Brain-Like Prediction: New Statistical Foundations for Prediction In the Face of Real-World Complexity

Paul J. Werbos

U.S. National Science Foundation

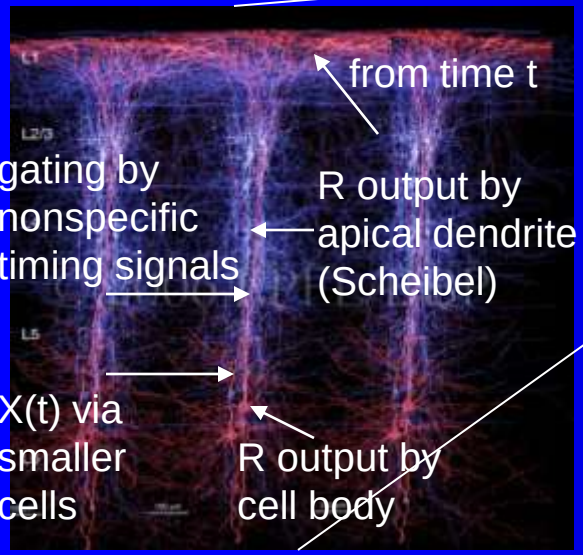
-- personal, not official, views

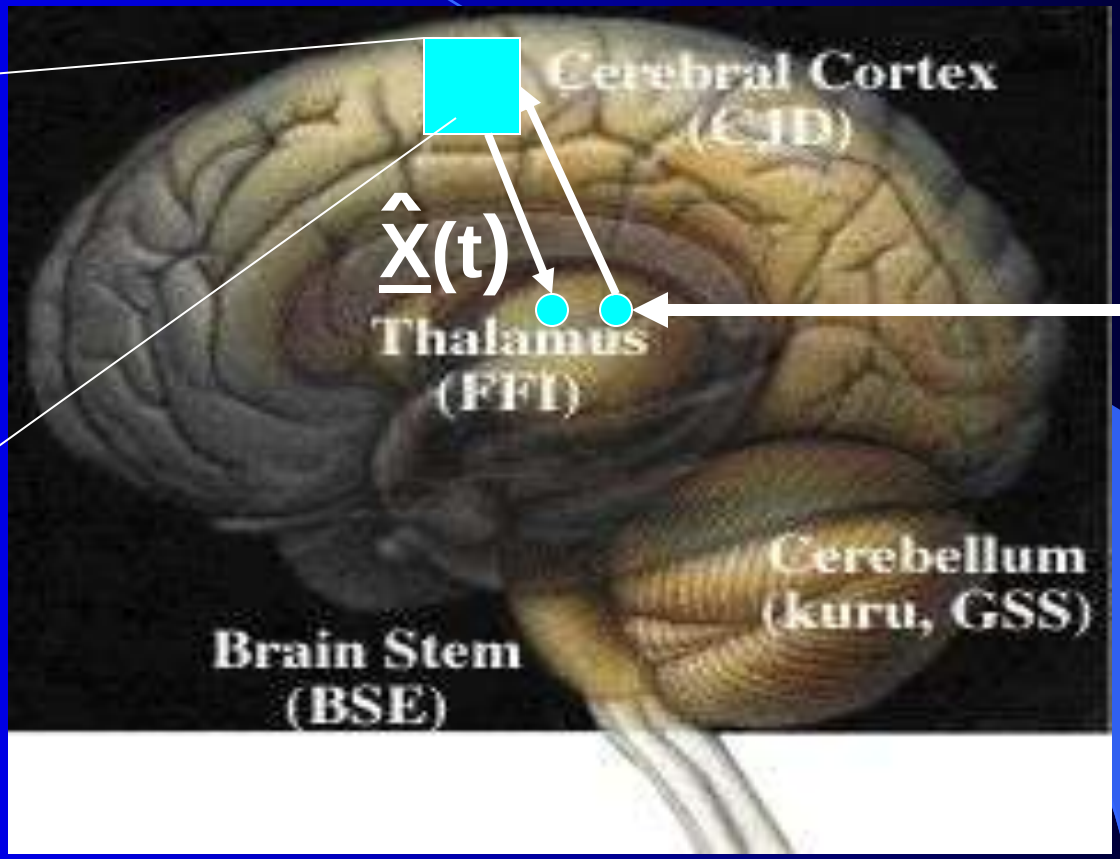
www.werbos.com, werbos@ieee.org

Outline

- What We Can Learn About Prediction From the Brain
- General Advice for Forecasting Competitions
- Dangerous Common Myths
  - Multilayer Perceptron (MLP) modeling is black magic
  - The "No Free Lunch Theorem" (not a theorem!)
  - Static Data Mining or Patterns Tell Us About Causality
  - Data-driven methods like learning cannot exploit domain knowledge
- Bayes versus Vapnik, and why dynamic robustness requires a "compromise"
- Model-Based Versus Precedent or Kernel Based Forecasting
  - **Generalize But Remember**
- Why Pr(Model) is crucial to brain ability to handle complexity

Ability to learn to "Predict Anything" Found in the Brain (Nicolelis, Chapin)





$X(t)$

(Richmond): "t+1" – t is .12 seconds. Each cycle has a forwards pass to predict, and a backwards pass to adapt

(Bliss, Spruston): found "reverse nMDA" synapse and backpropagation along dendrites

BUT: needs demonstration for more than just rat whiskers! We need "COPN2"!

What the Brain Teaches Us About Prediction

- **One universal system can learn to "predict everything**." No need for 125 different methods in 32 chapters. But "who pays for lunch"? How can it be possible?
- Can take full advantage of **massive parallel hardware** like CNN chips.
- **All predictions – including pattern recognition and memory – are in service to action**. What is true versus what is useful? It is always about "prediction of the future."
- **Incredible complexity** – learns nonlinear dynamic relations among millions of variables, based on only 10 data frames per second (300 million per year).

Question to Census Statistical Advisory Council (1979): What Principles Most Important in Building Or Understanding Such a Prediction System?

All said: They do not exist. It is impossible.
I would never use such a machine
even if I had it for free in my own lab.

Why It Was Seen As Impossible: 4 Schools of Thought in Statistics

- Probabilism ("We don't do inference. We just prove stuff.")
- Maximum Likelihood (Simplified from Jeffreys and Carnap)

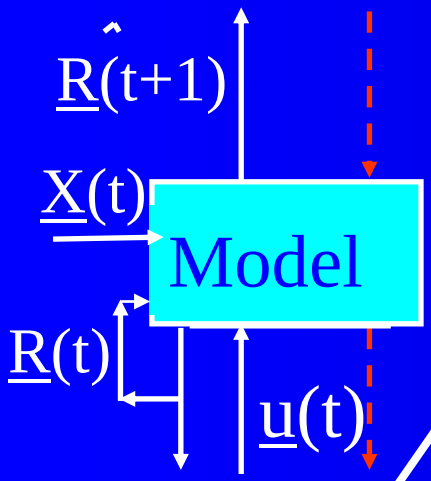
  Pr (Model & Weights W | Data) ≈ Pr (Data | Model & W)

  "Information geometry" (Rao) is basically in this group.

- Bayesian (e.g. Raiffa)
  - Sometimes Pr (Model &W | Data)

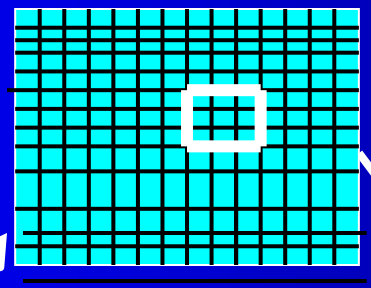
    = Pr (Data | Model & W)*Pr(Model & W)/Pr(Data)
  - Sometimes minimize utility-based loss function
- Robust statistics (Tukey, Mosteller): try to get useful results without assuming model must be true for some value of weights W

From Vector to Mammal



0. Vector Prediction (robustified SRN/TLRN) HIC Chapter 10 on web.

1. AT&T winning ZIP code recognizer and then CLION



Networks for inputs with more spatial complexity using symmetry – CSRN, ObjectNets, ….

Predicts What Will Happen Over Multiple Time Intervals Harmonized

2. reptile

Predicts and "Imagines The Possibilities" (Stochastic y=**f**(**X**,**e**). HIC Chapter 13 on web.)

3. Mouse

M. E. Bitterman (Scientific American 1969; Science later): Mouse learns to predict better in stochastic pattern recognition tasks, where turtles just slowly "go crazy." Cut mouse cortex, get turtle behavior. But reptile probably has stochastic capability, just not well-integrated.

