

Comparison of Hierarchical Temporal Memories and Artificial Neural Networks under noisy data

L. Rodriguez-Cobo*, J. Mirapeix, A. Cobo, and J.M. Lopez-Higuera

Photonics Engineering Group, University of Cantabria, Spain.

e-mail: luis.rodriguez@unican.es

March 10, 2014

Abstract

The ability of two different machine learning approaches to map non-linear problems from experimental data is evaluated under controlled experiments. A well-known machine learning algorithm (Artificial Neural Network) is compared against a new computing paradigm (Hierarchical Temporal Memory) under a controlled scenario. The chosen scenario is the detection of impacts in a cantilever beam under vibration instrumented with Fiber Bragg Gratings. The main characteristics of both of the machine learning approaches are analyzed varying environmental parameters such as the number of sensing points or their location. From the achieved results some clues can be extracted when dealing with noisy or partial data using different machine learning approaches.

1 Introduction

As the computing power increases, the ability of machine learning approaches to solve non-linear problems is also being improved. Some problems that could seem trivial having an a priori knowledge of them, can result too complex without this information. There are several scenarios that confirm this assumption such as pattern recognition, automatic navigation (Pomerleau, 1991) or job analysis (Fonseca and Navarrese, 2002), where machine learning algorithms have been successfully employed.

An important goal when using machine learning approaches, is to let the algorithm learn the a-priori knowledge by itself, allowing a correct response of the system. To perform this task, these algorithms are usually trained with the data to be classified indicating the correct output. There are lots of different machine learning algorithms applied to engineering fields (Reich and Barai, 1999) but they do not evolve in the same way when the incoming data are modified due to noise or environmental conditions changes, making one of them more suitable than the rest for a given application. This suitability can be studied by changing the environmental parameters of the problem such as, the amount of presented data, the pre-processing scheme, etc. . . These kind of studies can show some clues to improve the final application by establishing the number of acquisition points or determining their best location and pre-processing scheme.

In this work, two machine learning approaches, Hierarchical Temporal Memories and Artificial Neural Networks are applied to a well-known problem to study their final performance under different working conditions. The addressed problem is the impact detection in a vibrating cantilever beam. The beam has been instrumented with Fiber Bragg Gratings (Hill and Meltz, 1997), varying their installation procedure and position to obtain different points of view of the same test. There are several deterministic approaches that solves this problem (Bang et al., 2004; Frieden et al., 2011; Kirkby et al., 2011), even using machine learning approaches (Dua et al., 2001; Coelho et al., 2010), but all of them employ an a-priori knowledge of the problem to be solved. In the following section, an introduction to the chosen machine learning algorithms will be detailed. Then, the application scenario and data processing schemes will be also described. Finally, the experimental results will be depicted and discussed.

2 Machine learning algorithms

There are lots of machine learning algorithms but in this work two of them are compared in the same application: Artificial Neural Networks (ANN) and Hierarchical Temporal Memories (HTM). Introduced by Hawkins and Blakeslee (2004) and formalized by George and Hawkins (2009), HTM is a high level computation model inspired in the human neocortex, and a software implementation called NuPic developed by the company Numenta, Inc is offered. ANN is a well known, widely used supervised learning algorithm capable of modeling highly non-linear problems.

The ANN learning algorithm (Mehrotra et al., 1997) is inspired in the functional aspects of biological neural networks. It is a supervised learning

