Tropical varieties of neural networks

Yue Ren (Swansea University)

joint work with:

- Guido Montúfar (UC LA & MPI Leipzig)
- Leon Zhang (UC Berkeley)

23 September 2020

The expressivity of neural networks

Definition

A rectified linear unit (ReLU) network is a function of the form

$$\Phi: \quad \mathbb{R}^{n_0} \xrightarrow{\varphi_1} \mathbb{R}^{n_1} \xrightarrow{\varphi_2} \cdots \xrightarrow{\varphi_{L-1}} \mathbb{R}^{n_{L-1}} \xrightarrow{\varphi_L} \mathbb{R}^{n_L},$$

where $\varphi_i(x) = \max(W_i \cdot x + b_i, 0)$ for some $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ and $b_i \in \mathbb{R}^{n_i}$, and $\max(\cdot)$ denotes componentwise maximum.

The expressivity of neural networks

Definition

A rectified linear unit (ReLU) network is a function of the form

$$\Phi\colon \quad \mathbb{R}^{n_0} \xrightarrow{\varphi_1} \mathbb{R}^{n_1} \xrightarrow{\varphi_2} \cdots \xrightarrow{\varphi_{L-1}} \mathbb{R}^{n_{L-1}} \xrightarrow{\varphi_L} \mathbb{R}^{n_L},$$

where $\varphi_i(x) = \max(W_i \cdot x + b_i, 0)$ for some $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ and $b_i \in \mathbb{R}^{n_i}$, and $\max(\cdot)$ denotes componentwise maximum.

A maxout network of rank r is a function $\Phi\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ of the same form, except $\varphi_i(x) = \max(W_{i,1} \cdot x + b_{i,1}, \ldots, W_{i,r} \cdot x + b_{i,r})$.

The expressivity of neural networks

Definition

A rectified linear unit (ReLU) network is a function of the form

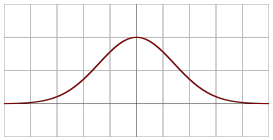
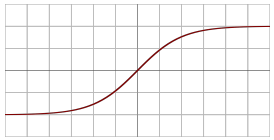
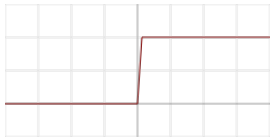
$$\Phi\colon \quad \mathbb{R}^{n_0} \xrightarrow{\varphi_1} \mathbb{R}^{n_1} \xrightarrow{\varphi_2} \cdots \xrightarrow{\varphi_{L-1}} \mathbb{R}^{n_{L-1}} \xrightarrow{\varphi_L} \mathbb{R}^{n_L},$$

where $\varphi_i(x) = \max(W_i \cdot x + b_i, 0)$ for some $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ and $b_i \in \mathbb{R}^{n_i}$, and $\max(\cdot)$ denotes componentwise maximum.

A maxout network of rank r is a function $\Phi\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ of the same form, except $\varphi_i(x) = \max(W_{i,1} \cdot x + b_{i,1}, \ldots, W_{i,r} \cdot x + b_{i,r})$.

Not covered in this talk:

- activations besides $\max(\cdot, 0)$:


gaussian

tanh

step

- convolutional / recurrent neural networks.

The expressivity of neural networks

Definition

A linear region of $\Phi\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ is a maximal connected subset of $\mathbb{R}^n$ on which Φ is linear.

The expressivity of neural networks

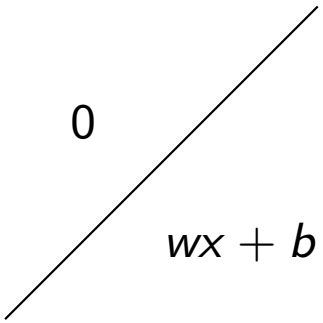
Definition

A linear region of $\Phi\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ is a maximal connected subset of $\mathbb{R}^n$ on which Φ is linear.

The maximal number of linear regions of a class of neural network is a measure of its capacity. The more it has, the better it approximates.

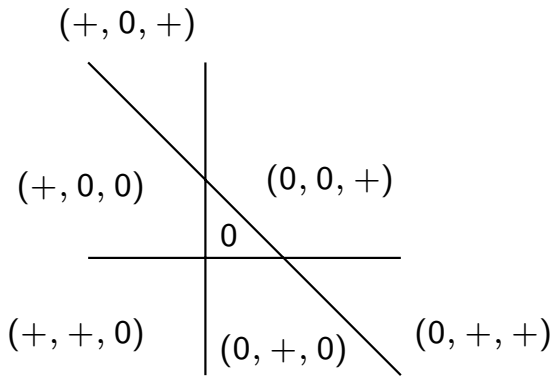
$$\Phi\colon \mathbb{R}^2 \longrightarrow \mathbb{R}$$

$$x \longmapsto \max(w \cdot x + b, 0)$$



$$\Phi\colon \mathbb{R}^2 \longrightarrow \mathbb{R}^3 \longrightarrow \mathbb{R}$$

$$x \longmapsto \begin{pmatrix} \max(w_1 \cdot x + b_1, 0) \\ \max(w_2 \cdot x + b_2, 0) \\ \max(w_3 \cdot x + b_3, 0) \end{pmatrix}$$



The expressivity of neural networks

Definition

A linear region of $\Phi\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ is a maximal connected subset of $\mathbb{R}^n$ on which Φ is linear.

