International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 1 | Issue – 6

Real-Time Data Representation Control in ConvolutionNeural Networks Based Indoor Wi-Fi Localization for Internet of Things

Dr. P. Srimanchari
Assistant Professor & Head, Department of Computer Applications, Erode Arts and Science College (Autonomus), Erode

Dr. G. Anandharaj
Assistant Professor & Head, Department of Computer Science and Applications, Adhiparasakthi College of Arts and Science, Vellore

ABSTRACT

The prediction of simultaneous limb motions is a highly desirable feature for the control of artificial limbs. In this work, we investigate different classification strategies for individual and simultaneous movements based on pattern recognition of myoelectric signals. Our results suggest that any classifier can be potentially employed in the prediction of simultaneous movements if arranged in a distributed topology. Compressive sensing has been successfully used for optimized operations in wireless sensor networks. However, raw data collected by sensors may be neither originally sparse nor easily transformed into a sparse data representation. This work addresses the problem of transforming source data collected by sensor nodes into a sparse representation with a few nonzero elements. Our contributions that address three major issues include: 1) an effective method that extracts population sparsity of the data, 2) a sparsity ratio guarantee scheme, and 3) a customized learning algorithm of the sparsifying dictionary. We introduce an unsupervised neural network to extract an intrinsic sparse coding of the data. As the underlying semiconductor technologies are getting less and less reliable, the probability that some components of computing devices fail also increases, preventing designers from realizing the full potential benefits of on-chip exascale integration derived from near atomic scale feature dimensions. As the quest for performance confronts permanent and transient faults, device variation, and thermal issues, major breakthroughs in computing efficiency are expected to benefit from unconventional and new models of computation, such as brain inspired computing. The challenge is then high-performance and energy-efficient, but also fault-tolerant computing solutions.

Keywords: *data representation, device variation, wireless sensor networks*

1. Introduction

In order to achieve simultaneous control using the direct mapping strategy, it would be enough to assign the MES of each muscle to its respective limb motion. However, this is practically impossible for several reasons, i.e., considerable myoelectric interference (crosstalk) is commonly found on superficial recordings and, inherently to an amputation, muscles are lost and so are the myoelectric control sites. The simultaneous control of two DoF using a direct scheme has been demonstrated by Kuiken*et al.* [1] in patients with targeted muscle re-innervations (TMR). This was achieved thanks to the TMR procedure which increases the number of independent control sites [2]. Unfortunately, even in TMR patients, it is not always possible to satisfactorily isolate MES in surface recordings, thereby admittedly making pattern recognition schemes desirable [2], [3].An alternative to the direct control scheme is the use of pattern recognition algorithms (classifiers) which map several inputs(mixed MES from different muscles) to several outputs(limb motions). Although this approach is potentially capable of providing simultaneous control,

most prosthetic research has focused on predicting individual motions which limits the control to a serial operation (a single motion at a time). A detailed review of prosthetic control has been provided by Scheme and Englehart [4].A sparsely-activated data (a few nonzero elements in a sample vector) may naturally exist for compressive sensing (CS) applications in wireless sensor networks (WSNs) such as the path reconstruction problem[1], indoor localization[2], and sparse event detection [3]. On the other hand, a sparse data representation cannot be easily induced in many other real-world contexts (e.g., in meteorological applications and environmental data gathering). In particular, noise patterns are usually presented in collected data from WSNs which greatly affect the performance of conventional sparsity-inducing (transformation) algorithms such as the Haarwavelet and discrete cosine transforms [4]. This motivates the quest for noise-robust and effective sparsity-inducing method for WSNs. One of the breakthroughs in recent deep learning paradigms for finding high level data abstractions is achieved by introducing sparsity constraints on data representations, e.g., the Kullback–Leibler divergence [5], rectifier function [6], and topographic coding [7]. These methods are introduced for extracting intrinsic features from the data in a similar way that the human brain does while encoding sensory organ data, e.g., the low percentage of spikes in a visual cortex [8].In particular, sparse deep learning methods generate sparse representations across training data for each single unit (i.e.,lifetime sparsity), and they neither guarantee sparsity for each input signal nor assert on the number of nonzero values in the sparse codes.

