Technology Maturity for Adaptive Massively Parallel Computing

First Workshop 2009

March 2-3, 2009

Portland, OR, USA

PetaVision: A Software Architecture for Performing Petascale Simulations of Visual Cortex

Craig E Rasmussen

Dan Coates (PSU), Garret Kenyon

New Mexico Consortium,

Los Alamos National Laboratory

PetaVision is a software framework what can it do, lead to?

- A neural scientist wants to
  - use “machine inspired computation” to provide a research vehicle to inform neural science community
- A computer scientist wants to
  - use “neural inspired computation” to inform computer science community on massively parallel computation and learning (plasticity, adaptability)

I will try to accomplish three things

- Describe a proof of concept of our petascale simulations of visual cortex
- Describe the software architecture of PetaVision, a software framework for building high-performance neural simulations
- Examine differences between machine inspired computer and neural inspired computing *(warning: pure speculation)*

Why can't thinking machines think?

- Caveat
  - I'm not a neural scientist
  - I dabble in computer science
- If only our computers were:
  - Bigger, faster, had more memory, …
- Perhaps we just need enough simulated neural circuits?
- Perhaps if we a better circuit diagram?

Are computers fast enough?

- 10^9 neurons x 10^3 Hz x 100 op
  - ~ 10^{14} ops
- Roadrunner computer at LANL
  - 16x180 = 2880 hybrid nodes with 4 IBM Cell cores
  - Each Cell core has 8 SPU cores
    - 2880 x 4 x 8 = 92160 compute cores
  - Each SPU ~ 200 x 10^9 flops (single precision)
    - 92160 x 200 x 10^9
  - > 10^{15} flops

Are our computers big enough?

- Visual cortex
  - ~ 10^9 neurons
  - ~ 10^{13} synaptic connections
- Los Alamos' Roadrunner
  - 92,160 IBM cores
  - ~ 10,000 transistors per core
  - ~ 10^9 transistors
  - where are the equivalent connections

Do computers have enough memory

- Assume memory is in the synaptic connections
  - 10^9 neurons x 10^4 connections x 1 byte
  - ~ 10^{13} bytes
- Roadrunner
  - > 10^{13} bytes

Do computers have enough bandwidth

- Assume bandwidth is in the synaptic connections
  - 10^{13} connections x 1000 Hz
  - ~ 10^{16} bytes/sec
- Roadrunner
  - 11520 Cell processors x 20 Gbytes/sec
  - ~ 10^{14} bytes/sec

