Neural networks for weather forecasting

Tim Bradshaw*

Incomplete draft, 16th December 2018

Contents

1 Introduction

Neural networks are likely to provide better weather forecasts in due course than current numerical models. If this is true then weather forecasting organisations that don't use them will be replaced by ones that do. Even though this only may be true, weather forecasting organisations should be investigating these techniques, today.

*tim.bradshaw@metoffice.gov.uk until the end of 2018, tfb@cley.com after that date.

AI has experienced successive cycles of hype and disappointment since the late 1950s, but there have been real developments. Neural networks, often now known as 'deep learning systems' or just 'AI' (and known in this paper as 'NNs') are close to the peak of the current hype cycle, from which they will decline, probably catastrophically. Nevertheless NNs can be extremely good solutions to some problems.

In particular NN models are likely to be highly successful for weather prediction. However they will not be trivial to design and deploy: cargo cult NN approaches are not going to work.

If NN models are successful then they will largely displace hand-crafted physics-based models (GCM models such as UM). Weather forecasting is a *service*, and consumers of the service care only about how good the forecasts are rather than how they are produced.

If this happens then organisations involved in weather forecasting, such as the Met Office, will need to adopt NN models or cease to exist: NNs are an *existential threat* to weather forecasting organisations.

This means that such organisations should be investigating NN models very seriously *now* so that, in the likely case that they are successful, they are not left behind.

[This is an incomplete draft assembled from various notes: it's longer than it should be, especially in the first section, and not final in several other ways.]

2 Two stories from the history of artificial intelligence

AI has a long and storied past. Some of these stories have useful lessons.

Don't believe the hype

In 1965, Herb Simon, who attended the 1956 Dartmouth workshop[1] where the term 'artificial intelligence' was coined, said:

> [...] machines will be capable, within twenty years, of doing any work a man can do[2].

In 1970, Marvin Minsky, an organiser of the Dartmouth workshop, said:

> In from three to eight years we will have a machine with the general intelligence of an average human being[3].

Neither of these statements turn out to have been true.

In 1973 the Lighthill report[4] said:

> In no part of the field [of AI] have the discoveries made so far produced the major impact that was then promised.

The Lighthill report all but killed AI research in the UK in the mid to late 1970s and similar things happened at the same time elsewhere: this was the second 'AI winter'[5].

[1] 1956 Dartmouth Summer Research Project on Artificial Intelligence

[2] Quoted in *AI: The Tumultuous Search for Artificial Intelligence*, Daniel Crevier, 1993

[3] Crevier p 96

[4] *Artificial Intelligence: A General Survey*, James Lighthill, 1953

[5] The first AI winter was a consequence of the failure of early machine translation projects in the mid 1960s. There have been at least three AI winters.

Since the very start of AI it has been subject to *hype cycles*: some wonderful new approach is discovered leading to uncontrolled enthusiasm and wild claims, some of them made in good faith and some to secure funding; funding duly follows; a few years pass and the approach turns out not to be quite as wonderful as it first seemed and the problems much harder than they first seemed; the funding bodies commission a report and funding is withdrawn, leading to an AI winter; the cycle then starts again with some new technique.

At the moment we are close to the peak of a hype cycle based around neural networks (NNs) driven by vast amounts of training data ('big data'): these have become so dominant that 'artificial intelligence' and 'machine learning' are now synonyms for NNs & big data: everything else that AI once was has been forgotten. There is no strong reason to believe that this is not just another cycle: the self-driving car problem will turn out to be harder than people expect it to be, and replacing humans by machines trained on vast accumulations of unstructured data will not turn out to be as easy as it is claimed to be. The bubble will burst and there will be another AI winter followed by another spring as some new trick is discovered.

But good things come from each cycle: the first gave us much of the foundation of what is now called 'computer science'; the first and second gave us interactive computing systems and programming languages which don't suck; the second and third gave us window systems and object-oriented programming. The current cycle is going to show us that some problems really can be solved by throwing vast amounts of training data at NNs: you can believe *some* of the hype.

Getting stuff done

A lot of early AI was about natural language processing: this was both because the US government was pretty interested in being able to automatically translate Russian into English and very happy to throw money at anyone who claimed to be able to solve that problem and because AI was done by academics, and a reasonable definition of an academic is 'a person who thinks reading and writing is hard and playing football is easy'[6].

