Proceedings of Machine Learning Research vol :1–6, 2026

Invited Open Problem: Is the Power of Deep Learning over Linear Models Inherently Distribution Dependent?

Vitaly Feldman VITALY.EDU@GMAIL.COM
Apple

Pritish Kamath PRITISH@ALUM.MIT.EDU
Google Research

Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago

Abstract

We ask whether distribution-*independent* SQ learning implies low dimension complexity, and whether anything learnable with (S)GD on a (benign) neural network under any input distribution is also learnable with a linear model.

1. Background and Framing

What is the power of deep learning over linear (or kernel) models? Or are linear models already "universal", so that anything learnable via deep learning, or perhaps any method, is also learnable via a linear model?

If we are only concerned with sample complexity, ignore computation, and consider intractable Empirical Risk Minimization (ERM) learning, Ben-David et al. (2002) already established linear (or kernel) models are not universal: there are hypothesis classes learnable with a small number of samples that require exponentially higher dimensionality (or norm) to represent with a linear model. More formally, Ben-David et al. defined the *dimension complexity* $\mathsf{dc}(\mathcal{H})$ of a binary hypothesis class $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ as the smallest dimension d s.t. there exists a feature map $\varphi : \mathcal{X} \to \mathbb{R}^d$ allowing linear representation of $\mathcal{H}$ (i.e. s.t. $\forall_{h\in\mathcal{H}}\exists_{w\in\mathbb{R}^d}\forall_x h(x) = \mathsf{sign}(\langle w, \varphi(x)\rangle)$), and proved that it could be much higher than the VC dimension of $\mathcal{H}$. It is not difficult to realize some of these classes, e.g. the class of all 2^n parities over n input bits, as small neural networks. This implies a sample complexity separation between ERM on neural networks vs learning with linear (or kernel) models.

But ERM on a neural network is neither worst-case tractable nor realistically implementable—instead neural networks are typically trained using (Stochastic) Gradient Descent (SGD, or similar gradient-based methods), which is rather different from ERM. Additionally, on the classes for which it is easy to show a sample complexity gap vs linear models, like parities, (S)GD fails at learning—at least for a worst-case input distribution.

Thanks to the implicit bias of Gradient Descent, it is also possible to show a sample complexity benefit of (S)GD on a neural network versus linear models: Gradient Descent on a "diagonal linear network" can learn 1-sparse functions on n inputs (i.e. the class $\mathcal{H} = \{x \mapsto \langle e_i, x\rangle \mid i = 1, \ldots, n\}$ of coordinate projections) using $O(\log n)$ samples (Woodworth et al., 2020; Chou et al., 2023), while linear models would require $\Omega(\mathsf{dc}(\mathcal{H})) = \Omega(n)$ samples. But the size of the network, and hence computational cost, is still $\Omega(\mathsf{dc}(\mathcal{H}))$. Thus, while there is an exponential sample complexity benefit here, there isn't such a benefit in terms of the size of the model or the computational cost.

This prompts us to modify the question about the power of deep learning over linear models, focusing on model size and computational cost: is anything learnable by (S)GD on a neural network also learnable using a linear model of dimension proportional to the size of the neural network and training time? Or are there problems learnable by a small neural network using a reasonable number of (S)GD iterations, that would require dimension exponentially larger to learn using a linear model?

