Algebraic Approach to Ridge-Regularized Mean Squared Error Minimization in Minimal ReLU Neural Network (arXiv:2508.17783; with R. Fukasaku, Y. Kabata)

Akifumi Okuno[1,2,3]

[1]Inst. Stat. Math., [2]SOKENDAI, [3]RIKEN (AIP/CBS)

https://okuno.net/slides/2026-02-ISM-ISI-ISSAS.pdf

What is Computational Algebra?

- $f_1, \ldots, f_r \in \mathbb{R}[\psi]$ are real polynomials (e.g., $f_1(\psi) = \psi_1^2\psi_3 + 2\psi_2 - 1$).

Roughly speaking, computational algebra can solve simultaneous *polynomial* equation[1]:

$$f_1(\psi) = 0,\ f_2(\psi) = 0, \quad \cdots, \quad f_r(\psi) = 0.$$

```
f1[ψ1_, ψ2_] := ψ1^2 - 2 ψ2^2;
f2[ψ1_, ψ2_] := ψ1^2 + 3 ψ2;

Solve[{f1[ψ1, ψ2] == 0, f2[ψ1, ψ2] == 0}, {ψ1, ψ2}]
解く
```

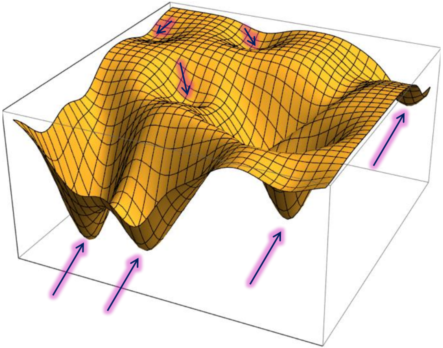
$$\left\{\{\psi1 \to 0, \psi2 \to 0\}, \left\{\psi1 \to -\frac{3}{\sqrt{2}}, \psi2 \to -\frac{3}{2}\right\}, \left\{\psi1 \to \frac{3}{\sqrt{2}}, \psi2 \to -\frac{3}{2}\right\}\right\}$$

Figure: A popular example: Mathematica

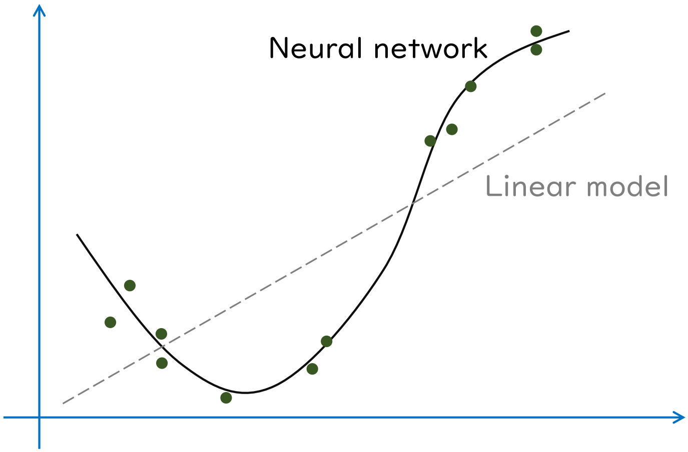
[1] It provides a simpler form of the affine variety $\mathbb{V}(f_1, \ldots, f_r) = \{\psi \in \Psi \mid f_1(\psi) = \cdots = f_r(\psi) = 0\}$

In This Study... Overview

- Using computational algebra, we enumerate all the local minima of the ReLU neural network loss functions.
  (Fukasaku, Kabata, and Okuno; arXiv:2508.17783)



Foundations and Challenges of Neural Networks



Neural networks are flexible nonlinear predictive models.

