The Bittersweet Lesson 😆

The strange case of inductive bias in Transformers

Felix Hill
21 Oct 2024

Do you remember a few years back when the notion of **inductive bias** was central to machine learning and AI research?

For example, not so long ago, a core part of my team's research was trying to design models with the right bias to optimally learn and/or represent the data in question (language, in our case).

But recently, the idea of inductive bias has become a bit, well, strange.

Any neural network that is not fully-connected can be thought of as having some inductive bias. A classic example is the convolutions in a convolutional neural network, used for finding things in images.

The conv net uses the same weights to process data from different parts of the image, allowing weights to specialise in 'seeing' specific features like eyes (suggesting an animal) or wheels (suggesting a vehicle). This inductive bias was 'inspired' by what we know about feature detectors in the eyes of animals.

Prior to the last few years at least, when building better models, we as researchers would try to understand the domain (e.g. vision), the data and/or the task, and then use this understanding to design inductive biases. In other words, we sought to determine which parts of the neural network should not be (fully) connected.

So far, so sensible. But what about the case of language and Transformers?

A known 'universal' fact about languages is that *local dependencies* are much more common than *long distance dependencies.* For example, in the phrase:

The banana is very very very very very very very yellow

There is a dependency between *banana* and *yellow* that we would like a model to pick up on, to be able to answer questions like "what colour is the banana?"

The dependency in this case is relatively long (9). But in the (presumably more common) sentence:

The banana is yellow.

The dependency length is only 2.

Dependency parsers say which words in a sentence depend on each other semantically or (grr*) syntactically. They typically build on annotations from highly skilled linguists.
If you apply a dependency parser across many languages, the results are unequivocal: the modal dependency distance is 1, the median distance is either 1 or 2. In short, in language, the distribution of dependencies is massively skewed towards the lower end.

But here's where things get strange.

Recurrent networks (RNNs) have a clear and very widely studied inductive bias toward representing local (short) dependencies. Further inductive biases like Long Short Term Memory (LSTM) are really just attempts to mitigate the underlying short-termism of the RNN. In contrast, Transformers have precisely zero bias when it comes to dependency length. RNNs have the right bias for language. Yet transformers outperform them consistently across a wide, and growing, set of tasks.

To reiterate; dependencies in language are almost all short. RNNs have a short term bias and Transformers don't. Yet Transformers win.

Strange indeed.

But, often when applying networks to language on particular tasks, the Transformer vs RNN gain is quite small (eg.. 90% plays 85%). I assume a lot of the difference is on (rare) test data where long-term dependency is important. There is evidence for this e.g. in the original attention paper by Bahdanau et al. On the majority of test data, where, as we know, long-term dependency is not important, RNNs and Transformers must be performing equally well.

So perhaps we should stop trying to design inductive biases that reflect the most salient aspects (or the most common modes) of the data. If the model is a good general model, it will learn about these salient phenomena easily, *precisely because they are so frequent*. They are *easy* for an unbiased model to learn about. There is no need for a bias to help.

But, critically, an adequately unbiased model will also be able to capture the rare phenomena in the data more easily than a model biased towards the modal phenomena, almost by definition. Hence, the Transformer, which is unbiased, manages long-distance dependencies far better than RNNs (even with LSTM, GRU etc).

So the conclusion could be that there is nothing left to design (other than the most unbiased network possible). That conclusion would indeed feel a bit like the 'Bitter

Lesson' of Rich Sutton. But that's wrong. It's OK. Provided computation, scale and data are limited (which they always will be), there will be more (domain-specific) biases to be designed. So the conclusion need not be bitter.

For instance, Transformers *do* have a very clear inductive bias. It may not be the case that attention is *all* you need*,* but (self) attention is certainly a highly useful inductive bias. The designers of Transformers showed that a network with multiple self-attention layers performs better, on language, than a network with only fully-connected layers. Self-attention is an inductive bias that *works*.

So what can we conclude from all this?

Maybe the answer does reside in the *bitter lesson* after all. In his post, Sutton argues that building models that are carefully designed to capture known phenomena in the data can be counterproductive. But he also advocates focusing solely on **scale**, **learning** and **search.** And the Transformer design focuses on all three.

Self-attention is known to **scale** well as it can be performed in parallel with highly optimised matrix multiplications.

Learning in a Transformer is facilitated by skip connections, ensuring good gradient flow even at the bottom layers of very deep models.

But, most importantly, self-attention is essentially a **search** mechanism. Multi-head self-attention enables the Transformer to (learn to) 'search' in parallel over many possible interpretations of a sentence, all in one feed-forward pass of the network. Consider the sentences:

Time flies like an arrow.

Fruit flies like a banana.

To make sense of these sentences, we can almost feel our mind searching; searching for the right sense of the words *flies* and *like*, (Dictionary,com lists 34 possible senses for the word 'like'). And it is precisely this sort of search (over word senses) that multi-head self-attention enables, all in a single forward pass of the Transformer.

So what can we conclude?

I. If it's possible for a human domain expert to discover from the data the basis of a useful inductive bias, it should be obvious for your model to learn about it too, so no need for that inductive bias.

II. Instead focus on building biases that improve either scale, learning or search

III. In the case of sequence models, any bias towards short-term dependency is needless, and may inhibit learning (about long-term dependency).

IV. Skip connections are good because they promote learning.

V. Most importantly: Self-attention is good because it (together with skip connections) enables an online, feed-forward, parallel search over possible sentence meanings.

So in some senses, the bitter lesson says it all. But, as a linguist, I find the self-attention bias to be a beautiful reflection of the way language works.

It just flies like a banana.

And that is why, for me at least — and hopefully for anyone interested in language and AI — it's an agreeable, not a bitter, lesson. 😄
