Ten Quick Tips for Deep Learning in Biology

This manuscript (permalink) was automatically generated from Benjamin-Lee/deep-rules@42f4353 on January 26, 2019.

Authors

- **Benjamin D. Lee**
  0000-0002-7133-8397 · Benjamin-Lee
  School of Engineering and Applied Sciences, Harvard University; Department of Genetics, Harvard Medical School; Lab41, In-Q-Tel

Introduction

Deep learning (DL), a subfield of machine learning (ML) implementing artificial neural networks with many layers, is increasingly used for the analysis of biological data [1]. Despite its growing popularity, DL itself remains an active area of research. Its everchanging complexity and lack of current beginner resources focused on biological applications pose large barriers of entry to newcomers who wish to utilize state-of-the-art DL in their research. Biological insight garnered from DL has been well-documented in the scientific literature, with applications ranging from predicting protein-drug binding kinetics [2] to identifying the lab-of-origin of synthetic DNA [3]. However, few resources articulate DL best practices to the scientific community. Most instructional literature focuses on ML broadly, rather than DL specifically, further limiting accessibility and reproducibility [4]. To address this issue, we solicited input from a diverse community of researchers, who wrote this manuscript collaboratively using the GitHub version control platform [5] and Manubot [6].

In the course of our discussions, several themes became clear: the importance of understanding and applying ML fundamentals as a baseline for utilizing DL, the necessity for extensive model comparisons and careful evaluation, and the need for critical thought in interpreting results generated by means of DL, among others. Ultmately, the tips we established range from high-level guidance to the implementation of best practices, and it is our hope that they will provide actionable, DL-specific advice for both new and experienced DL practitioners alike who would like to employ DL in biological research. By increasing the accessibility of DL techniques to biology, we aim to improve the overall quality and reproducibility of DL in the literature, enabling these powerful methods to be properly utilized to generate new scientific insights.

Tip 1: Concepts that apply to machine learning also apply to deep learning

Deep learning is a distinct subfield of machine learning, but it is still a subfield. Deep learning has proven to be an extremely powerful paradigm capable of outperforming “traditional” machine learning approaches, but it is not immune to the many limitations inherent to machine learning. Many best practices for machine learning apply to deep learning as well. For instance, deep supervised learning models should be trained, tuned, and tested on non-overlapping datasets. Those developing deep learning models should select data that are relevant to the problem at hand; non-salient data can hamper performance or lead to spurious conclusions. Furthermore, investigators should begin by thoroughly inspecting their data. When coupled with imprudence, data that is biased, skewed, or of low quality will produce models of dubious performance and limited generalizability. Biases in testing data can also unduly influence measures of model performance. For example, many conventional metrics for classification (e.g. area under the receiver operating characteristic curve or AUROC) have limited utility in cases of extreme class imbalance. As such, model performance should be evaluated with a carefully-picked panel of relevant metrics that make minimal assumptions about the composition of the testing data [7]. Extreme cases warrant testing the robustness of the model and metrics on simulated data for which the ground truth is known. Said simulations can be used to verify the correctness of the model’s implementation as well. Like all computational methods, deep learning should be leveraged in a systematic manner that is reproducible and rigorously tested.

Tip 2: Use traditional methods to establish performance baselines

Before diving into a fancy thousand-layer neural network, always implement at least a simple model to establish an adequate performance baseline. For example, researchers can build multinomial logistic regression or random forest models using the same software framework that is being used for DL and evaluate its classification performance. This approach will help researchers with assessing the complexity of the task at hand and debugging more complex DL architectures. The utility of these methods is evidenced by the recent development of hybrid models which combine DL and simpler models to improve robustness, interpretability, and confidence estimation [8,9]. Depending on the amount of available data and the type of tasks, DL models may not necessarily be the best performing one. As an illustration, the simple baseline models by Rajkomar et al. [10] achieved performance comparable with that of DL in a number of clinical prediction tasks using electronic health records, which may be a surprise to many.

