University Of Athens
Physics and Information Technology

Encoding of audio information for categorizing audio samples.

MSc Thesis

Author - Registration Number
Vasileios Lianos MSC - 2014512

Advisor
Mr. Reisis D.

MSc Thesis Committee
Mr. Reisis D., Mr. Fratzeskakis D. and Mr.Nistazakis E.

July 21, 2018

Contents

Introduction

The main idea behind this thesis is to implement an algorithm that can fully simulate the way the mammalian auditory system functions. More specifically, the implemented, for this paper, system simulates all the procedures that take place since an audio wave hits the ear drum till the moment neural spikes reach the brain. Furthermore, we made use of a powerful library implemented by Numenta [28] that simulates the way brain processes and learns neural spikes and tries to predict future inputs or even detect anomalies. This tool is called Hierarchical Temporal Memory (*HTM*) and is basically a machine-learning algorithm based on neural networks. The reason why it is chosen, lies on two very promising points. Firstly, it advertises that it is a good representation of what actually happens inside the mammalian brain. The second reason is that, it promises after having processed the audio input efficiently it can deduce a comparable representation of it. To function properly this library needs input of a specific form. In other words, to harmonize the HTM library into our system, we needed to implement a proper audio encoder that would provide the essential for the library inputs, the Sparse Distributed Representations (*SDRs*).

What is important to note here is that, to utilize the algorithm of this paper, one needs to make use of a time to frequency transform that will take the time samples of an audio signal and produce its spectral content. In the context of this paper, a specially designed for this need transform is used, the AAT [1].

The following chapters will give some insights of the HTM library and the specifics of the auditory encoder implemented. Finally, the experiment designed as a proof of concept will be presented along with some results indicating that the algorithm has an effective performance and leaves many promises to future endeavours.

Athens, July 21, 2018

Vasileios Lianos MSC

Abstract

Many applications simulating the features of a DJ, when selecting a specific series of music tracks, exist nowadays. What almost all of them lack, is that the DJ can process several aspects of a track and based on how "close" this set of aspects is between two tracks, he/she can determine if one track can be followed by another inside a program. The main goal of this paper is to present a way to process audio tracks in order to extract a comparable representation of them. We could then use these representations to embed them into an Euclidean space leading to their categorization and classification.

Chapter 1

The mammalian auditory system

As stated in the introduction, an effort has been made to firstly understand and then simulate how the auditory system operates in mammals. Several aspects of it will be documented here as they were essential to copying its behaviour into our model.

1.1 Ear cochlear

The mammalian ear consists of several parts. The most significant ones for transferring the audio information into the brain in signals that the brain can intercept (neuron spikes) are the following:

- Tympanic membrane
- Ossicles
- Cochlea

The tympanic membrane, or eardrum membrane, is a thin layer of tissue inside the human ear that receives sound vibrations from the outer air and transmits it to the inner ear by setting in motion three bones (Ossicles) behind it (Malleus, Incus, stapes / Hammer, Anvil, Stirrup) which in turn amplifies and then transfers mechanically the pulse to the entrance of the cochlea (Oval window). This oval window is a membrane that vibrates, thus sending the distortion received from the three tiny bones into the inner ear or cochlea.

The cochlea is a sensory organ, part of the auditory system, that forms a cochlea (as named) which, if it was to be uncoiled, would roll out to be about 33 mm long in women and 34mm in men, with about 2.28 mm of standard deviation for the population. The cochlea is also tonotopically organized, meaning that different frequencies of sound waves interact with different locations on the structure. The base of the cochlea, closest to the outer ear, is the most stiff and narrow and is where the high frequency sounds are transduced. The apex, or top, of the cochlea is wider and much more flexible and loose and functions as the transduction site for low frequency sounds.

The cochlea is a long coiled tube, with three channels divided by two thin membranes. The top tube is the scala vestibuli, which is connected to the oval window. The bottom tube is the scala tympani, which is connected to the round window. The middle tube is

the scala media, which contains the Organ of Corti. The Organ of Corti sits on the basilar membrane, which forms the division between the scala media and tympani.

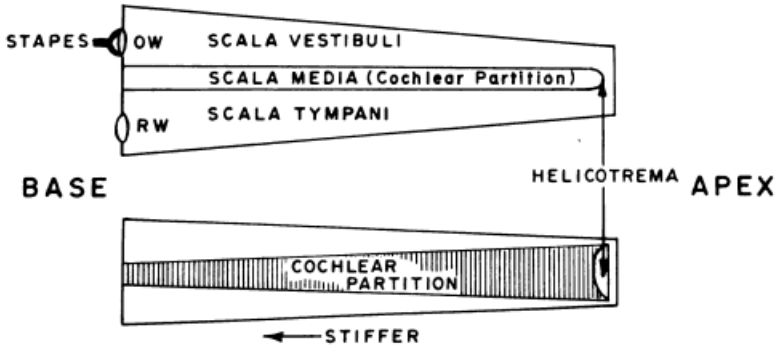

Figure 1.1: The scala in the cochlea

These three ducts (the vestibular, the tympanic and the cochlear or media scalae) are fluid-filled sections supporting a fluid wave driven by pressure across the basilar membrane. The wave is generated at the oval window and is decompressed at the round window. The scalae vestibuli and tympani are filled with perilymph, similar in composition to cerebrospinal fluid. The scala media, contains endolymph, a fluid similar in composition to the intracellular fluid found inside cells. The chemical difference between the fluids endolymph and perilymph fluids is important for the function of the inner ear due to electrical potential differences between potassium and calcium ions.

The structure of the cochlea is shown in the following figure (2).

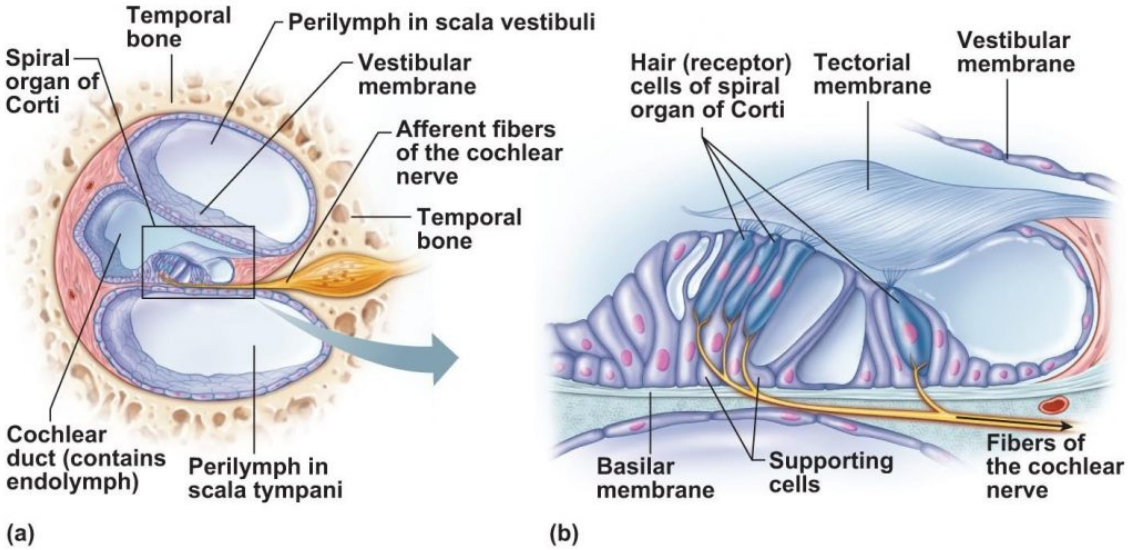

Figure 1.2: The structure of cochlea (a) and organ of corti(b)

The organ of corti is a sensory epithelium with the role of transforming audio sounds received as fluid waves through the three ducts into nerve signals. To perform this task the organ of corti has several Hair cells ,in particular three rows of outer Hair cells (OHCs)

and one row of inner hair cells (IHCs). Projecting from the tops of the hair cells are tiny finger-like projections called stereocilia, which are arranged in a graduated fashion with the shortest stereocilia on the outer rows and the longest in the center. This gradation is thought to be the most important anatomic feature of the organ of Corti because this allows the sensory cells superior tuning capability.

