Applied Artificial Intelligence for Law: The Use of Expert Systems and Decision Trees in Legal Practice and the Potential of Artificial Neural Networks

Alex Matheson[1]*

[1] *The University of Oxford*

Abstract: The literature on Artificial Intelligence (AI) as applied to legal practice can contain apparent paradoxes. On one hand, some researchers suggest that AI can transform law. On the other hand, others suggest that AI has more limited potential. This apparent tension can be understood by distinguishing different forms of AI technology. This paper reviews two contrasting types: First, it examines expert systems and their component 'decision trees'; second, it considers artificial neural networks (ANNs). Finally, it reviews the use of the former within the legal sector and the potential of the latter. Expert systems have fixed 'knowledge' that is codified into computer systems by human experts. To some extent, expert systems enjoy wide legal-sector adoption–at least for the 'decision tree' part of the technology. Improvements for the legal sector can be made by enriching decision trees using methods from decision science and by using algorithmically generated decision trees. ANNs are quite different. They are vast networks of connections within software systems that 'learn' from rich data through exposure to it. The mechanism of ANNs is based on the principle by which neuronal connections in biological brains become stronger – via exposure and training. This paper finds that ANNs have significant potential but face more limited legal-sector adoption than decision trees. Other forms of AI exist but are beyond the scope of this paper.

Keywords: legal artificial intelligence; legal AI; expert systems; ANNs; artificial neural networks; legal technology; legal practice; machine learning.

* alex.matheson@sant.ox.ac.uk

Barrister (England and Wales), Higher-Rights Solicitor (England and Wales), D.Phil (Candidate, Oxford), MBA (Graduand, London Business School), LL.M., M.A, PG.Dip, PG.Dip, LL.B.

Introduction

A very broad classical definition of Artificial Intelligence (AI) describes the field as that where machines are developed that can undertake tasks that would otherwise require human intelligence (Turing, 1950; Minsky, 1972; Raphael, 1976). This definition is also used for legal AI (Armour and Sako, 2020). The technologies that fall within the field thus change over time. For example, in recent decades computers playing chess were at the height of the imagination of many AI researchers (Turing, 1950; Hofstadter, 1979), but today electronic chess computers are commonplace (Lowry, 2021) and do not use the label 'AI'. John McCarthy is credited with saying *"as soon as it works, no-one calls it AI anymore"* (Meyer, 2011). For reasons like these, technical commentaries typically recognise that the field of AI is broad and changes over time (Russell and Norvig, 2016).

AI spans across many disciplines including, but not limited to, robotics, computer vision and artificial decision-making. The field of AI can be traced back to Alan Mathison Turing's 1936 work on a problem of computability (Turing, 1937; Stanford Encyclopaedia of Philosophy: 'Alan Turing', 2002). Others suggest with strained interpretation that it can be traced further back to Charles Babbage (Schwartz, 2019). The field was proposed by Turing in a public lecture in 1947 and has been developing in various forms since (Haenlein and Kaplan, 2019) albeit that research slowed for a period after attracting comparatively few researchers for some decades (Hofstadter, 1979). This slow period is often called the AI winter. Mathematics and statistical methods of logical problem-solving have formed a part of AI since the inception of the field and much research in this area progressed in the 1970s (Hofstadter, 1979).

During the 1980s and 1990s, 'expert systems' (described below) were the preeminent form of AI (Buchanan, 2005; Bench-Capon et al. 2012). Expert systems build upon both symbolic logic and the science of logical 'decision trees' to codify the knowledge of experts (including lawyers (Susskind, 1989)) into software programs. Expert systems research declined rapidly after the 1990s and there have been few real breakthroughs in this area since. Much of the legal AI in practice today stems from this period.

Another strand of AI, artificial neural networks (ANNs), was originally developed in the 1950s (Rosenblatt, 1957) but gained little traction for over half a century until a breakthrough in the early 2010s occurred, following which this strand has flourished. ANNs are the cutting edge of contemporary AI systems. ANNs are structures of code that are programmed to synthetically mimic biological neurons, their interconnections and their methods of learning within current technological constraints. ANNs are a type of machine learning which can be supervised or unsupervised (Berry and Others, 2020). 'Supervision' in this context typically means the training examples which are fed to a computer system for the purpose of learning are labelled with the target solution.

This paper first considers expert systems and their core technology, decision trees. It then examines limitations of this form of AI before turning to consider ANNs and how this newer AI technology can overcome the limitations of expert systems and decision trees.

Expert systems and decision trees

Expert systems

The phrase 'expert system' is used to describe a computerised diagnostic or decision-making system that asks a structured series of questions in the same way that a human expert might and then provides a suggested diagnosis or outcome (Bench-Capon et al. 2012). These expert systems rely heavily on codified expert knowledge and seek to mimic the ability of a human expert to reach a conclusion based on a subject's responses to a series of questions. In the series of questions, the next question to be asked will usually depend on the response given to prior questions (Hofstadter, 1999). This summary definition is expanded upon further below.

As summarised above, when codified expert knowledge is used in a computer system to allow a system or a non-expert operator to make decisions that would otherwise require a human expert, the resulting system can be described as an 'expert system'. An expert system often aims to replicate the interactive consultations that a human can have with a human expert in a particular field, i.e., instead of consulting a human expert, a person can consult the expert system to receive a diagnosis or prognosis. Early expert systems were installed from floppy disks to computer terminals and could be operated by a computer operator.

The phrase 'expert system' was unheard of before the 1960s, it had some limited usage over the next two decades and then spiked in popularity–at least amongst AI researchers–during the 1980s and early 1990s (Haenlein and Kaplan, 2019) (see also Google Ngram graph of the prominence of the subject 'expert system' over time across literature: <https://tinyurl.com/2wbbmd5x>). It has since dropped from the vocabulary of researchers and the area has been overtaken by the modern AI fields of machine learning and ANNs which are overwhelmingly more popular areas of research today. Expert systems could come back into fashion amongst AI researchers if a major future technological breakthrough occurs. This happened for ANNs from around 2012, which had long been in decline since the 1950s before the resurgence (Haenlein and Kaplan, 2019; Krizhevsky et al. 2012) (See the following Google Ngram graph of the prominence of the subjects 'neural networks', 'machine learning' and 'expert systems', again over time and across literature: <https://tinyurl.com/33j7w9bb>.) Alternatively, they could simply remain in use by legal practitioners owing to their predictability and transparency–as outlined below.

Expert systems were considered by early AI researchers in the 1980s to represent the most likely form of AI to penetrate legal practice (Buchanan, 2005; Bench-Capon et al. 2012) partly because they use a system to mimic consultation with a human expert. Expert systems are however today considered to be a historic form of AI by modern AI researchers.

As shown below, expert systems have not fully penetrated legal practice as expected, but many modern legal technology systems that provide legal process automation (which would not themselves be called 'expert systems') do use the 'decision tree' component of early expert systems as a method of codifying expert legal knowledge. The 'decision tree' part of expert systems technology has proven to be the most useful and resilient to technical change. Despite much technological potential for decision-making, it is shown below that decision trees are currently used in legal practice most frequently for simple automation tasks. They are used as mere tools to streamline the navigation of knowledge rather than as tools to augment cognitive processes or to allow systems to actually make optimal decisions. This is an area that can be improved upon. The adoption of expert systems by practitioners and the lack of adoption of ANNs (see below) suggests that the legal profession has been more comfortable allowing technology to augment human knowledge than to augment critical reasoning skills.

