The Ryskamp Learning Machine and Machine Learning Explainability

By Jeff Horton with contributions from Gino Mortillero and Rix Ryskamp

Introduction

In recent years, artificial neural networks (ANNs) have made significant contributions to technology and society. Despite these advancements, ANNs have lacked the ability to reveal their decisions and conclusions.

The Ryskamp Learning Machine is a new approach to machine learning and ANN design that inherently provides examination and explanation of its operations and conclusions on a very granular level. The system was designed and implemented by a research and development company called useAIble, with Rix Ryskamp as the company founder and technology inventor.

Current State of Explainability in Machine Learning

Operation as a "Black Box"

Learning as it occurs inside ANNs is essentially an educated trial and error method of drawing conclusions and creating mappings from a set of information or inputs to desired conclusions. The process includes thousands or millions of iterations where the state of weights inside the artificial neurons are mathematically calculated and adjusted until the desired state is achieved. With each calculation, an operation to aggregate information is performed using mathematical formulas. One of the artifacts of this process is that the complexity and volume of these calculations are beyond the human cognitive ability to follow, and the aggregations obscure the original data they collected.

These aggregations are often based upon statistical methods. When social statistics are reported in the aggregate we see results like "50% of people preferred product A". How the conclusion was reached is only knowable if you have access to the source data and methodologies used in the statistical determination. ANNs do not traditionally have a way to access the "source data" since, unlike in social statistics, the data has been aggregated so many times – sometimes thousands of times per second. This continuous aggregation makes following a logical thread representing how the network makes a decision almost impossible and gives rise to our "black box" problem.

External References:

- *". . . that makes the inner workings of neural networks like a black box, opaque even to the engineers who initiate the machine learning process."* (Marina Krakovsky, Stanford Engineering News, Finally, a peek inside the 'black box' of machine learning systems)
- *". . . they're something of a 'black box' when it comes to elucidating exactly how their results are generated."* (Charles McLellan, ZDNet, Inside the black box: Understanding AI decision-making)
- *"It's not always clear what happens inside -- you let the network organise itself, but that really means it does organise itself: it doesn't necessarily tell you how it did it."* (Nils Lenke, Nuance, Artificial Intelligence has to deal with its transparency problems by Shakti Acharya TechGig)

Explainability is a Necessary Capability

In traditional software programming, a system can be reviewed with deterministic conclusions drawn about why it behaves the way that it does. This is the fundamental and necessary process of "debugging" that software development could not proceed without. Despite several early attempts to provide tools for ANN explaining, no similar debugging method exists for ANNs.

External References:

- "*Whether it's an investment decision, a medical decision, or maybe a military decision, you don't want to just rely on a 'black box' method."* (Tommi Jaakkola, Professor MIT, The Dark Secret at the Heart of AI by Will Knight MIT Technology Review)
- *". . . in many real-world applications it will be desirable to examine the internal decision-making process in detail"* (Charles McLellan, ZDNet, Inside the black box: Understanding AI decision-making)
- *". . . it's also a bit unsettling, since it isn't completely clear how the car makes its decisions."* (Will Knight Senior Editor AI, MIT Technical Review, The Dark Secret at the Heart of AI)

Explainability is Difficult with Traditional ANNs

With the reliance of current ANNs on mathematical and statistical operations, trying to retrofit explainability on the system is an enormous aspiration. Currently there is no known method to accomplish this without significant compromise. Whether it is possible to create a mapping from a neuron network's internal complexity to the human cognitive ability may be debatable. A solution needs to be sufficient to stand up to public criticism and governmental regulation. Note that people have tried various strategies for explainability described below.

External References:

- *"The system is so complicated that even the engineers who designed it may struggle to isolate the reason for any single action… And you can't ask it: there is no obvious way to design such a system so that it could always explain why it did what it did… We've never before built machines that operate in ways their creators don't understand."* (Will Knight Senior Editor AI, MIT Technical Review, The Dark Secret at the Heart of AI)
- *"If you had a very small neural network, you might be able to understand it… But once it becomes very large, and it has thousands of units per layer and maybe hundreds of layers, then it becomes quite un-understandable. . ."* (Tommi Jaakkola, professor MIT, The Dark Secret at the Heart of AI by Will Knight MIT Technology Review)
- *"We haven't achieved the whole dream, which is where AI has a conversation with you, and it is able to explain… We're a long way from having truly interpretable AI . . ."* (Carlos Guestrin, professor University of Washington, The Dark Secret at the Heart of AI by Will Knight MIT Technology Review)
- *"If you look at the neural network, there won't be any logical flow that a human can understand, which is very different from traditional software . . ."* (Guy Katz, postdoctoral research fellow in computer science at Stanford, Finally, a peek inside the 'black box' of machine learning systems, by Marina Krakovsky Stanford Engineering News)

