Pre-read: computer vision and deep neural networks

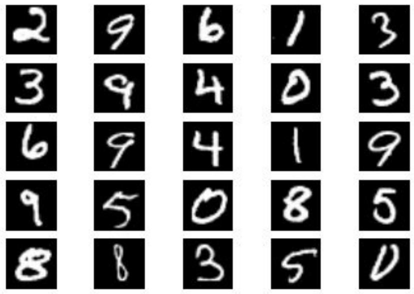
COS 597E/SOC 555, Princeton University, Fall 2020

A common reason for overoptimism about the capabilities of machine learning and deep neural networks is the recent progress in computer vision. This progress is genuine, and includes roughly human-level performance in tasks such as image classification — although this is highly dependent on how we set up the comparison — and arguably superhuman performance in some tasks involving generative modeling (example).

Our goal this week is to demystify computer vision, or at least image classification. Most importantly, we will identify aspects of the problem domain that have allowed deep learning to excel, and appreciate why these aspects don't apply to social prediction problems. Our goal is not to learn how to build computer vision models or to understand the state of the art in computer vision. If you would like to learn that, see COS 429 taught by Professors Jia Deng and Olga Russakovsky. Their contributions to computer vision include ImageNet and the ImageNet challenge, which we will read about this week. There are also online resources such as Stanford's CS 231n.

Two key ideas: Bayes error and structure

Perception problems are characterized by low Bayes error rates and high degrees of structure. Let's unpack this statement.



The figure shows a sample of images from MNIST, a landmark dataset used for handwritten digit recognition. To a human, there is little uncertainty about what the digits are; the challenge is teaching a computer to recognize them. Problems of this kind are said to have a low Bayes Error Rate (BER). The BER is the error rate of a hypothetical perfect classifier that knows the true class probabilities for each possible value of the feature vector. For MNIST, the BER is close to zero. The reason it may not be exactly zero is because there can be instances that look unrecognizable or halfway between two digits, say a 4 and a 9. Such instances do occur in the dataset, but they are rare. On the other hand, if you wanted to guess someone's weight given only their height, the BER would be comparatively high.

Our characterization of perception problems as "inverse" prediction problems (see course pre-read) may help understand why they have low BER. Since noise accumulates in the forward direction, the feature vector is a noisy, high-dimensional instantiation of the class label. In the case of handwritten digit recognition, we may imagine each writer as having a fixed mental template for each digit, which is rendered in a noisy way due to the limits of fine motor skills. Due to high dimensionality, the regions of the feature space corresponding to each class are mostly nonoverlapping. In fact, scripts and glyphs have likely evolved in a way that makes different symbols easy to tell apart despite the imperfections of handwriting. We hope this intuition is helpful, but we don't offer a proof that all perception problems have low BER, nor even a rigorous characterization of what is a perception problem.[1]

You will frequently encounter the concept of Bayes Error Rate in the readings. Both Breiman and Hand mention it. It is sometimes called irreducible error or aleatoric (as opposed to epistemic) uncertainty. For a concept that's invoked so often in the context of limits to prediction, it is surprisingly slippery. First, it assumes that the problem definition includes a fixed predictor set. An alternate perspective, better suited to predicting future events, is that the problem definition specifies the target variable but allows the researcher to collect whatever data is deemed relevant, within practical limits. In this perspective, asserting a nonzero BER requires an assertion about the nondeterminism of the universe. Even within the fixed-predictors perspective, in high-dimensional datasets there is no known general way to estimate BER — or even assert that it is nonzero — without theory and assumptions (whereas in the low-dimensional case, given sufficiently many labeled samples we can directly calculate the true class probabilities for each value of the feature vector). In other words, explaining limits to prediction by invoking Bayes error without articulating a specific theory, set of assumptions, or method simply begs the question. One of the main goals of this course is to go beyond these tautological explanations.

Let's turn to the concept of structure. Imagine explaining the appearance of objects like dogs and trees to someone who doesn't know what they look like. You may start with morphology: a dog has four legs and is covered in fur; a tree has a trunk and leaves. If someone didn't understand those concepts either, you may further explain them based on shape, color, texture, and so forth. Fur is fuzzy; leaves are green and sort of elliptical. At an even more basic level lie concepts that are so automatic to our visual system that we may not think to articulate them: a tree remains a tree even if it moves laterally in your field of vision. This example shows that the task of image classification contains several levels of hierarchical structure. Unfortunately, this structure lacks a known analytical description, i.e., in the form of equations. (The Bengio article has a nice, detailed explanation of this point.)

