COMPUTATIONAL BARRIERS FOR LEARNING SOME GENERALIZED LINEAR MODELS

Surbhi Goel
Microsoft Research NY

Rigorous Evidence for Information-Computation Trade-offs

GENERALIZED LINEAR MODELS (GLM)

Generalized linear model is a class of functions

$$\sigma_w : x \to \sigma(w \cdot x)$$

parameterized by an unknown weight vector $w \in \mathbb{R}^d$ and link function σ which is assumed to be a known 1-Lipschitz monotonic function.

$$\sigma(a) = \max(0, a)$$

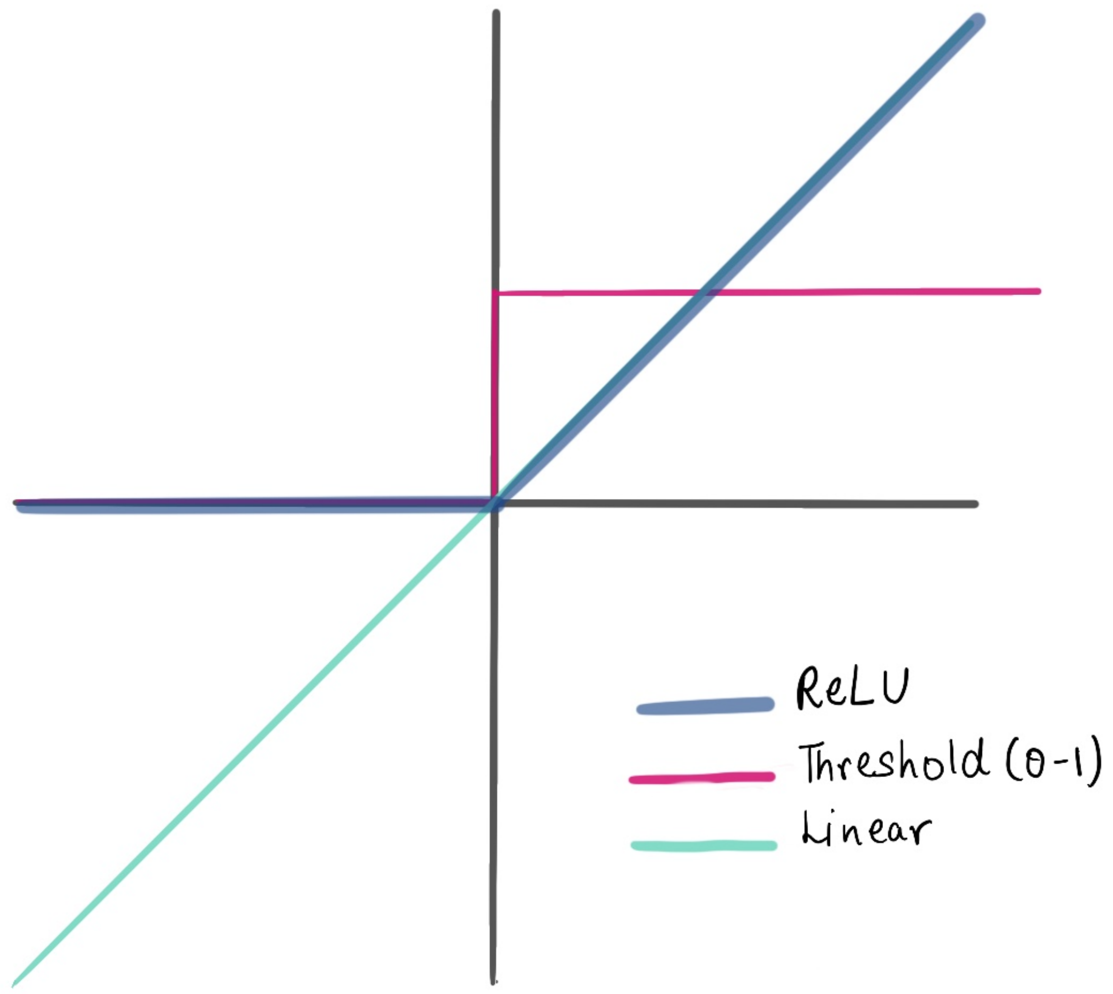
ReLU

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

Sigmoid

RECTIFIED LINEAR UNIT (RELU)