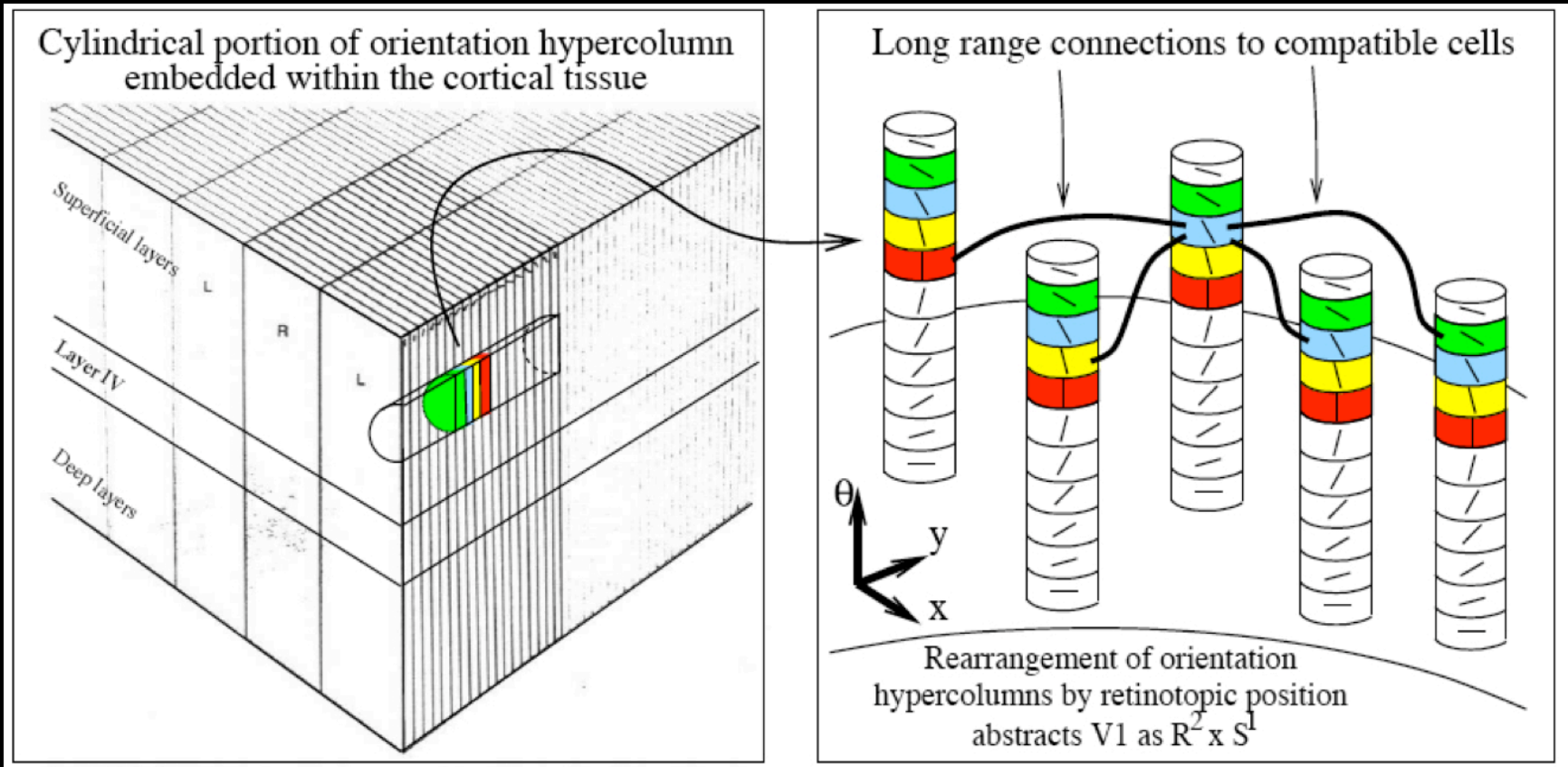
Why can't thinking machines think?

- Big enough
- Fast enough
- Has enough memory
- But lacks bandwidth
  - Using memory to simulate synaptic connections
  - Must time share access to memory
- So what, we'll just run 1000 times slower
  - if we program it correctly (access to memory)

