

BCM model as images generator

Tommaso Rondini

September 25, 2023

1 Introduction

The BCM model is a neural network which is able to classify images in clusters given a training data without labels (so it learns in unsupervised way). Looking to the neurons' weights after the training step they could be mapped into an image similar to the training ones. The “neuron's weights” are the importance that the neuron gives to each pixels to classify the images, so how much the intensity of that pixel is relevant to recognise if the image allow to the cluster or not. Obviously the weights are expressed by numbers and not by images, but they can be mapped to obtain a grey scale image. This concept can be applied to colour images too, using the weights of each channel as colour intensity in the synthetic image. In this report I use only grey scale images.

1.1 BCM model

The BCM model was originated by visual cortex neurons study conducted by Elie Bienenstock, Leon Cooper, and Paul Munro in 1982[2]. In following years it was gone to changes: I use the model proposed by Law and Cooper in 1994[7] and then adapted adding lateral connections thanks to the contribution of Castellani et al.[5]. In biology the lateral connections are synaptic links between neurons inside visual cortex. The connections can be positive (if a neuron recognizes a cluster of data, it suggests others to recognize the same type) or negative (it suggests others to find other weights so that they will find different clusters). In the BCM model a matrix is used to express these relationships and we call it *interaction matrix*[3]. I use two matrices in this project:

- a matrix where all the elements are equal and creates only negative connections, that I will call L ;
- a matrix that creates negative connections between all neurons except between couples, that I will call B .

The matrix B has a biological justification: in our brain the visual cortex is split in two hemispheres, the left and the right. Simplifying we can say that the info acquired by the left eye arrives to the right hemisphere and vice versa. But the eyes receive more or less the same info, so both the hemispheres recognise the same object. The B matrix reproduces this behaviour[1]. An example of B matrix is given in Eq.1.

$$\begin{bmatrix} 1. & 0.2 & -0.3 & 0.2 & -0.3 \\ 0.2 & 1. & -0.3 & -0.3 & 0.2 \\ -0.3 & -0.3 & 2.3 & -0.3 & -0.3 \\ 0.2 & -0.3 & -0.3 & 1. & 0.2 \\ -0.3 & 0.2 & -0.3 & 0.2 & 1. \end{bmatrix} \quad (1)$$

1.2 Synthetic images estimation

Once the synthetic images are created, we need to evaluate their quality in order to understand which combination of parameters reproduces the best results. First of all, a qualitative judgment is attended, simply looking the pictures. However, a quantitative opinion is necessary: I use the *Fréchet Inception Distance* (FID)[6], commonly used to evaluate synthetic images generated by *Generative Adversarial Networks* (GANs). It calculates the distance between the training images distribution and the synthetic images one. I try to use other methods, such as the *Peak Signal to Noise Ratio* (PSNR) and the *Structural SIMilarity* (SSIM), but they are generally used to compare two pictures at time and I have to manage datasets. I calculate the *Precision, Recall, Density and Coverage* (PRDC)[8], but it produces results discordant with visual and FID judgements.

1.3 Compilation info

The elements' values of the interaction matrices are copied by above articles, so L has a negative connection value -0.5 , while B has as negative connection -0.2 and as positive 0.3 . The implementation of the BCM model is the Python version of Curti et al.[4] with some adjustments in order to make faster the graphical representation of results. The training dataset is the Olivetti faces, that is a collection of ten pictures of forty people's faces in grey scale, 8bit and 64×64 size. To evaluate the FID I use a *Pytorch* module[9]. Each model is trained for 3000 epochs. The B matrix generates a synthetic "average" image of all the training images, while all the other neurons highlight the difference of the cluster which they select. Instead, the L matrix generates a synthetic image for each neuron. In consequence, the B matrix creates only one image for training, meanwhile the L creates as many images as the number of neurons (N). When the batch size is not specified it is 10. The FID's calculation requires a distribution with that to compare the synthetic images' one. I use the distribution of the all Olivetti dataset; in this way I can compare all the FIDs between them.

