**Machine learning for everyone**

*Lindsay Edwards, Autumn 2017*

Introduction

Hi there. My name’s Lindsay Edwards and I’m Head of the Respiratory Data Sciences group. Welcome to “Machine Learning and AI for everyone”. My goal today is to introduce machine learning – including a certain type of artificial intelligence called Deep Learning – to an audience of non-data scientists (hopefully you!). So let’s start with a quick poll – who here is a data scientist, or already uses machine learning routinely?

Even if you are a data scientist, I hope there’ll be enough in here to keep you interested.

Now one way to tackle machine learning is through something called computational thinking, and that’s the approach I’m going to take today. It’s very intuitive, especially for experimental scientists. However, this means that there will be code. So let’s have another quick poll:

1. Who already has at least one programming language? Not in depth, but do you know what ‘for loops’ and ‘if statements’ are?
2. Who would like to learn a programming language if they had the time?
3. Who thinks it will never happen?

Programming is a skill, like drawing. And arguably the best way to learn a language is to learn a few basics and then get stuck in. Today I’m going to run some live code using a very easy-to-read (and learn) language called Python. Hopefully, in the process, you will start to see how easy it really is. If you are interested in learning more about Python – or just programming in general - come and find me, there are some great resources I can recommend to get you started.

Let’s begin.

Definitions

Artificial Intelligence - AI - is everywhere at the moment, whether it’s [SLIDE] “Use this calculator to see if robots will take your job” or “End of humanity within decades” or (my favourite) “AI will \*not\* kill us…says Microsoft”. But what is AI? And what is machine learning? In the next half hour, I’m going to explain what machine learning is, discuss how machine learning is different to statistics, and put forward the hypothesis that experimental scientists are more naturally suited to a machine learning view of the world than a statistical one. Finally, I’m going to talk about Deep Learning, a type of machine learning that is behind all of the big advances in AI that you may have heard about recently.

[SLIDE]

Let’s start with definitions. Surprisingly, neither machine learning nor AI are new terms, and both date back to some time in the 1950s. Neither are terribly easy to define but let’s go with [CLICK] “apparently intelligent behaviour by machines” for AI and [CLICK] “machines learning tasks without being explicitly programmed to do them” for machine learning. Lots of AI is also machine learning – and vice versa – but there is one very specific type of machine learning – called [CLICK] Deep Learning – that is driving all the advances in AI that you would have heard of recently. More on that later.

Even though there is a lot of crossover, AI and machine learning are \*not\* the same thing. To illustrate that fact, let’s look at some counter examples.

[SLIDE]

Any AI that cannot learn is – by definition - not machine learning. So chess engines like Deep Blue – which famously beat Gary Kasparov in the 90s - and that had the rules of chess ‘programmed in’ are not machine learning. There was also a rather famous AI therapist back in the ‘60’s called ELIZA. ELIZA was cutting edge computation back then, but advances in technology means she can now live on your browser. Here she is…

6-7” [DEMO]

So… ELIZA was programmed with a set of rules to make her seem intelligent, but critically, she didn’t learn. So not machine learning. And not very impressive by today’s standards.

[CLICK]

Conversely, there are lots of machine learning applications that make no effort to appear intelligent. In fact, almost any machine learning these days that isn’t a neural network of some kind is not AI.

So: if it appears intelligent it’s AI, if it learns, it’s machine learning. But the two are not the same.

[SLIDE]

There is another, more operational, definition of machine learning that seems unwieldy at first, but actually gets us closer to the potential scope of ML.

“A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.”

This sounds very wordy, but actually just means that we have a [CLICK] **task** we want the computer to do, a way of measuring how well the computer **performs** that task; and **experiences** or data to feed it. If the computer uses the data and learns to get better at the task – that’s machine learning!

And with this definition, it’s easier to see how broad the scope can be. Here are just a few examples… [SLIDE]

* Classification
* Classification with missing inputs
* Regression
* Transcription (e.g. speech to text)
* Machine translation (e.g. natural language to natural language)
* Structured output (for example image captioning)
* Anomaly detection
* Synthesis and sampling (e.g. video game landscape generation)
* Imputation
* Denoising
* Gaming (Chess, Go, Atari)

10” Machine learning vs. statistics

I claimed, in my introduction, that machine learning is a more natural way of understanding data than statistics, if you happen to be an experimental scientist. Let’s unpack that statement a bit…

Statistics, as a discipline, was formed in an era when data was hard to come by, and computation was expensive. As a discipline, it contains two important ideas:

1. That unpredictable events (random variables) are best modelled via analysis of the mathematics of probability (for example, distributions). The resulting proofs can be impenetrable if you’re not a mathematician.
2. That the ultimate aim of these studies is to understand the world through the window of very limited available data.

Let’s use an example. If the average IQ of 100 chemists was higher than the average IQ of 100 biologists, does this mean that chemists are smarter than biologists? Or was it just a chance event? The statistical approach is to assume that these data fit a known model, and then leverage what we know about that model (from earlier mathematical analysis) to get an estimate of how generalizable this observation is. The problem for non-statisticians is that the number of models in their toolbox is limited, and because the theory is based on mathematical proofs, all the models come with a bunch of health warnings. If you violate these, the mathematics (and the models) break down.

There are other ways.

One alternative would be to actually run lots and lots of experiments. Recruit 200 scientists, over and over again.

Yet another would be to measure all the chemists and biologists and do away with the need for statistics at all. This used to sound like a daft idea, but in the era of Big Data, it can actually happen.

The final option (there may be more!) ... is to use a computer, and run lots and lots of simulations, perhaps re-using the original data.

