Synaptic Learning Rules: The Clusteron Volume Learning

David S. Touretzky

COMPUTATIONAL MODELS OF NEURAL SYSTEMS

Spring 1995

David S. Touretzky, Carnegie Mellon University

File: clusteron.frame, version of February 27, 1995 5:15 PM

Input Integration in Dendrites

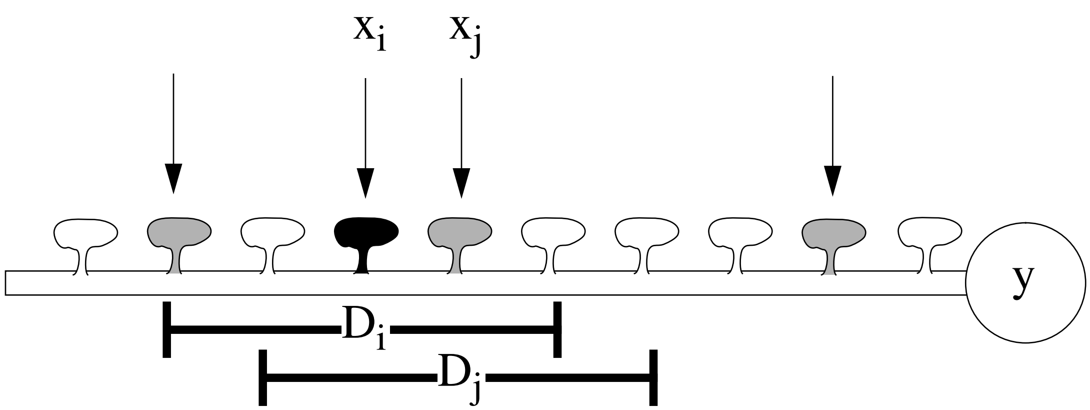
- The integrative behavior of the pyramidal cell's dendritic tree remains largely unknown.
- But we know the dendrites do not just passively sum epsps.
- They containvoltage-dependent membrane mechanisms, including:
    - NMDA receptors
    - spiking dendritic conductances
- Result: spatial structure of dendritic input is important.
- 100 synapses activated in 20 clusters of 5 may produce different results than 100 synapses randomly scattered throughout the dendritic tree.
- Pyramidal neurons may be sensitive to *clusters* of inputs.
- This makes them conjunctive feature detectors.
- They can also *learn* new clusters.

Research Strategy

1. Do biophysically accurate compartmental modeling of a pyramidal cell, to explore cluster sensitivity and a Hebb-like learning rule
2. Develop a simpler, abstract model to explore cluster learning in more depth. This is the **clusteron** model.
2. Examine applications of clusteron learning:

   - recognizing visual images
   - (suggested) learning to recognize objects from novel views using clusters of low-level visual features such as color and texture.

The Clusteron



Activation equation:

$$y = g\left(\sum_{i=1}^{N} w_i a_i\right)$$

Excitatory input from synapse i:

$$a_i = x_i \cdot \sum_{j \in D_i} w_j x_j$$

Clusteron Learning Rule

Consider a training set of P patterns:

$$T = \{X^{(1)}, X^{(2)}, \ldots, X^{(P)}\}$$

For *i*th synapse, compute total activation across the training set:

$$A_i = \sum_{i=1}^{P} a_i^{(P)}$$

Let θ be an activation threshold.

When $A_i < \theta$, transpose this synapse with another subthreshold synapse elsewhere on the dendritic tree.

Threshold θ starts at a high value, and is lowered by 1 whenever the number of transposed synapses fails to decrease for three consecutive epochs (passes through the training set.)

Result: gradual fine tuning of the synaptic population, forming clusters on the dendritic tree.