2 Comparison between matrices

As first study I check which matrix produces better images, so that other studies will be attended with it. For each matrix I attend two types of simulation: one using as dataset faces of the same person, the other using faces of different people. With the same dataset I change the number of neurons of the model: 5, 9, 13, 17 and 21. I repeat the same type of simulation five times, changing the dataset but not its features. Since for each person I have ten pictures, the datasets are composed by ten images for all the simulations. For each model with each dataset I save 64 synthetic images, changing the weights initialization.

In Fig.1 we see the FID depending on N . I little change the x axis' values since a better visualization, but all the simulations are done with the same (and integer) number of neurons. We see that for a few neurons the better results are achieved by simulation with faces of the same person. Increasing N , the type of matrix is more significant: the B matrix achieves better results with the same neurons. The same observations can be done looking the synthetic pictures. In Fig.2 some of the synthetic images of the simulations are showed. Since these result, future simulations are attended with B matrix.

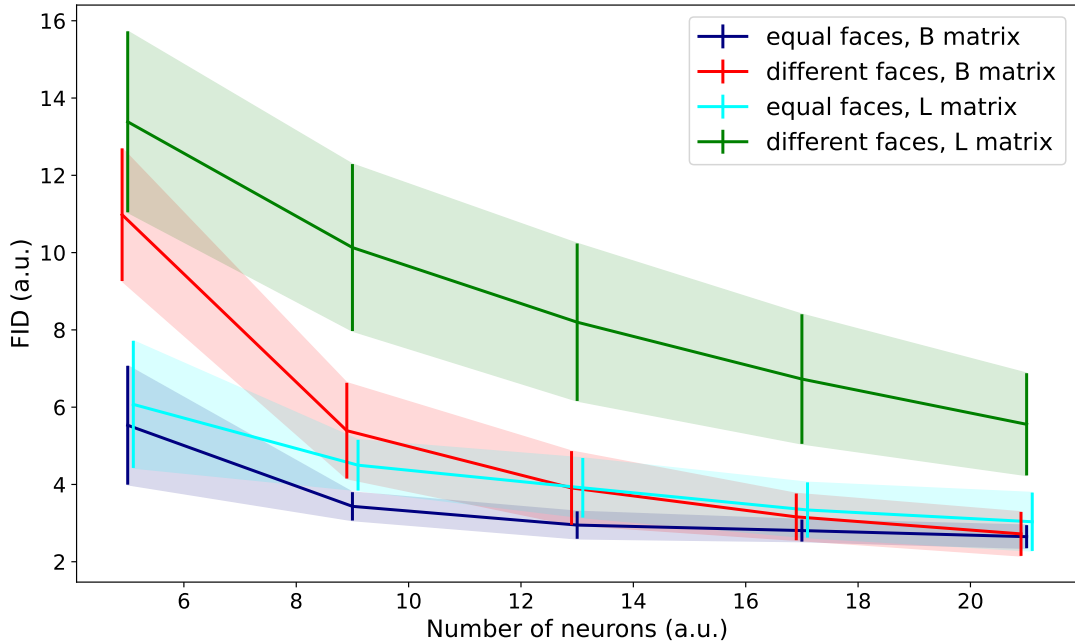


Figure 1: The evolution of the FID in relation to Olivetti dataset depending on the number of neurons. Two simulations are made with L matrix and two with B matrix; one with the pictures of the same person; the other with random faces. Each simulation is repeated five times and the error bars are the standard deviation of the FID in each case. Each dataset is made up ten pictures.

