Pre-read for Wednesday, February 21: Weather, part 2
Matt Salganik
Spring 2024

Note to readers:

- Given the purpose of this document will intentionally not have as much precision as a real piece of research, and I'll tend the caricature things to, I hope, make contrasts easier to see.
- Please feel free to add comments, questions, and disagreements using the comments feature.
- After class, I plan to turn this pre-read into a series of blog posts.

Abstract:

For this class we are going to see the 100 year evolution of numerical weather forecasting. To me this looks like an amazing story of scientific success, and I wonder if a process like this could happen for other problems, perhaps even some problems in sociology. Lynch (2008) tells the story of the past and somewhat the present. Schultz et al. (2021) hint at a possible future using deep learning, and Lam et al. (2023) show a step in that possible future. Rather than thinking of this as a story about weather, I encourage you to think of this as another example of what Sutton (2019) calls the "bitter lesson." If the bitter lesson is true, what else might it apply to in your field? Might we be able to "leapfrog" much of the work that went into numerical weather prediction, not need to figure out the physics, and just skip right to Lam et al (2023). What might be gained? What might be lost?

Note: These readings contain lots of information about meteorology. I encourage you not to get lost in the weeds. This is not a class about meteorology. Try to see this as an example of a process that might take place in some other field.

Introduction

"The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin." Such begins the blog post titled "The bitter lesson" by Richard Sutton (which you will read for this class). Basically, computing power doubles roughly every 1.5 years (Moore's Law), and this doubling swamps everything else. Sutton's post was not written about numerical weather prediction, but we are going to try to see how well it describes numerical weather prediction. And if it does capture the trajectory of numerical weather prediction, how might it predict the trajectory of other fields?

At a very high level you are going to read about the idea of using physics to predict the weather being replaced by using ML to predict the weather. The paper using ML, specifically deep

learning, to exceed state-of-the-art performance by physics-based models was just published in December 2023 so we don't yet know how this story will evolve.

One last note about terminology. In this class, we've contrasted ML models from dynamical models, and I now see that's not quite right. The stream of papers you will read for class Wednesday definitely uses dynamical models (learn some rule that maps the system at time t to t+1, measure the system at time t, then use the rule to predict t+1, t+2, …). However, the process that maps t to t+1 now comes from ML not physics.

The long arc of numerical weather forecasting

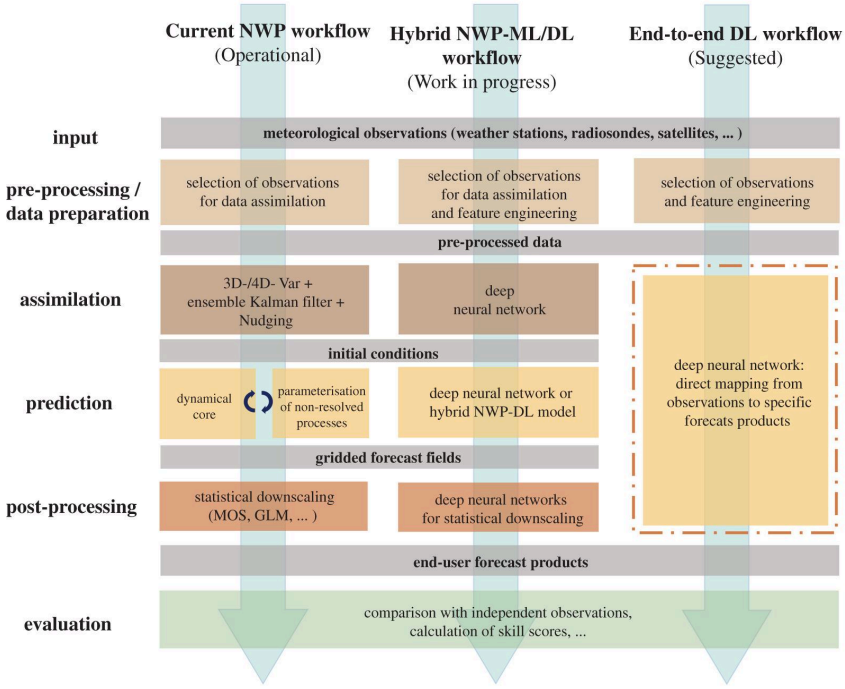
The origins of the field of numerical weather forecasting begin around 1900 and are attributed to Abbe and Bjerknes. They both had an idea that seems to me like Laplace's Demon. If we know the current state of the atmosphere and we know the rules of physics, then we can just project forward what the atmosphere will be into the future. At that point in time, the appropriate laws of physics were known. But, when Lewis Richardson attempted to put this idea into practice in the 1910s, he ran into two main problems: data and compute.