During the 1980s and 1990s, expert systems were viewed as the most promising field of AI research for application in the legal profession (Aggarwal, 2018 p.453; Ng, 2017), partly because solving many legal problems relies on the application of codified knowledge such as the multi-limbed tests of statutes and case law as well as on rule-based decision-making. Expert systems were also very much in vogue generally across many potential fields of application and showed great promise in many other domains that rely heavily on structured knowledge including medicine and social science (Buchanan 2012; Bench-Capon et al. 2012). Although research in the area has declined since the 1990s, expert systems still have two distinct advantages for legal practice.

The first advantage is that most routine legal practice is highly codified. Whilst machine learning might lend itself well to numerical problems and whilst ANNs might lend themselves well to the thorny problems of legal philosophy or more subjective legal problems (see below), expert systems lend themselves well to the standard codified types of problems that arise in routine day-to-day legal practice. Not all tasks however are conducive to expert systems: risk modelling and risk scoring is far better suited to statistical methods and machine learning (Khoylou, undated; Anderson 2007; Siddiqi, 2006; Lewis, 1990).

The second advantage is that expert systems do not suffer the 'black box' problem of other forms of AI such as ANNs (see below) (Aggarwal, 2018). An expert system might use a decision tree that contains millions of branches, but each can realistically be inspected and understood by a human at some level, unlike the billions of variously weighted connections in an ANN (see below) where even the basic task of estimating the required number of branches is a subject of research (see below). The feature of expert systems that gives rise to their openness to scrutiny also makes their outputs predictable. Earlier AI methods were far less sophisticated but more transparent (Hinton, 1976).

Given the previous practical problem of updating expert systems as a field developed (it involved duplicating and posting floppy disks with updates), it was historically the case that relatively stable areas of law were most suitable for expert systems. That constraint is now somewhat historic (Bench-Capon et al. 2012; Susskind presentation to Vanderbilt Law School, 2016).

Given the above two advantages, it is not surprising that a highly cited author in the field of legal AI, Richard Susskind, is from the arena of expert systems (Susskind, 1986) rather than e.g., the arenas of machine learning or ANNs (this could alternatively simply be an accident of timing for Mr Susskind as expert systems were the predominant form of AI at the time of his early doctoral research (see Google Ngram viewer: https://tinyurl.com/2wbbmd5x.

Susskind gives the (non-legal sector) example of TurboTax (Susskind, 2015; Also discussed at Online Courts Hackathon, 2017) as a modern-day expert system which has been highly successful in helping individuals with personal tax returns in the USA. TurboTax can, in many cases, bypass the need for hiring an accountant and is used by tens of millions of Americans (see https://turbotax.intuit.com) to prepare and file their tax returns.

Susskind has stated that many of the software tools that have penetrated legal practice today are based on the science of expert systems (Susskind presentation to Vanderbilt Law School, 2016). This view is true in current practice for the 'decision-tree' part of expert systems (i.e., the knowledge-base part), but it is less so for the inference engine part, i.e., the reasoning engine, or the 'symbolic logic' part of AI, (see distinction below). It is not entirely the case that expert systems as they were imagined in the 1980s and 1990s are pervasive in modern legal practice, but it is the case that the knowledge-base part of the technology is generally well adopted and that decision trees specifically are broadly adopted.

Components of expert systems

Expert systems have two key components: logical rules and a knowledge base. Other components include a user interface and an import/export capability. The rules govern how decisions are made and the knowledge base contains all of the decision options available. The logical rules are sometimes described as forming an 'inference engine' and the knowledge base has often been in the form of a decision tree. Other data structures exist for the knowledge base, but decision trees have the advantage that they use very simple and transparent logic. Like flowcharts, options are split into forking branches at the point of each decision. Furthermore, an expert can often easily codify his or her expertise into a decision tree but may struggle to do so in another data structure. To generate an expert system, human knowledge needs to be computerised. This has historically been done manually.

Decision trees

The phrase 'decision tree' is used to describe the most common way that computer systems structure the knowledge that is used for systems like expert systems. There are many conceptual methods of structuring knowledge for computer systems, including using linked tables, taxonomies and lists, but the 'decision tree' structure which would typically look like a sideways tree of forking branches if depicted, is a particularly useful method of codifying human knowledge. It is like an advanced flowchart in computer-readable form. Using a decision tree, an 'expert system' can navigate through the branches of a series of linked questions and can reach an appropriate output.

As outlined in the summary definition above, decision trees in computer systems are tools for storing expert knowledge in a structured fashion. The above definition can be expanded upon as follows: Decision trees (as far as decision science is concerned) can do slightly more than act as taxonomies of knowledge since they *can* contain the *rules* regarding which paths should be followed. This aspect of decision trees has not been adopted widely in the legal sector. Decision trees can thus contain codified knowledge with the intrinsic structure containing e.g., a simple form of logic (Hofstadter, 1979; Newell and Simon, 1976).

Two potential improvements for the use of decision trees for legal practice are described at the end of this section. These are 1) to use algorithms to generate legal decision trees automatically and 2) to enrich legal decision trees using proven methods from decision science so that humans can be taken out of 'the loop'. These improvements would 1) reduce the labour-intensive human burden of codifying knowledge for systems and 2) take legal technology systems beyond mere automation of knowledge access and closer to the realm of decision-making. These proposed improvements would only partially mitigate the limitations of expert system technology. More progress can be made with a different AI technology: ANNs which are discussed after this section.

How decision trees have penetrated legal practice

Richard Susskind's expert system from the late 1980s was a computerised method of navigating the then-emerging Latent Damages Bill (Susskind, 1986). Susskind called the emerging new law the 'latent damages system'. This became an Act in 1986. The decision tree that was developed for that statute streamlined a lawyers' work navigating through a vast number of potential paths through the complex legislation. It reduced the amount of legal research time needed to do so and meant that a human legal expert was not needed to navigate the Act. A non-expert could respond to a series of prompts on a computer system to do so. Originally, such a system could only be developed as a collaboration between a computer scientist and a legal subject matter expert, but today, software tools allow lawyers to codify their knowledge directly without the need for lawyer-programmer collaborations. One tool for computerising fields of knowledge without the need for a software programmer or computer science skills is called 'PrecisionTree'. The possibilities arising from expert systems have only marginally been realised by the legal profession (see below).

The adoption of decision trees by law firms

In recent years, a number of so-called 'no code' solutions have emerged for legal and other professionals. These are software tools that allow lawyers, and others, to build simple automation systems, usually based on decision trees (Common tools are called Bryter and Microsoft Power Apps: https://bryter.com and https://powerapps.microsoft.com/). Lawyers (or anyone who is not a programmer) can do so by dragging and dropping different process steps onscreen to design a decision tree. They can also add simple logical operators between steps (e.g., 'if … then…' operators). The tool then produces code in the background, bypassing the need for a software programmer to codify a decision tree within software code. The use of these tools takes lawyers away from profitable legal work whilst designing an automation routine with the hope often that the resulting tool can improve future profits, however it is often easier for a law firm to justify this temporary cost than to hire software programmers or to try to manage any form of software development project.

Research has shown that to develop and adopt AI properly, law firms must attract people from non-legal backgrounds such as computer science and must also develop multi-disciplinary teams (Parnham, Sako and Armour, 2021; Armour, Parnham and Sako, 2020). This is difficult for law firms to achieve since for a law firm, each lawyer in private practice is typically a profit centre (i.e. a part of the business that adds to the bottom line) and each non-lawyer is typically a cost centre (i.e. a part of the business that creates a cost rather than a profit). This reason, which this paper considers to be primary, is touched upon in the above cited works (Parnham, Sako and Armour, 2021; Armour, Parnham and Sako, 2020) where a number of other factors explaining the difficulty are also discussed. The opposite is true outside of law firms where in-house lawyers are seen as expensive back-office cost centres and the organisation's core staff (according to its sector) are more likely to be profit centres. It may be easier to persuade a software firm that it needs more lawyers than it is to persuade a law firm that it really *needs* any software programmers.

