Stanford University

Fast Convex Optimization for Two-Layer ReLU Networks:

Equivalent Model Classes and Cone Decompositions

Aaron Mishkin   Arda Sahiner   Mert Pilanci



Overview

Problem: Training neural networks is slow and sensitive.

Overview

Problem: Training neural networks is slow and sensitive.

Our Contribution: use convex reformulations for training.

Overview

Problem: Training neural networks is slow and sensitive.

Our Contribution: use convex reformulations for training.

1. **Equivalent Model Classes**: new convex reformulations of neural networks.

Overview

Problem: Training neural networks is slow and sensitive.

Our Contribution: use convex reformulations for training.

1. **Equivalent Model Classes**: new convex reformulations of neural networks.
2. **Cone Decompositions**: new connections between our convex training programs.

Overview

Problem: Training neural networks is slow and sensitive.

Our Contribution: use convex reformulations for training.

1. **Equivalent Model Classes**: new convex reformulations of neural networks.
2. **Cone Decompositions**: new connections between our convex training programs.
3. **Algorithms**: robust, tuning-free, and fast algorithms leveraging these connections.

I. 10 Years of Neural Nets

Context: Ten Years Since AlexNet

10 Years Ago: AlexNet won ILSVRC 2012 and started the modern "deep learning" movement in ML.

https://image-net.org/challenges/LSVRC/2012/results.html#abstract

Context: Ten Years Since AlexNet

10 Years Ago: AlexNet won ILSVRC 2012 and started the modern "deep learning" movement in ML.

AlexNet improved over the next best model by $\approx 10\%$ (top-5).

Key Techniques:

- "a large, deep convolutional neural network".
- "a very efficient GPU implementation of convolutional nets".
- "'dropout', a recently-developed regularization method that proved to be very effective."

https://image-net.org/challenges/LSVRC/2012/results.html#abstract

Context: ImageNet Today

AlexNet won with 84.69% top-five accuracy [KSH12].

Context: ImageNet Today

AlexNet won with 84.69% top-five accuracy [KSH12].

Today, models get 99.02% top-5 accuracy [Yua+21]!

(Using all sorts of tricks like pre-training, transformers, etc.)

Context: Slides from the Winners

How monster is the resulting feature sets

Compare to PASCAL classification task:

	# of training data	# of class	(assumed) training time
PASCAL	10,103	20	1 hour
ImageNet	1,200,000	1000	6000 hours = 250 days*
Ratio	120	50	6000

* Not including file I/O, networking delay, etc

☹ Life is short -- we need efficient training algorithms

What model goes here?

https://www.image-net.org/static_files/files/ILSVRC2010_NEC-UIUC.pdf

Context: Slides from the Winners — Revealed!

How monster is the resulting feature sets

Compare to PASCAL classification task:

	# of training data	# of class	(assumed) training time
PASCAL	10,103	20	1 hour
ImageNet	1,200,000	1000	6000 hours = 250 days*
Ratio	120	50	6000

* Not including file I/O, networking delay, etc

☹ Life is short -- we need efficient SVM training algorithms

https://www.image-net.org/static_files/files/ILSVRC2010_NEC-UIUC.pdf

Context: DALL·E 2

Generated by DALL·E 2

A bowl of soup that is a portal to another dimension as digital art.

https://openai.com/dall-e-2/

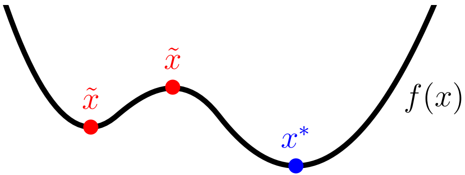
Context: Cost of Training DALL·E 2

DALL·E 2 has 5.5 billion parameters and took billions of Adam iterations to fit [Ram+22].

Context: Cost of Training DALL·E 2

DALL·E 2 has 5.5 billion parameters and took billions of Adam iterations to fit [Ram+22].

Main Challenge: neural networks are non-convex.



Context: Challenges Optimizing Neural Networks

Optimizing neural networks with SGD is hard!

- **Tuning**: step-size, momentum, batch-size, etc.
- **Model Churn**: new seed, different performance [Hen+18].
- **Certificates**: few/no guarantees.

Context: Challenges Optimizing Neural Networks

Optimizing neural networks with SGD is hard!

- **Tuning**: step-size, momentum, batch-size, etc.
- **Model Churn**: new seed, different performance [Hen+18].
- **Certificates**: few/no guarantees.

But these issues don't exist for convex models!

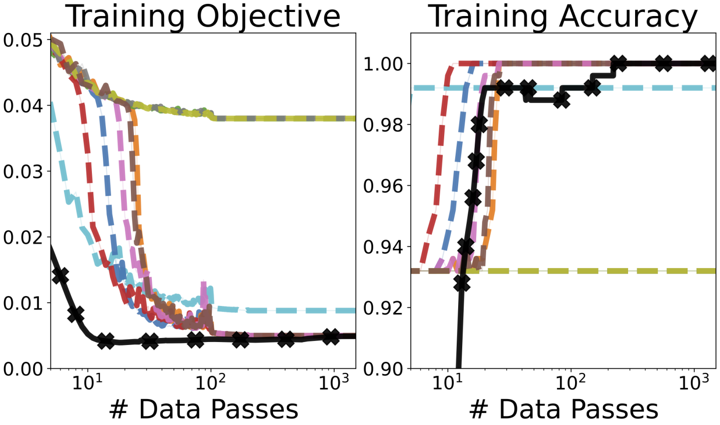
- **Tuning**: line-search, full-batch methods, acceleration, etc.
- **Model Churn**: strict/strong convexity gives uniqueness.
- **Certificates**: stationary points are global minima.

Context: Practical Challenges

Recovering a two-layer ReLU network from data generated by a two-layer ReLU network.

Context: Practical Challenges

Recovering a two-layer ReLU network from data generated by a two-layer ReLU network.



Context: Better Methods

We need better methods!

