

Portland State University

PDXScholar

Systems Science Friday Noon Seminar Series

Systems Science

2-26-2010

Biologically Inspired Computing: The DARPA SyNAPSE Program & The Hierarchical Temporal Memory

Dan Hammerstrom

Portland State University, dwh@pdx.edu

Follow this and additional works at: https://pdxscholar.library.pdx.edu/systems_science_seminar_series



Part of the [Computer Sciences Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Hammerstrom, Dan, "Biologically Inspired Computing: The DARPA SyNAPSE Program & The Hierarchical Temporal Memory" (2010). *Systems Science Friday Noon Seminar Series*. 36.

https://pdxscholar.library.pdx.edu/systems_science_seminar_series/36

This Book is brought to you for free and open access. It has been accepted for inclusion in Systems Science Friday Noon Seminar Series by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

Biologically Inspired Computing:

- The DARPA SyNAPSE Program &
- The Hierarchical Temporal Memory

Dan Hammerstrom
Electrical And Computer Engineering
Portland State University

2/26/10

1

Maseeh College of Engineering
and Computer Science

Intelligent Computing

- There is one class of problems that we still do not solve well
- These problems involve the interaction of a computing system with the real world
- Which, in part, involves a transformation and understanding of data at the boundary between the real world and the digital world
- These problems occur wherever a computer is interacting with the real world – which includes almost every embedded application
- An interesting opportunity for specialized hardware

2/26/10

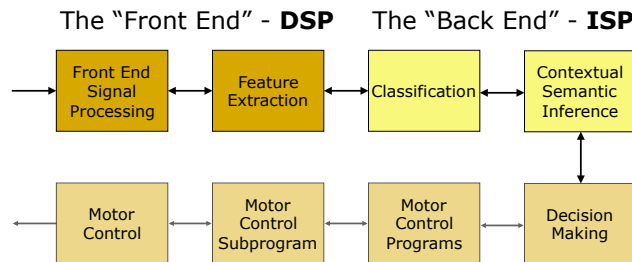
Hammerstrom

2

Maseeh College of Engineering
and Computer Science

Our Focus: Intelligent Signal Processing (ISP)

- ISP augments and enhances traditional DSP (Digital Signal Processing) by incorporating contextual and higher level knowledge of the application domain into the data transformation process



The "Front End"

- Front end processing is well understood, it is the realm of traditional digital signal and image processing
- Front end algorithms generally apply the same computation over large arrays of elements, they are data parallel, and communication tends to be local
 - An excellent example of such an architecture is the CNN (Cellular Non-linear Network) developed by Chua, Roska et al.
 - Most "neuromorphic" VLSI operates at the front end

But Then There's The "Back-End" ...

- In the early days of computing, "Artificial Intelligence" focused on the representation and use of contextual and semantic information
- Knowledge was generally represented by a set of rules
- However, these systems were "brittle," exhibiting limited flexibility, generalization, and graceful degradation
- They did not scale
- And they were unable to adapt dynamically (i.e., learn) within the context of most real world applications

The ISP Toolbox –
Still mostly empty after
all these years ...

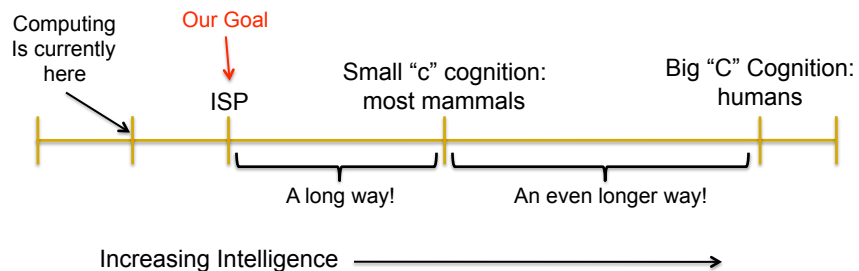


Some Desirable ISP Characteristics

- Solving the problem
- Massively parallel and low precision
- Self-organizing – in fact, system design becomes more the provisioning of organizing principles (Prof. Christoph von der Malsburg), than the specification of all operational aspects of the models
 - www.organic-computing.org
- Generalization, and graceful degradation
- Low power - the processing power of the brain is roughly 10^{15} operations per second which it accomplishes at a power dissipation of about 25 watts
- Scales - The scaling limitations of both symbolic and traditional neural network approaches constitute one of their biggest shortcomings
- Adaptive - Consequently another important characteristic of real systems is incremental, integrative adaptation or learning during system operation

The Scope of Our Project

- Conceptually one can think of “computational intelligence” as a spectrum
- And though not universally accepted, it has been hypothesized that this spectrum is more or less continuous from one end to the other

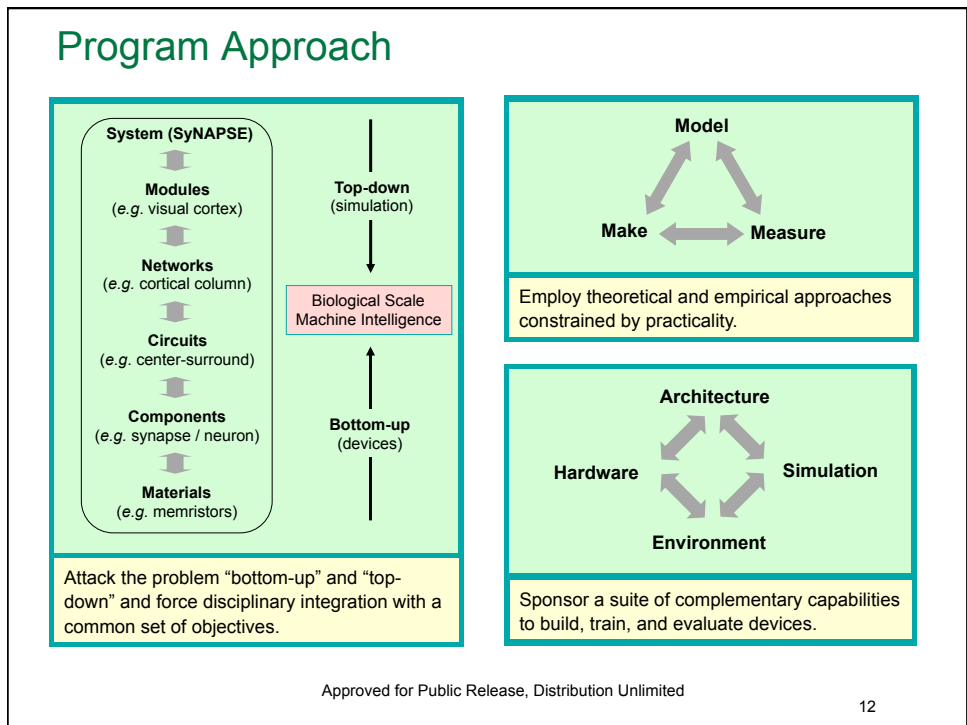
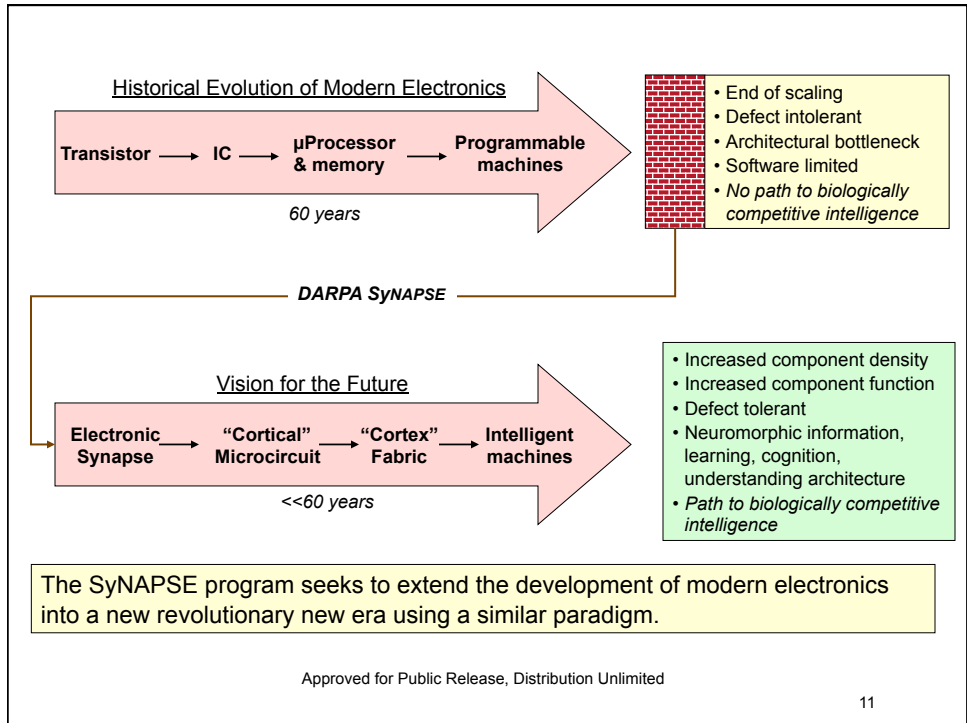


