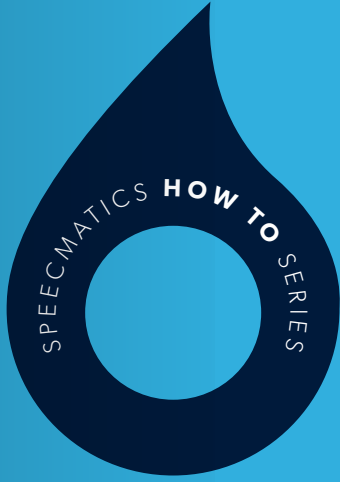


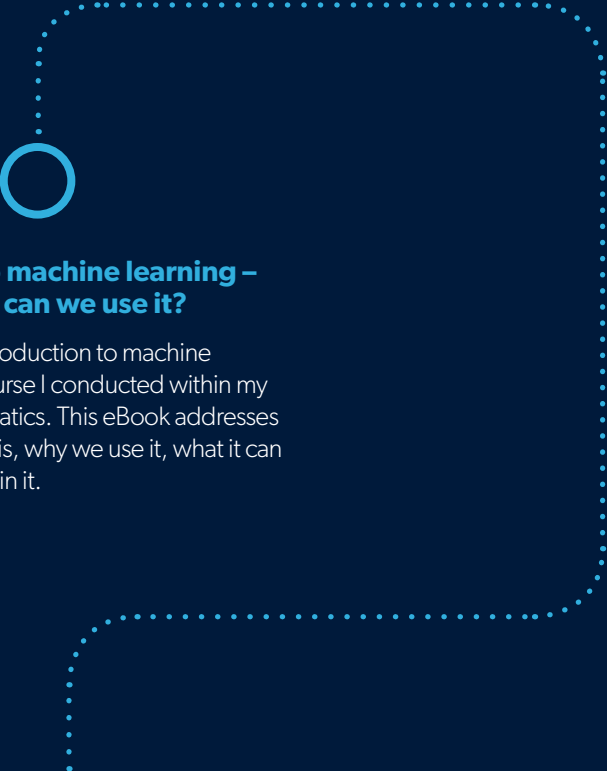


SPEECHMATICS



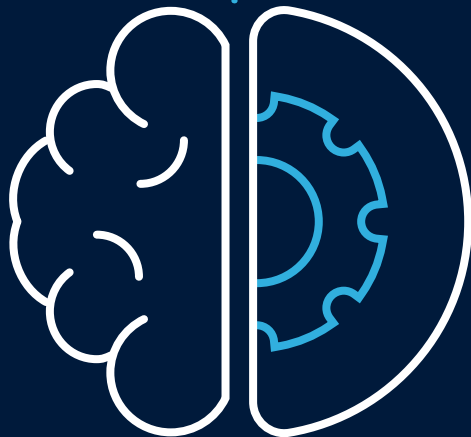
HOW TO

make the most of machine learning



An introduction to machine learning – what is it and how can we use it?

This is an entry-level introduction to machine learning based on a course I conducted within my organisation, Speechmatics. This eBook addresses what machine learning is, why we use it, what it can do, and how we can train it.



Part 1

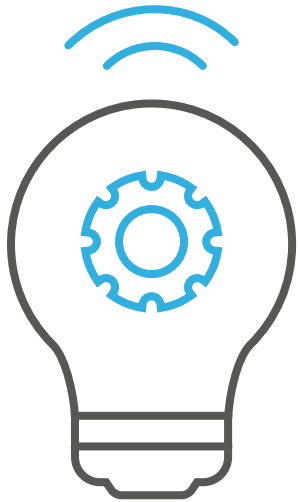
Is this eBook for you?

This eBook is designed for people with no, or very little background in machine learning and who want to learn more.

If you want to learn what some of the key terms and concepts mean then you have come to the right place; however, if it is deep technical detail you are after, you may be better served elsewhere. The eBook is adapted from a course I am co-ordinating within [Speechmatics](#) to help all of our staff better understand some of the machine learning progress we are making.