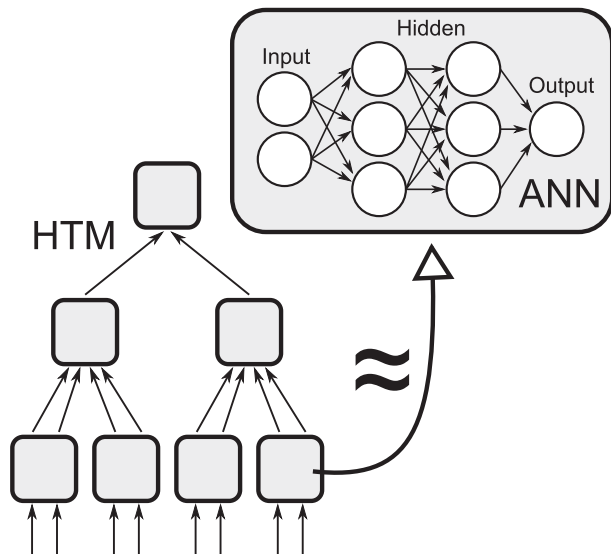


Figure 1: Schematic complexity of ANNs and HTMs

algorithm in which a group of interconnected neurons compute the same non-linear function (typically the sigmoid function) of the weighted inputs. ANNs have been widely used in many different fields including machine vision tasks (Chow and Cho, 2002) or industrial process control (Masri et al., 2000). Both algorithms require a training phase before performing the classification tasks to allow the required problem modeling.

The complex HTM algorithm can be described as a hierarchy of interconnected high-level identical nodes which are able to process and store information from their data inputs. Each high-level node of this model mimics the behavior of the cortical columns of the biological brain ($\approx 60,000$ neurons). Unlike ANNs, time is a very important aspect of HTMs: in the training phase, they are precisely the time variations of the input data what allows to identify their invariant patterns and to categorize them. HTMs have been used for different applications such as recognition of hand-written digits (Stolc and Bajla, 2010) or image retrieval (Bobier and Wirth, 2009), but most of them on machine vision applications.

As happens in ANNs, each node of a HTM computes the same algorithm but it has two differentiated stages. As shown in Fig. 2 (right), the first stage is the spatial pooler where a set of coincidence patterns (four at the figure) is created during the training phase. To get the coincidence patterns during the training phase, every time that new data arrive, their Euclidean distance is calculated against the set of already created quantization centers. If the distance of the new data is larger than a configuration parameter,

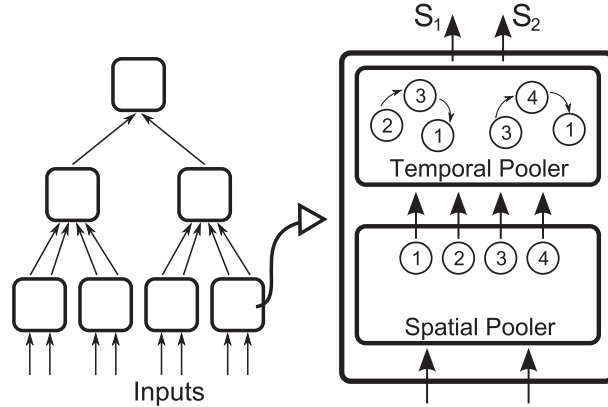


Figure 2: Architecture of HTMs. Detail of a HTM node

Max Distance, the pattern is added to the spatial pooler as a new center of quantization. The output provided by the spatial pooler is a vector with the probability distribution over the quantization centers, where the maximum of the output vector is the most probable pattern for a given input.

After the spatial pooler training phase, the training phase of the temporal pooler follows, where the temporal evolution of its input vectors (the probability distribution over the quantization centers of the spatial pooler) is used to progressively build a probabilistic transition matrix between the patterns. A set of the most probable pattern transitions (temporal sequences) is built, which is the training result in the temporal pooler. The training phase in the temporal pooler is mainly controlled by 2 parameters: Top Neighbors, which specifies the maximum number of coincidence patterns that are added simultaneously to a temporal sequence; and Transition Memory, which specifies how many input vector transitions are kept in the temporal pooler to track the time structure of coincidences while learning the time adjacency matrix.

The output of the temporal pooler is a vector with the probability distribution over the space of temporal sequences, being the maximum of this vector the most probable sequence found in the training phase. This information is passed to the upper nodes in the hierarchy.

Once the HTM is trained, the inference mode can be used to perform the classification task. There are several algorithms for the inference mode, the Gaussian algorithm is commonly chosen and it works as follows: for every new set of data at the inputs of the bottom nodes, the probability of being any of the already stored patterns of the spatial poolers is calculated assuming a Gaussian function of the Euclidean distance. The parameter of this Gaussian function can be also configurable for each node (Sigma). After that, the probability vector is passed to the temporal pooler and the probability of

being any of the stored temporal sequences is computed. The classification output computed by this algorithm is the vector with the probability of belonging to one of the temporal sequences.

