Perspectives on using deep learning for system health management

Samir Khan and Takehisa Yairi

Department of Aeronautics and Astronautics, University of Tokyo,

7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

khan@ailab.t.u-tokyo.ac.jp
yairi@ailab.t.u-tokyo.ac.jp

ABSTRACT

Having a robust health management and diagnostic strategy is an important part of a system's operational life cycle as it can be used to detect anomalies, analyze faults/failures and predict the remaining useful life of components. By utilizing condition data and on-site feedback, data models can be trained using machine learning and statistical concepts. Once trained, the logic for data processing can be embedded on on-board controllers whilst enabling real-time health assessment and analysis. More recently, deep learning has gained increasing attention due to its potential advantages with data classification and feature extraction problems. It is an evolving research area and hence its use for aerospace maintenance applications must be researched if it can be used to increase overall system resilience or potential cost benefits for maintenance, repair, and overhaul activities. This paper focuses on investigating the application of deep learning for system health management, therefore incorporating reliable redundancy at the adequate point in the system. Deep learning is discussed, some recent developments are reviewed to clarify potential applications, after which some research issues relating to their realization are highlighted.

1. INTRODUCTION

Health management can be described as the process of diagnosing and preventing system failures, whilst predicting the reliability and remaining useful lifetime (RUL) of its components (Sin and Jun 2015). The past few decades have experienced a proliferation of system health management research to help with all kinds of faults occurring at component level and up to the systems level (Huynh et al 2015). However, even though these concepts have been studied extensively (see Zhang and Jiang 2008, Patton 2013), most methods often require triggering mechanisms that are intelligent enough to collect enough data about the failing component, the nature of the fault, and its severity on the overall system performance. Whilst these technologies are typically focused on fault detection and isolation within individual subsystems, the growing maintenance costs facing today's engineering industry has prompted further research for novel architectures to reduce maintenance, repair and overhaul of complex high value assets. As a consequence, efforts are being concentrated on the integration of anomaly, diagnostic and prognostic technologies across systems and related platforms. Such capability to predict and isolate impending faults and failures can help maintain system performance in a cost-effective manner, whilst identifying ongoing issues to mitigate potential risks; and hence data exchange within diagnostic technologies has become a high priority research topic.

As the aerospace industry continuously strives to improve its performance, by delivering more reliable assets, with a higher availability; operational pressures reduce the time available for diagnostic investigations. Here, there is value of having many data collection sources that can be used to provide rich information (e.g. operating variables, environmental conditions, etc.) if a disruption occurs during operation (Russell and Benner 2010). However, most often data sources are disparate. With the ever-increasing size of data produced by modern systems, coupled with the complexities of contextual components (for correlating information); it can create barriers that were not anticipated by design engineers during the design phase of the system life cycle. This can also result in speculative replacements and higher levels of uncertainty during the diagnosis process (Khan et al, 2014). Due to the higher levels of interdependencies between assets, it becomes difficult (if not impossible) to assess as to why the failures appeared. Yet, the underlying engineering environment is expected to support the technological platforms as well as system availability requirements. In this context, novel approaches are required which can recognize and configure anomalies during operations; as well as mechanisms for making better decisions at the system-level. In the nominal environment, such problems require advanced capabilities to monitor in-service operations, record and share expert knowledge, and address critical aspects of on-board software. Generally, a system that can resolve issues autonomously can result in significant reduction of operational costs and increase in operational uptime, as the asset will not be taken out for corrective maintenance (Eti et al 2006). These would rather become evidence-based scheduled maintenance tasks which will reduce inspection costs, the required number of skilled labors, system down time, life-cycle cost of the system and emergency unscheduled maintenance. But effective identification of system faults (including the ones that have already occurred and that which are approaching) can still present a challenge. This can be due to various factors such as having a single diagnostic procedure for identification and isolation of any type of fault, insensitivity to operating conditions and unable to achieve reliable fault detection in time-varying conditions (Patton et al 2013). Of course, these factors are not inclusive, but they do indicate the two benefits that can be achieved by addressing them. Firstly, there will be improvement in the safety aspects. If failures can be diagnosed and assessed quickly, the remaining life of safety critical components can be regulated before they can cause any serious damage during operational service. Secondly, ongoing maintenance costs could be reduced with an increase in system availability. These two driving forces inevitably influence operational performance and the amount of maintenance required; a consequence of the design and the problems encountered during service. A direct result of such factors has contributed to the development of new technologies and techniques – which tends to add additional layers of complexities. Practitioners can gain general knowledge on these topics from several resources e.g. training, books, etc., but because of the availability of so many different fault analysis algorithms and health monitoring solutions, it can become difficult to rationalize this information. This leaves a gap as the number of solutions available, that require making choices that engineers might not be equipped to make. This paper attempts to discuss this issue.