It is worth noting that many conventional machine learning methods (e.g., support vector machines, random forests) require parameter tuning. Instead of assuming that DL is better than other machine learning methods, researchers should investigate whether the baseline models are rigorously fine-tuned. The performance comparison among DL models and many other ML approaches is informative only when the models are fairly compared.

Tip 3: Understand the complexities of training deep neural networks

Tip 4: Know your data and your question

In the era of easily accessible datasets, one sometimes starts analyzing data without a good understanding of the study design, namely why the data were collected and how. Having appropriate meta-data, a comprehensive data dictionary, and even the actual data collection protocol is essential for any analysis, including one that involves deep learning. It's even better to also have access to a subject matter expert who has either collected or analyzed this type of data before! For example, if the main reason why the data were collected was to test the impact of an intervention, it may be the case that a randomized controlled trial was performed. However, ethical or other study considerations may make this impossible or impractical, in which case the design may have been an observational one, either prospective or retrospective. These designs may also incorporate some amount of matching - for example, cases and controls may be selected so that the age range or weight distribution is similar. All of these different designs have different assumptions and caveats, which cannot be ignored during a data analysis. Many datasets are now passively collected or do not have a specific design, but even in this case it is important to know how individuals or samples were treated (for example, if all samples are from the same study site, if certain ethnic groups or zip codes are oversampled, if there are differences in processing dates or techniques.)

Systematic biases can lead to artifacts or "batch effects," which mean that instead of finding correlates with an outcome or grouping of interest, the investigator may find correlates with variables that are not of interest and obtain misleading results [11]. Other study design considerations that should not be overlooked include knowing whether a study involves biological or technical replicates or both. For example, are some samples collected from the same individuals at different time points? Are those time points before and after some treatment? If one assumes that all the samples are independent but that is in fact not the case, a variety of issues may arise, including having a lower effective sample size than expected.

In general, deep learning has an increased tendency for overfitting, compared to classical methods, due to the large number of parameters being estimated, making issues of adequate

sample size even more important (see Tip 7). For a large dataset, overfitting may not be a concern, but the modeling power of deep learning may lead to more spurious correlations and thus incorrect interpretation of results (see Tip 9). Finally, it is important to note that with the exception of very specific cases of unsupervised data analysis, it is generally the case that a molecular or imaging dataset does not have much value without appropriate clinical or demographic data; this must always be balanced with the need to protect patient privacy (see Tip 10). Looking at these data can also clarify the study design (for example, by seeing if all the individuals are adolescents or women) or at least help the analyst employing deep learning to know what questions to ask.

Tip 5: Choose an appropriate neural network architecture and data representation

Tip 6: Expect to tune hyperparameters extensively and systematically

Deep neural networks have the ability to approximate arbitrary continuous functions, as long as the neural network contains enough hidden nodes [12]. However, this flexibility makes the training process somewhat challenging. Users should expect to systematically evaluate the impact of numerous hyperparameters when they aim to apply deep neural networks to new data or challenges.

Neural network architectures also have their own odd nuances that affect hyperparameter portability. For example, in variational autoencoders (VAEs) there are two elements that are being optimized, reconstruction and distribution loss [13]. In common implementations, the relative weights of each are a function of the number of input features (more increase the importance of reconstruction loss) and the number of features in the latent space (more increase the importance of the distribution loss). Users who apply a VAE architecture to a new dataset with more input features, even without changing any hyperparameters, alter the relative weights of the components of the loss function.

This flexibility also makes it difficult to evaluate the extent to which neural network methods are well-suited to solving a task. Hu and Greene [14] discuss a Continental Breakfast Included (CBI) effect by which unequal hyperparameter tuning skews the evaluation of methods, especially those with performance that varies substantially with modest changes to hyperparameters. The implication of CBI on methods developers is discussed more in Tip 2 (`TODO: cgreene tie these together`). The implication of CBI on users of deep neural networks is that attaining performance numbers that match those reported in publications is likely to require an input of human and compute time for hyperparameter optimization.