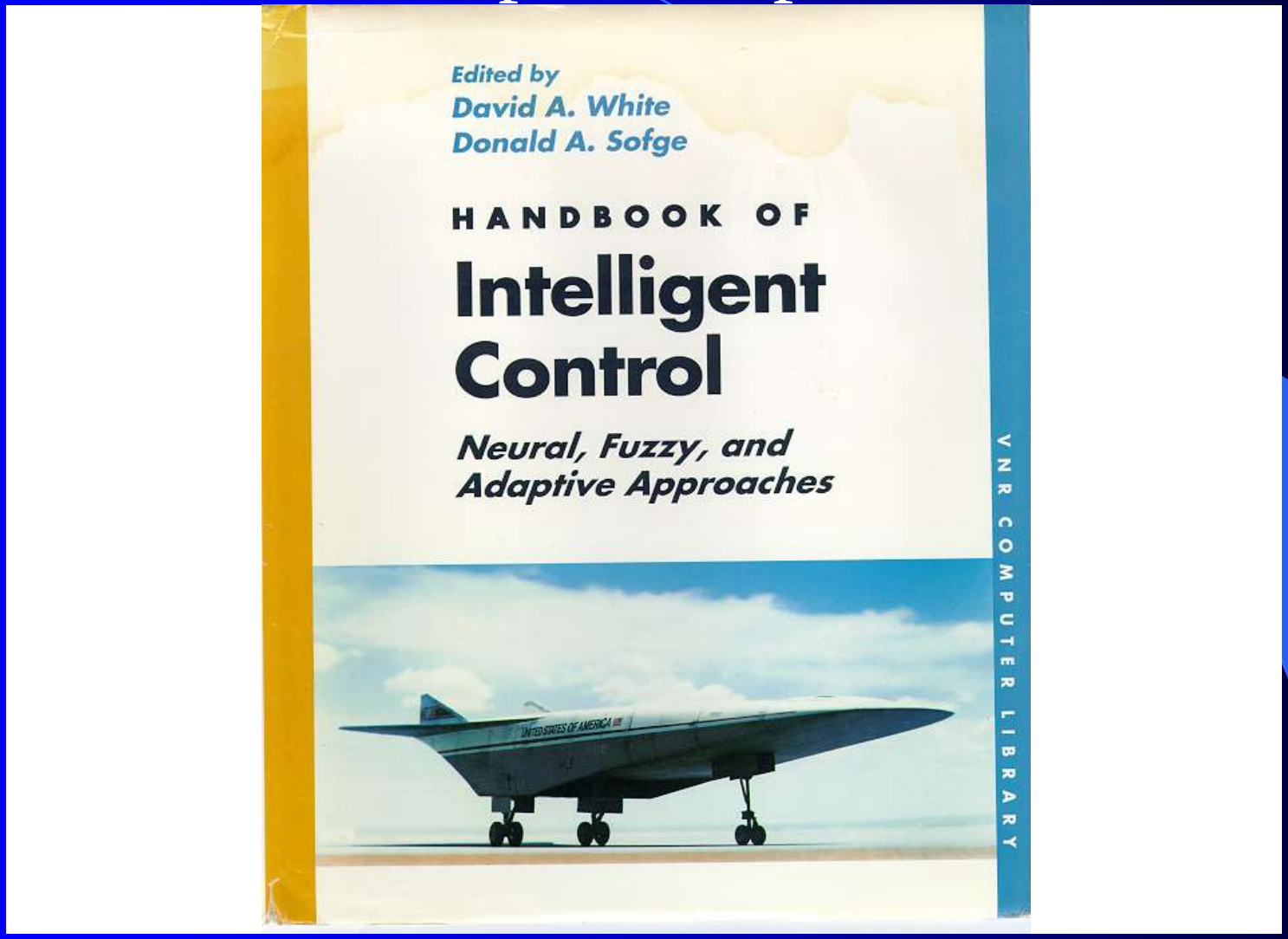
Universal Vector Prediction System: Principles To be Explained

- For smooth functions $\underline{\mathbf{Y}}=\underline{\mathbf{f}}(\underline{\mathbf{X}})$, Multilayer Perceptron (MLP) minimizes complexity and hence estimation error. Barron.
- For general functions $\underline{\mathbf{Y}}=\underline{\mathbf{f}}(\underline{\mathbf{X}})$, add simultaneous recurrence ($\underline{\mathbf{y}}[n+1]=f(\underline{\mathbf{y}}[n],\underline{\mathbf{X}})$ for Turing-like universality. SRN.
- For dynamic or time-series prediction, add time-lagged recurrence $\underline{\mathbf{Y}}(t)=\underline{\mathbf{f}}(\underline{\mathbf{Y}}(t-1)$, memory ($\underline{\mathbf{R}}(t-1)$), $\underline{\mathbf{X}}(t)$) for universal "NARMAX' capability (TLRN)
- Unify maximum likelihood (least squares training) with precedent-based forecasting, "uninformative priors" (penalty functions), & weights for multiperiod prediction and salience – especially for real-time "incremental" learning.

$\Rightarrow$ Learning speed also an issue, harder with better prediction. Many useful tricks known. Kozma/Ilin/Werbos patent just a useful start.

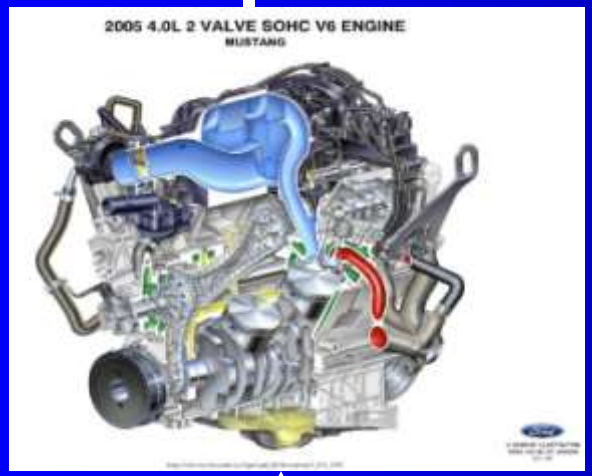
"HIC": NSF/McAir Workshop 1990

See 2nd half of chapter 10, posted on web



White and Sofge eds, Van Nostrand, 1992

Winner of IJCNN07 Forecasting Contest: Ford 1998: “All Ford Cars Will Have TLRNs by 2001, for on-board Diagnostics”



- How can one neural network predict and diagnose all Ford engines, without retraining, even as they change over time? TLRN: adaptive prediction even without learning! ICNN05: “A neural network which can predict anything.”
- IJCNN07, Prokhorov: TLRN prediction and control can improve mpg of Prius hybrid by 15% “at zero cost”!

Neural Network in Commercial Power Grid Hardware

- First deployment of deployment of recurrent neural network in the field in a commercial electric power grid. (Improved prediction to allow unprecedented monitoring and control of harmonics.) Harley, Georgia Tech.

Robot first copies human by learning to predict time-series of human

Schaal, Atkeson
NSF ITR project

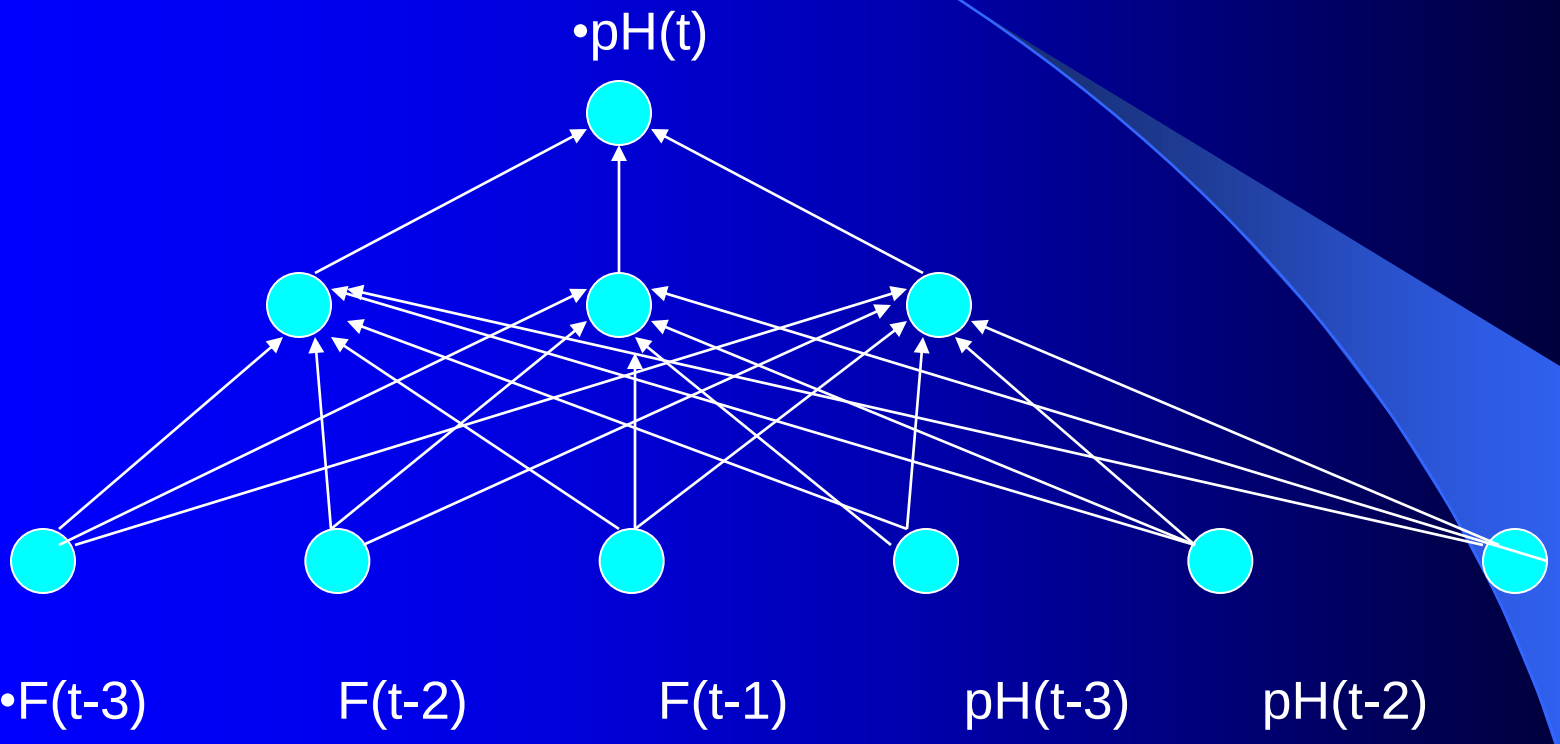
Learning allowed robot to quickly learn to imitate human, and then improve agile movements (tennis strokes). Learning many agile movements quickly will be crucial to enabling >80% robotic assembly in space.