$$\sigma(a) = \max(0,a)$$

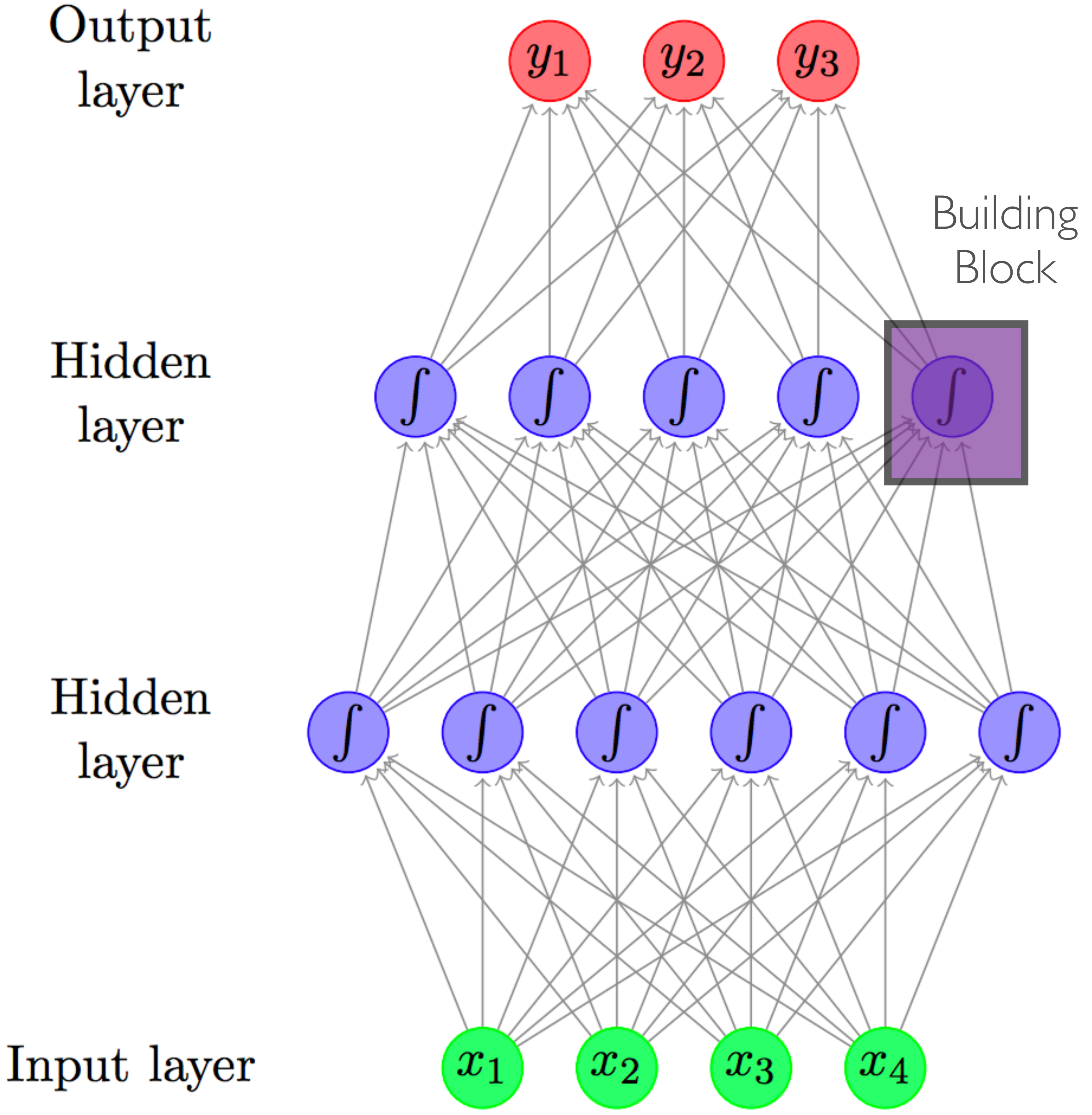


$$\text{ReLU}_w : x \to \max(0,\ w \cdot x)$$

Lies in-between a halfspace and a linear function

Challenging to learn

Easy to learn



Source: Yoav

LEARNING PROBLEM

INPUT

Access to samples $\mathcal{S}$ or queries from distribution $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{R}$

→ **Algorithm** →

OUTPUT

Hypothesis h which minimizes loss

$$\text{loss}(h) \leq \boxed{\min_w \text{loss}(\text{ReLU}_w)} + \epsilon$$

Best possible loss

Square loss: $\textbf{loss}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\frac{1}{2}\left(h(x) - y\right)^2\right]$

If h is of the form $\sigma_{\hat{w}}$ for some $\hat{w}$ then we call the algorithm a proper learner else improper

INFORMATION THEORETICALLY

- Complexity of the model can be bounded as long as the weights are bounded (Rademacher, parameter counting, . . .)

