Existence of optima in sparse matrix factorization and sparse ReLU networks training

Le Quoc Tung, 27 June 2023

Léon Zheng

Elisa Riccietti

Rémi Gribonval

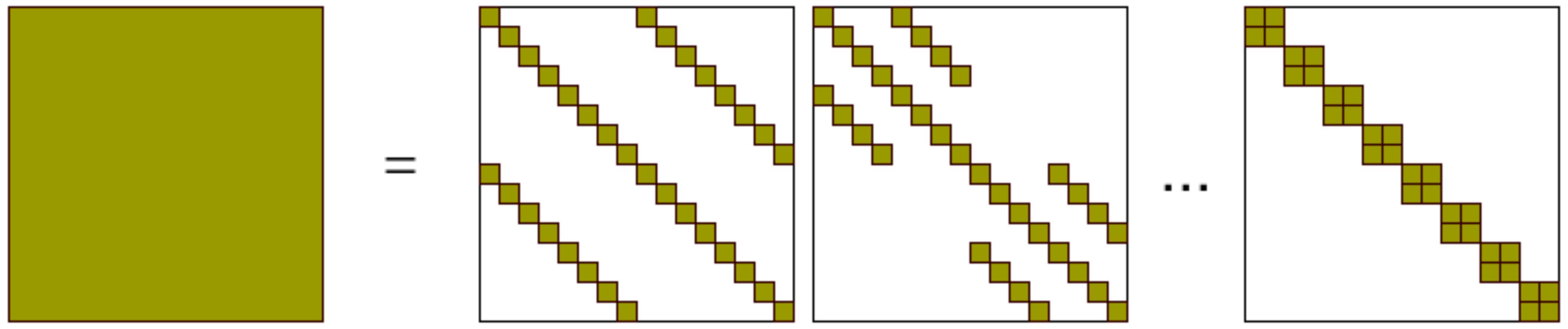
Sparse matrix factorization

OBJECTIVES: Given A, find some ***sparse*** matrices $X_\ell, \ell = 1,\ldots,L$, such that:

$$A \approx X_1 \ldots X_L$$

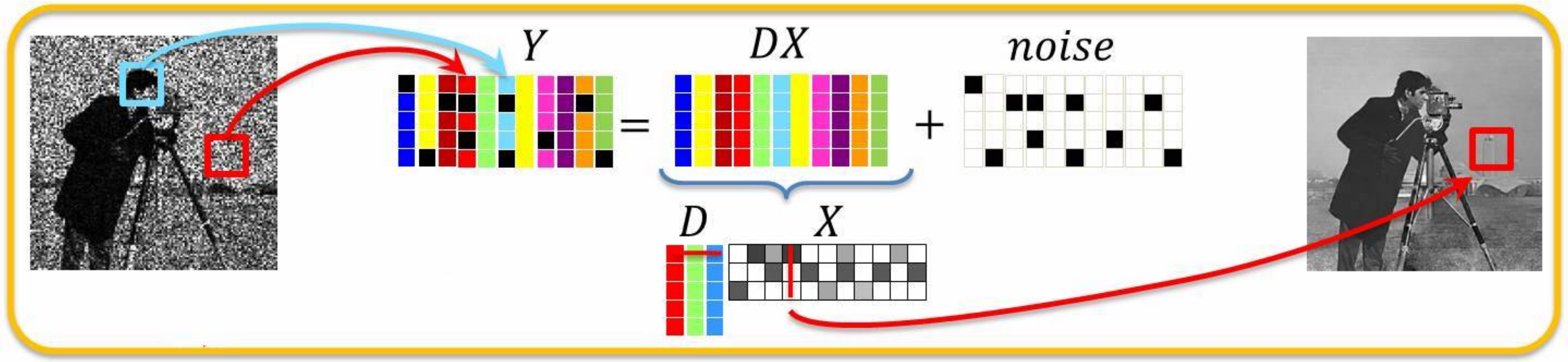
APPLICATIONS: Accelerating matrix-vector multiplication, data analysis, etc.

$$Ax \approx X_1(X_2 \ldots (X_L x)), \forall x$$



Fast Fourier Transformation

$Y = DX, \quad X$ sparse



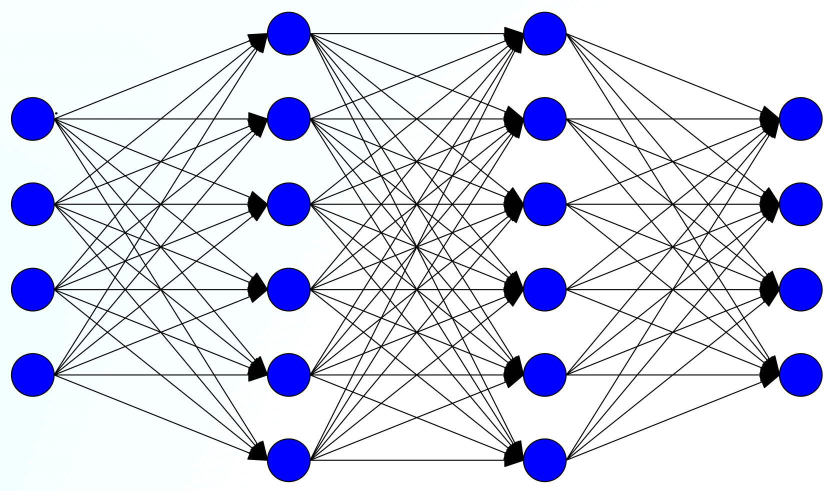
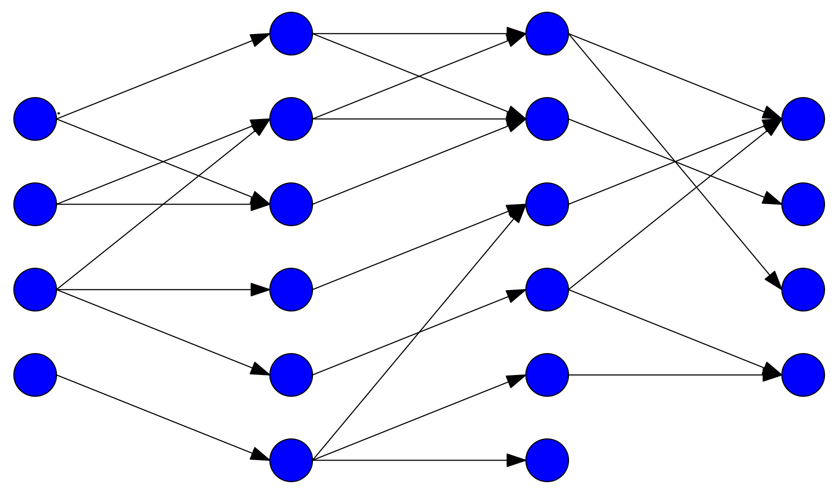
Dictionary learning

ReLU neural networks and sparse ReLU neural networks

DEFINITIONS: Given ***weight matrices*** $W^{(\ell)}$ and ***bias vectors*** $b^{(\ell)}, \ell = 1, \ldots, L$

$$x \mapsto W^{(L)}\sigma(\ldots\sigma(W^{(1)}x + b^{(1)}) + \ldots) + b^{(L)}$$

$\sigma : \mathbb{R} \mapsto \mathbb{R} : \sigma(x) = \max(0,x)$ is the ReLU activation function

