From Neural Networks to the Intelligent Power Grid: What It Takes to Make Things Work

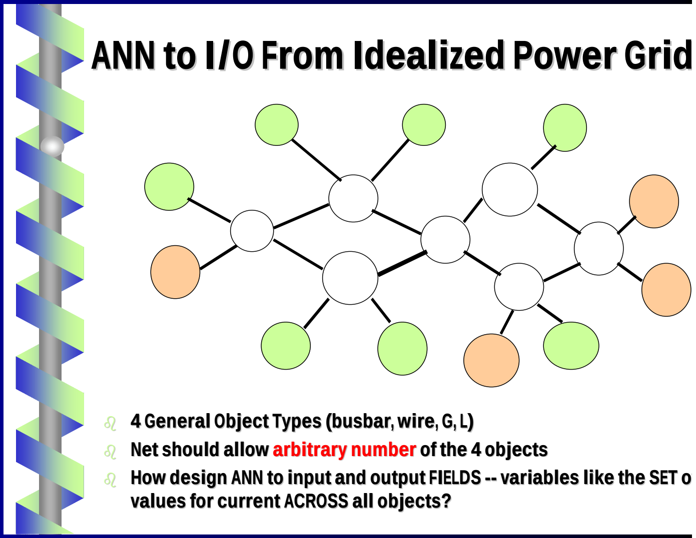
- What is an Intelligent Power Grid, and why do we need it?
- Why do we need neural networks?
- How can we make neural nets really work here, & in diagnostics/”prediction”/”control” in general?

What is a Truly Intelligent Power Grid?

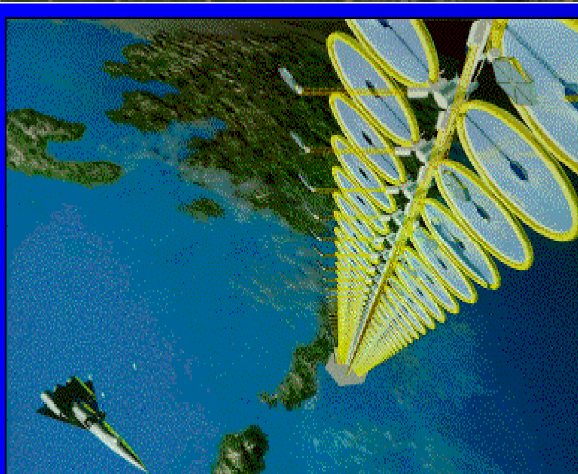
- True intelligence (like brain) $\Rightarrow$ foresight, $\Rightarrow$ ability to learn to coordinate all pieces, for optimal expected performance on the bottom line in future despite random disturbances.
- Managing complexity is easy– if you don't aim for best possible performance! The challenge is to come as close as possible to optimal performance of whole system.
- Bottom line utility function includes value added, quality of service (reliability), etc. A general concept. Nonlinear robust control is just a special case.
- Enhanced communication/chips/sensing/actuation/HPC needed for max benefit(cyberinfrastructure, EPRI roadmap)
- Brain-like intelligence = embodied intelligence, $\neq$ AI

Dynamic Stochastic Optimal Power Flow (DSOPF): How to Integrate the "Nervous System" of Electricity

- DSOPF02 started from EPRI question: can we optimally manage&plan the whole grid as one system, with foresight, etc.?
- Closest past precedent: Momoh's OPF integrates &optimizes many grid functions – but deterministic and without foresight. UPGRADE!
- ADP math required to add foresight and stochastics, critical to more complete integration.



Why It is a Life-or-Death Issue



HOW?

- www.ieeeusa.org/policy/energy_strategy.ppt
- Photo credit IEEE Spectrum

As Gas Prices ⇑ Imports ⇑ & Nuclear Tech in unstable areas ⇑, human extinction is a serious risk. Need to move faster.

Optimal time-shifting – big boost to rapid adjustment, $

Why It Requires Artificial Neural Networks (ANNs)

- For optimal performance in the general nonlinear case (nonlinear control strategies, state estimators, predictors, etc…), we need to adaptively estimate nonlinear functions. Thus we must use universal nonlinear function approximators.
- Barron (Yale) proved basic ANNs (MLP) much better than Taylor series, RBF, etc., to approximate smooth functions of many inputs. Similar theorems for approximating dynamic systems, etc., especially with more advanced, more powerful, MLP-like ANNs.
- ANNs more “chip-friendly” by definition: Mosaix chips, CNN here today, for embedded apps, massive thruput

Neural Networks That Actually Work In Diagnostics, Prediction & Control: Common Misconceptions Vs. Real-World Success

- Neural Nets, A Route to Learning/Intelligence
  - goals, history, basic concepts, consciousness
- State of the Art -- Working Tools Vs. Toys and Fads
  - static prediction/classification
  - dynamic prediction/classification
  - control: cloning experts, tracking, optimization
- Advanced Brain-Like Capabilities & Grids

Neural Nets: The Link Between Vision, Consciousness and Practical Applications

"Without vision, the people perish...."

What is a Neural Network?

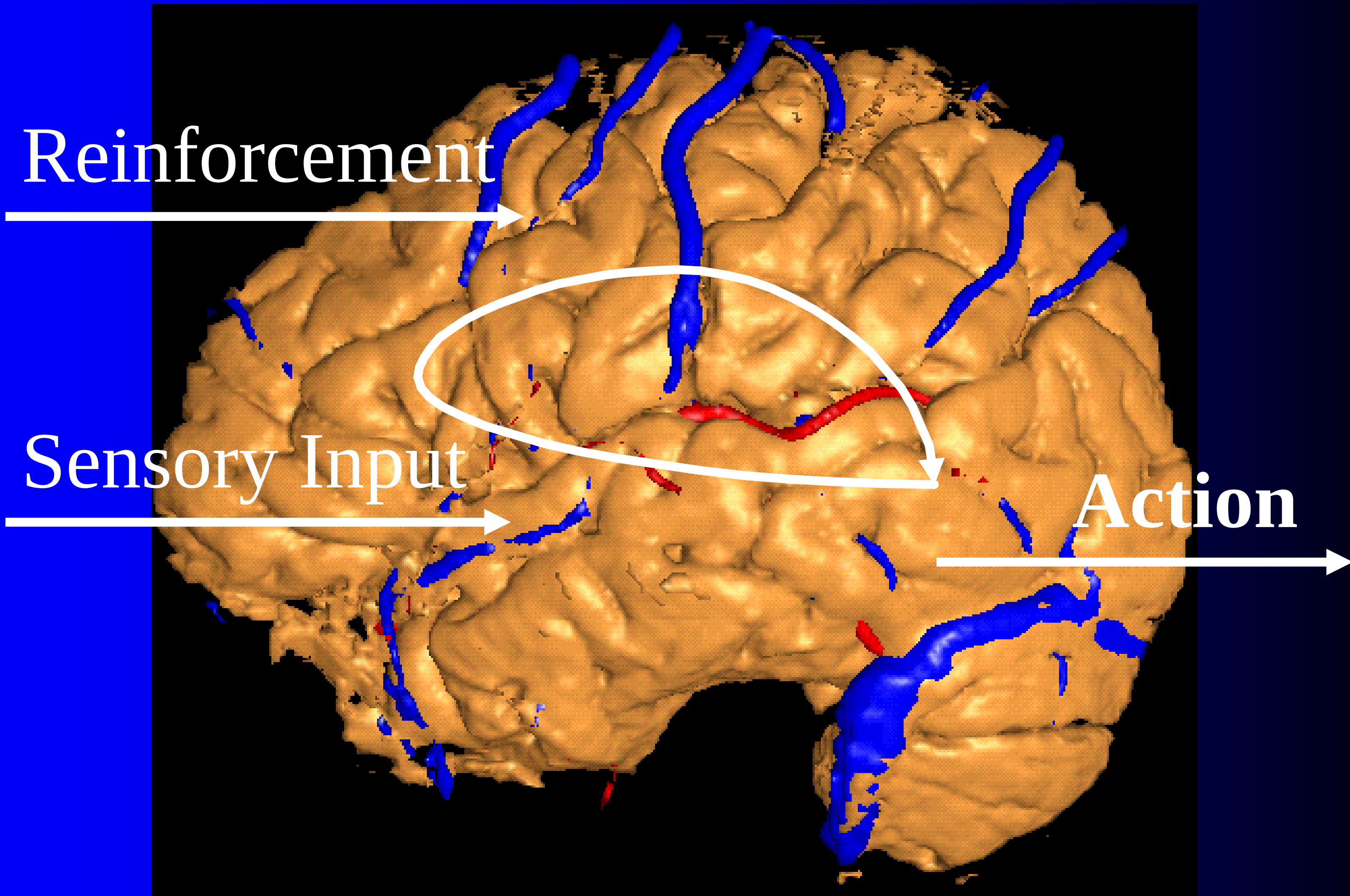
-- 4 definitions:"MatLab," universal approximators, 6th generation computing, brain-like computing

What is the Neural Network Field All About?

How Can We Get Better Results in Practical Applications?

Generations of Computers

- 4th Gen: Your PC. One VLSI CPU chip executes one sequential stream of C code.
- 5th Gen: “MPP”, “Supercomputers”: Many CPU chips in 1 box. Each does 1 stream. HPCC.
- 6th Gen or “ZISC.” Ks or Millions of simple streams per chip or optics. Neural nets may be defined as designs for 6th gen + learning. (Psaltis, Mead.)
  - New interest; Moore, SRC; Mosaix, JPL sugarcube, CNN.
- 7th Gen: Massively parallel quantum computing? General? Grover like Hopfield?



The Brain As a Whole System
Is an Intelligent Controller

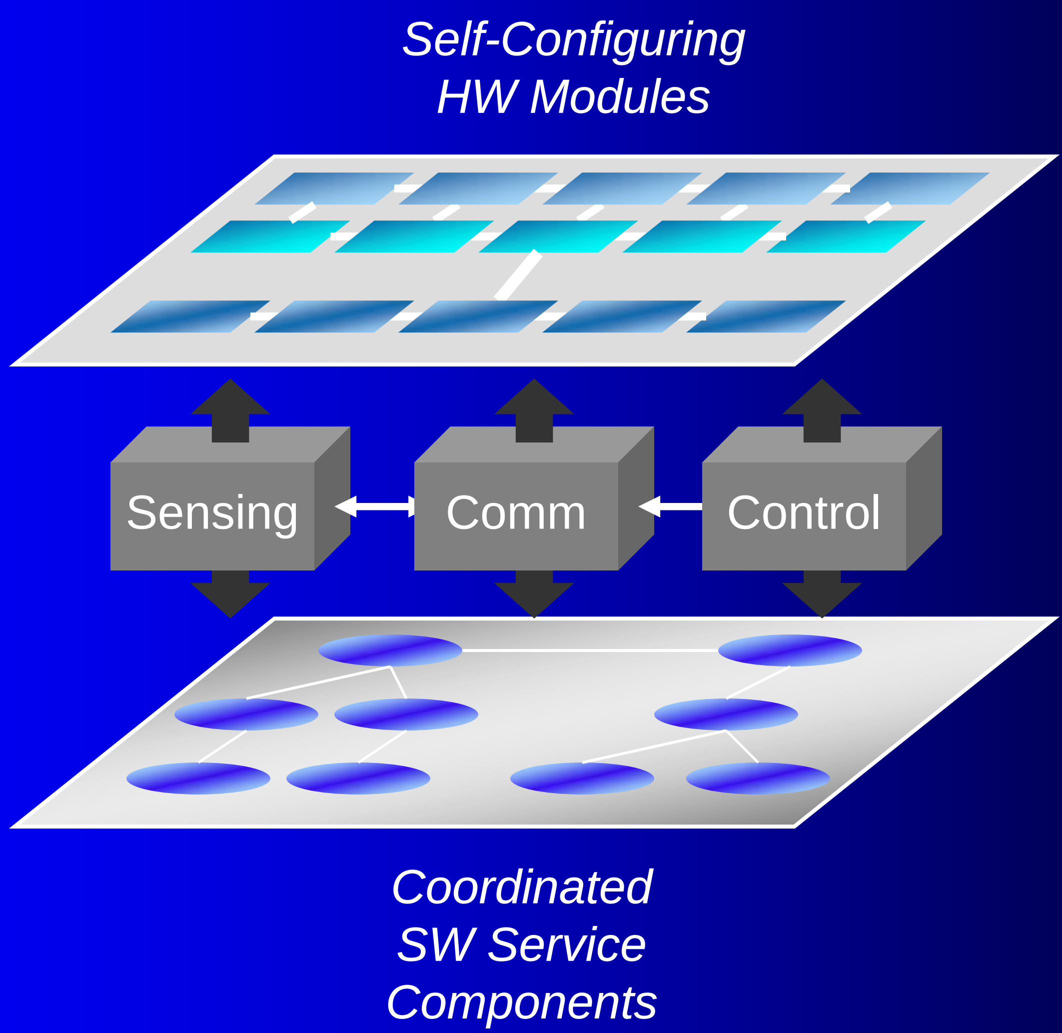
Unified Neural Network Designs:

The Key to Large-Scale Applications
& Understanding the Brain

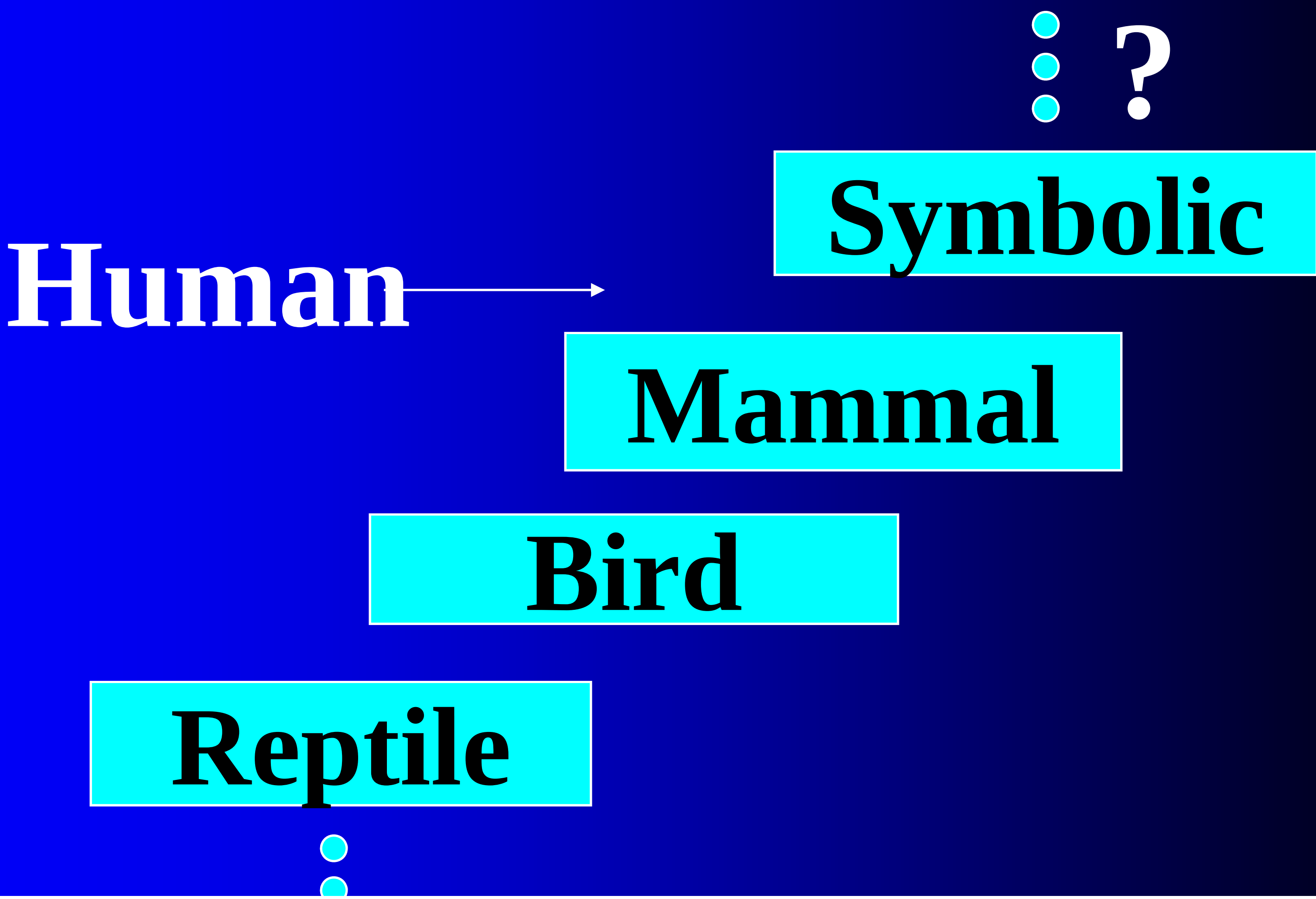
Electrical and Communications Systems(ECS) Cyber Infrastructure Investments

- **The Physical Layer – Devices and Networks**
  - **National Nanofabrication Users Network (NNUN)**
  - **Ultra-High-Capacity Optical Communications and Networking**
  - **Electric Power Sources, Distributed Generation and Grids**
- **Information Layer – Algorithms, Information and Design**
  - **General tools for distributed, robust, adaptive, hybrid control & related tools for modeling, system identification, estimation**
  - **General tools for sensors-to-information & to decision/control**
  - **Generality via computational intelligence, machine learning, neural networks & related pattern recognition, data mining etc.**
- **Integration of Physical Layer and Information Layer**
  - **Wireless Communication Systems**
  - **Self-Organizing Sensor and Actuator Networks**
  - **System on Chip for Information and Decision Systems**
  - **Reconfigurable Micro/Nano Sensor Arrays**
  - **Efficient and Secure Grids and Testbeds for Power Systems**

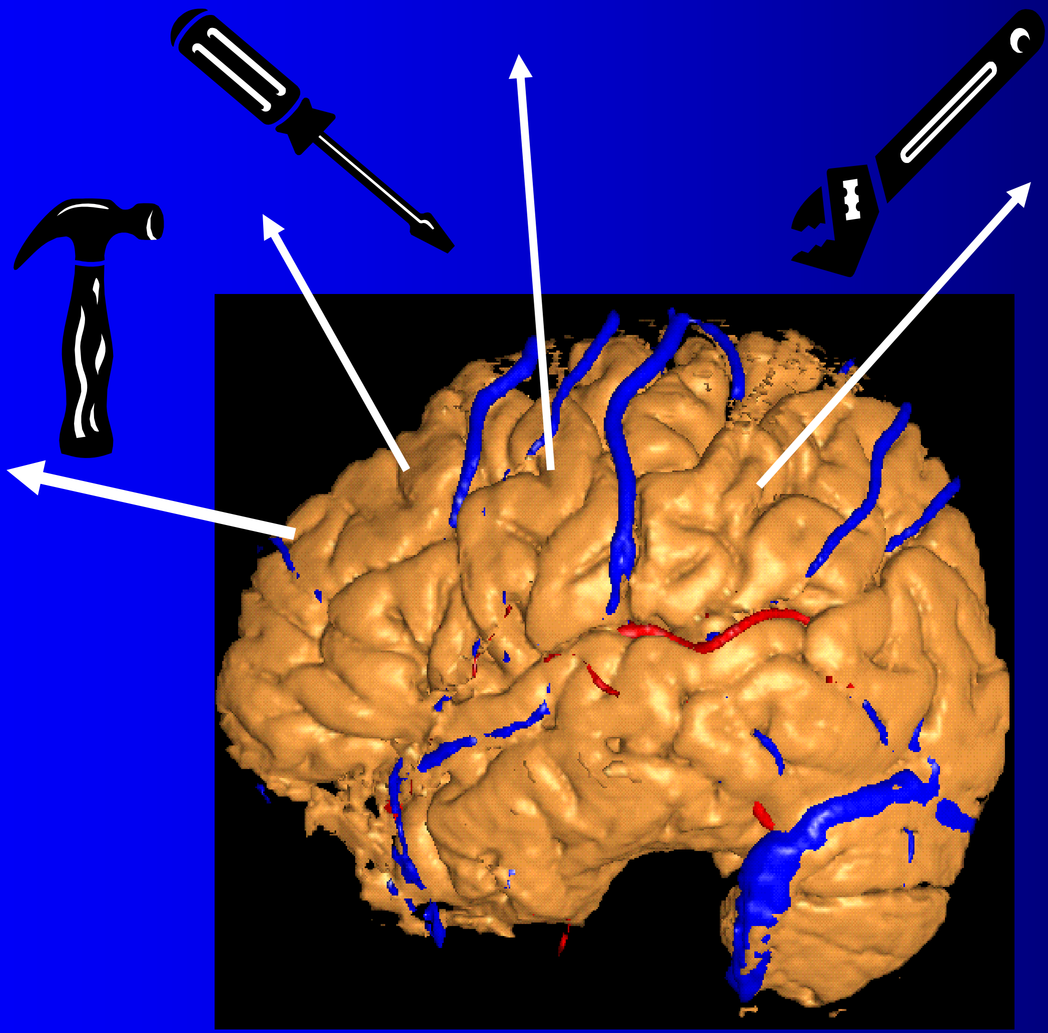
Cyberinfrastructure: The Entire Web From Sensors To Decisions/Actions/Control For Max Performance