To summarize, this week we're thinking about problems known to have a low Bayes Error Rate and a high degree of statistical (especially hierarchical and nonlinear) structure, but no known analytical description. What is the best way to teach this structure to the machine — should interrogate our visual system and code up the resulting insights, or assemble a ton of visual data and let the computer figure it out? This is one of the central questions in computer vision. Deep learning corresponds to the latter approach.

On the other hand, the true prediction problems that we'll discuss in most of the rest of the course are not known to have a low BER (and we have plenty of hypotheses as to why the BER might be high); they are not known to have much structure either. In fact, discovering more structure than is currently known (including analytic descriptions when possible) is one of the goals of prediction in those domains. This is because discovering and explaining hidden structure may lead to scientific understanding.

We're now ready for our first reading, which is a quasi-history of neural networks and deep learning. This blog series will be useful both as a refresher of basic concepts and for its timeline of major developments. You may find it useful to revisit the later parts covering recent history after doing the rest of the readings.

As you read the text, observe how the methods used to tackle perception problems are informed by the fact that such problems tend to have low BER. For example, the perceptron may seem superficially similar

[1] ImageNet contains some classes (e.g. "philanthropist") that don't correspond to visual concepts (Yang et al.). The existence of such classes introduces a lower bound on the BER.

to linear regression, but the perceptron learning algorithm assumes that the classes are perfectly linearly separable and totally fails if they are not! A perceptron would be useless in social science because it posits a world without noise, uncertainty, or probability and incorporates a hard, brittle decision threshold.

You'll notice that this and other readings make frequent references to researchers being inspired by the brain and biological neural networks. While the inspiration is no doubt real, we won't read too much into the biological analogy as an explanation for the effectiveness of deep neural networks. The reason is that researchers emphasize the similarities and ignore the differences. For example, the brain has no known equivalent of backpropagation, the workhorse of neural network training.

Remarks on Bengio's monograph

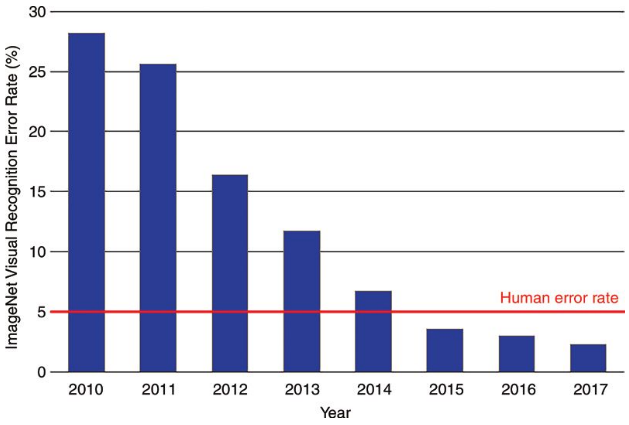
The main reading for Tuesday's class is Bengio's monograph. While his main goal is on how to *train* deep architectures, we will focus on the first three chapters which explain why deep architectures are necessary and useful in problem domains such as visual recognition.

The text excels at offering intuition and high-level arguments. Specifically, it gives many key reasons for the necessity and usefulness of deep learning, including:

- the presence of hierarchical structure reflecting different levels of abstraction
- the inefficiency of insufficiently deep architectures
- the ability of deep architectures to extract hierarchical structure from even unlabeled data
- the number of "variations" of the decision surface as a measure of problem difficulty
- the limits of local template matching
- the power of sparse distributed representations

These are all important ideas. As you read the text, make sure you understand them at an intuitive level.

Bengio's book was written in 2009 and has aged well considering that it predates the ascendance of deep learning in computer vision. Consider this sentence: "the current state-of-the-art in machine vision involves a sequence of modules starting from pixels and ending in a linear or kernel classifier, with intermediate modules mixing engineered transformations and learning, e.g., first extracting low-level features that are invariant to small geometric variations (such as edge detectors from Gabor filters), transforming them gradually (e.g., to make them invariant to contrast changes and contrast inversion, sometimes by pooling and sub-sampling), and then detecting the most frequent patterns." Bengio is advocating for automating this pipeline using deep learning, and that is exactly what happened, resulting in plummeting error rates on the ImageNet challenge.



Top-5 error rate on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) of classifying an image into one of 1,000 categories. Deep neural networks entered the picture with AlexNet in 2012.