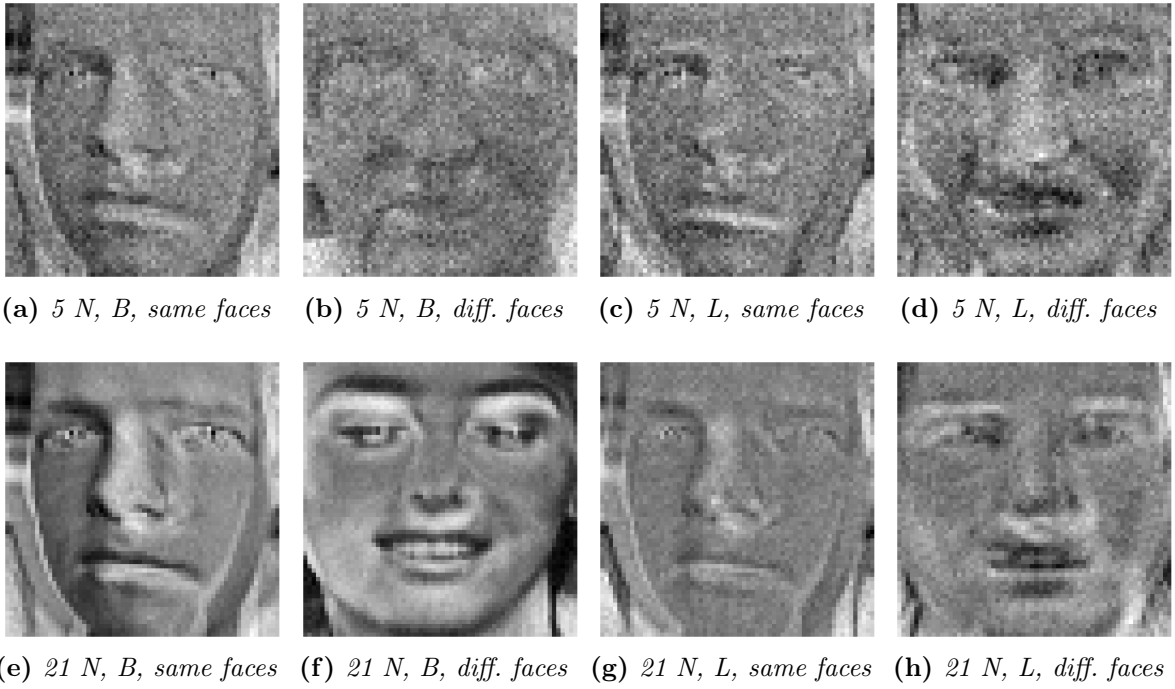


Figure 2: One synthetic picture for type of simulation with 5 and 21 neurons. The first four are synthetic images generated with 5 neurons with different matrices and datasets. Below each of them there is the synthetic image with the same matrix and data but generated with 21 neurons.

3 Comparison between dataset's size

Beyond the two previous simulations with B , I change the dataset's size to see which gives better results. Also for these simulations I repeat them five times and generate 64 images changing initial weights. The number of neurons are still 5, 9, 13, 17 and 21. The studied cases are:

- 5 pictures of the same person;
- 5 random pictures;
- 10 pictures of the same person (above simulation);
- 10 random pictures (above simulation);
- 20 pictures of the same two people;
- 20 random pictures.

The case of 20 picture of two people is a bit different from the others since it mixes two people. I do it because I need to increase the dataset's size and I have at the maximum 10. In the cases with 5 picture as dataset, the batch's size is 5.

In Fig.3 we see that increasing N the FID seems to converge, but for a few neurons the behaviour is totally different. With an high N , we notice that the larger datasets have a better FID than smaller ones. The same thing happens for datasets of the same person respect to random datasets. The datasets made of 20 pictures achieve the best FID (with great N) of all. We notice that the random dataset is better than the one made up two people's faces.

In Fig.4 I quote one synthetic image for simulation, always generated with 21 neurons. Qualitatively we can not see great differences between them, except the synthetic image generated with 5 pictures of the same person (Fig.4a), that is the worst. It is a confirmation that the FID is a good estimation of the visual evaluation, since in Fig.3 we see that the worst FID at 21 neurons is just with the 5 same person pictures dataset and all the other are more similar one each others.