Context: Better Methods

We need better methods!

- **Stable** — No mysterious failure modes.

Context: Better Methods

We need better methods!

- **Stable** — No mysterious failure modes.
- **Tuning-Free** — No grid-search.

Context: Better Methods

We need better methods!

- **Stable** — No mysterious failure modes.
- **Tuning-Free** — No grid-search.
- **Robust** — Work on a variety of problems.

Context: Better Methods

We need better methods!

- **Stable** — No mysterious failure modes.
- **Tuning-Free** — No grid-search.
- **Robust** — Work on a variety of problems.
- **Fast** — better than $O(1/\sqrt{T})$.

II. Equivalent (Convex) Model Classes

Convex Reformulations: Flavor of Results

Basic Idea: We start with a non-convex optimization problem and derive an equivalent convex optimization problem.

Convex Reformulations: Flavor of Results

Basic Idea: We start with a non-convex optimization problem and derive an equivalent convex optimization problem.

Equivalent means:

- The global minima have the same values: $p^* = d^*$
- We can map a solution u^* for one problem into a solution v^* for the other.
- Call this our *solution mapping*.

Convex Reformulations: Two-Layer ReLU Networks

Non-Convex Problem

$$\min_{w,\alpha} \underbrace{\| \sum_{j=1}^{m} (Xw_j)_+ \alpha_j - y \|_2^2}_{\text{Squared Error}} + \underbrace{\lambda \sum_{j=1}^{m} \|w_j\|_2^2 + |\alpha_j|^2}_{\text{Weight Decay}},$$

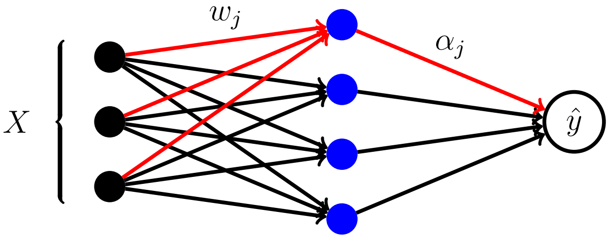
where $(x)_+ = \max\{x, 0\}$ is the ReLU activation.

Convex Reformulations: Two-Layer ReLU Networks

Non-Convex Problem

$$\min_{w,\alpha} \underbrace{\| \sum_{j=1}^{m} (Xw_j)_+ \alpha_j - y \|_2^2}_{\text{Squared Error}} + \underbrace{\lambda \sum_{j=1}^{m} \|w_j\|_2^2 + |\alpha_j|^2}_{\text{Weight Decay}},$$

where $(x)_+ = \max\{x, 0\}$ is the ReLU activation.



Convex Reformulations: Convex Problem

Convex Reformulation [PE20]

$$\min_u \| \sum_{j=1}^{p} D_j X(v_j - w_j) - y\|_2^2 + \lambda \sum_{j=1}^{p} \|v_j\|_2 + \|w_j\|_2$$
$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{w : (2D_j - I)Xw \geq 0\},$$

where $D_j = \text{diag}[\mathbb{1}(X g_j \geq 0)]$.

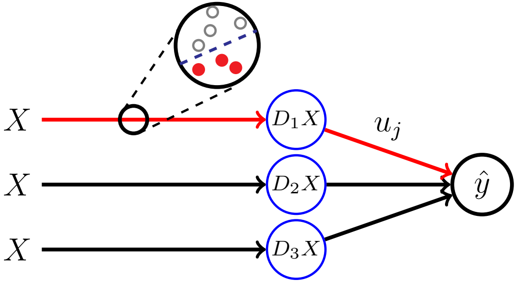
Convex Reformulations: Convex Problem

Convex Reformulation [PE20]

$$\min_{u} \| \sum_{j=1}^{p} D_j X(v_j - w_j) - y\|_2^2 + \lambda \sum_{j=1}^{p} \|v_j\|_2 + \|w_j\|_2$$

$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{w : (2D_j - I)Xw \geq 0\},$$

where $D_j = \mathrm{diag}[\mathbb{1}(Xg_j \geq 0)]$.



Convex Reformulations: Breaking it Down

$$\min_{u} \| \sum_{j=1}^{p} D_j X(v_j - w_j) - y\|_2^2 + \lambda \sum_{j=1}^{p} \|v_j\|_2 + \|w_j\|_2$$
$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{w : (2D_j - I)Xw \geq 0\},$$

where $D_j = \text{diag}[\mathbb{1}(X g_j \geq 0)]$.

- D_j is a ReLU activation pattern induced by "gate" g_j.

Convex Reformulations: Breaking it Down

$$\min_{u} \| \sum_{j=1}^{p} D_j X(v_j - w_j) - y\|_2^2 + \lambda \sum_{j=1}^{p} \|v_j\|_2 + \|w_j\|_2$$
$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{w : (2D_j - I)Xw \geq 0\},$$

where $D_j = \text{diag}[\mathbb{1}(Xg_j \geq 0)]$.

- D_j is a ReLU activation pattern induced by "gate" g_j.
  - $[D_j]_{ii} = 1$ if $\langle x_i, g_i \rangle \geq 0$ and 0 otherwise.

Convex Reformulations: Breaking it Down

$$\min_{u} \| \sum_{j=1}^{p} D_j X (v_j - w_j) - y \|_2^2 + \lambda \sum_{j=1}^{p} \|v_j\|_2 + \|w_j\|_2$$

$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{w : (2D_j - I)Xw \geq 0\}$$

where $D_j = \text{diag}[\mathbb{1}(Xg_j \geq 0)]$.

- D_j is a ReLU activation pattern induced by "gate" g_j.
  - $[D_j]_{ii} = 1$ if $\langle x_i, g_i \rangle \geq 0$ and 0 otherwise.
- Weight-decay regularization turns into "group ℓ_1" penalty.