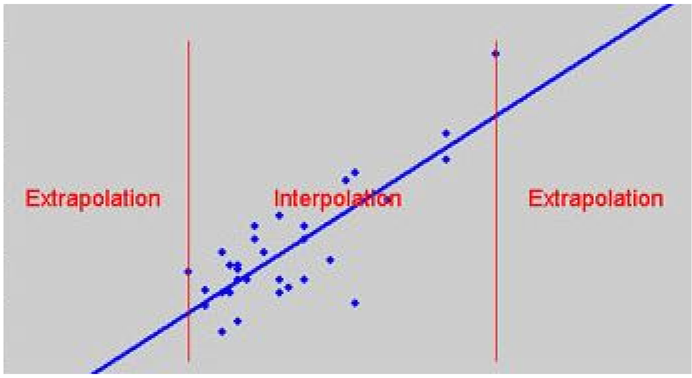
Bengio strongly emphasizes learning a hierarchy of representations via unsupervised pre-training. This is considered less important today than it was a decade ago. Nonetheless, the fact that it is *possible* to extract structure in the data using unsupervised methods is an important conceptual building block for understanding, even if not for engineering. We will see a very simple example in a hands-on way in Tuesday's breakout activity, which shows that tSNE alone is effective at clustering the digits in MNIST. In fact, tSNE is just a low-dimensional embedding / visualization algorithm that accomplishes this as a side effect; representation learning algorithms like autoencoders are much more effective at learning the structure in MNIST (and other visual datasets).

Bengio wrote this monograph not long after a breakthrough paper in 2006 that proposed a fast algorithm to train a deep architecture called Restricted Boltzmann Machines. In later chapters, he places much emphasis on RBMs, but it has turned out that a variety of deep architectures can be efficiently trained.

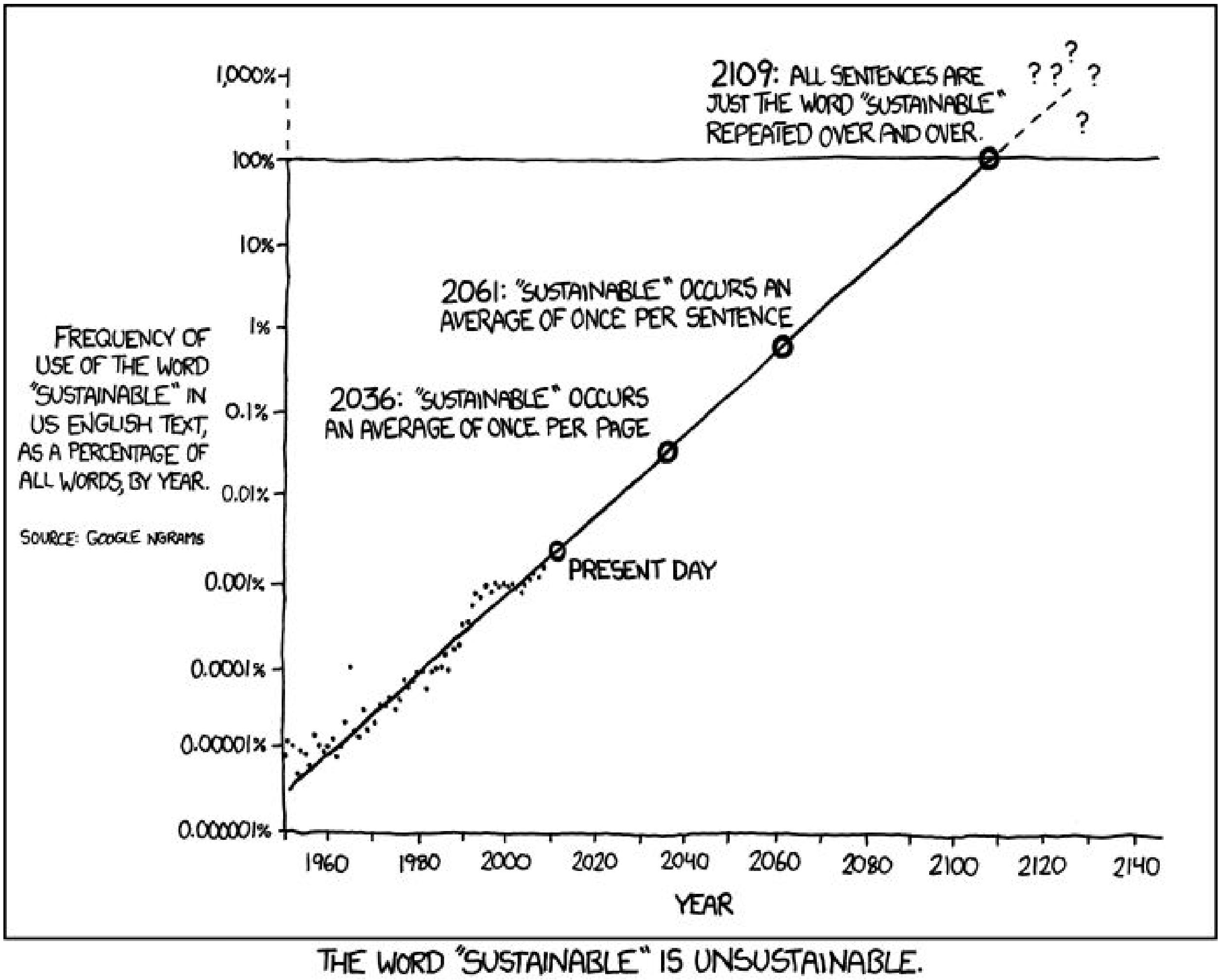
A limitation of Bengio's text is that the arguments are hand-wavy, despite some attempt at formalization in Chapter 2. It is not clear which of the intuitive claims correspond to known theorems (except for Theorem 2.1). For example, he defines *highly varying* functions as those for which a piecewise approximation (e.g., piecewise-constant or piecewise-linear) would require a large number of pieces. But what is a piece? Perhaps some functions that are intuitively highly varying can in fact be represented by a small number of highly complex "pieces". Indeed, as Bengio notes, this is what deep representation learning does, and a deep network can be seen as an efficient "factorization" of a large but shallow circuit (i.e. one that has a large number of simple pieces).