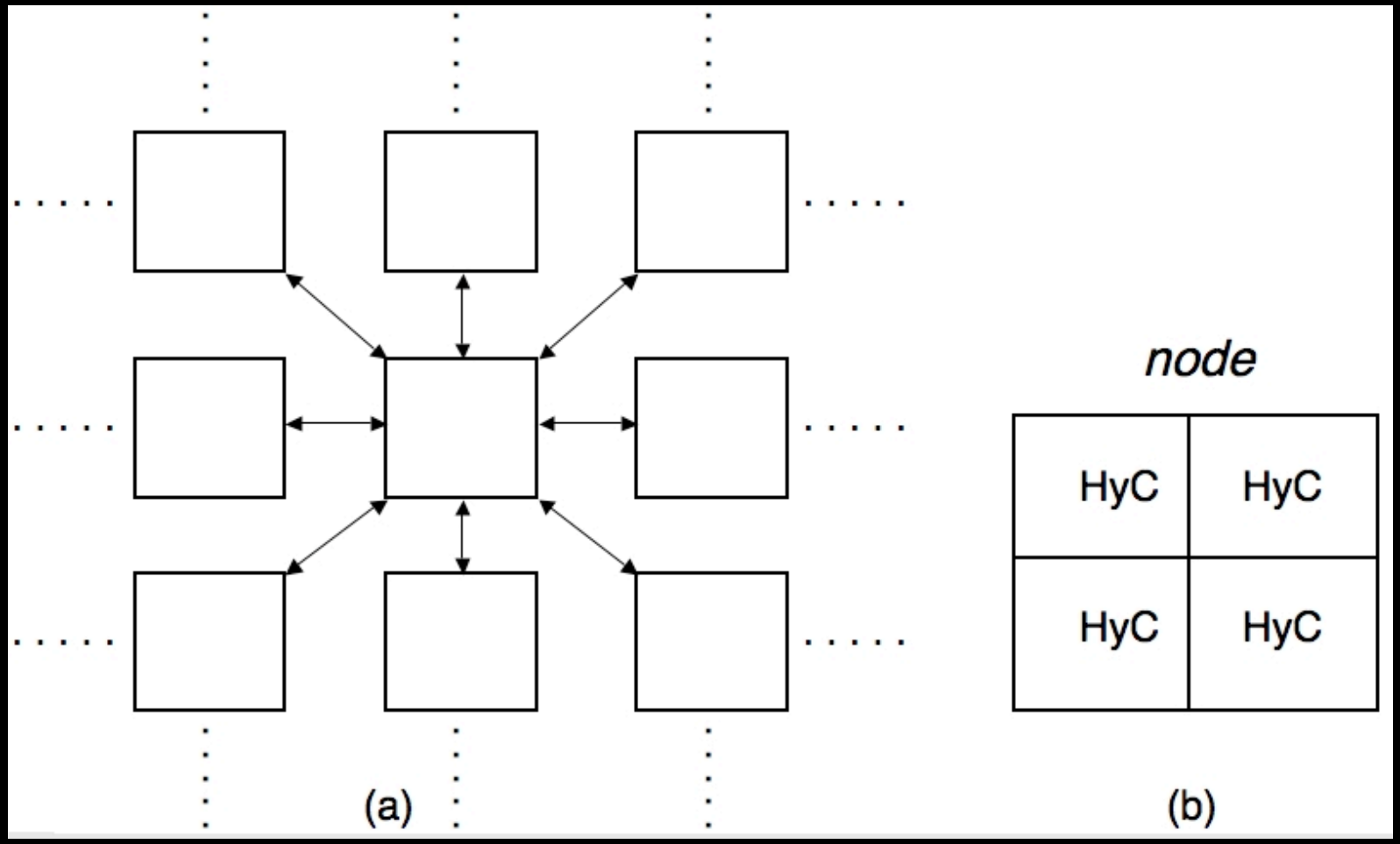
PetaVision: neural simulation proof of concept achieves 1.14 Petaflop/s

- Task was to examine possibility of designing and running a neural simulation *at scale*
  - what could we learn from about software design
- Cherry picked first application of PetaVision
- Ran leaky, integrate-and-fire neural model with cocircular excitation (Zucker)
  - One billion neurons
  - 1.14 Petaflop/s