Both approaches have particular advantages for the application discussed in the following section. On the one hand, ANNs are a relatively simple algorithm, but with capabilities of modeling highly non-linear patterns. On the other hand, HTMs are a more complex algorithm where a lot of parameters have to be set (apart from the architecture). HTMs can better process temporal sequences by using the temporal pooler, but they also need very high dimensional input vectors to work properly. On the contrary, ANNs work perfectly with small dimension input vectors, but they do not exhibit memory capabilities to process temporal sequences.

3 Application scenario

Detection of low energy impacts in a composite beam under vibration has been chosen to test both machine learning algorithms. This is a widely addressed problem in the literature where, by analyzing strain signals of different sensors, the impacts can be detected (Bang et al., 2004) and even located (Kirkby et al., 2011). In this work, a Glass Fiber Reinforced Plastic (GFRP) beam of 640 by 80 mm has been manufactured including 8 strain sensors located in different positions. One edge of the beam was fixed while the other one was attached to a electromagnetic shaker (TIRAvib 51110). The shaker was employed to periodically deform the beam adding extra noise to the measurements, hiding the low energy impacts. An electromagnetic actuator was also placed under the beam in 4 different positions to cause the impacts.

The 8 strain sensors were located at four different positions in the longitudinal axis but at two different depths: 4 of them were embedded into the composite beam and the other 4 were glued to the beam surface. The selected strain sensors are based on optical fiber technology, particularly on Fiber Bragg Gratings (FBGs) (Hill and Meltz, 1997), a technology highly compatible with composite materials (Frieden et al., 2011). The glued sensors have been attached to the beam surface after their manufacture; on the contrary, the embedded FBGs have been placed in the middle of the glass fiber plies during the beam manufacture. All the strain sensors have been interrogated using a commercial unit (si425 of Micron Optics) with a sampling frequency of 250 Hz, which limits the high frequency response of the impacts. To obtain the data, the shaker has been fed with a sinusoidal wave with a constant amplitude and a frequency varying between 1 and 20 Hz. For

each frequency, an electromagnetic actuator was employed to hit the beam 6 times in 4 different positions for each frequency (a total of 480 impacts) with a measured energy of 0.2-0.3J.

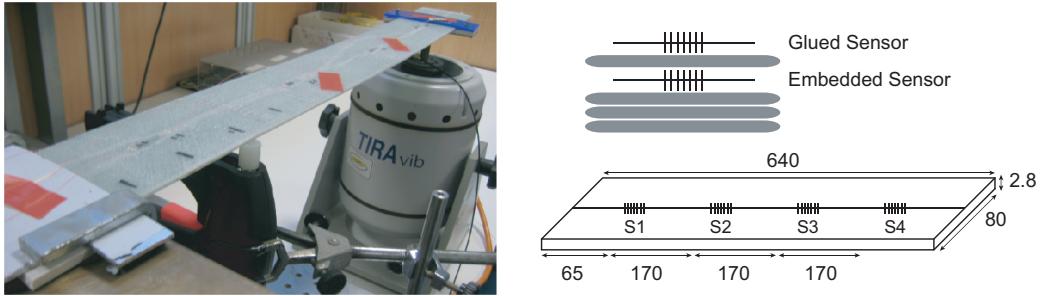


Figure 3: Photo of the experimental setup (left). Installation details of FBG sensors (right). Dimensions are in millimeters.

3.1 Fiber Bragg Grating (FBG)

In a simple way, a Fiber Bragg Grating (FBG) (Hill and Meltz, 1997) is a periodic variation of the refractive index in the optical fiber core that reflects specific wavelengths. The reflected wavelengths are centered around the Bragg wavelength, defined by $\lambda_{Bragg} = 2n_{eff}\Lambda$, where n_{eff} is the effective index (constant) in the fiber core and Λ is the period of the refractive index variation. By elongating the FBG, Λ is increased and, therefore, also the central wavelength (λ_{Bragg}). So, by measuring the central wavelength, the strain of the holder structure where the FBG is attached can be determined. This principle has been widely used for structural health monitoring in different fields such as civil engineering or renewable energies (Lopez-Higuera et al., 2011).

