Temporal Pooling Method for Rapid HTM Learning Applied to Geometric Object Recognition

[1,2]S. Štolc, [1,2]I. Bajla, [3]K. Valentín, [3]R. Škoviera

[1] Institute of Measurement Science, Department of Theoretical Methods,
Slovak Academy of Sciences, Bratislava, Slovakia

[2] Austrian Institute of Technology, GmbH,
Department Safety and Security, Seibersdorf, Austria

[3] Faculty of Mathematics, Physics, and Informatics, Department of Applied
Informatics, Comenius University, Bratislava, Slovakia

Abstract: *In the paper, we propose an alternative approach to the temporal pooling of the Hierarchical Temporal Memory (HTM) – a biologically inspired large-scale model of the neocortex. The novel method is compared with the conventional temporal pooling algorithm based on a smooth traversal of training images and their efficiency is demonstrated on a problem of the position, scale, and rotation-invariant recognition of simple geometric objects. Results have shown that the proposed method provides significantly faster convergence to the theoretical maximum classification accuracy than the conventional one.*

Keywords: Hierarchical Temporal Memory (HTM); Temporal Pooling; Rapid Learning; Image Explorer; Position, Scale, and Rotation-Invariant Pattern Recognition

1 Introduction

Hierarchical Temporal Memory (HTM), proposed by [1,2] and implemented as the free software package by Numenta, Inc. [3,4], is a recent development of an artificial intelligence network that excels at ambiguous pattern recognition problems. Promising results have been achieved in application of HTM to various pattern recognition/classification problems in machine vision, voice recognition, and other application areas [5,6]. These results usually refer to the situation in which the objects have fixed position and scale within the network's field of view (retina). The application of the HTM network to problems of the position, scale, and rotation-invariant recognition represents another challenging step in a development its functions. For the design of a successful classifier in this class of problems, characteristic by its extreme variability of a potential input, it is crucial to find an appropriate set of features, which are enough robust against expected object transformations. Moreover, it is very important to use an effective training algorithm, requiring neither too large training set nor too long training process.

In the HTM model, a key source of the invariance has been identified with the temporal learning, called the *temporal pooling*. The traditional approach to the temporal pooling [3,4] utilizes pattern sequences, called the *temporal sequences*, which are generated by a smooth traversal (exploration) of training images. This method often suffers from a slow convergence, causing that HTM requires a quite long training process to deliver reasonable results. In the paper, we propose an alternative approach to the temporal pooling that allows for a faster and more efficient training of the HTM networks. To demonstrate this ability, we conducted experiments with the position, scale, and rotation-invariant recognition of simple geometric objects, such as rectangles, triangles, or circles (represented as smoothed grey scale images) in which the efficiency of the both methods could be quantitatively compared.

2 Description of the HTM model

As proposed in [1,2,3], the HTM network forms a tree-shaped hierarchy of layers consisting of basic operational units called *nodes*. Each HTM node works in two modes – *learning* and *inference*. Within the learning phase the node performs two operations – *spatial pooling* and *temporal pooling*. Once these two steps are finished, the node can be switched to the inference mode. Although each node can be, in principle, trained by a different pattern sequence and contain its own spatial and temporal pooler data, a precondition of a successful position-invariant recognition is the equivalence of nodes at the same level. This, in fact, significantly simplify the whole training process, as only a single node needs to be trained at each network level and all the others share the same learned information.

Spatial pooling

In course of the spatial pooling, input patterns of each trained node are quantized into representative clusters. All clusters are characterized by their quantization center, which altogether form a codebook of spatial patterns approximating the input data. In the Numenta implementation of the spatial pooler, usually some kind of a smooth image explorer is used for collecting representative patterns. Their method depends strongly on an appropriate choice of the quantization parameter *maximum distance*, which specifies the size of the quantization clusters. Such an approach suffers from several weaknesses, which were addressed in [7]. In this study, we applied a different spatial pooling method that does not depend on any other parameters but the codebook capacity. This method randomly selects a required number of image patterns of a given size (e.g., 8x8 pixels) from provided training images and these patterns are afterwards considered as the quantization centers. In order to suppress appearance of irrelevant or empty patterns, the random selection is performed via the Metropolis-Hastings algorithm [8,9]. For running this algorithm, one needs to define two functions, the *pattern likelihood* and *pattern proposal* function. The pattern likelihood is a function of an image pattern X and accounts for assessing its relative relevance. For the purposes of this study, we used the following pattern likelihood function: $L(X) = (E(X^2))^k$, where $E(\cdot)$ is the arithmetic mean over the pattern intensity values, and k is a tunable constant (in our case, $k = 4$ worked very well). The proposed patterns were randomly sampled from the training images with coordinates uniformly distributed over the whole image extent.