Within the supervision of equipment, diagnostic systems based on conventional computing techniques are being replaced by Artificial Intelligence (AI) based ones which can increase efficiency of the monitoring technology. AI-based approaches can be categorized into (1) knowledge-driven (knowledge-based) approaches including expert system and qualitative-reasoning, and (2) data-driven approaches including statistical process control (SPC), machine learning approach and neural networks – including deep learning (Hinton et al 2012). Some approaches, such as probabilistic reasoning (e.g., Bayesian networks), may belong to both categories, because reasoning and learning cannot be distinguished. Yet, many of them are dependent on obtaining accurate (and sometimes complete) data on system models. Figure 1 illustrates some of the common AI approaches.

In general, the statistical characteristics of a system does not change until there is a fault; hence AI can effectively be used to carry out fault analysis and to make system level decisions based on information collected from a combination of sources[1]. Some of the more notable advancements have been captured in Table 1.

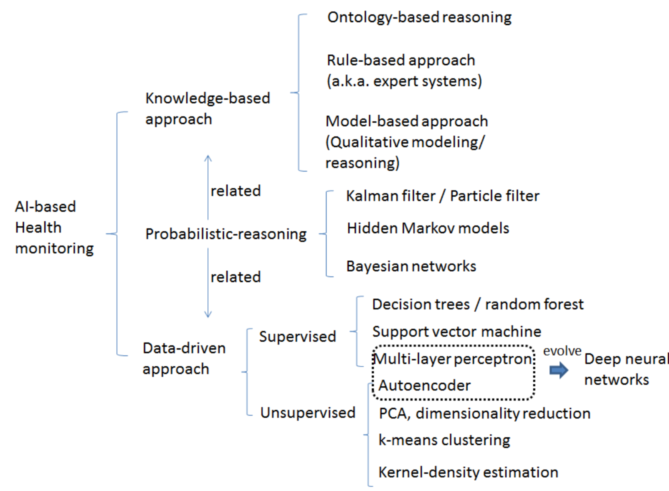

Figure 1: Categorization of AI based methods

Table 1: The use of various learning methods utilized in health management applications.

Technique	References
Parametric statistical modelling	Guttormsson et al. (1999), Keogh et al. (2006)
Nonparametric statistical modelling	Beyca et al. (2016), Desforges et al. (1998)
Neural networks	Purarjomandlangrudi et al. (2014), Bishop (1994), Campbell and Bennett (2001), Diaz and Hollmen (2002), Harries (1993), Jakubek and Strasser (2002), Li et al. (2002)
Spectral	Shui et al (2015), Fujimaki et al (2005)
Rule-based systems	Wang et al. (2015), Yairi et al (2001,2005,2006)
Deep learning	Jia et al (2016), dean et al (2016), Lv et al (2016)

The ever-increasing interest in using AI is a consequence of improved hardware performance and its reducing costs (Meziane 2009). Hence, applications on expert systems, fuzzy systems and Neural Networks (NN) have progressively developed over the past three decades. During this period, the discipline of software engineering had been promoted to accommodate the development of interacting systems. From a health management point-of-view, this means that systems

[1] Here, the use of the term AI incorporates various techniques such as expert systems, neural networks, support vector machine, fuzzy logic, and fuzzy-neural networks, that can be used autonomously or into each other to improve their efficiency and effectiveness.

have become more intricate and complicated, and any diagnostic analysis or prediction may require an even more rigorous procedure to elicit reliable and dependable system models. Yet, these methods still hold many benefits as compared to the conventional diagnostic approaches. One seemingly important development within AI community in the past few years, that has gained a lot of attention in the community, is deep learning. To the best of the authors' knowledge, deep learning methods have limited published material within the maintenance engineering community, let alone on health management, and hence is the motivation for this publication. Deep learning can be used for processing and analyzing large amounts of data. Depending on the application, various parameters such as vibration profiles, acoustic estimations, temperature, stress, oil analysis, etc. can be used to develop a history of the system on which qualitative and quantitative methods can possibly be applied for health management. Therefore, the authors hope to review and summarize some of the research potential for maintenance, reliability, and operations (MRO) applications, that maintenance practitioners might find useful.