The maximal number of linear regions of a class of neural network is a measure of its capacity. The more it has, the better it approximates.

Important questions

1. What is the maximal number of linear regions for a fixed architecture (= activation, number of levels, nodes per level)?
2. Which choices of parameters realize the maximal number?
3. What is the expected number of linear regions?

The expressivity of neural networks

Definition

A linear region of $\Phi\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ is a maximal connected subset of $\mathbb{R}^n$ on which Φ is linear.

The maximal number of linear regions of a class of neural network is a measure of its capacity. The more it has, the better it approximates.

Important questions

1. What is the maximal number of linear regions for a fixed architecture (= activation, number of levels, nodes per level)?
2. Which choices of parameters realize the maximal number?
3. What is the expected number of linear regions?

Motivation

1. Initialization of networks.
2. Training of networks.

State of the art on ReLU networks

Theorem (Zaslavsky 1975)

An generic arrangement of n_1 hyperplanes in $\mathbb{R}^{n_0}$ has $\sum_{j=0}^{n_0} \binom{n_1}{j}$ regions.

Corollary

A shallow ReLU network $\Phi\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, x \mapsto (\max(w_i \cdot x + b_i, 0)_{i=1,\dots,n_1}$ has at most $\sum_{j=0}^{n_0} \binom{n_1}{j}$ linear regions. The bound is sharp and attained by generic choices of parameters.

State of the art on ReLU networks

Theorem (Zaslavsky 1975)

An generic arrangement of n_1 hyperplanes in $\mathbb{R}^{n_0}$ has $\sum_{j=0}^{n_0} \binom{n_1}{j}$ regions.

Corollary

A shallow ReLU network $\Phi\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, x \mapsto (\max(w_i \cdot x + b_i, 0)_{i=1,\dots,n_1}$ has at most $\sum_{j=0}^{n_0} \binom{n_1}{j}$ linear regions. The bound is sharp and attained by generic choices of parameters.

Theorem (Montúfar-Pascanu-Cho-Bengio 2014)

There are deep ReLU networks $\Phi\colon \mathbb{R}^{n_0} \to \cdots \to \mathbb{R}^{n_L}$ with

$$\left(\prod_{i=1}^{L-1} \left\lfloor \frac{n_i}{n_0} \right\rfloor^{n_0}\right) \cdot \sum_{j=0}^{n_0} \binom{n_L}{j} \underset{\substack{\uparrow \\ \text{in case } n_i = n \text{ for } i>0}}{\in} \Omega\left((n/n_0)^{(L-1)n_0} n^{n_0}\right)$$

linear regions.

State of the art on ReLU networks

Corollary

A shallow ReLU network $\Phi\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_1}, x \mapsto (\max(w_i \cdot x + b_i, 0)_{i=1,\dots,n_1}$ has at most $\sum_{j=0}^{n_0} \binom{n_1}{j}$ linear regions. The bound is sharp and attained by generic choices of parameters.

Theorem (Montúfar-Pascanu-Cho-Bengio 2014)

There are deep ReLU networks $\Phi\colon \mathbb{R}^{n_0} \to \cdots \to \mathbb{R}^{n_L}$ with

$$\left(\prod_{i=1}^{L-1} \left\lfloor \frac{n_i}{n_0} \right\rfloor^{n_0}\right) \cdot \sum_{j=0}^{n_0} \binom{n_L}{j} \underset{\substack{\uparrow \\ \text{in case } n_i = n \text{ for } i>0}}{\in} \Omega\left((n/n_0)^{(L-1)n_0} n^{n_0}\right)$$

linear regions.

Observation (Hanin-Rolnick 2019)

Deep ReLU networks have surprisingly few linear regions ($\in \Omega(n_0^{n_0+\cdots+n_L})$).

