VISUAL FLOW DETECTION USING A RECURRENT XOR NEURAL NETWORK LAYER

BY

Rafi Qumsieh

rafiqumsieh@gmail.com

May 2019

ABSTRACT

Attention is a very useful mechanism that is found in many biological systems. It helps them navigate the environment by allowing them to focus on subsets of the presented input and filter out any input that is deemed unnecessary by the biological system at that moment. In this paper, a new method of visual flow/change detection is devised using a recurrent layer that consists of XOR gates as its basic units.

CLAIM

Nervous systems react to external stimuli to increase the survival chances of organisms equipped with them. In order to do so, nervous systems need to be capable of recognizing patterns in previous experiences, storing them, and adapting to new ones. The methods by which these neural systems achieve this have long been an interest of science.

Attention is an integral part of an animal's survival as it allows the animal to focus its computing power on a subset of the given stream of input that is relevant to surviving. For example, in the case of a prey, the animal will pay attention to changes in the environment, and if the animal recognizes a threat from that change, the animal's flight or fight system will take over to attempt to survive. In the case of a predator, the changes in the environment will help the predator run after its prey. These are just some examples that showcase the importance of attention. The main claim of this paper is: A recurrent layer whose units are XOR gates can act as a visual change detection system for an input stream.

SUPPORT

The support will be a loose one and for the specific case of two-dimensional black/white binary images, but can be generalized to any sequence of input. Let us start by inspecting the case of an animal that is looking at a field.

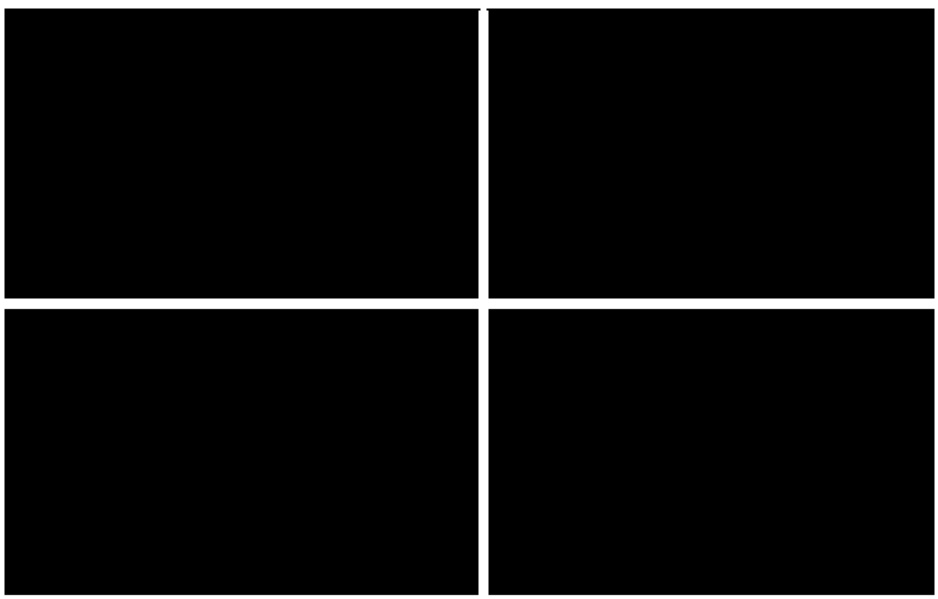

Figure 1: An 2x2 pixelated image

Now imagine the bottom-right pixel suddenly becomes active. The behaviour of most animals would be to shift attention and focus on that region.

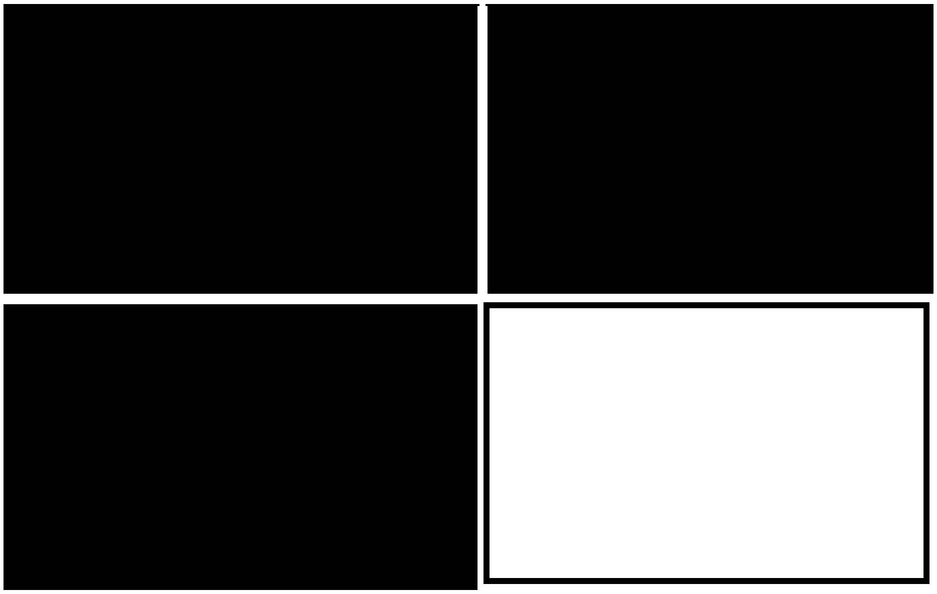

Figure 2: An 2x2 pixelated image with an active pixel in the bottom-right corner. The attention of the observing animal will be on the white region.