4 Experimental

The addressed application could be easily resolved by pre-processing the incoming data using a high-pass filter to remove the added noise. In this way, impacts can be easily detected by simply employing a threshold (Fig. 5), but this solution implies an a-priori knowledge of noise and perturbation frequency contents. Within this application scenario, characteristics of the available data were studied on two different machine learning schemes. The

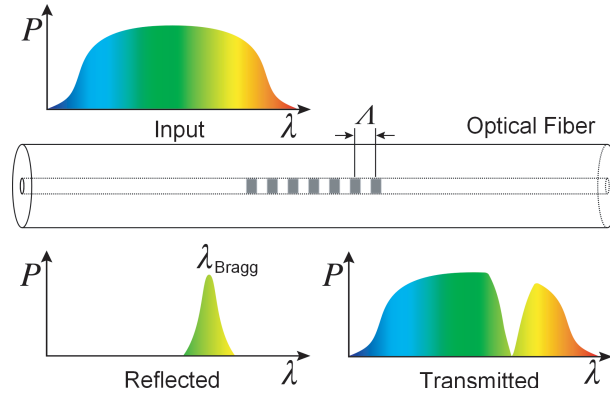


Figure 4: Basis of Fiber Bragg Grating

amount of fed data as well as their sensitivity have been evaluated taking into account the final performance.

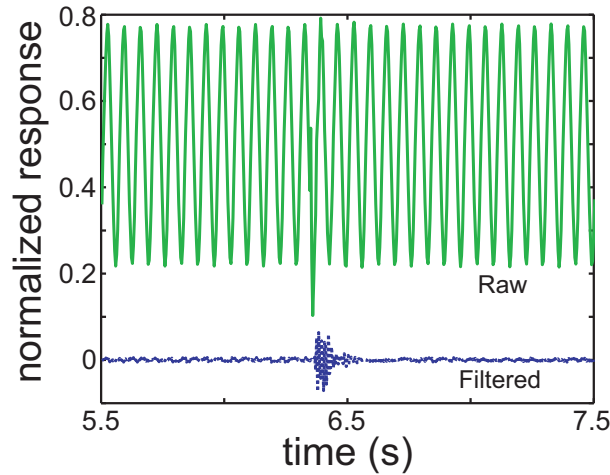


Figure 5: High pass filtered FBG signal. Impacts can be detected using a threshold.

4.1 Data processing

Before the training phase of both algorithms, a labeling step identifies each hit with a fixed duration of 0.36 seconds (90 samples at 250Hz). This value has been obtained by averaging the impact durations during a pre-processing step. All the strain data were scaled between 0 and 1 to allow a good numeric convergence of the algorithms. The available data have been split into 3 datasets for cross-validation. For each case, both algorithms were trained

with 2 datasets and tested with the remaining one. The process was repeated 3 times, making all the possible combinations, and the final performance was averaged. In each dataset, 2 of the 6 recorded hits for each situation were employed.

Impact detection is usually an asymmetric classification problem: there are more data in the “negative” state (non impact) than in the “positive” state (impact). In particular, in this work, there are 22.5% of samples labeled as impacts, so employing the classification rate metric can be confusing. In these cases, useful metrics are “precision”, “recall” and its harmonic mean “F1-score” (Goutte and Gaussier, 2005) that are defined as:

$$\begin{aligned}
 \textit{precision} &= \frac{\textit{true positives}}{\textit{true positives} + \textit{false positives}} \\
 \textit{recall} &= \frac{\textit{true positives}}{\textit{true positives} + \textit{false negatives}} \\
 F1_{\textit{score}} &= 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}
 \end{aligned} \tag{1}$$

In a simple way, a high recall means that the algorithm returns most of the “positive” states (impacts). High precision means that the algorithm returns more “true positive” states (impacts) than “false negative” states (non impacts).

