Research Statement
A Holistic Theory of the Deep Learning Methodology

Niladri Chatterji

The recent emergence of deep learning has led to rapid progress on tasks across natural language processing, computer vision, and speech processing. However, the underlying reason for the success of these models is not well understood. As their use becomes ubiquitous, it is an important open problem to develop a theoretical understanding of these models, and also develop methods to improve them. Empirically, it is evident that this success is not just due to the use of neural networks but due to the entire deep learning methodology that includes (a) model architecture, (b) optimizer, and (c) loss function. A successful theory must therefore study these components in unison. For example, a theoretical explanation for why overparameterized networks found by gradient descent generalize to unseen data despite perfectly fitting the noisy training data must account for the interplay between the optimization algorithm and model class. However, classical theoretical approaches only focused on the model class and therefore failed to differentiate between the "good" solution found by gradient descent from among the many "bad" solutions with zero training error. In contrast, my work develops a holistic theory to investigate the properties of the specific neural network solutions found by optimizers like gradient descent trained on popular loss functions.

Adopting this holistic viewpoint, I have studied several surprising phenomena in modern overparameterized neural networks. Returning to the question above, to understand why overparameterized neural networks trained by gradient descent generalize, my collaborators and I [18] developed a theoretical framework to show that the gradient descent dynamics on overparameterized models are intrinsically tolerant to noise in the training data and therefore lead to good solutions. In addition, I have also used this approach to study other questions like why overparameterized networks learn good features [19] and why gradient descent successfully optimizes non-convex training objectives[11, 10]. Furthermore, by leveraging these techniques that I developed to study generalization, I have also developed algorithms to obtain robust classifiers [28] and optimally select model sizes [6].

Understanding Why this Methodology is Successful

A dominant paradigm in deep learning is to train overparameterized neural networks, where the number of parameters in the model exceeds the number of datapoints, by gradient-based methods to nearly interpolate (perfectly fit) the datapoints. This learning paradigm presents many surprises and one aspect of my past work studies these.

Why do overparameterized neural networks generalize? Consider an example of a vision system that identifies traffic stop signs from data. Large training corpora are often noisy, and some stop signs in the training data could be mislabeled. Would a neural network model that fits this noisy training data identify a new unseen stop sign correctly? That is when is (over)fitting to this noisy training data benign? Empirically, we find overparameterized networks that perfectly fit noisy data generalize well to unseen test data. However, the underlying reason for this generalization is not well understood.

On the one hand, overparameterization engenders models the expressive power to represent complex functions, but on the other, this expressivity means that they might generalize poorly to new data by overfitting to the noise in the training data. Classical generalization theory, which focused solely on studying the properties of the model class, has failed to provide an answer to why networks learned by gradient descent tend to generalize. My research studies the interplay between the model class, optimizer, and loss function to characterize when and why solutions found in practice generalize.

We studied this phenomenon of benign overfitting in two-layer neural network classifiers trained with gradient descent on the logistic loss [18]. We showed that gradient descent not only successfully finds a zero loss solution to this non-convex problem, but the final solution recovers the optimal predictor despite overfitting to the noise. The key reason for this benign overfitting is that the gradient descent algorithm is inherently noise-tolerant, and it ensures that the influence that noisy training points can have is bounded at every iteration. Our work used a framework that I had previously developed to study linear classifiers [7] that allowed us to study the generalization error of the limiting interpolating model obtained by gradient descent. Our work was the first to show that nonlinear neural network classifiers can perfectly fit noisy training data and yet generalize to unseen test data.

Using this general framework to study the limiting interpolant obtained by gradient descent, I have also investigated other questions related to generalization. Specifically, I have studied the role that depth plays in generalization. A widespread belief is that depth plays an important role in generalizing to new data [22], however, precisely why it helps is unclear. My collaborators and I have studied this question with deep linear networks [8, 12] trained with gradient descent on the squared loss. Our work uncovered one aspect where depth *does not* help generalization. We showed that deep linear networks have the same *statistical variance* as shallow ones. As a consequence if a shallow neural network is susceptible to being misled by the noise in the training data, then a deeper network is too. Our study leaves open the possibility that depth can help generalization in other ways (such as having lower statistical bias) and this is something that I hope to understand in the future.