1st Neural Flight Control (90's): "Cloning" best human pilot in slowed-down "game" led to Robust controller for National Aerospace Plane model much faster than previous best controller. The neural contractor AAC became for a time Lead company in US hypersonics effort. Based on learning to predict human actions.

Advice to Neural Net Engineers

- Don't let this competition distract you from critical prediction tasks in engineering – clean, flexible car engines; power grids; batteries; manufacturing plants; chemical plants, etc. (www.werbos.com)
- Keep your eyes on the multivariate case –causal relations to enable control, brain-like complexity
- Fill in your weakness in general-purpose modular software (MatLab⇒C ⇒chip). Example: why do people use 10,000-crash broom-balancers instead of no-crash balancers?
- Create software which makes it quick and easy for you to compete here, and learn and disseminate
- Learn to improve your accuracy in the general case
- Learn & teach the underlying statistical principles – simple but crucial points, not well-known even to most statisticians

Myth 1: Training Multilayer Perceptrons (MLP) is not black magic, is not an alternative to statistics



•Any MLP represents a function **Y**=**f**(**X**,**W**), **X** the inputs, **W** the weights.
•Minimizing the mean square value of (**actual Y** – **f**(**X**,**W**)) over **W** is nonlinear regression. All the usual error and significance and standard error statistics apply. It's just a more general choice of **f** than usual (able to approximate any nonlinear smooth function efficiently) and it comes with faster more reliable convergence.
Standard errors are less with more data and fewer weights.

Myth #2: "No Free Lunch" Is Not a Theorem

- An understandable reaction to ad hoc "A is better than B" studies and the old "pick a chapter" psychology
- But: given two families of models or topologies, **g**(W_1) and **f**(W_2), if every model in **g** is close to a model in **f** but not vice-versa, then f is more powerful. "Almost-free lunch."
- Given enough data or given the right priors (favoring **g**-like points in **f**), **f** should always do much better than **g** or almost as well
- Examples:
  - ARMA beats AR: $x(t)+bx(t-1)=e(t)+ce(t-1)$, $c\neq 0$
  - (ARMA fits partially observed or noisy underlying AR.)
  - TLRN beats ARMA: $x(t)=e(t)+f(x(t-1),R(t-1))$
- BehavHeuristics airline seat forecasting example
- Most powerful if **f** is most universal approximator, fewer parameters. Neural vs. Taylor/translog, SRN versus MLP.

3: Correlation Versus Causality – Why Most Data Mining is Bogus and How We can Infer Causality

	Provinces Getting Poverty $	Provinces Not Getting Poverty $
Low Income	30	2
High Income	3	20

Human intuition or statistics for data at one time seem to say this povetry program causes low income! But think. The only legitimate way to judge impact is to predict the change form time t to t+1 from action at t – or, more generally and more accurately – to build models to predict the future as accurately as possible.

"Better statistical controls" simply mean ever better prediction of dynamics over time (never perfect). Our ability to act correctly is always limited by our knowledge of how the world works.

4. We Can Use Domain Knowledge

- Caveat: If I had only 30 data points in a national economic model, I would use prior knowledge to craft f, and would not use a neural network.
- Even in a learning approach, we can use "data pooling," in neural nets and in econometrics. (e.g. At www.werbos.com/energy.htm, I post most acurate model ever developed to predict industrial energy demand – exploiting pooling.) In time-series competition, you could pool across the time-series in the competition, to estimate domain tendencies! (Metalearning) "Poor man's object net" exploits symmetry across the objects in the data set.
- Initial weights can be what was learned in a related situation or trained to represent what humans would do. This is a crucial technique in many practical applications. One can even minimize an error function which penalizes deviations from those initial weights.

"Bayes" versus "Vapnik": today's debate

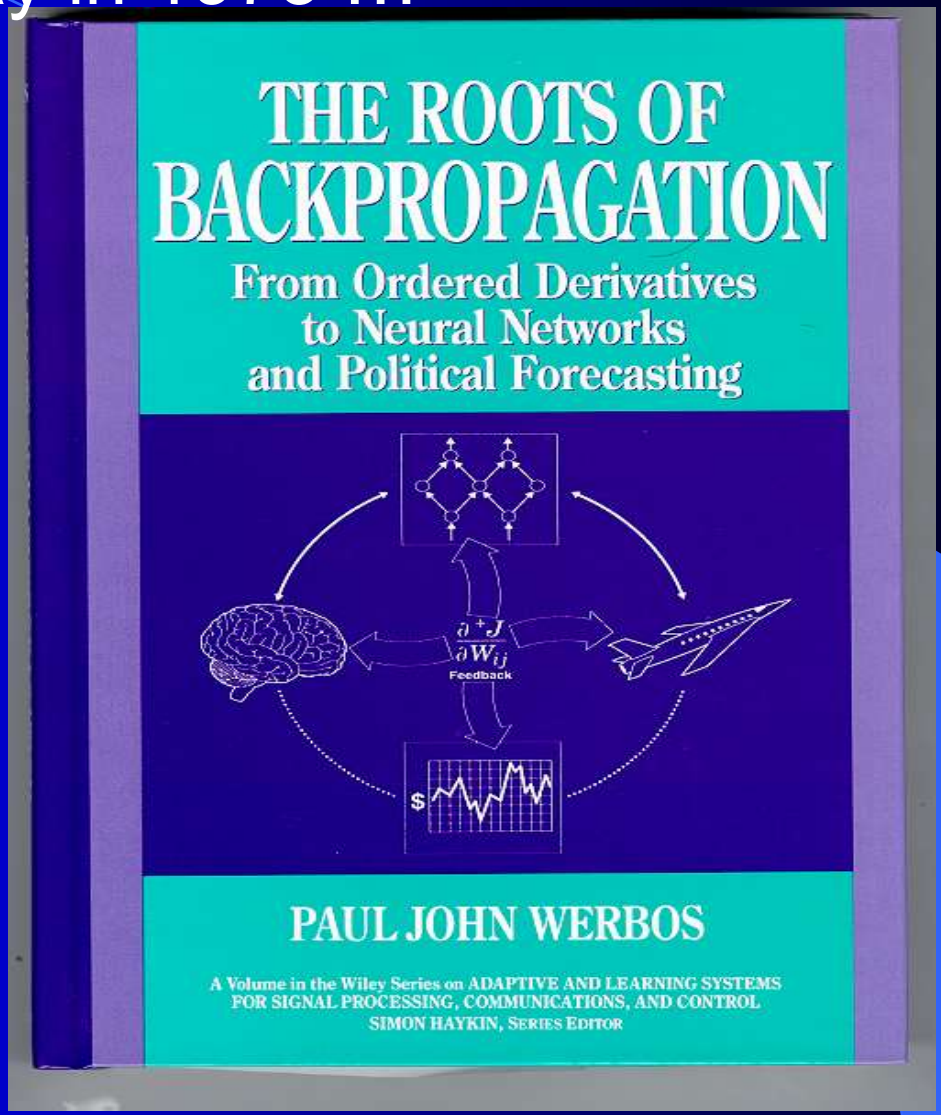
- Theorem: Pr(A|B) = Pr(B|A)*Pr(A)/Pr(B)
- Platonic Bayes:
  - Predict by using stochastic model Pr($\underline{\mathbf{x}}$(t)|past)
  - Find model with highest probability of being true: Pr(Model_W|database) = Pr(database|Model_W)* Pr(Model_W)/Pr(database)
  - Neural $\underline{\mathbf{x}}(t+1)=\underline{\mathbf{f}}(\underline{\mathbf{x}}(t),\ldots,W)+\underline{\mathbf{e}}(t)$ is just another stochastic model, with full NL regression statistics
  - Many variations; e.g. "Box-Jenkins" ARMA methods
  - "anything else is Las Vegas numerology"
- Vapnik says NO. "New" philosophy: if you want $, not truth, pick Model_W which would have maximized $ in the past (database)

Platonic Bayes fails very badly in some ways, as I learned the hard way in 1973 …

Vector ARMA (f) had twice the prediction error of simple extrapolator (g), on 100-year political data and simulated dirty datasets