- Sample complexity is polynomial in all parameters

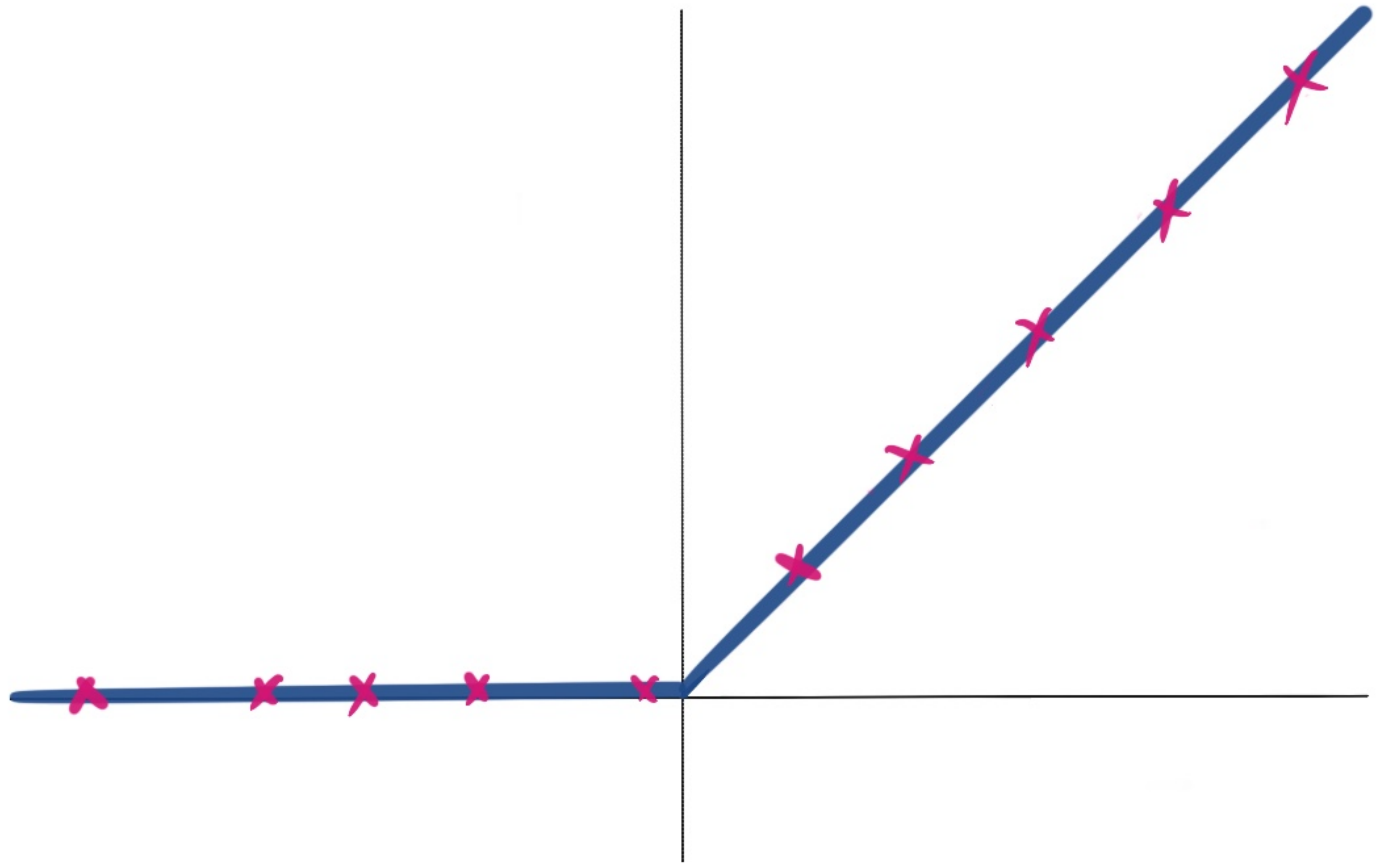
- Brute-force over the parameter space

Can we do this in a computationally efficient manner?

NOISE MODELS

Realizable (no noise)

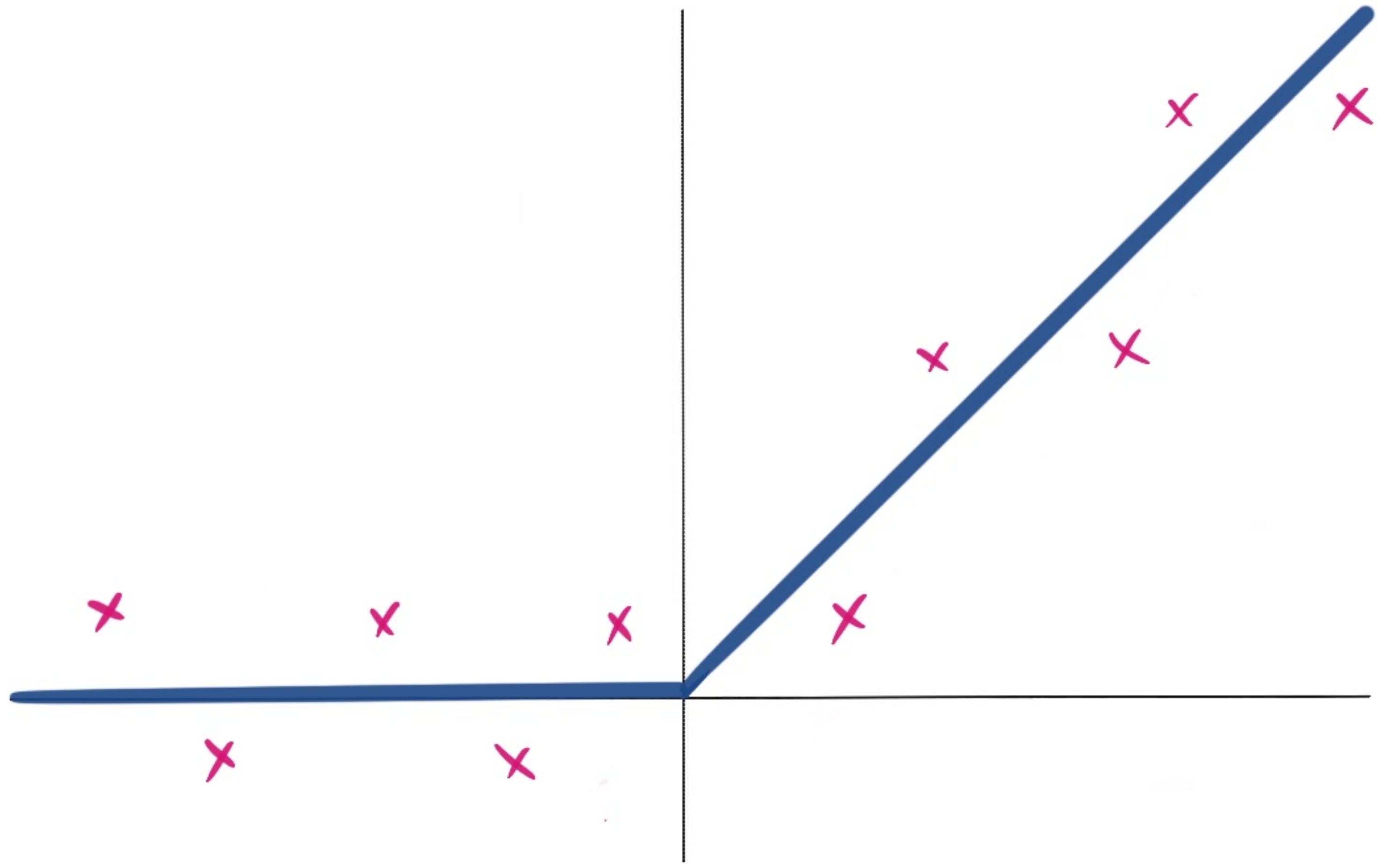
$y = \mathrm{ReLU}_w(x)$ for some $w \in \mathbb{R}^d$



Solvable in poly time via linear programming and by GD under additional assumptions *[Soltanolkotabi'17; Yehudai-Shamir'20]*

P-Concept (mean-0 noise)