Why do neural networks trained by gradient descent learn good features? A useful aspect of modern neural networks is that features learned by the model training on one task *transfer* over to other tasks via minimal or no retraining [17, 26]. We explored why the network learns good features in the case of a randomly initialized two-layer neural network trained with gradient descent on the logistic loss [19]. We found that the model architecture, random initialization, and logistic loss all play an important role in learning good features. The random initialization presents the network with many random features as "options", and then the process of gradient descent on the logistic loss picks out the good features among these and refines this "good subset" in subsequent iterations. Our work was among the first to show training neural networks outside the neural tangent kernel regime (where no feature learning occurs) can lead to a small loss solution while learning good features.

Why does gradient descent find neural networks that are global optima? Gradient descent is empirically extremely successful in training overparameterized neural networks, even though it involves solving highly non-convex problems. To understand why, we theoretically analyzed deep neural networks trained by gradient descent on the logistic loss [10, 11]. We established global convergence guarantees for sufficiently overparameterized deep neural networks and showed that gradient descent provably drives the logistic loss to zero. This analysis presented several challenges since minimizing the logistic loss to zero requires the weights to go off to infinity. Therefore, we could not operate in the well-studied neural tangent kernel (NTK) regime where the iterates remain bounded in a ball around the initial weights. Our analysis revealed that there are two phases in the training: (1) in the first phase overparameterization helps to implicitly smoothen the loss landscape, and (2) in the second phase it is the exponential tail structure of the logistic loss that is critical in successfully driving the loss to zero. Our results were among the first to provide end-to-end guarantees showing that gradient descent drives the logistic loss to zero for deep neural networks.

Changing and Improving the Deep Learning Methodology

Apart from understanding the surprises that the current deep learning methodology presents, I am also interested in improving the deep learning methodology further. My guiding principle on this front is to change a component in the deep learning pipeline (loss function/optimizer/architecture) and track how this change affects the entire learning algorithm.

Obtaining robust neural network classifiers (by changing the loss function). I have also used the general framework to bound generalization error of the limiting iterate of gradient descent to develop algorithms to obtain classifiers that are robust to distribution shift between train and test time. A classical method to address distribution shift is by training on an importance weighted loss function. However, previous attempts to obtain robust overparameterized neural network classifiers by training on such importance weighted losses had failed, which led the community to question whether such classical robustness interventions are fundamentally incompatible with the modern overparameterized regime [2]. By using this framework, in [28] we showed that importance weighting is not incompatible in the overparameterized regime, but instead the previous failures were a result of using importance weights along with popular exponentially-tailed losses like the logistic or cross-entropy loss. We showed that switching to other loss functions (that are polynomially-tailed) restores the effects of importance weighting. When our new loss function is applied in practice, we saw a 9% increase in test accuracy on standard distribution shift benchmarks compared to previous baselines.

Optimally selecting the model size (by changing the architecture). The performance of overparameterized models depend significantly on architectural design decisions. Further, overparameterized models require significant computational resources to train, and therefore it is important to make these design decisions correctly. Chief among these choices is deciding how large the model size should be relative to the sample size. To navigate this tradeoff between model size and sample size subject to a computational constraint, the ML community has resorted to building *empirical scaling laws* [25]—training many smaller models and then extrapolating based on their performance. However, this approach is resource intensive and can lead to inaccurate extrapolations [21]. In [6], again by deploying the general framework that I had developed previously, we developed theoretically grounded scaling laws by analyzing the generalization error of overparameterized two-layer neural networks trained by gradient descent as a function of both the model size (width) and the sample size. Due to the nonparametric nature of neural network models, we found that the test error decays at nonstandard rates (with exponents different than $1/2$ or 1) with both the number of samples and the width, providing an explanation for the nature of previous empirical scaling laws. Our theoretical results also allow us to choose the optimal model size as a function of the number of samples and provide guidance regarding how to optimally select the model size.

