Lifelong Machine Learning with Logic, Semantics and Natural Language Processing

Dr. Emdad Khan
Dept. of Computer Science
MUM (Maharishi U. of Management)
Fairfield, Iowa, U.S.A
ekhan@mum.edu

Dr. Emdad Khan
InternetSpeech, Inc
San Jose, California, U.S.A
emdad@internetspeech.com

Abstract— **A fully capable Learning System would need to have most learning capabilities of a human – self learning, creating knowledge, learning from experience, determining what to be learned and the like, and called as Lifelong Machine Learning (LML). With impressive growth of use of Machine Learning (ML) and Artificial Intelligence (AI) in many applications, the need for LML is becoming more attractive. By creating knowledge and learning from previous knowledge or experience continuously across tasks and across domains, LML can help AI-ML growth further.**

In this paper, we focus on a Lifelong ML approach using logic, semantics and Natural Language, especially with Semantic Engine using Brain-Like Approach (SEBLA). Natural Language is a nice & effective way to represent *knowledge.* It is an effective way to accumulate knowledge, learn from experience, and derive new knowledge & intelligence, as we do as human. We also use a Neural Network and Fuzzy Logic combination (NeuFuz) to convert existing data driven ML systems knowledge into Fuzzy logic rules and natural / semi-natural language sentences to easily integrate with Natural Language based knowledge. We show how Lifelong machine learning is achieved using some application examples.

Keywords— *Lifelong Machine Learning, Machine Learning, Natural Language Computing, Cognitive Computing, Semantic Engine, Artificial Intelligence, Neural Net, Fuzzy Logic, Neural-Fuzzy Systems*

I. INTRODUCTION

We have seen an impressive growth of use of Machine Learning (ML) and Artificial Intelligence (AI) in many applications. It is important to briefly explain the hidden reason for such growth. We have many good algorithms for problems where data do not need to change the algorithm or procedure (of course, data change the result) – for example, finding minimum, maximum, or average of a set of numbers. However, for many applications data need to change the procedure itself or some parameters of it, or derive (learn) the relationship between inputs and outputs (i.e. discover the hidden dynamics of the system) – for example in Modeling or Regression, Classification and Clustering. How do we write algorithms that can result highly accurate Spam filter, identify a fraud transaction, recognize a picture, produce accurate speech recognition / handwriting recognition, or machine translation system? These are all good examples of applications where data need to change the algorithm itself directly or via some parameters, or data need to derive hidden dynamics of the system i.e. algorithms needs to be automatically created (e.g. learning algorithms to drive a car i.e. auto-driving of vehicles) **as we cannot write such algorithms that will work well.**

This means that we need to have a mechanism that can extract the meaning of the data i.e. can create a model that the data well represent, and accordingly solves very complex problems that cannot be solved otherwise. Traditional algorithms can address some of such problems, especially when data size is small or data type is not complex, but not usually with acceptable accuracy (e.g. a Rule based approach will have combinatorial explosion and rule management will be a nightmare; a traditional statistical approach may not have a good generalization capability). The key reason is the fact that the nature of the data has strong influence on the algorithm to yield accurate results, and human experience / capability is not adequate to model such complex data. The problem become much worse with very large data. As the data is growing very fast e.g. we generate 2 exa bytes (10^{18}) of multimedia complex data everyday (the so called **Big Data**) from emails, social networks, sensors and the like.

ML algorithms can successfully address this issue of reliable modeling of complex systems and automate the process yielding reasonably accurate results for many applications. This is the main reason for the rapid growth of AI and ML. Deep Learning (DL) and Transfer Learning are making ML more capable and effective in solving more complex problem (e.g. image and object recognition).

However, existing Machine Learning (ML) algorithms are dominated by isolated learning (e.g. in Supervised Learning, a specific dataset for a specific task in a domain is used to train an ML for regression or classification). After learning (via training) is completed, the trained ML system is used to make prediction i.e. no more learning is done. With online learning, the system can learn on a continuous basis and make better prediction. However, the learned information is not converted

ISBN: 1-601-32-501-O, CSREA PRESS ©

into knowledge and hence cannot be easily transferred for another task or application.

