Stanford University

Optimal Sets and Solution Paths of ReLU Networks

Aaron Mishkin Mert Pilanci

ICML 2023

The Problem

Problem: We don't understand the solution space of neural networks nearly as well as that of GLMs.

The Problem

Problem: We don't understand the solution space of neural networks nearly as well as that of GLMs.

Consider the Lasso:

The Problem

Problem: We don't understand the solution space of neural networks nearly as well as that of GLMs.

Consider the Lasso:

1. **Optimal Sets**: we have an exact polyhedral characterization and simple criteria for uniqueness (general position) [Tib13].

The Problem

Problem: We don't understand the solution space of neural networks nearly as well as that of GLMs.

Consider the Lasso:

1. **Optimal Sets**: we have an exact polyhedral characterization and simple criteria for uniqueness (general position) [Tib13].
2. **Regularization Paths**: we know the (min-norm) solution path is continuous and piece-wise linear [OPT00].

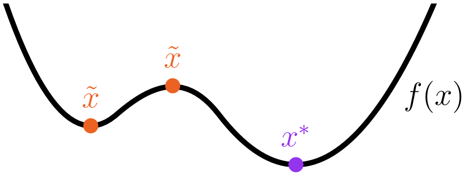
The Problem

Problem: We don't understand the solution space of neural networks nearly as well as that of GLMs.

Consider the Lasso:

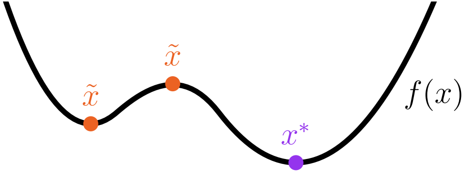
1. **Optimal Sets**: we have an exact polyhedral characterization and simple criteria for uniqueness (general position) [Tib13].
2. **Regularization Paths**: we know the (min-norm) solution path is continuous and piece-wise linear [OPT00].
3. **Algorithms**: we have efficient algorithms for homotopy [Efr+04] and computing minimal solutions [Tib13].

Non-convexity makes extensions beyond GLMs hard!



Challenges from Non-Convexity

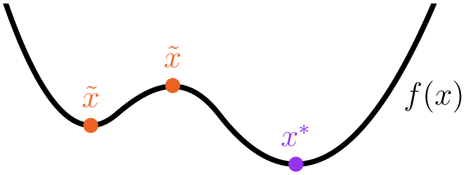
Non-convexity makes extensions beyond GLMs hard!



- **Optimality Conditions**: Stationarity $\not\Longrightarrow$ optimality. We have no global optimality criteria and no certificates.

Challenges from Non-Convexity

Non-convexity makes extensions beyond GLMs hard!

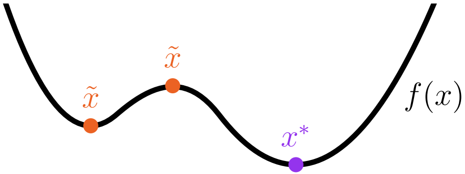


- **Optimality Conditions**: Stationarity $\not\Longrightarrow$ optimality. We have no global optimality criteria and no certificates.

- **Mathematical Tools**: We lose most of convex analysis and have to work with Clarke stationary points, etc.

Challenges from Non-Convexity

Non-convexity makes extensions beyond GLMs hard!



- **Optimality Conditions**: Stationarity $\not\Longrightarrow$ optimality. We have no global optimality criteria and no certificates.
- **Mathematical Tools**: We lose most of convex analysis and have to work with Clarke stationary points, etc.
- **Unintuitive Phenomena**: Surprising things happen even with toy neural networks!

Example: Discontinuous Paths

Consider training a toy neural network: given $(x_1, y_1), (x_2, y_2)$,

Example: Discontinuous Paths

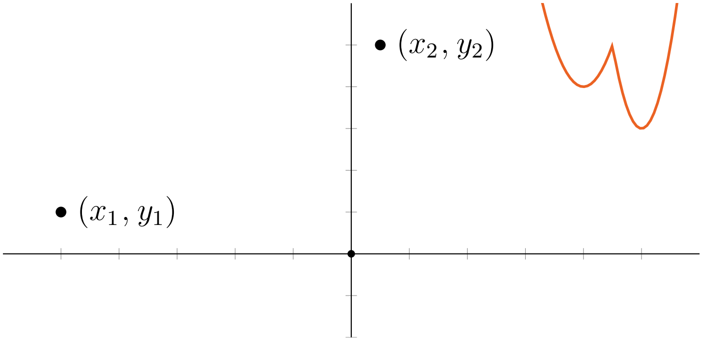
Consider training a toy neural network: given $(x_1, y_1), (x_2, y_2)$,

$$\min_{w_1} \frac{1}{2}((w_1 x_1)_+ - y_1)^2 + \frac{1}{2}((w_1 x_2)_+ - y_2)^2 + \lambda |w_1|.$$

Example: Discontinuous Paths

Consider training a toy neural network: given $(x_1, y_1), (x_2, y_2)$,

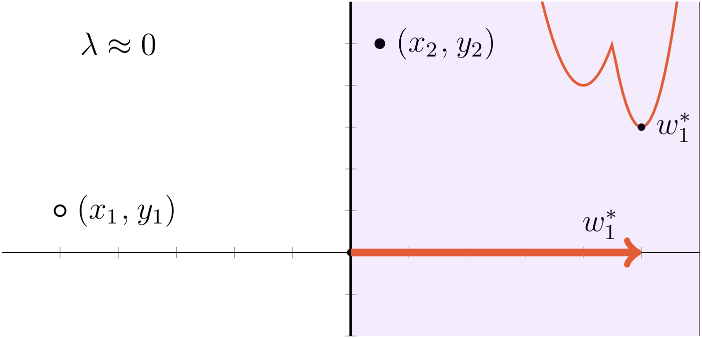
$$\min_{w_1} \frac{1}{2}((w_1 x_1)_+ - y_1)^2 + \frac{1}{2}((w_1 x_2)_+ - y_2)^2 + \lambda |w_1|.$$



Example: Discontinuous Paths

Consider training a toy ReLU network:

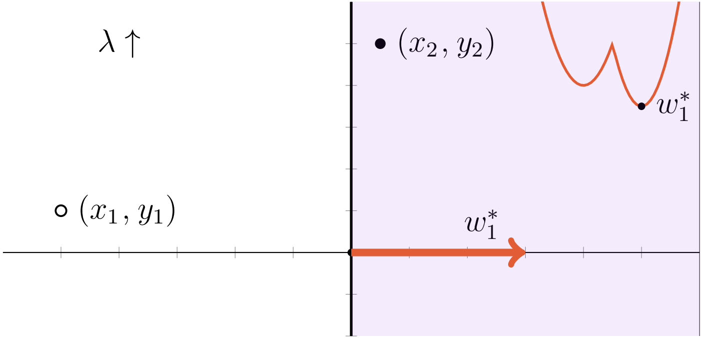
$$\min_{w_1} \frac{1}{2}((w_1 x_1)_+ - y_1)^2 + \frac{1}{2}((w_1 x_2)_+ - y_2)^2 + \lambda |w_1|.$$



Example: Discontinuous Paths

Consider training a toy neural network:

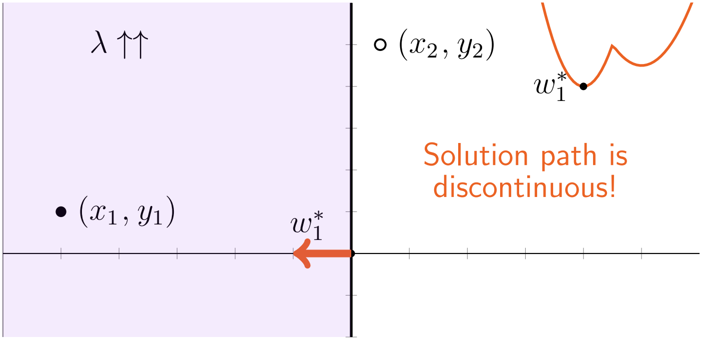
$$\min_{w_1} \frac{1}{2}((w_1 x_1)_+ - y_1)^2 + \frac{1}{2}((w_1 x_2)_+ - y_2)^2 + \lambda |w_1|.$$



Example: Discontinuous Paths

Consider training a toy neural network:

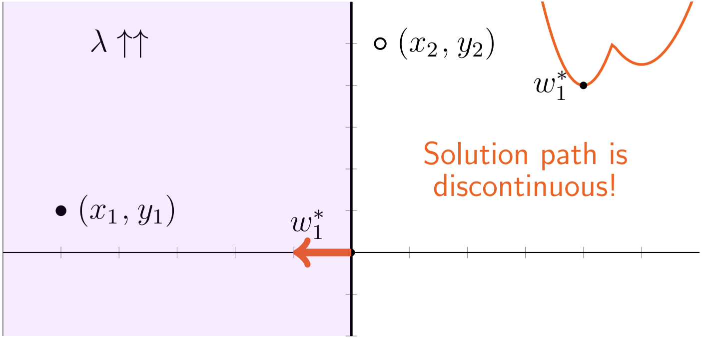
$$\min_{w_1} \frac{1}{2}((w_1 x_1)_+ - y_1)^2 + \frac{1}{2}((w_1 x_2)_+ - y_2)^2 + \lambda |w_1|.$$



Example: Discontinuous Paths

Consider training a toy neural network:

$$\min_{w_1} \frac{1}{2}((w_1 x_1)_+ - y_1)^2 + \frac{1}{2}((w_1 x_2)_+ - y_2)^2 + \lambda |w_1|.$$



Goal: Overcome these problems via convexification.

Our Contributions

Overall Approach: leverage convex reformulations of ReLU networks [PE20] as an analytical tool.

Our Contributions

Overall Approach: leverage convex reformulations of ReLU networks [PE20] as an analytical tool.

---

1. **Optimal Sets**: we characterize all optima of the non-convex training objective.

Our Contributions

Overall Approach: leverage convex reformulations of ReLU networks [PE20] as an analytical tool.

---

1. **Optimal Sets**: we characterize all optima of the non-convex training objective.
2. **Uniqueness**: we develop simple criteria for ReLU networks to admit unique solutions up permutation/split symmetries.

Our Contributions

Overall Approach: leverage convex reformulations of ReLU networks [PE20] as an analytical tool.

---

1. **Optimal Sets**: we characterize all optima of the non-convex training objective.
2. **Uniqueness**: we develop simple criteria for ReLU networks to admit unique solutions up permutation/split symmetries.
3. **Optimal Pruning**: we leverage our theory to give a poly-time procedure for computing minimal ReLU networks.

I. Background on Convex Reformulations

Convex Reformulations: Flavor of Results

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

Convex Reformulations: Flavor of Results

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

Equivalent means:

Convex Reformulations: Flavor of Results

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

Equivalent means:

- The global minima have the same values: $p^* = q^*$

Convex Reformulations: Flavor of Results

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

Equivalent means:

- The global minima have the same values: $p^* = q^*$
- We can map every global minimum u^* for one problem into a global minimum v^* of the other.

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

Equivalent means:

- The global minima have the same values: $p^* = q^*$
- We can map every global minimum u^* for one problem into a global minimum v^* of the other.
  - We call this the solution mapping.

Convex Reformulations: Two-Layer ReLU Networks

Non-Convex Problem (NC-ReLU)

$$\min_{W_1, w_2} \underbrace{\frac{1}{2}\| \sum_{j=1}^{m} (XW_{1j})_+ w_{2j} - y\|_2^2}_{\text{Squared Error}} + \underbrace{\lambda \sum_{j=1}^{m} \|W_{1j}\|_2^2 + |w_{2j}|^2}_{\text{Weight Decay}},$$

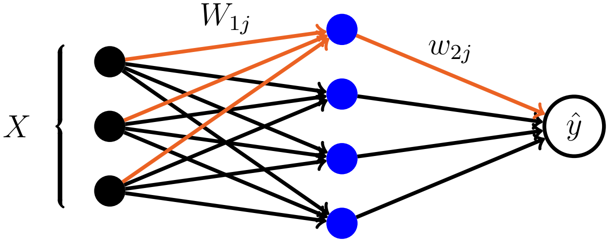
where $(x)_+ = \max\{x, 0\}$ is the ReLU activation.

Convex Reformulations: Two-Layer ReLU Networks

Non-Convex Problem (NC-ReLU)

$$\min_{W_1, w_2} \underbrace{\frac{1}{2}\|\sum_{j=1}^{m}(XW_{1j})_+ w_{2j} - y\|_2^2}_{\text{Squared Error}} + \underbrace{\lambda \sum_{j=1}^{m} \|W_{1j}\|_2^2 + |w_{2j}|^2}_{\text{Weight Decay}},$$

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

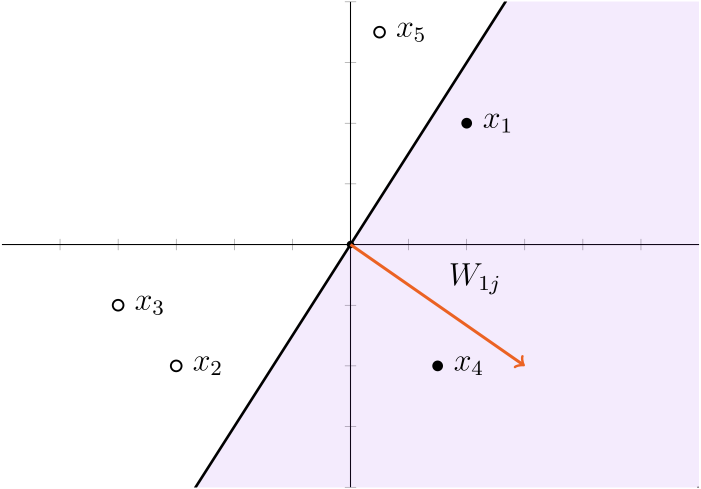


Aside: ReLU Activation Patterns

Each ReLU neuron is active on a half-space:

Aside: ReLU Activation Patterns

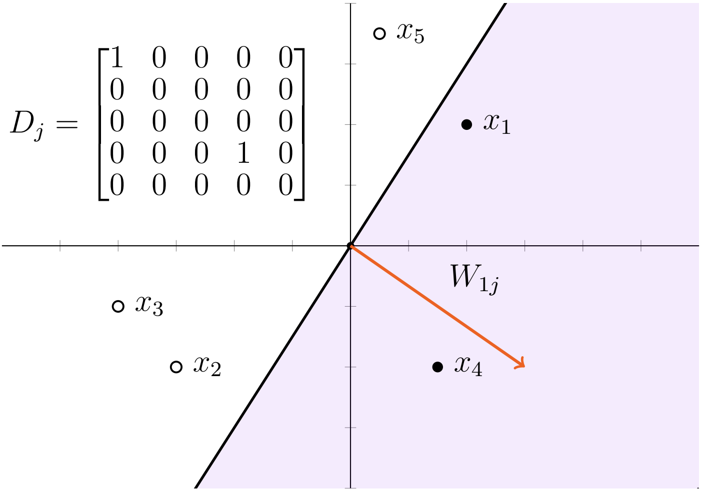
Each ReLU neuron is active on a half-space:



Aside: ReLU Activation Patterns

Each ReLU neuron is active on a half-space:

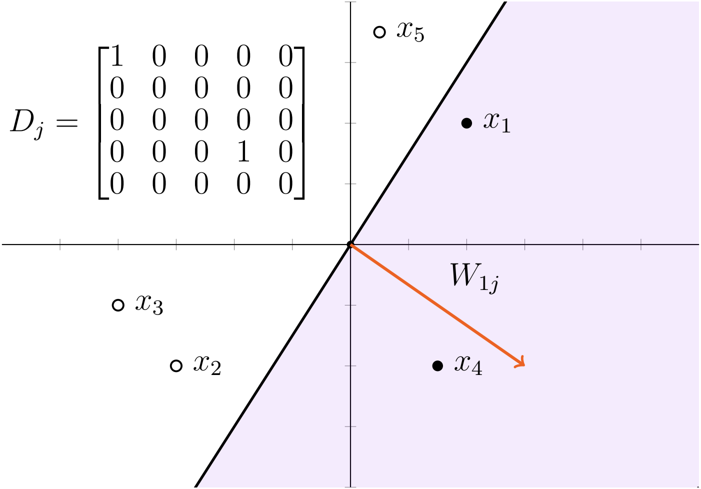
$$D_j = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Aside: ReLU Activation Patterns

Each ReLU neuron is active on a half-space:

$$D_j = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

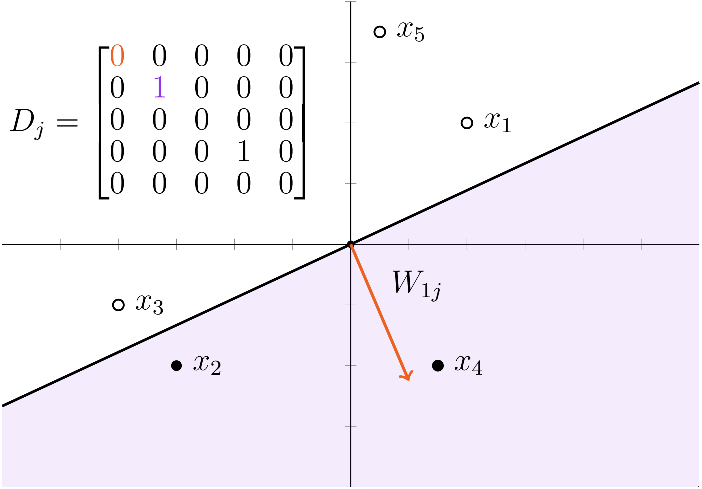


Activation Pattern satisfies $D_j X W_{1j} = (X W_{1j})_+$

Aside: ReLU Activation Patterns

Each ReLU neuron is active on a half-space:

$$D_j = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Activation Pattern satisfies $D_j X W_{1j} = (X W_{1j})_+$

Convex Reformulations: Convex Problem

Convex Reformulation (C-ReLU) [PE20]

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

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

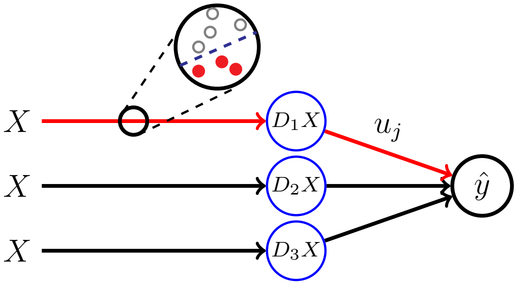
Convex Reformulations: Convex Problem

Convex Reformulation (C-ReLU) [PE20]

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

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

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



Convex Reformulations: Hardness

Result: if $m \geq m^*$ for some $m^* \leq n$, then C-ReLU and NC-ReLU are equivalent [PE20].

Convex Reformulations: Hardness

Result: if $m \geq m^*$ for some $m^* \leq n$, then C-ReLU and NC-ReLU are equivalent [PE20].

How "hard" is the convex program?

Convex Reformulations: Hardness

Result: if $m \geq m^*$ for some $m^* \leq n$, then C-ReLU and NC-ReLU are equivalent [PE20].

How "hard" is the convex program?

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

Convex Reformulations: Hardness

Result: if $m \geq m^*$ for some $m^* \leq n$, then C-ReLU and NC-ReLU are equivalent [PE20].

How "hard" is the convex program?

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

The **convex program** is:

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

Convex Reformulations: Hardness

Result: if $m \geq m^*$ for some $m^* \leq n$, then C-ReLU and NC-ReLU are equivalent [PE20].

How "hard" is the convex program?

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

The **convex program** is:

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

Convex Reformulations: Hardness

Result: if $m \geq m^*$ for some $m^* \leq n$, then C-ReLU and NC-ReLU are equivalent [PE20].

How "hard" is the convex program?

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

The **convex program** is:

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

We exchange one kind of hardness for another.

II. Optimal Sets of ReLU Networks

Optimal Set: Roadmap

Proof Roadmap:

Optimal Set: Roadmap

Proof Roadmap:

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.

Optimal Set: Roadmap

Proof Roadmap:

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.
2. Extend results to non-convex ReLU networks using the solution mapping.

Optimal Set: Roadmap

Proof Roadmap:

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.
2. Extend results to non-convex ReLU networks using the solution mapping.
3. Leverage explicit characterization of the optimal set for new insights and algorithms.

Optimal Set: Roadmap

Proof Roadmap:

1. **Characterize solutions to the convex reformulation using strong duality and KKT conditions.**
2. Extend results to non-convex ReLU networks using the solution mapping.
3. Leverage explicit characterization of the optimal set for new insights and algorithms.

C-ReLU: Strong Duality

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.

C-ReLU: Strong Duality

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.

C-ReLU Solution Set:

$$\mathcal{W}^*(\lambda) = \mathop{\arg\min}_{v_i, w_i \in \mathcal{K}_i} \left\{ \frac{1}{2} \left\| \sum_{D_i \in \tilde{\mathcal{D}}} D_i X (v_i - w_i), y \right\|_2^2 + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i\|_2 + \|w_i\|_2 \right\}.$$

C-ReLU: Strong Duality

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.

C-ReLU Solution Set:

$$\mathcal{W}^*(\lambda) = \mathop{\arg\min}_{v_i, w_i \in \mathcal{K}_i} \left\{ \frac{1}{2} \left\| \sum_{D_i \in \tilde{\mathcal{D}}} D_i X(v_i - w_i), y \right\|_2^2 + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i\|_2 + \|w_i\|_2 \right\}.$$

1. Convex objective + linear constraints $\implies$ strong duality!

C-ReLU: Strong Duality

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.

C-ReLU Solution Set:

$$\mathcal{W}^*(\lambda) = \mathop{\arg\min}_{v_i, w_i \in \mathcal{K}_i} \Bigg\{ \frac{1}{2} \Bigg\| \sum_{D_i \in \tilde{\mathcal{D}}} D_i X(v_i - w_i), y \Bigg\|_2^2 + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i\|_2 + \|w_i\|_2 \Bigg\}.$$

1. Convex objective + linear constraints $\implies$ strong duality!
2. Introduce dual variables ρ and analyze the KKT conditions.

C-ReLU: Strong Duality

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.

C-ReLU Solution Set:

$$\mathcal{W}^*(\lambda) = \operatorname*{arg\,min}_{v_i, w_i \in \mathcal{K}_i} \left\{ \frac{1}{2} \left\| \sum_{D_i \in \tilde{\mathcal{D}}} D_i X (v_i - w_i), y \right\|_2^2 + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i\|_2 + \|w_i\|_2 \right\}.$$

1. Convex objective + linear constraints $\implies$ strong duality!
2. Introduce dual variables ρ and analyze the KKT conditions.
3. Define $\theta = \begin{bmatrix} v_i \\ -w_i \end{bmatrix}$ and index D_i's from 1 to $2p$.