- There is increasing interest in using models from Computational Neuroscience, in particular cortical models as inspiration for new models of computation in general and intelligent computing in particular
- In Europe there is FACETS
 - The goal of the FACETS (**F**ast **A**nalog **C**omputing with **E**mergent **T**ransient **S**tates) project is to create a theoretical and experimental foundation for the realisation of novel computing paradigms which exploit the concepts experimentally observed in biological nervous systems.
- And in the US there is the DARPA SyNAPSE Program

Systems of Neuromorphic Adaptive Plastic Scalable Electronics



Dr. Todd Hylton, Program Manager
DARPA DSO




	Phase 0	Phase 1	Phase 2	Phase 3	Phase 4
Hardware	Component synapse (and neuron) development	CMOS process and core circuit development	CMOS process integration	~10 ⁶ neuron single chip implementation "Mouse" level	~10 ⁸ neuron multi-chip robot at "Cat" level
Architecture & Tools	Microcircuit architecture development	System level architecture development	~10 ⁶ neuron design for simulation and hardware layout	~10 ⁸ neuron design for simulation and hardware layout	Comprehensive design capability
Emulation & Simulation	Preparatory studies only	Simulate large neural subsystem dynamics	"Mouse" level benchmark (~ 10 ⁶ neuron)	"Cat" level benchmark (~ 10 ⁸ neuron)	
Environment	Preparatory studies only	Build Sensory, Planning and Navigation environments "Small mammal" complexity	Add Audition, Proprioception and Survival "All mammal" complexity	Add Touch and Symbolic environments	Sustain

*Currently only Phases 0 and 1 have been funded
We are now In Phase 1*

Approved for Public Release, Distribution Unlimited

13



- There are three contractors:
 - IBM - Dharmendra Modha
 - HP – Greg Snider
 - HRL (formerly the Hughes Research Lab) – Narayan Srinivasa

- The ultimate goal is to build a low-power, compact electronic chip combining a novel analog circuit design and a neuroscience-inspired architecture that can address a wide range of cognitive abilities—perception, planning, decision making and motor control

- "Our research progress in this area is unprecedented," says DARPA program manager Todd Hylton, Ph.D. "No suitable electronic synaptic device that can perform critical functions of a biological brain like spike-timing-dependent plasticity [an indicator of the capability to learn] has ever before been demonstrated or even articulated."

2/26/10

Hammerstrom

14

Maseeh College of Engineering
and Computer Science

HRL Team

- Hardware:
 - Analog circuits, HRL
 - Nano-devices, Wei Lu, Univ. of Michigan
 - Floating gate devices, Paul Hasler, Georgia Tech
 - Systems integration and global communication, Dan Hammerstrom, Portland State
- Neuroscience
 - Steve Grossberg, Boston University
 - Eugene Izhikevich, Jason Fleischer, et al., Neurosciences Institute
 - Jeff Krichmar, UC Irvine
 - Phil Goodman, University of Nevada, Reno
 - Giorgio Ascoli and Alexei Samsonovich, George Mason
- All three teams have completed phase 0 and are now in the middle of phase 1, which is scheduled to end in December 2010

2/26/10

Hammerstrom

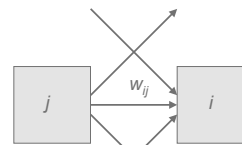
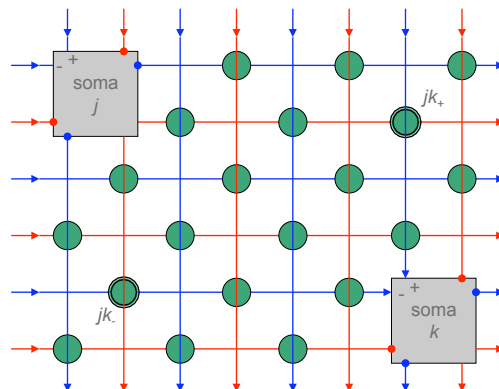
15

Maseeh College of Engineering and Computer Science

NEUROMORPHIC NETWORKS ("CROSSNETS")

Basic idea:

CMOS "somas" + nanowire "axons" and "dendrites" + nanodevice "synapses"