Levels of Intelligence



Why Engineers Need This Vision:



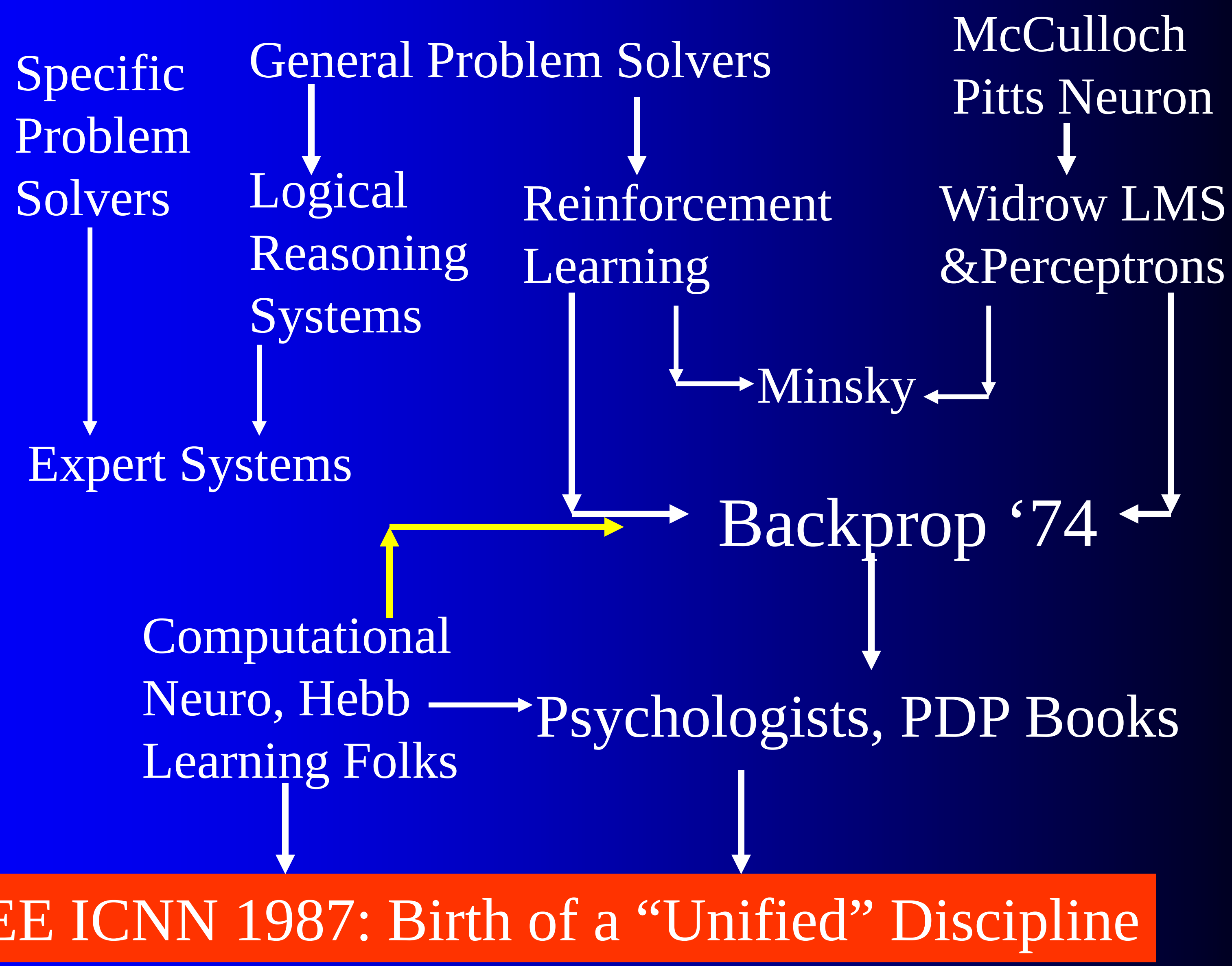
1. To Keep Track of MANY Tools

2. To Develop New Tools -- To Do Good R&D & Make Max Contribution

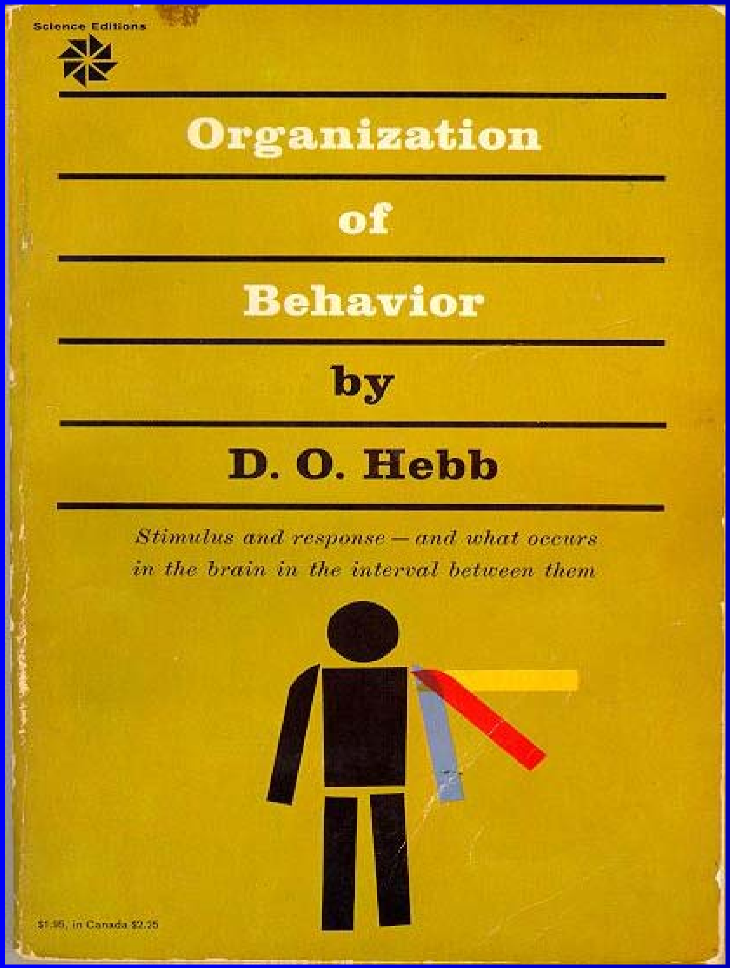
3. To Attract & Excite the Best Students

4. Engineers are Human Too...

Where Did ANNs Come From?



Hebb 1949: Intelligence As An Emergent Phenomenon or Learning



"The general idea is an old one, that any two cells or systems of cells that are especially active at the same time will tend to become 'associated,' so that activity in one facilitates activity in the other" -- p.70 (Wiley 1961 printing)

The search for the General Neuron Model (of Learning)

"Solves all problems"

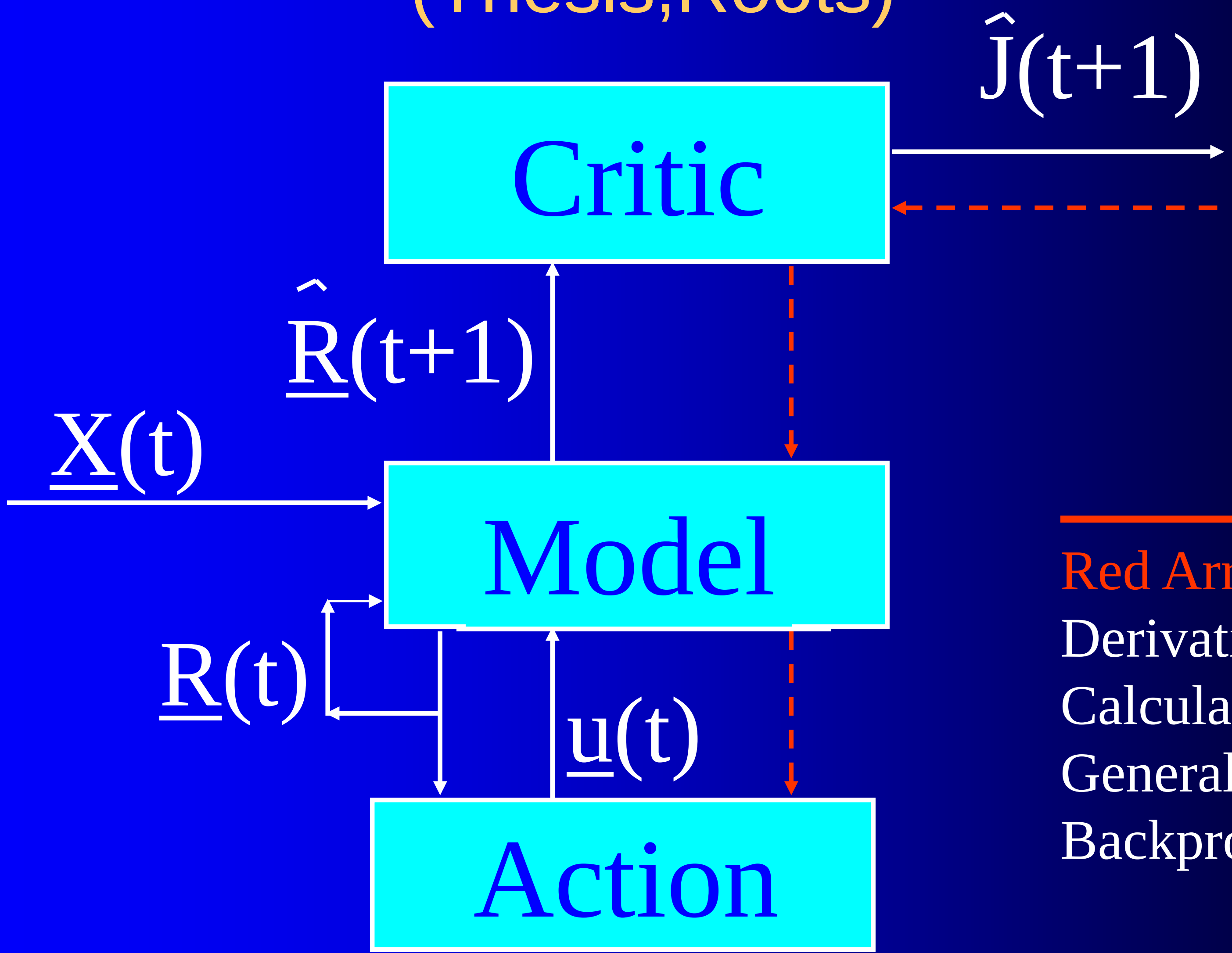
Claim (1964) : Hebb's Approach Doesn't Quite Work As Stated

- Hebbian Learning Rules Are All Based on Correlation Coefficients
- Good Associative Memory: one component of the larger brain (Kohonen, ART, Hassoun)
- Linear decorrelators and predictors
- Hopfield f(u) minimizers never scaled, but:
  - Gursel Serpen and SRN minimizers
  - Brain-Like Stochastic Search (Needs R&D)

Understanding Brain Requires Models Tested/Developed Using Multiple Sources of Info

- Engineering: Will it work? Mathematics understandable, generic?
- Psychology: Connectionist cognitive science, animal learning, folk psychology
- Neuroscience: computational neuroscience
- AI: agents, games (backgammon, go), etc.
- LIS and CRI

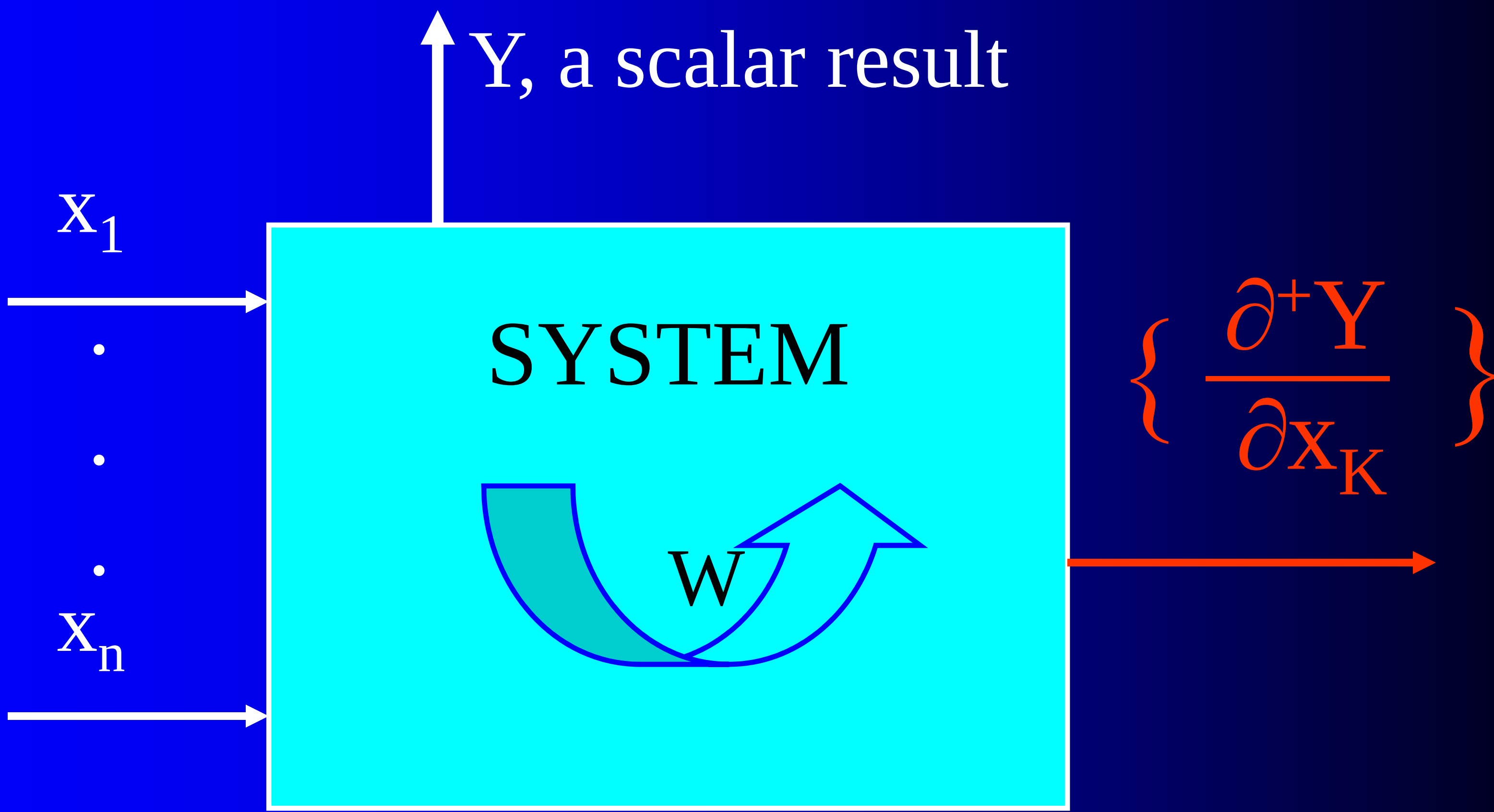
1971-2: Emergent Intelligence Is Possible If We Allow Three Types of Neuron (Thesis,Roots)



Red Arrows: Derivatives Calculated By Generalized Backpropagation

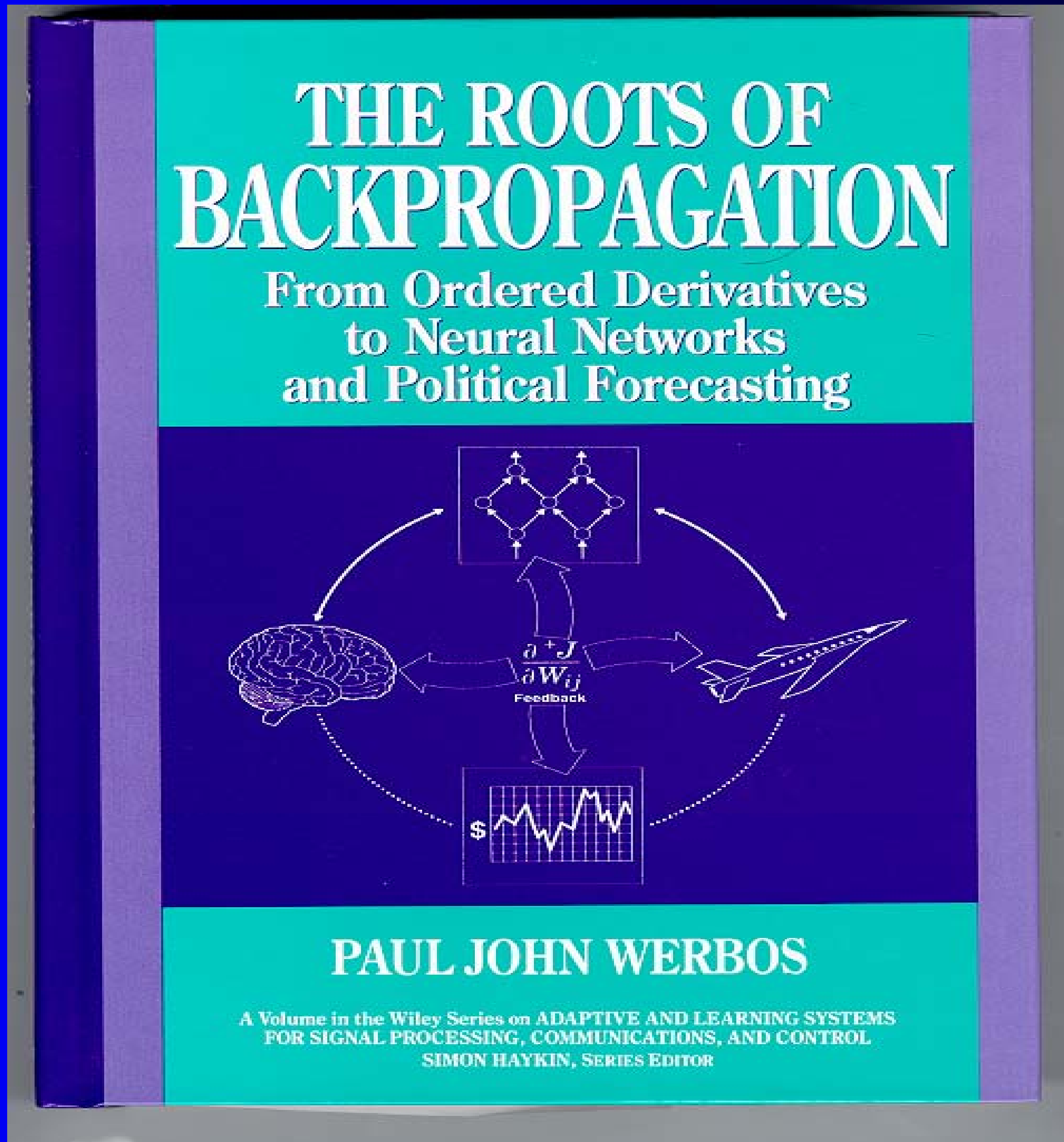
Harvard Committee Response

- We don’t believe in neural networks – see Minsky (Anderson&Rosenfeld, Talking Nets)
- Prove that your backwards differentiation works. (That is enough for a PhD thesis.) The critic/DP stuff published in ’77,’79,’81,’87..
- Applied to affordable vector ARMA statistical estimation, general TSP package, and robust political forecasting