2. TYPICAL LEARNING VS DEEP LEARNING

Neural networks have extensively been researched and applied to real world systems (as seen in Table 1). It is a network of interconnected nodes which are typically organized in layers – as input output and hidden layers where the processing is done via a system of weighted connections. Most neural networks contain some form of a learning rule that can change these weights according to the input that it is presented with[2]. In theory, a neural network replicates the human brain structure and consist of simple arithmetic functions that form an interconnected (and complex) architecture. They can represent highly nonlinear functions to carry out multi-input multi-output mapping by exposing the network to a predefined set of examples, observing the network, and adapting the values to reduce any differences. For this purpose, many methods can be applied to essentially train these values (or network behavior) that including a range of gradient based and optimization techniques. Neural networks can be organized in layers of non-linear transformations. When the number of layers becomes comparatively large, such an architecture can be called as deep neural network. In theory, a neural network with more than two layers i.e. input and output, can be classified as a deep architecture, however it is not just about the number of layers, but rather the idea of automated construction of more complex features on every step. This means that stacking other algorithms (such as a random forest) several times, use probabilities instead of class labels, and this can be considered as deep learning too. Back-propagation (which has existed for decades) theoretically allows to train a network with many layers. But before the advent of deep learning, researchers did not have widespread success training neural networks with more than 2 layers. This was largely because of the problem of vanishing and/or exploding gradients (Bengio et al, 1994). Prior to deep learning, the network was typically initialized using random numbers, and used the gradient of the network's weights with respect to the network's error. This helped to adjust the weights to better values in each training iteration. But with back propagation, evaluating the gradient involves using the chain rule and the need to multiply each layer's weight and gradients together across all the layers. This resulted in a lot of multiplications, especially for networks with more than 2 layers. If most of the weights across (many) layers are less than 1 and they are multiplied many times, then eventually the gradient just vanishes into a machine-zero and training stops. On the other hand, if most of the weights across many layers are greater than 1 and they are multiplied many times, then eventually the gradient explodes into a huge number and the training process becomes intractable.

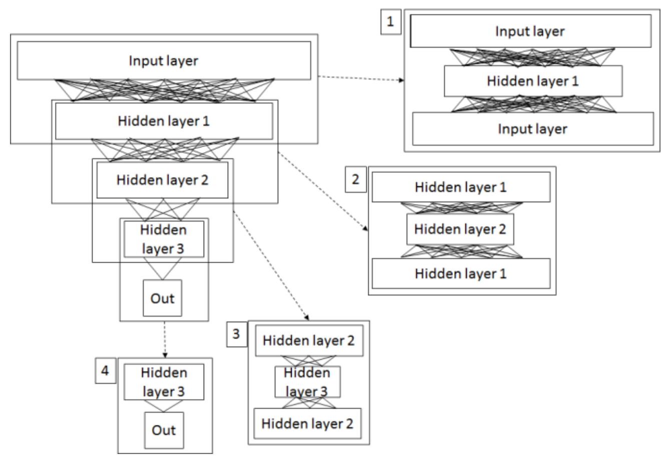

Figure 2: Adjusting the weights for each layer

Deep learning proposed a new initialization strategy: use a series of single layer networks - which do not suffer from vanishing/exploding gradients; to find the initial parameters. Figure 2 attempts to illustrate this process.

1. A single layer is used to find initial parameters for the first hidden layer. The approach uses the input data/vector to predict itself. By doing this, the layer learns something intrinsic about the data without the help of an output or label vector – that is often created by the human operator. The learned information is stored as the weights of the network for that layer.
2. The next layer uses the output from the first hidden layer to find initial weights for the second layer.
3. The process is repeated for the rest of the layers.

[2] The backward propagation of errors or backpropagation, is a common method of training thee weights in a network and is often used in conjunction with an optimization method such as gradient descent.

4. Finally, a softmax classifier[3] (logistic regression) is used to find initial parameters for the output layer.

Now that all the layers have been initialized through this pre-training process to values that are more suitable for the data, the network can now be trained using gradient descent techniques without the problem of vanishing/exploding gradients. Of course, the field has moved forward since this initial breakthrough, and many practitioners now argue that pre-training is not always necessary[4]. But even without pre-training, reliably training a deep network requires some additional sophistication, either in the initialization or training process beyond the older training approaches of random initialization followed by standard gradient descent.

