Sparse deep-learning algorithm for recognition and categorisation

K. Charalampous, I. Kostavelis, A. Amanatiadis and A. Gasteratos

Presented is a deep-learning method for pattern classification and object recognition. The proposed methodology is based on an optimised version of the hierarchical temporal memory (HTM) algorithm and it preserves its basic structure, along with a tree structure of connected nodes. The tree structured scheme is inspired by the human neocortex, which provides great capabilities for recognition and categorisation. The proposed method is enriched with more representative quantisation centres using an adaptive neural gas algorithm, and a more accurate and dense grouping by applying a graph clustering technique. Sparse representation using L_1 norm minimisation is embedded as a liaison between the quantisation centres and their grouping, reinforcing the proposed technique with advantages, such as a natural discrimination capability. The proposed work is experimentally compared with the aforementioned techniques as well as with state-of-the-art algorithms, presenting a better classification performance.

Introduction: In the last decades the field of pattern recognition and object classification has been enriched with a great variety of techniques, which attempt to provide a definite solution, most of them utilising structured mathematical models. However, reaching the human capabilities of recognising objects that have undergone complicated transformations, is a distant goal. Recently, a different approach which is inspired by the human brain structure has been proposed, suggesting a hierarchical structure of computational nodes [1]. Hierarchical temporal memory (HTM) proposes a merging between a bio-inspired hierarchical structure and well established mathematical models. This scheme can be applied in many different domains, such as in object recognition and classification, where the work in [2] yielded very promising results. The target application there was an object recognition one and, thus, the temporal information was irrelevant. Therefore, an alternative scheme concerning the part of the node that handles temporal information was suggested.

The methodology proposed in this Letter is a supervised machine learning technique that follows the HTM [1] notion but it also follows the idea to replace the temporal pooler of a node as well as to apply logpolar transformation to the initial input data, as in [2]. However, an adaptive version of neural gas [3] is chosen to quantise the input space, since the number of quantisation centres is not predefined, but depends on the percentage error of the input space representation. Furthermore, L_1 norm minimisation [4] is applied in order to transform the quantisation centres and express them as a linear combination of the input samples. Finally, the groups of the transformed quantisation centres are derived using the normalised cuts (Ncut) [5, 6] graph clustering algorithm. The resulting algorithm exhibits remarkable efficiency and improved performance.

HTM structure: The HTM is a network of nodes, placed in a tree-shaped hierarchy, divided into distinguished levels (Fig. 1). The number of nodes in level ν is $2^{2\lambda-\nu}$, where λ is the total number of levels. Data flow follows a bottom-up manner, starting from level 0 and then is fed to the next level in $n \times n$ patches, i.e. a node in level ν possesses children in level $\nu - 1$ and in turn, is a child of a node in level $\nu + 1$.

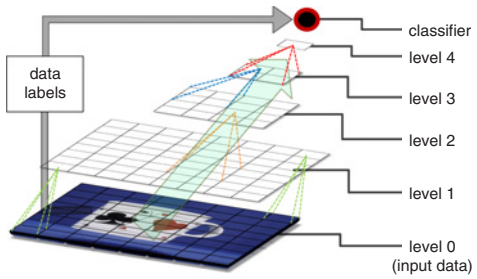

Fig. 1 *HTM structure*

Proposed node: A node in HTM is the fundamental computational unit, since it possesses the ability to process and store data. Initially, the spatial module operates in learning mode, seeking to learn an infinite space using a finite number of quantisation centres. This is achieved by a modified version of Neural Gas, where the number of quantisation centres increases until the percentage error of the space representation meets a predefined threshold T_g . As a result, a matrix $\mathbf{Q}_{M_0} \times N_0$ is extracted, where M_0 is the dimension of the input data and N_0 is the number of quantisation centres, i.e. each of its columns is a quantisation centre. Given the training data matrix $\mathbf{D}_{M_0} \times N_D$, where N_D is the number of data samples used in the adaptive Neural Gas, L_1 norm minimisation is applied on each quantisation centre, as follows:

$$\min_{\mathbf{r}_i} ||\mathbf{r}_i||_1 \qquad i = 1, 2, \ldots, N_0 \tag{1}$$

$$\text{s.t.} \quad ||\mathbf{q}_i - \mathbf{D}\mathbf{r}_i|| < \varepsilon \tag{2}$$

