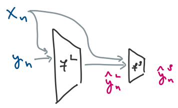
Week 9 - Distillation

Wednesday, March 19, 2025 7:13 PM

Origins

1. Distillation is the process of training a simple model (the "student") to mimic the predictions from a complex one ("teacher"). Specifically, suppose we are in a classification setting. Then, instead of training the simple model using the true labels as the response, distillation trains the simple model using the complex model's predicted probabilities as the response. Surprisingly, this can do better than training using the true labels, and it sometimes approaches the complex model's accuracy.



2. This idea was first developed at UW-Madison (Craven 1996), already for the purpose of explaining neural network models! But the idea was perhaps ahead of its time, and it would take about 20 years for it to sweep the machine learning community.

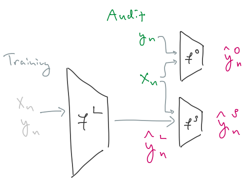
3. The catalyst for the renaissance was of course the arrival of deep learning, especially an influential paper that applied distillation to model compression (Hinton et al. 2015). Since student models have many fewer parameters than the original, they made it possible to use deep learning models on cell phones. It has also been useful for semi-supervised settings, since it only needs the teacher's predictions, not ground truth labels. Much more recently, it has been at the center of the OpenAI-Deepseek feud…

Distill-and-Compare

1. Most of the techniques we've explored in this class require access to the model that we want to explain. Even model agnostic, post-hoc techniques, like Shapley values, require us to evaluate the model on specifically chosen sets of input values. But this is not always possible.

2. Remember that one of our motivating examples was a ProPublica article that detected racial bias in the COMPAS criminal recidivism prediction model. This model is not public! Not only do we not know what kind of model it is, we can't even query it on any new examples. The only way the journalists did their analysis was by submitting a request to obtain past scores on real people.

3. Despite this constraint, we can still apply distillation! We treat the black box as a "teacher" and train an interpretable model as the "student." Further, by comparing the differences between the student model and the equivalent model trained from scratch, we can discover potential discrepancies between the data we have access to ("audit data") and the data used to train the black box. These were the main ideas of Tan et al. (2018). Let's review the details.



4. The interpretable student model is trained to mimic the black box teacher model using this loss:

$$\frac{1}{N}\sum_{n}^{N}\left(\hat{y}_n^{L} - f^{S}(x_n)\right)^2$$

The equivalent "outcome" model without access to the teacher is trained using:

$$\frac{1}{N}\sum_{n}^{N}\left(y_n - f^{O}(x_n)\right)^2$$

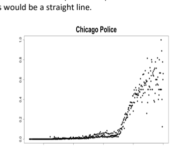
For example, in the reference, the simple model is an iGAM model, which can only ever look at one and two-way interactions:

$$x_n = [x_{n1}, \ldots, x_{nJ}]$$

$$f(x_n) = \sum_j h_j(x_{nj}) + \sum_{j \neq j'} h_{j,j'}(x_{nj}, x_{nj'})$$

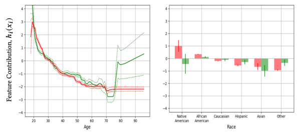
Both of these models are trained on an audit dataset. This can be whatever we have black box predictions for, and it need not be the training data.

5. One issue is that the complex model's probabilities might not be calibrated. In this case, it is better to calibrate these before training the student model. In the figure below, the x-axis is the risk score, and the y-axis is the fraction of examples in the positive class. If the risk were a well-calibrated probability, this would be a straight line.



6. The student model's components might be of independent interest, especially if it is able to accurately emulate the teacher's predictions. However, we need to be careful that there might be several interpretable models that all achieve similar performance, and they might be telling different stories about which features are important.

7. If we're willing to trust the trained student model, then it can also be helpful to compare the student with the outcome model. There may be individual features that have different effects in the student and outcome models. This can speak to differences in either (i) the data used to train the black box vs. the outcome model or (ii) the inductive biases of the black box. In the figure below, red is the student of the COMPAS model, and green is the outcome model.



8. The authors also claim that distillation lets us test for missing features. The argument is a bit heuristic, considering the other factors that could lead to differences in performance. The main idea is that, if there are many samples where both (i) the outcome model struggles to predict the ground truth and (ii) the student model struggles to emulate the black box, then perhaps there are features present in the original black box model training that aren't present in the audit data. This would explain why the black box was able to get better performance on those samples.

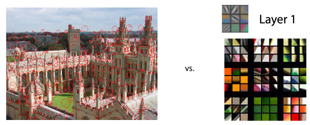
Table 1: Statistical test for likelihood of audit data missing key features used by black-box model.

Risk Score	Pearson ρ	Spearman ρ	Kendall τ
COMPAS	[0.10, 0.13]	[0.10, 0.14]	[0.08, 0.10]
Lending Club	[0.00, 0.03]	[-0.01, 0.01]	[-0.01, 0.01]
Stop-and-Frisk	[0.00, 0.01]	[-0.03, 0.01]	[-0.02, 0.01]
Chicago Police	[0.00, 0.01]	[0.01, 0.03]	[0.01, 0.02]

Distilling Features

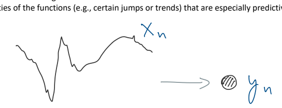
1. Distilling a black box's predictions is what most people mean when they talk about distillation. But there is a second kind of distillation that's worth knowing about: It's possible distill interpretable *features* from a complex model in a way that supports downstream applications.

2. To understand why this is a big deal, consider that one of the key characteristics of deep learning is that it performs representation learning more or less automatically. For example, in the network dissection notes, we saw that an image classification model could learn texture detectors simply by being trained to minimize classification error. Before deep learning, people in computer vision used to develop these feature extractors manually, e.g., the SIFT extractor (left panel below). Not only does deep learning bypass this labor-intensive feature engineering step, the automatically learned features are often more accurate.