4 Comparison between batch's size

Since in above section we see that the best FID is achieved by a dataset of 20 random faces, I decide to use as training dataset all the Olivetti's pictures. Then, I increase the number of neurons of the model (since better results are attended with high N) and I try with 21, 25 and 29.

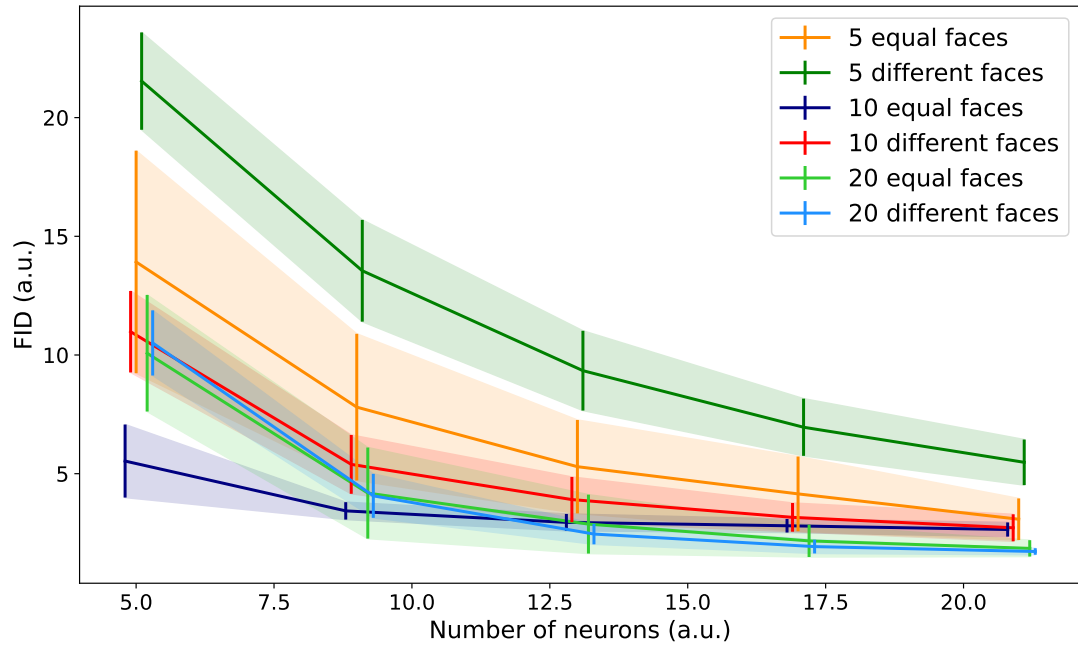


Figure 3: The evolution of the FID in relation to Olivetti dataset depending on the number of neurons. For the same dataset's size (5, 10 and 20) there are two simulation: one with pictures of the same person; the other with random pictures. Each simulation is repeated five times and the error bars are the standard deviation of the FID in each case. The interaction matrix is always B .



(a) *size 5, same faces*



(b) *size 5, diff. faces*



(c) *size 10, same faces*



(d) *size 10, diff. faces*



(e) *size 20, same faces*



(f) *size 20, diff. faces*

Figure 4: One synthetic picture for dataset's type. In the first column there are the picture generated with datasets made up same people's faces; in the second with datasets made up random faces. For each row, the dataset's size is the same, so in the 1, 2 and 3 row there are pictures with dataset's size of 5, 10 and 20 respectively. The number of neurons is always 21.

The last parameter to test is the batch size. Until now it was almost always 10; now I select 10, 20, 50 and 100. Since I use all the dataset, I can not reply the simulations five times, so in Fig.5 I have not the standard deviation of the FID.