After defining the metric, both algorithms, ANN and HTM, were fed with the same input data in three different scenarios: employing the data from the 8 FBG sensors, just employing the data from the 4 sensors glued to the surface and just employing the data from the 4 embedded sensors. Both machine learning algorithms are able to detect individual impacts from the raw FBG signals. By using different ways to feed the data, particular advantages of each approach are highlighted. In these applications, those sensors placed on the surface are more sensitive to structural deformation, but they are also more exposed to environmental conditions (noise). On the contrary, the embedded sensors are more protected but they exhibit a lower sensitivity. The sensitivity difference between embedded and glued sensors can be noticed in Fig. 6, where the signals from two sensors at the S1 location during an impact are depicted.

It must be noticed that the frequency sweep includes the resonant frequency of the structure (measured to be 16.4 Hz). Close to the resonant frequency, the vibration amplitude increases significantly, making more difficult the impact detection task. Trying to better understand both approaches, different data strategies were followed and the characteristics of each case are explained below:

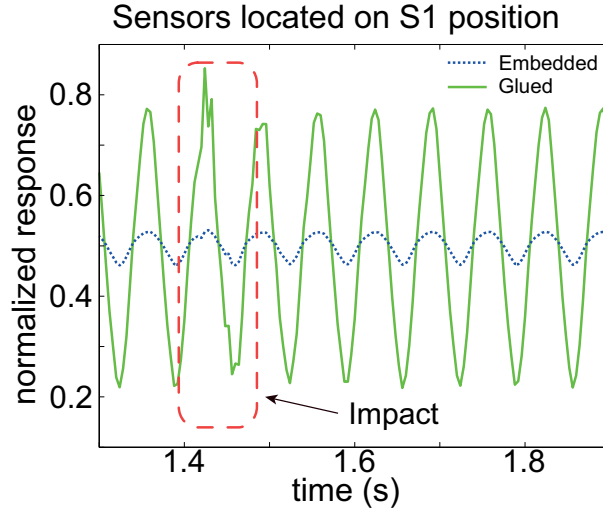


Figure 6: Deformation of S1 position sensors during an impact with $\text{Freq}=15\text{Hz}$

- Case A: Feeding with the 8 sensors data: In this situation, both surface-glued sensors and embedded sensors are included in the data fed to both algorithms. Therefore, there are highly redundant data in the non-impact state, but for impact states there is additional information due to the signal desynchronization between different sensors.
- Case B: Feeding with the 4 surface glued sensors data: By using the surface-glued sensors, there is still highly sensitive data from each position, but less redundant data.
- Case C: Feeding with the 4 embedded sensors data: The most sensitive signals (coming from the 4 glued sensors) are discarded and just the signals from the 4 embedded sensors are employed. As happens in Case B, with 4 sensors the data redundancy is weaker.

Besides different feeding data strategies, the algorithm architecture and parameters have been also swept trying to find an optimum configuration. All the configurations have been proposed to cover a wide set of modeling complexities trying to find the best fitting approach for this particular scenario. The ANN performance has been tested using 1 and 2 hidden layers, with a number of hidden neurons in each layer varying between 10 and 40 neurons (in steps of 10). The tested HTM architectures always consider 2 levels (as shown in Fig. 1, besides the top node), and the number of nodes in each level varies from 4 to 8 nodes for the first level and 2 to 8 for the second

level. The remaining HTM parameters have been empirically adjusted based in previous works for other classification problems (Rodriguez-Cobo et al., 2012), being related to the statistical properties of the input data (FBG signals for this scenario).

5 Results and discussion

The performance of both algorithms has been evaluated using the described metric (F1-score). The impact condition has been defined as a window of 64 samples of the vibration signal around the real instant of the impact. As there is not a clear end-of-impact indication within the vibration signal, and due to the presence of noise, a positive detection of the algorithms is considered when 16 of those 64 samples are labeled as “impact” in the output of the algorithm. In Fig. 7, a positive impact response of both algorithms is depicted (black line) against the desired output (solid line) and the individual decision of the algorithm (crosses) for each sample. In the HTM case, the individual decision is taken by selecting the higher output (non-impact and impact). For the ANN, the individual decision is taken by comparing the single exit against a given threshold. This threshold has been modified for each configuration in order to improve the F1-score.