3. The dream of distilling features then is to reverse engineer these predictive, automatically learned features, give them some clear interpretation, and then incorporate them into simple models. We'll see this in much more generally when we discuss mechanistic interpretability. For now, we'll focus one specific instance of distilling features: Adaptive Wavelet Distillation.

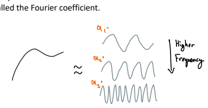
4. To motivate the idea, consider their clathrin fluoresence case study. This is a regression problem useful to analyze biological imaging data. The images are first preprocessed into short functional data, and the task is to predict a continuous response from these functions. It's possible to get good predictions using LSTMs. We want to use distillation to see whether there are certain properties of the functions (e.g., certain jumps or trends) that are especially predictive.



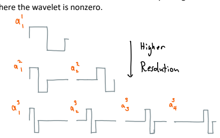
Wavelets Crash Course

1. Before we can see how Ha et al. (2021) solve this, we need to take a detour to the world of wavelets.

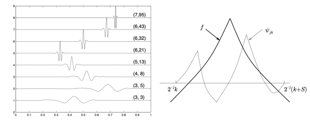
2. Even if you were not a math or engineering major, you have probably heard about Fourier series. The basic idea of Fourier series is that any periodic function can be decomposed into a mixture of sines and cosines with different frequencies. The sines and cosines serve as basis functions; the relative weight of each basis element is called the Fourier coefficient.



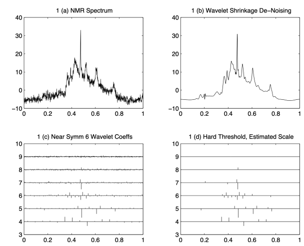
3. One difficulty with the Fourier basis is that the basis elements are global -- changing the coefficient for one basis element changes the resulting function everywhere in the input space. Wavelets are an alternative basis with many of the properties of the Fourier basis (e.g., they are orthonormal), but which are local. Changing the coefficient of one wavelet basis element only changes the value of the final function in the region where the wavelet is nonzero.



4. Wavelets are defined at several resolutions, meaning that some basis elements are nonzero over large regions of the input space, while others capture are only nonzero in very small regions. This makes them especially effective for modeling sharp jumps. The figure below is from Johnstone (2011). The function f on the right can be approximated by mixing basis wavelets like those on the left.



5. A classical application of wavelets is in denoising. By setting many of the small wavelet coefficients to zero, we can capture the essential shape of the function, even in places with sharp jumps. Each row corresponds to one resolution level, and each bar corresponds to one wavelet coefficient.

   i. 

6. There are many different valid wavelets. It's beyond our scope, but the wavelet decomposition can be implemented via a series of low and high-pass filters. This is what allows AWD to learn different wavelets by optimizing over candidate filter banks.

Adaptive Wavelet Distillation

1. The main idea of Ha et al. (2019) is that (i) we can use a deep learning model to guide the design of a wavelet basis and (ii) by setting most of the small coefficients to zero in a way that preserves prediction performance, we can identify the essential learned features.

2. For example, in the clathrin fluoresence case study, we don't simply denoise the functions in some model agnostic way -- we deliberately strip away all elements of the functions that are not being used by the deep learning model. The few remaining wavelet coefficients can also be used to derive a new, simple representation of the original functional data. These can be plugged into linear models. Surprisingly, these exceed performance of the original deep learning in all the examples they examined (though, check out the OpenReview reviews to see some comments about why they should have included examples where the performance deteriorated).

3. Let x[n] be the n^th input, and say f is the prediction of the original LSTM. The intuition above is encoded into different components of a particular optimization over filter banks h, g:

$$\underset{h,g}{\text{minimize}} \quad \mathrm{I} + \mathrm{II} + \mathrm{III}$$

The term (I) ensures that the "stripped away" input series doesn't look too different from the original:

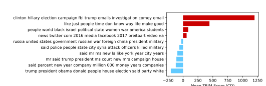
$$\mathrm{I} = \sum_n \left\| x_n - \text{Denoise}(x_n) \right\|_2^2$$

The term (II) is quite messy, but it just ensures that the learned filters induce a valid wavelet basis; e.g., it softly enforces an orthonormality constraint. The term (III) is key:

$$\mathrm{III} = \sum_n \left\| \text{TRIM}_f(\psi_{x_n}) \right\|_1$$

Here, TRIM is the gradient of the LSTM's response with respect to changes in each wavelet basis coefficient. While traditional wavelet denoising simply sets small coefficients to zero, AWD will only set coefficients to zero if they have small TRIM scores.

4. To get some intuition about TRIM, consider this figure from the paper that introduced the method. Instead of computing attribution scores for individual words in a document, it computes attribution scores for the topics that are present in that document. Since topics included much more context than the words, the attributions are more interpretable. Instead of topics, we can apply TRIM to the wavelet coefficients. Abstractly these are the same thing -- we compute an attribution on an alternative representation of the original sample.



5. The key interpretation in the clathrin case study is:

   AWD allows for checking what clathrin signatures [are predictive]; it indicates that the distilled wavelet aims to identify a large buildup in clathrin fluoresence… followed by a sharp drop

   Unfortunately, they only visualize the wavelet filters, and not the reconstructed signals.

6. While I have described this approach in the specific LSTM application, it could in principle apply to any domain where both deep learning and wavelets are used. For example, the authors consider an astronomical image application, and the reviewers point out that they could also be used on graph data (where graph signal processing gives the analog of classical wavelets).