A Hubel Weisel model for hierarchical representation of concepts in textual documents

Kiruthika Ramanathan (kiruthika_r@dsi.a-star.edu.sg),
Shi Luping (shi_luping@a-star.edu.sg), Chong Tow
Chong (chong_tow_chong@dsi.a-star.edu.sg)
Data Storage Institute, (A*STAR) Agency for Science,
Technology and Research, DSI Building, 5 Engineering Drive 1,
Singapore 117608

Abstract

Hubel Weisel models of the cortex describe visual processing as a hierarchy of increasingly sophisticated representations. While several models exist for image processing, little work has been done with Hubel Weisel models out of the domain of object recognition. In this paper, we describe how such models can be extended to the representation of concepts, resulting in a model that shares several properties with the PDP model of semantic cognition. The model that we propose is also capable of incremental learning, in which the knowledge is stored in the strength of the neuron connections. Degradation of old knowledge occurs as new knowledge is introduced to the system in a fashion that simulates decay theory in short term memory. The simulation model therefore captures several properties of cognitive conceptual memory, including generalization patterns, the role of rehearsal and, hierarchical representation.

Introduction

There exist several bottom-up approaches to hierarchical models of object recognition that are based on the visual cortex. They make use of Mountcastle's (1978) theory of uniformity and hierarchy in the cortical column and the model of simple to complex cells of Hubel and Weisel (1965), modeling how simple cells from neighboring receptive fields feed into the same complex cell, meaning that the complex cell has phase invariant response.

In this paper, we consider the following question. If the structure of the cortical column is uniform and hierarchical in nature and if the model of simple to complex cells can be used to model the visual cortex as discussed in prior works, then can such a model also be used to represent other modalities of information such as the concepts derived from text? We are therefore aiming to design a bottom up hierarchical memory for the representation of concepts, much the same way as it is designed for the representation of images. In this paper, we will define a concept as being a keyword in a document.

To deal with the dynamic nature of concept inputs, we look at incremental learning of concepts from two aspects relevant to concept representation from text – (a) with respect to new incoming features and (b) training of hierarchies. To perform this, we apply a set of geometric approximations to the incremental inputs and the existing memory, such that the new memory can be acquired without damage to the old ones.

Related work

Mountcastle (1978) showed that parts of the cortical system are organized in a hierarchy and that some regions are hierarchically above others. In general, neurons in the higher levels of the visual cortex represent more complex features with neurons in the IT representing objects or object parts (Hubel and Weisel, 1965). Hubel Weisel models have therefore been developed for object recognition (Cadieu et al., 2007; Fukushima, 2003) proposing a hierarchy of feature extracting simple (S) and complex (C) cells that allow for positional invariance. The combination of S-cells and C-cells, whose signals propagate up the hierarchy allows for scale and position invariant object recognition.

The idea of feature based concept acquisition has been well studied in psychological literature. Sloutsky (2003) discusses how children group concepts based on, not just one, but multiple similarities, which tap the fact that those basic level categories have correlated structures (or features). This correlation of features is also discussed in McClelland and Rogers (2003) who argue that information should be stored at the individual concept level rather than at the super ordinate category level allowing properties to be shared by many items.

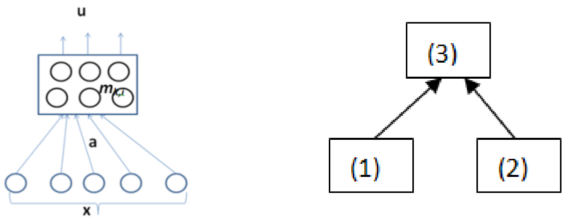
Our model is related to Hubel Weisel approaches in that it implements a hierarchical modular architecture for bottom-up propagation of conceptual information. To our knowledge, however, this is the first implementation of a Hubel Weisel approach to non-natural medium such as text, and has attempted to model hierarchical representation of keywords to form concepts.