Convex Reformulations: Breaking it Down

$$\min_{u} \| \sum_{j=1}^{p} D_j X(v_j - w_j) - y\|_2^2 + \lambda \sum_{j=1}^{p} \|v_j\|_2 + \|w_j\|_2$$

$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{w : (2D_j - I)Xw \geq 0\},$$

where $D_j = \text{diag}[\mathbb{1}(Xg_j \geq 0)]$.

- D_j is a ReLU activation pattern induced by "gate" g_j.
  - $[D_j]_{ii} = 1$ if $\langle x_i, g_i \rangle \geq 0$ and 0 otherwise.
- Weight-decay regularization turns into "group ℓ_1" penalty.
- The constraint $v_j \in \mathcal{K}_j$ implies

$$(Xv_j)_+ = D_j X v_j.$$

That is, v_j has the activation encoded by D_j.

Convex Reformulations: Hardness

$$p = \left| \left\{ D_j = \mathsf{diag}[\mathbb{1}(Xg_j \geq 0)] : g_j \in \mathbb{R}^d \right\} \right|$$

Convex Reformulations: Hardness

$$p = \left| \left\{ D_j = \mathsf{diag}[\mathbb{1}(X g_j \geq 0)] : g_j \in \mathbb{R}^d \right\} \right|$$

The **convex program** is:

- Exponential in general: $p \in O(r \cdot (\frac{n}{r})^r)$, where $r = \mathsf{rank}(X)$.
  - Bound comes from theory of hyperplane arrangements [Win66].

Convex Reformulations: Hardness

$$p = \left| \left\{ D_j = \mathsf{diag}[\mathbb{1}(X g_j \geq 0)] : g_j \in \mathbb{R}^d \right\} \right|$$

The **convex program** is:

- Exponential in general: $p \in O(r \cdot (\frac{n}{r})^r)$, where $r = \mathsf{rank}(X)$.
  - Bound comes from theory of hyperplane arrangements [Win66].
- Highly structured — it's a (constrained) generalized linear model!

Convex Reformulations: Hardness

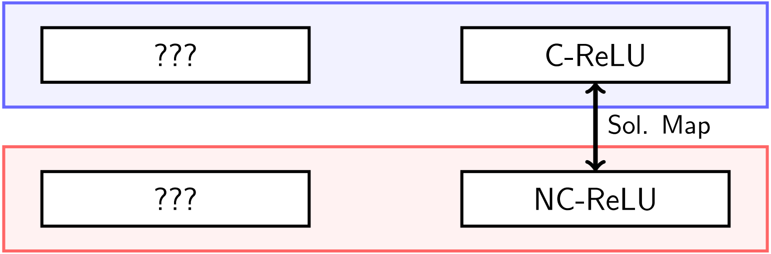
$$p = \left| \left\{ D_j = \mathsf{diag}[\mathbb{1}(X g_j \geq 0)] : g_j \in \mathbb{R}^d \right\} \right|$$

The **convex program** is:

- Exponential in general: $p \in O(r \cdot (\frac{n}{r})^r)$, where $r = \mathsf{rank}(X)$.
  - Bound comes from theory of hyperplane arrangements [Win66].
- Highly structured — it's a (constrained) generalized linear model!

We exchange one kind of hardness for another.

Convex Reformulations: Big Picture



Convex Reformulations: Unconstrained Relaxation

What can we do with the convex ReLU problem?

$$\textbf{C-ReLU}: \min_{u} \| \sum_{j=1}^{p} D_j X(v_j - w_j) - y\|_2^2 + \lambda \sum_{j=1}^{p} \|v_j\|_2 + \|w_j\|_2$$

$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{w : (2D_j - I)Xw \geq 0\},$$

Convex Reformulations: Unconstrained Relaxation

What can we do with the convex ReLU problem?

$$\textbf{C-ReLU} : \min_{u} \| \sum_{j=1}^{p} D_j X(v_j - w_j) - y\|_2^2 + \lambda \sum_{j=1}^{p} \|v_j\|_2 + \|w_j\|_2$$

$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{w : (2D_j - I)Xw \geq 0\},$$

Relaxation: drop the cone constraints and simplify to obtain,

$$\textbf{C-GReLU} : \min_{u} \| \sum_{j=1}^{p} D_j X u_j - y\|_2^2 + \lambda \sum_{j=1}^{p} \|u_j\|_2$$

Convex Reformulations: Unconstrained Relaxation

What can we do with the convex ReLU problem?

$$\textbf{C-ReLU} : \min_{u} \| \sum_{j=1}^{p} D_j X(v_j - w_j) - y\|_2^2 + \lambda \sum_{j=1}^{p} \|v_j\|_2 + \|w_j\|_2$$

$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{w : (2D_j - I)Xw \geq 0\},$$

Relaxation: drop the cone constraints and simplify to obtain,

$$\textbf{C-GReLU} : \min_{u} \| \sum_{j=1}^{p} D_j X u_j - y\|_2^2 + \lambda \sum_{j=1}^{p} \|u_j\|_2$$

What does it mean? Is it a neural network still?

Convex Reformulations: Gated ReLU Networks

Theorem 2.2 (informal): C-GReLU is equivalent to solving

$$\textbf{NC-GReLU}: \min_{W_1,\alpha} \frac{1}{2}\|\sum_{j=1}^{p} \phi_{g_j}(X, w_j)\alpha - y\|_2^2 + \frac{\lambda}{2}\sum_{j=1}^{p} \|w_j\|_2^2 + |\alpha_j|^2,$$

with the "Gated ReLU" [FMS19] activation function

$$\phi_g(X, u) = \text{diag}(\mathbb{1}(Xg \geq 0))Xu,$$

and gate vectors g_j such that

$$D_j = \text{diag}[\mathbb{1}(Xg_j \geq 0).$$

Convex Reformulations: Gated ReLU Networks

Theorem 2.2 (informal): C-GReLU is equivalent to solving

$$\textbf{NC-GReLU}: \min_{W_1, \alpha} \frac{1}{2} \| \sum_{j=1}^{p} \phi_{g_j}(X, w_j)\alpha - y\|_2^2 + \frac{\lambda}{2} \sum_{j=1}^{p} \|w_j\|_2^2 + |\alpha_j|^2,$$

with the "Gated ReLU" [FMS19] activation function

$$\phi_g(X, u) = \mathsf{diag}(\mathbb{1}(Xg \geq 0))Xu,$$

and gate vectors g_j such that

$$D_j = \mathsf{diag}[\mathbb{1}(Xg_j \geq 0).$$

Interpretation: if $u_j \notin \mathcal{K}_j$, then the activation must be decoupled from the linear mapping in the non-convex model.