Conventional Deep Neural Networks	Sparse Deep Neural Networks
	
The weight matrices are dense	**The weight matrices are sparse**

Sparse matrix factorization formulation

OPTIMIZATION FORMULATIONS:

Given A and $\mathscr{E}_j$ some sets of ***sparse*** matrices, solve:

$$\min_{S^{(1)},\ldots,S^{(J)}} \|A - \prod_{j=1}^{L} S^{(j)}\|_F^2 \text{ subject to: } S^{(j)} \in \mathscr{E}_j, \forall j \in \{1,\ldots,L\}$$

Choice of sparse matrices set $\mathscr{E}_j$

- k-sparse per row,
- k-sparse per column
- k-sparse in total

COMPLEXITY: Problem is **NP-hard** in general **(Malik, IPL 2017), (S.Foucart, H. Rauhut, ANNA 2013)**

Sparse ReLU neural networks (NNs) training

OPTIMIZATION FORMULATIONS:

Given data set $\mathscr{D} := (X, Y)$ and $\mathscr{E}_j$ some sets of ***sparse*** matrices, solve:

$$\min_{W^{(j)}, b^{(j)}} \quad \|Y - W^{(L)}\sigma(\ldots\sigma(W^{(1)}X + b^{(1)}) + \ldots) + b^{(L)}\|_F^2$$

$$\text{subject to:} \quad W^{(j)} \in \mathscr{E}_j, \forall j \in \{1, \ldots, L\}$$

Practical choice of sparse matrices set $\mathscr{E}_j$: k-sparse in total

(J. Frankle, M. Carbin, ICLR 2019), (S. Han, H. Mao, W-J. Dally, ICLR 2016)

COMPLEXITY: Not known yet.

Expected to be difficult since training classical ReLU NNs is **NP-hard**.

(R. Livni, S. Shalev-Shwartz, O. Shamir, NeuRIPS 2014), (D. Boob, S-S. Dey, G. Lan, Discrete Optimization 2022)

→ **How to deal with these problems?**

Fixed support matrix factorization

Back to sparse matrix factorization

SPECIAL CASE OF SPARSE MATRIX FACTORISATION

SPARSE MATRIX FACTORISATION

$$\min_{S^{(1)},\ldots,S^{(J)}} \|A - \prod_{j=1}^{L} S^{(j)}\|_F^2 \text{ subject to: } S^{(j)} \in \mathcal{E}_j, \forall j \in \{1,\ldots,L\}$$

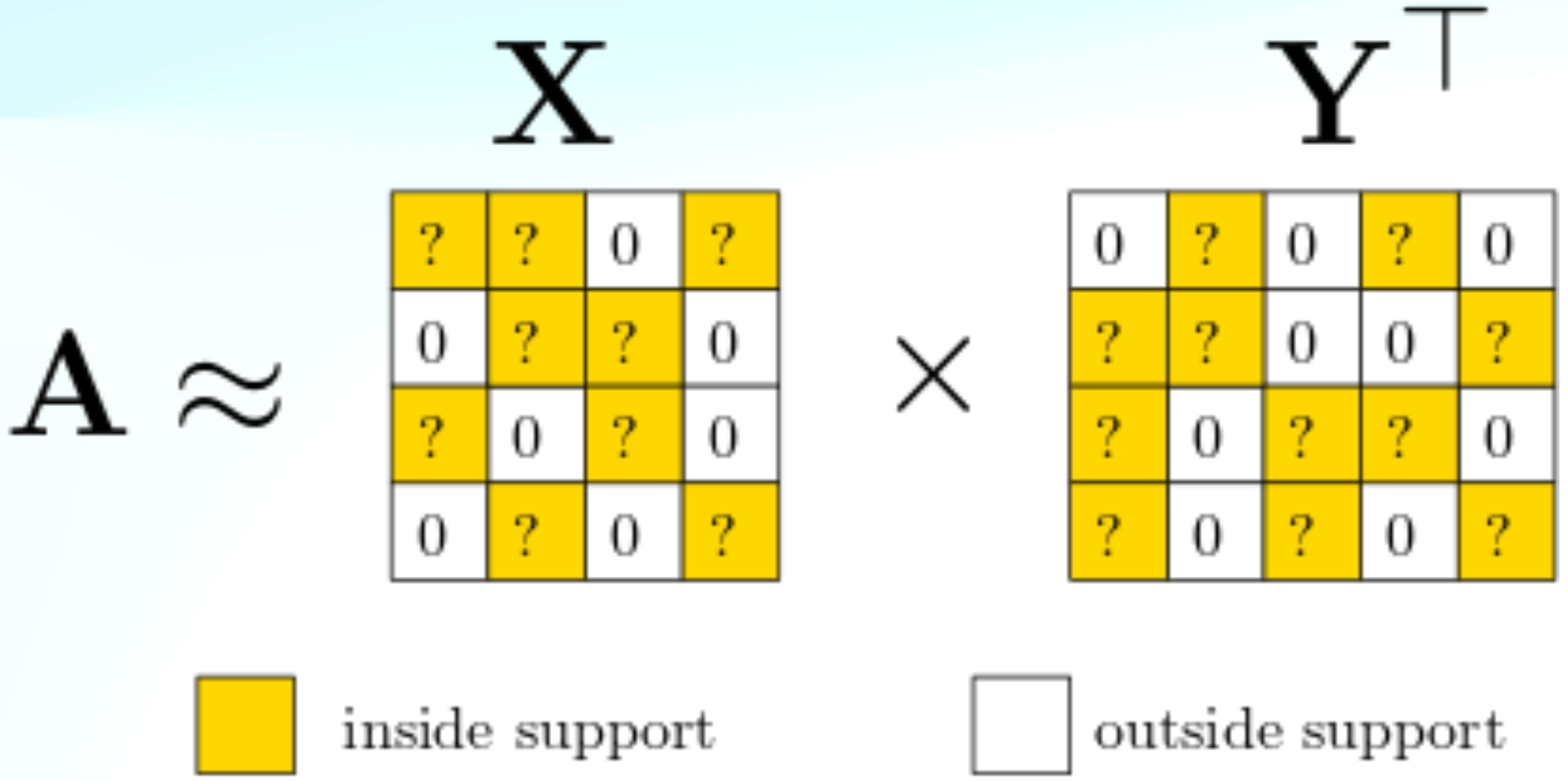
- $L = 2$
- $(\mathcal{E}_1, \mathcal{E}_2)$: set of matrices whose **support** are included in I and J

FIXED SUPPORT MATRIX FACTORISATION

$$\min_{X,Y} \|A - XY^\top\|_F^2 \text{ subject to: } \text{supp}(X) \subseteq I, \text{supp}(Y) \subseteq J$$

