Training Fully Connected Neural Networks is ∃ℝ-Complete

Daniel Bertschinger, ETH Zurich
Christoph Hertrich, LSE London → ULB Brussels → GU Frankfurt
Paul Jungeblut, Karlsruhe Institute of Technology
Tillmann Miltzow, Utrecht University
Simon Weber, ETH Zurich

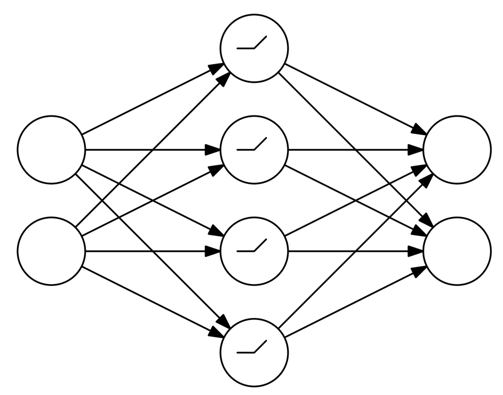
NEURAL INFORMATION PROCESSING SYSTEMS

Problem & Contribution

We determine the exact computational complexity of *empirical risk minimization*, i.e., the training problem for neural networks.

Training Problem:

Input:
- network architecture N: two layers and fully connected
- data points D
- target error γ



Question: Are there weights and biases such that the training error is at most γ?

Theorem: Training two-layer fully connected neural networks is ∃ℝ-complete. This holds even if:
- There are only two input neurons.
- There are only two output neurons.
- The number of data points is linear in the number of hidden neurons.
- The data has only 13 different labels.
- The target error is $\gamma = 0$.
- The ReLU activation function is used.

Theorem: Irrational numbers of arbitrary algebraic degree are required to train some two-layer fully connected to optimality.

∃ℝ: Existential Theory of the Reals

Definition: The complexity class ∃ℝ contains all problems that polynomial-time many-one reduce to ETR, i.e., deciding an existential first-order fomula of the form

∃ℝ

$$\exists X_1, \ldots, X_n \in \mathbb{R} : \varphi(X_1, \ldots, X_n).$$

⇢ polynomial equations/inequalities

P ⊆ NP ⊆ ∃ℝ ⊆ PSPACE

Intuition: ∃ℝ is a real analog of NP:
- SAT: existence of Boolean variables
- ETR: existence of real-valued variables

An ∃ℝ-Complete Problem: ETR-INV

Input: A formula $\Phi \equiv \exists X_1, \ldots, X_n : \varphi(X_1, \ldots, X_n)$ where φ is a conjunction (only $\wedge$) of constraints, each of the form $X_i + X_j = X_k$ or $X_i \cdot X_j = 1$.

Question: Is Φ true?

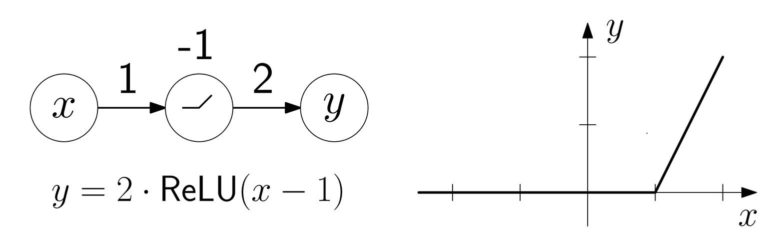
Promise: Φ is either false, or it has a solution with all $X_i \in [\frac{1}{2}, 2]$.

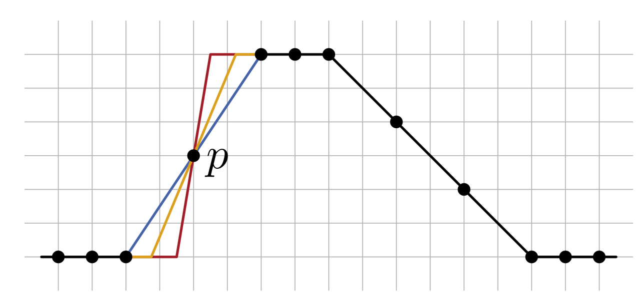
Proof Sketch: Reduction from ETR-INV

Goal: Given an ETR-INV instance, construct an equivalent instance of the neural network training problem in polynomial time.