$$1 = \mathbf{1}^T \mathbf{r}_i \tag{3}$$

where $\mathbf{q}_i$ is the ith quantisation centre within the node, $\mathbf{r}_i$ is a sparse reconstruction weight vector having length N_D , ε is the error tolerance and $\mathbf{1}$ is a vector of ones also of length N_D. The sparse vector $\mathbf{r}_i$ results from L_1 minimisation and represents the corresponding quantisation centre $\mathbf{q}_i$ as a linear combination of the training data $\mathbf{D}$. Both minimisation constrains attain very important geometric properties: the first one in (2) ensures that $\mathbf{r}_i$ is invariant to rotation and scaling, while the second one in (3) ensures that $\mathbf{r}_i$ is invariant to translation. The weight vectors form the matrix $\mathbf{R}_{N_D \times N_0} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{N_0}\}, i = 1, 2, \ldots, N_0$, which is fed to the Ncut graph clustering algorithm, so that the transformed quantisation centres are grouped. The number of these groups is not strictly defined. Since the quality of every cluster is described via a corresponding eigenvalue, the number of groups will expand until an eigenvalue of a group reaches a predefined threshold T_c. By the conclusion of this procedure, an indicator j declares the group where each original quantisation centre belongs, i.e. $\mathbf{Q}_{M_0 \times N_0} = \{\mathbf{q}_1^j, \mathbf{q}_2^j, \ldots, \mathbf{q}_i^j, \ldots, \mathbf{q}_{N_0}^{N_c}\}, i = 1, 2, \ldots, N_0, j = 1, 2, \ldots, N_c$, where N_c is the number of groups. Once the groups have been formed, the node can switch to inference mode. Given a test data matrix $\mathbf{T}_{M_0 \times N_t}$, where N_t is the number of test samples, L_1 norm minimisation is applied, as follows:

$$\min_{\mathbf{o}_i} ||\mathbf{o}_i||_1 \qquad i = 1, 2, \ldots, N_t \tag{4}$$

$$\text{s.t.} \quad ||\mathbf{t}_i - \mathbf{Q}\mathbf{o}_i|| < \varepsilon \tag{5}$$

$$1 = 1^T \mathbf{o}_i \tag{6}$$

where $\mathbf{o}_i$ is the sparse reconstruction weight vector of length N_0, representing the corresponding test sample $\mathbf{t}_i$ as a linear combination of the quantisation centres $\mathbf{Q}$ and forms the matrix $\mathbf{O}_{N_0 \times N_t} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{N_t}\}$. Considering that the quantisation centres have been grouped, not only the contribution of each individual centre is available, but also the contribution of the respective group. Finally, a vector $\mathbf{a}_i$ of length N_c is formed for every test sample $\mathbf{t}_i$, where $\mathbf{a}_i(j) = mean\{\mathbf{o}_i^j(1), \mathbf{o}_i^j(2), \ldots, \mathbf{o}_i^j(N_0)\}, \forall j \in [1, 2, \ldots, N_c]$, implying that every element of the vector $\mathbf{a}_i$ holds the average value of the contributions of the quantisation centres, belonging to the same group j. As a result, the matrix $\mathbf{A}_{N_c \times N_t} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{N_t}\}$ is formed. Both learning and inference modes of an arbitrary node are shown in Fig. 2.

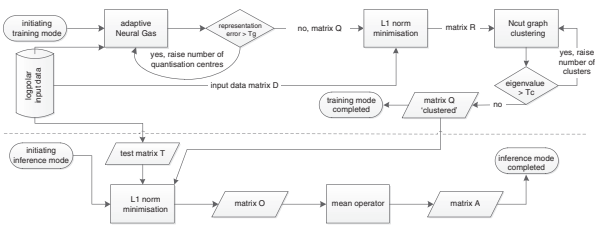

Fig. 2 *Proposed node*

Experimental results: The experimental validation was performed on the ETH-80 dataset, which is a collection of 256 by 256 pixel coloured images, belonging to eight different classes. Following the original HTM methodology, all images were linearly scaled to 32 by 32 pixels. Both in [2] and here, the logpolar transformation was applied, resulting in 32 by 32 pixel images. To assess the experimental results and the generalisation capacity of the examined techniques, the tenfold cross-validation was applied. Since the aforementioned methodologies are parametric models, the necessity of proper tuning to derive