Current efforts to Create Explainability in ANNs

There are a number of organizations currently attempting or using various strategies to provide explainability. These efforts have succeeded to varying degrees but each with a downside. The efforts can essentially be categorized into several strategies:

1. **Trade accuracy for explainability** – it is possible to use a more rule-based approach to machine learning that is inherently easier to explain than an ANN. However, these methods are not as accurate as an ANN in their conclusions. It is also important to remember that the machine learning advancements in recent years that have resulted in widespread public awareness and adoption have been the area of ANNs and not rule-based machine learning.
2. **Reduce data inputs or use specially designed data or tools to deduce ANN operation** – these methods do in fact provide insight into ANN operations. However, they either compromise the free use of data or only provide insight via deductions. Whenever these methods include problem specific data to help with inferences they chip away at a core value of machine learning, reusability.
3. **Extract pieces of information giving indications** –This strategy attempts to extract portions of an image or create a visualization of some sort that gives a user insight into learning. Several high-profile efforts are currently underway in this category and this seems to be the most promising area of "black box" research for ANNs. However, this type of information and insight is quite minimal compared to the entirety of ANN

operations. It's comparable to an English only speaker trying to understand a Mandarin conversation by picking out a few words from any given sentence. While it is helpful, it is not reliable. It is certainly not ready for making life and death or business critical decisions.

(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

DARPA XAI Project

useAIble defines explainability as:

1. Complete access to all data and rationale behind network operation and conclusions.
2. Provide direct access to data from machine learning engine (versus indirect access as explained above).
3. Provide access both while learning is occurring and after it has completed.

No current explainability strategies meet these criteria and there is skepticism as to whether it is achievable.

External References:

- *"Some models are extremely accurate but not very interpretable. Some models are highly interpretable, but not all that accurate. And AI researchers are starting to design models that are optimized for both – though skepticism remains about whether this is even possible."* (Janet Wagner, SiftScience.com, Machine Learning Isn't Always a Black Box)

Impact on Real World Deployments

Health and Medical

One of the more promising opportunities for machine learning is the area of health care. Significant advancements have occurred in improved accuracy of diagnoses and outcome prediction. However, the issue of transparency in these systems has not been overcome. Medical professionals and their patients need explainability to have confidence in the systems.

External References:

- *"We can build these models, but we don't know how they work."* (Joel Dudley, Lead Deep Patient Research Group Mount Sinai Hospital New York, MD Anderson Cancer Center's IBM Watson project fails)

Self-Driving Vehicles and Autonomous Aircraft

Machine learning technological development for self-driving vehicles, which ranges from autopilot features on cars, to self-driving trucks to autonomous aircraft, are receiving more effort and funding than many other areas of research. Because of the obvious safety concerns as well as governmental regulation, this is another area that cannot proceed to adoption without reliable explainability.

External References:

- *"The race by automakers and technology firms to develop self-driving cars has been fueled by the belief that computers can operate a vehicle more safely than human drivers. But that view is now in question after the revelation on Thursday that the driver of a Tesla Model S electric sedan was killed in an accident when the car was in self-driving mode."* (Bill Vlasic And Neal E. Boudette, NY Times, Self-Driving Tesla Was Involved in Fatal Crash, U.S. Says)
- *"That opacity can be worrisome when it comes to using neural networks in safety-critical applications, like preventing aircraft collisions."* (Guy Katz, a postdoctoral research fellow in computer science at Stanford, Finally, a peek inside the 'black box' of machine learning systems, by Marina Krakovsky Stanford Engineering News)

Finance and Banking

The applications for machine learning in the finance and banking sectors is great. These include investment and trading, credit and loan automation, banking services and many others. In fact, penetration into this domain has already begun. Again, governmental regulation as well as investor scrutiny makes the lack of explainability an issue that needs to be overcome. Investors, customers, regulators, and auditors all expect to know why critical financial decisions are being made. For machine learning to take hold in the financial industry it must meet the scrutiny of these and other parties.

External References:

- *". . . banks...are now turning their attention to more complex machine-learning approaches that could make automated decision-making altogether inscrutable . . . "*(Will Knight Senior Editor AI, MIT Technology Review, The Dark Secret at the Heart of AI)

Criminal Justice and Military

Machine learning's advancements in predicting human behavior has made it relevant to some criminal justice areas. Social issues of equality and legal rights also have proponents and sceptics for opportunities and concerns for machine learning. Similarly, inroads have been made by the military into machine learning for robotics and other operations. Although these domains are different with regard to external accountability, the basic need for transparency remains high. In any potentially life altering or even ending situations the decisions being made must be clear and understandable. In these fields the public's lack of trust in military automatization or in criminal justice automation exacerbates these issues intensely.