Idea: A single ReLU neuron computes a continuous piecewise linear function with one flat part and one sloped part.

more ReLUs ⇝ more bends



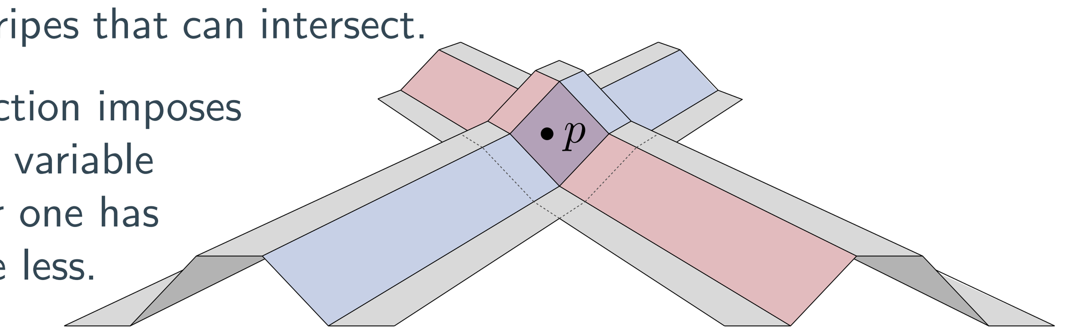


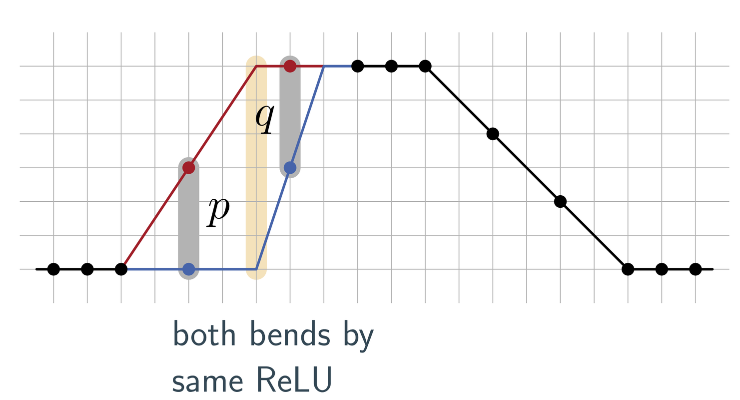
Variable Gadget: 12 data points that must be fit with 4 ReLU neurons.

Unique, except for the segment through p. Its slope represents the value of a variable.

Linear Dependencies: In two input dimensions, variable gadgets become stripes that can intersect.

A data point in the intersection imposes a linear dependency: If one variable contributes more, the other one has to proportionally contribute less.
⇝ Copying & Addition

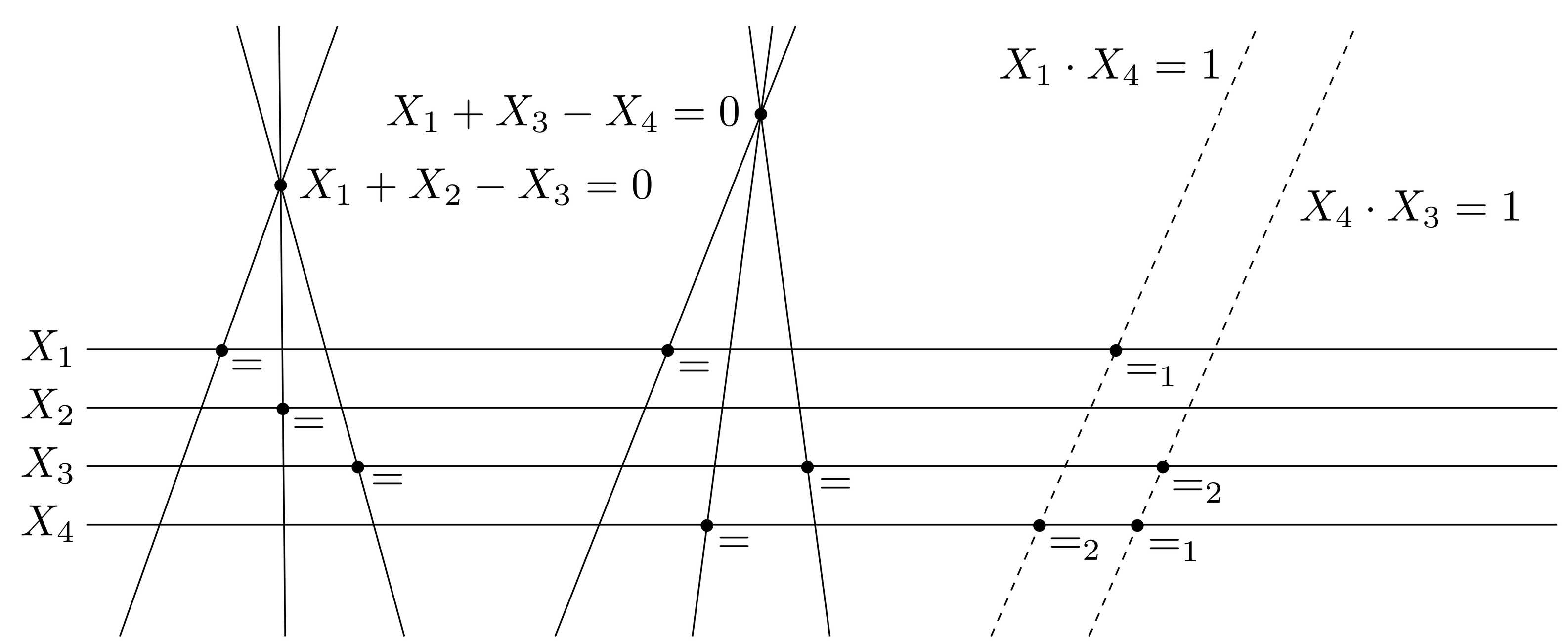




Inversion Gadget: 13 data points that must be fit with 5 ReLU neurons.: Data points p and q have different labels in the two output dimensions.