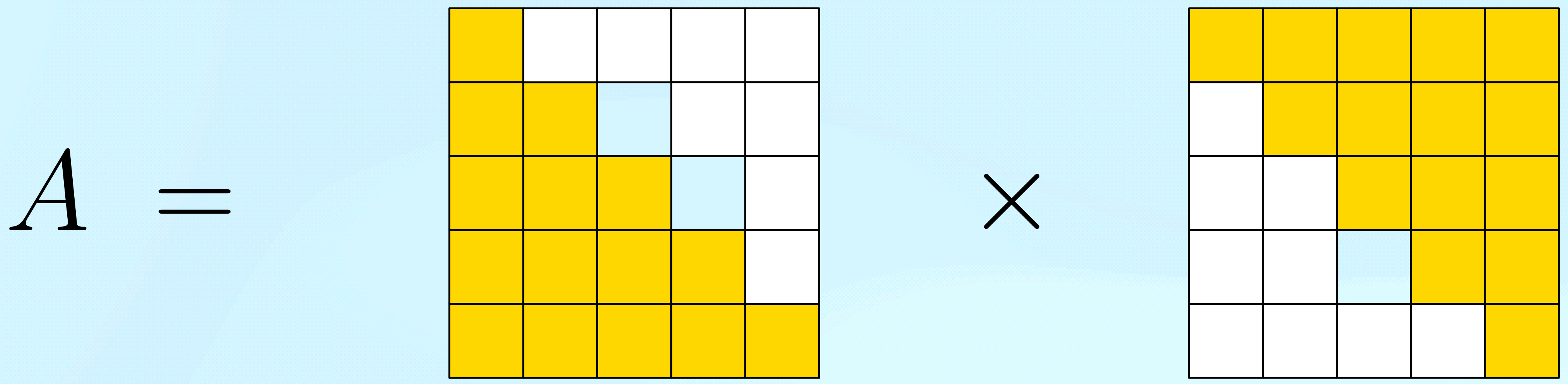
Fixed support matrix factorization (FSMF)

$$\min_{X,Y} \|A - XY^\top\|_F^2 \text{ subject to: } \mathrm{supp}(X) \subseteq I, \mathrm{supp}(Y) \subseteq J$$

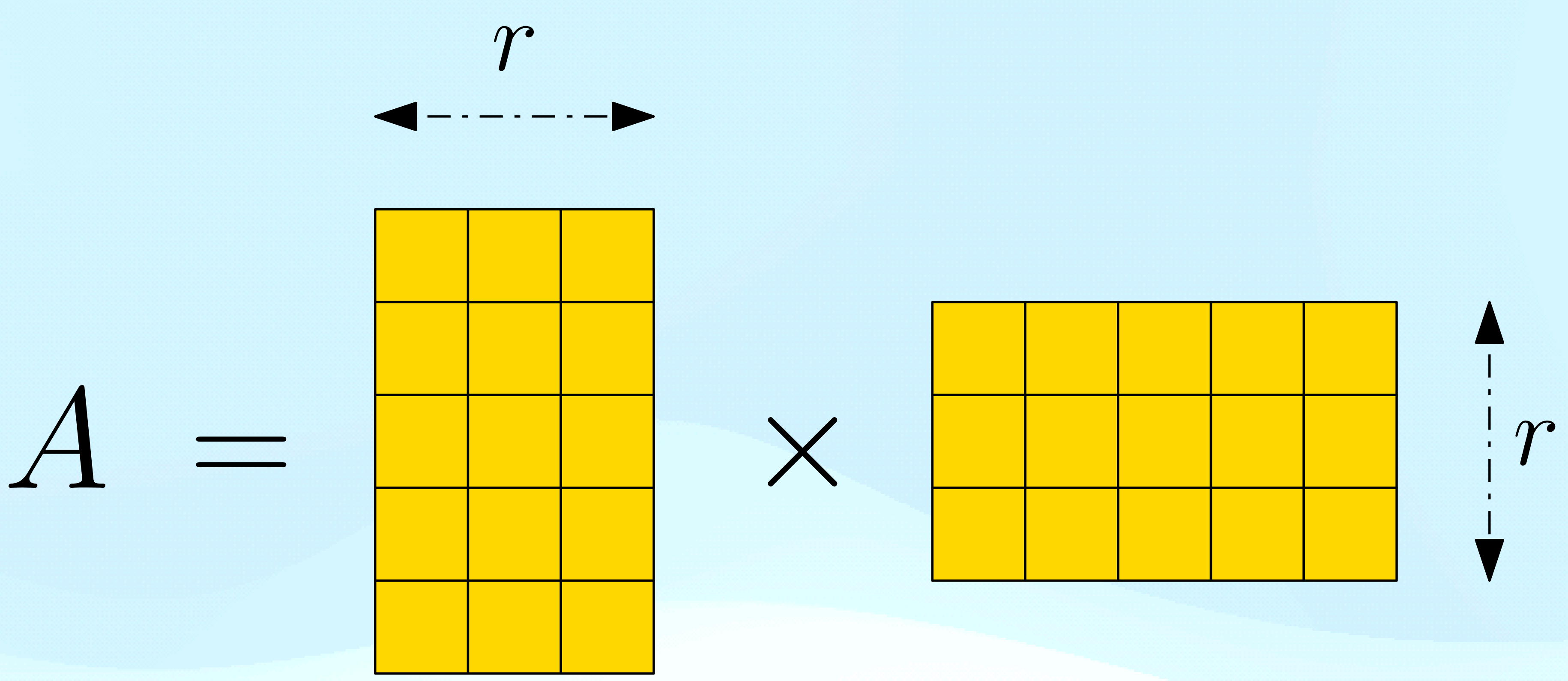


SUPPORT CONTRAINTS

Why Fixed Support Matrix Factorization?

