

What is *Exhaustive-Learning*? (WIEL)

--- DRAFT ---

Paul Cadaret

UNICON Inc, Rancho Santa Margarita, CA, 92688

Phone: 949-742-7538; Fax: 877-243-6431; Email: [alias-renew-2009\[at\]unicon.ws](mailto:alias-renew-2009[at]unicon.ws)

Abstract

UNICON's recent participation in the *Small Business Innovation Research (SBIR)* programs at the DOD and DOE has enabled us to envision how very large and fast artificial neural networks can be constructed. We believe that our patent-pending *CogniMax® pattern recognition technology* [1] allows us to employ very large and fast neural networks in a way that enables a concept we call "*exhaustive learning*" to be applied for the benefit of certain types of challenging technical problems. This paper attempts to describe the concept of *exhaustive learning* through a discussion of what it is, when it is useful, how it can be used, and how we have found it to be an important solution development tool.

1 What is the concept of *Exhaustive-Learning*?

Our *CogniMax® pattern recognition technology* development efforts over the last few years have given us reason to consider the potential implications of the construction and use of large and fast neural networks. Additionally, given the availability of large supercomputing systems methods may be found to effectively construct large and fast neural networks using those systems as well. As a result, we have considered how it might be possible to construct and use such systems to address certain types of challenging problems within government and industry. Recently we have focused on the capabilities of our pattern recognition technology as the basis for the definition of a set of large and fast neural network capabilities because we have expended a significant amount of time modeling our design approach. Our models have then provided us with a baseline set of capabilities for large and fast neural network configurations and we have used these system capability definitions as we have explored various types of pattern recognition enabled solutions.

Given through some means the availability of large and fast neural networks we then have the capability to learn a great deal of information about a problem domain that can be exploited in real-time for some purpose. This information can be considered to be a form of *knowledgebase* that can be exploited when developing certain types of system solutions. Fuzzy pattern recognition capabilities are inherent with many types of neural network systems. This implies that the knowledgebases constructed can be queried in ways that are quite different from those associated with traditional database technology. Because knowledgebases can be queried in a fuzzy way, this means that it is possible for us to perform highly structured and generalized queries. In other words, we can control the degree of query imprecision (or fuzziness) and we can control where and how much imprecision is allowed. This is a very powerful feature.

We will now briefly describe what fuzzy pattern recognition is and how our definition of a fuzzy knowledgebase differs from a typical database management system. With traditional databases one typically forms a query that is constructed from a series database table or field tests. As an example, one might search for records where the NAME field is "John" and the AGE field is between 25 and 30. In a limited sense this is a fuzzy pattern recognition operation. If we have a need to perform queries on dozens or hundreds of such fields fuzzy pattern recognition methods become far more complex. As an example, let's consider a FBI criminal record search example. We're looking for a male suspect; whose name is John, Joe, or similar; whose height is approximately to 58 inches; whose weight is approximately 180 pounds; whose hair color is "dark"; who was last known to be in the Los Angeles area. Although one could construct a database query to extract matching records those records returned would not necessarily indicate the **best fit** for the search query; neither would they likely be provided as a list of the best matches in some prioritized order. A prioritized best fit search response would likely require additional post-processing. This is one type of problem where a knowledgebase capable of fuzzy pattern recognition and search methods would likely be advantageous.

As another example, let's consider the problem of a database of facial images. In this case we ask the question: *how can we utilize existing database technology to form queries to identify the best match with a new unknown facial image to be identified?* It is likely clear to the reader that trying to use existing database technology in such an application is not a good fit for such an application. What is needed in such an application is a means to perform fuzzy pattern recognition over a potentially large number of variables. In this case, the variables are the pixels associated with the facial images. Even with such a capability, such a problem is more complex than just fuzzy pattern recognition between

[1] COGNIMAX is a trademark of UNICON Inc.

some pixel-block-1 and pixel-block-2. Other image processing methods would likely be needed to prepare the two images to assess proper correlation and to measure such correlation using some relevant metrics.