Given that economic incentives make an immediate switch to multi-disciplinary teams difficult for law firms, no-code solutions and their simple decision tree approaches have the potential to act as stepping-stones to allow new ways of generating value and profit to be attempted by firms to justify the changes needed to adopt this simple form of AI more broadly.

Weaknesses of the current approach

There are three main weaknesses of expert systems and decision tree technology. The first weakness is that systems using these technologies are relatively 'fixed' meaning that they are limited by the knowledge that has been programmed into them in advance by human experts (Armour and Sako, 2020 describe this as 'top-down'). The second weakness is that expert systems and decision trees are only useful in scenarios that human experts are able to map. Many complex or one-off aspects of legal practice are very difficult to map and instead require practical experience. This is especially true for bespoke or novel legal problems as well as for the interactions with clients that are needed to understand and define a legal problem before a solution is attempted. The third weakness is that typically, most legal decision trees are simple flowchart-style decision trees which are used in systems that do little computationally to replace the higher-order cognitive processes that legal practitioners undertake when thinking deeply about a matter, but do a great deal to streamline routine legal work and to replace the procedural knowledge that lawyers otherwise rely on. If AI is a combined function of both artificial 'knowledge' and artificial 'thinking' then expert systems and decision trees are impressive in their ability to deal with structured knowledge but cannot be described as demonstrating any real artificial thinking beyond a very low-order type of thinking. Modern AI researchers do not find the status-quo impressive and the potential for improvement is recognised by legal scholars (Higgins et al. 2020).

As noted above, the expert systems of the 1980s and 1990s used ‘inference engines’ to help human operators to choose efficient paths through their decision trees but much of this technology proved to be overly complex and impractical. This aspect of expert systems technology has not penetrated legal practice. The only aspect of expert systems technology to have been adopted in the legal sector in any real sense is the decision tree aspect. Most legal technology thus comprises of mere knowledge navigation tools. Human reasoning is, for the most part, still needed to choose paths at each junction of a decision tree.

Potential improvements

Using algorithmically generated decision trees

The decision trees described above are of the hand-crafted variety which humans can interact with directly. Many computer scientists will recognise the term ‘decision tree’ from a number of functions that can exist deep within computer systems. They are often used within systems for categorisation tasks where a system algorithmically extrapolates a classification tree from a body of pre-existing data so that future data can be arranged. The term ‘classification tree’ is usually used in such cases.

To illustrate how this works, imagine a 19th century gold miner with the task of sorting through placer deposits to try to find gold or other precious metals. The human gold panner can classify the deposits by working them through an appropriate selection of panning sieves and screens. The hole sizes in the selection of panning sieves and screens used for the task would need to match the range of granularity that exists in the placer deposit for a given location and for the types of precious metal sought. The gold miner’s first task is to determine a suitable selection of sieves and screens for the task. While an expert would immediately choose the right sizes of sieves and screens by looking at the placer deposit and by drawing upon prior experience, a completely inexperienced and unintelligent gold miner could, by simply blindly using a long process of trial-and-error, select a perfect set of sieves and screens. A programmed machine could also perform the same task simply via automated means of trial and error and a process of elimination. In this illustration, the machine’s systematic method of finding a suitable series of sieves and screens to sort placer deposits is akin to a computer algorithm’s systematic method of finding a suitable classification tree to sort data. Through automated trial, error and elimination, with no need for expertise, a system can be programmed to build a classification tree from a sample of data. This is akin to Richard Feynman’s dumb filing clerk metaphor for computers (Feynman, 1996). I.e., computers simply carry out entirely mechanical and clerical operations quickly but appear intelligent.

It has been shown that computer systems can be programmed to extract a decision tree algorithmically by systematically processing and organising the text of a long and complex statute (Mingay et al. 2022), but this is a relatively recent and small, unusual, example of such a technique. This technique is possible so long as the structure of the text of the statute can be interrogated using computer-based rules. This is useful so long as crafting the required rules is not more onerous than manually developing a decision tree from the statute.

Borrowing from the domain of decision science

To improve upon the status quo, and make better use of decision trees, another approach that could provide a major improvement would be for the legal profession to borrow from the domain of decision science (which is sometimes called business analytics or quantitative decision making) when using decision trees (see: Winston and Albright, 2020; Wisniewski, 2009). Arguably, the term 'science' lends undue authority to this type of collection of basic mathematical principles (McDonald, 2017). The decision trees of decision science are not simply navigation tools like those described above but are enriched with information to allow either decisions to be taken by a human operator with better information available at each step, or to allow a system to calculate an optimal route through a decision tree in the same way that a satellite navigation system not only displays a digital map but can calculate an optimal route through it. These are decision trees that not only contain structured information about a set of decisions but are also enriched with information about possible outcomes at each decision junction.

How does this work? To achieve this enrichment some mathematics is involved but it is relatively simple. First, for each factor of concern, the expected outcome is given a numerical value at each outcome point of the decision tree. Then, the probability of achieving the outcome is calculated at each decision node and each potential path is given a calculated value (Magee, 1964). This allows an algorithm, without human intervention, to decide upon an optimal path through the decision tree.

For example, in a decision tree showing potential paths of a litigation process, if the only factors of concern are the legal costs involved and the potential quantum of a litigation settlement or damages award at the end, then the cost of each step can be added to each branch of the decision tree and the most likely average quantum gain or loss arising from taking each step from a statistical perspective can also be calculated and added to each branch. Here, statistical and actuarial skills are more important than legal experience.

These enriched decision trees are well understood by business analysts in large corporations and in the insurance sector where actuarial science is well understood (Magee, 1964), but they are rare in legal practice. The phrase 'enriched' here is used as a shorthand phrase to distinguish the decision trees of decision science or business analytics and the simpler mere navigation tools that are used in legal technology at present.

Remaining limitations

Even if the above two proposed improvements are to be incorporated into legal technology systems, the systems would still broadly suffer the limitations described above to some extent. Of these limitations, the dominant one is that complex tasks cannot be prescribed in a 'top down' fashion (Armour and Sako, 2020). ANNs offer significantly more promise.

Artificial Neural networks

Artificial Neural networks (ANNs) operate unlike traditional computational or data science techniques. Rather than processing-controlled inputs using pre-defined logical algorithms– which represents the predominant method of symbolic AI and data science, ANNs are designed to use algorithms to mimic the basic behaviour of the neurons and synapses of human and animal brains (Lillicrap et al. 2020). Traditional computing can be said to be 'symbolic' since it takes symbolic representations of concepts and manipulates these using formulas to produce other symbolic representations. ANNs, sometimes called neural nets, can deal with richer and more complex problems than traditional logical computing methods can. Andrew Ng (2016) says that each ANN can undertake tasks needing about one moment of thought. The differences materialise at a high level of abstraction. ANNs are algorithmically driven logical circuits but do not solve problems using logic. They are more sophisticated than the 'inference engines' of expert systems.

It is appropriate to say ANNs mimic 'basic' biological brain functions since our understanding of the brain is limited and biological neural networks are currently immeasurably superior in terms of general flexibility than any currently available artificial counterpart. Experts disagree as to the timescale when ANNs will reach or surpass broad human capabilities (Kurzwell, 2005; Stoop, 2021). Some say it may occur in decades, others say it may occur in thousands of years–if at all.