the optimised results emerges. Algorithms in [1] and [2] share the same parameters, with the first one being the criterion to add new quantisation centres. The proper values (0.45 and 0.08, respectively) were determined using exhaustive search. The appropriate value for the learning rate parameter was derived in the same fashion and set to 0.1 for both [1] and [2]. In the proposed method the counterparts of these parameters are T_g and T_c. Both the parameters were determined through exhaustive search, leading to $T_g = 0.8$ and $T_c = 0.75$. In the top node of the hierarchy, two classifiers were assessed, namely the k-nearest neighbours (k-nn) and the linear support vector machines (SVMs). k-nn is a multi-class classifier and, thus, only one parameter needs to be fixed, i.e. the number of neighbours which was set to $k = 3$. Since the SVM is a binary classifier the one-vs-all approach was followed. C is defined as a penalty factor for non-separable points. Optimal classification results were yielded by setting $C = 1000$. Principal component analysis (PCA) + SVM was used as a reference framework to compare against the aforementioned techniques, which is also suitable for solving pattern recognition problems. In PCA, 95% of initial information was kept, and $C = 100$ was set to the corresponding SVM. The configuration of the dataset was the same as the one described in [1]. The derived results for both classifiers are shown in Tables 1 and 2, respectively.

Table 1: Classification accuracy using k-nn of original HTM, the one in [2] (optimised) and proposed method

Class	Original %	Optimised %	Proposed %
Apple	75.23	83.32	87.31
Car	78.46	89.51	91.08
Cow	76.17	82.50	84.16
Cup	79.56	89.96	90.94
Dog	77.32	84.12	84.51
Horse	72.45	81.23	85.57
Pear	79.11	84.31	89.63
Tomato	71.34	80.56	84.70
Average	76.21	84.44	87.23

Table 2: Classification accuracy using linear SVM of original HTM, that in [2], PCA and proposed method

Class	Original %	Optimised %	PCA %	Proposed %
Apple	85.12	93.35	85.67	96.28
Car	90.12	98.65	89.41	99.16
Cow	87.35	94.22	82.64	96.72
Cup	91.45	98.65	90.03	99.41
Dog	90.46	93.31	85.93	97.91
Horse	88.30	95.28	87.92	97.83
Pear	91.80	94.67	92.16	96.72
Tomato	86.12	93.46	82.84	97.57
Average	88.84	95.20	87.07	97.70

Conclusion: Presented is a deep learning method for pattern recognition and object classification, based on sparse representation. It exploits well-known machine learning and optimisation techniques, endowing the proposed algorithm with enhanced classification capabilities and, thus, outperforming its predecessors in categorisation tasks.

30 March 2012
doi: 10.1049/el.2012.1033
One or more of the Figures in this Letter are available in colour online.

K. Charalampous, I. Kostavelis, A. Amanatiadis and A. Gasteratos (*Laboratory of Robotics and Automation, Department of Production and Management Engineering, Democritus University Of Thrace, 12 Vas. Sofias Str, GR-67100, Xanthi, Greece*)

E-mail: kchara@pme.duth.gr

References

1 George, D., and Jaros, B.: 'The HTM learning algorithms', Whitepaper, Numenta Inc., 2007
2 Kostavelis, I., and Gasteratos, A.: 'On the optimization of hierarchical temporal memory', *Pattern Recognit. Lett.*, 2011, **33**, pp. 670–676
3 Martinetz, T., Berkovich, S., and Schulten, K.: '"Neural-gas". Network for vector quantization and its application to time-series prediction', *IEEE Trans. Neural Netw.*, 1993, **4**, pp. 558–569
4 Qiao, L., Chen, S., and Tan, X.: 'Sparsity preserving projections with applications to face recognition', *Pattern Recognit.*, 2010, **43**, pp. 331–341
5 Shi, J., and Malik, J.: 'Normalized cuts and image segmentation', *IEEE Trans. Pattern Anal. Mach. Intell*, 2000, **22**, (8), pp. 888–905
6 Cour, T., Yu, S., and Shi, J.: 'Normalized cut segmentation code', (Copyright) 2004 University of Pennsylvania, Computer and Information Science Department