C-ReLU: Optimality Conditions

We form the Lagrangian for the convex reformulation:

$$\mathcal{L}(\theta, \rho) = \frac{1}{2} \| \sum_{i=1}^{2p} D_i X \theta_i - y \|_2^2 + \lambda \sum_{i=1}^{2p} \|\theta_i\|_2 - \sum_{i=1}^{2p} \left\langle K_i^\top \rho_i, \theta_i \right\rangle,$$

where $K_i = (2D_i - I)X$.

C-ReLU: Optimality Conditions

We form the Lagrangian for the convex reformulation:

$$\mathcal{L}(\theta, \rho) = \frac{1}{2} \| \sum_{i=1}^{2p} D_i X \theta_i - y \|_2^2 + \lambda \sum_{i=1}^{2p} \|\theta_i\|_2 - \sum_{i=1}^{2p} \left\langle K_i^\top \rho_i, \theta_i \right\rangle,$$

where $K_i = (2D_i - I)X$.

The KKT conditions are necessary and sufficient for optimality:

C-ReLU: Optimality Conditions

We form the Lagrangian for the convex reformulation:

$$\mathcal{L}(\theta, \rho) = \frac{1}{2}\| \sum_{i=1}^{2p} D_i X \theta_i - y\|_2^2 + \lambda \sum_{i=1}^{2p} \|\theta_i\|_2 - \sum_{i=1}^{2p} \left\langle K_i^\top \rho_i, \theta_i \right\rangle,$$

where $K_i = (2D_i - I)X$.

The KKT conditions are necessary and sufficient for optimality:

- Define the **optimal model fit**: $\hat{y} = \sum_{i=1}^{2p} D_i X \theta_i^*$.

C-ReLU: Optimality Conditions

We form the Lagrangian for the convex reformulation:

$$\mathcal{L}(\theta, \rho) = \frac{1}{2}\| \sum_{i=1}^{2p} D_i X \theta_i - y\|_2^2 + \lambda \sum_{i=1}^{2p} \|\theta_i\|_2 - \sum_{i=1}^{2p} \left\langle K_i^\top \rho_i, \theta_i \right\rangle,$$

where $K_i = (2D_i - I)X$.

The KKT conditions are necessary and sufficient for optimality:

- Define the **optimal model fit**: $\hat{y} = \sum_{i=1}^{2p} D_i X \theta_i^*$.
- The Lagrangian is stationary when,

$$\underbrace{X^\top D_i(\hat{y} - y) + K_i^\top \rho_i}_{q_i} \in \partial \lambda \|\theta_i\|_2.$$

C-ReLU: Optimality Conditions

We form the Lagrangian for the convex reformulation:

$$\mathcal{L}(\theta, \rho) = \frac{1}{2} \| \sum_{i=1}^{2p} D_i X \theta_i - y \|_2^2 + \lambda \sum_{i=1}^{2p} \|\theta_i\|_2 - \sum_{i=1}^{2p} \left\langle K_i^\top \rho_i, \theta_i \right\rangle,$$

where $K_i = (2D_i - I)X$.

The KKT conditions are necessary and sufficient for optimality:

- Define the **optimal model fit**: $\hat{y} = \sum_{i=1}^{2p} D_i X \theta_i^*$.
- The Lagrangian is stationary when,

$$\underbrace{X^\top D_i(\hat{y} - y) + K_i^\top \rho_i}_{q_i} \in \partial \lambda \|\theta_i\|_2.$$

- It turns out each "block correlation" q_i is unique WLOG!

C-ReLU: Implications of Stationarity

Stationary Lagrangian:

$$X^\top D_i(\hat{y} - y) + K_i^\top \rho_i =: q_i \in \partial \lambda \|\theta_i^*\|_2.$$

C-ReLU: Implications of Stationarity

Stationary Lagrangian:

$$X^\top D_i(\hat{y} - y) + K_i^\top \rho_i =: q_i \in \partial \lambda \|\theta_i^*\|_2.$$

Non-zero Blocks:

- Suppose $\theta_i^* \neq 0$.

C-ReLU: Implications of Stationarity

Stationary Lagrangian:

$$X^\top D_i(\hat{y} - y) + K_i^\top \rho_i =: q_i \in \partial \lambda \|\theta_i^*\|_2.$$

Non-zero Blocks:

- Suppose $\theta_i^* \neq 0$.
- Then $\nabla_\theta \lambda \|\theta_i^*\|_2 = \lambda \frac{\theta_i^*}{\|\theta_i^*\|_2}$ and there exists $\alpha_i > 0$ for which,

$$q_i = \lambda \frac{\theta_i^*}{\|\theta_i^*\|_2} \implies \theta_i^* = \alpha_i q_i.$$

C-ReLU: Implications of Stationarity

Stationary Lagrangian:

$$X^\top D_i(\hat{y} - y) + K_i^\top \rho_i =: q_i \in \partial \lambda \|\theta_i^*\|_2.$$

Non-zero Blocks:

- Suppose $\theta_i^* \neq 0$.
- Then $\nabla_\theta \lambda \|\theta_i^*\|_2 = \lambda \frac{\theta_i^*}{\|\theta_i^*\|_2}$ and there exists $\alpha_i > 0$ for which,

$$q_i = \lambda \frac{\theta_i^*}{\|\theta_i^*\|_2} \implies \theta_i^* = \alpha_i q_i.$$

- Every optimal $\theta_i^* \neq 0$ is collinear with the (unique) q_i vector.

C-ReLU: the Optimal Set

- **Optimal Fit:** $\hat{y} = \sum_{i=1}^{2p} D_i X \theta_i^*$.
- **Block Correlation:** $q_i := X^\top D_i(\hat{y} - y) + K_i^\top \rho_i$.

C-ReLU: the Optimal Set

- **Optimal Fit:** $\hat{y} = \sum_{i=1}^{2p} D_i X \theta_i^*$.
- **Block Correlation:** $q_i := X^\top D_i(\hat{y} - y) + K_i^\top \rho_i$.
- **Support Set:** $\mathcal{S}_\lambda = \{i \in [2p] : \exists \theta \in \mathcal{W}^*(\lambda),\ \theta_i \neq 0\}$.

C-ReLU: the Optimal Set

- **Optimal Fit:** $\hat{y} = \sum_{i=1}^{2p} D_i X \theta_i^*$.
- **Block Correlation:** $q_i := X^\top D_i(\hat{y} - y) + K_i^\top \rho_i$.
- **Support Set:** $\mathcal{S}_\lambda = \{i \in [2p] : \exists \theta \in \mathcal{W}^*(\lambda),\ \theta_i \neq 0\}$.

Proposition (Informal)

Fix $\lambda > 0$. The optimal set of the C-ReLU problem is given by

$$\begin{aligned}\mathcal{W}^*(\lambda) = \Big\{\theta : \sum_{i=1}^{2p} D_i X \theta_i = \hat{y} \\ \forall\, i \in \mathcal{S}_\lambda, \theta_i = \alpha_i q_i, \alpha_i \geq 0, \\ \forall\, j \in [2p] \setminus \mathcal{S}_\lambda, \theta_j = 0, \Big\}\end{aligned}$$

Returning to our Roadmap

Proof Roadmap:

Returning to our Roadmap

Proof Roadmap:

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.
2. **Extend results to non-convex ReLU networks using the solution mapping.**
3. Leverage explicit characterization of the optimal set for new insights and algorithms.