Synthetic neural networks–those that exist in digital computer systems–have nonetheless become highly sophisticated and can achieve astounding capabilities that were previously thought to be beyond the realm of computation. Successful example applications are translating languages and recognising images. ANNs show the type of intelligence that our ancestors have previously only observed in animals, and which has never previously been replicated mechanically. We have long utilised the learning abilities of animals by training e.g., dogs or even birds to assist us and ANNs allow us to now exploit the storage and processing ability of machines by training them to 'learn' how to process information to assist us. This can be seen by comparing historical animal training methods to machine learning-based technique (e.g., Ferreira, 2020).

What Artificial Neural Networks are

At a fundamental level, we know that human thinking is derived from a vast web of paths between an immense network of neurons connected by synapses in the brain and across the nervous system. (Note: in computer science literature, the American spelling of neuron is preferred (Mehta, 2020)). Connections are formed and strengthened in a quasi-evolutionary process through human learning over time. Ideas in the brain are formed through micro-clusters of links between neurons at synapses at neural junctions and the nature of neuron connections in the brain is a field of ongoing discovery (Trafton, 2015). We process our experience of the world through these neural networks in our brains which 'generalise' solutions to problems (Stoop, 2021) and this discovery that micro-clusters of links in the biological brain can generalise solutions led to early ANN research (NY Times, 1958; Rosenblatt, 1957). The simplest fractional fragment of an idea might cause an incalculable number of neural pathways to 'light up' in the brain and a basic moment of thought might engage an enormous and even further incalculable number more. Human brains, which require more energy than other animal brains, have a significantly larger number of neurons which provides humans with our advantage in intelligence. The human brain has and uses a very high number of neurons and uses a large amount of energy as compared to the brains of members of other species (Bardone, 2019). Synthetic (artificial) neural networks simulate the way the biological human brain learns via the development and strengthening of artificial neural pathways in computer systems. With ANNs, software-generated logical computer circuits are designed to behave like biological neurons and vast numbers can be generated in software and connected together with a network of logical links between them akin to biological neural pathways. Researchers are training ANNs of increasing size and their capabilities are therefore continually growing in scale.

After a long history of failure in this subfield of AI between the 1950s and 2010s which had slowed after some promising developments occurred in the late 1940s and 1950s inspired by Alan Mathison Turing (Buchanan, 2005; McCarthy et al. 1955), a revival with enormous advances occurred in 2012, starting in the area of computer vision. An astounding improvement on prior neural-network attempts at labelling images triggered this revival. This was achieved by the computer scientist Geoffrey Hinton and two research students, Alex

Krizhevsky and Ilya Sutskever using tens of millions of artificial software-generated neurons (Krizhevsky, Sutskever and Hinton, 2012). This ground-breaking demonstration of their AI system revived the field. The period before this ground-breaking advance is often referred to as the winter of AI. Hinton, who had also made significant contributions to the field in the 1980s, received the Turing Award for his work. Modern systems now use far more artificial neurons but still operate at a scale of magnitude less than the biological human brain although the methods of counting differ for synthetic and biological neural networks (Barrett et al. 2019; Bae, 2021; Kriegeskorte, 2015).

ANNs are trained or improved when sophisticated algorithms traverse and strengthen different paths through a structured neural network arranged in lines of code. This occurs through a process of training in a quasi-evolutionary fashion by exposing the network to inputs from a large number of training samples which have been tagged by humans. Training is maintained until a network that functions for a particular purpose is developed (Aggarwal, 2018). The functionality of the ANN is driven by the strength of millions of different pathways across the vast repeatedly criss-crossing network between input and output options. The most efficient paths are strengthened through the network by this training process (Olah, 2017). Many untrained ANNs might be set up with the same original structure of connections prior to training and the same random weights for connections, but the final weights of the connections will differ as a result of the training exercise. It is these weighted connections that have utility. This can be described as a 'bottom-up' approach (Armour and Sako, 2020).

One ANN might be trained to recognise images whilst another ANN that started with exactly the same structure, and originally looked identical to the original ANN that was later trained to recognise images, might instead be trained to recognise sounds. Two differently structured ANNs with a different number of neurons connected in different ways can also be trained for exactly the same task by being exposed to the same training samples. They would however develop differently and perform at different levels of accuracy. This is akin to e.g., a human child and a human adult in the same household with different numbers and structures of biological neurons in their brains learning the same new foreign language together but at different levels of comprehension as a result of the same exposure to foreign words (Brysbaert et al. 2016). Choosing the size and structure of the network itself can involve trial and error. A smaller ANN might outperform a larger ANN in some cases in the same way that children appear to be better at learning languages than adults (Brysbaert et al. 2016).

Unlike a trained human brain which cannot be cloned (i.e., the training needs to be repeated for each human), a trained ANN can be copied and distributed at scale like other software systems and code samples can be shared between programmers using programming community websites. ANNs can also process information at a larger scale than the human brain (e.g., translation tools can deal with millions of pages or hours of audio and image recognition tools can deal with millions of images) but presently, unlike the human brain, ANNs are limited to narrow scopes of application.

The difference between expert systems and ANNs

The key difference between old-fashioned expert systems using decision trees and modern ANNs for present purposes can be explained with a metaphor from the built environment: the placement of footpaths in urban areas. Until the 1990s, when a hospital complex, university campus or other group of buildings was developed, the footpaths across grounds were typically rigidly designed in advance and laid out by engineers before anyone stepped foot on the new development. In the 1990s however, when developing the Illinois Institute of Technology, the architect Rem Koolhaas elected not to design the footpaths in advance, but instead to allow natural footpaths from the movement of people to form across the grounds (Bramley, 2018).

The architect could then revisit the area after several years and make permanent paths wherever natural paths had been formed by the natural trampling of grass along routes that proved convenient in practice in the absence of pre-prescribed designed routes. This was not an entirely new idea in the 1990s, but Koolhaas' implementation seems to be the source from which the concept was popularised. The approach of Koolhaas was a far less rigid but a far more useful method of choosing the locations of footpaths but it required a lot to be left to chance.

A number of university campuses have used this method in recent decades and these human-formed paths are called 'desire paths' in the literature. This is akin to the fundamental difference between expert systems' decision trees and ANNs. The decision trees of expert systems use paths for decision-making that are prescribed in advance by experts crafting their knowledge in a structured fashion into the memory of computer systems. Nothing needs to be left to chance. ANNs, by contrast, use potentially billions of paths for solving problems and decision-making that are formed entirely by training via flows of data and feedback loops within a system, through the traversing of a network and the forming the paths. The training exercise to develop these paths, however, can be costly and can raise practical challenges (Aggarwal, 2018; Russell and Norvig, 2016). Many instances of trial and error from data running through the system create natural paths which an engineer can later freeze in time once the neural pathways are established to make them permanent, or indeed a system can be designed to allow endless ongoing improvements over time where paths may change over time.

This capability of ANNs to find ways of solving specific problems allows them to overcome the limitations of expert systems. Two examples below show how ANNs can be trained to undertake complex cognitive processes.

Artificial Neural Networks for image recognition

Understanding how ANNs achieve image recognition demonstrates their potential for dealing with many other types of rich and complex information–such as those used for legal reasoning.

To start with the problem and the reason why the task needs something like ANNs: traditional computing works in a somewhat mechanical fashion whereby numerical inputs are processed by a computer which acts as a calculating machine. They are fed through mathematical algorithms so that an output can be produced. The information in an image is just too rich, even once extracted in a numerical format, to allow it to be processed computationally to reach any sensible output. As shown below, one cannot imagine manually designing a mathematical formula to convert the information in an image file into a textual label.