How many layers? Deep learning works because of the architecture of the network, but more importantly, the optimization routine applied to that architecture. As can be noted in Figure 2, each hidden layer is connected to many other hidden layers within the overall network. When an optimization routine is applied to the network, each hidden layer can become an optimally weighted, non-linear combination of the layers below it. As the size of each sequential hidden layer keeps decreasing, each hidden layer becomes a lower dimensional projection of the previous one. So, the information from each layer is being summarized in each subsequent layer of the deep network by a non-linear, optimally weighted, and lower dimensional projection. None-the-less, the training process can be a challenging and lengthy task when the network has many layers and multiple connections between layers and neurons; but nowadays many researchers are implementing this training phase in Graphical Processing Units (GPUs) to leverage the power of parallel processing and reduce training time. However, once trained, classifying information becomes straightforward and fast to complete. Several existing algorithms have been extended for deep learning. These include supervised ones like logistic regression, multilayer perceptron, and deep convolutional networks. To be reliable, these methods would require information – labelled data – of all probable anomalies and the nominal states. Collecting such data for a dynamic system, such as an aircraft engine, may be difficult to realize, as intermittent faults are rare and heavily depend on the environmental conditions (Khan et al, 2014). In contrast, unsupervised learning algorithms aim to learn from unlabeled data and generate a model. A summary of all these algorithms has been summarized in Figure 3.

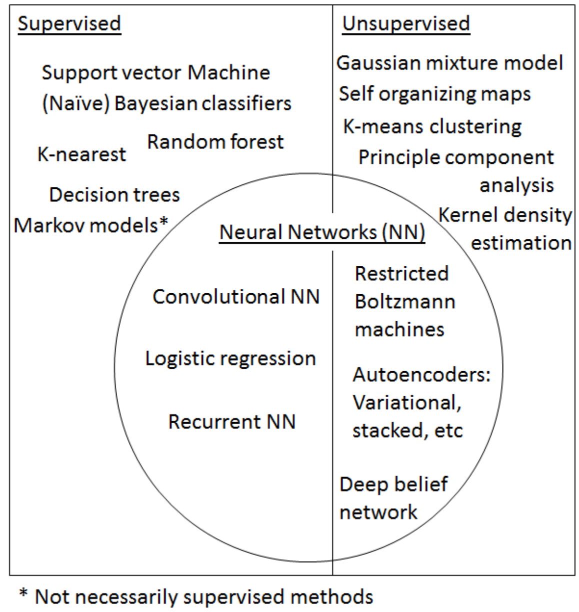

Figure 3: Techniques to be considered for deep learning

It should be noted that different learning algorithms for deep architectures will have different characteristics. E.g. Stacked Auto-Encoders (SAE) and Deep Belief Networks (DBNs) are unsupervised learning algorithms and hence they learn a model of the input distribution from which one can generate samples. They can also can be seen as unsupervised feature learning algorithms and hence be used to pre-train (from labelled or unlabeled data) features. These features can then be used as initialization for a supervised neural network. There are many of other unsupervised representation learning algorithms and yet there has been only few quantitative comparisons between them. It seems that DBNs and SAEs behave very similarly in terms of quality of the features learned (Litjens et al 2017).

3. APPLICATION FOR SYSTEM HEALTH MANAGEMENT

Enabling health management requires three constituents: fault detection, fault classification and fault prediction. Also, different types of faults (such as intermittent, incipient, degradation, etc.) will need to be classified in to various categories. This can be achieved by considering this as a binary classification problem, where either there is a fault or no. Once a fault has been detected, it can be further analyzed for other features. In the past, McDuff et al (1989) had used neural networks on F16 flight line data for diagnostics

[3] Softmax is a generalization of logistic regression that can be used for multi-class. In contrast, the standard logistic regression can be used for binary classification tasks.

[4] Pre-training helps to achieve two goals: further optimize layers and reduce overfitting. But if the initialization of weights is done correctly, pre-training is not always needed. This is because pretraining will require many training samples and a lengthy training window (the first few layer will change slowly). This reduces the usefulness of the approach.

purposes. The authors acknowledged its capabilities to carry out multiple fault diagnosis, predictions and reconfigurations, and the ability to work with inaccurate or incomplete rules. Adaptive resonant theory was used to train the data due to its ability to learn faster than other methods. Since different fault scenarios can be used to verify the efficacy of an approach, researchers found interest in integrating different diagnostic methods and developing a hybrid approach for this purpose. Volponi et al (2003) had used Kalman filters and neural networks methodologies to find the malfunction and deviations from the normal engine behavior. Another possible combination is the integration of neural networks is with the genetic algorithm (GA) method. In Kobayashi and Simon (2005), a neural network part of the scheme is used for engine components fault diagnostic and the GA is applied for sensor bias detection and estimation. By integration the two methods one can take benefit from each method's advantage. Neural networks enable nonlinear estimation and GA methods bring more robustness. The results helped in terms of improved fault detection and reduced false alarms.