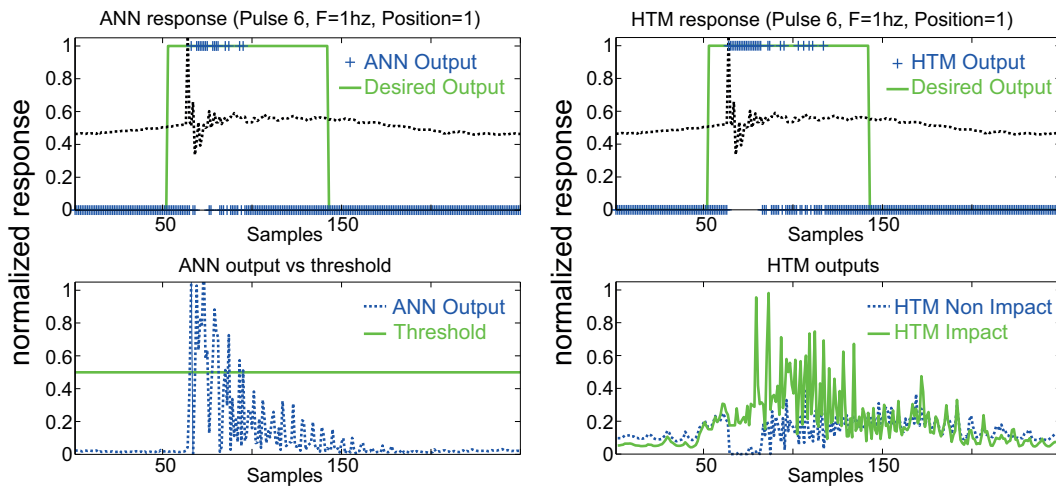


Figure 7: Positive output of both algorithms for an impact

As shown in Fig. 7, ANN tends to label less individual samples as an “impact” than HTM does. ANN is a memory-less algorithm in which the present output just depends on the current sample fed to the algorithm. This lack of memory causes a more impulsive response of the ANN, in contrast to

HTMs, where the output depends on the previous and current samples. The achieved responses of both algorithms under the three different approaches are described below:

5.1 Case A: Feeding with data from 8 sensors (both embedded and surface-glued)

In this case, there are 4 data sources with a high sensibility (sensors glued to the surface) and other 4 with less sensibility (sensors embedded into the GFRP). During the non-impact state, the less sensitive sensors practically do not add extra information to the others but, during the impact state, the embedded sensors add an extra point-of-view to the algorithms, making easier the classification task. ANN works rather well with a 2 level architecture of 30 hidden neurons per level by having an averaged F1score of 0.967 establishing the threshold in 0.5. HTM seems to also exhibit a good performance with an averaged F1score of 0.987 for a 8-8 architecture (8 nodes in level 1 and 8 nodes in level 2).

As shown, both approaches work properly with this case. Although ANNs don't have memory to work with temporal sequences, there are still enough different points of view of each "impact", thus the algorithm is able to properly detect this situation. In general, HTMs classify temporal sequences better when an enough amount of data is available. For this case, the raw samples coming from the 8 sensors are fed into the algorithm, so there are few inputs for making spatial patterns in lower nodes. This fact establishes a limit in the hierarchy complexity to a few nodes in level 1; in fact, simpler architectures such as 4-2 works also rather well with an F1score of 0.971.