Tip 7: Address deep neural networks' increased tendency to overfit the dataset

Overfitting is one of the most significant dangers faced by a deep learning practitioner. Put simply, overfitting occurs when a model fits patterns in the training data too closely, includes noise or non-scientifically relevant perturbations, or in the most extreme case, simply memorizes patterns in the training set. This subtle distinction is made clearer by seeing what happens when a model is tested on data to which it was not exposed during training: just as a student who memorizes exam materials struggles to correctly answer questions for which they have not studied, a machine learning model that has overfit to its training data will perform poorly on unseen test data. Deep learning models are particularly susceptible to overfitting due to their relatively large number of parameters and associated representational capacity. To continue the student analogy, a smarter student has greater potential for memorization than average one and thus may be more inclined to memorize.

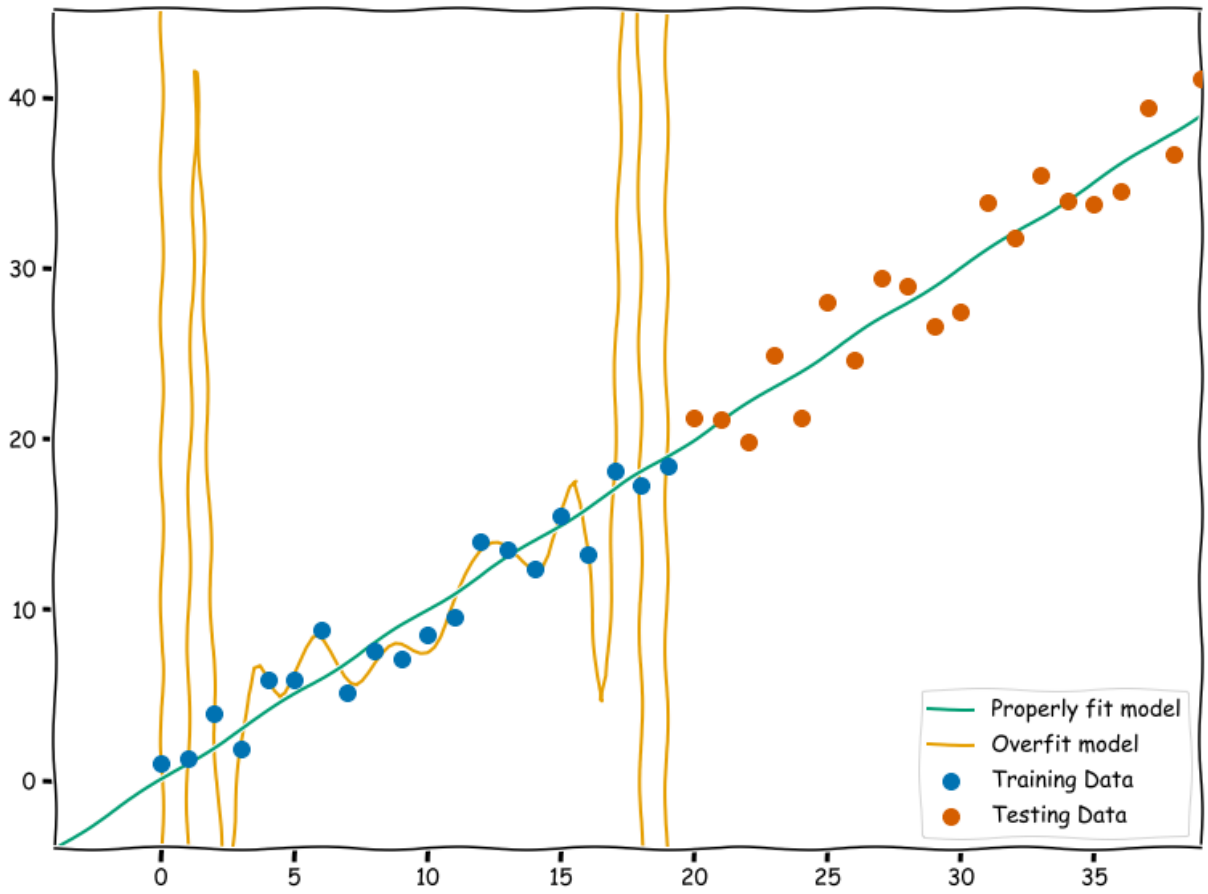

Figure 1: A visual example of overfitting. While a high-degree polynomial gets high accuracy on its training data, it performs poorly on data that is has not seen before, whereas a simple linear regression works well. The greater representational capacity of the polynomial is analogous to using a larger or deeper neural network.

The simplest way to combat overfitting is to detect it. This can be done by splitting the dataset into three parts: a training set, a tuning set (also commonly called a validation set in the machine learning literature), and a test set. By exposing the model solely to the training data during fitting, a researcher can use the model's performance on the unseen test data to measure the amount of overfitting. While a slight drop in performance from the training set to the test set is normal, a significant drop is a clear sign of overfitting (see Figure 1 for a visual demonstration of an overfit model that performs poorly on test data). Additionally, there are a variety of techniques to reduce overfitting during training including data augmentation and regularization techniques such as dropout [15] and weight decay [15]. Another way, as described by Chuang and Keiser, is to identify the baseline level of memorization of the network by training on the data with the labels randomly shuffled and to see if the model performs better on the actual data [16]. If the model performs no better on real data than randomly scrambled data, then the performance of the model can be attributed to overfitting.