Derived from these observations, it is commonly claimed that the majority of neural network models, abstracted from biological ones, have built-in or intrinsic fault tolerance properties due to their parallel and distributed structure, and the fact that usually they contain more neurons or processing elements than the necessary to solve a given problem, i.e., some natural redundancy due to over provisioning. However, claiming such an equivalent fault tolerance only on the basis of rough architectural similarities therefore cannot hold true in general, especially for small size neural networks [9], [10]. Furthermore, the assessment of fault tolerance across different neural models still remains difficult to generalize, due to fault tolerance is network- and application-dependent, an inconsistent use of the principal concepts exists, and the lack of systematic methods and tools for evaluation across neural models. Computational studies have shown that neural networks are robust to noisy inputs and they also provide graceful degradation due to their resilience to inexact computations when implemented in a physical substrate. The tolerance to approximation, for instance, can be leveraged for substantial performance and energy gains through the design of custom low-precision neural accelerators that operate on sensory input streams [11]_[13]. However, in practice, a neural network has a very limited fault tolerance capability and, as a matter of fact, neural networks cannot be considered intrinsically fault tolerant, without a proper design. Furthermore, as a consequence of computation and information are naturally distributed in neural networks, error confinement and replication techniques, key to conventional fault tolerance solutions, cannot be applied directly so as to limit the error propagation when implemented in potentially faulty substrates. Obtaining truly fault tolerant neural networks is still a very attractive and important issue to obtain more biological plausible models, both for i) artificial intelligence based solutions, where, for instance, pervasive embedded systems will require smart objects fully merged with the environment in which they are deployed to cope with unforeseeable conditions[14]_[16], and ii) as a source to build reliable computing systems from unreliable components, as suggested by [17].Rooted on the neural paradigm computing systems might take advantage of new emerging devices at nano scale dimensions and deal both with manufacturing defects and transient faults as well [18], [19] and even considers faults/errors an essential and intrinsic part of the design. In this last direction, the robustness and the potential fault tolerant properties of neural models call for attention as permanent and transient faults, device variation, thermal issues, and aging will force designers to abandon current assumptions that transistors, wires, and other circuit elements will function perfectly over the entire lifetime of a computing system, relying mainly on digital integrated circuits [20][23].

2. Problem Formulation

Neural networks are claimed to have a built-in or intrinsic fault tolerance property mainly due to their distributed connectionist structure. Fault tolerance in a neural network is directly related to the redundancy introduced because of *spare capacity* (over-

provisioning), i.e., when the complexity of the problem is less than the raw computational capacity that actually the network can provide. Nevertheless, the analysis and evaluation of fault tolerance remain difficult because many different architectural and functional features under diverse conceptual frameworks are usually involved, and there are no common systematic methods or tools for evaluation. Technical and quantitative reasoning about these features calls for clear definitions, highlighting their similarities and differences, as those concepts appear indifferent contexts and areas of application.

A *fault* is an anomalous physical condition in a system that gives rise to an error. An *error* is a manifestation of a fault in a system, the deviation from the expected output, in which the logical state of an element differs from its intended value. A *failure* refers to a system's inability to perform its intended functionality or behavior because of errors in its elements or perturbations in its environment. Propagation of an error to the system level results in system failure, however, a fault in a system does not necessary result in an error or failure as it might go inactivated. A fault is said to be active when it produces an error; otherwise it is called dormant. Faults can be classified by their temporal characteristics as follows:

A **permanent** fault is continuous and stable with time; it is mainly a result of an irreversible physical damage.

A **transient** fault may only persist for a short period of time and it is often result of external disturbances. Transient faults, which recur with some frequency, are called **intermittent**. Usually, an intermittent fault results from marginal or unstable device operation and they are more difficult to detect than permanent ones. Transient and intermittent faults cover the vast majority of faults which occur in digital computing systems built with the current semiconduct or technology. Even, future implementation technologies are expected to suffer transient faults due to a reduced device quality, exhibiting a high level of process and environmental variations as well as considerable performance degradation due to the potential high stress of materials.