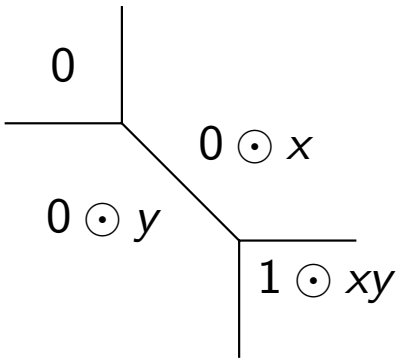
Tropical polynomials and raised Newton polytopes

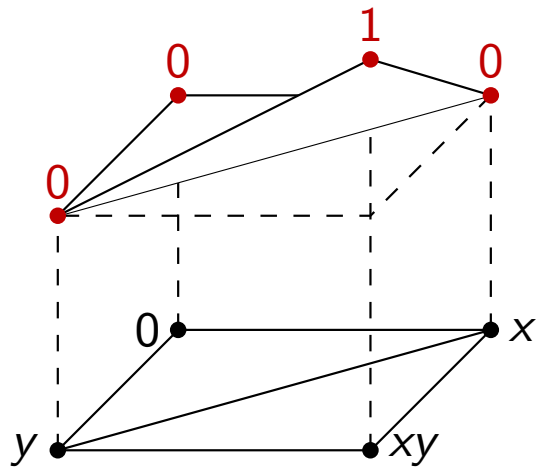
Recall

tropical numbers: $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$, $a \oplus b := \max(a, b)$, $a \odot b := a + b$

tropical r-nomials $\longleftrightarrow$ mixed subdivisions of r points

$0 \oplus 0 \odot x \oplus 0 \odot y \oplus 1 \odot xy$





linear regions $\longleftrightarrow$ vertices of the mixed subdivision

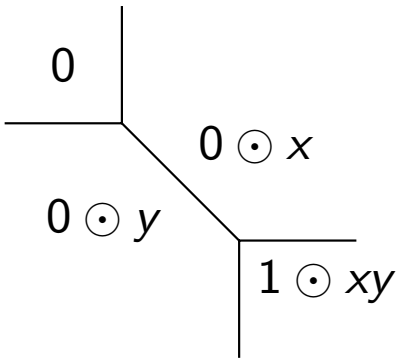
Tropical polynomials and raised Newton polytopes

Recall

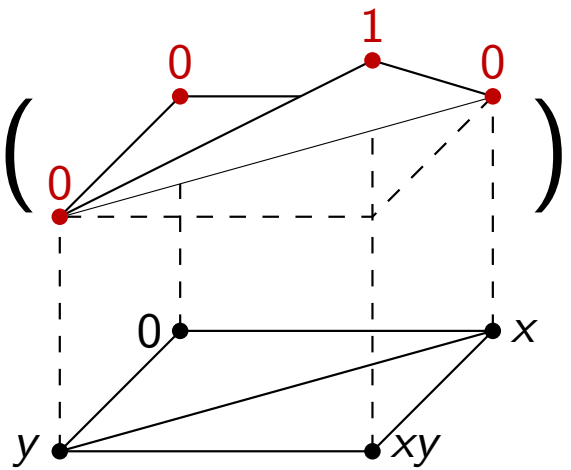
tropical numbers: $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\}$, $a \oplus b := \max(a, b)$, $a \odot b := a + b$

tropical r-nomials $\longleftrightarrow$ mixed subdivisions of r points

$0 \oplus 0 \odot x \oplus 0 \odot y \oplus 1 \odot xy$



raised Newton polytope $= \mathrm{Conv}\Big($



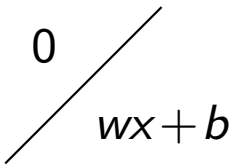
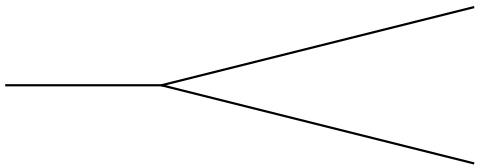
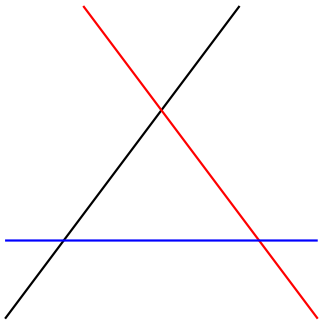
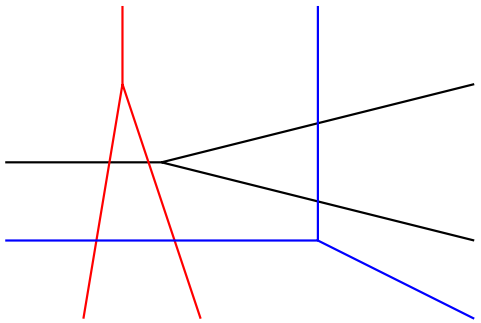
$\Big)$

