Test Set Verification Is An Essential Step in Model Building

Thomas P. Quinn[1*], Vuong Le[1], and Adam P. A. Cardilini[2]

[1]Applied Artificial Intelligence Institute, Deakin University, Geelong, Australia
[2]School of Life and Environmental Sciences, Deakin University, Geelong, Australia
* *contacttomquinn@gmail.com*

Abstract

1. Recently, Christin et al. published an article that reviewed the field of deep learning and offered advice on how to train a deep learning model.

2. We write here to emphasize the importance of model verification, which can help ensure that the model will generalize to new data.

3. Specifically, we discuss the importance of using a test set for model verification, and of defining an explicit research hypothesis.

4. We then present a revised workflow that will help ensure that the accuracy reported for your deep learning model is reliable.

Remarks

Deep learning is a valuable tool for solving many problems that face life science researchers today. The field has quickly earned the keen interest of scientists from many disciplines, and the popularity of deep learning will likely grow for some time. Articles like the one presented by Christin et al. play an important role in encouraging domain experts to pursue deep learning [1]. We write here to emphasize the importance of model verification, which can help ensure that the model will generalize to new data. We feel that even the simplest deep learning workflows should include model verification for two (related) reasons:

The first is technical: Any model trained without verification is unreliable for use with real-world data. This is because iterative model tuning can overfit the validation subset (which is actually *part* of the training data). Although it is often necessary to try several models before finding the best one, the goal of deep learning is not only to develop a model that works well on the validation subset. Rather, the ultimate goal of deep learning (or any machine learning method) is to develop a model that works well for completely new real-world data. Once model tuning begins, the validation subset no longer resembles completely new data: information from the validation subset has begun to "leak" into the training procedure, and the model may become overfit to the validation subset. It is helpful to think about this "data leakage" as akin to a student having access to an exam and memorizing the answers instead of learning the material. In other words, "data leakage" can cause a model to memorize particular patterns that are specific to the validation subset instead of learning general concepts that work well for unseen data sets. The more that the analyst tunes their model, the more the model might "memorize" the answers, and thus the more unreliable the validation subset accuracy becomes. By repeatedly tuning the model, one could unknowingly train an overfit model that works well for the validation subset but fails to generalize to new data.

The second is conceptual: Machine learning is not only a tool that enables experimental science, but is also an experimental science itself [2]. When developing a new deep learning model in ecology, the researcher performs a methodological experiment that is nested within a larger ecological experiment. As such, model accuracy is not only measured to evaluate the reliability of a model; it is also measured to test an explicit methodological hypothesis that follows from a research question [3]. This hypothesis should specify the model architecture, the available data, and the intended application. For example, the hypothesis might state *A CNN with VGG architecture can reliably classify photos of field-collected passerine bird nests, after training on 2500 photos of bird nests provided by the Natural History Museum.* In the context of model verification, thinking about big accuracies as "good" is akin to thinking about small p-values as "good": these metrics, big or small, are all important because they help evaluate the hypothesis being examined. If the hypothesis is confirmed, researchers may then use this model as a tool to classify their field-collected passerine bird nest images. However, deep learning is not a panacea: even with the optimal model, some data sets will never result in useful predictions.

The solution to both problems is to use a test set for model verification [4]. Model verification will (1) help ensure that the model can be trusted on unseen data, and (2) allow the researcher to evaluate the methodological hypothesis. Ideally, this test set is collected separately from the training set. When a separate test set is not available, the practitioner should set aside a test set from the available data, then split the remaining training samples into training and validation subsets. Either way, the test set should represent the intended use case and is only used *once* to verify a model at the end of the workflow. For a model trained to classify field-collected passerine bird nests, the test set should contain field-collected passerine bird nests (regardless of the training data used). Once accuracy is measured for the test set, the workflow is over and the methodological hypothesis is evaluated. If the model performance on the test set is not sufficient for use, then the hypothesis is rejected. It is invalid to re-tune the model based on the test set results, or else the test set starts "leaking" into the training procedure too. In other words, if you use the test set to tune the model, it is no longer a test set; it is now part of the training set! In fact, it is becoming more common to use single-blinded test set verification, where an external party measures the accuracy of the final model.

It is worth noting that cross-validation is a popular way of dividing the data, and could be used to carve out a validation subset *or* a test set. When using cross-validation, the model must still be verified on at least one test set that is totally separate from the validation subset(s). Importantly, cross-validation is not an alternative to test set verification, though it could be used to facilitate test set verification. Either way, the analyst must perform model verification, especially if they wish to tune their model. We include here an amended flowchart that illustrates model verification using a test set. Following this updated workflow will help ensure that the accuracy reported for your deep learning model is reliable.