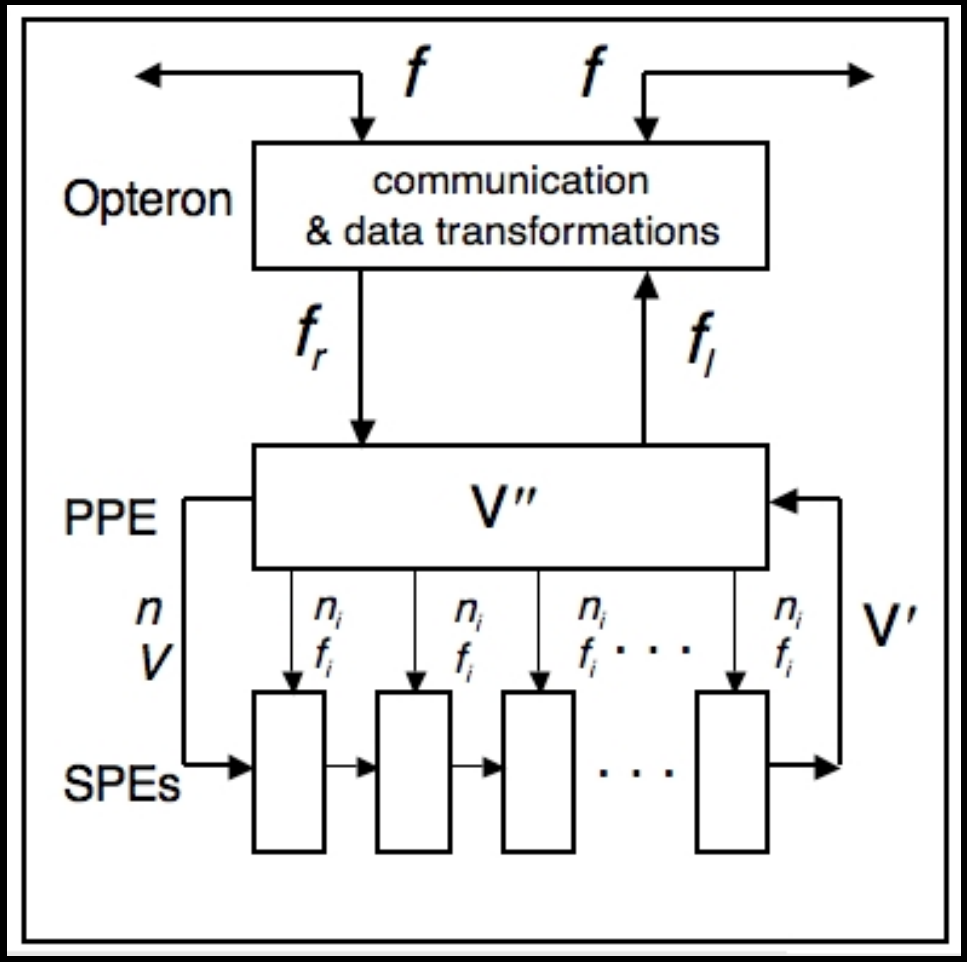
Hypercolumns distributed retinotopically communicates locally



