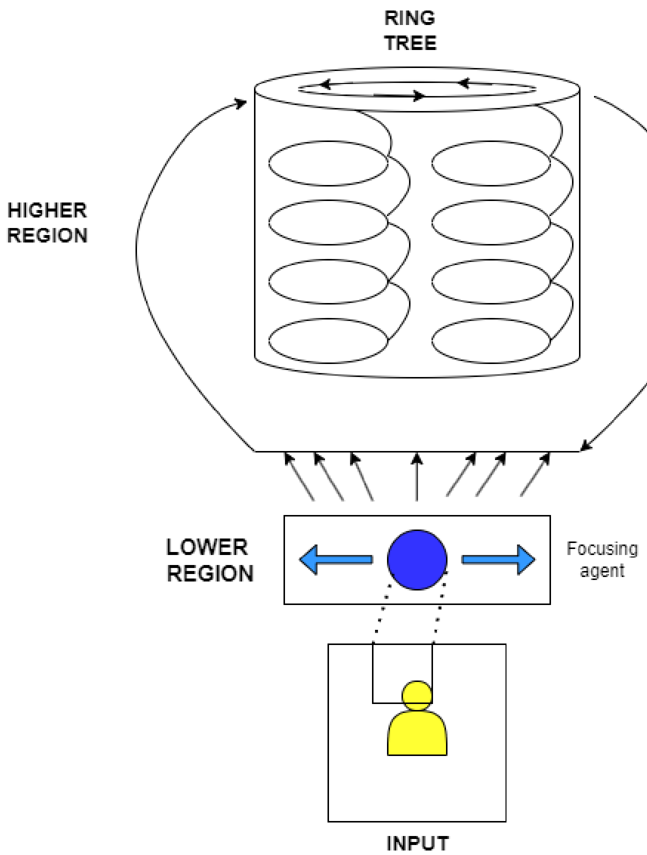
CRX Lite

This model comprises two systems: a lower region (small jump) and a higher region (higher jump). The lower region processes small bits of the overall input image and feeds them into the Ring tree. The Ring tree, receiving multiple input bits from the lower region, generates outputs by identifying similarities between the incoming inputs and stored information.

The lower region includes a focusing agent that collects the smallest truths (non-reducible forms of information) and responds to requests from the higher region to locate specific smallest truths within the input source. Additionally, the lower region possesses a short-term memory that collectively recalls these small-term memories. It then provides the correctly ordered short-term memory to the higher region, enabling it to retrieve the corresponding long-term memory in sequence.

In essence, the lower region identifies lower-level similarities, while the higher region identifies higher-level similarities within the input source.



Focusing agent

A focusing agent scans the input, processing it bit by bit and feeding each piece into the Ring. This agent moves along the input source, storing the scanned positions for future recall. Consequently, whenever the agent revisits a particular position, it will automatically move in the stored direction. This ensures that the same information, in the same sequence, can be retrieved, which facilitates the activation of long-term memory in higher brain regions. This storage of sequence is what we call the 'smallest truth,' a learned feature of any input. Because the smallest truth is an emergent, inherent property of an information processing system

Ring tree

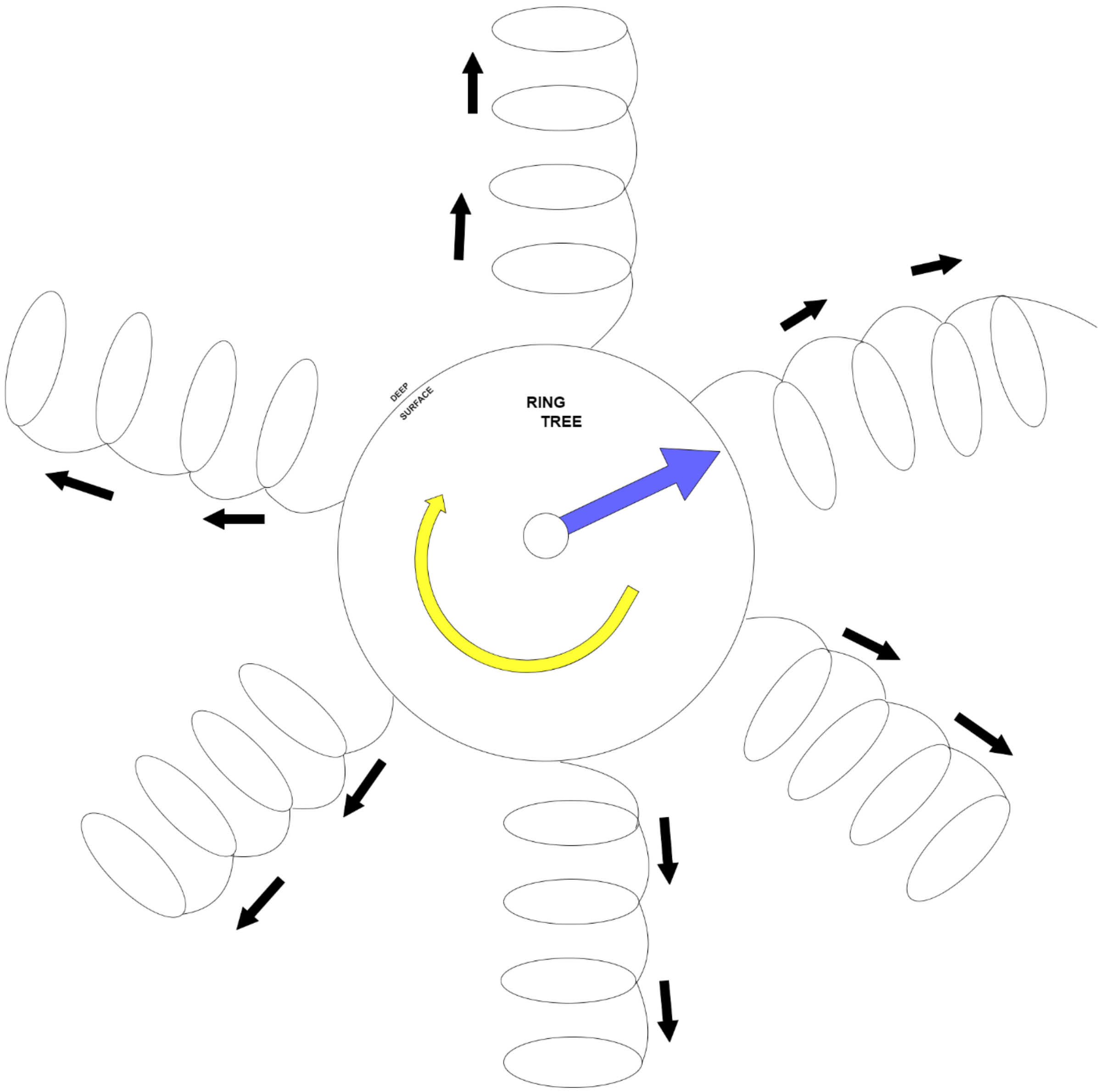
This method offers an alternative approach to creating a biologically inspired neuron model. Traditional models require tracking the position and movement of billions of neurons across each iteration, a computationally intensive task. Leveraging modern technology, we can instead extract the core logic of the brain and implement it in a more streamlined, custom-designed manner.