External References:

- *"The U.S. military is pouring billions into projects that will use machine learning to pilot vehicles and aircraft, identify targets, and help analysts sift through huge piles of intelligence data. Here more than anywhere else, even more than in medicine, there is little room for algorithmic mystery, and the Department of Defense has identified explainability as a key stumbling block."* (Will Knight Senior Editor AI, MIT Technology Review, The Dark Secret at the Heart of AI)
- *". . . in May this year, COMPAS, a proprietary risk assessment algorithm that's widely used to decide on the freedom or incarceration of defendants passing through the US criminal justice system was alleged by online investigative journalism site ProPublica to be systematically biased against African Americans compared to whites. Although Northpointe (the for-profit company behind COMPAS) disputed ProPublica's statistical analysis, generating further controversy, the widespread use of closely-guarded proprietary algorithms in sensitive areas such as criminal justice is a cause for concern at the very least."* (Charles McLellan, ZDNet, Inside the black box: Understanding AI decision-making)
- *"Rachael Tatman, a National Science Foundation Graduate Research Fellow in the Linguistics Department at the University of Washington, found that Google's speech recognition system performed better for male voices than female ones when auto-captioning a sample of YouTube videos, a result she ascribed to 'unbalanced training sets' with a preponderance of male speakers. As Tatman noted, a few incorrect YouTube captions aren't going to cause any harm, but similar speech recognition biases in medical or connected-car applications, for example, would be another matter altogether."* (Charles McLellan, ZDNet, Inside the black box: Understanding AI decision-making)

Opaque by Their Very Nature

To better understand why ANNs operate as a black box, it's necessary to look closer at how they function and why. Probably the most significant divergence between the design of an ANNs and traditional software development is a paradigm difference between the use of mathematics and statistics as the algorithm (ANN design) versus manual coding as the algorithm (software development). ANNs use a series of mathematical functions to manipulate the state of the network over thousands or millions of iterations, aggregating data in each operation, to produce a conclusion or prediction. Tracking the relationship between original inputs and determined outcomes becomes increasingly difficult with each aggregation and calculation. These systems can often aggregate data millions of times before producing results. This lack of tracking, by a human or through data, makes ANNs opaque by their very nature.

> *". . . there's a potential fly in the AI ointment: the workings of many of these algorithms are not open to scrutiny -- either because they are the proprietary assets of an organisation or because they are **opaque by their very nature**." (Charles McLellan, ZDNet, Inside the Black Box: Understanding AI Decision Making)*

An even more fundamental understanding of ANN design is to recognize *what* the mathematical and statistical operations are doing. At its most basic level, an ANN is being exposed to a sequence of experiences (through iterations of data passing) and each time these sequences occur, the ANN is applying an aggregation to those experiences. The type of aggregation can vary widely from one ANN to another, but all are solidly based upon this paradigm.

Consider the level of complication that these millions of aggregations add to our problem's complexity. If this were the only way for machines to learn successfully, maybe we would have to live with it. However, many people believe that a more sophistical algorithm could simplify the problem and possibly expand the types of problems that machine learning can handle. This is notable in "The Master Algorithm" by Dr. Pedro Domingos. Dr. Domingos speculates that a better core algorithm may be the solution to new horizons in machine learning.

$$\left| \begin{array}{c} \begin{aligned} a_1^{(2)} &= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\ a_2^{(2)} &= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \\ a_3^{(2)} &= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \\ h_{W,b}(x) = a_1^{(3)} &= f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)}) \end{aligned} \\ = \\ \begin{aligned} J(\theta) = &\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[-y_k^{(i)}\log((h_\theta(x^{(i)}))_k) - (1-y_k^{(i)})\log(1-(h_\theta(x^{(i)}))_k)\right] + \\ &\frac{\lambda}{2m}\left[\sum_{j=1}^{25}\sum_{k=1}^{400}(\Theta_{j,k}^{(1)})^2 + \sum_{j=1}^{10}\sum_{k=1}^{25}(\Theta_{j,k}^{(2)})^2\right]. \end{aligned} \end{array} \right| * 10^{10,000,000} > \text{Human Cognitive Ability}$$

Complicated Nature of Deep Learning Methods

Clearly, as ANNs are applied to real world applications across a variety of industries, the need for explainability is a must and the disconnect between human understanding and current ANN design is an obstacle.

Gartner’s popular “Hype-Cycle” demonstrates how many technologies are not yet passing to the “Slope of Enlightenment”. It is clear that the “black box” issue will continue to fuel this type of negative publicity and will likely inhibit the full adoption of ANNs until it is solved.

Many AI-related technologies are approaching, or have already reached, the 'peak of inflated expectations' in Gartner's Hype Cycle, with the backlash-driven 'trough of disillusionment' lying in wait.

Image: Gartner / Annotations: ZDNet

Since the design of current ANNs are inherently opaque, the solution may be a different approach to design that altogether avoids the problem. This philosophy drives useAIble’s patent pending system for machine learning.

