Comparison of deep learning models for predictive maintenance

R Naren¹ and J Subhashini²

¹M-Tech-Embedded System Technologies, Department of electronics & Communication Engineering, SRM Institute of Science & Technology, Kattankulathur, Kanchipuram, Tamil Nadu, India
²Assistant Professor, Department of Electronics & Communication Engineering, SRM Institute of Science & Technology, Kattankulathur, Kanchipuram, Tamil Nadu, India

E-mail: narenr94@gmail.com

Abstract. There is a clear intersection between the Internet of Things (IoT) and Artificial Intelligence (AI). IoT is about connecting machines and making use of the data generated from those machines. AI is about simulating intelligent behaviour in machines of all kinds. As IoT devices will generate vast amounts of data, then AI will be functionally necessary to deal with these huge volumes if we’re to have any chance of making sense of the data. AI is beneficial for both real-time and post event processing: Post event processing – identifying patterns in data sets and running predictive analytics, e.g. the correlation between traffic congestion, air pollution and chronic respiratory illnesses within a city centre. Real-time processing – responding quickly to conditions and building up knowledge of decisions about those events, e.g. remote video camera reading license plates for parking payments.

1. Introduction

Using data exchange and automation in manufacturing is referred to as Industry 4.0. Nine crucial technologies that the Industry 4.0 comprises of are: Horizontal and Vertical System Integration, Cybersecurity, Autonomous Robots, Simulation, the Industrial Internet of Things, The Cloud, Additive Manufacturing, Data & Analytics, and Augmented Reality. These technologies enable dynamic communication between Robots, Instruments and Humans to provide increased and monitorable improvement in productivity. For the sake of this paper we are going to concentrate on only two of the nine above mentioned principle technologies, namely, Industrial Internet of Things and Data & Analytics.

We live in an analog world, The Internet of Things puts sensors and networking elements on these analog processes we encounter every day. The Internet of Things has the potential to be implemented in both the industrial and domestic domain, it can be implemented for anything as monitoring power consumption of household utility to monitoring and analysing complex industrial procedures. IoT along with Industry 4.0 is bound to create massive amount of data. Having the necessary means to collect analyse and process this data quickly and effectively is the challenging part. In addition, information generated must be understandable by humans, this is challenging as most data today are chaotic and un-formatted.

Some of the common challenges encountered by organisations when implementing IoT and AI are with accessibility, application and analysis of IoT data. If data is segregated from multiple sources in
form of clusters statistical analysis maybe performed on them. But, if the application requires the prediction of failure or breakdown of a process say end of life of a drill bit or failure of the machinery all together, this requires the data to be interleaved with the process. The data generated by big organisation that have implemented tags or sensors to the fundamental units by virtue of size is truly massive. This massive amount of data is impractical to be processed by conventional analytic tools and predictive techniques. But, AI by its very nature offers an effective solution for problems of this very nature. Un-supervised and clustering algorithms enable us to easy find deviants from a regular patter with high efficiency for large amount of data. Likewise, these AI-enabled IoT systems are able to bring to light commonalities and patterns within the data that are invisible to the human eye or are beyond the scope of tradition techniques.

Predictive maintenance is a proactive technique by which it is possible to determine duration within which the machine is probable to fail, this allows maintenance to be scheduled beforehand. This prediction is made by taking the data from sensors that are connected various parts of the machinery that affect the procedure and the algorithm finds the pattern of normal functionality and as also when the machinery nears failure, thus enabling prediction of failure before it happens. Condition monitoring equipment that asses the condition of the machinery against the process makes predictive maintenance possible. Using this condition monitoring equipment along with IoT to feed the Predictive algorithm a reliable and effective system can be established for this application. Information acquired from this will give the organisation the idea of what equipment needs attention and at what time.

Neural networks are set of pattern recognition algorithms that are fundamentally modelled after the human capability of memory retention. They like the human brain are trained over multiple similar exposures to recognise distinct patterns and form classifications. But, unlike the human brain the patterns Neural Networks find exist in the numerical domain in form of vectors, and the outcome from this recognition also exists in the numerical domain in the form of vectors. Classification is basically more than one outcome is expected and the network is able to make clear predictive distinction between them. You might call this a static prediction. Similarly, deep learning algorithms are able to correlate present events to future events. The future event is outcome in perspective to these types of applications. Deep learning makes no distinction in the nature of the data that is input to it, that is, a time series data set will be treated as static data with time being just another data variable. Given a time series, deep learning can make future predictions such as finding the next number in the series of numbers even though the number predicted will only come to happen in the future.