So we can say that, in this case, the attention was given to the sub-region that had a significant change in the input. The question that follows, then: How can we create a system that can shift attention to subregions based on changes? One way would be to simply calculate the pixel-by-pixel difference between input at time t and input at time $t+1$. Then attention can be focused on the sub-regions with highest changes. This method, however, cannot be integrated into a neural network since it does not rely on the usage of neurons to calculate differences in inputs at different times. Therefore, the method is not a plausible candidate to explain how the process of shifting attention happen biologically.

The mechanism this paper discusses calculates the pixel-by-pixel binary difference between the input at time t and the input at time $t+1$ and is a plausible candidate for how the attention shifting process happens biologically as it relies on neurons to achieve attention. To explain the mechanism, first we need to explain how an XOR (exclusive OR) gate works.

i_1/i_2 **(XOR)**	**1**	**0**
1	0	1
0	1	0

Table 1: Truth table of an XOR operator given two inputs i_1 and i_2. Rows are the values of input i_1. Columns are the values of input i_2.

As we can see from the table above, the XOR gate gives an output of 1 only when the inputs are different, and it gives an output of 0 when the inputs are the same. We can use this property to calculate the change/difference between an input at some time t and the input at the next time step $t + 1$.

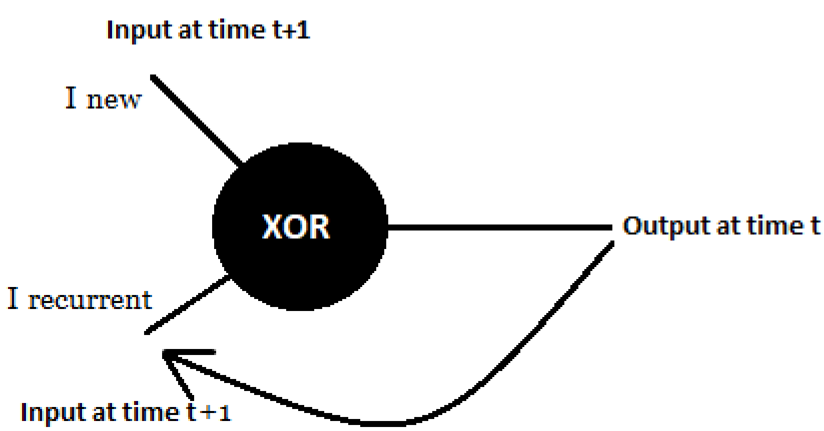

Figure 3: A Recurrent XOR unit

By feeding the output at the current time step back as an input to the next time step, we are essentially calculating the difference between the value of the input at this time step t and the next time step $t+1$. This will only activate the output neuron if the values of the input are different. Adding this to a stream of input images, for example, will only activate regions where the inputs are changing and thus will filter out any repeated information in the environment.

Formally:

Let F be the set of all black and white images/frames of size $M \times N$. So $F : \{f \in F | f = (m, n, b) \, where \, m, n \in \mathbb{Z}^{+} and \, b \in \{0, 1\}\}$. Define a map $\pi : F \times F \rightarrow F$. The map π takes two images/frames and produces a pixel-wise XOR operation on the values of the two images/frames. So $\pi(X, Y)$ works this way: $\forall x \in X$ where $x = (m, n, b)$ find $y \in Y$ where $y = (m, n, v)$ and where $m = 0, 1, ..., M$ and $n = 0, 1, ..., N$ and calculate output $(m, n, XOR(b, v))$. Now take a sequence X_i of such images where each element in $X_i \in F$. We will produce a new output filtered sequence Y_i of images where each element in $Y_i \in F$ from the input sequence X_i where $Y_i = \pi(X_{i-1}, X_i)$. A couple of properties on $\pi :$ $\pi(X_0) = X_0$ and $\pi(X_n) = X_n$.

This is how theoretically the system can work, so one way we can implement an XOR gate in a neural network can be achieved with the following configuration:

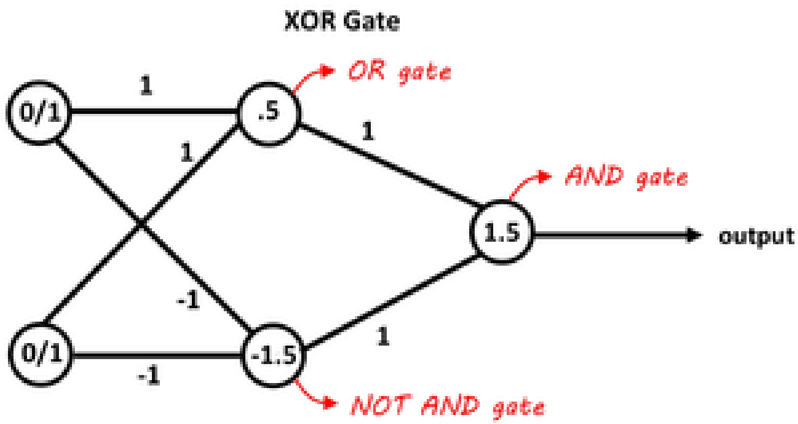

Figure 4: XOR unit implementation using a two perceptron configuration. Credit: Aditya V. D

Adding such a configuration to each pixel in the input layer of a neural network will allow it to detect changes and only pass neurons that have changed since last input was fed.

PLAN

A rigorous generalized proof for the claims made in this paper is still needed. Future work to improve the mechanism can include the following:

1. Improving on the system to handle continuous values for inputs (e.g. grayscale images) and outputs.

2. Finding ways to output values other than 0 for regions without any change in input. This comes from the observation that as we focus on objects, the other objects outside our field of focus are still visible to us and are not completely dark.

3. Looking at changes in a sequence of input data rather than just two input data at a time.

4. Looking at recurrent units of other logical gates (AND, OR, XNOR, etc.) and seeing what they mean.