-means

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

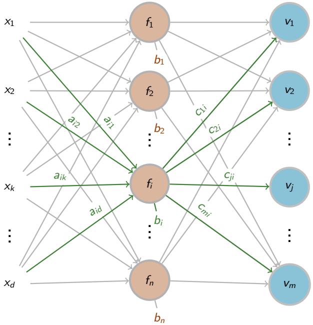


- Zonotope K-means doesn't generalize directly to multiple output nodes.



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

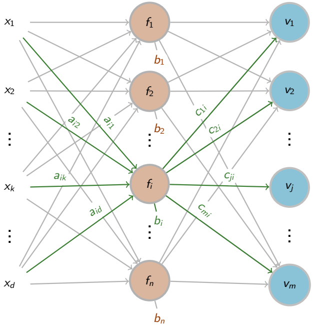
Neural Path K-means



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

Neural Path K-means

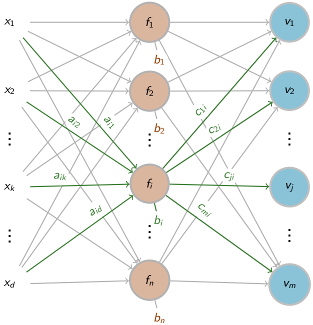
1. For each node form the vector of weights of incident edges.



- 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.



- 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

$$\frac{1}{\rho} \cdot \max_{\boldsymbol{x} \in \mathcal{B}} |v(\boldsymbol{x}) - \tilde{v}(\boldsymbol{x})| \leq K \cdot \delta_{\max} + \left(1 - \frac{1}{N_{\max}}\right) \sum_{i=1}^{n} |c_i| \| \left(\boldsymbol{a}_i^T,\, b_i\right) \|$$

Neural Path K-means Bound

$$\frac{1}{\rho} \cdot \max_{\boldsymbol{x} \in \mathcal{B}} \|v(\boldsymbol{x}) - \tilde{v}(\boldsymbol{x})\|_1 \leq \sqrt{m} K \delta_{\max}^2 + \sqrt{m} \left(1 - \frac{1}{N_{\max}}\right) \sum_{i=1}^{n} \|C_{:,i}\| \left\| \left(\boldsymbol{a}_i^T,\, b_i\right) \right\| + \frac{\sqrt{m}\delta_{\max}}{N_{\min}} \sum_{i=1}^{n} \left(\left\| \left(\boldsymbol{a}_i^T,\, b_i\right) \right\| + \|C_{:,i}\| \right) + \sum_{j=1}^{m} \sum_{i \in \mathcal{N}_j} |c_{ji}| \left\| \left(\boldsymbol{a}_i^T,\, b_i\right) \right\|$$

- ✓ 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.

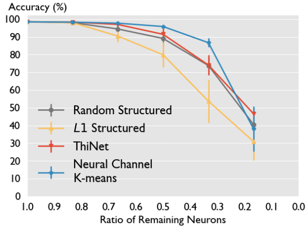
✓ Binary classification tasks.

Percentage of Remaining Neurons	MNIST 3/5			MNIST 4/9		
	Smyrnis et al., 2020	Zonotope K-means	Neural Path K-means	Smyrnis et al., 2020	Zonotope K-means	Neural Path K-means
100% (Original)	99.18 $\pm$ 0.27	99.38 $\pm$ 0.09	99.38 $\pm$ 0.09	99.53 $\pm$ 0.09	99.53 $\pm$ 0.09	99.53 $\pm$ 0.09
1%	99.11 $\pm$ 0.36	99.39 $\pm$ 0.05	99.32 $\pm$ 0.03	99.01 $\pm$ 0.09	99.46 $\pm$ 0.05	99.35 $\pm$ 0.17
0.3%	99.18 $\pm$ 0.36	99.25 $\pm$ 0.37	99.19 $\pm$ 0.41	98.81 $\pm$ 0.09	98.22 $\pm$ 1.38	98.22 $\pm$ 1.33

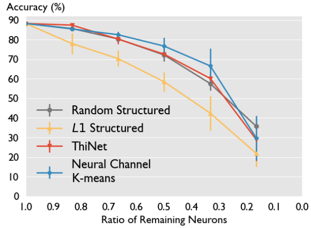
✓ Multiclass classification tasks.

Percentage of Remaining Neurons	MNIST		Fashion-MNIST	
	Smyrnis and Maragos, 2020	Neural Path K-means	Smyrnis and Maragos, 2020	Neural Path K-means
100% (Original)	98.60 $\pm$ 0.03	98.61 $\pm$ 0.11	88.66 $\pm$ 0.54	89.52 $\pm$ 0.19
10%	93.48 $\pm$ 2.57	96.89 $\pm$ 0.55	80.43 $\pm$ 3.27	86.04 $\pm$ 0.94
5%	92.93 $\pm$ 2.59	96.31 $\pm$ 1.29	–	83.68 $\pm$ 1.06

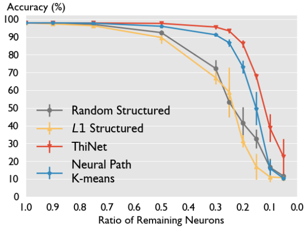
Experimental Evaluation II: Comparison with Thinet and baselines.



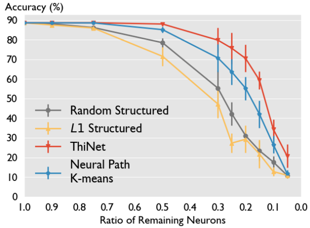
(a) LeNet5, MNIST



(b) LeNet5, F-MNIST



(c) deepNN, MNIST



(d) deepNN, F-MNIST

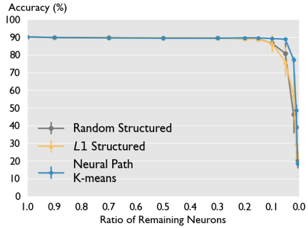
LeNet5

- ✓ 1 hidden layer with 84 neurons.

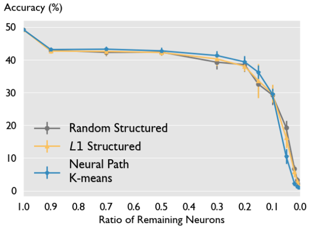
Custom deep network

- ✓ 3 hidden layers.

Experimental Evaluation III: Larger datasets



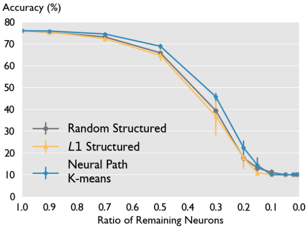
(a) CIFAR-VGG, CIFAR10



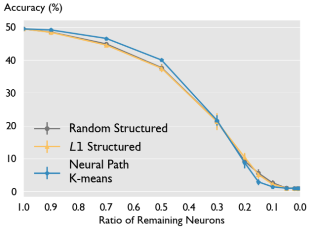
(b) CIFAR-VGG, CIFAR100

CIFAR-VGG

✓ 1 hidden layer of size 512.



(c) AlexNet, CIFAR10



(d) AlexNet, 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