Definition of Neural Networks

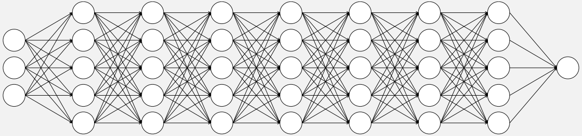
- **Linear regression model**:

$$f_\theta^{\mathrm{LM}}(x) = Wx + b$$

- **Neural network** (whose special case is the perceptron):

$$f_\theta^{\mathrm{NN}}(x) = W^{(Q+1)}\sigma\left(W^{(Q)}\sigma\left(\cdots\sigma\left(W^{(1)}x + b^{(1)}\right)\cdots\right) + b^{(Q)}\right) + b^{(Q+1)}.$$

  - σ is the activation function, applied elementwise (e.g., $1/(1+\exp(-z))$ or $\mathrm{ReLU}(z) = \max\{0, z\}$).
  - Many other architectures exist beyond this form.
  - When the number of layers Q is large, we refer to it as a deep neural network.
  - NN has *universal approximation capability*.

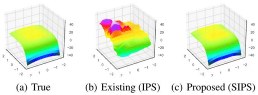


A Wide Variety of Applications

(Generated by ChatGPT)

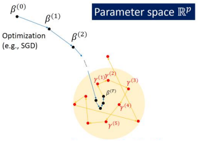
...But Reality Is Not That Simple

- From the viewpoint of statistical science, many essential issues remain unresolved.



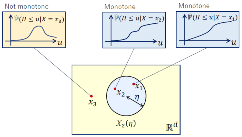
More Expressive Siamese NN

Okuno et al. (AISTATS2019)



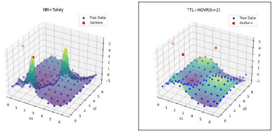
WAIC + Overparameterized NN
+ Langevin dynamics

Okuno and Yano (JCGS2023)



NN + Ordinal Regression

Okuno and Harada (JCGS2024)



NN + Variation Regularization

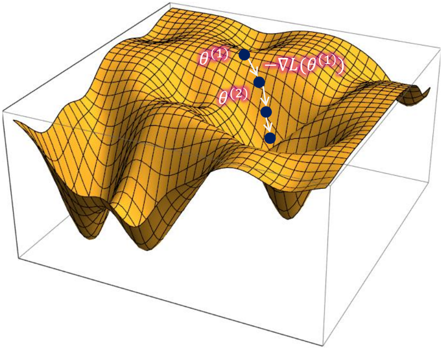
Okuno and Yagishita (in revision)

Despite substantial progress, the theoretical picture remains unclear.

Core Difficulties: The Loss Landscape Is Extremely Bumpy

- Gradient descent update:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma \nabla L(\theta^{(t)}).$$

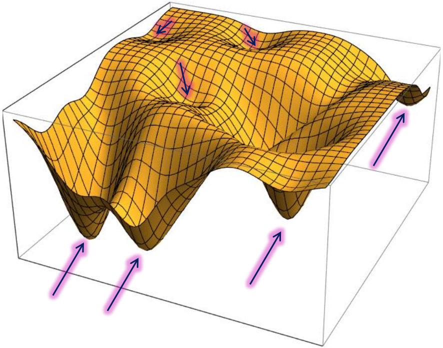


- For concave (single-valley) functions, many theoretical guarantees exist.
- For multimodal loss, the convergence limit depends on initial parameter.
- We can't know: <u>how many solutions? are they isolated?</u>

Goals and Starting Points

What We Ultimately Want to Do

We want to *enumerate all local minima* of the loss function.



- How many solutions?
- Are they isolated? or form high-dimensional solution sets?
- We leverage computational algebra to list all the solutions!

Algebraic Representation of ReLU Activation

The ReLU activation $\sigma(z) = \max\{0, z\}$ can be expressed via activation patterns.

For fixed $W \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, and $x \in \mathbb{R}^d$, there exists $e = e(W, b, x) \in \{0, 1\}^m$ such that

$$\mathrm{ReLU}(Wx + b) = \mathrm{diag}(e)(Wx + b),$$

where $\mathrm{diag}(e)$ is the diagonal matrix with diagonal entries e.

