Rebel Cortex 1.0

A Visual Recognition Software Project

Louis Savain

7/25/2011

I Rebel, Therefore I Am

Design document for Rebel Cortex, a visual recognition software project. Note: As this is a first draft, some of the information in this document is either missing or incomplete. This document is subject to change frequently. Keep an eye out for updates.

Introduction

Rebel Cortex (or RC for short) is a full color, visual recognition software program written in the C# programming language for the MS Windows operating system. It is based on fully scalable and biologically plausible principles of sensorimotor learning and pattern recognition. The goal of the project is to implement a recognition system that is potentially as good as a human at understanding a visual scene. I say 'potentially' only because full scene understanding is impossible unless an intelligent system has a comprehensive set of effectors and sensors with which to interact with its environment.

Possible uses of RC include not only the usual object recognition and identification tasks but also the detection of a variety of human/animal behaviors and expressions such as frowning, smiling, laughing, blinking, fear, head turning, winking, talking, walking, running, limping, kicking, sitting, lying down, standing, friendly, threatening, armed, etc. RC can be the enabling technology for a secure, video-enabled, facial recognition and identification system that cannot be fooled by still photographs.

RC consists of a movable and zoom-capable virtual *eye*, a circular *retina*, a *sensory cortex*, a *motor cortex* and an *effector layer*.
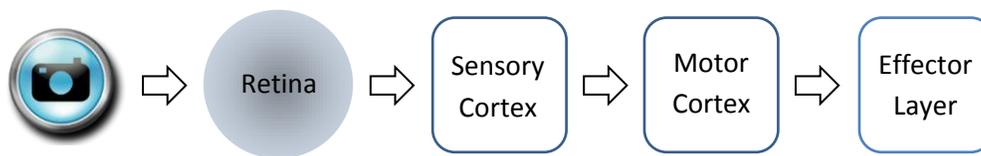

The retina is roughly patterned after the human retina. It sports center-surround perceptive fields and motion detectors. The sensory cortex has two signal processing regions called the *signal separation layer* (SSL) and the *tree of knowledge* (TOK). The SSL is a 10-stage, fixed-interval, feedback layer that separates signals from their source streams and channels them into unique paths. The TOK is a variable-interval, hierarchical memory structure that contains the system's knowledge. The motor cortex is under the direct control of the sensory cortex. For now, its actions are limited to moving the focus of the eye from one location of the input image to another and to zooming in and out of the image. The motor cortex does not do any signal processing of its own. It is essentially an action selection module. The effector layer contains *motor command* cells (or *effectors*) and a mechanism for detecting and resolving motor conflicts. The effectors directly control the movements of the eye.

It is important to realize that this design does not limit RC to visual processing. It could just as easily process auditory or tactile inputs, if necessary. Likewise, RC can handle all sorts of motor actions. In the future, RC will be expanded to handle multiple sensor and effector modalities. With the addition of a motivation mechanism RC can be an ideal brain for a sophisticated learning robot.

The following pages contain a description of the main components and subcomponents of Rebel Cortex followed by a brief discussion of the underlying theory.

The Retina

The retina consists of two layers of cells. The first layer is a circular sheet of color light sensors that receives its inputs from an external image array. Currently, the diameter of the circle is specified to be 500 pixels but this will probably change later after experimentation, depending of the speed of the computer.
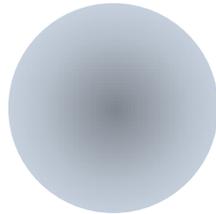
Sensor density is higher at the center of the retina.
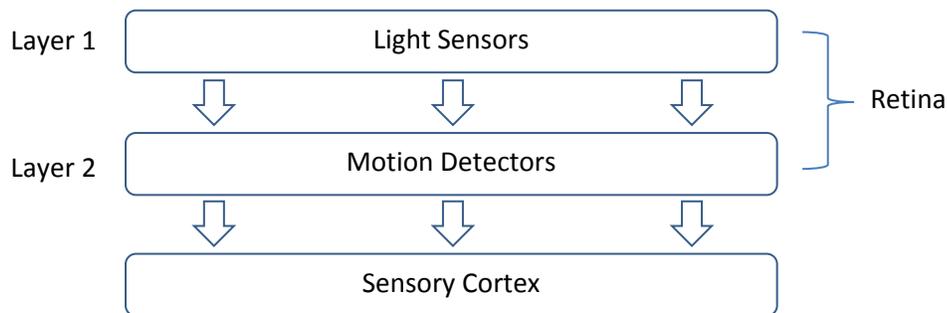
The second layer receives its inputs directly from the first layer and sends its outputs to the sensory cortex for further processing. It consists of motion detection cells that are connected to the light sensors in a center-surround arrangement. They are comparable to the retinal ganglion cells (RGC) in the human eye.
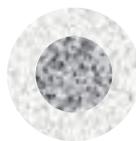
A center-surround perceptive field has an initially specified diameter of 10 pixels. The optimal size will be determined during experimentation. The fields partially overlap in order to capture fine contiguous movements from one field to another. A center-surround perceptive field makes it possible to detect the motion of a light stimulus in various directions at a small spot on the retina. The second layer sends its output signals directly to the signal separation layer of the sensory cortex. Each field contains multiple motion detectors that are connected to multiple light sensors. Every motion detector has two inputs, one from a light sensor located at the center of the perceptive field and another from a sensor located on the periphery (see description of motion detectors in the next section).