Additionally, one must be sure that their data are not skewed or biased, such as by having confounding and scientifically irrelevant variables that the model can pick up on [17]. In this case, simply holding out test data is insufficient. The best remedy for confounding variables is to know your data and to test your model on truly independent data.

Tip 8: Do not necessarily consider a DL model as a black box

Tip 9: Don't over-interpret predictions

Deep learning models can make predictions with high accuracy, but we need to take care to correctly interpret these predictions. We know that the basic tenets of machine learning also apply to deep learning (Tip 1), but because deep models can be difficult to interpret intuitively, there is a temptation to anthropomorphize deep models. We must resist this temptation.

A common saying in statistics classes is "correlation doesn't imply causality". While we know that accurately predicting an outcome doesn't imply learning the causal mechanism, it can be easy to forget this lesson when the predictions are extremely accurate. A poignant example of this lesson is [18,19]. In this study, the authors evaluated the capacities of several models to predict the probability of death for patients admitted to an intensive care unit with pneumonia. Unsurprisingly, the neural network model achieved the best predictive accuracy. However, after fitting a rule-based model, the authors discovered that the hospital data implied the rule "HasAsthma(x) => LowerRisk(x)". This rule contradicts medical understanding - having asthma doesn't make pneumonia better! This rule was supported by the data (pneumonia patients with a history of pneumonia tended to receive more aggressive care), so the neural network also learned to make

predictions according to this rule. Guiding treatment decisions according to the predictions of the neural network would have been disastrous, even though the neural network had high predictive accuracy.

To trust the reasoning and scientific conclusions of deep learning models, combine knowledge of the data (Tip 4) with inspection of the model (Tip 8).