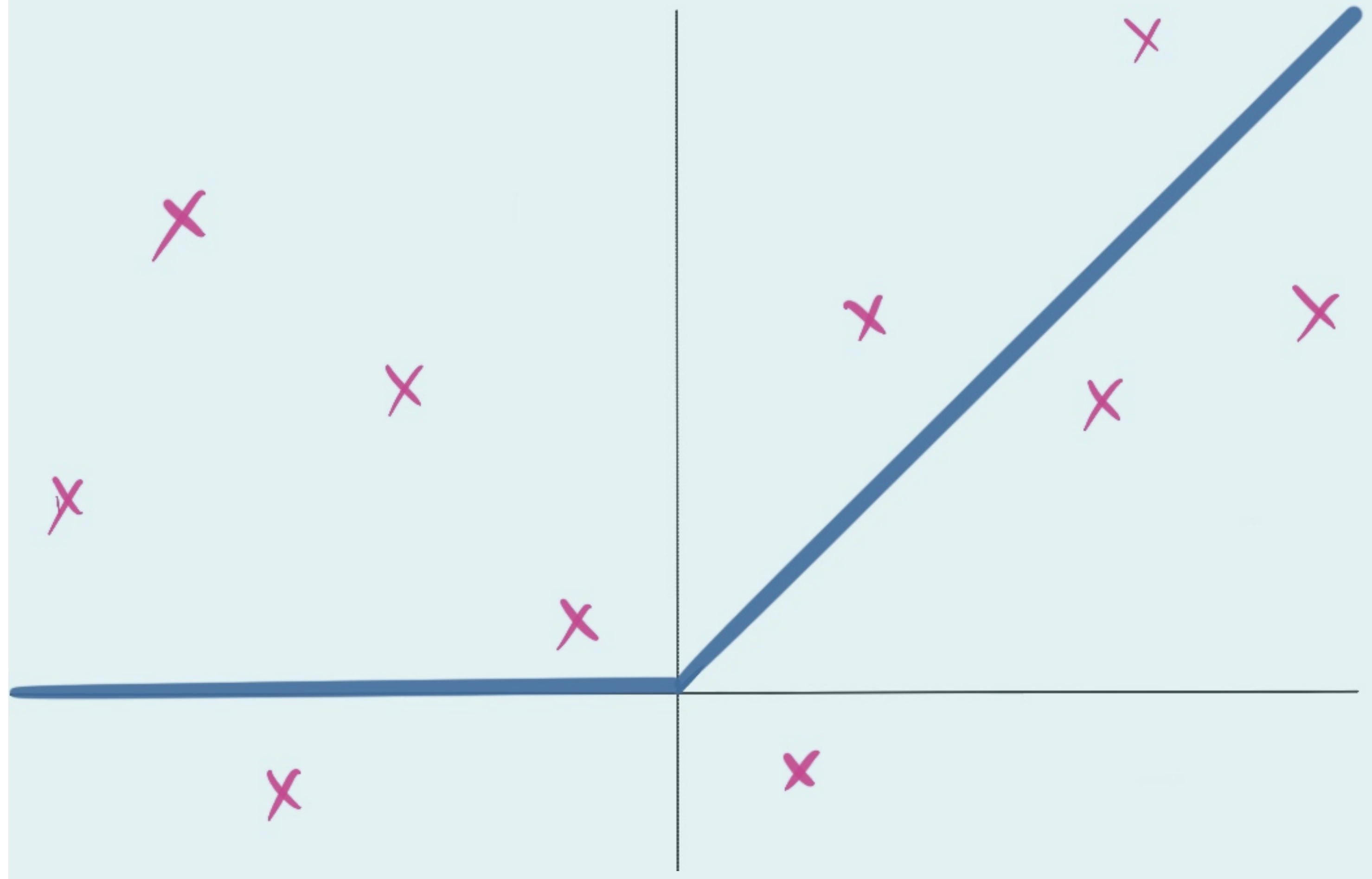
$\mathbb{E}[y \mid x] = \mathrm{ReLU}_w(x)$ for some $w \in \mathbb{R}^d$



Solvable in poly time using a convex surrogate *[Kalai-Sastry'08; Kakade-Kalai-Kanade-Shamir'11]*

Agnostic (arbitrary noise)

y has no restrictions



Not solvable in poly-time

THIS TALK

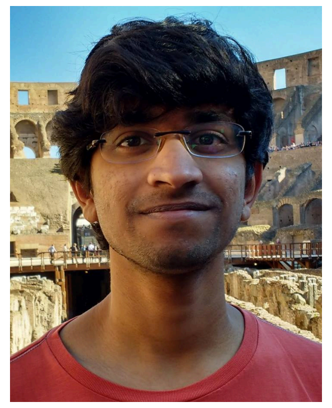
What is the precise computational complexity for this problem?

Part 1: Conditional Hardness even with Bounded Weights and Inputs

Joint work with Adam Klivans, Pasin Manurangsi and Daniel Reichman

Part 2: Unconditional Hardness even under Gaussian Marginals

Joint work with Aravind Gollakota and Adam Klivans



PART 1:

CONDITIONAL HARDNESS OVER UNIT BALL

RELU REGRESSION ERM

- **Input:** Set S of samples $(x, y) \in$ $B(d,1) \times [0,1]$
- **Goal:** Find $w \in$ $B(d,1)$ minimizes $\mathbb{E}_{(x,y)\sim S}\left[\frac{1}{2}\left(\mathrm{ReLU}_w(x) - y\right)^2\right]$ Proper learning

Training problem is equivalent to learning problem if we choose $\mathscr{D}$ to be uniform on S

We remove the scale in the problem by restricting x, w to have norm ≤ 1

MAIN RESULT

Goel-Klivans-Manurangsi-Reichman'21

Under a certain Exponential Time Hypothesis (ETH), there is no $2^{o(1/\epsilon^2)}$poly(d) time algorithm for proper ReLU regression up to additive error ϵ.

Bounds poly in d due to norm bound on input and weight

A simple algorithm that iterates over all possible sign-patterns for polynomially many samples matches the lower bound (approach by *[Arora-Basu-Mianjy-Mukherjee'18]*)

Our result gives a separation between *proper* and *improper* learning for ReLU regression!