Transparent by its very Nature

The Ryskamp Learning Machine

The Ryskamp Learning Machine (RLM) operates differently than traditional ANNs. First, it does not aggregate data as part of the learning process. In fact, the RLM works by saving as much data as it possibly can. It can recall every input combination it ever sees, what logic was used to find a solution for that input combination, which solution was chosen (a solution is a combination of outputs), and what feedback was received (e.g. score, etc.). The fact that the RLM remembers everything makes transparency natural.

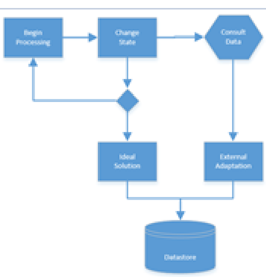
“The fact that the RLM remembers everything makes transparency natural.”

This new system or paradigm focuses on using today's advances in hardware, today's complex problems, while using many concepts pioneered by 60 years of machine learning innovation.

Many of today's machine learning methods attempt to describe everything in mathematical formulas. While math and statistics have been great for categorization, pattern recognition, classification, and the other problems associated with machine learning, these methods have their limitations.

Human learning is complex and when abstracted we see that it uses logic as much as math. Let's look at an example:

> When a human learns, they begin with categorization yet they end with specific knowledge: Bob walks into a room full of strangers. He immediately notices that some are clustered on the south side of the room and some are clustered on the north side of the room. After walking around, he realizes that everyone on the south side of the room is discussing management and everyone on the north side of the room is discussing marketing. For the moment, Bob has inferred that people on the south side of the room are marketers and the people on the north side of the room are managers. This is classical categorization.
>
> Then Bob meets Cindy. Cindy is a manager yet she is friends with the marketers and Bob finds her on the south side of the room. Bob and Cindy get talking and Bob learns that Cindy has three children, a boat, and two cats. These facts about Cindy are all specific knowledge. The fact that she breaks the categorization of "marketers in the north" and "managers in the south" is a piece of specific knowledge that breaks or overrides Bob's initial inference and the layout of the room. This is analogous to how the RLM is able to handle "edge cases" so well.
>
> The combination of categorization and specific knowledge is something we take for granted in human learning. Since this information is all stored in a non-aggregated format the RLM can recall Bob and Cindy's interactions later when trying to asses why certain decisions were made. This open data store of all collected information makes it easy to explain decisions made by the RLM for any given point in time.



VS

$$\begin{aligned}\mathbb{E}[\mathbf{x}] &= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \int \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \mathbf{x}\,\mathrm{d}\mathbf{x} \\ &= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \int \exp\left\{-\frac{1}{2}\mathbf{z}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{z}\right\} (\mathbf{z}+\boldsymbol{\mu})\,\mathrm{d}\mathbf{z} \qquad (2.58)\end{aligned}$$

Logic vs. Purely Mathematical Machine Learning

This approach has many benefits. For information on other benefits see http://useaible.com/7-machine-learning-breakthroughs/.

"In our attempt to understand the complex world of machine learning (or learning in general) we cannot always leap frog into mathematics. The reality may be such that our current mathematics are woefully incapable of describing what is happening."

Carlos E. Perez - DeepLearningPatterns.com

RLM Architectural Details

Introduction

To accept that state aggregation for machine learning isn't the only option, it is necessary to understand, at a more technical level, how an alternate system is designed and implemented. If the RLM doesn't operate on formula based aggregation, how does it operate?

The first fundamental divergence of the RLM is the replacement of formula based state aggregation with logic based processes that describe the flow and operation of generic machine learning.   In addition, the RLM includes a number of other divergent concepts, resulting in advancements, that fall out of the new architecture of logical processes; namely *best-known solution, similar neuron categorization, and randomized learning.* The similar neuron categorization is a method for selecting a set of related solutions to situations that are similar to the given situation. The best-known solution is a concept where the RLM intelligently considers similar solutions to the given situation and identifies the best fit. Randomized learning is an algorithm that simulates human learning which begins with attempting random efforts and ends with specific actions that have proven successful.

Logical Processes vs Formula Based State Aggregation

As has been described above, current ANN architecture is based on a series of mathematical formulas that iteratively change the state of hidden neurons. The RLM is based on a new architecture that defines and implements, in a logical way, the processes, operations and rules that define how learning takes place. These processes take into account learning observations from both human type learning and machine based learning.   This RLM architecture overcomes several limitations of the formula based aggregation method; namely 1) data used for aggregation is not lost 2) instant identification of exception cases without lengthy re-training 3) inherent transparency and explainability, and 4) training speed with more efficient processing of neurons as only one neuron is processed per cycle versus the entire set of neurons.

Given that the RLM is based on implemented logical processes, it may be suspected that the implementation is specific to a particular domain or data set and that it isn't truly a generic learning system. This is not the case. The code that implements the RLM system is a generic code base implementing generic operations and rules for a generalized learning process. It does not include any code that is specific to a particular domain or learning situation. In fact, the system has been applied, as is, to a variety of divergent situations. Visit the useAIble website for online examples or download the source code examples at www.useaible.com/code.