Why can't mathematicians write a formula and algorithm for image recognition? A computer file containing an unencoded and uncompressed image is stored in a computer system as a very large number (i.e., a string of characters in computer memory), made up of small parts with values in specific patterns to represent colours (Charlap, 1995). A part of the string will contain information about the image format and another (for some image formats) can be split up into smaller strings and stacked as an array or matrix of pixels–but this level of abstraction is not required for present purposes. This describes a bitmap image where each pixel has a numerical value representing a mix of red, green and blue light (note: when the long numerical string is split and stacked as an array/matrix, the result is a pixel grid). A JPEG image is encoded so that shapes of colour and gradient tones are saved to reduce file size so that each pixel does not need to be remembered. A GIF image is encoded so that geometric shapes are remembered, again to reduce file size. Returning to the bitmap format, a single red pixel might have the number 100,0,0 (100 parts red, no parts green, no parts blue) (this is a simplification for present purposes). A one-megapixel image (which is small) can thus be stored in computer memory as an extremely long numerical string (which can set out a matrix or array) with different digits representing each pixel of the image. Other formats are more compressed, but less simple to describe. Words in computer memory are also stored as numbers since each character of an alphabet has a numerical identifier within a computer system. What is needed for an image recognition task is for a computer system to take one enormous numerical string (an image) and match it to a selection of smaller numbers (words) with varying degrees of confidence. It is unfathomable to try to think of a calculation or algorithm that a human might be able to craft in advance that might take the very large numerical input string of an image file in computer memory as its input and, through computation, find matching smaller numbers (words) and degrees of confidence.

With an ANN however, the very long list of input numbers can be matched to input neurons, which connect across a vast network of connected artificial neurons to a final output set of words at the end with a degree of confidence for each potential match. Across the network, every digit in the long input numerical string maps to every possible label at the end through layers that contain junctions connecting everything in one layer to everything in another. This starting point is an 'untrained' ANN. With training using examples, the links between inputs and labels can be lit-up across the vast criss-crossing network of connections and layers so that specific patterns of features naturally become more closely linked to specific labels. Millions of training examples might be needed for usefully weighted paths to form. The fact that networks have a depth of criss-crossing layers gives rise to the term 'deep learning'. The ANN system is designed to increase the weight of the interconnections that are activated with each item of training. Eventually the weights build up so that without human intervention, each input image in the training set (each long number) will be mapped through the network to the correct label (or combination of labels). The layers of interconnected junctions in the middle of the network mean that as part of the process, patterns in the image (in the input number) cause certain clumps to activate. Future images which the system may be exposed to which were not part of the training set that have similar patterns are likely to cause the same clumps to activate and thus be mapped to the correct label (or combination of labels) at the end. This fine-tuning of weightings of network connection would be impossible for a human to define by hand, but it evolves naturally from the training exercise in the same way that biological brains learn, at least at a basic level. ('Fine-tuning' is used here in a descriptive sense. The phrase can also be used in a technical sense to describe the latter phase of training a general-purpose model.)

The training exercise will teach the ANN how much strength to ascribe to each link between neurons so that however a combination of patterns in an image is detected at the start of the network, links through each layer of the network will eventually cause a best match of a possible image label at the end of the network to be found along with other close matches with varying degrees of confidence. An ANN might find many possible labels for an image with each having a different resulting strength and the image detection algorithm will typically be set to select the strongest match, or the top five matches. Once trained, the system will be able to label new images that it has not seen before.

In any trained ANN, the connections are so vast, and the tunings so finely weighted that it is impossible to observe the network and comprehend quite how the fine-tuning 'works' completely. All we can know is that it works, not precisely how. This is the starting point for a vast body of research into 'explainable AI'–the details of which are beyond the scope of this paper.

Within the vast network of weights, some undecipherable mechanism for solving a problem will have been found through the evolutionary process described above. ANNs are structured in matrix-like layers so that multi-stage methods of solving are found.

Using Artificial Neural Networks for difficult tasks

The potential application of ANNs to legal problems is a result of their general capacity to be adapted. ANNs have a remarkable ability to be trained to learn to undertake a wide variety of tasks that would be complex to describe in precise line-by-line instructions: they can be trained to deal with many types of problems in many industries that would typically otherwise require human effort.

An example of the application of an ANN to a complex task is as follows: In the agricultural industry, male chicks are of no use for egg production so over the last century, trained humans have sorted male and female chicks using a variety of mostly visual methods (Horsey, 2002). Cognitive scientists have struggled to understand how humans can develop expert chick sexing abilities at a cognitive level, but the skill can be taught, and many schools of chick sexing exist (Williams, 2017). Following human selection, the vast majority of male chicks are culled early with very few kept for meat production. Female chicks are allowed to survive to be used for industrial egg production. Selective breeding mitigated the mass-culling of male chicks to a small extent, but it has long been the practice of the industry to sort male chicks of a few days old and cull them using industrial machinery–typically transporting them via conveyor belts to chick grinders called macerators. Globally, billions of male chicks are killed each year. In-ovo sexing, the practice of determining the sex of eggs before they hatch to avoid the culling of male chicks, has been a developing field over the last decade with various techniques involving hormones and biomarker tests arising (Zhu et al. 2021; Galli, 2018), yet these methods have proven to be more expensive than male chick culling so have gained little traction except in jurisdictions where legislation has intervened. Returning to the opening proposition of this paragraph: In recent years, it has been discovered (in part by Adam Rivers of the USDA Agricultural Research Service) that ANNs can be trained for in-ovo sexing and remarkable progress has been made to reduce the cost of the exercise, increase the accuracy level and allowing the process to be completed earlier in an egg's development. Computers have, through trial and error, been taught a seemingly magical skill with the potential of saving many male chicks from being born only to be ground in macerators. This is a skill that can be learned through experience but as-yet eludes any description so cannot be taught by direct instruction.

Scale and complexity

A trained ANN may have many hundreds of billions of finely tuned weights ascribed to possible links between synthetic neurons in order to function, but this remarkable and unimaginably complex fine-tuning is merely a result of the training exercise–not the result of intervention or purposeful fine-tuning by any intelligent agent. Indeed, the fine-tuning is often so complex that it could not be achieved manually. In the same way that a mountain will develop an efficient network of streams and waterfalls of ideal widths and depths for taking water to the river or sea at ground level, which can deal with the wide variety of weather conditions the mountain might face which would be very complex for a human engineer to design, so too an ANN will be fine-tuned through training to deal with the types of inputs that it is designed to deal with.

For machine translation tasks, previous AI techniques involved hundreds of linguists codifying language data into taxonomies for computers to read, with mappings between taxonomies and language rules carefully hand-crafted by linguistic experts. ANNs not only do not need this onerous human codifying effort but they significantly outperform the previous techniques that did. They can simply be trained from bodies of pre-translated text or speech and can find better patterns in language than a human expert could codify (Thompson, 2019).

ANNs can only be trained effectively with enormous libraries of training examples, and they improve with ongoing training.

The ‘black box’ problem

The ‘black box’ problem refers to the fact that a trained neural network with potentially billions of finely tuned weightings for connections between artificial neurons is so vast that it is practically impossible to know precisely what happens between each layer of connection (Aggarwal, 2018). Some form of reduction of abstraction between themes and concepts occurs at each layer of artificial neurons as processing occurs deeper into the network, but the precise method of working is unknowable. We simply know that a trained system works by observing its results. The inner workings are theoretically understandable, but practically impossible to decipher since if all the humans on the planet were able to co-operate on the task of trying to manually decode the working of a trained system, then it could in most cases take an incalculable number of human lifetimes (longer than the age of the universe) to untangle a trained network. It comes as no surprise that ANNs like human brains are ‘black boxes’: we cannot explain precisely how their (or our own) neural connections discriminate.