Given the above discussion we can see that certain classes of problems exist where a large number of variables are involved and the type of queries needed can benefit from fuzzy pattern recognition to provide query-result information as a list in best-fit order. Additionally, such problems may require a prioritized list of best-fit type matches as a response so that additional post-processing steps can be employed. If we can learn an extensive amount of information about a very narrow problem domain one could reasonably say that the knowledgebase generated is **omniscient** in its narrowly focused area. Although developing such a knowledgebase may take some time to generate, in principle it seems reasonable that such a knowledgebase can be constructed. To date we have explored a variety of problems in government and industry where the type of technical challenges to be addressed appeared to be massive in scale or scope and as a result these problems were seemingly intractable if approached using traditional computational methods. These have been indicators to us that exhaustive-learning enabled solution approaches might be a productive area of exploration.

Knowledge itself can be thought of as an enabling data resource when it is organized in the form of fast-searchable knowledgebases. A fast/fuzzy pattern recognition engine provides a capability to rapidly search properly organized knowledgebases. The ability to perform fuzzy queries with such an engine enables the generalization of previously learned knowledge. Together these resources and capabilities enable exhaustive-learning methods to be employed to address massive-data real-time problems.

In summary, we believe that exhaustive-learning is a solution enabling method that is based on the construction and exploitation of a large scale knowledgebases related to one or more aspects of a particular problem domain. It is important to note that we do not believe that exhaustive-learning methods typically yield solutions by themselves. Thus far as we have explored a variety of challenging technical problems in government and industry and we have observed that exhaustive-learning is more of a solution enabler. It allows us to look at very challenging technical problems involving massive data in some new and creative ways. This new perspective has thus far enabled us to envision elegant solutions to some extremely challenging technical problems that are likely not effectively solvable using traditional computing methods. Therefore, we have found it to be a valuable solution enabling method.

2 When are exhaustive-learning methods useful?

As mentioned above, we believe that the concept of exhaustive-learning is primarily a solution enabler. Typically, to be useful some aspect of the problem to be solved typically has a massive-data characteristic associated with it or available to be extracted from it. To begin our utility discussion here we start by providing a few examples. Below are a few significant summaries based on our prior research work exploring exhaustive-learning solutions:

1. Our **CAUNN** paper [2](#) described how an unusually fast computational system could be constructed using a Radial Basis Function (RBF) neural network to manage a large hyperdimensional feature space to deliver computational results. The problem domain addressed would likely require **>1M neurons** to the effective.
2. Our **MSVAPM** paper [3](#) described how an unusually fast video piracy detection system could be constructed using a RBF neural network to manage a large knowledgebase of video frame signatures. Such signatures would be generated to reflect certain characteristics of every video frame ever produced by every Hollywood studio. The problem domain addressed was estimated to likely require **~10.8B neurons** to the effective.
3. Our **WBDUNN** paper [4](#) described how an unusually fast behavior monitoring and characterization system could be constructed using a variety of large/fast RBF neural networks. Such a system could be used to manage a large knowledgebase of video game behavioral signatures. Such behavioral signatures would allow *Blizzard Entertainment* [5](#) to perform real-time monitoring of all of the gamespaces within their extremely

[2](#) Our “Computational Acceleration Using Neural Networks” (**CAUNN**) paper can be found within the **DTIC IR&D Collection** (www.dtic.mil) with Accession-Number **08241202** and within the SPIE online library (<http://spie.org/x399.xml>)

[3](#) Our “Massive Scale Video Anti-Piracy Monitoring” (**MSVAPM**) paper can be found within the **DTIC IR&D Collection** (www.dtic.mil) with Accession-Number **08241203** and within the SPIE online library (<http://spie.org/x399.xml>)

[4](#) Our “WOW Bot Detection Using Neural Networks” (**WBDUNN**) paper can be found within the **DTIC IR&D Collection** (www.dtic.mil) with Accession-Number **08241204**.

[5](#) See www.blizzard.com

popular *World Of Warcraft* videogame ⁶ so that automated game play using “bots” can be reliably detected. The problem domain addressed was estimated to likely require ~14T neurons to be effective.

In each of the examples listed above the problem to be addressed had some element of massive scale and hence massive data associated with it. After some study we observed that if certain attributes of the massive data could be recognized in real-time, then we could exploit this capability to enable an effective solution to the overall problem. Normally, the ability to perform fuzzy search queries on massive datasets in realtime is very difficult to effectively perform with traditional computing methods. However, if one has access to a large and fast neural network system then we can address such needs in some new, creative, and real-time ways.