So a lot of early AI people either were linguists or knew a lot about linguistics. Linguists regard natural languages as being described by grammars, which contain rules which describe what is well-formed and what isn't, so a lot of early AI consisted of trying to write down grammars for natural languages and understanding how to write parsers based on those grammars and so on.

While it's pretty clearly the case that grammars matter for natural language (anyone can parse 'the cat bit the dog' into 'S(NP(DET(the) N(cat)) VP(V(bit) NP(DET(the) N(dog))))'), it turned out that natural language grammars are remarkably subtle. Different schools arose as to how grammars should be written and what the grammar supposed to be in our heads might look like, and after years of work even the best 'broad coverage' grammars did depressingly badly when pointed at corpora of real text (parsing spoken language, which was the real aim, remained conveniently computationally implausible).

An obvious approach was to try to *learn* the grammar: given a corpus of text suitably labelled as grammatical or not it should be possible for a machine to learn the rules of grammar. After all this

[6] This turns out to be embarrassingly wrong, although, unfortunately, many academics still believe it.

is what children do. Pretty quickly theorems were proved which showed that this had a problem: the amount of input needed to learn a language 'from cold' was absurdly higher than the amount to which children were exposed.

The linguists' response to this problem was to say that, 'well, we're linguists and we're interested in human language, so we need to understand what inbuilt structure humans have in their heads which lets them to do this trick': the idea was that humans in some sense must already *know* how natural languages work and merely had to fit the particular language they were learning around it[7]. So the problem became one of, somehow, working out what this structure was which would teach us a lot about how humans worked and also let us teach it to a machine which could then learn languages efficiently.

But there was another approach, which was to say that none of that mattered at all. Since we actually *have* very large quantities of training data and machines which don't bore easily, we can simply throw the data at the machine and let it learn, even if this is completely implausible as a mechanism by which humans learn language. After all, the people paying for this wanted *something that worked* and didn't care very much *how* it worked. The customers *didn't care* what went on in a human's head when they translated Russian to English, if a machine could translate Russian to English, no matter how that was done: *getting stuff done* was what mattered.

Linguists *hated* the second approach: what were we learning about natural language if we ended up with some opaque blob which, apparently by magic, could translate between natural languages?

[7]This notion is often called deep structure and is due to Chomsky.

We already *had* lots of opaque blobs like that: bilingual people. But the pragmatists realised that the thing they had been asked to do was *make machines translate between languages more cheaply than humans could* and that by doing that *they could get paid*.

And today we have systems which do a sometimes-acceptable job of translating natural languages, and certainly a good enough job to snoop on intercepted communications for 'interesting' things which, after all, was the goal, and all of those systems were created by the use of huge masses of training data. Getting stuff done turns out to matter: the pragmatists got paid, the linguists didn't[8].

What this means

AI has been through a number of cycles of hype followed by failure and there is no reason to believe that we are not simply in the hype stage of the current cycle. During the hype stage absurdly optimistic claims are made, often to get funding: these claims are almost always false. *Despite this* useful results have come out of each cycle. What will come out of the current cycle is that computational problems where there is a huge amount of training data available can often be solved very effectively by neural networks, although the solutions will be extremely opaque.

There are two approaches to solving many computational problems: one is to try and understand what is actually happening in the world and to build a model based on that understanding; the other is to simply arrive at a system which gets the right answer whether or

[8] Today the descendants of the pragmatists run companies with names like 'Google'; the descendants of the linguists live in linguistics departments and worry about old Norse.

not it models what is actually happening in a transparent way and whether or not it is comprehensible at all. Neither approach is better but they are not equivalent: the first approach provides understanding while the second may provide better answers in practice. For many problems, neural networks are a good match to the second approach. Which approach is better depends on what you want to achieve, but consumers of the solution generally do not care at all about anything but getting better answers and will prefer the second approach if it is cheaper, gets better answers, or both.

3 Weather forecasting

I don't think there are any plausible approaches to forecasting the weather which don't involve simulation: some kind of model is built, fed initial conditions and then one or more copies are run forward to predict what may happen, with all the usual caveats about chaos making the prediction hard. But there are two approaches to *building* the model which will simulate the weather.

Two approaches

The traditional approach is to understand the physics[9] and write a system which numerically solves the equations to a lesser or greater degree of accuracy. This has been pretty successful of course.