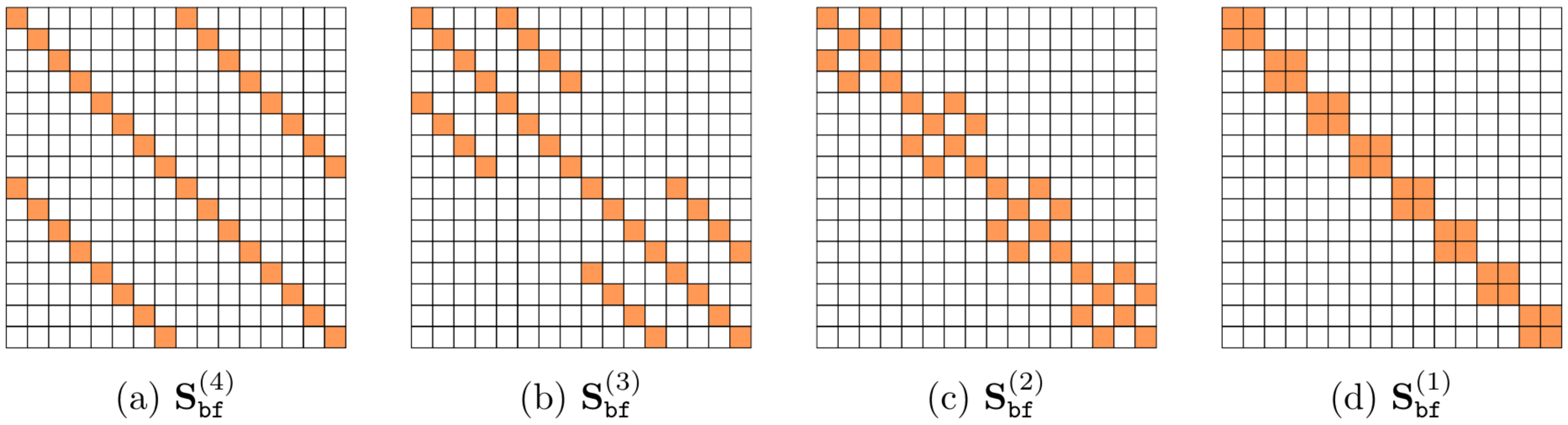


LU decomposition



Low rank approximation

Hierarchical matrix



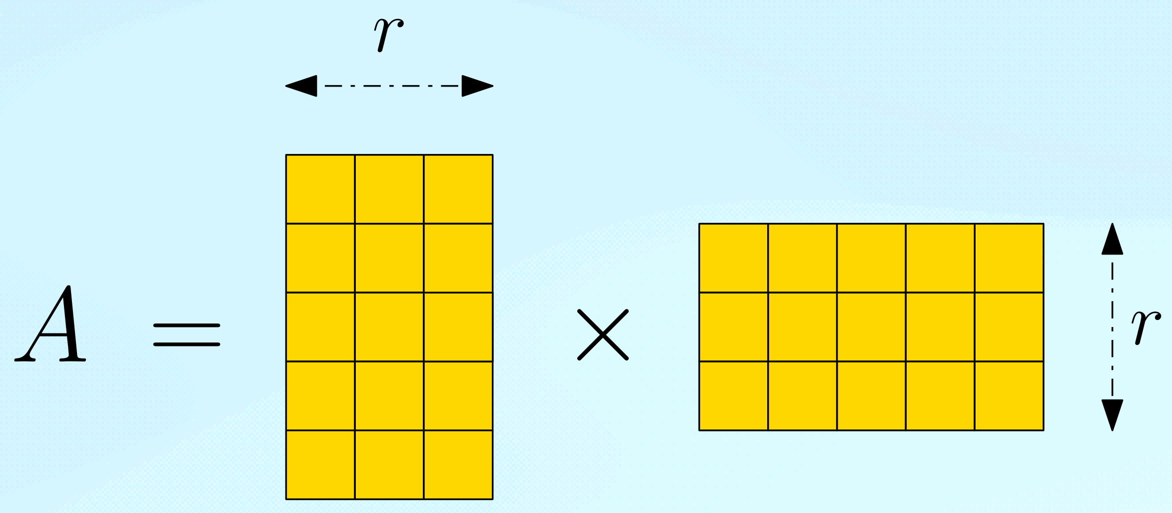
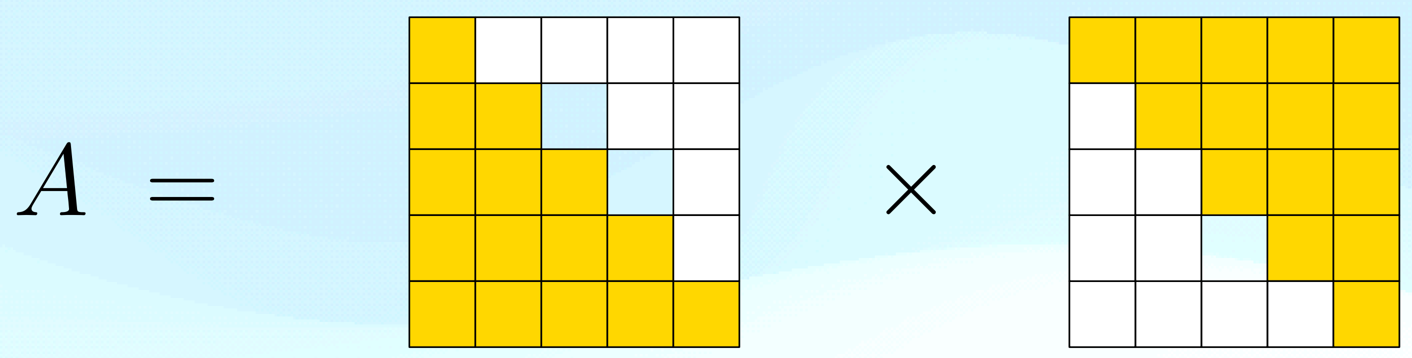
Butterfly matrix/factorization

Known results on (FSMF)

- For arbitrary (I, J), (FSMF) is *NP-hard* to solve. — NP-hardness
- There are instances (A, I, J) where (FSMF) has no optimal solution. — Ill-posedness
- For certain structured (I, J), (FSMF) has a polynomial algorithm. — Tractability
- With the same family of structured (I, J), loss function of (FSMF) has no local minima. — Benign landscape

(Q-T. Le, E. Riccietti, R. Gribonval, SIAM Journal of Matrix Analysis and Applications, 2023)