Why is everyone so excited about machine learning anyway?

You may have heard (perhaps from a friendly tech evangelist) that right now we are in a golden age for artificial intelligence. It isn't the first, and perhaps it's not the last either.



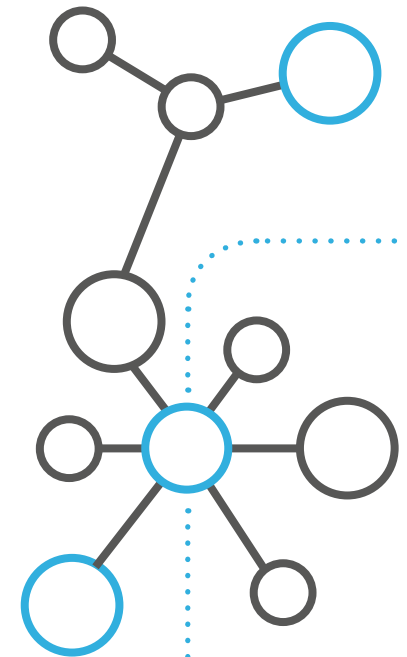
Your friendly tech evangelist (FTE) may also tell you that this has been driven, in large parts, by improvements in machine learning algorithms that have in turn been driven by hardware – especially graphics processing units (GPUs) – providing orders of magnitude more compute than previous generations had access to. Your FTE may then go on to say that this means we can exploit larger datasets and build bigger models that can do more remarkable things. The FTE is also likely to be excited by the vibrant open source community that has opened up access to this technology faster and more easily than we have previously seen in technology development cycles.

The friendly evangelist might not have spent long explaining what all of this meant though, and it is easy to get lost in all the jargon. Hopefully by the end of this eBook, all of the above will start to make a little more sense.

What is machine learning? How does it differ from artificial intelligence or statistics?

Let's start with some definitions of terms. There are lots of overlapping fields around machine learning. I won't try to cover them all in this eBook, but I'll at least look over two of the larger ones: artificial intelligence and statistics, and try to find some dividing lines between them and machine learning.

Note that the lines between these fields are not in reality, sharply drawn. They are moving over time and on occasion, hotly debated. The following are some insights I find useful but are by no means the whole picture.



SIMPLE DEFINITION

A simple definition of machine learning to start. Machine learning involves turning a dataset into a model. The model is then used to perform some task based on fresh data inputs.

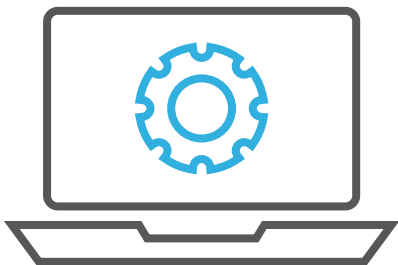
Machine learning vs statistics

Certain statisticians will tell you that this current fad for machine learning is nothing but repeating what they (the statisticians) have been doing for decades – even centuries! Indeed, my definition above sounds very much like the bread and butter of a statistician – making a model out of data. In many cases, they are right in that there is a lot of overlap between the fields, particularly in the techniques used.

At a high level, I like to think that a key difference between machine learning and statistics is the direction in which they look:

Statistics is about looking **backwards** at data, to determine what it tells us about the environment from which that data is drawn.

Machine learning is about looking **forwards** from data, to create a tool that can predict new events in similar environments to that from which the data is drawn.



The following table summarises this and a few other differences that I like to draw between the two (with examples in italics).