One crucial principle to adhere to is the linearity of thought, specifically how ring formations evolve with each piece of information. I have achieved this by simply storing each ring formation sequentially. This circumvents the substantial computational burden that the brain handles effortlessly but poses a challenge for computers. For instance, in the brain, each neuron behaves agentically, driven by receptors and the effects of neurochemical triggers. Replicating this 'aliveness' in a computer is complex; mimicking it is far more resource-intensive than the actual biological process. To emulate this, we would need to create billions of 'dead' neurons and, for each iteration, modify their behaviour billions of times. This would demand an immense computational effort. Therefore, I have chosen to focus solely on storing the ring formation sequence for each piece of information, rather than simulating neuronal 'aliveness.' This sequential storage method is significantly more efficient than attempting to simulate an entire neural network transitioning between 'dead' and 'alive' states

The cycles or rings are stored in a tree-like structure that expands as the model generates more rings in response to incoming impulses. This tree can branch off into secondary rings if multiple rings are needed from a primary ring. If inputs from the source, already stored in the main root, arrive in a different sequence, branches are created. Ring trees are structured such that activating any ring within the tree triggers the subsequent activation of rings located below it. After activating a ring, the model proceeds to activate the next in-line ring (either in a branch or the main root, as confirmed by a dual process), and this process continues until either the rings are exhausted or a change in input, detected by the dual process, shifts the model's focus to a different ring tree.

