

Formal Analysis of Models for the Mammalian Vision System

Bedeho Mender¹

Abstract. In this paper we present a rigorous definition of classification in a common family of models for the mammalian vision system, and provide a formal analysis of classification power in this family. We characterize the power of a single unit, and subsequently demonstrate how to reduce a network of arbitrary topology and size to such a single unit. We finally show how such a network, via a reduction to a single unit, is equivalent to a simple 3 layer feed forward network of threshold units.

1 INTRODUCTION

The goal of this paper is to demonstrate the usefulness of formal analysis of cortex models and biologically inspired vision systems in particular. If such models aspire to be not only devices of utility but also accurate descriptions of cortical computation, then foundational insight into their computational mechanisms is a necessary step towards a full understanding of the cortex.

With our maturing understanding of cortical computation in mammals, particularly in the visual cortex, there have been developed a number of biologically inspired vision systems, e.g. K. Fukushima [3], M. Riesenhuber and T. Poggio [2], G. Wallis and E. T. Rolls [1] and D. George and J. Hawkins [4]. We consider the model of D. George and J. Hawkins, called Hierarchical Temporal Memory (HTM), first outlined in the 2004 book “On Intelligence” by J. Hawkins as a biological model of cortex. This model is particularly interesting to study, not only because of the significant empirical performance it has demonstrated [4] or because it subsumes similar models [5], but also because it is being developed actively and enjoys a growing community of developers using a general purpose computing platform based on it².

2 HIERARCHICAL TEMPORAL MEMORY

In this paper we consider fully trained HTM networks ready for classification tasks, for more information on how learning, prediction and other features work, please consult [5, 4, 6].

A HTM network is a tree of nodes where the input is fed into the leaf nodes and the result is outputted from the top node. The leaf nodes are given non-overlapping sections of the total network input, and the internal nodes are given input from their children.

A leaf node has a finite list of vectors from its input space, called spatial patterns, and these are partitioned into groups called temporal

groups.³ Given some input $\mathbf{x}$, the output of the node is a vector with a score for each of its temporal groups, where the score of a group is the maximum score among all spatial patterns in that group. The score of each individual spatial pattern is a simple matching score between the pattern and the input $\mathbf{x}$.

An internal node also has a finite list of vectors called coincidence vectors. These are as long as the number of children of the node, and each position in a coincidence vector references a temporal group in the corresponding child node. So for example if a node has a coincidence vector (2, 4, 6), then the node has three children and this specific coincidence is referencing the second, fourth and sixth temporal group in the children respectively. As in the leaf nodes, these coincidence vectors are partitioned into temporal groups and the score of each group is the maximum score among all coincidence vectors in that group. The score of each coincidence is calculated by taking the product of the scores of the temporal groups in the children referenced by the coincidence.

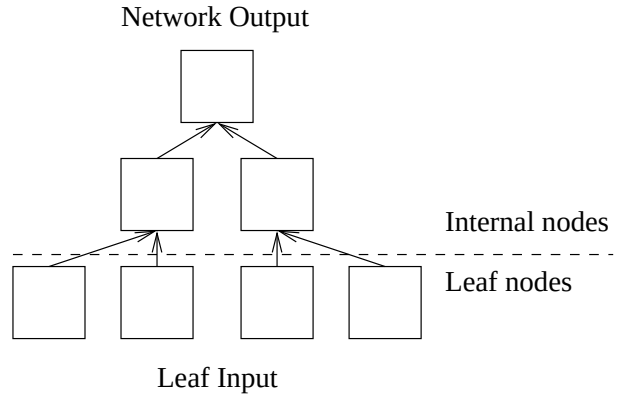


Figure 1. A simple three layer HTM network.

Before summarizing this into precise definitions we establish convention for rest of the paper. We consistently use boldface style vectors and regular style scalars, so let $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ and $\mathbf{y} = (y_1, \dots, y_m) \in \mathbb{R}^m$. Vector concatenation of $\mathbf{y}$ onto $\mathbf{x}$ is denoted $\mathbf{x} \oplus \mathbf{y}$, that is $\mathbf{x} \oplus \mathbf{y} = (x_1, \dots, x_n, y_1, \dots, y_m)$. We may project onto a contiguous section, or just one position, of a vector by writing $\mathbf{x}[i, j] = (x_i, \dots, x_j)$ or $\mathbf{x}[i] = x_i$ respectively. We use the standard

¹ University of Oslo, Norway, email: bmmender@ifi.uio.no

² Numenta Inc. (<http://www.numenta.com>) is a private company, founded by D. Dubinsky, D. George and J. Hawkins, developing a general purpose computing platform, called NuPiC, based on the HTM model.