Global connectivity using MPI



Communication and computation on one IBM Cell with 8 SPEs



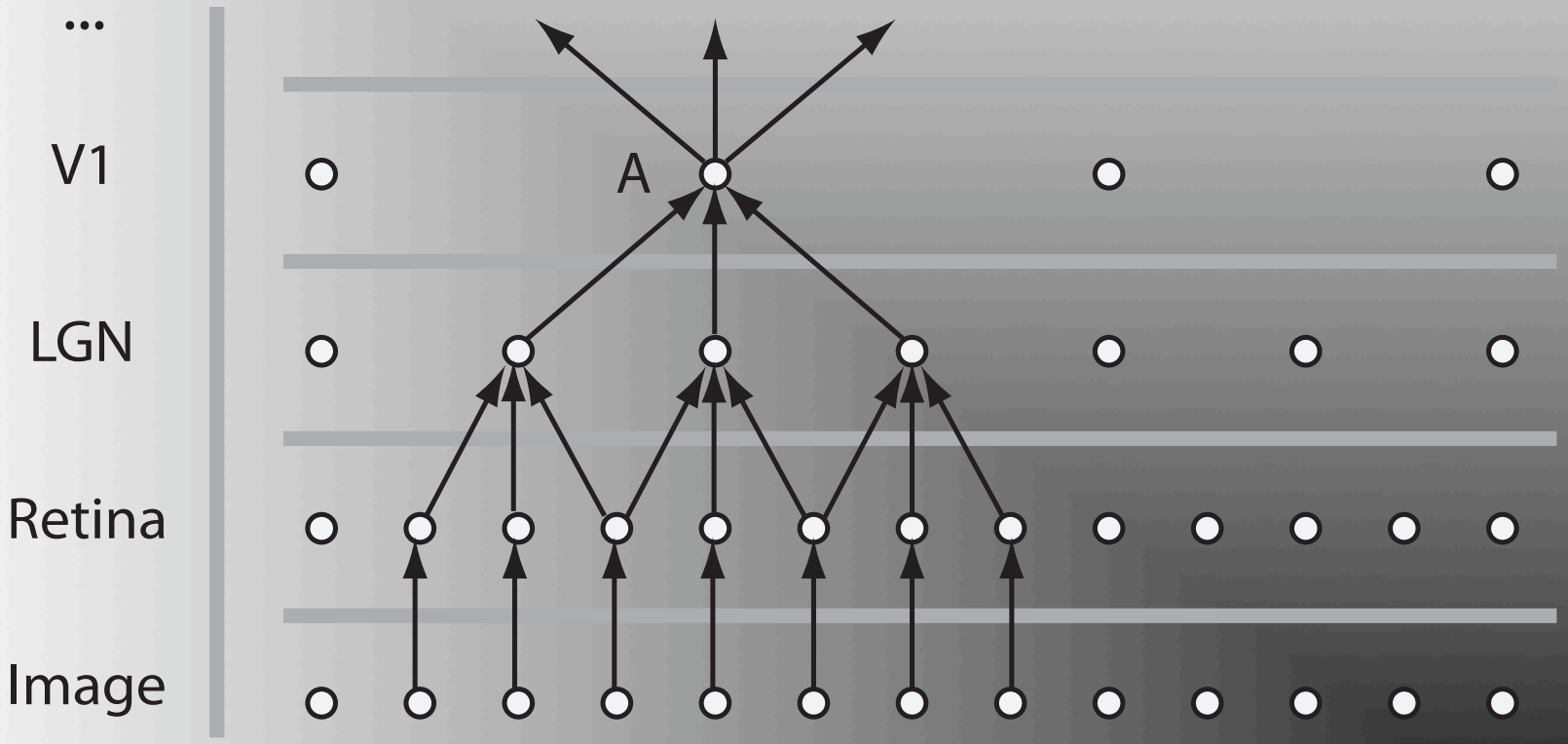
Conclusions from running our proof of concept model at scale

- It can be done
- Scales linearly with node count
  - global communication using MPI is not a limitation
- Programming is hard
- Memory access pattern is everything
  - must compute on *contiguous elements* of memory
  - otherwise you run 1000 times slower