The FID's trend does not highlight any common features or a more efficient batch. The two best results have batch 100 and N respectively equal to 21 and 29, but with N equal to 25 the FID has a spike. On the contrary, the FID with batch 20 has a minimum with N equal to 25. With batch of 10 and 50 FIDs are monotonic: the first is decreasing, the second is increasing.

The synthetic images generated with the smallest FID (batch equal to 100 and N equal to 21) are qualitative good, too, as you can see in Fig.6. Unfortunately, using all the dataset most of them are very similar and, between 64 generated, there are only three different faces.

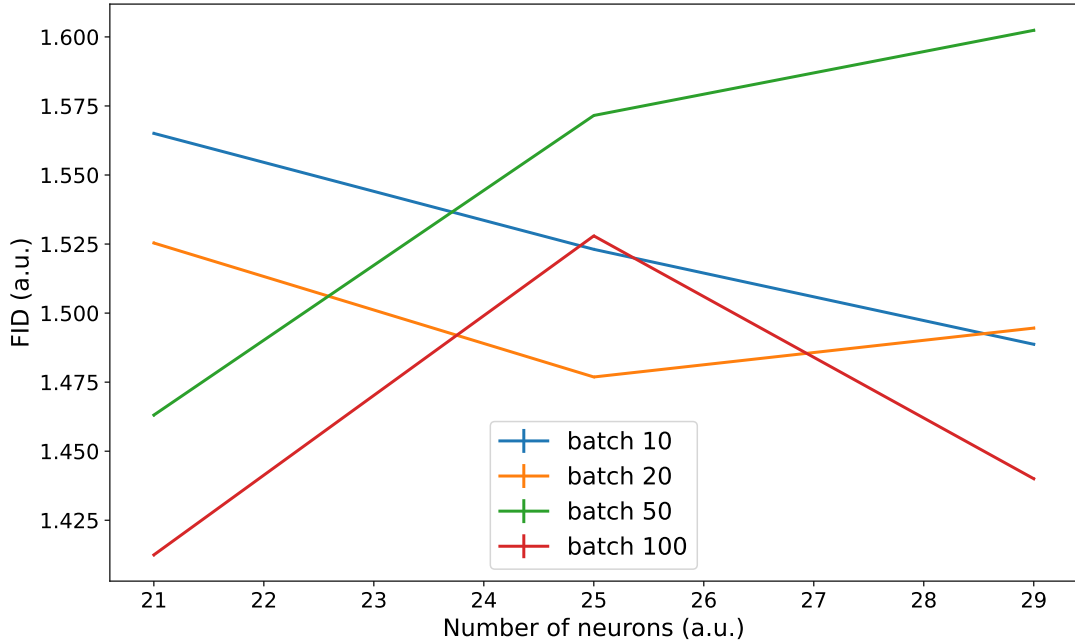


Figure 5: The evolution of the FID in relation to Olivetti dataset depending on the number of neurons. There it is used always all the dataset, but changing the batch's size. The interaction matrix is always B .

5 Conclusions

Given a dataset, to generate synthetic images throw a BCM model is on the whole possible. You get images resembling those of training mapping neurons' weights. Modifying the parameters and the dataset you achieve quite good images: the better combination