2. Literature Survey

Machine Learning is a technique used to obtained a predictive outcome using existing knowledge without requiring any explicit coding for the particular application [1]. It mostly revolves around the concept of using computer codes to analyse existing data to “make sense of” or find patter in the seemingly chaotic data set in a predictive manner [2]. The objective of machine learning is not to mimic human conscious decision making but rather to enable the code to by itself without any human interference to be able to recognize and categorize data on basis of patter they exhibit. Thus, significantly reducing man-power required for the task of categorization. But even though machine learning algorithms far exceed the performance of humans in simple and semi-linear applications, they lack the performance and predictive capability by a large margin in complex application. Computational learning theory is a means of validating and evaluating the predictive capabilities of a machine learning algorithm statistically. Machine learning in all fundamental sense seems to resemble the characteristics of data mining and prediction algorithms [3,4].

Multiple studies have been conducted and papers have been published on how to establish and train a machine learning algorithm to customised advertising for very user. Some other popular applications for machine learning include, spam filtering, data mining, anomaly detection, fraud detection and customizing news feeds [5,6]. For formatted data with distinct outcomes machine learning can be
classified on basis of their utilization as supervised, semi-supervised and re-enforced machine learning [7,8].

General machine learning algorithms such as SVM or Feed Forward Neural Networks consider each set of data as an independent entity thus making their performance poor in scenarios where the data is sequential and the order of the sequence plays a major role in the characterisation of the dataset. Hidden Markov Models (HMM) and high order Markov chains are alternative techniques for analysing time series data [9-12]. But, the state space for these algorithms increase exponentially when size of dataset is increased making them impractical for most real time applications that by nature consist of huge datasets [13]. Classic sequential pattern [14,15] and rule mining methods [16,17] are useful to analyse and evaluate data as long as they are not too big, which they are in most cases. However, these methods fail to handle long sequences and large datasets. Also, these algorithms require that there are no duplications within the dataset, but most data in real time are unpredictable and can yield duplicated data. Thus, LSTM (Long Short Term Memory), GRU (Gated Recurrent Unit) are the algorithms that will be used and compared for the scope of this paper with RNN (Recurrent Neural Network) serving as a benchmark to validate their performances.

3. Methods

3.1. Recurrent Neural Network

Recurrent Neural Network (RNN) is a type of Neural Network where the output from previous step are fed as feedback to the current step. Conventional Neural Networks lack this feedback mechanism to make predictions that are temporal in nature. RNN solved this issue by introducing a hidden layer that remembers all the past data series, this hidden layer is then fed to the network along with the input.

![RNN block-diagram](image)

**Figure 1.** RNN block-diagram.

In discrete time supervised learning, data arrive at real time in the form of one vector at a time. During calculation of result every non-input unit computes its next value using its current value non-linearly with the new input fed to it. Given target activations can also be taken from some output. For example, an application to recognise digits from spoken voice input sequence, the final output at the end of the sequence may be a label classifying the digit. Figure 1 shows the RNN block-diagram.

The hidden layer functions as a memory remembering all the calculation that have taken place before. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity.

3.2. Long Short Term Memory

Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM solved the issue of long-term dependencies of RNN in which the RNN cannot predict accurately of data that have dependencies in the output over a long-term but is efficient only for short-term predictions. As the gap length between the input and output that depends on increases RNN is basically ineffective. LSTM overcomes this drawback. Figure 2 shows the LSTM block-diagram.
Cells retain the information and the gates manipulate the “memory” already present. The first gate is called Forget Gate: The functionality of this gate is to remove information that is no longer useful in the cell. Two inputs $x(t)$ (current input) and $h(t-1)$ (previous cell output) are input to this gate and multiplied with weight matrices and then added with the bias. The result is processed by an activation function that makes the output binary in nature. If the outcome is 0 then the information no longer holds any value and is deleted, and similarly if 1 is obtained the information is kept for use in the future. Second gate is called The Input gate: This gate is responsible for adding new useful information to the cell. The information is bounded using a sigmoid function and the is filtered to find values to be remembered using inputs $h(t-1)$ and $x(t)$. Tanh function is used to make a vector whose values range between $-1$ and $+1$ which contains all the possible values from $h(t-1)$ and $x(t)$. The product of vector and regulated value yields useful information. The final gate is called The Output gate: This gate is responsible to select relevant information that needs to be fed as output for then cell. First, tanh function is used to generate a vector. Then, similar to previous gates regulated information is obtained using sigmoid function and then filtering using inputs $h(t-1)$ and $x(t)$. Product of the vector and regulated information yields the output and input to the next cell.