As seen in the picture above, the image area presented to the visual system is much bigger than the retina (semi-transparent circle). However, the system can rapidly and autonomously move its focus from one position on the image to another and can even zoom in and out. Furthermore, it continually moves in tiny random motions called *saccades*, even when fixated on a single location on the image. Without the saccades, the SSNs would not fire and the RC would be blind. I chose this design in order to demonstrate RC's autonomous motor behavior and its ability to focus its attention on different parts of an image. This is also a prelude to RC's eventual incorporation into an autonomous robot with a full complement of sensors and effectors.

Light Sensors

There are three types of light sensors, one for each of the primary colors: red, blue and green. All sensors respond to a light stimulus the same way, by emitting discrete signals called *spikes*. This is true regardless of the intensity of the light. Counterintuitively, the light intensity level at a small spot on the retina is not encoded in either the amplitude or the frequency of the spikes. Intensity at a spot on the retina, as perceived by the brain, is solely a function of how many light sensors are activated at that spot. In other words, activation density is a function of the level of light intensity impinging on the retina.

Note that a sensor does not fire continually in response to a light stimulus. A sensor responds only to the onset (positive detection) or offset (negative detection) of a light stimulus. Thus every sensor has two outputs, one positive and one negative. This is not exactly the way it works in the human retina but the principles are the same. Retinal sensors send their output signals to the second retinal layer where they serve as inputs to the motion detectors.

Motion Detectors

A motion detector is a special cell or neuron that detects when two signals in separate sensory streams are separated by a fixed temporal interval (10 milliseconds). In other words, it detects when two signals arrive in close succession. Motion detectors are similar to the signal separation neurons (SSNs) that are used in the sensory cortex for fixed-interval temporal learning (see below). The main difference between a motion detector and a cortical SSN is that there is no learning in the retina. Every motion detector has its two inputs hardwired to two light sensors, one of which is located in the center and the other on the periphery of its perceptive field. Half of the motion detectors attached to a given

perceptive field detect motion toward the center of the field. The others detect motion away from the center.
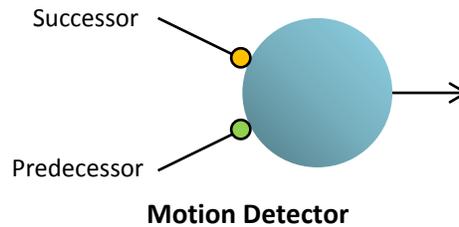
Motion Detector

As shown above, a motion detector has two inputs called the predecessor and the successor. The detector fires if it receives a successor signal a short interval (10 milliseconds) after the arrival of the predecessor signal. In order for movement to be detected, a stimulus must disappear at one spot on the retina and reappear at another spot a short time later. In other words, the predecessor input must receive a negative signal 10 milliseconds before the arrival of a positive signal at the successor input, at which time the cell fires. In addition, it makes sense that both the predecessor and successor inputs of any given motion detector are connected to light sensors of the same type: red, blue or green.

Sensory Cortex (under construction)

The job of the sensory cortex is to make sense of the many signals it receives from various sensors. It channels, stores and organizes signals in a hierarchical memory structure that is tolerant to noise, missing data and local malfunctions. The memory store must be constructed in such a way that predictions are easy to make.