The Inner Hair cells in the organ of corti serve as mechanoreceptors for hearing. They transduce the vibration of sound into electrical activity in nerve fibers, which is transmitted to the brain. Outer hair cells are a motor structure. Sound energy causes changes in the shape of these cells, which serves to amplify sound vibrations in a frequency specific manner. Lightly resting atop the longest cilia of the inner hair cells is the tectorial membrane, which moves back and forth with each cycle of sound, tilting the cilia, which is what elicits the hair cells' electrical responses.

Moving the cilia of IHCs has the result of opening special conductive channels towards the inside of the cell. As mentioned before IHCs reside on the organ of corti which in turn resides in the scala media which is filled with endolymph. Potassium is the major cation in the endolymph and is thought to be responsible for carrying the receptor currents in the cochlea. The influx of positive ions from the endolymph in the scala media depolarizes the cell, resulting in a receptor potential. This receptor potential opens voltage gated calcium channels; calcium ions then enter the cell and trigger the release of neurotransmitters at the basal end of the cell. The neurotransmitters diffuse across the narrow space between the hair cell and a nerve terminal, where they then bind to receptors and thus trigger action potentials in the nerve. In this way, the mechanical sound signal is converted into an electrical nerve signal. Repolarization of hair cells is done in a special manner. The perilymph in the scala tympani has a very low concentration of positive ions. The electrochemical gradient makes the positive ions flow through channels to the perilymph.

The spectral and modal properties of the basilar membrane enables the decompose of the entering acoustic signal into individual frequency components. The signal's individual frequency components that have been "decoded" in this way are transformed in the organ of Corti into electrical signals that are conveyed to the relevant area of the nervous system through the structure of nervous fibers. However, the principle governing the transformation of the membrane vibrations in the oval window into basilar membrane vibrations presents a fundamental problem which has yet to be completely satisfactorily resolved. The aforementioned problem is truly critical in nature, as its incorrect description and definition may give rise to various misleading considerations and concepts of the system and further may become a possible cause of subsequent related errors and constructs. This may then result in totally non-functioning diagram of the function and transformation of acoustic signals while passing through the cochlea. In the context of the present paper, we will be content to just cite two prominent theories about this:

- Helmholtz's place theory, also known as the sympathetic resonance theory.
- Békésy's travelling wave theory.

1.2 Auditory System

> Every region of the neocortex performs the same basic operations. What makes the visual cortex visual is that it receives input from the eyes; what makes the auditory cortex auditory is that it receives input from the ears.

The part of the mammalian brain responsible for higher order functions is the neocortex. It is the top layer of the cerebral hemispheres, 2-4 mm thick, and made up of six layers, labelled I to VI (with VI being the innermost and I being the outermost). The neocortex is part of the cerebral cortex (along with the archicortex and paleocortex - which are cortical parts of the limbic system). It is involved in higher functions such as sensory perception, generation of motor commands, spatial reasoning, conscious thought, and in humans, language. The neocortex consists of grey matter surrounding the deeper white matter of the cerebrum. While the neocortex is smooth in rats and some other small mammals, it has deep grooves (sulci) and wrinkles (gyri) in primates and several other mammals. These folds serve to increase the area of the neocortex considerably. In humans it accounts for about 76% of the brain's volume. It is the neocortex that receives the spikes of the auditory nerve described in the previous section.

The above citation is Vernon Mountcastle's. What is important to highlight here is the homogeneity of the neocortex. Types and patterns of cells across all its span are identical regardless of the sector or the task that is assigned to that sector. In other words, even if some parts of the neocortex process audio, others process vision and so forth, all of these functions are based on the same neural procedures.

Neocortex is made up of neurons. The state of the neocortex is defined as the number of the neurons that are currently active. An active neuron is one that generates spikes or action potentials. Although the exact number of neurons is not known, it has recently came to our attention that whenever we take a snapshot of the active ones, the resulting activity diagram will be very sparse. Only a small percentage of the neurons are spiking at a given time.

There are multiple theories trying to describe the way the neocortex works. Some of them describe it as a huge database where brain refers to whenever it needs to process information received from the body-sensors (eyes, skin etc).

A most promising theory is that describing the neocortex as a hierarchical memory. This theory is embraced by the HTM machine-learning library. This theory states that our brain relies on sequential and hierarchical memory. Some simple examples of sequential memory is that we can cite the alphabet from A through Z with no substantial effort as we have memorized the exact sequence. But if we were to cite it reversely, we would face difficulties as this sequential pattern is not stored. A Another example that illustrates the power in this theory is the following sentence:

> Yuo cna porbalby raed tihs esaliy desptie teh msispeillgns.

Our brain has learned the words as sequences of letters and by combining only the first, the last and maybe the relative length of the word if can, rather easily, understand the

word at hand. If it was functioning simply as a database (of words in this specific example) it would not be able to distinguish the words.

The auditory model described in the introduction has some physical limitations.

- SPEED LIMITATION: To transduct electrical signals into the brain, cillia onto hair cells move due to sound travelling through the cochlea, which has the result of opening channels to the inner of the hair cell so that the potassium can gather inside and thus trigger the underlying neurotransmitters. If mechanically induced opening and closing of the ion channels of the stereocilia is to modulate the transmembrane potential by changing the resting current through the hair cells, then the channels of any one hair cell must collectively have an electrical impedance approximately equal to that of the base of the cell. This expectation is confirmed by the measurements of Sellick and Russell (1978), who showed that the resistance of guinea pig inner hair cells (IHCs) was reduced by at most 50% when driven at very high SPLs by low-frequency stimuli. Thus, the receptor current is determined by the state of the ion channels and by the basal properties of the cell, and the receptor potential is determined by the electrical impedance of the cell membrane. Typically, cell membranes have large shunt electrical capacitances and so the receptor potentials are low-pass filtered representations of the receptor current, restricted to rise times on the order of a millisecond or so. Because the transmembrane potential determines the afferent synaptic response, this means that a simple hair cell is incapable of encoding sounds that vary on a time scale significantly faster than a millisecond.
- DYNAMIC RANGE LIMITATION: A typical synapse needs approximately 1 mV of receptor potential to trigger the transmission, but this potential saturates at some decades of millivolts. This introduces an upper and a lower limit: the upper is introduced by the aforementioned saturation and the lower by the threshold properties of the synapse itself.

