Math Modeling, Week 9

Recurrent networks

Full matrix of connections (bidirectional)

w_{ij}: weight from node j to node i

b_i: bias for node i

Network state evolves over time, to some equilibrium state or stationary distribution

Hopfield network

Recurrent network model of pattern completion
- Attractor network: settles to nearest stored state
- Content-addressable memory: retrieves a pattern from partial information

Architecture
- Symmetric connections, $w_{ij} = w_{ji}$
- No self-connections, $w_{ii} = 0$
- Binary activation, ± 1

Dynamics
- Asynchronous deterministic updating
- $a_i \leftarrow \mathrm{sign}\left(\sum_j w_{ij} a_j + b_i\right)$, with i chosen randomly on each step

Energy function
- Measure of discordance of a network state (cf. Smolenky's harmony)
- $E(\mathbf{a}) = -\sum_{ij} a_i a_j w_{ij} - \sum_i a_i b_i = -\mathbf{a}^{\mathrm{T}} W \mathbf{a} - \mathbf{b}^{\mathrm{T}} \mathbf{a}$
- Update rule always reduces energy
  - $\Delta E = \Delta a_i \cdot \left(-\sum_j w_{ij} a_j - b_i\right)$: negative if a_i changed
- Attractors are local minima

Hebbian learning
- Set network to some state $\mathbf{a}$
- $\Delta w_{ij} = \frac{1}{N} a_i a_j$
- Trained patterns become attractors

Stability of trained patterns
- Train on patterns $\mathbf{a}^k$ for $k \in \{1, \dots, K\}$
  - $w_{ij} = \frac{1}{N} \sum_k a_i^k a_j^k$
- Consider network in state $\mathbf{a}^l$
- $a_i^{\mathrm{in}} = \sum_j w_{ij} a_j^l = \frac{1}{N} \sum_k \sum_j a_i^k a_j^k a_j^l = a_i^l + \frac{1}{N} \sum_{k \neq l} \sum_j a_i^k a_j^k a_j^l$
- Crosstalk term
  - Interference among patterns
  - Determines storage capacity of network, i.e. before trained patterns are no longer attractors
  - Random patterns, large N: phase transition at $K \approx .138 \cdot N$