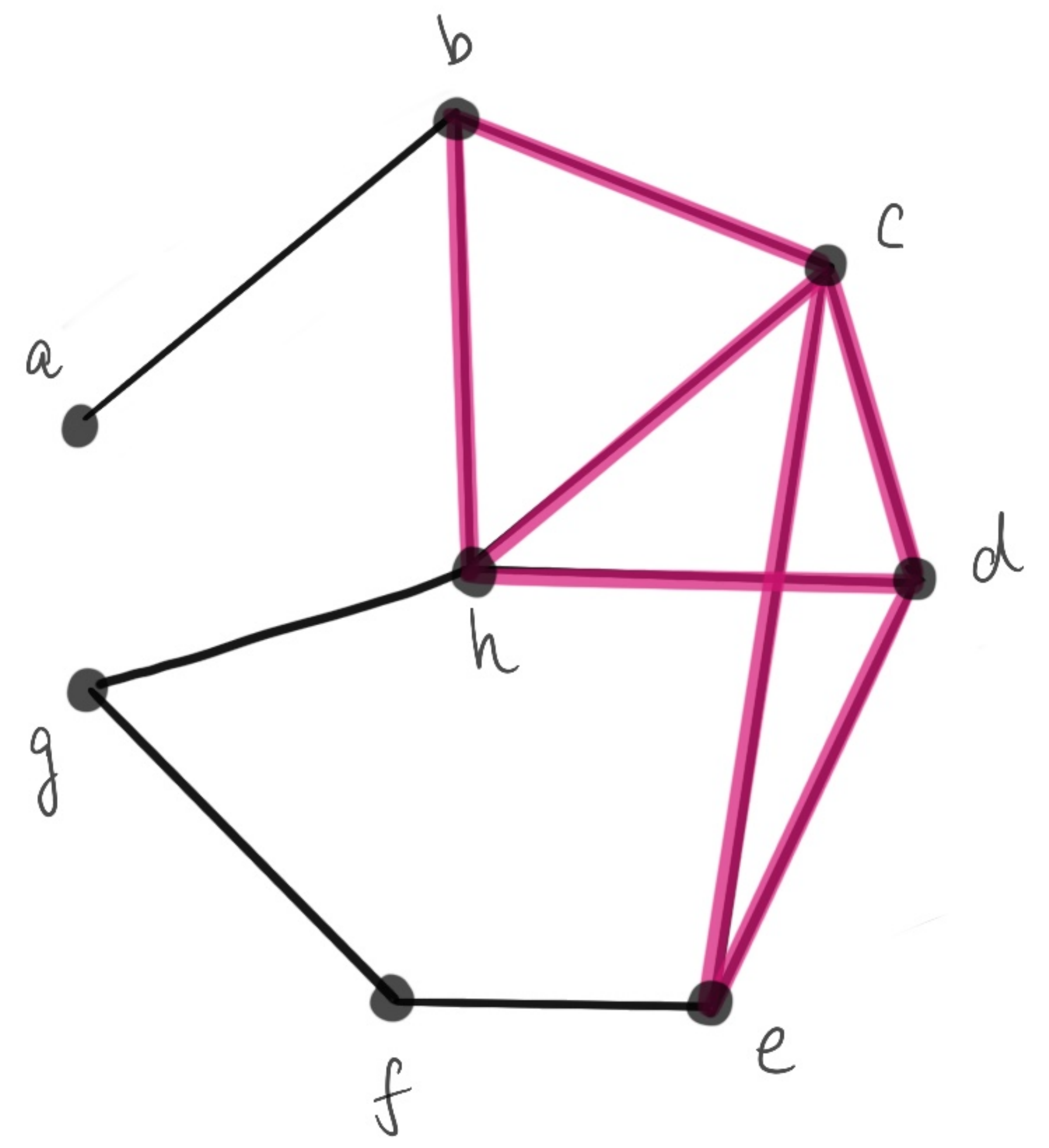
Best-known *improper* algorithm runs in time $2^{O(1/\epsilon)}$poly(d) *[Goel-Kanade-Klivans-Thaler'17]*

HYPOTHESIS

Densest κ-Subgraph (DκS)

Input: Graph G of size $n, \kappa \in \mathbb{N}$

Goal: Find κ-vertex subgraph with max number of edges



Gap-ETH for Densest κ-Subgraph

Goel-Klivans-Manurangsi-Reichman'21

There is no $2^{o(n)}$-time algorithm that can approximate Densest κ-Subgraph within a constant factor.

$o(n)$-level of the Sum-of-Squares Hierarchies do not give constant factor approximation for DκS even for bounded degree graphs *[Alon-Arora-Manokaran-Moshkovitz-Weinstein'11; Manurangsi'17]*

MAIN CHALLENGE

- Can approximate the optimal ReLU up to δ in $2^{O(1/\delta^2)}\text{poly}(d)$ time using dimensionality reduction and a δ-net

$\delta = \sqrt{\epsilon}$ implies ϵ additive sq-loss for no noise

- When the intended solution is "almost" correct then we can get a $2^{O(1/\epsilon)}$ algorithm

- Thus we need to construct an instance where the intended solution is also far from the label