Indeed, there are many known examples of problems learnable by (S)GD on a neural network, that would require dimension exponential in the network size and SGD computational effort to represent and learn using linear models (and this in turn also provides a lower bound on the sample complexity using kernel methods). However, as far as we are aware, all of these are *distribution-dependent*. That is, these are examples of a hypothesis class $\mathcal{H}$ and *an explicit input distribution* $\mathcal{D}$ over $\mathcal{X}$, such that for any $h^\star \in \mathcal{H}$, (S)GD on samples $x \sim \mathcal{D}$ labeled (perhaps noisily) by $h^\star$, succeeds in learning. A simple example is the class of parities mentioned above, with a distribution $\mathcal{D}$ which is a mixture of a uniform distribution over $\{\pm 1\}^n$ and a distribution which is uniform over 1-sparse inputs. Although parities are hard to learn in the worst case (with noisy labels, or with a Statistical Query (SQ) algorithm (Kearns, 1998)), this specific input distribution creates strong correlation between bits in the support of the parity and the output, and can thus be easily learned, e.g. by a simple SQ algorithm. Indeed, (S)GD on a neural network can also pick up these correlations and learn the parity (Malach et al., 2021; Medvedev et al., 2026). However, even on this "easy" distribution, it is still impossible to even approximately represent parities using linear separators in dimension $2^{o(n)}$. See Malach et al. (2021) for a discussion of how other separation results are also distribution-dependent. More recently, the staircase property (Abbe et al., 2021a, 2022) has been used to understand situations where (S)GD on a neural net (or alternatively, SQ algorithms) succeed, even when linear (or kernel) methods fail. Importantly, the staircase property is *distribution-dependent* as the basis used in the staircase is orthonormal w.r.t. the specific input distribution $\mathcal{D}$. In fact, in the parity example, the specific input distribution used can be understood as changing the basis so as to provide a staircase.

What we ask here is whether such a separation (between the size and computational cost of deep learning vs the dimensionality needed for linear learning) is inherently *distribution-dependent*, or whether it is possible to show hypothesis classes that are learnable by (S)GD on a (simple, benign and realistic) neural network w.r.t. *all* input distributions, but have dimension complexity much higher than the size of the network and number of (S)GD steps.

Viewing Statistical Query (SQ) learning as a useful guide for understanding deep learning, and realistic noise-tolerant learning more generally, we also pose a similar (though not formally related) question about the relationship between distribution-independent SQ learning and the dimension complexity: If a hypothesis class $\mathcal{H}$ is (m, τ)-SQ-learnable for all input distributions, does this imply $\mathsf{dc}(\mathcal{H}) = O(m/\tau^2)$?

2. Learning with SGD over Neural Networks

We first formally define an SGD learning procedure over a neural network. For concreteness and simplicity we focus on *fully connected ReLU networks* over inputs in $\mathcal{X} = \{\pm 1\}^n$. An architecture is thus defined by a depth L and widths $n_1, n_2, \ldots, n_{L-1}$ where we have $n_0 = n$ input units and $n_L = 1$ output unit. The parameters $\theta = (\theta_1, \theta_2, \ldots, \theta_L) \in \mathbb{R}^S$ of the network are the $S := \sum_{i=1}^{L} n_i n_{i-1}$ weights on each of the edges, and the output of the network is defined as $f_\theta(x) = \theta_L \sigma(\theta_{L-1} \sigma(\cdots \sigma(\theta_1 x) \cdots))$, where $\sigma(z) = \max(0, z)$ operates elementwise. For an in-

put distribution $\mathcal{D}$ over $\mathcal{X}$ and a target $h^\star \in \{\pm 1\}^{\mathcal{X}}$, SGD with stepsize η operates as follows: each weight of $\theta^{(0)}$ is drawn independently from a Gaussian, where elements in θ_i have variance $1/n_{i-1}$ (i.e. standard initialization). At each step $t = 0, \ldots, T-1$, we draw a sample $x^{(t)} \sim \mathcal{D}$ and perform the update: $\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \nabla_\theta \ell(h^\star(x^{(t)}) f_{\theta^{(t)}}(x^{(t)}))$, where $\ell(z) := \log(1 + e^{-z})$ is the logistic loss and then return the predictor $\hat{h}(x) = \text{sign}\left(\sum_{t=\lceil T/2 \rceil}^{T} f_{\theta^{(t)}}(x)\right)$. We denote the error achieved by $\hat{h}$ as $\mathcal{L}_{\mathcal{D},h^\star}(\hat{h}) := \Pr_{x\sim\mathcal{D}}[\hat{h}(x)h^\star(x) < 0]$.