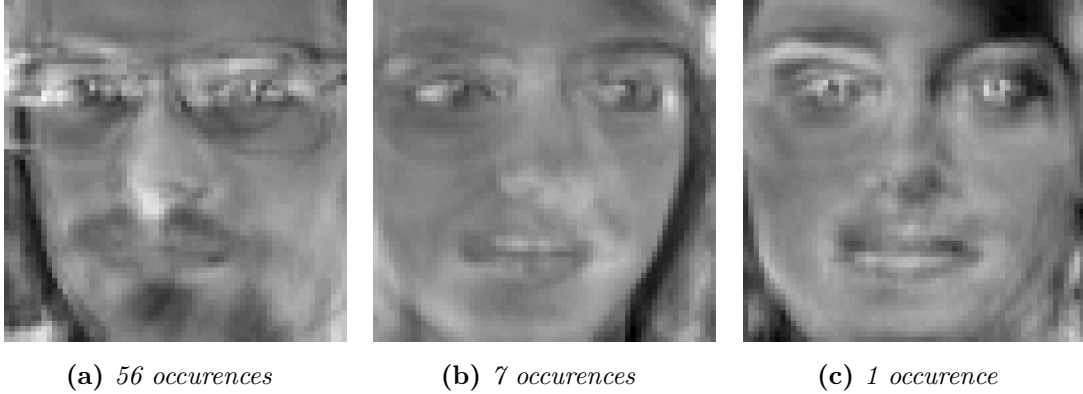


Figure 6: Three synthetic pictures with batch’s size equal to 100 generated by a BCM model with 21 neurons and interaction matrix of type B . All the other generated images (61) are similar to these three. The occurrence of each example is reported in its sub-caption.

of them that I found is with 21 neurons, interaction matrix of type B and batch’s size equal to 100, using as training dataset the whole Olivetti. The FID of this simulation is 1.412.

However the synthetic pictures are neither qualitatively or quantitatively indistinguishable from the training ones. I am not able to determine some rules to find the best results, since increasing N does not always reduce FID either (see Fig.5). I can say that the matrix B return better results, but generates less images.

Ultimately, if on the whole the model can be used to synthesize images, at the state of the art it can not generate qualitatively indistinguishable images from real ones, much less to compete with others existing synthesizers. Furthermore, the latter can generate hypothetically infinite images once they are trained; on the contrary the BCM model needs to be trained for each synthetic image.

References

- [1] Gregorio Berselli. *Memorization capability of the BCM model*. https://github.com/Grufoony/BCM_selectivity. 2023.
- [2] Elie Bienenstock, Leon Cooper, and Paul Munro. “Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex”. In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 2 (Feb. 1982), pp. 32–48. DOI: [10.1142/9789812795885_0006](https://doi.org/10.1142/9789812795885_0006).
- [3] L N Cooper and C L Scofield. “Mean-field theory of a neural network.” In: *Proceedings of the National Academy of Sciences* 85.6 (1988), pp. 1973–1977. DOI: [10.1073/pnas.85.6.1973](https://doi.org/10.1073/pnas.85.6.1973).
- [4] Nico Curti et al. *plasticity - Unsupervised Neural Networks with biological-inspired learning rules*. <https://github.com/Nico-Curti/plasticity>. 2020.
- [5] H Shouval G C Castellani N Intrator and L N Cooper. “Solutions of the BCM learning rule in a network of lateral interacting nonlinear neurons”. In: *Network: Computation in Neural Systems* 10.2 (1999), pp. 111–121. DOI: [10.1088/0954-898X/10/2/001](https://doi.org/10.1088/0954-898X/10/2/001).
- [6] Martin Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *NIPS’17* (2017), pp. 6629–6640. URL: <https://dl.acm.org/doi/10.5555/3295222.3295408>.
- [7] C C Law and L N Cooper. “Formation of receptive fields in realistic visual environments according to the Bienenstock, Cooper, and Munro (BCM) theory.” In: *Proceedings of the National Academy of Sciences* 91.16 (1994), pp. 7797–7801. DOI: [10.1073/pnas.91.16.7797](https://doi.org/10.1073/pnas.91.16.7797).
- [8] Muhammad Ferjad Naeem et al. “Reliable Fidelity and Diversity Metrics for Generative Models”. In: *ICML’20* (2020). DOI: [10.48550/arXiv.2002.09797](https://doi.org/10.48550/arXiv.2002.09797).
- [9] Maximilian Seitzer. *pytorch-fid: FID Score for PyTorch*. <https://github.com/mseitzer/pytorch-fid>. Version 0.3.0. Aug. 2020.