III. Recurrent Neural Networks

A. The Hopfield Network

Typical Artificial Neuron

Equations

Net input:
\[ h_i = \sum_{j=1}^{n} w_{ij} s_j - \theta \]
\[ h = Ws - \theta \]

New neural state:
\[ s'_i = \sigma(h_i) \]
\[ s' = \sigma(h) \]

Hopfield Network

- Symmetric weights: \( w_{ij} = w_{ji} \)
- No self-action: \( w_{ii} = 0 \)
- Zero threshold: \( \theta = 0 \)
- Bipolar states: \( s_i \in \{-1, +1\} \)
- Discontinuous bipolar activation function:
\[ \sigma(h) = \text{sgn}(h) = \begin{cases} -1, & h < 0 \\ +1, & h > 0 \end{cases} \]
What to do about $h = 0$?

- There are several options:
  - $\sigma(0) = +1$
  - $\sigma(0) = -1$
  - $\sigma(0) = -1$ or $+1$ with equal probability
  - $h_i = 0 \Rightarrow$ no state change ($s_i' = s_i$)
- Not much difference, but be consistent
- Last option is slightly preferable, since symmetric

Positive Coupling

- Positive sense (sign)
- Large strength

Negative Coupling

- Negative sense (sign)
- Large strength

Weak Coupling

- Either sense (sign)
- Little strength

State = $-1$ & Local Field < 0

State = $-1$ & Local Field > 0
State Reverses

State = +1 & Local Field > 0

State = +1 & Local Field < 0

State Reverses

NetLogo Demonstration of Hopfield State Updating

Run Hopfield-update.nlogo

Hopfield Net as Soft Constraint Satisfaction System

- States of neurons as yes/no decisions
- Weights represent soft constraints between decisions
  - hard constraints must be respected
  - soft constraints have degrees of importance
- Decisions change to better respect constraints
- Is there an optimal set of decisions that best respects all constraints?
Demonstration of Hopfield Net Dynamics I

Run Hopfield-dynamics.nlogo

Convergence

- Does such a system converge to a stable state?
- Under what conditions does it converge?
- There is a sense in which each step relaxes the "tension" in the system
- But could a relaxation of one neuron lead to greater tension in other places?

Quantifying "Tension"

- If $w_{ij} > 0$, then $s_i$ and $s_j$ want to have the same sign ($s_i s_j = +1$)
- If $w_{ij} < 0$, then $s_i$ and $s_j$ want to have opposite signs ($s_i s_j = -1$)
- If $w_{ij} = 0$, their signs are independent
- Strength of interaction varies with $|w_{ij}|$
- Define disharmony ("tension") $D_{ij}$ between neurons $i$ and $j$:
  \[ D_{ij} = -s_i w_{ij} s_j \]
  - $D_{ij} < 0 \Rightarrow$ they are unhappy
  - $D_{ij} > 0 \Rightarrow$ they are happy

Total Energy of System

The "energy" of the system is the total "tension" (disharmony) in it:

\[
E \{ s \} = -\frac{1}{2} \sum_{i} \sum_{j \neq i} s_i w_{ij} s_j
\]

Another View of Energy

The energy measures the degree to which the neurons' states are in disharmony with their local fields (i.e. of opposite sign):

\[
E \{ s \} = -\frac{1}{2} \sum_{i} \sum_{j} s_i w_{ij} s_j
\]

Review of Some Vector Notation

\[
x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = (x_1, \ldots, x_n)^T \quad \text{(column vectors)}
\]

\[
x^Ty = \sum_{i=1}^{n} x_i y_i = x \cdot y \quad \text{(inner product)}
\]

\[
xy^T = \begin{pmatrix} x_1y_1 & \cdots & x_1y_n \\ \vdots & \ddots & \vdots \\ x_ny_1 & \cdots & x_ny_n \end{pmatrix} \quad \text{(outer product)}
\]

\[
x^TMx = \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} x_i x_j \quad \text{(quadratic form)}
\]
Do State Changes Decrease Energy?