So perhaps the concept of *highly varying* can be defined by further reference to the depth of architecture needed to represent it. But then the statement that deep architectures are necessary for highly varying functions reduces to the tautology that deep architectures are necessary when deep architectures are necessary! There is no doubt that deep architectures are necessary for *some* tasks, but there is nothing formal that tells us why this is true of perception tasks, only the intuition about a hierarchy with multiple levels of abstraction (and the empirical evidence that we've tried and failed with shallow architectures).

"Local template matching" is relevant to limits to prediction. To see the connection, we must understand the difference between interpolation and extrapolation. Both are instances of making predictions on unseen data, but in interpolation, those data fall within the range of previously observed values of the predictors, whereas in extrapolation they fall outside that range. Extrapolation is typically less accurate than interpolation, although of course there is no mathematical reason this is always true. Predicting the future often involves extrapolation because some of the predictors (or their combinations) may take values not seen in the past. Think of the world before and after covid-19.



XKCD illustrates the dangers of extrapolation



Local template matching is a type of interpolation and Bengio shows how a wide variety of ML methods fall into this category. Further, although it is not clear from the text, deep learning is not necessarily a way to transcend the limits of interpolation either. It's just a way to apply interpolation on a transformed, semantically meaningful space rather than the space of raw sensory input. Indeed, the final layer of many deep architectures used for image classification is just a logistic regression. Even more tellingly, modern facial recognition proceeds by computing low-dimensional (e.g. 128-dimensional) embeddings of face images and then simply finding the nearest neighbor in the embedding space. We'll explore this further in a breakout activity.

Bengio does express the view that sparse distributed representations, as he calls them, can enable generalization to categories (classes) not seen in the training data. This is a hallmark of human reasoning. While abilities like zero-shot learning are impressive, visual reasoning remains a relatively nascent area of research.

Interlude: historical notes

Armed with the understanding we've developed so far, let's briefly revisit the history, which will motivate the importance of datasets and prepare us for Thursday's readings.

After an initial wave of overenthusiasm about 1-layer neural networks was crushed by Perceptrons (1969), a second wave of research established all the core principles of modern deep learning *by the 1980s*, including the importance of architecture depth, learning representations, convolutional networks, and training DNNs by backpropagation. Despite successful commercial applications, notably handwritten ZIP code recognition, this research hit limits because of things like the vanishing gradient problem that led to the pervasive belief that DNNs cannot be efficiently trained. In retrospect, the core reasons for these limits were not scientific but rather related to engineering and hardware.