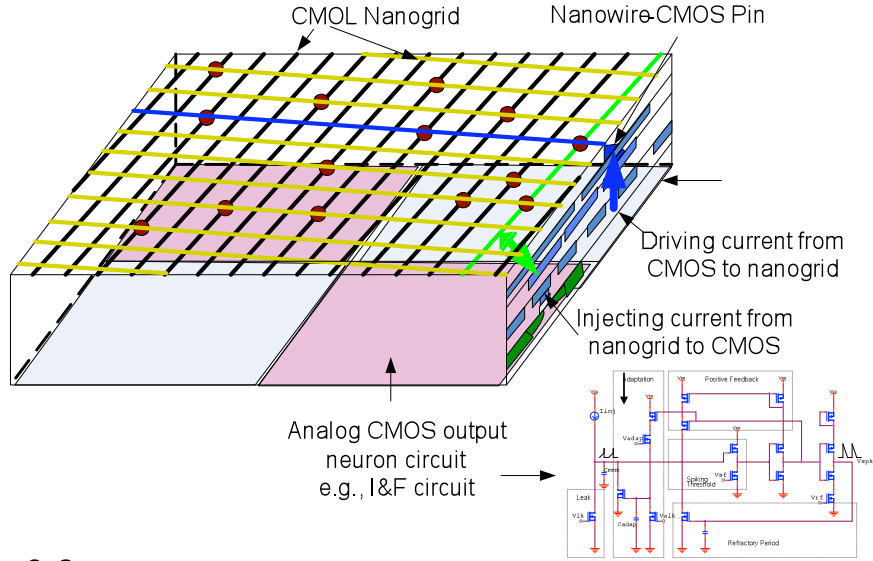
$$x_i = \sum_j w_{ij} y_j$$

$$w_{jk} = \{-1, 0, +1\}$$

Generic structure of a feedforward CrossNet

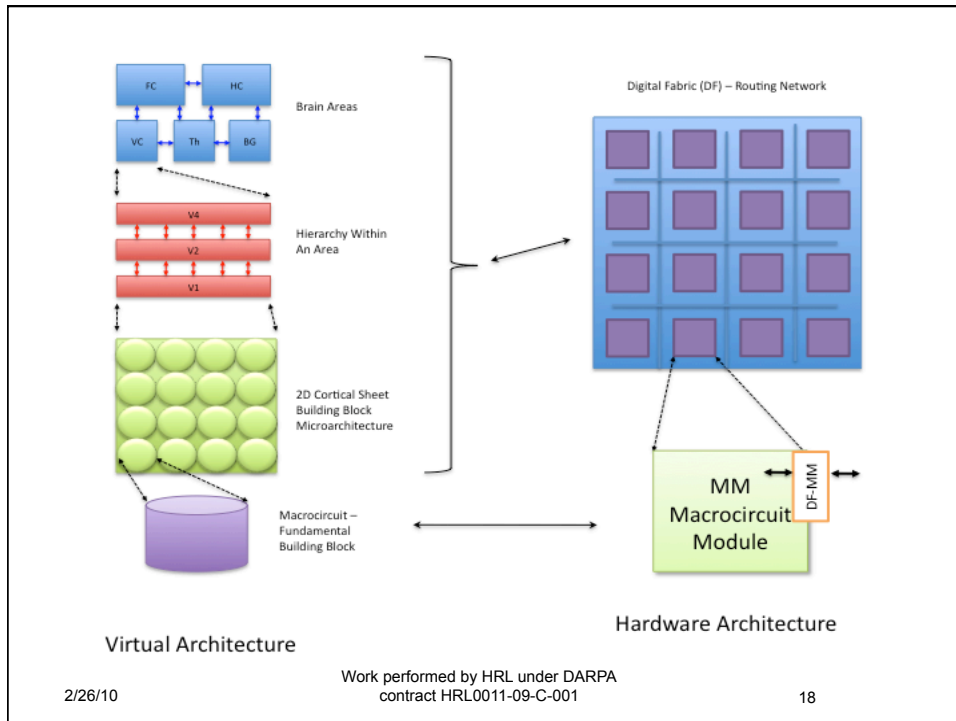
S. Fölling *et al.* (2001)
O. Turel *et al.* (2004)

Structural View of Mixed-Signal CMOL Design (Each CP) - Gao



C. Gao

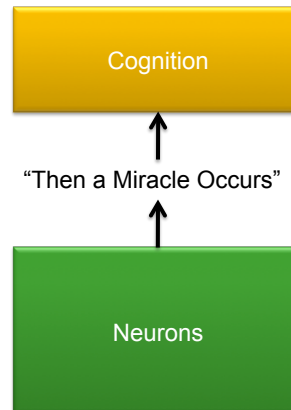
17



However ...

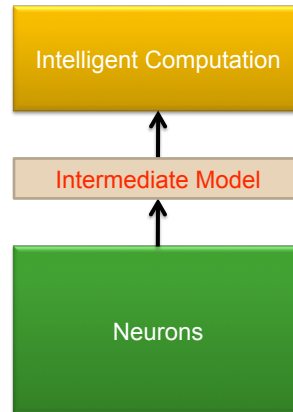
The "Gap"

- So do neural techniques lead to advanced ISP?
- Most Computational Neuroscience is weak in making the jump from spiking neurons with learning rules such as STDP to Cognition
 - The SyNAPSE program has this problem
- Even solutions to the more narrowly defined ISP back-end problem are not obvious



Modular Intermediate Form

- One way to possibly bridge the gap is to define a computational model that “spans” the gap
- A candidate has been proposed by Albus and many others:
- The Cortical Computation Unit (CCU)



Desirable Characteristics

- Modular
- Distributed representation
- Hierarchical, bi-directional information flow
- Massively parallel
- Scales
- Learning / Self-organizing
- Does a kind of Bayesian inference
- Solves the problem ...

Albus: What is the path to success for reverse engineering the brain?

Pick the right level of resolution

- overall system level (central nervous system)
- AI and Cognitive Neuroscience units (e.g., cortical regions)
- macro-computational units (e.g., cortical hypercolumns & loops)
- micro-computational units (e.g., cortical microcolumns & loops)
CCUs
- neural clusters (e.g., spinal and midbrain sensory-motor nuclei)
- neurons (elemental computational units) – input/output functions
- synapses (electronic gates, memory elements) – synaptic phenomena
- Mainstream Neuroscience & Neural Nets (y) – molecular phenomena

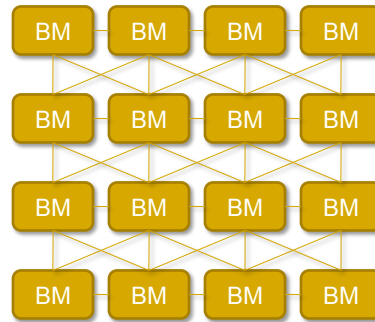
Slide courtesy James Albus, "Reverse Engineering the Visual System" From the PSU / Intel / ONR / NSF "Massively Parallel, Adaptive Computing" Workshop, March 2009

Modular Hierarchies

- Computer engineers make extensive use of modular, hierarchical design, can we assume the same for these models?
- In neocortex the fundamental unit of computation appears to be the cortical minicolumn (Mountcastle)
 - A minicolumn is a vertically organized group of about 80-100 neurons which traverses the thickness of the gray matter (~3 mm) and is about 50 μ in diameter
 - Neurons in a column tend to communicate mostly vertically with other neurons in the different layers in the same column
- These are subsequently organized into larger columns variously called just "columns", "cortical columns", "hypercolumns", or sometimes "modules"
 - Note, columnar organization is not universally accepted in the neuroscience community

“Bayesian Memory” (BM) Building Block

- An approximation to a CCU
- A BM sees only a subset of its input BMs and each BM's subset is slightly different
- Inference is performed over small sub-blocks
- The number of blocks increases linearly
- Relies heavily on sparse, distributed representations

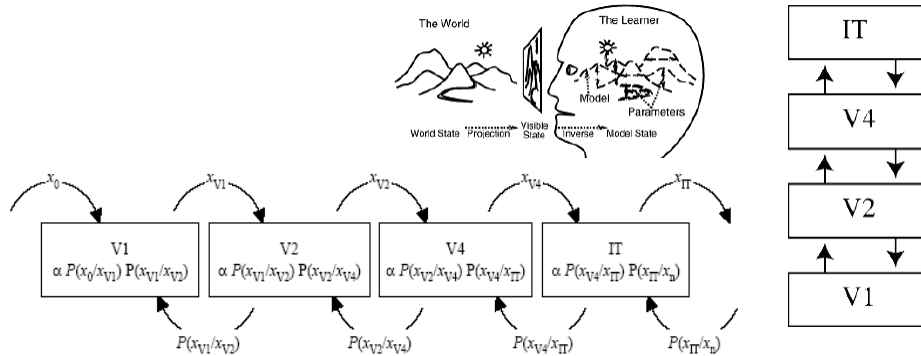


From *Big Brain* by Gary Lynch and Rick Granger (Palgrave McMillan 2008):

