Refined Deep Learning for Digital Objects Recognition via Betti Invariants

Darian M. Onchis[1](✉), Codruta Istin[2], and Pedro Real[3]

[1] West University of Timisoara, Timisoara, Romania
darian.onchis@e-uvt.ro
[2] Politehnica University of Timisoara, Timisoara, Romania
istin.codruta@cs.upt.ro
[3] University of Seville, Seville, Spain
real@us.es

Abstract. In this paper, we make use of the topological invariants of 2D images for an accelerated training and an improved recognition ability of a deep learning neural network applied to digital image objects. For our test images, we generate the associated simplicial complexes and from them we compute the Betti numbers which for a 2D object are the number of connected components and the number of holes. These information are used for training the network according to the corresponding Betti number. Experiments on the MNIST databases are presented in support of the proposed method.

Keywords: Deep learning · Betti numbers · Handwritten digits

1 Introduction

The intelligent classification of data objects with human eye precision and speed or if possible even more accurate, is one of the current ambitions in computer analysis of images and patterns. From quantum algorithms to topological data analysis to the state of the art deep learning architecture, the human mind is trying to overcome the condition of the machines and if possible his own native condition [2–5,12].

In this context, the search for refined methods of classification involves designing new algorithms that could make use of the research developments in scientific cross-fields. In here, we merge the use of Betti numbers from computational topology with the training of a deep learning network to enhance the speed and the accuracy of the recognition for the handwritten digits.

Similar merges have been proposed in [4–6], but the areas of application are different from our approach. For example in [4], the authors propose a Topology-Net for biomolecular property predictions. In [5], the topological characteristics of digital models were employed for geological core and in [6] a geometric data analysis for image understanding is proposed and this analysis is based on manifold learning.

M. Vento and G. Percannella (Eds.): CAIP 2019, LNCS 11678, pp. 613–621, 2019.
https://doi.org/10.1007/978-3-030-29888-3_50

Our method is based on designing a first layer of the deep learning net used to feeding the data to the network. So the network will be trained with extra-information on the object to be discovered. As a concrete example, instead of training the network for the entire MNIST database, we train three networks for the corresponding three Betti numbers 0, 1 and 2 characterizing the number of holes. The digits 1, 2, 3, 4, 5, 7, have 0 holes then 0, 6 and 9 have 1 hole, and respectively the digit 8 has 2 holes. Smaller networks allow a more flexible and faster training and possibly even more accurate results.

We evaluate our constructive model on the MNIST database which contains 60,000 training and 10,000 test examples of handwritten digit images, each of which is 28 × 28 pixels (784-dimensional) and belong to one of 10 classes [1]. We randomly partition the original training set into training and validation sets with a ratio of 5 : 1.

Neural networks are widely used nowadays, but they have some drawbacks. A major one consists of the time spent to train the network, that is directly proportional to the accuracy of the results. We present in this work a different and novel approach that improves the accuracy and decreases the time spent for training the network. In order to do this, we make use of deep learning feed forward networks in which we alter the first layer of the network to act like a filter for the other layers that keep their standard role. The filtering is necessary for maintaining only the useful information in the database, such that the training will be performed optimally. For this, in the first step, we compute the boundaries of the objects and their holes and afterwards, we eliminate the impossible matching from the database that is not in our recognition interest.

The structure of the paper is the following: in the next section we introduce a theoretical brief about homology, topological spaces and Betti numbers, followed by a section with the presentation of the algorithm. In the last section, the final performances of our approach will be presented on the MNIST database, to emphasize the benefits stated by the theoretical algorithm.

2 Theoretical Brief

We describe briefly in this section the basis of the computational topology theory before applying them to the deep learning network.

We begin with homology which provides valuable information about topological spaces, by observing sets that intuitively have no boundary, but are on the boundary of other sets [13]. These sets are representative cycles of a homology hole, seen as an equivalence class. Algebraic homology information with coefficients in a field could be defined as the set of processed and structured linear algebraic data describing in some sense its (co)homology classes and the relations between them. We talk about homology and cohomology information as a whole due to the fact that homology and cohomology classes are measured using, up to dimension, the same algorithmic strategy of connectivity clustering over the initial topological data and they both provide the same measurable information quantities, up to "duality" [9,11]. A simple example of (co)homology information is provided by the numerical topological invariants called Betti numbers. If

X is a cell complex embedded in $\mathbf{R}^3$, Betti numbers β_0, β_1 and β_2 respectively measure the number of different connected components, (co)homological tunnels and cavities of X.