Gated ReLU Networks: Proof Sketch

The proof reduces C-GReLU to NC-GReLU and vice-versa.

Roadmap:

1. Manipulate NC-GReLU to remove invariance to certain scale re-parameterizations.
2. Merge second-layer weights into first-layer weights.

Gated ReLU Networks: Proof Sketch

The proof reduces C-GReLU to NC-GReLU and vice-versa.

Roadmap:

1. Manipulate NC-GReLU to remove invariance to certain scale re-parameterizations.
2. Merge second-layer weights into first-layer weights.

The prediction function is

$$f_{w,\alpha}(X) = \sum_{j=1}^{p} \phi_{g_j}(X, w_j)\alpha$$

Gated ReLU Networks: Proof Sketch

The proof reduces C-GReLU to NC-GReLU and vice-versa.

Roadmap:

1. Manipulate NC-GReLU to remove invariance to certain scale re-parameterizations.
2. Merge second-layer weights into first-layer weights.

The prediction function is

$$\begin{aligned} f_{w,\alpha}(X) &= \sum_{j=1}^{p} \phi_{g_j}(X, w_j)\alpha \\ &= \sum_{j=1}^{p} \mathsf{diag}(\mathbb{1}(X g_j \geq 0)) X w_j \cdot \alpha_j. \end{aligned}$$

Gated ReLU Networks: Proof Sketch

The proof reduces C-GReLU to NC-GReLU and vice-versa.

Roadmap:

1. Manipulate NC-GReLU to remove invariance to certain scale re-parameterizations.
2. Merge second-layer weights into first-layer weights.

The prediction function is

$$\begin{aligned} f_{w,\alpha}(X) &= \sum_{j=1}^{p} \phi_{g_j}(X, w_j)\alpha \\ &= \sum_{j=1}^{p} \mathsf{diag}(\mathbb{1}(Xg_j \geq 0))Xw_j \cdot \alpha_j. \end{aligned}$$

Invariant to scale re-parameterizations of the form

$$w_j' = w_j \cdot \beta, \quad \alpha_j' = \frac{\alpha_j}{\beta_j}.$$

Gated ReLU Networks: Sketch Continued

Let (w^*, α^*) be a solution to NC-GReLU.

Gated ReLU Networks: Sketch Continued

Let (w^*, α^*) be a solution to NC-GReLU.

By Young's inequality,

$$2 \sum_{j=1}^{p} \|w_j^*\|_2 \, |\alpha_j^*| \leq \sum_{j=1}^{p} \|w_j^*\|_2^2 + |\alpha_j^*|^2$$

Gated ReLU Networks: Sketch Continued

Let (w^*, α^*) be a solution to NC-GReLU.

By Young's inequality,

$$2 \sum_{j=1}^{p} \|w_j^*\|_2 \, |\alpha_j^*| \leq \sum_{j=1}^{p} \|w_j^*\|_2^2 + |\alpha_j^*|^2$$

Equality is achieved for $w_j' = w_j^* \cdot \beta$, $\alpha_j' = \alpha_j^* / \beta$, where

$$\beta = \sqrt{\frac{|\alpha_j^*|}{\|w_j^*\|_2}}.$$

Gated ReLU Networks: Sketch Continued

Let (w^*, α^*) be a solution to NC-GReLU.

By Young's inequality,

$$2 \sum_{j=1}^{p} \|w_j^*\|_2 \, |\alpha_j^*| \leq \sum_{j=1}^{p} \|w_j^*\|_2^2 + |\alpha_j^*|^2$$

Equality is achieved for $w_j' = w_j^* \cdot \beta$, $\alpha_j' = \alpha_j^* / \beta$, where

$$\beta = \sqrt{\frac{|\alpha_j^*|}{\|w_j^*\|_2}}.$$

But f is invariant to such re-parameterizations!

Gated ReLU Networks: Sketch Continued

Let (w^*, α^*) be a solution to NC-GReLU.

By Young's inequality,

$$2 \sum_{j=1}^{p} \|w_j^*\|_2 \, |\alpha_j^*| \leq \sum_{j=1}^{p} \|w_j^*\|_2^2 + |\alpha_j^*|^2$$

Equality is achieved for $w_j' = w_j^* \cdot \beta$, $\alpha_j' = \alpha_j^* / \beta$, where

$$\beta = \sqrt{\frac{|\alpha_j^*|}{\|w_j^*\|_2}}.$$

But f is invariant to such re-parameterizations!

$$\begin{aligned}
\frac{1}{2}\| \sum_{j=1}^{p} \phi_{g_j}(X, w_j^*)\alpha_j^* - y\|_2^2 + \frac{\lambda}{2} \sum_{j=1}^{p} \|w_j^*\|_2^2 + |\alpha_j^*|^2 \\
\geq \frac{1}{2}\| \sum_{j=1}^{p} \phi_{g_j}(X, w_j')\alpha_j' - y\|_2^2 + \lambda \sum_{j=1}^{p} \|w_j'\|_2 |\alpha_j'|
\end{aligned}$$