linear regions $\longleftrightarrow$ vertices of the mixed subdivision

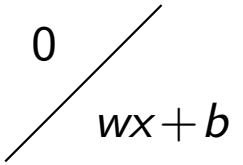
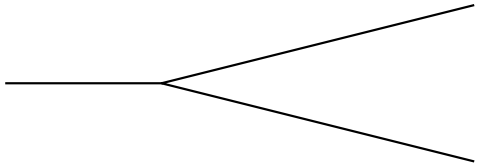
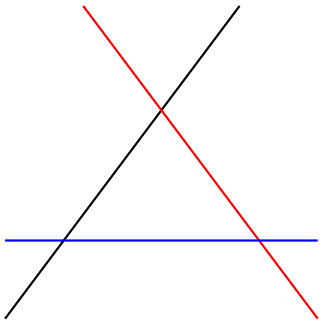
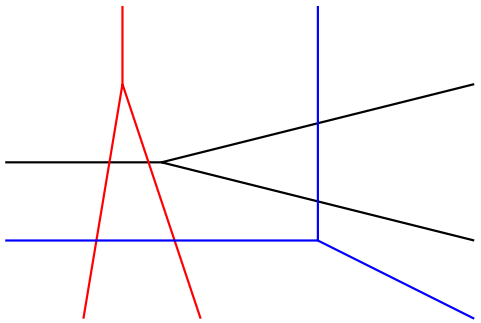
$\|$

lower vertices of raised Newton polytope

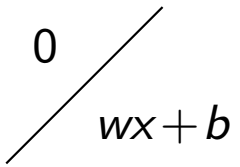
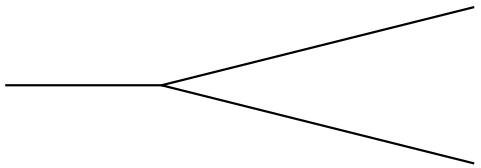
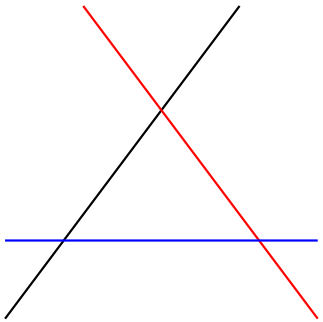
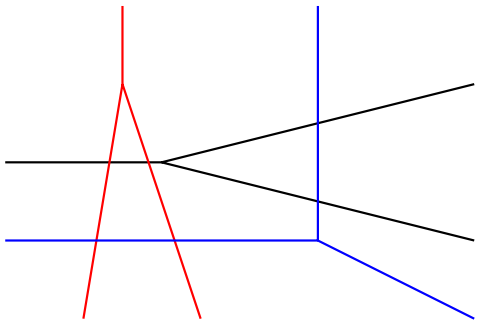
Tropical polynomials and raised Newton polytopes

	ReLU	maxout
single unit $\Phi: \mathbb{R}^n \to \mathbb{R}$	 hyperplane	 varieties of tropical r-nomials
multi units $\Phi: \mathbb{R}^n \to \mathbb{R}^k$	 arrangement of hyperplanes	 arrangement of trop varieties
# regions	**Zavlasky**	**???**

Tropical polynomials and raised Newton polytopes

	ReLU	maxout
single unit $\Phi: \mathbb{R}^n \to \mathbb{R}$	 hyperplane	 varieties of tropical r-nomials
multi units $\Phi: \mathbb{R}^n \to \mathbb{R}^k$	 arrangement of hyperplanes	 arrangement of trop varieties $\longleftrightarrow$ product of trop polys
# regions	**Zavlasky**	**???**

Tropical polynomials and raised Newton polytopes

	ReLU	maxout
single unit $\Phi: \mathbb{R}^n \to \mathbb{R}$	 hyperplane	 varieties of tropical r-nomials
multi units $\Phi: \mathbb{R}^n \to \mathbb{R}^k$	 arrangement of hyperplanes	 arrangement of trop varieties $\longleftrightarrow$ Minkowski sum of raised Newton polys $\updownarrow$ product of trop polys
# regions	**Zavlasky**	**???**

Upper bound theorem for Minkowski sums

Question

Let $P_1, \ldots, P_m \subseteq \mathbb{R}^n$ be convex polytopes with r vertices each.
How many vertices are there in $P_1 + \cdots + P_k$?

Upper bound theorem for Minkowski sums

Question

Let $P_1, \ldots, P_m \subseteq \mathbb{R}^n$ be convex polytopes with r vertices each.
How many vertices are there in $P_1 + \cdots + P_k$?

Answer 1

If $m < n$, the the number vertices of $P_1 + \cdots + P_m$ is at most r^m.
The bound is sharp.

Upper bound theorem for Minkowski sums

Question

Let $P_1, \ldots, P_m \subseteq \mathbb{R}^n$ be convex polytopes with r vertices each.
How many vertices are there in $P_1 + \cdots + P_k$?

Answer 1

If $m < n$, the the number vertices of $P_1 + \cdots + P_m$ is at most r^m.
The bound is sharp.

