1713 - Artificial Intelligence (AI)		
BENCHMARKS	**ACTIVITIES BY YEAR**	
	2017	
	'1-1	'2-1
Internet searches: 1 Articles	Objectives	Heuristic design
OBJECTIVES	**RESULTS**	
Learn new methods: ? Achieve results: ? Optimality: % Completeness: % Accuracy and precision: % Execution time: 2.2 seconds Virus scanning: 99.8 %		
UNCERTAINTIES & KEY VARIABLES	**CONCLUSIONS**	
1 - Technological uncertainty		
Data sets & structures	Y	
Hardware (architectures, devices...)	Y	
Heuristics for decision making		
SPECIFIC DETAILS ON		
2 - Heuristics (computer science)		
admissibilities		
combinations		Y
types - top down, inferred, rule of thumb		
	METHODS	
Analysis Trials Prototypes Lines of code	60000000 1	
	COSTS	
Hours Materials $ Subcontractor $		

Project Details:

Scientific or Technological Objectives:

Measurement	Current Performance	Objective	Has results?
Learn new methods (?)	(not set)	(not set)	No
Achieve results (?)	(not set)	(not set)	No
Optimality (%)	(not set)	(not set)	No
Completeness (%)	(not set)	(not set)	No
Accuracy and precision (%)	(not set)	(not set)	No
Execution time (seconds)	3.2	2.2	No
Virus scanning (%)	97	99.8	No

The following information is based on the article " The challenges of patenting artificial intelligence, November 27, 2017" by Stacy Rush of Ridout Maybee, LLP as published in Canadian Lawyer. Many of the issues related to patents apply to R&D Grant & SR&ED tax credit eligibility.

Artificial intelligence’s “black box” decision-making presents challenges for AI and machine learning innovators who want to file for patents. Practitioners need to consider the unique properties of AI technology to secure meaningful and enforceable patent protection for these inventions.

The problem: “Black box” AI algorithms

Machine learning relies on training a piece of software to make decisions by providing feedback on the output it produces while processing a set of training data.

The programmers create the initial structure of the software and define the feedback heuristics used to train it, but the software produced by the training process is often a jumble of weights and interconnections between nodes in a neural network or some similarly human-illegible chunk of math.

Thus, while AI systems created via ML often exhibit highly effective decision-making, they do so without providing their creators (or anyone else) any meaningful insight as to the underlying logic of the system.

Field of Science/Technology:

Software engineering and technology (2.02.09)

Project Details:

Intended Results:	Develop new processes, Develop new materials, devices, or products, Improve existing processes, Improve existing materials, devices, or products
Work locations:	Commercial Facility, Lab
Key Employees:	
Evidence types:	Records of resources allocated to the project, time sheets; Progress reports, minutes of project meetings; Test protocols, test data, analysis of test results, conclusions; Design, system architecture and source code; Project planning documents; Design of experiments; Records of trial runs

Scientific or Technological Advancement:

Uncertainty #1: Technological uncertainty

Patents and artificial intelligence

A patent is a bargain with the public. An inventor applying for a patent is required to disclose the nature of the invention, including instructions for building it, to the public — in exchange, he or she is granted a limited monopoly, usually for a period of 20 years. The public is granted access to the workings of the invention in perpetuity thereafter.

In the context of AI and ML, inventors are faced with hard decisions about how to satisfy this disclosure requirement. What constitutes proper disclosure about the workings of a “black box” AI invention when the inventor herself does not always understand how it works?

Further, how much detail is needed to describe the intricate (and commercially sensitive) training protocols by which a mature ML system is created?

A national patent office will examine the invention through the lens of public policy, including policies against the monopolization of abstract ideas: How can the inventor overcome the issues outlined above and describe the workings of the algorithm so as to satisfy the patent examiners — and the courts — that inventors are entitled to a patent for their inventions?

Considerations for patenting an ML system

One fundamental question that requires consideration is what parts of the technology are being claimed as the invention? An inventor will need to make choices about how much they want a patent to focus on protecting the processes by which the system is created, trained and validated, as opposed to focusing on the final product deployed after these processes have run their course.