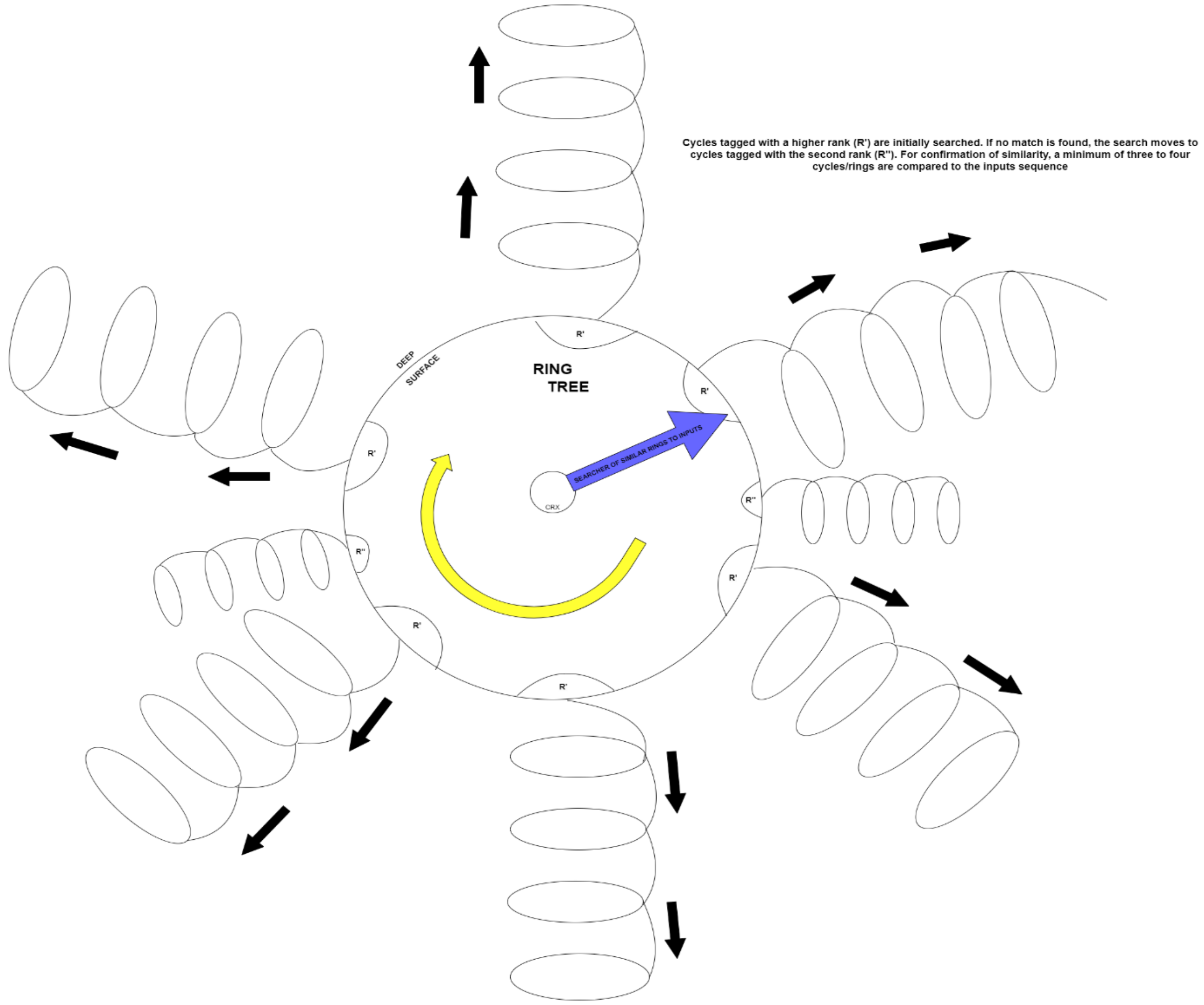
The ring trees are attached to a surface where a searcher function identifies similarities between incoming inputs and the attached ring trees, triggering their activation. To determine similarity, at least three to four rings from the ring tree and three to four bits of input are considered. Below this surface, in what we call the 'deep,' all ring trees reside, and all connections are contained within this deep layer.

Over time, the model evolves to establish intricate connections within the deep layer, where all rings are subsets of the smallest truths. Because our reality excels at providing contrast and diversity, yet also maintains similarity (as humans, being similarity-lovers, have created all things in reality based on similarities between other information), this combination leads to the activation of different ring trees with similar smallest truths, resulting in novel outputs



R’ – R’’ – R’’ – R^N

The model ranks the ring trees based on their frequency of activation and the extent of their branching, arranging them in the order R' – R'' – R''' – RN on the surface. The searcher initially seeks a match within the R' tagged ring trees. If no match is found, it proceeds to R'', and so on. The size of this R-TAG is dynamically adjusted, as the ranking prioritizes different rings based on changes in branch connections and activation counts, which fluctuate over time with varying input exposures. These Rn tagged cycles represent common information from the model's environment. By exposing this common information on the surface, the model significantly reduces search time.

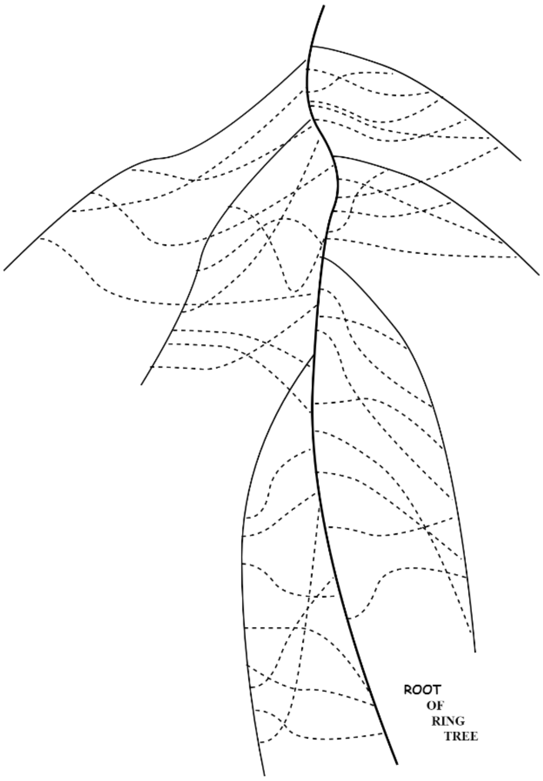


Root

The root structure expands as more information is input into the model. This information is stored in the form of roots. The root structure is advantageous because it allows us to simulate how each ring affects the formation of other rings, while preserving the sequential order of the information.

