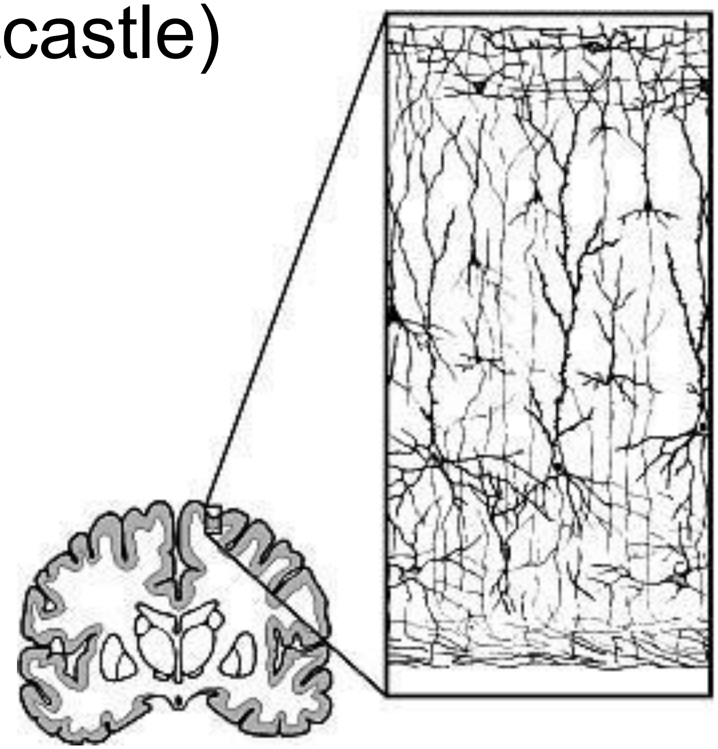
Scalable Inference in Hierarchical Models of the Neocortex

Thomas Dean
Brown University

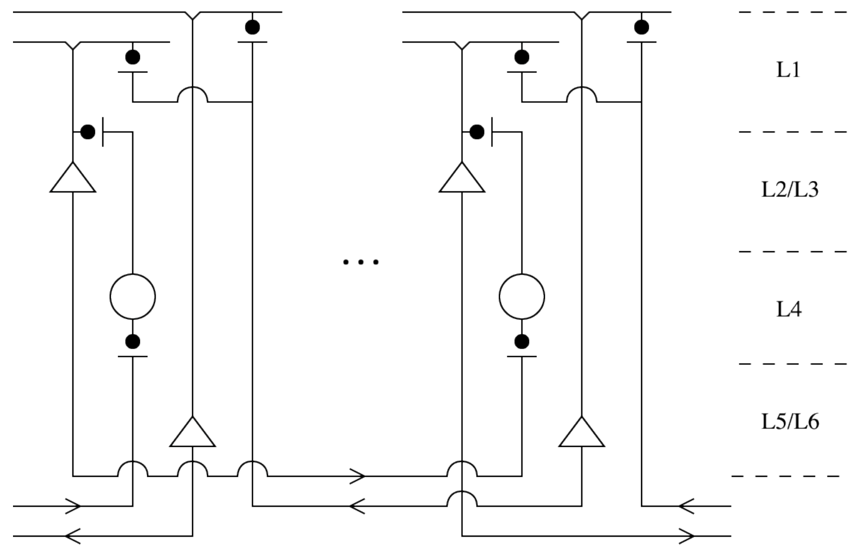
Anatomical Characteristics

- Folded sheet the size of a large napkin
- Regular structure replicated throughout
- Cortical columns (Mountcastle)

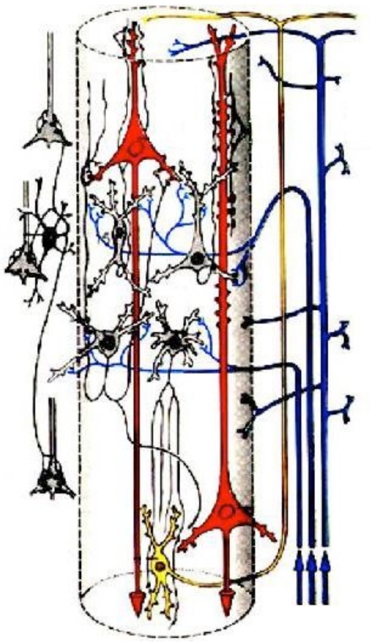


Cortical Columns

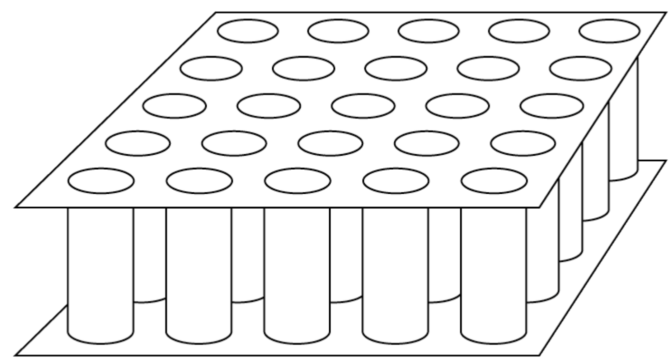
(Gilbert, 1993)



(Braitenburg and Schuz, 1991)

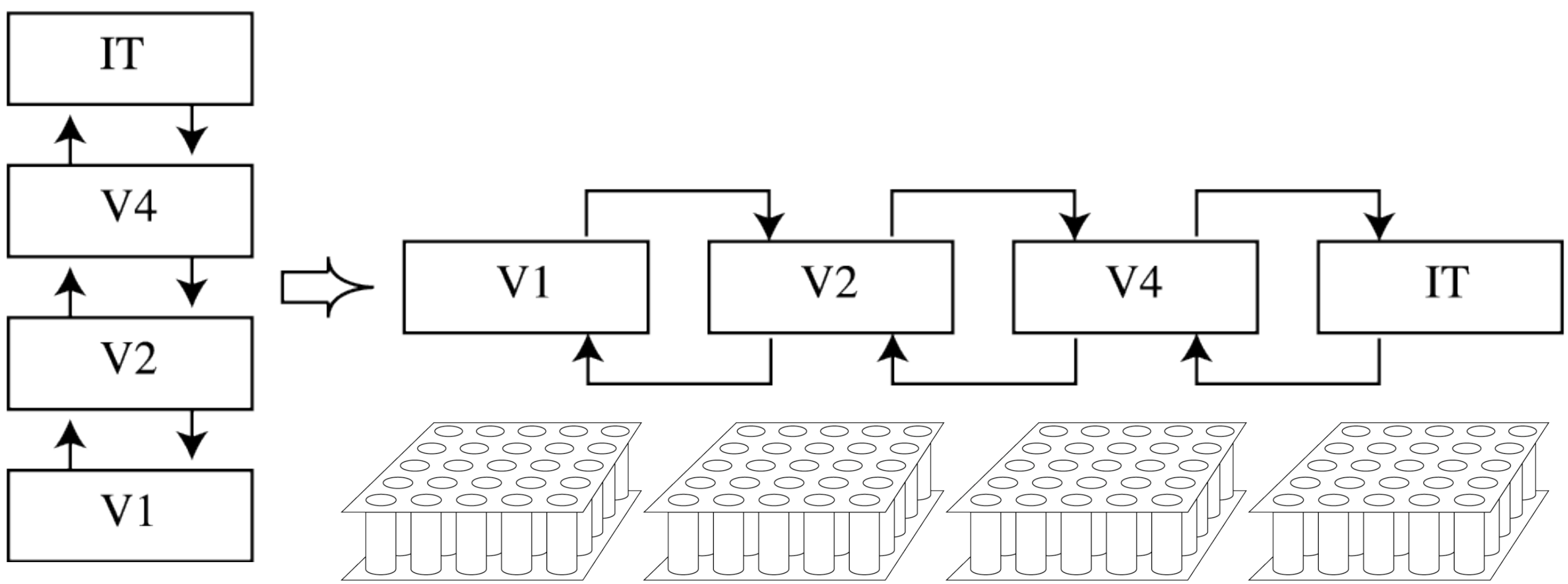


Receptive fields are mapped preserving spatial relationships (Hubel and Weisel)

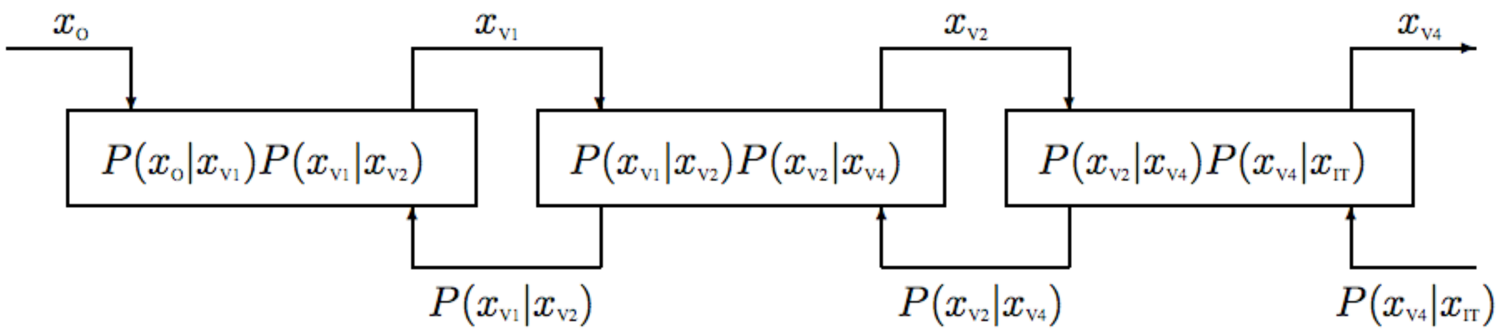


Functional Characteristics

- Hierarchical associative memory
- Pattern recognition and completion
- Powerful invariant representations
- Multiple modalities and resolutions



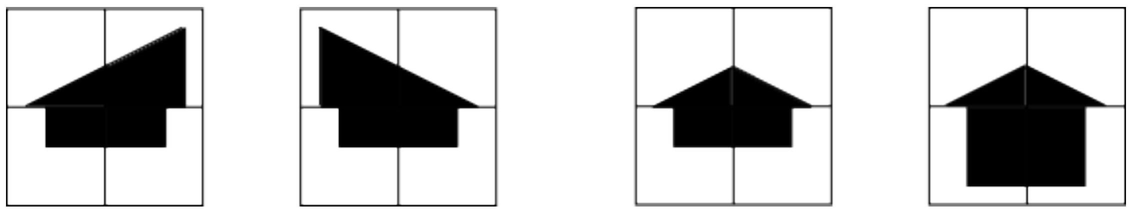
Lee and Mumford Model



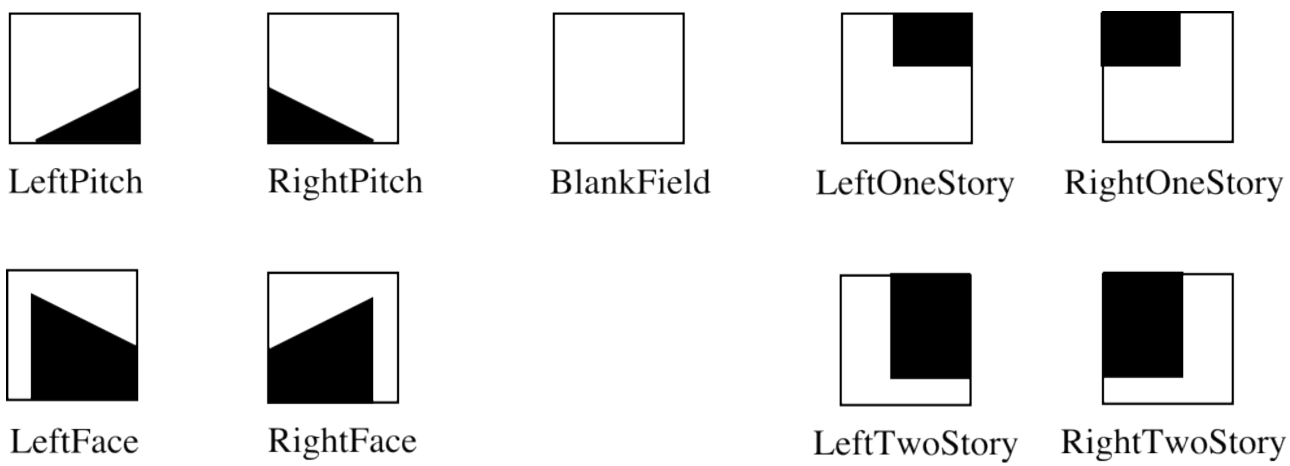
- Hierarchical model of the visual cortex
- Generative Bayesian statistical model
- Markov random fields (MRFs)
- Bottom-up data $\{x_i\}$ are fed forward
- Top-down priors $\{P(x_i|x_{i+1})\}$ are fed back