Best ReLU

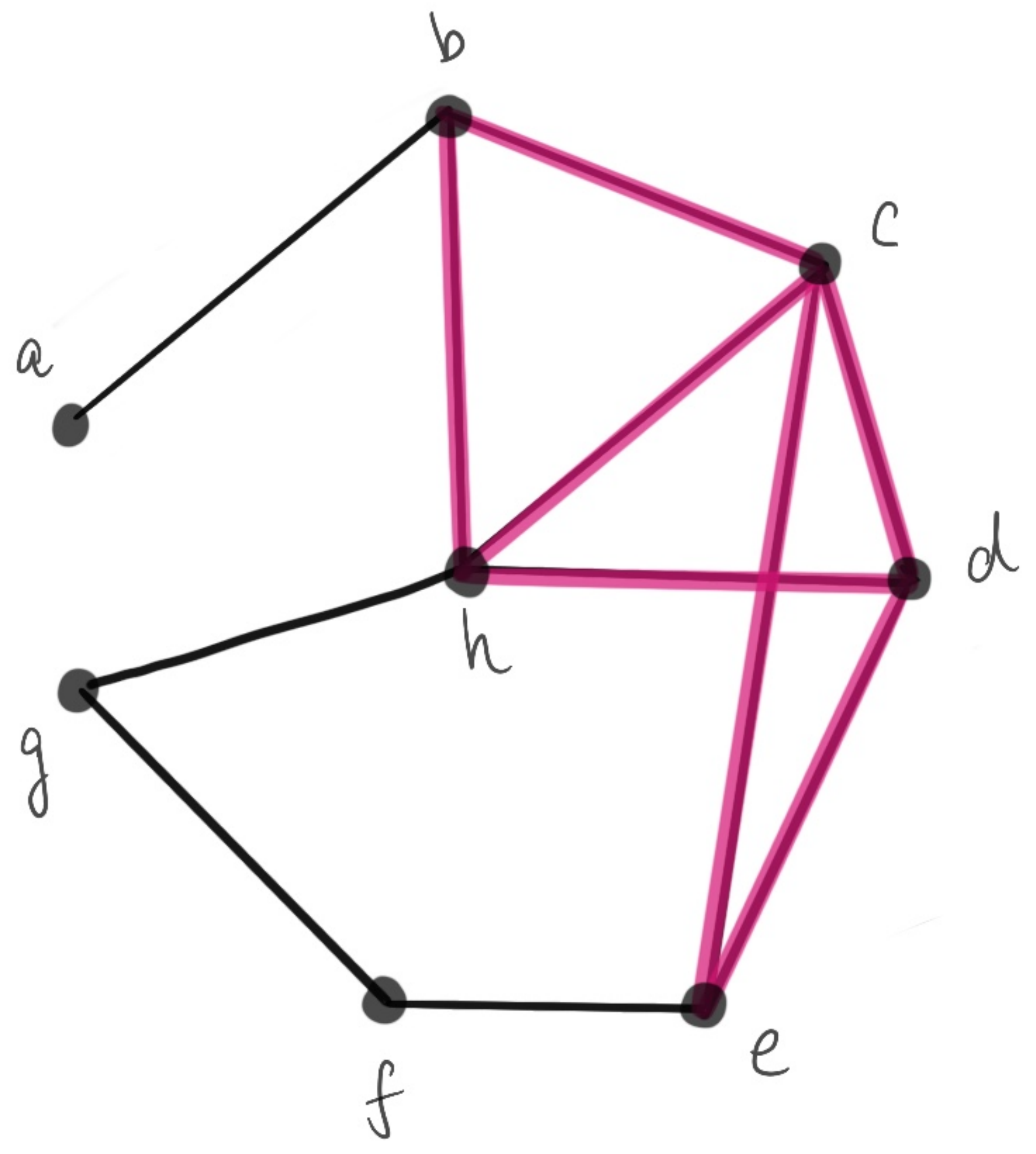
$$(y - y'')^2 - (y - y')^2 \leq \delta^2 + 2\delta\,|\,y' - y\,|$$

True label

δ-approx to best ReLU $|y' - y''| \leq \delta$

If this is $O(1)$ then error is δ not δ^2
So we need $\delta = \epsilon$

REDUCTION



Densest κ-Subgraph (DκS)

Input: Graph G of size $n, \kappa \in \mathbb{N}$

Goal: Find κ-vertex subgraph with max number of edges

Cardinality Constraint

For every vertex $j \in [n]$ create $\sigma(w_j - 0.5/\sqrt{\kappa}) = 1$

Among w of norm 1, loss is minimized when κ coordinates are set to $1/\sqrt{\kappa}$

Edge Constraint

For every edge (i, j) create $\sigma(w_i + w_j - 0.75/\sqrt{\kappa}) = 1$

Error is small when both i and j are selected

Note that error on these constraints is a constant as needed

PART 2:
UNCONDITIONAL HARDNESS OVER GAUSSIAN MARGINALS

GAUSSIAN INPUT SETUP

- Back to the learning problem (not ERM)

Well-behaved distribution

- We further assume input x is distributed according to Gaussian $\mathcal{N}(0, I)$

Common assumption in many works
[Ge-Lee-Ma'18; Du-Zhai-Poczos-Singh'18; Safran-Shamir'18;…Awasthi-Tang-Vijayaraghavan'21]

- Work in the Statistical Query (SQ) computational model
- Our results extend to Sigmoid and Sign activations

Agnostically learning half-spaces

THE STATISTICAL QUERY MODEL *[KEARNS'98]*

Don't see individual samples (x, y), instead make "statistical queries" to an oracle

Oracle

Query

$$\varphi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$$

$$\tau \in [0,1]$$

Tolerance

$$\mathbb{E}[\varphi(x, y)] \pm \tau$$

Learner

Tolerance → sample complexity

queries → runtime complexity

POWER OF SQ MODELS

- SQ model puts a restriction on the computational model
- SQ allows for unconditional (without assumption) computational lower bounds
- Many standard ML algorithms can be implemented as SQ algorithms including moment-based methods, gradient descent etc.
- Parities can't be learned in the SQ model: Gaussian elimination is not SQ

Previous part allowed any algorithm but required hardness assumption in contrast here we restrict the computational model

MAIN RESULT

Goel-Gollakota-Klivans'20

Any SQ algorithm for agnostically learning ReLU needs super-polynomial number of queries or super-polynomial tolerance.