Roughly speaking, "homotopy holes" of objects (those related to generalized "parametrized and oriented closed curves") are theoretically attainable from homology's ones [10], but these methods have an enormous complexity in time and space [8]. An easier relation is given by the Euler-Poincaré characteristic (see [7]), defined in local terms as the alternate sum of the number of cells in each dimension. This number is the most simple example of homotopy invariant that can also be obtained from global homological information (Betti numbers).

A simplicial complex consists of a set of *vertices* and a set of *simplices* (over that set of vertices) satisfying some simple conditions that we now describe. We make no requirement on vertices other than that they admit a total order. An *n-dimensional ordered simplex*, or just *n-simplex*, over a set V of vertices is an $(n + 1)$-tuple of distinct vertices from V. (An ordered triangle is thus a 2-simplex.) An ordered simplex is *oriented* iff we do not distinguish between two orderings that differ by an even permutation (one that can be expressed as an even number of swaps). Oriented simplices therefore have a well-defined notion of "inside" and "outside".

An *n-dimensional, oriented, pure simplicial complex*, or just *n-complex* for short, is a pair $K = (V, S)$ where V is a set of vertices and S is a set of oriented simplices satisfying the following conditions:

1. Every vertex in V determines a 0-simplex in S. We usually do not distinguish between a vertex v and its associated 0-simplex (v).
2. Every sub-simplex of a simplex in K is also a simplex of K. That is, if $s \in S$ and $t \leq s$, then $t \in S$ (where $\leq$ is the ordered sub-simplex relation).
3. Every simplex $s \in S$ is a sub-simplex of some n-simplex in S. That is, there are no n'-simplices with $n' < n$ other than those that are faces of an n-simplex in S.

A *closed surface* is a 2-complex in which the link of every 0-simplex is a simple, closed polygon having the vertex as an interior point.

Now, we can build a continuous object out of the point cloud, hoping to get a good approximation of the original object. One way to do that is by centering a ball of fixed radius on each sample point. There are some smart techniques based on this idea: e.g., the construction of Vietoris-Rips or alpha complexes [14]. Out of these constructions, we can compute topological invariants; typically they are the dimensions of the homology modules at various dimensions k, called Betti numbers. Substantially, they count the numbers of k-cycles, i.e. connected components and voids in the object. The Betti numbers that one obtains depend on the radius of the balls. While varying the radius, there may be k-cycles which persist, and a good guess is that those may correspond to the true k-cycles of the sampled object.

3 Algorithm Description

The proposed algorithm makes use of the concrete relation between the computation of Betti numbers and the training of the neural network. Betti numbers already shown their use in computational topology, graphs theory and even in the determination of a neural network capacity [4–6]. But their use in the training of the deep learning networks was less explored. Therefore, we are considering in here the context in which a neural network is trained for object recognition in a 2D image. With the use of Betti numbers we can quickly discriminate during the training phase between the wheels of a car or bike and other objects with less or more number of holes. This information is useful for training as it does a separation of the objects to be recognized in smaller and more lighter neural networks.

We present below a complete and exhaustive algorithm description as we have implemented it.

```
1. Loading database
2. Dataset normalization of size $s \times (m \times n)$, image by image
3. Extract a point set from an image given as $m \times n$ matrix
4. Compute 2D simplicial complexes
5. Compute Betti numbers and determine the number of holes
   (a) Compute plot points
   (b) Compute vertices to form triangles
   (c) Determinate the number of unconnected elements
   (d) Determinate the number of holes
6. Send the image to the network corresponding to training the image with the
   same Betti number as the number of holes determined
7. Set beginning time for each network
8. Train the network
9. Set and compute end time for each network
```

For the MNIST database the maximum number of holes is 2 for number 8, so in order to have a concrete exemplification, we designed a network with 3 sub networks. We have designed and implemented a neural network as seen in Fig. 1. The first layer of the network computes the Betti numbers and depending on the number of holes determined, the image is sent to the corresponding network for training. The number of networks has to be equal to the number of holes that training images might have.