In other words, existing ML algorithms mainly do instance learning and does not support accumulative learning with knowledge creation [1]. The generalization capabilities of such systems are closely related to data, task and domain used to train, and hence are limited in scope (Transfer Learning can help to a good extent though for some applications). But such systems do not create knowledge and hence cannot learn from previous knowledge or experience across tasks and across domains.

However, recently there have been some good works that can help Lifelong machine learning i.e. can create knowledge from what was learned, use that knowledge to learn more and repeat the process like we do as human. But, such methods use algorithmic and statistical approaches for knowledge creation which do not scale up well and less flexible to model human-like learning.

It is important to note that existing ML approaches are great (and have been very successful in many applications) but they are data driven i.e. such algorithms cannot learn anything without a large dataset. On the other hand, humans do not learn using a large dataset. Unlike existing ML algorithms, we as human do not use a training dataset to understand / learn something. When we read a book or a teacher teaches us, we use semantics (meaning) and logic to understand the concept. A numerical example (s) can be used to explain the concept but it usually is not a necessity.

In this paper we emphasize on Lifelong Machine Learning (LML) using Natural Language based approach to further advance ML. Natural Language representation is something that we as human use very effectively in *accumulating knowledge, processing knowledge, deriving new knowledge (Cognitive Computing) and converting knowledge to intelligence*. Hence, we use Natural Language based Lifelong Learning. We also do learn many things by observation (e.g. looking into an object or scene) only. But we use Natural Language to explain or describe such visual based learning. The key in Natural Language is "Semantics" i.e. the "Meaning". Logic is another key to process meaning and create new knowledge, cognition and intelligence.

We believe applying Natural Language based human learning approach to LML is a logical approach that can make LML more effective and successful. Since existing ML is mainly driven by numerical / structured data based learning and the learned information is not represented in Natural Language, it is important to convert such learned information automatically into Natural Language / Natural Language Like rules so that such can be easily integrated to Natural Language based Lifelong Machine Learning framework.

We focus on a Lifelong ML approach using Logic, Semantics and Natural Language, especially with Semantic Engine using Brain-Like Approach (SEBLA). Natural Language is an effective way to accumulate knowledge, learn from experience, and derive new knowledge & intelligence, as we do as human. We also use a Neural Network and Fuzzy Logic combination (NeuFuz) to convert existing numerical / structured data driven ML systems knowledge into Fuzzy logic rules and natural / semi-natural language sentences to easily integrate with Natural Language based knowledge. We show how Lifelong machine learning is achieved using some application examples. Initial implementations show promising results.

The rest of the paper is organized as follows: Section II discusses "Importance and Issues With Data Driven Learning", Section III discusses "Logic, Semantics and Natural Language Driven Learning", Section IV discuses a "Lifelong Machine Learning System Driven By Natural Language Semantics" – it emphasizes on Learning & Knowledge Creation, how Knowledge, Cognition and Intelligence make decisions and take actions using a sample application, Section V discusses how to "Integrate numerical Data Driven Learning with Semantics & Logic Driven Learning", followed by Section VI - Conclusions.

II. IMPORTANCE AND ISSUES WITH DATA DRIVEN LEARNING

Data Driven approach is the main paradigm of all ML algorithms. All Supervised, Unsupervised, and Reinforcement Learning use many data to learn from data. Data must be enough, cleaned, of good quality and represent all possible corners of the data space (an important step called Data Preprocessing & Data Engineering). Appropriate features must be extracted from the input data (called Feature Engineering) to help the learning process of ML algorithms. Feature Engineering (FE) is one of the key complex steps in ML, especially when data is complex and there are many inputs. Deep Learning (DL) is helping FE process significantly by automating FE, especially, when processing images using "auto-encoding" & "convolution". But for many other applications, such approach does not work very well for FE.