First, data. Richardson and his colleagues did not know the current state of the atmosphere because they didn't have tools to measure the state of the atmosphere all over the world and up into the stratosphere; weather balloons had not been invented and they certainly didn't have satellites. Second, compute. All the computation had to be done by hand and this was slow, actually slower than the weather. This led Richardson to dream: "Perhaps some day in the dim future it will be possible to advance the computations faster than the weather advances and at a cost less than the saving to mankind due to the information gained. But that is a dream."

Fast forward 100 years—and skipping over lots of hard and interesting work—the data and compute problems have not been "solved" but have been dramatically changed. Now we have lots of historical weather data, and we have much faster computers. For example, Lynch (2008) (which you will read for class) notes that from ENIAC—one of the first computers roughly around 1950—to HPCF—a state of the art supercomputer from around 2000—the amount of computing power increased by a factor of 10^10, which is one hundred billion (100,000,000,000). The constraints that we face today are much different than those that Richardson faced in the 1910s.

Given that we are now in a data and compute rich environment relative to Richardson in the 1910s, what if we just give up on physics and just throw deep learning at the problem.

This is the idea addressed in Schultz et al. (2021), which you will read for class. Of particular interest is Figure 1. Roughly the column on the left is the standard approach (as of 2021) and it relies on a lot of domain expertise. Schultz et al. advocate for the approach at the right, which requires a lot of deep learning expertise. The switch from the approach on the left to the approach on the right seems aligned with Sutton's Bitter Lesson, and it sets the stage for the Lam et al. paper which was published just a few months ago.

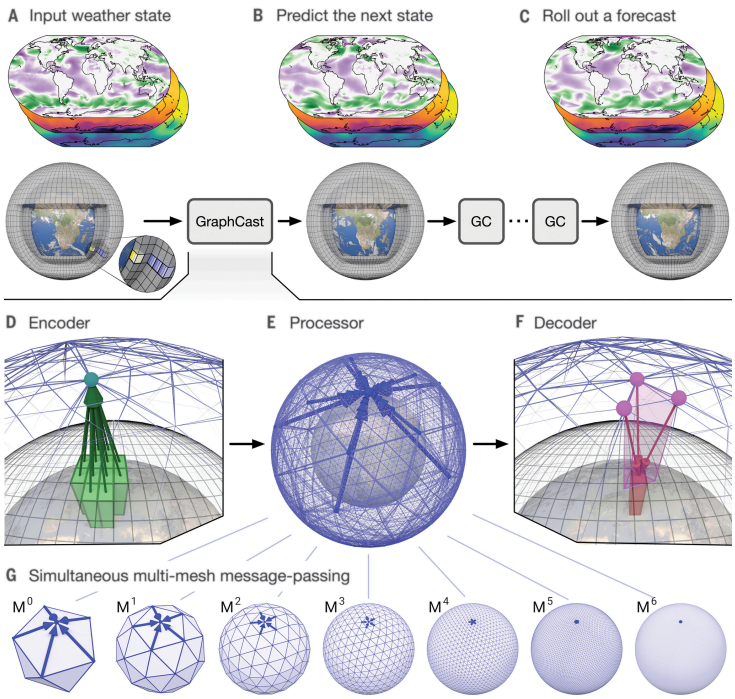


[Fig 1 from Schultz et al. (2021). The column on the left is the standard approach (as of 2021) and it relies on a lot of domain expertise. Schultz et al. advocate for the approach at the right, which requires a lot of deep learning expertise. This is the approach taken by Lam et al. (2023).]

Graphcast

Lam et al. published Graphcast in December of 2023. You'll read this paper for class. Together with the comms team at Google DeepMind, they also produced a nice video and blog post that you will read as well. In general, you should be skeptical of blog posts by large companies touting their research breakthroughs. However, I think this one is reasonable (at least given my limited understanding of meteorology).

Roughly, Graphcasts, which is summarized in Fig 1 says, let's just take all the data and put it in a deep learning model. The paper then shows that this approach leads to predictions that are better than state-of-the-art predictions using the physics-based approach.



[Fig 1 of Lam et al. (2023). Schematic of GraphCast.]