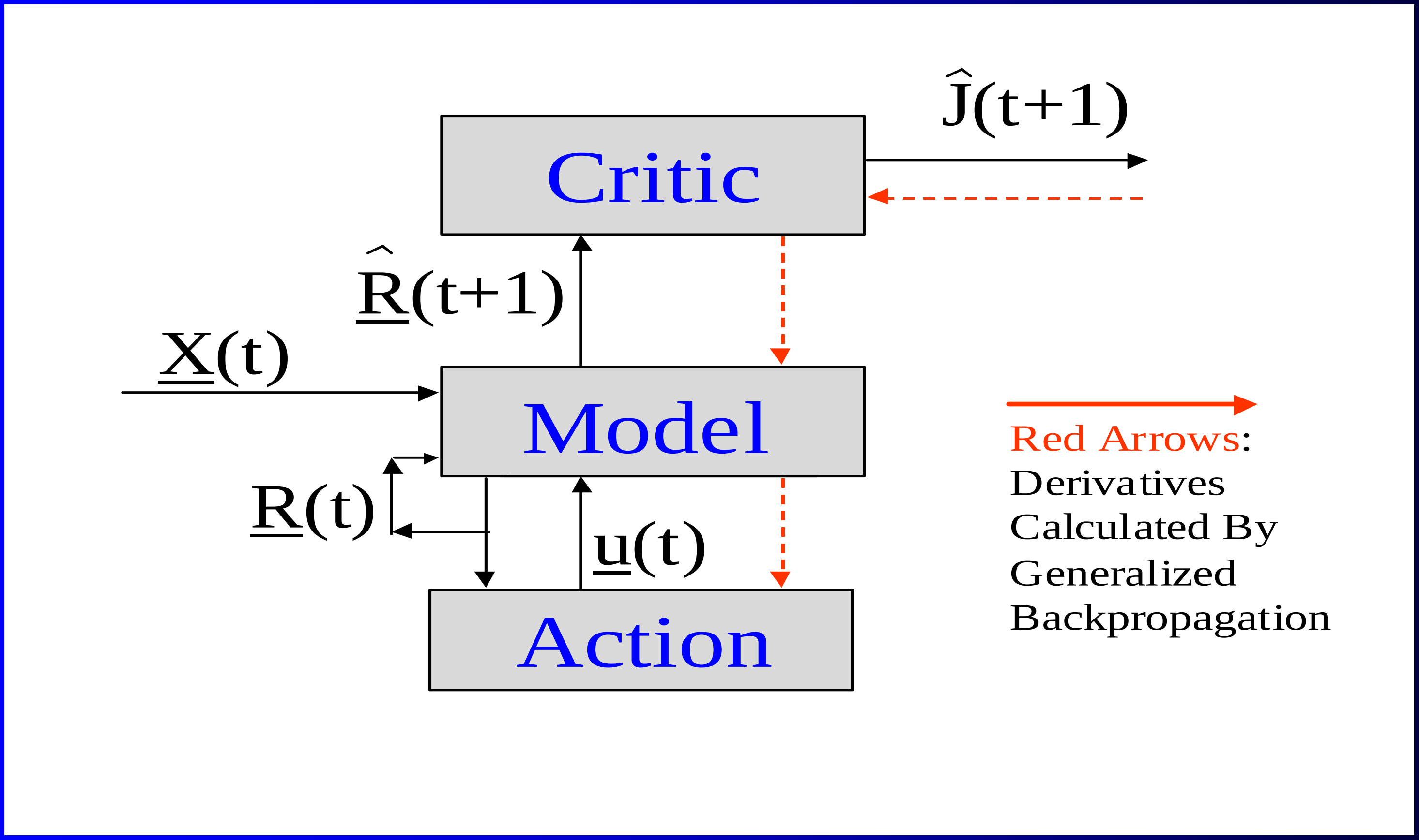


(Inputs x_k may actually come from many times)

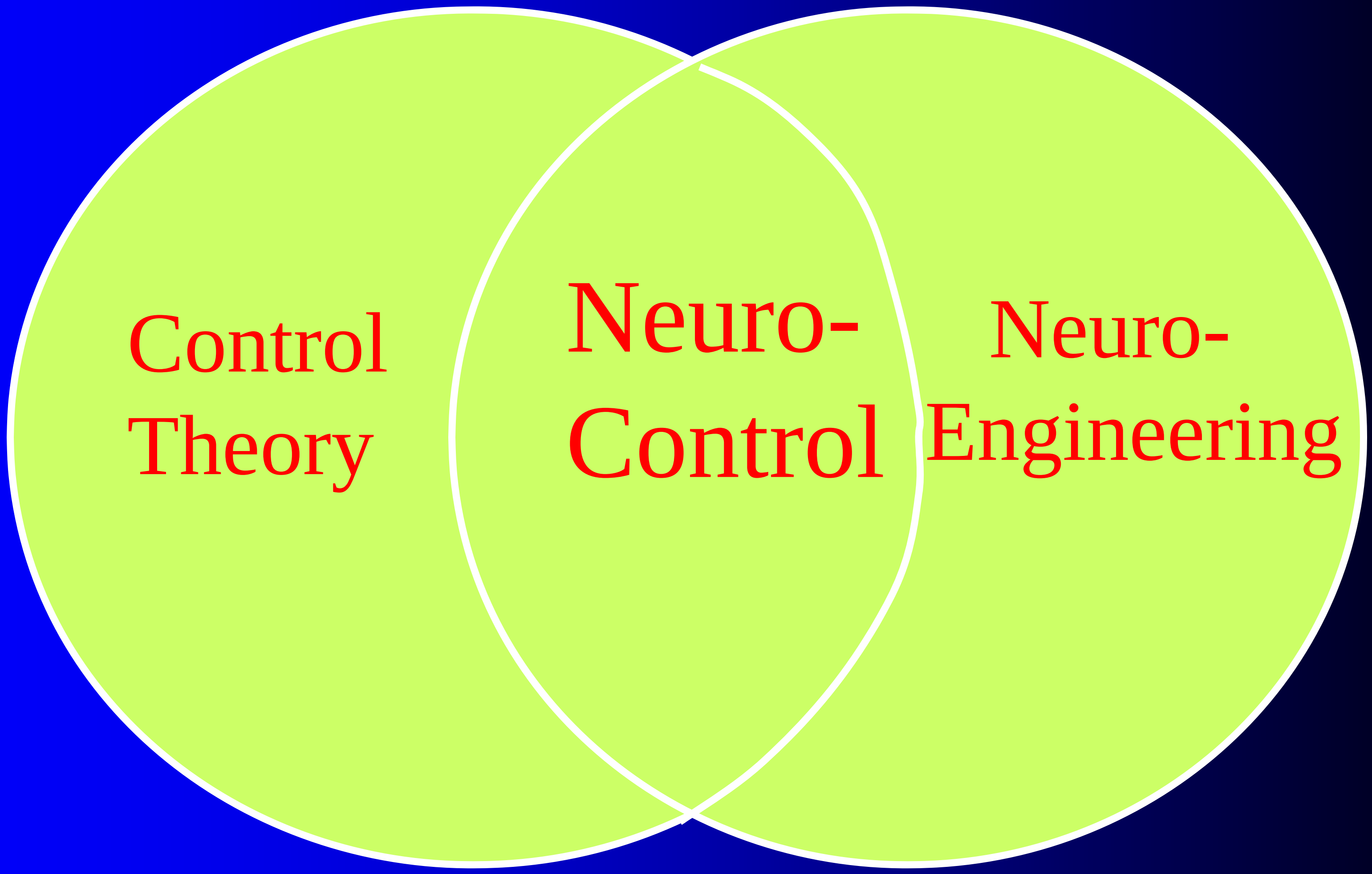
Backwards Differentiation: But what kinds of SYSTEM can we handle? See details in AD2004 Proceedings, Springer, in press.


THE ROOTS OF
BACKPROPAGATION
From Ordered Derivatives
to Neural Networks
and Political Forecasting
$\frac{\partial^+ J}{\partial W_{ij}}$
Feedback
$
PAUL JOHN WERBOS
A Volume in the Wiley Series on ADAPTIVE AND LEARNING SYSTEMS
FOR SIGNAL PROCESSING, COMMUNICATIONS, AND CONTROL
SIMON HAYKIN, SERIES EDITOR

To Fill IN the Boxes:
(1) NEUROCONTROL, to Fill in Critic or Action;
(2) System Identification or Prediction (Neuroidentification) to Fill In Model



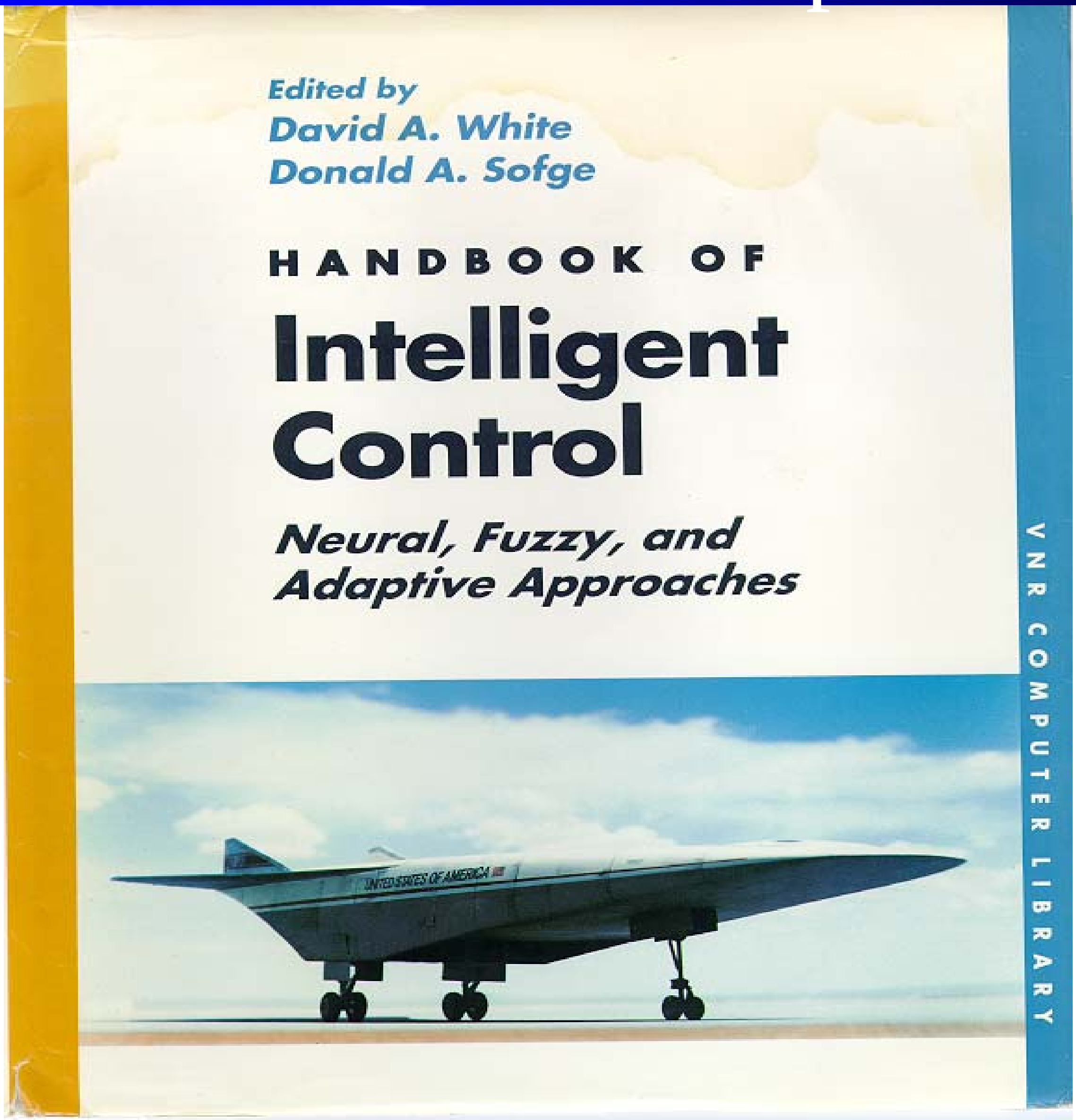
NSF Workshop Neurocontrol 1988



Miller, Sutton, Werbos, MIT Press, 1990

Neurocontrol is NOT JUST Control Theory!

NSF/McAir Workshop 1990



White and Sofge eds, Van Nostrand, 1992

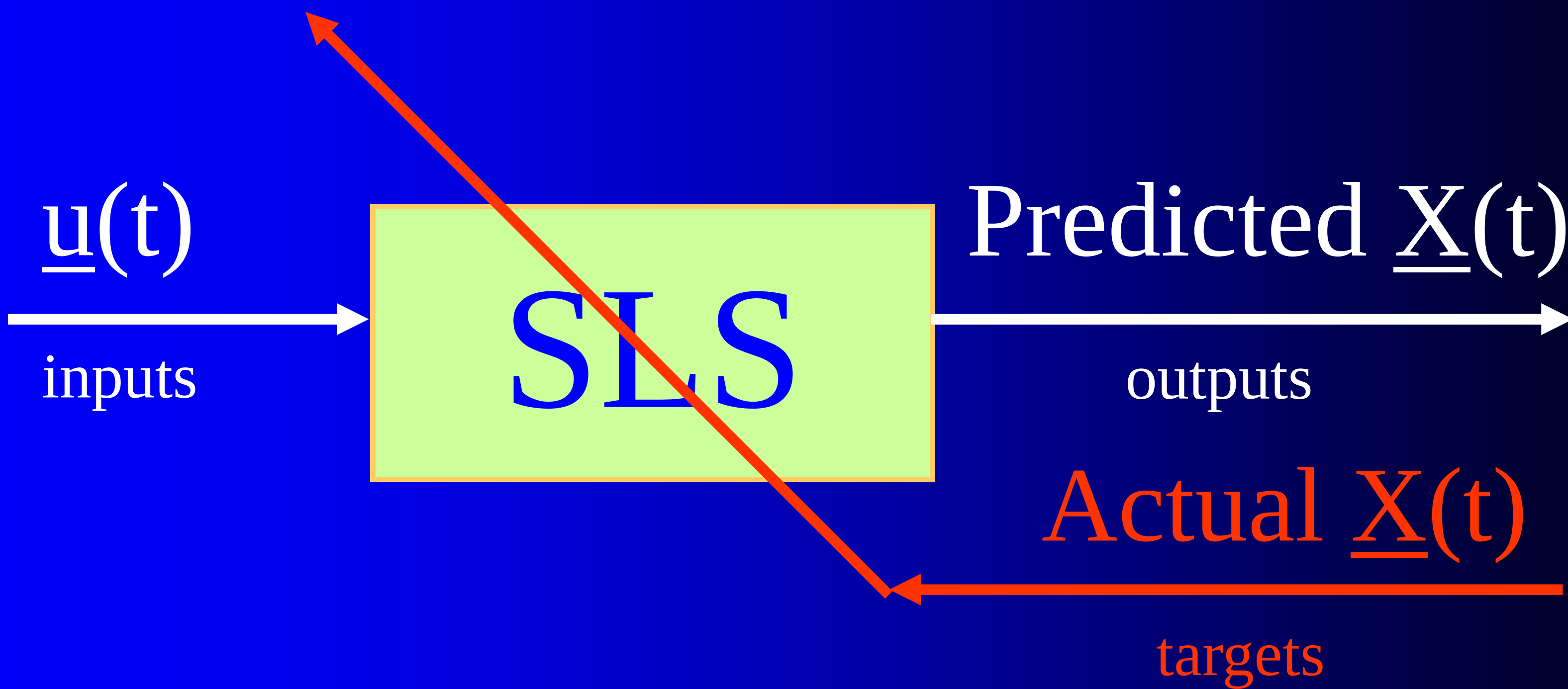
“What Do Neural Nets & Quantum Theory Tell Us About Mind & Reality?”

In Yasue et al (eds), *No Matter, Never Mind -- Proc. Of Towards a Science of Consciousness*, John Benjamins (Amsterdam), 2001 & arxiv.org

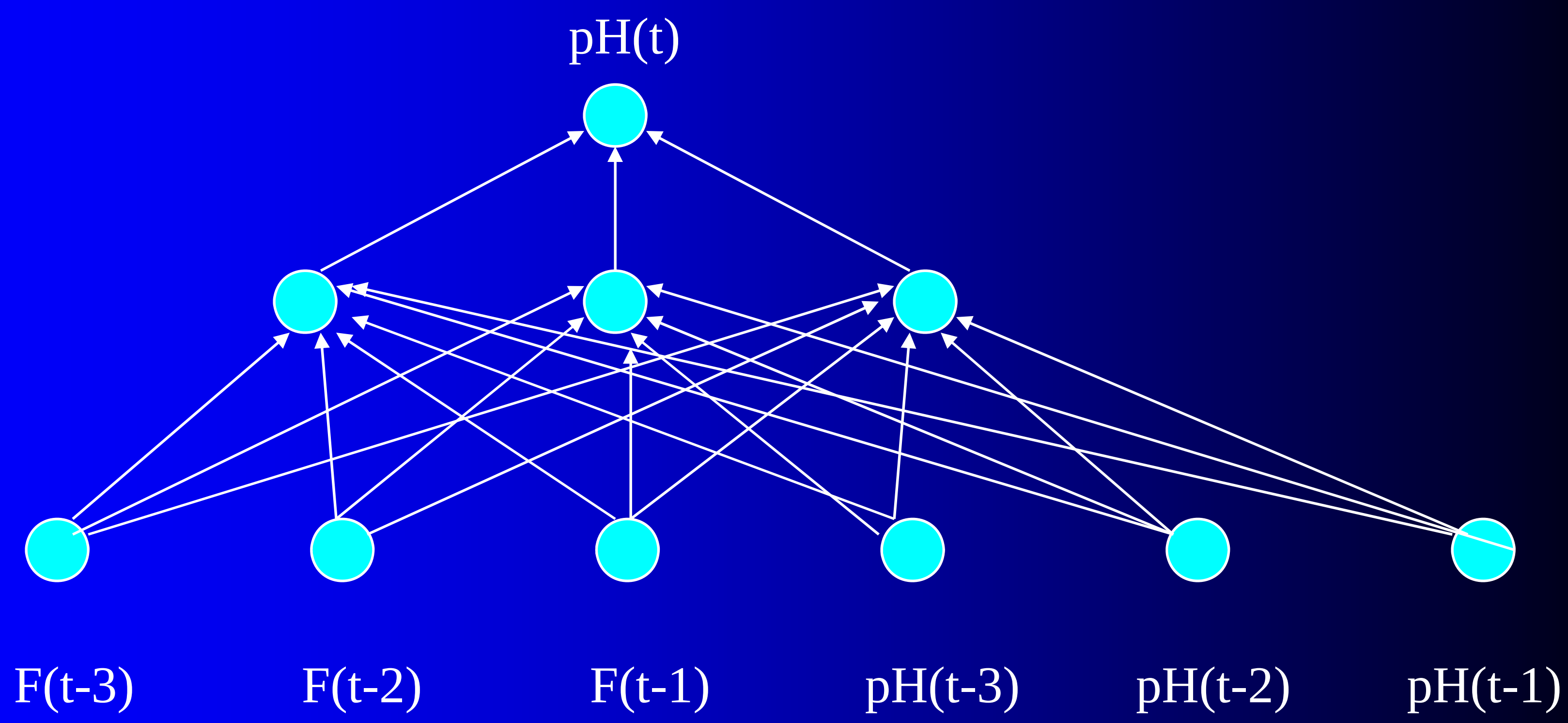
3 Types of Diagnostic System

- All 3 train predictors, use sensor data $\underline{X}(t)$, other data $\underline{u}(t)$, fault classifications F_1 to F_m
- Type 1: predict $F_i(t)$ from $\underline{X}(t)$, $\underline{u}(t)$, MEMORY
- Others: first train to predict $\underline{X}(t+1)$ from $\underline{X}$,$\underline{u}$,MEM
  - Type 2: when actual $\underline{X}(t+1)$ 6σ from prediction, ALARM
  - Type 3: if prediction net predicts BAD $\underline{X}(t+T)$, ALARM
- Combination best. See PJW in Maren, ed, *Handbook Neural Computing Apps*, Academic, 1990.