If a problem area can be addressed by traditional computing means, then the concept of exhaustive-learning will likely provide little if any additional benefit. However, if the problem to be solved appears to require the manipulation and exploitation of massive data associated with the problem and such exploitation must be performed in the context of real-time constraints, then there is a high likelihood that exhaustive-learning methods may prove useful to the development of elegant system solutions.

3 How can exhaustive-learning be used?

As a solution enabler the concept of exhaustive-learning is not necessarily a highly targeted approach. Based on our efforts to date exploring a variety of challenging problems in government and industry we have come to the conclusion that the range of potential applications for exhaustive-learning methods is quite broad. Some examples would again likely be helpful here to illustrate our belief:

- A. In our **CAUNN** paper we describe how a series of somewhat unusual RBF feature-space maps can be constructed where the neurons are placed within the feature-space at regular intervals and each neuron would be modified to contain additional metadata that is associated with the computational system of interest. Using a variety of feature-space maps a computational solution can be developed that provides varying levels of computational precision across the breadth of the computational feature space. One of the interesting aspects of the solution described is that it largely relies upon the use of pre-trained neural networks; however, we also identified how a dynamic-learning strategy can be effectively employed to achieve certain useful objectives. Another interesting aspect of the solution described was the heavy reliance on fast ensemble neural network methods to implement an effective composite computational solution.
- B. In our **MSVAPM** paper we describe a more traditional use of a single large/fast RBF neural network that is used to provide a list of “best match” knowledgebase query results. The solution described shows how a massive knowledgebase can be constructed that contains signatures for every video frame that every Hollywood studio has ever produced. When a new unknown video file is uploaded to a filesharing web site (such as YouTube) the system described can exploit the knowledgebase previously generated and extract a series of “best match” video frame signatures. Because overlap is possible video frame temporal signatures are also considered as a post-processing step to greatly improve the overall video-segment recognition reliability of the system. Because this problem area is only concerned with identifying video piracy, the solution described relies entirely upon a pre-trained neural network knowledgebase.
- C. In our **WBDUNN** paper we describe a real-time behavioral monitoring system that is again based on the assumed availability of numerous large-fast RBF neural network systems each with the capability to maintain multiple knowledgebases. In the solution described only some very basic characteristics of “bot” behavior are assumed, however, the number of possible actions available to a bot-directed game-character is truly vast. Because of this we determined that the direct identification of bot-directed game-character behavior is not possible. As we explored this very challenging problem at length we were able to identify 2 potential weaknesses in bot-directed game-character behavior that we believe could be used to distinguish bot-directed game play from human-directed game play (with 11M subscribers). One important metric that was described was the development of a rate-of-learning metric that was itself based on exhaustively learning prior game-character position over time. A second method described was the identification of certain patterns of motion that could be clearly observed when the problem utilizes an unusual feature-space map that is formed after an unusual coordinate transformation.

In each of the examples above the exhaustive-learning methods employed were only a part of the overall solution. The exhaustive-learning capabilities enabled other methods to be employed that were then more directly tied to the

⁶ Blizzard claims to have 11M subscribers worldwide.

resulting solution. We believe that exhaustive-learning methods that are based on the availability of large/fast neural network engines allow the creation of knowledgebases that capture certain problem specific data. Such data is often relevant to certain aspects of the problem to be solved but by itself does not directly provide a solution. The ability to perform fuzzy pattern recognition on such knowledgebases in real-time then provides opportunities for somewhat indirect solutions to be developed. Such methods are generally unavailable to solution developers when they employ more traditional computing methods. Although the problems we have explored and the solutions developed to date have been quite obscure, they would likely never have been considered possible without the enabling concept of exhaustive learning.

3.1 An exhaustive-learning workflow description

Figure 1 below is provided as a means to illustrate the process of developing solutions when exhaustive-learning methods are employed. The figure starts off with a complex problem (in red) that has massive-data attributes associated with it. Because the solution required is real time in nature, processing massive-data to develop a real-time solution is often a goal that is not feasible using traditional computational methods. This is the type of problem that can often be enabled by applying exhaustive-learning methods. In the figure we show how the problem can be transformed into a more solvable form. The green box shows the same problem where the source of the massive-data stream has been studied so that useful solution-relevant metrics can be identified. In the figure the various elements of the transformation are each labeled using yellow boxes; in our textual description we refer to these labels as [A], [B], [C], and so on.