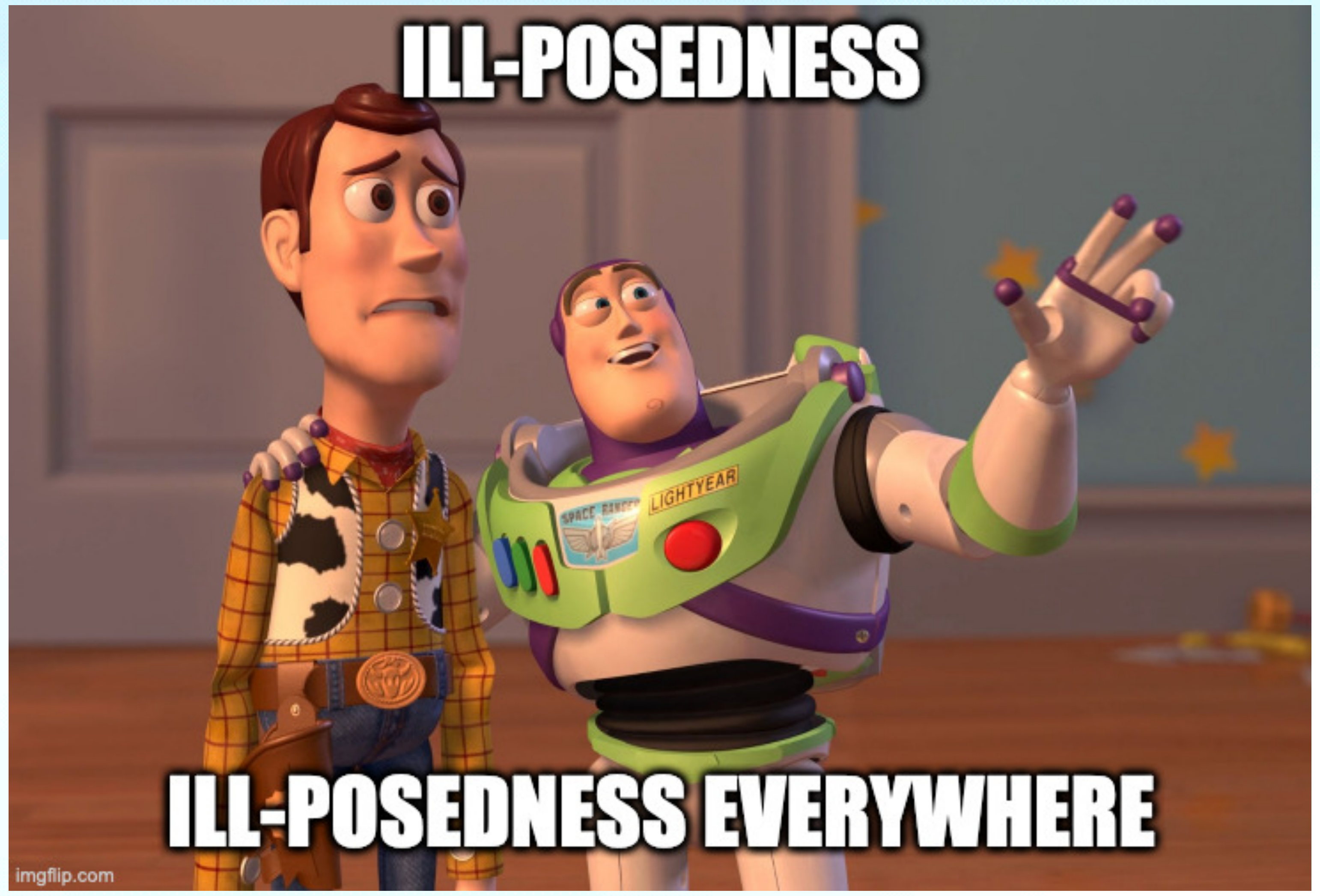
Existence of optimal solutions of FSMF

Low rank matrix approximation	LU decomposition
	
• Approximate a matrix A by a rank r matrix. • Optimal solutions are given by computing the truncated Singular Value Decomposition for ***any*** matrix A.	• $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, I = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, J = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ • Infimum is zero. • A cannot be factorised into lower and upper-triangular matrices.

WELL-POSED

ILL-POSED

Similar phenomenon



Tensor decomposition (order at least three)	TENSOR RANK AND THE ILL-POSEDNESS OF THE BEST LOW-RANK APPROXIMATION PROBLEM VIN DE SILVA* AND LEK-HENG LIM†
Matrix Completion	Low-Rank Matrix Approximation with Weights or Missing Data is NP-hard Nicolas Gillis[1] and François Glineur[1]
Robust Principle Component Analysis	Matrix rigidity and the ill-posedness of Robust PCA and matrix completion* Jared Tanner†‡ Andrew Thompson§ Simon Vary†
(Classical) Neural Network Training	**Best *k*-Layer Neural Network Approximations** **Lek-Heng Lim[1] · Mateusz Michałek[2,3] · Yang Qi[4]**

Existence of optimal solutions of FSMF (cont)

Given support constraints (I, J), is there a matrix A that makes (FSMF) have no optimal solution?

Given support constraints (I, J), is there a data set $\mathscr{D}$ that makes the training sparse ReLU NNs have no optimal solutions?

$$\min_{W^{(j)}, b^{(j)}} \quad \|Y - W^{(L)}\sigma(\ldots\sigma(W^{(1)}X + b^{(1)}) + \ldots) + b^{(L)}\|_F^2$$

subject to: $W^{(j)} \in \mathscr{E}_j, \forall j \in \{1, \ldots, L\}$

$\mathscr{E}_j$: set of matrices whose **support** are *fixed*.

Similar assumption to (FSMF)

Origin of ill-posedness

Reformulation of (FSMF)

ORIGINAL FORMULATION

$$\min_{X,Y} \|A - XY^\top\|_F^2 \text{ subject to: } \operatorname{supp}(X) \subseteq I, \operatorname{supp}(Y) \subseteq J$$

Change of variables

NEW FORMULATION

$$\min_{B \in \mathcal{E}_{I,J}} \|A - B\|_F^2 \text{ where } \mathcal{E}_{I,J} := \{XY^\top \mid \operatorname{supp}(X) \subseteq I, \operatorname{supp}(Y) \subseteq J\}$$

PROJECTION A ONTO THE SET $\mathcal{E}_{I,J}$

Equivalence: closedness - well-posedness

A NECESSARY AND SUFFICIENT CONDITION

THEOREM

(I, J) is well-posed if and only if $\mathscr{E}_{I,J}$ is a closed set in the usual topology of $\mathbb{R}^{m \times n}$

REMINDER:

A set X is closed if the limit of any convergent sequence of elements of X is an element of X.

Equivalence: closedness - well-posedness

PROOF

$\Rightarrow$ If (I, J) is well-posed:

By contradiction, assume that $\mathcal{E}_{I,J}$ is not closed.

By definition, there exists $A \notin \mathcal{E}_{I,J}$ such that there is a sequence $\{B_n\}_{n\in\mathbb{N}}, B_n \in \mathcal{E}_{I,J}$ s.t.:

$$\lim_{n\to\infty} B_n = A.$$

Consider the (FSMF) with (A, I, J):

- The infimum is zero (take the sequence $\{B_n\}_{n\in\mathbb{N}}$)
- The infimum is not attained ($A \notin \mathcal{E}_{I,J}$)

Equivalence: closedness - well-posedness

PROOF (CONT)

$\Rightarrow$ If $\mathscr{E}_{I,J}$ is closed:

$\Rightarrow$ Since $0 \in \mathscr{E}_{I,J}$ is closed, for *any* instance of (FSMF) with (A, I, J), the infimum is at most $C = \|A\|_F^2$.

$$\min_B \|A - B\|_F^2 \text{ where } B \in \mathscr{E}_{I,J} \cap \mathbf{B}(A, \|A\|_F)$$

Ball centered at A and radius $\|A\|_F$

Important trick: $\mathscr{E}_{I,J} \cap \mathbf{B}(A, \|A\|_F)$ is compact (bounded and closed).

$\|A - \cdot\|_F^2$ is a continuous function.

Conclusion

Given a support constraint (I, J), decide whether (I, J) is **well-posed**.

⬇ ⬆

Given a support constraint (I, J), decide whether $\mathscr{E}_{I,J}$ is **closed**.

An algorithm to decide the closedness of $\mathscr{E}_{I,J}$

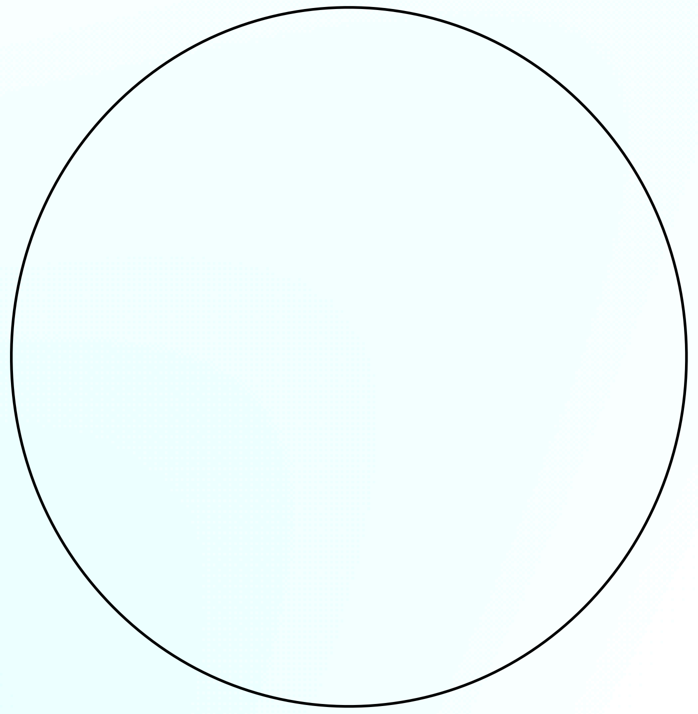
Real algebraic geometry and its algorithm