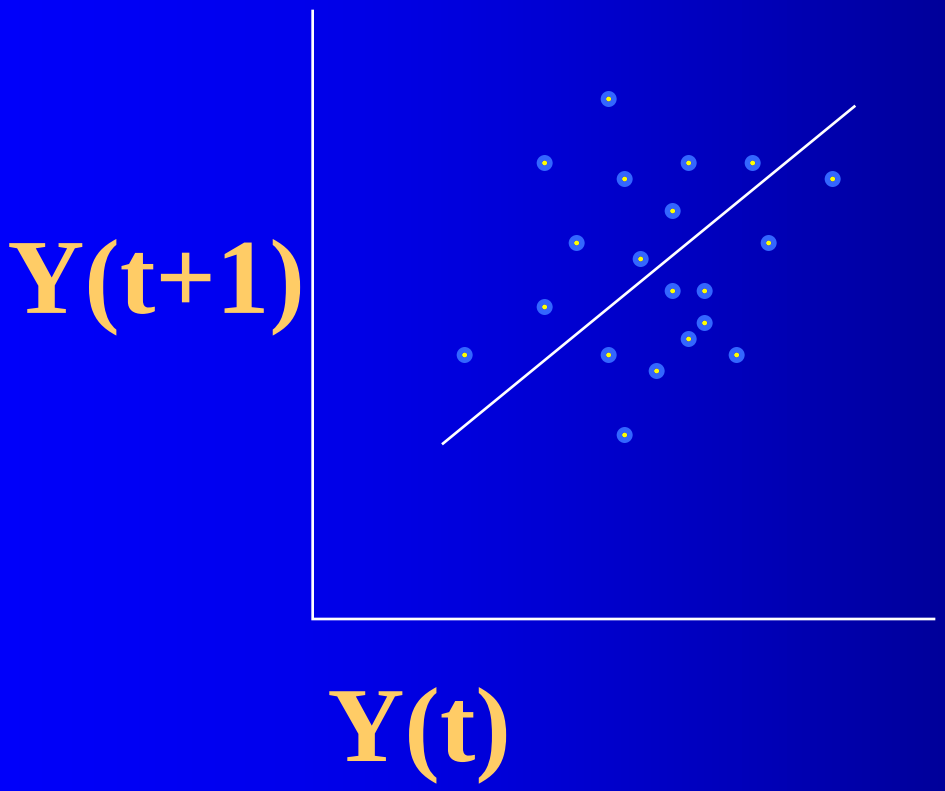
"Vapnik" style "pure robust method"

BRAINS absolutely require multiperiod robustness beyond what Platonic Bayes offers