Gated ReLU Networks: Sketch Continued

Now we use positive homogeneity of the norm,

$$\begin{aligned}\frac{1}{2}\| \sum_{j=1}^{p} \phi_{g_j}(X, w_j')\alpha_j' - y\|_2^2 + \lambda \sum_{j=1}^{p} \|w_j'\|_2 |\alpha_j'| \\ = \frac{1}{2}\| \sum_{j=1}^{p} \phi_{g_j}(X, w_j' \cdot \alpha_j') - y\|_2^2 + \lambda \sum_{j=1}^{p} \|w_j' \cdot |\alpha_j'|\|_2\end{aligned}$$

Gated ReLU Networks: Sketch Continued

Now we use positive homogeneity of the norm,

$$
\begin{aligned}
&\frac{1}{2}\|\sum_{j=1}^{p}\phi_{g_j}(X, w'_j)\alpha'_j - y\|_2^2 + \lambda\sum_{j=1}^{p}\|w'_j\|_2|\alpha'_j| \\
&\qquad = \frac{1}{2}\|\sum_{j=1}^{p}\phi_{g_j}(X, w'_j\cdot\alpha'_j) - y\|_2^2 + \lambda\sum_{j=1}^{p}\|w'_j\cdot|\alpha'_j|\|_2 \\
&\qquad = \frac{1}{2}\|\sum_{j=1}^{p}\phi_{g_j}(X, w''_j) - y\|_2^2 + \lambda\sum_{j=1}^{p}\|w''_j\|_2
\end{aligned}
$$

Gated ReLU Networks: Sketch Continued

Now we use positive homogeneity of the norm,

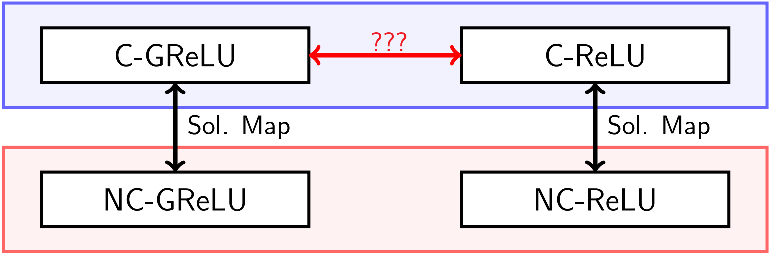
$$\begin{aligned}
&\frac{1}{2}\| \sum_{j=1}^{p} \phi_{g_j}(X, w_j')\alpha_j' - y\|_2^2 + \lambda \sum_{j=1}^{p} \|w_j'\|_2 |\alpha_j'| \\
&\quad = \frac{1}{2}\| \sum_{j=1}^{p} \phi_{g_j}(X, w_j' \cdot \alpha_j') - y\|_2^2 + \lambda \sum_{j=1}^{p} \|w_j' \cdot |\alpha_j'|\|_2 \\
&\quad = \frac{1}{2}\| \sum_{j=1}^{p} \phi_{g_j}(X, w_j'') - y\|_2^2 + \lambda \sum_{j=1}^{p} \|w_j''\|_2 \\
&\quad \geq \min_{w} \frac{1}{2}\| \sum_{j=1}^{p} \phi_{g_j}(X, w_j) - y\|_2^2 + \lambda \sum_{j=1}^{p} \|w_j\|_2 \quad (\textbf{C-GReLU}).
\end{aligned}$$

This completes the sketch.

What do we do with these models?

Gated ReLu Networks: Big Picture

What do we do with these models?



III. Cone Decompositions

Cone Decompositions: Gated ReLU Networks

Question: when are Gated ReLU and ReLU networks equivalent?

Cone Decompositions: Gated ReLU Networks

Question: when are Gated ReLU and ReLU networks equivalent?

Consider special case where $\lambda = 0$.

$$\textbf{C-GReLU} : \min_u \| \sum_{j=1}^{p} D_j X u_j - y \|_2^2.$$

V.S.

$$\textbf{C-ReLU} : \min_u \| \sum_{j=1}^{p} D_j X (v_j - w_j) - y \|_2^2.$$

$$\text{s.t. } v_j, w_j \in \mathcal{K}_j := \{ w : (2D_j - I) X w \geq 0 \},$$

Cone Decompositions: Equivalent Statement

Equiv. Question: when does $u_j = v_j - w_j$ for some $v_j, w_j \in \mathcal{K}_j$?

Cone Decompositions: Equivalent Statement

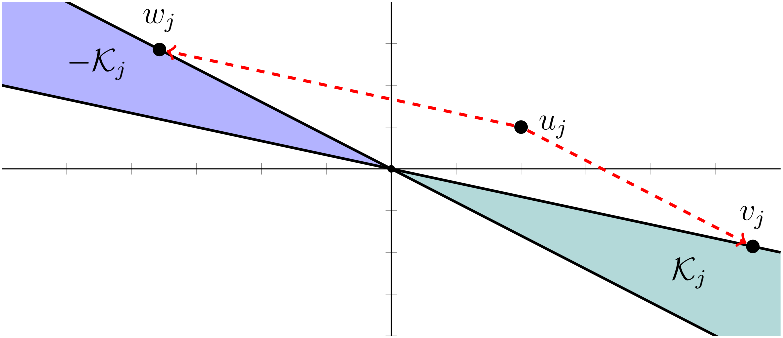
Equiv. Question: when does $u_j = v_j - w_j$ for some $v_j, w_j \in \mathcal{K}_j$?

Answer: when $\mathcal{K}_j - \mathcal{K}_j = \mathbb{R}^d$ and a "cone decomposition" exists.

Cone Decompositions: Equivalent Statement

Equiv. Question: when does $u_j = v_j - w_j$ for some $v_j, w_j \in \mathcal{K}_j$?

Answer: when $\mathcal{K}_j - \mathcal{K}_j = \mathbb{R}^d$ and a “cone decomposition” exists.



Cone Decomposition: Basic Result

Recall: $\mathcal{K}_j = \{w : (2D_j - I)Xw \geq 0\}$.

Cone Decomposition: Basic Result

Recall: $\mathcal{K}_j = \{w : (2D_j - I)Xw \geq 0\}$.