Neural networks researchers persevered for decades through the skepticism they encountered. This story has now so often been told that it has led to the reverse myth that deep learning is great for all kinds of problems and is encountering prejudice from uninformed outsiders such as social scientists.

In 2006 there was an apparent breakthrough in training an architecture called Restricted Boltzmann Machines, which was enough to rekindle interest in neural networks. In retrospect this development was much less scientifically important than originally thought. Nonetheless, it sustained the interest of the community long enough that the feasibility of training deep architectures could be clearly established. Once that happened, there was no going back.

Turning specifically to computer vision, the real breakthrough was the release of ImageNet in 2009. For Thursday, we will read a press article about the creation of ImageNet, as well as the ImageNet paper itself. The ImageNet contest dataset (ILSVRC) is described in a separate paper; we haven't assigned it due to the overlap in content but you may find it useful to refer to it while doing the other readings.

The significance of ImageNet and ILSVRC didn't become widely apparent until AlexNet won the challenge in 2012 using a deep learning approach with a massive drop in error rate over the 2011 winner. While it included several technical innovations, in our opinion they are much less significant than the core principles of training deep architectures that had been developed much earlier.

The centrality of datasets

Our characterization of the dataset as the real breakthrough becomes less surprising when we consider that datasets like ImageNet perform three roles that have no parallel or only a weak parallel to datasets used for social prediction.

First, they serve as benchmarks for algorithmic progress. If our goal is to use machine learning to predict, say, individual life trajectories, the instances are individuals. This means that even if there is no benchmark dataset and every researcher samples a different set of individuals for learning, they are all in principle attempting the same prediction task and can compare results with each other, even if this way of doing things would be too laborious in practice.[2] But for a task like object recognition, there is no obvious

[2] The practical difficulties of creating representative samples are quite serious; see, for example, the WEIRD crisis in psychology. Similarly, the debate we encountered about the accuracy of expert forecasts of geopolitical events has the same primary cause. However, for various reasons, in these domains it is not

way to sample images from the visual world; it has to be constructed by the researcher. It is easy to underappreciate how strong this limitation is; when reading the ImageNet paper, make note of the myriad choices they had to make and the different choices they could have made. Without benchmark datasets, it is practically impossible to compare algorithmic performance and tell if progress is being made. Specifying the distribution precisely enough that another sample can be drawn from it may be harder than just releasing the dataset!

An unfortunate side effect of the centrality of benchmark datasets is that the community becomes hyper-focused on those datasets and turns into a giant competition. The *formal* competitions such as ILSVRC are carefully structured in a way that promotes scientific progress; much more damaging is the *informal* competition that seems to inevitably emerge, resulting in unfortunate outcomes such as insightful papers being rejected because they failed to beat the state of the art, or unoriginal papers being published because they did beat the state of the art by (scientifically insignificant) application of greater computing power.[3]

This point has been made endlessly; our goal here is to point out that this is not purely a cultural difference but is in fact a consequence of scientific differences. In short, even specifying a problem usually involves reference to a benchmark dataset because that's the simplest way to specify the input distribution. Once a community converges around a standard benchmark, competition is a tiny, tempting step away.

The story of computer vision can be told in terms of benchmark datasets, each representing a greater scale and complexity than the previous generation. Thus, datasets have had a second function of spurring the development of optimization algorithms and architectures as well as allowing hyperparameter tuning. This is related to, but distinct from, the benchmarking function. For example, some of the innovations introduced by AlexNet and other deep architectures become useful only at the scale of ImageNet and not, say, PASCAL-VOC or CIFAR-10. Again the parallels to social prediction tasks are weak; there, new datasets enable discoveries about the domain but rarely engender new algorithmic insights.

A third function of datasets (or at least ImageNet) is to provide a source of world knowledge. The images in each ImageNet category are a sample, but the set of categories itself strives to be comprehensive; it is more like a census. It has become very common to pre-train models on ImageNet and transfer the knowledge gained to an altogether different visual task; what the two tasks share in common is the underlying visual structure of the world.

As you read the ImageNet paper, make note of how the goals, approach, and decisions differ from a social science dataset. In particular, note how the need for scale trumps many other considerations.

If a single dataset spurs the development of algorithms, models, and architectures by hundreds of researchers for years, has it led to adaptive overfitting through hyperparameter tuning on the test set? The Recht et al. paper tests this hypothesis by constructing a new test set that follows the ImageNet (and

a viable approach to simply agree on a benchmark dataset and ignore the messy question of specifying the distribution.