Supervised Learning Systems (SLS)



SLS may have internal dynamics but no “memory” of times t-1, t-2...



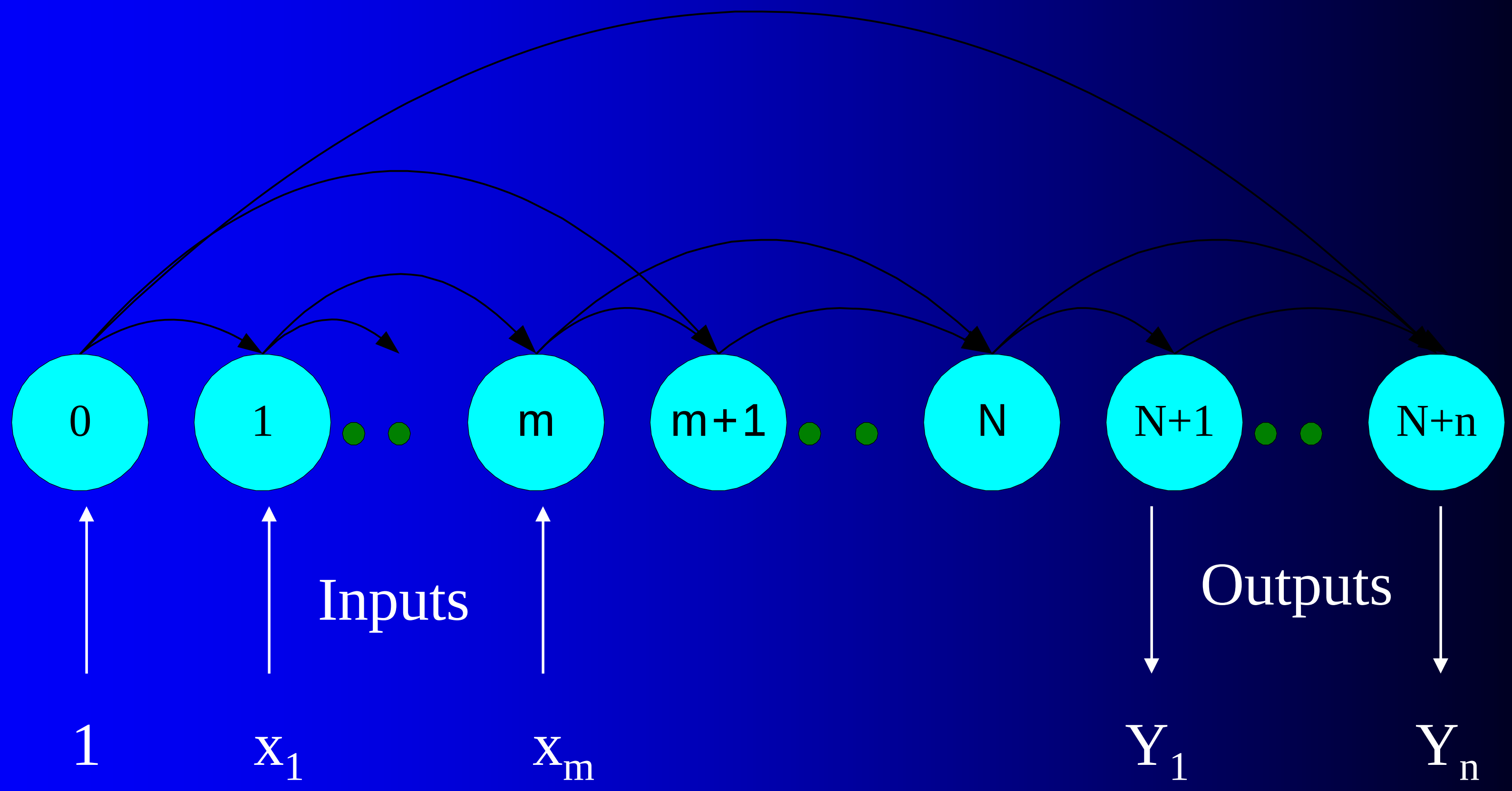
Example of TDNN used in HIC, Chapter 10

TDNNs learn NARX or FIR Models, not NARMAX or IIR

CONVENTIONAL ANNS USED FOR FUNCTION APPROXIMATION IN CONTROL

- Global: Multilayer Perceptron (MLP)
  - Better Generalization, Slower Learning
  - Barron’s Theorems: More Accurate Approximation of Smooth Functions as Number of Inputs Grows
- Local: RBF, CMAC, Hebbian
  - Like Nearest Neighbor, Associative Memory
  - Sometimes Called “Glorified Lookup tables”

Generalized MLP



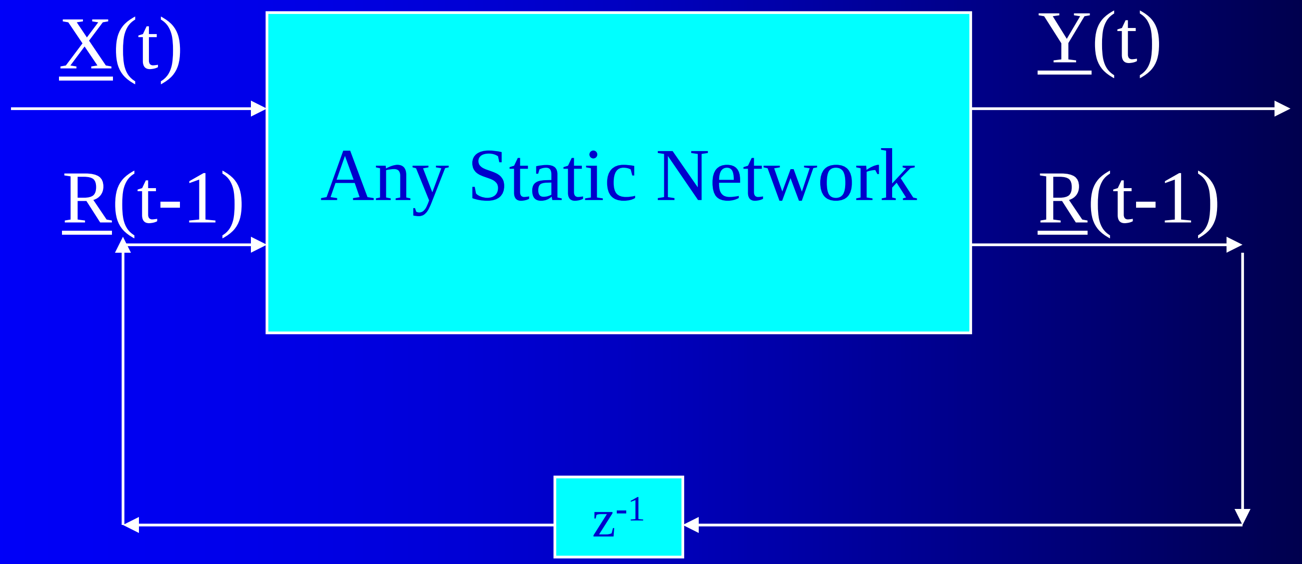
No feedforward or associative memory net can give brain-like performance! Useful recurrence--

- For short-term memory, for state estimation, for fast adaptation – time-lagged recurrence needed. (TLRN = time-lagged recurrent net)
- For better Y=F(X,W) mapping, Simultaneous Recurrent Networks Needed. For large-scale tasks, SRNs WITH SYMMETRY tricks needed – cellular SRN, Object Nets
- For robustness over time, “recurrent training”

Why TLRNs Vital in Prediction: Correlation ≠ Causality!

- E.g.: law X sends extra $ to schools with low test scores
- Does negative correlation of $ with test scores imply X is a bad program? No! Under such a law, negative correlation is hard-wired. Low test scores cause $ to be there! No evidence + or – re the program effect!
- Solution: compare $ at time t with performance changes from t to t+1! More generally/accurately: train dynamic model/network – essential to any useful information about causation or for decision!

The Time-Lagged Recurrent Network (TLRN)



$\underline{Y}(t)=\underline{f}(\underline{X}(t), \underline{R}(t-1))$; $\underline{R}(t)=\underline{g}(\underline{X}(t), \underline{R}(t-1))$

$\underline{f}$ and $\underline{g}$ represent 2 outputs of one network

All-encompassing, NARMAX($1 \equiv n$)

Felkamp/Prokhorov Yale03: >>EKF,≈ hairy

4(5) Ways to Train TLRNs (SRN) (arXiv.org, adap-org 9806001)

- "Simple BP" – incorrect derivatives due to truncated calaculation, robustness problem
- BTT – exact, efficient, see Roots of BP ('74), but not brain-like (back time calculations)
- Forward propagation – many kinds (e.g, Roots, ch.7, 1981) – not brainlike, O(nm)
- Error Critic– see Handbook ch. 13, Prokhorov
- Simultaneous BP – SRNS only.

4 Training Problems Recurrent Nets

- Bugs – need good diagnostics
- "Bumpy error surface" – Schmidhuber says is common, Ford not. Sticky neuron, RPROP, DEFK (Ford), etc.
- Shallow plateaus – adaptive learning rate, DEKF etc., new in works…
- Local minima – shaping, unavoidable issues, creativity

GENERALIZED MAZE PROBLEM

Jhat(ix,iy) for all 0<ix,iy<N+1

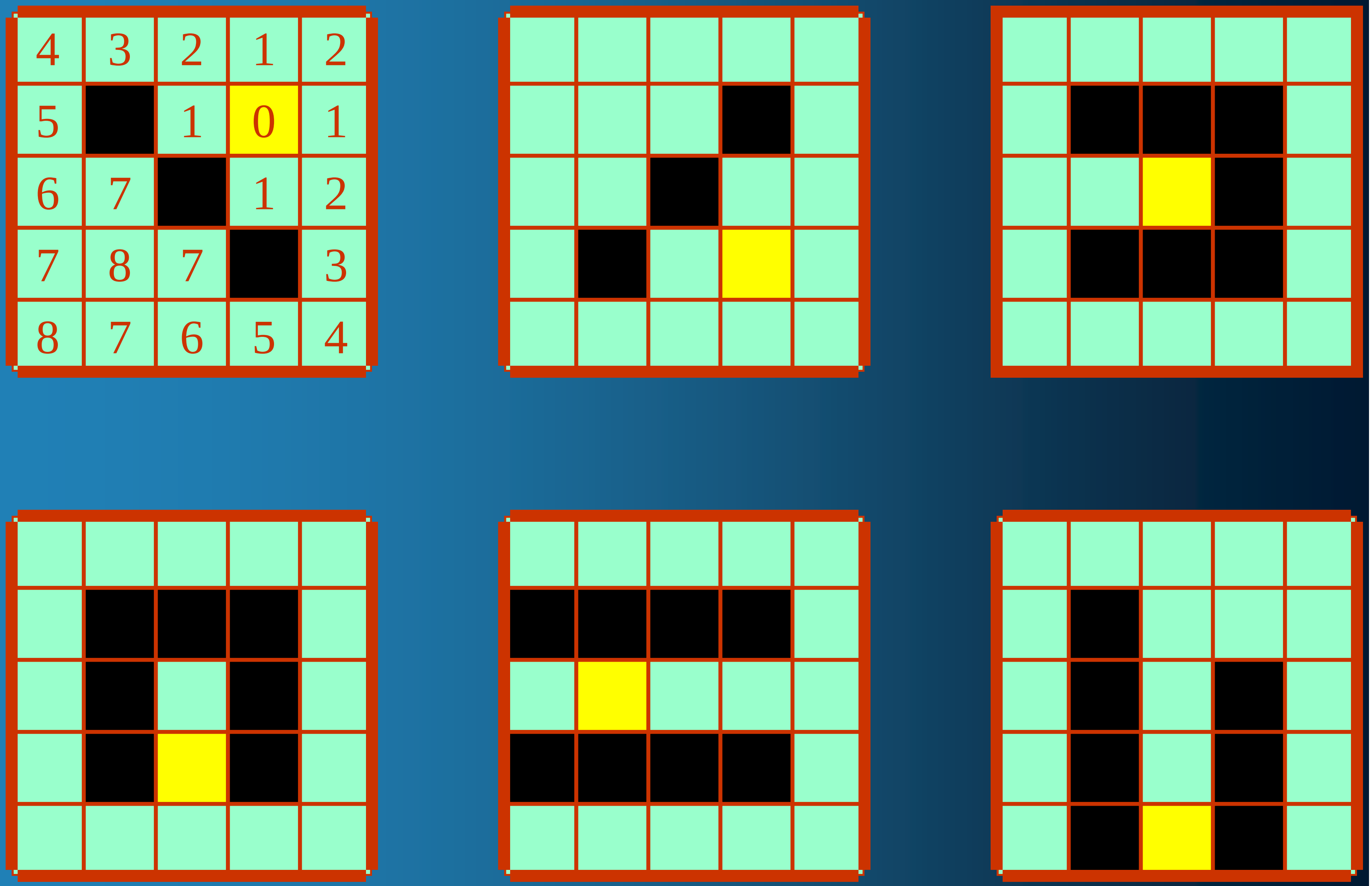
(an N by N array)

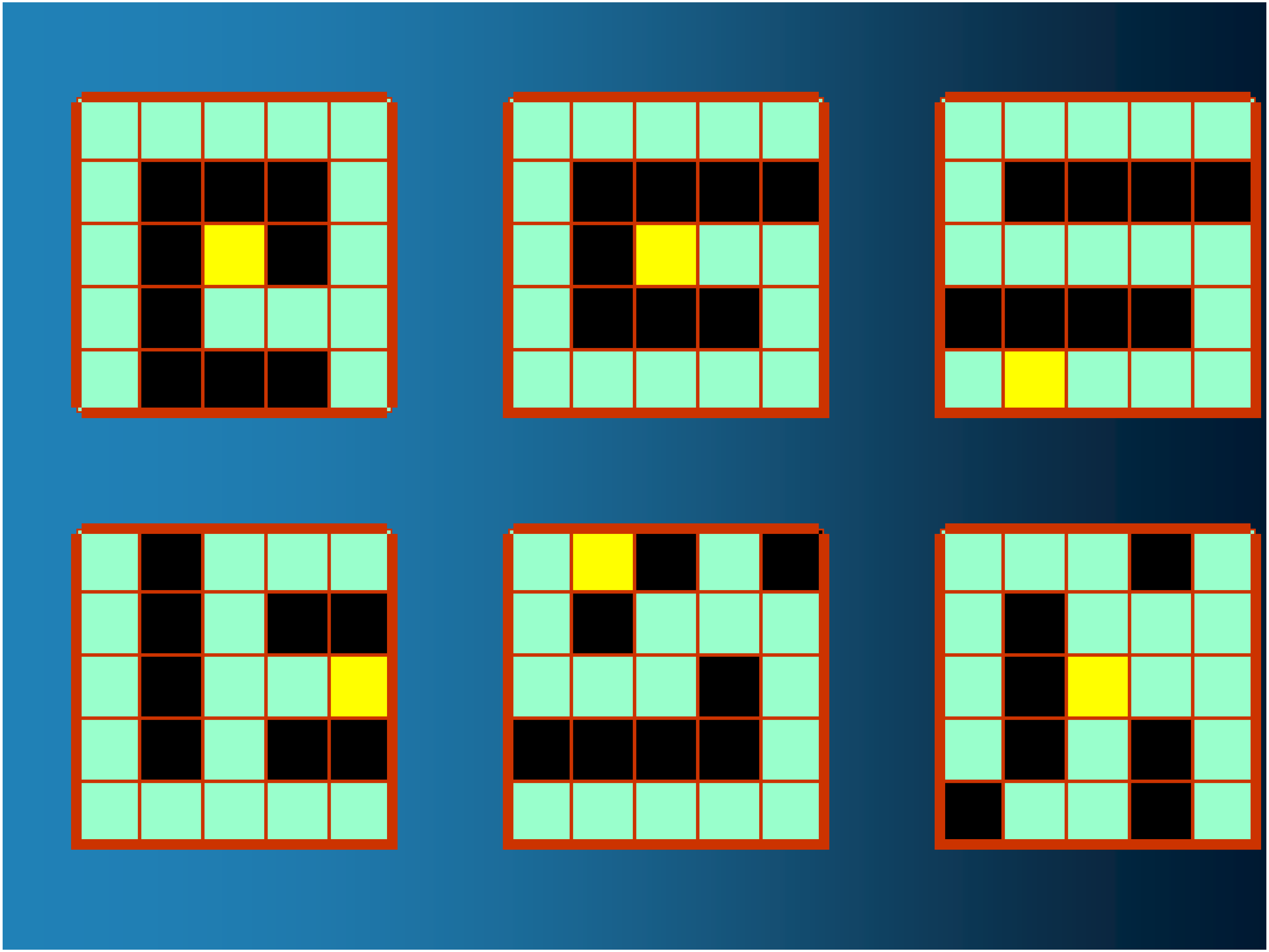
NETWORK

Maze Description

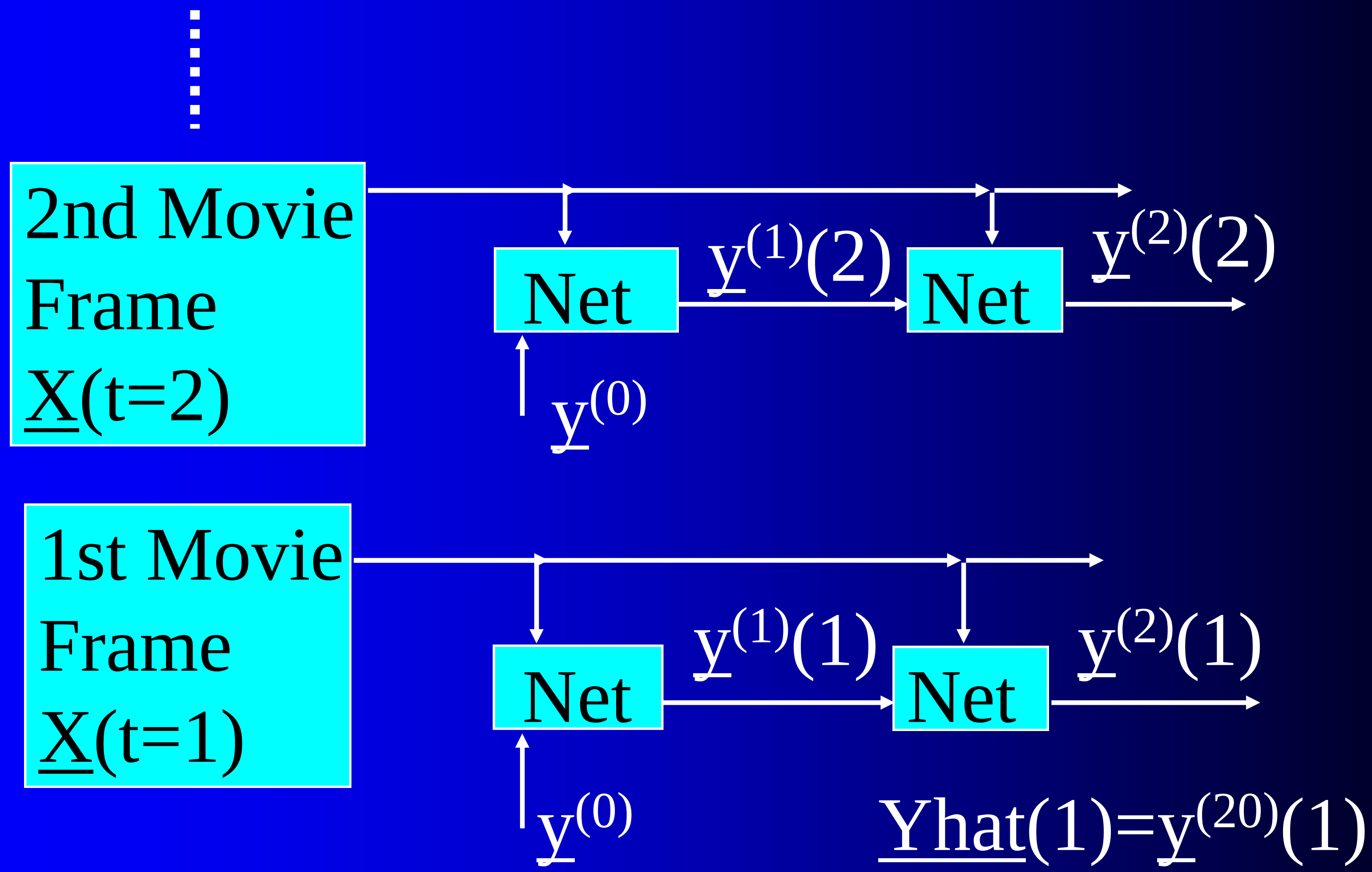
- Obstacle (ix,iy) all ix,iy
- Goal (ix,iy) all ix,iy