Temporal pooling

In the temporal pooling step, the quantization centers are being grouped according to their temporal coherence within the training sequence of patterns. The resulting non-overlapping sets of codebook patterns are called *temporal groups*. The original HTM theory postulated several conditions, which are to be satisfied by any pattern sequence used for training HTM networks. The most crucial one is the condition of a smooth translation of objects within the network's retina, meaning that the position, rotation, scale, or illumination change smoothly in time[1]. Up to now, when dealing with the static images containing no inherent temporal information (unlike video streams), the temporal pooling has been accomplished by means of some sort of a smooth traversal of training images (e.g., along the horizontal lines or a smooth Brownian-like random walk), in course of which a sequence of image patterns is generated. Hereinafter, we refer to this type of the image exploration as the *smooth explorer*. The generated pattern sequence then serve for estimating the temporal statistics reflecting temporal coherences of the codebook patterns. The concept the smooth image exploration, however, can be implemented many different ways, which may end up in significantly different temporal groups in terms of their invariance. Usually the reason is that different temporal pooling approaches may provide differently accurate temporal statistics, though based on the same training data. As the temporal learning is most important factor influencing the invariance provided by HTM, an efficient temporal pooling algorithm is crucial for its functionality. The construction of a novel more efficient temporal pooler was the objective of our research and it will be described in details in Section 3.

Inference

In the inference mode, each HTM node produces a vector of beliefs for all memorized temporal groups, given arbitrary input pattern. The resulting belief vectors are then passed to the next network level, where they serve as inputs for superior HTM nodes. For calculating beliefs over the temporal groups, we applied principle of a strong lateral inhibition called *“winner-take-all”*. According to this system, only one temporal group in a time can be active, meaning that the winning temporal group receives belief of 1 and the rest belief of 0. The active temporal group is always the one, which contains the codebook pattern that is closest (in L_2 sense) to the current input pattern.

Supervised classification

Usually at the very top of the HTM hierarchy, there resides a supervised classifier responsible for assigning predefined object categories (classes) to the concatenated belief vectors coming from HTM nodes on the top most network level. In the papers, one can find HTMs combined with various supervised classifiers, e.g., KNN, SVM, or MLP. In this study, we considered a simple *nearest neighbor* (NN) approach, as our intention was to investigate qualities of two temporal pooling algorithms, regardless of capabilities of any employed supervised classifier. The NN method is most suited for such a task, as its generalization power merely depends on the organization of the input data.

3 Alternative Approach to the Temporal Pooling Providing More Accurate Temporal Statistics

As already suggested, the main drawback of the Numenta-like smooth explorer is its slow convergence to the theoretical maximum classification accuracy. When the problem domain is large, pattern sequences produced by the smooth explorer need to be rather long to capture the

[1] Be aware that such a condition does not imply generation of pattern sequences, which are smooth in the sense of Euclidean metric, i.e., L_2 distance between patterns appearing nearby in a sequence is not necessarily small.

data in its entirety, assuring sufficiently accurate temporal statistics[2]. When processing the training sequence, codebook patterns, which occur nearby in the training sequence (representing a virtual time), generate updates of the structure called the *time adjacency matrix* (TAM)[3]. In each training step, TAM is increased at the locations corresponding with the co-occurring codebook patterns according to the update function defined as follows [3,4]:

$$U(d) = \begin{cases} TM - d + 1 & \text{if } 0 < d \leq TM, \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

where $d \in \mathbb{N}$ is the temporal distance of the two patterns (i.e., the number of temporal transitions separating given two patterns in the training sequence) and TM is the Numenta parameter *transition memory*, which gives the maximum accepted number of temporal transitions.