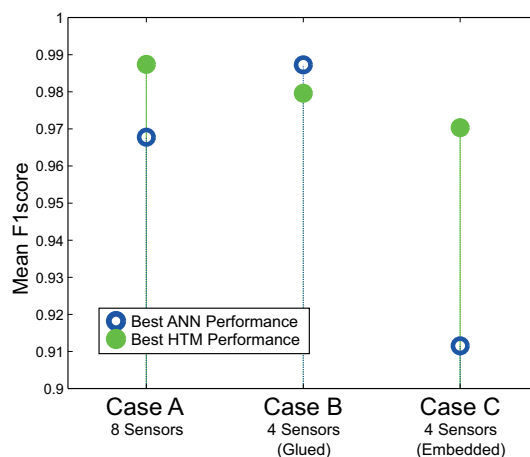


Figure 8: Best ANN and HTM performance achieved for each case

5.2 Case B: Feeding with the 4 surface glued sensors data

The most sensitive sensors are still used for the classification task. However, the amount of redundant data coming from the embedded sensors is reduced, but with the higher sensitive sensor samples there are enough data for the classification problem. The ANN algorithm is directly fed with the 4-dimension vectors. On the contrary, for the HTM the same 4-dimension input vector is repeated reversed to give extra information allowing more complex architectures to find the spatial patterns. This step has been skipped on the ANN because no improvement is obtained by repeating the same data. The best performance with the ANN approach is achieved for this scenario, as shown in Fig. 8. In this case, the most sensitive data from the glued sensors have been used having enough points of view of the impact state and reducing the noise data coming from the embedded. The best achieved F1score is 0.987 with 2 hidden layers of 30 nodes each one and by establishing the threshold to 0.5. In the case of the HTM, there are still very meaningful data coming from the 4 glued sensors but the number of inputs is very low to get advantage of the spatial inference of this approach. However, HTM seems to work pretty well by achieving a F1score of 0.979 with complex architecture such as 8-8. With a 8-8 architecture the spatial patterns are mostly discarded to benefit the temporal sequences (8 nodes in level 2). The other approach also exhibits a good performance: having into account the spatial patterns by using less nodes in level 1 with a simpler architecture (4-2). The obtained F1score is 0.970 for this architecture.

5.3 Case C: Feeding with the 4 embedded sensors data

In this case, the data are coming from the 4 embedded sensors, which have a lower strain sensitivity and higher noise. As in Case B, for the HTM tests, the same 4-dimensional input vector has been repeated in the input space in reverse order to allow architectures with more input nodes. Case C is the worst case for both algorithms because only the less sensitive and noisiest data are available. For a memory-less algorithm such as ANNs, working with less sensitive data is more complicated because it is not possible to separate the noise component from the truly information. By using ANN the best F1score achieved in this case is 0.911 with 2 hidden layers of 20 nodes each one and by establishing the threshold to 0.4. On the other hand, the HTM seems to work better than the ANN. The capability of this algorithm to manage temporal sequences is reflected on the obtained averaged F1score of 0.970. This performance has been achieved with an architecture of a

few nodes in the first level and more in the second one (4-8). By having few nodes in the first level, the spatial patterns can be grouped in order to create clearer temporal sequences with less noise which feed the second level, where the main classification task is performed over the temporal sequences of the first level. In summary, with high sensitive and redundant data, both approaches exhibit a good performance, despite the errors caused by noisy redundant data on ANNs. With the data from the surface-glued sensors (higher sensitivity and less noise), both approaches still work well and the ANN works even better by not having the noise coming from the redundant data. Finally, with the data from the embedded sensors, the HTM still have a very good classification performance due to its memory capabilities, what can be noticed in the number of nodes of the second level.

6 Conclusion

In this work, two machine learning algorithms have been tested using a controlled scenario. A GFRP beam instrumented with optical fiber strain sensors has been hit 480 times at 4 different locations while some extra noise were added using an electromagnetic shaker. A high level temporal algorithm (Hierarchical Temporal Memories) has been compared to the well known Artificial Neural Networks to detect these impacts. Different ways of feeding the experimental data to both algorithms exhibit their benefits and disadvantages when dealing with noisy data. Several configurations of both algorithms have been tested to achieve their best performance. Particular benefits of each approach can be exploited in order to reduce the number of sensors and/or to locate them in critical positions when installed in a real structure. The obtained data have been employed to train and test several architectures of both approaches.