3.3. Gated Recurrent Unit Networks

To solve the Vanishing-Exploding gradients problem often encountered during the operation of a basic Recurrent Neural Network, many variations were developed. One of the most famous variations is the Long Short Term Memory Network (LSTM). One of the lesser known but equally effective variations is the Gated Recurrent Unit Network (GRU). Unlike LSTM, it consists of only three gates and does not maintain an Internal Cell State. The information which is stored in the Internal Cell State in an LSTM recurrent unit is incorporated into the hidden state of the Gated Recurrent Unit. This collective information is passed onto the next Gated Recurrent Unit figure 3 shower the GRB lock diagram.

The first gate is, Update Gate($z$): It determines how much of the past knowledge needs to be passed along into the future. It is analogous to the Output Gate in an LSTM recurrent unit. Second gate is, Reset Gate($r$): It determines how much of the past knowledge to forget. It is analogous to the combination of the Input Gate and the Forget Gate in an LSTM recurrent unit. The third gate is,
Current Memory Gate($\overline{h}_t$): It is often overlooked during a typical discussion on Gated Recurrent Unit Network. It is incorporated into the Reset Gate just like the Input Modulation Gate is a sub-part of the Input Gate and is used to introduce some non-linearity into the input and to also make the input Zero-mean. Another reason to make it a sub-part of the Reset gate is to reduce the effect that previous information has on the current information that is being passed into the future.

4. Evaluation

4.1. Datasets

The data set chosen to test the performances of above mentioned models is a subset of a dataset released by NASA back in 2008 for a competition to test accuracy of different models at predicting the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate.

The dataset file contained a total of 14 files. Of which 13 are text files and 1 pdf file. The text files contain, 4 train data files, 4 test data files, 4 Remaining useful life (RUL) and 1 readme file. The four files from each category corresponds to one of four conditions under which the reading had been obtained. For the sake of this paper the first data sub-set with the simplest conditions was chosen for simulation. The test data for this category has readings from 100 engine units and a total of 20,631 data sequences. The test data has 13,096 data sequences for the same 100 units of engines. And the RUL file contained the correct value to be predicted for the 100 units of engines in the test data by the model used. The test and train data contain 21 sensor data along with engine unit id, RUL and three operational settings data.

4.2. Testbench

The train data and test data are segmented into frames of size 50 each. These frames are fed into the models as input at one go. And each frame is displaced from its previous frame by a single data sequence.

The neural network setup consists of the input layer of the desired model type (RNN/LSTM/GRU) of 100 units (neurons), followed by another layer of the model type (RNN/LSTM/GRU) of 50 units to provide a gradual descent to the final dense layer (all neurons of a dense layer are connected to all neurons of the previous layer) of unit size 1, with linear activation function as the desired output is regressive.

An early termination condition is also set to terminate the training if the loss doesn’t decrease for 10 epochs. The simulation will run for a maximum of 100 epochs.

The dataset is divided in the ratio 7:3 where 7 units are used for training and 3 units for validation.

4.3. Results

The simulation was run on the same host machine for the same size dataset with no other parallel running processes during the execution, to avoid interference of external factor.

From figures 4 and 5 it can be deduced that though the accuracy of prediction improves exponentially with total number of iterations/epochs, at around 20 epochs it can be noted that there seems to be fluctuations in the accuracy that deems that RNN algorithm after a certain accuracy is unreliable in its predictiveness.
Figure 4. RNN loss vs epoch.

Figure 5. RNN $r^2$ vs epoch.
In figure 6 and figure 7 it can be seen that compared to RNN the LSTM algorithm has a slower initial learning curve but provides much smoother and accurate readings with higher epochs.

**Figure 6.** LSTM loss vs epoch.

**Figure 7.** LSTM $r^2$ vs epoch.
In Figure 8 and Figure 9 it can be seen that compared to RNN the LSTM algorithm has a slower initial learning curve but provides much smoother and accurate readings with higher epochs.