Learning and Data Persistence

The heart of the RLM's ability to explain its decisions and conclusions is data persistence and the data store. Based on the RLM posit that all learning data can and should be persisted, because of pervasive and cheap storage, the architecture doesn't discard any data.

The result of this simple architectural element is a complete and extensive repository of the networks progress, related data and outcomes. It can be used for evaluation, ranging from a quick decision explanation to analysis of a data set's affect on learning, to activity of the network's learning process.

The logical nature of the RLM allows this data to be combined with the core RLM algorithms to explain how decision were made at any point in time.

RLM Persistence of all Learning Data

The RLM has several types of data that are essential to both general machine learning concepts as well as the unique way in which the RLM architecture works. With each cycle of the RLM learning process, inputs with specific data values and resulting outputs are persisted to a data store. RLM SmartNeurons for each input/output mapping are also an important part of the architecture and are persisted. Two concepts that help to understand explainability inside the RLM are *cases and solutions*. A case represents a single attempt by the RLM to provide a decision, action, categorization or recommendation. The "case" data structure contains all data and information relevant to a single learning point. A solution represents the best learning or accomplishment of the RLM for a given set of inputs at a given point in time. It includes a collection of cases that are used with various algorithms to determine the best outcome or output.

Best Known Solution

In the RLM, the basic unit of calculation is called an Rneuron. An Rneuron represents a unique set of input values. When this unique set of input values appears, historical information associated with the Rneuron is used to choose a best-known solution. This best-known solution is drawn from a number of cases where the network has previously encountered this specific combination of data. Based upon feedback provided to the network, the RLM is able to identify which of these solutions performed “best”. The best-known solution will be selected when randomization does not apply and when enough information is not available for categorization (see below).

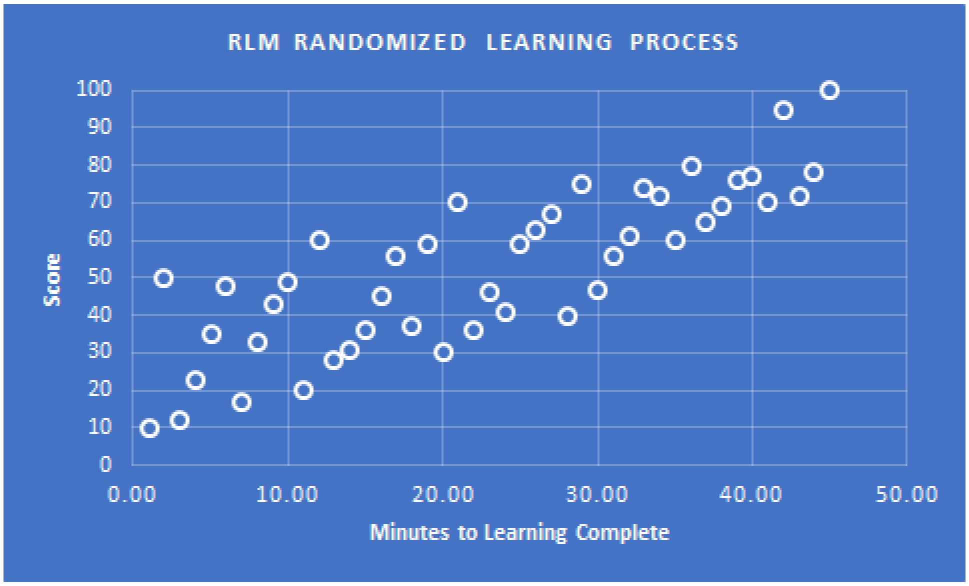
Categorization Through Similar Rneurons

So, what happens when no historical data has been associated with a unique combination of inputs? This is where the categorization or generalization feature of the RLM performs. When a lack of adequate historical data is available for an Rneuron, it will look for associate “neurons”. This means it will look at RNeurons with similar values, rather than exact values, to make generalizations about a solution. Our example above with Bob in a room with certain people on the south side and certain people on the north side is a good generalization of when this would occur. When a lack of specific information exists, Bob’s inference of people on the north side would be substituted for specific information about a new person encountered.

Randomized Learning

The algorithms for randomized learning and applying best-known solutions are similar to how the human mind works as it learns. For example, when a child is completely new to a situation or object, say a red ball, they try random actions. Maybe the child will sit on the ball, put it in their mouth, or kick it. Each time the child tries an action they get feedback; sitting on the ball results in falling off, putting it in their mouth results in a bad taste, kicking it may results in cheering parents. The child will likely continue trying random actions on the ball for some time. Yet with every repeated action, the randomness decreases until finally, the child is familiar with the ball and all the activities that can be performed with it and their associated outcomes.

The RLM implements a similar randomized process. Whenever the system is learning, there is some probability of the system selecting a randomized output over best-know solutions or associated RNeurons. As more data is collected and results are processed, randomness is likely to occur less often than selecting known results in both humans and in the RLM. As in the child and ball example above, randomness is essential to learning and will occur to a varying degree over a long period of time.



