Some Damaging Delusions of Deep Learning Practice (and How to Avoid Them)

Arun Kumar Supun Nakandala Yuhao Zhang

University of California, San Diego
{arunkk, snakanda, yuz870}@eng.ucsd.edu

1 CONTEXT FOR OUR TALK AND WORK

This is not a talk by "DL researchers." In fact, we are "outsiders" to the DL/ML world, having never published a full paper at NeurIPS, ICLR, ICML, or KDD! So, what do we have to say and why speak here? We are *data systems* researchers interested in helping "democratize" DL, specifically from the usability and scalability standpoints. This talk distills our *critical analysis* of some major issues we see in DL research and practice on these fronts.

Our public health use cases. We have been collaborating with public health researchers at UCSD for 3 years on building DL models to study the impact of sedentary behaviors on health. They have TB-scale labeled time series datasets from hip and/or wrist worn accelerometers from various cohorts, including breast cancer survivors and people in assisted living facilities. By building ML classifiers to identify sitting activities, they can analyze more cohorts. Their prior pipeline based on RandomForest and handcrafted features had a balanced accuracy of about 76%. We built CNN-LSTM models that raised it massively to 92%. Our models are now the state of the art for this task in their field, leading to 2 public health journal papers [5, 11] and our models being used by other researchers [4]. How did we accomplish this? We did not need fancy DL algorithmic chops–we used largely off-the-shelf architectures for time series data. Our secret sauce was enabling *high throughput model selection on large-scale data*. That is what this talk is about: the importance of model selection and systems issues in DL.

The Cerebro DL platform. A DL model's accuracy on a given dataset is controlled by tuning 3 key factors: data/feature representation, neural architecture, and hyperparameters; collectively we call these the model selection triple (MST) [6]. Tuning MSTs is crucial for DL accuracy. But due to our dataset's size, we needed to *scale out* to a cluster to reduce model building times. Alas, DL tools such as TensorFlow and PyTorch failed to scale well out of the box. Thus, we are building Cerebro, a new lightweight DL platform on top of such DL tools that efficiently scales DL model selection to *sharded data* on a cluster and optimizes for the *throughput* of trying many MSTs. We published about it at top data systems venues [7, 17, 18].

Based on all of our above experiences, we share some major delusions that we saw/see being pervasive in DL practice as a cautionary tale. We hope our insights on how to avoid these delusions will help DL researchers, practitioners, and systems builders alike.

2 MODELING-RELATED DELUSIONS

(1) Damaging Delusion: Model selection is not needed. ML theory teaches us that balancing bias, variance, and noise controls generalization. There is no free lunch. Tuning MSTs is *inevitable*. In our case, we had to tune time window sizes, sizes of the model's layers, and some training hyperparameters. Yet, we see a common delusion that proper model selection is not needed. Many DL users, even top researchers, skip even hyperparameter tuning! There is delusional propaganda that architecture tricks are "all you need" or more labeled data is all one needs. A new delusion is that "double descent" is a panacea: keep bloating models to avoid overfitting.

Figure 1 shows examples from our work that rebuts all the above delusions. A1 shows results for our sitting prediction task on a cohort of adults [11]. We try 2 values each for time windows, number of layers, learning rate, and L2 regularizer. We see 2 clear clusters and significant variability in both. As a different example, B1 and B2 show results on ImageNet for ResNet50 and VGG16, each with 2 learning rates, L2 regularizers, and batch sizes. B1 uses grid search. B2 uses the AutoML procedure HyperOpt. The conclusion is inescapable: *rigorous model selection is key to get good accuracy and not squander one's labeled data.* As for double descent, A2 shows results for top 13 of 24 models for the same task as A1 but on a cohort of children. We doubled architecture sizes but overfitting persists (quadrupling was similar). But training runtimes jumped many folds, slowing our research. *The interpolation regime is non-trivial to realize in practice and is often counterproductively expensive.*

How to Avoid It: Model selection-first thought processes. DL users *must put model selection first*, not make it an afterthought. Think of data/feature representations, architectures, and hyperparameters to tune. List all MSTs tried in DL papers, not cover them up. Use higher level APIs such as Keras and/or AutoML procedures to try *a set of MSTs at a time*, not ad hoc trials. Automate logging and tracking of all MSTs tried and their results. Maintain provenance over time. In Cerebro, we adopted all these principles and offer familiar APIs to make DL model selection more seamless [7].