- Suppose that neuron $k$ changes state
- Change of energy:
  \[
  \Delta E = E[s] - E[s'] = -\sum_{j \neq k} w_{kj} s_j + \sum_{j \neq k} w_{kj} s_j
  \]
  \[
  = -(s_k' - s_k) \sum_{j \neq k} w_{kj} s_j
  \]
  \[
  = -\Delta s_k h_k < 0
  \]

Energy Does Not Increase

- In each step in which a neuron is considered for update:
  \[E(s(t + 1)) - E(s(t)) \leq 0\]
- Energy cannot increase
- Energy decreases if any neuron changes
- Must it stop?

Proof of Convergence in Finite Time

- There is a minimum possible energy:
  - The number of possible states $s \in \{-1, +1\}^n$ is finite
  - Hence $E_{min} = \min \{E(s) | s \in \{\pm 1\}^n\}$ exists
- Must reach in a finite number of steps because only finite number of states

Conclusion

- If we do asynchronous updating, the Hopfield net must reach a stable, minimum energy state in a finite number of updates
- This does not imply that it is a global minimum

Lyapunov Functions

- A way of showing the convergence of discrete- or continuous-time dynamical systems
- For discrete-time system:
  - need a Lyapunov function $E$ ("energy" of the state)
  - $E$ is bounded below ($E(s) > E_{min}$)
  - $\Delta E < (\Delta E)_{max} \leq 0$ (energy decreases a certain minimum amount each step)
  - then the system will converge in finite time
- Problem: finding a suitable Lyapunov function

Example Limit Cycle with Synchronous Updating
The Hopfield Energy Function is Even

- A function \( f \) is odd if \( f(-x) = -f(x) \), for all \( x \)
- A function \( f \) is even if \( f(-x) = f(x) \), for all \( x \)
- Observe:

\[
E\{-s\} = \frac{1}{2}(-s)^TW(-s) = -\frac{1}{2}s^TWs = E\{s\}
\]
Part 3A: Hopfield Network

Basins in Bipolar State Space

Demonstration of Hopfield Net Dynamics II
Run initialized Hopfield.nlogo

Storing Memories as Attractors

Example of Pattern Restoration

Example of Pattern Restoration
Example of Pattern Restoration

Example of Pattern Restoration

Example of Pattern Completion

Example of Pattern Completion

Example of Pattern Completion

Example of Pattern Completion
Part 3A: Hopfield Network

Example of Pattern Completion

Example of Association

Example of Association

Example of Association

Example of Association

Example of Association
Applications of Hopfield Memory

- Pattern restoration
- Pattern completion
- Pattern generalization
- Pattern association

Hopfield Net for Optimization and for Associative Memory

- For optimization:
  - we know the weights (couplings)
  - we want to know the minima (solutions)
- For associative memory:
  - we know the minima (retrieval states)
  - we want to know the weights

Hebb’s Rule

“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.”

—Donald Hebb (The Organization of Behavior, 1949, p. 62)

“Neurons that fire together, wire together”

Example of Hebbian Learning: Pattern Imprinted

Example of Hebbian Learning: Partial Pattern Reconstruction

Mathematical Model of Hebbian Learning for One Pattern

Let \( W_{ij} = \begin{cases} x_i x_j, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \)

Since \( x_i x_i = x_i^2 = 1 \), \( W = xx^T - I \)

For simplicity, we will include self-coupling:

\( W = xx^T \)
A Single Imprinted Pattern is a Stable State

• Suppose \( W = xx^T \)
• Then \( h = Wx = xx^T x = nx \)
since \( x^T x = \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} (\pm 1)^2 = n \)
• Hence, if initial state is \( s = x \), then new state is \( s' = \text{sgn}(nx) = x \)
• For this reason, scale \( W \) by \( 1/n \)
• May be other stable states (e.g., \(-x\))

Questions

• How big is the basin of attraction of the imprinted pattern?
• How many patterns can be imprinted?
• Are there unneeded spurious stable states?
• These issues will be addressed in the context of multiple imprinted patterns

Imprinting Multiple Patterns

• Let \( x_1, x_2, \ldots, x_p \) be patterns to be imprinted
• Define the sum-of-outer-products matrix:
\[
W_{ij} = \frac{1}{p} \sum_{k=1}^{p} x_{i}^k x_{j}^k
\]
\[
W = \frac{1}{p} \sum_{k=1}^{p} (x_{i}^k)^T
\]