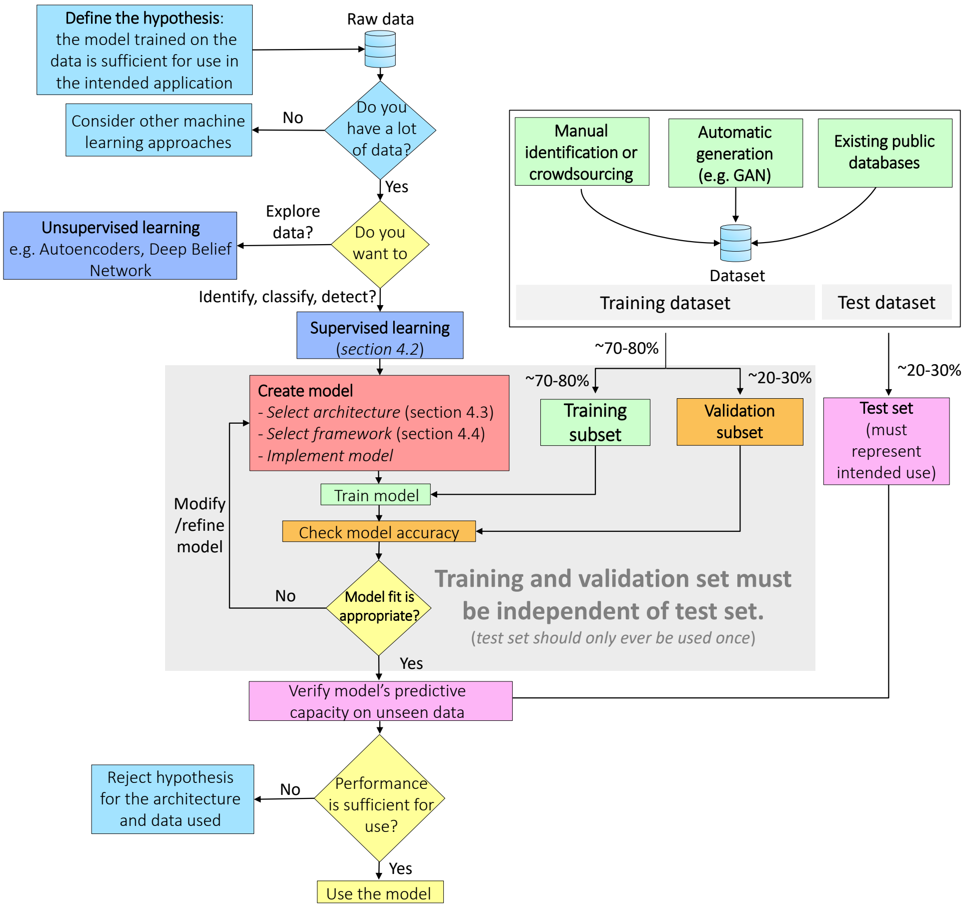
Define the hypothesis: the model trained on the data is sufficient for use in the intended application
Raw data
Do you have a lot of data?
No
Consider other machine learning approaches
Yes
Explore data?
Do you want to
Unsupervised learning e.g. Autoencoders, Deep Belief Network
Identify, classify, detect?
Supervised learning (section 4.2)
Manual identification or crowdsourcing
Automatic generation (e.g. GAN)
Existing public databases
Dataset
Training dataset
Test dataset
~70-80%
~70-80%
~20-30%
~20-30%
Training subset
Validation subset
Test set (must represent intended use)
Create model
- Select architecture (section 4.3)
- Select framework (section 4.4)
- Implement model
Train model
Check model accuracy
Modify /refine model
No
Model fit is appropriate?
Yes
Training and validation set must be independent of test set.
(test set should only ever be used once)
Verify model's predictive capacity on unseen data
Reject hypothesis for the architecture and data used
No
Performance is sufficient for use?
Yes
Use the model

Figure 1: This figure presents an updated flowchart, derived from the original publication, with a few changes. First, the practitioner should begin by defining a methodological hypothesis that follows from a research question. This hypothesis should specify the model architecture, the available data, and the intended application. For example, the hypothesis might state *A CNN with VGG architecture can reliably classify photos of field-collected passerine bird nests, after training on 2500 photos of bird nests provided by the Natural History Museum.* Second, the practitioner should build a test set that is independent from the training data. The test data should represent the intended application. In the non-ideal case that the test data must be carved out from the training data, the analyst should take care to minimize bias with respect to potential confounders. Third, the practitioner should use test set verification. In test set verification, the performance of the tuned model is measured for new real-world data once and only once, allowing the analyst to evaluate their hypothesis. Our updated flowchart also recognizes the possibility that the tuned model does not work. In this case, the explicit hypothesis is rejected.

1 Data Availability

This article does not use any data.

2 Author Contributions

TPQ, VL, and APAC conceptualized the work. TPQ prepared a first draft. APAC updated the flowchart. All authors contributed to the final article.

References

[1] Sylvain Christin, Éric Hervet, and Nicolas Lecomte. Applications for deep learning in ecology. *Methods in Ecology and Evolution*, 10(10):1632–1644, 2019.

[2] Pat Langley. Machine Learning as an Experimental Science. *Machine Learning*, 3(1):5–8, August 1988.

[3] Leo Breiman. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199–231, August 2001.

[4] Davide Chicco. Ten quick tips for machine learning in computational biology. *BioData Mining*, 10(1):35, December 2017.