Bounds scale with dimension since input norms are $\approx \sqrt{d}$

For ReLU and Halfspaces: lower bound scales as $d^{\Omega((1/\epsilon)^c)}$ for some constant $1 > c > 0$

Improves on previous $d^{\Omega(\log(1/\epsilon))}$ bound by *[Goel-Karmalkar-Klivans'19]* for ReLU and by *[Klivans-Kothari'14]* for Halfspaces

For Sigmoids: lower bound scales as $d^{\Omega(\log^2(1/\epsilon))}$ No result was known for Sigmoids

See concurrent work by *[Diakonikolas-Kane-Zarifis'20]* and subsequent work by *[Diakonikolas-Kane-Pittas-Zarifis'21]* which generalizes and tightens this result

OUR APPROACH

- Standard SQ lower bounds work by constructing a large class of functions that are nearly orthogonal
- We prove via a reduction using known SQ lower bounds instead of explicitly constructing a family of functions

$$\sigma(w \cdot x)$$

Learner for GLM $\mathscr{A}$ in agnostic model

$$\longrightarrow$$

Learner for two-layer NN in the realizable model

$$\psi\left(\sum_{i=1}^{k} a_i\, \sigma(W_i \cdot x)\right)$$

ψ takes the input and maps to $[-1,1]$

OUR APPROACH

$\sigma(w \cdot x)$

Learner for GLM $\mathscr{A}$ in agnostic model

Learner for two-layer NN in the realizable model

$$\psi\left(\sum_{i=1}^{k} a_i\,\sigma(W_i \cdot x)\right)$$

ψ on the outside is important to get general SQ lower bounds

Goel-Gollakota-Klivans'20; Diakonikolas-Kane-Kontonis-Zarifis'20

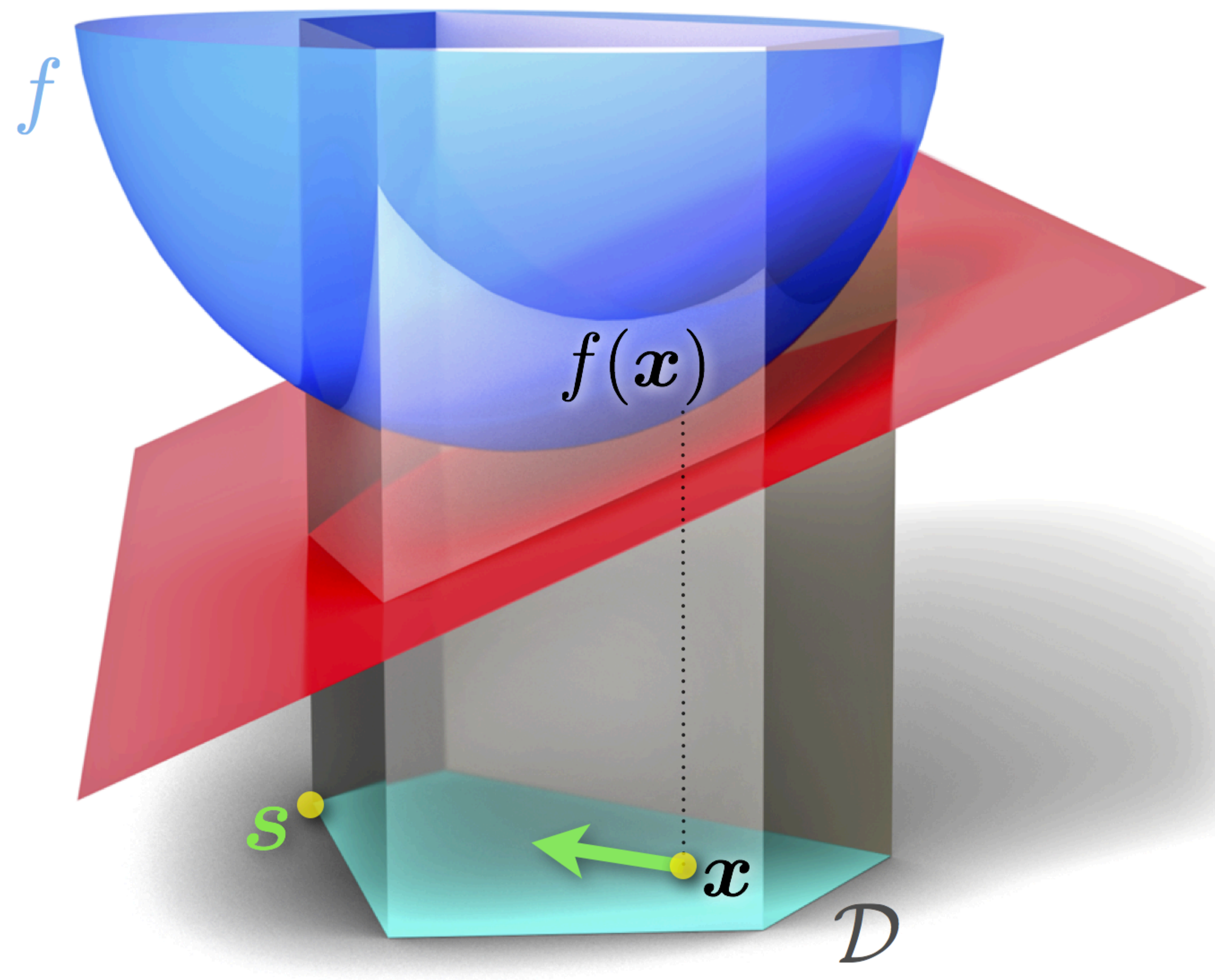
Any SQ algorithm for learning the above NN in the *realizable noise model* needs super-polynomial number of queries or super-polynomial tolerance.

FRANK WOLFE

To minimize a function, at each step we find an element in our set that maximizes inner-product with the negative of the gradient and update our current estimate

Algorithm 1 Frank–Wolfe gradient descent over a generic inner product space

Start with an arbitrary $z_0 \in \mathcal{Z}$.
for $t = 0, \ldots, T$ **do**
  Let $\gamma_t = \frac{2}{t+2}$.
  Find $s \in \mathcal{Z}$ such that $\langle s, -\nabla p(z_t)\rangle \geq \max_{s' \in \mathcal{Z}'} \langle s', -\nabla p(z_t)\rangle - \frac{1}{2}\delta\gamma_t C_p$.
  Let $z_{t+1} = (1-\gamma_t)z_t + \gamma_t s$.
end for