Generative Statistical Models

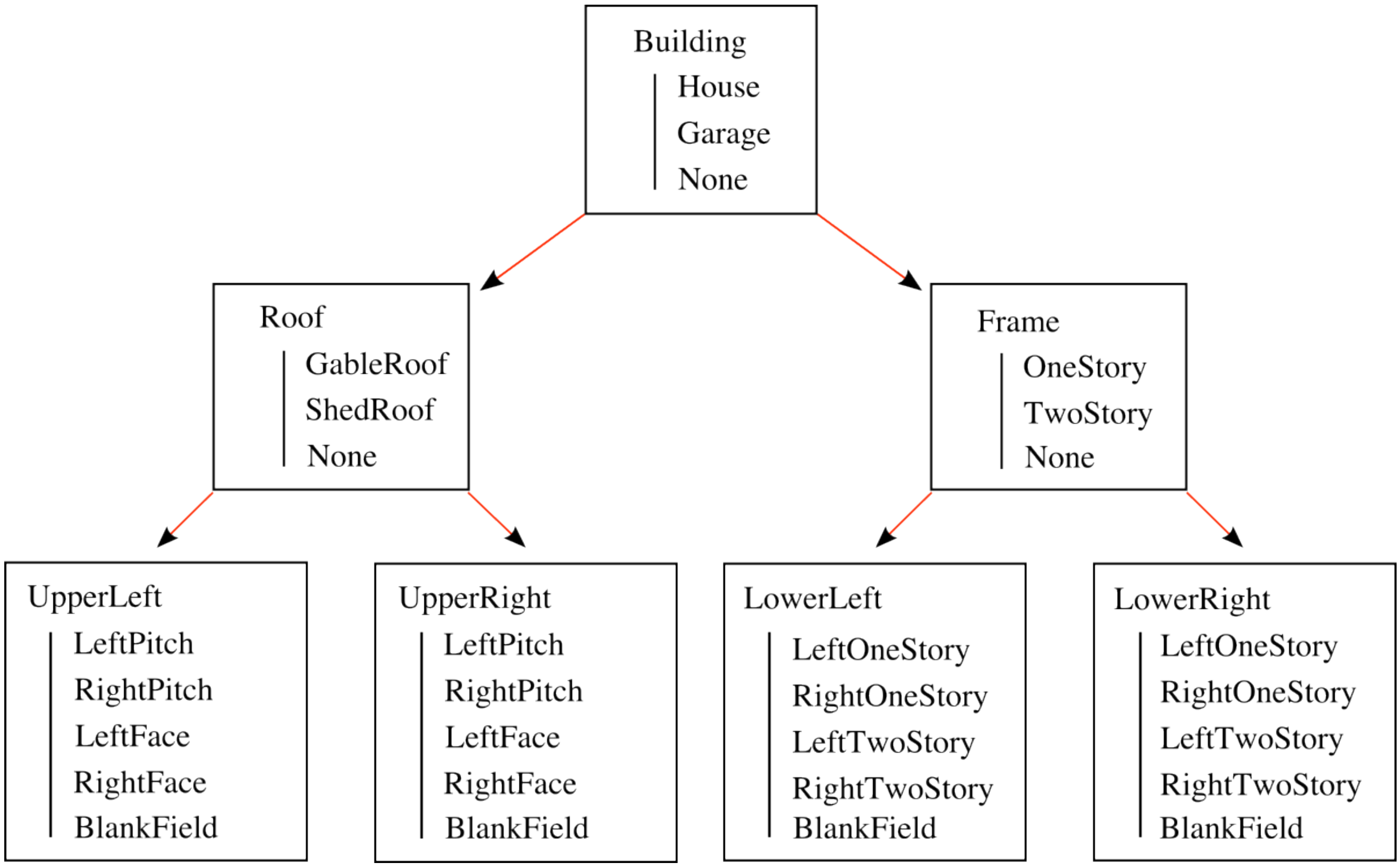
- Recognizing simple patterns

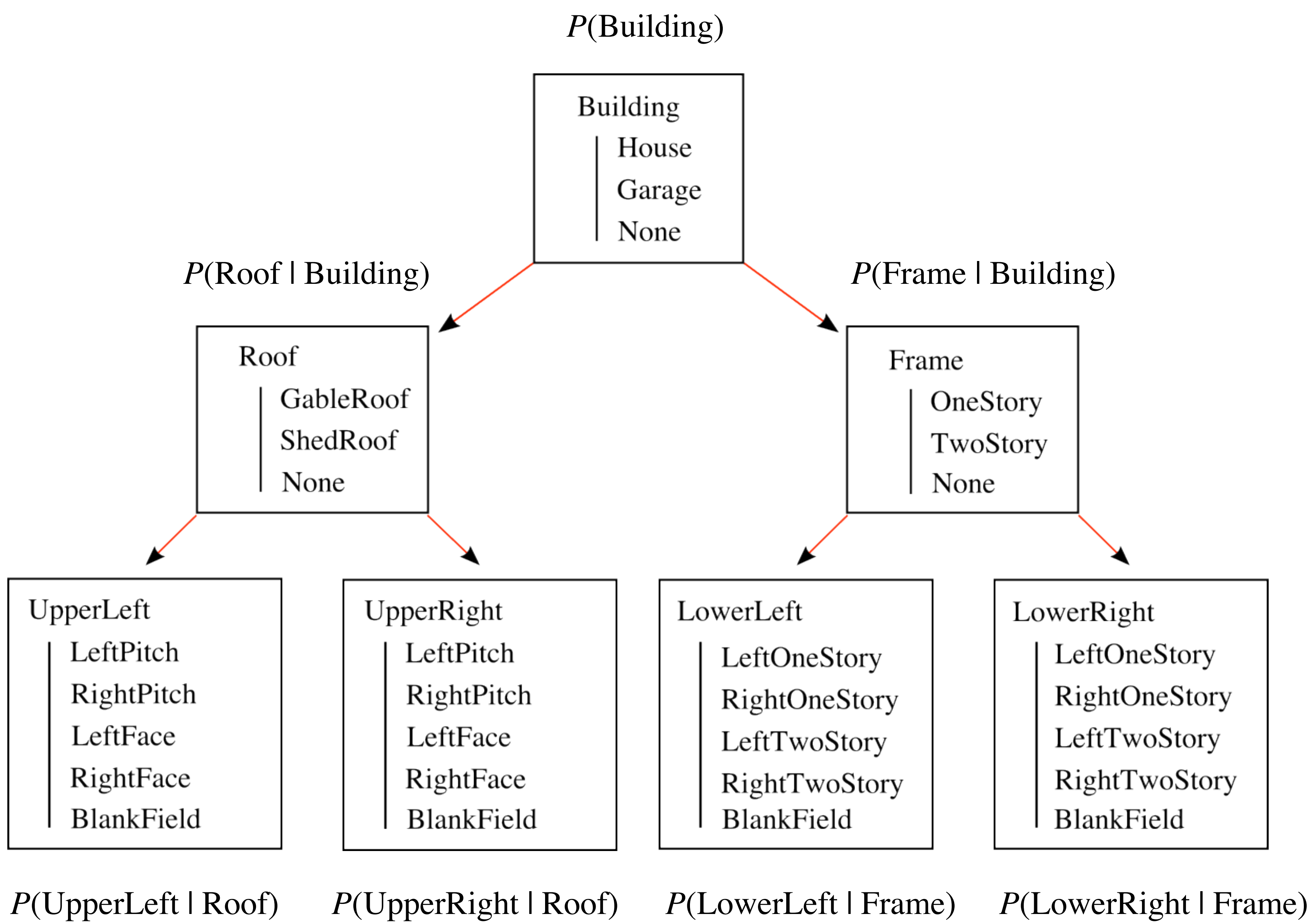


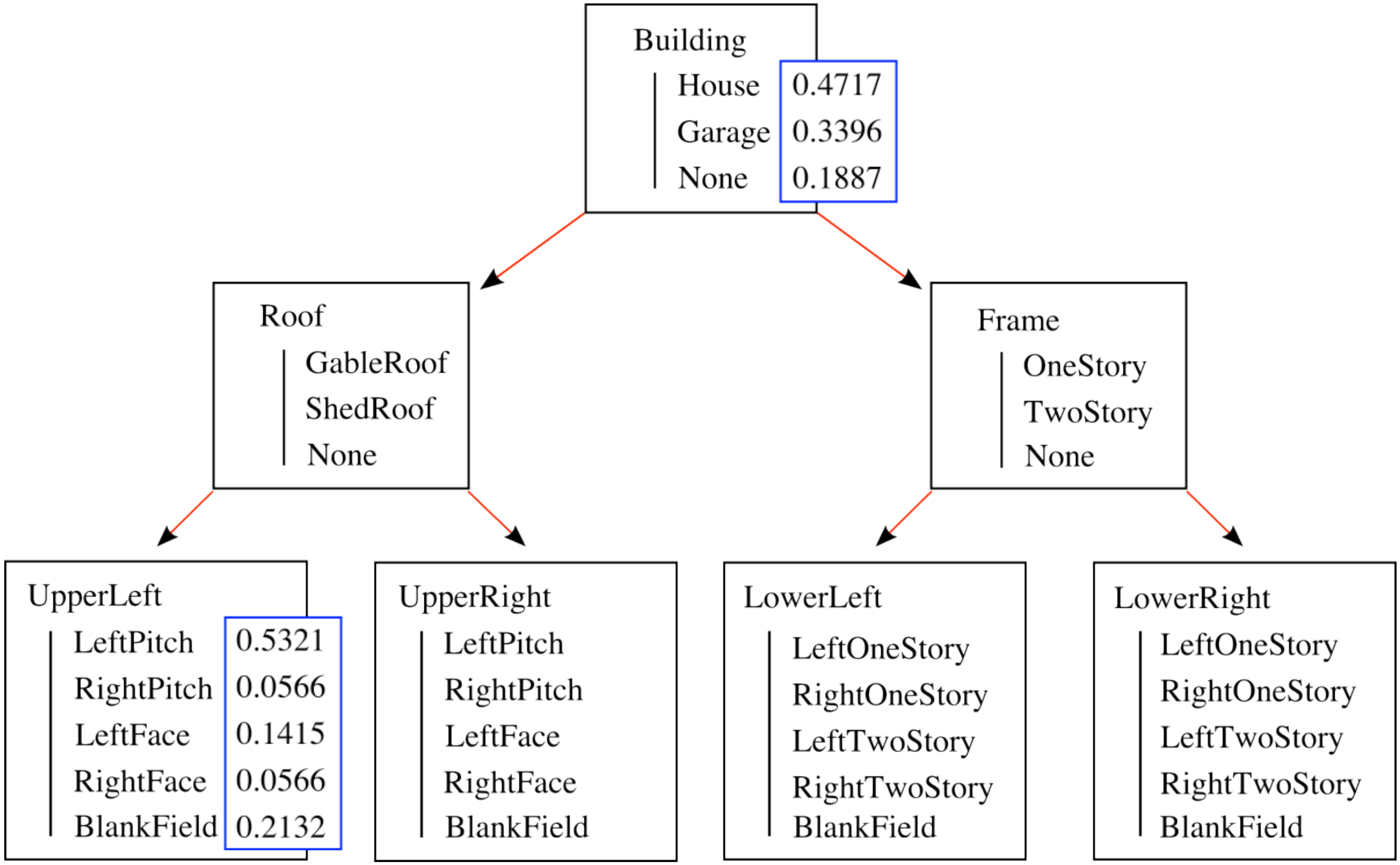
- Receptive fields and tuned filters

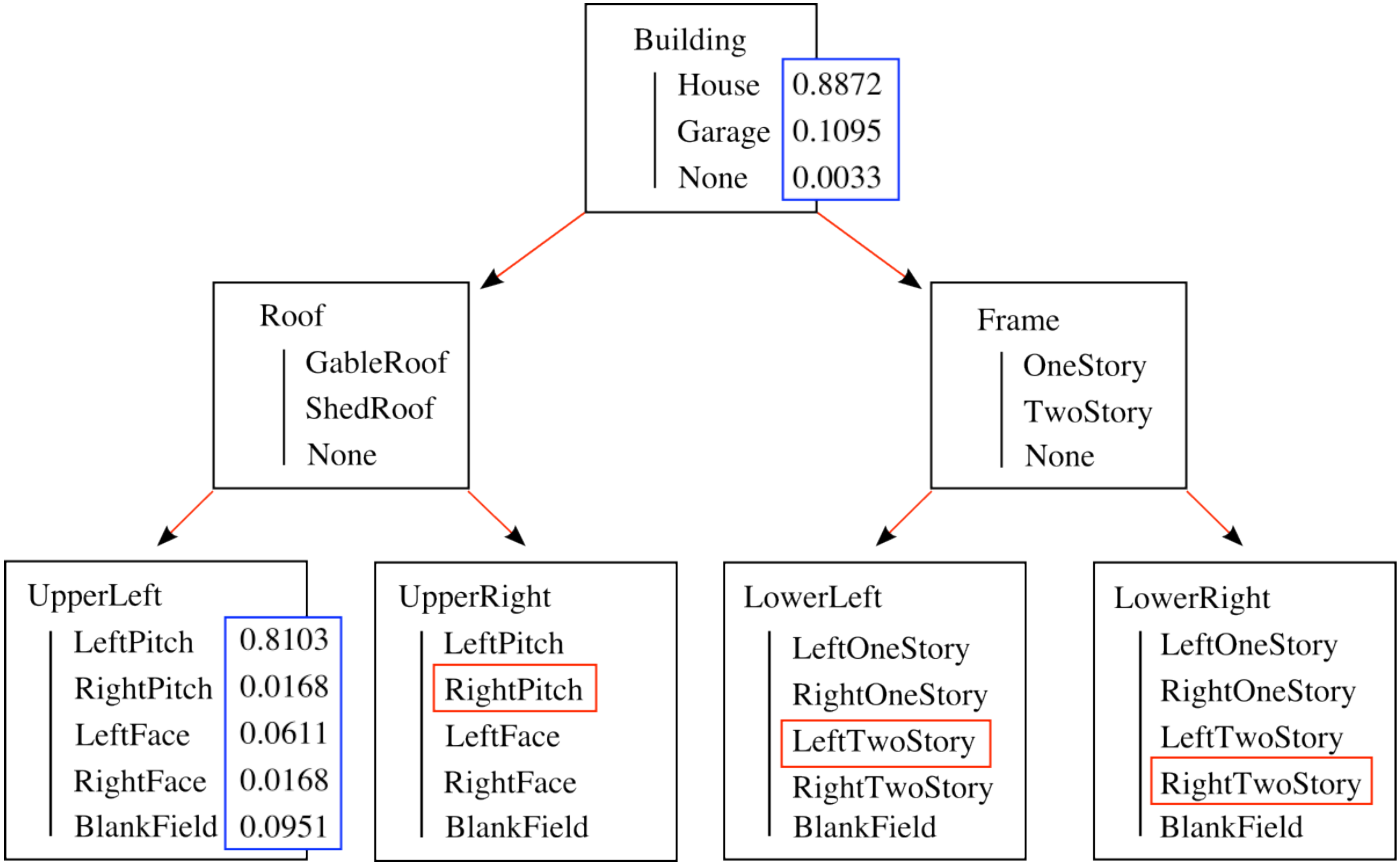


Hierarchy and Compositionality



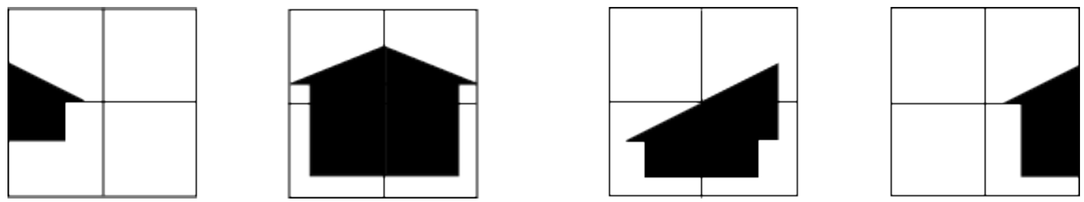

P(Building)
Building
House
Garage
None
P(Roof | Building)
P(Frame | Building)
Roof
GableRoof
ShedRoof
None
Frame
OneStory
TwoStory
None
UpperLeft
LeftPitch
RightPitch
LeftFace
RightFace
BlankField
UpperRight
LeftPitch
RightPitch
LeftFace
RightFace
BlankField
LowerLeft
LeftOneStory
RightOneStory
LeftTwoStory
RightTwoStory
BlankField
LowerRight
LeftOneStory
RightOneStory
LeftTwoStory
RightTwoStory
BlankField
P(UpperLeft | Roof)
P(UpperRight | Roof)
P(LowerLeft | Frame)
P(LowerRight | Frame)


Building
House 0.4717
Garage 0.3396
None 0.1887
Roof
GableRoof
ShedRoof
None
Frame
OneStory
TwoStory
None
UpperLeft
LeftPitch 0.5321
RightPitch 0.0566
LeftFace 0.1415
RightFace 0.0566
BlankField 0.2132
UpperRight
LeftPitch
RightPitch
LeftFace
RightFace
BlankField
LowerLeft
LeftOneStory
RightOneStory
LeftTwoStory
RightTwoStory
BlankField
LowerRight
LeftOneStory
RightOneStory
LeftTwoStory
RightTwoStory
BlankField

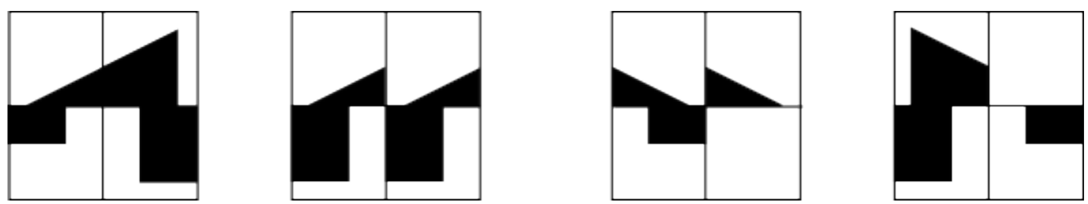

Building
House
Garage
None
0.8872
0.1095
0.0033
Roof
GableRoof
ShedRoof
None
Frame
OneStory
TwoStory
None
UpperLeft
LeftPitch
RightPitch
LeftFace
RightFace
BlankField
0.8103
0.0168
0.0611
0.0168
0.0951
UpperRight
LeftPitch
RightPitch
LeftFace
RightFace
BlankField
LowerLeft
LeftOneStory
RightOneStory
LeftTwoStory
RightTwoStory
BlankField
LowerRight
LeftOneStory
RightOneStory
LeftTwoStory
RightTwoStory
BlankField

Representational Challenges

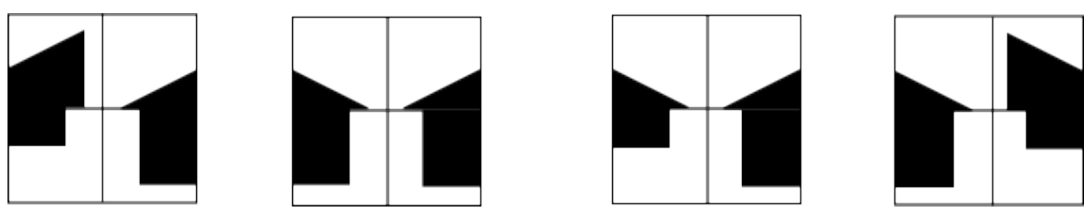
- Translation and scale invariance



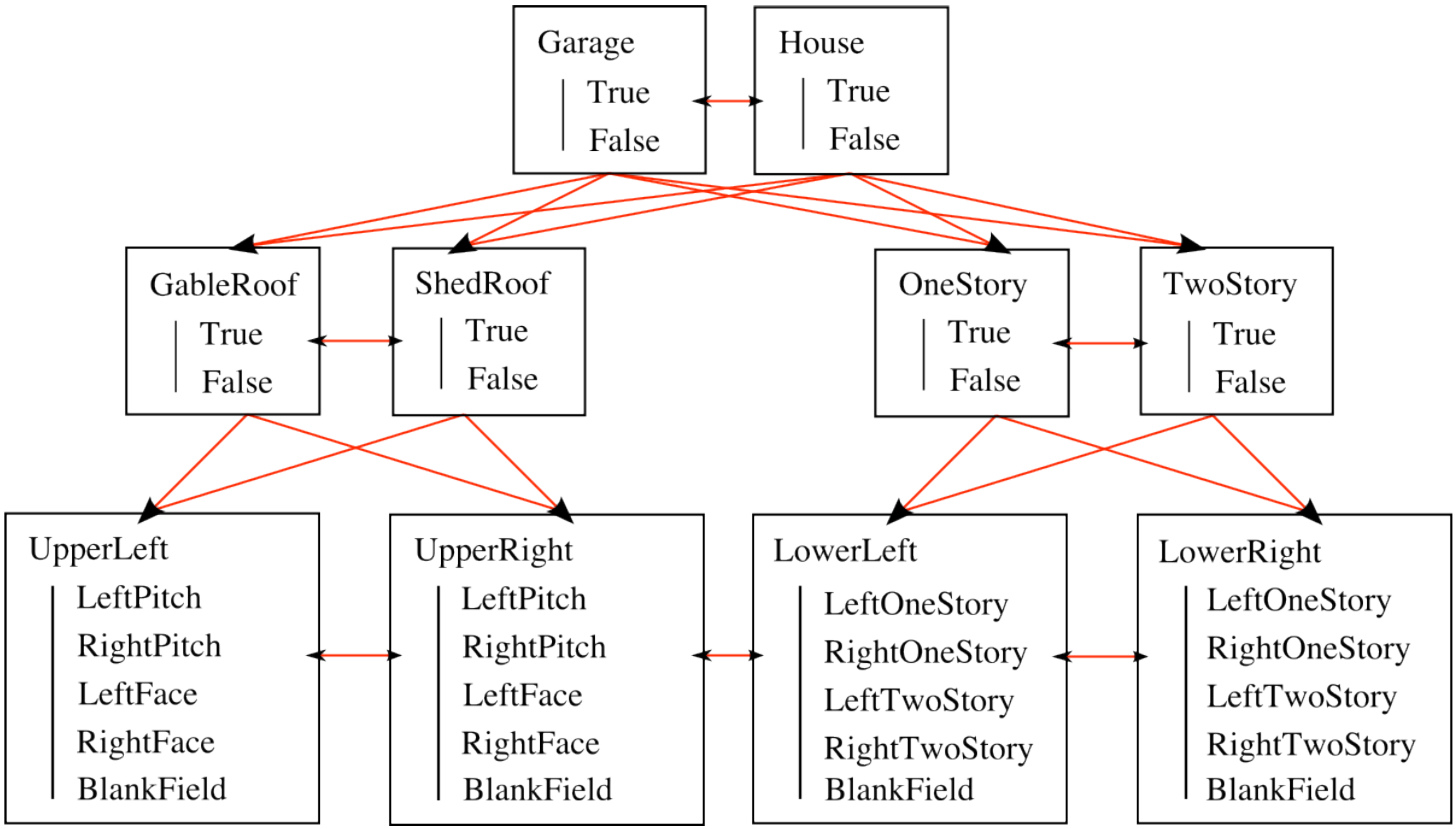
- Compositionality constraints



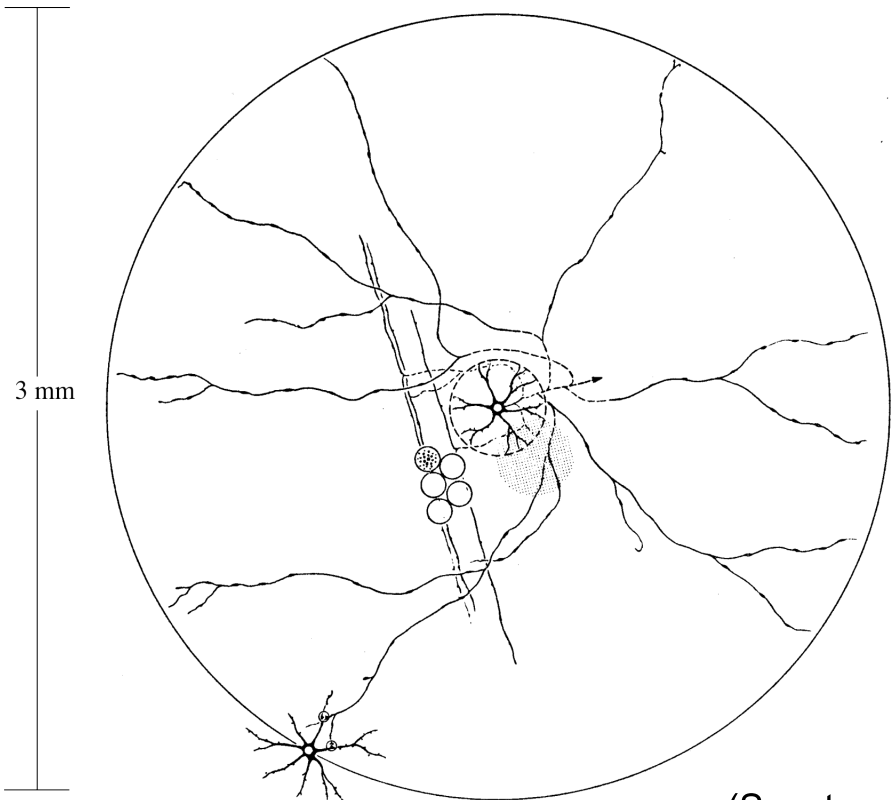
- Multiple instances of a concept



Dependent Random Variables



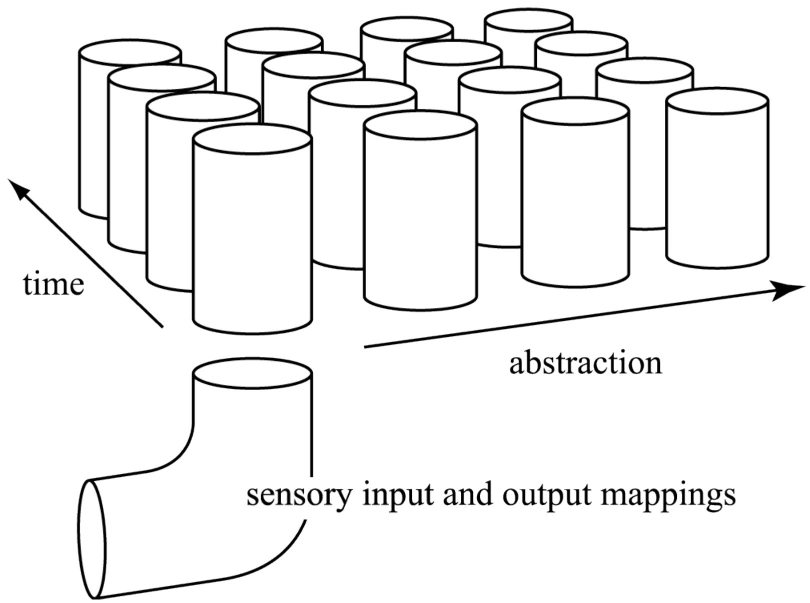
Cortical Connectivity



(Szentagothai, 1978)