A. Classifiers and Classifier Topologies

There is a wide variety of fundamentally different pattern recognition algorithms, and although some of them are inherently capable of simultaneous classification (e.g., MLP), others are limited by their design to produce a single output (e.g., linear discriminant analysis). A mixture of these classifiers was evaluated in this work: Linear Discriminant Analysis (LDA) as a statistical classifier part of discriminant analysis; Multi-Layer Perceptron (MLP) as a supervised Artificial Neural Network (ANN); Self-Organized Map (SOM) as an unsupervised ANN and Regulatory Feedback Networks (RFN) as a new paradigm in classification based on negative feedback rather than learning.

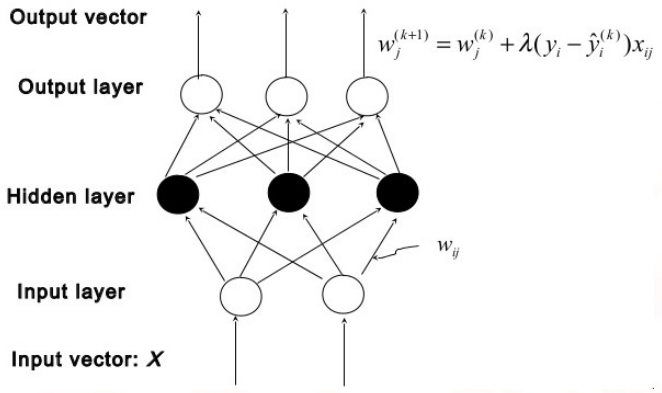


Figure: 2.1 A Multi-Layer Feed-Forward Neural Network

Although some of these algorithms compute the most likely pattern/class by majority voting (single output), they can be split into different topologies using dedicated classifiers, thus enabling simultaneous predictions (mixed outputs). One way of achieving mixed outputs is the creation of several binary classifiers, which is known as problem transformation by *binary relevance*. The following topologies (transformations) were used in this study.

1) *Single:* This is the simplest and most commonly used topology, where all inputs feed a single classifier which is trained to discriminate all labels. In order to be used for simultaneous predictions, the classifier is also fed with information relating to the mixed classes during the training/learning process. Although the number of outputs remained the same as that of the individual classes, simultaneous prediction is possible because more than one output can be activated in parallel.

2) *All Movements as Individual (AMI):* Similar to a single classifier but applies the *label power set* problem transformation method, which means creating a new label for each mixed movement. The

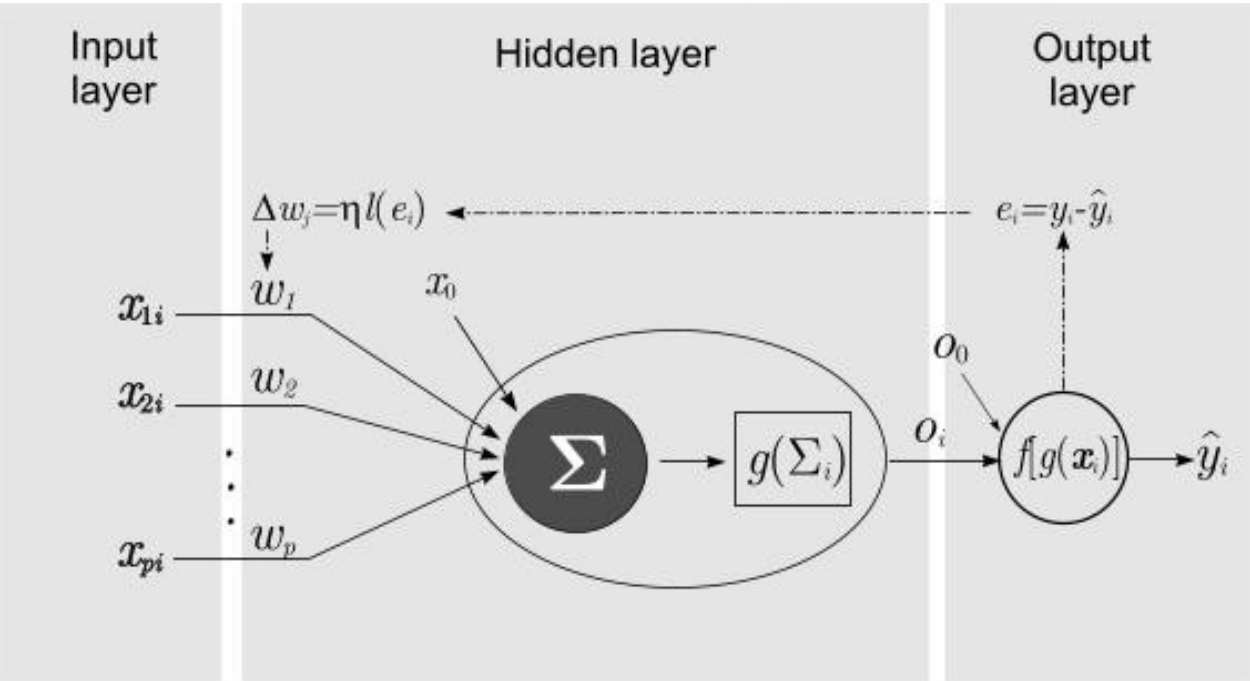
number of outputs is there for expanded to the total number of classes. In this case, only one output is activated at a time.