Effect of Clusteron Learning

- Assume the training set T is drawn randomly from the population of 10,000-bit binary vectors.
- Patterns are sparse: $\rho = 0.01$, so each pattern has 100 1 bits.
- Clusteron has 10,000 sites at which to sample the input vector.
- The learning rule causes synapses to cluster together in positions corresponding to multiple 1's in the training patterns.
- The clusteron "memorizes" the training patterns this way.
- Learning increases the clusteron's response to these patterns.
- For patterns not in the training set, the correlation between positions of their 1 bits and those of the training set is zero.
- Hence, on average learning does not increase the clusteron's response to patterns not in T.
- Result: the clusteron learns to "recognize" patterns in T by producing a high response.

Clusteron Recognition Performance

- Training set contained 1000 patterns, 10,000 bits long.
- Clusteron diameter $D_i = 21$.
- Learning algorithm was run for 200-300 epochs.
- Resulting clusters were mapped onto a full compartmental model of a cluster-sensitive pyramidal cell.
- Test set contained five classes of patterns:
  1. training patterns
  2. random mixtures of two training patterns
  3. random mixtures of three training patterns
  4. training patterns with 50% of active bits replaced at random
  5. control patterns (not in training set)
- Response to training patterns was 4.6 times higher than response to controls: 0.80 vs. 0.17 spikes/100 msec.
- Response to mixtures and corrupted patterns was intermediate.

Comparison with a Perceptron

- A perceptron is a linear-threshold unit.
- Use binary weights: set $w_i = 1$ if x_i is active in more than 50% of the input patterns, else $w_i = 0$.
- Perceptron shows about 90% accuracy in discriminating 1000 learned patterns from controls. Clusteron did better.
- Clusteron has about 3 times the memory capacity of a perceptron, for a given performance level.
- Passive dendritic tree (equivalent to the linear portion of the perceptron) does not perform as well as the perceptron.

Populations of Clusterons

- A single clusteron learns only a tiny random sample of the second-order features of the training set.
- Multiple clusterons are thus essentially statistically independent samples of the feature space.
- Using a population of clusterons can give much higher recognition accuracy than a single clusteron.
- With 50 clusterons, we can discriminate 1000 training patterns from 1000 controls with 99.95% accuracy.
- Note that a population of 50 perceptrons would gain nothing over a single perceptron, since each perceptron represents the entire *first order* statistics of the training set.

Extrapolation to Cortex

- Assume a 5x5 mm slab of cortex containing 10,000 minicolumns.
- Each minicolumn contains 50 cluster-sensitive output neurons.
- Slab is assumed to map 10,000-bit input patterns into 10,000 bit output patterns.
- It's an associative memory.
- Patterns are assumed sparse (1% active = 100 bits)
- The slab can learn roughly 100,000 random input/output associations with near perfect accuracy.

Biological Assumptions Reviewed

1. Cluster-sensitive dendrites: a second-order activation function.
2. Gain control mechanism to limit moment-to-moment variation in the number of active synapses.
3. Combinatorially rich axo-dendritic interface structure that affords each input line access to many sites on the dendrite where synapses may be formed.
4. Process of continual synapse formation.
5. Locally-acting Hebbian learning rule capable of stabilizing or destabilizing newly-formed synapses.

Hebbian Learning Rules

Linear activation function:

$$y = \sum_{j=1}^{n} w_j x_j = w^T \bullet x$$

Simple Hebbian learning rule:

$$\Delta w(t) = \eta x(t) y(t)$$

Assuming weights change slowly relative to speed of pattern presentations, we have:

$$\langle \Delta w_i \rangle = \langle x_i y \rangle = \langle x_i \sum_{j=1}^{n} w_j x_j \rangle = \langle \sum_{j=1}^{n} x_j x_i w_j \rangle = \sum_{j=1}^{n} C_{ij} w_j$$

$$C_{ij} = \langle x_i x_j \rangle$$

Correlation Matrix

- Cij is the correlation matrix for the input vector x.
- It is equivalent to the covariance matrix when the input components have zero means.
- This is a symmetric matrix, so:
  - real eigenvalues
  - complete set of ortho-normal eigenvectors
- The eigenvectors represent synaptic weight patterns that can develop independently of one another.
- The eigenvector with the largest eigenvalue will determine the final weight patterns that develop.