Dual process

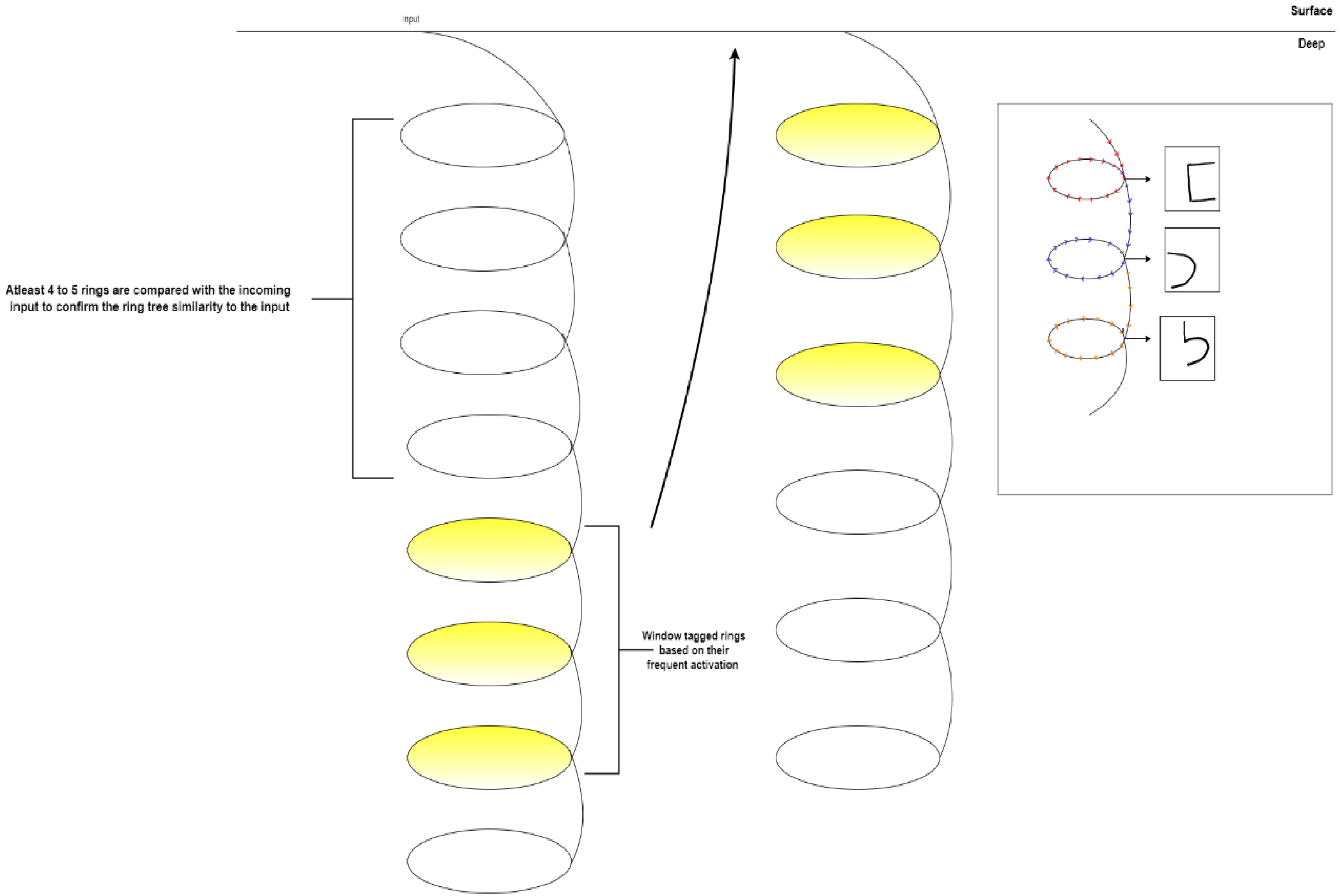
The dual process is a mechanism by which the model seeks confirmation from the input source to validate its path traversal. When a ring is activated, it triggers the activation of the next ring, which is then output by the model. This subsequent ring acts as an expected input, and the model, using the focusing function in the lower region, attempts to locate this expected input within the input source. If the expected input is found, the model continues along the path, confirming each step. If, however, the expected input is absent from the input source, the model restarts the process by activating attached ring trees at the surface, prioritizing common information likely to be present in the input source. The dual process applies to both the main root and any branches.



Storage

Information, or smallest truths, is stored sequentially, one by one, in the form of trees. These trees are composed of rings, which contain information about patterns. These patterns describe the input, output, and connections that create the pathways linking inputs to outputs. Rings are arranged sequentially, and they can form branches and connections between ring trees. The ring trees are attached to a surface, where the searcher has access to the first few nodes of each tree. The searcher then determines the similarity between the incoming input and the initial inputs of the ring tree.

Information entering the model is stored as cycles or rings, forming a tree-like structure. The model creates a 'window' on the cycles or rings of each tree based on their frequent activation. These window-tagged rings represent commonly occurring information, allowing the model to easily identify frequently received data. After a certain threshold, these window-tagged rings and subsequent attached rings gets separated and form a distinct tree structure onto the surface. Because the model prioritizes checking rings on the surface before exploring deeper roots, the window-tagged rings facilitate easier searching and the filtering of relevant information.



Retrieving

Inputs retrieve stored information by traversing down the root. This retrieval process generates separate branches as required by the input. Utilizing the dual process, the model produces an output and seeks confirmation to continue along the path. As the model progresses down the root, each node yields an output and simultaneously searches for the expected input to confirm further traversal. If a node has branches, the model pauses and expects additional input to determine the correct path—whether to continue along the main root or a branch.

Removal of stored information

Information removal is accomplished by decaying branch connections for branches, merging the most frequently activated branch into the main structure, and converting the main structure into a branch for easier removal.

The removal of a branch adheres to this rule: if a ring has a high concentration of branches, or if the number of branches on a ring increases significantly, the model completely removes the branch with the fewest activations from that crowded set of branches.

Rules for making a rings into the ring tree

As the model receives the smallest truths of the inputs, it can create rings. These rings represent collections of neurons and their connections, or patterns. When these smallest truths are stacked as rings in a tree-like structure, they form a ring tree. These trees can branch into separate ring trees if highly activated, window-tagged rings are present.

