iqr Tutorial 05-Camera-Negative image

iqr tutorials will provide you with a practical introduction to using the neural simulation software and give you an insight into the principles of connectionist modeling. They are intended to be complementary to the detailed operation manual.
The home page of iqr is at iqr.sourceforge.net. Up to date information, documentation and tips and tricks ca be found in the iqr wiki (sourceforge.net/p/iqr/wiki/Home/).
The repository of iqr packages is here: sourceforge.net/projects/iqr/files/. iqr is open source software. You can browser the entire source code of iqr here: http://sourceforge.net/p/iqr/code/HEAD/tree/. Please contribute to iqr by reporting bug (sourceforge.net/p/iqr/bugs/) and requesting features (sourceforge.net/p/iqr/feature-requests/).

Aims

- *In this tutorial we will learn how to invert the activity of a group. We will see that if we apply this operation to the camera output, we can obtain a "negative" image.*
- *The result of this tutorial is implemented in the file “negative.iqr”. However, the file is not complete. As an exercise, you have to set the missing parameters while reading the tutorial.*

Introduction

For the next tutorials we will use a camera and the iqr “video module” (please check the Tutorial “Camera-Basic interfacing” for more details). In general we can consider a camera as a matrix of sensors; each pixel gives the reading of a particular sensor in terms of amount of light; if we use a 160x120 camera output we are dealing with almost 20 thousand individual sensors.

When we use a sensor device, the first step is to check the range of its output signal. In iqr we can this by setting the output of the sensor module to a neuron group of linear thresholds and then check for the output in the space plot. The min and max values in the bar of the space plot will give us an idea of the approximate range.

In the previous tutorial we searched for the brightest spot in an image. This task was easy because the video module represents brighter spots with higher values. Therefore, we could just use an Integrate and Fire (IaF) neuron and set a threshold so that only neurons with higher inputs were active. But had we searched for the darker spots instead of the brighter spots, then we would had faced a problem. How can we set a threshold and let pass only the things that are below it?

A solution is to invert the original image, inverting an image results in the “negative image” (as the negatives produced by old-fashioned cameras) where the darker spots are now the clearer. In our case, we obtain a group whose output is the negative of the original group's output; higher values are now encoded with lower values and vice versa. The file “negative.iqr” shows how this can be simply done using three groups.

Before proceeding we have to consider that IQR has certain limitations inspired by biological neural networks: in IQR standard neurons only encode positive values. Therefore, in iqr the inversion of a certain positive value 'x' can't be coded as '-x', because a neuron's activity can't be negative.

Then, a first constrain for our operation is that the result of the inversion has to be positive.

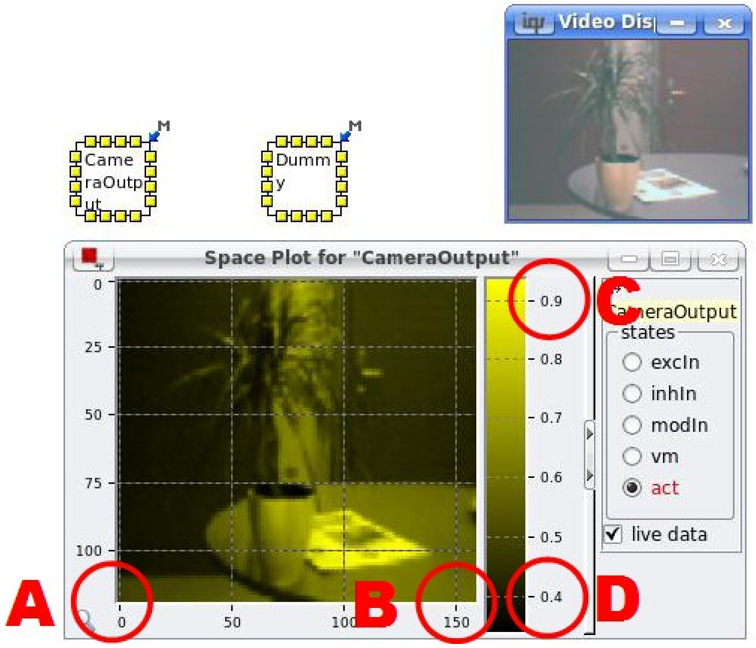

Figure 1 Output of the camera seen as a “space plot” of the group “CameraOutput”. The group has approx. 100 rows (A) and approx. 150 columns (B) (the true values are 120 and 160). From the upper (C) and lower (D) values of the “color bar” we see that the maximum value mapped is between 0.9 and 1.0 and the minimum between 0.3 and 0.4

A second desirable feature is to preserve the range of the output values. An example: if the pixel values of the original image range from 0 to 1, then we want the pixel values of the negative image to range from 0 to 1 as well.

The simpler inversion operation that respect both constrains is '1-x'.

Formula

Negative = 1 – Original

Implementation

File: “negative.iqr”

Processes

Again, we will use just one process “camera” with its properties set as in tutorial 1.

Groups

We need 3 groups + “Dummy”. “Original", "Negative" and "Bias".