The design of the sensory cortex is based on the assumption that all phenomena in the world obey certain rules and that this regularity is faithfully captured by the senses and is ultimately reflected in the temporal relationships between sensory signals. It turns out that there can only be two kinds of temporal relationships; signals can be either concurrent or sequential. This realization is the basis of perceptual learning and recognition. The sensory cortex uses two distinct processing regions to accomplish its task, the signal separation layer (SSL) and hierarchical memory (the tree of knowledge).

Signal Separation Layer

The sensory cortex receives multiple parallel streams of signals from the motion detectors in the retina. The problem is that the meaning of the signals in a stream change depending on the circumstances. For example, signal *A* may arrive 10, 20 or 90 milliseconds after signal *B* depending on the conditions. It also happens that a signal may be temporally correlated with multiple other signals. We need a mechanism that separates signals from their source streams according to their temporal correlations with other signals. After separation, the system channels the signals into unique paths for further processing. This is the job of the SSL.

Signal Separation Neuron
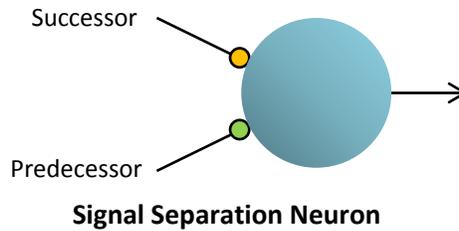
The SSL is divided into 10 feedback areas filled with signal separation neurons (SSN). Every SSN assumes a fixed 10 millisecond interval between successor and predecessor. So the only way to test for correlations over multiple intervals is to use multiple feedback loops. My hypothesis is that there are 10 such loops in the human sensory cortex for a total of 10 possible fixed intervals, the maximum interval being about 100 milliseconds. This requires 10 delay/staging areas that form a looping cascade as seen in the diagram below.
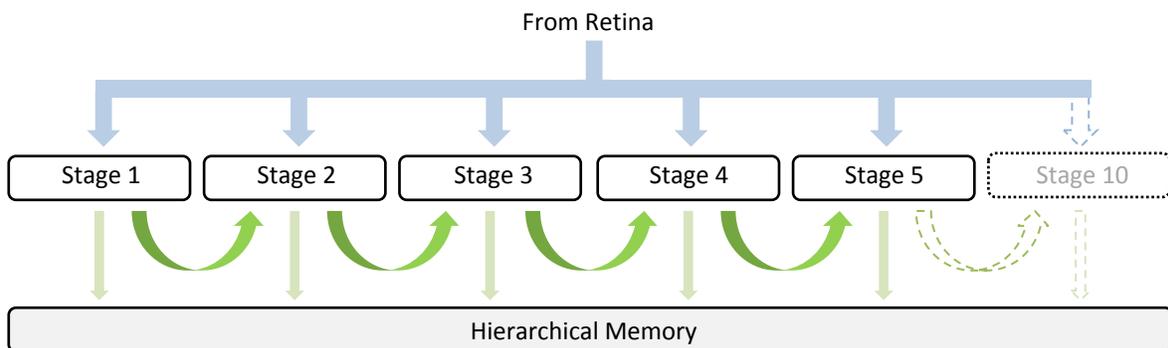
Note that all the input streams (light blue) from the retina are connected to every SSL stage. Each stage picks out a number of signals and channels them to distinct new streams (light green). Every stage except stage 1 makes only successor connections with the sensory streams. Stage 1 makes both predecessor and successor connections with the input streams. The feedback streams (dark green) make only predecessor connections. This prevents duplications. Note also that every stage (except the last one) sends its output streams to hierarchical memory in addition to feeding them back to the next stage.

Fixed-Interval Learning

Fixed interval learning in the SSL consists of using many signal separation neurons to find fixed sequential correlations between signals arriving in different streams. It is a simple process. The idea is that a good predecessor connection is one that successfully predicts the arrival of a successor signal.