- This is a polyhedral cone which we rewrite as

$$\mathcal{K}_j = \bigcap_{i=1}^{n} \{w : [S_j]_{ii} \cdot \langle x, w \rangle \geq 0\},$$

where $S_j = (2D_j - I)$.

Cone Decomposition: Basic Result

Recall: $\mathcal{K}_j = \{w : (2D_j - I)Xw \geq 0\}$.

- This is a polyhedral cone which we rewrite as

$$\mathcal{K}_j = \bigcap_{i=1}^{n} \left\{ w : [S_j]_{ii} \cdot \langle x, w \rangle \geq 0 \right\},$$

where $S_j = (2D_j - I)$.

Proposition 3.1 (informal): If X is full row-rank, then $\mathrm{aff}(\mathcal{K}_j) = \mathbb{R}^d$ and $\mathcal{K}_j - \mathcal{K}_j = \mathbb{R}^d$.

Cone Decomposition: Basic Result

Recall: $\mathcal{K}_j = \{w : (2D_j - I)Xw \geq 0\}$.

- This is a polyhedral cone which we rewrite as

$$\mathcal{K}_j = \bigcap_{i=1}^{n} \left\{w : [S_j]_{ii} \cdot \langle x, w \rangle \geq 0\right\},$$

where $S_j = (2D_j - I)$.

Proposition 3.1 (informal): If X is full row-rank, then $\text{aff}(\mathcal{K}_j) = \mathbb{R}^d$ and $\mathcal{K}_j - \mathcal{K}_j = \mathbb{R}^d$.

Unfortunately, there is no extension to full-rank X.

Cone Decompositions: Not All Cones are Equal

Alternative Program: show we don't need "singular" cones $\mathcal{K}_j$,

$$\mathcal{K}_j - \mathcal{K}_j \subsetneq \mathbb{R}^d.$$

Cone Decompositions: Not All Cones are Equal

Alternative Program: show we don't need "singular" cones $\mathcal{K}_j$,

$$\mathcal{K}_j - \mathcal{K}_j \subsetneq \mathbb{R}^d.$$

Proposition 3.2 (informal): Suppose $\mathcal{K}_j - \mathcal{K}_j \subset \mathbb{R}^d$. Then, there exists $\mathcal{K}_i$ for which $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$ and $\mathcal{K}_j \subset \mathcal{K}_i$.

Cone Decompositions: Not All Cones are Equal

Alternative Program: show we don't need "singular" cones $\mathcal{K}_j$,

$$\mathcal{K}_j - \mathcal{K}_j \subsetneq \mathbb{R}^d.$$

Proposition 3.2 (informal): Suppose $\mathcal{K}_j - \mathcal{K}_j \subset \mathbb{R}^d$. Then, there exists $\mathcal{K}_i$ for which $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$ and $\mathcal{K}_j \subset \mathcal{K}_i$.

Interpretation: if optimal $u_j^* \neq 0$, then set

$$u_i' = u_j^* + u_i^*.$$

It is possible to show this causes no problems.

Cone Decompositions: Proof Sketch

Proof: Works by iteratively constructing $\mathcal{K}_i$ s.t. $\mathcal{K}_j \subset \mathcal{K}_i$.

Cone Decompositions: Proof Sketch

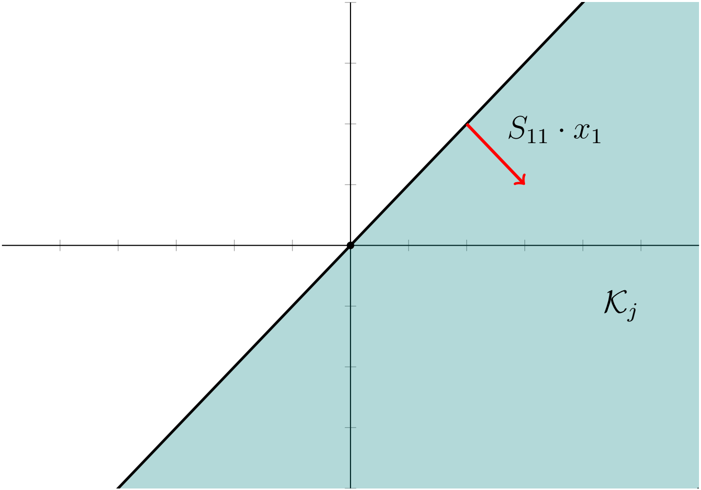
Proof: Works by iteratively constructing $\mathcal{K}_i$ s.t. $\mathcal{K}_j \subset \mathcal{K}_i$.

We sketch a simpler statement:

Proposition 3.2 (informal): Suppose $\mathcal{K}_j = \{0\}$. Then, there exists $\mathcal{K}_i$ for which $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$ and $\mathcal{K}_j \subset \mathcal{K}_i$.