To overcome weaknesses of the smooth explorer, we proposed a novel method of the temporal learning, the so-called *pair-wise explorer*. Instead of generating smooth random walks through images, our explorer performs the HTM training using pairs of relevant patterns sampled from a hypothetic infinite random walk, which crosses each image coordinate in each direction. These pairs of patterns are randomly sampled from the training images, so that distances $\hat{d}$ of their coordinates follow the probability distribution $P(\hat{d})$, which is proportional to the update function $U(d)$ (see Eq. (3.1)), given some reasonable conversion between $\hat{d}$ and d (e.g., $d = \lfloor \hat{d} \rfloor$). Each pair of patterns is considered as an extremely short temporal sequence, which produces exactly one TAM update having a constant influence and that ends immediately after processing the second pattern. Afterwards, the training can continue with processing another pair of patterns. The whole training is finished when requested number of TAM updates is performed.

When the temporal learning is over, the requested number of temporal groups is generated using the *agglomerative hierarchical clustering* (AHC). We have achieved good results with the *UPGMA*[4] linkage and the dissimilarity measure $D_{i,j}$ given as:

$$D_{ij} = \begin{cases} 1 - TAM_{ij} \Big/ \sqrt{\max\limits_{i}(TAM_{ij}) \max\limits_{j}(TAM_{ij})} & \text{if } i \neq j, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

4 Classification Experiments with Simple Geometric Objects

In order to demonstrate advantages of the pair-wise explorer over the smooth explorer, we conducted experiments with the position, scale, and rotation-invariant recognition of simple geometric primitives of three classes – circles, triangles, and rectangles. The objects were arbitrarily scaled, rotated and translated within the network's retina of 64x64 pixels. The

[2] Note that the overtraining effect does not apply to the temporal pooling. The longer training sequence is taken, the more stable and accurate temporal statistics is obtained.

[3] TAM is a square matrix, where each row and column corresponds to a single codebook pattern. Thus, each coordinate in TAM has a unique association with a particular pair of codebook patterns. In our experiments, TAM was always updated symmetrically.

[4] Unweighted Pair Group Method with Arithmetic Mean (UPGMA)

considered HTM network consisted of a single layer of non-overlapping nodes, each looking at the patch of 8x8 pixels. In accordance with the image size and types of used patterns, we set the codebook size to 512, the requested temporal group count to 64, and the transition memory to 4.

The classification accuracy was investigated with respect to two variable parameters. The first observed parameter was the number of training images varied from 10 to 300 per class, whereas the number of testing images was fixed at 300 per class. The second parameter was the number of TAM updates ranging between 1024 and 32768, which specified the length of training for the both temporal pooling methods and an equivalent basis. The classification accuracy was evaluated 10 times independently for each combination of the variable parameters and the average values have been used.

5 Results and Conclusions

In the performed classification experiments, the accuracy of NN classifier in the original space takes the values within the interval ⟨0.378, 0.701⟩.The classification accuracy for any of two HTM methods with sufficiently long training (i.e., sufficiently high number of TAM updates) increased up to 13.4 % in comparison with NN classifier (see Fig. 1, left). The proposed method of temporal pooling outperforms the conventional one with regard to both investigated parameters (i.e., the number of training images and the number of TAM updates). With the increasing number of TAM updates the difference between the two methods decreases asymptotically. For lower numbers of TAM updates within the interval ⟨2896, 4096⟩, our method yields the accuracy improvement of 1.104 in comparison with the conventional one. It corresponds to the cases for which the training sequence is quite short in the context of a given problem domain (see Fig. 1, right, and Tab. 1).