2) Damaging Delusion: Full automation is a panacea. A new delusion is that AutoML, especially automated neural architecture search, is a panacea for model selection. But most DL users *cannot afford the high cost of such elaborate searches*, especially if marginal utility is low relative to manual tuning. Reinforcement learning and similar metaheuristics often *obfuscate the design process*, rendering them even less usable in most cases outside of Big Tech.

How to Avoid It: Intermittent human-in-the-loop. DL users need *better semi-automated tools*, bridging the spectrum of automation [6]. Make it easy to spawn sets of MSTs (including with AutoML), intervene to stop low-accuracy MSTs, and clone/modify high-accuracy MSTs over time. We are building such capabilities in an intermittent interface for Cerebro [8]. Such "dialogue with the

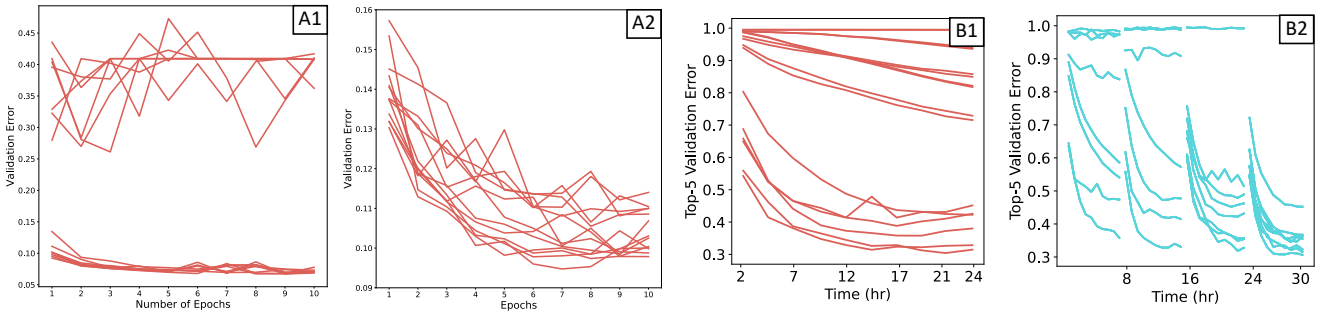

Figure 1: (A): Model selection results on our public health datasets. (B) Model selection results on the ImageNet benchmark.

algorithms" reins in DL's resource bloat, while combining both the ease of AutoML and the prudence of human intuition.

3) Damaging Delusion: Pre-trained models are a panacea. The rise of model hubs, HuggingFace library, etc. has led to a new delusion that pre-trained models have "solved" DL. But off-the-shelf models help only a few well-defined tasks. Most DL users in the real world have *bespoke tasks with custom semantics* and often multimodal data. They still need to train their own DL models.

How to Avoid It: Transfer learning as model selection. *View pre-trained models as seeds for task-specific models.* On images and text, one can extract layers of a pre-trained model as new feature representations. Entire architectures can sometimes be reused, with their weights finetuned. In many other cases, some layers can be frozen but others retrained, with some new layers added perhaps. All these are just new forms of MSTs. In Cerebro, we are supporting hierarchies of APIs overloading Keras-style syntax to make all these forms of transfer learning easier [12, 13].

3 SYSTEMS-RELATED DELUSIONS

1) Damaging Delusion: Almighty cloud is a panacea. The public cloud whales preach a "gospel of gluttony" urging DL users to use ever more GPUs, bigger GPUs, more memory, and more machines. But there is a conflict of interest here: the more resources enterprises and smaller Web firms rent on public clouds, the more of their money the whales pocket. Yet, many DL systems builders have a delusion that "scalability" just means "throw more machines at it," with low regard for resource efficiency and total costs.