Understanding tradeoffs between memory and performance (by changing the optimizer). Neural network weights are sometimes sparsified to reduce memory costs [20]. However, empirically sparse networks had been observed to perform much worse than dense counterparts, even when both perfectly interpolate the training data [3]. This led us to ask, is there a better training algorithm that yields sparse models, or is there a fundamental tradeoff between memory and performance in the overparameterized interpolating regime? We studied this question with linear models [9] and showed that there is indeed a fundamental tradeoff between memory and performance. We proved an information-theoretic lower bound of the performance of any sparse model and showed that their performance can be exponentially worse than a dense model, even when the underlying ground truth is sparse. The key intuition driving our lower bound is that all of the extra parameters are akin to slack variables, and help spread the effect of the training noise in many directions. Our result shows that the performance benefits of modern overparameterized models come at the cost of high memory usage.

Other Work

My work has sought to develop a rigorous statistical methodology for practical problems.

Understanding Diffusion Algorithms. Markov chain Monte Carlo algorithms like Langevin Monte Carlo (LMC) and Hamiltonian Monte Carlo (HMC) have long been used in wide variety of statistical inference tasks [24], and more recently to train generative diffusion models [27] and differentially private models [1]. However, unlike gradient-based optimization algorithms, these gradient-based MCMC algorithms are relatively poorly understood and explored. By drawing connections from probability theory, our work established the first theoretical guarantees for accelerated [16] and variance reduced [5] versions of Langevin Monte Carlo when the negative log-densities are convex. Further, we also showed that it is possible to sample efficiently using Langevin Monte Carlo from non-convex objectives even when optimization is provably impossible [15, 23]. We also were the first to establish fundamental limits on the abilities of Monte Carlo algorithms and showed via information-theoretic lower bounds that the popular first-order algorithm stochastic gradient Langevin Monte Carlo is optimal [4].

Sample-Efficient Reinforcement Learning. As the application of reinforcement learning moves beyond games (where the samples are relatively cheap) into the real-world, it is becoming increasingly important to develop algorithms that are sample efficient without foregoing performance. To attain this goal I have made algorithmic progress along two fronts. Reinforcement learning algorithms rely on the availability of a reward function which is often unavailable in practice. In [14], we developed algorithms that eschew the need for reward function at every step, and only rely binary signal (success/failure) at the end of each episode. Another avenue to improve sample efficiency is by an adaptive choice of the policy class. For any problem, the optimal policy class should be rich enough so as to contain a good policy within it, but not too rich that is requires too many samples to learn. In [13], we developed a sequential hypothesis testing approach to optimally model select between policy classes in the linear contextual bandit problem.

Next Steps

Apart from advancing research in the directions highlighted above and I would also like to explore the following themes going forward.

Surprises and challenges in large pretrained models. Large pretrained models have demonstrated several surprising phenomena like incontext learning (where the model learns to perform new tasks without any weight updates but only through a few input-output pairs provided during test time), and they learn general features that easily transfer over to other tasks. I would build on my past work on overparameterized neural networks to develop a theoretical foundation to understand these new phenomena. Apart from these surprises, training these large models also presents significant challenges. For example, the dynamics of gradient algorithms are often unstable and can lead to failures. I hope to develop new algorithms to address these challenges.

Data as another leg of the methodology. I would like to understand how the choice of training data interacts with the three other components highlighted above and how it affects the model's ability to generalize. Another question involving data is in the modern pretraining-finetuning paradigm of learning. Here a practical problem is how one should choose the data to train a pretrained model so that it can adapt via minimal finetuning to many different downstream tasks. I hope to develop principled methods to select this data to enable the model to learn such multipurpose features.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. McMahan, I. Mironov, K. Talwar, and L. Zhang. “Deep learning with differential privacy”. In: *Conference on Computer and Communications Security*. 2016.

[2] J. Byrd and Z. Lipton. “What is the effect of importance weighting in deep learning?” In: *International Conference on Machine Learning (ICML)*. 2019.

[3] X. Chang, Y. Li, S. Oymak, and C. Thrampoulidis. “Provable benefits of overparameterization in model compression: From double descent to pruning neural networks”. In: *Conference on Artificial Intelligence (AAAI)*. 2021.

[4] **N. Chatterji**, P. Bartlett, and P. Long. “Oracle lower bounds for stochastic gradient sampling algorithms”. In: *Bernoulli* (2022).