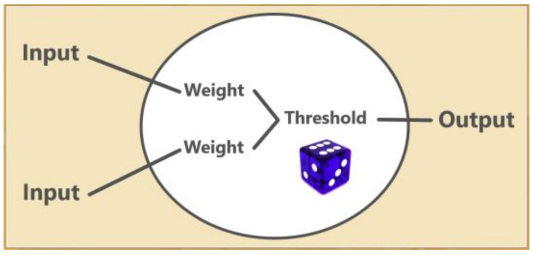
Decrease of Randomization over Learning Process

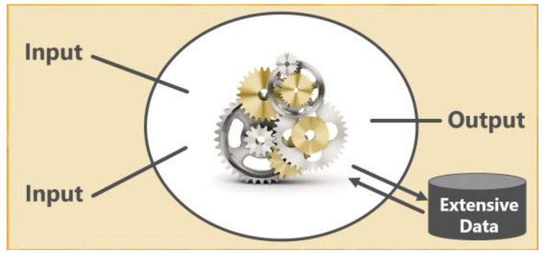
Dynamically Created SmartNeurons

A concept that is fundamental to the RLM system and that is closely tied to data persistence and explainability, is the way the internal or intermediate neurons are created and managed (dubbed SmartNeurons). In a typical ANN, the number of neurons are configured before training begins and usually remain static throughout. In the RLM, neurons are created on an as-needed basis according to what data combinations and patterns are presented to the system.

This is one of the characteristics of the RLM that makes it easier and quicker for a software developer to apply. The developer doesn't need to make educated guesses about the neuron population nor go through a trial and error process to find the configuration required for success. Since the RLM is already in the business of categorizing data patterns and looking at relationships among neurons, it is in a better position than the developer to efficiently create and manage neurons.

A beneficial result of the RLM using rules and logic processes to manage neurons is a substantial increase in learning speed. The RLM doesn't go through the entire neuron population on each iteration. It determines which neurons are applicable to a given input data pattern and only operates on those. This means that it is not necessary to touch a large number of neurons on each iteration. This can often reduce the time per cycle by orders of magnitude.





Traditional neuron vs. an RLM SmartNeuron

The RLM Learning Visualizer

A Tool for Viewing and Explaining RLM Solutions

The RLM source codes includes a toolkit that provides an RLM Learning Visualizer (RLV) as well as allowing a customized learning visualizer to be built. The RLV is used for viewing and investigating the solutions that are generated by the system. It can be used for gaining necessary insight into the learning process as well as gaining explanations of why certain decision were recommended.

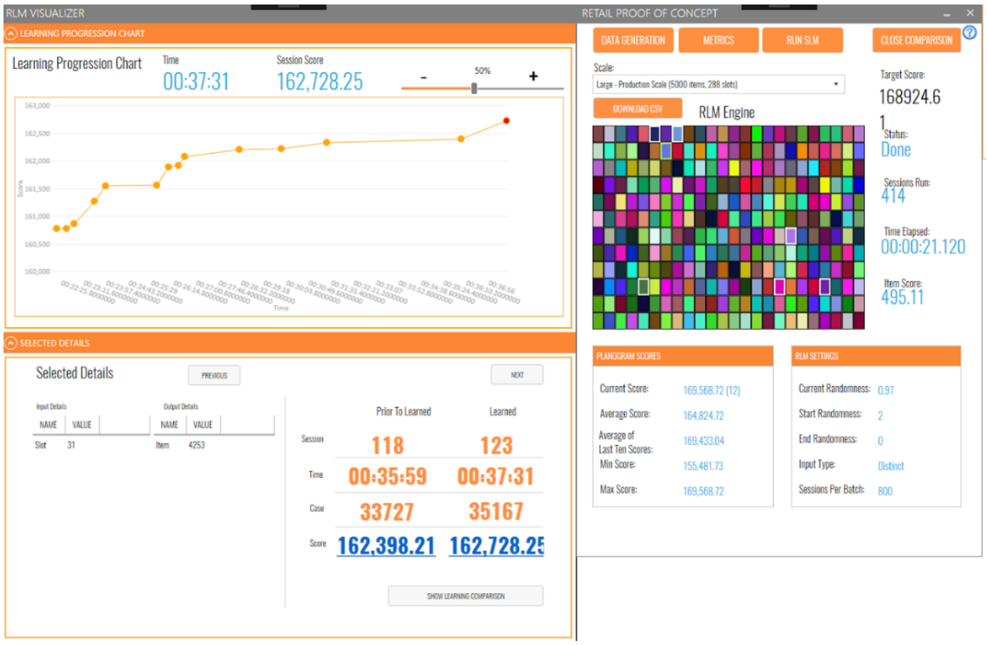
The system provides a type of summarization view that abstracts and filters the data in a way that makes it much easier and quicker to get insight and rationale for decisions and recommendations. One interesting and complicating aspect of applying machine learning to varied and disparate domains is that *what* the system learns and produces during deployment, as well as the terms and concepts used to explain the results, can be substantially different. To this end, the RLV provides both a plugin mechanism and a customization feature that allows the tool to adapt to differing domains. The RLV plugin mechanism allows the implementers of the RLM, for a given domain, to provide a view for the solution that is specific, understandable and useable for that field. Also, the entire UI of the RLV is customizable to provide for the use of different terms that make sense to each given domain.