- Example: If $Wx + b = (3, -2, 2, 1, -1)$, then $e = (1, 0, 1, 1, 0)$ and

$$\mathrm{ReLU}(Wx + b) = (3, 0, 2, 1, 0) = \mathrm{diag}(e)(Wx + b).$$

- Arora et al. (2018), Pilanci and Ergen (2020), Mishkin et al. (2022), etc.

Generalization to Multi-layer Networks

For parameters $\theta = (W^{(\ell)}, b^{(\ell)})_{\ell=1}^{L}$ and fixed input $x \in \mathbb{R}^d$, each layer $\ell = 1, \ldots, L$ has an activation pattern $e^{(\ell)} = e^{(\ell)}(\theta, x) \in \{0, 1\}^{m_\ell}$ such that

$$f^{\mathrm{NN}}_{\theta,E}(x) = W^{(Q+1)}\mathrm{diag}(e^{(Q)})\Big\{W^{(Q)}\mathrm{diag}(e^{(Q-1)})\{\cdots \\ \cdots \mathrm{diag}(e^{(1)})(W^{(1)}x + b^{(1)})\cdots\} + b^{(Q)}\Big\} + b^{(Q+1)}.$$

- If $E = (e^{(\ell)})$ is fixed, the ReLU network reduces to matrix product.
- The loss

$$\ell_{\lambda,E}(\theta) = \sum_{i=1}^{n}\{y_i - f^{\mathrm{NN}}_{\theta,E}(x_i)\}^2 + \lambda\|\theta\|_2^2$$

  becomes a polynomial in the parameters.

Our Basic Idea

- The loss $\ell_{\lambda,E}(\theta)$ is a polynomial in θ.
- Its minimizer should satisfy the estimating equation:

$$\frac{\partial \ell_{\lambda,E}(\theta)}{\partial \theta} = 0,$$

  which is also a polynomial system.
- This is precisely the type of problem addressed by computational algebra.

Our Work (Fukasaku, Kabata, and Okuno; arXiv:2508.17783)

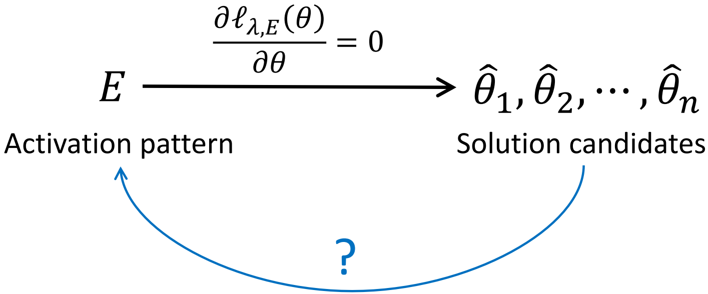
So in Principle...

$$\frac{\partial \ell_{\lambda,E}(\theta)}{\partial \theta} = \frac{\partial \left\{ \sum_{i=1}^{n} (y_i - f_{\theta,E}(x_i))^2 + \lambda \|\theta\|_2^2 \right\}}{\partial \theta} = 0$$

If we could simply solve this equation, everything would be resolved⋯

But, things are not so easy in practice...

Difficulties



Problem:
Solutions does not necessarily correspond to the activation pattern E

Our strategy

$$E_1 \xrightarrow{\frac{\partial \ell_{\lambda,E_1}(\theta)}{\partial \theta} = 0} \hat{\theta}_{E_1,1}, \hat{\theta}_{E_1,2}, \cdots, \hat{\theta}_{E_1,n_1}$$

$$E_2 \xrightarrow{\frac{\partial \ell_{\lambda,E_2}(\theta)}{\partial \theta} = 0} \hat{\theta}_{E_2,1}, \hat{\theta}_{E_2,2}, \cdots, \hat{\theta}_{E_2,n_2}$$

$$E_3 \xrightarrow{\frac{\partial \ell_{\lambda,E_3}(\theta)}{\partial \theta} = 0} \hat{\theta}_{E_3,1}, \hat{\theta}_{E_3,2}, \cdots, \hat{\theta}_{E_3,n_3}$$