System architecture

The system that we describe here is organized in a bottom up hierarchy. This means that the component features are represented before the representation of concept objects. Our learning algorithm exploits the property of this hierarchical structure. Each level in the hierarchy has several modules. These modules model cortical regions of concept memory. The modules are arranged in a tree structure, having several children and one parent. In our paper, we call the bottom most level of the hierarchy level 1, and the level increases as one moves up the hierarchy. The keywords from a document form the inputs to the system. These are directly fed to level 1. Level 1 modules resemble simple cells of the cortex, in that they receive their inputs from a small patch of the input space. Several level 1 modules tile the input space, possibly with overlap. A module at level 2 covers more of the input space when compared to a level 1 module. It represents the union of the input space of all its children level 1 modules. However, a level 2 module obtains its inputs only through its level 1 children. This pattern is repeated in the hierarchy. Thus, the module at the tree root (the top most level) covers the entire input space, but it does so by pooling the inputs from its child modules. In the visual cortex, the level 1 can be considered analogous to the area V1 of the cortex, level 2 to area V2 and so on.

Learning the first batch of information

To understand how the model learns, let us consider the inputs and outputs of a single module $m_{k,i}$ in level k of the system as shown in Figure 1a. Let **x**, representing connections $\{x_j\}$ be the input pattern to the module $m_{k,i}$. **x** is the output of the child modules of $m_{k,i}$ from the level *k-1*, and **a** represent the weights of the competitive network. The vector **a** is used to represent the connections $\{a_j\}$ between **x** and the cells in the module $m_{k,i}$. The output of a neuron in the module $m_{k,i}$ is given by $u = \sum_j a_j x_j$.



(a) (b)

Figure 1a. Inputs and outputs to a single module $m_{k,i}$. b. The concatenation of information from the child modules of the hierarchy to generate inputs for the parent module

During learning, each neuron in $m_{k,i}$ competes with other neurons in the vicinity. Of the large number of inputs to a given module, a neuron is activated by a subset of them. The neuron then becomes the spatial center of these patterns. To ensure that there are no garbage neurons, we adopt in our creation of the module, a model of Growing SOM (GSOM) (Alahakhoon et al., 2000).

When all the modules at level k finish training, the second stage of learning occurs. This comprises the process by which the parent modules learn from the outputs of the child modules. Here, consider the case shown in Figure 1b where the module 3 is the parent of modules 1 and 2. Let **x(1)** be the output vector of module 1 and **x(2)** be the output vector of module 2. **x(i)** represents the Euclidean distance from the input pattern to the each output neuron *i* of the child modules. The input to module 3, $\mathbf{I(3)} = \mathbf{x(1)}||\mathbf{x(2)}$, is the concatenation of the outputs of modules 1 and 2. A particular concatenation represents a simultaneous occurrence of a combination of concepts in the child module. Depending on the statistics of the input data, some combinations will occur more frequently, while others will not. During the second stage of learning, the parent module learns the most frequent combinations of concepts in the levels below it. A GSOM is again used in the clustering of such combinations. The learning process thus defined can be repeated in a hierarchical manner.

Incremental learning

In this and the subsequent sections of the paper, we will use the terms *batch 0* to represent the first batch of documents. *Batch 1* refers to the subsequent set of documents. Once the system learns the documents in *batch i*, only the hierarchical structure and the neuron architecture are retained. All other information regarding the documents presented is discarded.

Incremental learning poses a challenge in Hubel Weisel based computational models due to three reasons. (1) Damage to the knowledge represented by old neurons which is fundamental in competitive learning. (2) Propagation of information in the hierarchical architecture. The number of output neurons of each child node increases with the introduction of the incremental batch. The input dimensions of the parent node are therefore changed and incompatible with the dimensions of the previous batch. (3) The irregularity in the input data dimensions. Where keywords are defined as concepts to be processed by the system, the keywords in an incremental batch will not be a subset of those in the previous batch. The architecture therefore has to provide rules for the generation and growth of new modules with respect to incoming incremental data.