Rules for making a connections

If the input sequence finds related or similar rings, the impulse propagates down that particular ring tree as long as the dual process requirements are satisfied.

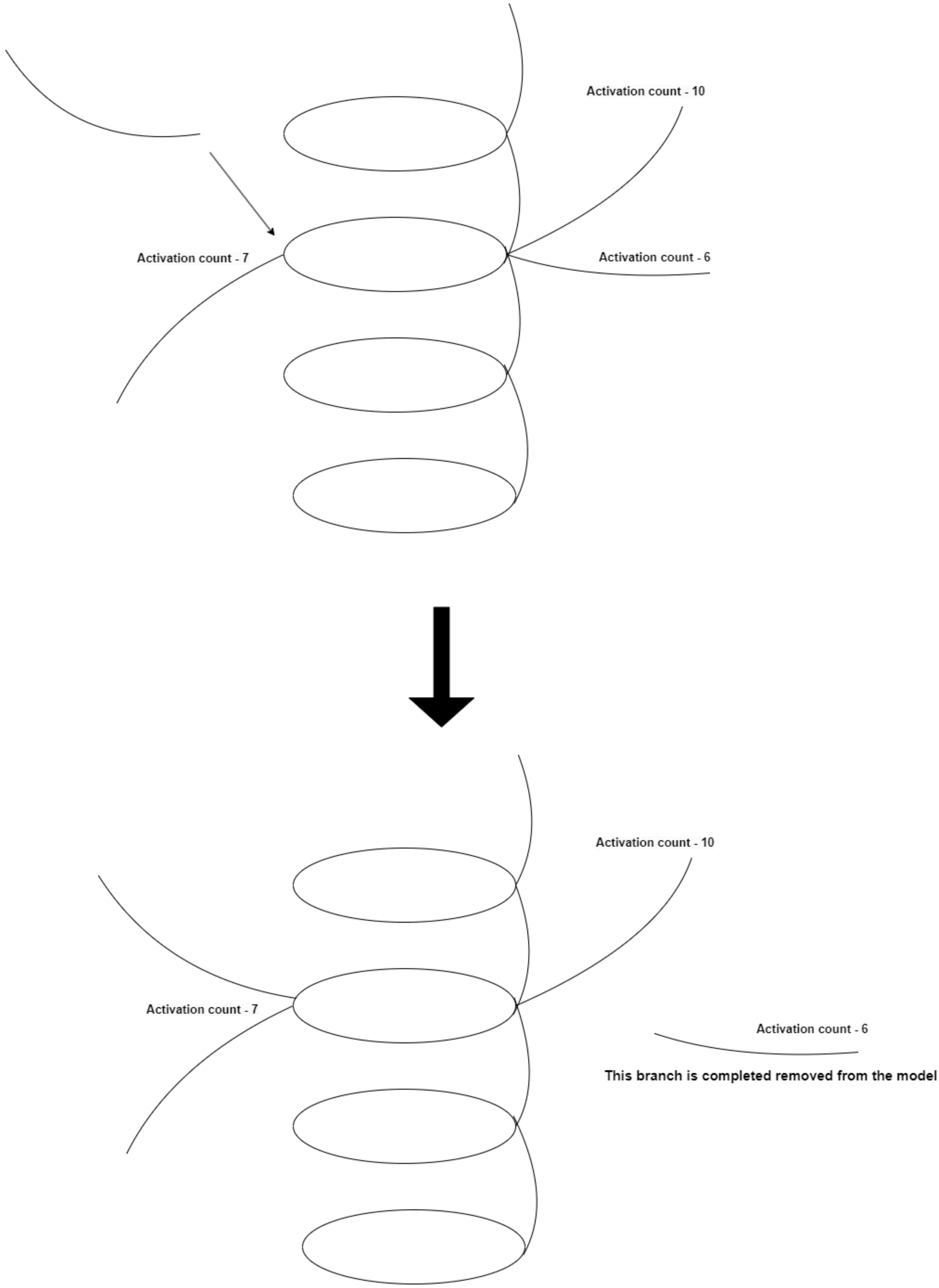
If the next incoming input cannot find a similar ring:

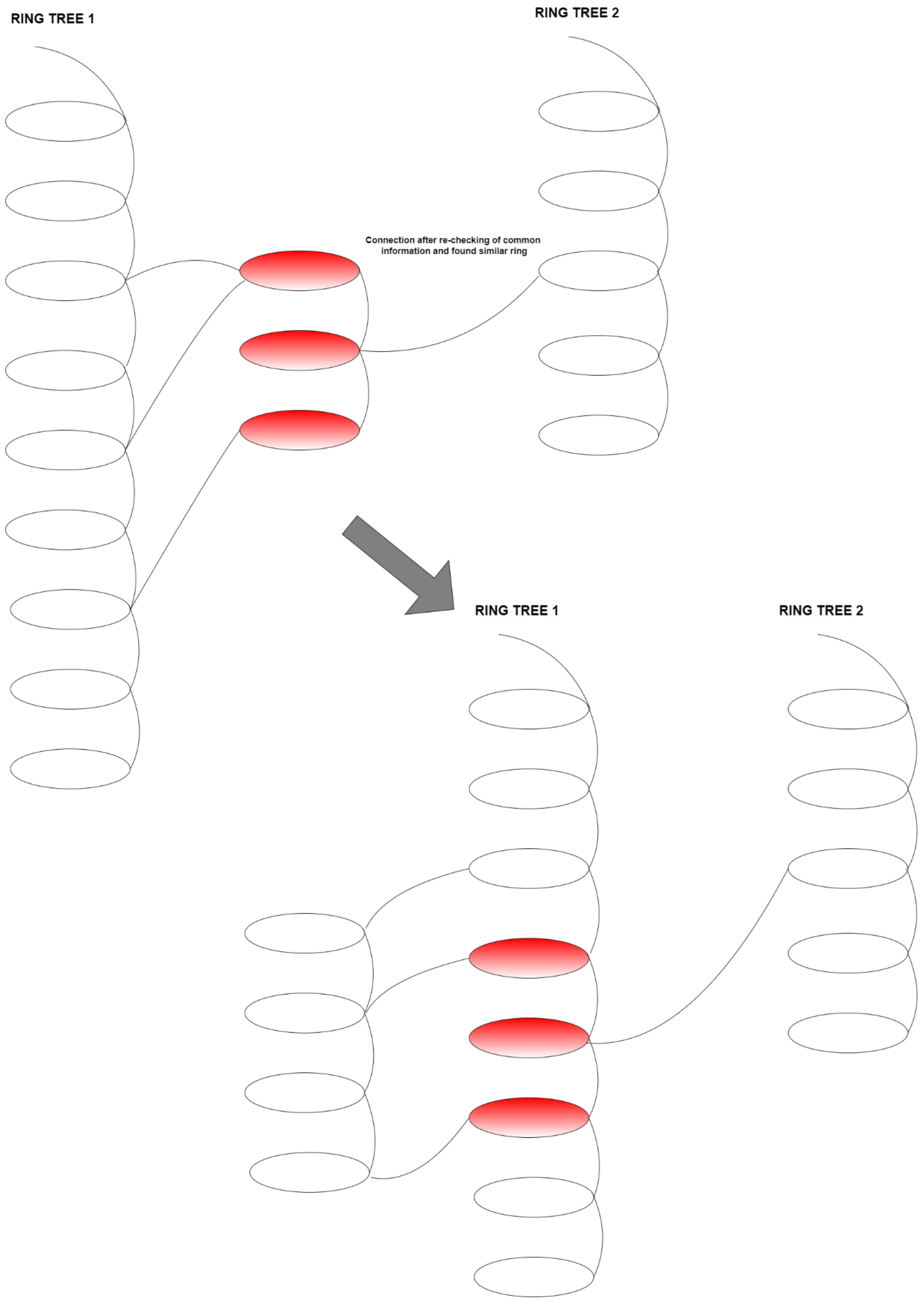
1. It searches for similar rings within the main structure of the same ring tree. If found, it creates a separate branch.
2. If no similar ring is found within the same ring tree, the model temporarily holds this path. It then retrieves another input from the source, which will contain common information. This common information activates other ring trees by finding similarities at the ranked ring tree positions. The model may encounter previously held rings within the roots. It then compares the similarities between the two held rings. If they are similar, a connection is established between these two ring trees at those specific rings—this applies to both main roots and branches when searching for residing information.
3. If no similarity is found, it is assumed that this information does not exist in the model's storage. Therefore, new rings are created and appended to the currently active ring.

Activations

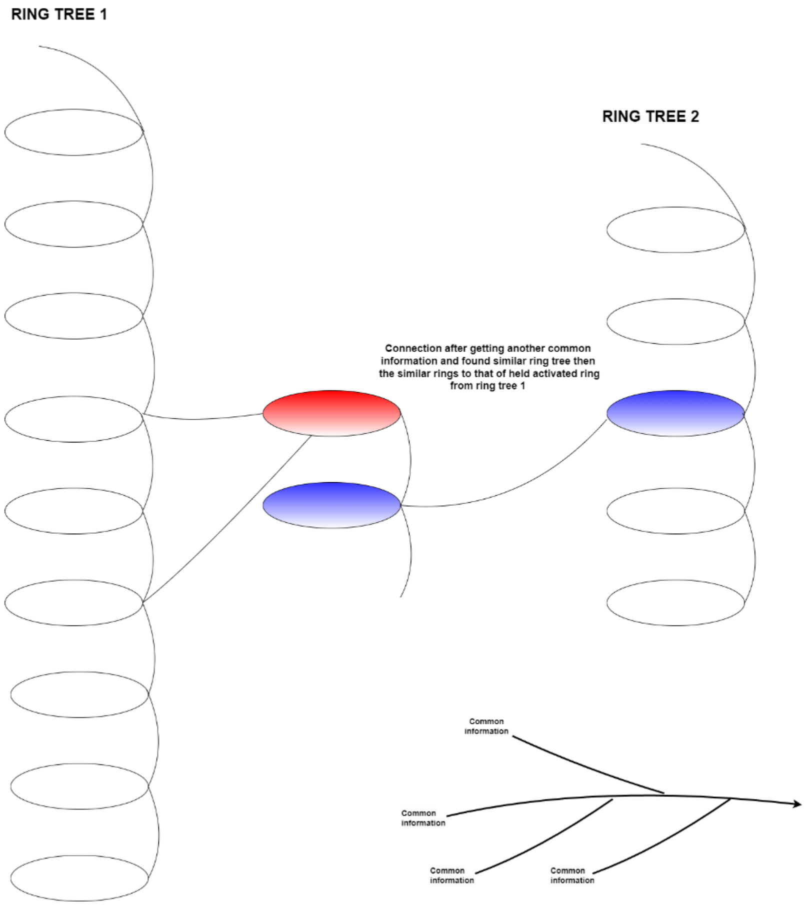
1. As an impulse travels down the ring tree, and the dual process requirements are met, outputs are generated along the path.
2. During ring processing, if a ring contains a branch, the dual process requires additional input to determine the similarity between the branch and the main structure, thereby deciding the