An alternative approach is to not do that at all, but rather build a system which can, itself, *learn* to simulate the weather: a system

[9] Here I mean 'physics and everything built from it' so including chemistry &c in the usual arrogant physicist's way.

which can be trained to simulate the weather, in other words, based on observations. As far as I'm aware such an approach has not been tried on any significant scale.

The first approach is rather like the linguist's approach to teaching a computer how to handle natural language: understand the grammar of the language and implement this on a computer. It's much better than that, because the nature of physical systems is (perhaps surprisingly) much better understood than the nature of natural language, and because physical systems have well-defined notions of approximation meaning it makes sense to get an answer that is approximately correct (although chaos limits this). The huge advantage of the first approach is that the model is, within limits, comprehensible: you can find and alter the bit of it which models a particular part of the physical system, and people who understand the physical system can build models like this based on their understanding of the physics.

The second approach is the 'getting stuff done approach': although the result will be a system which will simulate the weather, *how* it does this will almost certainly be entirely opaque. It won't be easy to look at some part of the model and map it onto the part of the physical system it is simulating, because no part of the process of arriving at the model cared about that: all it cared about was how good the results of the simulation were. The second approach is therefore completely unhelpful to someone who wants to use simulation to understand weather or, more importantly, climate: it's a tool which does one thing – forecast the weather – but which does it in a way which is almost certainly not comprehensible to human beings.

Believe some of the hype

The second approach also builds on the current hype cycle in AI: it will, obviously, be some kind of neural network model trained on huge amounts of data. But I think there are good reasons to believe that this might be a case where this approach could work extremely well.

There is copious training data. There is obviously a really huge amount of data which can be used to drive a model, which NNs love. But NN models need *training* data in general: they need to be told how well they did so they can correct their weights. And weather is almost the best example it's possible to think of of this: if we want to predict, say, rainfall in 24 hours time, then, if we wait 24 hours, we know how much rain actually fell, and we can use that data to teach the model how do to better. *And this is true for everything, all the time*: every time the model makes *any* prediction about the state at some future time then, at that future time, we know what the state actually is and can use that information to train the model. This is the sort of situation NN people dream about.

In fact the amount of training data is unbounded: as time goes on there is always more. This means that the model can be trained iteratively, essentially for ever. As time goes on it can get better and better (obviously within limits imposed by inherent unpredictability, although it will become better at saying when its own predictions are no good as well).

There is a well-defined notion of error. The model is trying to predict a number of continuous fields (rainfall, wind, temperature &c &c), and these fields have well-defined notions of error (in the simplest case just subtract the predicted field from the actual field). Again, this is something NNs love.

There are existing models which get reasonable results. This might sound like a reason not to consider NN models: see below for why it's not. But it also means that if the NN model predicts something which is absurd then these predictions can be removed from the pool. And in the early stages of training the NN models will presumably make many absurd predictions. So existing, physical, models can be used to help bootstrap the NN models by constraining them.

But not quite all the hype

I think it's very likely that NNs will turn out to do an extremely good job of weather forecasting: everything about the problem looks like something that NNs are good at. But they might not: NNs are not a solution to all problems, and there are some kinds of problem for which a trained NN might do well where it is nevertheless very hard, or impossible, to actually train it: there have been significant improvements in training techniques since the early days of NNs but there are still problems.

NNs are also not just black boxes into which you pour training data and out of which come solutions: the current cargo cult approach to NN solutions[10] is a symptom of the peak of the current AI hype

[10] When people are selling expensive courses by saying 'many people think they need to

cycle and tells you nothing about how easy they are to use. They're pretty easy to use if you want to recognise pictures of cats, but not so easy if you want to do something more interesting. Forecasting the weather with NNs is still going to require a lot of design and understanding and considerable computational resources.

Some of the hype is enough

If NNs work for forecasting the weather they are likely to be extremely successful at it, especially as data volumes increase. It is possible they will not work although I can see no reason why they should not. I think it's rather important that NNs should be tried for weather forecasting for reasons I'll talk about in the next section.

4 Reasons to be cheerful

The obvious reason not to try an NN approach to weather forecasting is that it is, bluntly, *not meteorology*: the model produced by training an NN to predict the weather is unlikely to be comprehensible to humans, and people who have spent a lot of time training to be meteorologists are going to find that all their skills are much less use, and terrifyingly, perhaps they will be no use at all. This is a very bad reason not to try for several reasons.