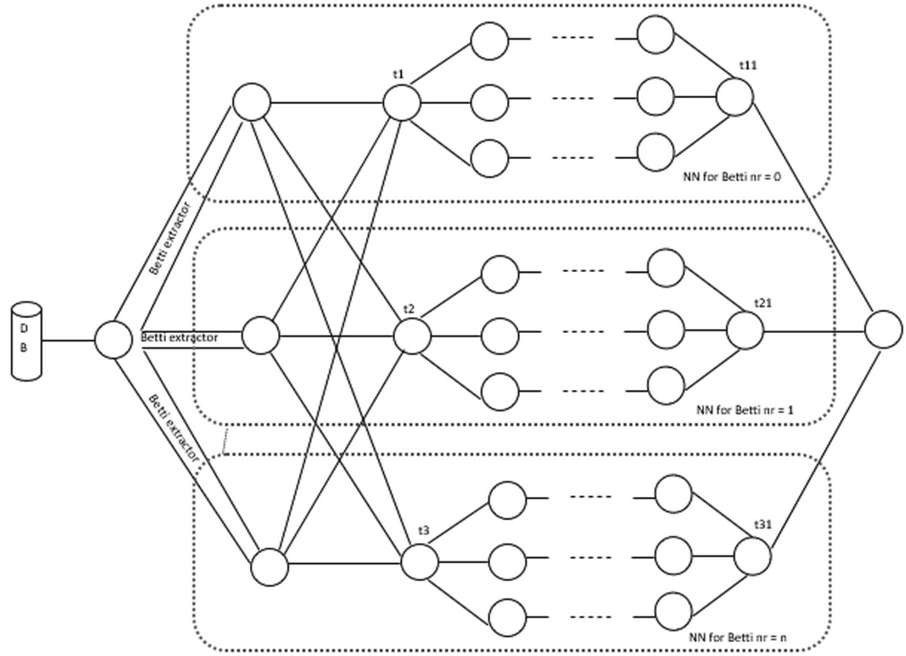
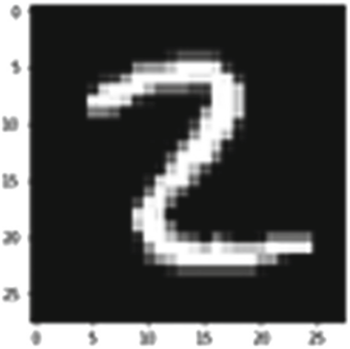
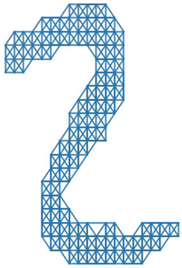


Fig. 1. Network architecture



Two in MNIST database



Two after simplicial complexes were computed

The experiment must compare the accuracy of each network after training therefore we performed multiple sets of independent trainings i.e. training one time each of the 3 networks is an element of the set. Each set of training can have a number of epochs that can vary between the set elements.

On our way to compute the Betti numbers, we determine the simplicial complexes, the number of unconnected components and the number of holes from an image.

Considering the image representing a 2 in the MNIST database, the simplicial complexes were computed and the output image can be seen in the second picture. As mentioned in the description of the algorithm, first, the points of every vertex were computed and then the triangles were formed. For this particular example, there are 314 points and 157 vertices. An example of points coordinates as they result from the PYTHON code looks like below:

```
[[ 0.14285714  0.42857143]
[ 0.14285714  0.46428571]
[ 0.14285714  0.5 ]
[ 0.14285714  0.53571429]
[ 0.14285714  0.57142857]
[ 0.17857143  0.32142857], etc].
```

For this image we get the Betti number 0 because we do not have any hole and we get the number of unconnected components 0, because the image has all the components connected. From the fact that the number has no holes, we can conclude that the possible numbers can be only 2, 3, 4, 5 and 7, because all the other numbers have at least one hole.

This topological computation is performed on the first layer of the neural network, due to the fact that depending on the result of the Betti numbers, the images are sent to the corresponding training sub network so both the accuracy and the training time are improved. The accuracy is enhanced and sometimes, as can be seen from Table 1, the errors can be eliminated even before the neural network training and the compilation time can be diminished by more than half, as the tests will show.