Preventing Damage to Old Memories: This problem is tackled using a sampling method using pseudo data inspired from Liu et al (2008). The algorithm implemented summarizes data distribution in a cluster map. Given neuron **a** in a GSOM of N neurons, consider the closest neuron **b**, **a,b** Є N , their midpoint is given by $\mathbf{a} + \mathbf{b}/_{2}$. We generate a random set of vectors around neuron **a**, bounded on both sides by $\mathbf{a} + \mathbf{b}/_{2}$. These pseudo vectors generated during the training of batch *k* implicitly reconstruct the data used to train batches 0 to *k-1*.

Incremental learning in a hierarchy Let us consider Figure 2, where the modules α and β are child modules of γ.. At batch 0, the training sets $\mathbf{x_\alpha}$ and $\mathbf{x_\beta}$, consisting of p_0 patterns each are used to generate the neurons $\mathbf{y_\alpha}$ and $\mathbf{y_\beta}$.

$$\forall\ i\text{€}p_0,\ \mathbf{x_{\gamma,i}} = \left[\left|\mathbf{x_{\alpha,i}} - \mathbf{y_\alpha}\right| \| \left|\mathbf{x_{\beta,i}} - \mathbf{y_\beta}\right|\right] \qquad (1)$$

is passed to node γ. The vectors $\mathbf{x_\alpha}$, $\mathbf{x_\beta}$ and $\mathbf{x_\gamma}$ are then discarded.

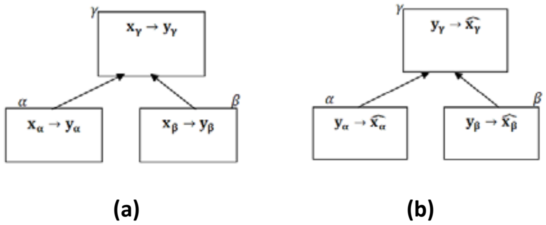

γ
$x_\gamma \rightarrow y_\gamma$
α
$x_\alpha \rightarrow y_\alpha$
$x_\beta \rightarrow y_\beta$
β
(a)
γ
$y_\gamma \rightarrow \widehat{x_\gamma}$
α
$y_\alpha \rightarrow \widehat{x_\alpha}$
$y_\beta \rightarrow \widehat{x_\beta}$
β
(b)

Figure 2. (a) Incremental learning stages. At batch 0, the training patterns at level 1, $\mathbf{x_\alpha}$ and $\mathbf{x_\beta}$ cluster to form the neurons $\mathbf{y_\alpha}$ and $\mathbf{y_\beta}$. For simplicity, we consider that only one neuron is generated after training batch 0. (b) Batch 1 and the approximation of the pseudo vectors $\widehat{\mathbf{x_\alpha}}$, $\widehat{\mathbf{x_\beta}}$ and $\widehat{\mathbf{x_\gamma}}$

When batch 1, consisting of p_1 vectors is now introduced, $\widehat{\mathbf{x_\alpha}}$ and $\widehat{\mathbf{x_\beta}}$ are approximated from $\mathbf{y_\alpha}$ and $\mathbf{y_\beta}$ respectively and used along with the new batch to train the GSOM modules α and β. After training, the neurons of the level 1 nodes y_α and y_β adapt to $y_\alpha{}'$ and $y_\beta{}'$. A set of pseudo data $\widehat{\mathbf{x_\gamma}}$ are approximated from the neuron $\mathbf{y_\gamma}$.

From equation 2, $\widehat{\mathbf{x_\gamma}}$ represents the Euclidean of $\widehat{\mathbf{x_\alpha}}$ and $\widehat{\mathbf{x_\beta}}$ from $\mathbf{y_\alpha}$ and $\mathbf{y_\beta}$ respectively, i.e, for a set of *p₀'* pseudo data,

$$\forall\ i\text{€}p_0{}',\ \widehat{\mathbf{x_{\gamma,\imath}}} = \left[\left|\widehat{\mathbf{x_{\alpha,\imath}}} - \mathbf{y_\alpha}\right| \| \left|\widehat{\mathbf{x_{\beta,\imath}}} - \mathbf{y_\beta}\right|\right] \qquad \mathbf{(2)}$$

However, during the training of batch 1, the measure for $\widehat{\mathbf{x_\gamma}}$ should be the distance to $y_\alpha{}'$ and $y_\beta{}'$, the updated neuron vectors. A set of adapted pseudo vectors $\widehat{\mathbf{x_\gamma}}\mathbf{'}$ should therefore be approximated.