More recently, deep learning methods were introduced to looked at a fault diagnosis and *learn* the deep architectures of fault data (Lv et al 2016). The research makes use of stacked autoencoders to improve the network's learning capability and demonstrates the potential of deep learning. The authors appreciated that, unlike image data, fault characteristics can vary over time making it difficult to classify and to develop a deep learning architecture. Clearly, after the diffusion of NN within AI applications, practitioners started adopting using them to carry out diagnosis to make system health management decisions where the classical "if–then and do" commands can be utilized to carry out most complex actions[5]. The major advantage is of retrieving and processing signals; and the knowledge-base which contains all the possible architectures corresponding to the considered fault modes. This knowledge-base can be used as an account for various attributes required for learning, and hence can compute and store tables or curves of diagnostic indexes for different faults, whilst working in different operating conditions. This can also archive heuristic rules and expert knowledge [6] coming from on field experience in order to overcome some of the limits of incomplete system models. These functions have been applied extensively for condition monitoring and fault diagnosis (Nelles, 2013). For health management activities, in particular, neural networks are often employed as statistical modelling and prediction algorithms. This can be treated as either a density estimation and prediction problem or as a classification and regression problems (Sutharssan et al, 2015).

Furthermore, Lee et al (2016) have investigated the use of convolutional neural networks for analyzing acoustic signals in the midst of noise. The authors were motivated by the fact that most existing signal analysis methods are largely dependent on the physical behavior/characteristics of the system being analyzed, which warrants regular re-tuning of algorithms for new acoustic profiles. Although, training for a deep learning system can be slow; but in testing (running) time these systems are usually quite fast when run on GPUs. Traditional methods can be much slower than deep learning methods during test time. It should also be noted only very recently organizations have been focusing on optimizing neural based computations, and in the near future are expected to see silicon chips that are designed especially for these systems. Another strategy to reducing training time is to precondition the input data to extract features that are not required.

But regardless of which technique is used, its real-time implementation and related theoretical formulations must be transformed into an algorithm. There are predominately three issues in this respect and they occur in various forms: these are data sampling considerations, the size of data and the implementation architecture. In practice, these issues are not solely associated with AI implementation, but for real-time system in general. None-the-less they are linked and therefore, it is important to understand the nature of the problem. Firstly, many health management design engineers lack the knowledge on develop deep learning architectures. This issue seems to be more prevalent in academia as compared to industry. Libraries and packages found in popular mathematical software's, such Matlab and R for model development and analysis, seem to be outdated for deep learning purposes as the ease of implementing some functionalities (such as Rectified Linear Unit (ReLU) activation, optimizers such as rmsprop, adagrad, and batch normalization), makes it difficult to learn and realise effective deep nets. Matlab has excellent support for traditional time series modelling e.g., in its signal processing packages, but for working with deep recurrent neural networks such as Long Short-Term Memory (LSTM) or Gated Recurrent unit (GRU), tensorflow is the much better option (Abadi et al, 2016). In addition, other packages like keras or lasagne[7] can be used to build upon tensorflow to allow for even simpler, optimizing and designing of network structures, e.g., ones involving novel networks that have embedding, feed forward, recurrent all together.

[5] such as programs execution, file management, etc.

[6] The symptom to fault map can be stored to identify the healthy state of a system

[7] Keras and Lasagne provide high-level functionality to enable deep learning algorithms e.g. pooling, backpropagation and optimization routines. Theano provides the back-end capabilities for computation (Barranco et al, 2016).

Another major issue is associated with the cost of such systems. The development of a complete health management system that is integrated in to the entire system architecture can be expensive, especially as more data will be required to train a network for varying fault characteristics. Using artificial fault data to train is not ideal and can result in incorrect performance attributes of real world applications. For example, in an aircraft engine it is vital to detect the system deviations from its nominal behavior as early as possible (Khan et al, 2014). Anomalous operation of the engine can affect undesirably both the engine function and the aircraft mission management. Perhaps, if the mathematical model of the system in addition to different fault models and their progression are available, model-based approaches are more accurate and effective. However, that is a research field in its own right as it can be difficult to find the explicit mathematical models due to system complexity and uncertainties. As a result, the accuracy of the results will decrease. Most of the recent success of deep learning have been in applications of supervised learning in computer vision and natural language processing with deep (convolutional, recurrent) neural networks. But deep learning techniques have been proposed for anomaly/novelty detection in time series data. These also include both recurrent neural networks and standard networks (Şeker et al, 2003). These models have been used for anomaly detection in engines, power demand, network (failures/intrusions), and novelty detection in music, etc. Deep learning has also been applied to time series modelling (Busseti et al, 2012). However, it should be mentioned that the benefit of using deep nets is still subjective to its application and discipline, e.g., it may return limited benefits when used for time series prediction as compared to its application in pattern recognition. Perhaps a greater CPU energy consumption does not justify its use.