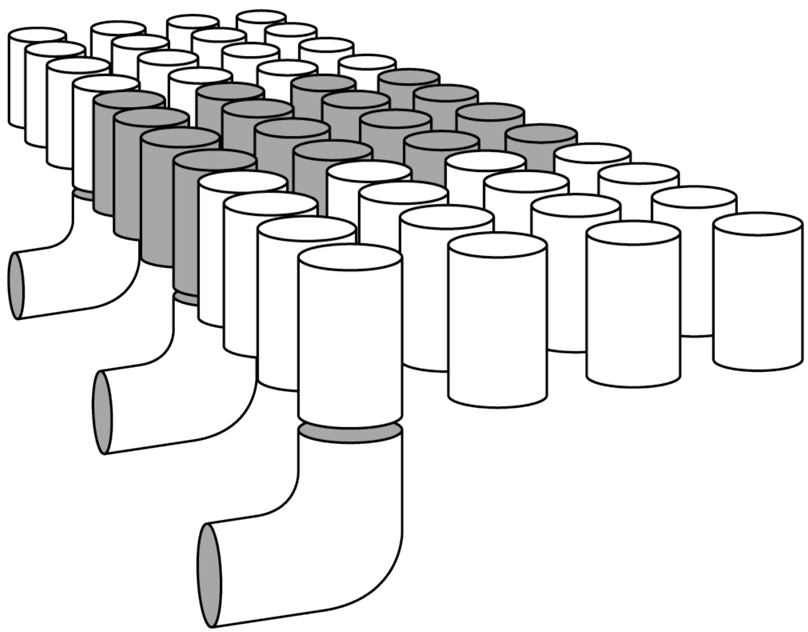
- 10^{15} connections
- 10^{11} neurons
- Small-world graph
- Small diameter

Space, Time and Abstraction



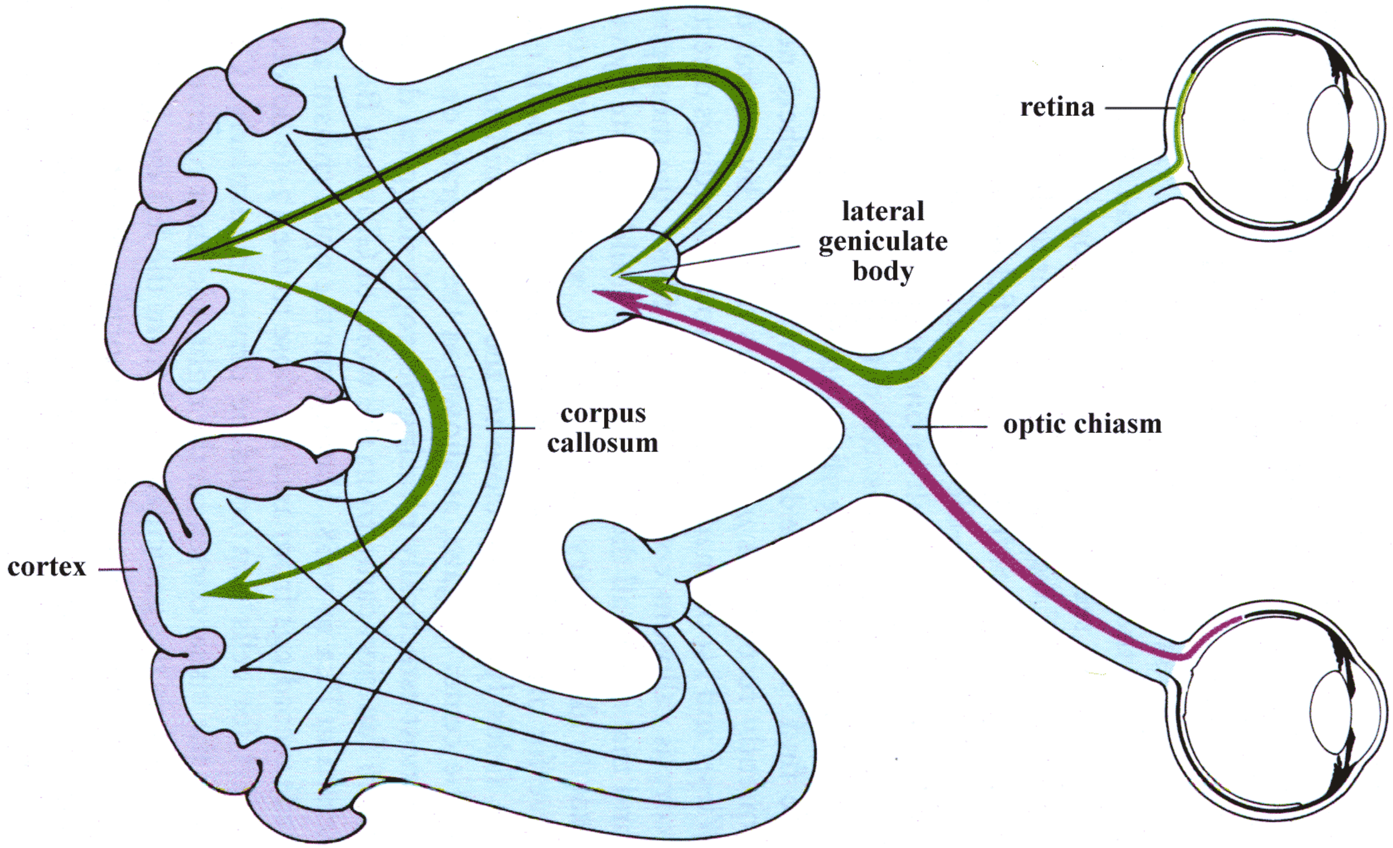
- Spatial relationships
- Temporal relationships
- Layers of abstraction

Multiple Modalities

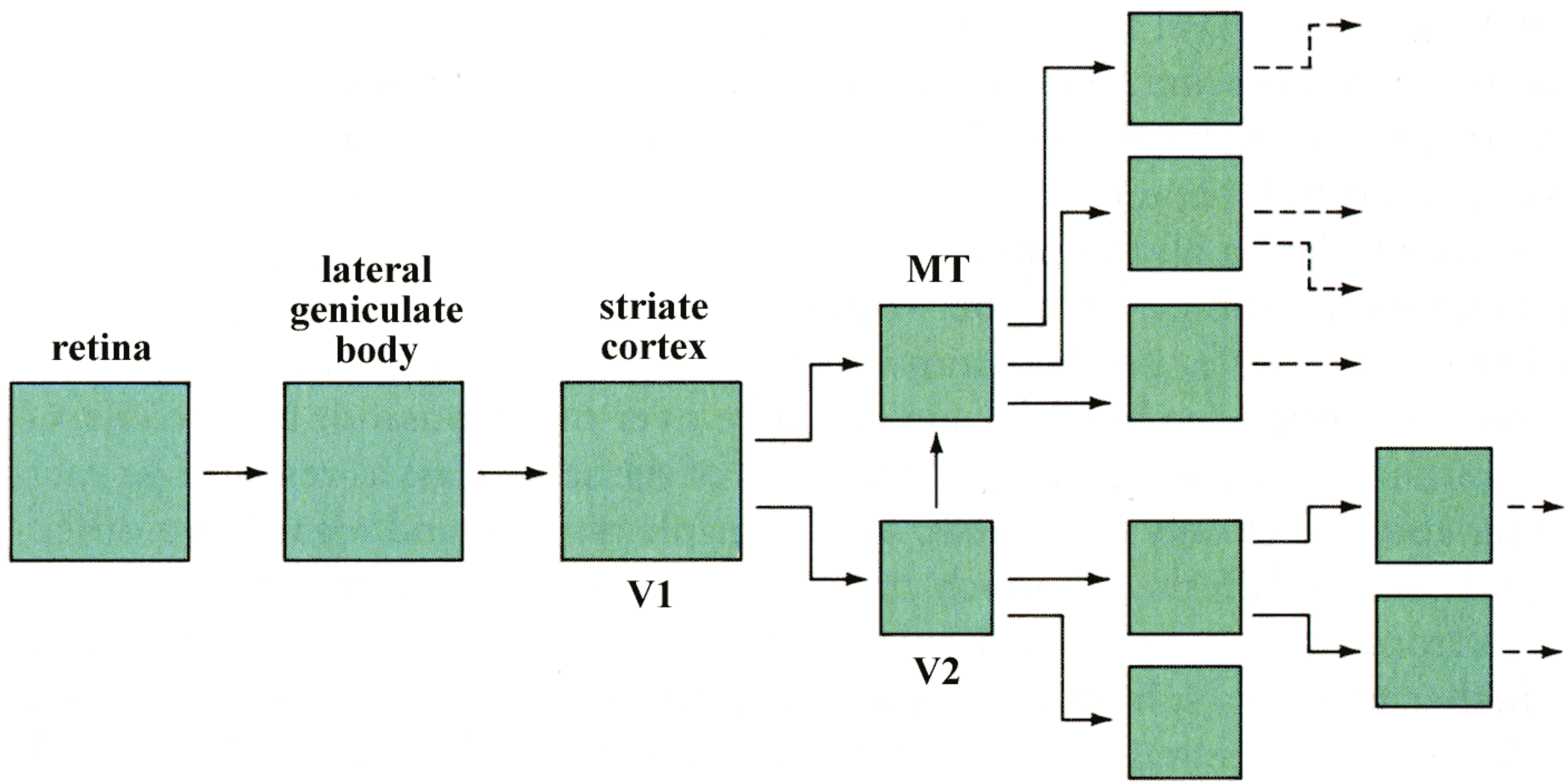


- Multiple resolutions
- Sensor integration
- Sensory correlation

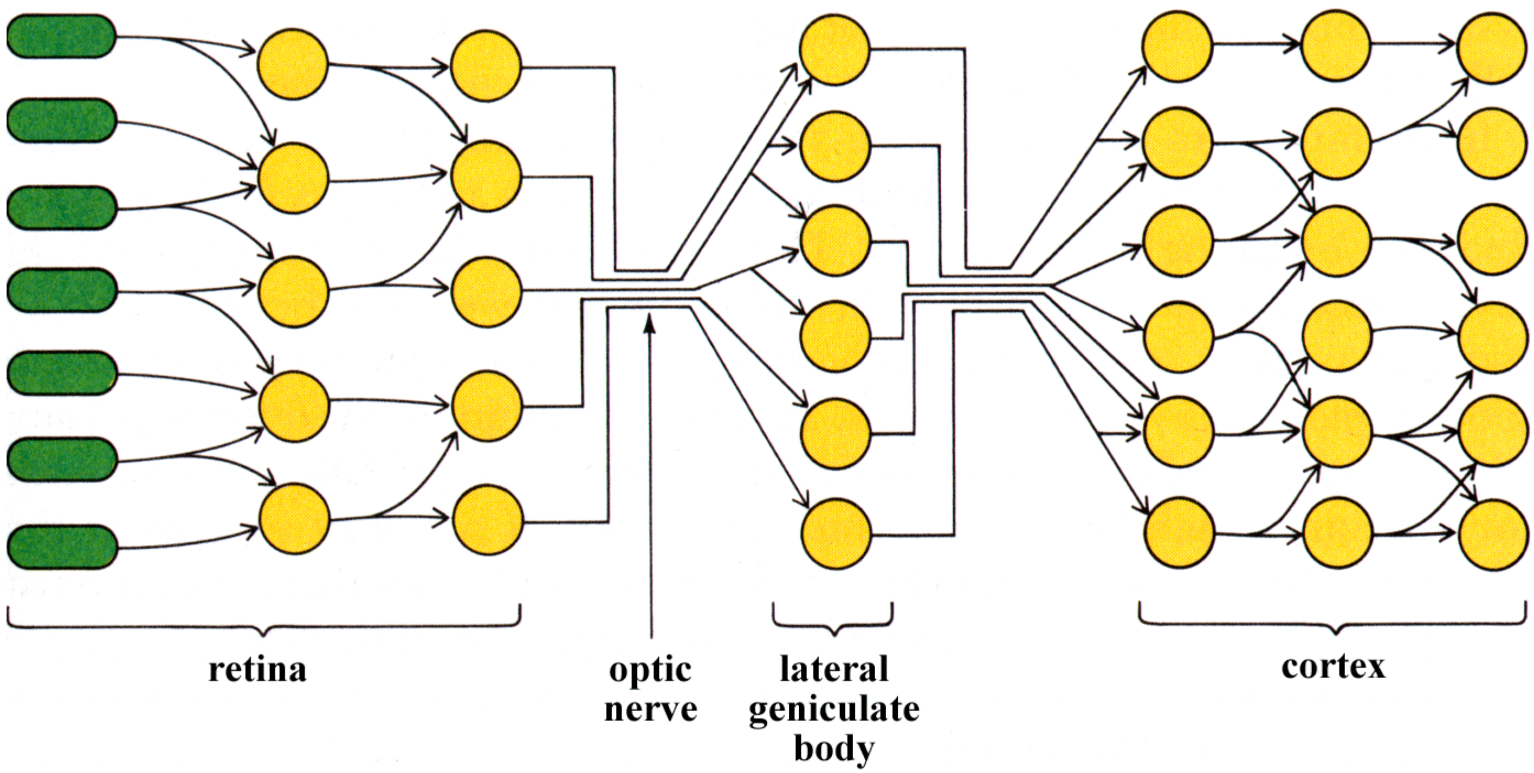
Primary Visual Pathways



Computational Modules

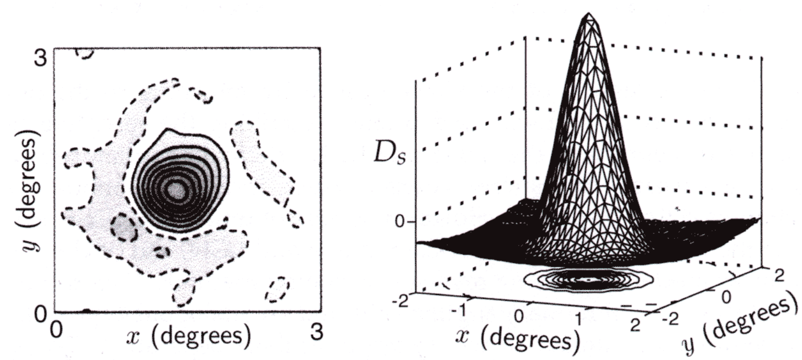


Neural Circuitry

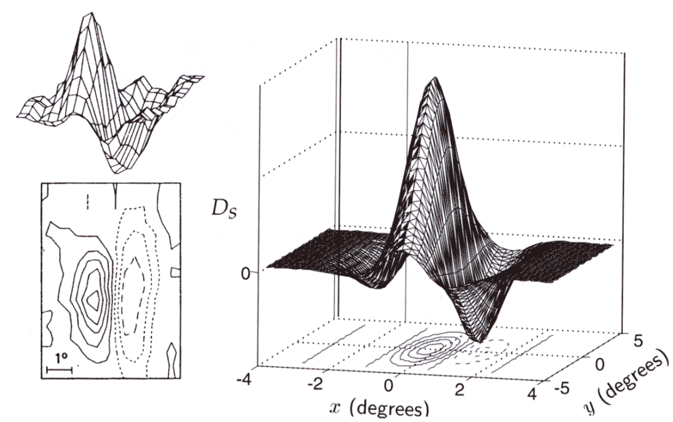


Simple Cells/Receptive Fields

- Center-surround cells in the retina and lateral geniculate nuclei — difference of Gaussians



- Edge-sensitive cells in V1 — product of Gaussian and sinusoidal functions, Gabor functions

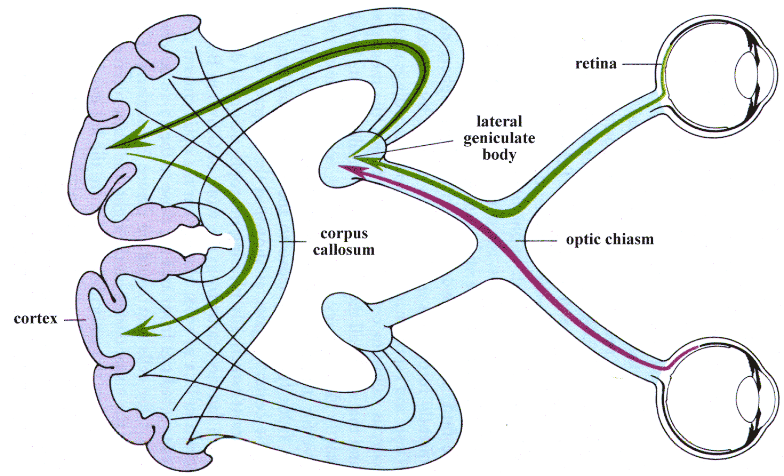


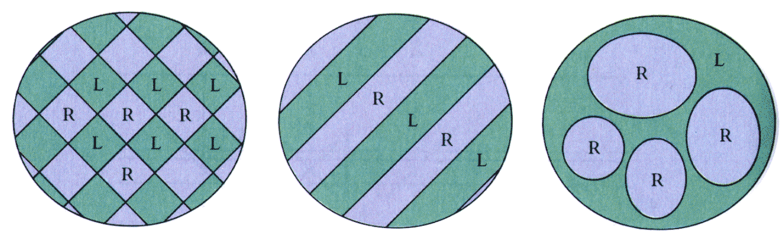
Simple versus Complex Cells

- Simple Cells
  - highly *selective* in their responses
  - partitioned into excitatory and inhibitory regions
- Complex Cells
  - able to implement *invariant* features
  - represent ~3/4 of the cells in the striate cortex
- Both Types
  - can respond to different orientations
  - can respond to spatio-temporal stimuli

Primary Visual Cortex

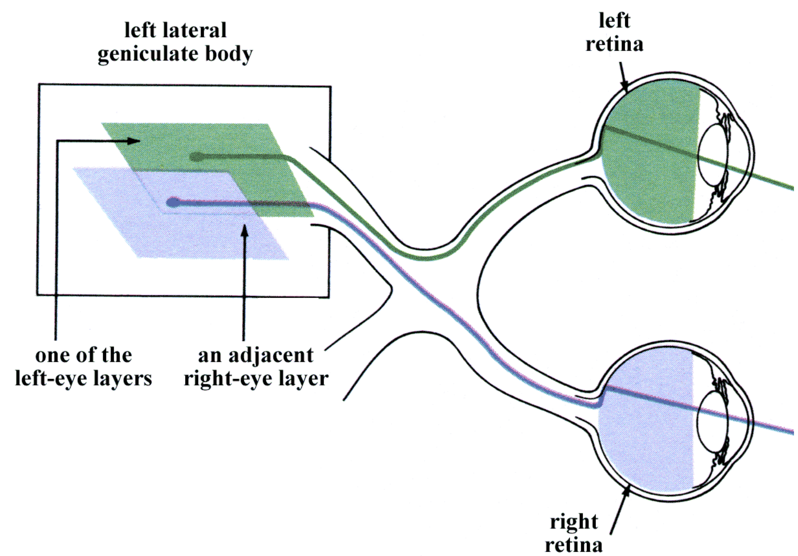
- Left and Right Stimuli
- Orientation Features
- How is it organized?



