

The Ryskamp Learning Machine

A Technical Overview

Introduction

Are you ready for an algorithmic-level shift in machine learning? useAble has invented a class of artificial neural networks that sets a new standard in performance, accuracy and problem-solving abilities. As Neil Lawrence (University of Sheffield and Amazon AI) has stated "So while we are making considerable progress on tasks which were once thought extremely difficult or impossible, the truth is that the progress is driven far more by the availability of data than an improvement in algorithms." useAble's patent-pending technology addresses Mr. Lawrence's concern with a new core algorithm and neural network management architecture that is very different from that of its predecessors.

useAble's algorithm, dubbed the Ryskamp Learning Machine (RLM), has shifted away from the purely mathematically-based neural network algorithms in favor of a more human-like logic based learning and recall algorithm. While the algorithm, associated methods and systems do include the categorization and pattern recognition abilities of traditional neural networks, it also includes the more human-like ability to recognize exceptional cases after only brief learning exposure. For example, the human brain solves a classic maze by using both generalization and specific case recall without the need for thousands upon thousands of attempts. Unlike traditional machine learning systems, useAble's system can perform this same function just as a human would. After a single run through of a maze, it can recall the correct steps every time, while traditional systems must attempt the maze many times before neurons can recall this level of granular specificity. With this combination of pattern recognition and specific case recall, the network training process reaches a state of convergence, with this type of problem, exponentially faster and with more accuracy than traditional weight-based networks, which can only learn after many iterations. This is all accomplished without sacrificing performance on traditional pattern recognition or any other type of problem. Stated another way, this is an additional native feature, not a trade off as you might see between two traditional neural networking algorithms.

useAble has also re-architected the creation and management of neurons and layers from a static approach to a dynamic and more intelligent approach. This introduces the ability to solve a wider spectrum of problem types than traditional machine learning algorithms. The useAble solution learns on small devices without help from large cloud-based server farms and data centers. Since the core foundation of the algorithm is logic-centric rather than math-centric, useAble is also able to record and recall all decisions, conclusions and related data. This moves the RLM away from the significant problem of "black box" networks that cannot explain critical decisions. The combination of a pre-defined logical neural structure, logic centric algorithms, and extensive past decision storage creates an environment where diagnostics are not just possible yet are easy and native.

The structure of the network is simple and straightforward, consisting of only developer defined inputs, outputs and a few configuration parameters with the remaining aspects of the network are handled automatically by the RLM. This contrasts with the traditional network where the developer must painstakingly choose and specify a combination of discrete components (such as algorithm, activation

function, optimization function, etc.) This inherent simplicity in RLM usage takes away the need for extensive developer training or exclusive AI experts to conduct serious AI projects. A standard developer can now access the power of machine learning without dedicating years to specific knowledge acquisition.

Finally, an interesting artifact of using a logic-based neural network versus a mathematically-based approach is that the entire race to provide server farms and high-end GPUs becomes obsolete. Because useAble's technology is a revolutionary core algorithmic change and not just a new AI toolkit or adaptation of existing approaches to a specific domain, its influence and effects can be realized in a wide range of market domains, implementation strategies, and across technology boundaries. The overall solution simply uses less resources than its traditional counterparts.

The Seven Ryskamp Learning Machine Breakthroughs

A Neural Network Paradigm Shift

SmartNeurons with Cognitive Type Logic over Mathematics

"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

UseAble's RLM implements a logic based neural network vs. a mathematically based approach. This does not require the parameterized type categorization that is often associated with math based approaches such as traditional neural networks.

The RLM moves away from:

- Activation and optimization Functions
- Rudimentary weight calculations

In favor of:

- Remembering Best Known outcomes
- Intelligent creation & use of neurons

Breakthrough #1

A More General Problem Solving Engine for Real World Applications

Traditional machine learning algorithms tend to excel at specific types of problems and are less effective or even ineffective at solving other categories of problems. useAble's machine learning engine has been shown to solve a variety of divergent problems. This is attributed to several factors including intelligent management of neurons, intelligent persistence of neuron state and a paradigm shift in evolution of the network structure during training.

For several examples of diverse problems sets that the RLM can solve, visit useAble's [Machine Learning Challenge](#).

Note that the challenges at the learning challenge site are very different types of problems. The first is a lunar lander that is considered a "101" machine learning game, the second is a simple maze, and the third is a logistics simulation game that does not use traditional "inputs" i.e. the engine must simply provide a value for each output (setting in the game) and then receive a score. While traditional approaches work well on the lunar lander, they do not on the maze or the logistics simulation game. Neither problem fits well into what neural networks and other machine learning engines do today.

The code for these challenges is available at GitHub (see Resources below) and we encourage you to play around with your own combinations.

Breakthrough #2

Neural Network Approximation + Human Type Specific Knowledge Recall

The human brain can generalize things into broad categories that become more specific over time, as well as instantly handle exceptional edge cases within a category that must be handled in an exceptional way. For example, a person only needs to touch a hot stove once to permanently learn that a bright red electrical unit on a stove is dangerous.

useAble's RLM learns the way a human learns. It initially uses general categories which get more specific as learning progresses. The RLM is also designed to identify specific cases and recall them instantly when encountered again in the future. Traditional machine learning typically fails at correctly handling edge cases in this context. In current AI implementations, programmers typically use a combination of machine learning and manual algorithmic coding to deal with this.

For an example of Specific Knowledge Recall, look at either the [Maze Sample app in Python](#) or [Maze Sample app in C#](#), or for those who prefer a visual experience see the WPF Maze Sample

all available at GitHub (see resources below).