**Signal Separation Neuron During Learning**
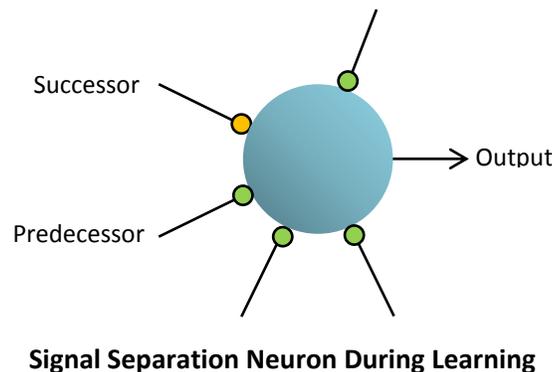
Essentially, every input stream makes single permanent successor connections (yellow) with multiple SSNs in the SSL. During learning, the SSNs make as many random predecessor connections (green) with the input streams as possible. Every predecessor connection is given an initial low strength level. Every time a predecessor signal arrives, the predecessor connection is slightly weakened. Every time a successor signal is successfully predicted, the predictor's connection is strongly strengthened. The first predecessor to reach a predetermined adult strength is declared the winner, at which point all other predecessor connections are severed. Subsequently, every time a predicted successor signal arrives, the SSN fires to signal the detection.

The assumption is that, since successful predictions are known to have occurred frequently in the past, they will occur frequently in the future. In this light, the purpose of the signal separation layer is not to make predictions but to emit signals when successful predictions occur. We can look at the SSL as a sort of signal filter or sorter, one which channels incoming signals into distinct paths according to their relative timing.

Hierarchical Memory

Hierarchical memory is what I have been calling the tree of knowledge (TOK). It is the center of the brain's intelligence because it is the repository of the brain's knowledge. The TOK is a hierarchical network of sequences that builds knowledge from the ground up. Signals from the SSL are fed directly into the bottom level of the hierarchy where they are combined into concurrent patterns.

Hierarchical Memory
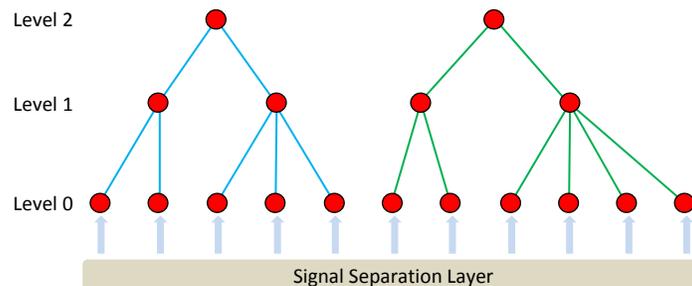
There are several powerful advantages to using a hierarchy of sequences. For one, it makes for a very compact storage system because lower level sequences are reused multiple times by higher level ones. Second, it is universal since it uses a single principle to process all types of knowledge. Third, invariant pattern recognition becomes a simple matter of activating a branch in the tree. Finally, it makes predictions, goal-seeking behavior and adaptation easy.

## Invariant Recognition, Patterns and Sequences

Only the nodes at the bottom level (level 0) of the hierarchy receive concurrent signals (aka patterns) from the SSL. The nodes at the upper levels, by contrast, receive sequential signals from the level immediately below them. This is crucial because true invariant recognition can only be accomplished in a hierarchy of sequences.

One of the things that distinguish RC from other hierarchical models is that most of them use both concurrent and sequential signals at all levels of the hierarchy. The conventional wisdom is that the nodes at the bottom level of the hierarchy are designed to recognize only very small areas of the image and that, as one goes up the hierarchy, bigger and bigger areas are recognized in terms of lower level patches, and so on. At the top of the hierarchy, the entire scene is recognized all at once. The idea is that the size of the visual receptive field increases as one goes up the hierarchy.

There is every reason to suppose that the receptive field does not change at all. Looking at the paragraph above, try as we may, we find it impossible to see everything at once. We can only focus on a tiny part of the image at a time and we have to scan the image in order to read the entire paragraph. I am proposing instead that visual recognition works just like speech recognition and that the hierarchy above the bottom level consists exclusively of sequences. The main difference is that, whereas speech recognition is passive, visual recognition requires the active participation of the intelligent system. This is the reason that RC is designed to use effectors to move its eye around the image.