Setting aside the details about how the forecast was created, which are beyond my area of expertise, it is impressive to me to note how much care went into forecast verification. Most of the paper is about forecast verification not forecast creation. A few things to note: 1) they are comparing against a strong baseline: HRES (High-Resolution Forecast), which is the top deterministic operational system in the world and is produced by European Center for Medium-Range Weather Forecasts (ECMWF); 2) they are also comparing against a second strong baseline, the best previous ML-based system Pangu-Weather, a similar system developed at Huawei (Bi et al. 2023); 3) they compare performance for both routine and severe weather; 4) in their evaluation they draw on the meteorological concept of "skill". Roughly, skill measures accuracy relative to some baseline. For example, because of sensitive dependence we can't make a skillful forecast of the weather in Princeton on July 4, but we can make a reasonably accurate one (it will be hot). The limits of weather forecasting are about skill not accuracy.

Conclusions

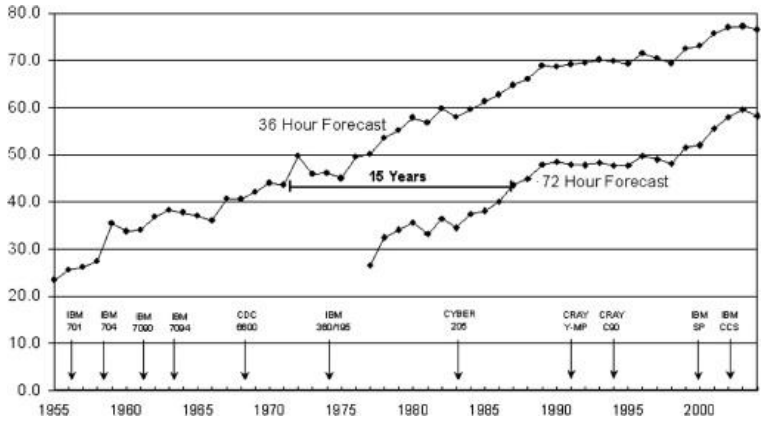
The results of Lam et al. (2023) seem to suggest that the Bitter Lesson has now been learned by meteorologists, but I guess time will tell.

This leads to a few things I'm wondering about, but don't know:

- Could Graphcast be used for climate simulation? For example, what if we want to know what will happen to the climate if we continue to release CO_2 at current rates? That

seems to require some kind of ability to extrapolate from current conditions to new conditions. I can see how that might be possible with a physics-based model, but I'm not sure about this deep learning model.

- What if we were going to try this trick somewhere else? For example, the International Monetary Fund (IMF) publishes the World Economic Outlook, which makes financial projections for the entire world. You could think of this as roughly a "one year weather forecast" for the economy of the world. What would it look like to embrace the "bitter lesson" and take a fully data-driven, compute-heavy approach to this task? What would be gained and what would be lost?
- What is the role of domain expertise in a project like GraphCast and the bitter lesson in general? GraphCast seems to rely on scientific understanding to know what are the right features to include in the model and what are the right things to evaluate. How do we combine domain expertise with the bitter lesson? (I would note that in sociology, there still seems to be a lot of emphasis on using ML for feature selection, which strikes me as qualitatively different from GraphCast.)
- What evidence would convince you to replace the physics-based model with the ML-based model for real, routine weather forecasts? What are things you'd definitely want to see before making that change?
- What would it mean to do sociology in a way that tried to harness Moore's Law? I would argue that our current approach pretty much ignores it.
- How do we make sense of these readings compared to our understanding from Monday that the atmosphere is subject to sensitive dependence on initial conditions leading to a ceiling for the best possible skillful forecast? Could we learn about that ceiling from this approach? How would we interpret the results of GraphCast if we didn't know what the ceiling was?
- Should we view GraphCast as a qualitative change or the continuation of a long-running series of incremental improvements as shows in Lynch Fig 4?



[Fig 4 from Lynch (2008). There are a series of improvements that have happened in numerical weather prediction over the past 50 years that have lead to continually improving accuracy. Is GraphCast just another step in this long process, or should we think of it as something qualitatively new?]

Let's close with a last word by Sutton and the a troubling question:

> “This is a big lesson. As a field, we still have not thoroughly learned it, as we are continuing to make the same kind of mistakes. To see this, and to effectively resist it, we have to understand the appeal of these mistakes. We have to learn the bitter lesson that building in how we think does not work in the long run. The bitter lesson is based on the historical observations that 1) AI researchers have often tried to build knowledge into their agents, 2) this always helps in the short term, and is personally satisfying to the researcher, but 3) in the long run it plateaus and even inhibits further progress, and 4) breakthrough progress eventually arrives by an opposing approach based on scaling computation by search and learning. The eventual success is tinged with bitterness, and often incompletely digested, because it is success over a favored, human-centric approach.”

What if the approaches that are satisfying to us as researchers are the ones that are actually impeding our progress?