Both of these limitations would be catastrophic for hearing if it were not for the break down of a broadband signal into its frequency components from the auditory system. This is analogous to breaking a single broadband signal in many narrow-band signals and transmitting them through narrow-band channels. To do so one would need some spatially put narrow band filters. The information carried by each such channel would be easily handled as a filter can only slowly change its amplitude and phase (the rate is inversely proportional to its bandwidth). If we were to try to translate this in terms of electronics, one could easily say that a processor would be needed for the signal before it reaches the processing unit (brain) that would take the whole signal as an input, analyze it through some narrow-band and ideally spaced filters and finally give as a response the output of these filters. Once the preprocessing on the input signal is performed, the output is obtained by the modeled hair cells of the cochlea. Since, in the physical model, the hair cells are laid out along the organ of cochlea, the preprocessor component of this notion simulates this exact phenomenon. The only job left for the modeled hair cells is the triggering of the underlying neurons through their synapses.

The following figure (3) shows how hair cells are polarized and depolarized. The immediate effect of polarization of hair cells, given that the polarization exceeds a certain value (threshold), is that its underlying synapses can be triggered, resulting into producing spikes that will then be driven to the brain through the auditory nerve.

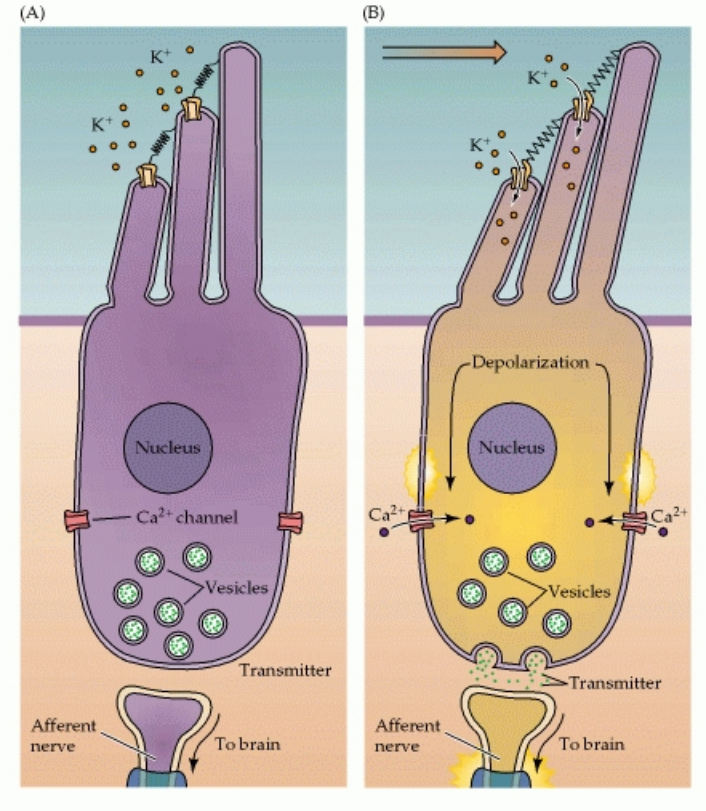

Figure 1.3: The polarization and depolarization of a hair cell

Another interesting fact is that, a refractory period applies to synapses of hair cells. A refractory period is defined as the amount of time it takes for an excitable membrane to be ready for a second stimulus once it returns to its resting state following an excitation. The refractory period in a neuron occurs after an action potential and generally lasts one millisecond.

One last detail that played a significant role in the implementation is that cillia of hair cell, due to their mechanical nature, are sensitive to the speed of the incoming wave as well as its amplitude. This led us to pass to our system the velocities of the input signal after taking the spectral content of it.

Chapter 2

The HTM library

> "In an HTM-based system, knowledge is inherent in the data, not in the algorithms"
>
> Biological And Machine Intelligence, 2017

Following the neocortex's sparse neuron activity as described before, this library tries to assign sparse representations to incoming (to be processed) input. These representations are, and from now on, called as SDRs (Sparse Distributed Representation). In essence they are vectors composed from thousands of bits. We call them sparse as the amount of -on- bits (1's) are proportionally fewer than the -off- bits (0's). Typically a 2% sparsity is chosen, according to the brain model.

A common problem to be solved for everyone dealing with AI[1] is the representation of information. Another problem is the correlations between this information. Brains though, do not have such problems as the neurons and in particular their spikes is the "language" that describes each kind of data to be processed. Another interesting fact is that brains intercept the biological SDR's without knowing the semantics of them. As long as they are correctly derived and encoded from sensory organs the brain will be able to learn and process them efficiently. This is a property respected in the HTM as well.

The significance of the SDRs is that they are semantically constructed. Let us think for example the ASCII table. The letter 'F' is represented by the hexadecimal '46' whereas the letter 'V' by the hexadecimal '56'. Although these two representations are similar by 50% they share no meaningful semantics whatsoever. With SDRs the -on- bits encode semantic properties of the processed input. If two SDRs where to share one -on- bit, that would mean that the original inputs would share a semantic property. Analogously the more -on- bits they share the more similar the inputs would be.

The aforementioned property mirrors the ability of the neocortex to make generalized representations. From little data, humans are able to predict a representation. Hearing only a few tones of a song we are able to tell which song is this. Without even having to hear a significant part of it. If we close our eyes (fewer data) only by a few touches of

[1] Artificial Intelligence.

an object we are able to find out what object we have in front of us, even to distinguish between very similar ones; if it is a cup or a pot for example.

Data are derived from our sensors; our eyes, our ears etc. These sensory organs are attached with neurons which trigger in response to the incoming stimuli and result to the sparse representation arriving in the brain. HTM needs such 'organs'. Components to interpret the physical data and construct the analogous SDR. HTM learning algorithms will work with any data as long as it is properly encoded into an SDR.

It can easily be concluded that, an SDR is a snapshot of an input at a specific time. Brains need a constant flow of such representations to function properly. Our eyes move several times in a second, we need to pass our fingers through a surface to intercept an object each time generating a corresponding representation. Thus our brain receives a flow of such representations. This is why this paper is in favour of theories stating that neocortex is a sequence learning organ rather than a database based memory system.

To learn such sequences HTM uses a specially designed algorithm called Temporal Memory (TM). This algorithm is a memory of sequences and their transitions and variations. This is beneficial and crucial for machine learning as this algorithm is capable of adapting to changes in data streams occurring real time. This is a very important feature of this library as this is immensely derived from mammalian brains and is crucial for their survival. This functionality will be extensively described in the next chapters.

2.1 From SDRs to HTM learning algorithm

Numenta, in their BAMI paper [5] also states: "The activity of biological neurons is more complex than a simple 1 or 0. Neurons emit spikes, which are in some sense a binary output, but the frequency and patterns of spikes varies considerably for different types of neurons and under different conditions. There are differing views on how to interpret the output of a neuron. On one extreme are arguments that the timing of each individual spike matters; the inter-spike time encodes information. Other theorists consider the output of a neuron as a scalar value, corresponding to the rate of spiking. However, it has been shown that sometimes the neocortex can perform significant tasks so quickly that the neurons involved do not have enough time for even a second spike from each neuron to contribute to the completion of the task. In these tasks inter-spike timing and spike rate can't be responsible for encoding information. Sometimes neurons start spiking with a mini-burst of two to four spikes in quick succession before settling into a steadier rate of spiking. These mini-bursts can invoke long lasting effects in the post-synaptic cells, i.e. the cells receiving this input."

What is essential to be stated here is that individual spiking of neurons does not matter to the neocortex functionality. It is the population of neurons that matters most. A single neuron could theoretically stop working for a period of time and the effect to the whole network would be of little importance. Having said that and taking into consideration that the rate of spiking of every individual cell is a function of its "ideal" receptive field, all HTM implementations work well up to now without implementing variable spiking of neurons. This strategy is followed in our implementation too.