3) ***Ago/Antagonist-Mixed (AAM):*** This topology assumes that the motions are paired in ago/antagonist movements (essentially a DoF). There are as many classifiers as DoF and each classifier is fed with the feature vectors of at least three classes; two of them are the movements related to the DoF, and the third is a mixed class combining all the other movements. A fourth class is optional if the *rest* class (no motion) is available. The output vector contains the winner motion from each classifier, It is formed by classifiers, considering that there is a *rest* class.

The number of movements for the individual classification study was 11 (hand open/close, wrist flexion/extension, pro/supination, side grip, fine grip, agree or thumb up, pointer or index extension, and *rest*), recorded by four equally spaced bipolar electrodes. The movements involved in the simultaneous study were hand open/close, wrist flexion/extension and pro/supination, plus all their possible combinations—that is three DoF with six individual and 20 mixed motions for a total of 27 classes (considering the *rest* class) recorded using eight bipolar electrodes. These movements were selected because they are currently feasible using commercially available prosthetic devices. A recording session for simultaneous movements lasted 10.4 min, and information from all classes was used to train the classifiers.

3. Fault-Tolerant Neural Networks

Fault tolerance in neural networks depends on the definition of the acceptable degree of performance and its intended application at a high level of abstraction, fault tolerance, within neural models, might be analyzed by the effects of errors in the main operators that support the whole neural computational task, rather independent from their intended physical implementation. In fact, this has been the practice in most works reported in the literature. In a more comprehensive and structured approach, after this initial step, physical faults affecting a specific implementation can be mapped onto such errors so that the expected fault tolerance of a given architectural implementation of a neural model can be estimated, and further by identifying critical neural components, complementary and ad-hoc fault tolerance policies can be further applied to enhance the properties of the neural model implementation. In neural networks, an error model can be defined depending only on the neuron behavior itself, rather independent of its physical implementation, which is usually targeted to a digital substrate, so as to estimate the influence of faults on the neural computation from the initial design stages.



- As unexpected values of signals in the communication channels due to faulty interconnections or noise.
- In the synaptic weight or the associated computation, this in the absence of implementation details can be considered as indistinguishable.
- In the neuron body itself, affecting the summation or the evaluation of the nonlinear activation function.

The stuck-at model essentially allows investigating fault tolerance at the behavioral level, independently of the actual implementation or detailed characteristics of physical faults. It abstracts and simplifies faults into stuck-at values affecting single components. Such an abstraction has been widely used in testing of digital circuits and has proved to be sufficient to model a large number of physical faults. Some other faults/errors can be even considered for neurons but they can mask each other in the sense that it can be undistinguishable which fault occurred, for instance a fault in the synaptic operation itself (multiplication) instead of a fault in the weight storage. Considerations on physically realistic fault models for analog VLSI neural networks are also needed. Among the most important works reported in the literature regarding fault models in neural networks, mostly feed forward multilayer networks, are the following. Sequin and Clay used a bottom-up approach to categorize the types of faults that usually might occur in neural networks looking at the main components that comprise a network and focusing in fault cases that yield a worse effect on the overall performance of the network.