STATISTICS	MACHINE LEARNING
<p>Make conclusions</p> <p><i>'Data says hippies are 90% more likely to take drugs.'</i></p>	<p>Make predictions</p> <p><i>'We can predict what drugs a hippie has taken from a photo of their face.'</i></p>
<p>Focus on rigour of results</p> <p><i>'Our results have a 10% chance of being incorrect.'</i></p>	<p>Focus on making cool things happen</p> <p><i>'Our results are awesome – look at them!'</i></p>
<p>User provides the model structure</p> <p><i>'Assume all participants can be modelled by a Gaussian curve.'</i></p>	<p>Model 'learns' its structure</p> <p><i>'We used a convolution layer to learn to detect edges in pictures.'</i></p>
<p>Smaller number of variables, simple structure in noisy data</p>	<p>More variables, more complex structure in data which is visible over noise</p>

I like to learn by example, so here is how I'd like to think mythical stereotypical statisticians and machine learning researchers might treat some plane crash statistics:

Statistician:

I have a hypothesis that plane crashes are correlated with Easterly winds. If we take the weather data and run this specific correlation model with crash statistics, we can see that they correlate with less than 5% chance of being incorrect.

Machine learning researcher:

Let's gather all the data we possibly can and put it in one big model without any preconceptions. I think we can teach it to predict which planes will crash with an accuracy of 90%.

To finish off, here are some amusing quotes I found from discussions about the difference between these two fields:

"In that case, maybe we should get rid of checking of models and assumptions more often. Then maybe we'd be able to solve some of the problems that the machine learning people can solve but we can't!

"Machine learning is statistics minus any checking of models and assumptions."

"I don't know what machine learning will look like in ten years, but whatever it is I'm sure Statisticians will be whining that they did it earlier and better."

Machine learning vs artificial intelligence

There seem to be lots of variation between definitions for artificial intelligence, so I'm not going to give you just one definition, but four!

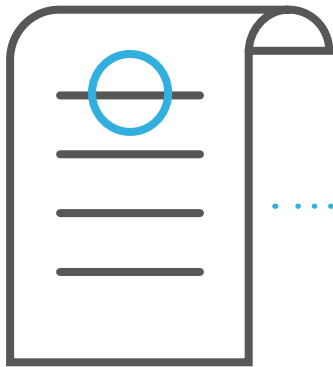
A RATHER GENERAL DEFINITION:
 'Intelligence' demonstrated by machines, rather than humans/other animals.

THE 'SOFT' DEFINITION OF AI:
 A tool capable of 'intelligent' actions within a specific domain.

THE 'HARD' DEFINITION OF AI:
 Something broadly able to complete any intellectual tasks a human could.

THE 'AI' EFFECT:
 AI is anything that hasn't been done yet.

The last one is a rather tongue in cheek observation that once a problem is solved, such as beating humans at chess, the solution comes to be labelled as something other than intelligence, even though a human solving the problem in a similar way may be regarded as very intelligent!



More generally, the important thing to note is that all of these really revolve around applications of AI – not in how the AI itself is generated. If we were to invite humans to write rules on how they would respond in specific scenarios, with sufficient work there is no doubt we could create something that could fulfil the criteria on page 11 (or at least the first three!). Though, that might not be a practical thing to do...

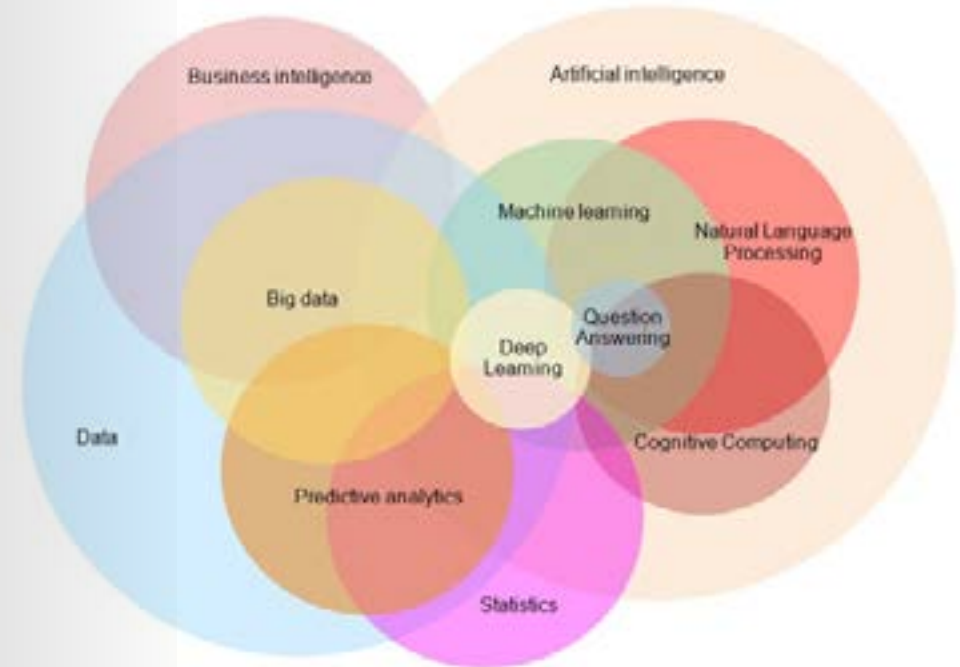
Which is where machine learning comes in. Machine learning is one way to build the systems, directly from data, that exhibit varying levels of artificial intelligence, and hence it is typically regarded as a subfield of AI.