NC-ReLU: Using the Solution Mapping

2. Extend results to non-convex ReLU networks using the solution mapping.

NC-ReLU: Using the Solution Mapping

2. Extend results to non-convex ReLU networks using the solution mapping.

We need to do some accounting for model symmetries:

NC-ReLU: Using the Solution Mapping

2. Extend results to non-convex ReLU networks using the solution mapping.

We need to do some accounting for model symmetries:

- **Permutations**: Re-ordering neurons inside the layers.

NC-ReLU: Using the Solution Mapping

2. Extend results to non-convex ReLU networks using the solution mapping.

We need to do some accounting for model symmetries:

- **Permutations**: Re-ordering neurons inside the layers.
- **Splits**: Splitting a neuron into two collinear neurons.

NC-ReLU: Using the Solution Mapping

2. Extend results to non-convex ReLU networks using the solution mapping.

We need to do some accounting for model symmetries:

- **Permutations**: Re-ordering neurons inside the layers.
- **Splits**: Splitting a neuron into two collinear neurons.

Theorem (Informal)

Suppose $m \geq m^*$. Then the optimal set for NC-ReLU up to permutation/split symmetries is

$$\begin{aligned}\mathcal{O}_\lambda = \big\{(W_1, w_2) :\ & f_{W_1, w_2}(X) = \hat{y}, \\ & \forall\, i \in \mathcal{S}_\lambda, W_{1i} = \left({}^{\alpha_i}/\lambda\right)^{1/2} q_i, w_{2i} = \left(\alpha_i \lambda\right)^{1/2}, \alpha_i \geq 0 \\ & \forall\, i \in [2p] \setminus \mathcal{S}_\lambda, W_{1i} = 0,\ w_{2i} = 0\big\}.\end{aligned}$$

NC-ReLU: Surprises from the Optimal Set

Theorem (Informal)

Suppose $m \geq m^*$. Then the optimal set for NC-ReLU up to permutation/split symmetries is

$$\begin{aligned}\mathcal{O}_\lambda = \big\{(W_1, w_2) :\ & f_{W_1, w_2}(X) = \hat{y}, \\ & \forall\, i \in \mathcal{S}_\lambda, W_{1i} = \left({}^{\alpha_i}/\lambda\right)^{1/2} q_i, w_{2i} = (\alpha_i \lambda)^{1/2}, \alpha_i \geq 0 \\ & \forall\, i \in [2p] \setminus \mathcal{S}_\lambda, W_{1i} = 0,\ w_{2i} = 0\big\}.\end{aligned}$$

NC-ReLU: Surprises from the Optimal Set

Theorem (Informal)

Suppose $m \geq m^*$. Then the optimal set for NC-ReLU up to permutation/split symmetries is

$$\begin{aligned}\mathcal{O}_\lambda = \big\{&(W_1, w_2) : \ f_{W_1, w_2}(X) = \hat{y}, \\ &\forall\, i \in \mathcal{S}_\lambda, W_{1i} = \left({}^{\alpha_i}/\lambda\right)^{1/2} q_i, w_{2i} = (\alpha_i \lambda)^{1/2}, \alpha_i \geq 0 \\ &\forall\, i \in [2p] \setminus \mathcal{S}_\lambda, W_{1i} = 0, \ w_{2i} = 0\big\}.\end{aligned}$$

Surprising Properties of the Optimal Set:

NC-ReLU: Surprises from the Optimal Set

Theorem (Informal)

Suppose $m \geq m^*$. *Then the optimal set for NC-ReLU up to permutation/split symmetries is*

$$\mathcal{O}_\lambda = \big\{(W_1, w_2) :\ f_{W_1,w_2}(X) = \hat{y},$$
$$\forall\, i \in \mathcal{S}_\lambda, W_{1i} = (\alpha_i/\lambda)^{1/2} q_i, w_{2i} = (\alpha_i \lambda)^{1/2}, \alpha_i \geq 0$$
$$\forall\, i \in [2p] \setminus \mathcal{S}_\lambda, W_{1i} = 0,\ w_{2i} = 0\big\}.$$

Surprising Properties of the Optimal Set:

- Given the ordering induced by D_i, every optimal neuron W^*_{1i} is positively collinear!

NC-ReLU: Surprises from the Optimal Set

Theorem (Informal)

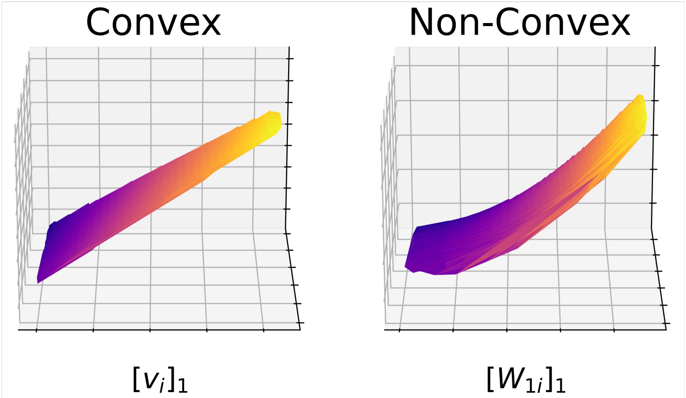
Suppose $m \geq m^*$. Then the optimal set for NC-ReLU up to permutation/split symmetries is

$$\begin{aligned}\mathcal{O}_\lambda = \big\{&(W_1, w_2) : f_{W_1, w_2}(X) = \hat{y}, \\ &\forall\, i \in \mathcal{S}_\lambda, W_{1i} = ({}^{\alpha_i}/{\lambda})^{1/2} q_i, w_{2i} = (\alpha_i \lambda)^{1/2}, \alpha_i \geq 0 \\ &\forall\, i \in [2p] \setminus \mathcal{S}_\lambda, W_{1i} = 0,\, w_{2i} = 0\big\}.\end{aligned}$$

Surprising Properties of the Optimal Set:

- Given the ordering induced by D_i, every optimal neuron W_{1i}^* is positively collinear!
- Up to permutation/split symmetries the optimal set is connected!

NC-ReLU: Appearance of Solution Sets



- The non-convex parameterization maps the convex polytope of solutions into a curved manifold.

NC-ReLU: Exploring the Optimal Set

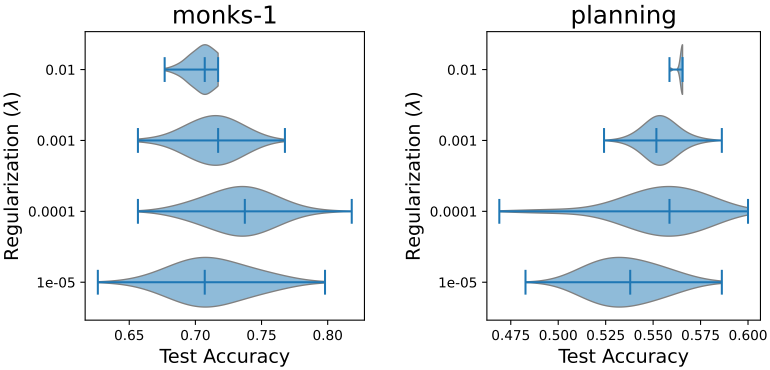
- Take 10,000 samples from the set of optimal neural networks.

NC-ReLU: Exploring the Optimal Set

- Take 10,000 samples from the set of optimal neural networks.
- All samples have (i) **same training accuracy**, (ii) **same model norm**, but can generalize differently.