Details of Operation

As described previously, since the RLM is not an extensive sequence of mathematical functions, all data for each cycle of the system is automatically persisted to the data store. This means it is possible to track the status of both the learning process and the prediction process after deployment. Every aspect of the RLM generated model can be inspected to see what decision was made, what data is related to that specific decision, and what the state of the model was both before and after that decision.

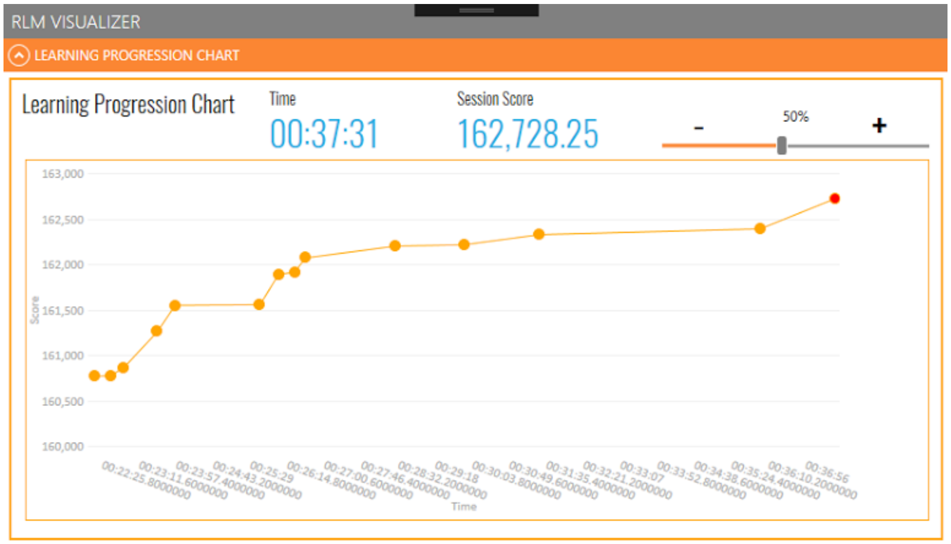
The system records learning events to properly display the development of the scores from the early sessions to the last. The RLV is composed of three panels that each display a different view into the system and separate set of data. The *Timeline Panel* displays a history, from beginning to end, of the significant learning points or solutions in the given model. The *Details*

Panel displays all the data relevant to a chosen solution. The *Output Visualizer* is a panel that shows a graphical visualization of a solution that is specific to a particular domain.



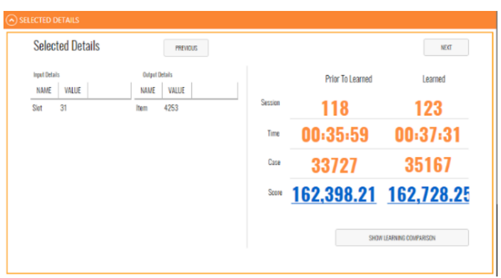
RLM Learning Visualizer

The Timeline Panel is a graph with the y-axis being a score and the x-axis being time from the start of learning to the end of learning. What range a score can span is specific to a given domain and is also not absolute but relative to other scores. Because the extent of data the RLM stores is usually quite large, it would be unusually cumbersome to sort through it all when trying to get an explanation of why the RLM concluded something. For this reason, the Timeline Panel essentially filters out only the significant learning points in the data store. This makes it easier to quickly see how learning or operation has progressed and inspect any number of solutions along the way. The granularity of this solution filtering can be adjusted to be finer or coarser via a simple slider bar control in the panel