SEMI-ALGEBRAIC SET

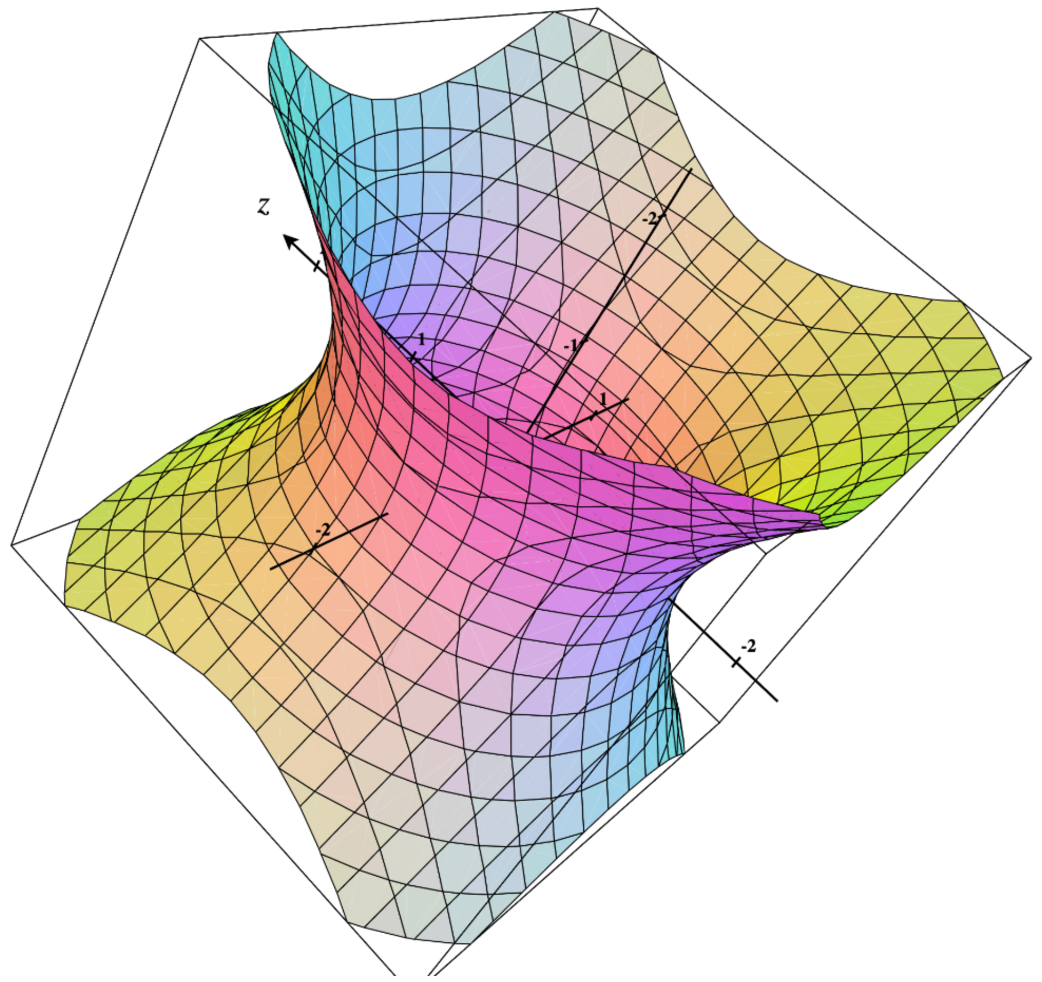
$$\bigcup_{i \in \mathcal{I}} \{x \in \mathbb{R}^n \mid P_i(x) = 0 \wedge \bigwedge_{j=1}^{\ell} Q_{i,j}(x) > 0\}, \mathcal{I} \text{ is finite}$$

where $P_i, Q_{i,j}$ are **polynomials**

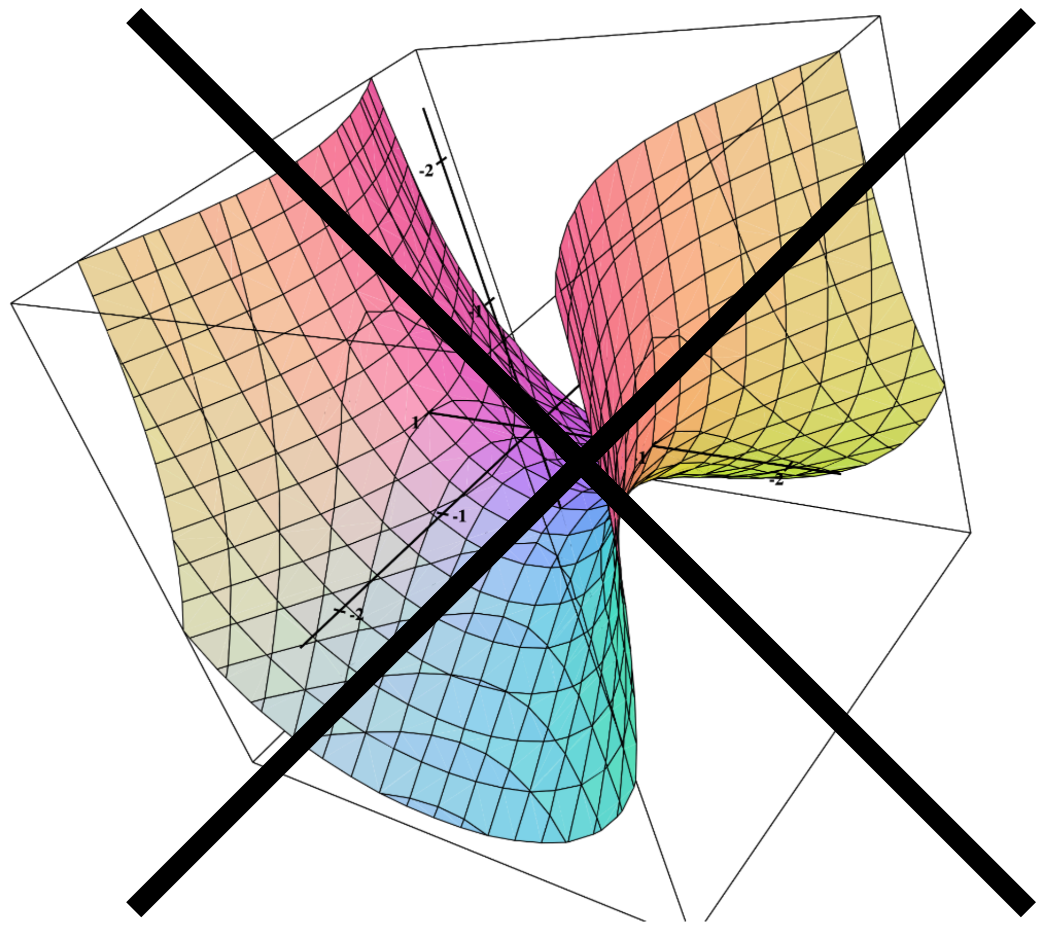
EXAMPLE:



$\{(x, y) \mid x^2 + y^2 = 1\}$



$\{(x, y, z) \mid x^2 - y^2 + z^2 = 2\}$



$\{(x, y, z) \mid x^2 - y^2 + e^z = 2\}$

$\mathscr{E}_{I,J}$ is a semi-algebraic set

THEOREM

For any (I, J), $\mathscr{E}_{I,J}$ is a semi-algebraic set

REMINDER: $\mathscr{E}_{I,J} := \{XY^\top \mid \text{supp}(X) \subseteq I, \text{supp}(Y) \subseteq J\}$

How to find the set of polynomials describing $\mathscr{E}_{I,J}$?