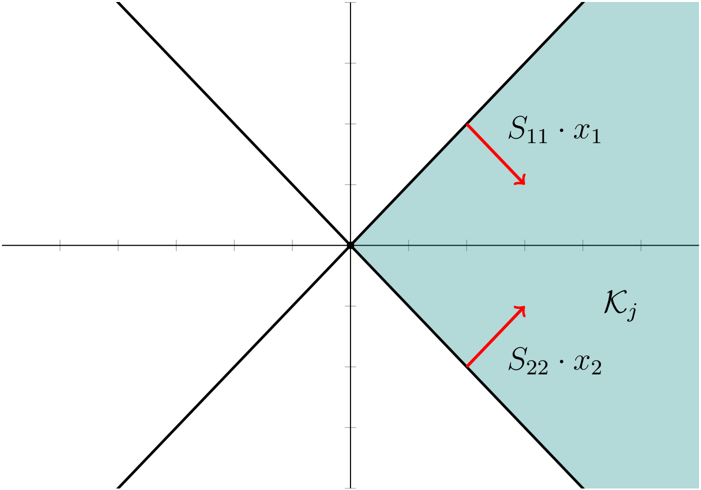
Cone Decompositions: Proof Sketch

$$\mathcal{K}'_j = \{w : [S_j]_{11} \cdot \langle x_1, w \rangle \geq 0\}$$



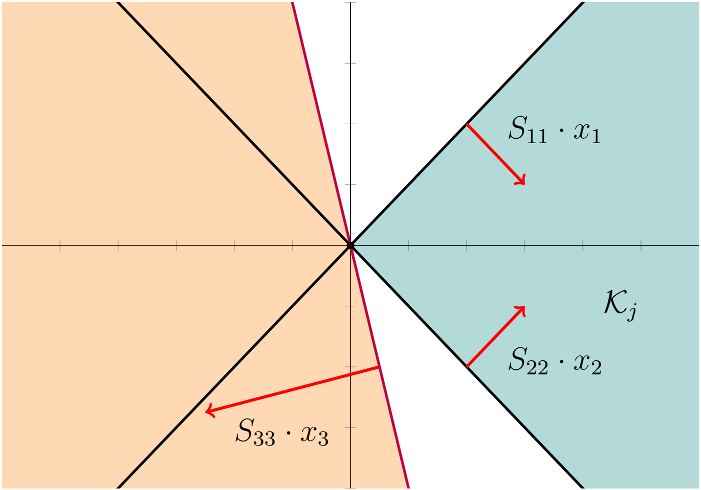
Cone Decompositions: Proof Sketch

$$\mathcal{K}_j'' = \mathcal{K}_j' \cap \{w : [S_j]_{22} \cdot \langle x_2, w \rangle \geq 0\}$$



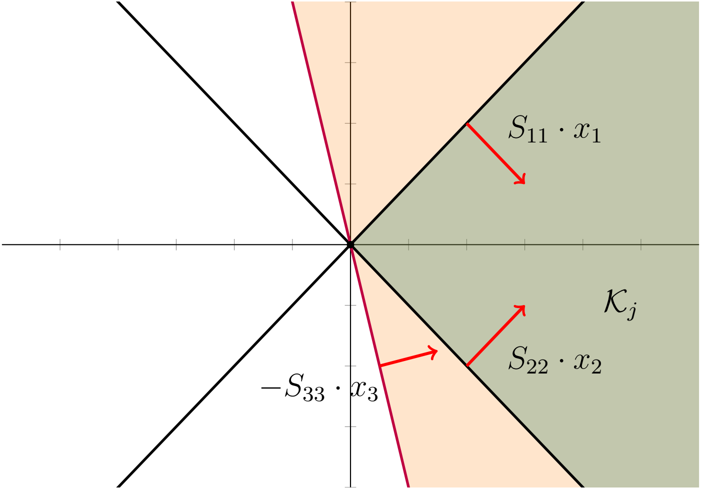
Cone Decompositions: Proof Sketch

$$\mathcal{K}_j''' = \mathcal{K}_j'' \cap \{w : [S_j]_{33} \cdot \langle x_3, w \rangle \geq 0\}$$



Cone Decompositions: Proof Sketch

$$\tilde{\mathcal{K}}_j''' = \mathcal{K}_j'' \cap \{w : -[S_j]_{33} \cdot \langle x_3, w \rangle \geq 0\}$$



Cone Decomposition: Main Result

- The real proof is more complex, but this is the core idea.
  - Build $\mathcal{K}_i$ by switching signs of $[S_j]_{ii}$.
  - Equivalent to turning on/off activations.
- Leads to our main approximation result.

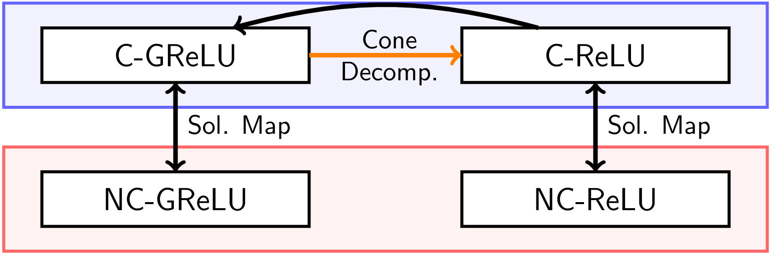
Cone Decomposition: Main Result

- The real proof is more complex, but this is the core idea.
  - Build $\mathcal{K}_i$ by switching signs of $[S_j]_{ii}$.
  - Equivalent to turning on/off activations.
- Leads to our main approximation result.

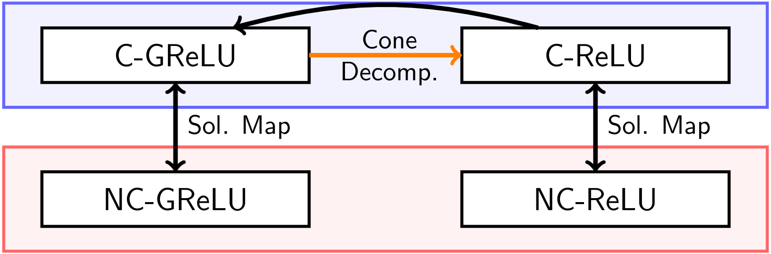
Theorem 3.7 (informal): Let $\lambda \geq 0$ and let p^* be the optimal value of the ReLU problem. There exists a C-GReLU problem with minimizer u^* and optimal value d^* satisfying,

$$d^* \leq p^* \leq d^* + 2\lambda\kappa(\tilde{X}_{\mathcal{J}}) \sum_{D_i \in \tilde{\mathcal{D}}} \|u_i^*\|_2.$$

Cone Decompositions: Big Picture



Cone Decompositions: Big Picture



Takeaways:

- Gated ReLU and ReLU model classes are the same.
- We can convert between them at will.

IV. Algorithms

Algorithms: ReLU by Cone Decomposition

Using cone decompositions **in practice**.

Algorithms: ReLU by Cone Decomposition

Using cone decompositions **in practice**.