Open Question 1 (SGD Learning vs. Dimension Complexity) *Is there a constant C such that for all $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ over $\mathcal{X} = \{\pm 1\}^n$, and $\varepsilon < 1/4$, if there exists a fully connected ReLU network with S parameters in total, stepsize η and number of steps T, such that for every input distribution $\mathcal{D}$, every $h^\star \in \mathcal{H}$, SGD on the said architecture yields expected error $\mathbb{E}\,\mathcal{L}_{\mathcal{D},h^\star}(\hat{h}) \le \varepsilon$ (expectation over the initialization and SGD sampling), then* $\mathsf{dc}(\mathcal{H}) \le C \cdot TS$.

We do not view the specifics here as critical, and we could also consider other architectures, activation functions, loss functions, step size schedules and GD variants. We cannot, however, allow an arbitrary architecture or an arbitrary initialization: With a specialized unrealistic architecture, activation function, or initialization, it is possible to "cheat" and simulate any algorithm via (S)GD on the network (Abbe and Sandon, 2020; Abbe et al., 2021b). Instead of formally defining "benign" or "natural" networks, we state the open problem in terms of the simplest architecture. However, the question is relevant and interesting for any "benign" architecture that avoids such "cheating".

3. Statistical Query Learning

A τ-statistical query (SQ) oracle for input distribution $\mathcal{D}$ and target $h^\star$, on input query $q : \mathcal{X} \times \{\pm 1\} \to [-1, 1]$ and tolerance τ returns an arbitrary value v such that $|v - \mathbb{E}_{x\sim\mathcal{D}}\, q(x, h^\star(x))| \le \tau$. A (randomized) (m, τ)-statistical query (SQ) algorithm operates by making a sequence of m queries to a τ-SQ oracle where each query can depend on all previous responses and can be selected at random, and then returns a predictor $\hat{h} : \mathcal{X} \to \{\pm 1\}$.

Open Question 2 (SQ Learning vs. Dimension Complexity) *Is there a constant C such that for every class $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$, over any domain $\mathcal{X}$, and any $\varepsilon < 1/4$, if there exists an (m, τ)-SQ algorithm s.t. for every input distribution $\mathcal{D}$ and every $h^\star \in \mathcal{H}$, the algorithm returns a predictor $\hat{h}$ with $\mathbb{E}\,\mathcal{L}_{\mathcal{D},h^\star}(\hat{h}) \le \varepsilon$ (expectation over the randomness of the algorithm), then* $\mathsf{dc}(\mathcal{H}) \le C \cdot m/\tau^2$.

The converse of Open Question 2 is known to hold up to a dependence on $n = \log|\mathcal{X}|$ (the log size of the domain, or bit complexity of representing an element $x \in \mathcal{X}$). Specifically, if $\mathsf{dc}(\mathcal{H}) = d$ then there is an (m, τ)-SQ PAC learning algorithm for $\mathcal{H}$ with $m, 1/\tau = \text{poly}(d, n, 1/\varepsilon)$. This result is a corollary of the algorithm of Dunagan and Vempala (2004) for learning halfspaces, which accesses data via two variants of the Perceptron algorithm and can be straightforwardly converted to a SQ algorithm (see also Appendix A of Balcan and Feldman, 2013). The query complexity and the inverse tolerance of the resulting algorithm are polynomial in d and $\log(1/\gamma)$, where γ is the margin achieved by the linear separator. It is also well known that bit complexity lower bounds the achievable margin as $\log(1/\gamma) = O(nd \log d)$ (e.g. Lemma 10 in Gonen et al., 2013).

We briefly note that *weak* and distribution-independent learnability via Correlational SQs (a restriction of SQ model where the queries are of the form $q(x, y) = y \cdot q'(x)$) is known to be equivalent to polynomial margin complexity (Feldman, 2008), which is a stronger notion than dimension

complexity. Distribution independent CSQ learning is however a very restricted model of learning in which even Boolean conjunctions are not efficiently PAC learnable (Feldman, 2011). It also known that there exists function classes SQ learnable efficiently over any fixed distribution but not efficiently SQ-learnable distribution-independently (Feldman, 2017).