Table 1. Recognition percent

[7.] 7 0.99588446515	[0.] 0 0.961333326967
[2.] 2 0.865965937884	[6.] 6 0.619316634394
[1.] 1 0.988815228181	[9.] 9 0.955976339534
[0.] 0 0.985114050493	[0.] 0 0.989992193154
[4.] 4 0.989545536094	[1.] 1 0.994334666874
[1.] 1 0.993095777184	[5.] 5 0.894572800157
[4.] 4 0.968422037592	[3.] 3 0.89274171303
[9.] 9 0.809672938457	[9.] 9 0.964876201986
[5.] 6 0.376798989843	[7.] 7 0.995512296425
[9.] 9 0.935884560644	[4.] 4 0.996539732474

As it can be seen on line 9 of the recognition test, on a regular neural network, there is a mistake. It should have been five, but the network recognized a 6. By computing the Betti numbers first, this mistake would not have been possible to appear because five has ho hole, while 6 has one hole.

The Fig. 2 shows the initial picture from MNIST and the image after the topological computation of the simplicial complexes. Also the computation time for the simplicial complexes of each picture is shown.

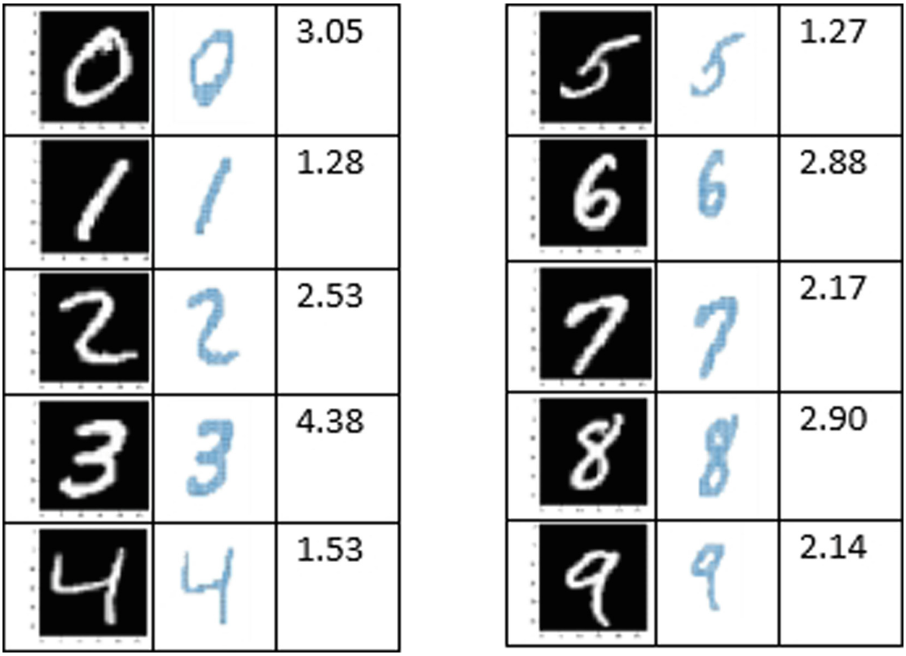

3.05
1.28
2.53
4.38
1.53
1.27
2.88
2.17
2.90
2.14

Fig. 2. Computational timings simplicial complexes

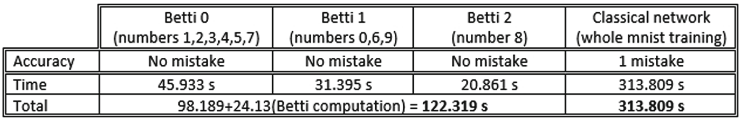
As it can be seen, the computation time for this images varies between 1.53 s and 4.38 s. The variation of time computation with respect to the number of vertices determined, will be discussed at Testing.

4 Testing

Experiments showed that by determining, for example the boundaries and number of holes from numbers, if a hole is found, than the number can be only 0, 6 or 9, so the training of the network could be done only with these numbers.

Integrating the topological computation to the first layer of the neural network has two important parts: the first one is actually computing the simplicial complexes and the second one is to send the picture to the corresponding sub network for training.

The tests were performed on a standard PC with Intel core I7 processor and 8G RAM. The comparison was made between a standard deep feed forward neural network trained with backpropagation and on the network showed in Fig. 1. Both time and accuracy were tested. For the consistency of the comparison we restricted the number of hidden layers to 3 in both the standard network and the proposed network on all the three sub networks. The results can be observed in Fig. 3.