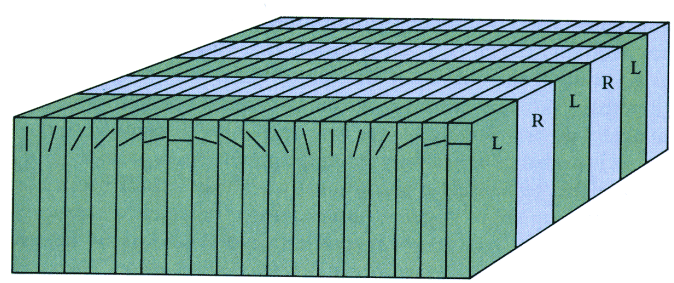


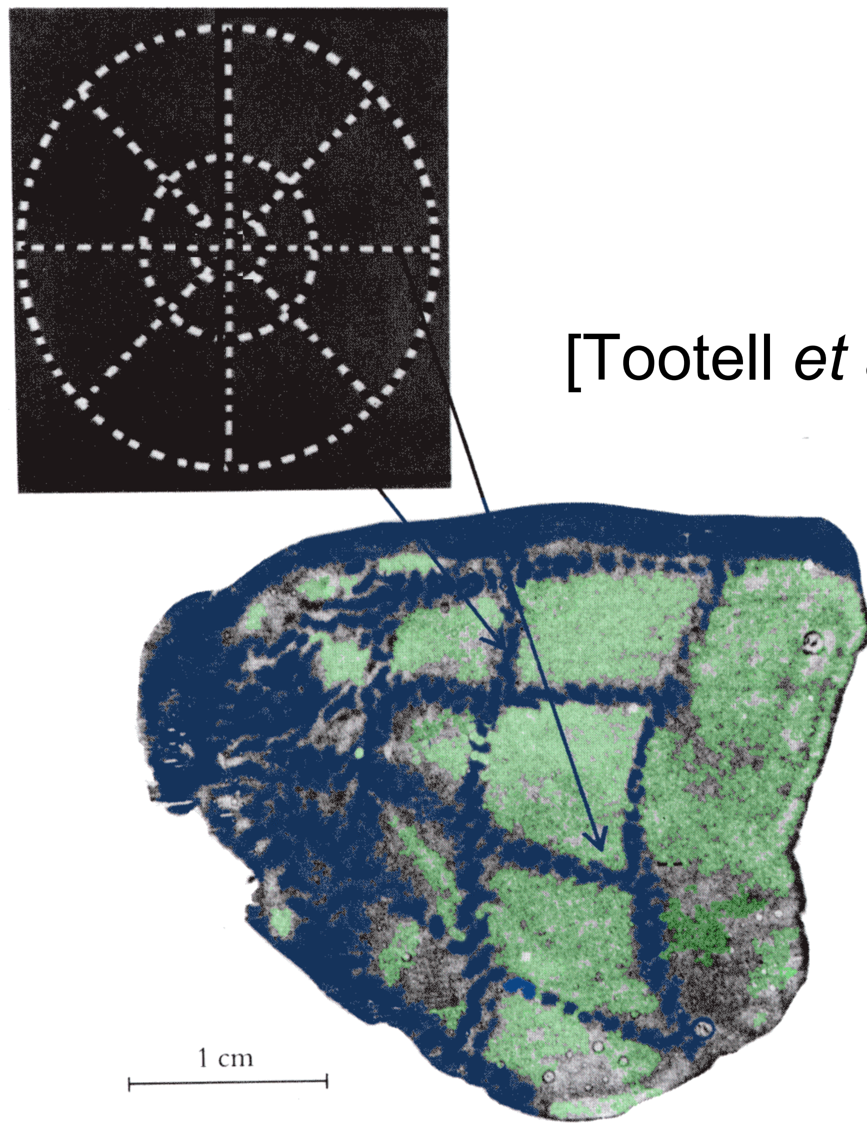
Organization of V1(area 17)

- Interleave left and right visual slices



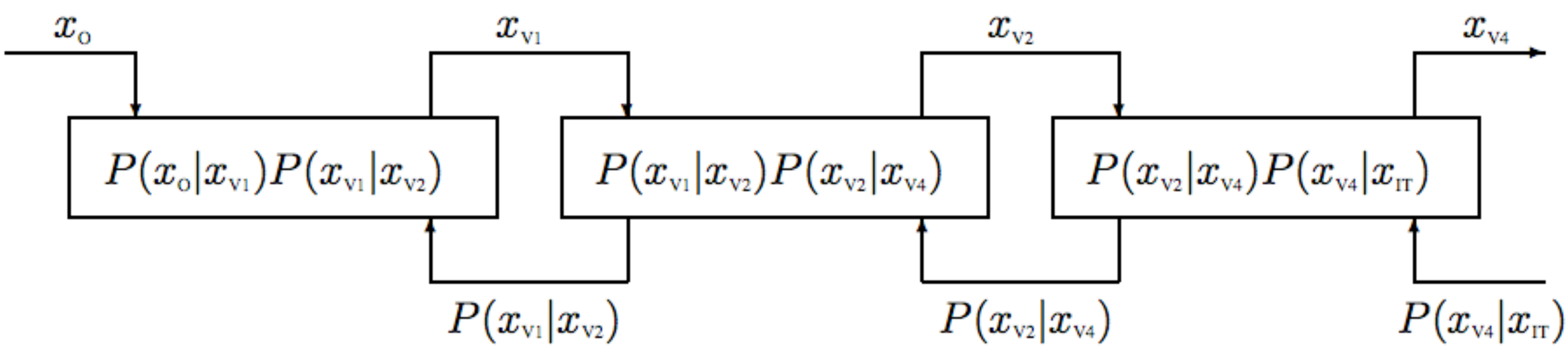
- String together the orientations from 0° to 180° in about 10° increments





[Tootell *et al.*, 1982]

Learning and Inference

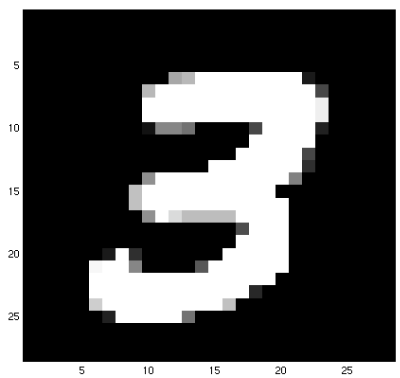


Learning

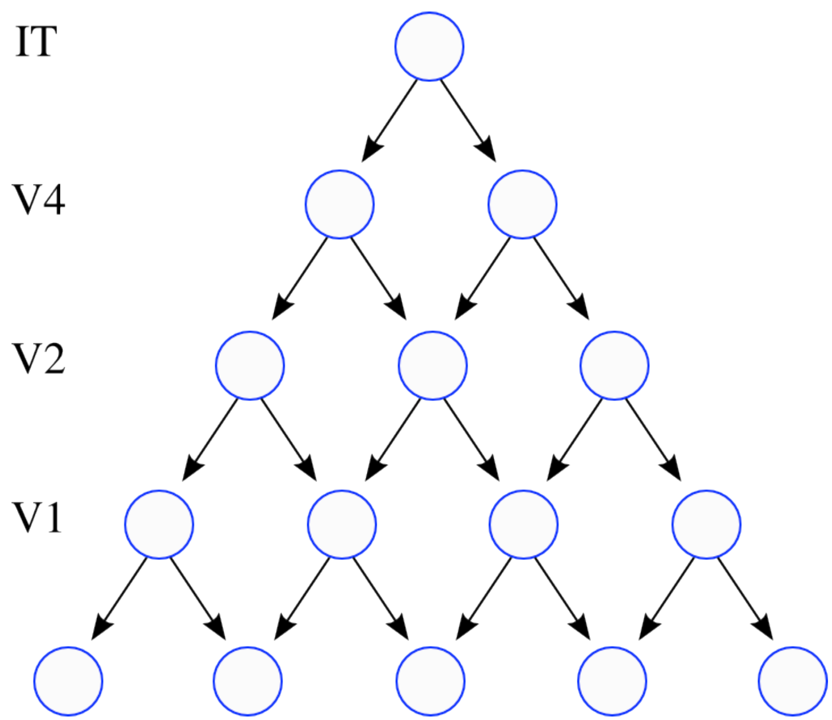
- Learning graphical models from data
  - Both structure and parameter learning (Jordan, 1998)
- Learning hierarchical invariant feature networks
  - Neocognitron (Fukushima *et al.,* 1983)
  - MAX operations (Riesenhuber and Poggio, 1999)
  - Slow feature analysis (Wiskott and Sejnowski, 2002)
  - Multiple-cause vector quantification (Ross and Zemel, 2003)
  - Hierarchical Bayesian networks (George and Hawkins, 2005)
  - Learning deep belief nets (Hinton, Osindero and Teh, 2005)

Learning (continued)

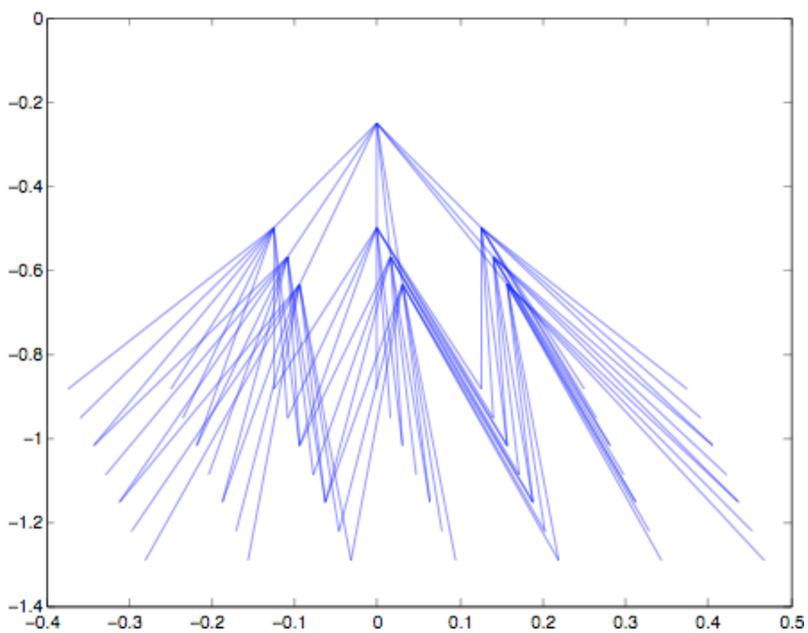
- Learning to recognize hand-written digits

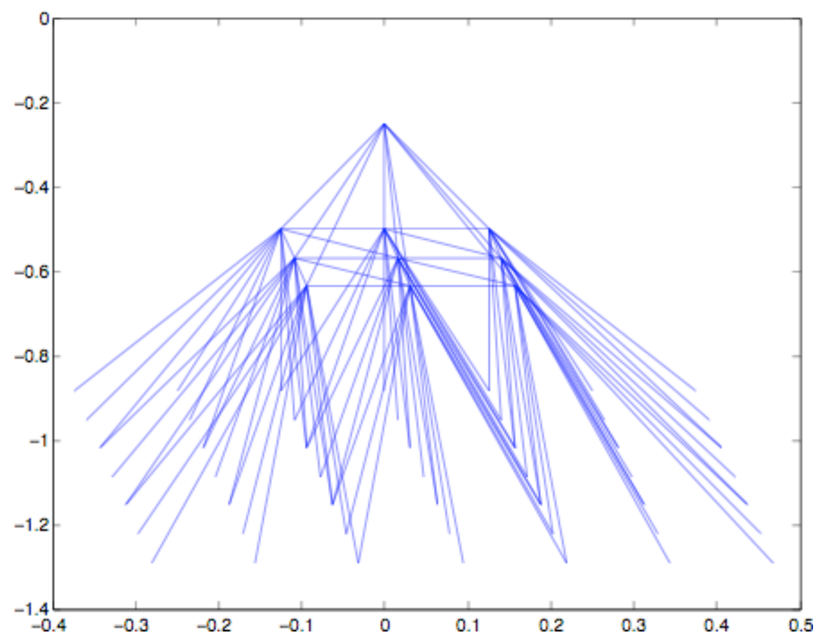


- Specify hierarchical network structure



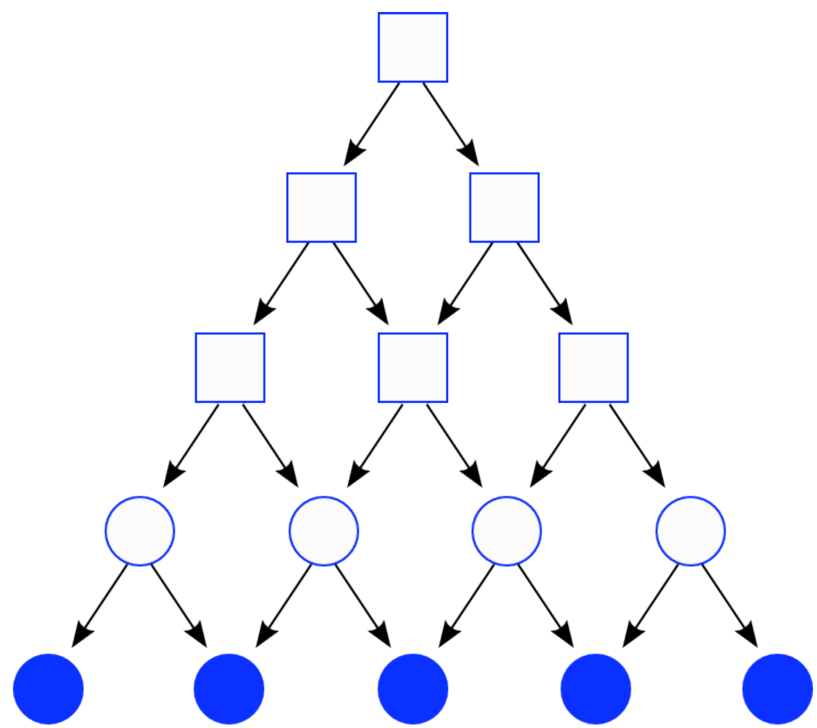
Pyramid-graph Bayes Networks



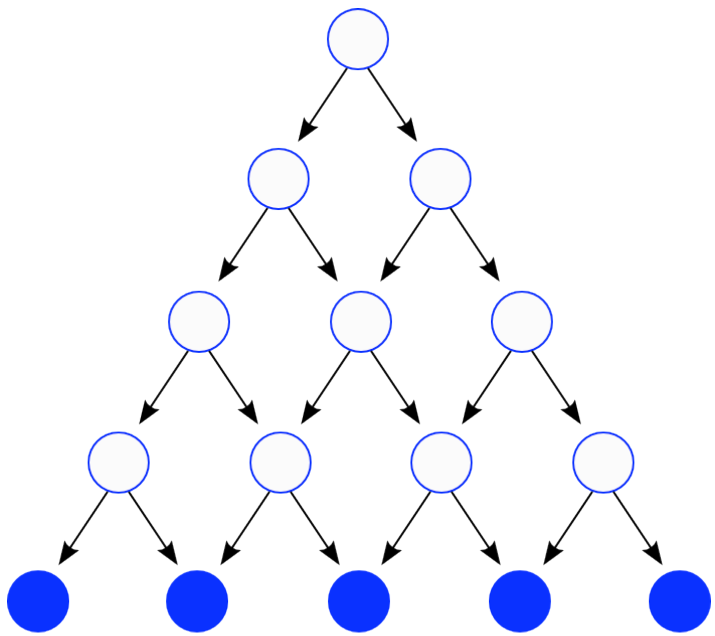


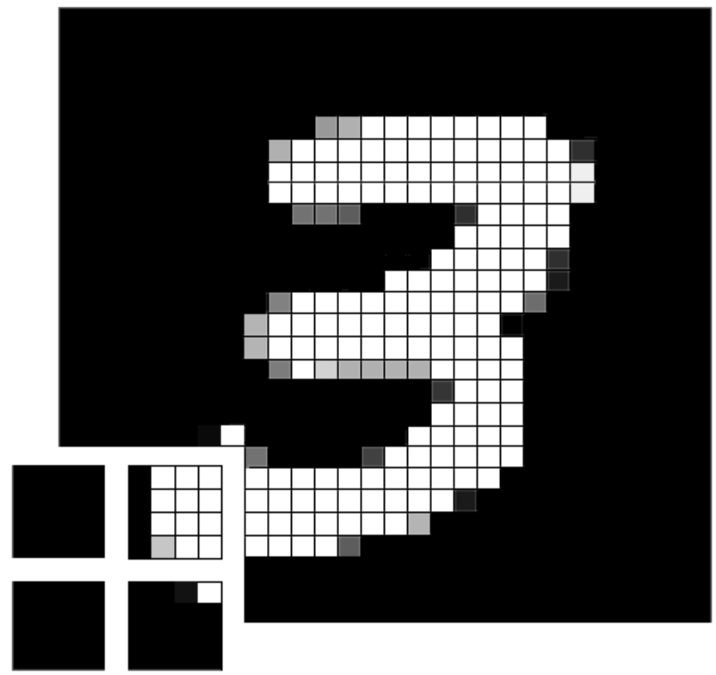
Learning (continued)

- Learn layer-by-layer from the bottom-up



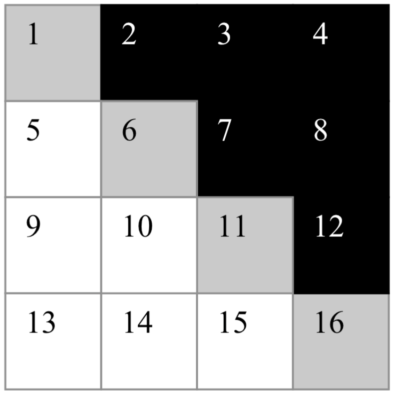
Input Layer of Simple Cells



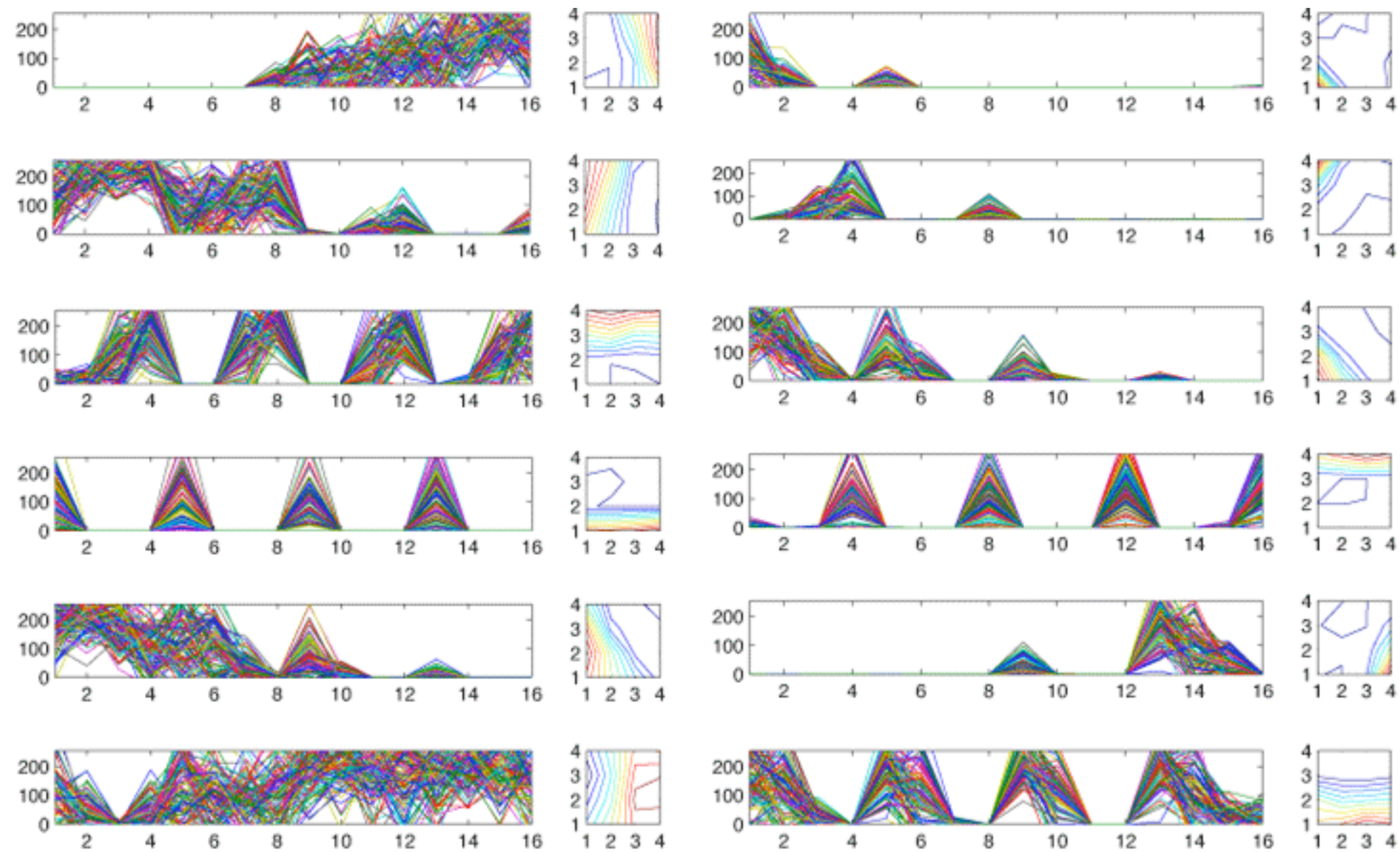


- Mixtures of Gaussians

$$p(\mathbf{x} \mid \theta) = \sum_k \alpha_k \mathrm{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$$