- “Although the ‘front end’ circuits of the brain, with their point-to-point circuit designs, specialize in their own particular visual and auditory inputs, the rest of the brain converts these to random-access encodings in association areas throughout cortex. ... these areas take initial sensory information and construct grammars
- “These are not grammars of linguistic elements, they are grammatical organizations (nested, hierarchical, sequences of categories) of percepts – visual, auditory, and other
- “Processing proceeds by incrementally assembling these constructs ... these grammars generate successively larger ‘proto-grammatical fragments,’ eventually constituting full grammars

- “They thus are not built in the manner of most hand-made grammars; they are statistically assembled, to come to exhibit rule-like behavior, of the kind expected for linguistic grammars
- “Proto-grammatical fragments capture regularities that are empirically found to suffice both for recognizing and generating grammatical sequences
- “Auditory pathways in our brains grew and lengthened building voice-sounds into words, words into phrases, phrases into sentences”

- Information needs to flow both ways
- Assume that conditional probabilities / priors - model the world
- Bi-Directional Belief propagation – e.g., visual cortex model
- Inference as the basic computation



Lee and Mumford Visual cortex model

A BM module then has two parts:

- An **input** that approximately learns the probability distribution of its inputs
 - A table of vectors, which is called a codebook and implements Vector functionality
 - Approximates the input probability distribution
 - Learns in an unsupervised manner by allocating new vectors and/or moving existing vectors
- A Vector Quantizer is an example of such a function – an “entropy” maximizing data reducer

- An **output** that creates a new representation of the codebook vectors to send up the hierarchy
 - A table of vectors, one for each codebook vector
 - The output vectors are sparse and are of a higher dimension than the space they span
 - In some implementations they are random, in Numenta's HTM they just pass up the index of the winning codebook vector
 - Ideally they would self-organize, as in a Self-Organizing Map (SOM) to capture the one or two dimensions of highest invariance of the input space

- The winning codebook vector is the most likely given some input
- W is some vector weight
- In VQ terms it specifies the width of the region surrounding the codebook vector
- It can be thought of as the "prior" probabilities

↑ Winning Output Vector

Input Vector	W	i	Output Vector
1000100111000	0.12	0	10010101000
0001101001010	0.05	1	01011100110
.	.	.	.
.	.	.	.
.	.	.	.
0111010101000	0.001	n	10001000100

↑ Evidence for Input Vectors

Belief Propagation

- The model looks good, but something missing
- It is generally assumed that biological systems perform a kind of inference over the knowledge they have learned
- If so, then perhaps Bayesian Inference
 - **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference**, by Judea Pearl, Morgan Kaufmann, 1988, ISBN-10: 1558604790
- Assume that our modular hierarchy is a directed Bayesian network
 - Vertices are objects which have local information and carry out local computations by updating of probability distribution via message passing

Most Likely Computation – Influence From Above

- OK, I lied, the computation of the most-likely codebook vector is actually more complicated
- The reason is that it is a result of the influence of “belief” propagated both from above and from below

Vector from upper BM

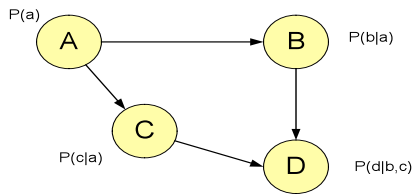
Evidence for Output Vectors ↑ ↓ ↑ Winning Output Vector

Input Vector	W	i	Output Vector
1000100111000	0.12	0	10010101000
0001101001010	0.05	1	01011100110
.	.	.	.
.	.	.	.
.	.	.	.
0111010101000	0.001	n	10001000100

Winning Input Vector ↓ ↑ ↑ Input Vector Evidence

Input Vector

A Simple Bayesian Network

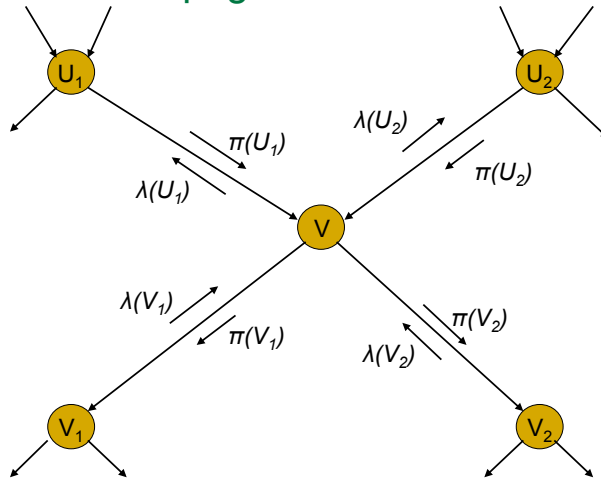


CPT for node D, there are similar tables for B and C

$P(d b,c)$	d_1	d_2
b_1, c_1	0.5	0.5
b_2, c_1	0.3	0.7
b_1, c_2	0.9	0.1
b_2, c_2	0.8	0.2

CPT is “Conditional Probability Table”

Pearl's Belief Propagation



Singliar Slides

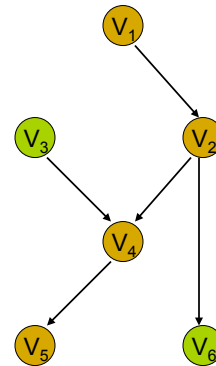
Maseeh College of Engineering and Computer Science

The Evidence

- Evidence – values of observed nodes
 - $V_3 = T, V_6 = 3$
- Our belief in what the value of V_i 'should' be changes
- This belief is propagated
- As if the CPTs became

$V_3=T$	1.0
$V_3=F$	0.0

P	$V_2=T$	$V_2=F$
$V_6=1$	0.0	0.0
$V_6=2$	0.0	0.0
$V_6=3$	1.0	1.0

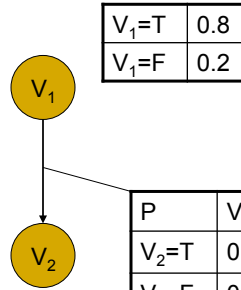


Slides by Tomas Singliar, CMU

Maseeh College of Engineering and Computer Science

The π Messages

- What are the messages?
- For simplicity, let the nodes be binary



The message passes on information

What information? Observe:

$$P(V_2 | V_1) = P(V_2 | V_1=T)P(V_1=T) + P(V_2 | V_1=F)P(V_1=F)$$

The information needed is the CPT of $V_1 = \pi_{V_1}(V_1)$

π Messages capture information passed from parent to child

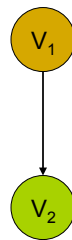
Singular Slides

Maseeh College of Engineering and Computer Science

The λ Messages

- What about λ ?

Assume $E = \{V_2\}$ and compute by Bayes rule:



$$P(V_1 | V_2) = \frac{P(V_1)P(V_2 | V_1)}{P(V_2)} = \alpha P(V_1)P(V_2 | V_1)$$

The information not available at V_1 is the $P(V_2 | V_1)$. To be passed upwards by a λ -message. Again, this is not in general exactly the CPT, but the *belief* based on evidence down the tree.

