

Clustering within Integrate-and-Fire Neurons for Image Segmentation

Phill Rowcliffe, Jianfeng Feng, and Hilary Buxton

School of Cognitive and Computing Sciences, University of Sussex, Brighton, BN1 9QH, England

Abstract. An algorithm is developed to produce self-organisation of a purely excitatory network of Integrate-and-Fire (IF) neurons, receiving input from a visual scene. The work expands on a clustering algorithm, previously developed for Biological Oscillators, which self-organises similar oscillators into groups and then clusters these groups together. Pixels from an image are used as scalar inputs for the network, and segmented as the oscillating neurons are clustered into synchronised groups.

1 Introduction

Integrate-and-Fire neurons have been studied within the field of computer vision for some time now. One aspect of IF neurons is that they can act as natural oscillators and can be thought of as pulsed-coupled oscillators (oscillators which fire a pulse when a threshold is reached, then reset to zero).

One application of such a network of oscillators is that of clustering [1]. In [6], Rhouma and Frigui introduced a clustering algorithm which was both insensitive to the initialisation of the network and did not require the optimising of a sophisticated iterative objective function. They took pulse-coupled Integrate & Fire (IF) oscillators and partitioned them into synchronised groups, with a simple objective function based on a dissimilarity measure between the oscillators and the input data. Networks of oscillators were self-organised into clusters where the number was a product of the coupling function.

Here the algorithm developed in [6] is extended for use on a fully connected network of IF neurons, and used to achieve image segmentation on face image inputs. The neurons are based on the Gerstner model [3] which is a simplified model of the single state equation [5] derived from the Hodgkin-Huxley model [4].

The rest of the paper is outlined as follows. Section 2 will consist of a brief review of related literature, in particular [6], which forms the basis on which this research has been developed. Section 3 will look at the Integrate-and-Fire model used and the clustering algorithm. In Section 4 the results from the application of this model, using images of faces as inputs, are presented. Finally, in section 5 conclusions on findings and discussions of how this model will be taken further, are explained.

2 Related Work

Clustering, as an unsupervised method of classification, is a means by which data is partitioned into naturally present subsets, or groups. Wang et als [7] LEGION model segments data using a locally connected network, but was found to be sensitive to noisy data and its initialisation stage. Rhuoma and Frigui [6] produced an algorithm which was not computationally complex and which also addresses some of these inherent problems within clustering. In [6] a model was developed which was both insensitive to the initialisation of the network and which used a simple dissimilarity measure to update the weight values. It took pulse-coupled oscillators, with excitatory and inhibitory connections, and partitions them into synchronised groups, with an objective function based around a simple dissimilarity measure. Networks of oscillators were self-organised into clusters (the number a product of the coupling function).

The oscillators used here though, are IF neurons, with spike response. They are based on the Gerstner model [3] which is a simpler description of the Hodgkin-Huxley model [4] previously simplified as a single state equation [5]. He has shown how the IF model is a special case of the *Spike Response Model* developed in [3] from the results Gerstner et al showed in [5]. However Feng in [2] recently showed that the IF and HH models do behave differently, infact in opposite ways when correlated inputs are considered.

3 Model

3.1 Neurons

The neurons in this network are IF Neurons, with purely excitatory connections. The basic model of an IF neuron has each neuron consisting of a current $I(t)$, a capacitance C and resistance R , similar in fact to a circuit. Just like in a circuit the current input into a unit $I(t)$, is composed of two components, the current going through the capacitor I_c and that going through the resistor I_r , i.e. $I(t) = I_c + I_r$. Using Ohm's law $V = I \times R$, and the definition of *capacitance* (i.e. $C = q/v$, where q is the unit charge and V the voltage, or membrane potential), we can rewrite the current equation as:

$$I(t) = C \frac{dv}{dt} + \frac{V(t)}{R} \quad (1)$$

Allowing C and R to have a value of 1, and integrating over δt this becomes:

$$V(t + \delta t) = V(t) - \delta t V(t) + \delta t I(t) \quad (2)$$

The neurons receive inputs from two different sources: the *Input image* (x_i being the pixel input to neuron i) and *other neurons* in the network, which are in the form of *spikes*. A process has been used to simulate a spike train with l_j being the output spike from neuron j .

$$l_j(t) = (t - t_{thres}) \exp\left(\frac{-(t - t_{thres})}{T}\right) \quad (3)$$

Here t_{thres} is the time that the neuron firing the spike reached threshold, and t , the present time. T is the time constant with which the membrane potential decays. Each neuron has a weighted coupling w connecting them to each other. At any point in time a neuron receives inputs from both other neurons in the network, and the pixels in the input scene. The voltage change is therefore:

$$V_i(t + \delta t) = V_i(t) - \delta t V_i(t) + \delta t \left(\frac{x_i}{S} + \sum w_{ji} l_j(t) \right) \quad (4)$$

where w_{ji} is the weighted connection between neurons j and i , and l_j the output spike from neuron j . S is a scaling factor used to normalise the pixel input.