In Euclidean space, we can we can visualize the problem as shown in Figure 3,

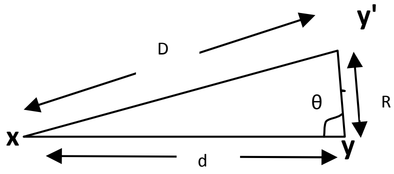


Figure 3. Approximation of incremented pseudo vector for levels 2 and above in the hierarchy

We consider two cases, (a) **y'** is not the winning neuron for the pattern **x**. (b) **y'** is the winning neuron.

Case (a). $\boldsymbol{y'_\alpha}$ *is not the winning neuron, i.e, R<<d*

For ease of analysis, assume that d=1

$$D \cong \frac{1}{\pi}\int_0^\pi \left|\boldsymbol{y}'(\boldsymbol{\theta}) - \boldsymbol{x}\right| d\theta$$

$$D \cong \frac{1}{\pi}\int_0^{\frac{\pi}{2}} \left|\mathbf{y}'(\theta) - \mathbf{x}\right| + \left|\mathbf{y}'(\pi - \theta) - \mathbf{x}\right| d\theta$$

Where $f(\theta) = \left|\mathbf{y}'(\boldsymbol{\theta}) - \mathbf{x}\right| + \left|\mathbf{y}'(\boldsymbol{\pi} - \boldsymbol{\theta}) - \mathbf{x}\right|$,

$$D \cong \tfrac{1}{\pi}\textstyle\int_0^{\frac{\pi}{2}} f(\theta) \qquad (3)$$

$$f(\theta) = \sqrt{(1 - cos\theta R)^2 + (sin\theta R)^2} \\ + \sqrt{(1 + cos\theta R)^2 + (sin\theta R)^2}$$

Let $a = \left|\boldsymbol{y}'(\boldsymbol{\theta}) - \boldsymbol{x}\right|$ and $b = \left|\boldsymbol{y}'(\boldsymbol{\pi} - \boldsymbol{\theta}) - \boldsymbol{x}\right|$

We obtain

$$f(\theta) = \sqrt{2}\sqrt{1 + R^2 + ab} \qquad (4)$$

Where $E_0 = ab$,,

$E_0 =$
$\sqrt{(1 - cos\theta R)^2 + (sin\theta R)^4 + \sin^2(\theta R)[2 + 2cos^2\theta]}$
(11)

$$\cong 1 - \frac{cos^2(\theta R)}{2} + \frac{3\sin^2(\theta R)}{2} \qquad (5)$$

Substituting (5) into (4), we obtain

$$f(\theta) = 2\left(1 - \tfrac{R^2}{8} + \tfrac{1}{2}sin^2(\theta R)\right) \qquad (6)$$

Substituting (6) into (3), we obtain

$$D = = 1 + \tfrac{R^2}{8} + \tfrac{R^2}{\pi}\textstyle\int_0^{\frac{\pi}{2}} sin^2\,\theta d\theta \qquad (7)$$

Solving (7), we have

$$D = 1 + \frac{3R^2}{8} \qquad (8)$$

Based on Figure 3, if we approximate a $\theta = {}^{\pi}/_{2}$, we obtain $D = 1 + \frac{R^2}{2}$. In implementation, to satisfy (8) we use the inequality (9) to assign the value of θ.

$$\frac{4\pi}{9} < \theta < {}^{\pi}/_{2} \qquad (9)$$

We can therefore conclude that, a θ value specified by inequality (9) can be used to re-generate the dataset $\widehat{\mathbf{x_{\gamma,\imath}}}[0,1,\ldots,j,\ldots k_\alpha]$, where $\widehat{\mathbf{x_{\gamma,\imath}}}[\boldsymbol{j}] \equiv \left|\widehat{\mathbf{x_{\alpha,\imath}}} - \mathbf{y'_\alpha}\right|$ and $\mathbf{y'_\alpha}$ is not the winning neuron.