Think of a variable gadget with two slopes representing two real values that depend on each other non-linearly. ⇝ Inversion

Global Arrangement of the Gadgets:



Discussion

Advancement of the State of the Art:

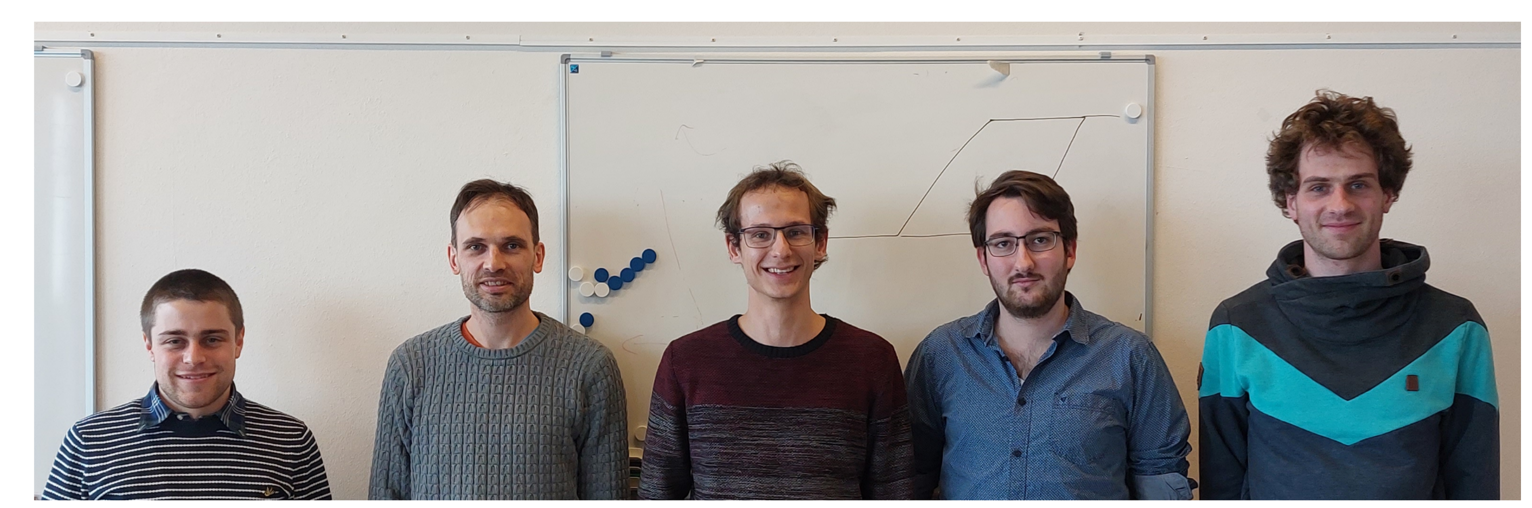
- Arora, Basu, Mianjy and Mukherjee (ICLR 2018) prove NP-membership for the single output case. Our result explains, why this was never generalized to more than one output dimension.
- Abrahamsam, Kleist and Miltzow (NeurIPS 2021) prove ∃ℝ-hardness for adversarial network architectures. We strenghten their result by proving that ∃ℝ-hardess is inherent to the problem itself.

Implications:

- It is widely believed that NP ⊊ ∃ℝ
  - ⇒ NN training is more difficult than NP-complete problems
  - ⇒ Tools like mixed-integer programming or SAT-solving not sufficient.
- Our results do not rule out good heuristics. In fact, stochastic gradient descent seems to be an effective tool for other ∃ℝ-complete problems, too.

Limitations and Open Questions:

- ∃ℝ-completeness heavily relies on precision. Is NN training in NP if we allow small additive errors?
- Which other extra-assumptions make training tractable?
- We consider only the *training* error. Any implications on *generalization*?
- Can we transfer our results to deeper architectures?



Daniel Till Christoph Simon Paul

References

Mikkel Abrahamsen, Linda Kleist, and Tillmann Miltzow (2021). "Training Neural Networks is ∃ℝ-complete" In: Advances in Neural Information Processing Systems 34

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee (2018). "Understanding Deep Neural Networks with Rectified Linear Units" In: International Conference on Learning Representations

Related Poster at NeurIPS 2023: V. Froese and C. Hertrich: Training Neural Networks is NP-Hard in Fixed Dimension