Answer 2 (Fukuda-Weibel, Karavelas-Konaxis-Tzanaki, Adiprasito-Sanyal; 2007 onward)

If $m \geq n$, the the number vertices of $P_1 + \cdots + P_m$ is at most

$$\binom{m-1}{n} + \sum_{j=0}^{n} \binom{m}{j} (r-1)^j$$

The bound is sharp and attained by *Minkowski-neighbourly* polytopes.

Upper bound theorem for shallow maxout networks

Applying the upper bound theorem to raised Newton polytopes yields:

Corollary

Let $\Phi\colon \mathbb{R}^n \to \mathbb{R}^m$ be a shallow maxout network of rank r. Then the number of linear regions of Φ is sharply bounded above by

$$\begin{cases} r^m & \text{if } m < n, \\ \approx \binom{m-1}{n} + \sum_{j=0}^{n} \binom{m}{j}(r-1)^j & \text{if } m \geq n. \end{cases}$$

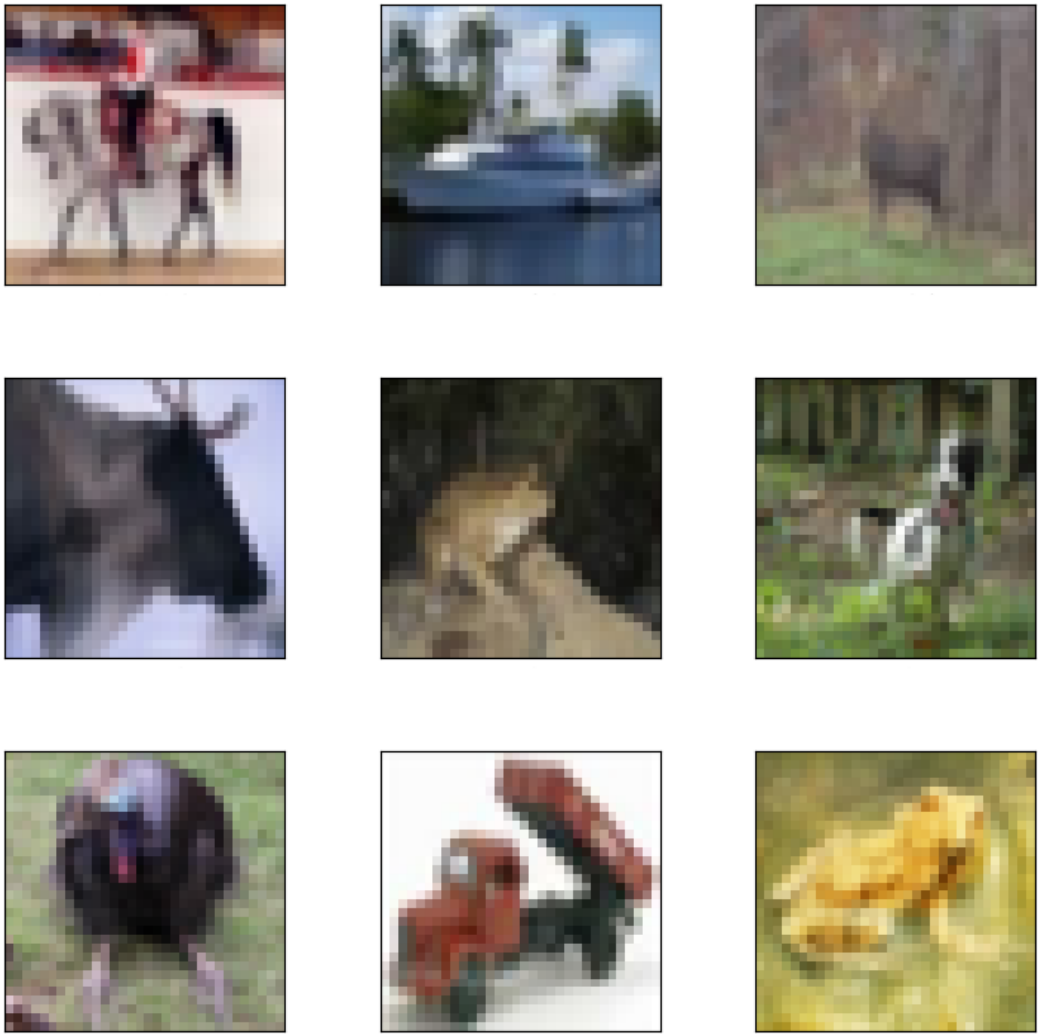
Corollary

Let $\Phi\colon \mathbb{R}^n \to \mathbb{R}^m$ be a shallow maxout network of rank r with weights and biases sampled i.i.d. from $\mathcal{N}_{0,1}$. Then the expected number of linear regions is

$$\begin{cases} \in O(\log(r)^{n^2} m^n) & \text{if } m < n, \\ \in O(\log(r)^{m \cdot n}) & \text{if } m \geq n. \end{cases}$$

First experimental results

We train on the CIFAR-10 dataset, which consists of 32x32px color images of airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks.