A significant difference between brains (and consequently AI) and computers is the way their memory implementations work. Computers have what is called RAM (Random Access Memory) and its characteristic is that one can access a specific value of a variable stored in it provided that the address of the variable is known. In the brain there is no such thing. Data are stored associatively. Neurons spiking due to a stimulus produce sparse representations in the brain and these are linked to the next to come representations and so forth. If a recollection of such a representation is to be made, it is done through association with other such representations.

Neurons can also perform predictions. When participating in stimuli that they consider known they enter their polarized state. This does not mean that they are spiking at the moment. An excellent physical example of this is that as humans we are not aware of the process of predictions but as soon as a change of what we consider normal happens, we are instantly aware of it.

2.2 Encoding SDRs

After obtaining data from the input signal, HTM requires a proper encoding to take place. In the context of the present paper, the input data is the sound. The cochlear encoding that is implemented for the purposes of this paper is based and mirrors the biological model. The following figure (5) depicts exactly this fact.

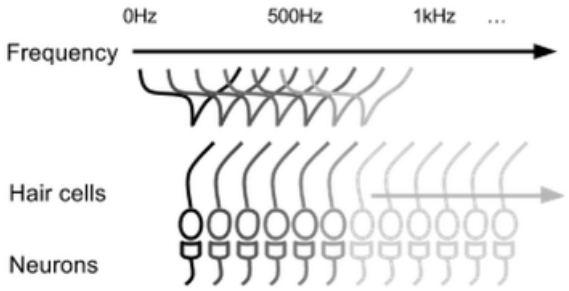

Figure 2.1: Biological hair cell encoder

A proper encoder for HTM algorithms should respect the following restrictions, as described in their papers:

1. Semantically similar data should result in SDRs with overlapping active bits.
2. The same input should always produce the same SDR as output.
3. The output should have the same dimensionality (total number of bits) for all inputs.
4. The output should have similar sparsity for all inputs and have enough one-bits to handle noise and subsampling.

By modeling the cochlea and the way it has hair cells arrayed along its length and by letting them to have synapses which can spike according to the input stimulus they receive, we ended up with a proper encoder that can produce sparse representations of the input signals. Below, an effort is being made to present the steps we follow in order to properly encode input signal.

We first try to analyze the signal to its participating frequencies. To do so, we utilize a transform specially designed to simulate the biological model. This transform is the AAT mentioned in the introduction. Then, we obtain from this transform the velocity of the input signal as in the cochlear, it is the velocity of sound that mostly stimulates the cilia on the hair cells.

After we have obtained the spectral content of the input, we pass it on the "auditory encoder" implemented for the needs of this paper. We implemented an entity to simulate the hair cell that will take a fragment of the input data that corresponds to the "cochlear" area assigned to that cell. According to the values of the input, it will be able to gather "potential". According to this gathered value it then has an array of topologically put entities representing the hair cell's synapses which are picked and triggered to spike. A synapse that has just spiked is then put into a refractory state for a given period and is prevented from spiking for as long as it is in this state. It is clear up to this point how close to the biological model is the implemented encoder. The output of all these synapses from the hair cells collectively is the desired encoding of the input signal.

The encoder is not implemented to match the sparsity requirements of the HTM library. It simply operates as the hair cells get stimulated enough to produce spikes. To meet with the required, by the HTM library, sparsity we have introduced a new component; the condenser. The condenser's sole job is to gather SDRs as time passes and simply OR them altogether until a desired sparsity is met. It is essential here to remind the reader of the very useful characteristics and dynamics of the UNION[2] property that accompany SDRs.

After having encoded the input signal into an SDR sequence, an effort is being made to end up with one representing SDR of the whole above sequence. The HTM library promises this possibility, as it can receive large sequences of SDRs and by processing them it can deduce them to a single one. To perform such deductions, it associates several aspects of the input and tries to learn sequences of it.

In the following figure (6) the whole system is displayed.

Following the neocortex's hierarchical structure of regions, HTM functionalises two components for learning and deducing SDRs, the spatial pooler and the temporal memory.

2.3 The spatial pooler (SP)

Spatial pooling of HTM tries to model the function of neocortex that assembles SDRs from different sensory organs and yet manages to learn such sequences. It is basically the interface of the temporal memory which perceives the input data and transcodes them into properly encoded SDRs.

SP is formed to respect some basic properties. The first property is to always form fixed-sparsity representations of the input signal. This would basically mean that such SDRs are similarly recognized by the HTM learning algorithm causing the same spikes. This makes the algorithm more robust and fail tolerant. Another property is that the system makes use of all available resources for the learning process. This leads to the fact

[2]Refer to the Appendix A.