---

1. Solve the gated ReLU problem:

$$u^* \in \arg\min_{u} \| \sum_{j=1}^{p} D_j X u_j - y \|_2^2 + \lambda \sum_{j=1}^{p} \|u_j\|_2$$

Algorithms: ReLU by Cone Decomposition

Using cone decompositions **in practice**.

1. Solve the gated ReLU problem:

$$u^* \in \arg\min_{u} \|\sum_{j=1}^{p} D_j X u_j - y\|_2^2 + \lambda \sum_{j=1}^{p} \|u_j\|_2$$

2. Solve a cone decomposition:

$$v_j^*, w_j^* \in \arg\min_{v_j, w_j} \left\{ L(v_j, w_j) : v_j - w_j = u_j^* \right\}$$

Algorithms: ReLU by Cone Decomposition

Using cone decompositions **in practice**.

1. Solve the gated ReLU problem:

$$u^* \in \arg\min_{u} \| \sum_{j=1}^{p} D_j X u_j - y\|_2^2 + \lambda \sum_{j=1}^{p} \|u_j\|_2$$

2. Solve a cone decomposition:

$$v_j^*, w_j^* \in \arg\min_{v_j, w_j} \left\{ L(v_j, w_j) : v_j - w_j = u_j^* \right\}$$

3. Compute corresponding ReLU model.

Algorithms: Solving the Convex Programs

We develop two algorithms for solving the convex reformulations:

- **R-FISTA**: a restarted FISTA variant for Gated ReLU.
- **AL**: an augmented Lagrangian method for the (constrained) ReLU Problem.

Algorithms: Solving the Convex Programs

We develop two algorithms for solving the convex reformulations:

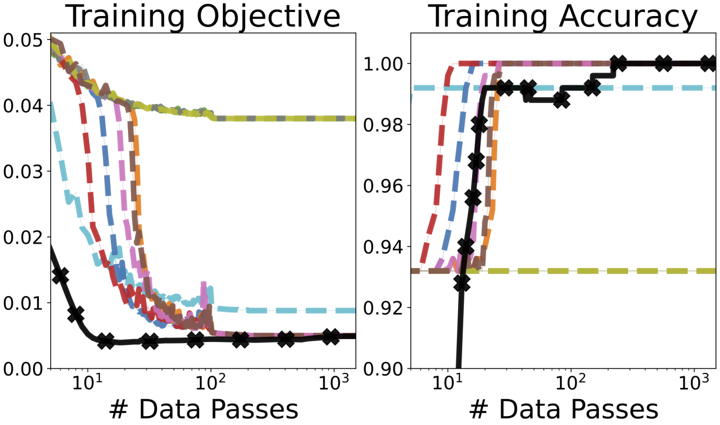
- **R-FISTA**: a restarted FISTA variant for Gated ReLU.
- **AL**: an augmented Lagrangian method for the (constrained) ReLU Problem.

And we can use all the convex tricks!

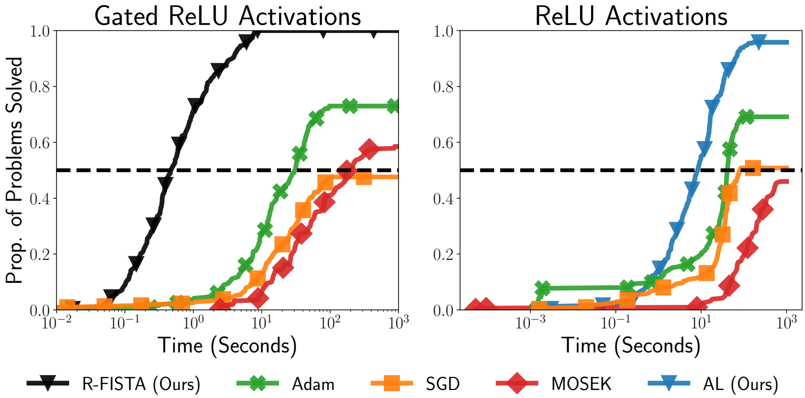
- **Fast**: $O(1/T^2)$ convergence rate.
- **Tuning-free**: line-search, restarts, data normalization, ...
- **Certificates**: termination based on min-norm subgradient.

Algorithms: Completing the Picture

Returning to our first example...



Algorithms: Large-Scale Robustness



- Generated by 438 training problems taken from UCI repo.
- R-FISTA/AL solve more, faster, than SGD and Adam.

Pause.

Our Contributions.

Our Contributions.

- We develop new convex reformulations of two-layer neural networks with **gated ReLU** activations.

Our Contributions.

- We develop new convex reformulations of two-layer neural networks with **gated ReLU** activations.
- We approximate the ReLU training problem by **unconstrained** convex optimization of a Gated ReLU network.

Our Contributions.

- We develop new convex reformulations of two-layer neural networks with **gated ReLU** activations.
- We approximate the ReLU training problem by **unconstrained** convex optimization of a Gated ReLU network.
- We propose and **exhaustively evaluate** algorithms for solving our convex reformulations.

Try our Code!

References I

Jonathan Fiat, Eran Malach, and Shai Shalev-Shwartz. "Decoupling gating from linearity". In: *arXiv preprint arXiv:1906.05032* (2019).

Peter Henderson et al. "Deep Reinforcement Learning That Matters". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 2018, pp. 3207–3214.

References II

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. 2012, pp. 1106–1114.

Mert Pilanci and Tolga Ergen. "Neural Networks are Convex Regularizers: Exact Polynomial-time Convex Optimization Formulations for Two-layer Networks". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. 2020, pp. 7695–7705.

References III

Aditya Ramesh et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents". In: *CoRR* abs/2204.06125 (2022). arXiv: 2204.06125.

Robert O Winder. "Partitions of N-space by hyperplanes". In: *SIAM Journal on Applied Mathematics* 14.4 (1966), pp. 811–818.

Lu Yuan et al. "Florence: A New Foundation Model for Computer Vision". In: *CoRR* abs/2111.11432 (2021).