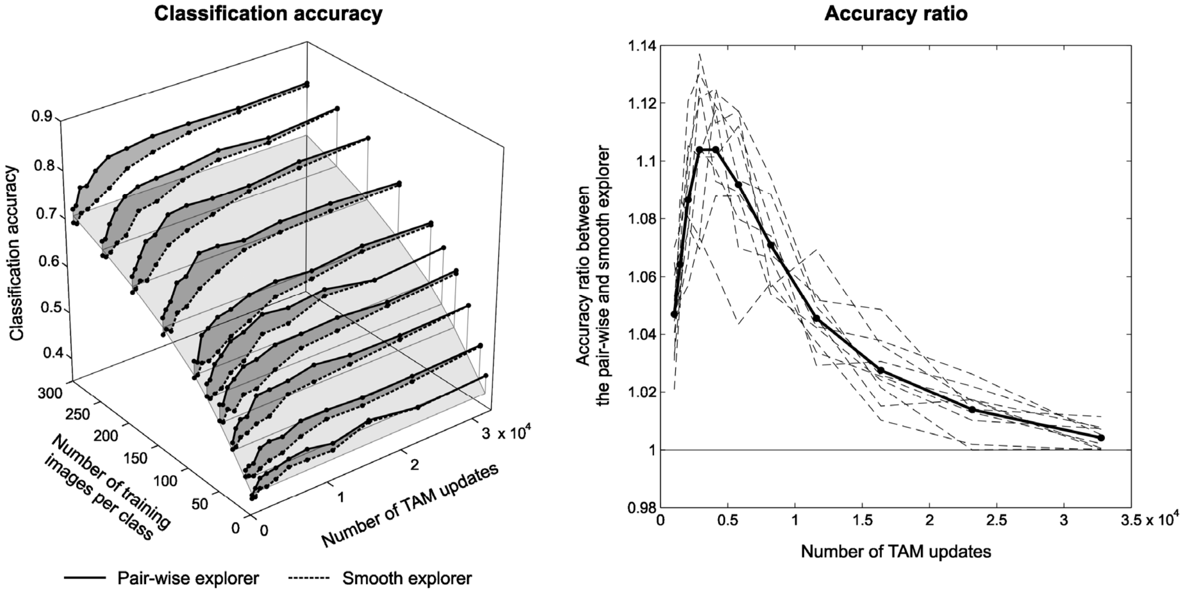

Fig. 1: (Left) The plot demonstrates that the pair-wise method outperforms the Numenta-like smooth method in terms of faster convergence mostly in the range of lower numbers of TAM updates. The gray surface represents the classification accuracy achieved by NN classifier performed in the original data space. (Right) The plot shows the classification accuracy gain of the pair-wise explorer over the smooth explorer.

Tab. 1: The actual values of classification accuracy ratio of the two temporal pooling methods, which have been averaged over variously sized training sets. The maxima of ratio are emphasized.

Number of TAM updates	1024	1448	2048	**2896**	**4096**	5793	8192	11585	16384	23170	32768
Mean accuracy ratio	1.047	1.064	1.086	**1.104**	**1.104**	1.092	1.071	1.046	1.028	1.014	1.004

The results have shown that, in contrast to the conventional method, the proposed novel temporal pooling method yields significantly faster convergence to the theoretical maximum classification accuracy with respect to both the length of the training sequence and the number of training samples. Therefore we suggest using this method instead of the conventional one, especially when dealing with complex large domain problems.

6 Acknowledgement

This work has been supported by the Slovak Grant Agency for Science (project No. 2/0019/10) and partially by the Foundation of Tatrabanka, through program E-talent for young scientists in the field of applied informatics (project No. 2010et019).

References

[1] Hawkins, J. and Blakeslee, S. (2004). *On intelligence.* Henry Holt and Company, New York.

[2] George, D. and Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLoS Computational Biology* 5(10). DOI 10.1371/journal.pcbi.1000532.

[3] Numenta (2008). Hierarchical Temporal Memory, concepts, theory, and terminology. Document version 1.8.0.

[4] Numenta (2009). Numenta node algorithms guide, NuPIC 1.7.

[5] Csapó, A., Baranyi, P., and Tikk, D. (2007). Object categorization using VFA-generated nodemaps and Hierarchical Temporal Memories. In: *5th IEEE Int. Conf. on Computational Cybernetics*, 257–261.

[6] Sassi,F., Ascari, L., and Cagnoni, S. (2009). Classifying human body acceleration patterns using a Hierarchical Temporal Memory. In: R. Serra & R. Cucchiara, eds. *AI*IA 2009: Emergent Perspectives in Artificial Intelligence*. Berlin, Heidelberg, 496–505.

[7] Štolc, 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 Revue* 10(2), 28–49. DOI 10.2478/v10048-010-0008-4.

[8] Metropolis, N., et al. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics* 21, 1087–1092.

[9] Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 97–109.