The Branch, Short-Term Memory and Attention

Each branch of the tree of knowledge represents a single recognized object or phenomenon. One of the rules used in RC is that only one branch of the tree is active at a time. The branch concept is a rather important part of intelligence because it is the basis of the attention and short-term memory mechanism of the brain. Whichever branch is active is what the system is paying attention to.
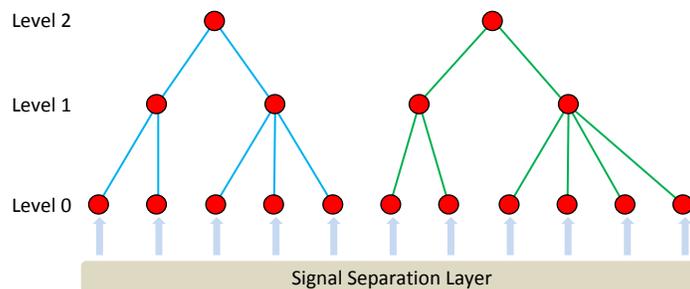

Hierarchical Memory

At any one time, every branch in the tree of knowledge is asleep except for one (or two, if counting both hemispheres of the brain). The active branch stays awake for a short time (about 12 seconds in the human brain) and immediately goes back to sleep. This allows another branch to wake up and do its thing, thus shifting the brain's attention to something else. During waking hours, sensory signals arriving from the SSL continually stimulate the nodes at the bottom level of the hierarchy. There are many branches that are always on the verge of waking up (becoming active) but they can do so only if the current active branch falls back to sleep. As soon as that happens, the sleeping branch with the highest level of sensory stimulation wakes up and takes the place of the previous active branch. There are times when a sleeping branch may pre-empt an awake branch and force it to fall back to sleep before its time. This occurs occasionally because, a) the sleeping branch may be very important for the avoidance of discomfort or harm; or b) the number of sensory stimuli a sleeping branch is receiving from the SSL is high enough to overcome the active branch. I will get back to this topic later when I discuss adaptation and motivation.

Seven-Node Sequences

The fundamental building block of memory is a sequence of up to seven nodes. Why seven? Primarily because it is the short-term memory capacity of the human brain. With the exception of bottom-level nodes, every node is a sequence of lower level nodes. Again, bottom level nodes receive signals directly from the signal separation layer.

Seven-Node Sequence
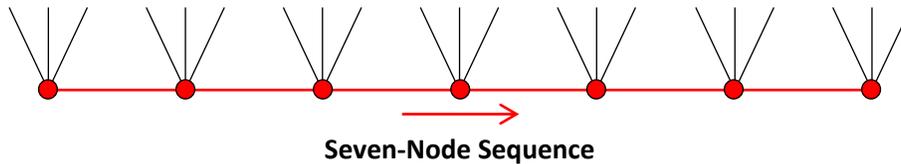
The red arrow indicates the direction order of the arriving signals. The temporal intervals between the nodes are not necessarily equal and can change at any time. Note that there are occasions (having to do with planning and anticipation) when sequences must be played back internally. It follows that there must be some kind of biological linkage between nodes in a sequence. However, sequence playback is a topic that I will cover in an upcoming chapter. Note also that each node receives multiple concurrent signals from the SSL. Most researchers in the computational neuroscience community have a bad habit of calling concurrent signals, *spatial or spatiotemporal signals*. I think it is a misnomer for the simple reason that concurrency has nothing to do with spatiality. In fact, the same mechanism is used in the processing of many different types of sensory signals, not just visual inputs.

Proportional Intervals

Unlike the SSL, the TOK assumes that the intervals between signals are both variable and proportional to one another. In other words, if the interval between two nodes in a sequence is decreased by 1/3, every other interval in that sequence must also decrease by 1/3. This is absolutely crucial because it is the only way that the brain can properly predict the timing of future events. I don't know exactly how the brain handles the timing of intervals in sequence memory but, judging from the performance of animals, athletes, musicians and dancers, I can infer that timing is very quick and very precise, probably on the order of milliseconds.