ML algorithms use various techniques including Gradient Descent to minimize a loss function (e.g. Mean Squared Errors (MSE) in Back Propagation in Neural Nets, and Information Gain in Decision Tree).

While these are all good and very successful, there are multiple issues, especially from LML standpoint and when data size is very large. Some key issues are:

(a) Getting enough reliable data.
(b) Preparing the data
(c) Feature Engineering that would work well for learning
(d) Determining appropriate ML algorithms that would provide best "generalization" capability for a particular application domain.
(e) Avoiding overfitting
(f) Supporting on-line learning
(g) Dealing with highly unbalanced data set

ISBN: 1-601-32-501-O, CSREA PRESS ©

(i) Limited generalization capability
(h) Converting learned information into knowledge that can be accumulative (to support LML)

While there are various algorithms to address these key issues (mainly for a-i), addressing knowledge accumulation to support LML (h) is very difficult; in fact not possible today in general. However, we can use learned information of some tasks / domains into some other tasks / domains using today's Transfer Learning. Similarly, without having a wider generalization capability, its use for accumulating knowledge and transferring knowledge will remain very limited. In fact, these (a-h) are the key issues that strongly limit the capabilities of Data Driven Learning, especially to help LML.

Let's consider the Bias-Variance tradeoff (related to generalization). The equation for expected sum-of-square error is [2]

$$E = noise^2 + variance + bias^2 \qquad \ldots\ldots\ldots\ldots\ldots\ldots..(1)$$

The first of the three terms on the right side of the equation is beyond our control. It is the irreducible error and is the variance of the test data. The second term is variance, and the third term is the square of the bias. The variance tells us how much **x*** (a new data after a model is trained) changes depending on the particular training set that was used, while the bias tells us about the average error of $h(\mathbf{x}^*)$ where h is the hypothesis. It is possible to exchange bias and variance, so that we can have a model with low bias (meaning that on average the outputs are current), but high variance (meaning that the answers wibble around all over the place) or vice versa, but we can't make them both zero – for each model there is a tradeoff between them. However, for any particular model and dataset there is some reasonable set of parameters that will give the best results for the bias and variance together, and part of the challenge of model fitting is to find this point.

This means that for a particular dataset and model we can try to optimize the bias-variance and get some generalization. But the bias-variance tradeoff has a limited bound.

Thus, existing data driven ML approaches have 2 key issues from LML standpoint:

(a) Representing learned information (knowledge) in suitable form
(b) Limited generalization capability

It is clear that without converting a learned model into some good knowledge representation, accumulation and transfer of knowledge will remain difficult. Similarly, without having a wider generalization capability, its use for accumulating knowledge and transferring knowledge will remain limited.

Contrast this with human capabilities in learning, generalizing, accumulative learning and transferring knowledge across domain. *Human does it very easily mainly because of knowledge representation in natural language along with semantics & logic* (more in Section III).

However, as already stated, existing data driven ML are very important, essential and useful. So, we need to come up with ways to efficiently integrate such ML approaches with Natural Language based knowledge representation & processing to ensure a good LML system (more in Section V).

III. LOGIC, SEMANTICS AND NATURAL LANGUAGE DRIVEN LEARNING

The key capabilities, namely, classification, regression and clustering of existing numerical/structured data driven machine learning algorithms are not adequate for natural language processing, natural language understanding, natural language computing and cognitive computing. In [3], we have presented a new algorithm using new paradigm to better deal with ML for unstructured data. We call it MLANLP (Machine Leaning Algorithms for Natural Language Processing). An inherent property of any natural language computing and cognitive computing is the semantics. Machines need to compute semantics / meaning - by properly using the word semantics to compute the meaning of a sentence; and properly using the meaning of sentences to compute the meaning of a paragraph; and properly using the meaning of paragraphs to compute the meaning of the whole document. Machines also need to learn (and refine learning) semantics, especially the semantics of words. One common important element for all these is learning how to derive (i.e. learn) new semantics and new knowledge. This is more needed for cognitive computing.