Problem with the simple learning rule:
weights grow without bound.

Weight Decay (Local Constraint)

One way to limit growth of weights is to add a local decay term for each synapse (Oja 1982):

$$\Delta w_i = \eta \, [x_i y - w_i y^2]$$

- The subtrahend implements a form of LTD. It depends only on postsynaptic activity.
- This learning scheme extracts the first principal component of the training set.
- In other words, the weight vector will point in the average direction of the training set vectors.

Clipping (Local Constraint)

Limit the range of values a weight can attain. This has been proposed by many investigators:

$$\Delta w_i = \begin{cases} 0 & w < w_{\min} \\ \eta\,(x_i - \theta_{\mathrm{pre}})\,(y - \theta_{\mathrm{post}}) & w_{\min} < w < w_{\max} \\ 0 & w_{\max} < w \end{cases}$$

- When activation is below threshold θ, the difference is negative. Usually treated as zero in this case.
- Variations on the rule include nonlinear activation functions for postsynaptic term, and sliding threshold values.
- Problem with weight clipping: sensitive dependency on initial conditions.
- Final weight pattern is no longer completely determined by the principal eigenvector.

Weight Normalization (Global Constraint)

Weight update step:

$$w_i(t) = w_i(t-1) + \Delta w_i(t)$$

Multiplicative weight normalization:

$$w \leftarrow \frac{w}{\sum_i w_i}$$

Subtractive weight normalization:

$$w_i(t) \leftarrow w_i(t) - z$$

$$z = \frac{1}{n} \sum_{j=1}^{n} [w_j(t) - w_j(t-1)]$$

Problem: **each synapse requires access to entire weight vector.**

Volume Learning

- The vertebrate CNS produces nitric oxide (NO) subsequent to NMDA receptor stimulation.
- NO can diffuse through cell membranes and the synaptic cleft.
- Retrograde transmission: NO could diffuse from the postsynaptic side of a synapse to the presynaptic side.
- NO could further diffuse to other nearby synapses, permeating a local volume of tissue.
- NO can influence both synaptic plasticity (LTP) and synaptic transmission.
- Thus, NO could be a global signal useful for controlling learning.

Volume Learning Rule

$$\Delta w(t) = \eta\, [x(t) - \theta_{pre}]\, [\mu(r, t) - T_{pre}]$$

η is the learning rate

θ_{pre} is the presynaptic threshold

μ(r,t) is the local concentration of NO at distance r

T_{pre} is a presynaptic concentration threshold

Concentration of NO evolves as:

$$\frac{\partial}{\partial t}\mu(r, t) = \nabla^2\mu(r, t) - \kappa\mu(r, t) + p(r, t)$$

κ is the exponential decay rate of the substance

p(r,t) is the production rate

Problem: **can be overwhelmed by over-active inputs, because T_{pre} is fixed.**

Volume Learning Rule 2

$$\Delta w(t) = \eta\,[x(t) - \theta_{pre}]\,[\mu(r, t) - \bar{\mu}(r, t)]$$

$\bar{\mu}(r, t)$ is the time-averaged value of $\mu(r, t)$.

This formulation compensates for ambient levels of NO, so only departures from the ambient value trigger learning.

Predictive Hebbian Learning

Input cell learns to predict imminent firing of output cell, and strengthen the synaptic connection between them.

$$\Delta w(t) = \eta\,[x(t-h)]\;[\mu(r,t) - \bar{\mu}(r,t)]$$

This rule can be extended from unsupervised synaptic learning to more global reinforcement learning schemes where μ is a diffuse reward signal **r**.

Suggestion: **r** should (1)cause weight change at previously-activated synapses, but also (2) inhibit future synaptic plasticity in the region.

This prevents the reinforcement pathway from predicting the onset of input from cortical areas , which would result in backward conditioning.