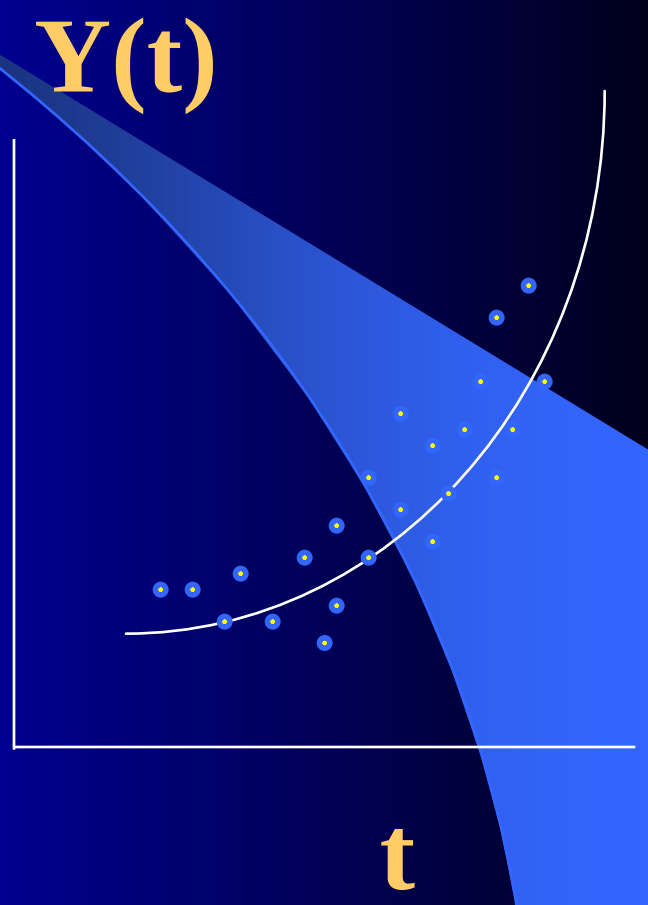


1974 Harvard PhD in subject of statistics, Mosteller on committee (Dempster help)

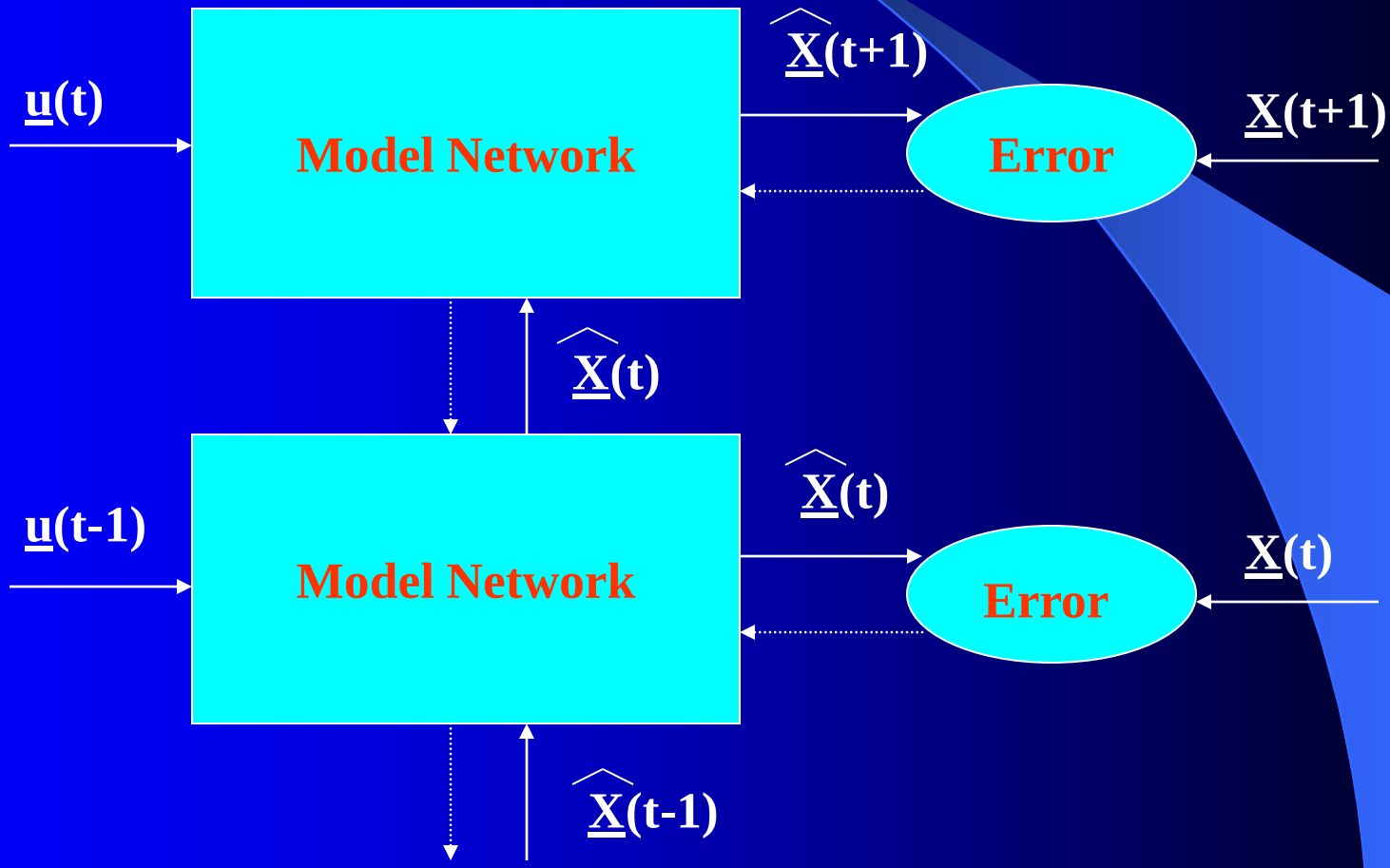
Conventional Least Squares

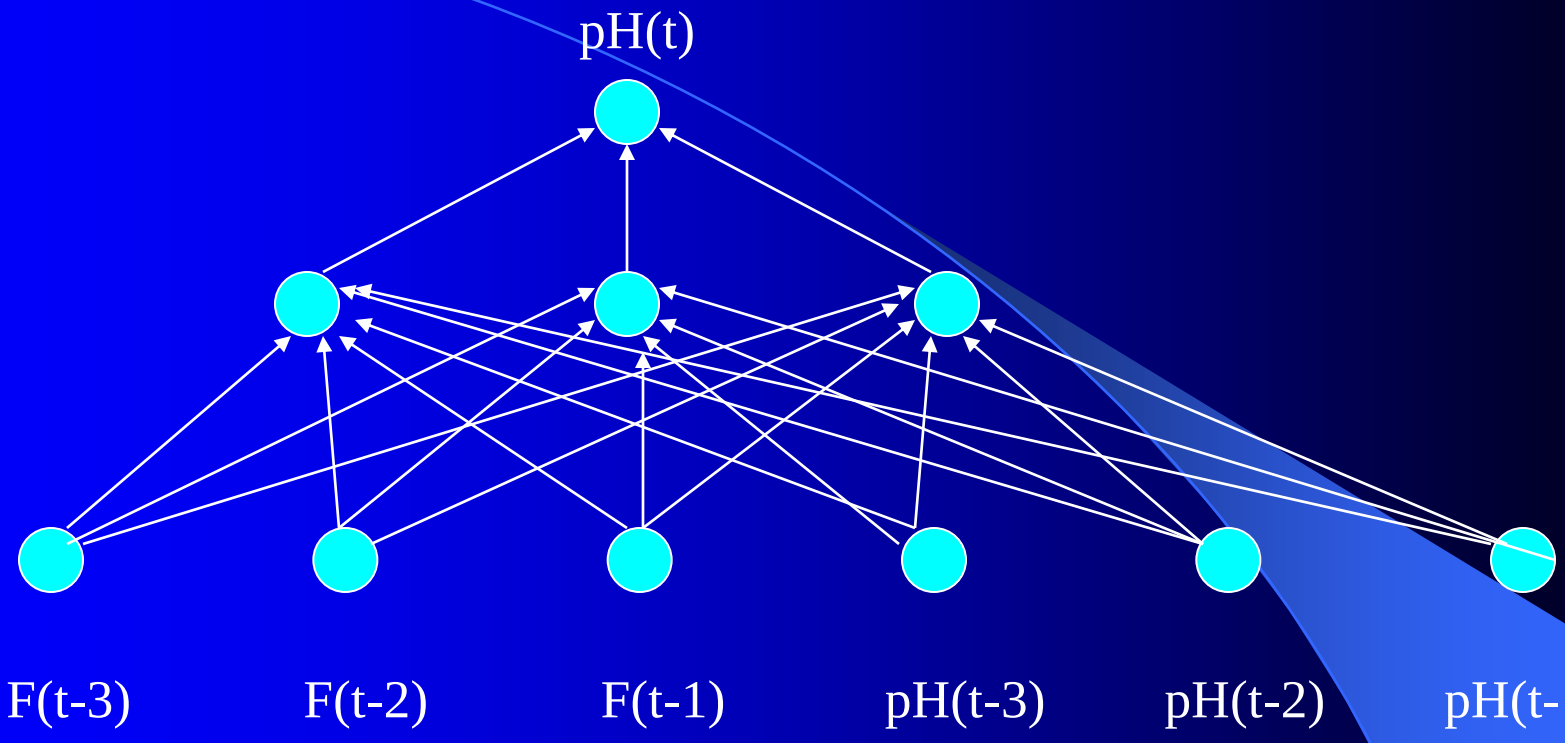


Pure Robust



PURE ROBUST METHOD

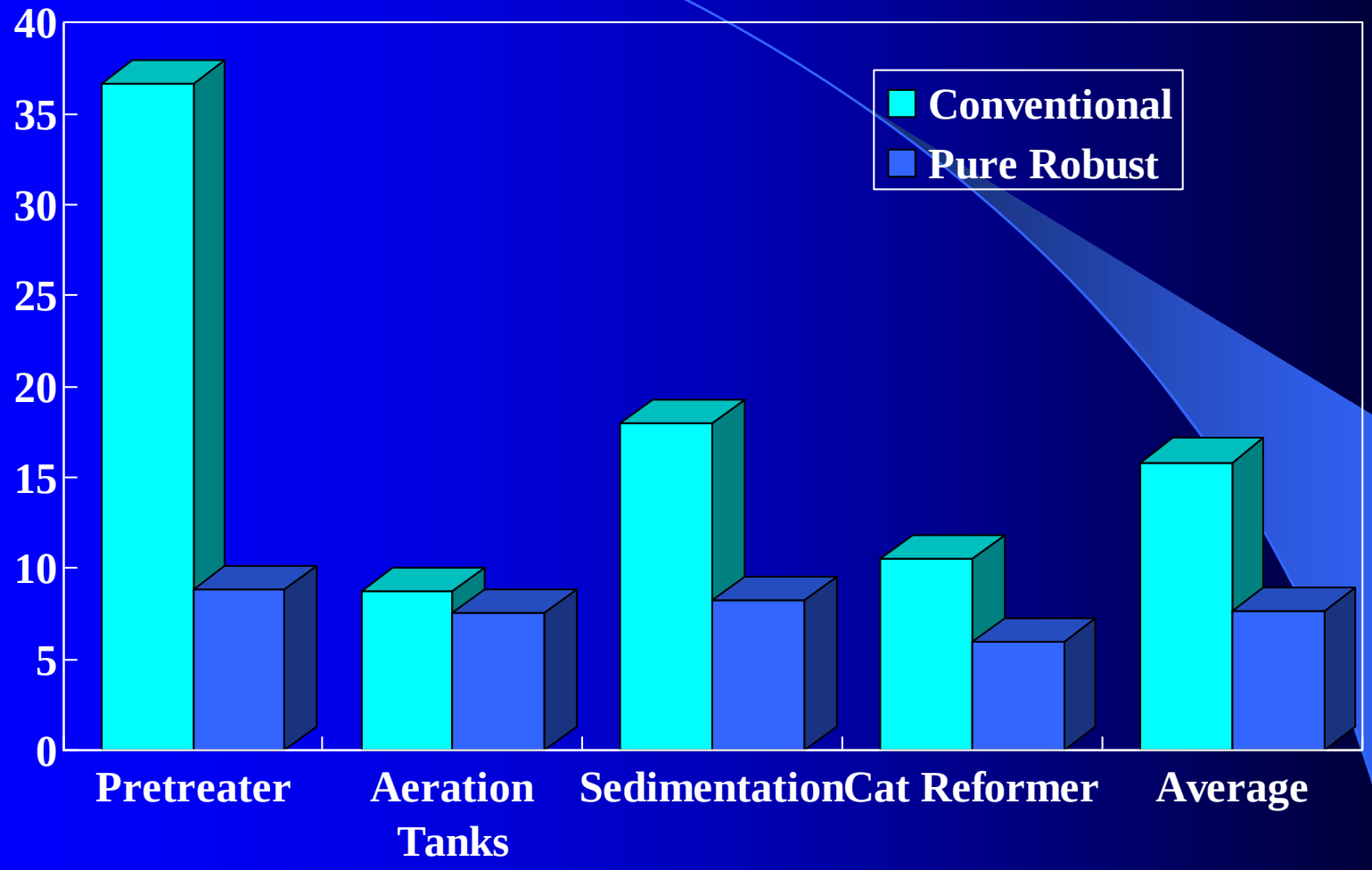




Example of TDNN used in HIC, Chapter 10

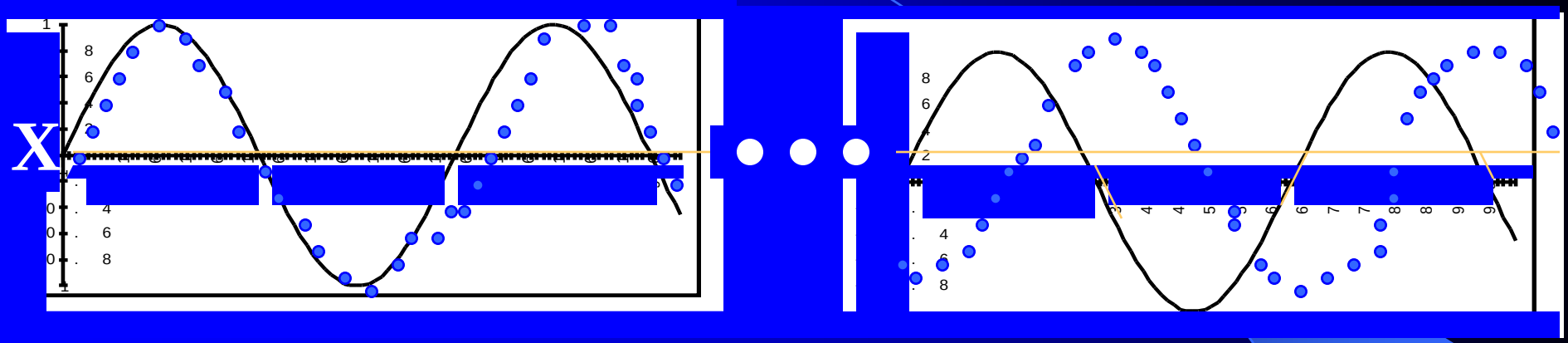
TDNNs learn NARX or FIR Models, not NARMAX or IIR

Prediction Errors (HIC p.319)



- Greatest advantage on real-world data (versus simulated)
- Full details in chapter 10 of HIC, posted at www.werbos.com.
- Statistical theory (and how to do better) in second half of that chapter.

But Pure Robust ("Vapnik") Can Fail Badly Too: Phase Drift



$$R(t+1) = R(t) + w + e_p(t)$$

$$X(t) = \sin R(t) + e_m(t)$$

TINY

A unified method cut GNP errors in half on Latin American data, versus maximum likelihood and pure robust both (SMC 78, econometric).

"Vapnik" approach is not new even in the static case

- Utilitarian Bayes: google "Raiffa Bayesian": pick model and weights W so as to minimize a loss function L.
- Example of the issue: to weight or not weight your regression (in actual DOE/EIA model and conflict model):

  Energy(state,year)=a*income(state,year)+e(year) (1)

  (energy(state,year)/income(state,year)=a+e(year) (2)

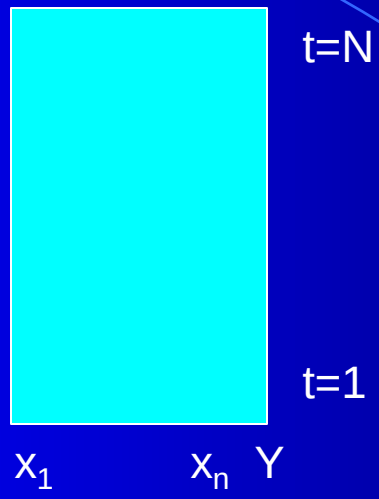
  If big states different, equation (1) is more consistent

  If big states few, (2) has more information, less random error

  Platonic approach: use F tests to see which is more true, but..

NonBayesian methods in econometrics for consistency under more general conditions

Model-Based Versus Precedent-Based: Which Is Better?