Moreover, proper **actions** need to be learned or computed based on the meaning of a sentence or query. Thus, ML for unstructured data would need to learn logic, semantics and determine appropriate actions. Hence, the ML for unstructured data need to focus on semantics driven and logic driven learning as opposed to existing numerical data driven learning that usually minimizes error with respect to some objective function.

Semantics driven and logic driven learning paradigm mainly uses good explanation / logic rather than training using a large dataset. It is similar to the way human learn. We learn very easily when someone explains or teaches. A few examples may be used to enhance the teaching process. But our main part of the learning is based on explanation and logic. We do not use numerous data to learn something. In contrast, numerical data driven ML systems learn from many examples of numerical data and do not use explanation or logic to learn. This is the main reason such systems can only do limited key functions like regression, classification and clustering, and with limited generalization.

The generalization capability of semantics and logic based approach is much higher. To quantify this, we would need different mechanisms than used in equation (1) in Section II.

ISBN: 1-601-32-501-O, CSREA PRESS ©

For example, we can use various ratios and then combine those in some weighted form:

a. X1 = Intent detection ratio: total number of intents correctly identified / total number of sentences or intents
b. X2 = Action detection ratio: total number of actions correctly identified / total number of actions
c. X3 = Summarization ratio: total number of summarizations correctly done / total number of summarizations
d. X4 = Inference Ratio: total number of inference correctly done / total number of inferences
e. More ratios

NLP Generalization = W1.X1 + W2. X2 + W3.X3 +W4.X4 + …… Wn.Xn ………………………… (2)

where Wi is weight for Xi so that we have different weighting mechanisms for Xis.

A key component of MLANLP is a Semantic Engine, SEBLA (Semantic Engine using Brain-Like and Brain-Inspired Algorithms, [3], [4]). The main theme of our approach in SEBLA is to use each word as object with all important features and function which together represent the semantics. In our human natural language based communication, we understand the meaning of every word even when it is standalone i.e. without any context. Sometimes a word may have multiple meanings which get resolved with the context in a sentence. The next main theme is to use the semantics of each word to develop the semantics / meaning of a sentence as we do in our natural language understanding as human. Similarly, the semantics of sentences are used to derive the semantics or meaning of a paragraph. The 3rd main theme is to use natural semantics as opposed to existing "mechanical semantics" of Predicate logic or Ontology or the like.

Key features of a word are the features that basically defines the word including its functions i.e. what it does. For example, the key features of a ball are:
- something that is round,
- something that rolls *(one function it performs)*
- something that bounces when it hits a wall *(another function it performs)*

The color of the ball is a secondary feature as one can identify whether an object is a ball or not without using its color. So, a ball is represented in the Word Feature (WF) table as
{Move, roll, round, bounce back, play,..}, which is also its semantics.

At sentence level we consider semantics of each parsed word in a sentence. For each of such words,

a) Get the Function words (WF)
b) Get the World Knowledge (WK) words

E.g. for the word "ball", the function words are {ball, move, roll, round, play..} as mentioned above. Now consider a sentence like,

"I open door", the semantics of which is

I {person} open {open, unlock, push, pull,..} door {a thing that blocks, close, open, move,…} ……………………. (3)

MLANLP uses Natural Language and Logic driven approach, and hence, can accumulate knowledge (by using semantics of words, sentences, paragraphs, and documents), derive new knowledge and also determine actions. Thus, it is very critical for LML as described in next 2 Sections.

IV. A LIFELONG MACHINE LEARNING SYSYEM DRIVEN BY NATURAL LANGUAGE, SEMANTICS AND LOGIC (LMLS_NL_SEM_LOGIC)

Integrative learning or accumulative learning with existing approaches is very difficult and limited. Natural Language driven learning can handle these issues easily as such learning paradigm uses logic and semantics for learning and not much depends on the data. It also represents knowledge using Natural Language. As already mentioned, logic, semantics and Natural Language based Knowledge representation help much more robust generalization, accumulation of knowledge, integration of knowledge from other domains, and derivation of new knowledge & intelligence.