We send the output of the “camera” module to the “Original” group, as we did in the previous tutorial. This group is connected to the group “Negative", where we will obtain the negative image. For the inversion operation we require the "bias" group as well that will play the role of the "1" in the formula. For technical reasons we still need the "Dummy" group.

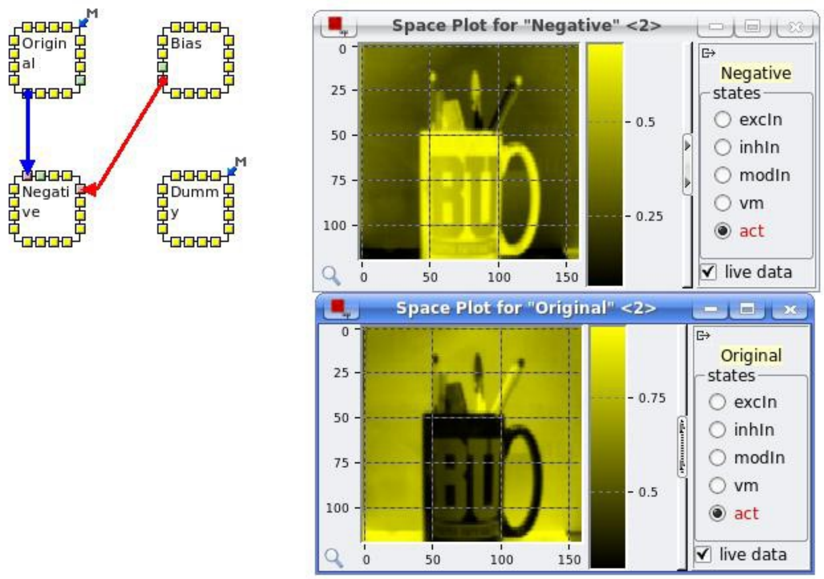

Figure 2: qr circuit, original mug (bottom) and negative mug (top).

We connect “Original” and “Negative” with an inhibitory connection. As both groups have the same size, by adding a connection we will obtain 1-to-1 links between the neurons in both groups; this means that a neuron in a certain position within the source group will send its output to a neuron in the same position in the target group. This is the default setup, it can be modified using the connection properties as we will see later. However, the default is setup is the one we want for this connection.

The result we obtain after the linking these two groups is that the higher the value in “Original” the lower the value in “Negative” (and vice versa). At this point we are still not done because the values in “Negative" are negative; we are only obtaining "-x" and not "1-x" . For the "Negative" group we still need a "default" excitatory input. We solve this using the "Bias" group. It consists of a single "random spike" neuron connected to the group "Negative" with an excitatory connection.

There are two important points

- We don't want activity of the “Bias” group to be “random”; we want it to be constantly on. We achieve this by setting the spike probability to 1.
- We have 1 neuron in the “Bias” group and 19800 neurons (160 x 120) in the “Negative” group. Then this only neuron has to send output to all neurons in the target group, meaning that there has to be a connection between the one neuron in “Bias" and all the neurons in "Negative". If this is the case, then each time the source neuron generates an output (as 1), all the neurons in "Negative" receive this 1.

At the level of IQR we can generate this kind of connectivity in two different ways:

- Set the “pattern mapping” to “ForEach”.
- Set the arborization type to “ArbAll” and its parameter direction to “PF”.

The first configuration means that we generate a connection between all possible neuron couples, where a couple contains one neuron from the source and one from the target group. For the second configuration, you can check what it does tutorial 2.

Warning

Connecting two groups of 160 x 120 means creating almost 20 thousand individual links (or synapses). These are a lot connections, and you could notice it by seeing that now iqr needs a bit more time (several seconds) to start running the simulation. You can also notice it if you open a “connection plot". You better avoid to open a connection plot when it contains so many synapses. It could hang your IQR for more than a minute.

Links to other domains

Image processing

You can find this function implemented in Photoshop. It is simply called “invert". A difference is that photoshop does this in a color image, whereas we only did it for a gray scale image. However, we could attain the same result working with three channels (RGB) and inverting each of them separately (and then we sent them to an output module)

Biology: systems controlled by dis-inhibition.

Biological neurons work as thresholded elements. They fire when the internal electrical charge exceeds a tipping point. Real neurons can not change the sign of their output (this is known as the “Dale's principle"): they can be excitatory or inhibitory, but they can not have both effects.

Nonetheless, some brain areas perform an “inversion” computation. The way it is achieved is as we did it in this exercise: a persistently active neuron(s) is controlled by an inhibitory neuron(s). These circuits are said to work by “dis-inhibition". An example is found in the cerebellum: The Cerebellar Cortex controls the Cerebellar Nuclei by dis-inhibition: when the cerebellar cortex stops the cerebellar nuclei starts and vice versa.

Exercises

- Set the missing parameters in the file “negative.iqr” in order to have the simulation running. Report the changes.
- Change the neuron type of the second group to “Integrate and Fire”. Set the right threshold value in order to obtain the negative version of result of tutorial 4. Provide screen shots of the result and explain how you did it.