---

spend years studying advanced math first [to learn AI], but that's just not true: many top practitioners today studied at [our company] without anything beyond high school or basic undergrad math' you know there is something pretty wrong.

Getting stuff done matters

Weather forecasting is a *service*: consumers of weather forecasts don't actually care how they are made. They don't mind if they are made using hand-crafted physically-based meteorological models as at present, by models derived by machine such as NNs or by reading tea leaves: all they care about is the statistical accuracy of the forecasts. If NNs (or, indeed, tea-leaves) produce statistically better forecasts than hand-crafted physically-based models than the consumers of weather forecasts will use them instead, and will stop paying for the other sorts of forecasts.

It's not certain that NNs will work, although I think it's very likely they will. If they do work, then they have a lot of good properties, the most significant of which is that they are going to improve over time without model development. So, if *anyone* tries an NN model and has success with it then they are eventually, with high probability, going to be doing a better job of weather forecasting than anyone who doesn't try. The only approach to stopping this would be by preventing people who are using NNs or other machine-learning approaches from getting access to the data they need to train their models: in other words to use monopoly power to prevent possibly-better forecasting technology from being explored. I don't think that kind of protectionist approach is even slightly defensible, and it almost certainly won't work since the data is largely publicly funded in the first place.

And people are exploring NN and other variant approaches to weather forecasting.

- Dark Sky, while not a pure NN system, does a very clever trick[11]: its aim in life it to answer a rather simple question which is important to people who go outside: 'is it going to rain in the next hour or less, here?' It does this by acquiring rainfall radar maps, cleaning them up using a neural network, and then doing what essentially is linear prediction (they are a bit shy about how this works). That's a terrible approach to forecasting the weather but it works quite well if your time-horizon is very short because what is actually happening has a Taylor series expansion. I don't have non-apocryphal evidence but I'm pretty sure Dark Sky does a better job of answering the question it sets out to answer than Met Office forecasts do: I certainly trust it more if I'm going for a walk[12].

- David Gagne at NCAR is using NNs to predict hailstorms, as described in this article.
- Microsoft are using AI (which will mean NNs) to predict the weather.

There will be other people doing this: the list above is just what I found on very casual searching.

So this is not just a hypothetical case: people are investigating NNs for weather forecasting, today.

[11] This description is really of the original, 2011-era Dark Sky: it does quite a lot more than this now. The original approach is described here.

[12] Dark Sky is helped by not being a batch system: if I ask the Met Office app what the weather is it gives me something the model predicted possibly several hours ago, while Dark Sky tells me what it predicts based on information that's a few minutes old, about what will be happening in half an hour (and its grid size is tiny since its based on the resolution of the radar images that drive it and its doing hardly any computation on each grid element): unsurprisingly it very often does a better job.

If NN-based forecasting is going to work better than hand-crafted model forecasting then *any organisation which wants to stay in the weather forecasting business needs to be doing NN-based forecasting*. This means that they need to be investigating it, today. I would expect that the Met Office wants to stay in the weather forecasting business.

The Persistence of Memory

So, from the meteorologist's perspective this looks fairly grim: if NN-based forecasting works then it looks like traditional hand-crafted physics-based NWP approaches are doomed, and a lot of meteorologists will be out of a job. The only way to stay afloat is going to be to bite the NN bullet. This is a pretty unattractive proposition, I think.

If things are this bad there's no way out if NNs work: either your employer moves to using NN-based forecasting and you lose your job, or they don't, they get displaced by organisations who do, cease to exist, and you lose your job.

But things are probably not quite this bad for several reasons.

Cargo cult NNs will not work. Because NNs ('deep learning systems', 'AI') are extremely fashionable a number of *cargo cult* 'solutions' are being offered. These are prepackaged applications or libraries which you merely need to plug in and feed with suitable data for the desired results to appear. Even better, you may be able to feed your data to someone *else's* system and they will produce the desired results for you: your cargo cult can be cloud-based for double fashionability.

These systems might work if your goal is to do things a lot of other people already do such as spotting patterns in customer behaviour and, in the second case, you don't mind becoming a vassal state to the company who owns the machines where the system runs. But they're not going to work for forecasting the weather, because forecasting the weather is actually hard, and because it's not desirable to outsource your key competence, however fashionable that might be.