In the same way that we cannot look inside a human brain to see how it functions, we are now in an era when we cannot look inside a computer ‘brain’ to see how it functions. This is problematic if bias exists but is not detected within an ANN as a result of training (Dubber et al 2020).

Returning to the ‘desire paths’ metaphor, one might imagine it possible for, e.g., a historic bias of a higher number of male science students to cause future issues of bias to persist in the final paths. This could be caused by, e.g., male students walking between a science lab to the male dormitories leaving thick paths in the grass to be frozen in time by the returning architect who builds a path along the route to the disadvantage of future female science students who were not present when the natural paths were formed so did not create desire paths where fixed paths were later built. This exemplifies how bias can become captured, frozen and perpetuated by an AI system.

Of course, applying any systematic scientific methods to business can raise ethical questions regardless of whether AI is involved (McDonald, 2017), but ANNs present specific challenges because their precise methods of working are unknown. There is a great deal of research work seeking to reduce the difficulty of this ‘black box’ problem, a sub-field often called explainable-AI (Vedaldi, 2019), but the fundamental problem of issues such as bias still exists for computers–as it does for humans (Higgins, 2020; Zuckerman, 2020).

Application to legal practice

ANNs are useful for dealing with rich and complex data sets that contain patterns that cannot be dealt with by expert systems, mathematics, logic or traditional algorithms. They are therefore useful for complex legal problems such as questions over precisely how overlapping existing laws should be applied to novel situations that arise in real life. Typically, ANNs have required large data sets (LeCun et al. 2016) but emerging artificial learning techniques are reducing this need (Aggarwal, 2018; Hinton and Salakhutdinov 2006) making ANNs more practical for use in legal practice.

ANNs are already used by systems that support high-stakes litigation to unearth relevant documents, in complex transactions, to review documents and to highlight legal risks (Zuckerman, 2020). These tasks were traditionally performed by human practitioners charging hourly fees meaning that some aspects of legal work have changed, however, more broadly, the impact of ANNs on legal practice has to date been limited.

Potential uses of Artificial Neural Networks in legal practice

Examples of what ANNs might feasibly be trained to do (given enough training examples) include to:

1. Select the legal principles that apply to a given set of facts.
2. Predict case outcomes based on past cases with similar facts.
3. Detect which field of law is engaged by a problem.
4. Propose solutions for jurisdictional problems or conflict-of-law problems.
5. Replace a traditional expert-system that might otherwise take many human lifetimes to handcraft.
6. Examine a legal document and determine what type of document it is.
7. Detect gaps in legal documents.
8. Identify 'loopholes' across a large body of legislation or a large collection of contracts.
9. Evaluate quantum levels.
10. Identify whether witness evidence is likely to be believed or disbelieved.

There are many other possibilities, but a limited list of ten examples has been selected to maintain brevity. It should be noted that each application here is a narrow micro-task.

Application of Artificial Neural Networks to multi-stage legal processes

Many tasks of legal practice involve different mixes of cognitive processes. By combining together trained ANNs that perform narrow functions, either in collaboration or in competition, more complex systems can be developed. For example, when a human lawyer responds to a client's question about the law, a sequence of cognitive processes is required. Replicating the sequence of processes would require a combination of ANNs that are trained to complete different linked narrow tasks such as (again selecting 10 examples):

1. Determining whether the question being asked is a legal question at all;
2. identifying the jurisdiction and scope of the question;
3. identifying the body of law that is engaged;
4. understanding the framework of the broad body of law that has been engaged;
5. identifying the issues from the facts (based on the broad body of law);
6. breaking the question down into sub-questions;
7. identifying the correct narrow elements of law for each sub-question;
8. applying the narrow law to the facts;
9. organising a proposed answer; and,
10. summarising the proposed answer for a client.

A practising lawyer dealing with a legal issue will typically undertake the above micro-steps or a similar arrangement of similar steps.

As another example, preparing a witness statement on behalf of a client for trial could involve the following sub-processes (again selecting 10 examples):

1. Identifying the relevant law (which may have sub-processes, see above);
2. taking a broad statement initially from a witness (transcribing what the witness says orally);
3. identifying how facts engage different areas of law and selecting the relevant legal tests that facts need to speak to (or not);

4. identifying gaps where legal tests require further facts or logical gaps in the witness' story;
5. developing (ethical) prompting questions where the witness might (or might not) fill gaps;
6. removing irrelevant parts of evidence;
7. removing evidence that should be provided by an expert or other witness;
8. removing hearsay evidence or other evidence that is procedurally objectionable;
9. choosing an appropriate order for the evidence (by theme or chronology); and,
10. summarising the evidence appropriately in the witness's own words.

This shows that a system's functionality can be maximised by using ANNs in combination.

Competition between ANNs also enhances their potential, specifically where creativity is needed. AI art for example involves one ANN that is trained to generate random results and another ANN that is trained to test how realistic each result is based on trained prior examples (Aggarwal, 2018). There is no shortage of creative cognitive processes needed in legal practice and thus no shortage of potential for ANNs.

Conclusions and recommendations

Six broad conclusions and recommendations can be drawn from the above analysis:

1. Apparent paradoxes both arising in the legal AI literature and between researchers and legal practitioners can, to some extent, be explained by the fact that different writers use the term 'AI' to mean different things. A researcher may envisage ANNs when AI is discussed and say that legal practice can be fundamentally transformed by AI, whilst a legal practitioner may envisage decision trees and conclude that AI is already part of his or her legal practice and consider that there is little real scope for transformation. I.e. to some extent, the fact that very different types of AI exist can explain the conflict of visions that arises for legal AI.
2. It is only partially correct to say that 'expert systems' have been adopted by the legal sector. The inference engine part of expert systems technology has not been widely adopted, but the knowledge base part of the technology has been widely adopted using decision trees.
3. The technology that has been adopted in legal practice has, to date and, involved the relatively simple and understandable automation of knowledge-based processes. Lawyers have been more comfortable allowing systems to augment their knowledge than to allow systems to augment their critical thinking skills.
4. There is great potential for the legal sector to use algorithmically generated decision trees. These have been used in other fields and the application of these to a legal context is a promising recent innovation.
5. Another area for improvement is for law firms to adopt the enriched decision trees of decision science. These are used in business analytics and have much utility, but these have not yet penetrated the legal profession at any scale.
6. Beyond improving decision tree technology, ANNs provide a fundamentally different approach to legal problem-solving and have significant potential for the legal sector. ANNs have the potential to tackle more complex, multi-stage, legal tasks especially if multiple, potentially more complex, ANNs are used in combination.

Acknowledgements

The author is grateful to Emeritus Professor Adrian Zuckerman of Oxford University for feedback and to Dr Luke Herbert-Anderson for sharing papers.

References

AGGARWAL, Charu C.. *Neural networks and deep learning : a textbook [electronic resource]*. Cham, Switzerland: 2018.

ALBRIGHT, S. Christian y WINSTON, Wayne L.. *Business analytics : data analysis and decision making*. 5th edition. ed. Delhi: 2015.

ANDERSON, Raymond. *The credit scoring toolkit : theory and practice for retail credit risk management and decision automation [electronic resource]*. Oxford: Oxford University Press, 2007.

ARCHETTI, Francesco, C, y ELIERI, Antonio. *Bayesian optimization and data science [electronic resource]*. Cham: 2019. SpringerBriefs in optimization.

