Restrict Boltzmann Machines

- Generative model, capable of learning latent representations
- \( v \): observable binary data (e.g., an image), \( h \): latent binary vectors

\[
p(x = (v, h)) = \frac{1}{Z} e^{-E(x)}
\]
\[Z = \sum_x e^{-E(x)}, \quad E(x) = -v^T W h\]

Maximum Likelihood Training

\[
\hat{\nabla}_W L = \left[ \frac{1}{N} \sum_i \mathbb{E}[h|v_i] \right] \quad \text{easy to compute}
\]

Markov–Chains Monte Carlo

- Sample \( x^{(t+1)} \sim p(x|v^{(t)}) \), randomly initialized \( x^{(0)} \)
- \( \mathbb{E}[vh^T] \) estimated from \( v_i^T h_i^{(t)} \), with fixed \( K \)
- Asymptotic unbiased-ness requires steady state, not guaranteed
- Contrastive Divergence: initialize \( x^{(0)} \) from Training data

Markov-Chains Las Vegas

- Strong Markov Property
- Tours are Sample Paths from the Steady State Distribution

\[
T := x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow x^{(3)} \rightarrow x^{(4)} \rightarrow x^{(5)} \rightarrow x^{(6)} = x^{(0)}
\]

\[
e^{-E(x)} \sum_{x \in \text{high prob states}} vh^T
\]

is an unbiased estimate of \( Z \cdot \mathbb{E}[vh^T] \).
- Similarly, \( e^{-E(x)}|T| \) is an estimate of the partition function \( Z \).
- Exactly Sampled from the steady state, but Random Running Time
- Uses all states in the chain to compute estimate

Stopping Sets

\[
\uparrow P(r(S)) \Rightarrow \downarrow \text{running time}
\]

\[
\text{Relax tour stopping condition: require } x^{(t)} \in S \text{ instead of } x^{(t)} = x^{(0)}
\]

Tour Behavior

- Extremely Heavy Tail
- \( P(\text{tour length} = 1) > 99\% \) for 32 hidden neurons

\[
\text{Shorter tours } \Rightarrow \text{Memorized data}
\]

- Longer tours \( \Rightarrow \) Better generalization

- Ignoring long tours aids generalization

Gradient Estimates (Las Vegas Slope)

Stopping Set \( \equiv \) Training Data (high probability states)

- Use Tours which finish in \( \leq K \) steps \( \Rightarrow \) limit max. running time
- Heavy tail allows ignoring long tours
- Better training of RBMs on MNIST

\[
n_H = 32
\]

<table>
<thead>
<tr>
<th>Method</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD-1</td>
<td>-167.3 (2.7)</td>
<td>-166.6 (2.8)</td>
</tr>
<tr>
<td>CD-2</td>
<td>-153.0 (4.9)</td>
<td>-152.1 (4.7)</td>
</tr>
<tr>
<td>CD-3</td>
<td>-147.8 (0.5)</td>
<td>-147.0 (0.5)</td>
</tr>
<tr>
<td>VLS-1</td>
<td>-133.8 (0.5)</td>
<td>-133.5 (0.5)</td>
</tr>
<tr>
<td>VLS-2</td>
<td>-146.7 (0.5)</td>
<td>-146.5 (0.5)</td>
</tr>
<tr>
<td>VLS-3</td>
<td>-147.0 (1.1)</td>
<td>-147.1 (1.2)</td>
</tr>
</tbody>
</table>

MNIST test log-likelihood (higher is better)

<table>
<thead>
<tr>
<th>Method Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_H = 32 )</td>
</tr>
<tr>
<td>MLE</td>
</tr>
<tr>
<td>MCMC</td>
</tr>
<tr>
<td>MCLV</td>
</tr>
<tr>
<td>LVS</td>
</tr>
</tbody>
</table>

Future Work

- Apply techniques to directed models (Variational models and GANs)

Code Repository

https://github.com/PurdueMINDS/MCLV-RBM