ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

Stacked Denoising Autoencoders for the Automatic Recognition of Microglial Cells' State

Sofia Fernandes[1], Ricardo Sousa[1], Renato Socodato[2] and Luís Silva[1] *

1- Instituto de Investigação e Inovação em Saúde (i3S) and
Instituto de Engenharia Biomédica (INEB)

2- Instituto de Investigação e Inovação em Saúde (i3S) and
Instituto de Biologia Molecular e Celular (IBMC)
Universidade do Porto, Porto, Portugal.

Abstract. We present the first study for the automatic recognition of microglial cells' state using stacked denoising autoencoders. Microglia has a pivotal role as sentinel of neuronal diseases where its state (resting, transition or active) is indicative of what is occurring in the Central Nervous System. In this work we delve on different strategies to best learn a stacked denoising autoencoder for that purpose and show that the transition state is the most hard to recognize while an accuracy of approximately 64% is obtained with a dataset of 45 images.

1 Introduction

Microglia are the immune resident cells of the Central Nervous System (CNS). They are involved in normal brain physiology and homeostasis, regulating synaptic remodeling and neuronal tissue repair [1]. In a healthy CNS, microglia are considered to be in a resting state and display a ramified shape (see Fig. 1 A, D). Unlike the term "resting" may suggest, resting microglia are responsible for scanning its surrounding neuronal parenchyma in order to detect threats to the CNS. After the detection of a threat, microglial cells undergo dramatic shape changes, transiting from a resting-like into amoeboid activated morphology (a transformation process which is accompanied by the rounding of their cell bodies). Although microglia are also associated with neuroprotective functions, their sustained activation or overactivation can potentially result in neuronal cell loss [2]. Therefore, the identification of the transition state of microglia is an important tool for the diagnosis of such activation states both in acute and chronic brain diseases since subtle shape changes may anticipate microglia activation. Figure 1 shows some examples of collected images from each of the three microglial cell's states.

Until recently, the identification of a microglial cell's state via its morphology required the intervention of one or more experts (it should be noted that there are no clear boundaries between the states as the morphology of the cells is highly variant even within the same class; see Fig. 1 for some examples). To overcome this issue, we propose the application of deep neural networks, namely

*This work was financed by FEDER funds through the Programa Operacional Factores de Competitividade – COMPETE and by Portuguese funds through FCT – Fundação para a Ciência e Tecnologia in the framework of the project PTDC/EIA-EIA/119004/2010.

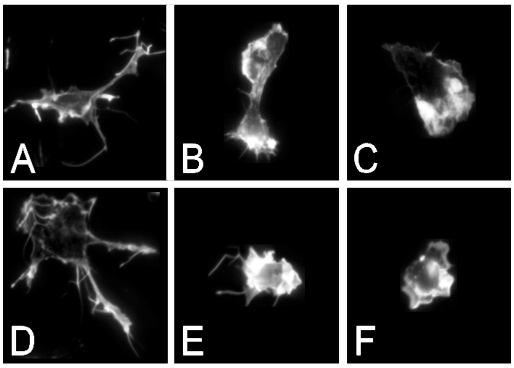

Fig. 1: Morphology of microglial cells. A,D: resting microglial cells; B,E: transition microglial cells; C,F: active microglial cells.

stacked denoising autoencoders (SdAs) [3], to build an automatic classifier of the state of microglial cells.

2 Stacked Denoising Autoencoders

An autoencoder (AE), also known as auto-associator, is a feedforward neural network, usually with a single hidden layer, built to reconstruct its input. The reconstruction process is partitioned in a phase of encoding followed by a phase of decoding. Given an input $\mathbf{x}$, the model applies an encoding mechanism, resulting from the processing of the hidden layer, and the result, $c(\mathbf{x})$, is then subject to a decoding mechanism, resulting from the processing of the output layer, so that the final result, corresponding to the output of the AE, is a representation $\tilde{\mathbf{x}} = d(c(\mathbf{x}))$ of the input $\mathbf{x}$.

If, alternatively, the training is guided in an effort to reconstruct the original input from a corrupted version of it, the AE is dubbed as denoising autoencoder (dAE) [3]. Essentially, when training a dAE, each input $\mathbf{x}$ is first subject to a mechanism of corruption. Then this corrupted version of $\mathbf{x}$ is processed in an analogous manner as in an AE with the aim of reconstructing the original input $\mathbf{x}$. The dAEs were proposed in the context of deep neural networks (DNNs) training, namely, in the framework of greedy layer-wise training. The key idea was to divide the training of a DNN in two stages: a (layer-wise) pre-training phase and a fine-tuning phase. In the case of the dAEs approach, the pre-training phase consists in an unsupervised training of each hidden layer of the DNN, separately and sequentially, from the lowest-level layer to the highest-level layer, regarding it as the hidden layer of a dAE. The input of each dAE is obtained by processing the original (network) input through the previous already trained layers of the DNN and by subjecting the processing result to the mechanism of corruption. Thus, after the training of the first dAE, which has, as hidden layer, the first hidden layer of the DNN, its output layer (decoding) is discarded and the trained hidden layer is used to obtain the input for the next dAE which has, as hidden layer, the second hidden layer of the DNN. This process is repeated for the other hidden layers until all the hidden layers of the DNN are trained.