First experimental results

We train on the CIFAR-10 dataset, which consists of 32x32px color images of airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

In the following let Φ denote a maxout network of rank 5 of the form

$$\Phi\colon \mathbb{R}^{3\times 32\times 32} \longrightarrow \underbrace{\mathbb{R}^n \longrightarrow \cdots \longrightarrow \mathbb{R}^n}_{L \text{ hidden layers}} \longrightarrow \mathbb{R}^{10}$$

We initialize its parameters in two ways:

1. maximizing the number of linear regions per layer (Fukuda-Weibel),
2. sampled i.i.d. from $\mathcal{N}_{0,1}$.

First experimental results

We train on the CIFAR-10 dataset, which consists of 32x32px color images of airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

In the following let Φ denote a maxout network of rank 5 of the form

$$\Phi\colon \mathbb{R}^{3\times 32\times 32} \longrightarrow \underbrace{\mathbb{R}^n \longrightarrow \cdots \longrightarrow \mathbb{R}^n}_{L \text{ hidden layers}} \longrightarrow \mathbb{R}^{10}$$

We initialize its parameters in two ways:

1. maximizing the number of linear regions per layer (Fukuda-Weibel),
2. sampled i.i.d. from $\mathcal{N}_{0,1}$.

Difference in accuracy after training for 5 epochs:

	L=2	L=3	L=4	L=5	L=6
n=200	-0.01%	0.30%	-1.33%	-3.98%	-1.04%
n=400	0.04%	0.70%	0.70%	-1.44%	-0.15%
n=800	0.40%	1.00%	1.69%	2.23%	3.50%
n=1600	0.73%	1.09%	2.14%	2.07%	2.75%

(average of 10000 runs)

First experimental results and ongoing work

Ongoing

Improve method for initializing maxout network

$$\Phi: \quad \underbrace{\mathbb{R}^{n_0} \xrightarrow{\varphi_1} \mathbb{R}^{n_1} \xrightarrow{\varphi_2} \cdots \xrightarrow{\varphi_i} \mathbb{R}^{n_i}}_{=:\Phi_i} \xrightarrow{\varphi_{i+1}} \cdots \xrightarrow{\varphi_{L-1}} \mathbb{R}^{n_{L-1}} \xrightarrow{\varphi_L} \mathbb{R}^{n_L},$$

so that

- number of linear regions as large as possible

First experimental results and ongoing work

Ongoing

Improve method for initializing maxout network

$$\Phi: \quad \underbrace{\mathbb{R}^{n_0} \xrightarrow{\varphi_1} \mathbb{R}^{n_1} \xrightarrow{\varphi_2} \cdots \xrightarrow{\varphi_i} \mathbb{R}^{n_i}}_{=:\Phi_i} \xrightarrow{\varphi_{i+1}} \cdots \xrightarrow{\varphi_{L-1}} \mathbb{R}^{n_{L-1}} \xrightarrow{\varphi_L} \mathbb{R}^{n_L},$$

so that

- number of linear regions as large as possible
- (Hanin-Rolnick 2018) $\mathbb{E}(\Phi_i(\mathcal{N}_{0,1})) = 0$ and $\sigma(\Phi_i(\mathcal{N}_{0,1})) =: \sigma_i$ not growing exponentially

First experimental results and ongoing work

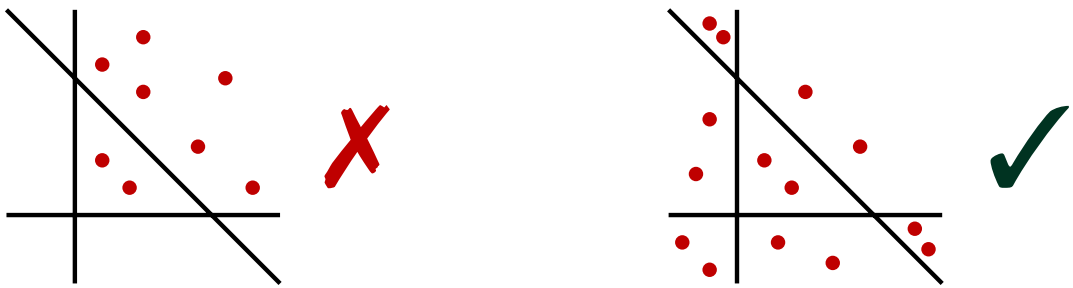
Ongoing

Improve method for initializing maxout network

$$\Phi: \quad \underbrace{\mathbb{R}^{n_0} \xrightarrow{\varphi_1} \mathbb{R}^{n_1} \xrightarrow{\varphi_2} \cdots \xrightarrow{\varphi_i} \mathbb{R}^{n_i}}_{=:\Phi_i} \xrightarrow{\varphi_{i+1}} \cdots \xrightarrow{\varphi_{L-1}} \mathbb{R}^{n_{L-1}} \xrightarrow{\varphi_L} \mathbb{R}^{n_L},$$