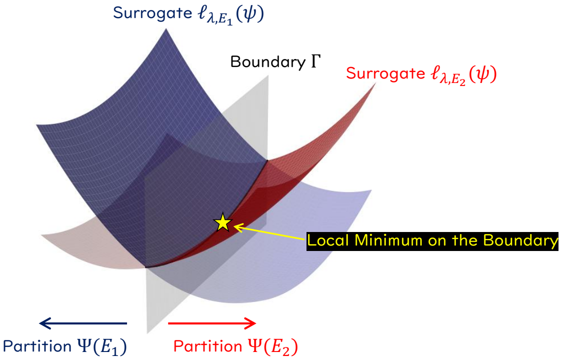
$$\vdots$$

Merge solutions

Filter out candidates
that does not satisfy the activation pattern

Why Boundary Solutions Are Difficult

- For neighboring activation patterns $E_1, E_2 \in \{-1, +1\}^{n \times L}$, the surrogate losses ℓ_{λ, E_1} and ℓ_{λ, E_2} may each have minimizers *on the shared boundary*.



- Across the full space Ψ, neither surrogate may produce local minima. Yet *on the boundary*, switching between the surrogates can create new local minima.

Local Minima on the Boundary

- A point ψ lies on a boundary if $\xi_{i\ell}(\psi) = [\![b_\ell , x_i]\!] + c_\ell = 0$ for some (i, ℓ).
- Solve the Lagrange multiplier system:

$$\frac{\partial}{\partial \psi}\{\ell_{\lambda,E}(\psi) + \beta\,\xi_{i\ell}(\psi)\} = 0$$

which is a system of rational equations.

FKO (arXiv:2508.17783) Theorem 2

Any local minimum of ℓ_λ is either

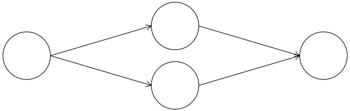
(1) an interior local minimizer of some region $\Psi(E)$, or

(2) a local minimizer on a boundary between regions.

- Hence all local minima arise as solutions of polynomial (or rational) equations.

A Concrete Example

- Input dimension $d = 1$, number of units $L = 2$, sample size $n = 5$.



$$(x_1, y_1) = (-0.17,\ 0.05), \quad (x_2, y_2) = (0.44,\ 1.02), \quad (x_3, y_3) = (-1.00,\ 0.61),$$
$$(x_4, y_4) = (-0.40,\ -0.36), \quad (x_5, y_5) = (-0.71,\ -1.32).$$

- The number of possible activation patterns is $2^{nL} = 1024$.[2]

[2] So we must compute 1024 Gröbner bases!

Obtained Solutions:

Under the setting in the previous slide,
the ridge-regularized loss function for the ReLU neural network has:

- 1 one-dimensional solution set (in the interior of a partition), and
- 8 isolated local minima (on the boundary).

Observations:

- All isolated minima lie on the activation-pattern boundary...!
- Ridge regularization does not necessarily make the minimizers isolated.

Detected one-dimensional solution set:

A one-dimensional solution set of $\psi = (b_{11}, b_{12}, c1, c2)$ is specified by:

$$c_1 - \frac{17b_{11}}{100} > 0, \quad c_2 - \frac{17b_{21}}{100} > 0, \quad \frac{11b_{11}}{25} + c_1 > 0, \quad \frac{11b_{21}}{25} + c_2 > 0,$$

$$c_1 - B_{11} < 0, \quad c_2 - B_{21} < 0, \quad c_1 - \frac{2b_{11}}{5} < 0, \quad c_2 - \frac{2b_{21}}{5} < 0,$$

$$c_1 - \frac{71b_{11}}{100} < 0 \quad c_2 - \frac{71b_{21}}{100} < 0,$$

$$0 = b_{11} + R_1 c_1^7 + R_2 c_1^5 c_2^2 + R_3 c_1^5 + R_4 c_1^3 c_2^4 + R_5 c_1^3 c_2^2 + R_6 c_1^3 + R_7 c_1 c_2^6 \cdots - R_{10} c_1,$$

$$0 = b_{21} + R_{11} c_1^6 c_2 + R_{12} c_1^4 c_2^3 + R_{13} c_1^4 c_2 + R_{14} c_1^2 c_2^5 + R_{15} c_1^2 c_2^3 + R_{16} c_1^2 c_2 \cdots - R_{20} c_2,$$

$$0 = c_1^8 + 4c_1^6 c_2^2 + R_{21} c_1^6 + 6c_1^4 c_2^4 + R_{22} c_1^4 c_2^2 + R_{23} c_1^4 + 4c_1^2 c_2^6 + R_{24} c_1^2 c_2^4 \cdots - R_{30},$$

where $R_1, R_2, \ldots, R_{30}$ are complicated rational numbers.

Coefficients I

$$R_1 = \frac{8061831845311915622677137119327762091177021647160801855468750}{799152119487995315448496053126952456312807787539926712735352143},$$
$$R_2 = \frac{24185495535935746868031411357983286273531064941482405566406250}{799152119487995315448496053126952456312807787539926712735352143},$$
$$R_3 = \frac{16592903810388605869109122181308724918558592156970414314140625}{114164588498285045064070864732421779473258255362846673247907449},$$
$$R_4 = \frac{24185495535935746868031411357983286273531064941482405566406250}{799152119487995315448496053126952456312807787539926712735352143},$$
$$R_5 = \frac{33185807620777211738218244362617449837117184313940828628281250}{114164588498285045064070864732421779473258255362846673247907449},$$
$$R_6 = \frac{3631820373341883747515837259737976349533140846091509424078088125}{913316707986280360512566917859374235786066042902773385983259592},$$
$$R_7 = \frac{8061831845311915622677137119327762091177021647160801855468750}{799152119487995315448496053126952456312807787539926712735352143},$$
$$R_8 = \frac{16592903810388605869109122181308724918558592156970414314140625}{114164588498285045064070864732421779473258255362846673247907449},$$
$$R_9 = \frac{3631820373341883747515837259737976349533140846091509424078088125}{913316707986280360512566917859374235786066042902773385983259592},$$
$$R_{10} = \frac{3986185952593039040079422065453083833713933848669678031131169525}{799152119487995315448496053126952456312807787539926712735352143},$$
$$R_{11} = \frac{8061831845311915622677137119327762091177021647160801855468750}{799152119487995315448496053126952456312807787539926712735352143},$$
$$R_{12} = \frac{24185495535935746868031411357983286273531064941482405566406250}{799152119487995315448496053126952456312807787539926712735352143},$$
$$R_{13} = \frac{16592903810388605869109122181308724918558592156970414314140625}{114164588498285045064070864732421779473258255362846673247907449},$$
$$R_{14} = \frac{24185495535935746868031411357983286273531064941482405566406250}{799152119487995315448496053126952456312807787539926712735352143},$$
$$R_{15} = \frac{33185807620777211738218244362617449837117184313940828628281250}{114164588498285045064070864732421779473258255362846673247907449},$$
$$R_{16} = \frac{3631820373341883747515837259737976349533140846091509424078088125}{913316707986280360512566917859374235786066042902773385983259592},$$