In the fine-tuning phase, the whole DNN is trained using a gradient descent method like backpropagation [4] where the initial weights of the output layer are randomly initialized while the weights of the hidden layers are the ones obtained in the pre-training phase. The goal of the pre-training phase is to obtain a better initialization of the hidden layers' weights and biases when compared to a random initialization.

3 Experimental Procedures and Results

3.1 Dataset

The dataset consists of a set of images from microglial cells which were acquired in a DMI6000B inverted microscope using the ORCA-Flash4.0 V2 (Hamamatsu Photonics) CMOS camera. Images were exported as raw 16-bit TIFF using the LAS AF software with the original metadata preserved. TIFFs had their background subtracted in FIJI [5] using the roller-ball ramp in between $35-50\%$ pixel radius. Images were segmented in FIJI using a modification of the triangle threshold algorithm for epifluorescence images. Each thresholded microglial cell was delineated using the particle analyze tool in calibrated images and exported to FIJI ROI manager.

After isolating each cell in an image (the images vary in size and shape), a panel of four experts decided, by majority voting, the corresponding label (class) for each microglia. A total of 45 images were acquired and labeled, resulting in the following class distribution: 7 images (15.6%) of cells in the resting state; 22 images (48.9%) of cells in the transition state; 16 images (35.5%) of cells in the active state. It should be noted that the experts' labeling was not consensual for all the 45 cells of the dataset. In particular, there were 2 cells (4% of the dataset) in which only two experts were in agreement, 16 cells (36% of the dataset) in which three experts were in agreement and 27 cells (60% of the dataset) in which the assigned label was consensual. The first 2 cells were classified in transition and active. Moreover, within the mentioned 16 cells, 3 were classified as resting, 11 as transition and 2 as active. In the remaining 27 cells (whose classification was consensual) 4 cells were assigned as resting, 10 as transition and 13 as active (more dataset info at `https://goo.gl/BlTxoO`).

In order to simplify the notation, the resting, transition and active states are coded as C1, C2 and C3, respectively.

3.2 Experimental Setting

We used Theano [6], an open source Python library for numerical computation that allows an efficient use of GPU capabilities in order to increase the processing speed. All the computations were performed using a GTX 770 GPU.

The experimental component was driven through simulations following a common structure. In each simulation, all images were resized to an image with dimensions of 30 × 30 and flatten to a vector of 900 entries which was used as the input of the network. The dataset was then divided into training, validation and test sets with 40%, 20% and 40% of the samples, respectively. The samples were normalized according to the maximum and minimum values of the samples in the training set (after the normalization each component/entry of the input assumed a value in $[0, 1]$).

An SdA of a fixed architecture of 5 hidden layers with 500 sigmoid neurons each and a softmax output layer was trained until the validation error started to increase significantly (executing a maximum of 1000 fine-tuning epochs). The parameters used were: 200 epochs for pre-training, 0.01 for the pre-training learning rate, 0.2 for the fine-tuning learning rate, 10 for the batch size and 0.1 for the corruption rate. The latter implies that 10% of the input components are randomly chosen to be set to 0 (the input components set to 0 may vary within samples). Other values for such parameters were tested in some preliminary experiments but no significant changes were observed.

The above procedure was repeated 20 times (by randomly shuffling the data) in order to capture the general performance of the model regardless of the samples used to train. The reported performance was the average test set accuracy $(1 - \text{BER}) \times 100\%$ over the 20 models, as measured by the Balanced Error Rate:

$$\text{BER} = \frac{E_{C1} + E_{C2} + E_{C3}}{3}. \tag{1}$$

E_{C1}, E_{C2} and E_{C3} are the error rates of each class.

3.3 Results

In the presence of such a small dataset as ours, with a highly unbalanced class distribution, we started by considering a balanced training set with 3 samples from each class. Given that a training set of 9 elements is poor, we used an approach inspired by [7] which consists in increasing the training set with rotated versions (equally spaced) of the original images. Thus, the training sets were obtained by randomly selecting images so that the class distribution in the dataset is respected; the rotations (of the selected images) were then added. The number of rotations considered for each image within a class was nearly the same and was computed so that each class had the same number of elements for training. The values considered for the number of elements per class in the training set were 30, 50, 70 and 100.