Seven-Node Sequence
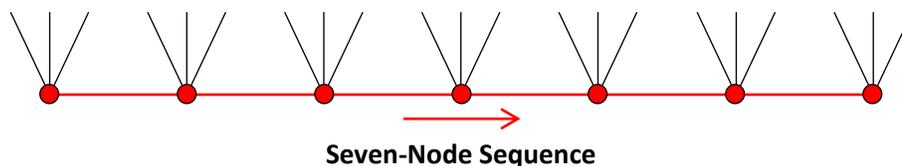
A common example of the use of proportional intervals by the brain is evident in the way we process music. When we listen to a familiar musical tune, we can immediately infer the tempo of the tune within the first few notes and we can instantly adjust our expectation of the timing of the rest of the tune. If the tempo changes, we can instantly readjust our internal timing to the new tempo. The one thing that stays invariant when an interval varies is the proportionality. Proportional intervals are also essential to scale-invariant visual recognition. When we scan the picture of a face, for example, our brains automatically change the timing of the relevant intervals to compensate for distance and magnification.

Some researchers ([Jeff Hawkins](#) and [Dileep George](#) of [Numenta, Inc.](#)) maintain that the use of time proximity as a supervisor is sufficient for invariant object recognition. I think they are mistaken. The time proximity between nodes does help in maintaining the continuity of recognition during transformation but it is not enough to handle full scale-invariant recognition.

Temporal Learning in the Tree of Knowledge

Temporal learning consists of discovering the only two types of relationships that can exist among sensory signals: they are either concurrent or sequential. Some, like the folks at Numenta, believe that the best approach is to learn commonly occurring concurrent patterns first and that only after a number of concurrent patterns have been learned, can they be tried in various sequences. I think that the best approach is to learn both types of patterns simultaneously.

There are some in the visual recognition research community who are convinced that some sort of fancy math based on Bayesian belief propagation is needed in order to compensate for the uncertainty and noise that plague the sensory space. They are mistaken, in my opinion. There is no time for complex computation during learning and recognition in the brain simply because neurons are too slow for that sort of thing.

In a way, learning in the TOK is similar to learning in the SSL: a temporal pattern that occurred frequently in the past is likely to occur in the future. The question is, how are patterns detected in a sequence? In my research, I found that the best way to design any learning system is to imagine how it might be done with just neurons and synapses. It occurred to me that if two signals are temporally correlated, they have the same frequency. This is true whether they are concurrent or sequential. This seems like a fairly straightforward principle that should be amenable to an algorithmic solution.

Node Sharing
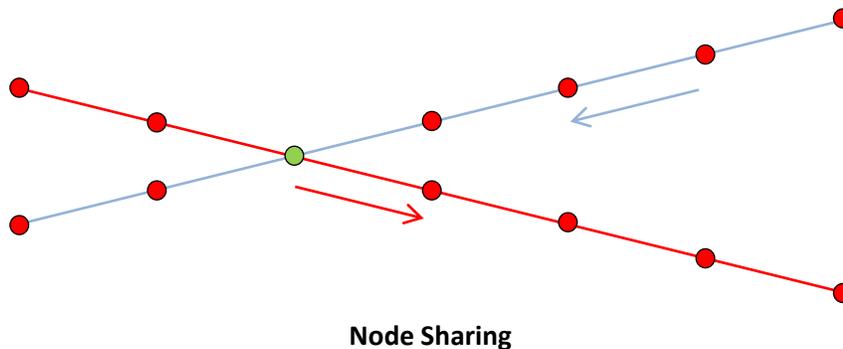
The only caveat is that a node may belong to multiple distinct sequences. That is to say, two or more sequences may share one or more nodes. Keep this in mind because this will come in handy later when I discuss temporal learning.

The Temporal Learner

How can we use these observations to design a fast and effective temporal learner? There are a huge number of inputs coming in from the SSL and the cortex needs a way to quickly connect them to as many bottom-level nodes as possible. In my opinion, the brain solves this problem with the use of two fitness criteria.

Seven-Node Sequence
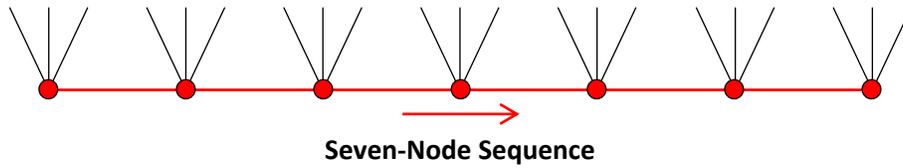
The first fitness criterion is frequency.

To be continued.

Anticipation and Pattern Completion

Coming soon.

Motor Cortex

Coming soon.

Motor Layer

Coming soon.

Adaptation

Coming soon.

Intelligence Theory

Coming soon.