Designing and configuring NN models will be something that actually requires expert meteorological knowledge, and will continue to do so.

Hand-crafted models will continue to be needed. Although I expect that NN models will fairly quickly outperform hand-crafted physics-based models, those models will still be needed for at least four reasons.

Firstly hand-crafted models are more likely to remain sane than NN models in the early stages. There's no rule that says that an NN won't get some mad idea into its head and start, occasionally, making predictions which are completely physically insane. And because it's so opaque there's really no way of telling whether it's gone mad or not until the real data comes in and tells you it has. If the NN converges to some reasonable state then these episodes of madness will stop, but they will certainly happen early on. A hand-crafted model can provide a good sanity check and can weed out insane NN predictions. Indeed it should be possible to use a rather coarse hand-crafted model to prune clearly insane members of an ensemble of NN models fairly early on thus allowing more resources to be spent on the ensemble members which are making sensible predictions.

Secondly hand-crafted models are far more transparent than NN models. A hand-crafted model is designed to represent the physics in some comprehensible way, while an NN is not. So if you, for instance, train an NN to predict precipitation only then the chances are quite high that somewhere in it is going to be a representation of, say, surface wind, but that representation may be utterly opaque, because it won't have been anything that it was ever asked to make *not* opaque. There's basically no chance of being able to point into some part of the model and say 'surface wind is here'. So anyone who decides that surface wind is interesting won't be able to extract it from the model without training a new one, which might take months or longer. A hand-crafted model, on the other hand can just have the data pulled out of it.

Thirdly, *hybrid* models may well be better than either pure NN models or hand-crafted models. There are a lot of ways a hybrid model could be put together, with perhaps one extreme being using an NN to evaluate members of an ensemble of forecasts by a hand-crafted model, and the other being using a hand-crafted model to constrain forecasts made by an NN. I expect this will continue to be an interesting area to explore even if pure NN models win in the shorter term.

The fourth reason is perhaps the most important of all: climate.

Climate (or: getting stuff done is not always enough)

While NN models are an almost perfect fit for weather forecasting they are, perhaps surprisingly, a terrible fit for climate modelling. This is for two reasons.

Sparseness of training data. NNs are likely to work for weather prediction because the training data is so copious: if you want to predict the weather a given time ahead then you simply predict, wait until that amount of time has elapsed and you have training data, and then you iterate this process. You can't do that for climate: if you want to predict the climate a century ahead you can neither wait for a century for the training data nor can you iterate the process.

Opacity of NN models. Even if climate modelling by an NN is technically practical it's an absolutely terrible answer to the questions people actually want to answer. If I run some NN model and it predicts 4 degrees of warming by 2100 the first thing people will ask is 'why does it predict that?'. And the best answer to that question is 'because some opaque blob of weights which neither I nor any human understands told me that', which is a *terrible* answer: it's essentially the same as 'a voice in my head told me'. Given the political sensitivity of climate modelling this is not going to be an answer anyone will accept, and nor should they.

So climate modelling is a really good example of a place where a transparent physics-based model is the only reasonable answer. And that's ultimately because the people who are interested in climate[13] are *not* just interested in a statistically-good prediction (whatever that even means in this case): they're interested in *why* the prediction is what it is. Climate modelling requires hand-crafted physics-based models, and there's no way around that.

---

[13] Which should be 'everybody'.

Notes

Climate, again

[This is not part of the main document but rather some leftover material.]

I presume that the climate modelling part of the Met Office can't itself fund the development and maintenance of UM: to some extent it is parasitic on the weather forecasting part. If NN models largely or completely displace hand-crafted models for weather this may be difficult for climate people, who need such models.

On page 13 I mentioned that Dark Sky benefits from not being a batch system: its 'forecasts' (really, interpolations) are based on data that is often only a few minutes old. As I write this the Met Office forecast is more than two hours old. Quite independently of the NN question, if the Met Office wants to compete with systems like Dark Sky it needs not to be using a batch system which can be hours out of date.

The names of things: 'neural networks' (NNs), 'artificial neural networks' (ANNs) & 'connectionism' are the same thing (the last of these is probably not a fashionable term any more). 'Machine learning' substantially means NNs now. 'Deep learning' & 'deep learning networks' mean NNs. 'AI' now substantially means NNs as well although this usage is historically wrong and probably transient.