ARMOUR, John, et al. "Unlocking the Potential of AI for English Law." International Journal of the Legal Profession, vol. 28, no. 1, Taylor and Francis, 2020.

ARMOUR, John and SAKO, Mari, *AI-enabled business models in legal services: from traditional law firms to next-generation law companies* (Journal of Professions and Organization 2020, 7, 27-46)

BAE, Hyojin, KIM, Sang Jeong y KIM, Chang-Eop. "Lessons From Deep Neural Networks for Studying the Coding Principles of Biological Neural Networks". *Frontiers in systems neuroscience*. 2021, vol 14, p. 615129–615129.

BAIERLEIN, Ralph. "Probability Theory: The Logic of Science Probability Theory: The Logic of Science , E. T. Jaynes Cambridge U. Press, New York, 2003. (727 pp.). ISBN 0-521-59271-2". *Physics today*. 2004, vol 57, num. 10, p. 76–77.

BARRETT, David GT, MORCOS, Ari S y MACKE, Jakob H. "Analyzing biological and artificial neural networks: challenges with opportunities for synergy?". *Current opinion in neurobiology*. 2019, vol 55, p. 55–64.

BAYES, Mr. y MR. PRICE, . "An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S.". *Philosophical Transactions (1683-1775)*. 1763, vol 53, p. 370–418.

Bench-Capon, T, Araszkiewicz, M, Ashley, K, Atkinson, K, Bex, F, Borges, F, Bourcier, D, Bourgine, D, Conrad, J. G, Francesconi, E, Gordon, T. F, Governatori, G, Leidner, J. L, Lewis, D. D, Loui, R. P, Mccarty, L. T, Prakken, H, Schilder, F, Schweighofer, E, Thompson, P, Tyrrell, A, Verheij, B, Walton, D. N Y Wyner, A. Z. "A History Of Ai and Law in 50 papers: 25 years of the international conference on AI and Law". *Artificial intelligence and law*. 2012, vol 20, num. 3, p. 215–319.

BERRY, Michael W, MOHAMED, Azlinah Hj y WAH, Yap Bee. *Supervised and unsupervised learning for data science [electronic resource]*. Cham: Springer, 2020. Unsupervised and semi-supervised learning.

BORDONE, Melina Paula, SALMAN, Mootaz M., TITUS, Haley E., AMINI, Elham, ANDERSEN, Jens V., CHAKRABORTI, Barnali, DIUBA, Artem V., DUBOUSKAYA, Tatsiana G., EHRKE, Eric, ESPINDOLA DE FREITAS, Andiara, BRAGA DE FREITAS, Guilherme, GONÇALVES, Rafaella A., GUPTA, Deepali, GUPTA, Richa, HA, Sharon R., HEMMING, Isabel A., JAGGAR, Minal, JAKOBSEN, Emil, KUMARI, Punita, LAKKAPPA, Navya, MARSH, Ashley P. L., MITLÖHNER, Jessica, OGAWA, Yuki, PAIDI, Ramesh Kumar, RIBEIRO, Felipe C., SALAMIAN, Ahmad, SALEEM, Suraiya, SHARMA, Sorabh, SILVA, Joana M., SINGH, Shripriya, SULAKHIYA, Kunjbihari, TEFERA, Tesfaye Wolde, VAFADARI, Behnam, YADAV, Anuradha, YAMAZAKI, Reiji y SEIDENBECHER, Constanze I.. "The energetic brain–A review from students to students". *Journal of Neurochemistry*. 2019, vol 151, num. 2, p. 139-165.

BRAMLEY, Ellie. "Desire paths: the illicit trails that defy the urban planners" *The Guardian.* <https://www.theguardian.com/cities/2018/oct/05/desire-paths-the-illicit-trails-that-defy-the-urban-planners> (5 Oct 2018) accessed Q4 2022

BRYSBAERT, Marc, STEVENS, Michaël, M, , ERA, Pawel y KEULEERS, Emmanuel. "How many words do we know? Practical estimates of vocabulary size dependent on word definition, the degree of language input and the participant's age". *Frontiers in psychology*. 2016, vol 7, p. 1116–1116.

BUCHANAN, Bruce G. "A (very) brief history of artificial intelligence". *The AI magazine*. 2005, vol 26, num. 4, p. 53–60.

CHARLAP, David, "The BMP File Format, Part 1" <https://drdobbs.com/architecture-and-design/the-bmp-file-format-part-1/184409517> (1 Mar 1995) accessed Jan 2023.

DALE, Andrew I. *A history of inverse probability : from Thomas Bayes to Karl Pearson*. 2nd ed. ed. New York ; London: Springer, 1999. Sources and studies in the history of mathematics and physical sciences.

DEMIDOV, Vadim V. *DNA beyond genes : from data storage and computing to nanobots, nanomedicine, and nanoelectronics [electronic resource]*. Cham: Springer, 2020.

DUBBER, Markus Dirk, PASQUALE, Frank y DAS, Sunit. *The Oxford handbook of ethics of AI [electronic resource]*. New York: 2020. Oxford handbooks online.

FARRAR, John H y DUGDALE, Anthony M. *Introduction to legal method*. 2nd ed. / John H. Farrar and Anthony M. Dugdale. ed. London: Sweet & Maxwell, 1984.

FERREIRA et al 'Deep learning-based methods for individual recognition in small birds' (Methods Ecol Evol. 2020 11: 1072–1085) <https://doi.org/10.1111/2041-210X.13436> accessed Jan 2022.

FEYNMAN, R. "Feynman Lectures on Computation". Addison-Wesley Publishing Company, Inc 1996

GALLI, Roberta, PREUSSE, Grit, SCHNABEL, Christian, BARTELS, Thomas, CRAMER, Kerstin, KRAUTWALD-JUNGHANNS, Maria-Elisabeth, KOCH, Edmund y STEINER, Gerald. "Sexing of chicken eggs by fluorescence and Raman spectroscopy through the shell membrane". *PloS one*. 2018, vol 13, num. 2, p. e0192554–e0192554.

GOOGLE, Ngram graph of prominence of 'expert system' over time: <https://tinyurl.com/2wbbmd5x> accessed Q2 2022.

GOOGLE, Ngram graph of prominence of 'neural networks', 'machine learning' and 'expert systems' over time: <https://tinyurl.com/2wbbmd5x> accessed Q2 2022.

GRINDROD, Peter. *Mathematical Underpinnings of Analytics: Theory and Applications*. Oxford: Oxford University Press, 2014.

HAENLEIN, Michael y KAPLAN, Andreas. "A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence". *California management review*. 2019, vol 61, num. 4, p. 5–14.

HERVEY, Matt y LAVY, Matthew. *The law of artificial intelligence*. First edition. ed. London: 2021.

HIGGINS, Andrew, LEVY, Inbar and LIENART, Thibaut. *The Bright but Modest Potential of Algorithms in the Courtroom* (Ch 6 of Principles, Procedure and Justice: Essays in Honour of Adrian Zuckerman 113-132 OUP, 2020)

HINTON, G.E y SALAKHUTDINOV, R.R. "Reducing the Dimensionality of Data with Neural Networks". *Science (American Association for the Advancement of Science)*. 2006, vol 313, num. 5786, p. 504–507.

HORSEY, RICHARD, 'The art of chicken sexing' UCL Working Papers in Linguistics 14 (2002) https://www.phon.ucl.ac.uk/home/PUB/WPL/02papers/abstracts/horsey.html <accessed Jan 2022>

HOFSTADTER, D.R.. *Godel, Escher, Bach: An Eternal Golden Braid*. Basic Books, 1999.