- The messages are $\pi(V)=P(V|E^+)$ and $\lambda(V)=P(E^-|V)$

Singular Slides

Maseeh College of Engineering and Computer Science

Combination of evidence

$$\begin{aligned}
 P(V | E) &= P(V | E_V^+, E_V^-) = \alpha' P(E_V^+, E_V^- | V) P(V) = \\
 &\alpha' P(E_V^- | V) P(E_V^+ | V) P(V) = \alpha P(E_V^- | V) P(V | E_V^+) = \\
 &\alpha \lambda(V) \pi(V) = BEL(V)
 \end{aligned}$$

- α is the normalization constant
- normalization is not necessary (can do it at the end)
- but may prevent numerical underflow problems

Singliar Slides

Maseeh College of Engineering
and Computer Science

Messages to pass

- We need to compute $\pi_{X|Y}(x)$

$$\pi_{X|Y}(x) = \alpha \pi_X(x) \prod_{k \neq j} \lambda_{Y_k X}(x)$$

- Similarly, $\lambda_{X|Y}(x)$, X is parent, Y child
- Symbolically, group other parents of Y into $V = V_1, \dots, V_q$

$$\lambda_{Y_j X}(x) = \sum_{y_j} \lambda_{Y_j}(y_j) \sum_{v_1, \dots, v_q} p(y | v_1, \dots, v_q) \prod_{k=1}^q \pi_{V_k Y_j}(v_k)$$

Singliar Slides

Maseeh College of Engineering
and Computer Science

The Pearl Belief Propagation Algorithm

- Iterate until no change occurs
 - (For each node X) if X has received all the π messages from its parents, calculate $\pi(x)$
 - (For each node X) if X has received all the λ messages from its children, calculate $\lambda(x)$
 - (For each node X) if $\pi(x)$ has been calculated and X received all the λ -messages from all its children (except Y), calculate $\pi_{xY}(x)$ and send it to Y.
 - (For each node X) if $\lambda(x)$ has been calculated and X received all the π -messages from all parents (except U), calculate $\lambda_{xU}(x)$ and send it to U.

- Compute $BEL(X) = \lambda(x)\pi(x)$ and normalize

Singliar Slides

Maseeh College of Engineering
and Computer Science

Most Graphs are not Polytrees

- Cutset conditioning
 - Instantiate a node in cycle, absorb the value in child's CPT
 - Do it with all possible values and run belief propagation
 - Sum over obtained conditionals
 - Hard to do
 - Need to compute $P(c)$
 - Exponential explosion - minimal cutset desirable (also NP-complete)
- Clustering algorithm
- Approximate inference
 - Sampling methods
 - Loopy BP

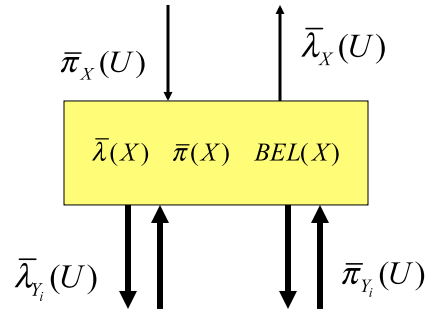
Singliar Slides

Maseeh College of Engineering
and Computer Science

Now To Add BBP to The BM

- A node in our hierarchy then represents a variable and is part of a larger, acyclic graph
- Child regions Y_1 and Y_2 , parent U

- 1) $\lambda(x_k) = \prod_j \lambda_{y_j}(x_k)$
- 2) $\lambda(u_m) = \sum_x \lambda(x) P(x | u_m)$
- 3) $\pi(x_k) = \sum_u P(x_k | u) \pi_x(u)$
- 4) $BEL(x_k) = \alpha \lambda(x_k) \pi(x_k)$
- 5) $BEL(x_k) = \alpha \pi(x_k) \prod_{i \neq j} \lambda_{y_i}(x_k)$



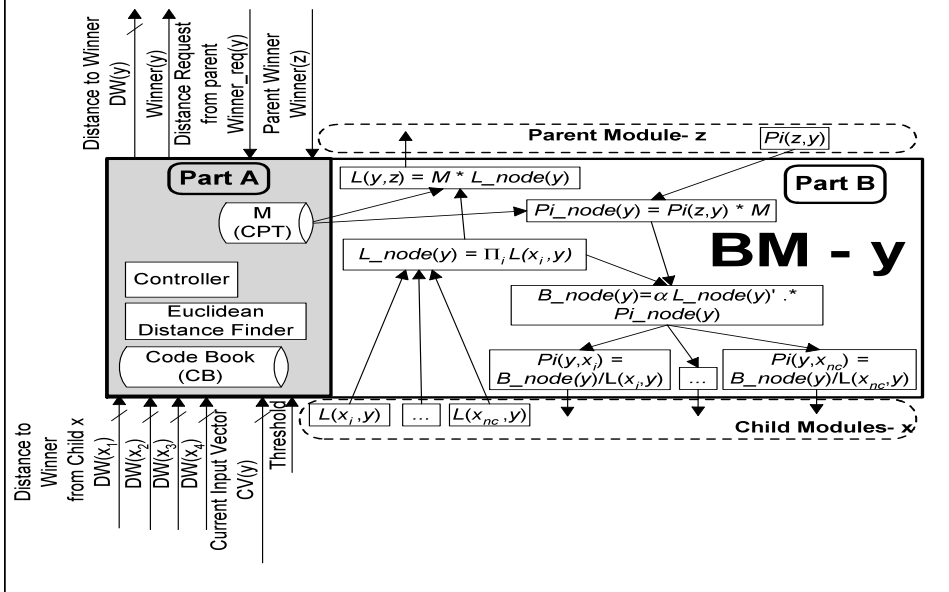
2/26/10

Hammerstrom

43

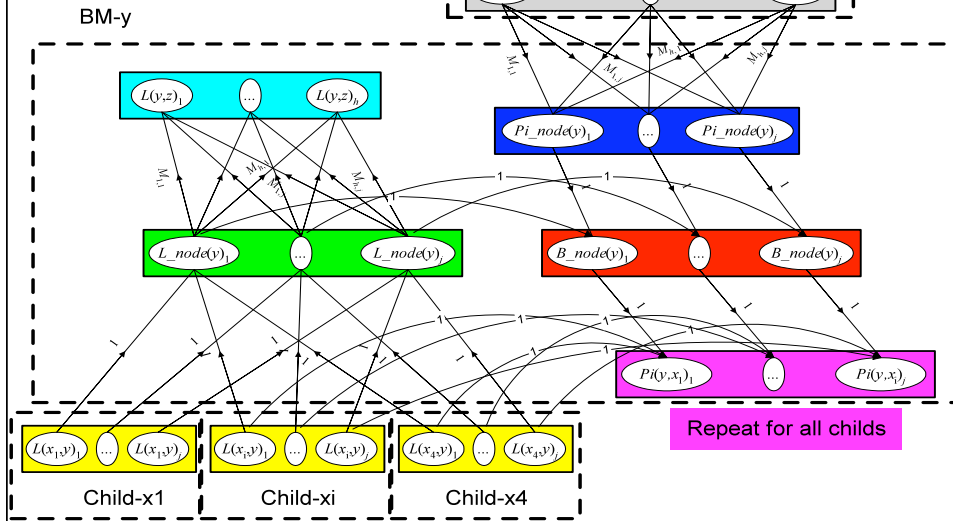
Maseeh College of Engineering and Computer Science

Voilà - A "Bayesian Memory"



Neural Network Equivalent of BBP-PA

- For 4K CB entries
- No. Neurons ~ 32e3
- Synapses ~ 34e6
- NN derived from Hawkins' paper



HTM – Hierarchical Temporal Memory – Version 2

Numenta

- The HTM algorithm is the work of Jeff Hawkins and Dileep George
- Jeff (Palm Pilot inventor) founded the Redwood Neuroscience Institute, <http://redwood.berkeley.edu>
- From which has emerged a synthesis of a number of existing and new ideas of cortical operation
- These are highlighted in his book, "On Intelligence"
- The models have worked so well that he has now spun out a company, Numenta, Inc., www.numenta.com
- Our work has borrowed heavily from Jeff and Dileep