³ These names (spatial pattern, temporal group, coincidence) are due to how this information is learned during training, and while this is not relevant in this presentation, we choose to use the original names for the sake of consistency.

norm, that is $\|\mathbf{x}\|^2 = \sum_{i=1}^n (\mathbf{x}[i])^2$. Finally, a cover for some set $\mathcal{T}$ is a family of sets $\{\mathcal{T}_1, \dots, \mathcal{T}_p\}$ such that $\mathcal{T} = \cup_i \mathcal{T}_i$, and this cover is called a partitioning if $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset$ for $i \neq j$.

Definition 1 (Leaf Node). A leaf node is defined as $\mathcal{G} = (l, r, \sigma, \mathcal{S}, \mathcal{T})$ where; $1 \leq l \leq r$ and (l, r) is called the receptive field; $\sigma \in \mathbb{R}$ is called the spread; $\mathcal{S} \subset \mathbb{R}^{r-l+1}$ is finite and called the spatial set; $\mathcal{T} = \{\mathcal{T}_1, \dots, \mathcal{T}_p\}$ is a partitioning of $\mathcal{S}$, and the partitions are called temporal groups. We may indicate the receptive field of a node by writing $\mathcal{G}_{l,r}$. For any $\mathbf{x} \in \mathbb{R}^d$ where $d \geq r$ we define

$$\mathcal{G}(\mathbf{x}) = (z_1(\mathbf{x}[l, r]), \dots, z_p(\mathbf{x}[l, r])),$$

where

$$z_i(\mathbf{u}) = \max_{\mathbf{c} \in \mathcal{T}_i} \left\{ e^{-\frac{\|\mathbf{u}-\mathbf{c}\|^2}{2\sigma^2}} \right\}.$$

Definition 2 (Internal Node). A network with receptive field (l, r) is inductively defined to be either a leaf node $\mathcal{G}_{l,r}$, or as $\mathcal{H} = (\mathcal{H}_{r_1, r_2}^1, \dots, \mathcal{H}_{r_{2n-1}, r_{2n}}^n, \mathcal{C}, \mathcal{T})$ where; $\mathcal{H}_{r_1, r_2}^1, \dots, \mathcal{H}_{r_{2n-1}, r_{2n}}^n$ are networks such that $l = r_1 < \dots < r_{2n} = r$ and $r_3 - r_2 = \dots = r_{2n} - r_{2n-1} = 1$; $\mathcal{C} \subseteq \{(c_1, \dots, c_n) \mid 1 \leq c_i \leq d_i\}$ is called the coincidence set and d_i is the number of temporal groups in $\mathcal{H}^i$; $\mathcal{T} = \{\mathcal{T}_1, \dots, \mathcal{T}_p\}$ is a partitioning of $\mathcal{C}$, and the partitions are called temporal groups. For any $\mathbf{x} \in \mathbb{R}^d$ where $d \geq r$ we define

$$\mathcal{H}(\mathbf{x}) = (z_1(\mathbf{x}), \dots, z_p(\mathbf{x})),$$

where

$$z_i(\mathbf{x}) = \max_{\mathbf{c} \in \mathcal{T}_i} \left\{ \prod_{j=1}^n \mathcal{H}^j(\mathbf{x})[\mathbf{c}[j]] \right\}.$$

3 CLASSIFICATION

It turns out that convex polytypes⁴ are essential for characterizing leaf node classification, and a convenient way of representing a convex polytype is by keeping the coefficients of the bounding hyperplanes in matrix form.

Definition 3. A cover $\{\mathcal{T}_1, \dots, \mathcal{T}_p\}$ for $\mathbb{R}^n$ is said to be a finite union of convex polytypes when for each $i = 1, \dots, p$ there exist $m \times n$ matrices $A_1^i, \dots, A_{k_i}^i$ and vectors $\mathbf{b}_1^i, \dots, \mathbf{b}_{k_i}^i \in \mathbb{R}^n$ such that

$$\mathbf{x} \in \mathcal{T}_i \Leftrightarrow A_j^i \mathbf{x} \leq \mathbf{b}_j^i \text{ for some } j = 1, \dots, k_i.$$

In order to state the key result about leaf node classification, we also need to consider how a network assigns category labels to different points in the input space.