At arXiv.org, nlin-sys, see adap-org 9806001

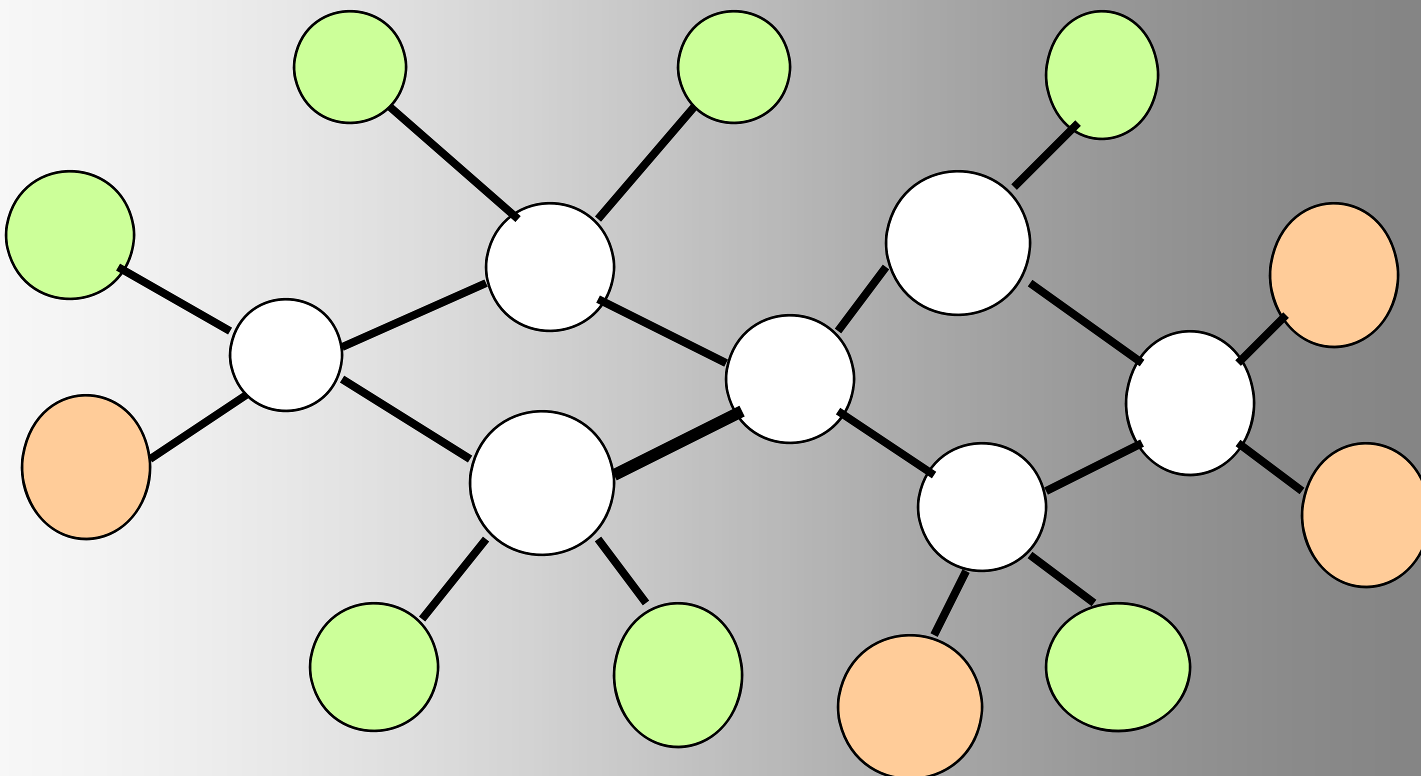

4 3 2 1 2
5 1 0 1
6 7 1 2
7 8 7 3
8 7 6 5 4



IDEA OF SRN: TWO TIME INDICES t vs. n



ANN to I/O From Idealized Power Grid



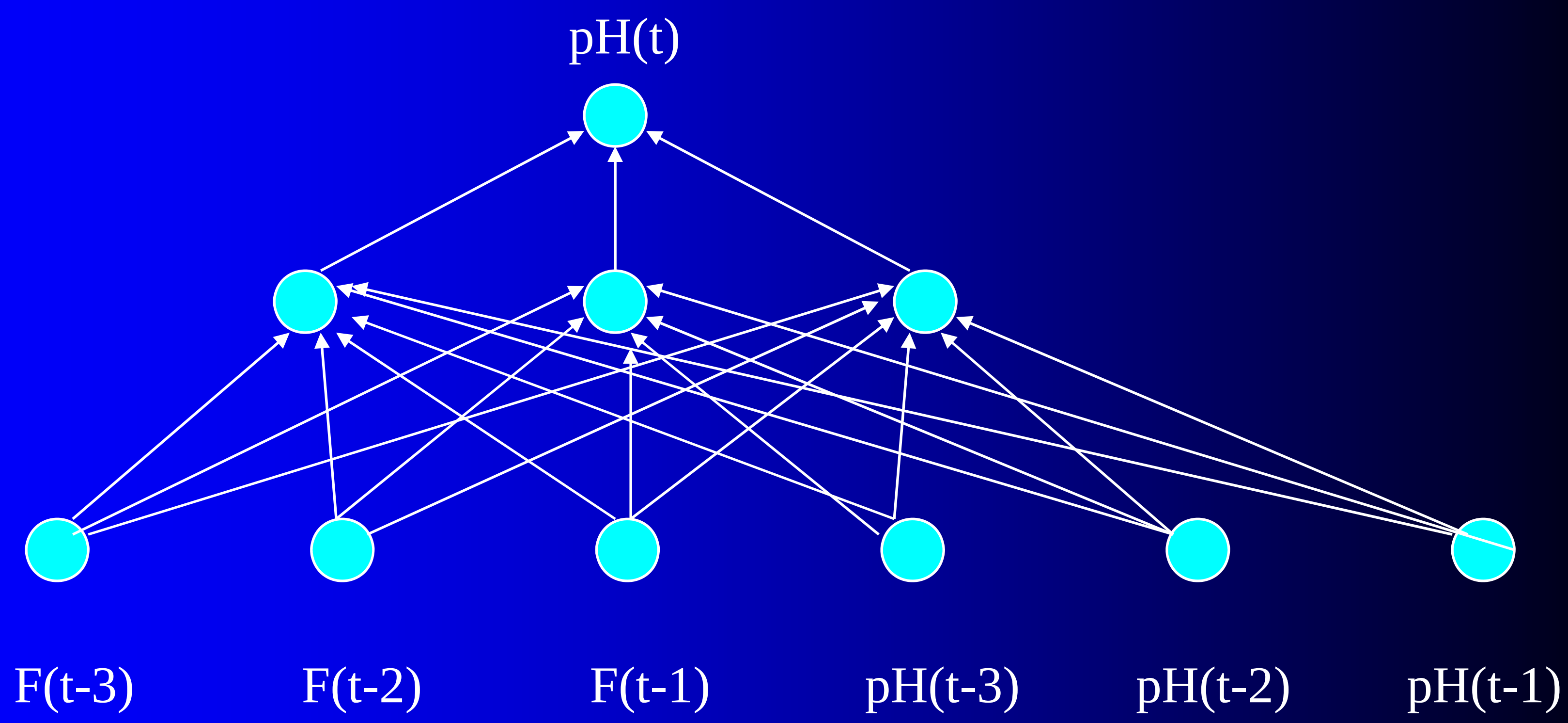
- 4 General Object Types (busbar, wire, G, L)
- Net should allow arbitrary number of the 4 objects
- How design ANN to input and output FIELDS -- variables like the SET of values for current ACROSS all objects?

Training: Brain-Style Prediction Is NOT Just Time-Series Statistics!

- One System does it all -- not just a collection of chapters or methods
- Domain-specific info is 2-edged sword:
  - need to use it; need to be able to do without it
- Neural Nets demand/inspire new work on general-purpose prior probabilities and on dynamic robustness (See HIC chapter 10)
- SEDP&Kohonen: general nonlinear stochastic ID of partially observed systems

Three Approaches to Prediction

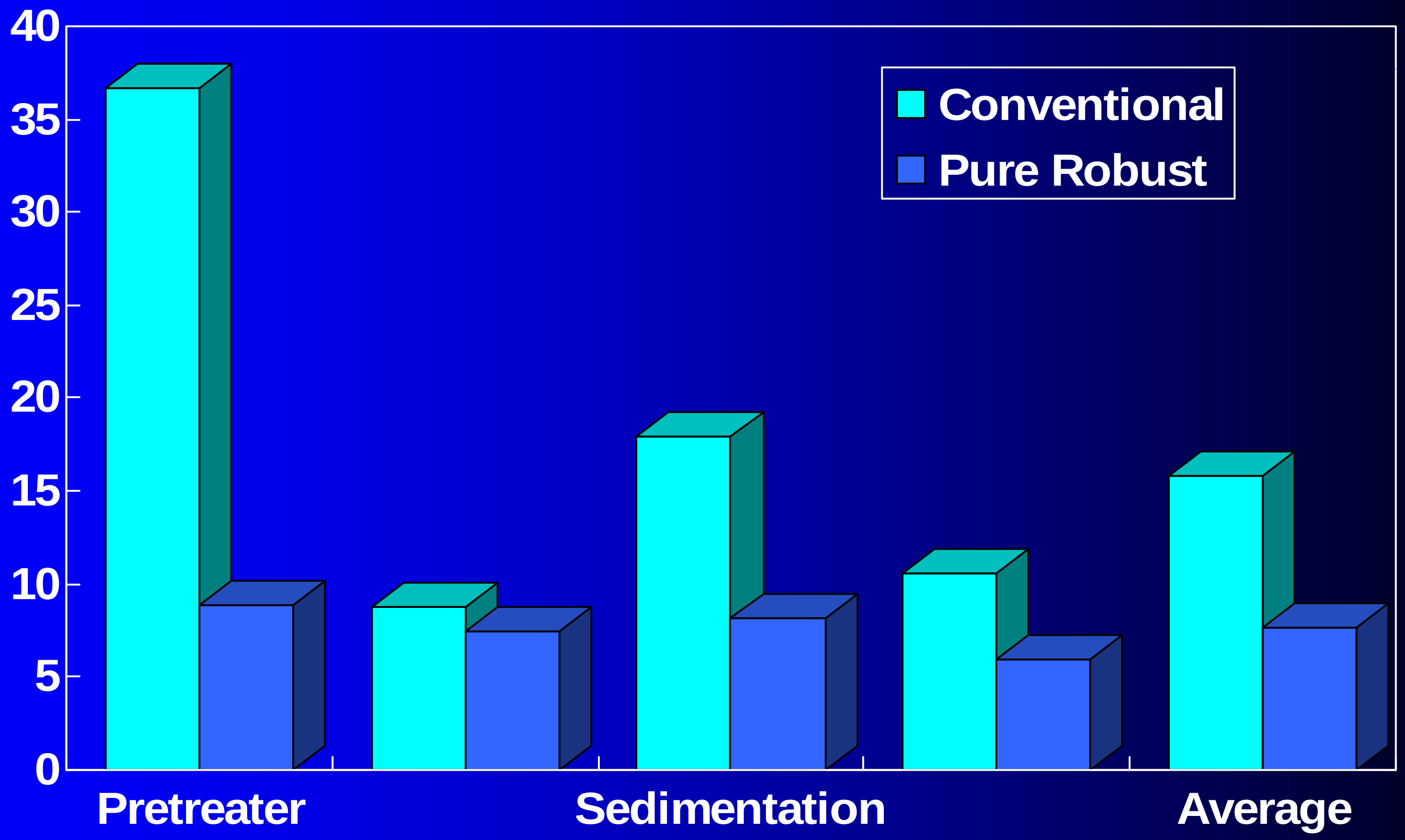
- Bayesian: Maximize Pr(Model|data)
  - "Prior probabilities" essential when many inputs
- Minimize "bottom line" directly
  - Vapnik: "empirical risk" static SVM and "sytructural risk" error bars around same like linear robust control on nonlinear system
  - Werbos '74 thesis: "pure robust" time-series
- Reality: Combine understanding and bottom line.
  - Compromise method (Handbook)
  - Model-based adaptive critics
- Suykens, Land????



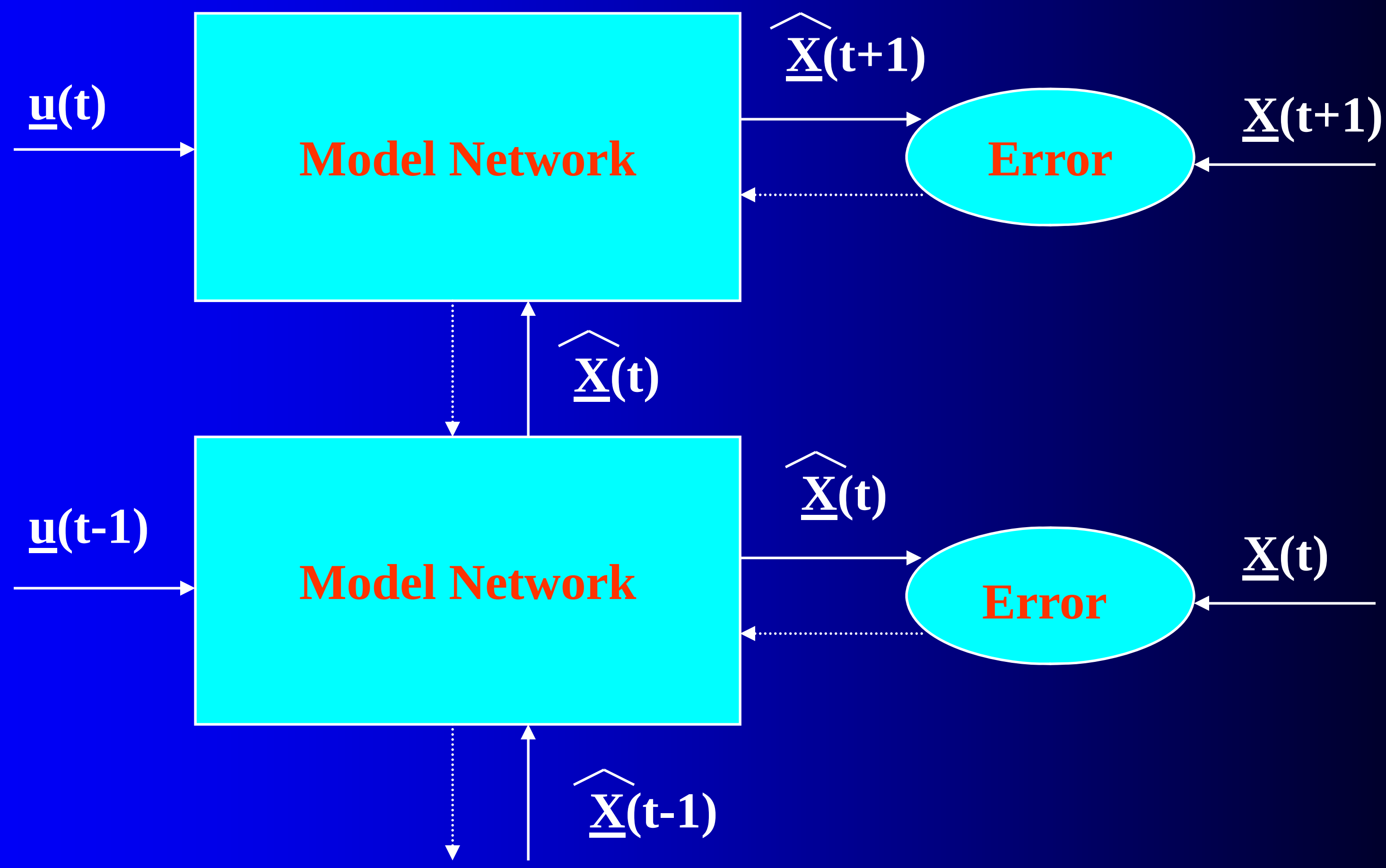
Example of TDNN used in HIC, Chapter 10

TDNNs learn NARX or FIR Models, not NARMAX or IIR

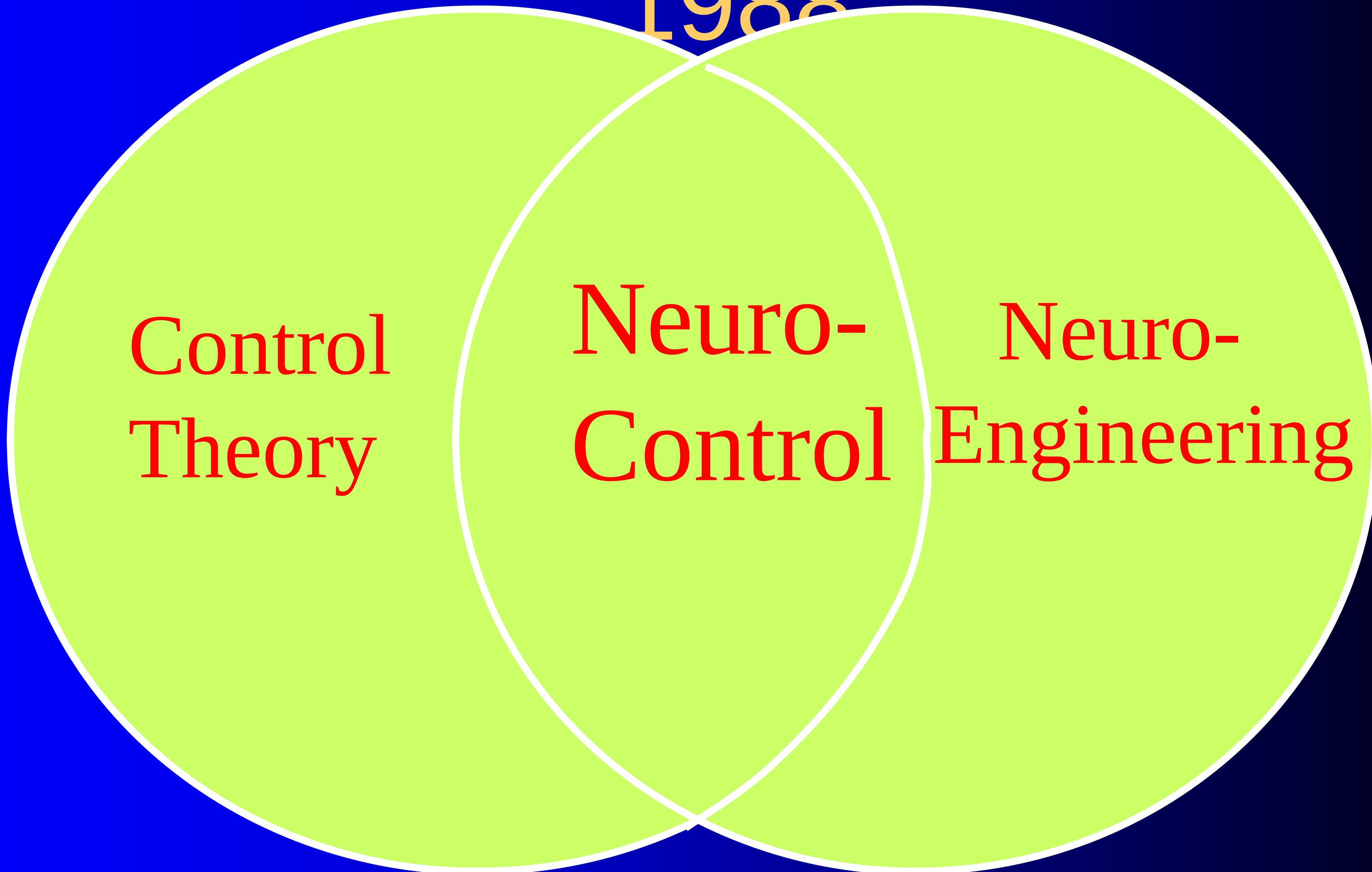
Prediction Errors (HIC p.319)