Case (b): $\boldsymbol{y'_\alpha}$ *is the winning neuron*

If y' is the winning neuron, a random value of θ, $0 < \theta < \pi$ can be used to regenerate $\widehat{\mathbf{x_{\gamma,\imath}}}[j] \equiv \left|\widehat{\mathbf{x_{\alpha,\imath}}} - \mathbf{y'_\alpha}\right|$, where y'_α is the winning neuron.

Dealing with the problem of new input dimensions: A rule based approach of creating a new module to process the new keywords is preliminarily proposed to deal with the dynamically increasing input dimensions. A module is trained and connected to parents only if the number of concepts that it represents increases above a predefined threshold. In order to avoid overcrowding, heuristic rules have been put into place such that a parent has atmost three children.

Experimental results

To illustrate the cognitive properties of the training model, we train the system using 21 concepts from 200 documents. The concepts included ideas such as "birds", "animals", "flowers", "trees" etc, same as the ones used by McClelland and Rogers (2003). The following preprocessing was performed to the documents. First, the contents were analyzed and the stopwords removed. The concept terms were stemmed and grouped using Wordnet (Fellbaum, 1998) before a tf.idf weighing scheme was used to select the most relevant concepts to the batch. For visualization purposes,

Hierarchical identification of concepts from wiki documents

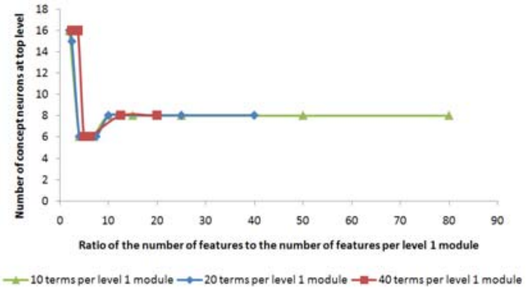


Figure 4. The number of concept neurons at the top of the hierarchy vs. the number of features per bottom level module

In this section we observe how our hierarchical model captures the properties of semantic cognition outlined by McClelland and Rogers (2003, 2008). The training data used by McClelland and Rogers is intuitively designed based on common sense knowledge. Our system, on the other hand is trained using information from 200 text descriptors of the concepts from wikipedia. The snippets varied in length from 50 word descriptions to 500 word descriptions. Figure 4 illustrates the number of concept neurons at the top level as a function of the ratio of the number of features to each level 1 module and the total number of features. When there are only two layers in the hierarchy, a larger number of concept neurons (16) are generated. The number of concept neurons converges to between 6 and 8 for all other architectures. Typically, for a six concept cluster, the concept of penguin is separate from that of other clusters. This is shown in Figure 6.

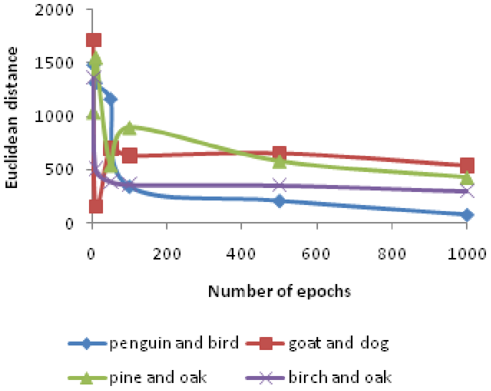


Figure 5. Euclidean distance between various concepts vs. the number of training epochs

In Figure 5, we observe the evolution of the Euclidean distance between concepts. The training shows empirical properties of convergence. The distances between the various concepts are stable after 500 epochs of training. We can also observe promising results from the concept representation point of view. For instance, the Euclidean distance between "pine" and "oak", for instance, is larger than the Euclidean distance between "birch" and "oak", which belong to the same family.

