

iqr Tutorials 4 and 5

These 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. Hence, detailed instructions on which buttons to press are not included here in most cases.

Links

- [iqr page](http://iqr.sourceforge.net): <http://iqr.sourceforge.net>
- [repository of iqr packages](http://sourceforge.net/project/showfiles.php?group_id=190531) (http://sourceforge.net/project/showfiles.php?group_id=190531)

Bugs and features on sourceforge

- [Submit bug report](http://sourceforge.net/tracker/?group_id=190531&atid=933732) (http://sourceforge.net/tracker/?group_id=190531&atid=933732)
- [Submit feature request](http://sourceforge.net/tracker/?group_id=190531&atid=933735) (http://sourceforge.net/tracker/?group_id=190531&atid=933735)

Code on sourceforge

- [browse SVN repository](http://sourceforge.net/svn/?group_id=190531) (http://sourceforge.net/svn/?group_id=190531)

Table of Contents

iqr Tutorials 4 and 5.....	1
Introduction	2
Tutorial 4: Classification.....	3
Tutorial 5: Negative Image.....	6

Introduction

For the next tutorials we will use the WEBCAM SPC220NC and the iqr "video module". 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.

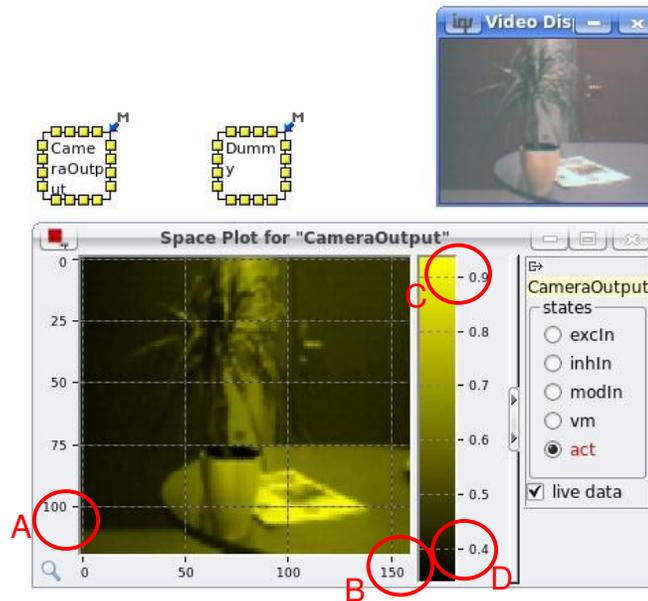


Fig. 1: Output of the camera seen as a "space plot" of the group "CameraOutput". The group has approx. 100 rows (A) and approx. 150 columns (B) (the true values are 120 and 160). From the upper (C) and lower (D) values of the "color bar" we see that the maximum value mapped is between 0.9 and 1.0 and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

When we use a sensor device, the first step is to check the range of its output signal. In iqr we can do 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.

Tutorial 4: Classification

- *In this tutorial we will learn to discriminate a group input into two different classes. We will apply this operation to real time video in order to obtain a 2-color output.*
- *The result of this exercise is implemented in the file "classification.iqr". However, the file is not complete. **As an exercise, you have to set the missing parameters while reading the tutorial.***

Introduction

Most information that we can get via sensors is expressed as numerical values in a range. For instance, the value of pixel in a monochrome image indicates the luminosity level in a region of the image. In some cases this may be encoded by an integer number between 0 and 255, thus providing 256 different possible values. In the case of the output of our video module the values are encoded as decimal numbers between 0 and 1, and to fulfill your curiosity, there are 256 different values as well (256 is an important number for computers).

In some situation though we may be interested in reducing all this amount of information, 256 or more different values, to just two: 0 and 1. And with 0 and 1 we may be encoding dark or clear, too close or far away, noisy or silent, etc.

This is one of the easiest operation that we can apply to the data given by a sensor, it consists in setting a threshold and consider that only values above that threshold correspond to the "active" class or 1.

Example: Discrimination between classes of spots in an image

Let's think of a task where we have to search for a white object in an image, for instance a football or a (pale) face. If we are provided with a monochrome camera, a first step towards finding a white ball could be to discriminate clear spots from darker ones. We can do this by setting a threshold.

Quite likely, the main object that your webcam will find in its visual field is your face. With such working material one of the games you can play is to search for the threshold value that better isolates your face from the background. Based on luminance (gray level), this will be an easy task if, for instance, your skin color is clearer than the background (if your skin color is darker than the background you have to jump to the "Negative Image" exercise).

Implementation

Processes

To implement this operation in IQR we only need one process with its type set to "video module". We have to select the "output" of the camera to monochrome

("Process properties", tab "output", checkbox "monochrome").