PROJECTION THEOREM

Let X be semi-algebraic, $Y = \{y \mid \exists x, (x, y) \in X\}$ is also semi-algebraic

$\mathscr{E}_{I,J}$ is a semi-algebraic set (cont)

PROJECTION THEOREM

Let X be semi-algebraic, $Y = \{y \mid \exists x, (x, y) \in X\}$ is also semi-algebraic.

PROOF (THAT $\mathscr{E}_{I,J}$ IS SEMI-ALGEBRAIC):

Consider $\mathscr{A} := \{(A, X, Y) \mid \|A - XY^\top\|_F^2 = 0 \wedge \text{supp}(X) \subseteq I \wedge \text{supp}(Y) \subseteq J\}$.

polynomial $X_{i,j} = 0, \forall (i,j) \notin I$ $Y_{i,j} = 0, \forall (i,j) \notin J$

Therefore, $\mathscr{A}$ is semi-algebraic.

To conclude, projection of $\mathscr{A}$ to the first term is $\mathscr{E}_{I,J}$ (because $\|A - XY^\top\|_F^2 \Rightarrow A = XY^\top$)

→ **Therefore, we can use tools from real algebraic geometry to decide the closedness of $\mathscr{E}_{I,J}$**

Deciding the closedness of $\mathscr{E}_{I,J}$

$\mathscr{E}_{I,J}$ is a closed set if and only if $\overline{\mathscr{E}_{I,J}} \setminus \mathscr{E}_{I,J}$ is empty

REMINDER: Given a set $\mathscr{A}$, $\overline{\mathscr{A}}$ is the set of limits of sequence of $\mathscr{A}$.

$$\overline{\mathscr{E}_{I,J}} \setminus \mathscr{E}_{I,J} = \{A \mid \forall X, \forall Y, \text{supp}(X) \subseteq I \wedge \text{supp}(Y) \subseteq J \wedge \|A - XY^\top\|^2 > 0\} \quad \mathscr{E}^C_{I,J}$$

$$\bigcap \{A \mid \forall \epsilon > 0, \exists X, \exists Y, \text{supp}(X) \subseteq I \wedge \text{supp}(Y) \subseteq J \wedge \|A - XY^\top\|^2 < \epsilon\} \quad \overline{\mathscr{E}_{I,J}}$$

→ **Using (generalised) projection theorem, $\mathscr{E}^C_{I,J}$, $\overline{\mathscr{E}_{I,J}}$, $\overline{\mathscr{E}_{I,J}} \setminus \mathscr{E}_{I,J}$ are semi-algebraic sets**

Deciding the closedness of $\mathscr{E}_{I,J}$

$$\overline{\mathscr{E}_{I,J}} \backslash \mathscr{E}_{I,J} = \{A \mid \forall X, \forall Y, \mathrm{supp}(X) \subseteq I \wedge \mathrm{supp}(Y) \subseteq J \wedge \|A - XY^{\top}\|^2 > 0\}$$
$$\bigcap \{A \mid \forall \epsilon > 0, \exists X, \exists Y, \mathrm{supp}(X) \subseteq I \wedge \mathrm{supp}(Y) \subseteq J \wedge \|A - XY^{\top}\|^2 < \epsilon\}$$

- Using *quantifier elimination algorithm*, we can decide the emptiness of the semi-algebraic set $\overline{\mathscr{E}_{I,J}} \backslash \mathscr{E}_{I,J}$.

(S. Basu, R. Pollack, M-F Roy, Algorithms in Real Algebraic Geometry)

- The complexity of the algorithm is $O\left(4^{C^k}\right)$, where:
  - C is a universal constant.
  - $k = mn + 2(|I_1| + |I_2|) + 1$

Size of the matrix product

Size of the supports

Recap of the algorithm

(I, J) is well-posed?

→

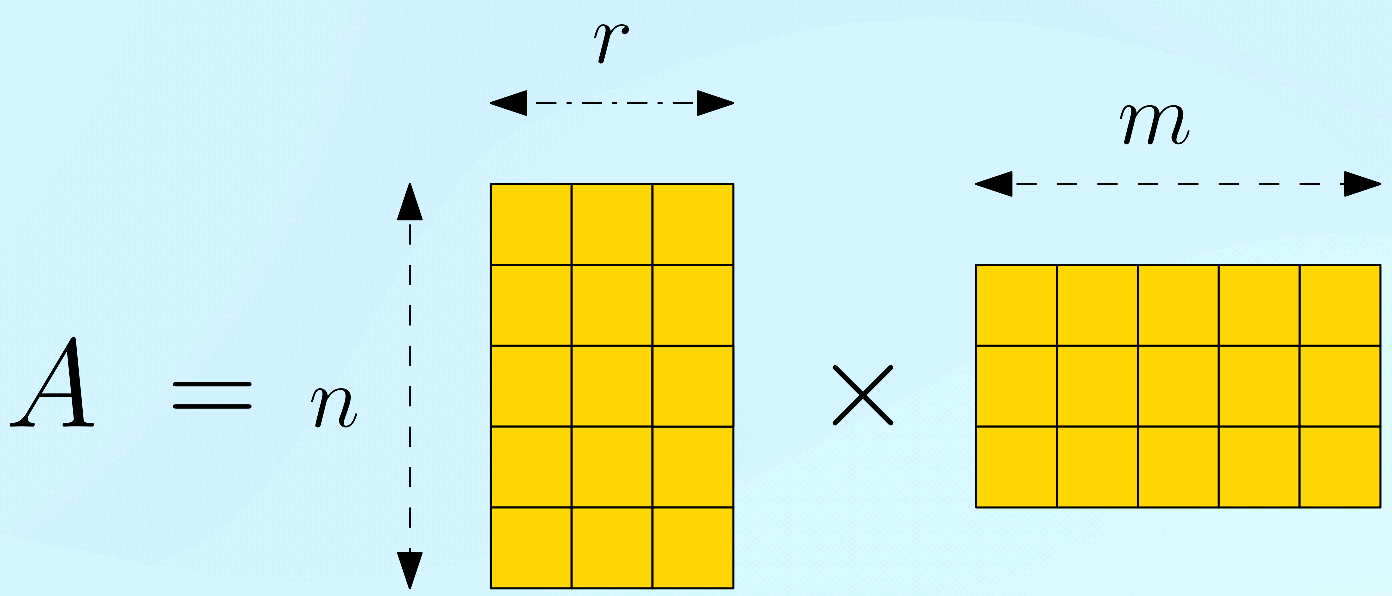
$\mathscr{E}_{I,J}$ is closed?