3.2 Algorithm

The algorithm operates in two main parts. The first part works on the network of oscillating neurons and adjusts the lateral weight connections between the neurons. The clustering aspect of synchronised groups of neurons is the second part of the algorithm.

Oscillatory Adjustment. The adjustment to the connections between the neurons is designed to force similar groups to threshold at the same time. The method used to cluster neurons is that described in [6] as *Clustering of Object Data*. The purpose of the algorithm is to partition the neurons into groups. Each pixel in the input image is represented by a neuron and similar inputs are grouped together into the same cluster as follows:

Let X be the set of N neurons, and P be the set of M prototypes. The *prototypes* represent the clusters within the network, [6] recommends this to be around $N/4$ for best results. The next neuron to reach threshold is identified, say X_j . The closest prototype to this neuron is also identified, say P_k . The distances are then calculated between all the other neurons and this prototype. These distances d will be used to update the weight values on neurons connected to neuron X_j using the new updated (normalised) distance \tilde{d} as follows:

$$d_{jk} = \begin{cases} ll d_{jk} & \text{if neuron does not belongs to any group,} \\ \frac{\sum_{l \in G} d_{lk}}{|G_m|} & \text{if neuron belongs to group } G_m. \end{cases} \quad (5)$$

where G is the group of synchronised neurons $G = G_1, \dots, G_m$ (grouping is explained in the next section), m the number of synchronised groups so far. The current state or *phase* of neuron i , at a particular point in time, is represented by ϕ_i . In synchronising neurons the phases will be equal and the firing of one neuron will bring the others to their membrane threshold. Therefore, the coupling of neuron i is updated by $\epsilon_i(\phi_j)$, where:

$$\epsilon_i(\phi_j) = \begin{cases} ll C_E [1 - (\frac{d_{jk}}{\delta_0})^2] & \text{if } d_{jk} \leq \delta_0, \\ C_I [(\frac{d_{jk}-1}{\delta_0-1})^2 - 1] & \text{if } d_{jk} > \delta_0. \end{cases} \quad (6)$$

Here, C_E and C_I are the excitatory and inhibitory coupling strengths, respectively and δ_0 is the dissimilarity measure between the input pixels.

Grouping. At the end of each iteration the groups of synchronised neurons are updated. Firstly all the other neurons which had been brought to threshold are identified and then one of the following cases is carried out:

- Create a new group if none of the identified neurons, which have reached threshold, have been assigned to a group and update the number of groups by 1.
- If several of the neurons which have reached threshold belong to an existing group, say G_m , then assign all these neurons to G_m .
- If several of the neurons which have reached threshold belong to groups G_{m_1}, \dots, G_{m_q} then delete their association with the $q-1$ groups and assign these neurons to G_{m_1} .
- if no neurons reach threshold, then do nothing.

The iterative cycles are stopped when the number of groups stabilise.

4 Results

The algorithm discussed in section 3 has been implemented using a network of Integrate-and-Fire neurons. Here results are presented using a variety of faces as input. The images are greyscale and fed in as a bitmap. All the experiments took between 5-8 iterations before the grouping stabilised. The weight updating values C_E and C_I used were 0.1 and 0.01 respectively and the dissimilarity measure $\delta_0 = 0.001$.

The initial images sampled were approximately 280 x 380 pixels square. These were reduced to input images 44 x 58 by simply removing rows and columns of pixels, no sampling or other technique was used.

Figure (1)(a) shows the original photograph sampled for input. Image (b) shows the initial state of the network with all the neurons black, indicating they have not been assigned to any group. Images (c) to (f) show the progress of the algorithm through iterations 2 to 5. It can be seen that more and more of the black, unassigned neurons are grouped together. Neurons which are clustered into the same group are assigned the same colour until at the end of the iterative cycle all neurons are assigned to a group.

Figures (2) and (3) shows 2 faces which took 8 iterations until stabilisation.

In Figure (2) the algorithm finally merged small, similar groups together which had in the previous iterations been kept separate. This effect is seen to a greater degree in Figure (3) where the presence of noise is treated in the same way. The input image used in Figure (3) contained lots of noise and the algorithm appeared to be particularly robust when such noise was present, clustering these stray pixels together with similar ones. However, this does have the effect, as in Figure (2), of absorbing segments of the image which would ideally have been kept separate. This effect is a result of the scalar input. Adjusting scale parameter S so that small variations in pixel value are preserved, allows the algorithm to segment the image to achieve a finer granularity. The algorithm, as in Rhouma and Frigui's experiments [6], has shown a robustness to the effects of noise.