so that

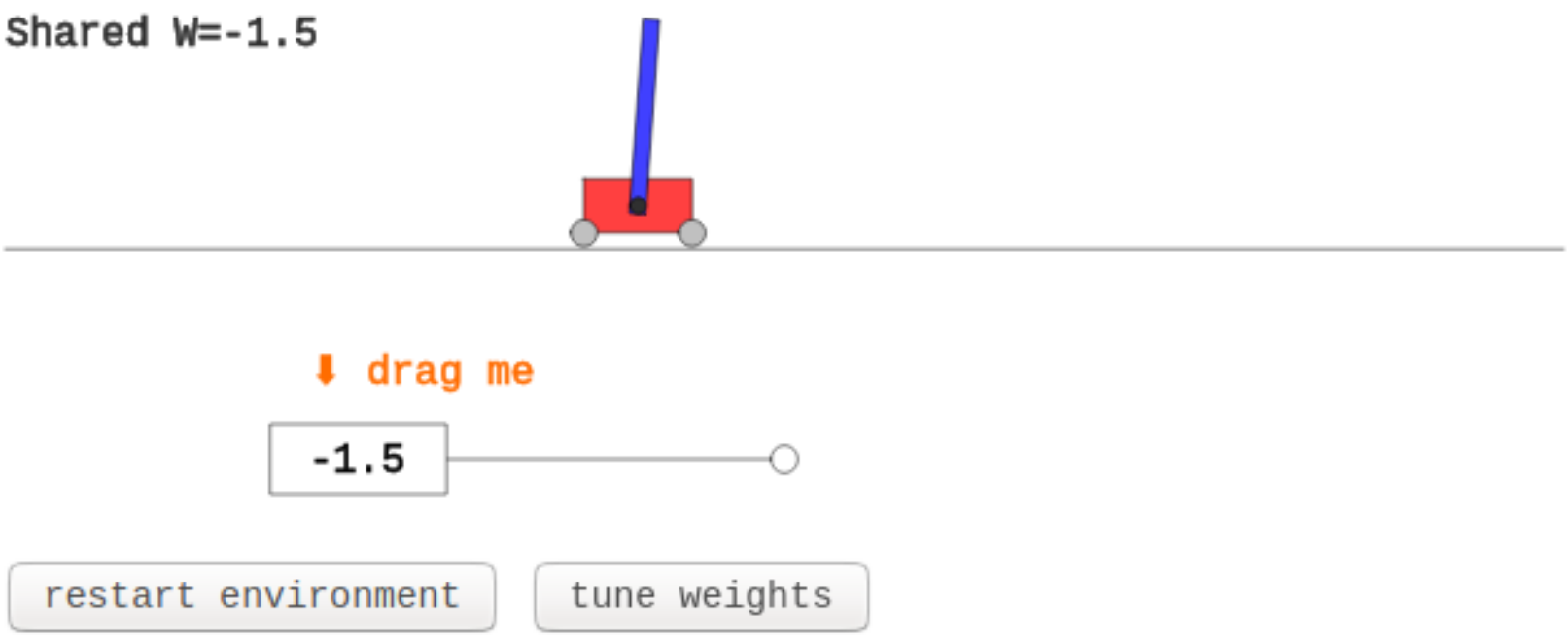
- number of linear regions as large as possible
- (Hanin-Rolnick 2018) $\mathbb{E}(\Phi_i(\mathcal{N}_{0,1})) = 0$ and $\sigma(\Phi_i(\mathcal{N}_{0,1})) =: \sigma_i$ not growing exponentially
- (Steinwart 2019) $\mathrm{Trop}(\varphi_{i+1})$ *well spaced* w.r.t. $\mathcal{N}_{0,\sigma_i}$



Bonus: weight agnostic networks and tropical linear spaces

Observation (Gaier-Ha 2019)

Certain networks architectures perform well on certain tasks even without training the weights (see `weightagnostic.github.io/`).