We will need two groups "CameraOutput" and "dummy". We will send to "camera output" the information we want to process and to "dummy" the one we discard. To see that, check the properties of the the process "camera" in the tab "module to groups". See that the "Grayscale" output is sent to "CameraOutput".

Groups

The iqr implementation of this tutorial is quite minimalistic, as said above we need two groups, but only one of them ("CameraOutput") is involved in the classification (Fig 2).

Both groups size has to match the camera resolution: 160x120. This is set in the group "properties", section "topology", for the "topologyRect" (press the "edit" button) the values 160 and 120. Whenever we refer to a groups "size" we refer to this "width" and "height". For the group "dummy" we do not care of anything else.

The group "cameraOutput" does the job. First of all, recall that we want to give only two kind of outputs: 0 and 1. The neuron type "Integrate and Fire" does it, when it "spikes" it outputs "1". When it is not spiking it outputs 0. Therefore all we have to do is set the appropriate properties:

- "persistence" and "probability": unless otherwise stated, is safe to set these values to "0" and "1" in most of the cases (this holds for "Integrate and Fire", "Linear threshold" and "Sigmoid neurons".
- "membrane potential reset": if "persistence" is set to 0 this parameter has no effect, but again, set this value to "0" unless you have good reasons to do otherwise.
- "threshold": this is the only parameter that matters for this exercise. By setting its value we decide the extent of the "active are" that we will see in the image.

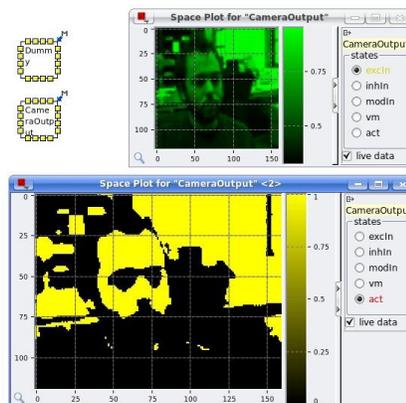


Fig. 2: Face of the bearded author of this exercise with a messy office landscape as background. The output of the camera appears in green in the top right. The yellow space plot shows the result of the thresholding process.

As you will see, IQR does a good job in processing the data at real time. Keep in mind that you CAN modify parameter values while an IQR model is running, you can not

modify the neuron types, connections, etc. Changes to parameters are effective once you press the apply button. In this exercise you can modify the "threshold" value up and down until you get your favorite output. Test the «0» and «1» cases.

Links to other domains

Maths

In maths the operation performed by this module is known as Heaviside function.

Photoshop

If you ever worked with an image processing software, like Adobe Photoshop, you can find this manipulation in the program. Indeed in Photoshop its name "threshold" and it does exactly the same as our exercise. The difference is that what you do in Photoshop for one image, in IQR you do it to a stream of images in real time.

Exercises

1. Complete the IQR module with the missing parameters. Report them.
2. Change the Integrate and Fire neuron to a Linear threshold neuron ("persistence" and "probability" set to 0 and 1 respectively). Set the threshold to the same value you used for the "Integrate-and-Fire" neuron. Describe the result.

Tutorial 5: Negative Image

- *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

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 have 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.

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 respects both constraints 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 ob-

tain 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.

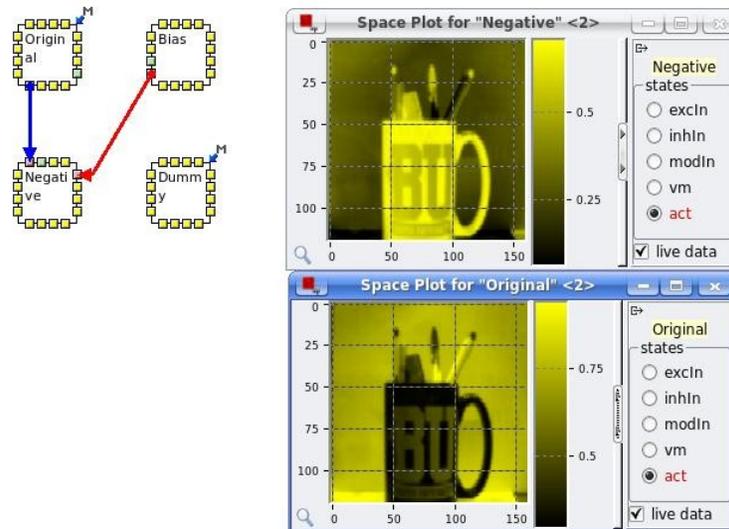


Fig. 3: iqr circuit, original mug (bottom) and negative mug (top).

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

1. 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.
2. 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:

1. Set the "pattern mapping" to "ForEach".
2. 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 grayscale 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

1. Set the missing parameters in the file "negative.iqr" in order to have the simulation running. Report the changes.
2. 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.