- Model-based: Pick W to fit Y=f(x,W) across examples t. Given a new x(T), predict Y(T) as f(x(T),W).
- Precedent-Based: Find t whose x(t) is closest to x(T). Predict Y(T) as Y(t). Kernel is similar, weighted sum of near values.
- Best is optimal hybrid, needed by brain. "Syncretism" – chapter 3 of HIC. Keep training f to match examples or prototypes in memory, especially high-error examples. Predict Y(t) by f plus adjustment for errors of f in nearby memory. Closest so far: Principe kernel applied to model residuals; Atkin's memory-based learning. Exact fits Freud's description of ego versus id in neurodynamics.

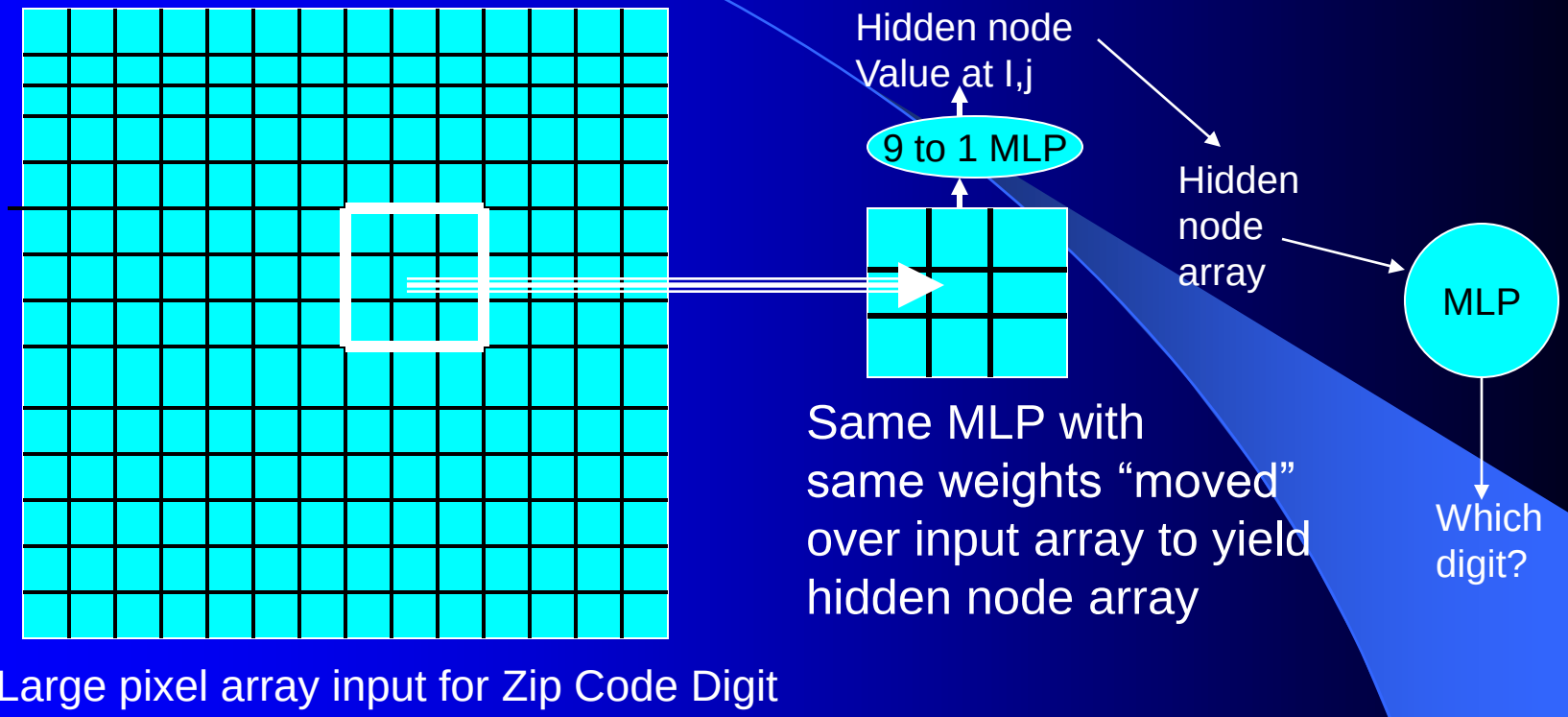
Example of Freud and Syncretism

- A Freudian story:
  - Nazi hurts child, a traumatic memory
  - For years, he is terrified when anyone in black shirt appears (precedent based prediction/expectation) – the kernel-based "id" is at work!
  - Later he learns about Nazis in subjective model of world (f), "ego"
  - After that learning, if he relives that memory (trains on memory), f error on the memory is low; memory loses power to cause irrational bias
- Key corollaries:
  - False hope from memory is as dangerous as false fear
  - We still need id when exploring new realms we can't yet reliably predict

The Prior Term Pr($Model_W$) is crucial, in Bayesian or robustified statistics

- Not just specific domain knowledge, but key basic principles like Occam's Razor – that Pr($Model_W$) is greater for simpler models. See Emmanuel Kant: "apriori analytic." New jargon: "uninformative priors" and "metastatistics."
- Under old school "flat priors," human brain could not exist. Too many variables.
- 1977: to handle complexity (many input variables), ridge regression – empirical Bayes, estimated pr(W_i).
- For ANNs: penalty functions, robustified by allowing redundancy (Phatak); symmetry (see brain paper). Symmetry+TLRN and proper loss function was how we got 6% per month above Dow in 1990's..

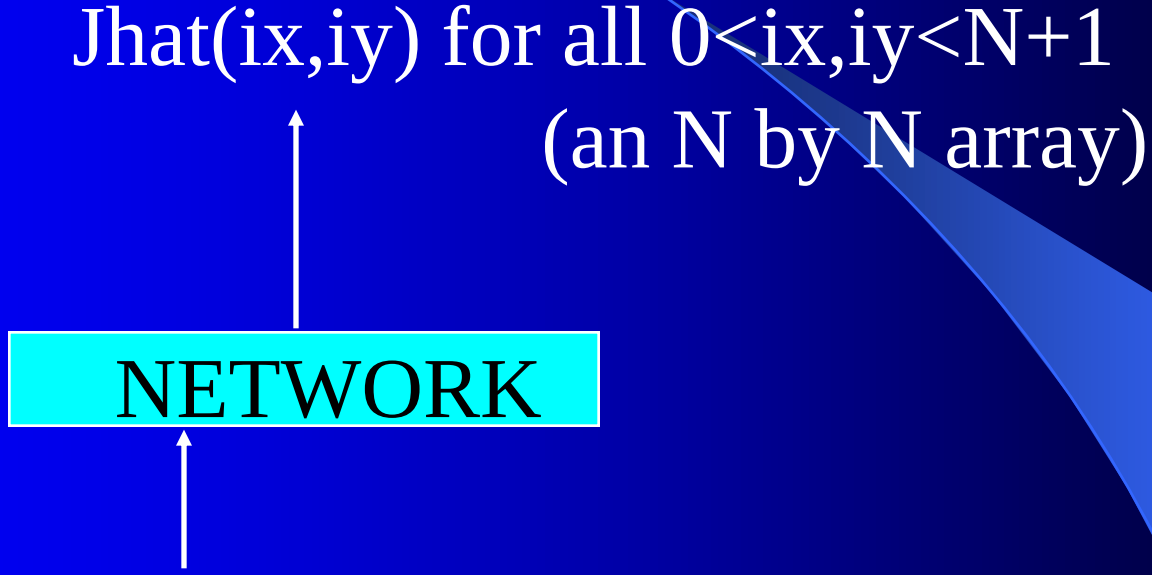
Moving Window Net: Clue Re Complexity



- Best ZIP Code Digit Recognizer Used "Moving Window" or "conformal" MLP! (Guyon, LeCun, AT&T story, earlier…)
- Exploiting symmetry of Euclidean translation crucial to reducing number of weights, making large input array learnable, outcomes.

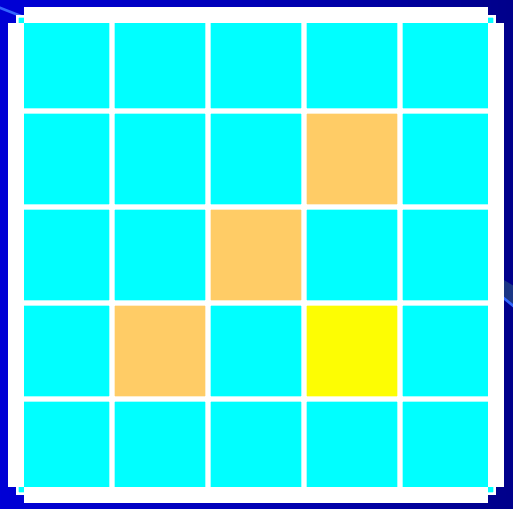
Cellular SRN: The Recurrent (SRN) Generalization of "Conformal MLP"