4. Taxonomy of Fault Tolerance

A general but widely adopted frame is to classify fault tolerance as *passive* or *active*, based on the principles and mechanisms that they exploit to achieve fault tolerance and particularly emphasized for fault tolerance in neural models. We follow this frame in reviewing the literature related to neural networks fault tolerance, and we principally focus on methods and techniques to enhance fault tolerance passively. Nonetheless, other important works on fault tolerance in neural networks are also briefly referred throughout this review. An empirical study of the influence of the activation function on fault tolerance properties of feed forward neural networks is presented, showing that the activation function largely has relevance on fault tolerance and the generalization property of the network. Review some representative works on active fault tolerance in neural networks. Before going into details, it is worth to point out that the majority of reported works has been focused in feed forward neural networks and few attempts have been made to improve fault tolerance in some other neural models. Works that discuss and analyze fault tolerance of non-feed forward neural networks, even though some works do not propose any specific technique for fault tolerance improvement. This issue is of importance since the studies and results in the literature concerned with fault tolerance in feed forward neural networks, despite of its importance (e.g. for deep learning), are difficult to generalize and directly apply across other different neural models.

4.1 Passive Fault Tolerance

In the proposed taxonomy, as schematically the reviewed works are classified based on their main strategies to achieve or improve fault tolerance in the recall stage of neural networks without considering retraining, i.e., we mainly focus on passive fault tolerance. Since only passive fault tolerance is considered in depth herein, the main mechanisms to provide the needed redundancy or fault masking to enhance fault tolerance will be presented. Each technique is explained based on its characteristics, design objectives, and the considered fault types in the performed study. Three main categories, in the passive approach, are identified, which group together related methods and techniques to enhance the intrinsic fault tolerance capabilities of neural networks: i) explicitly augmenting redundancy, ii) modifying learning/training algorithms, and iii) neural network optimization with constraints.

5. Results

In order to appreciate the detection quality of the different options, and considering the highly unbalanced distribution of service labels, we provide the following performance metrics for each option: accuracy, precision, recall, and F1. Considering all metrics, F1 can be considered the most important metric in this scenario. F1 is the harmonic mean of precision and recall and provides a better indication of detection performance for unbalanced datasets. F1 gets its best value at 1and worst at 0.All the activation functions were Rectified Linear Units(ReLU) with the exception of the last layer with Softmax activation. As expected, in general, the more features render better

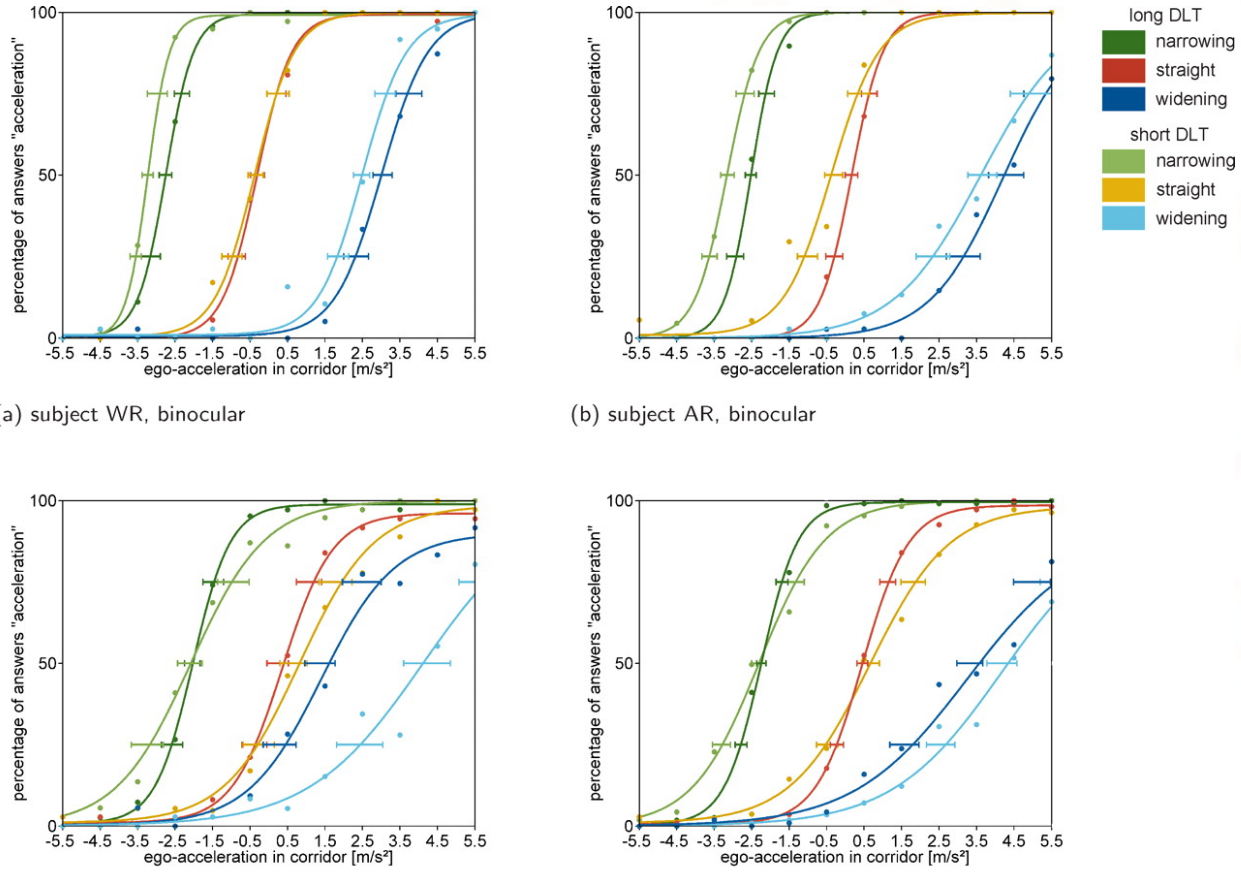
results. But, interestingly the packets inter-arrival time (TIMESTAMP) gives slightly worse results when added to the full features set. It seems it provides some not well aligned information with the source and destination ports, because, as soon as we take away the source and destination port, it is clear it becomes again an important feature. It is also interesting to appreciate the importance of the feature TCP window size (WIN SIZE), being more important than TIMESTAMP when operating with a reduced set of features.