**Figure 8.** GRU loss vs epoch.

**Figure 9.** GRU $r^2$ vs epoch.
Figures 10, 11 and 12 show the comparison of the predictions made by RNN, LSTM and GRU respectively on a part of the dataset that wasn’t used for training.

**Figure 10.** RNN Predictions.

**Figure 11.** LSTM Predictions.
It can be clearly seen that GRU provides better accuracy overall, when compared to RNN and GRU. Table 1 provides an insight on the various parameters that were compared amongst the algorithms during testing and evaluation segments of the simulation.

| Table 1. Comparison of RNN VS LSTM VS GRU. | RNN | LSTM | GRU |
|------------------------------------------|-----|------|-----|
| Average time/epoch or response time (sec) | 13  | 31   | 24  |
| Epochs executed before termination of training | 85  | 59   | 70  |
| Training Period (mins) | 18.5 | 30.5 | 28  |
| R^2 (training) | 0.77 | 0.79 | 0.85 |
| R^2 (validation) | 0.701 | 0.8088 | 0.8134 |

5. Conclusion
In this paper, we presented a comparison among three neural networks capable of processing time sequence data, over a considerably big dataset to quantify and compare their performance.

Here, we can conclude that the accuracy was highest in GRU model with 9.411% and 7.05% increase in R squared score during training compared to RNN and LSTM respectively and 13.5% and 1.3% increase in R squared score during validation as compared to RNN and LSTM respectively, while still sporting a good training period, which is 7.1% lesser than LSTM but 35% greater than RNN. Thus, GRU is the better of the three models tested for the specific prediction analysis.

6. References
[1] Holzinger A, Plass H, Holzinger K, Crisan C G, Pintea C M, and Palade V 2017 A glass-box interactive machine learning approach for solving NP-hard problems with the human-in-the-loop arXiv preprint arXiv:1708.01104
[2] Ian H, Witten, Frank E, Hall M A, and Pal C J 2016 Data Mining: Practical machine learning tools and techniques *Morgan Kaufmann*
[3] Kohavi R 1998 Glossary of terms Machine Learning 30 271274
[4] Mannila H 1996 Data mining: machine learning, statistics, and databases Scientific and Statistical Database Systems. Proceedings Eighth International Conference on. IEEE
[5] Ongsulee P and Pariwat 2017 Artificial intelligence, machine learning and deep learning ICT and Knowledge Engineering 15th International Conference on IEEE
[6] Friedman J H 1988 Data mining and statistics: What's the connection? Computing Science and Statistics 29 3-9
[7] Jain A K, Murty M N and Flynn P 1999 Data clustering: a review ACM Comput Surveys. 31, 264–323
[8] Dutton M and Conroy G V 1997 A review of machine learning The Knowledge Engineering Review 12 341-367
[9] Kinnebrew J S and Biswas G 2011 Comparative action sequence analysis with hidden markov models and sequence mining Knowledge Discovery and Data
[10] Scholz M 2016 R package clickstream: Analyzing clickstream data with markov chains Journal of Statistical Software 74 1–17
[11] Melnykov V 2016 ClickClust: An R package for model-based clustering of categorical sequences Journal of Statistical Software 74 1–34
[12] Jurafsky D and Martin J H 2018 Hidden Markov Models Speech and Language Processing 2nd Edition
[13] Lipton Z C, Berkowitz J, and Elkan C 2015 A critical review of recurrent neural networks for sequence learning arXiv preprint arXiv:1506.00019
[14] Zaki M J 2001 SPADE: An efficient algorithm for mining frequent sequences Machine learning 42 31–60
[15] Fournier-Viger P, Gomariz A, Campos M and Thomas R 2014 Fast vertical mining of sequential patterns using co-occurrence information Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer 40–52
[16] Lo D, Khoo S C, and Wong L 2009 Non-redundant sequential rulestheory and algorithm Information Systems 34 438–453
[17] Xu L D, He W, and Li S 2014 Internet of Things in Industries: A Survey IEEE Transactions on Industrial Informatics 10 2233-2243
[18] Yi L, Deng X, Wang M, Ding D, and Wang Y 2017 Localized Confident Information Coverage Hole Detection in Internet of Things for Radioactive Pollution Monitoring IEEE Access 5 18665-18674
[19] Bhuiyan M Z A, Wang G, Wu J, and Cao J 2017 Dependable Structural Health Monitoring Using Wireless Sensor Networks IEEE Transactions on Dependable and Secure Systems 14 363-376