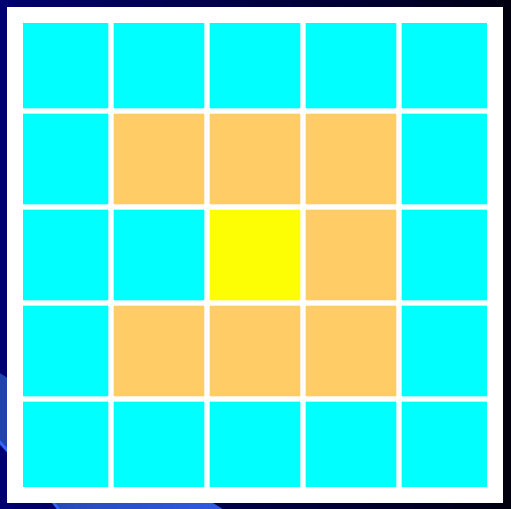
GENERALIZED MAZE PROBLEM

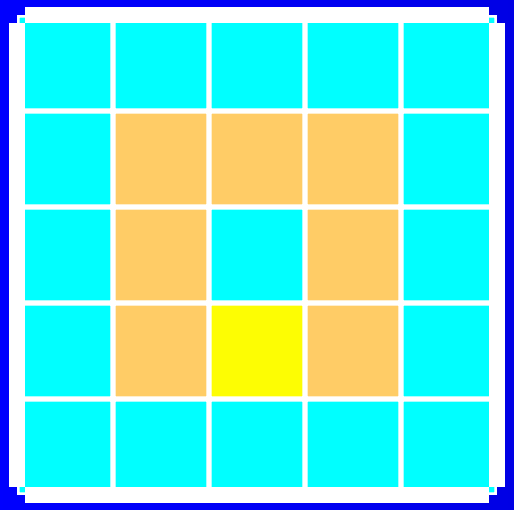


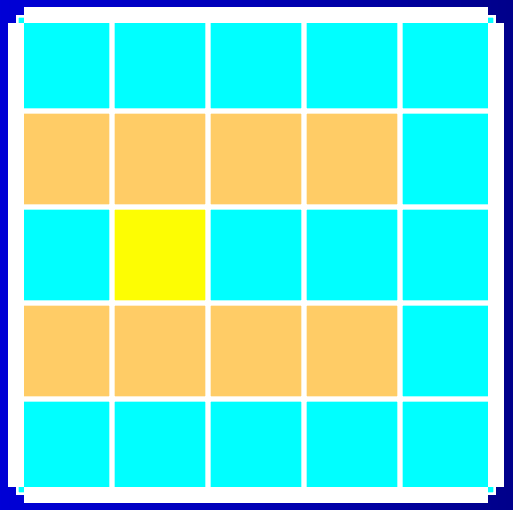
Rapid learning algorithm by Kozma, Ilin, Werbos: IEEE Transactions on Neural networks, June 2008

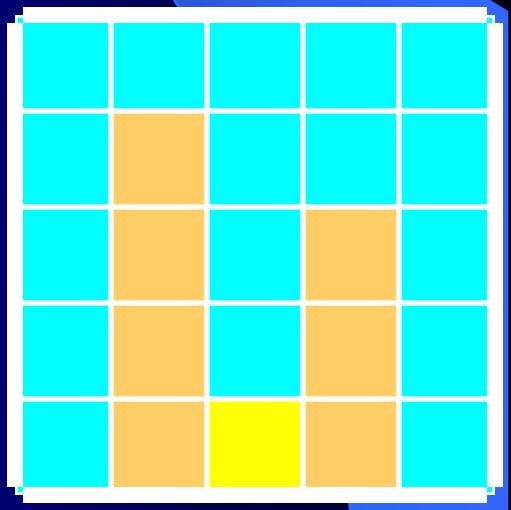

4 3 2 1 2
5 1 0 1
6 7 1 2
7 8 7 3
8 7 6 5 4

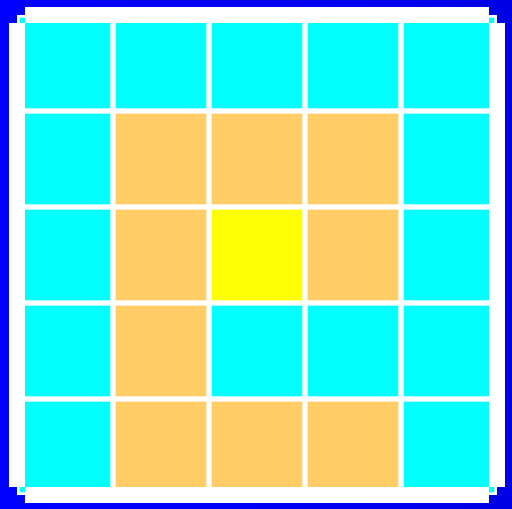
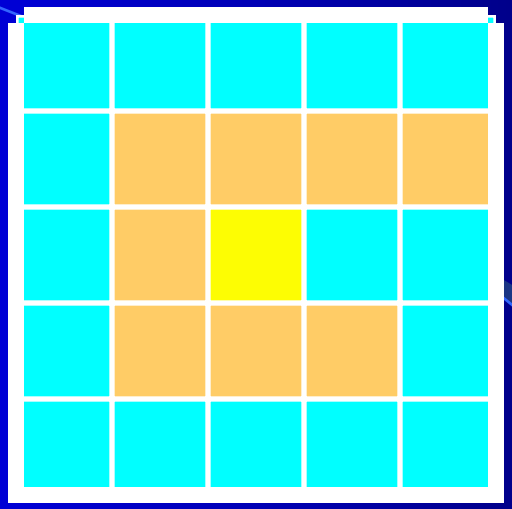
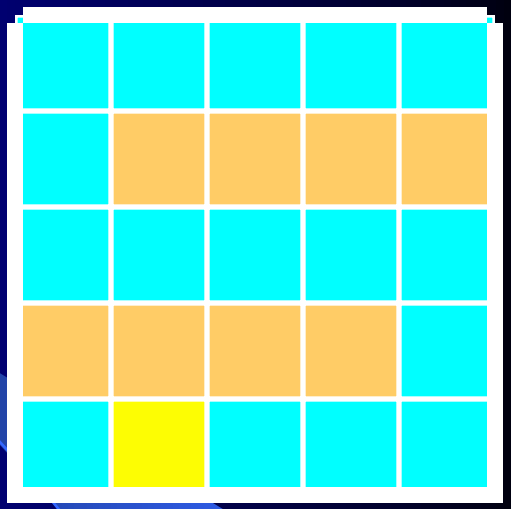
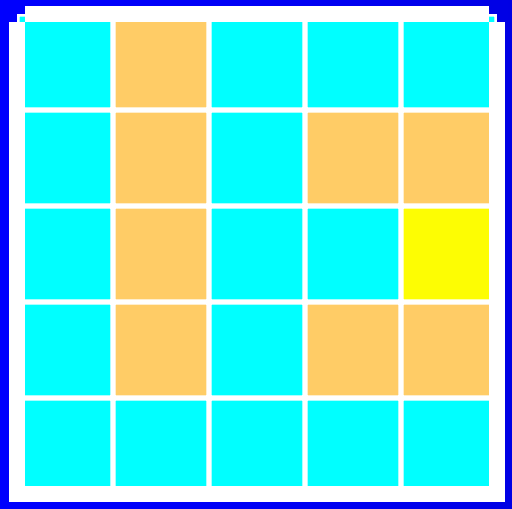
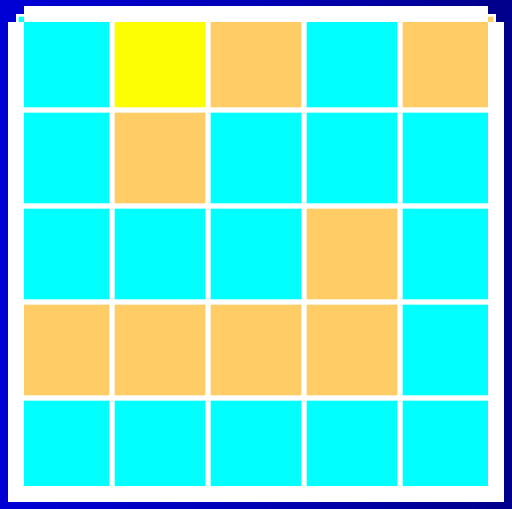
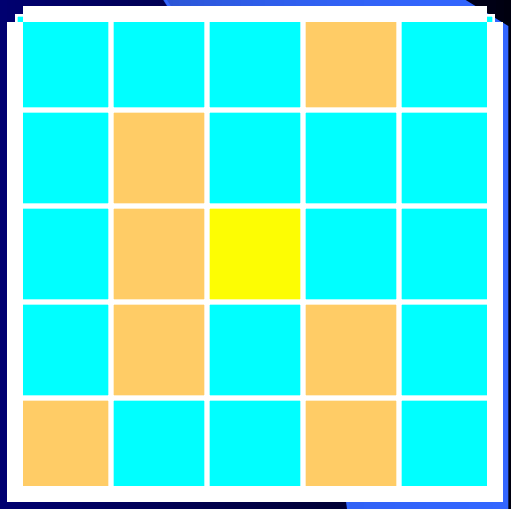