One connection between the two open problems follows from the fact that learnability with an approximate version of gradient oracle in which the oracle is allowed to return an arbitrary vector that is close to the true gradient (in ℓ_2 norm), is known to be equivalent to SQ learning (Feldman et al., 2021; Abbe et al., 2021b). Thus, an upper bound on the dimension complexity in terms of the number of steps and size of the network in Open Question 1 for such an approximate version of gradient descent would also imply an upper bound on the dimension complexity in terms of the SQ complexity. This connection between SQ learnability and approximate SGD is also used in a recent work of Karchmer and Malach (2025). Their work is motivated by understanding the power of SGD on neural networks and aims to prove an upper bound on an approximate variant of dimension complexity for SQ learnable classes. However the claimed upper bounds do not hold due to a flaw in the proof (Karchmer and Malach, 2026).

4. Relaxations and Generalizations

Polynomial Instead of Linear. Even if a linear upper bound cannot be established, it would be interesting to show any polynomial upper bound, $\mathsf{dc}(\mathcal{H}) = \mathrm{poly}(S, T)$ or $\mathsf{dc}(\mathcal{H}) = \mathrm{poly}(m, 1/\tau)$— or, conversely, to rule out such a bound.

Dependence on the Size of the Domain. Open Question 2 is stated for an arbitrary domain, and without a dependence on the size of the domain. As discussed in Section 3, bounding SQ learnability in terms of the dimension complexity (the converse of Open Question 2) using existing approaches requires a dependence also on $n = \log|\mathcal{X}|$. It might therefore be reasonable to allow such a dependence also in the bound in Open Question 2 and settle for $\mathsf{dc}(\mathcal{H}) = \mathrm{poly}(m, 1/\tau, n)$.

Probabilistic Variants. The learning algorithms often rely crucially on randomization, for example in the random initialization of the network or initial solution in the algorithm of Dunagan and Vempala (2004). While we do not know of any counterexamples to the conjectured upper bounds on the deterministic dimension complexity, it may be that such counterexamples exist solely due to the deterministic nature of the dimension complexity. To address this we also formulate our conjecture with two probabilistic variants of dimension complexity. The weakest relaxation is to consider a distribution over mappings and allow a failure probability (e.g. Chornomaz et al., 2025). Specifically we ask the same questions for $1/2$-confident dimension complexity $\mathsf{dc}^{1/2}(\mathcal{H})$ in place of $\mathsf{dc}(\mathcal{H})$ which we formally define below. For $\delta > 0$, $\mathsf{dc}^{\delta}(\mathcal{H})$ is the smallest d for which there exists a distribution $\mathcal{P}$ over embeddings $\varphi : \mathcal{X} \to \mathbb{R}^d$ such that for all distributions $\mathcal{D}$ over $\mathcal{X}$ and all $h^\star \in \mathcal{H}$, it holds that $\Pr_{\varphi\sim\mathcal{P}}\left[\inf_{w\in\mathbb{R}^d} \mathcal{L}_{\mathcal{D},h^\star}(h_{w,\varphi}) = 0\right] \geq 1-\delta$, where $h_{w,\varphi}(x) := \langle w, \varphi(x)\rangle$.

A stronger relaxation is the probabilistic dimension complexity $\mathsf{dc}_\varepsilon(\mathcal{H})$ (Kamath et al., 2020) which also allows approximate representation. Formally, $\mathsf{dc}_\varepsilon(\mathcal{H})$ is the smallest d for which there exists a distribution $\mathcal{P}$ over embeddings $\varphi : \mathcal{X} \to \mathbb{R}^d$ such that for all distributions $\mathcal{D}$ over $\mathcal{X}$ and all $h^\star \in \mathcal{H}$, it holds that $\mathbb{E}_{\varphi\sim\mathcal{P}}\left[\inf_{w\in\mathbb{R}^d} \mathcal{L}_{\mathcal{D},h^\star}(h_{w,\varphi})\right] \leq \varepsilon$. For this notion we ask whether (SGD or SQ) learning to within error ε implies an upper bound on $\mathsf{dc}_{C\cdot\varepsilon}(\mathcal{H})$. We remark that this stronger notion itself does not imply efficient learnability (or polynomial SQ complexity) due to approximate representation. It still captures the approximation power of linear classifiers.