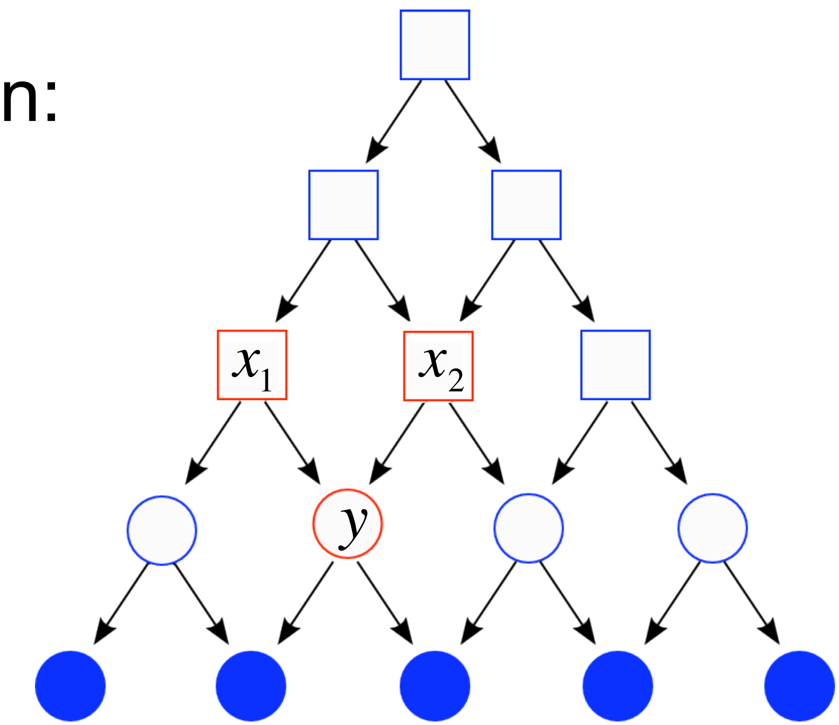


Mixture of Gaussians



Learning (continued)

- For each node y learn: $P(y \mid \text{Parents}(y))$



Complex Cells

Tabular nodes are profligate in numbers of parameters

$$P(x \mid \mathrm{Parents}(x))$$

Naïve Bayes and noisy OR nodes are more sparing

$$P(\mathbf{x} \mid y) = \prod_i P(x_i \mid y)$$

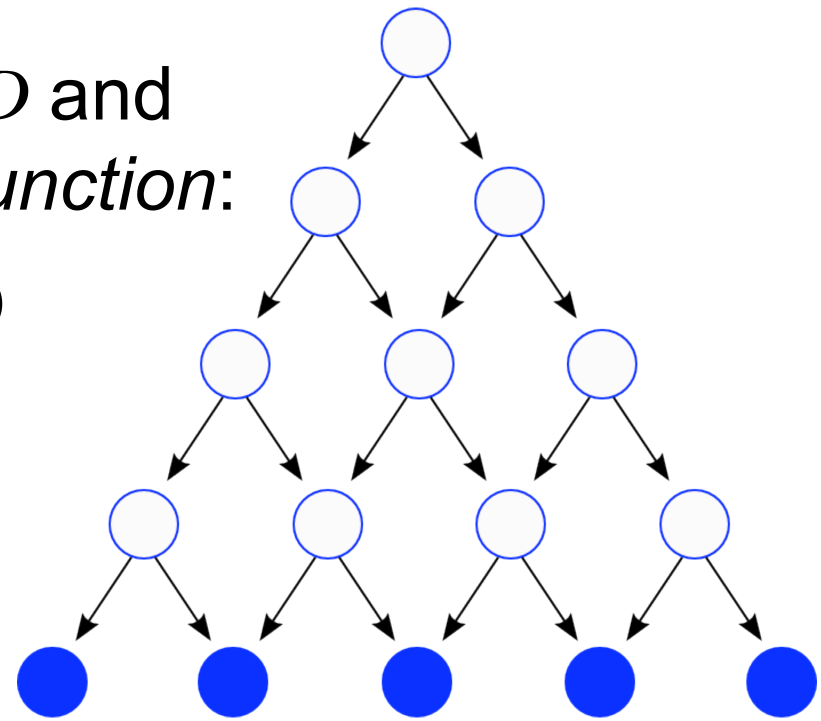
$$P(x = 0 \mid \mathbf{y}, \alpha) = \exp\left\{ \sum_i y_i \log \alpha_i \right\}$$

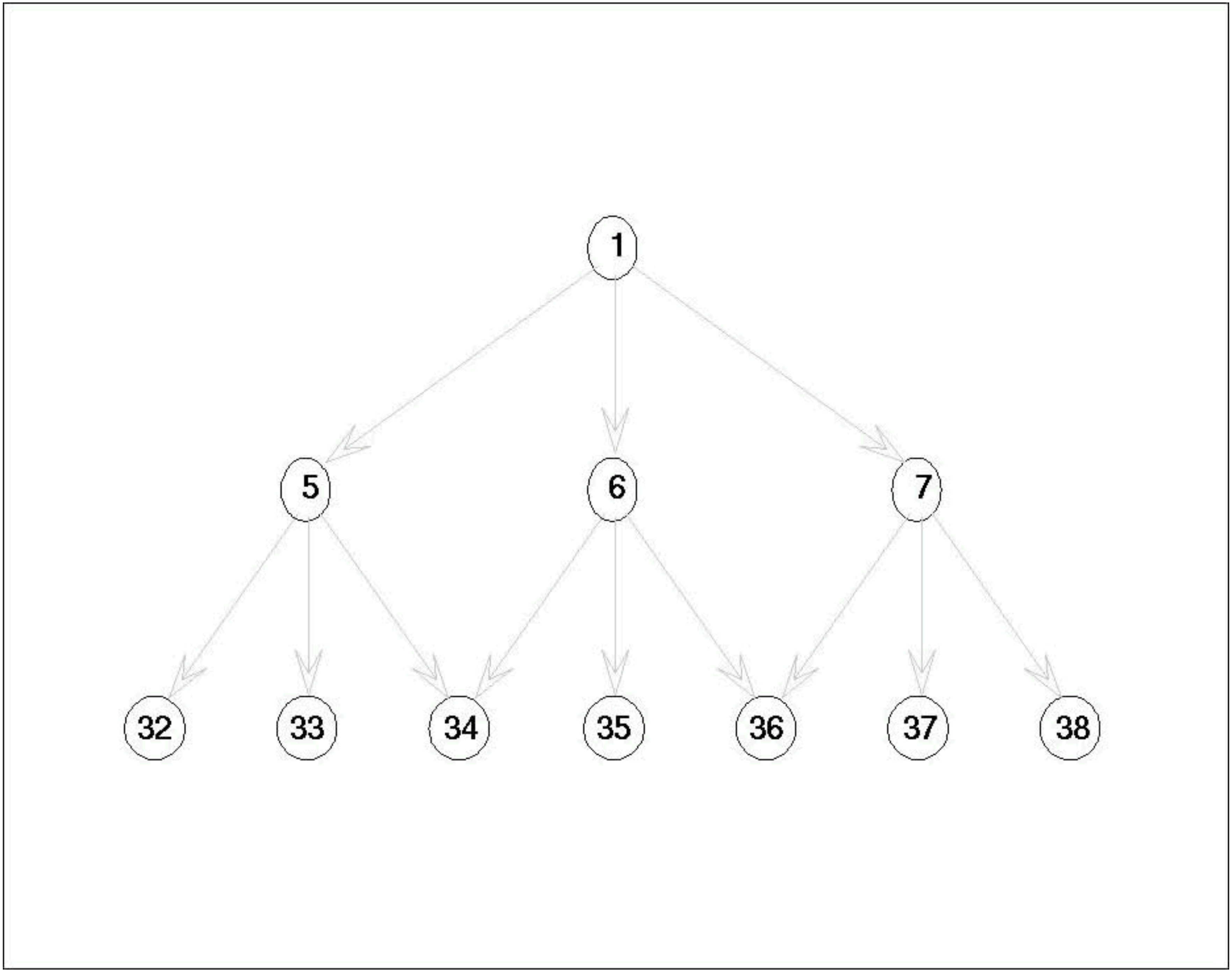
All three can be used to implement invariant features

Learning (continued)

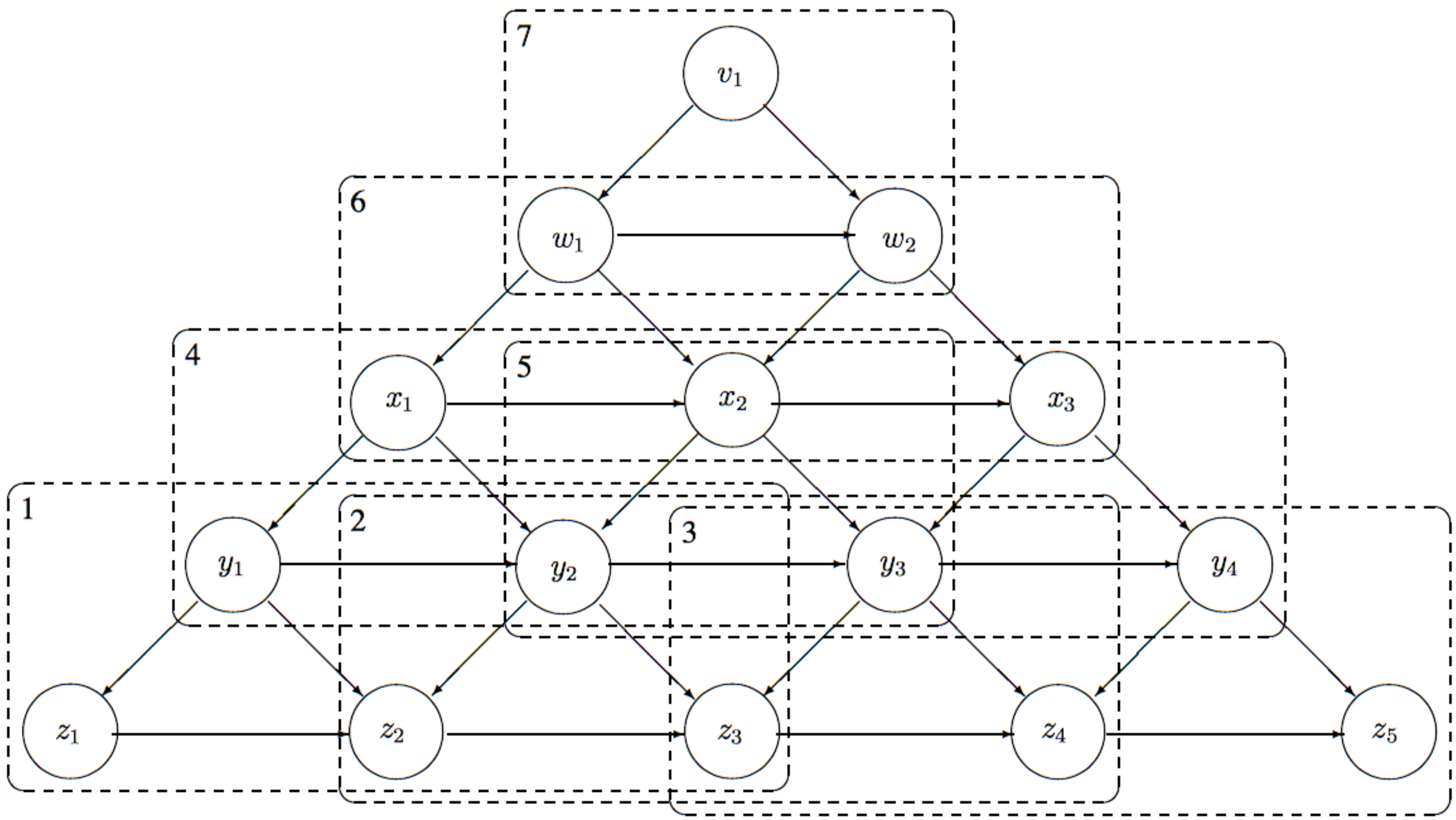
- Observe input data D and compute the *belief function*:

$$\forall i,\ \mathrm{Bel}(x_i) = P(x_i \mid D)$$

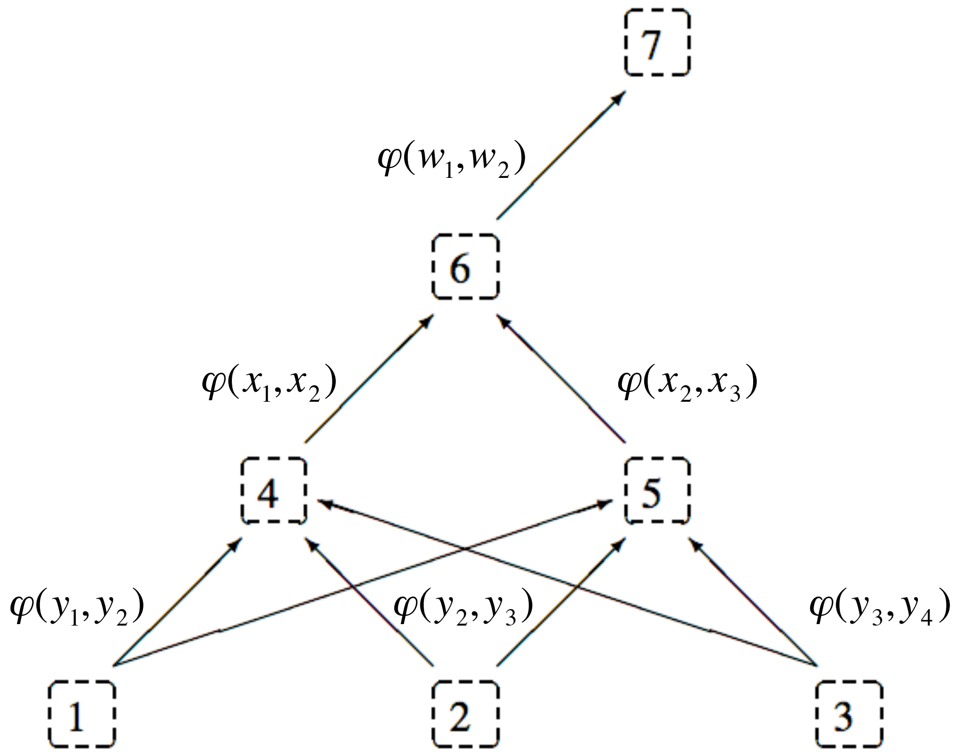



1
5
6
7
32
33
34
35
36
37
38

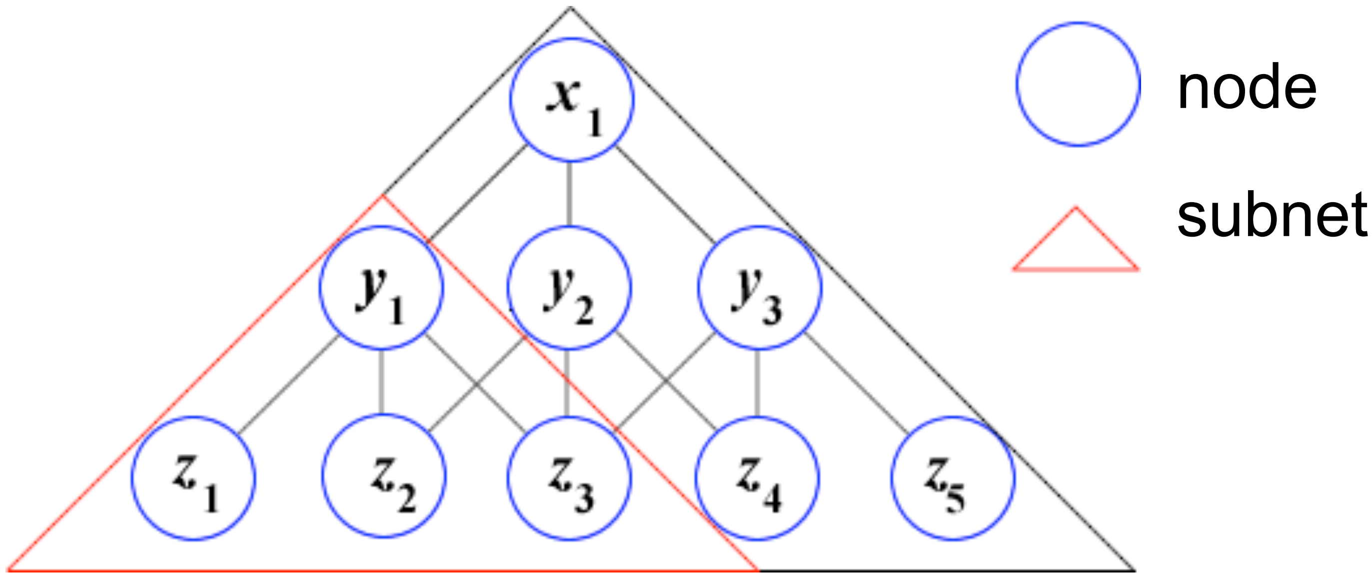
Subnet Decomposition



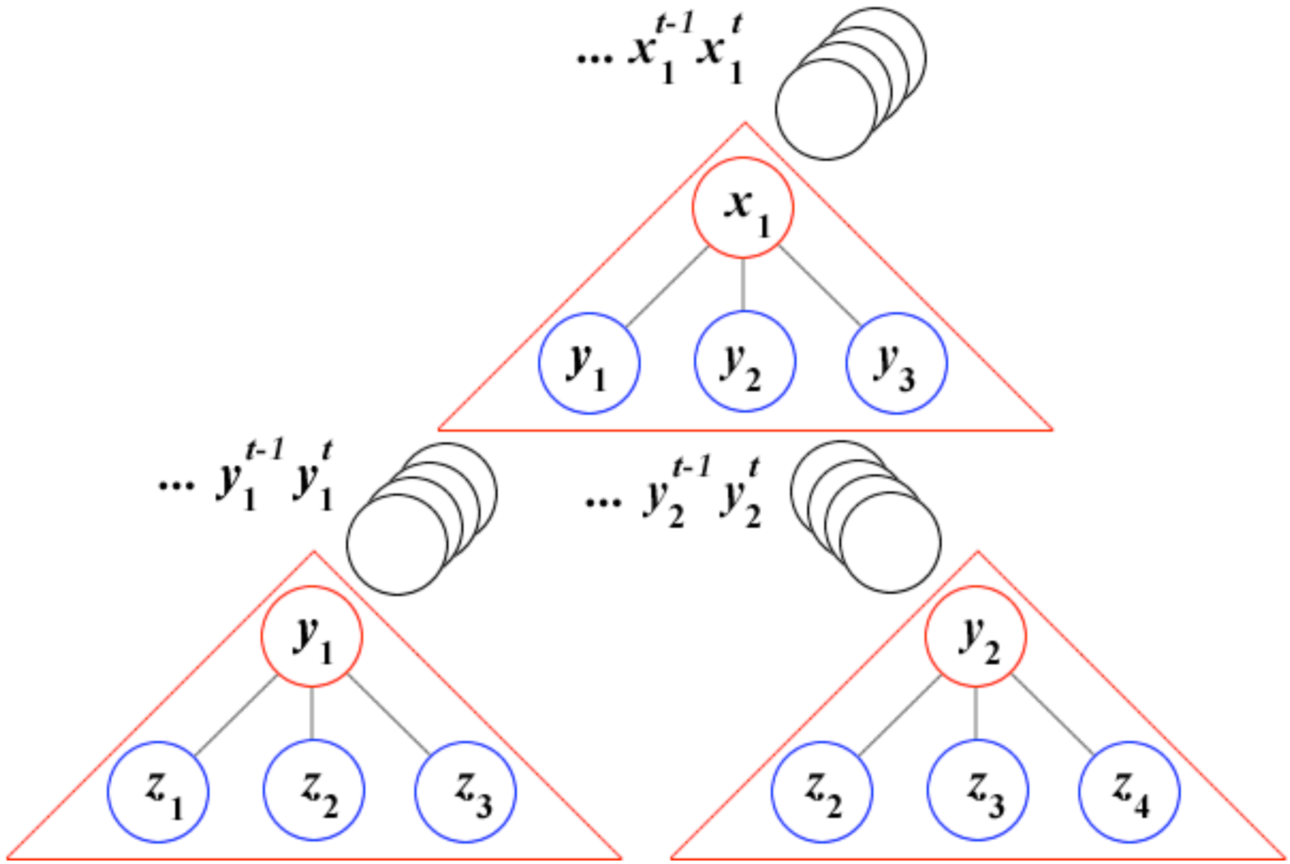
Propagation in Subnet Graphs



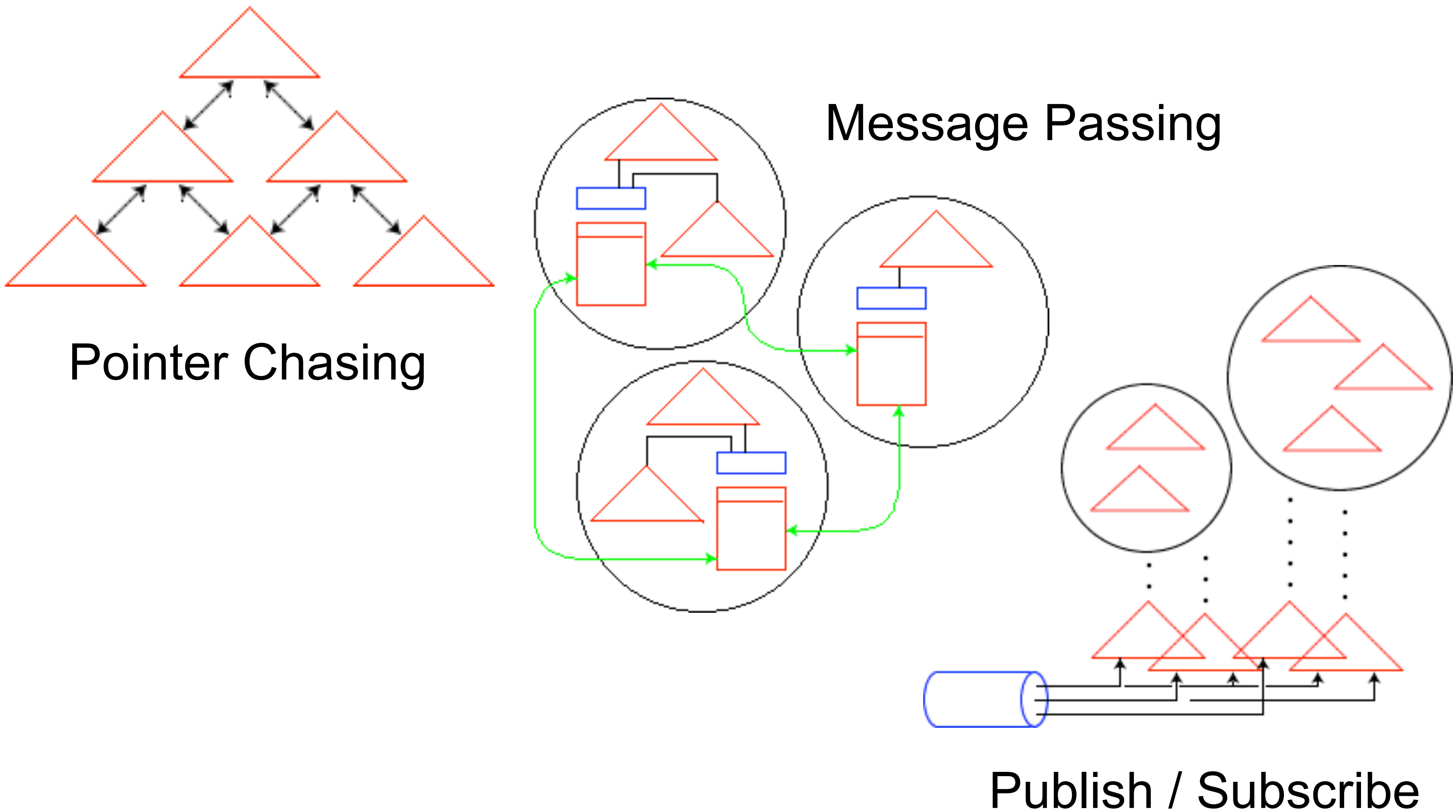
Pyramid-graph Fragment



Subnet Sample Propagation

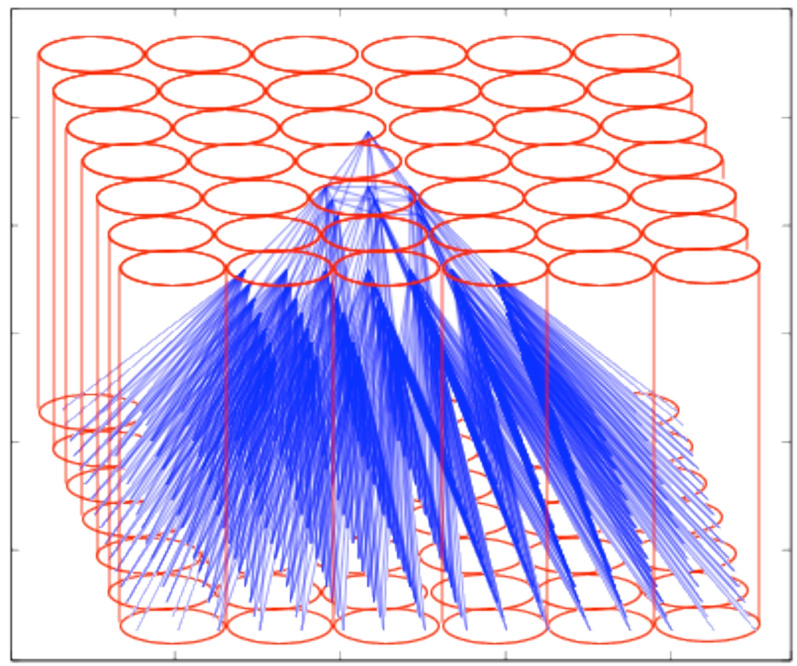


Serial and Parallel Implementation

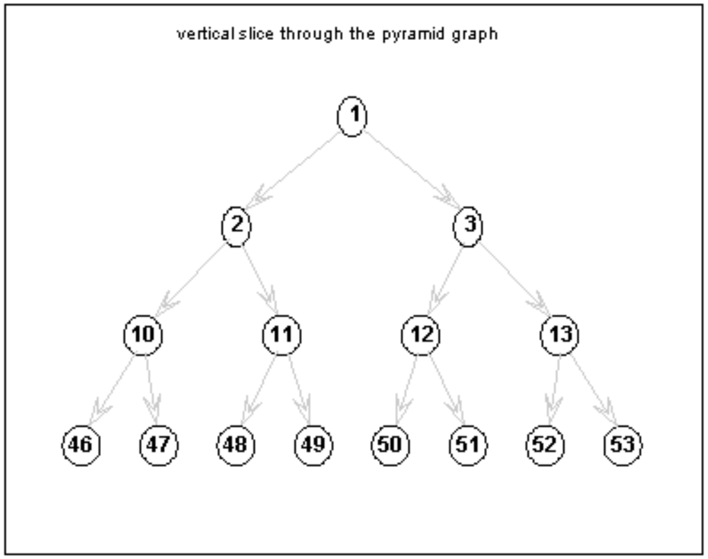


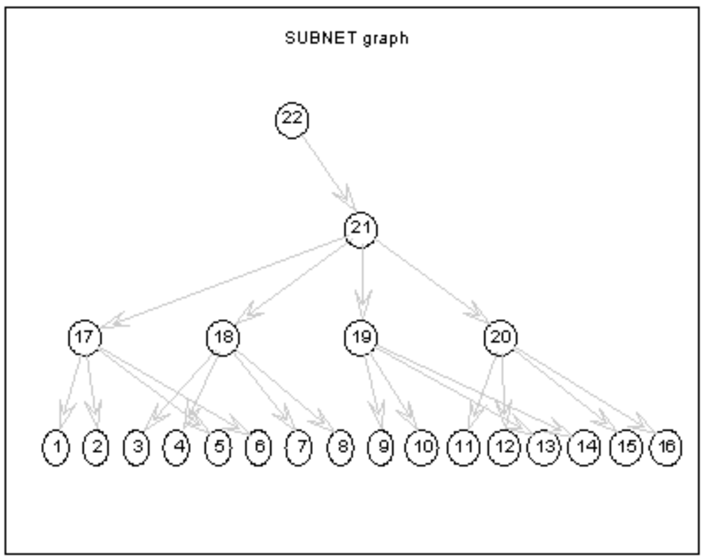
Simple Columnar Architecture

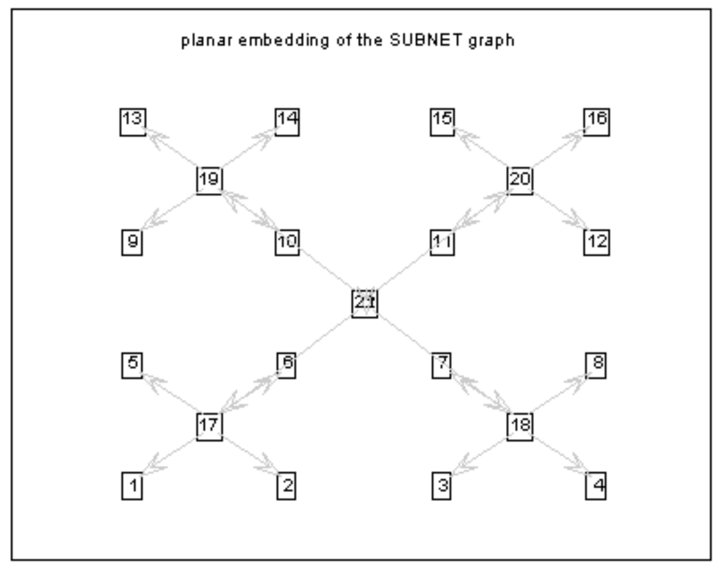
- One processor per column of nodes
- One process per subnet structure
- Performs variants of generalized belief propagation (GBP) [Yedidia, Freeman and Weiss, 2002]



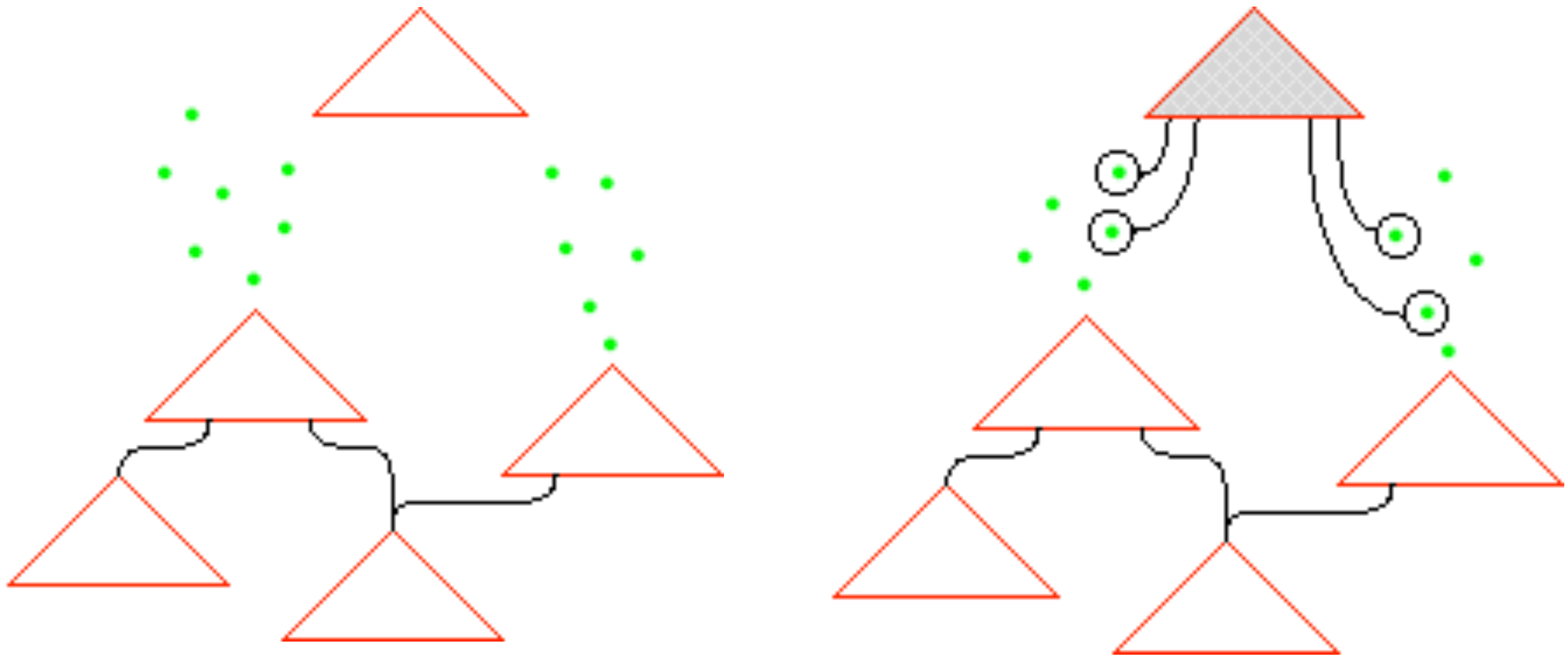
Message Passing Interface





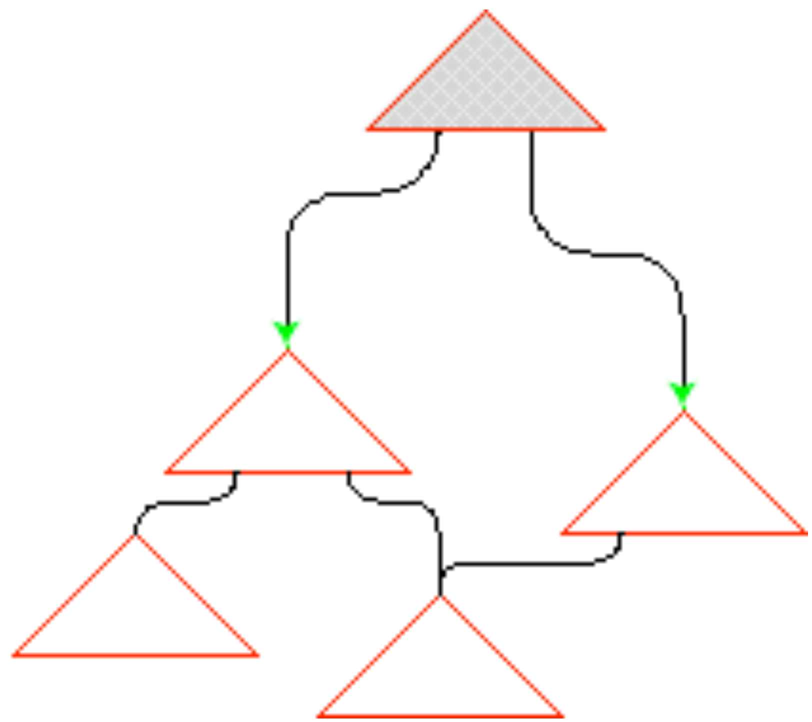


Publish / Subscribe



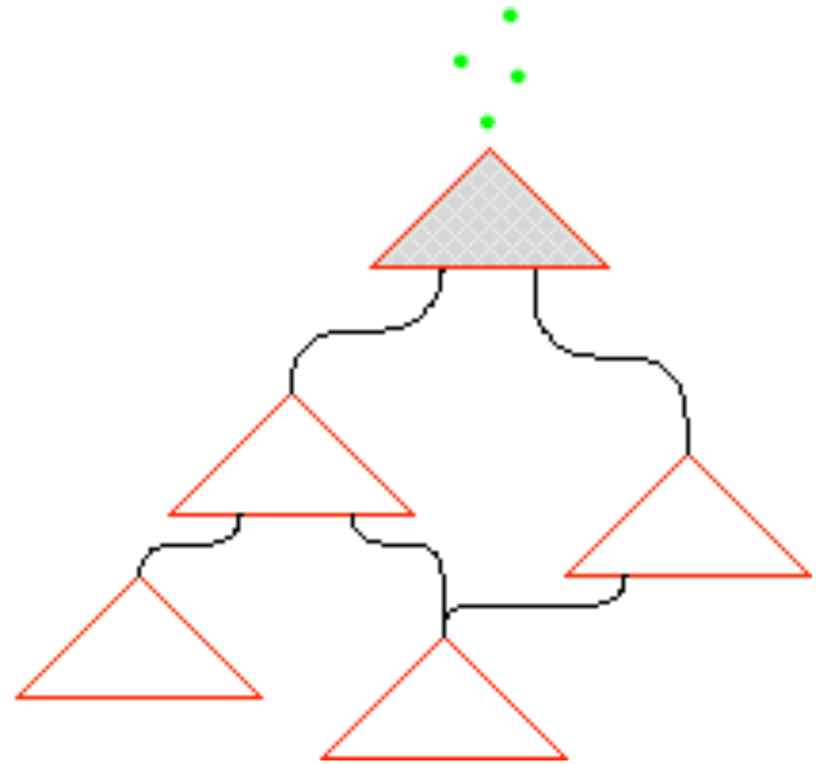
Unsubscribed & Unpublished

- Collect broadcasts
- Identify temporally & spatially proximate set of data publishers
- Bid on subscriptions
- Enter into contracts