Of course, the problem of complexity associated with its implementation is still an issue. There is a need to place some limits on it and perhaps aim to achieve a reasonable system performance. Here, the long-term goal is to appreciate the limitations that real-time AI would bring and then attempt to simplify the problem. Understanding these trade-offs is an on-going motivation of this work. Complexity limitations for real time systems could be categorized in terms of limitations in the structure of the computation, selecting criterions for acceptable solutions, the cost of reasoning and task predictability:

1. Structure of the computation: To limit complexity, a limit needs to be imposed on how many layers are required for the deep learning architecture. This will dictate the number of computations involved.
2. Selecting criterions for acceptable solutions: The assumption here is to satisfy a requirement rather than to find an optimal solution. Theses 'satisfying' solutions often take the form of using heuristic problem solving technique and is only useful if non-optimal solutions exist for a problem.
3. The cost of reasoning: A good way to describe this to consider the cost of control against the cost of safety. Within safety critical applications, this assumption will be difficult to justify unless the cost of control is predictable, e.g., do the computations result in feedback delays which can affect system stability?
4. Task predictability: Like all learning algorithms, deep architectures needs to make accurate predictions to help avoid excessive runtime costs.

4. DISCUSSIONS

The following fault tolerance problem is currently being investigated for using deep learning in health management. In this case, the solution needs to warrant a controlled system performance under varying conditions – indicating a deterministic system response. Any loss of information can be critical to the stability of the system; on the other hand, any delay in processing information can result in an inaccurate control response. Figure 4 depicts a control scheme consisting of a system, two controllers and a neural network based fault analyzer. A number of sensors (in this case x_0, x_1, x_2) are used for feedback. The notion is to observe the sensor outputs (in the midst of noise) and reach a conclusion with regards to faults. This is done by verifying if the system output data is within the specified tolerances and would require collecting data and training the net off-line. During the analysis, the net needs to distinguish if the fault is from failed sensors, the system response, or a result of system degradation – the authors expect that this is where the deep architecture will be able to provide the granularity required to make the distinction. This can be built to allow compensating the control response online during operation. This can also aid in substituting sensor output estimates in case of sensor failures (or partially destroyed/missing readings). In case of a sensor failure, data from other sensors can still help with a good estimation to replace the failed sensor and virtually provide an input until a maintenance action is carried out. However, this problem would become more complex as there are stability and phase lag requirements that should be met during implementation.

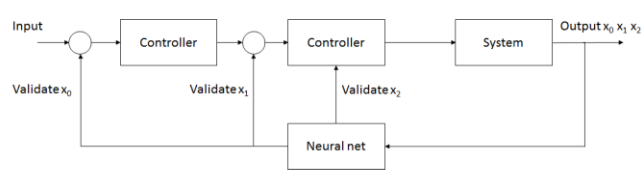

Figure 4: The closed loop setup

4.1. The gap between industry and academia

Current research work on deep learning is being led by industry, whereas academia has been slow to adopt it. Despite the fact there was already a lot of research work on deep networks in the past – it seems that the field of application hindered it. The following are some academic challenges and subjective views which have been stated by researchers and practitioners alike, with regards to today's disparity in deep learning research activities between industry and academia:

1. Funding calls may influence current academic research directions and having to spend time write grants, sit on committees, etc. Currently funding bodies are paying scant attention to deep learning techniques for maintenance applications, which currently warrants some fundamental work – perhaps even below technology readiness level 4.
2. A universal issue is only having access to publicly available datasets which is may not be suitable for the application under study. Deep learning is often cited to require large data sets, which is not necessarily available to academia.
3. Not having as large engineering teams to develop large-scale systems, and other computational resource limitations (as compared to industry).
4. Machine learning techniques often require a lot of parameter and framework tuning, and it is not always clear as to which architecture will work better. This can be frustrating for engineering academics, who may be beginners to deep learning concepts.
5. Industry has clear business goals, in contrast to scientific ones.