References

Emmanuel Abbe and Colin Sandon. On the universality of deep learning. In *Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e7e8f8e5982b3298c8addedf6811d500-Abstract.html.

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. The staircase property: How hierarchical structure can guide deep learning. In *Neural Information Processing Systems (NeurIPS)*, 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/a45a1d12ee0fb7f1f872ab91da18f899-Abstract.html.

Emmanuel Abbe, Pritish Kamath, Eran Malach, Colin Sandon, and Nathan Srebro. On the power of differentiable learning versus PAC and SQ learning. In *Neural Information Processing Systems (NeurIPS)*, pages 24340–24351, 2021b. URL https://proceedings.neurips.cc/paper/2021/hash/cc225865b743ecc91c4743259813f604-Abstract.html.

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. The merged-staircase property: A necessary and nearly sufficient condition for SGD learning of sparse functions on two-layer neural networks. In *Conference on Learning Theory (COLT)*, volume 178 of *Proceedings of Machine Learning Research*, pages 4782–4887. PMLR, 2022. URL https://proceedings.mlr.press/v178/abbe22a.html.

Maria-Florina F Balcan and Vitaly Feldman. Statistical active learning algorithms. *Advances in neural information processing systems*, 26, 2013.

Shai Ben-David, Nadav Eiron, and Hans Ulrich Simon. Limitations of learning via embeddings in euclidean half spaces. *J. Mach. Learn. Res.*, 3:441–461, 2002. URL https://jmlr.org/papers/v3/bendavid02a.html.

Bogdan Chornomaz, Shay Moran, and Tom Waknine. On reductions and representations of learning problems in euclidean spaces. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 2043–2054, 2025.

Hung-Hsu Chou, Johannes Maly, and Holger Rauhut. More is less: Inducing sparsity via overparameterization. *Information and Inference: A Journal of the IMA*, 12(3):iaad012, 2023. doi: 10.1093/imaiai/iaad012.

John Dunagan and Santosh Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 315–320, 2004.

Vitaly Feldman. Evolvability from learning algorithms. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 619–628, 2008.

Vitaly Feldman. Distribution-independent evolvability of linear threshold functions. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 253–272. JMLR Workshop and Conference Proceedings, 2011.

Vitaly Feldman. A general characterization of the statistical query complexity. In *Conference on learning theory*, pages 785–830. PMLR, 2017.

Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. *Mathematics of Operations Research*, 46 (3):912–945, 2021.

Alon Gonen, Sivan Sabato, and Shai Shalev-Shwartz. Efficient active learning of halfspaces: An aggressive approach. *Journal of Machine Learning Research*, 14(79):2583–2615, 2013. URL http://jmlr.org/papers/v14/gonen13a.html.

Pritish Kamath, Omar Montasser, and Nathan Srebro. Approximate is good enough: Probabilistic variants of dimensional and margin complexity. In *Conference on Learning Theory (COLT)*, Proceedings of Machine Learning Research, pages 2236–2262. PMLR, 2020. URL http://proceedings.mlr.press/v125/kamath20b.html.

Ari Karchmer and Eran Malach. The power of random features and the limits of distribution-free gradient descent. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 29078–29097. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/karchmer25a.html.

Ari Karchmer and Eran Malach. Personal communication, 2026.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Eran Malach, Pritish Kamath, Emmanuel Abbe, and Nathan Srebro. Quantifying the benefit of using differentiable learning over tangent kernels. In *International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pages 7379–7389. PMLR, 2021. URL http://proceedings.mlr.press/v139/malach21a.html.

Marko Medvedev, Idan Attias, Elisabetta Cornacchia, Theodor Misiakiewicz, Gal Vardi, and Nathan Srebro. Positive distribution shift as a framework for understanding tractable learning. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2602.08907. To appear.

Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory (COLT)*, Proceedings of Machine Learning Research, pages 3635–3673. PMLR, 2020. URL http://proceedings.mlr.press/v125/woodworth20a.html.