PURE ROBUST METHOD



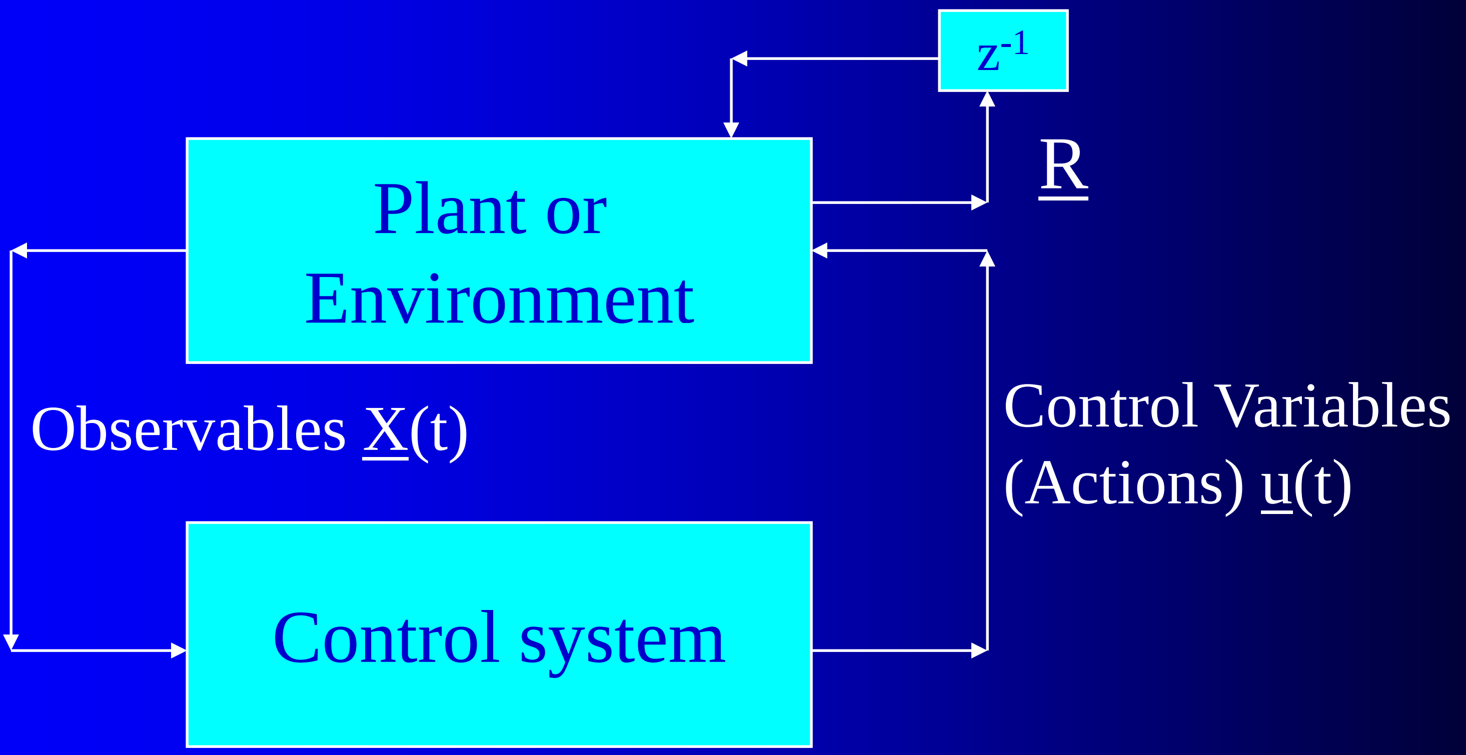
NSF Workshop Neurocontrol 1988



Miller, Sutton, Werbos, MIT Press, 1990

Neurocontrol is NOT JUST Control Theory!

What Is Control?



- t may be discrete (0, 1, 2, ...) or continuous
- "Decisions" may involve multiple time scales

Major Choices In Control (A Ladder)

•SISO (old) versus. MIMO (modern & CI)

•Feedforward versus Feedback

•Fixed versus Adaptive versus Learning

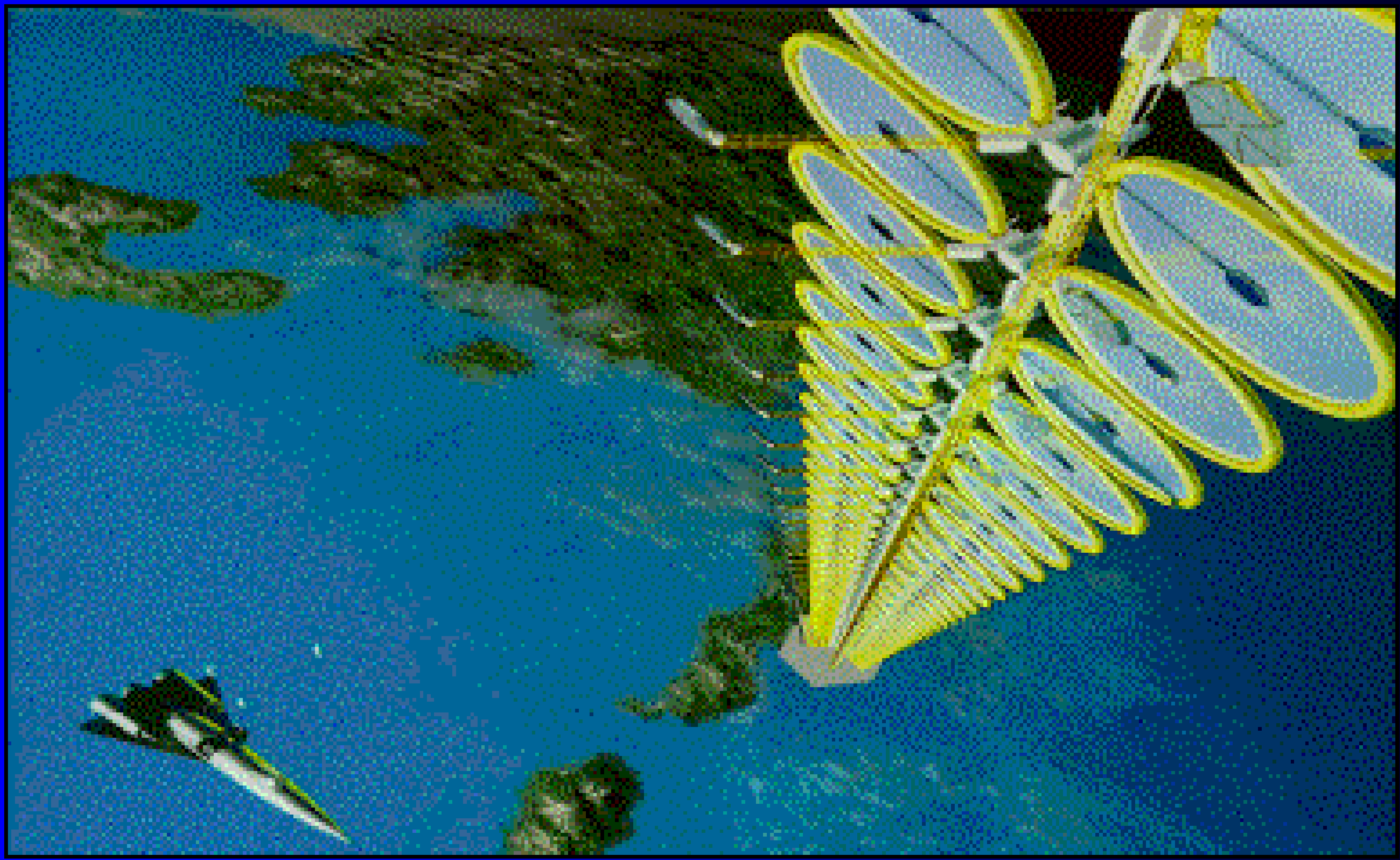
— e.g <u>learn</u> to <u>adapt</u> to changing road traction

•Cloning versus Tracking versus Optimization

3 Design Approaches/Goals/Tasks

- CLONING: Copy Expert or Other Controller
  - — What the Expert Says (Fuzzy or AI)
  - — What the Expert Does (Prediction of Human)
- TRACKING: Set Point or Reference Trajectory
  - — 3 Ways to Stabilize; To Be Discussed
- OPTIMIZATION OVER TIME
  - — n-step Lookahead vs. LQG (Stengel, Bryson/Ho)
  - — vs. Approximate Dynamic Programming (Werbos)

NSF-NASA Workshop on Learning/Robotics For Cheaper (Competitive) Solar Power



See NSF 02-098 at www.nsf.gov &URLs

Human mentors robot and then robot improves skill

Schaal, Atkeson
NSF ITR project

Learning allowed robot to quickly learn to imitate human, and then improve agile movements (tennis strokes). Learning many agile movements quickly will be crucial to enabling >80% robotic assembly in space.

Three Ways To Get Stability

- Robust or H Infinity Control
  (Oak Tree)
- Adaptive Control (Grass)
- Learn Offline/Adaptive Online
  (Maren 90)
  - "Multistreaming" (Ford, Felkamp et al)
  - Need TLRN Controller, Noise Wrapper
  - ADP Versions: Online or "Devil Net"