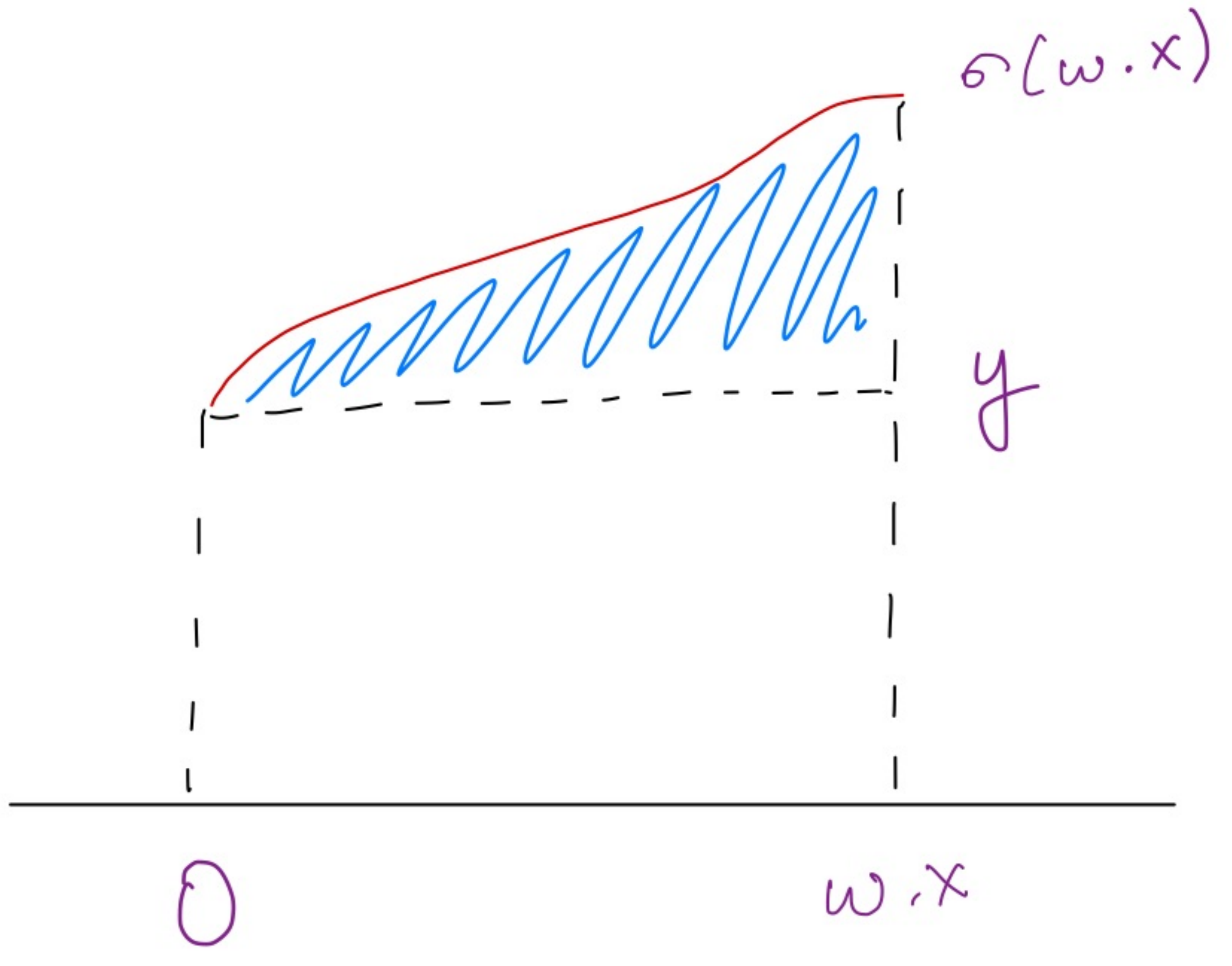
Source: Wikipedia

FUNCTIONAL FRANK-WOLFE

- We use Frank Wolfe on the function space using a convex surrogate loss functional

$$\text{loss}_{\text{surr}}(f) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\int_0^{f(x)} (\psi(a) - y)da\right]$$

- This update step turns out to be equivalent to solving the agnostic GLM problem on a residual



- Each time we add a new neuron to our existing linear combination

Square loss would have let us learn sum of neurons which would get a CSQ lower bound not general SQ lower bound

COMPLETING THE REDUCTION

- We can simulate the queries using the SQ oracle of the original problem
- We can bound the number of times the inner optimization is run

Using standard FW proof (surrogate is convex)

- If the inner loop was efficient then we could learn the two-layer NN

However, this is a contradiction

Surrogate loss can handle the non-linearity in the second layer

Part 1:

- ***Distribution agnostic proper*** ReLU regression ***over bounded inputs*** is not tractable under a ***conditional hardness assumption***

  ➕ Separation b/w proper and improper

  ➕ Holds against all algorithms

  ➖ Discrete/specialized input distribution

Part 2:

- ReLU regression with even ***Gaussian inputs*** is not tractable ***unconditionally for Statistical Query Algorithms***

  ➕ Unconditional hardness even for benign distribution

  ➖ Holds against only SQ algorithms

WHAT NEXT?

- We want to avoid this computational barriers under reasonable assumptions
- These hardness results indicate what assumptions do not suffice to get positive results

(Relaxed) **Goal**: Output hypothesis h such that:

$$\text{loss}(h) \leq C \min_{c \in \mathscr{C}} \text{loss}(c) + \epsilon$$

*Diakonikolas-**Goel**-Karmalkar-Klivans-Soltanolkotabi'20*

There exists an algorithm that approximately learns the ReLU over any isotropic log-concave distribution using $\tilde{O}(d/\epsilon)$ samples in time $\tilde{O}(d^2/\epsilon)$.

WHAT NEXT?

- We want to avoid this computational barriers under reasonable assumptions
- These hardness results indicate what assumptions do not suffice to get positive results

(Stronger) **Assumptions**: Underlying distribution has additional structure

Goel-Klivans'17

There exists poly-time algorithms over bounded domain if the marginal distribution has strong Eigen-value decay.

WHAT NEXT?

- We want to avoid this computational barriers under reasonable assumptions
- These hardness results indicate what assumptions do not suffice to get positive results