Figure 6.a shows the top two levels of a five level hierarchy of hierarchy of concepts obtained (10 concepts per GSOM module and 160 concepts used in training). We observe, as is the result in McClelland and Rogers that similar concepts tend to be near each other in space. For instance, "canary" and "sparrow" tend to be closer to each other, but far from "penguin". In some cases, super ordinate terms, such as "bird", "tree" etc are mined as part of the hierarchy. There are some interesting observations that can be made here. We can see that the highest level (level 5) shows general concepts while level 4 shows the concepts one level lower. i.e., while the neuron 1 refers to "animals", the neuron box "2" refers to more detailed differentiation of neuron 1. Further to this, the system also shows some intermediate level categorization characteristics that taps item frequency effects. In McClelland and Rogers' paper, they describe it as the process by which certain descriptive terms such as "tree", "bird" and "dog" tend to be acquired earlier than the super ordinate terms such as "plant" or "animal" or more specific terms such as canary, pine or poodle. The general consensus for this is that parents use certain intermediate level words more frequently when speaking to children. As such, intermediate concepts, based on their frequency of usage, are also clustered more tightly into intermediate groups within super ordinate concepts.

In the experimental data, some concepts were used more frequently than others of the same sub category. These include birch (12 instances) vs. 10 instances of pine and maple, 16 instances of rose vs. 3 instances of daisy and 7 instances of sunflower. It is seem that the more frequent concepts are tied together with the super ordinate concept neuron (dog is tied with animal, rose with flower, sparrow with bird) and so on.

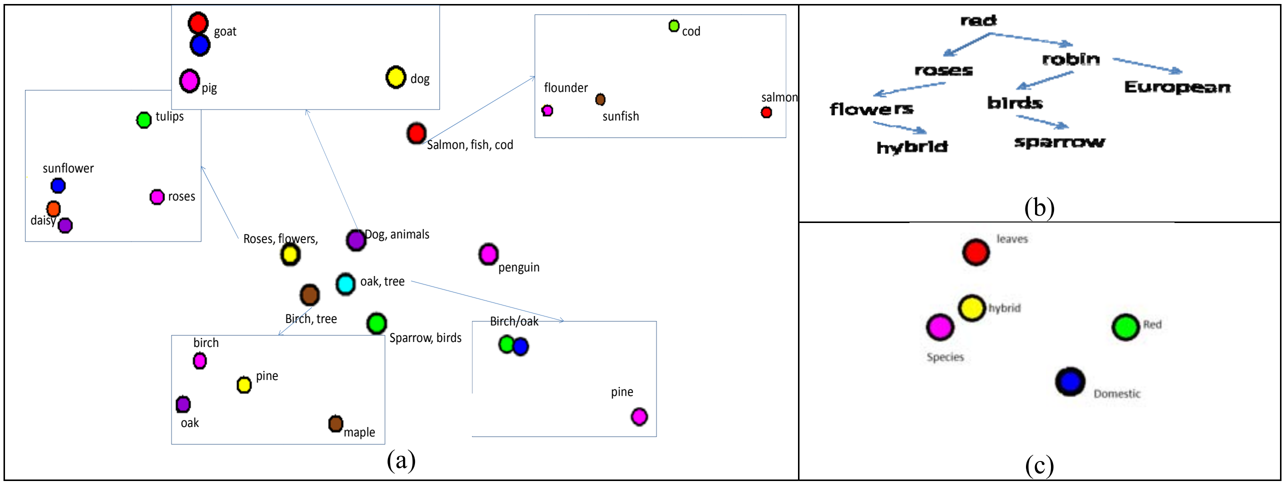


Figure 6 (a) Hierarchical representation of the 21 concepts in the memory system. (b) Grouping due to secondary characteristics – “leaves” groups all trees, “red” for rose, salmon etc. (c) Associative retrieval of concepts based on activation of high correlation neurons in the hierarchy.

At the lower level, we can also observe alternate similarity grouping which are based on individual features. For instance, at level 1 module, one might observe neurons denoting concepts such as “leaves” grouping birch, maple and oak. The concept “grow” groups objects of all categories. “Red” indicates objects that are red – rose, cod, salmon. One such lower layer representation is shown in Figure 6.c.