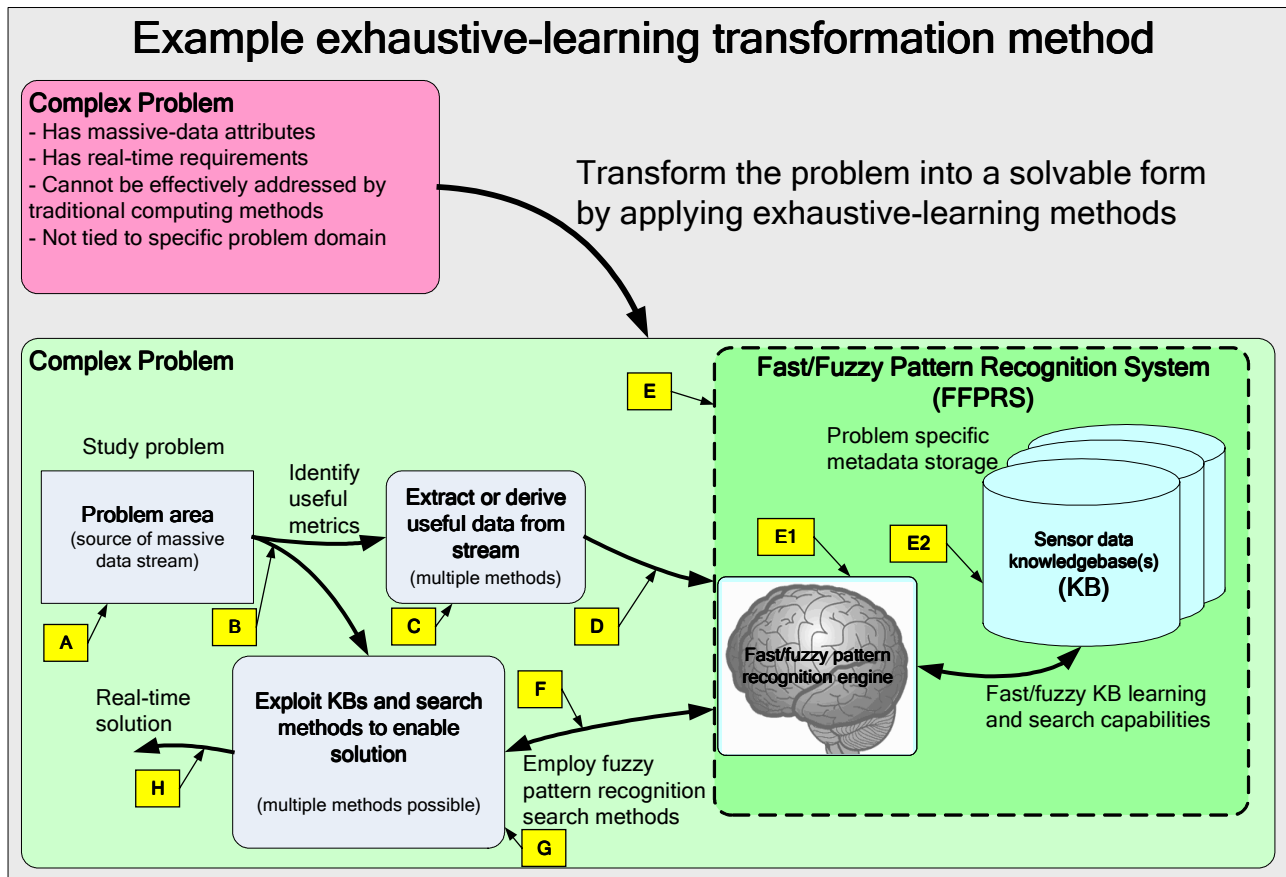


Figure 1 - Example Exhaustive-Learning Transformation Method

To describe the solution development process we start by identifying the problem to be solved [A] and studying the available data [B] with the intent of identifying useful metrics relevant to a potential solution. A means is then identified to extract or otherwise derive useful metadata from the massive-scale data stream. At this point we do not care how long it takes to extract these metrics; this problem will be addressed shortly. With metrics identified we then develop a system [C] that can be used to extract or derive useful solution-relevant metadata [D] from the data

stream. The data [B] and the metadata generated [D] is then presented to a *Fast/Fuzzy Pattern Recognition System* (FFPRS) [E] where the data and metadata can be **learned** and stored in one or more knowledgebases [E2] in an application-specific manner.