Definition 4. For any $n, m > 0$, a function $f : \mathbb{R}^n \rightarrow \mathcal{P}(\{\ell_1, \dots, \ell_m\})$ is called a classification in dimension n of size m when for each ℓ_i there exists an $\mathbf{x} \in \mathbb{R}^n$ such that $\ell_i \in f(\mathbf{x})$. Given classification f , we define $\mathbb{R}_{\ell_i}^n = \{\mathbf{x} \in \mathbb{R}^n \mid \ell_i \in f(\mathbf{x})\}$, and $\mathbb{R}_f^n = \{\mathbb{R}_{\ell_1}^n, \dots, \mathbb{R}_{\ell_m}^n\}$, which is a cover for $\mathbb{R}^n$.

For any network $\mathcal{H}_{l,r}$ with n temporal groups we define the classification $f_{\mathcal{H}} : \mathbb{R}^{r-l+1} \rightarrow \mathcal{P}(\{\ell_1, \dots, \ell_n\})$ by

$$f_{\mathcal{H}}(\mathbf{x}) = \left\{ \ell_i \mid i \in \operatorname{argmax}_{1 \leq j \leq n} \{\mathcal{H}(\mathbf{x})[j]\} \right\}.$$

⁴ A polytype is a n -dimensional generalization of the two dimensional polygon, and a convex polytype is one which defines a convex set. They are often defined as intersections of a finite number of half-spaces, which again are characterized by hyperplanes.

Theorem 5. $\mathbb{R}_{f_{\mathcal{G}}}^n$ is a finite unions of convex polytypes for any leaf node $\mathcal{G}_{l,n}$.

The second key result establishes how a network may be reduced to an equivalent singel leaf node.

Theorem 6. For any network $\mathcal{H}_{l,r}$ where all leaf nodes have the same spread σ , there exists a leaf node $\mathcal{G}_{\mathcal{H}}$ with spread σ and receptive field (l, r) such that for any $\mathbf{x} \in \mathbb{R}^d$ where $d \geq r - l + 1$ we have

$$\mathcal{H}(\mathbf{x}) = \mathcal{G}_{\mathcal{H}}(\mathbf{x}).$$

Finally we show that a finite union of convex polytypes cover may be captured quite naturally using a 3 layer feed-forward network of linear threshold units. This means that since a full network may be reduced to a singel unit as mentioned in Theorem 6, and by Theorem 5 such a unit naturally induces a finite union of convex polytypes cover on its input space, by Theorem 7 this singel unit has an equivalent feed-forward network classifying it.

Theorem 7. For any finite union of convex polytypes cover $\{\mathcal{T}_1, \dots, \mathcal{T}_p\}$ for $\mathbb{R}^n$ there exists 3 layer feed-forward neural network classifying it.

4 DISCUSSION

The flavor of networks considered here by no means represent the full range of networks that one can have in the HTM model. The model has matured significantly, although the fundamental concept of hierarchical temporal and spatial pooling remains the same. Requiring all leaf nodes to share the same spread is not such a severe restriction as it may seem because

- If all leaf nodes receive input from the same domain (e.g. bottom nodes in a vision network) we expect that they should learn and be exposed to the same low level spatial patterns (e.g. horizontal, vertical, diagonal lines and shapes) and thus have the same spread.
- The spread is a hand tuned parameter which cannot be learned, so it is in practice often left the same across all nodes.
- A common technique in training such networks is what is called node cloning, which means that you only train a single node pr. network level, and then just replicate that node when you do classification.

REFERENCES

- [1] G. Wallis and E. T. Rolls. A model of invariant object recognition in the visual system. Prog. Neurobiol., 51:167-194, 1997.
- [2] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. Nat. Neurosci., 2:1019-1025, 1999.
- [3] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cyb., 36:193-202, 1980.
- [4] George D, Hawkins J, 2009 Towards a Mathematical Theory of Cortical Micro-circuits. PLoS Comput Biol 5(10): e1000532. doi:10.1371/journal.pcbi.1000532.
- [5] George D. HOW THE BRAIN MIGHT WORK: A hierarchical and temporal model for learning and recognition. PhD Thesis Stanford, June 2008: <http://numenta.com/for-developers/education/DileepThesis.pdf>
- [6] Numenta Inc. Zeta1 Algorithms Reference Version 1.5. Available upon request.