direction of travel. Specifically, the dual process continuously requests three to four input bits from the source and compares these bits with the bits in the branch and main structure rings. The impulse then proceeds in the direction where the higher similarity is found.

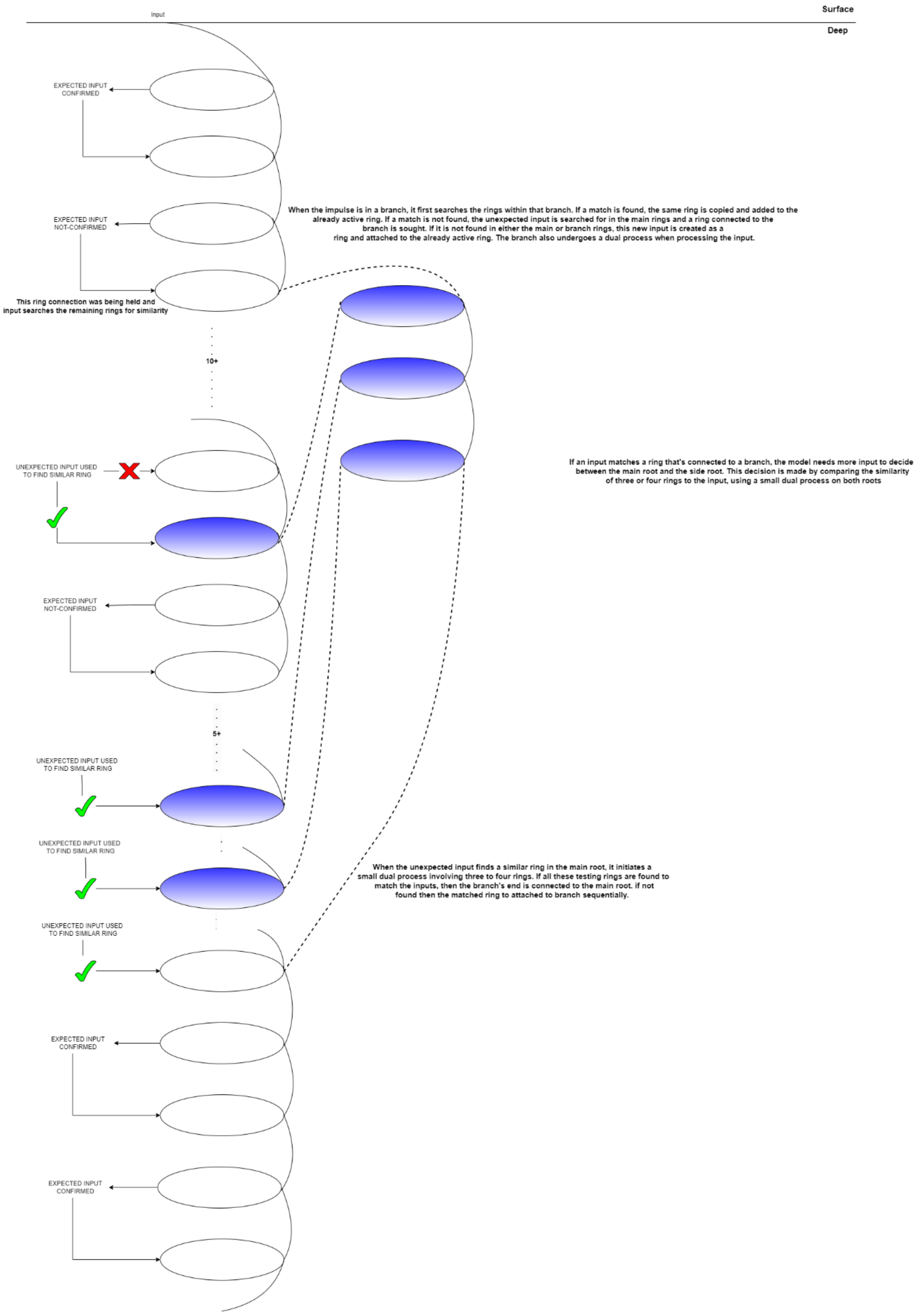



RING TREE 1
RING TREE 2
Connection after re-checking of common information and found similar ring
RING TREE 1
RING TREE 2

The model retains the activated ring after searching through the main structure and the activated branch. This rule applies to both rings activated in the main structure and in the branch. The dual process then searches for common information from the source that may be similar to the ranked rings at the surface. If common information is found that exhibits similarity to a ranked ring, the model continues using that ring tree until it encounters a point where it becomes stuck due to insufficient information, similar to a previously stuck ring tree. This ring is also placed on hold. Now, the two rings held are compared for similarity. If similarity is found, a connection is made, and processing continues. If not, a new ring is added to the existing activated ring. If no common information from the source can activate the ranked ring tree, a new ring is added under the activated ring.