But a clear assessment of the technology readiness level for deep learning in health management might be difficult assess. The influential factors include software/hardware demonstration, proven reliability of the implementation design and an assessment of potential impact on overall cost of research and development. An important indicator in this context can be an integration readiness levels proposed for evaluating the complexity of integrating these techniques into existing applications.

5. CONCLUSIONS

Deep neural networks are known to overcome the vanishing gradient problem; which was severely limiting the depth of neural networks. It is as simple as that. Neural networks are trained using backpropagation gradient descent. That is, the weights are updated for each layer as a function of the derivative of the previous layer. The problem is that the update signal can get lost as the depth of the neural network increases. Therefore, practitioners would often only use neural networks with a single hidden-layer. But now, as there is the possibility to implement large neural networks, this has opened a door of opportunities to techniques such as auto-encoders for unsupervised problems, convolutional neural networks to classify images, recurrent neural networks for time series, etc.

This paper provides a brief description of deep learning methods and discusses its potential for health management applications. Although the effectiveness of the various approaches was not addresses, it presents some of the recent advances and problems of the engineering community. It is difficult to assess whether it will be at the academic frontier in upcoming years – as academic research now-a-days tends to have a short shelf-life and get replaced by new ideas and trendy topics. But due to recent industrial efforts, the machine learning field has moved quickly, and perhaps a few years from now may look nothing like what is call "deep learning". That being said, there is an unprecedented interest from a number of technology organizations, other academic disciplines, and even the general public, on the topic. Despite the hype and how academics perceive it, deep learning seems quite valuable in the monetary sense. It has enabled an array of real-world commercial products and services that were not technologically feasible before and hence it could prove useful for the aerospace maintenance community.

ACKNOWLEDGEMENT

This research was supported by the Japanese Society for the Promotion of Science (JSPS). Grant number: P16079.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*. https://arxiv.org/pdf/1603.04467

Barranco, R. C., Rodriguez, L. M., Urbina, R., & Hossain, M. S. (2016). Is a Picture Worth Ten Thousand Words in a Review Dataset?. arXiv preprint arXiv:1606.07496. https://arxiv.org/pdf/1606.07496

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.

Beyca, O. F., Rao, P. K., Kong, Z., Bukkapatnam, S. T., & Komanduri, R. (2016). Heterogeneous Sensor Data Fusion Approach for Real-time Monitoring in Ultraprecision Machining (UPM) Process Using Non-Parametric Bayesian Clustering and Evidence Theory. IEEE Transactions on Automation Science and Engineering, 13(2), 1033-1044.

Bishop, C., 1994. Novelty detection and neural network validation. In: Vision, Image and Signal Processing, IEE Proceedings, Vol. 141, No. 4. IET, pp. 217

Campbell, C. and Bennett, K., 2001. A Linear Programming Approach to Novelty Detection, Cambridge, UK: Cambridge Press.

Desforges, M., Jacob, P., and Cooper, J., 1998. Applications of Probability Density Estimation to the Detection of Abnormal Conditions in Engineering. pp. 687–703.

Diaz, I. and Hollmen, J., 2002. Residual Generation and Visualization for Understanding Novel Process Conditions. IEEE, pp. 2070–2075

Eti, M. C., Ogaji, S. O. T., & Probert, S. D. (2006). Reducing the cost of preventive maintenance (PM) through adopting a proactive reliability-focused culture. Applied energy, 83(11), 1235-1248.

Busseti, E., Osband, I., Wong, S. (2012). Deep Learning for Time Series Modeling CS 229 Final Project Report. http://cs229.stanford.edu/proj2012/BussetiOsbandWong-DeepLearningForTimeSeriesModeling.pdf

Fujimaki, R., Yairi, T., & Machida, K. (2005). An anomaly detection method for spacecraft using relevance vector learning. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 785-790). Springer Berlin Heidelberg.

Guttormsson, S., Marks, R., El-Sharkawi, M. and Kerszenbaum, I., 1999. Elliptical novelty grouping for online short-turn detection of excited running rotors. IEEE Transactions of Energy Conversions, 14(1), 14.

Harries, T., 1993. Neural Network in Machine Health Monitoring. Professional Engin.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), 82-97.

Huynh, K. T., Barros, A., & Bérenguer, C. (2015). Multi-level decision-making for the predictive maintenance of-out-of-: F deteriorating systems. IEEE Transactions on Reliability, 64(1), 94-117.

Jakubek, S. and Strasser, T., 2002. Fault-Diagnosis Using Neural Networks with Ellipsoidal Basis Functions. pp. 3846–3851.