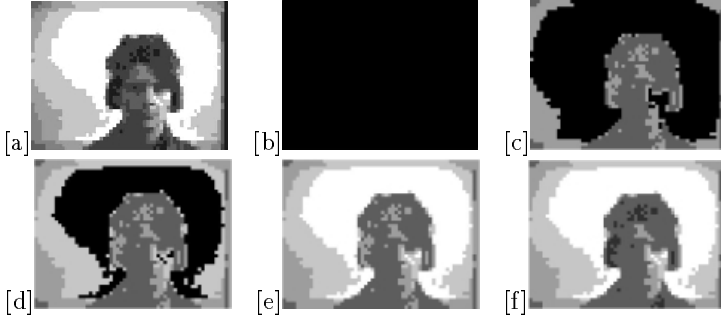


Fig. 1. Clustered neurons tested on an image of a face over 5 iterations. (c) to (f) shows the iterative states of the network on input image (a) from starting state (b).

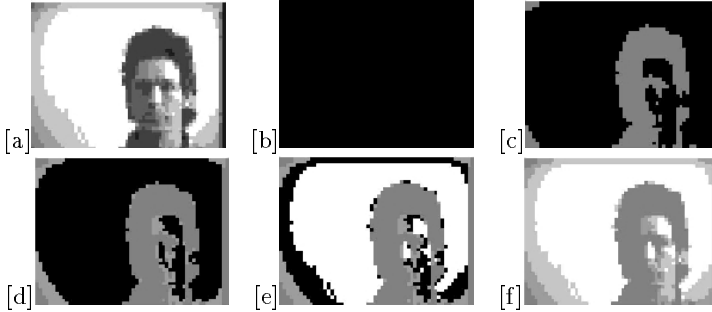


Fig. 2. From input image (a) and starting state (b), figures (c) to (f) show the segmentation of the image over 8 iterative steps. Some groups are merged together over the iterations resulting in the final segmented image (f).

5 Conclusion

The implementation of Rhouma and Frigui's algorithm on more realistic biological neurons has produced positive results when applied to image segmentation. It works effectively with single images and produces positive results even when noise is present in the data.

The IF neurons, though more representative of biological neurons than those used in [6], are still only approximates of more realistic biological models currently being used in today's research. A natural progression is to develop the model of the neuron further, and produce a closer representation of those seen in biology. This might be achieved by developing the algorithm on the FHN model of neurons as shown in [2] to more closely approximate the HH model than the IF model does.

The algorithm here has only been tested on a single layer of neurons and has yet to be developed to deal with multiple layers. This would seem an inevitable

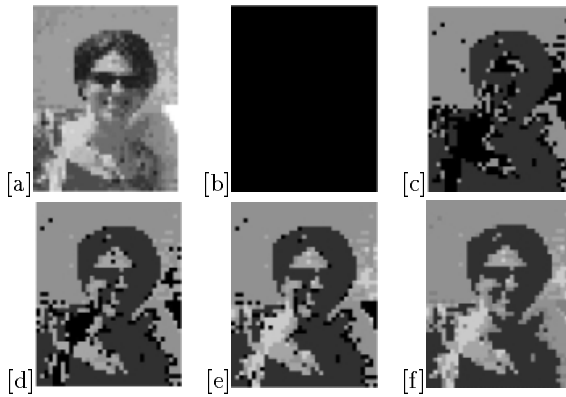


Fig. 3. Clustered neurons tested on a noisy input image, (a), from starting state (b), over 8 iterations shown in (c) to (f). The results appear robust to the effects of noise.

next step especially if our objective is to identify ways of clustering more realistic biological neurons, the architecture of networks of neurons being of an equally important consideration.

References

1. Bressloff, B.C., and Coombes, S. Synchrony in an array of integrate and fire neurons with dendritic structure. *Phys. Rev. Lett.*, vol. 78 pp. 4665–4668, 1997.
2. Feng, J. Is the integrate-and-fire model good enough? – a review. *Neural Networks*, vol. 14, pp.955–975, 2001.
3. Gerstner, W. The spike response model. In *The Handbook of Biological Physics* vol. 4 (Ch.12), pp. 469–516, Moss, F., and Gielen, S. Eds. Elsevier Science, 2001.
4. Hodgkin, A.L. and Huxley, A.F. A quantitative description of ion currents and its applications to conductance and excitation in nerve membranes. *J. Physiol. (London)*, no. 117, pp. 500–544, 1952.
5. Kistler, W.M., Gerstner, W., and van Hemmen, J.L. Reduction of Hodgkin-Huxely equations to a single-variable threshold model. *Neural Comput.*, no. 9, pp. 1015–1045, 1997.
6. Rhouma, M.B.H., and Frigui, H. Self-organisation of biological oscillators with application to clustering. *IEEE Trans. Patt. Analysis Mach. Intell.*, vol. 23, no. 2, Feb. 2001.
7. Chen, K., Wang, D.L., and Liu, X. Weight adaptation and oscillatory correlation for image segmentation. *IEEE Trans. Neural Network*, vol. 11, no. 5, pp. 1106–1123, Sept. 2000.