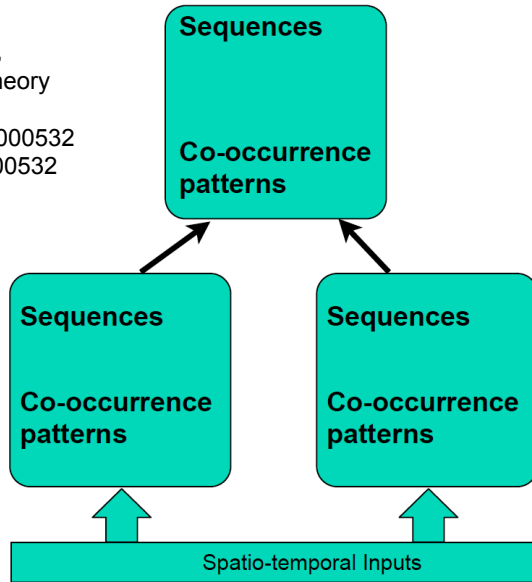
- Based on neuroscience principles, Jeff proposed that Cortex performs the following:
 1. Learns sequences of patterns
 2. Operates auto-associatively
 3. Captures invariants
 4. Is organized hierarchically, and
 5. Based on fundamental Bayesian principles
- The George / Hawkins model starts with a fairly general Bayesian module, very similar to the BM presented earlier
 - "A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex," D. George and J. Hawkins, *Proceedings of the ICJNN 2005*
- These modules then are combined into a hierarchy to form the Numenta *Hierarchical Temporal Memory* (HTM)

- **Hierarchical** -- HTMs are organized as a tree-shaped hierarchy of nodes. Each node implements a learning and memory function, that is, it encapsulates an algorithm
 - Lower-level nodes receive large amounts of input and send processed input up to the next level
 - In that way, the HTM Network abstracts the information as it is passed up the hierarchy

- **Temporal** -- During training, the HTM application must be presented with objects as they change over time
 - For example, during training of the Pictures application, the images are presented first top to bottom, then left to right as if the image were moving over time
 - Note that the temporal element is critical: The algorithm has been written to expect input that changes gradually over time

- **Memory** -- An HTM application works in two stages, which can be thought of as training memory and using memory
 - During training, the HTM Network learns to recognize patterns in the input it receives. Each level in the hierarchy is trained separately
 - In the fully trained HTM Network, each level in the hierarchy knows -- has in memory -- all the objects in its world
 - During inference, when the HTM Network is presented with new objects, it can determine the likelihood that an object is one of the already known objects.

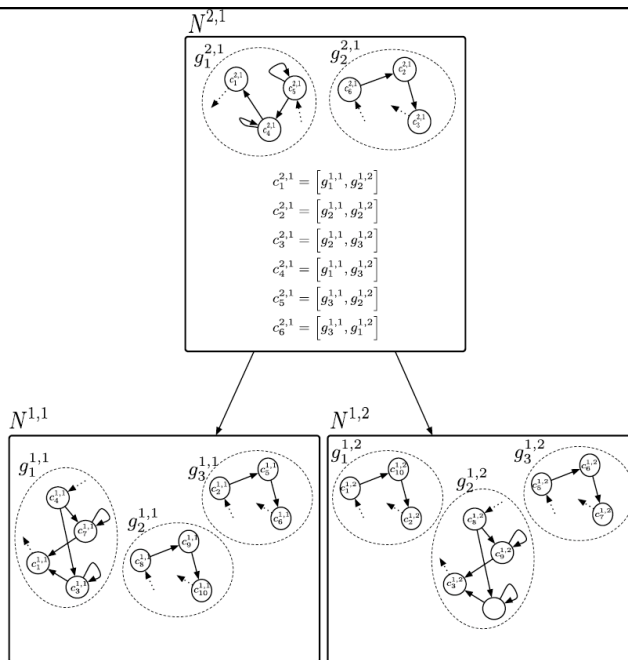
George D, Hawkins J, 2009,
 "Towards a Mathematical Theory
 of Cortical Micro-circuits,"
PLoS Comput Biol 5(10):e1000532
 doi:10.1371/journal.pcbi.1000532



2/26/10

51

Courtesy Dileep George 2009



2/26/10

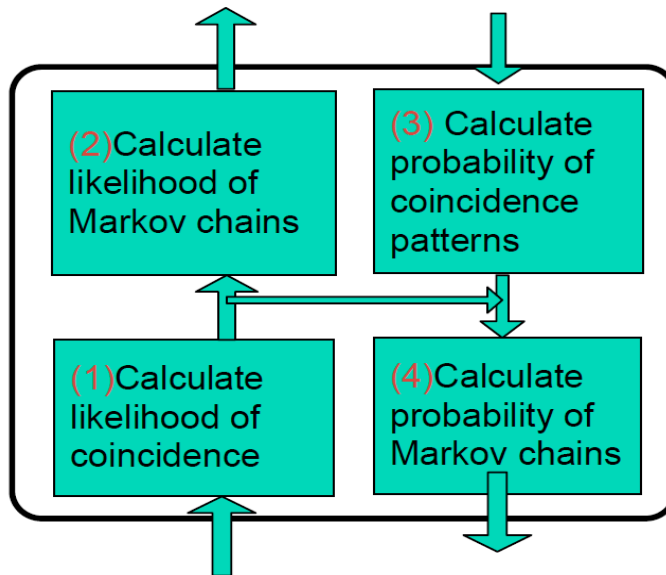
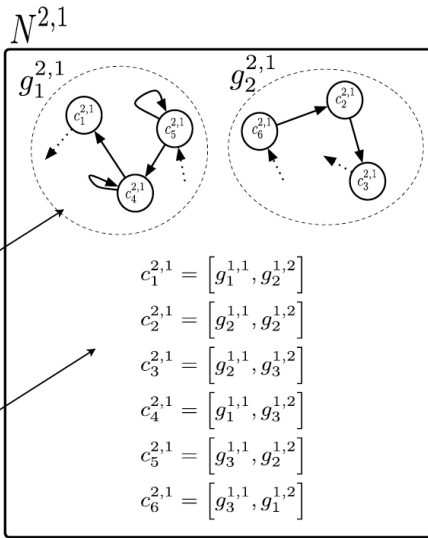
52