[5] **N. Chatterji**, N. Flammarion, Y. Ma, P. Bartlett, and M. Jordan. “On the theory of variance reduction for stochastic gradient Monte Carlo”. In: *International Conference on Machine Learning (ICML)*. 2018.

[6] **N. Chatterji**, P. Liang, and T. Hashimoto. “Understanding sample and model size tradeoffs in two-layer neural networks (in preparation)”. 2022.

[7] **N. Chatterji** and P. Long. “Finite-sample analysis of interpolating linear classifiers in the overparameterized regime”. In: *Journal of Machine Learning Research (JMLR)* (2021).

[8] **N. Chatterji** and P. Long. “Deep linear networks can benignly overfit when shallow ones do”. In: *arXiv preprint arXiv:2209.09315* (2022).

[9] **N. Chatterji** and P. Long. “Foolish crowds support benign overfitting”. In: *Journal of Machine Learning Research (JMLR)* (2022).

[10] **N. Chatterji**, P. Long, and P. Bartlett. “When does gradient descent with logistic loss find interpolating two-layer networks?” In: *Journal of Machine Learning Research (JMLR)* (2021).

[11] **N. Chatterji**, P. Long, and P. Bartlett. “When does gradient descent with logistic loss interpolate using deep networks with smoothed ReLU activations?” In: *Conference on Learning Theory (COLT)*. 2021.

[12] **N. Chatterji**, P. Long, and P. Bartlett. "The interplay between implicit bias and benign overfitting in two-layer linear networks". In: *Journal of Machine Learning Research (JMLR)* (2022).

[13] **N. Chatterji**, V. Muthukumar, and P. Bartlett. "OSOM: A simultaneously optimal algorithm for multi-armed and linear contextual bandits". In: *International Conference on Artificial Intelligence and Statistics (AISTATS).* 2020.

[14] **N. Chatterji**, A. Pacchiano, P. Bartlett, and M. Jordan. "On the theory of reinforcement learning with once-per-episode feedback". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

[15] X. Cheng, **N. Chatterji**, Y. Abbasi-Yadkori, P. Bartlett, and M. Jordan. "Sharp convergence rates for Langevin dynamics in the nonconvex setting". In: *arXiv preprint arXiv:1805.01648* (2018).

[16] X. Cheng, **N. Chatterji**, P. Bartlett, and M. Jordan. "Underdamped Langevin MCMC: A non-asymptotic analysis". In: *Conference on Learning Theory (COLT).* 2018.

[17] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. "DeCAF: A deep convolutional activation feature for generic visual recognition". In: *International Conference on Machine Learning (ICML).* 2014.

[18] S. Frei, **N. Chatterji**, and P. Bartlett. "Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data". In: *Conference on Learning Theory (COLT).* 2022.

[19] S. Frei, **N. Chatterji**, and P. Bartlett. "Random feature amplification: Feature learning and generalization in neural networks". In: *arXiv preprint arXiv:2202.07626* (2022).

[20] S. Han, H. Mao, and W. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding". In: *arXiv preprint arXiv:1510.00149* (2015).

[21] J. Hoffmann et al. "Training compute-optimal large language models". In: *arXiv preprint arXiv:2203.15556* (2022).

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* (2017).

[23] Y. Ma, **N. Chatterji**, X. Cheng, N. Flammarion, P. Bartlett, and M. Jordan. "Is there an analog of Nesterov acceleration for gradient-based MCMC?" In: *Bernoulli* (2021).

[24] C. Robert and G. Casella. *Monte Carlo statistical methods.* Springer Science & Business Media, 2013.

[25] J. Rosenfeld, A. Rosenfeld, Y. Belinkov, and N. Shavit. "A constructive prediction of the generalization error across scales". In: *International Conference on Learning Representations (ICLR).* 2019.

[26] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. "CNN features off-the-shelf: An astounding baseline for recognition". In: *Computer Vision and Pattern Recognition (CVPR) Workshop.* 2014.

[27] Y. Song and S. Ermon. "Generative modeling by estimating gradients of the data distribution". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

[28] K. A. Wang, **N. Chatterji**, S. Haque, and T. Hashimoto. "Is importance weighting incompatible with interpolating classifiers?" In: *International Conference on Learning Representations (ICLR).* 2021.