Table: 5.1 Adaptive Motion Data Representation

Motion	Original			The proposed method			Clip-based PCA with IK [3]			PGA-IK [13]		
	Defect	Natural.	Faithful.	Defect	Natural.	Faithful.	Defect	Natural.	Faithful.	Defect	Natural.	Faithful.
Mix set	6.48	3.56	–	**5.81**	**3.61**	**3.81**	7.48	3.39	3.54	–	–	–
09_06	0	4.86	–	0	4.86	4.86	0	4.86	4.86	0	4.86	4.86
17_08	11.62	2.54	–	1.85	4.31	4.46	2.77	4.08	**4.77**	**1.31**	**4.38**	4.35
15_04	12.14	3.71	–	8.07	3.75	**3.96**	7.79	3.11	3.21	**6.14**	**4.11**	3.18
85_12	4.79	**4.14**	–	**4.21**	3.96	**3.93**	7.14	3.57	3.5	4.57	3.64	3.82
17_10	8.29	3	–	**1.14**	**4.64**	**4.57**	3.21	3.89	**4.57**	5.29	3.29	4.14

The mix data set contains eleven clips of walking, dancing, jumping, sporting motions from dataset in Table 1 and 2. Others are motions in Table 3. "Defect" represents the number of visible defects noticed by subjects; "natural." represents the score of naturalness (from 1 to 5. 1: very poor, 3: acceptable, 5: very satisfactory); "faithful." represents the score of faithfulness (from 1 to 5).

The Motion Test for simultaneous movements was performed by ten subjects using the MLP classifier in Single, OVA, and AAM topologies in order to investigate whether differences in real-time prediction exist despite their practically identical offline accuracy No difference was found between the Single and AMM topologies when subjects were asked about their perceived performance. Conversely, they all reported that OVA was the least stable, which is in line with its poor performance on this test. On the other hand, the Single topology for MLP was chosen as the simultaneous strategy because of its simplicity and since no considerable difference was found compared with AAM.