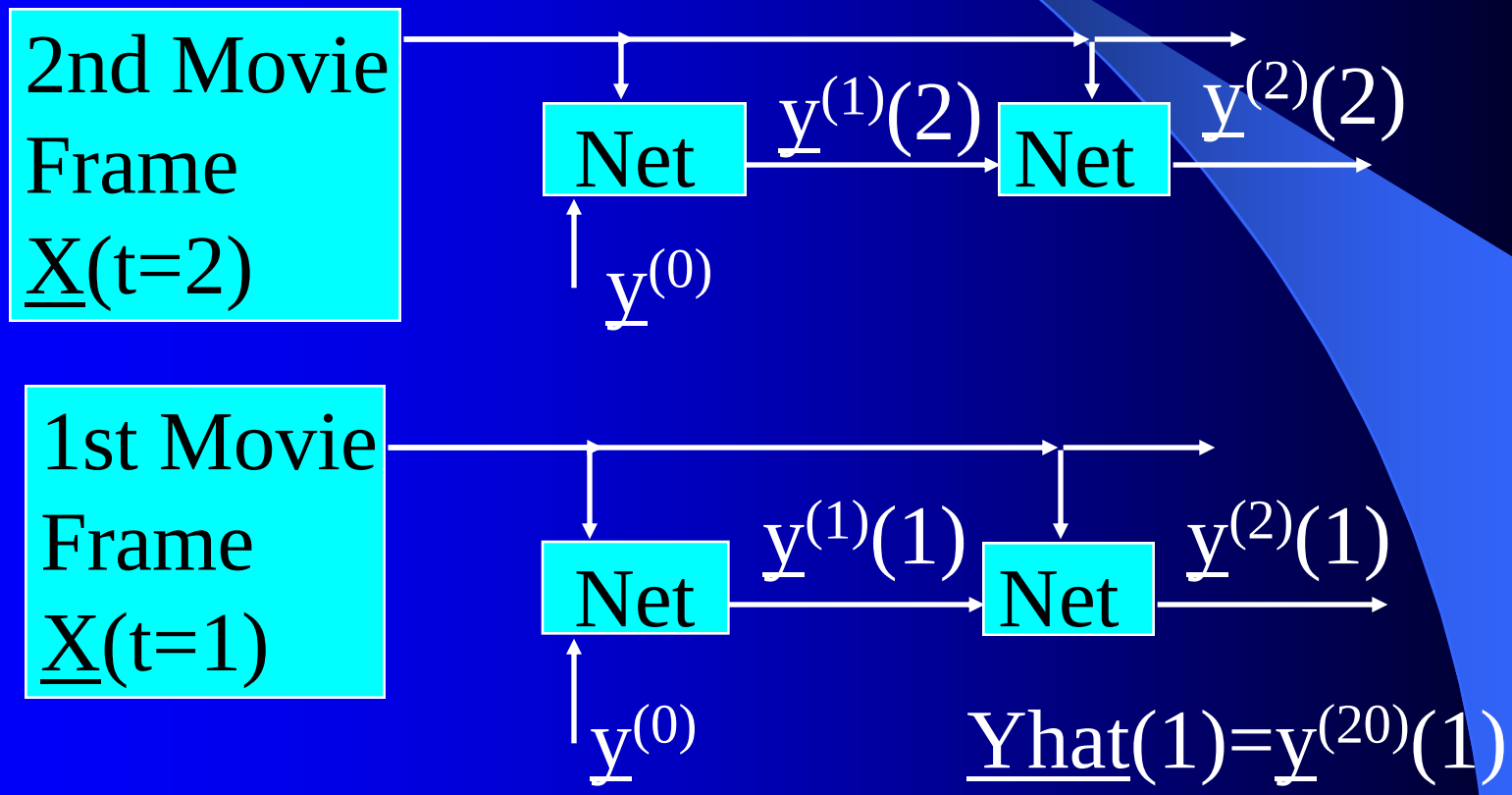




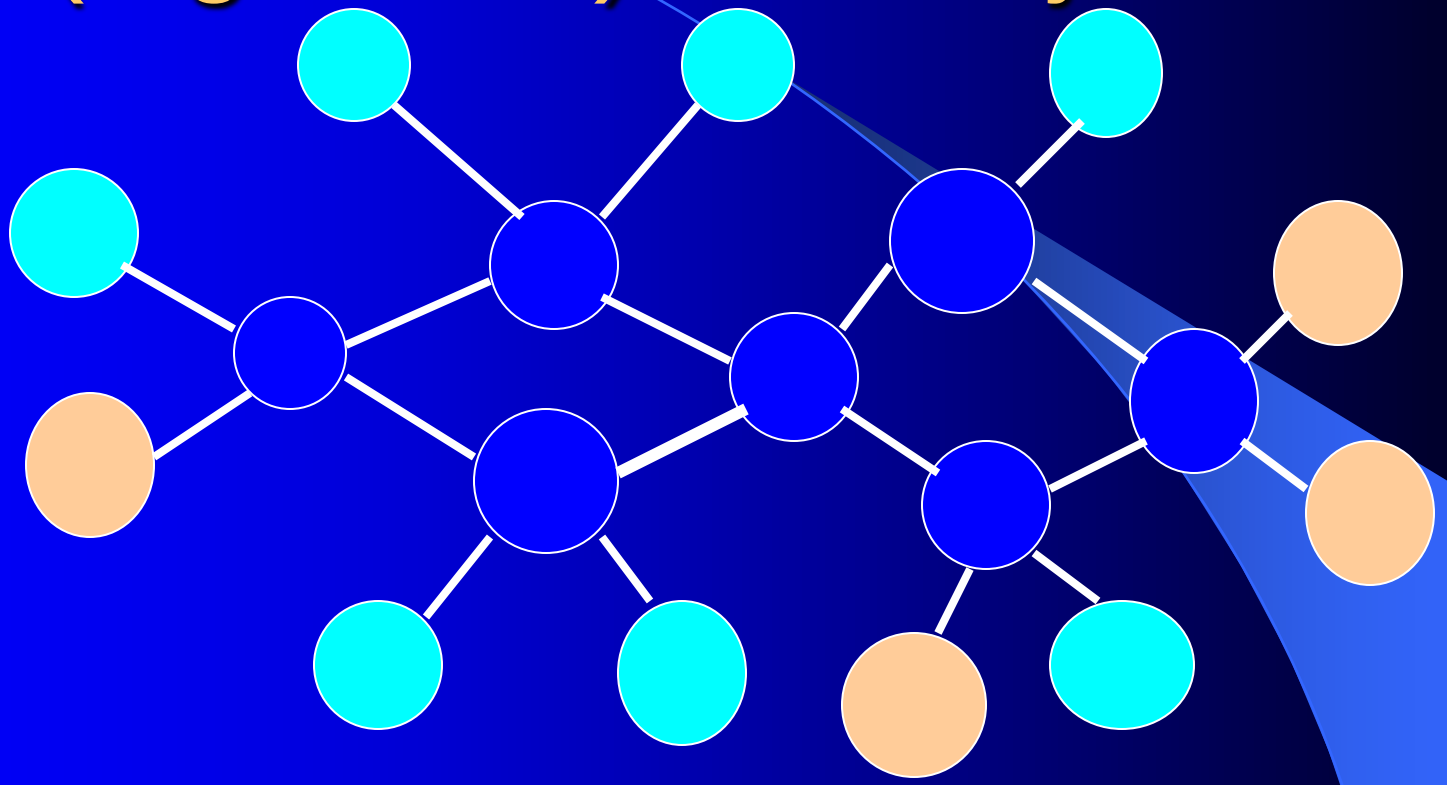







IDEA OF SRN: TWO TIME INDICES t vs. n

(Simultaneous Recurrent Net is not equivalent to "Simple Recurrent Net" later proposed by psychologists!!!)



Spatial Symmetry in the General Case (e.g. Grids): the Object Net

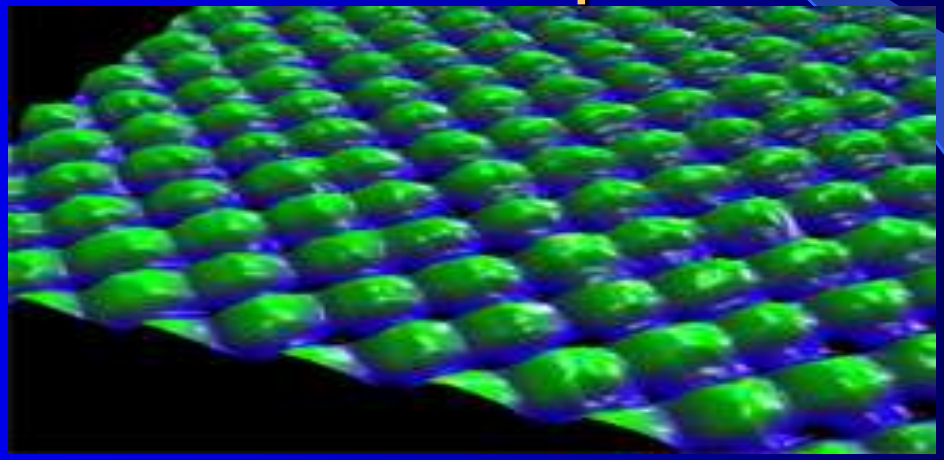


- 4 General Object Types (busbar, wire, G, L)
- Net should allow arbitrary number of the 4 objects
- How design ANN to input and output FIELDS -- variables like the SET of values for current ACROSS all objects?
- Great preliminary success (Fogel's Master Class Chess player; U. Mo. Power)
- But how learn the objects and the symmetry transformations???? (Brain and images!!)

David Fogel (Proc IEEE 2004): World's First System which LEARNED Master-Class Performance in Chess

- Evolutionary computing (EC) to train a game-player worked for tic-tac-toe, but not checkers
- EC to train a multilayer perceptron (MLP) to serve as a CRITIC (an ADP value function) was enough to beat checkers but not chess
- EC to train a feedforward Object Net as a Critic was enough to beat chess
- Prediction: A full (recurrent) ObjectNet Critic can get to master class in Go. Will Wunsch get there first?

Key technology challenge for CLION: how can we use learning to get best performance from new chips like CNN, with thousands of processors per chip, across a huge segment of the market for computation?



Key enablers:

- New chips are suitable for nonlinear function approximators like CSRN, ObjectNets which can handle more complexity than traditional Taylor series, neural nets, lookup tables, etc.
- Kozma/Ilin/Werbos paper shows how they can be trained
- Neural net research shows general-purpose ways to use them