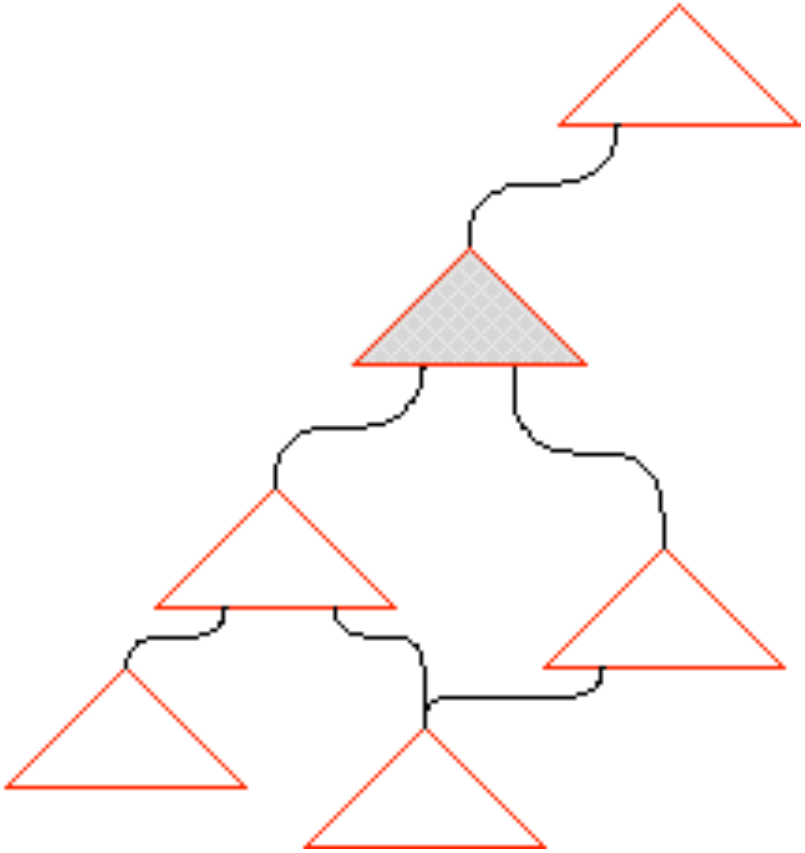
Unpublished

- Acquire a sufficiently large input sample
- Estimate parameters
- Broadcast data feeds for subscription



Trained & Published

- Accept bids for subscription services
- Enter into contracts
- Monitor publisher & subscriber signatures



MapReduce Implementation

Terminal — emacs

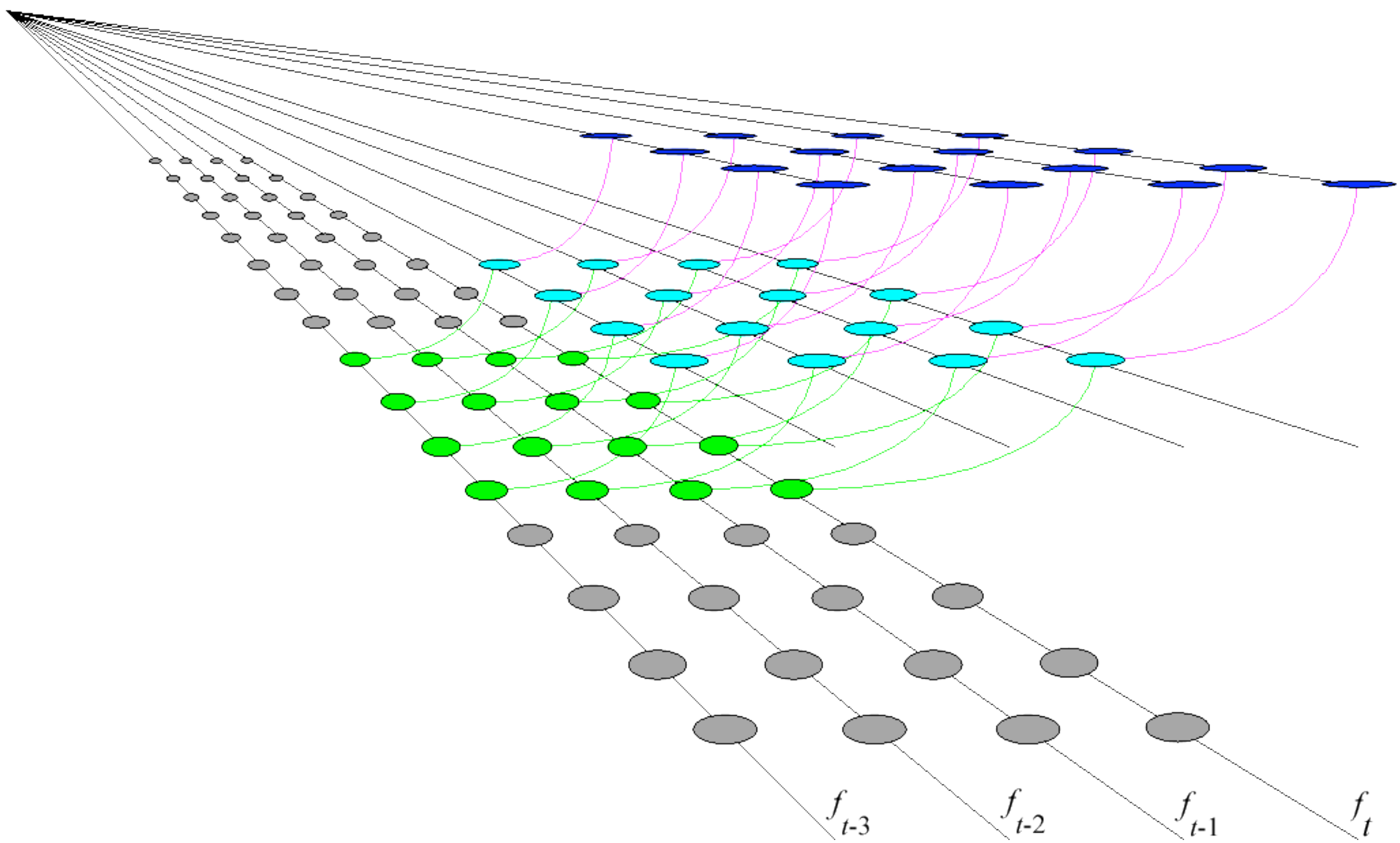
File Edit Options Buffers Tools C++ Help

```cpp
#include "mapreduce/mapreduce.h"

class Propagator : public Mapper {
 public:
  virtual void Map(const MapInput& input) {
    const int subnet_rank = input.key();
    const int[] observed = input.val();
    // Load the Subnet with the specified rank.
    Subnet& s = load(subnet_file(subnet_rank));
    // Update the Subnet using the observed values.
    enter_evidence(s, observed);
    // Emit parent-subnet/tuple assignments.
    for ( int i = 0; i < s.num_sources(); i++ ) {
      for ( int j = 0; j < s.source(i).num_consumers(); j++ ) {
        Emit(s.source(i).consumer(j), tuple(i, s.sample(i)));
      }
    }
    // Write the updated Subnet back to file.
    save(s, subnet_file(subnet_rank));
  }
}
Register_Mapper(Propagator);

class Consolidator : public Reducer {
 public:
  virtual void Reduce(ReduceInput& input) {
    const int subnet_rank = input.key();
    const tuple[] pairs = input.val();
    // Collect node/value tuples for each Subnet.
    Emit(subnet_rank, cat(sort(pairs,1),2));
  }
}
Register_Reducer(Consolidator);
```

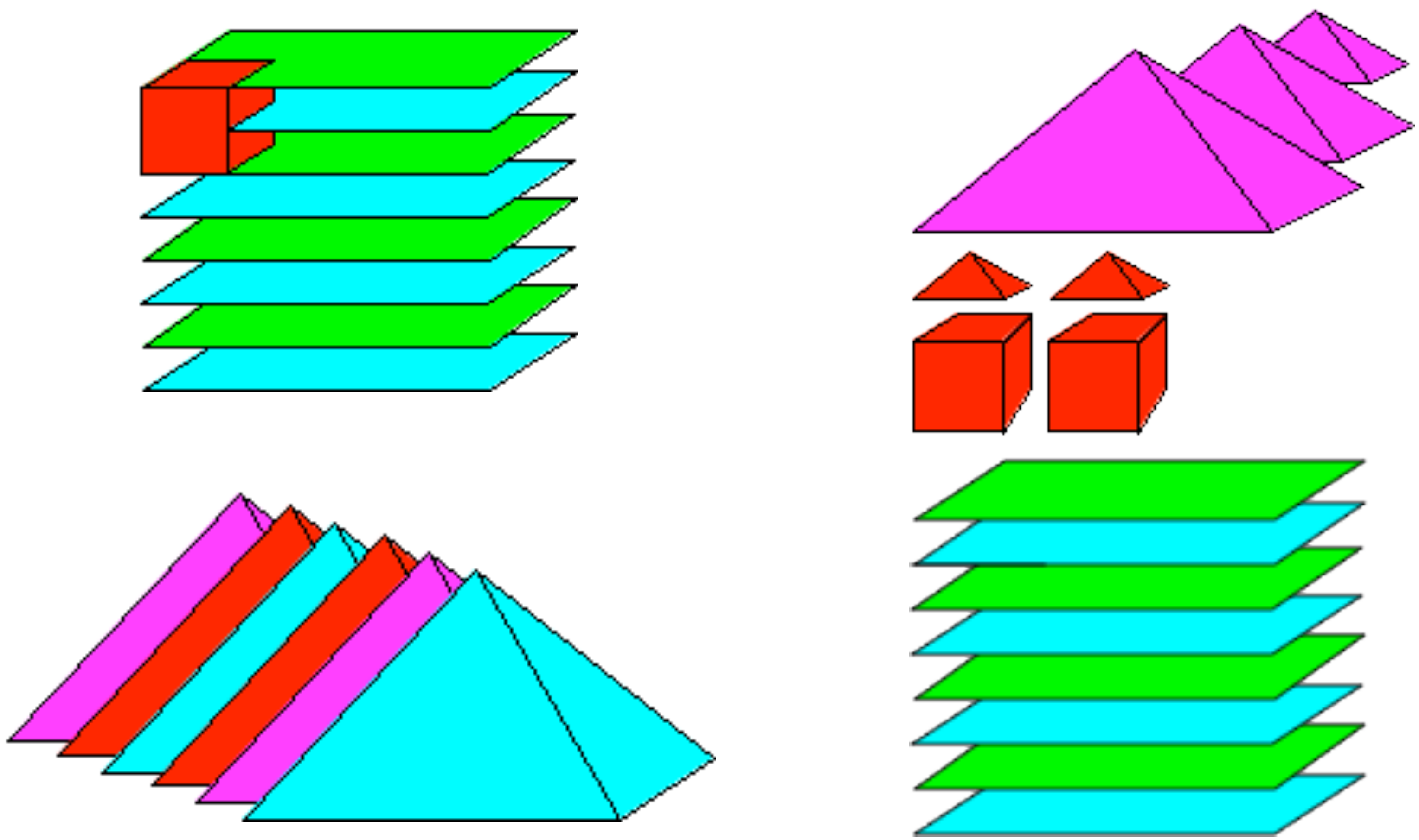
```cpp
int main(int argc, char** argv) {
  const PyramidGraphBayesNet pbn = load(argv[1]);
  for (int i = 0; i < pbn.num_levels(); i++) {
    MapReduceSpecification spec;
    // Propagate sample from bottom to top of the hierarchy.
    for ( int j = 1; j < pbn.level(i).num_subnets; j++ ) {
      MapReduceInput* input = spec.add_input();
      input->set_format("int");
      input->set_mapper_class("Propagator");
      input->set_key(pbn.level(i).subnet(j).rank());
      input->set_filepattern(pbn.level(i).subnet(j).rank());
    }
    // The output of one iteration becomes input to the next.
    MapReduceOutput* output = spec.output();
    output->set_format("int");
    output->set_reducer_class("Consolidator");
    // All of the results but the last are ignored.
    MapReduceResult* result;
    if ( !MapReduce(spec, &result) ) abort();
  }
  return 0;
}
```

```
----:---F1  mapreduce.cpp      (C++ Abbrev)--L2--Top-------------
----:---F1  mapreduce.cpp      (C++ Abbrev)--L50--Bot-------------
Wrote /Users/tld/presentations/05/AIMATH-05/figures/parallel/mapreduce.cpp
```