	Betti 0 (numbers 1,2,3,4,5,7)	Betti 1 (numbers 0,6,9)	Betti 2 (number 8)	Classical network (whole mnist training)
Accuracy	No mistake	No mistake	No mistake	1 mistake
Time	45.933 s	31.395 s	20.861 s	313.809 s
Total	98.189+24.13(Betti computation) = **122.319 s**			**313.809 s**

Fig. 3. Total neural network timing

For computing the time for the proposed architecture, the time for each sub network was computed (see t1-t11, t2-t21, t3-t31) and also the time for Betti computation was considered. The difference between the proposed method and architecture overcomes the classical network by reducing the time to half. Also, the accuracy of our work did not give any mistakes, while the classical one misplaced a five with a six.

5 Conclusions

We introduced in here a novel method for improved recognition of objects in 2D images by merging computational topology invariants and deep learning neural networks. By splitting the full network into several small deep learning networks depending on the corresponding Betti numbers we have obtained a better accuracy in the recognition and also a faster training time. The next step is to try different numbers of layers for the small networks architecture and to validate the procedure also on noisier data sets.

Acknowledgement. The second author acknowledges the support of the Project HPC-EUROPA3 (INFRAIA- 2016-1-730897), with the support of the EC Research Innovation Action under the H2020 Programme.

References

1. http://yann.lecun.com/exdb/mnist/
2. Lloyd, S., Garnerone, S., Zanardi, P.: Quantum algorithms for topological and geometric analysis of data. Nat. Commun. **7**, 10138 (2016)
3. Cang, Z., Mu, L., Wei, G.W.: Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. PLoS Comput. Biol. **14**(1), e1005929 (2018)
4. Cang, Z., Wei, G.: TopologyNet: topology based deep convolutional and multi-task neural networks for biomolecular property predictions. PLoS Comput. Biol. **13**(7), e1005690 (2017)

5. Gilmanov, R.R., Kalyuzhnyuk, A.V., Taimanov, I.A., Yakovlev, A.A.: Topological characteristics of digital models of geological core. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) CD-MAKE 2018. LNCS, vol. 11015, pp. 273–281. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99740-7_19
6. Miranda Jr., G.F., Thomaz, C.E., Giraldi, G.A.: Geometric data analysis based on manifold learning with applications for image understanding. In: 2017 30th SIBGRAPI Conference On Graphics, Patterns and Images Tutorials (SIBGRAPI-T), SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing, Niteroi, Brazil, 17–20 October, pp. 42–62 (2017). Brazilian Comp Soc; UFRJ; PUC Rio; NVIDIA; IBM; Univ Fed Fluminense, Inst Computacao; Univ Fed Rio Janeiro, Programa Engn Sistemas Computacao; Pontificia Univ Catolica Rio aneiro, Dept Informatica; Univ Fed Rio Janeiro; ACM SIGGRAPH; CAPES; CNPq; SIBGRAPI 2017
7. Alexandroff, P.S.: Combinatorial Topology. Dover, New York (1998)
8. Cadek, M., Krcal, M., Matousek, J., Vokrinek, L., Wagner, U.: Polynomial-time computation of homotopy groups and Postnikov systems in fixed dimension. SIAM J. Comput. **43**(5), 1728–1780 (2014)
9. Díaz-del-Río, F., Real, P., Onchis, D.: A parallel homological spanning forest framework for 2D topological image analysis. Pattern Recogn. Lett. **83**, 49–58 (2016)
10. Hurewicz, W.: Homology and homotopy theory. In: Proceedings of the International Mathematical Congress of, p. 344 (1950)
11. Real, P., Diaz-del-Rio, F., Onchis, D.: Toward parallel computation of dense homotopy skeletons for nD digital objects. In: Brimkov, V.E., Barneva, R.P. (eds.) IWCIA 2017. LNCS, vol. 10256, pp. 142–155. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59108-7_12
12. Feichtinger, H.G., Onchis, D.M., Wiesmeyr, C.: Construction of approximate dual wavelet frames. Adv. Comput. Math. **40**(1), 273–282 (2014)
13. https://pi.math.cornell.edu/~hatcher/AT/AT.pdf
14. https://geometrica.saclay.inria.fr/team/Fred.Chazal/papers/CGLcourseNotes/main.pdf