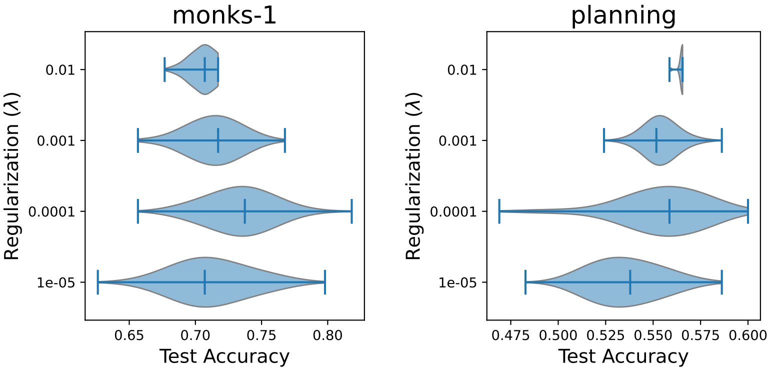
NC-ReLU: Exploring the Optimal Set

- Take 10,000 samples from the set of optimal neural networks.
- All samples have (i) **same training accuracy**, (ii) **same model norm**, but can generalize differently.



NC-ReLU: Exploring the Optimal Set

- Take 10,000 samples from the set of optimal neural networks.
- All samples have (i) **same training accuracy**, (ii) **same model norm**, but can generalize differently.



Implicit regularization is critical to generalization.

III. Optimal Pruning

Optimal Pruning: the Final Step

Proof Roadmap:

Optimal Pruning: the Final Step

Proof Roadmap:

1. Characterize solutions to the convex reformulation using strong duality and KKT conditions.
2. Extend results to non-convex ReLU networks using the solution mapping.
3. **Leverage explicit characterization of the optimal set for new insights and algorithms.**

Optimal Pruning: the Polytope of Solutions

3. Leverage explicit characterization of the optimal set for new insights and algorithms.

Optimal Pruning: the Polytope of Solutions

3. Leverage explicit characterization of the optimal set for new insights and algorithms.

$$\mathcal{W}^*(\lambda) = \big\{\theta : \sum_{i=1}^{2p} D_i X \theta_i = \hat{y},$$
$$\forall\, i \in \mathcal{S}_\lambda, \theta_i = \alpha_i q_i, \alpha_i \geq 0,$$
$$\forall\, j \in [2p] \setminus \mathcal{S}_\lambda, \theta_j = 0\big\}$$

Optimal Pruning: the Polytope of Solutions

3. Leverage explicit characterization of the optimal set for new insights and algorithms.

$$\mathcal{W}^*(\lambda) = \Big\{\theta : \sum_{i=1}^{2p} D_i X \theta_i = \hat{y}, \\ \forall\, i \in \mathcal{S}_\lambda, \theta_i = \alpha_i q_i, \alpha_i \geq 0, \\ \forall\, j \in [2p] \setminus \mathcal{S}_\lambda, \theta_j = 0\Big\}$$

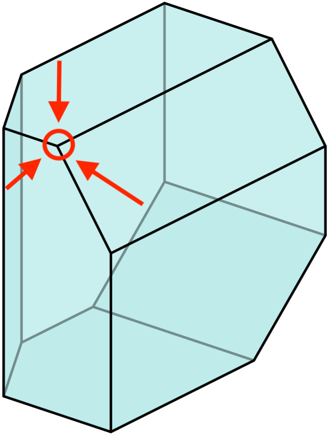
The C-ReLU optimal set is a convex polytope!

Optimal Pruning: the Polytope of Solutions

3. Leverage explicit characterization of the optimal set for new insights and algorithms.

$$\mathcal{W}^*(\lambda) = \Big\{\theta : \sum_{i=1}^{2p} D_i X \theta_i = \hat{y},$$
$$\forall\, i \in \mathcal{S}_\lambda, \theta_i = \alpha_i q_i, \alpha_i \geq 0,$$
$$\forall\, j \in [2p] \setminus \mathcal{S}_\lambda, \theta_j = 0\Big\}$$

The C-ReLU optimal set is a convex polytope!



Figure

Optimal Pruning: Vertices

1. Stack the q_i vectors into a matrix $Q = \begin{bmatrix} | & & | \\ q_1 & \cdots & q_{2p} \\ | & & | \end{bmatrix}$.

Optimal Pruning: Vertices

1. Stack the q_i vectors into a matrix $Q = \begin{bmatrix} | & & | \\ q_1 & \cdots & q_{2p} \\ | & & | \end{bmatrix}$.
2. The C-ReLU Optimal Set in α space is then,

$$\begin{aligned} \mathcal{W}^*(\lambda) &= Q_{\mathcal{S}_\lambda}\Big\{\alpha \succeq 0 : \sum_{i \in \mathcal{S}_\lambda} (D_i X q_i)\alpha_i = \hat{y}, \Big\} \\ &= Q_{\mathcal{S}_\lambda}\mathcal{P}_{\mathcal{S}_\lambda}. \end{aligned} \tag{1}$$

Optimal Pruning: Vertices

1. Stack the q_i vectors into a matrix $Q = \begin{bmatrix} | & & | \\ q_1 & \cdots & q_{2p} \\ | & & | \end{bmatrix}$.
2. The C-ReLU Optimal Set in α space is then,

$$\begin{aligned} \mathcal{W}^*(\lambda) &= Q_{\mathcal{S}_\lambda}\{\alpha \succeq 0 : \sum_{i \in \mathcal{S}_\lambda} (D_i X q_i)\alpha_i = \hat{y}, \} \\ &= Q_{\mathcal{S}_\lambda} \mathcal{P}_{\mathcal{S}_\lambda}. \end{aligned} \tag{1}$$

3. $\bar{\alpha} \in \mathcal{P}_{\mathcal{S}_\lambda}$ is a vertex iff $\{D_i X q_i\}_{\bar{\alpha}_i \neq 0}$ are linearly independent.

Optimal Pruning: Vertices

1. Stack the q_i vectors into a matrix $Q = \begin{bmatrix} | & & | \\ q_1 & \cdots & q_{2p} \\ | & & | \end{bmatrix}$.
2. The C-ReLU Optimal Set in α space is then,

$$\begin{aligned} \mathcal{W}^*(\lambda) &= Q_{\mathcal{S}_\lambda}\left\{\alpha \succeq 0 : \sum_{i \in \mathcal{S}_\lambda} (D_i X q_i)\alpha_i = \hat{y}, \right\} \\ &= Q_{\mathcal{S}_\lambda} \mathcal{P}_{\mathcal{S}_\lambda}. \end{aligned} \tag{1}$$

3. $\bar{\alpha} \in \mathcal{P}_{\mathcal{S}_\lambda}$ is a vertex iff $\{D_i X q_i\}_{\bar{\alpha}_i \neq 0}$ are linearly independent.

Are these vertices special in some way?

Optimal Pruning: Minimal Models

Definition: An optimal C-ReLU model θ^* is minimal if there does not exist another optimal model θ' with strictly smaller support.

Optimal Pruning: Minimal Models

Definition: An optimal C-ReLU model θ^* is minimal if there does not exist another optimal model θ' with strictly smaller support.

- NC-ReLU: minimal $\iff$ sparsest (neuron-wise) model.

Optimal Pruning: Minimal Models

Definition: An optimal C-ReLU model θ^* is minimal if there does not exist another optimal model θ' with strictly smaller support.

- NC-ReLU: minimal $\iff$ sparsest (neuron-wise) model.

Proposition 3.2 (informal): For $\lambda > 0$, $\theta \in \mathcal{W}^*(\lambda)$ is minimal iff the vectors $\{D_i X q_i\}_{\alpha_i \neq 0}$ are linearly independent.

Optimal Pruning: Minimal Models

Definition: An optimal C-ReLU model θ^* is minimal if there does not exist another optimal model θ' with strictly smaller support.

- NC-ReLU: minimal $\iff$ sparsest (neuron-wise) model.