Another issue that requires consideration is the question of subject matter eligibility. To qualify as proper subject matter for a patent in these jurisdictions, an AI-based technology generally needs to be something more than an “abstract idea.” The patent office needs to be convinced that a specific “technical problem” is being overcome by the invention in order to have the patent issued. (See Alice Corp. v. CLS Bank International). This hurdle is easier to overcome if the application includes claims directed to specific implementations of the technology, such as:

Specific hardware (e.g. sensors, remote devices, autonomous vehicle controls, processor architectures);
Specific details about the training data or how that data is processed by the system;
Specific data structures implementing an AI or ML system (e.g. a neural network);
Specific heuristics being used for decision-making and/or training feedback; and
Technical improvements to the functioning of a computer.

The most significant underlying key variables are:

SPECIFIC DETAILS ON (unresolved), Hardware (architectures, devices...), Data sets & structures, Heuristics for decision making (unresolved)

Technology or Knowledge Base Level:

Activity #1-1: Specific vs General Objectives (Fiscal Year 2017)

Methods of experimentation:

Method	Experimentation Performed
Analysis / simulation:	60000000 alternatives
Trials:	1 runs / samples

With regard to AI and ML inventions, the United States Patent and Trademark Office and U.S. courts have provided some guidance to patent eligibility:

- as one example, a patent application for an ML system that simply describes possible specific data structure implementations (e.g. neural networks) may be rejected by the USPTO (See Ex Parte Kirshenbaum, Board of Patent Appeals and Interferences, Appeal 2007-3223),

- whereas an application that restricts the claimed monopoly to such a specific implementation may be allowed (See Ex parte Bramlett, Patent Trial and Appeal Board, Appeal 2015-002707).

By including technical implementation details of the AI or ML system in the patent claims and specification, the inventor should be able to overcome common pitfalls. However, there are trade-offs involved in this choice.

Specifically, if patent claims limit an invention to a specific implementation, the patent may be difficult to enforce:

first, because a patent owner may not be able to detect when a competitor is using an invention as part of their product or service and,

second, because it may be easy for a competitor to design around the specific implementation claimed in the patent by employing an alternative implementation.

Results:

These considerations of enforceability take practitioners outside of the purely technical realm and require practitioners to consider the details of the market in which the business operates. Before drafting the claims, a patent agent should be fully briefed on these market factors, including:

- Who the competitors are;
- Which features of the new technology competitors are most likely to infringe;
- How such infringement could be detected; and
- Which other market actors a company would (or could) sue in the event of infringement.

Conclusion:

An effective patent strategy to protect AI-related inventions should take into account all of the issues of patentability and enforceability set out above, tailored to the realities and needs of the business.

Practitioners need to consider the unique properties of AI technology to secure meaningful and enforceable patent protection for these inventions.

Stacy Rush is an associate in the Toronto office of Ridout & Maybee LLP – a boutique intellectual property firm. Stacy's practice focuses on assisting with the preparation and prosecution of patent applications in the areas of software and computer-related devices, gaming, and medical device technologies.

Matt Norwood is an associate in the Toronto office of Ridout & Maybee and assisted in the writing of this column. Matt's focus is directed towards litigation of intellectual property disputes and advising clients on IP licensing matters. Matt also assists with the drafting and prosecution of patent applications with an emphasis on electrical and computer related technology.

Significant variables addressed: Data sets & structures, Hardware (architectures, devices...)

Documentation:

- Uploaded to RDBASE.NET: The challenges of patenting artificial intelligence _ Canadian Lawyer Mag.pdf (475KB)
- Offline Documents: Typical documentation

Uncertainty #2: Heuristics (computer science)

HEURISTIC (COMPUTER SCIENCE) - FROM WIKIPEDIA

In computer science, artificial intelligence, and mathematical optimization, a heuristic (from Greek "I find, discover") is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution.

This is achieved by trading optimality, completeness, accuracy, or precision for speed. In a way, it can be considered a shortcut.

A heuristic function, also called simply a heuristic, is a function that ranks alternatives in search algorithms at each branching step based on available information to decide which branch to follow. For example, it may approximate the exact solution.

TRADE OFF CRITERIA:

The trade-off criteria for deciding whether to use a heuristic for solving a given problem include the following:

Optimality: When several solutions exist for a given problem, does the heuristic guarantee that the best solution will be found? Is it actually necessary to find the best solution?

Completeness: When several solutions exist for a given problem, can the heuristic find them all? Do we actually need all solutions? Many heuristics are only meant to find one solution.

Accuracy and precision: Can the heuristic provide a confidence interval for the purported solution? Is the error bar on the solution unreasonably large?

Execution time: Is this the best known heuristic for solving this type of problem? Some heuristics converge faster than others. Some heuristics are only marginally quicker than classic methods.