Fig. 1 shows Natural Language, semantics and logic driven LML architecture. User inputs (query, request, others; using text, voice & other methods) are processed by SEBLA, MLANLP & Logic, and (a) stored in the Natural Language driven knowledgebase (KB), and then (b) further processed by Q&A, Summarization, Inference and other Engines to create appropriate outputs to the

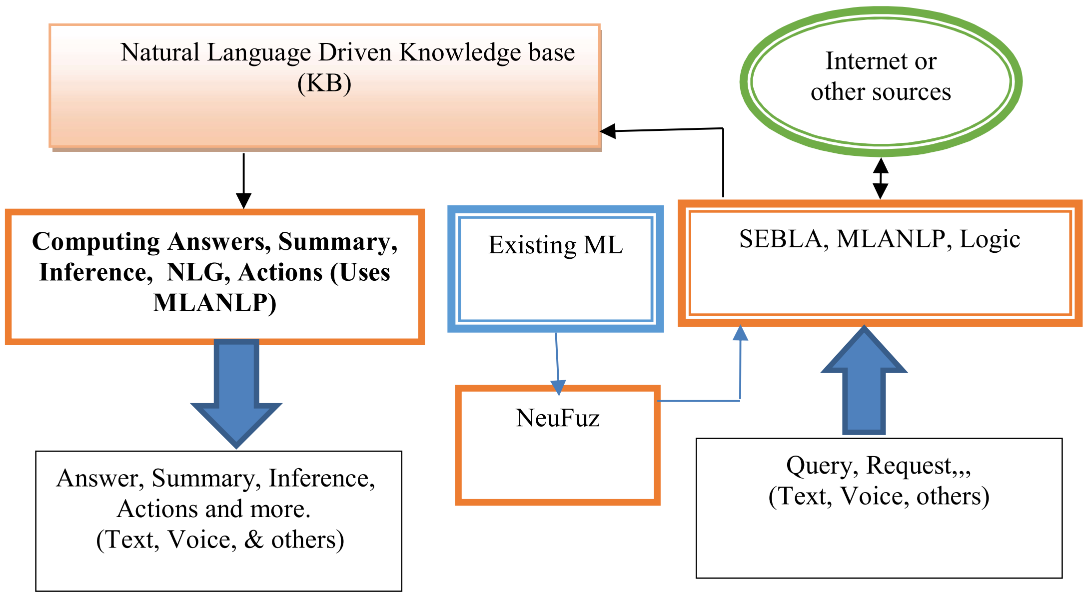

Fig. 1: A Natural Language Driven Lifelong Machine Learning (LML) Architecture (LMLS_NL_SEM_LOGIC). Uses NeuFuz (Neural Nets and Fuzzy Logic Combination) to integrate existing ML. Well suited for Advanced Analytics, Cognitive Computing and Intelligent Agent.

user. If some necessary information is not available in the KB, SEBLA will get the relevant information from the Internet (or similar sources), process it as appropriate and then add to KB. Existing ML Systems are integrated via NeuFuz (Neural Network & Fuzzy Logic Combination) where ML knowledge of a trained Neural Net is converted into Fuzzy Logic Rules which are then processed by SEBLA, MLANLP & Logic in the same way like other user inputs.

Let's use an example application to explain the concept. Consider 2 meeting minutes in a sequence to go for a vacation (Table 1).

Meeting 1: May 20; Attendees: Bob, Ron, Shelly & Sandy
Topics: Duration of the vacation, Countries to visit, Making a cost estimate [Start date June 21]

Minutes: 2 weeks vacation, Would like to visit China, and Australia, About $7,000 per person.
Some key conversations:
1. "We must see the Great Wall; and visit Beijing & Shanghai".
2. Bob: "Sounds good. But I prefer to stay longer in Shanghai".
3. "Let's talk about key places to visit for Australia once we are in China".