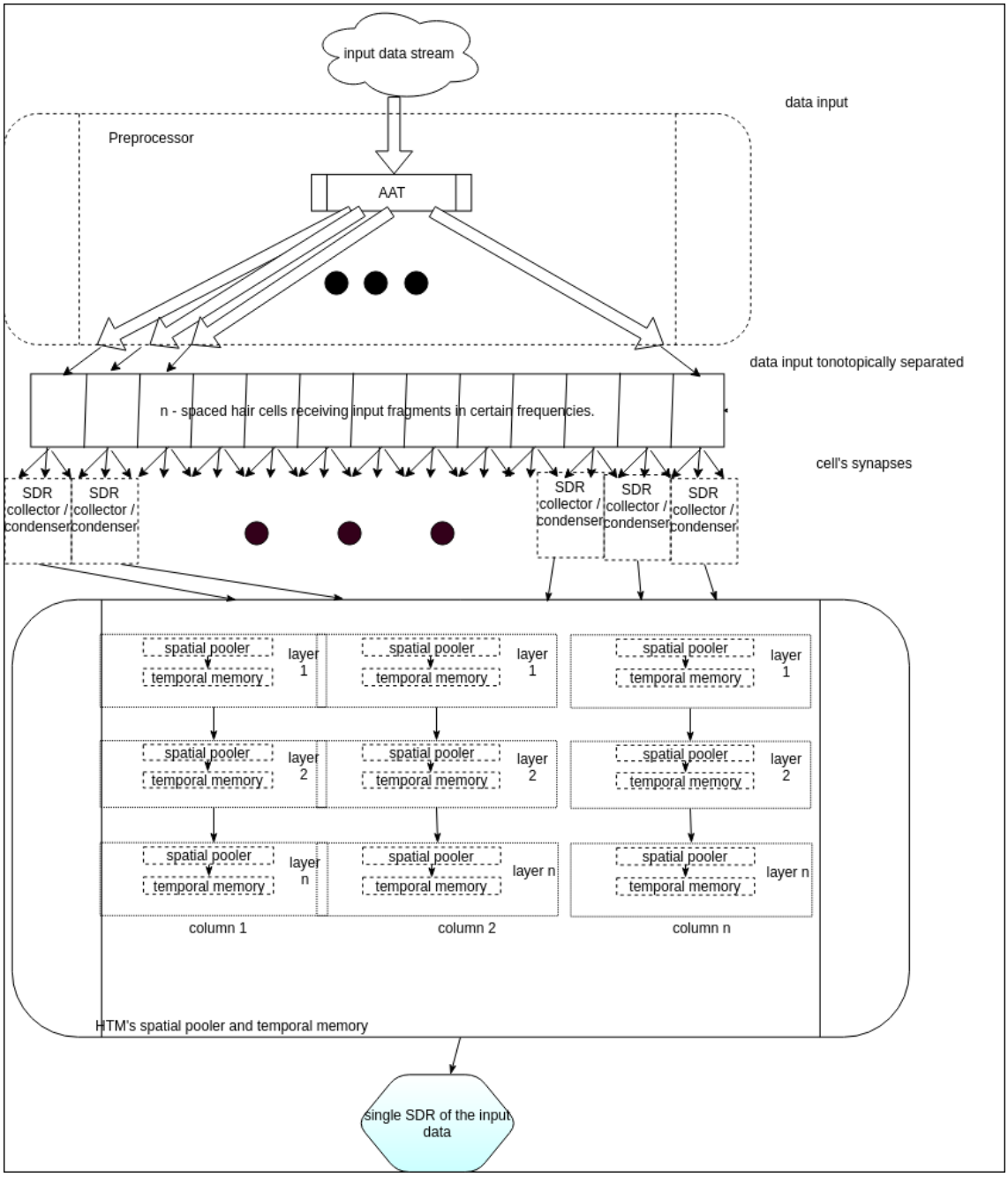

Figure 2.2: Auditory Encoder

that it is prevented to have neurons that are more active than others, i.e if the system has a given number of neurons it is preferable to utilize specific fragments of them for spiking in response of a sequence resulting to having almost all neurons being used. An also important property that SP is introducing to produced SDRs is that, it makes them noise robust, in a fashion that the output representation is relatively insensitive to small changes in the input. A fourth property is the flexibility. Like neocortex SP is sensible to changes of the input statistics. This property is particularly important for applications with continuous data streams that has fast-changing statistics (Cui et al., 2016a).

2.4 The temporal memory (TM)

Neurons in the neocortex have thousands of excitatory synapses. Some of them, the proximal synapses, have a large effect on the probability of a cell generating an action potential. However, a majority of them, the distal ones, do not influence this probability as much. These synapses act as processing units and respond to spatially located activations. The activation of several distal synapses within close spatial and temporal proximity can lead to a local dendritic NMDA spike and consequently a significant and sustained depolarization of the soma (Antic et al., 2010; Major et al., 2013). This fact has led researchers to believe that these synapses act like pattern-recognising and learning units.

HTM theory models this behaviour by introducing a system capable of learning patterns in incoming SDRs despite the amount of noise and variability inserted. The next figure (7) depicts exactly this fact.

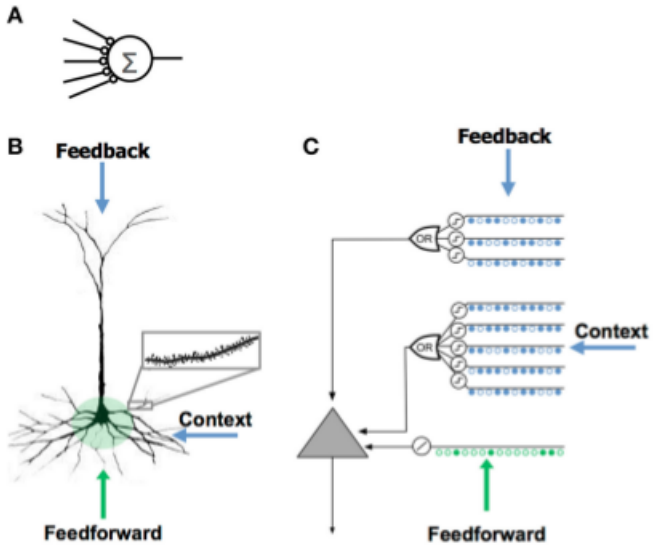

Figure 2.3: Pyramidal neurons {B} as opposed to what is usually used in AI {A}. {C} offers an insight on how input is to be passed to such cells.

Neurons in this model receive three kinds of input. It divides the synapses of a cell into three zones (according to the biological model), the proximal the basal and the apical. The proximal receives the feedforward data, the basal contextual data (from nearby cells in the same cortical region) and the apical feedback input. As proposed by the creators of HTM,

this division serves as follows:

- Proximal synapses define the classic receptive field of a cell.
- Basal synapses learn transitions in sequences.
- Apical synapses invoke a top-down expectation.

Hawkins and Ahmad stated in their paper[3]:

> "We then propose a neuron model where patterns detected on proximal dendrites lead to action potentials, defining the classic receptive field of the neuron, and patterns detected on basal and apical dendrites act as predictions by slightly depolarizing the neuron without generating an action potential. By this mechanism, a neuron can predict its activation in hundreds of independent contexts. We then present a network model based on neurons with these properties that learns time-based sequences".

In the same paper they support that, temporal memory can be used to implement a sequential memory system that can be used for predictive applications as well as learning systems. This aspect of TM is being used to feed sequences learned and processed from adjacent neurons so that we can state that every neuron is "put into context". It will not operate by processing a segment of the input signal as it was separately received. It will try to associate the information with the information from the immediate neighbours.

Temporal memory of HTM learns sequences of input data and tries to project this process on to higher levels. To do so, it enlarges the input SDRs by a factor, thus giving each bit of the input a column of bits. When something is patronized by the system it projects the input bit in some bits (usually just one) in corresponding column. When something is entirely new, due to a newly inserted sequence of data or simply due to an anomaly in the data, it projects it to the whole column. The latter is called a burst. What's more, the TM can operate in two modes; the learning and the inferring. A sequence should always firstly be passed at learning mode for some times and then passed at inferring mode so that it can efficiently induce the desired outcome.

[3]Hawkins and Ahmad, 2016.

Chapter 3

The learning network

In the neocortex, regions are organized into columns and layers. Layers are interchangeably connected through the same columns and among their neighbour columns. With respect to that, HTM library proposes a model structured the same way. It is important to note here that, there is yet to be discovered why exactly there are such connections between layers in neocortex. This is the main reason why in almost all artificial neural networks such connections are not present.

In HTM, layers are organized in a hierarchical way to model an input object. From layer to layer several *sensations* of the object are being monitored and encoded into what can be perceived as a *feature* of the object. Such features can be passed as a feedforward signal for the next level layers or as context for the same level layers. On the top level layer these features combined can finally encode a whole object. On our implementation, the last layer perceives all encoded aspects of the object and unifies them (using the UNION property of SDRs) to end up with a single representation of the object holding every feature of it.

Each layer consists of an SP, a TM and a Union component which ensures that the SDRs reaching each layer are of some parametrized density. Using the latest library of HTM that enables TM receiving contextual and feedback input as well as feedforward, we ended up with the following setup for our end experiment (figure 8).

For the needs of this paper, we simply present a single layer and the ability it exhibits to learn audio data input encoded in sequences of SDRs. We will leave the implementation of the full system, illustrated in figures 7 and 8, for future work, as there are a series of issues in the current version of the library, but promising updates are announced. To demonstrate that a single layer can learn a sequence we will observe the behaviour of the output of the TM component. There should not be much bursting. Spiking should be done in a mostly polarized way.