(a) subject WR, binocular

(b) subject AR, binocular

(c) subject FF, binocular

(d) all subjects, binocular

6. Conclusion

This problem is currently being addressed using different, but not always mutually exclusive, approaches such as conforming dry surface electrodes, TMR, and in our case, implanted neuromuscular interfaces permanently communicated through an Osseo integrated implant. In any case, if these controllers are to be used, this work suggests that simultaneous control must be considered, as it increases the overall controllability without considerable increment incomplexity. In this work, we have introduced a sparsity-inducing algorithm for data aggregation of non-sparse signal in wireless sensor networks. The proposed method consists of three steps: data collection, offline training and modelling, and online sparse code generation. The modelling scheme is based on a neural network with three layers, where the sparse codes are exposed at the hidden layer's neurons. A cost function is introduced as a sparsity nomination scheme. Then, a shrinking mechanism is used to switch off the least dominant neurons in the hidden layer, while asserting on the number of generated nonzero values in the sparse code. The resulting scheme can be used in many applications such as in compressive sensing based data aggregation schemes. For future research, we will analytically study the energy consumption and computational burdens of the proposed scheme.

REFERENCE

1) T. Kuiken, G. Dumanian, R. Lipschutz, L. A. Miller, and K. Stubblefield,"The use of targeted muscle reinnervation for improved myoelectricprosthesis control in a bilateral shoulder disarticulation amputee,"*Prosthet. Orthot.Int.*, vol. 28, no. 3, pp. 245–253, 2004.

2) T. Kuiken, G. Li, B. A. Lock, R. D. Lipschutz, L. A. Miller, K. A.Stubblefield, and K. B. Englehart, "Targeted muscle reinnervation forreal-timemyoelectric control ofmultifunction artificial arms," *J. Amer.Med. Assoc.*, vol. 301, no. 6, pp. 619–628, 2009.

3) P. Zhou, M. M. Lowery, K. B. Englehart, H. Huang, G. Li, L. Hargrove,J. Dewald, and T. Kuiken, "Decoding a new neural machineinterface for control of artificial limbs," *J. Neurophysiol.*, vol. 98, no.5, pp. 2974–2982, Nov. 2007.

4) E. J. Scheme and K. Englehart, "Electromyogram pattern recognitionfor control of powered upper-limb prostheses: State of the art and challengesfor clinical use," *J. Rehabil. Res. Dev.*, vol. 48, no. 6, p. 643,2011.

5) P. Herberts, C. Almström, R. Kadefors, and P. D. Lawrence, "Handprosthesis control via myoelectric patterns," *Acta. Orthop. Scand.*, vol.44, no. 4, pp. 389–409, Jan. 1973.

6) D. Yatsenko, D. McDonnall, and K. S. Guillory, "Simultaneous,proportional, multi-axis prosthesis control using multichannel surfaceEMG," in *29th Ann. Int. Conf. IEEE EMBS*, Lyon, France, Aug.23–26, 2007, pp. 6134–6137.

7) N. Jiang, K. B. Englehart, and P. Parker, "Extracting simultaneous andproportional neural control information for multiple-DOF prosthesesfrom the surface electromyographic signal," *IEEE Trans. Biomed.Eng.*, vol. 56, no. 4, pp. 1070–1080, Apr. 2009.

8) S. Muceli and D. Farina, "Simultaneous and proportional estimationof hand kinematics from EMG during mirrored movements at multipledegrees of freedom," *IEEE Trans. Neural. Syst. Rehabil. Eng.*, vol. 20,no. 3, pp. 371–378, May 2012.

9) N. Jiang, J. L. Vest-Nielsen, S. Muceli, and D. Farina, "EMG-basedsimultaneous and proportional estimation of wrist/hand dynamics inuni-lateral trans-radial amputees," *J. NeuroengineeringRehabil.*, vol.9, p. 42, 2012.

10) B. A. Lock, K. Englehart, and B. Hudgins, "Real-time myoelectriccontrol in a virtual environment to relate usability vs. accuracy," in*Proc. MyoElectric Controls/Powered Prosthetics Symp.*, Fredericton,Canada, Aug. 17–19, 2005.

11) G. Li, A. E. Schultz, and T. Kuiken, "Quantifying pattern recognitionbasedmyoelectric control of multifunctional transradial prostheses,"*IEEE Trans. Neural. Syst. Rehabil. Eng.*, vol. 18, no. 2, pp. 185–192, Mar. 2010.

12) E. J. Scheme, K. B. Englehart, and B. S. Hudgins, "Selective classificationfor improved robustness of myoelectric control under nonidealconditions," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 6, pp. 1698–1705, Jun. 2011.