Meeting 2: June 29; In Shanghai. Attendees: Bob, Ron, Shelly & Sandy [Possible more meetings since May 20]
Topics: Plan change; How to accommodate conflicting interest

Minutes: "Sandy has an emergency and would need to return soon". Bob would like to skip Australia and thinking about going to Indonesia.
Some conversations:
1. Bob: – "I don't mind to go to Indonesia by myself"
2. "Shelly – I would prefer to go to Australia"
3. "Ron – I am not sure whether to stay longer in China or go to Australia with Shelly. Well, may be I have seen enough in China".

Inference: Where possibly all travelers will be on June 30?

Table 1: Two meeting minutes in a sequence to go for a vacation. Shows how LML integrates the knowledge and draw inference.

Semantics of each sentence in each meeting are computed using SEBLA (see Section III). Action Words of each sentence (mainly verbs) are used to determine the Intent and Actions for each sentence using the MLANLP (Section III). Total Actions, content for a Query, Summary or Inference from a few paragraphs / docs are computed using various blocks as appropriate (Fig 1). So, the KB for meeting 1 and meeting 2 are accumulated easily and placed in the KB. The World Knowledge (WK, Common sense computation & inference) is also included in KB (e.g. 2 weeks vacation for 2 countries means one week for each country as nothing specifically is mentioned otherwise). KB can also create or derive new knowledge (using MLANLP & Logic).

Using the KB generated for the 2 meetings, the following can be inferred easily by LMLS_NL_SEM_LOGIC :

On June 30, Sandy will be in San Francisco; Shelly will be in Australia, Bob will be in Indonesia and Ron will most possibly in Australia. The inference for the Ron is the most difficult one. But the conversation #3 in meeting 2, implies with some confidence that Ron has seen China enough. In case more information is given, a better inference could be made. This is just a simple example to explain the concept. Of course, for complex cases, the system may fail sometime to give correct answers. But, in general, it will still give something reasonable. KB will usually have hierarchy and various sub-domains (like for History, Economics, Science etc.) so that it is manageable, easily searchable and tries to help avoid combinatorial explosion when integrated with the existing ML system (Section V).

V. INTEGRSATING EXISTING ML SYSTEM WITH LML SYSTEM DRIVEN BY NATURAL LANGUAGE, SEMANTICS AND LOGIC (LMLS_NL_SEM_LOGIC)

It is very important to integrate existing ML system with LML system so that we have a more complete, capable and efficient LML system, like LMLS_NL_SEM_LOGIC. A novel way to do so is to use a Neural-Fuzzy system where a Neural Network is trained and then its knowledge (the weights) is automatically converted to Fuzzy Logic Rules and Membership Functions. Since Fuzzy Logic Rules are similar to Natural Language, such Rules can be easily integrated with LMLS_NL_SEM_LOGIC discussed in Section IV.

NeuFuz [9, 10, 12] is such a system where Artificial Neural Net (ANN) algorithms are used to generate fuzzy logic rules and membership functions. The combination of learned fuzzy rules, membership functions, and a fuzzy design technique based on new fuzzy inferencing and defuzzification methods significantly improves performance, accuracy, and reliability and reduces design time. NeuFuz also minimizes system cost by optimizing the number of rules and membership functions. Fig. 1 shows how NeuFuz is integrated with LMLS_NL_SEM_LOGIC.

The neural net is properly architected so that it maps well to fuzzy logic rules and membership functions (Fig. 2). The 1st layer neurons in Fig.2 include the fuzzification process whose task is to match the values of the input variables against the labels used in the fuzzy control rule. The 1st layer neurons and the weights between layer 1 and layer 2 are also used to define the input membership functions. In fact, it is difficult to do both fuzzification and learning membership functions just by one layer of neurons.