3.1 Parallelling the network

The first version of the implemented algorithm was sequential. This led to several hours of waiting of a single tone to give results. To reduce such delays we implemented a message exchanging parallel system to handle every aspect of the problem. We introduced entities

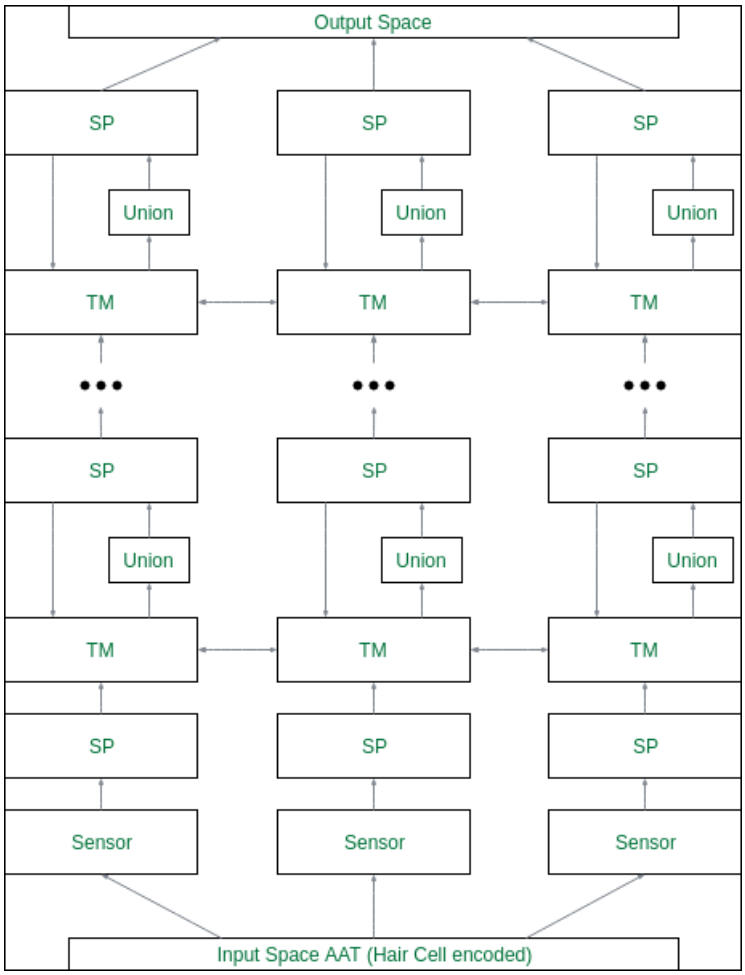

Figure 3.1: Auditory Neural Network

wrapping all key methods of hair cell encoding and HTM processing, that would be invoked upon receiving appropriate messages. This gave us the ability to induce encapsulation and thread distribution upon our code. This is the main reason that enabled us to having nicely arranged pipelines of workers each and every one of them performing its duties independently.

Chapter 4

The experiment

Below, we present the steps followed during the experiment.

1. First, we take the audio sample. For the proof of concept we have chosen some single tones for a rather narrow time window as computations are still rather time consuming. It is crucial to note here that we have run our experiments with some audio samples taken from the **GTZAN genre collection**[1].
2. We pass this input to a specially designed time to frequency transform, that simulates the mammalian auditory system as described above.
3. We then obtain the velocities of the outcome of the transform. This is the input of our hair cell encoder.
4. We stimulate the hair cell encoder implemented for the needs of the present paper and we obtain a series of bit streams representing the spiking each auditory neuron is sending on the auditory nerve.
5. This sequence of bit streams is collected and then passed through the SDR learning memory system after condensing it to meet the requirements of the library used.
6. Coherently with the practises proposed by the HTM we firstly pass the input several times as the learning step through the first layer of the system and then when we are confident that it has learned it sufficiently, we pass it once more this time by configuring it to infer an output. This is then to be passed to the next layer and so forth.
7. To realize that the experiment works we simply calculate the percentage of columns in the outcome of the temporal memory that do not contain bursts. According to the HTM library and documentation, the TM bursts a column with on bits when it encounters a sequence or a pattern for the first time or out of context.

For simplicity and for faster calculations we narrowed down the system to one that handles only a single octave. In addition, this was essential to take more specific measurements as a single tone in a given octave is possible to have spectral components in adjacent octaves. Given that the human ear can perceive 10 octaves (approximately 20 Hz - 20 kHz), we

[1]This dataset was used for the well known paper in genre classification "Musical genre classification of audio signals" by G. Tzanetakis and P. Cook in IEEE Transactions on Audio and Speech Processing 2002.

chose the middle (fifth) one (~261Hz). We also chose to use only two columns for this experiment.

At the beginning, we experimented with a single tone right in the middle of this octave (F5), then with two tones at the edges of the octave (D5+A5). We proceeded with two tones that were closer to the middle of the octave (E5+G5) and finally with the three tones right in the middle of the octave (E5+F5+G5). These experiments were chosen for demonstrating differences that can be induced to measurements according to the task in hand. In other words, specific parametrization must take place in order for the system to perform optimally.

As already highlighted, HTM is yet in an early development and research state so we faced some issues using it.

Our system has two layers with two groups, each of them receiving their neighbour's outputs and transmitting to them their own. To perform such transactions we needed to save the state of our components. Such states, would gradually grow bigger as we were passing the audio input through the network. After a certain point and size HTM library fails to perform serialization and deserialization of such objects.

Given that it would take some repetitions for the first layer to learn the input and stop bursting, the second layer would receive a sequence as feedforward input that was not coherent during this time. This would also be retransmitted back to the first layer as feedback. This would not be a problem as eventually, the first layer will manage to learn the input, thus providing to the second layer a stable representation. This would lead the second layer to learn the input after some time.

We include measurements of the experiment with the two layers to demonstrate that even with the aforementioned problems, the second layer seems to slowly learn the input as it gradually stops bursting.

To avoid this problem we repeated the experiment with only one layer. Measurements taken indicated that the system converged rapidly to sufficient numbers.

Chapter 5

Presenting the results

The first diagrams (Figure 9, 10) represent the first experiment where two layers were being used. It is obvious that the first layer manages to learn sufficiently the input whereas the second will need much more tries to learn it at the same level. What is promising here is that second layer's curves are ascending, indicating that, eventually, this layer would learn the input as well.

However, the second diagrams (Figure 11) demonstrate the good ability of a single layer to learn an input. We can observe that it took much fewer times that the input had to be passed through the system for it to reach the learning levels of the first experiment, and in some cases it reached significant higher levels (i.e Figure 11 - D5-A5 line).

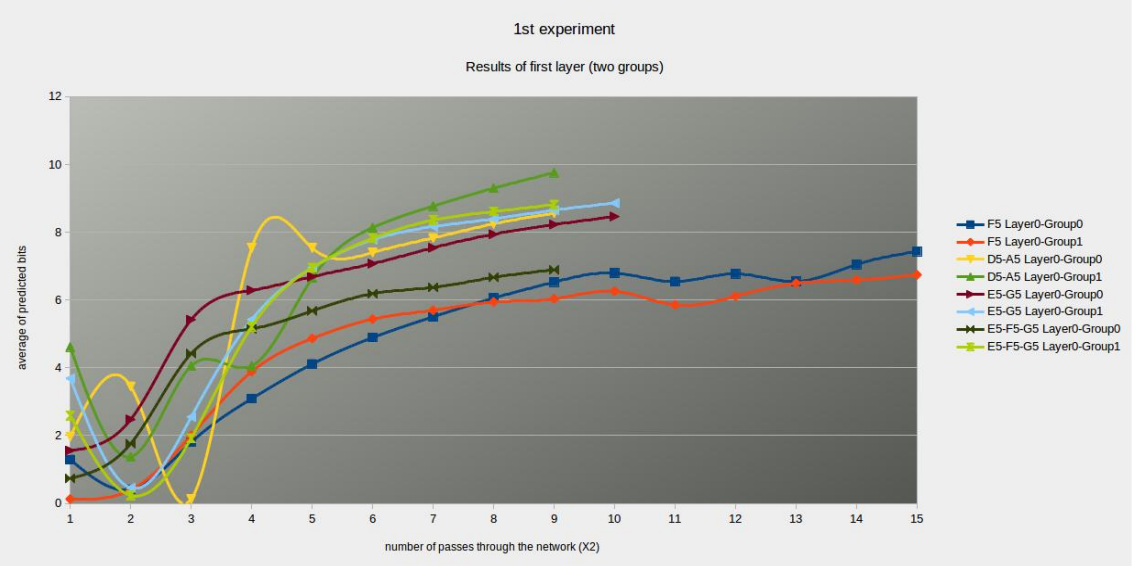

Figure 5.1: Combined diagram of the 1st layer measured during the first experiment.