Tip 10: Don’t share models trained on sensitive data

One of the greatest opportunities for deep learning in biology is the ability for deep learning techniques to incorporate representation learning to extract information that can not readily be captured by traditional methods [20]. The abundance of features for each training example means that the representation learning of the deep learning models can capture information-rich abstractions of data during the training process. Therefore with both deep learning and traditional machine learning models (*e.g. k*-nearest neighbors models, which learn by memorizing the full training data), it is imperative not to share models trained on sensitive data. Applying deep learning to images of cats from the internet does not pose significant ethical, legal, or privacy problems; this is not the case when dealing with classified, confidential, trade secret, or other types of biological data that cannot be shared. For example, adversarial training techniques such as model inversion attacks can be used to exploit model predictions to recover recognizable images of people’s faces used for training [21]. These risks are even more significant in deep learning compared to traditional machine learning techniques due to the greater representational capacity of the models. This is achieved by the large number of model weights, even in a relatively small project, that allow deep learning to model high-dimensional non-linear relationships among data. It is this enhanced modeling capacity that allows the model to learn more robust and nuanced features of specific data, leading to the danger of revealing the underlying sensitive data. When training deep learning models on sensitive data, be sure not to share the model weights directly, and use privacy preserving techniques [22] such as differential privacy [23,24] and homomorphic encryption [25,26] to protect sensitive data.

Conclusion

References

1. **Opportunities and obstacles for deep learning in biology and medicine**
Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, … Casey S. Greene
Journal of The Royal Society Interface (2018-04) https://doi.org/gddkhn
DOI: 10.1098/rsif.2017.0387 · PMID: 29618526 · PMCID: PMC5938574

2. **VAMPnets for deep learning of molecular kinetics**
Andreas Mardt, Luca Pasquali, Hao Wu, Frank Noé
Nature Communications (2018-01-02) https://doi.org/gcvf62
DOI: 10.1038/s41467-017-02388-1 · PMID: 29295994 · PMCID: PMC5750224

3. **Deep learning to predict the lab-of-origin of engineered DNA**
Alec A. K. Nielsen, Christopher A. Voigt
Nature Communications (2018-08-07) https://doi.org/gd27sw
DOI: 10.1038/s41467-018-05378-z · PMID: 30087331 · PMCID: PMC6081423

4. **Ten quick tips for machine learning in computational biology**
Davide Chicco
BioData Mining (2017-12) https://doi.org/gdb9wr
DOI: 10.1186/s13040-017-0155-3 · PMID: 29234465 · PMCID: PMC5721660

5. **Ten Quick Tips for Deep Learning in Biology. Contribute to Benjamin-Lee/deep-rules development by creating an account on GitHub**
Benjamin Lee
(2019-01-26) https://github.com/Benjamin-Lee/deep-rules

6. **Open collaborative writing with Manubot**
Daniel S. Himmelstein, David R. Slochower, Venkat S. Malladi, Casey S. Greene, Anthony Gitter
(2018-12-31) https://greenelab.github.io/meta-review/

7. **Comparison of Deep Learning With Multiple Machine Learning Methods and Metrics Using Diverse Drug Discovery Data Sets**
Alexandru Korotcov, Valery Tkachenko, Daniel P. Russo, Sean Ekins
Molecular Pharmaceutics (2017-11-13) https://doi.org/gcj4p2
DOI: 10.1021/acs.molpharmaceut.7b00578 · PMID: 29096442 · PMCID: PMC5741413

8. **Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning**
Nicolas Papernot, Patrick McDaniel
arXiv (2018-03-13) https://arxiv.org/abs/1803.04765v1

9. **To Trust Or Not To Trust A Classifier**
Heinrich Jiang, Been Kim, Melody Y. Guan, Maya Gupta
arXiv (2018-05-30) https://arxiv.org/abs/1805.11783v2

10. **Scalable and accurate deep learning with electronic health records**
Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M. Dai, Nissan Hajaj, Michaela Hardt, Peter J. Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, … Jeffrey Dean
npj Digital Medicine (2018-05-08) https://doi.org/gdqcc8
DOI: 10.1038/s41746-018-0029-1

11. **Tackling the widespread and critical impact of batch effects in high-throughput data**
Jeffrey T. Leek, Robert B. Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W. Evan Johnson, Donald Geman, Keith Baggerly, Rafael A. Irizarry
Nature Reviews Genetics (2010-09-14) https://doi.org/cfr324
DOI: 10.1038/nrg2825 · PMID: 20838408 · PMCID: PMC3880143