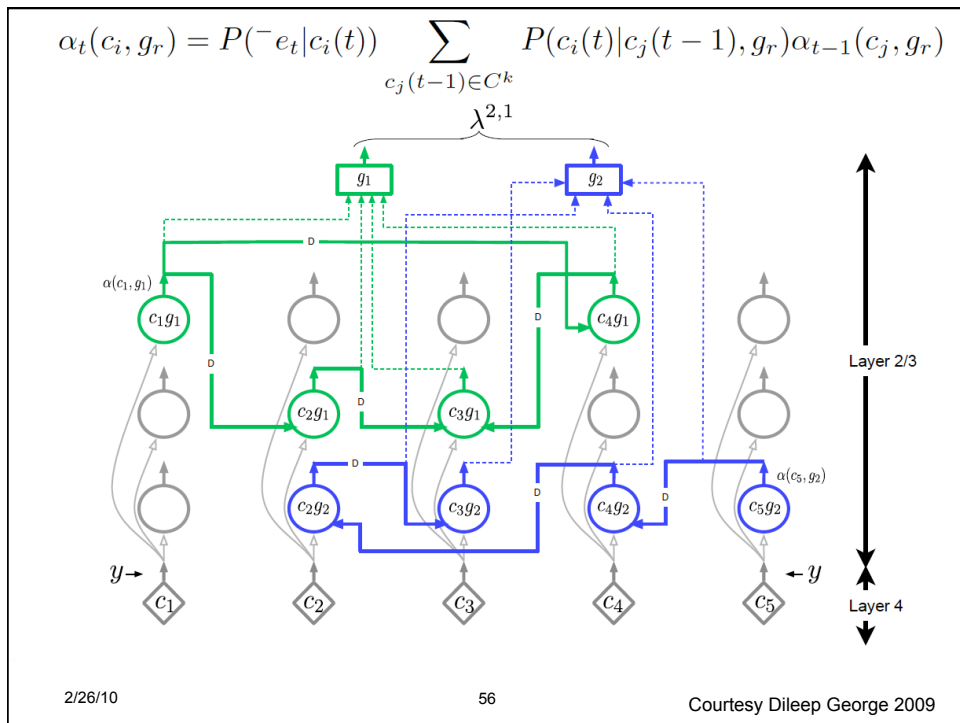
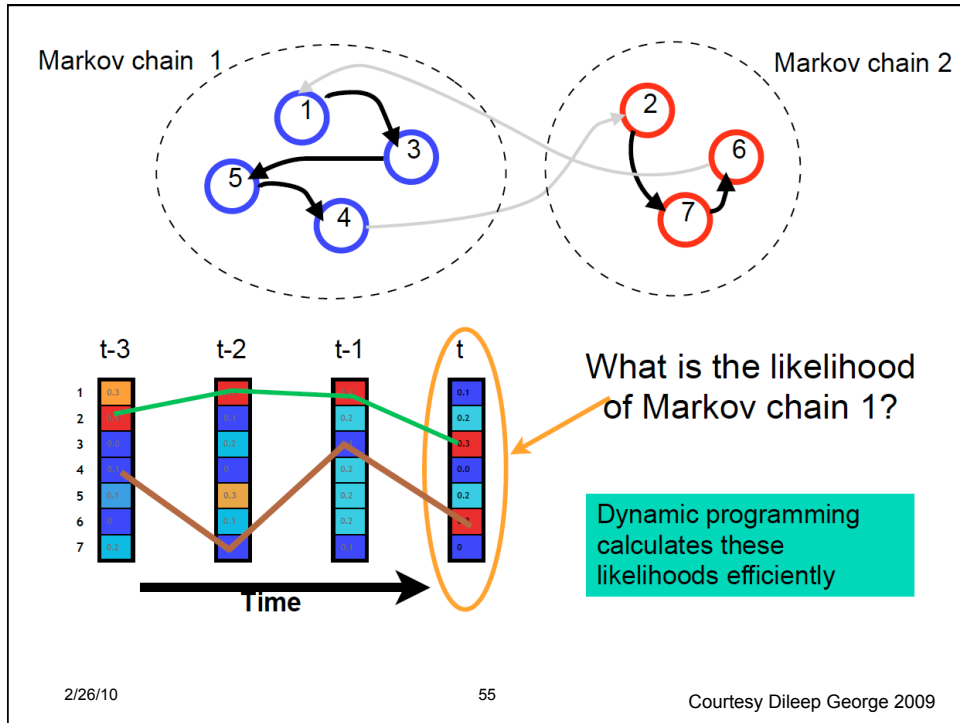
Courtesy Dileep George 2009

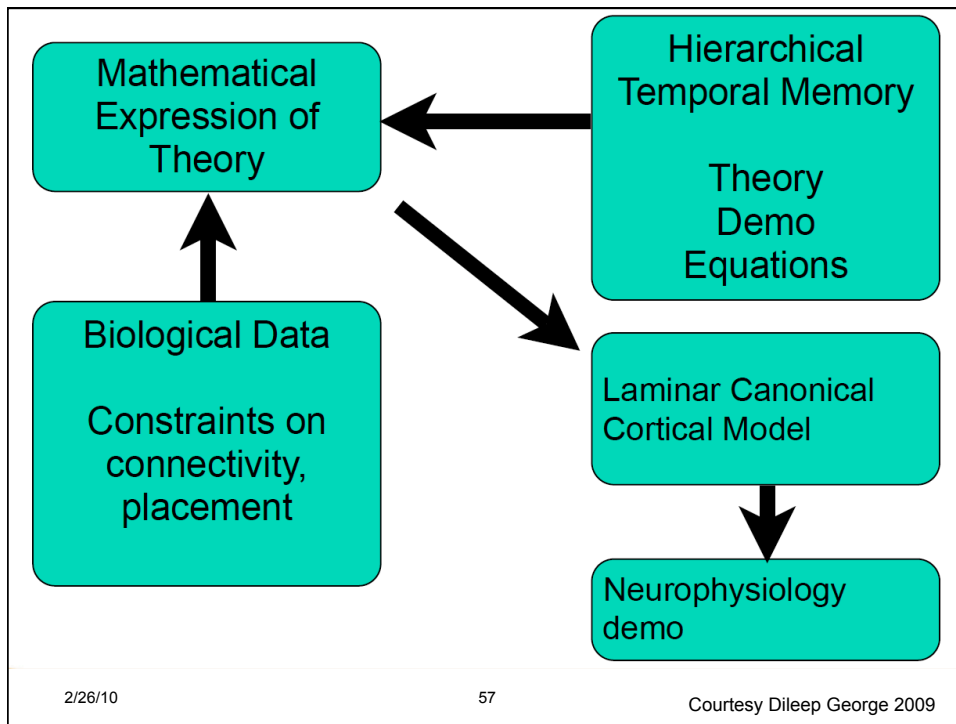
- Level by level learning
- Each node
 - Stores co-occurrence patterns
 - Learns sequences


Markov chains
(sequences)

Coincidence patterns







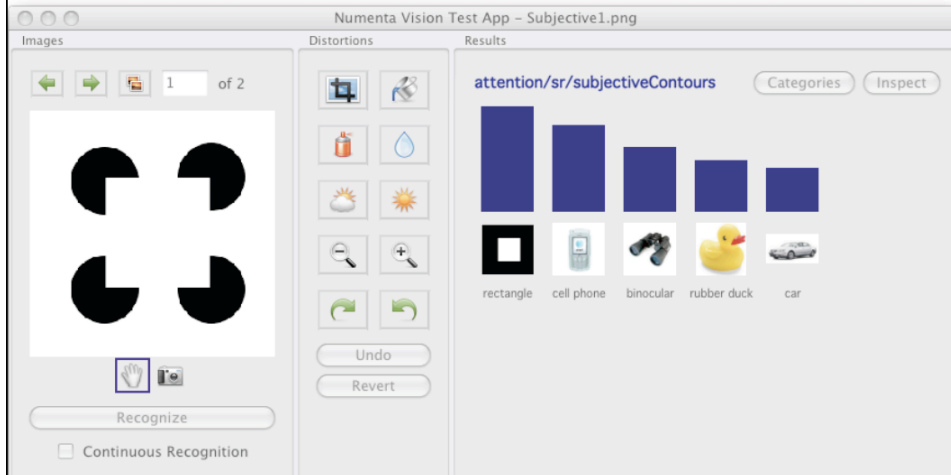

 Portland State
UNIVERSITY

- Hierarchy in space and time
 - Evidence for biology abstracting in space and time as signals proceed up the hierarchy
- Feed-forward and feedback connections
- Common cortical algorithm
- Inference using Bayesian belief propagation
- Sparse Distributed Representations
- Prediction using temporal context
- Biologically accurate

- Several computational vision applications
 - www.numenta.com

2/26/10 Hammerstrom 58 Maseeh College of Engineering and Computer Science

Kanisza Square Illusion ...