Coefficients II

$$R_{17} = \frac{806183184531191562267713711932776209117702164716080185546875\,0}{7991521194879953154484960531269524563128077875399267127353521 43},$$

$$R_{18} = \frac{16592903810388605869109122181308724918555859215697041431414062 5}{11416458849828504506407086473242177947325825536284667324790744 9},$$

$$R_{19} = \frac{3631820373341883747515837259737976349533140846091509424078088125}{913316707986280360512566917859374235786066042902773385983259592},$$

$$R_{20} = \frac{3986185952593039040079422065453083833713933848669678031131169525}{799152119487995315448496053126952456312807787539926712735352143},$$

$$R_{21} = \frac{916767969161863 07}{5836063856703750},$$

$$R_{22} = \frac{9167679691618630 7}{1945354618901250},$$

$$R_{23} = \frac{10799719744535841949933618669}{263849322371155043062500 00},$$

$$R_{24} = \frac{91676796916186307}{1945354618901250},$$

$$R_{25} = \frac{10799719744535841949933618669}{1319246611855775215312500 0},$$

$$R_{26} = \frac{1170757087686584669238812}{3298116529639438038281 25},$$

$$R_{27} = \frac{91676796916186307}{5836063856703750},$$

$$R_{28} = \frac{10799719744535841949933618669}{26384932237115504306250000},$$

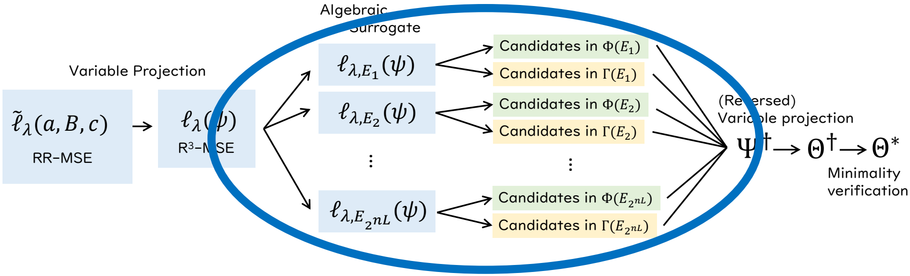
$$R_{29} = \frac{1170757087686584669238812}{329811652963943803828125},$$

$$R_{30} = \frac{16870323239553700909764929 29}{10306614155123243869628906 25}.$$

Towards the Future

Remaining Challenges

- The computational cost is extremely large.
  - Increasing the number of parameters ⇒ both per-pattern computation and parallel load increase.
  - Increasing the sample size ⇒ the number of activation patterns increases exponentially.



- Future work includes parallelization and fast computation of Gröbner bases for the associated polynomial systems.

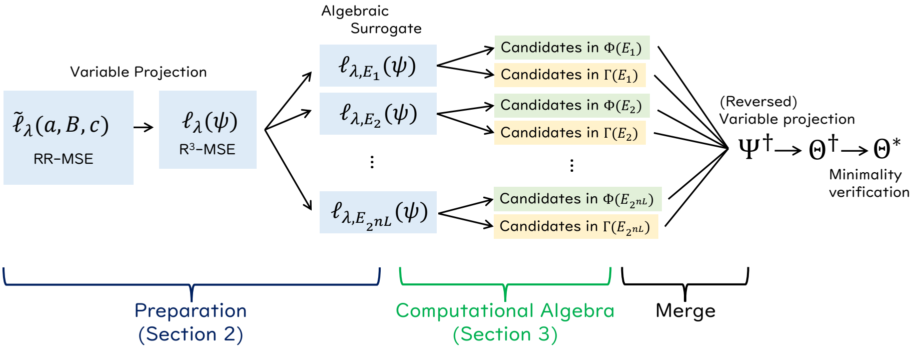
arXiv:2508.17783

Please feel free to contact me: okuno@ism.ac.jp

https://okuno.net/slides/2026-02-ISM-ISI-ISSAS.pdf

Details

Overall Procedure



- Enumerating interior local minimizers (candidates) is relatively straightforward.
- Boundary solutions, however, are much more subtle.