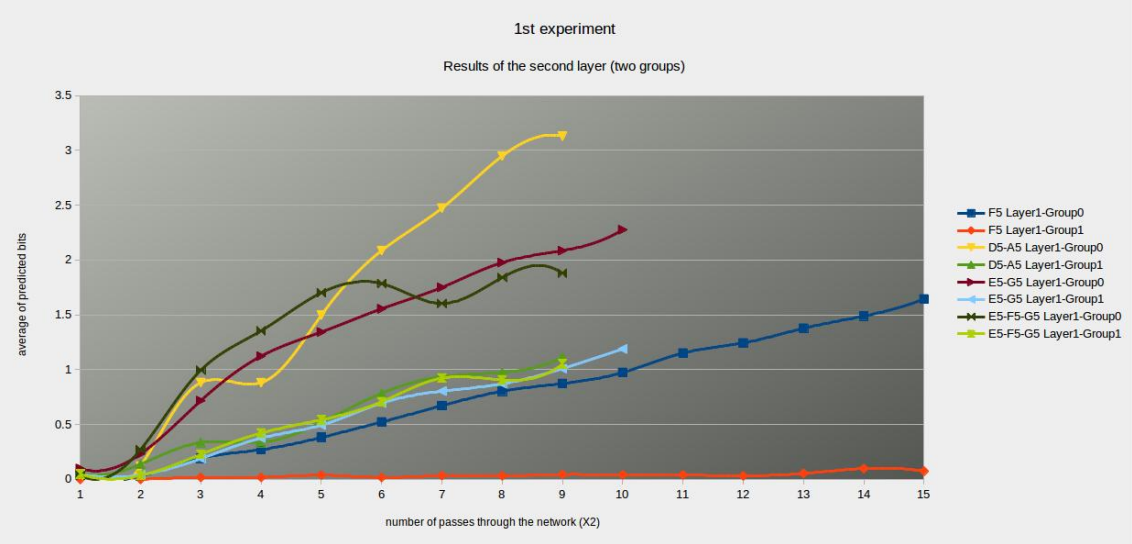

Figure 5.2: Combined diagram of the 2nd layer measured during the first experiment.

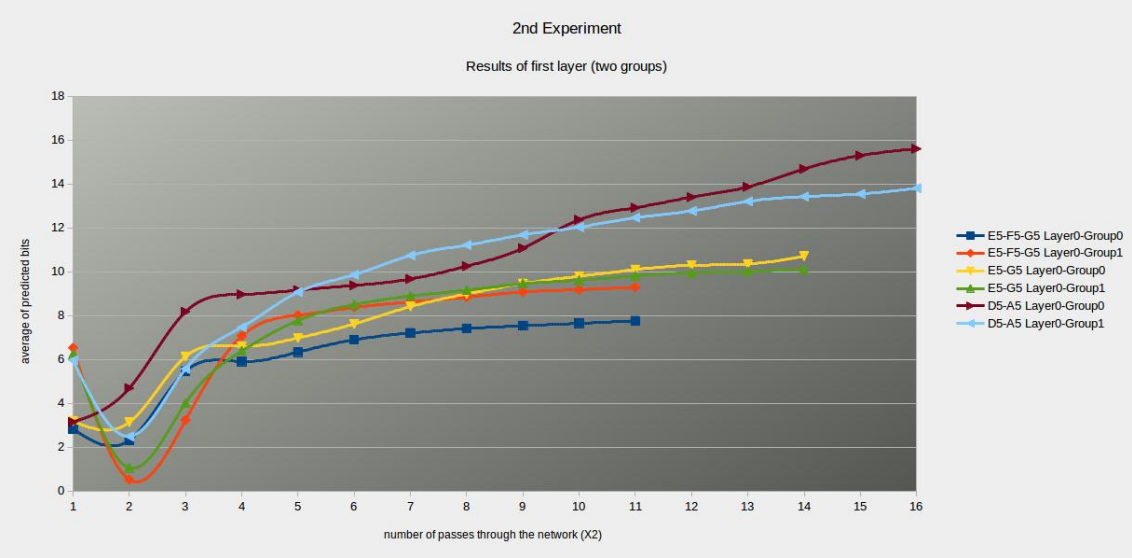

Figure 5.3: Combined diagram of the only layer measured during the second experiment.

Chapter 6

Future work

As stated before, the current version of HTM presents some blocking issues for our goal. As announced by the creators of the HTM, significant updates are expected to be released.

In these updates, they will fully implement the functionality of the learning process. Currently, the deduction of a single representation is not implemented. It can rather associate the input with a random pre-provided representation. We are also expecting to fully resolve the serialization issues, described above.

These two issues are the main reason why we did not manage to completely implement the system from figures 6 and 8.

When these releases will be available, an effort will be made to fully implement our designed system and will be put into use for extracting a single representation from each input data. These representations will then form a new sequence of input data, that will represent a whole genre for example. When having done this for a satisfying amount of residents of these genres we will end up with a single representation for each genre. At that point, the comparison of a SDR of a single candidate of a genre with the representation of this genre will define a probability of this candidate being a member of that specific genre. Thus, we will end up with an application that will be able to determine whether a track belongs to a group of tracks sharing some characteristics. What is very promising here, is that the accuracy of the application will continuously grow as more and more tracks are embedded into the groups defined.

Chapter 7

Authorship

For the authorship of this paper, the framework Markdown has been chosen, which was compiled with Pandoc, using the TEX Live and/or the MiKTEX LATEX compiler.

The template was inspired by the **Thesis template in Markdown**[1]. Special thanks to the authors of this project are due, as they made their project publicly available thus making the formatting of the present paper easier.