How to Avoid It: Resource-efficiency matters. It is hard for DL users to control system efficiency. So, the responsibility falls on DL systems builders to *optimize for overall resource efficiency*, total costs, and energy use, not pursue ad hoc scaling. DL users must aim to adopt and push for such systems. Often, such holistically optimized systems yield lower runtimes too, which in turn helps DL users with accelerating their exploration. We are pursuing this design philosophy in Cerebro by re-imagining DL computations as new kinds of query models and devising a suite of novel query optimization techniques for DL systems [7]. This applies to both model building and model deployment, e.g., for explaining deep CNN inference results as we showed with Krypton [14–16].

2) Damaging Delusion: Data/task/model parallelism alone suffices to scale. Parallelism is necessary for high-throughput model selection. But most DL platforms today use *basic task parallelism*, which copies the full dataset to each worker and bloats the dataset's storage/memory footprints. For instance, on a 10-machine cluster our 1TB dataset will bloat to 10TB! While this may seem innocuous at small scale, it compounds to a massive headache at large scale. Yet, tools such as Ray, Google Vizier, Dask, Celery, and Determined suffer this pitfall. Reading data from remote storage every epoch mitigates this issue a bit but it will waste the network massively, often with 1000x redundant reads. *Sharded data parallelism* approaches, e.g., Horovod, Petuum, and PyTorch DDP offer more data scalability. But they all suffer from low throughput because they train only one model at a time, which often compound to over 10,000x network costs in model selection scenarios [17]. Finally, *model parallelism* approaches train one large DL model on multiple GPUs. But they offer low to no real speedups and also complete ignore the data scalability bottleneck. Overall, all these *one-dimensional system designs are too primitive*, resulting in bloated resource footprints and costs for large-scale DL.

How to Avoid It: Hybrid-parallel systems. DL systems builders must *end this false trichotomy* of task-, data-, and model-parallelism and *aim for hybrid parallelism schemes* that optimize overall resource efficiency. Use the bigger picture of the overarching model selection process, specified via the higher-level APIs, instead of only parallelizing one model at a time or copying whole datasets. In Cerebro, we are taking a first-principles approach to both scalability and parallelism in DL systems, inspired by time-tested lessons from research on RDBMSs, dataflow systems, and operations research.

We are devising a suite of novel hybrid parallelism schemes that enable true scalability along all major axes of interest in DL: dataset size, model size, number of models/tasks, number of workers, number of data sub-groups, and even data example size [7]. Three key recent examples include a new hybrid of task- and data-parallelism for model selection that substantially reduces resource footprints and/or runtimes against all prior data-scalable approaches [17], a new hybrid of task- and data-parallelism for more efficient bulk execution of model selection over sub-groups of data (an increasingly common practice) [9], and a new hybrid of task- and model-parallelism for model selection that outperforms all prior model-scalable approaches on both runtime and GPU utilization [10]. All

of our ideas are easy to integrate with existing DL tools without needing to modify their internal code, which can ease adoption. We have been prototyping Cerebro with support for both PyTorch and TensorFlow.

4 CONCLUSION AND FUTURE OF CEREBRO

We hope this conversation on the importance of model selection and system efficiency at scale is helpful to the DL world, especially practitioners. We have been infusing these lessons into Cerebro, a first-of-its-kind model selection platform for DL that aims for seamless scalability along all axes powered by multi-query optimization. Cerebro is fully open sourced [3] and also integrated with Apache Spark [1]. We invite all DL users to try it, and we welcome critical feedback. Some of our ideas have been adopted by Apache MADlib and shipped by VMware for enterprise customers [18]. Our ongoing research is tackling more bottlenecks in scalability, resource efficiency, and usability of DL. We welcome inquires by DL users in both domain sciences and at companies on any new scalability bottlenecks they face in practice. We also plan to support cloud-native, serverless, and budget-aware execution in due course and also add more high-level vertical-specific APIs [7]. Please monitor our project webpage for details [2]. Ultimately, we hope Cerebro bridges the worlds of DL and data systems to help truly democratize large-scale DL.

Speaker Bio. Arun Kumar is an Assistant Professor in the Department of Computer Science and Engineering and the Halicioglu Data Science Institute at the University of California, San Diego. His research interests are in data management and systems for ML-based data analytics, with a focus on scalability and usability.

ACKNOWLEDGMENTS.