From these results, one can expect this model to perform a sort of associative retrieval of the kind that humans perform. For instance, if one sees the concept “gold”, one would think of perhaps “ring” and subsequently “marriage” and then “family”, “children” and so forth. In the same way, firing of the neuron “red” at level 1 will lead to firing of corresponding nodes at various levels. This will lead to other concepts being fired, both sequentially and parallely. For instance, probing the model with “red” leads to the sequential firing as described in Figure 6.b.

Each concept can, in turn, be used as a probe to activate relevant neurons. In this sense, the model describes the human chain of thought process. Work is in progress to study how this process can be modeled and how the firing can be controlled by the wiring strength between modules.

Incremental learning performance

Overview of incremental learning with 5 concepts and 3 batches: In this section, instead of introducing all the concepts are one go, concepts are learnt in batches. The figure below shows the evolution of the incremental concept network at the top level for the first three batches (a) tulips and sunflowers only (b) sparrow and sunfish (c) salmon. In batch 1, the top level generalizes, creating concept representation of flower and animal. Lower layers now portray the differentiated concepts. In batch 2, the concept of salmon was introduced. At this juncture, the old information from sunfish is sub grouped under fish and sparrow is considered a separate entity. These experiments suggest merit in the approximations that we have described earlier in the paper.

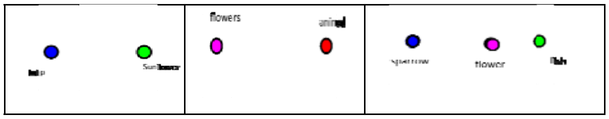


Figure 7. The top-level evolution of incremental concept representation. (a) Batch 0 tulip and sunflower. (b) Batch 1: sparrow and sunfish. (c)Batch 2: salmon

Hierarchical representation in incremental learning

In this experiment, concepts were introduced one by one, beginning with “birch” and “cod”. At some juncture concepts are reintroduced to investigate the effect of new data and rehearsal on old data. Some of the results obtained are discussed in this section.

Figure 8a shows the evolution of how the concepts “birch” and “betula” are represented (“Betula” is the scientific name for “birch”). Birch is the first concept that was introduced to the system. At batch 0, the distinction between the concepts “birch” and “betula” appears in level 3 of the hierarchy. To the system, “betula” and “birch” are two distinct concepts, though with a low Euclidean distance of 39.81. When batch 2 is introduced, the distinction between the two concepts moves one level lower to level 2 and eventually to level 1. However, as more concepts are introduced to the system, the presence of the new information makes the system lose the distinction between the concepts “birch” and “betula”, and the Euclidean distance between the concepts reduces to 0 at batch 7. However, at batch 10, when the concept of birch is reintroduced, the Euclidean distance between the two terms increases before gradually decreasing to 0 once again. A similar result is also observed in the relationship between terms “canary” and “islands”.

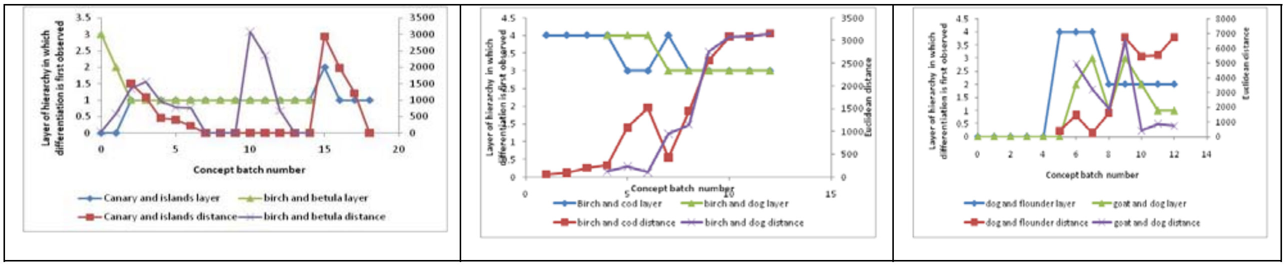


Figure 8. Evolution of the hierarchical and Euclidean relationship between the concepts (a)“birch” and “betula” vs “canary” and “islands” (b) “Birch” and “cod” vs. “birch” and “dog” (c) “dog” and “flounder” vs. “goat” and “dog”