Definition of Covariance

Consider samples \((x_1^1, y_1^1), (x_2^2, y_2^2), \ldots, (x_N^N, y_N^N)\)
Let \( \overline{x} = \{x_i^i\} \) and \( \overline{y} = \{y_i^i\} \)
Covariance of \( x \) and \( y \) values:
\[
C_{xy} = \sum_{i=1}^{N} (x_i^i - \overline{x})(y_i^i - \overline{y})
\]
\[
= \sum_{i=1}^{N} x_i^i y_i^i - \overline{x}\overline{y} - x^i \overline{y} - x^i \overline{x} + \overline{x}\overline{y}
\]
\[
= \left( x^i \overline{y} - \overline{x} y^i + \overline{x} \overline{y} \right) - \left( x^i \overline{y} - \overline{x} y^i + \overline{x} \overline{y} \right)
\]
\[
= \left( x^i y^i - \overline{x} \overline{y} - \overline{x} \overline{y} + \overline{x} \overline{y} \right)
\]
\[
C_{xy} = \left( x^i y^i \right) - \overline{x} \overline{y}
\]

Weights & the Covariance Matrix

Sample pattern vectors: \( x_1, x_2, \ldots, x_p \)
Covariance of \( i^{th} \) and \( j^{th} \) components:
\[
C_{ij} = \sum_{k=1}^{p} x_{i}^k x_{j}^k
\]
If \( \forall i: x_i = 0 \) (\( \pm 1 \) equally likely in all positions):
\[
C_{ij} = \sum_{k=1}^{p} x_{i}^k x_{j}^k
\]
\[
\therefore \ nW = pC
\]

Characteristics of Hopfield Memory

• Distributed ("holographic")
  – every pattern is stored in every location (weight)
• Robust
  – correct retrieval in spite of noise or error in patterns
  – correct operation in spite of considerable weight damage or noise
Stability of Imprinted Memories

- Suppose the state is one of the imprinted patterns \( \mathbf{x}^m \)
- Then: 
  \[ h = \mathbf{W}\mathbf{x}^m = \left[ \frac{1}{n} \sum_{k \neq m} \mathbf{x}^k \cdot \mathbf{x}^m \right] \mathbf{x}^m \]
  \[ = \frac{1}{n} \mathbf{x}^m \cdot \mathbf{x}^m + \frac{1}{n} \sum_{k \neq m} \mathbf{x}^k \cdot \mathbf{x}^m \]
  \[ = \mathbf{x}^m + \frac{1}{n} \sum_{k \neq m} (\mathbf{x}^k \cdot \mathbf{x}^m) \mathbf{x}^k \]

**Interpretation of Inner Products**

- \( \mathbf{x}^k \cdot \mathbf{x}^m = n \) if they are identical  
  - highly correlated
- \( \mathbf{x}^k \cdot \mathbf{x}^m = -n \) if they are complementary  
  - highly correlated (reversed)
- \( \mathbf{x}^k \cdot \mathbf{x}^m = 0 \) if they are orthogonal  
  - largely uncorrelated
- \( \mathbf{x}^k \cdot \mathbf{x}^m \) measures the crosstalk between patterns \( k \) and \( m \)

**Cosines and Inner Products**

If \( \mathbf{u} \) is bipolar, then \( ||\mathbf{u}||^2 = \mathbf{u} \cdot \mathbf{u} = n \)

Hence, \( \mathbf{u} \cdot \mathbf{v} = n \cos \theta_{uv} \)

Hence \( h = \mathbf{x}^m + \sum_{k \neq m} \mathbf{x}^k \cos \theta_{km} \)

**Conditions for Stability**

**Stability of entire pattern:**

\[ \mathbf{x}^n = \text{sgn} \left( \mathbf{x}^n + \sum_{k \neq m} \mathbf{x}^k \cos \theta_{km} \right) \]

**Stability of a single bit:**

\[ x_i^n = \text{sgn} \left( x_i^n + \sum_{k \neq m} x_i^k \cos \theta_{km} \right) \]