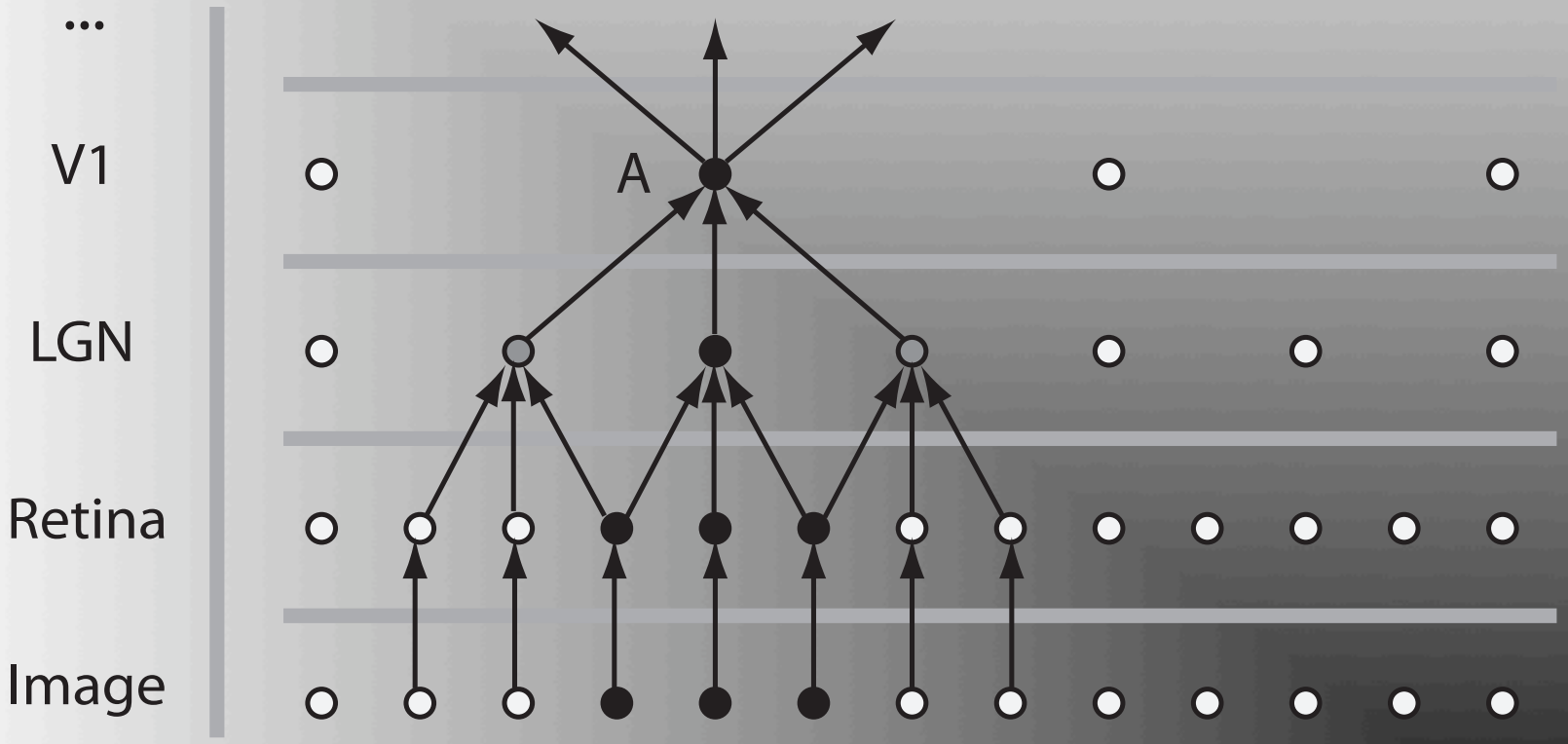
PetaVision software architecture is very simple

- HyPerColumns
  - manage data and control flow on a node
- HyPerLayers
  - manage multiple neural units within a layer
- HyPerConnections
  - connect neural layers (can be inhibitory)
  - synaptic connections are plastic/adaptable
  - Most of *science research* goes here