12. **Approximation capabilities of multilayer feedforward networks**
Kurt Hornik
Neural Networks (1991) https://doi.org/dzwxkd
DOI: 10.1016/0893-6080(91)90009-t

13. **Auto-Encoding Variational Bayes**
Diederik P Kingma, Max Welling
arXiv (2013-12-20) https://arxiv.org/abs/1312.6114v10

14. **Parameter tuning is a key part of dimensionality reduction via deep variational autoencoders for single cell RNA transcriptomics**
Qiwen Hu, Casey S Greene
Cold Spring Harbor Laboratory (2018-08-05) https://doi.org/gdxxjf
DOI: 10.1101/385534

15. http://dl.acm.org/citation.cfm?id

16. **Adversarial Controls for Scientific Machine Learning**
Kangway V. Chuang, Michael J. Keiser
ACS Chemical Biology (2018-10-19) https://doi.org/gfk9mh
DOI: 10.1021/acschembio.8b00881 · PMID: 30336670

17. **Confounding variables can degrade generalization performance of radiological deep learning models**
John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, Eric K. Oermann
arXiv (2018-07-02) https://arxiv.org/abs/1807.00431v2

18. **An evaluation of machine-learning methods for predicting pneumonia mortality**
Gregory F. Cooper, Constantin F. Aliferis, Richard Ambrosino, John Aronis, Bruce G. Buchanan, Richard Caruana, Michael J. Fine, Clark Glymour, Geoffrey Gordon, Barbara H. Hanusa, … Peter Spirtes
Artificial Intelligence in Medicine (1997-02) https://doi.org/b6vnmd
DOI: 10.1016/s0933-3657(96)00367-3

19. **Intelligible Models for HealthCare**
Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, Noemie Elhadad
Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15 (2015) https://doi.org/gftgxk
DOI: 10.1145/2783258.2788613

20. **Convolutional Networks on Graphs for Learning Molecular Fingerprints**
David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, Ryan P. Adams
arXiv (2015-09-30) https://arxiv.org/abs/1509.09292v2

21. **Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures**
Matt Fredrikson, Somesh Jha, Thomas Ristenpart
Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15 (2015) https://doi.org/cwdm
DOI: 10.1145/2810103.2813677

22. **A generic framework for privacy preserving deep learning**
Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, Jonathan Passerat-Palmbach
arXiv (2018-11-09) https://arxiv.org/abs/1811.04017v2

23. **Deep Learning with Differential Privacy**
Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, Li Zhang
Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16 (2016) https://doi.org/gcrnp3
DOI: 10.1145/2976749.2978318

24. **Privacy-Preserving Distributed Deep Learning for Clinical Data**
Brett K. Beaulieu-Jones, William Yuan, Samuel G. Finlayson, Zhiwei Steven Wu
arXiv (2018-12-04) https://arxiv.org/abs/1812.01484v1

25. **SIG-DB: Leveraging homomorphic encryption to securely interrogate privately held genomic databases**
Alexander J. Titus, Audrey Flower, Patrick Hagerty, Paul Gamble, Charlie Lewis, Todd Stavish, Kevin P. O'Connell, Greg Shipley, Stephanie M. Rogers
PLOS Computational Biology (2018-09-04) https://doi.org/gd6xd5
DOI: 10.1371/journal.pcbi.1006454 · PMID: 30180163 · PMCID: PMC6138421

26. **The AlexNet Moment for Homomorphic Encryption: HCNN, the First Homomorphic CNN on Encrypted Data with GPUs**
Ahmad Al Badawi, Jin Chao, Jie Lin, Chan Fook Mun, Sim Jun Jie, Benjamin Hong Meng Tan, Xiao Nan, Khin Mi Mi Aung, Vijay Ramaseshan Chandrasekhar
arXiv (2018-11-02) https://arxiv.org/abs/1811.00778v1