13) M. Ortiz-Catalan, R. Brånemark, and B. Håkansson, "BioPatRec: Amodular research platform for the control of artificial limbs based onpattern recognition algorithms," *Source Code Biol. Med.*, vol. 8, p. 11,2013.

14) A. M. Simon, L. J. Hargrove, B. A. Lock, and T. Kuiken, "Targetachievement control test: Evaluating real-time myoelectric pattern-recognition control of multifunctional upper-limb prostheses," *J.Rehabil. Res. Dev.*, vol. 48, no. 6, pp. 619–628, 2011.

15) L. J. Hargrove, K. Englehart, and B. Hudgins, "A comparison of surfaceand intramuscular myoelectric signal classification," *IEEE Trans.Biomed. Eng.*, vol. 54, no. 5, pp. 847–853, May 2007.

16) W. J. Krzanowski, *Principles of Multivariate Analysis: A User's Perspective*.New York, NY, USA: Oxford Univ. Press, 1988.

17) S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed.Upper Saddle River, NJ, USA: Prentice Hall, 1999.

18) G. Qu, H. Zhang, and C. Hartrick, "Multi-label classification withBayes' theorem," in *Proc. 4th Int. Conf. Biomedical Engineering andInformatics (BMEI)*, 2011, pp. 2281–2285.

19) G. Tsoumakas and I. Katakis, "Multi-label classification: Anoverview," *Int. J. Data Warehous. Min.*, vol. 3, no. 3, pp. 1–13,2007.

20) L. J. Hargrove, E. J. Scheme,K. B. Englehart, and B. S.Hudgins, "Multiplebinary classifications via linear discriminant analysis for improvedcontrollability of a powered prosthesis," *IEEE Trans. Neural. Syst. Rehabil. Eng.*, vol. 18, no. 1, pp. 49–57, Jan. 2010.

21) N. Garc´ıa-Pedrajas and D. Ortiz-Boyer, "Improving multiclass patternrecognition by the combination of two strategies," *IEEE Trans. PatternAnal. Mach. Intell.*, vol. 28, no. 6, pp. 1001–1006, Dec. 2006.

22) M. Ortiz-Catalan, R. Brånemark, and B. Håkansson, "Evaluation ofclassifier topologies for the real-time classification of simultaneouslimb motions," in *Proc. 35th Ann. Int. Conf. IEEE EMBS*, Osaka,Japan, Jul. 3–7, 2013.

23) M. Ortiz-Catalan, BioPatRec [Online]. Available: http://code.google.com/p/biopatrec Jan. 2014

24) R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithmfor bound constrained optimization," SIAM J. on Scientific Computing,vol. 16, no. 5, pp. 1190–1208, 1995.

25) R. Kohavi et al., "A study of cross-validation and bootstrap for accuracyestimation and model selection," in Proc. Int. Joint Conf. Artificial Intell.,vol. 14, no. 2, 1995, pp. 1137–1145.

26) "Sensorscope: Sensor networks for environmental monitoring," 2007.[Online]. Available: http://lcav.epfl.ch/op/edit/sensorscope-en

27) P. Domingos, "A few useful things to know about machine learning,"Commun.of the ACM, vol. 55, no. 10, pp. 78–87, 2012.

28) Y. Bengio, "Practical recommendations for gradient-based training ofdeep architectures," in Neural Networks: Tricks of the Trade. Springer,2012, pp. 437–478.

29) H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse codingalgorithms," in Proc. 19th Advances in Neural Inform. Process. Syst.,2006, pp. 801–808.

30) J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learningfor sparse coding," in Proc. 26th Annu.Int. Conf. Machine Learning.ACM, 2009, pp. 689–696.

31) F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion,O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al.,"Scikit-learn: Machine learning in python," The J. of Machine LearningResearch, vol. 12, pp. 2825–2830, 2011.

32) K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair,S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensornetwork data fault types," ACM Trans. Sensor Networks, vol. 5, no. 3,p. 25, 2009.

33) Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimized approximation algorithmin neural networks without overfitting," IEEE Trans. Neural Netw.,vol. 19, no. 6, pp. 983–995, 2008.