Breakthrough #3

Changes the Cost/Benefit Ratio of Machine Learning vs Traditional Coding

Machine learning is typically only used when machine learning *must* be used. Machine learning projects require tremendous resources including AI data scientists and/or developers with highly specific domain knowledge, intense hardware requirements and months or years of network training. Because of these high barriers to ML implementation, most programming tasks are still hand coded by developers using traditional coding approaches. UseAble's RLM enables a simple, fast and reliable API with a simple scoring mechanism. Developers can now choose this approach over the costlier manual algorithmic design and coding for more tasks than is currently the norm.

For an example of quickly and easily replacing manual code, that often wouldn't be thought of as AI applicable, contact useAble for a demo of a complex Retail Plan-o-Gram [here](#).

In this project, useAble spent 90% of resources dealing with user experience issues, as the logic was already provided by the RLM. Traditionally the bulk of the coding time would be spent on the logical algorithms that perform the back-end analysis. This focus on user experience changes the way projects are managed and approached and gives developers more time to focus on the user experience.

Breakthrough #4

Learn in a Fraction of Time and Resources

Over the past few decades, exponential advancement in computing hardware have occurred. Most traditional neural networks are rooted in algorithms that were designed prior to this massive change in hardware. The RLM leverages these newly available resources to fundamentally improve the neural network design and structure. The RLM operates easily on a standard 4 Core system with a common GPU.

The design of the RLM focused its priorities on 1) Real world business requirements and results and 2) Efficient hardware usage. The resulting RLM concepts and system are both significantly faster and surprisingly straight forward. To date, the RLM has been substantially faster and more accurate compared to other popular systems.

To see a case where the RLM learns significantly faster than other well-known AI systems, explore the source code for the Lunar Lander in either [Python](#) or [C#](#).

Also note that the framework provided on GitHub is simple enough that users may easily design

their own tests and games to see how the RLM compares to other machine learning techniques.

Breakthrough #5

Easily Re-trainable

It's generally accepted that a machine learning system will conduct a long period of training prior to being useful. Any changes to data or configuration requires a lengthy retraining process. At useAble, we generally reject this notion because it cripples many real world iterative implementation attempts. An inherent but very fortunate artifact of the RLM's design for Specific Knowledge Recall is that it can reach a state of convergence or "sufficient learning" in a fraction of the time typically expected. This makes required retraining during an iterative development cycle practical and doable.

This is particularly powerful in our plan-o-gram tool where users can optimize a retail shelf space plan, dramatically change the scoring criteria and then immediately reoptimize based on the new criteria. This "real-time" adaptation to a problem, criteria, environment and other factors is not possible with current training period expectations that often take weeks, months and sometimes years.

While the plan-o-gram tool is an excellent example of this breakthrough, a user can use any open source application, or write their own, to see how quickly the RLM can re-train after changes are made.

Breakthrough #6

Easily Implementable

useAble's RLM has the potential to transform the AI industry from a model where scientists and domain experts are required to produce an AI product to one where ordinary developers can do it unassisted. This is similar to the transition that happened in data persistence from the early days of databases, which required domain expert implements, to the current SQL toolkits allowing any developer to easily produce a solution.

Because traditional machine learning can often be difficult to implement, it is generally only used to solve problems where machine learning is the *only* option. UseAble changes this criterion. Since our system is a general-purpose problem solver, the developer is not required to have a broad knowledge of AI algorithms and how they apply to various problem domains. In addition, the simple and straight forward API negates the need for a highly-specialized understanding of toolkit graphs and configuration settings. In fact, almost no previous training is required. This is a fundamental departure from a field where machine learning scientists and experts are required for almost any AI usage.

To get a more detailed understanding of how straight forward and easy it is to apply, review the actual API [here](#).

Breakthrough #7

Explanation of Decisions

With the typical mathematical-based neural network, weights, biases, activation and optimization functions make understanding how the network trains difficult and explaining why it makes particular decisions almost impossible. Unfortunately, this creates a huge barrier to public deployment of neural networks in certain domains. useAble's neural network is not mathematically based and its logic based approach requires data, not statistics. Therefore, the RLM natively tracks all information about decisions as part of its core algorithm. The RLM provides a simple API to recall and analyze network activity, prediction and decisions. The so called Black Box Problem is a non-issue with the Ryskamp Learning Machine.

You can find a sample app demonstrating how the RLM recalls decisions and data [here](#).

In Conclusion

useAble's Ryskamp Learning Machine, while still a neural network architecture, departs fundamentally from the current and traditional theory of mathematically based neural network structure. Its use of coding logic to create and manage the neurons, and produce output, creates a paradigm shift that results in a number of significant changes and improvements over existing neural networks and other machine learning algorithms. Specifically, these changes are...

1. A More General Problem Solving Engine for Real World Applications
2. Neural Network Approximation + Human Type Specific Knowledge Recall
3. Changes the Cost/Benefit Ratio of Machine Learning vs Traditional Coding
4. Learn in a Fraction of Time and Resources
5. Easily Re-trainable
6. Easily Implementable
7. Explanation of Decisions

To both publicize and validate this new type of neural network and related concepts, useAble has made the entire system available for evaluation through open source code. We invite you to consider, test and comment on the ideas presented in this solution.

Resources

[The RLM on GitHub](#)

[useAble Home Page](#)

[useAble's Head to Head Challenge](#)

[useAble Video Blog - Running with Scissors](#)