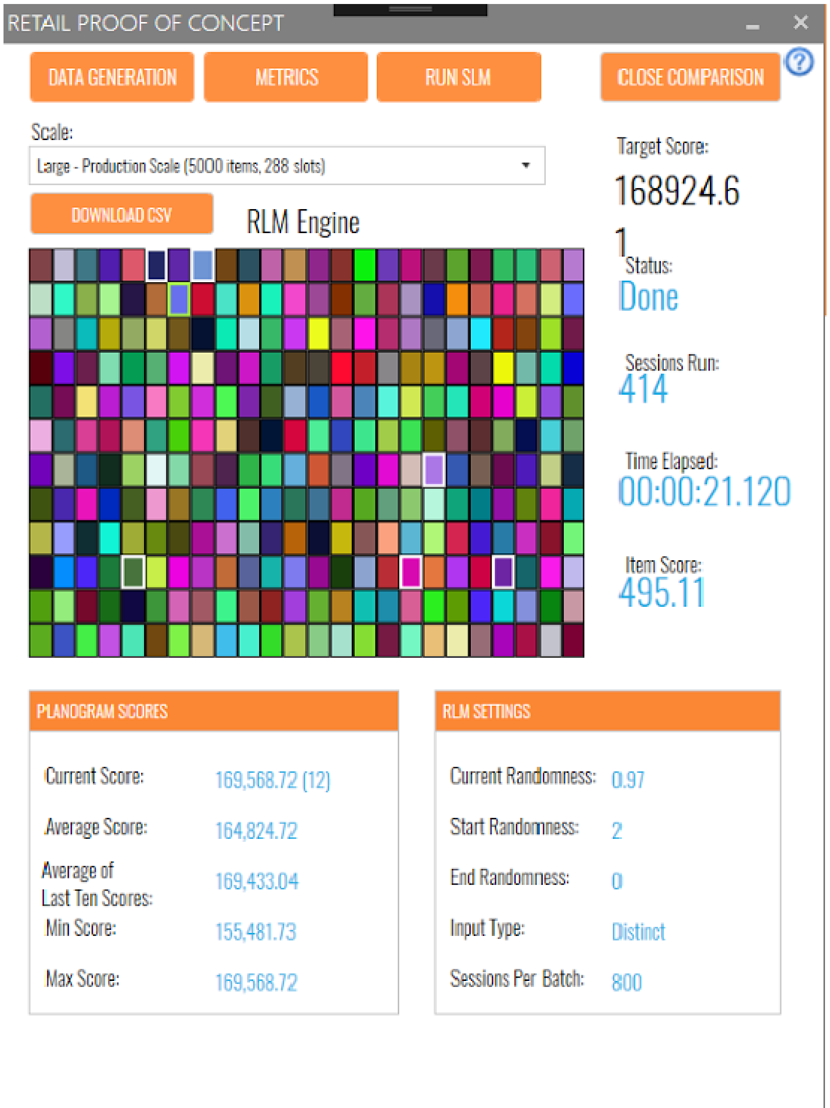
Learning Timeline Panel

The Details Panel is where the details of a selected solution can be inspected. This panel displays the data for whatever solution has been selected in the Timeline Panel. A comprehensive list of both input and output values, that are related to the selected solution, are displayed here. This is a significant set of information in understanding RLM decisions, as it can be determined what input data was used to arrive at the given output. Other data related to the selected solution can be seen here including session number, time, score and percent improvement over the previous solution. Via previous and next controls, the display can be moved chronologically through the datastore.



Details Panel

The Solution Visualizer is useful in that it gives a graphical representation of a solution that is more meaningful to a person familiar with a particular domain or field. This solution could be the final result of RLM learning or any solution along the path. What the panel displays is dependent on what is appropriate for the given domain. For example, the Retail Store Plan-o-Gram displays a store shelf with the recommended items populating the shelf. This panel is actually a plug-in to the RLV, as the content it displays needs to be defined by the implementer for the domain.



API for External Reporting Integration

Many companies that use machine learning to improve business processes will have their own reporting system to access raw data directly. From this context, the RLV could be considered a reference implementation showing what RLM data is available and how to access it. For such a reporting system, all underlying data is available for retrieval via an RLV API or even directly from the data source. This allows integration of the RLM, a specific model's data and possible portions of the RLV into any external reporting system.

Conclusion

As has been demonstrated, the machine learning community is well aware of both the advances and practical applications of machine learning as well as the deficiencies of these systems in providing insights and explanations for their operations and decisions. Because of governmental oversight, safety factors, and general expectations for accountability, ANNs currently have a significant obstacle to general deployment in many market sectors. There have been numerous attempts by companies and organizations to deploy ANN based systems, but were unable to proceed because of the lack of machine learning explainability.

> *"There's already an argument that being able to interrogate an AI system about how it reached its conclusions is a fundamental legal right. Starting in the summer of 2018, the European Union may require that companies be able to give users an explanation for decisions that automated systems reach. This might be impossible, even for systems that seem relatively simple on the surface, such as the apps and websites that use deep learning to serve ads or recommend songs. The computers that run those services have programmed themselves, and they have done it in ways we cannot understand. Even the engineers who build these apps cannot fully explain their behavior." (Will Knight Senior Editor AI, MIT Technology Review,* The Dark Secret at the Heart of AI)

The Ryskamp Learning Machine is a new machine learning architecture and system implementation that inherently provides transparency and explainability into its learning process, decisions and actions. This functionality fills a critical void that is needed for machine learning to be deployed in many market sectors. The RLM system is open sourced for evaluation purposes and is available for use today. www.useaible.com/code.