But what about big data, deep learning and other buzzwords?

These three are only a part of the jargon in the field. There are many other words. I won't try to explain them all – instead, I will use the image on page 13 to show how at least one person views the overlap between variations of these fields.

Note: This image contains our three terms as we have described them. Machine learning nestling in as a subfield of AI and statistics overlapping both.

Big Data Dictionary



Credit: sastat.org.za/sasa2017/big-data-dictionary

FINAL THOUGHTS

Some final summary thoughts on machine learning vs AI vs statistics

- Labelling stuff is hard!
- Machine learning is probably a subfield of AI
- Machine learning overlaps many other related fields
- At its core, machine learning is about getting a machine to learn something from data, with as few preconceptions as possible.

Why do we use machine learning?

Sometimes I get pushback from some particularly old school engineers asking why we don't just write a set of rules instead of spending all this time and equipment on complicated machine learning code and training models. In particular, there is one colleague who insists if we tried hard enough, we could replace all of our machine learning code with regex...

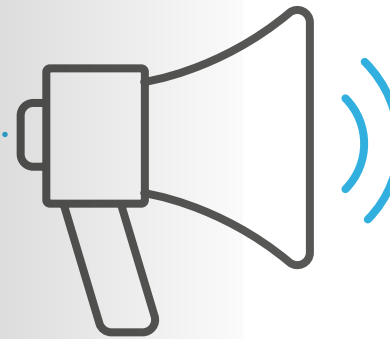
So, here are some reasons why we use machine learning rather than rules-based systems, expert systems or other systems that rely on the domain-specific knowledge and insight of the designer, rather than tasking the system to generate knowledge and insight for itself. I will use speech recognition as a use case to demonstrate my points, comparing a machine learning system to a human trying to write a series of rules, in italics.

1. Experts have their limits

We cannot write rule-based programmes for everything humans can do as we simply don't know how our brain does it in sufficient detail.

The precise details of how we, humans, convert sounds into words in our minds are incomplete. We know some of the pieces, but not all of them.

Therefore, any expert system of speech recognition will be starting from incomplete knowledge and will have to make approximations and guesses. Without that additional 'edge' over machine learning approaches, we can't reasonably expect to outperform an approach that tries to generalise directly from data.



2. Complexity

Complex tasks would require increasingly complex programs to solve. Every new edge case would need more code. Machine learning needs minimal code for the complexity it solves.

In principle, to add a new audio context (perhaps people speaking over loudspeakers) to a model just means training your model on new training data – nothing else in the codebase necessarily needs to change.

Our expert, however, will need to add new code to deal with the situation – perhaps a loudspeaker distortion cancelling module. However, they need to be careful that doesn't interfere with the car noise filter they added a week last Friday, or that patch they put out a year ago to prevent people with colds being misunderstood, or...

3. Generalisability

An expert system will work for the cases the designer has thought about. A machine learning system will (should!) work for cases it has never seen before.