From the achieved results it can be concluded that the HTM exhibits a better performance when only partial data are employed. Specifically, when the signal to noise ratio is lower (i.e. only using the embedded sensors), the difference between the performance of both approaches increases, offering the HTM better results. On the contrary, when there are enough sensing points, simpler approaches such as ANNs are better to model the problem without a-priori knowledge.

This study suggests that HTMs offer an improved performance when dealing with noisy data, especially for scenarios with a reduced number of sensing points. It might be interesting to apply this approach to field data, for example from the structural monitoring of wind turbine blades using FBG technologies.

Acknowledgments

Special thanks to Roberto Perez Sierra for his valuable collaboration with the experimental works. This work has been supported by the project TEC2010-20224-C02-02 and grant AP2009-1403.

References

- Bang H, Park S, Kim D, Hong C and Kim C (2004) Impact monitoring in smart composites using stabilization controlled fbg sensor system. *Proceedings of SPIE* 5384: 279–86.
- Bobier B and Wirth M (2009) Content-based image retrieval using hierarchical temporal memory. *Proceedings of the 16th ACM international conference on Multimedia* ACM:925–928.
- Chow T and Cho SY (2002) Industrial neural vision system for underground railway station platform surveillance. *Advanced Engineering Informatics* 16(1): 73–83.
- Coelho C, Hiche C and Chattopadhyay A (2010). Machine learning approach to impact load estimation using fiber bragg grating sensors. *Proceedings of SPIE* 7648:764810.
- Dua R, Watkins S, Wunsch D, Chandrashekhara K and Akhavan F (2001) Detection and classification of impact-induced damage in composite plates using neural networks. *International Joint Conference on Neural Networks* 1:681–686.
- Fonseca DJ and Navarrese D (2002) Artificial neural networks for job shop simulation. *Advanced Engineering Informatics* 16(4):241–246.
- Frieden J, Cugnoni J, Botsis J and Gmür T (2011) Low energy impact damage monitoring of composites using dynamic strain signals from fbg sensors-part i: Impact detection and localization. *Composite Structures*. 94(2):438–445.
- George D and Hawkins J (2009) Towards a mathematical theory of cortical micro-circuits *PLoS computational biology* 5(10):1000532.
- Goutte C and Gaussier E (2005) Lecture Notes in Computer Science. In: Advances in Information Retrieval (Springer) *A probabilistic interpretation of precision, recall and F-score, with implication for evaluation*. Berlin Heidelberg, pp. 345–359.

- Hawkins J and Blakeslee S (2004) *On Intelligence*. Owl Books.
- Hill KO and Meltz G (1997) Fiber bragg grating technology fundamentals and overview *Journal of Lightwave Technology* 15(8): 1263–1276.
- Kirkby E, de Oliveira R, Michaud V and Manson J (2011) Impact localisation with fbg for a self-healing carbon fibre composite structure. *Composite Structures*. 94(1): 8–14
- Lopez-Higuera J, Rodriguez Cobo L, Quintela Incera A and Cobo A (2011) Fiber optic sensors in structural health monitoring. *Journal of Lightwave Technology* 29(4): 587–608.
- Masri S, Smyth A, Chassiakos A, Caughey T and Hunter N (2000) Application of neural networks for detection of changes in nonlinear systems. *Journal of Engineering Mechanics* 126(7): 666–676.
- Mehrotra K, Mohan CK and Ranka S (1997) Elements of artificial neural networks. The MIT Press.
- Pomerleau DA (1991) Efficient training of artificial neural networks for autonomous navigation. *Neural Computation* 3(1): 88–97.
- Reich Y and Barai S (1999) Evaluating machine learning models for engineering problems. *Artificial Intelligence in Engineering* 13(3): 257–272.
- Rodriguez-Cobo L, Garcia-Allende PB, Cobo A, Lopez-Higuera JM and Conde OM (2012) Raw material classification by means of hyperspectral imaging and hierarchical temporal memories *Sensors Journal, IEEE* 12(9):2767–2775.
- Stolc S and Bajla I (2010) On the optimum architecture of the biologically inspired hierarchical temporal memory model applied to the hand-written digit recognition. *Measurement Science Review* 10(2): 28–49.