2/26/10

59

Courtesy Dileep George 2009

Mapping To Biology ...

- In the PLOS paper, "Towards a Mathematical Theory of Cortical Micro-circuits," they also speculate on mapping the algorithms to cortical circuitry
 - Use known facts about cortical organization to map belief propagation to cortical layers
 - "The vertical dimension of the cortical rectangle is only a few layers deep, the horizontal dimension is variable"
 - The states of the region are represented by neurons along the horizontal dimension of the cortical region
 - They then divide the horizontal dimension of the cortical region into a number of Compartments, where each compartment corresponds to a particular state of the region
 - This subdivision corresponds to a columnar organization of the cortex

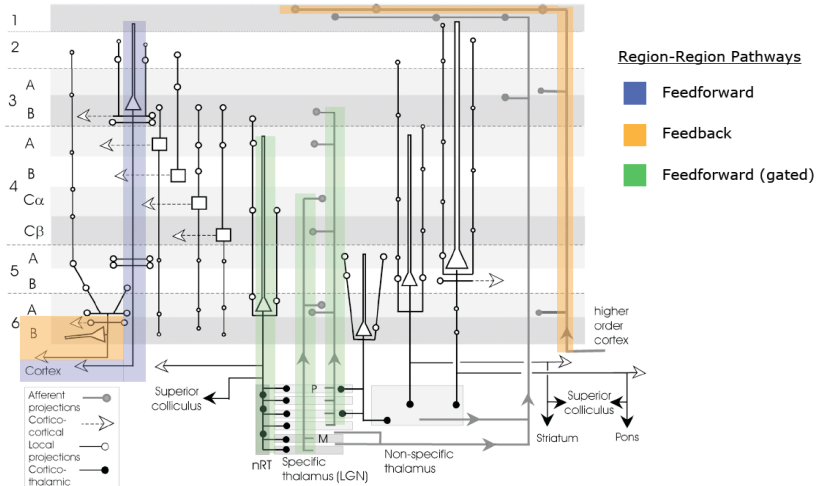
2/26/10

Hammerstrom

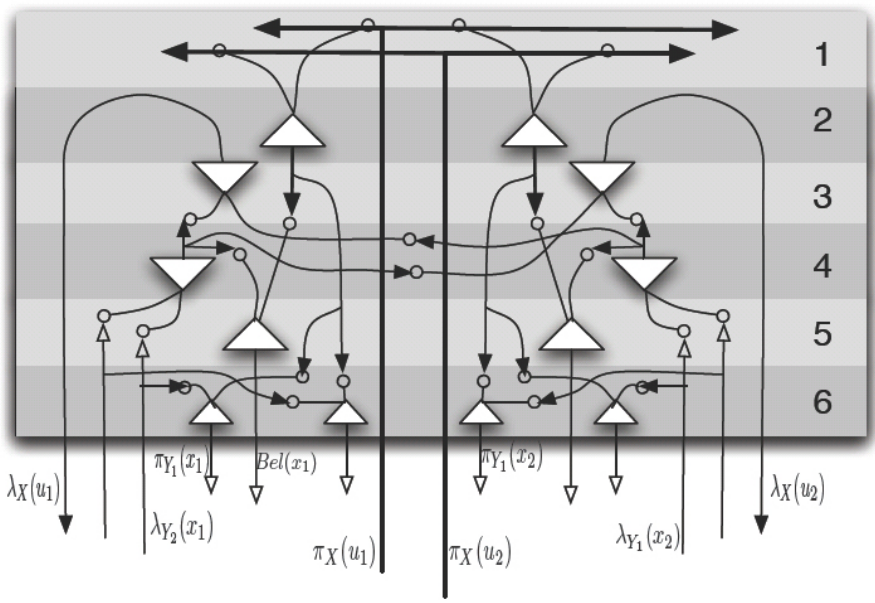
60

Maseeh College of Engineering
and Computer Science

Cortical Layers: Cells and Connectivity



16 From: Alex Thompson and Peter Bannister



2/26/10

62

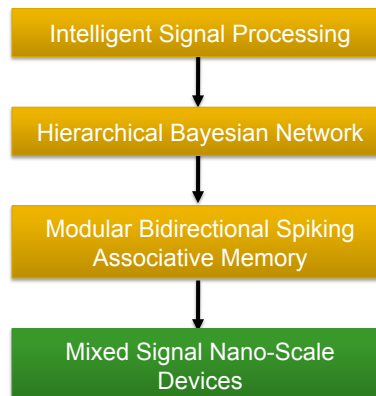
Courtesy Jeff Hawkins 2009

■ It is an interesting time!

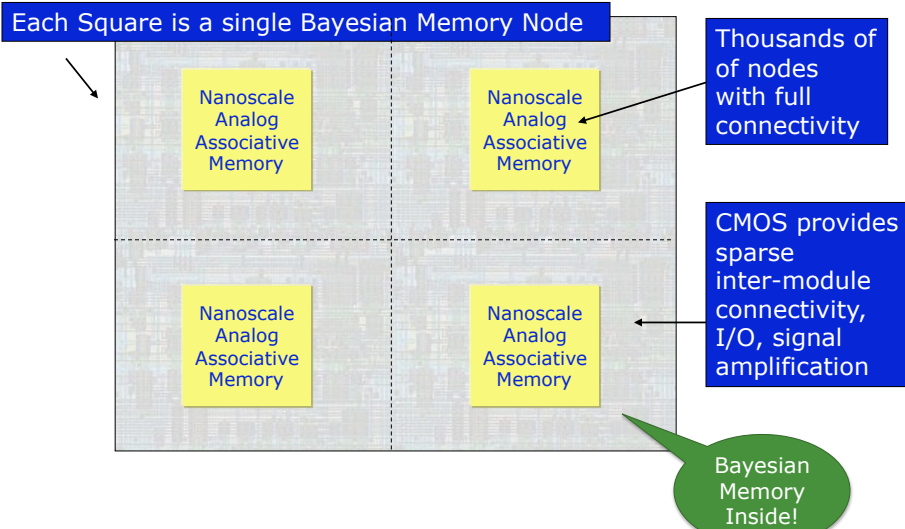
- SyNAPSE probably won't meet its original goals, but it will push the field forward – assuming there is no catastrophic failure – or too much hype ...
 - IEEE Tech Blog, "Cat Fight Brews Over Cat Brain"
- I personally believe that Jeff is on the right track
- And he is in this for the long haul, like the Terminator he will just keep on attacking this problem ...
- And for the work I do, there are all kinds of interesting hardware possibilities – especially with nano and molecular electronics!

A Path From Nanowires to ISP

- Our approach is top-down, not bottom up
- There is a large range of implementation options
 - 1000 Atom processors
 - Neuromorphic VLSI
 - Nano-grids
 - Other nano ...



Our Goal – A Commercial Product: The Field Adaptable Bayesian Array (FABA)



FABA – Long Term Goal

- A roughly 1 inch die containing several billion CMOS transistors and close to a trillion molecular devices
- Operating at over 10 Tera-Ops
- Extensive fault / defect tolerance
- Performs real-time, adaptive bayesian inference over very complex spatial and temporal knowledge structures
- Available in a portable, hand-held, low power devices