Fig. 3 shows a multiple layer implementation for fuzzification and membership function generation. Both linear (L) and non linear (NL) neurons are used. With an input level of x, the output of layer 1 neuron is gl.x where gl is the gain of neuron in layer 1. The input of layer 2 neuron is gl.x.W1. Continuing this way, we have the input of layer 4 neuron, z as

$$z = (g1.x.W1.W2.g2 + b).w3 \quad \text{...} \quad (4)$$
$$= (a.x + b).c$$

where a = gl.g2.WI.W2, c = W3, g2 = gain of layer 2 neuron.

The output of layer 4 is the membership function which is same as the output of layer 1 in Fig 2. Thus, NeuFuz uses a 6 layer neural net [9]. Neurons in Layers 1-4 correspond to the

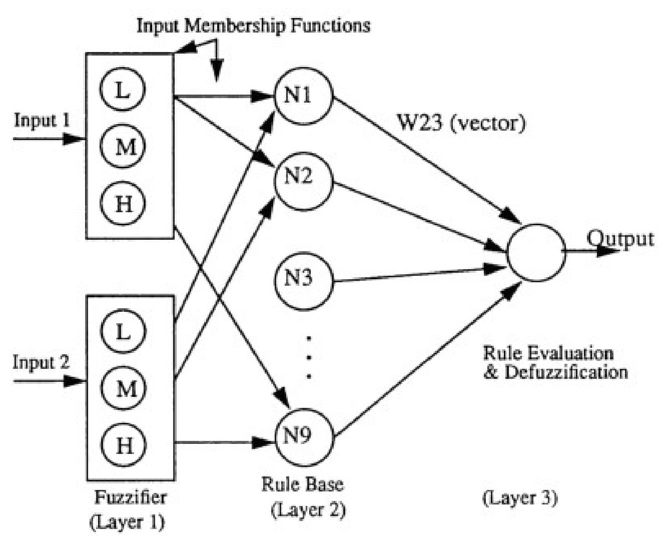

Fig. 2: Neural Network and corresponding Fuzzy Logic representation in simplified form. The net is first trained with system input-output data. Learning takes place by appropriately changing the weights between the layers. After learning is completed, the final weights represent the fuzzy rules and membership functions. The learned neural net, as shown above, can generate output very close to the desired outputs. Equivalent fuzzy design can be obtained by using generated fuzzy rules and membership functions.

membership functions. Neurons in layer 5 (Layer 2 in Fig. 2) correspond to fuzzy logic rules.

A sample Fuzzy Logic Rule is (equation 5):

"If Input 1 is Low and Input 2 is Low THEN the output is W23" ……………………………………………………….(5)

where W23 is the weight between layers 2 and 3 in Fig. 2 The neuron in layer 3 does the rule evaluation and defuzzification.

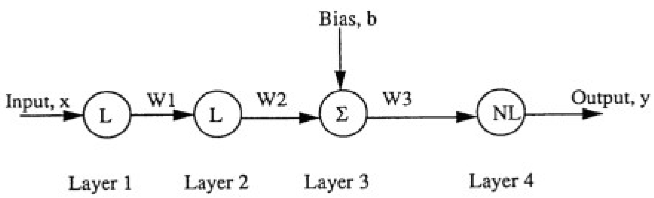

Fig. 3 Neural network architecture to learn Fuzzy membership functions.

The Neural Fuzzy approach typically uses nonlinear membership functions (as shown in Fig. 2 & 3). The advantage of using a nonlinear membership function is that the system knowledge can be distributed evenly between the rule base and the membership function base. This results in a reduced rule base, help prevent combinatorial explosion, saves memory and overall cost.