The learning process can be either static or dynamic in nature. If statically-learned we might accumulate knowledge regarding the incoming data stream for some time prior to using it in an online fashion. We generally strive to generate a knowledgebase that is “*omniscient enough*” to be useful in our limited problem domain. The metadata might be generated by a system [C] from the data stream [B] using some computationally intensive and slow batch process. This off-line generated “knowledge” (in the form of metadata) regarding the real-time data stream could then be learned and incorporated within the FFPRS knowledgebases [E2]. If our knowledgebases were to contain data that is dynamically-learned within an operational environment, then a similar strategy would likely be employed on a periodic basis. The rate at which learning could occur in this case would be limited by the data processing capability of the data processing system [C] being used to generate the metadata.

Let us now consider the speed of learning and the speed of knowledge exploitation. Learning need not be fast relative to the solution ultimately developed. We often envision whether an offline knowledge creation mechanism [C] could be employed that uses 10, 100, or 1000 processors to process a recorded datastream during a knowledgebase creation process. Even one processor [C] given enough time could create an extremely large and effective knowledgebase. The important point to consider is whether the knowledgebase created is large enough (and omniscient enough) in the problem space so that we can reasonably expect it to provide the answers we need to support a viable solution. Planning is important here. Of course, where some form of real-time learning is helpful to the development of an effective solution we must take into account the processing power available to a system [C] so that learning performance expectations can be clearly understood.

Whether the FFPRS knowledgebases [E2] that are created are associated with a static-learning or a dynamic-learning strategy, the result is the same. We will ultimately will acquire significant amounts of “knowledge” regarding a real-time problem area data stream and maintain this information within one or more knowledgebases [E2]. The information acquired allows us to associate these bits of knowledge with data patterns that have been previously observed. Later, as new data arrives from the problem area source [B] the data stream can then be inspected and processed in real-time by a real-time processing system [G] using the fast-search capabilities of the FFPRS search engine(s) [E1] to access one or more knowledgebases [E2]. We believe that the ability to exploit a very capable FFPRS [E] and the knowledgebases contained within it [E2] in real-time is a powerful solution enabler. Because by definition the types of FFPRS we envision is fast and scalable we are then **conceptually enabled** to develop real-time solution strategies useful to a wide variety of extremely challenging problem-areas.

Again, we do not see exhaustive-learning solution methods as a narrow “silver-bullet” type of solution method. Instead, we see the concept of exhaustive-learning as an enabler that is broad in scope and it allows us to look at extremely challenging problems in new ways so that a variety of unusual pattern recognition based methods can be explored to provide powerful real-time solutions. Problem areas that are associated with massive-data are certainly one type of problem for which exhaustive-learning can be beneficially applied. However, a variety of other potential application areas likely exist as well.

4 The importance of exhaustive-learning methods

Our exploration thus far of potential applications for exhaustive-learning methods has convinced us that these methods can be extremely useful when attempting to develop solutions to certain types of very challenging problems. Thus far we have been able to envision elegant solutions to problems that originally appeared to us to be either extremely difficult or impossible to solve using traditional computing methods. Although the exhaustive-learning methods described are not suitable for every type of problem one might encounter, thus far we have observed that exhaustive-learning methods provide a valuable new tool in the arsenal of ideas that can be used to assault certain types of extremely challenging massive-data problems.

5 Summary

This paper was generated for Program Managers, Scientists, and Engineers to provide a quick introduction to UNICON’s technology, our ideas related to massive scale pattern recognition, and the associated solution-enabling concept of exhaustive-learning. It is our hope that the discussion in this document will provide the average reader with a reasonable understanding regarding how **massive scale pattern recognition** and **the concept of exhaustive-learning** can be used to address a variety of very challenging technical problems associated with massive-data streams.