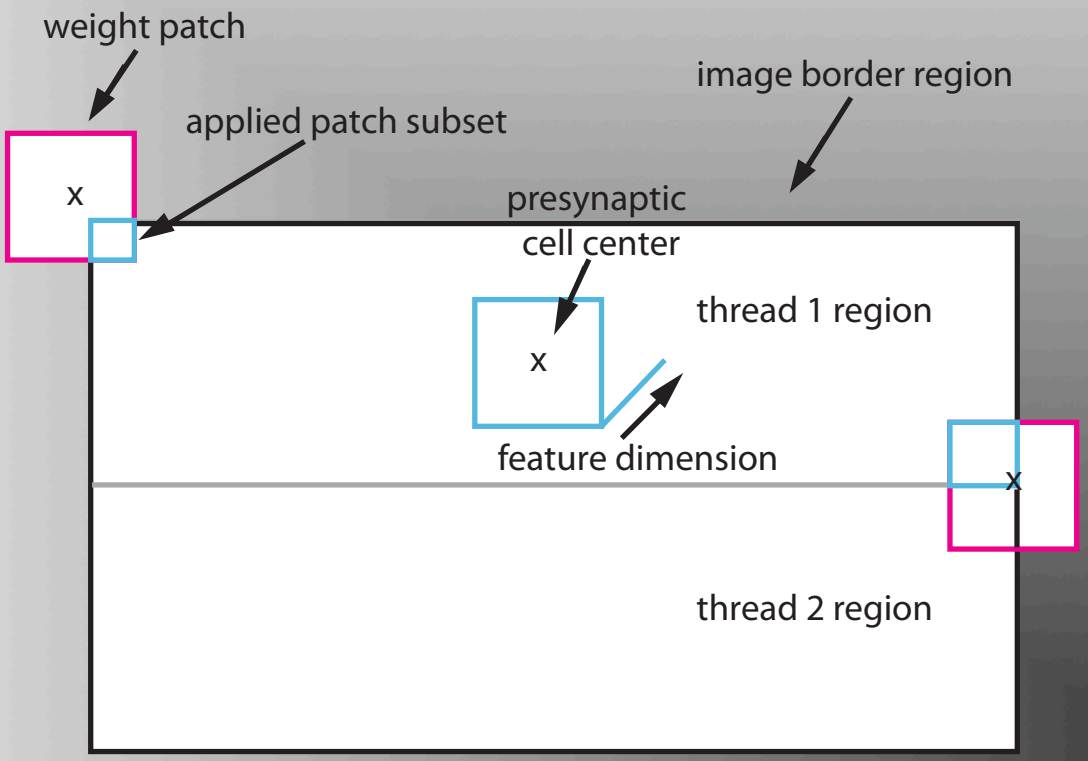
Neurons in layers are connected



Spiking neuron signals propagate (only feed-forward network shown)



Representation of connections: retinotopic patches of weights



Lessons we can learn from neural computing

- Some ideas I think are important
- Others are pretty speculative
  - but interesting

Real computers don’t make mistakes

- Visual cortex thrives on noise

Real computers make precise computations

- Orientation tuning in V1 is very broad
  - 20 degrees or more
  - overlapping implies multiple neurons will respond
- Apparently visual cortex takes advantage of the hierarchy to refine tuning
  - hyperacuity
  - when needed with attention?

Real computers are fully utilized

- THE FLOATING POINT UNIT must be fully utilized (LINPACK benchmark)
- Neuronal activity is inhibited to save energy
  - controversial statement, but consider
  - 20% of energy budget taken up by human brain
  - 80% of brain's energy goes into spiking
  - so you can't tell me that an animal that computes more efficiently isn't better adapted

Real computer architectures are flat and broad, not deep

- Roadrunner has a three deep hierarchy
  - roundly criticized for this
    - current codes won't run on it, communication is too complicated, too hard to program, …
    - all true
- Neural computing is broad *and* deep
  - visual cortex takes advantage of multiple layers

Real computers use dynamic routing for communication

- Message passing (MPI) is dynamic, any point to any point
- Neural routing tables (axons and dendritic branches) are static
- Neural communication channels are mostly local with a few long range connections

Real computers are programmed intelligently

- Well, so were my kids
  - and they are still learning
  - it's remarkable how long it takes
- But most neural programming is not via intelligent design
  - watch a child learn to walk

Real computers have programs with branches

- Neurons have dendrites but no “go to” statements
  - Hum?
  - But they learn and dynamically respond to input
- PetaVision code by design is not branchy
  - waste cycles (multiply by zero) rather than branch
  - better for load balance too

Real computers are digital with a binary encoding

- Neurons are spiky so neural computing is binary, on or off
- Hum, but what do spikes encode
  - rate (frequency, amplitude?)
  - arrival time

Real computers have synchronized clocks

- I don’t know of any synchronicity in visual cortex

Real computers have Von Neumann architectures

- A model for execution of a stored, static program
- Neural computation is dynamic
  - neural programs learn
- Synaptic weights vary with
  - with attention
  - in response to feedback

Real computers are finite state machines

- Neural computers are ….
- Let's see
  - Billions of neurons
    - each can be modeled as an FSM
  - Each neuron with 10K connections
    - each connection conducting current or not
- So why not an FSM
  - Otherwise stuck with Cartesian duality

Real computers are Turing equivalent

- Perhaps I'll stop while I'm ahead
  - or at least not too far behind

Petavision is a software framework

- Neural scientist specifies
  - layers
    - mostly just names
  - connections between layers
    - in terms of tunable specifications
- Get two levels of parallelism for free
  - message passing and multi-threading is hidden
- http://sourceforge.net/projects/petavision/

BACKUP