Figure 8b shows the representation of the concept “birch” with respect to the concepts “dog” and “cod”. “Birch” is introduced to the system at batch 0, “cod” at batch 1 and “dog” at batch 4. The differentiation between the concepts “birch” and “cod” is at level 4 and converges to level 3. By batch 8, the concepts of “dog” and “cod” are of the same distance from “birch”. At this juncture, the system at level 3 no longer distinguishes between “birch”, “cod” and “dog”, but makes a distinction between “plant” and “animal”. Figure 8c shows a similar relationship of concepts “dog” with the concepts “flounder” and “goat”. The flounder-dog distinction converges to level 2 (from Figure 8b, we can see that the plant-animal distinction occurred at level 3) while the dog-goat distinction converges to level 1. The Euclidean distance between the concept terms “dog” and “goat” converges to approximately 700 which is close to the value that is obtained through batch learning (from Figure 5).

Conclusions and further work

In summary, our model attempts to propose a hierarchical Hubel Weisel model for the acquisition of concepts from text such that the concepts are represented in a hierarchical connectionist network. The result is a new framework that we have applied in two scenarios. The first is concept acquisition where we have shown that the system is able to represent everyday concepts in a hierarchical fashion, in a manner similar to the PDP model. The system was interestingly also able to perform chain retrieval, in that when “red” was given as a probe to the system, it was able to retrieve “robin” and by association “sparrow”. Secondly, we have modeled information approximation and incremental learning, which models some properties of short term memory.

There are several directions for further work in this area. In addition to the pertinent issues of improving computation time and processing algorithms to make the system able to handle large sets of data, one important direction is the incorporation of semantic information into the hierarchical architecture. As of now, this information is ignored and only the statistical properties of keywords are taken into consideration in the generation of the concept hierarchy. Work is under process to integrate semantic information into the model. Work is also under progress to include common sense knowledge in the model. We expect that these additions will make the model more cognitively accurate. In addition to this, we are also incorporating other aspects of cognition such as attention; interest etc to study the generation and behavior of the cognitive map.

References

Cadieu, C., Kouh, M., Pasupathy, A., Conner, C., Riesenhuber, M., & Poggio, T.A. (2007). *A Model of V4 Shape Selectivity and Invariance*. J ournal of Neurophysiology 98, 1733-1750

George D, Hawkins J (2005), *A hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex*, International Joint Conference on Neural Networks, 3, 1812-1817

Vernon Mountcastle (1978), *An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System*, The Mindful Brain (Gerald M. Edelman and Vernon B. Mountcastle, eds.) Cambridge, MA: MIT Press.

Hubel D and Weisel T (1965), *Receptive fields and functional architecture in two non striate visual areas (18 and 19) of a cat*, Journal of NeuroPhysiology 28, pp229-289

Fukushima K(2003), Neocognitron for handwritten digit recognition 1, Neurocomputing , 51C, 161-180

Alahakhoon D, Halgamuge S K, Srinivasan B (2000), *Dynamic Self Organizing maps with controlled growth for Knowledge discovery*, IEEE Transactions on neural networks, 11(3), pp601-614

Liu M, Liu Y C, Wang X L (2008), IGSOM: *Incremental clustering based on self organizing map*, International Conference on Intelligent Information hiding and multimedia Signal Processing, IIHMSP’08, pp 885-890

Fellbaum C (1998), *Wordnet: An electronic lexical database*, MIT Press

Mc Clelland J L and Rogers T T (2003), *The parallel distributed processsing approach to semantic cognition*, Nature Reviews Neuroscience, 4(4), pp310-322

Sloutsky VM (2003), *The role of similarity in the development of categorization*, Trends in Cognitie Sciences, 7, 246-251

Rogers T T, McClelland J L (2008), *Precis of Semantic Cognition, a Parallel distributed Processing approach*, Brain and Behavioral Sciences, 31, pp 689-749