→

$\overline{\mathscr{E}_{I,J}} \setminus \mathscr{E}_{I,J}$ is empty?

How does the algorithm work in practice?

Low rank approximation



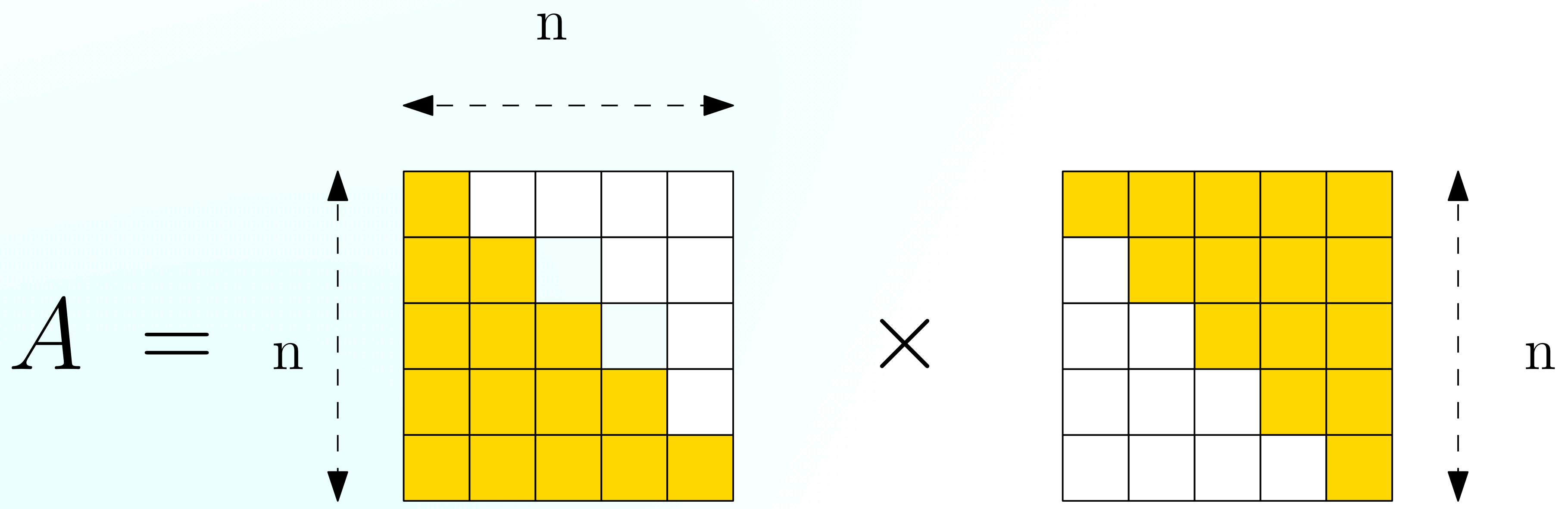
✔ $m = n = r = 2$

```
quantifiersElimination — -zsh — 80×24
(base) tung@dhcp-67-169 quantifiersElimination % python fullsupport.py
Running time: 0.0036940574645996094
True
```

✘ $m = n = 3, r = 2$

```
quantifiersElimination — -zsh — 80×24
(base) tung@dhcp-67-169 quantifiersElimination % python fullsupport.py
^CRunning time: 2112.0239312648773
None
```

LU decomposition



✔ $n = 2$

```
quantifiersElimination — -zsh — 80×24
(base) tung@dhcp-67-169 quantifiersElimination % python LU2x2.py
Running time: 0.013816118240356445
False
```

✘ $n = 3$

```
quantifiersElimination — -zsh — 80×24
(base) tung@dhcp-67-169 quantifiersElimination % python LU3x3.py
^CRunning time: 3202.279525756836
None
```

Perspectives

- ✓ Given support constraint (I, J), its well-posedness is ***decidable***.
- ✓ The algorithm generalises easily to multi-factors ($L > 2$).

But,

- ✗ The complexity for the algorithm is doubly exponential.
- ✗ Using quantifier elimination algorithm (a general algorithm) does not provide any insight properties of $\mathscr{E}_{I,J}$.

Well-posedness of sparse ReLU neural networks

Fixed support sparse ReLU neural networks

Given data set $\mathscr{D} := (X, Y)$, solve:

GENERAL

$$\min_{W^{(j)}, b^{(j)}} \quad \|Y - W^{(L)}\sigma(\ldots\sigma(W^{(1)}X + b^{(1)}) + \ldots) + b^{(L)}\|_F^2$$

subject to: $W^{(j)} \in \mathscr{E}_j, \forall j \in \{1,\ldots,L\}$

FIXED SUPPORT

$$\min_{W^{(j)}, b^{(j)}} \quad \|Y - W^{(L)}\sigma(\ldots\sigma(W^{(1)}X + b^{(1)}) + \ldots) + b^{(L)}\|_F^2$$

subject to: $\text{supp}(W^{(j)}) \in I_j, \forall j \in \{1,\ldots,L\}$

DÉJÀ VU: closedness vs well-posedness

Given a support constraint $(I_1, \ldots, I_L)$, is the training problem well-posed (i.e., **for all data set** $\mathscr{D}$, optimal solutions always exist)?

The support constraint $(I_1, \ldots, I_L)$ make training problem **well-posed** if and only if for all input sets X, the image $W^{(L)}\sigma(\ldots\sigma(W^{(1)}X + b^{(1)}) + \ldots) + b^{(L)}$ is **closed**.

Sufficient condition for well-posedness

THEOREM

For **two-layer** neural networks ($L = 2$) with **output dimension** equal to **one**, any support constraint makes the training problem **well-posed**.

COROLLARY

For **two-layer** neural networks ($L = 2$) with **output dimension** equal to **one**, constraints $\mathcal{E}_j := \{X \mid \|X\|_0 \leq k_j\}, j = 1,2$ makes the training problem well-posed.

Necessary condition for well-posedness

THEOREM

For **two-layer** neural networks ($L = 2$) with support constraint (I, J), the **well-posedness** of training problem implies the **closedness** of $\mathscr{E}_{I,J}$.

this is decidable

THEOREM

For fixed support neural networks with support constraint $(I_1, \ldots, I_L)$, the **well-posedness** of training problem implies the **closedness** of $\mathscr{E}_{I_1,\ldots,I_L}$.

Necessary condition for well-posedness

THEOREM

For **two-layer** neural networks ($L = 2$) with support constraint (I, J), the **well-posedness** of training problem implies the **closedness** of $\mathscr{E}_{I,J}$.

The condition is just necessary because when there is **no constraint** on the support, the training problem is ill-posed for certain data set.

(L-H. Lim, M. Michalek, Y. Qi, Constructive Approximation 2019)

Contribution and future works

TAKE AWAY MESSAGE

- Ill-posedness of (FSMF) is decidable, not yet tractable.
- Link between sparse matrix factorization and sparse ReLU neural networks.

POSSIBLE IMPROVEMENT?

- Better algorithms to decide the ill-posedness of (FSMF)
- When the problem is well-posed, is there polynomial algorithm for (FSMF)
- A full characterization of ill-posedness of sparse ReLU neural networks

https://arxiv.org/abs/2306.02666

THANK YOU