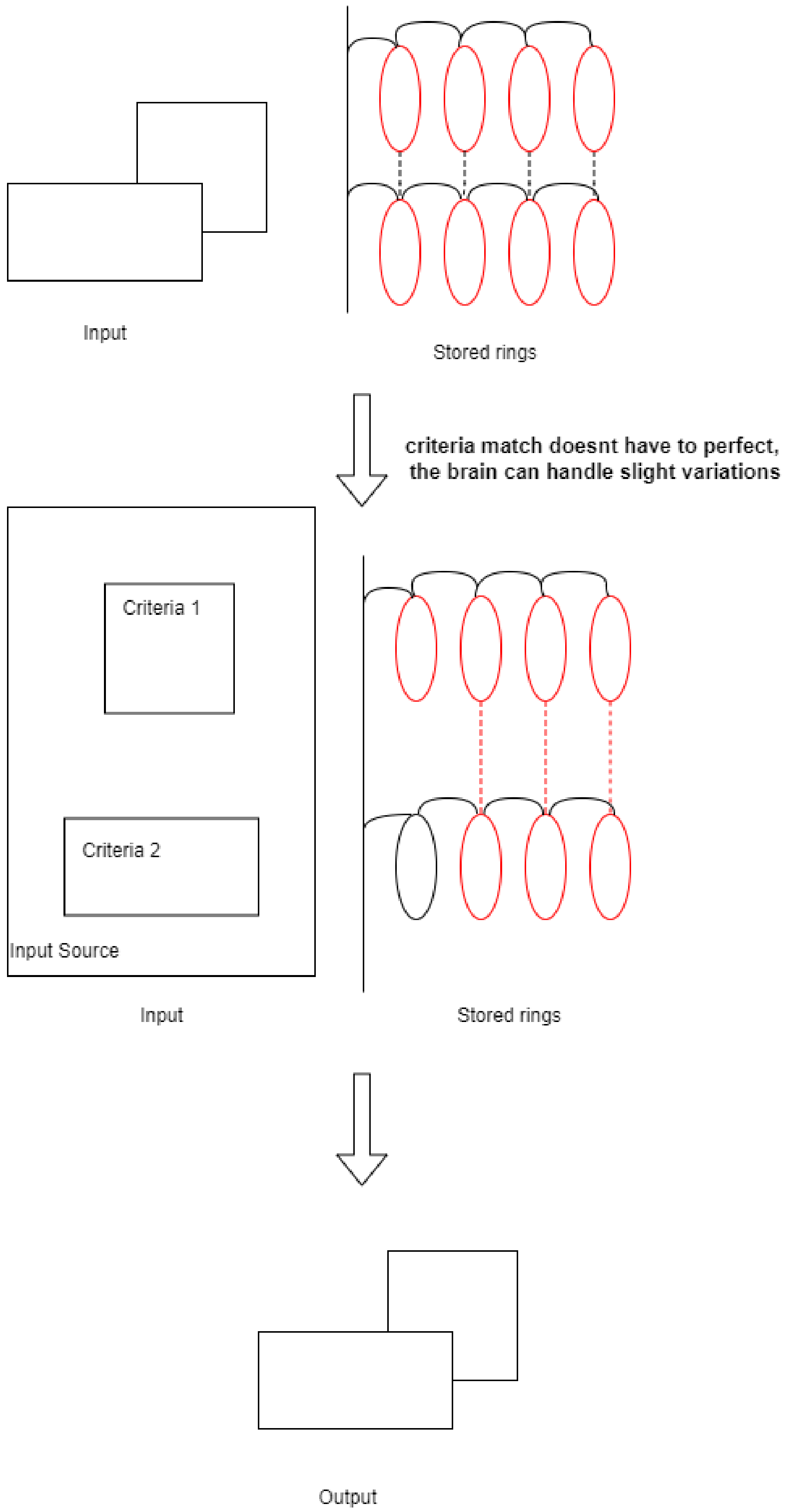
Branching is the process by which a ring tree creates a combined copy of different rings from the same ring tree, as demanded by the inputs, and retrieves them as a new kind of input. It works as follows: information finds a good matching ring tree and descends down the root in a dual process. This process produces an output and receives confirmation to continue. If the confirmation is positive, an impulse travels down the root. If the confirmation is negative, the previously activated ring is held, and the model searches the remaining rings attached to that ring tree. If a match is found, a separate branch is created, with connections from both original connections. If a match isn't found, the information is added to the already activated ring.



How this model does reasoning?

Initially, all information is stored in ring trees. Subsequently, connections between ring trees are formed based on the relatedness of their rings.

Reasoning involves applying the rules of one piece of information to another, which occurs when incoming information satisfies the criteria of already stored information (where these rules are shared).



Is Recreation being enough?

At a micro-level, the recreation of individual sets of the entire input occurs. However, at a macro-level, it may appear that something more than mere recreation is happening. This is because our brain recognizes the individual components (smallest truths) within the input source, even though we are not consciously aware of them. We do not know what constitutes a smallest truth in any given input source; only the brain does. Smallest truths are not fixed; they are changeable, evolving based on the brain's exposure to specific inputs. Smallest truths are essentially variations and contrasts that the brain inevitably processes. Through plasticity, the brain can combine two or more smallest truths into a single pathway, driven by the frequency of exposure.