Detailed Setup and Simplifying Assumptions

- For simplicity, restrict attention to a network with $Q = 1$ hidden layer:[3]

$$f_\theta^{\mathrm{NN}}(x) = [\![\, a\,,\, \mathrm{ReLU}(Bx + c)\,]\!]\,, \quad \theta = (a, B, c),$$

  where the number of units is L ($a, c \in \mathbb{R}^L$, $B \in \mathbb{R}^{L \times d}$).

- Eliminate a in advance. Define $\psi = (B, c)$ and consider

$$\ell_\lambda(\psi) = \min_a \left\{ \sum_{i=1}^n (y_i - f_\theta(x_i))^2 + \lambda \|\theta\|_2^2 \right\}.$$

- The minimizer in a is given analytically (ridge regression), so $\ell_\lambda(\psi)$ becomes a rational function. We therefore minimize $\ell_\lambda(\psi)$ algebraically.

[3] The essential ideas extend to general depth.

Activation Patterns and Partitioning of Parameter Space

- Consider a dataset $\{(x_i, y_i)\}_{i=1}^{n}$.
- Define $\xi_{i\ell}(\psi) = [\![b_\ell , x_i]\!] + c_\ell$ and

$$e_{i\ell} = e_{i\ell}(\psi) = \begin{cases} 1 & \text{if } \xi_{i\ell}(\psi) \geq 0, \\ -1 & \text{if } \xi_{i\ell}(\psi) < 0. \end{cases}$$

  (We now use ± 1 instead of $\{0, 1\}$ for convenience.)
- Then

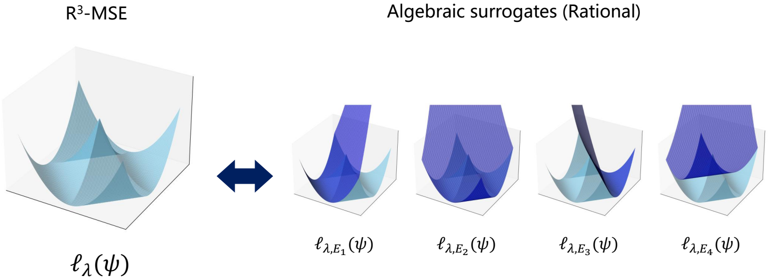
$$\mathrm{ReLU}(\xi_{i\ell}(\psi)) = \frac{e_{i\ell} + 1}{2} \xi_{i\ell}(\psi).$$

- Define the region of parameters yielding activation pattern E:

$$\Psi(E) = \{\psi \in \Psi \mid \xi_{i\ell}(\psi) e_{i\ell} \geq 0, \ \forall i, \ell\}.$$

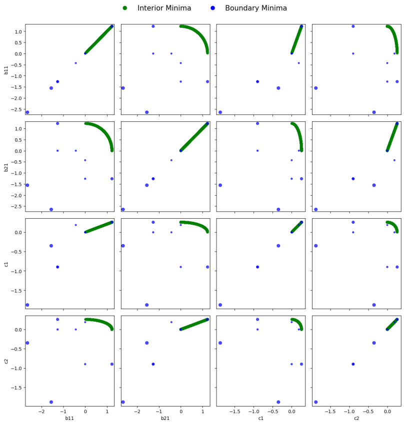
Function Decomposition and Surrogate Losses

- Our true objective is to minimize $\ell_\lambda(\psi)$.
- Partition parameter space into $\Psi(E_1), \Psi(E_2), \ldots$ based on activation patterns. In each region, $\ell_\lambda(\psi)$ equals a surrogate $\ell_{\lambda,E}(\psi)$ consistent with pattern E.



- The solutions (especially, interior points of each region) of $\frac{\partial \ell_{\lambda,E}(\psi)}{\partial \psi} = 0$ can be obtained by computational algebra.

Visualization of Local Minima



- Despite ridge regularization, an entire 1-dimensional solution set appears.
- All isolated points turned out to lie on boundaries.