A speech recognition system will have been trained on thousands of hours of audio data. That will include hundreds or thousands of different voices, each one with slightly different characteristics that it will take into account.

It is not practical to expect a human to take all of those different voices into account when crafting a rules-based system – there are only so many things a human can hold in working memory at once. This will lead to the system being far more proficient at the voices the expert has focussed on, to the detriment of others.

Of course, there are use cases where expert systems are still outperforming machine learning. Particularly in situations where suitable training data for machine learning is sparse. But in plenty of fields, we are now seeing that those expert systems simply can't keep up and are being outperformed by machine learning equivalents in terms of absolute performance, in the speed of development, and in the number of person-hours required to build a system.

What can we do with machine learning?

There are many problems machine learning can solve. In this section, I am going to talk about four broad fields of use case for it, with examples of each.



Classification

Classification is perhaps the 'classic' use case for machine learning. Classification means working out what class a data point belongs to. A simple task might be, for example, looking at a picture of a pet and asking if it is a dog or a cat.

A machine learning model would typically solve this problem by taking the raw input data (in this case the RGB pixel values of the image of the pet), running it through a model that eventually converts it into a vector of length equal to the number of classes, each item in the vector corresponding to some form of probability output for the appropriate class (in this case cat or dog).

Some more example use cases:

- Given an x-ray, does this patient have a broken bone or not?
- Does this iris scan match anyone in our database?
- What telephone number did our caller utter?

Regression

In regression, the task is to predict a real-valued number based on a data point. This is not constrained to a limited number of possibilities but could be any number (perhaps within a defined range). A simple task might be, for example, looking at a stock's trading price and predicting what it will be next morning.

A machine learning model would typically solve this problem by taking the input data (in this case all of the historic data for that particular stock), running it through a model that eventually converts it into a single number, perhaps with some confidence intervals (representing the next day's trading number).

Some more example use cases:

- Given these personal details, what is the percentage chance this person will default on their mortgage?
- How many people are likely to turn up for the flight to Hannover at 10am tomorrow, given historic attendance rates and ticket purchasing information?
- What temperature will it be tomorrow, given these satellite images?

Clustering

Clustering involves grouping data points in such a way that examples in the same group are more similar to each other than those in other groups. A simple task might be, for example, dividing rainbow marbles into piles of similar colour.

A machine learning model would typically solve this by taking all the data points and then outputting a cluster label for each data point. The numbers and names of the clusters may or may not have been predefined by the user, or the machine learning model may generate them. The model might, for example, decide to separate the marbles into four piles, corresponding to marbles that are closest to being green, blue, yellow and red.

Some more example use cases:

- Dividing customers into buying groups in order to analyse common purchasing patterns
- Analysing crime prevalence and locations of particularly high crime
- Document clustering to find commonalities

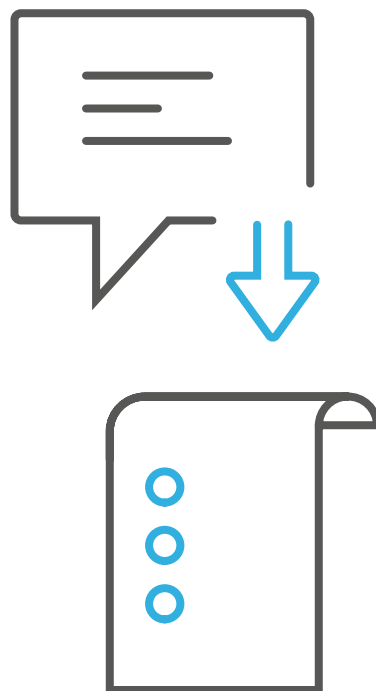
Dimensionality reduction

Dimensionality reduction is reducing the size of a dataset by either deleting or merging some of its variables. A simple task might be, for example, compression of an audio track.

A machine learning model would typically solve this problem by taking some data and outputting it in a new form that was much smaller than before without losing the key information. The model might, for example, keep only the frequency components corresponding to a particular human voice in an audio recording.