Proposition 3.2 (informal): For $\lambda > 0$, $\theta \in \mathcal{W}^*(\lambda)$ is minimal iff the vectors $\{D_i X q_i\}_{\alpha_i \neq 0}$ are linearly independent.

- NC-ReLU: minimal if $(XW_{1i})_+$ are linearly independent.

Optimal Pruning: Minimal Models

Definition: An optimal C-ReLU model θ^* is minimal if there does not exist another optimal model θ' with strictly smaller support.

- NC-ReLU: minimal $\iff$ sparsest (neuron-wise) model.

Proposition 3.2 (informal): For $\lambda > 0$, $\theta \in \mathcal{W}^*(\lambda)$ is minimal iff the vectors $\{D_i X q_i\}_{\alpha_i \neq 0}$ are linearly independent.

- NC-ReLU: minimal if $(XW_{1i})_+$ are linearly independent.

Our Results:

1. We prove vertices of $\mathcal{W}^*(\lambda)$ are minimal models.

Optimal Pruning: Minimal Models

Definition: An optimal C-ReLU model θ^* is minimal if there does not exist another optimal model θ' with strictly smaller support.

- NC-ReLU: minimal $\iff$ sparsest (neuron-wise) model.

Proposition 3.2 (informal): For $\lambda > 0$, $\theta \in \mathcal{W}^*(\lambda)$ is minimal iff the vectors $\{D_i X q_i\}_{\alpha_i \neq 0}$ are linearly independent.

- NC-ReLU: minimal if $(XW_{1i})_+$ are linearly independent.

Our Results:

1. We prove vertices of $\mathcal{W}^*(\lambda)$ are minimal models.
2. There are at most n neurons in a minimal model.

Optimal Pruning: Minimal Models

Definition: An optimal C-ReLU model θ^* is minimal if there does not exist another optimal model θ' with strictly smaller support.

- NC-ReLU: minimal $\iff$ sparsest (neuron-wise) model.

Proposition 3.2 (informal): For $\lambda > 0$, $\theta \in \mathcal{W}^*(\lambda)$ is minimal iff the vectors $\{D_i X q_i\}_{\alpha_i \neq 0}$ are linearly independent.

- NC-ReLU: minimal if $(XW_{1i})_+$ are linearly independent.

Our Results:

1. We prove vertices of $\mathcal{W}^*(\lambda)$ are minimal models.
2. There are at most n neurons in a minimal model.
3. We give a poly-time algorithm for computing minimal models starting from any model θ.

Optimal Pruning: Algorithm

Algorithm Compute Minimal Model

Input: data matrix X, solution θ.

$k \leftarrow 0$.

$\theta^k \leftarrow \theta$.

while $\exists \beta \neq 0$ s.t. $\sum_{i \in \mathcal{A}_\lambda(\theta^k)} \beta_i D_i X \theta_i^k = 0$ **do**

  $i^k \leftarrow \arg\max_i \left\{ |\beta_i| : i \in \mathcal{A}_\lambda(\theta^k) \right\}$

  $t^k \leftarrow 1/|\beta_{i^k}|$

  $\theta^{k+1} \leftarrow \theta^k (1 - t^k \beta_i)$

  $k \leftarrow k + 1$

end while

Output: final weights θ^k

Optimal Pruning: Algorithm

Algorithm Compute Minimal Model

Input: data matrix X, solution θ.
$k \leftarrow 0$.
$\theta^k \leftarrow \theta$.
while $\exists \beta \neq 0$ s.t. $\sum_{i \in \mathcal{A}_\lambda(\theta^k)} \beta_i D_i X \theta_i^k = 0$ **do**
  $i^k \leftarrow \arg\max_i \left\{ |\beta_i| : i \in \mathcal{A}_\lambda(\theta^k) \right\}$
  $t^k \leftarrow 1/|\beta_{i^k}|$
  $\theta^{k+1} \leftarrow \theta^k (1 - t^k \beta_i)$
  $k \leftarrow k + 1$
end while
Output: final weights θ^k

- NC-ReLU: Returns sparsest network that is still optimal!

Optimal Pruning: Algorithm

Algorithm Compute Minimal Model

Input: data matrix X, solution θ.
$k \leftarrow 0$.
$\theta^k \leftarrow \theta$.
while $\exists \beta \neq 0$ s.t. $\sum_{i \in \mathcal{A}_\lambda(\theta^k)} \beta_i D_i X \theta_i^k = 0$ **do**
  $i^k \leftarrow \arg\max_i \left\{ |\beta_i| : i \in \mathcal{A}_\lambda(\theta^k) \right\}$
  $t^k \leftarrow 1/|\beta_{i^k}|$
  $\theta^{k+1} \leftarrow \theta^k(1 - t^k \beta_i)$
  $k \leftarrow k + 1$
end while
Output: final weights θ^k

- NC-ReLU: Returns sparsest network that is still optimal!
- Complexity: $O\left(n^3 l + nd\right)$ starting from l neurons.

Optimal Pruning: Theory vs Practice

Optimal Pruning in Theory:

- Starts from any neural network and returns a neuron-sparse model with the same predictions and weight-norm.

Optimal Pruning: Theory vs Practice

Optimal Pruning in Theory:

- Starts from any neural network and returns a neuron-sparse model with the same predictions and weight-norm.
- But it can't prune past minimum width: $m^* \leq n$...

Optimal Pruning: Theory vs Practice

Optimal Pruning in Theory:

- Starts from any neural network and returns a neuron-sparse model with the same predictions and weight-norm.
- But it can't prune past minimum width: $m^* \leq n\ldots$

Optimal Pruning in Practice:

- We propose a simple heuristic based on least-squares to prune past minimal models.

Optimal Pruning: Theory vs Practice

Optimal Pruning in Theory:

- Starts from any neural network and returns a neuron-sparse model with the same predictions and weight-norm.
- But it can't prune past minimum width: $m^* \leq n \dots$

Optimal Pruning in Practice:

- We propose a simple heuristic based on least-squares to prune past minimal models.
- Our heuristic works with any neuron pruning rule.

Optimal Pruning: Theory vs Practice

Optimal Pruning in Theory:

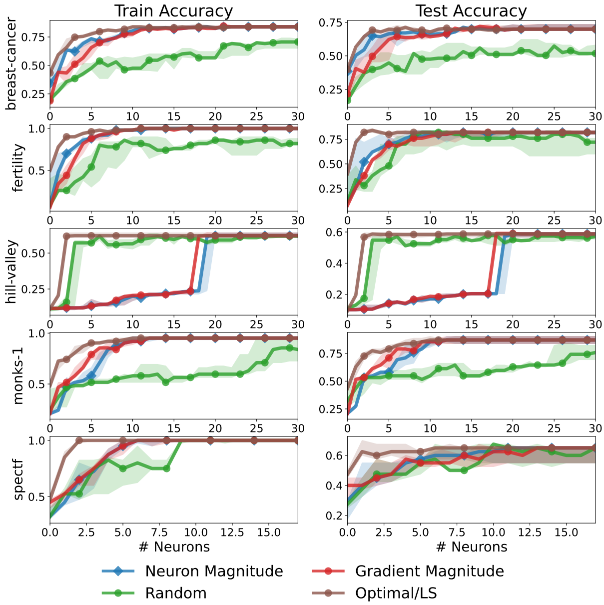
- Starts from any neural network and returns a neuron-sparse model with the same predictions and weight-norm.
- But it can't prune past minimum width: $m^* \leq n \ldots$

Optimal Pruning in Practice:

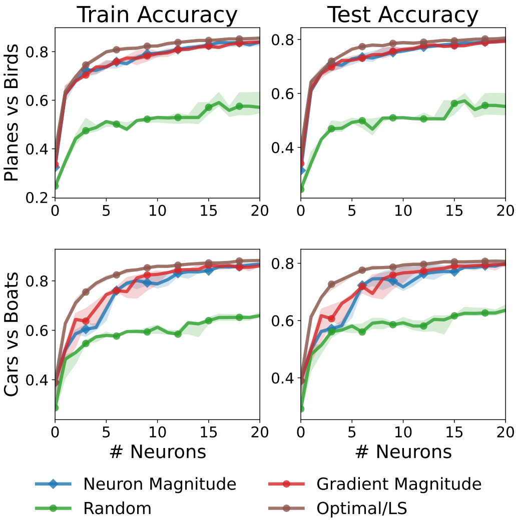
- We propose a simple heuristic based on least-squares to prune past minimal models.
- Our heuristic works with any neuron pruning rule.