**Sufficient Conditions for Instability (Case 1)**

Suppose \( x_i^n = -1 \). Then unstable if:

\[ (-1) + \sum_{k \neq m} x_i^k \cos \theta_{km} > 0 \]

\[ \sum_{k \neq m} x_i^k \cos \theta_{km} > 1 \]
**Sufficient Conditions for Instability (Case 2)**

Suppose $x_i^n = +1$. Then unstable if:

$$ (+1) + \sum_{k \neq m} x_i^k \cos \theta_{km} < 0 $$

$$ \sum_{k \neq m} x_i^k \cos \theta_{km} < -1 $$

**Sufficient Conditions for Stability**

$$ \left| \sum_{k \neq w} x_i^k \cos \theta_{kw} \right| \leq 1 $$

The crosstalk with the sought pattern must be sufficiently small.

**Capacity of Hopfield Memory**

- Depends on the patterns imprinted
- If orthogonal, $p_{\text{max}} = n$
  - but every state is stable $\Rightarrow$ trivial basins
- So $p_{\text{max}} < n$
- Let load parameter $\alpha = p / n$

**Single Bit Stability Analysis**

- For simplicity, suppose $x^k$ are random
- Then $x^k \cdot x^m$ are sums of $n$ random $\pm 1$
  - binomial distribution $\approx$ Gaussian
  - in range $-n, \ldots, +n$
  - with mean $\mu = 0$
  - and variance $\sigma^2 = n$
- Probability sum $> t$:
  $$ \frac{1}{2} \left[ 1 - \text{erf} \left( \frac{t}{\sqrt{2n}} \right) \right] $$

[See "Review of Gaussian (Normal) Distributions" on course website]

**Approximation of Probability**

Let crosstalk $C_{ij}^n = \frac{1}{n} \sum_{k \neq j} x_i^k (x^j \cdot x^n)$

We want $\Pr \{ C_{ij}^n > 1 \} = \Pr \{ nC_{ij}^n > n \}$

Note: $nC_{ij}^n = \frac{1}{n} \sum_{k \neq j} x_i^k x_j^n$

A sum of $n(p - 1) = np$ random $\pm 1$

Variance $\sigma^2 = np$

**Probability of Bit Instability**

$$ \Pr \{ nC_{ij}^n > n \} = \frac{1}{2} \left[ 1 - \text{erf} \left( \frac{n}{\sqrt{np}} \right) \right] $$

$$ = \frac{1}{2} \left[ 1 - \text{erf} \left( \sqrt{2} \alpha \right) \right] $$
Part 3A: Hopfield Network

**Tabulated Probability of Single-Bit Instability**

<table>
<thead>
<tr>
<th>$P_{\text{error}}$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1%</td>
<td>0.105</td>
</tr>
<tr>
<td>0.36%</td>
<td>0.138</td>
</tr>
<tr>
<td>1%</td>
<td>0.185</td>
</tr>
<tr>
<td>5%</td>
<td>0.37</td>
</tr>
<tr>
<td>10%</td>
<td>0.61</td>
</tr>
</tbody>
</table>

(Table from Hertz & al. Int. Theory Neural Comp.)

**Spurious Attractors**

- **Mixture states:**
  - sums or differences of odd numbers of retrieval states
  - number increases combinatorially with $p$
  - shallower, smaller basins
  - basins of mixtures swamp basins of retrieval states $\Rightarrow$ overload
  - useful as combinatorial generalizations?
  - self-coupling generates spurious attractors

- **Spin-glass states:**
  - not correlated with any finite number of imprinted patterns
  - occur beyond overload because weights effectively random

**Basins of Mixture States**

$$x_{mix}^{i} = \text{sgn}(x_{i}^{k_{1}} + x_{i}^{k_{2}} + x_{i}^{k_{3}})$$

**Fraction of Unstable Imprints ($n = 100$)**

**Number of Stable Imprints ($n = 100$)**

**Number of Imprints with Basins of Indicated Size ($n = 100$)**
Summary of Capacity Results

- Absolute limit: $p_{\text{max}} < \alpha_c n = 0.138 n$
- If a small number of errors in each pattern permitted: $p_{\text{max}} \propto n$
- If all or most patterns must be recalled perfectly: $p_{\text{max}} \propto n / \log n$
- Recall: all this analysis is based on random patterns
- Unrealistic, but sometimes can be arranged