Example from Hypersonics: Parameter Ranges for Stability (H

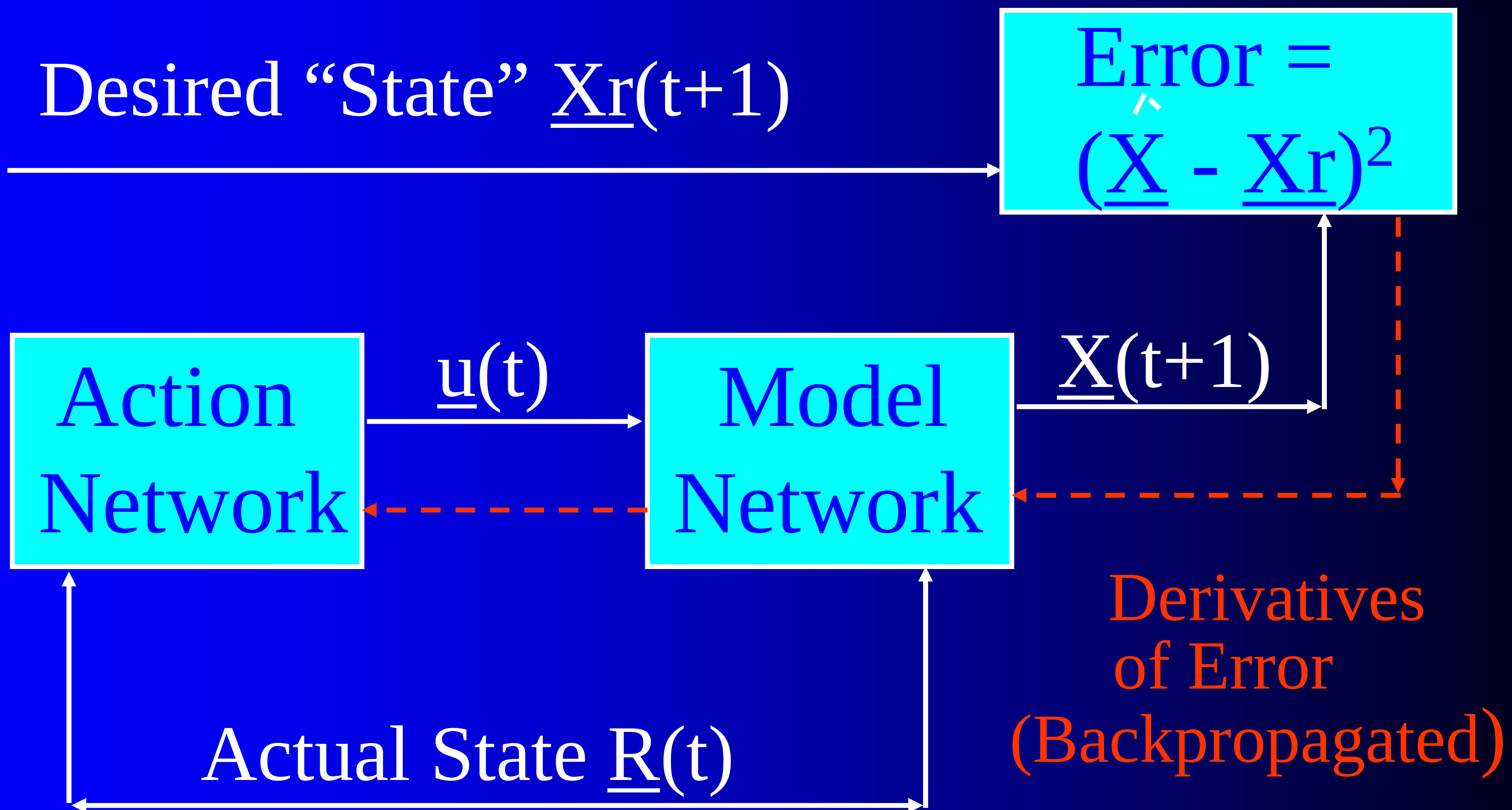
θ_2

θ_1

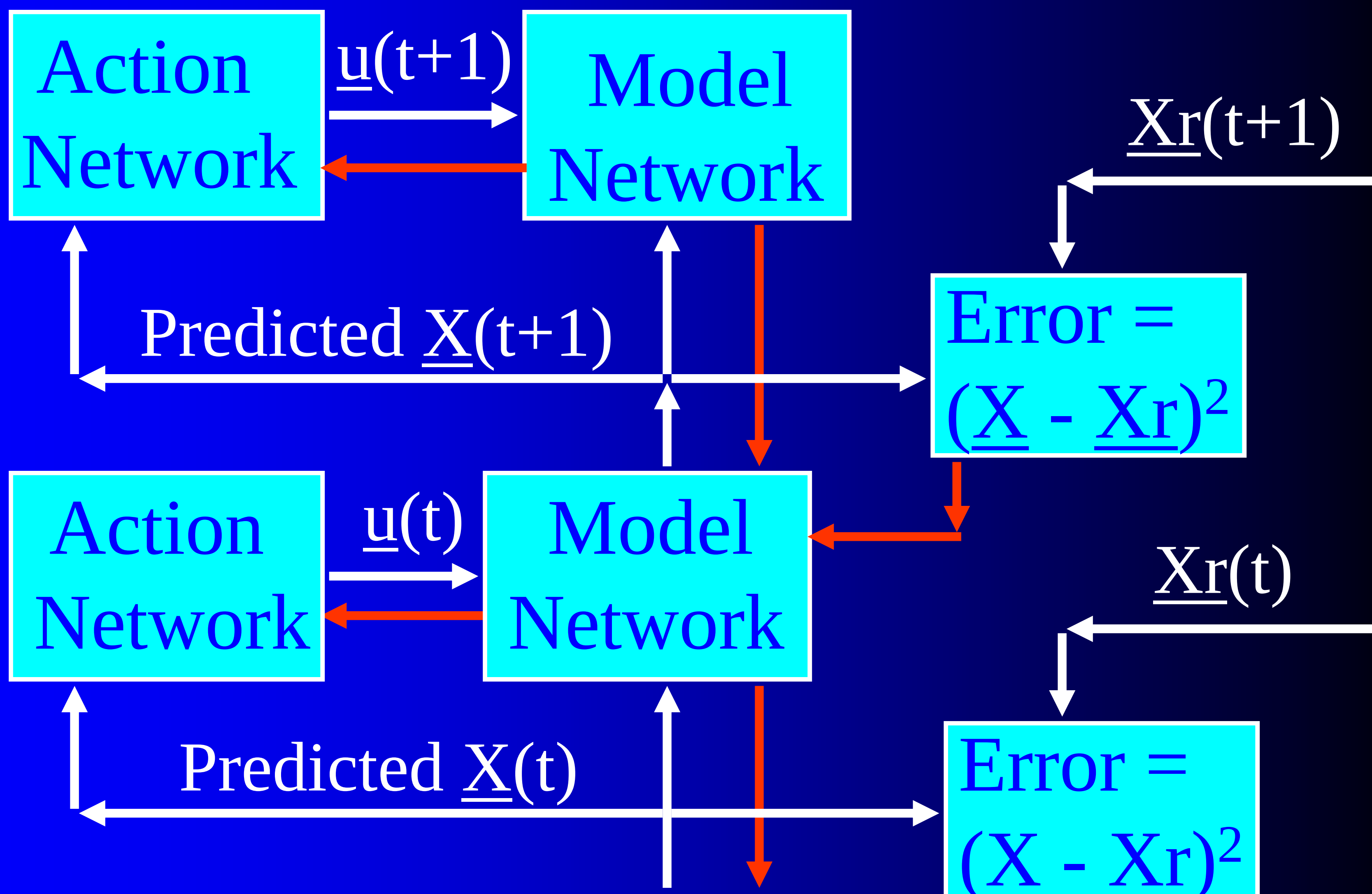
Center of Gravity at 12 Meters

Center of Gravity at 11.3 Meters

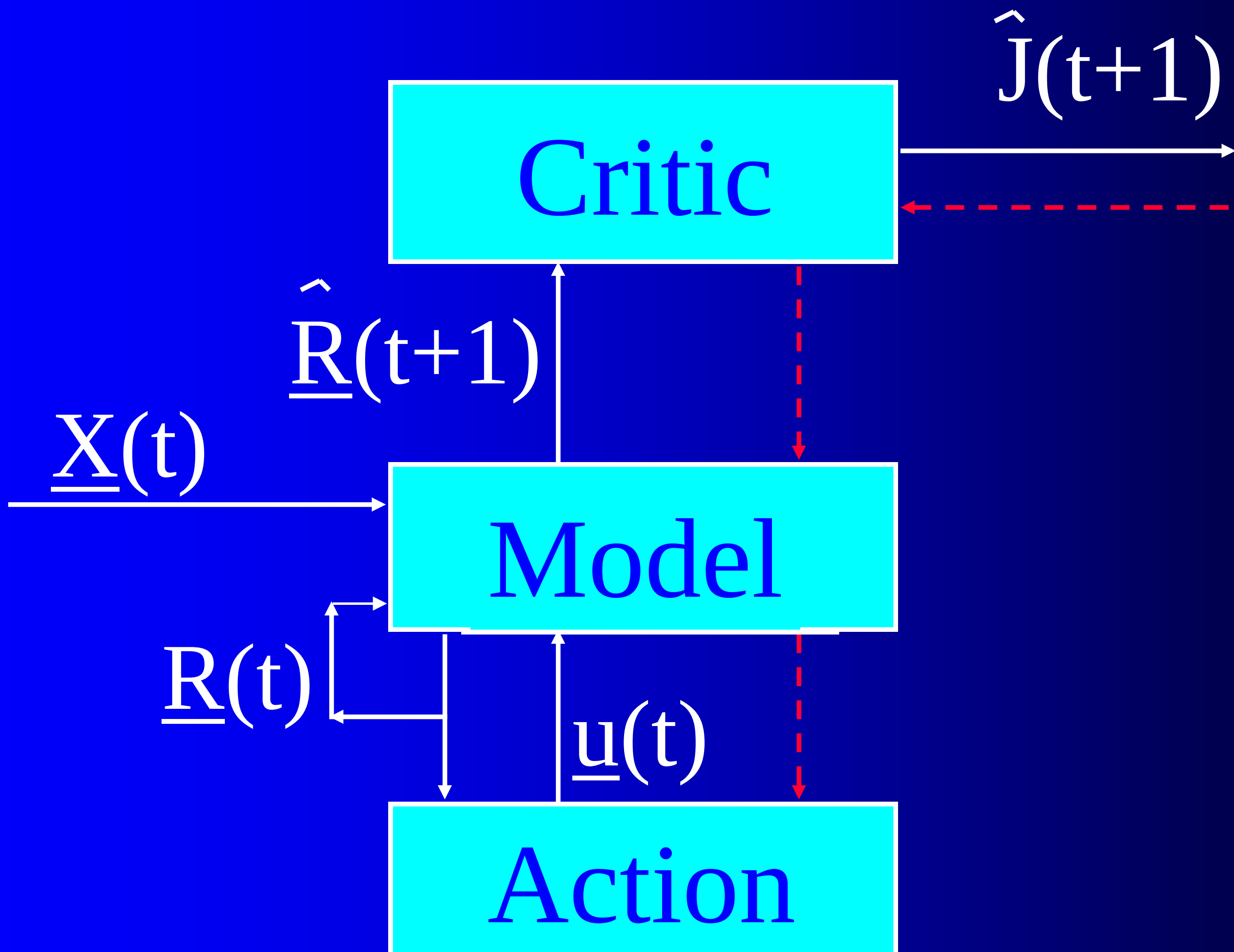
Idea of Indirect Adaptive Control



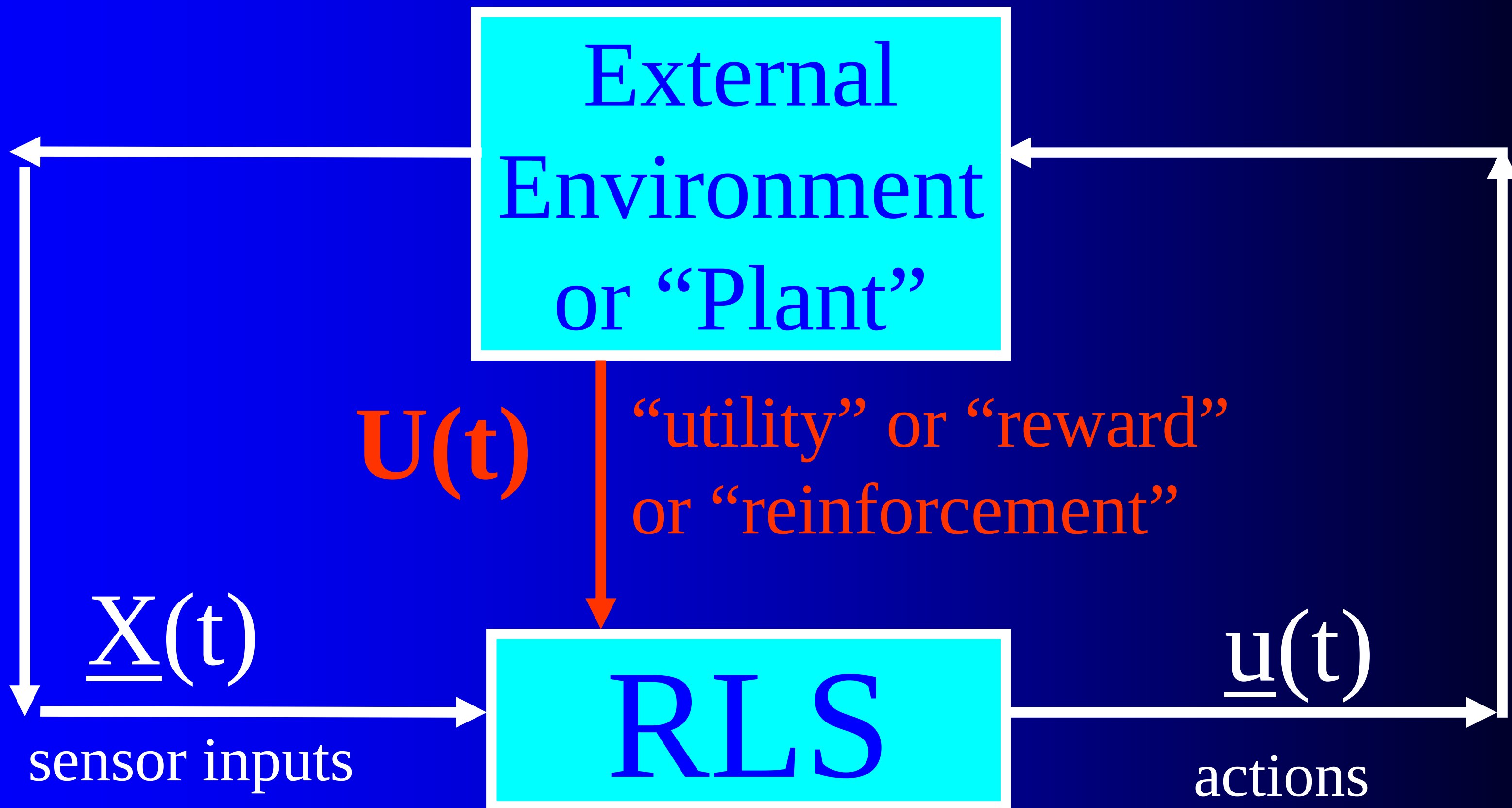
Backpropagation Through Time (BTT) for Control (Neural MPC)



Level 3 (HDP+BAC) Adaptive Critic System



Reinforcement Learning Systems (RLS)



RLS may have internal dynamics and "memory" of earlier times t-1, etc.

Maximizing utility over time

Model of reality

Utility function U

Dynamic programming

$$J(\mathbf{x}(t)) = \mathop{\mathbf{Max}}_{\mathbf{u}(t)} \left\langle U(\mathbf{x}(t), \mathbf{u}(t)) + J(\mathbf{x}(t+1)) \right\rangle / (1+r)$$

Secondary, or strategic utility function J

Beyond Bellman: Learning & Approximation for Optimal Management of Larger Complex Systems

- Basic thrust is scientific. Bellman gives exact optima for 1 or 2 continuous state vars. New work allows 50-100 (thousands sometimes). Goal is to scale up in space and time -- the math we need to know to know how brains do it. And unify the recent progress.
- Low lying fruit -- missile interception, vehicle/engine control, strategic games
- New book from ADP02 workshop in Mexico www.eas.asu.edu/~nsfadp (IEEE Press, 2004, Si et al eds)

Emerging Ways to Get Closer to Brain-Like Systems

- IEEE Computational Intelligence (CI) Society, new to 2004, about 2000 people in meetings.
- Central goal: "end-to-end learning" from sensors to actuators to maximize performance of plant over future, with general-purpose learning ability.
- This is DARPA's "new cogno" in the new nano-info-bio-cogno convergence
- This is end-to-end cyberinfrastructure
  - See hot link at bottom of www.eng.nsf.gov/ecs
- What's new is a path to make it real

4 Types of Adaptive Critics

- Model-free (levels 0-2)*
  - Barto-Sutton-Anderson (BSA) design, 1983
- Model-based (levels 3-5)*
  - Werbos Heuristic dynamic programming with backpropagated adaptive critic, 1977, Dual heuristic programming and Generalized dual heuristic programming, 1987
- Error Critic (TLRN, cerebellum models)
- 2-Brain, 3-Brain models

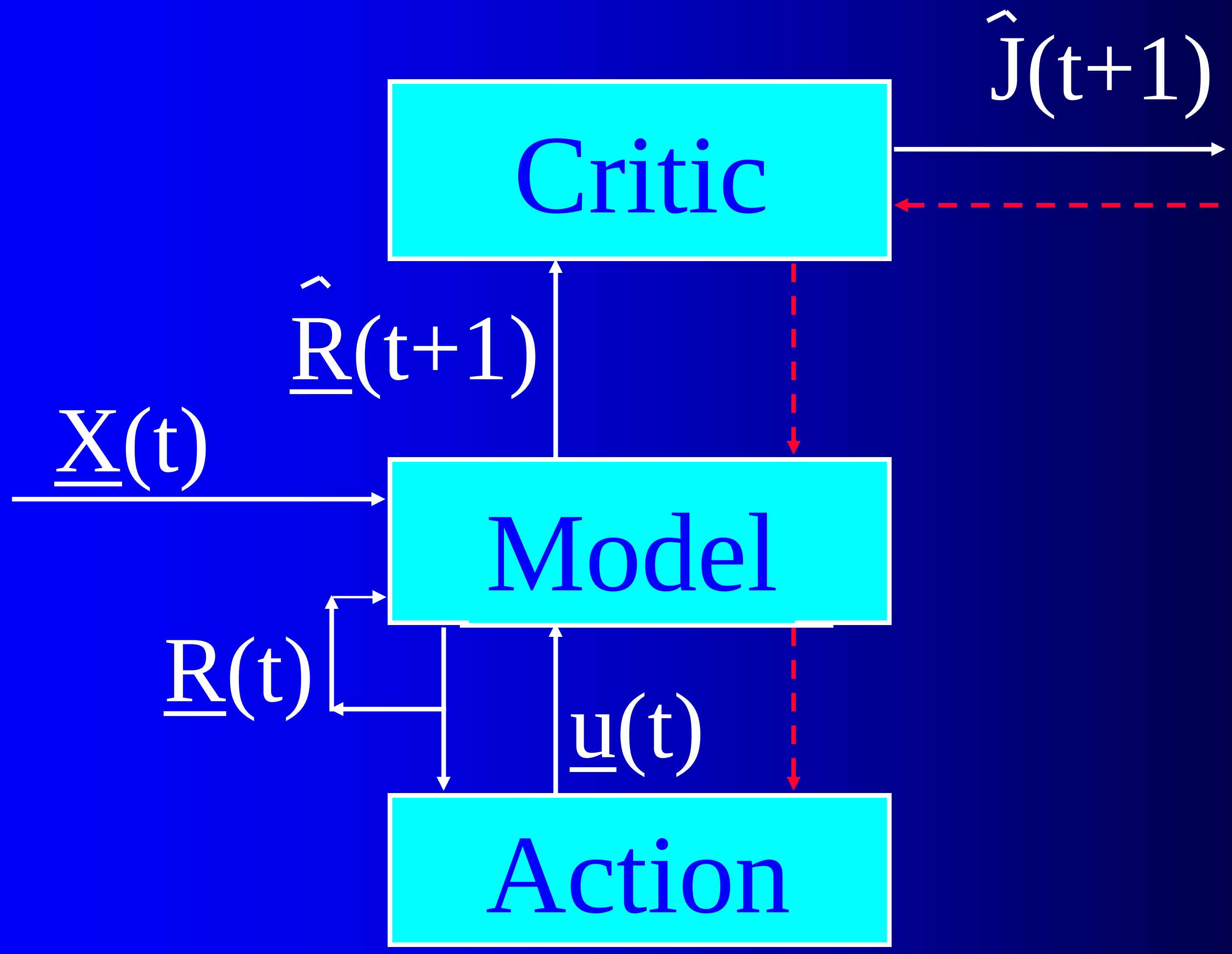
Beyond Bellman: Learning & Approximation for Optimal Management of Larger Complex Systems

- Basic thrust is scientific. Bellman gives exact optima for 1 or 2 continuous state vars. New work allows 50-100 (thousands sometimes). Goal is to scale up in space and time -- the math we need to know to know how brains do it. And unify the recent progess.
- Low lying fruit -- missile interception, vehicle/engine control, strategic games
- Workshops: ADP02 in Mexico ebrains.la.asu.edu/~nsfadp; coordinated workshop on anticipatory optimization for power.

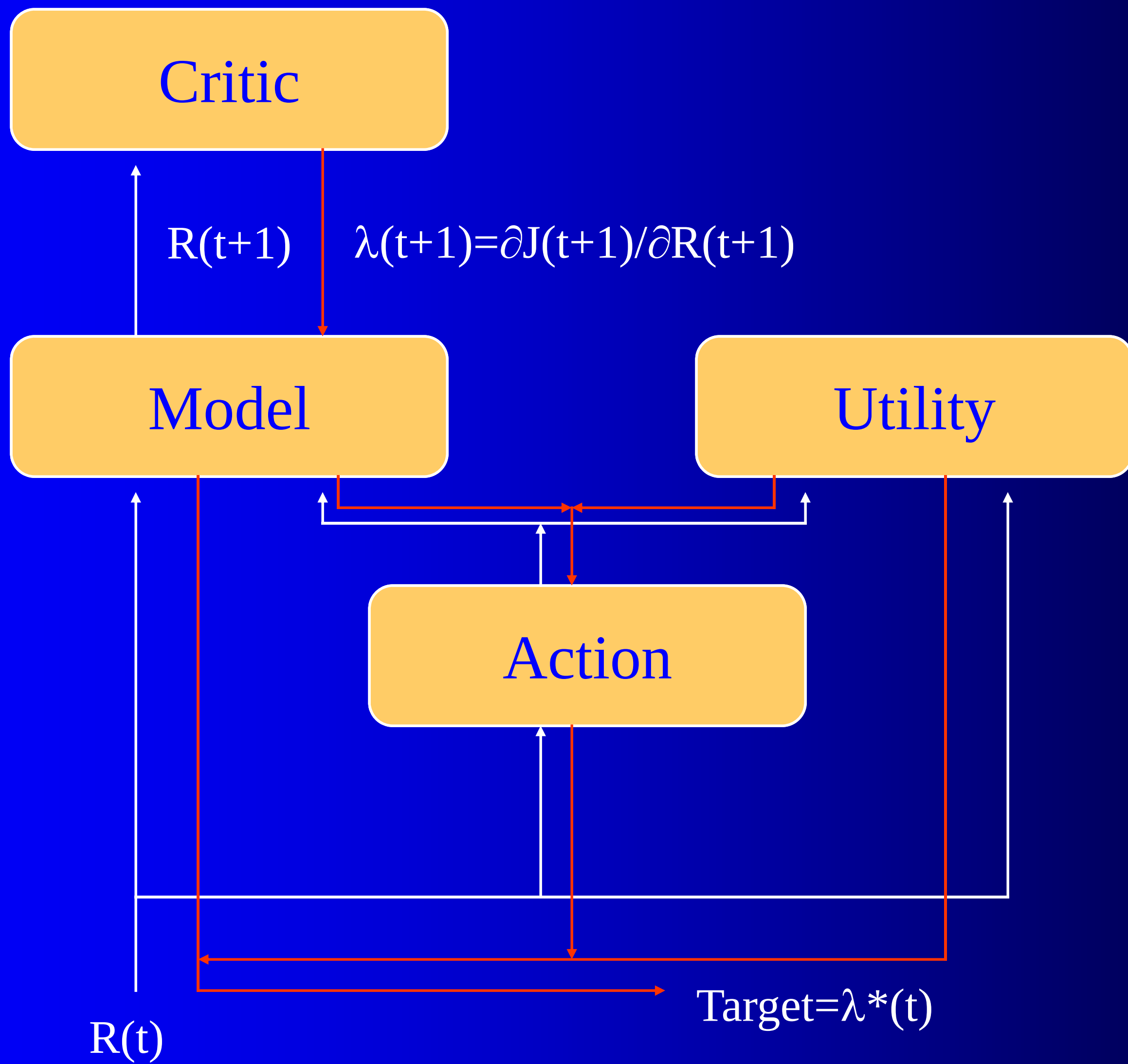
New Workshop on ADP: text/notes at www.eas.asu.edu/~nsfadp

- Neural Network Engineering
  - Widrow 1st ‘Critic’ (‘73), Werbos ADP/RL (‘68-’87)
  - Wunsch, Lendaris, Balakrishnan, White, Si,LDW......
- Control Theory
  - Ferrari/Stengel (Optimal), Sastry, Lewis, VanRoy (Bertsekas/Tsitsiklis),Nonlinear Robust...
- Computer Science/AI
  - Barto et al (‘83), TD, Q, Game-Playing, ...
- Operations Research
  - Original DP: Bellman, Howard; Powell
- Fuzzy Logic/Control
  - Esogbue, Lendaris, Bien

Level 3 (HDP+BAC) Adaptive Critic System



Dual Heuristic Programming (DHP)



Don Wunsch, Texas Tech ADP Turbogenerator Control CAREER 9702251, 9704734, etc.

- Stabilized voltage & reactance under intense disturbance where neuroadaptive & usual methods failed
- Being implemented in full-scale experimental grid in South Africa
- Best paper award IJCNN99

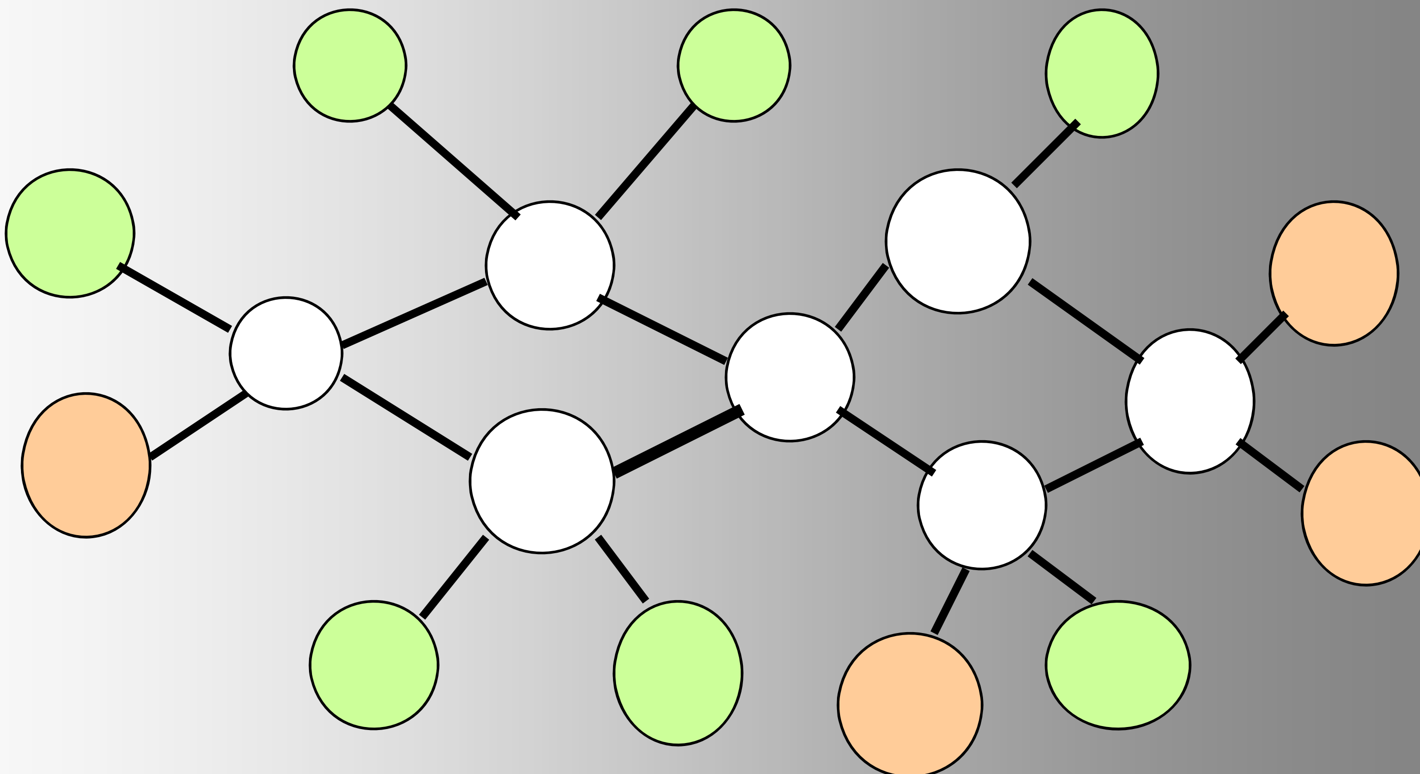
Uses of the Main Critic Designs

- HDP=TD For DISCRETE set of Choices
- DHP when action variables <u>u</u> are continuous
- GDHP when you face a mix of both (but put zero weight on undefined derivative)
- See arXiv. org , nlin-sys area, adap-org 9810001 for detailed history, equation,

From Today's Best ADP to True (Mouse-)Brain-Like Intelligence

- ANNs For Distributed/Network I/O: "spatial chunking," ObjectNets, Cellular SRNs
- Ways to Learn Levels of a Hierarchical Decision System – Goals, Decisions
- "Imagination" Networks, which learn from domain knowledge how to escape local optima (Brain-Like Stochastic Search BLiSS)
- Predicting True Probability Distributions

ANN to I/O From Idealized Power Grid



- 4 General Object Types (busbar, wire, G, L)
- Net should allow arbitrary number of the 4 objects
- How design ANN to input and output FIELDS -- variables like the SET of values for current ACROSS all objects?