KHOYLOU, Jalal (undated) <https://www.credit-scoring.co.uk/modeller> accessed Jan 2022.

KOVAS, Y., & SELITA, F. 'Oedipus Rex in the Genomic Era'. (Palgrave Macmillan 2021

KRIEGESKORTE, Nikolaus. "Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing". *Annual review of vision science*. 2015, vol 1, num. 1, p. 417–446.

KRIZHEVSKY, Alex, SUTSKEVER, Ilya y HINTON, Geoffrey E.. *ImageNet Classification with Deep Convolutional Neural Networks*. Red Hook, NY, USA: Curran Associates Inc.. 2012.p. 1097–1105.

KRIZHEVSKY, Alex, SUTSKEVER, Ilya y HINTON, Geoffrey. "ImageNet classification with deep convolutional neural networks". *Communications of the ACM*. 2017, vol 60, num. 6, p. 84–90.

KUCHARSKI, Adam. *The perfect bet : how science and maths are taking the luck out of gambling*. London: 2016.

KURZWEIL, Ray. *The singularity is near: When humans transcend biology*. Penguin, 2005.

LECUN, Yann, BENGIO, Yoshua y HINTON, Geoffrey. "Deep learning". *Nature*. 2015, vol 521, num. 7553, p. 436–444.

LEWIS, Edward M.. *An introduction to credit scoring*. San Rafael, Calif.: Athena Press, 1990.

LILLICRAP, Timothy P, SANTORO, Adam, MARRIS, Luke, AKERMAN, Colin J y HINTON, Geoffrey. "Backpropagation and the brain". *Nature reviews. Neuroscience*. 2020, vol 21, num. 6, p. 335–346.

LOWRY, Cameron. *When Moore's Law Killed Chess: How Strategy Games Redefined Intelligence in AI* (University of Chicago Intersect, Vol 15, No 1, 2021).

MAGEE, John. 'Decision Trees for Decision Making' (Harvard Business Review 1964) <https://hbr.org/1964/07/decision-trees-for-decision-making> checked Oct 2022.

MAY, Leo. *The Bayesian rule : simplification and geometrical visualization*. Regensburg: Roderer, 1998. Theorie und Forschung. Mathematik ; Bd. 11.

MCCARTHY, John, MINSKY, Marvin L y SHANNON, Claude E. "A proposal for the Dartmouth summer research project on artificial intelligence - August 31, 1955". *The AI magazine*. 2006, vol 27, num. 4, p. 12–14.

MCDONALD, Duff. *The golden passport : Harvard Business School, the limits of capitalism, and the moral failure of the MBA elite*. New York: 2017.

MCGRAYNE, Sharon Bertsch. *The theory that would not die [electronic resource] : how Bayes' rule cracked the enigma code, hunted down Russian submarines, and emerged triumphant from two centuries of controversy*. New Haven [Conn.: Yale University Press, 2011. Ebook central.

MEHTA, Arpan R, MEHTA, Puja R, ANDERSON, Stephen P, MACKINNON, Barbara L H y COMPSTON, Alastair. "Etymology and the neuron(e)". *Brain*. 2019, vol 143, num. 1, p. 374-379.

MINGAY, H.R.F.; HENDRICUSDOTTIR, R.; CEROSS, A.; BERGMANN,J.H.M. Using Rule-Based Decision Trees to Digitize Legislation. Prosthesis. 2022,4,113–124. https://doi.org/10.3390/prosthesis4010012

MINSKY, Marvin. "Steps toward Artificial Intelligence". *Proceedings of the IRE*. 1961, vol 49, num. 1, p. 8–30.

MINSKY, Marvin. *Computation : finite and infinite machines*. London: Prentice-Hall International, 1972. Prentice-Hall series in automatic computation.

NEWELL, Allen y SIMON, Herbert A.. "Computer Science as Empirical Inquiry: Symbols and Search". *Commun. ACM*. 1976, vol 19, num. 3, p. 113–126.

Ng, Andrew. *What Artificial Intelligence Can and Can't Do Right Now*, Harvard Business Review. 2016).

NY Times. 'New Navy Device Learns by Doing; Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser' 1958

OLAH, C. et al, others 'Feature Visualization' (2017) <https://distill.pub/2017/feature-visualization/> accessed Jan 2022.

PARNHAM, R., SAKO, M. and ARMOUR, J. (2021). AI-assisted lawtech: its impact on law firms (white paper). Oxford: University of Oxford. December 2021.

RAPHAEL, Bertram. *The thinking computer : mind inside matter*. San Francisco: W. H. Freeman, 1976. Series of books in psychology.

ROSENBLATT, Frank. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

RUSSELL, Stuart J y NORVIG, Peter. *Artificial intelligence [electronic resource] : a modern approach*. Third edition / contributing writers, Ernest Davis [and seven others].; Global edition. ed. Boston: 2016. Prentice Hall series in artificial intelligence.

SIDDIQI, Naeem. *Credit risk scorecards : developing and implementing intelligent credit scoring*. Hoboken, N.J. : Chichester: Wiley ; John Wiley [distributor], 2006.

SIME, Stuart. *A practical approach to civil procedure [electronic resource]*. Twenty-fourth edition. ed. Oxford: 2021. Legal practice course manuals.

STONE, Marcus. *Cross-examination in criminal trials*. 3rd ed. ed. Haywards Heath: Tottel, 2009.

STOOP, Ruedi. "Note on the Reliability of Biological vs. Artificial Neural Networks". *Frontiers in physiology*. 2021, vol 12, p. 637389–637389.

SUSSKIND, Richard E. *Expert systems in law : a jurisprudential inquiry*. 1986. Published at: SUSSKIND, Richard E. *Expert systems in law : a jurisprudential inquiry*. Oxford: Clarendon Press, 1989.

SUSSKIND, Richard E and SUSSKIND, Daniel. 2015. *The Future of the Professions : How Technology Will Transform the Work of Human Experts.* First ed. Oxford United Kingdom: Oxford University Press.

THOMPSON, Nicholas. Wired Magazine, <https://www.wired.com/story/ai-pioneer-explains-evolution-neural-networks/> (May 2019) accessed Jan 2022

TURING, A. M. "On Computable Numbers, with an Application to the Entscheidungsproblem". *Proceedings of the London Mathematical Society*. 1937, vol s2-42, num. 1, p. 230–265.

TURING, A. M.. "I.—Computing Machinery and Intelligence". *Mind*. 1950, vol LIX, num. 236, p. 433-460.

VARIOUS. “Susskind, Richard: Expert Systems in Law (Book Review)". *Law Quarterly Review*. 1988, vol 104, p. 171

VEDALDI, Andrea, MONTAVON, Gregoire, HANSEN, Lars Kai, SAMEK, Wojciech y MULLER, Klaus-Robert. *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer, 2019. LNCS sublibrary. SL 7, Artificial intelligence.

ZDZIARSKI, Jonathan A. *Ending spam [electronic resource] : Bayesian content filtering and the art of statistical language classification*. 1st ed. ed. San Francisco: No Starch Press, 2005. Ebook central.

ZHU, Z.H., Z.F. YE, y. TANG, . "Nondestructive identification for gender of chicken eggs based on GA-BPNN with double hidden layers". *Journal of Applied Poultry Research*. 2021, vol 30, num. 4, p. 100203.

ZUCKERMAN, A. A. S. *Zuckerman on civil procedure : principles of practice*. 2nd ed. ed. London: Sweet & Maxwell, 2006.

ZUCKERMAN, Adrian. "Artificial intelligence Implications for the legal profession, adversarial process and rule of law". *Law quarterly review*. 2020, vol 136, num. July, p. 427–453.