This work was supported in part by an NSF CAREER Award under award number 1942724, the NIDDK of the NIH under award number R01DK114945, a Hellman Fellowship, and gifts from VMware. The content is solely the responsibility of the authors and does not necessarily represent the views of any of these organizations.

REFERENCES

[1] Cerebro integration with Apache Spark, Accessed July 9, 2021. https://databricks.com/session_na20/resource-efficient-deep-learning-model-selection-on-apache-spark/.
[2] Cerebro project webpage, Accessed July 9, 2021. https://adalabucsd.github.io/cerebro.html/.
[3] Cerebro system download and documentation, Accessed July 9, 2021. https://adalabucsd.github.io/cerebro-system/.
[4] Deep Postures repository on GitHub, Accessed July 9, 2021. https://github.com/ADALabUCSD/DeepPostures.
[5] M. A. Greenwood-Hickman, S. Nakandala, M. M. Jankowska, F. Tuz-Zahra, J. Bellettiere, J. Carlson, P. R. Hibbing, J. Zou, A. Z. LaCroix, A. Kumar, and L. Natarajan. The cnn hip accelerometer posture (chap) method for classifying sitting patterns from hip accelerometers: A validation study. *Medicine and Science in Sports and Exercise Journal*, 2021.
[6] A. Kumar, R. McCann, J. Naughton, and J. M. Patel. Model Selection Management Systems: The Next Frontier of Advanced Analytics. *SIGMOD Rec.*, 44(4):17–22, May 2016.
[7] A. Kumar, S. Nakandala, Y. Zhang, S. Li, A. Gemawat, and K. Nagrecha. Cerebro: A Layered Data Platform for Scalable Deep Learning. In *Proceedings of the 2021 Conference on Innovative Data Systems Research*, CIDR '21, 2021.
[8] L. Li, S. Nakandala, and A. Kumar. Intermittent Human-in-the-Loop Model Selection using Cerebro: A Demonstration. *Proc. VLDB Endow.*, 14, 2021.
[9] S. Li and A. Kumar. Towards an Optimized GROUP BY Abstraction for Large-Scale Machine Learning. *Proc. VLDB Endow.*, 14, 2021.
[10] K. Nagrecha and A. Kumar. Hydra: A Scalable and Optimized Data System for Large Multi-Model Deep Learning. *Technical Report*, 2021.
[11] S. Nakandala, M. M. Jankowska, F. Tuz-Zahra, J. Bellettiere, J. A. Carlson, A. Z. LaCroix, S. J. Hartman, D. E. Rosenberg, J. Zou, and A. Kumar. Application of convolutional neural network algorithms for advancing sedentary and activity bout classification. *Journal for the Measurement of Physical Behaviour*, 1(aop):1–9, 2021.
[12] S. Nakandala and A. Kumar. Vista: Optimized System for Declarative Feature Transfer from Deep CNNs at Scale. In *Proceedings of the 2020 International Conference on Management of Data*, SIGMOD '20, 2020.
[13] S. Nakandala and A. Kumar. Nautilus: An Optimized System for Deep Transfer Learning over Evolving Training Datasets. *Technical Report*, 2021.
[14] S. Nakandala, A. Kumar, and Y. Papakonstantinou. Incremental and Approximate Inference for Faster Occlusion-based Deep CNN Explanations. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, 2019.
[15] S. Nakandala, A. Kumar, and Y. Papakonstantinou. Query Optimization for Faster Deep CNN Explanations. *SIGMOD Rec.*, 49(1):61–68, 2020.
[16] S. Nakandala, K. Nagrecha, A. Kumar, and Y. Papakonstantinou. Incremental and Approximate Computations for Accelerating Deep CNN Inference. *ACM Trans. Database Syst.*, 45(4):16:1–16:42, 2020.
[17] S. Nakandala, Y. Zhang, and A. Kumar. Cerebro: A Data System for Optimized Deep Learning Model Selection. *Proc. VLDB Endow.*, 13(11):2159–2173, 2020.
[18] Y. Zhang, F. McQuillan, N. Jayaram, N. Kak, E. Khanna, O. Kislal, D. Valdano, and A. Kumar. Distributed Deep Learning on Data Systems: A Comparative Analysis of Approaches. *Proc. VLDB Endow.*, 14, 2021.