Jia, F., Lei, Y., Lin, J., Zhou, X., & Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. Mechanical Systems and Signal Processing, 72, 303-315.

Khan, S., Phillips, P., Jennions, I., & Hockley, C. (2014). No Fault Found events in maintenance engineering Part 1: Current trends, implications and organizational practices. Reliability Engineering & System Safety, 123, 183-195.

Keogh, E., Lin, J., Lee, S.-H. and Herle, H., 2006. Finding the most unusual time series subsequence: Algorithms and applications. Knowledge Information Systems, 11(1), 1–27

Kobayashi, T., & Simon, D. L. (2005). Hybrid neural-network genetic-algorithm technique for aircraft engine performance diagnostics. Journal of Propulsion and Power, 21(4), 751-758.

Lee, D., Siu, V., Cruz, R., & Yetman, C. (2016, January). Convolutional Neural Net and Bearing Fault Analysis. In Proceedings of the International Conference on Data Mining (DMIN) (p. 194). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Li, Y., Pont, M. and Jones, N., 2002. Improving the performance of radial basis function classifers in condition monitoring and fault diagnosis applications where unknown faults may occur. Pattern Recognition Letters, 23(5), 569–577

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., van Ginneken, B. and Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. arXiv preprint arXiv:1702.05747.

Lv, F., Wen, C., Bao, Z., & Liu, M. (2016, July). Fault diagnosis based on deep learning. In American Control Conference (ACC), 2016 (pp. 6851-6856). IEEE.

Meziane, F. (Ed.). (2009). Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects: New Prospects. IGI Global.

McDuff, R. J., Simpson, P. K., & Gunning, D. (1989, September). An investigation of neural networks for F-16 fault diagnosis. I. System description. In AUTOTESTCON'89. IEEE Automatic Testing Conference. The Systems Readiness Technology Conference. Automatic Testing in the Next Decade and the 21st Century. Conference Record. (pp. 351-357). IEEE.

Nelles, O. (2013). Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Springer Science & Business Media.

Patton, R. J., Frank, P. M., & Clark, R. N. (Eds.). (2013). Issues of fault diagnosis for dynamic systems. Springer Science & Business Media.

Purarjomandlangrudi, A., Ghapanchi, A. H., & Esmalifalak, M. (2014). A data mining approach for fault diagnosis: An application of anomaly detection algorithm. Measurement, 55, 343-352.

Russell, B. D., & Benner, C. L. (2010). Intelligent systems for improved reliability and failure diagnosis in distribution systems. IEEE Transactions on Smart Grid, 1(1), 48-56.

Şeker, S., Ayaz, E., & Türkcan, E. (2003). Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery. *Engineering Applications of Artificial Intelligence*, *16*(7), 647-656.

Shin, J. H., & Jun, H. B. (2015). On condition based maintenance policy. Journal of Computational Design and Engineering, 2(2), 119-127.

Shui, H., Jin, X., & Ni, J. (2015). An Anomaly Detection and Diagnosis Method Based on Real-Time Health Monitoring for Progressive Stamping Processes. In ASME 2015 International Manufacturing Science and Engineering Conference (pp. V002T05A021-V002T05A021). American Society of Mechanical Engineers.

Sutharssan, T., Stoyanov, S., Bailey, C. and Yin, C. (2015). Prognostic and health management for engineering systems: a review of the data-driven approach and algorithms. The journal of engineering. doi: 10.1049/joe.2014.0303

Volponi, A. J., DePold, H., Ganguli, R., & Daguang, C. (2003). The use of Kalman filter and neural network methodologies in gas turbine performance diagnostics: a comparative study. Journal of engineering for gas turbines and power, 125(4), 917-924.

Wang, G., Li, Q., Sun, J., & Meng, X. (2015). Telemetry data processing flow model: a case study. Aircraft Engineering and Aerospace Technology: An International Journal, 87(1), 52-58.

Yairi, T., Hori, K., & Hirama, K. (2005). Qualitative map learning based on covisibility of objects. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 35(4), 779-800.

Yairi, T., Kawahara, Y., Fujimaki, R., Sato, Y., & Machida, K. (2006, July). Telemetry-mining: a machine learning approach to anomaly detection and fault diagnosis for space systems. In 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06) (pp. 8-pp). IEEE.

Yairi, T., Kato, Y., & Hori, K. (2001, June). Fault detection by mining association rules from house-keeping data. In proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (pp. 18-21).

Zhang, Y., & Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. Annual reviews in control, 32(2), 229-252.