Neural Network Modeling of Self-Similar Teletraffic

Homayoun Yousefi'zadeh
Center for Pervasive Communications
Electrical and Computer Engineering Dept.
University of California, Irvine
http://www.ece.uci.edu/~hyousefi/pub.html

Self-Similar Teletraffic

- Burst Within Burst Architecture
- Traffic Spikes Ride on Longer Term Ripples, that in Turn Ride on Longer Term Swells, So on and So forth
- Structural Similarities across a Wide Range of Time Scales
- Teletraffic Patterns Exhibit Fractal Behavior w/ No Actual Burst Length

Second-Order Self-Similarity

- Slowly Decaying Variances
  - $var(X^{(m)}) \sim k_2 m^{(-b)}$ as $m \to \infty$ with $0 < b < 1$
- Long Range Dependence
  - Non-Summable Autocorrelations, $\sum_n R(n) = \infty$
- $1/f$ Noise Property
  - Power Law near the Origin

    $f(l) = k_3 l^{-g}$ as $l \to \infty$ with $0 < g < 1$

Self-Similarity Evidence

- Ethernet LAN Traffic: Leland et al.
- ISDN Traffic: Hellstern et al.
- Variable Bit Rate (VBR) Video: Beran et al.
- Common Channel Signaling Network (CCSN): Duffy et al.

Self-Similar Teletraffic Modeling

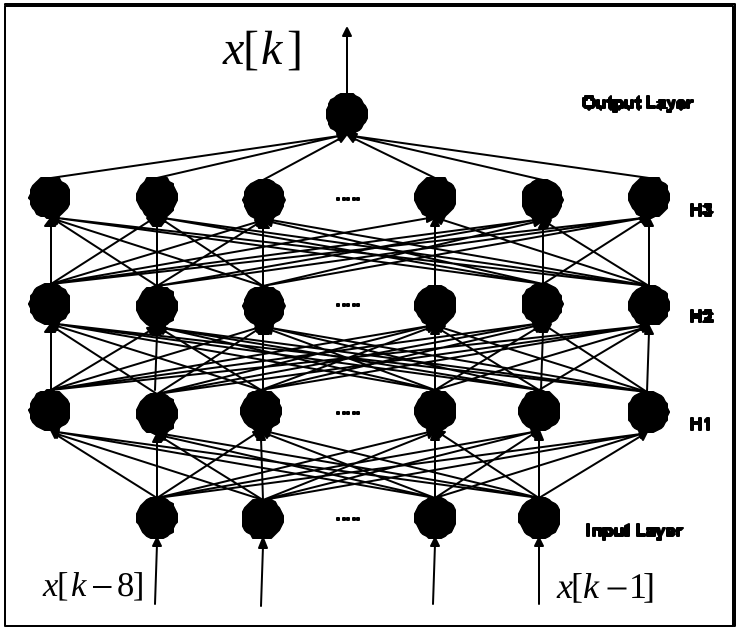
- Statistical Approach
  - Leland, Willinger, Taqq, Willson
- Chaotic Systems Approach
  - Erramilli, Singh, Prutti
- Neural Network (NN) Approach
  - Yousefi'adeh, Jonckheere

Perceptron NN Traffic Modeling

- Fully Connected
- Feed Forward
- BPA Learning
- Inputs:

  $x[k-8]$ thru $x[k-1]$
- 3 Hidden Layers

  w/ 20 PEs Each
- Output: $x[k]$



Back Propagation Algorithm

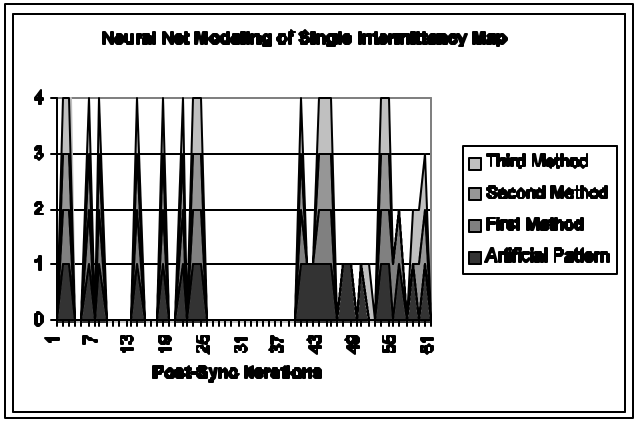
- Start from an Initial Set of Weighting Functions
- Do {
  - Propagate Forward from Input u to Output J
  - Calculate Absolute Output Error $E = \frac{1}{2}\sum_k (u_k - J_k)^2$ and Calculate Backward Scaled Relative Errors of Each PE
    $$e_j[s] = x_j[s].(1 - x_j[s]).\sum_k \{e_k[s+1].w_{kj}[s+1]\}$$
  - Calculate per PE Variations of Weighting Functions
    $$\underbrace{\Delta w_{ji}[s]}_{(k+1)-thstep} = lc.e_j[s].\{x_i[s-1] + k.e_i[s-1]\} + M(\underbrace{\Delta w_{ji}[s]}_{k-thstep})$$

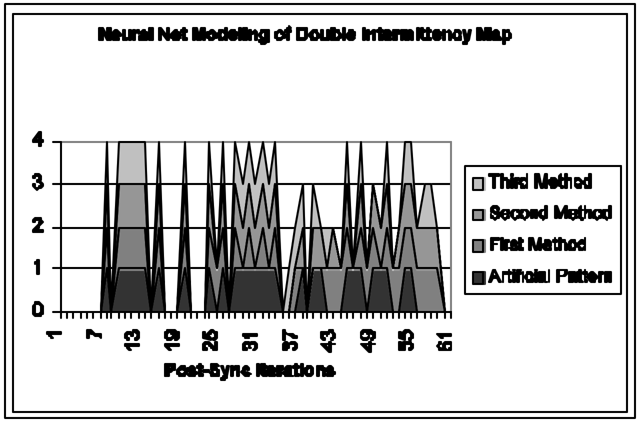
} While (Error > Bound)

NN Teletraffic Modeling

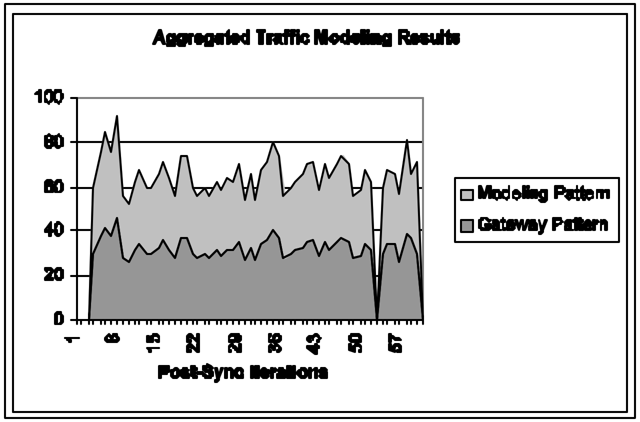
- Intermittency Maps Artificial Traffic
- Modeling Alternatives:
  1. Discrete 0 and 1 per PKT Traffic Samples
     - Low Learning Speed
  2. Predicting Samples of Chaotic Maps
     - Indirect Modeling w/ the Same Threshold
  3. Normalized Artificial Traffic Samples
     - Direct Modeling Utilizing Continuous Artificial Samples between 0 and 1

Complexity & Modeling Results





- BPA Complexity
  - Time: $O(IN)$
  - Space: $O(N)$



Numerical Issues

- Time Consuming Learning Algorithm Due to Dynamics Complexity
- Iterative Learning Is Required as the Result of Facing Chaotic Divergence
- Impact of Initial Weighting Functions of NN in the Algorithm Convergence