I want you to notice what’s different about these last two. With Big Data, you may not need to generalise at all. You can actually get a measure of everything. Or at least, such huge numbers that generalisability is moot. Alternatively you can stick with your limited data, but leverage it with lots of (now) cheap, fast computation.

Let’s talk about p-values quickly, you’ll see why. P-values are the mainstay of modern inferential statistics. Let’s start with a definition (just because a surprising number of people – including the odd statistician – get this wrong):

p = the probability that, were you to run exactly the same experiment again, you would see a result at least as extreme as the one in front of you, *if in fact there was nothing there*

One way to think about probability is as the ‘long-run frequency’ (“if I did this a million times, what fraction of times would this thing occur?”). So we can recast our definition of a p-value as:

p = the fraction of times, if you ran exactly the same experiment a ton (say, 100,000 times), that you would see a result at least as extreme as the one in front of you, if in fact there was nothing there*.*

I wonder if you can compute a p-value without using traditional statistics at all. Let’s revisit out chemists vs biologists question, using some code...

[DEMO]

What I’ve shown you is an example of how we can use a computer, and computational problem solving, to get a statistical answer. But I only used p-values to make the point that you get near enough the same answer both ways. When doing machine learning we don’t tend to bother with p-values. Instead we normally take a measure of performance (say accuracy) and let the computer learn from the data while it improves accuracy.

There are lots of different machine learning approaches – confusingly we call them models - and they all have different names: random forests, support vector machines, gradient boosted trees. They work in different ways, but they all use the same philosophy. They use computing power to solve the problem, and leave the mess for the mathematicians to clean up later.

Before I move on to AI, there is a very very important concept I need to introduce, and that is the concept of overfitting. Machine learning approaches are very powerful, and can find patterns where there are none, much more so than statistical models. It is the classic double-edged sword.

Overfitting is where a very flexible model learns the data ‘off by heart’. Let me give you an example.

[PLOTTED DATA EXAMPLE]

1. 5 misclassified points
2. Perfect score!
3. 9 misclassified points
4. 6 misclassified points

Unlike statistics, we do not often have years of theory and proofs to help us understand whether we are overfitting or not. So as is so often the case in machine learning, we are pragmatic. The very best ways to protect against overfitting are: [SLIDE]

1. Use a model that is only as flexible as you need, and no more so
2. [CLICK] As soon as you get your data, take some, put it to one side, and as a final check, see how well your model works on this ‘new’ data. This is called validation.

Again, this second point should be very familiar to applied scientists. It is the ‘confirmatory experiment’.

Just be aware – not everyone takes the care they should to protect against overfitting. It’s very tempting to use very flexible, fancy models. If you do, it can be oh so easy to fool yourself into thinking that you have a much better model than you actually do.

OK, a last recap: Traditional statistics is about *knowing the theory* (including the assumptions). Machine learning is about *running the experiment*. You can use computational thinking to answer many questions, or better still, write some code!

AI and Deep Learning

OK, now finally to AI. All the advances in AI that you have heard about recently are due to advances in a branch of machine learning called Deep Learning. After this talk you will hear about four applications of Deep Learning and neural networks from across the business. But what’s different about Deep Learning, why is it gaining traction suddenly now, and what can we expect it to do for us?

Deep Learning is a type of neural network. Neural networks are a way of taking very simple models and stacking them together like Lego bricks. Not all neural networks are Deep Learning. As a loose definition, Deep Learning requires at least two hidden layers (more about this in a second).

Often when we draw neural networks, we draw them like this [SLIDE]. The data flows in here, and each node (little circle) is where some processing happens. Usually it comprises adding up the stuff coming in, and then applying some other transformation to it.

We’ve known about neural networks for many years, but advances in computing power and in the field itself have suddenly made them useful in a way that simply wasn’t possible till recently. So why are they so different to all the other machine learning models out there?

One of the first big successes for Deep Learning was object recognition in images. Who here uses Facebook? In the early days of Facebook you may remember that you needed to tag photos yourself. Now that’s done by Deep Learning. Deep Learning has absolutely smashed the previous records for tasks like this, and here’s why.

Because the networks are deep and complex, they are able to break images down into features, and then use those features to recognise objects.

[SLIDE] This is a rather nice visualisation of a relatively simple eight layer convolutional neural network that can be used for handwriting recognition.

[WALK THROUGH THE NETWORK]

But here’s the key: Deep Learning figures out the best features for itself. Because of this, it can recognise a face even if it’s obscured or rotated away, because it builds the face up from the features. A line here, a smudge there. OK, that’s Lindsay.

This xkcd slide sums it up nicely. Just a few years ago, recognising a bird in an image was considered virtually impossible. Today it’s commonplace.

And it’s this ability to break things down, to capture abstract – almost human – concepts, that makes Deep Learning so powerful. Whether it’s constructing sentences from ‘meaning’ or recognising faces in an image, Deep Learning is transforming our world. It is at its very best when consuming vast amounts of unstructured data – text, speech, images, music video and the like.

In fact, some of the most impressive examples of AI implementations recently take this concept a step further – by mapping one medium (say, audio or video) to this abstract feature space, and then mapping back out to something else (say, text).

EXAMPLES.

But it’s not all good news. Neural networks are incredibly flexible, which means they are wide open for overfitting. Typically they need a lot of labelled data, which we often don’t have. It also means that they may be out-performed by simpler machine learning models when the data are limited, even though it’s tempting to use Deep Learning just so you can wear the badge.

Summary

That’s it. I hope you’ve enjoyed the talk. Please feel free to follow up with me on these, or any related points. Thank you.