Interactive Demo

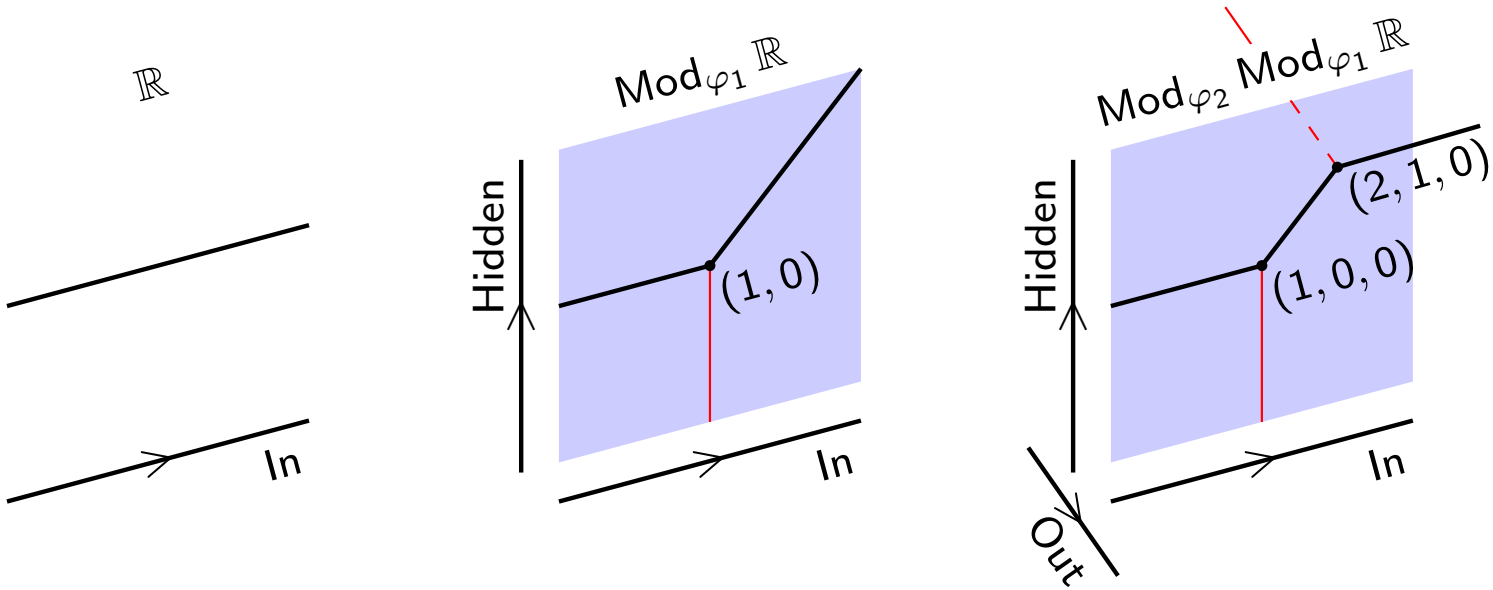
A weight agnostic neural network performing *CartpoleSwingup* task. Drag the slider to control the weight parameter and observe the performance at various shared weight parameters. You can also fine-tune the individual weights of all connections in this demo.

Bonus: weight agnostic networks and tropical linear spaces

Observation

Maxout networks whose weights are unit vectors ($w \cdot x = x_i$) correspond to tropical linear spaces of dimension n_0 in $\mathbb{R}^{n_0+\cdots+n_L}$.

$$\begin{aligned} \Phi: \quad & \mathbb{R} \xrightarrow{\varphi_1} \mathbb{R} \xrightarrow{\varphi_2} \mathbb{R} \\ & x \longmapsto x \oplus 0 \\ & y \longmapsto -1 \odot y \oplus 0 \end{aligned}$$



Bonus: weight agnostic networks and tropical linear spaces

Observation (Gaier-Ha 2019)

Certain networks architectures perform well on certain tasks even without training the weights (see `weightagnostic.github.io/`).

Observation

Maxout networks whose weights are unit vectors ($w \cdot x = x_i$) correspond to tropical linear spaces of dimension n_0 in $\mathbb{R}^{n_0+\cdots+n_L}$.

Speyer's f-Vector Theorem (2008-2009)

Tropical linear spaces have at most $\Omega(n_0^{n_0})$ maximal cells.

Corollary

Maxout networks whose weights are unit vectors only has at most $\Omega(n_0^{n_0})$ linear regions (compared to $\Omega(n_0^{n_1+\cdots+n_L})$ for unrestricted weights).

Summary and outlook

What has been done:

- First sharp bounds on the max number of linear regions for shallow maxout networks based on recent results in convex and tropical geometry.
- New initialization strategy for deep maxout networks which yields better performing networks after training.

Experiments were made with the help of:

Summary and outlook

What has been done:

- First sharp bounds on the max number of linear regions for shallow maxout networks based on recent results in convex and tropical geometry.
- New initialization strategy for deep maxout networks which yields better performing networks after training.

Experiments were made with the help of:

What will be done:

- Comprehensive study of deep networks using tropical geometry
  - Combinatorics of tropical fewnomial varieties
  - Upper bound theorem for Minkowski sums of degenerate polytopes
- Explore implications for the training of neural networks.

Image credits

Slide 2 Activation plots by Laughsinthestocks taken from Wikipedia (link) and licensed under CC BY-SA 4.0.

Slide 7 CIFAR-10 images taken from the tensorflow catalog (link) and licensed under CC BY 4.0.