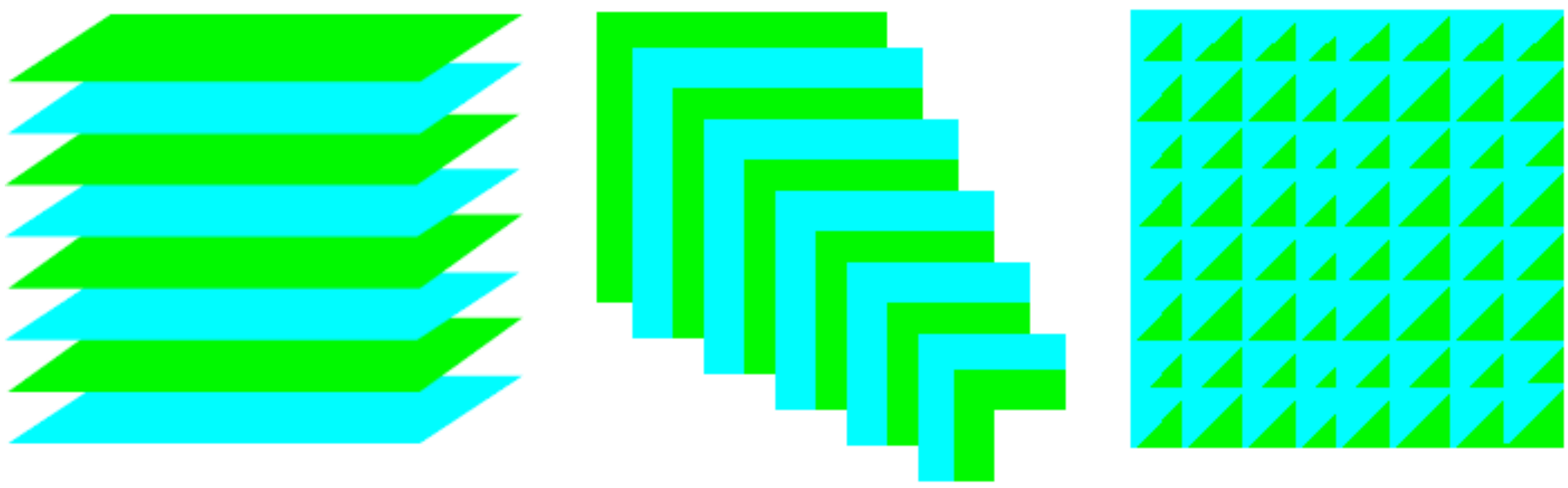
Some Perspective on Time



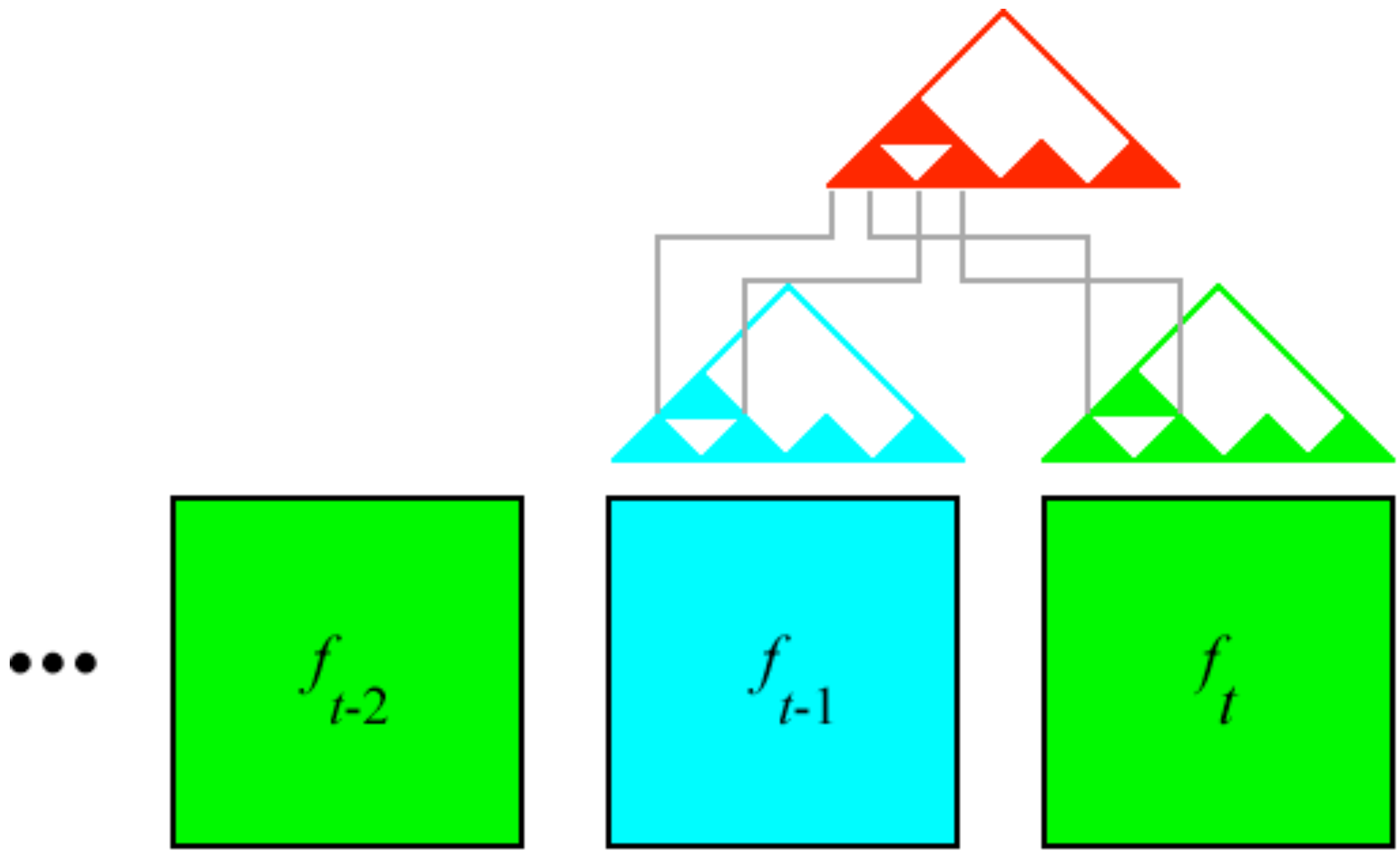
Spatiotemporal Receptive Fields



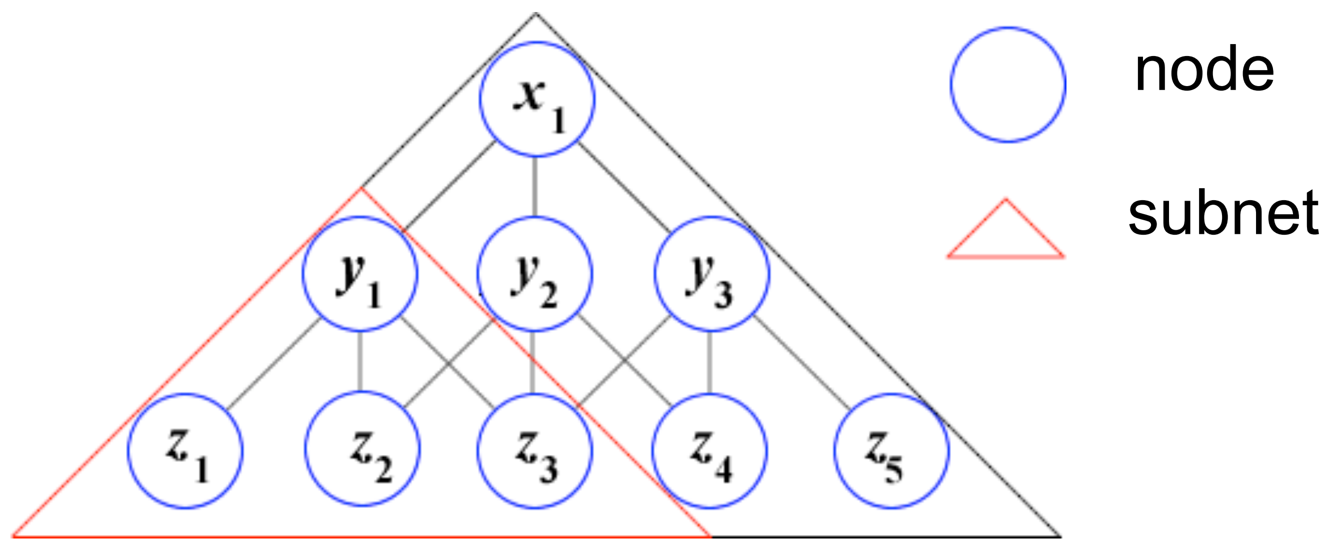
Organizing Time and Space



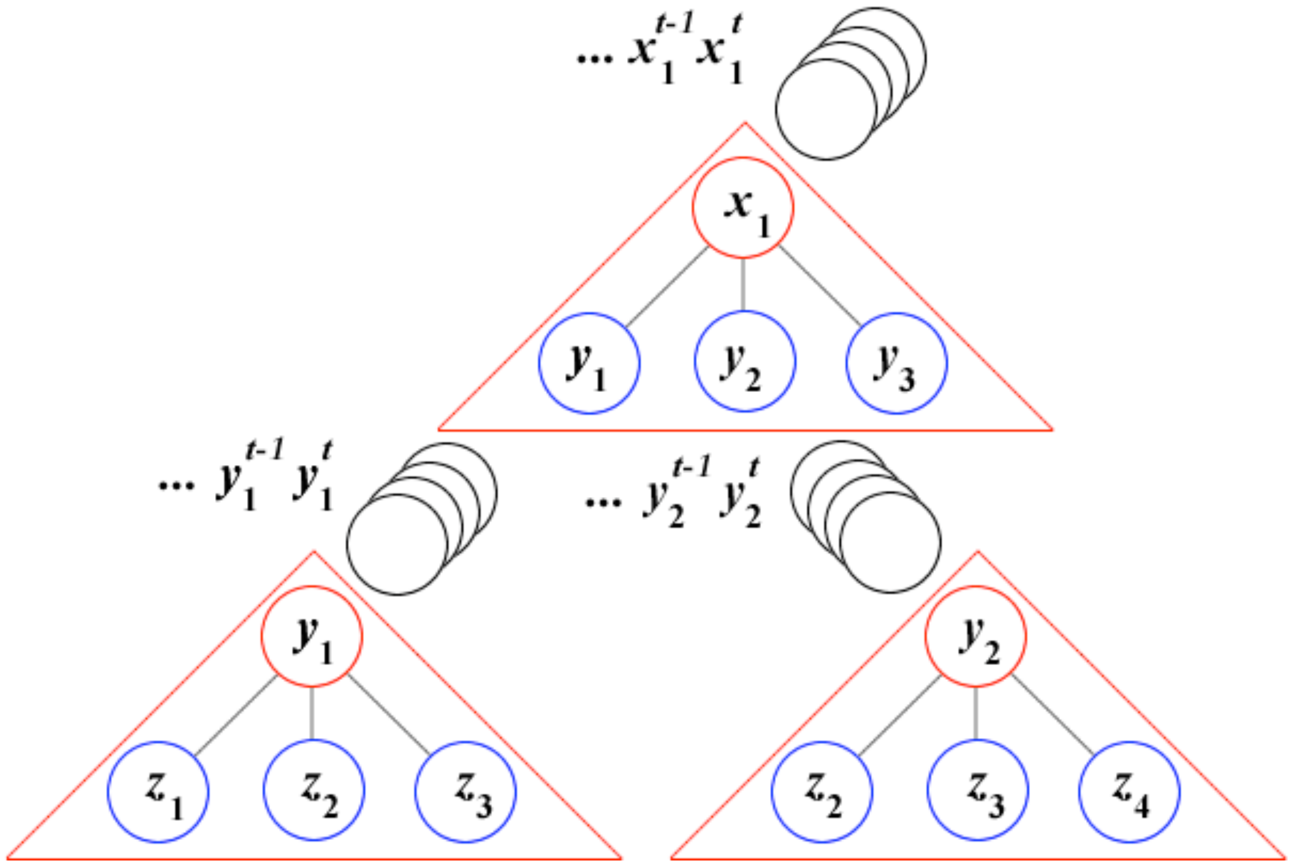
Video Frame Buffers



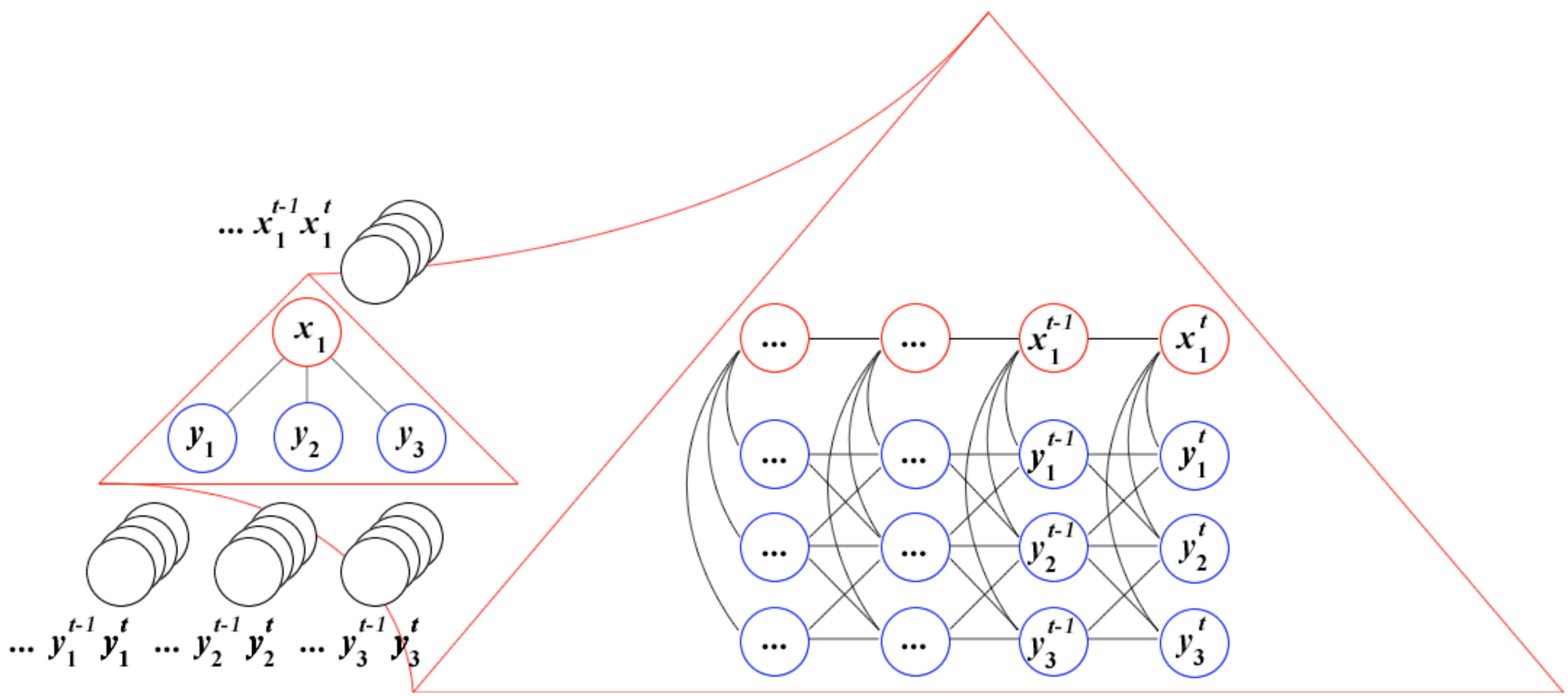
Pyramid-graph Fragment



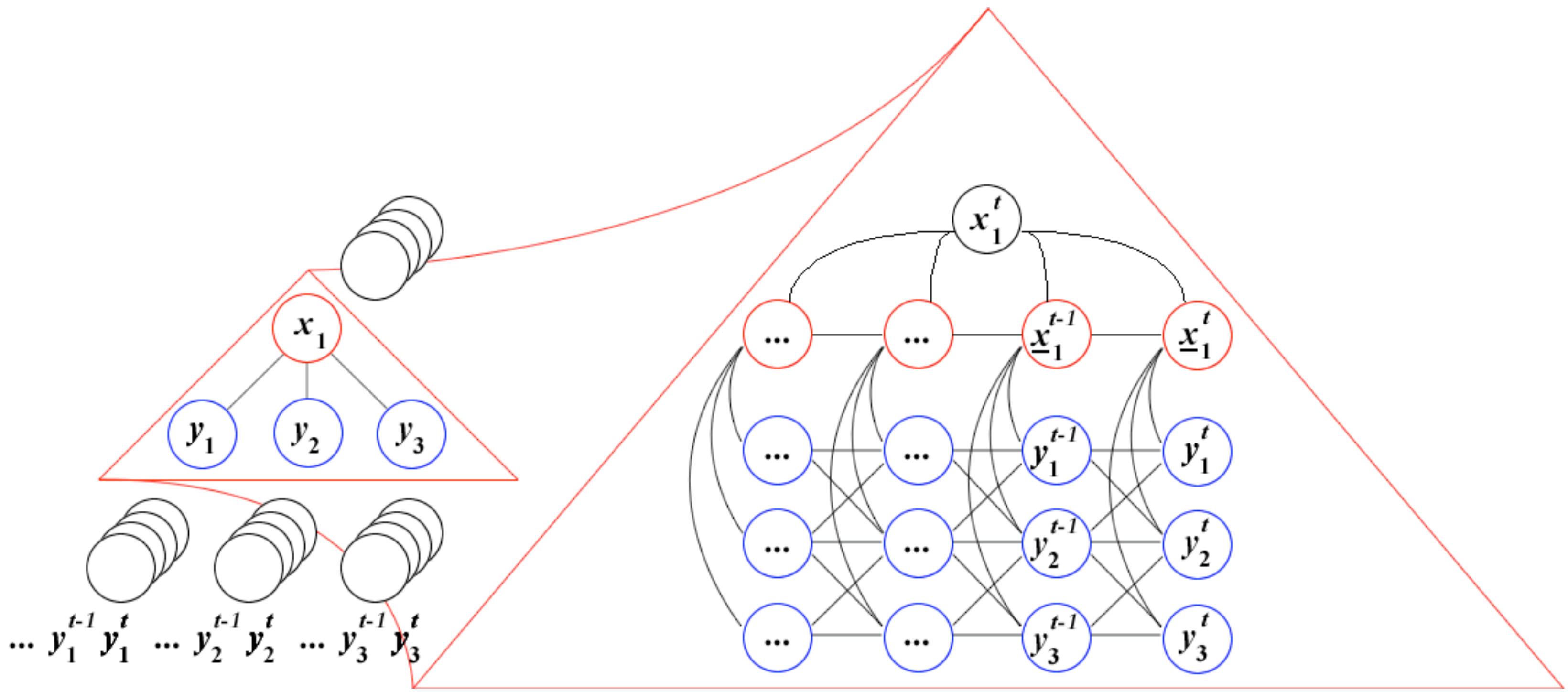
Subnet Sample Propagation



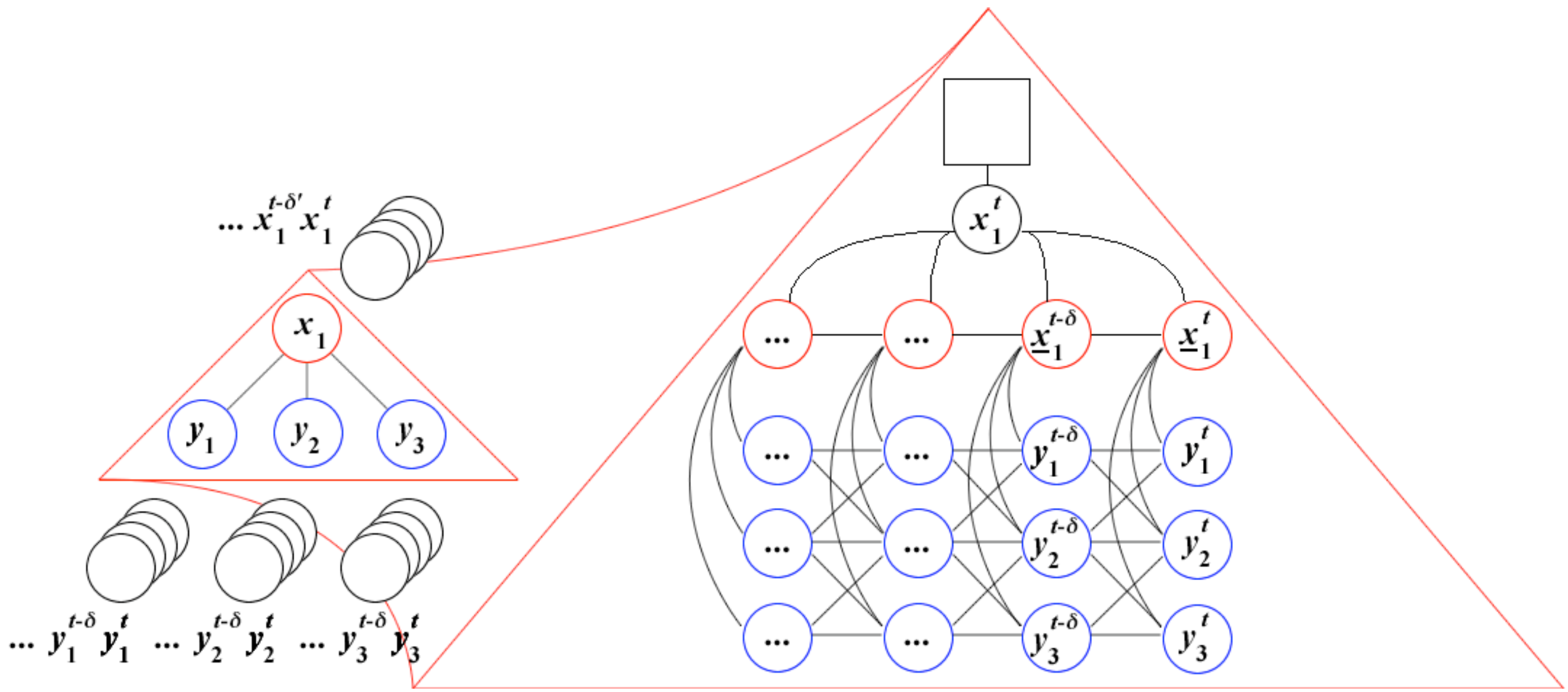
Inferring Spatiotemporal Features



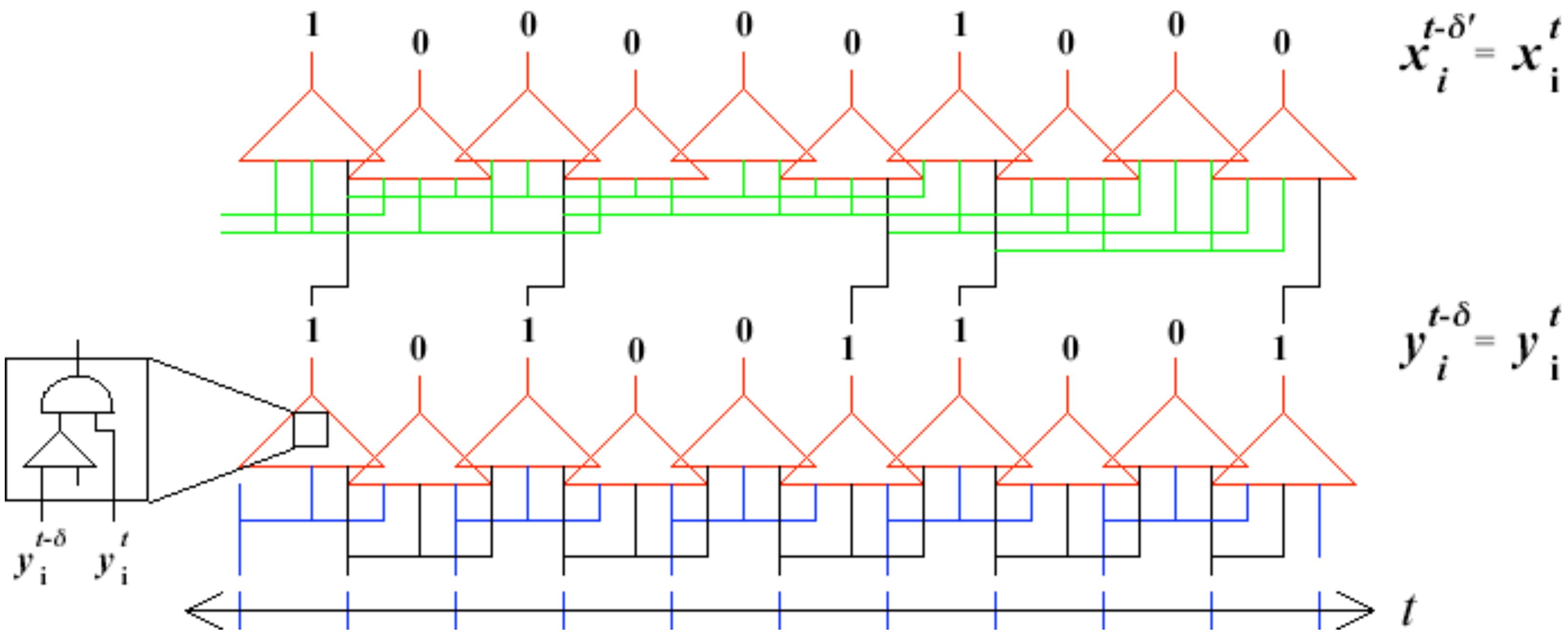
Hierarchical Hidden Markov Model



Subnet Sample Propagation



Variable Subnet Spatial Resolution



References

- G. Burns, R. Daoud and J. Vaigl. LAM: An Open Cluster Environment for MPI. In John W. Ross, editor, *Proceedings of Supercomputing Symposium '94.* University of Toronto. pp. 379-386. 1994.
- Dean, J. and Ghemawat, S. MapReduce: Simplified Data Processing on Large Clusters. *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*. San Francisco, CA. pp. 137-150, 2004.
- S. Fine, Y. Singer and N. Tishby. The Hierarchical Hidden Markov Model: Analysis and Applications. *Journal of the Optical Society of America Association,* **20**(7):1237-1252, 2003.
- K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernnetics,* **36**(4):93-202, 1980.
- D. George and J. Hawkins. A Hierarchical Bayesian Model of Invariant Pattern. *Proceedings of the International Joint Conference on Neural Networks,* 2005.
- M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience,* **2**(11):1019-1025, 1999.
- L. Wiskott and T. Sejnowski. Slow feature analysis: unsupervised learning of invariances. *Neural Computation,* **14**(4):715-770, 2002.

Picture Credits

- M. F. Bear, B. W. Connors and M. A. Paradiso. *Neuroscience: Exploring the Brain (2nd Edition).* Lippincott, Williams and Wilkins (2002).
- P. Dayan and L. F. Abbott. *Theoretical Neuroscience*. MIT Press (2001).
- D. H. Hubel. *Eye, Brain and Vision*. Scientific American Library (1988).