Most importantly, a Neural Fuzzy System's learning and generalization capabilities allow generated rules and membership functions to provide more reliable and accurate solutions than alternative methods. In conventional approaches, one writes rules and draws membership functions, then adjusts them to improve the accuracy using trial-and-error methods. However, with the proper combination of fuzzy logic and neural networks (such as NeuFuz), it is possible to completely map (100%) the neural net knowledge to fuzzy logic. This enables users to generate fuzzy logic solutions that meet a pre-specified accuracy of outputs. This is possible because the neural net is able to learn to a pre-specified accuracy, especially for the training set (the accuracy for the test set can be controlled to be very close to the accuracy of the training set by properly manipulating the learning parameters), and learned knowledge can be fully mapped to fuzzy logic. Full mapping of the neural net to the fuzzy logic is possible when the fuzzy logic algorithms are all based on the neural net architecture. Such an elegant feature is not possible in conventional fuzzy logic, in that one cannot write fuzzy rules and membership functions to meet a pre-specified accuracy. NeuFuz capabilities and performance can further be improved using Recurrent Fuzzy Logic [11, 12].

VI. CONCLUSIONS

We have presented a Lifelong Machine Learning (LML) system using a Natural Language, Semantics and Logic driven approach (LMLS_NL_SEM_LOGIC). Natural Language is an effective way to represent knowledge, accumulate knowledge, learn from experience, and derive new knowledge & intelligence, as we do as human. LMLS_NL_SEM_LOGIC achieves Lifelong learning and create appropriate actions using SEBLA (Semantic Engine using Brain-Like Approach) and MLANLP (Machine Learning Algorithms for Unstructured data). Initial results are encouraging.

Existing numerical data driven ML systems are very important and have been very successful. But such systems are based on isolated learning, and LML is not supported, although Transfer-Learning and On-line learning help for certain applications. We have presented an elegant solution to integrate LML with existing ML using a combination of Neural Nets and Fuzzy Logic (NeuFuz), thus enabling Lifelong learning for existing ML systems in a general way.

REFERENCES

[1] Z. Chen et al, Lifelong Machine Learning, Morgan & Claypool publishers, November 2016.

[2] S. Marsland, Machine Learning: An Algorithmic Perspective, 2nd Edition, 2015, CRC Press

[3] E. KHAN, MACHINE LEARNING ALGORITHMS FOR NATURAL Language Semantics and Cognitive Computing", The 2016 International Conference on Computational Science and Computational Intelligence (CSCI'16), Dec 15-17, 2016, Las Vegas.

[4] E. Khan,: Natural Language Understanding Using Brain-Like Approach: Word Objects and Word Semantics Based Approaches help Sentence Level Understanding, US Patent files 2011–US Patent Publication 2014/0032574A1

[5] E. Khan, "Natural Language Processing: Key for Next Generation Big Data and Data Science", Plenary talk *at 15th International Conference on Applied Computer and Applied Computational Science, March 18-20, 2016* http://www.wseas.org/wseas/cms.action?id=12693

[6] E. Khan, "Next generation web - intelligent search, question answering, summarization and more", INTERNATIONAL JOURNAL of COMPUTERS AND COMMUNICATIONS, (NAUN & UNIVERSITY PRESS), Vol. 9, June 2015.

[7] E. Khan, " Intelligent Internet: Natural Language and Question & Answer based Interaction", International Journal of Computers and Communications, (NAUN & UNIVERSITY PRESS) Oct. 2013.

[8] E. Khan, "Natural Language Computing: A Necessity for Cognitive Computing and Intelligent Internet", Plenary Talk, 17th International Conference on AI, Knowledge Engineering & Data Bases, Feb 25, 2017, Cambridge University, U.K.

[9] Lakhmi Jain; etal, Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications by N.M. Martin CRC Press, CRC Press LLC ISBN: 0849398045 Pub Date: 11/01/98

[10] E. Khan, Neural Networks Apparatus and Method for Pattern Recognition, U.S Patent # 5058180, Oct 15, 1991.

[11] E. Khan, Neural Network Based Fuzzy Logic Systems & Methods, U.S Patent # 8828812, Oct 27, 1998.

[12] E.Khan, "NeuFuz: An Intelligence Combination of Fuzzy Logic, with Neural Nets", IEEE, IJCNN, 1992, Japan.

ISBN: 1-601-32-501-O, CSREA PRESS ©