[3] Another downside to a field oriented around one-dimensional, competitive pursuit is that it becomes structurally difficult to address cultural biases in models and classifiers. If a contestant takes steps to prevent dataset bias from propagating to their models, there will be an accuracy drop (because accuracy is judged on a biased dataset) and fewer people will pay attention to the work.

CIFAR-10) sampling procedures as closely as possible. It concludes that overfitting is not happening, whether vanilla or adaptive overfitting. This is a surprising result; the paper offers two possible explanations, but does not test them. On the other hand, it does find substantial accuracy drops on the new test sets caused by the fact that the original datasets consist of "easier" instances despite Recht et al.'s careful attempts to reproduce the sampling procedure.

In one sense this can be read as a criticism of benchmark datasets: the accuracy figures reported on such datasets do not constitute generalizable knowledge. The authors seem to favor this interpretation (e.g. "On ImageNet, the accuracy loss amounts to approximately five years of progress in a highly active period of machine learning research.") On the other hand, the results can be seen as a vindication of the benchmark dataset approach. After all, the relative order of model performance was almost exactly preserved on the new test sets, which means that the knowledge gained about which models work well did turn out to be generalizable. It suggests that even if the ImageNet team had gone to great lengths to specify a sampling procedure instead of releasing a benchmark dataset, there would have been substantial differences in the accuracies on different realizations of this procedure, making performance comparison difficult. Thus, the community's practice of agreeing on a dataset without necessarily thinking carefully about the data distribution seems to have worked well.

Final thoughts

Our goal was to explain why computer vision is possible rather than how to build models. For this reason we've almost completely ignored the aspect that the community spends most of its time on, namely optimization. One useful thing to know is that variants of gradient descent work well on a range of models, architectures, and problems, given enough data and computation. Of course, great ingenuity is required to get it to work well in practice and avoid local minima, run in as close to linear time as possible, and avoid the ever-present danger of overfitting given that models may have more parameters than training instances. Without diminishing the efforts of researchers working on these problems, we subjectively think that this isn't where the "magic" of deep learning resides.

Another useful thing to know about optimization is the difference between loss functions and scoring functions (which may superficially look similar). Because optimization is hard, the choice of loss function in machine learning is generally whatever makes efficient convergence to the global minimum most likely for a given optimization algorithm and architecture. The loss function is not necessarily tied to the scoring function, which is the evaluation criterion. The scoring function is part of the problem statement while the loss function is part of the solution space. [4]

Culturally speaking, work at the bleeding edge of deep learning and computer vision has the character of art and engineering more than math and science; engineering success tends to advance faster than theory and understanding, although the latter tend to follow close behind. This results in a pedagogy gap, with

[4] Furthermore, in perception problems, since success is more dependent on the ability to learn the structure in the data and less dependent on a careful treatment of uncertainty, the choice of scoring function is much less important than in social prediction; simple measures tend to work well and the relative performance of algorithms is not strongly dependent on the choice of scoring function. For example, the ImageNet challenge is scored on accuracy of categorical outputs; if an analogous approach were used in (say) the Fragile Families Challenge, the optimal approach would be to always output '0' for rare outcomes such as eviction if we assume that P[Y=1 | X=x] < 0.5 for all *x*.

the latest successes of deep learning often reported as mysterious in the absence of accessible explanations.

Our final pair of readings are a recent controversial blog post arguing that incorporating human knowledge in AI is pointless and even counterproductive, and a rebuttal to this post. One of the points in the rebuttal is that convolutional neural nets are an example of an architecture that exploits human understanding of the visual world, without which the modern performance of computer vision would be impossible. These points are valid, but they are relatively weak and don't change the main claim that deep learning has obviated careful feature engineering in many domains.

This pair of blog posts is merely the latest iteration in a long-running debate. What these authors (and many others) miss entirely is the distinction between low-BER / high-structure domains, in which the "bitter lesson" is mostly true, and high-BER / low-structure domains in which theory-based extrapolation is necessary and the "bitter lesson" is mostly false. In addition to perception, the main problem domain mentioned in the "bitter lesson" post is computer chess and computer go, which are notable for being games of complete information where the BER is exactly zero.

The hubris among technologists about the uselessness of domain knowledge has time and again resulted in simplistic and incorrect problem formulations, overconfident proclamations about predictive performance, and blind spots relating to bias and fairness that would have been obvious to domain experts.