Simple Approach to Grid-Grid Prediction in Feedforward (FF) Case

- Train 4 FF Nets, one for each TYPE of object, over all data on that object.
- E.g.: Predict Busbar(t+1) as function of Busbar(t) and Wire(t) for all 4 wires linked to that busbar (imposing symmetry).
- Dortmund diagnostic system uses this idea
- This IMPLICITLY defines a global FF net which inputs <u>X</u>(t) and outputs grid prediction

ObjectNets: A Recurrent Generalization (with patent)

- Define a global FF Net, FF, as the combination of local object model networks, as before
- Add an auxiliary vector, $\underline{y}$, defined as a field over the grid (just like $\underline{X}$ itself)
- The structure of the object net is an SRN:
  - $\underline{y}^{[k+1]} = FF(\ \underline{X}(t), \underline{y}^{[k]}, W)$
  - prediction (e.g. $\underline{X}(t+1)) = g(\underline{y}^{[\infty]})$
- Train SRNs as in xxx.lanl.gov, adap-org 9806001
- General I/O Mapping -- Key to Value Functions

Four Advanced Capabilities

- ANNs For Distributed/Network I/O: “spatial chunking,” ObjectNets, Cellular SRNs
- Ways to Learn Levels of a Hierarchical Decision System
- “Imagination” Networks, which learn from domain knowledge how to escape local optima (Brain-Like Stochastic Search BLiSS)
- Predicting True Probability Distributions

Forms of Temporal Chunking

- Brute Force, Fixed “T”, Multiresolution
  - “Clock Based Synchronization”, NIST
  - e.g., in Go, predict 20 moves ahead
- Action Schemas or Task Modules
  - “Event Based Synchronization”:BRAIN
  - Miller/G/Pribram, Bobrow, Russell, me...

Lookup Table Adaptive Critics 1

$$\begin{bmatrix} U_1 & \cdots & U_N \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix}$$

$\langle U(\underline{x}) \rangle =$

SUM (over i) $U_i \, p_i$

$= U^T \underline{p}$ or $U^T \underline{x}$

Where $p_i = \Pr(x_i)$ AND $M_{ij} = \Pr(x_i(t+1) \mid x_j(t))$

Review of Lookup Table Critics 2

Bellman: $J(\underline{x}(t)) = \langle U(\underline{x}(t)) + J(x(t+1)) \rangle$

$$J^T\underline{x} = U^T\underline{x} + J^TM\underline{x}$$

$$J^T = U^T(I-M)^{-1}$$

Learning Speed of Critics...

- Usual Way: $J^{(0)} = U$, $J^{(n+1)} = U + M^T J^{(n)}$
  - After n iterations, J(t) approximates
  - $U(t) + U(t+1) + \ldots + U(t+n)$
- DOUBLING TRICK shows one can be faster: $J^T = U^T(I+M)(I+M^2)(I+M^4)\ldots$
  - After n BIG iterations, J(t) approximates
  - $U(t) + U(t+1) + \ldots + U(t+2^n)$

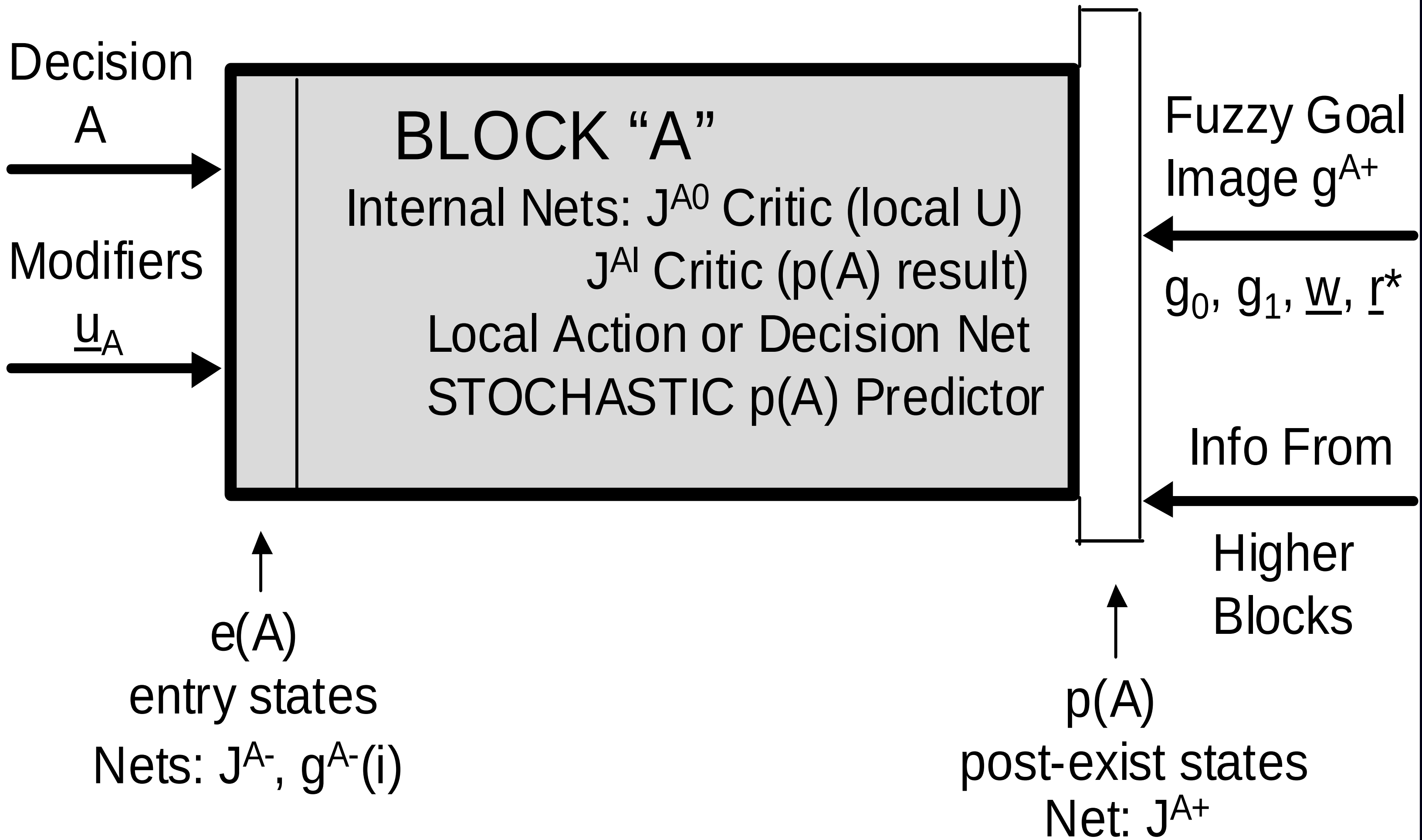
But: What if M is Sparse, Block Structured, and Big??

- M-to-the-2-to-the-nth Becomes a MESS
- Instead use the following equation, the key result for the flat lookup table case:

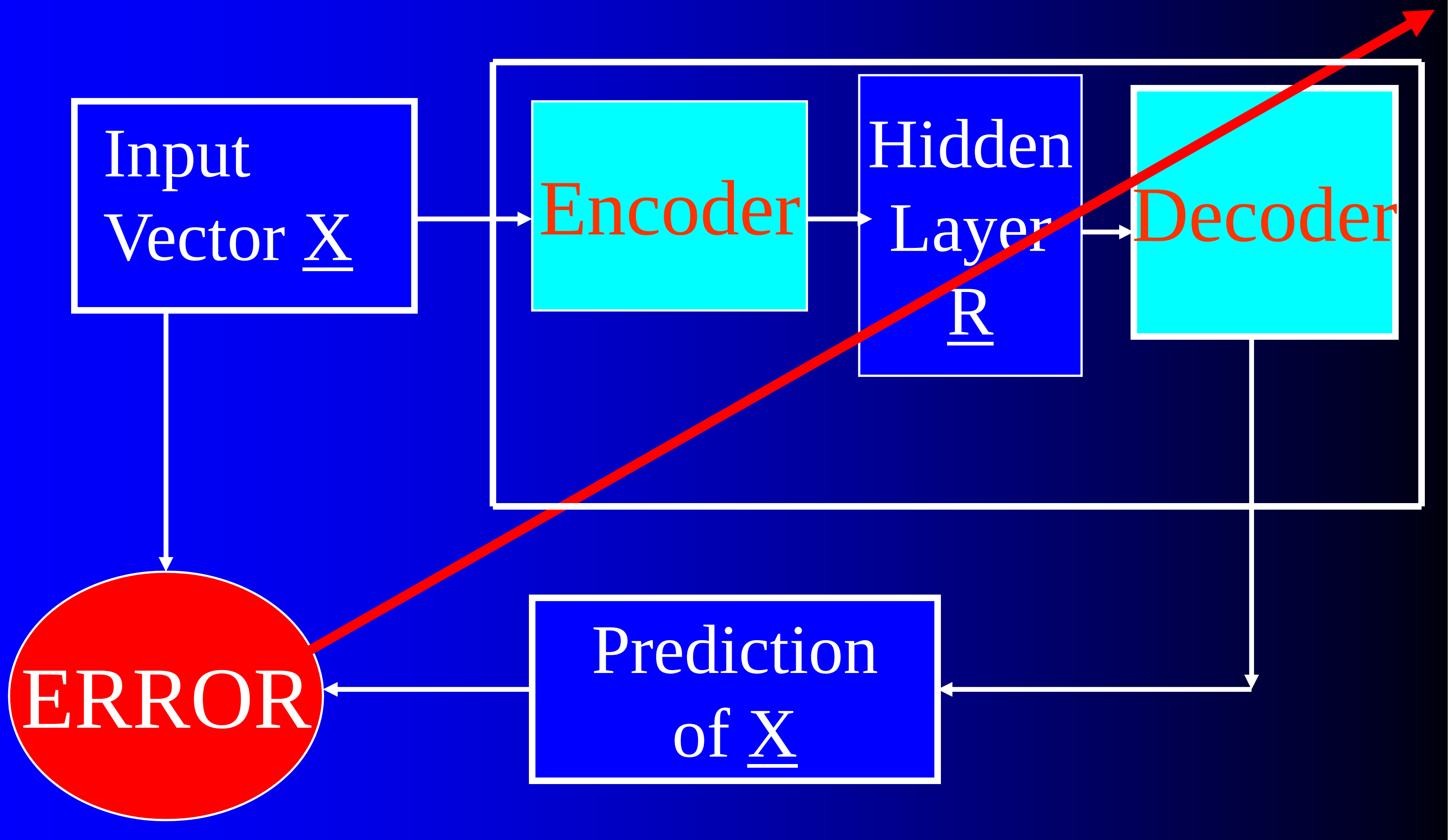
$$J_i^T = (J_i^A)^T + \text{SUM (over j in N(i))} \; J_J^T (J^B)_i^J$$

where JA represents utility within valley i before exit, and JB works back utility from the exits in New valleys j within the set of possible next valleys N(i)

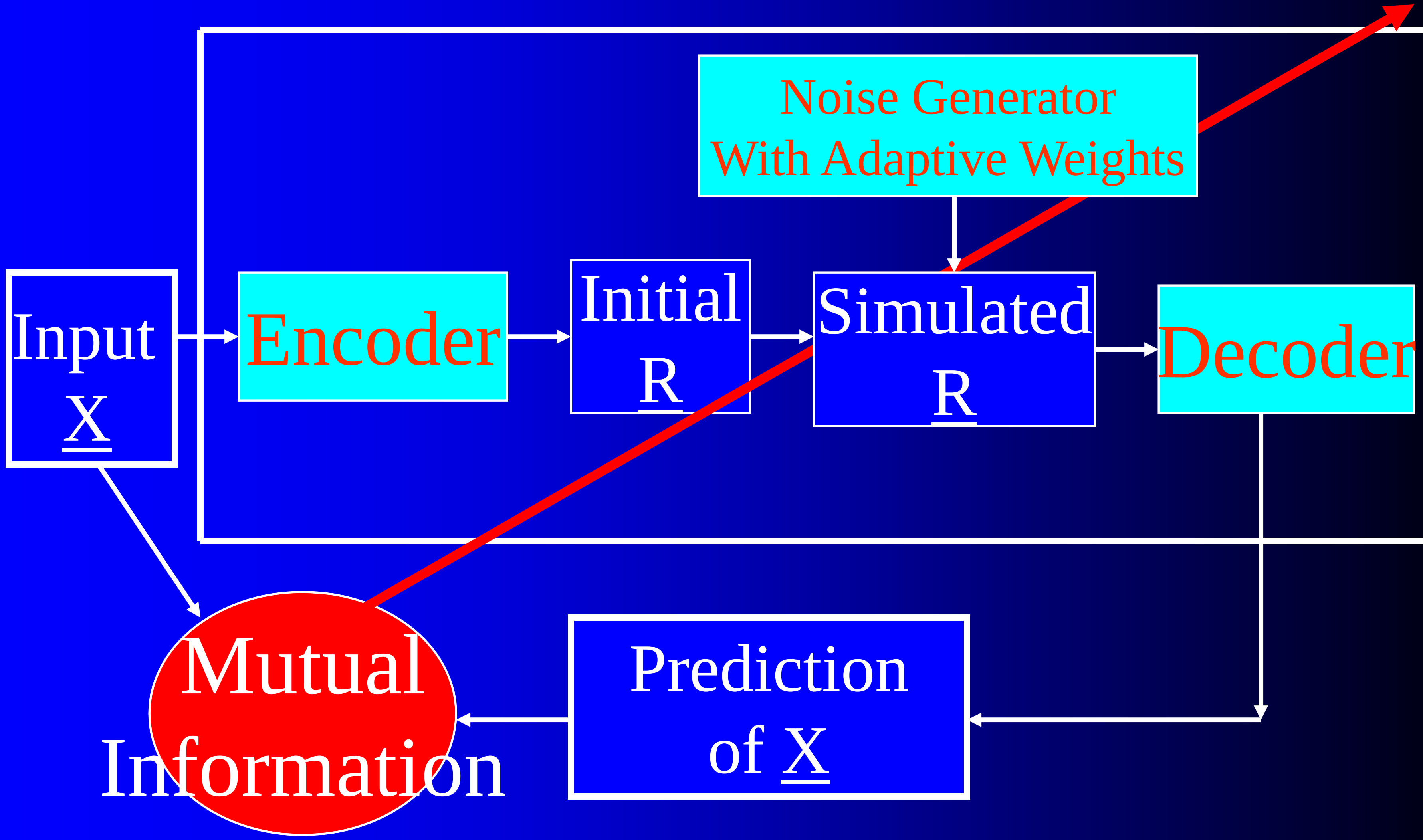
Structure of a Decision Block



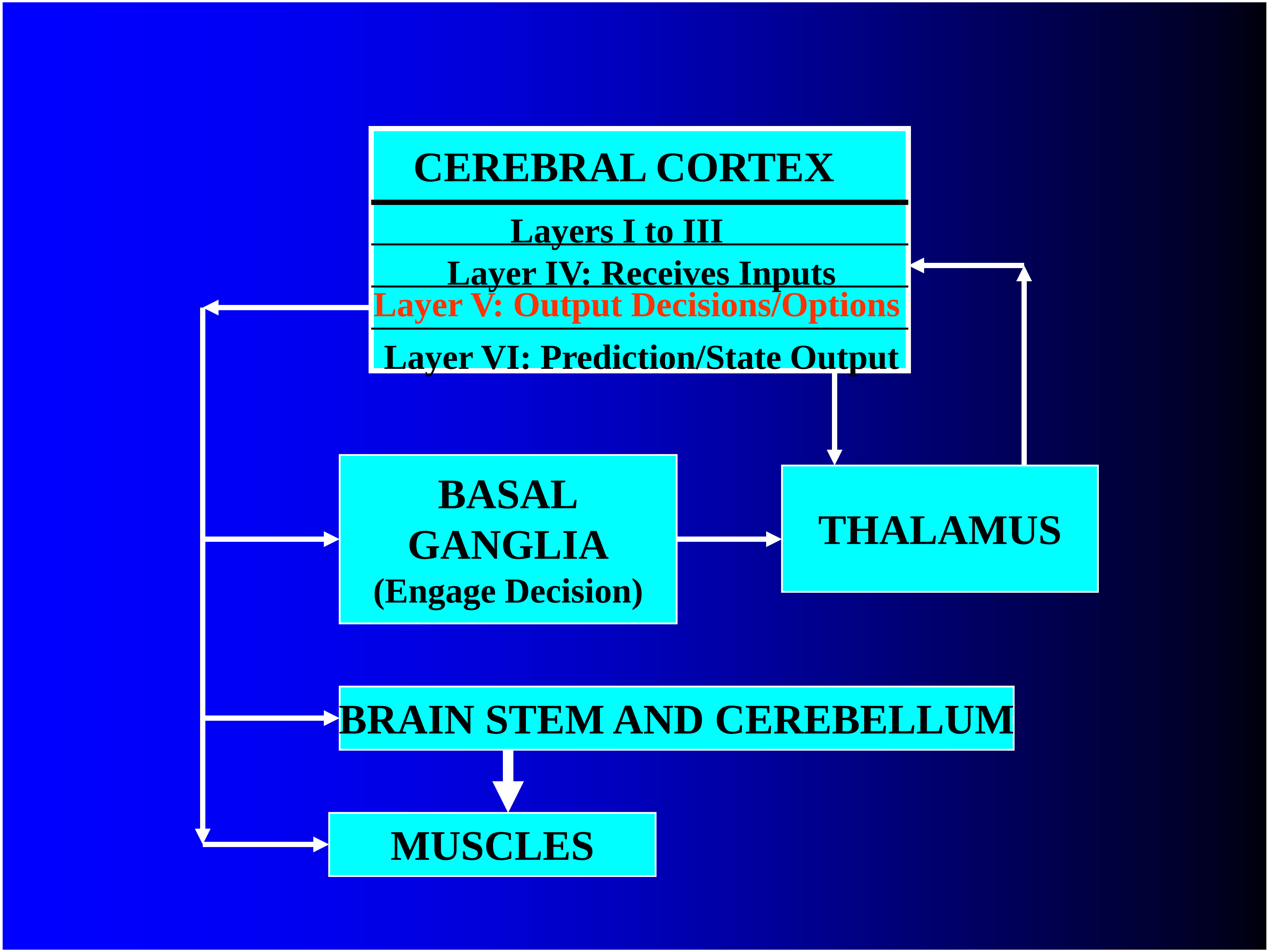
Conventional Encoder/Decoder (“PCA”)



Stochastic ED (See HIC Ch. 13)



Full Design Also Does the Dynamics Right


CEREBRAL CORTEX
Layers I to III
Layer IV: Receives Inputs
Layer V: Output Decisions/Options
Layer VI: Prediction/State Output
BASAL GANGLIA
(Engage Decision)
THALAMUS
BRAIN STEM AND CEREBELLUM
MUSCLES