Let's see how this does on real data!

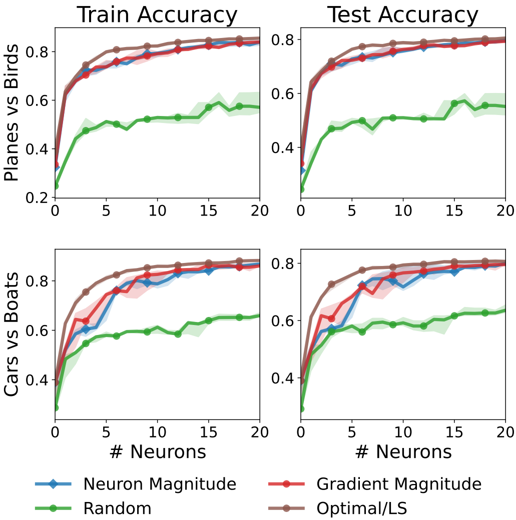
(Sub)-Optimal Pruning: UCI Datasets



(Sub)-Optimal Pruning: CIFAR-10



(Sub)-Optimal Pruning: CIFAR-10



(Sub)-optimal pruning dominates the naive baselines!

Summary

Our Contributions.

Our Contributions.

- **Optimal Sets**: We derive the set of all optimal two-layer ReLU neural networks.

Our Contributions.

- **Optimal Sets**: We derive the set of all optimal two-layer ReLU neural networks.
- **Regularization Paths**: We have some continuity results (see paper) and are working on more.

Our Contributions.

- **Optimal Sets**: We derive the set of all optimal two-layer ReLU neural networks.
- **Regularization Paths**: We have some continuity results (see paper) and are working on more.
- **Algorithms**: We give a poly-time algorithm for optimally pruning ReLU networks.

Try our Code!

References I

[Efr+04] Bradley Efron et al. "Least angle regression". In: *The Annals of statistics* 32.2 (2004), pp. 407–499.

[OPT00] Michael R Osborne, Brett Presnell, and Berwin A Turlach. "A new approach to variable selection in least squares problems". In: *IMA journal of numerical analysis* 20.3 (2000), pp. 389–403.

[PE20] 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.

[Tib13] Ryan J Tibshirani. "The lasso problem and uniqueness". In: *Electronic Journal of statistics* 7 (2013), pp. 1456–1490.

References II

[Win66] Robert O Winder. "Partitions of N-space by hyperplanes". In: *SIAM Journal on Applied Mathematics* 14.4 (1966), pp. 811–818.

Bonus: Explicit Optimal Set

We gave a characterization of $\mathcal{W}^*(\lambda)$ that depends on

$$\mathcal{S}_\lambda = \{i \in [2p] : \exists \theta \in \mathcal{W}^*(\lambda),\ \theta_i \neq 0\}.$$

Alternative expression involves additional linear constraints.

Bonus: Explicit Optimal Set

We gave a characterization of $\mathcal{W}^*(\lambda)$ that depends on

$$\mathcal{S}_\lambda = \{i \in [2p] : \exists \theta \in \mathcal{W}^*(\lambda),\ \theta_i \neq 0\}.$$

Alternative expression involves additional linear constraints.

$$\begin{aligned}\mathcal{W}^*(\lambda) = \big\{\theta : \forall\, i \in \mathcal{E}_\lambda, \theta_i = \alpha_i q_i, \alpha_i \geq 0,& \\ \forall\, j \in [2p] \setminus \mathcal{E}_\lambda, \theta_j = 0, \sum_{i=1}^{2p} D_i X \theta_i = \hat{y},& \\ \forall\, i \in [2p], K_i \theta_i \geq 0, \langle \rho, K_i \theta_i \rangle = 0.\big\}&\end{aligned}$$

Bonus: Explicit Optimal Set

We gave a characterization of $\mathcal{W}^*(\lambda)$ that depends on

$$\mathcal{S}_\lambda = \{i \in [2p] : \exists \theta \in \mathcal{W}^*(\lambda),\ \theta_i \neq 0\}\,.$$

Alternative expression involves additional linear constraints.

$$\begin{aligned}\mathcal{W}^*(\lambda) = \big\{\theta : \forall\, i \in \mathcal{E}_\lambda, \theta_i = \alpha_i q_i, \alpha_i \geq 0,& \\ \forall\, j \in [2p] \setminus \mathcal{E}_\lambda, \theta_j = 0, \sum_{i=1}^{2p} D_i X \theta_i = \hat{y},& \\ \forall\, i \in [2p], K_i\theta_i \geq 0, \langle \rho, K_i\theta_i \rangle = 0.\big\}&\end{aligned}$$

More complex, but also **explicit**.

Bonus: Solution Mapping for C-ReLU

Given (v^*, w^*), an optimal non-convex ReLU network is given by

C to NC:

$$W_{1i} = v_i^* / \sqrt{\|v_i^*\|}, \quad w_{2i} = \sqrt{\|v_i^*\|}$$
$$W_{1j} = w_i^* / \sqrt{\|w_i^*\|}, \quad w_{2j} = -\sqrt{\|w_i^*\|}.$$

Bonus: Solution Mapping for C-ReLU

Given (v^*, w^*), an optimal non-convex ReLU network is given by

C to NC:

$$W_{1i} = v_i^* / \sqrt{\|v_i^*\|}, \quad w_{2i} = \sqrt{\|v_i^*\|}$$
$$W_{1j} = w_i^* / \sqrt{\|w_i^*\|}, \quad w_{2j} = -\sqrt{\|w_i^*\|}.$$

- Optimal convex weights satisfy $v_i^* = \alpha_i q_i$ so that

$$\|v_i^*\|_2 = \alpha_i \|q_i\|_2 = \alpha_i \lambda.$$

Bonus: Solution Mapping for C-ReLU

Given (v^*, w^*), an optimal non-convex ReLU network is given by

C to NC:
$$W_{1i} = v_i^* / \sqrt{\|v_i^*\|}, \quad w_{2i} = \sqrt{\|v_i^*\|}$$
$$W_{1j} = w_i^* / \sqrt{\|w_i^*\|}, \quad w_{2j} = -\sqrt{\|w_i^*\|}.$$

- Optimal convex weights satisfy $v_i^* = \alpha_i q_i$ so that
$$\|v_i^*\|_2 = \alpha_i \|q_i\|_2 = \alpha_i \lambda.$$

Recall structure of **non-convex optima**:

$$\begin{aligned}\mathcal{O}_\lambda = \big\{&(W_1, w_2) : \ f_{W_1, w_2}(X) = \hat{y}, \\ &\forall\, i \in \mathcal{S}_\lambda, W_{1i} = ({}^{\alpha_i}/\lambda)^{1/2} q_i, w_{2i} = (\alpha_i \lambda)^{1/2}, \alpha_i \geq 0 \\ &\forall\, i \in [2p] \setminus \mathcal{S}_\lambda, W_{1i} = 0, \ w_{2i} = 0\big\}.\end{aligned}$$