[1] https://github.com/FTSRG/thesis-template-markdown

Bibliography

1. “Analysing audio signals for audio-sorting applications”, Fragoulides 2018
2. “Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex”, Hawkins and Ahmad, 2016
3. “The HTM Spatial Pooler— A Neocortical Algorithm for Online Sparse Distributed Coding”, Hawkins, Ahmad and Cui, 2017
4. “A Theory of How Columns in the Neocortex Enable Learning the Structure of the World”, Hawkins, Ahmad and Cui, 2017
5. “Biological and machine intelligence - BAMI”, Numenta 2017
6. “Hearing - 2nd edition”, Brian C. J. Moore
7. “Learning a metric for music similarity”, Slaney, Weinberger, White, International Symposium on Music Information Retrieval, September 2008.
8. “Machine learning approaches to music similarity”, dissertation by Brian McFee, University of California
9. “Analysis and synthesis of music using the auditory transform”, John Paul Stautner, MIT
10. “Robust similarity metrics between audio signals based on asymmetrical spectral envelope matching.”, Mathieu Lagrange, Roland Badeau, Gaël Richard
11. http://www.music-ir.org/mirex/wiki/MIREX_HOME
12. https://nba.uth.tmc.edu/neuroscience/s2/chapter12.html
13. https://www.omicsonline.org/open-access/the-standing-acoustic-wave-principle-within-the-frequency-analysis-ofacoustic-signals-in-the-cochlea-.php?aid=80276&view=mobile
14. http://sites.music.columbia.edu/cmc/MusicAndComputers/
15. http://wondergressive.com/neocortex-how-human-memory-works/
16. https://en.wikipedia.org/wiki/Neocortex
17. https://en.wikipedia.org/wiki/Cochlea
18. https://en.wikipedia.org/wiki/Hair_cell
19. https://en.wikipedia.org/wiki/Refractory_period_(physiology)
20. http://www.cochlea.org/en/
21. https://www.youtube.com/watch?v=yVT7dO_Tf4E
22. A series of online lessons of HTM available on “https://www.youtube.com/user/OfficialNumenta”
23. A course “Artificial Intelligence” available on “https://ocw.mit.edu/courses/electrical-

engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/index.htm"
24. A course "Sensory Systems" available on "https://ocw.mit.edu/courses/brain-and-cognitive-sciences/9-04-sensory-systems-fall-2013/"
25. https://www.sciencedaily.com/terms/neocortex.htm
26. http://mir.ilsp.gr/
27. https://librosa.github.io/librosa/
28. https://numenta.com
29. http://marsyasweb.appspot.com/download/data_sets/
30. https://github.com/FTSRG/thesis-template-markdown

List of Figures

Appendix A

Mathematical properties of SDRs

As stated before, SDRs have semantics of the input data encoded in them. Keeping this in mind, if two representations share at least one active bit, this means that the two inputs have a common characteristic feature. Let us think the following: the letter 'r' is represented by '01110010' and letter 'Q' by '01010001'. First of all, it is obvious that these representations are not semantically encoded. Another aspect is that if one was to store such values, it would be essential to store the whole information. In case of sparse representations (in our case 2% sparse) we would be able to limit that to the extent of the sparsity by storing only the places of the active bits.

A very surprising feature of SDR's is that even by storing only a random subset of the total number of the active bits, we can still end up with a fidel representation of the encoded input data stream. This could easily lead to some cases of a false-positive match but on the one hand this is extremely rare and on the other hand, two SDR's sharing some exact same active bits even if they differ by the rest of them, would mean that they are semantically similar.

Another very useful feature, deriving from the sparsity of such encodings, is the UNION feature. Obtaining a new SDR simply by OR-ing some given SDRs would result to a new one which would be more dense but would keep essentially every characteristic of the originating ones. This is very helpful as this feature makes possible to determine if an input is part of the original SDRs used to form the unified one. Again we could be driven to some false- positive deductions but due to the high sparsity of the SDRs contained in the UNION, the possibility of such a result is very low.

Let all SDRs be considered as binary vectors and denoted as x = [b0, ... bn-1] for the SDR x. With n we denote the size of such a vector, meaning that it has an amount of n positions. The sparsity s of the vector will be the fraction of the n bits that will be active. The cardinality w of this vector will be the amount of the active (1's) bits. To compare SDRs we will use a property called **overlap** and it will be the amount of active bits that two SDRs share in common places. The matching of two SDRs will be determined with a threshold and will imply that the two SDRs match sufficiently.

An SDR of size n and cardinality w can represent a total number of unique SDRs given by the following formula $\binom{n}{W} = \frac{n!}{w!(n-w)!}$. It must be stated here that although the amount

of representations of these sparse SDRs may be much smaller than some denser ones, these SDRs can represent astronomical numbers of encodings. For example with a typical value of $n = 1024$ and $w = 20$ (2% sparsity) the total number of representations is *5.47994E+41*.

The probability of two SDRs (x, y) being exactly the same is $P(x = y) = \frac{1}{\binom{n}{w}}$ which is considerably small. So we can state that such SDRs are distinct.

To compare two SDRs we use the overlap notion. Firstly let us examine the effects of SDR matching. Let x be an SDR encoding of size n and with w_x on bits. If we were to calculate the amount of elements in a set of SDRs of size n and with w bits on, sharing b bits in common with the x vector we would have to calculate the following formula: $|\Omega_x(n, w, b)| = \binom{w_x}{b} * \binom{n-w_x}{w-b}$.

As stated before, we match SDRs by setting a threshold of how much we want them to be alike so we can be sure that they are the same representation. If we set this threshold to the number of on bits w, we would be intolerable of noise in our system. A single bit of noise would end up in a false-negative outcome. In general, lowering this threshold allows our implementation to be less sensitive and more noise robust. Of course by doing so, we increase the probability of false-positives. The false positive probability of two SDRs of size n and cardinality w can be measured by the following formula: $f_{p_w^n}(\theta) = \frac{\sum_{b=\theta}^{w} |\Omega_x(n,w,b)|}{\binom{n}{w}}$ Some indicating values of this is that with size 1024 and cardinality 32 this probability is ~10^{-18} whereas with size 2048 is ~10^{-22}.

As we explained earlier an prominent advantage of SDRs is that we can rely on subsampled SDRs to compare. Let x be an SDR and x' be a subsampled version of x. Whilst it is self-evident that x' will always match x, we search the probability of false positive matches of x' with vectors that are not x. $f_{p_{w_y}^n}(\theta) = \frac{\sum_{b=\theta}^{w_{x'}} |\Omega_{x'}(n,w_y,b)|}{\binom{n}{w_y}}$. Some indicating values of the above formula are that having a total number of bits $n = 4000$ and a number of subsampled bits $w_y = 32$ the false-positive probability is ~10^{-18} whereas with *n=2000* the probability decreases to ~10^{-13}, both extremely small probabilities.

The following figure (4) illustrates the UNION property.

We want to determine the probability of false positive matches based on the UNION property. Although self-determining, it is vital to note here that, the probability of false negatives is zero. Let M be the number of the OR-ed SDRs, n the size of them, w their cardinality and let θ be the similarity threshold as described previously. The probability of an exact match (θ = w) will be: $p_0 = (1 - s)^M$, where s = $\frac{w}{n}$. The false positive probability therefore will be: $p_{f_p} = (1 - p_0)^w$.

This technique is similar to the derivation of the false positive rate for Bloom filters (Bloom,1970 Broder and Mitzenmacher, 2004). An indicating value of the above is that with $M = 4$ and $n = 20000$ the probability of false positives rises to ~10^{-13}. Of course by lowering the threshold θ, one can make the implementation more robust to noise.

$$\boldsymbol{x}_1 = [01000000000010000000 \ldots 010]$$
$$\boldsymbol{x}_2 = [00000000000000000010 \ldots 100]$$
$$\boldsymbol{x}_3 = [10100000000000000000 \ldots 010]$$
$$\vdots$$
$$\boldsymbol{x}_{10} = [00000000000000110000 \ldots 010]$$

$$\boldsymbol{X} = \boldsymbol{x}_1 OR \boldsymbol{x}_2 OR, \ldots, \boldsymbol{x}_{10}$$

$$\boldsymbol{X} = [11100000000110110000 \ldots 110]$$

$$\boldsymbol{y} = [10000000000001000000 \ldots 001]$$

$$\therefore match(\boldsymbol{X}, \boldsymbol{y}) = 1$$

Figure A.1: UNION property