Examples include:

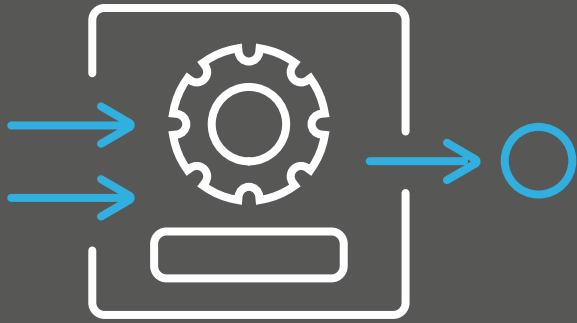
- Summarisation of a text document
- Data compression
- Often an intermediate step in other machine learning techniques to reduce compute demands



Summary

I always like a small table to summarise things.

TASK	INPUT	OUTPUT
Classification	A data point	A probability for each possible class
Regression	A data point	A numerical prediction
Clustering	A set of data points	A cluster label for each data point
Dimensionality Reduction	A data point	A data point with fewer variables (smaller!)



Part 2

How do we train those machine learning models?

There are various ways to train machine learning models. Much like in Part 1, where I gave four general types of problem machine learning could solve, I'll now give three general ways in which to train a machine learning system.

These methods all hold one thing in common. A model is created by taking some data, using it to update the model such that it performs better on that data, then repeating the cycle many times. This typically involves pushing the data through the model, looking at the outputs and then calculating how wrong it is compared to some metric. This 'wrongness' we then call an error signal, and we adjust the model in such a way that if that same data were to be pushed through the model, it would do a bit better the next time. This is sometimes called iterative learning.

Supervised learning

Supervised learning is probably the most straightforward case. With supervised learning, a machine learning model is fed **labelled** data during its training phase. When each training data point is fed to the model it is then very easy to detect how wrong it is – you simply compare the output to the label, and the difference can be used as an error signal to correct the model.

The key challenge in supervised learning is making sure your model ‘converges’ on the best values that it can. One of the core problems within this is making sure that it is using the correct ‘learning rate’ at all times – essentially a parameter that tells it how much the model is allowed to change every time it ‘learns’ in an iteration. If the learning rate is too small you may end up in what is called a ‘local minimum’ – a state which seems to be optimal, but actually does not have as good performance as if it used more radical steps in learning. Alternatively, your learning rate may be too high, and the model may never settle (‘converge’)

on any particular values, instead moving around rapidly without any coherent pattern. Balancing these two possibilities is a first step to creating a high-quality supervised learning algorithm.

Supervised learning is typically used in classification and regression use cases.

Some examples:

- Image identification: each image is labelled (cat, dog, etc.)
- Spam detection: each training email is labelled as either okay or spam
- Handwriting recognition: each image of a word is accompanied by a label of what the word is

Unsupervised learning

With unsupervised learning, the machine learning model is not given a label, and the task is typically to work out what underlying patterns and structure there might be in the data. Clustering is perhaps an easy way to understand this. In clustering, the model is trying to extract similarities and contrasts between data points that may not be immediately apparent.

The main challenge within unsupervised learning is dealing with the lack of prespecified objectives for what you would like it to output. Designing an algorithm that can nonetheless produce meaningful and useful results requires careful thought. For example, in blind clustering, you do not have labels on the data, but you may try to optimise the model so that it maximises the between-cluster distance and minimises the within-cluster distance. We can use those distances to form the error signal.

Unsupervised learning has a lot of potential, as it has the most data available to train on. It is far easier to obtain unlabelled data than it is to obtain data with clean and meaningful labels to train from.

Unsupervised learning is typically used in clustering and dimensionality reduction use cases.

Reinforcement learning

Reinforcement learning is a little bit different and more complex again. In a reinforcement system, people normally talk about a ‘reward system’ during training, whereby a model is given a ‘reward’ when the actions it takes lead to good outcomes. The model then changes its parameters to try and make sure it takes the actions that lead to higher rewards, rather than actions that lead to lower rewards.