In some cases, it may be difficult to decide whether the solution found by the heuristic is good enough, because the theory underlying that heuristic is not very elaborate.

PITFALLS:

Some heuristics have a strong underlying theory; they are either derived in a top-down manner from the theory or inferred from experimental data. Others are just rules of thumb learned empirically without even a glimpse of theory. The latter are exposed to a number of pitfalls.

When a heuristic is reused in various contexts because it has been seen to "work" in one context, without having been mathematically proven to meet a given set of requirements, it is possible that the current data set does not necessarily represent future data sets (see: overfitting) and that purported "solutions" turn out to be akin to noise.

Statistical analysis can be conducted when employing heuristics to estimate the probability of incorrect outcomes. To use a heuristic for solving a search or a knapsack problem, it is necessary to check that the heuristic is admissible.

If a heuristic is not admissible, it may never find the goal, either by ending up in a dead end of graph.

The most significant underlying key variables are:

combinations, types - top down, inferred, rule of thumb (unresolved), admissibilities (unresolved)

Technology or Knowledge Base Level:

Benchmarking methods & sources for citings:

Benchmark Method/Source	Measurement	Explanatory notes
Internet searches	1 Articles	Wikipedia illustrations of computer science heuristics

Activity #2-1: Heuristic design (Fiscal Year 2017)

Methods of experimentation:

Examples

Simpler problem
One way of achieving the computational performance gain expected of a heuristic consists of solving a simpler problem whose solution is also a solution to the initial problem. Such a heuristic is unable to find all the solutions to the initial problem, but it may find one much faster because the simple problem is easy to solve.

Traveling salesman problem
An example of approximation is described by Jon Bentley for solving the traveling salesman problem (TSP) so as to select the order to draw using a pen plotter. TSP is known to be NP-Complete so an optimal solution for even a moderate size problem is intractable.

Instead, the greedy algorithm can be used to give a good but not optimal solution (it is an approximation to the optimal answer) in a reasonably short amount of time. The greedy algorithm heuristic says to pick whatever is currently the best next step regardless of whether that precludes good steps later. It is a heuristic in that practice says it is a good enough solution, theory says there are better solutions (and even can tell how much better in some cases).

Search
Another example of heuristic making an algorithm faster occurs in certain search problems. Initially, the heuristic tries every possibility at each step, like the full-space search algorithm. But it can stop the search at any time if the current possibility is already worse than the best solution already found. In such search problems, a heuristic can be used to try good choices first so that bad paths can be eliminated early (see alpha-beta pruning).

Newell and Simon: heuristic search hypothesis
In their Turing Award acceptance speech, Allen Newell and Herbert A. Simon discuss the heuristic search hypothesis: a physical symbol system will repeatedly generate and modify known symbol structures until the created structure matches the solution structure.

Each successive iteration depends upon the step before it, thus the heuristic search learns what avenues to pursue and which ones to disregard by measuring how close the current iteration is to the solution. Therefore, some possibilities will never be generated as they are measured to be less likely to complete the solution.

A heuristic method can accomplish its task by using search trees. However, instead of generating all possible solution branches, a heuristic selects branches more likely to produce outcomes than other branches. It is selective at each decision point, picking branches that are more likely to produce solutions.

Virus scanning
Many virus scanners use heuristic rules for detecting viruses and other forms of malware. Heuristic scanning looks for code and/or behavioral patterns indicative of a class or family of viruses, with different sets of rules for different viruses.

If a file or executing process is observed to contain matching code patterns and/or to be performing that set of activities, then the scanner infers that the file is infected.

The most advanced part of behavior-based heuristic scanning is that it can work against highly randomized polymorphic viruses, which simpler string scanning-only approaches cannot reliably detect.

Heuristic scanning has the potential to detect many future viruses without requiring the virus to be detected somewhere, submitted to the virus scanner developer, analyzed, and a detection update for the scanner provided to the scanner's users.

Results:

ADVANCEMENT FROM MACHINE LEARNING & AI COULD BE IN THE FORM OF:

1) SOFTWARE & HEURISTIC TECHNOLOGY

2) IMPROVED MATERIALS & DEVICES (NON-SOFTWARE)

Conclusion:

TECHNOLOGICAL ADVANCEMENTS WILL LIKELY BE IDENTIFIABLE IN PROJECTS THAT PROVIDE "SPECIFIC" OBJECTIVES & SCOPE.

Significant variables addressed: combinations