Since the original images are of different size and shape, we repeated the same experiments by considering squared images (a black background was added to the images so that the they become square before downsizing). The purpose was to maintain the image ratio when resizing to 30 × 30. This approach, in which all the images have a squared shape, will be called *squared* while the initial approach will be denoted as *non-squared.* Results are shown in Table 1.

	Images/class				
	3	30	50	70	100
NS	54.81(7.51)	55.83(8.82)	60.05(6.46)	60.73(7.67)	60.91(6.42)
S	56.99(7.57)	63.75(7.46)	62.74(7.36)	65.16(7.03)	65.26(7.25)

Table 1: Average performance (in % over 20 repetitions, standard deviations in parentesis) for the original 3-class problem varying the number of elements per class. Results for squared (S) and non-squared (NS) images are shown.

We start by observing that artificially increasing the training set size is beneficial for the SdA learning. Still, it is clear that it is sufficient to use 50 (non-squared) or 30 (squared) images per class (the performances obtained with 70 and 100 images were not substantially higher). It is also clear that the use of squared images is important. In fact, by using this strategy the downsizing step maintains the microglial cell aspect ratio which is fundamental as the recognition is based on the morphology of such cells. Overall, the results show that this is a hard problem with the best performance around 64%, with squared images. Standard deviations are also high due to the small size of the original dataset.

To further investigate where does the difficulty lies, we repeated the study by considering some 2-class sub problems, namely C1 vs C2, C1 vs C3 and C2 vs C3. The results are shown in Table 2.

		Images/class			
		3	30	50	70
C1 vs C2	NS	52.64(10.88)	58.03(12.67)	61.49(9.41)	60.87(10.68)
	S	56.39(12.24)	64.57(8.68)	65.43(8.97)	64.29(10.69)
C1 vs C3	NS	82.92(10.53)	83.63(8.27)	84.88(7.81)	84.86(8.53)
	S	82.29(9.58)	85.5(5.84)	85.75(7.03)	86.00(7.43)
C2 vs C3	NS	63.61(12.45)	67.42(9.37)	68.73(10.30)	67.31(8.97)
	S	66.39(11.76)	69.96(11.76)	70.92(11.55)	70.92(11.55)

Table 2: Average performance (in % over 20 repetitions, standard deviations in parentesis) for the 2-class sub problems varying the number of elements per class. Results for squared (S) and non-squared (NS) images are shown.

The worst performances were obtained in the C1 vs C2 problem, while the best performances were obtained in the sub problem C1 vs C3. This was expected due to the high morphological difference between cells in the states C1 and C3 (see Sect.1). It is particularly evident the difficulty in distinguishing the transition state (class C2) from the others, specifically from class C1. In fact, the resting and transition states share some morphological similarities (such as elongated body and sharper boundaries) that hinder the SdA learning. We also observe that the performance of the models obtained in each repetition is highly variant. This is essentially due to the small size of the dataset and consequent heterogeneity within classes, making the models too dependent on the particular training set. A slight improvement is also obtained when squared images are used. Also, the use of rotations is still beneficial although with a smaller impact for the C1 vs C3 problem.

4 Conclusions

In this work, we applied stacked denoising autoencoders to the classification problem of identifying a microglial cell's state. By considering the original 3-class problem as well as some binary sub problems of it, we conclude that the transition state, being the most important to detect, is also the most difficult to recognize, mainly from the resting state. This is due to the morphological similarities between those classes that cause some disagreement even between the experts panel (please refer to Table 2 at `https://goo.gl/BlTxo0`). We also found important to square the images in order to maintain their aspect ratio preventing, in that way, morphological distortions with the downsizing step. Rotated versions of the original images were also used to diminish the effect of having a small size dataset. Moreover, the generalization ability of the created models may also have been compromised because the dataset is not enough representative of the microglia population's morphology (even within the same class). It is also important to recall that 40% of the cells' classifications were not consensual which may indicate the existence of wrong labels affecting the performance of the models. Globally, the experiments show that it is essential to collect more data in order to improve the results. For these reasons and future work, we are now collecting more data in order to improve the recognition of microglial cells' state.

References

[1] M. W. Salter and S. Beggs. Sublime microglia: expanding roles for the guardians of the CNS. *Cell*, 158:15 – 24, 2014.

[2] K. Saijo and C. K. Glass. Microglial cell origin and phenotypes in health and disease. *Nature Reviews Immunology*, 11:775–787, 2011.

[3] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, 2008.

[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[5] J. Schindelin, I. Arganda-Carreras, E. Frise, and *et al.* Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9:676 – 682, 2012.

[6] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[7] T. Amaral, L. M. Silva, L. A. Alexandre, C. Kandaswamy, J. M. de Sá, and J. M. Santos. Transfer learning using rotated image data to improve deep neural network performance. In *Image Analysis and Recognition*, volume 8814 of *Lecture Notes in Computer Science*, pages 290–300. Springer International Publishing, 2014.