In a sense, this is similar to supervised learning, in that there are labels. However, the labels are not as clearly assigned to each data point. In reinforcement learning, for example, the model may take an action that does not lead to any rewards until tens or hundreds of actions later. In other cases, the model may get a reward immediately after it has taken each action.

It is also complicated by the exploitation/exploration trade-off. A good reinforcement learning algorithm should explore the possible set of actions it could take to make sure it is not missing a better approach, whilst also exploiting the best sets of actions it has discovered so far.

Setting out a clear reward structure that can compensate for outcomes not being clear until a long time after actions have been taken, and getting the exploration/exploitation trade-off right, are two of the biggest challenges in reinforcement learning.

Some examples:

- Models that can play various games, such as Chess, Go or StarCraft (the reward is winning the game, so it only gets granted a long time after the early actions in a game are taken)
- Automated car driving (the rewards here include not crashing, getting to the destination, and getting there fast).



The role of GPUs

I mentioned at the start that graphics processing units (GPUs) have been a key part of the current golden age in artificial intelligence and machine learning. But why?

The simple answer is that **using these devices has unlocked massive amounts of compute for relatively little cost**. Lots of the techniques used in machine learning have been around for decades, or even longer, but were unable to be scaled up enough to produce results that could compete with other approaches.

Meanwhile, gamers wanted bigger and better graphics on their video games and specialised units were created that could perform many processes in parallel to achieve that. It turns out that many of the operations required to perform machine learning training (such as matrix multiplication) are also very well suited to massive parallelisation, and so machine

learning researchers started using them to train their models. Using a single GPU is estimated to have given at least a 10-100 times speed up, compared to using a CPU. More recently, some researchers have started using many of these GPUs, or even similar custom-designed hardware for machine learning in parallel with each other, giving even more orders of magnitude faster processing.

To give you an idea of just how powerful these hardware units are, a single Titan X (a popular GPU card for machine learning enthusiasts, released in 2015) can perform more floating point operations per second than the world's fastest supercomputers of the twentieth century. Using these devices has led to an explosion in model size used in machine learning, and from that, a hockey stick improvement in the performance of these models.

That's (nearly) all folks!



I hope you've enjoyed this brief introduction to machine learning and that it can help more people start on the journey to understanding this rapidly advancing field that is primed to revolutionise many aspects of our lives. There is a lot more to learn...

Quiz Finish!

When I ran this as a course at Speechmatics, I ended this session with a quiz – so let's do so here too! There are six questions. In the course, I got everyone in the room to stand up and then got them to sit down once they got a question wrong. One person managed to survive the whole way and stay standing! Can you be a machine learning superstar as well?

Questions...

1. Was the first artificial neural network model proposed before or after the start of the Second World War?
2. Was Backpropagation (the main way we train neural networks) described before or after the Simpsons first appeared on TV?
3. Was Tony's landmark paper 'A recurrent error propagation network speech recognition system' published before or after Bill Clinton became president?
(This refers to Tony Robinson – the Founder of Speechmatics)
4. Did Deep Blue beat Garry Kasparov at Chess before or after Charles and Diana divorced?
5. Were GPUs first used for machine learning before or after the last time Italy won the World Cup?
6. Was TensorFlow (the most widely used machine learning platform today) released before or after Trump announced his bid to run for President?

[Answers on the next page!](#)

Quiz answers...

- 1. After!** McCulloch and Pitts created a computational model for neural networks in 1943, the Second World War started in 1939.
- 2. Before!** Backpropagation was first described in 1986, The Simpsons first appeared in 1987.
- 3. Before!** The paper was in 1991; Clinton became president in 1993.
- 4. After!** Charles and Diana divorced in 1996; Deep Blue faced Kasparov in 1997.
- 5. After!** CUDA was first released in 2007 and the first papers really demonstrating it for deep learning were in 2009. Italy last won the World Cup in Germany in 2006.
- 6. After!** Trump announced his bid in June 2015; Tensorflow was released in November 2015.



SPEECHMATICS