Restricted Boltzmann Machines

- Generative model, capable of learning latent representations
- $v$: observable binary data (e.g. an image), $h$: latent binary vectors
- $p(x = (v, h)) = \frac{1}{Z} e^{-E(x)}, \quad Z = \sum_x e^{-E(x)}, \quad E(x) = -v^T Wh$

Maximum Likelihood Training

$V_w L = \frac{1}{N} \sum_{n=1}^{N} v_n [E[h|v_n]]^{\text{easy}} - [E[vh]]^{\text{intractable}}$

Markov-Chain Monte Carlo

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

Markov-Chain Las Vegas

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

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

$\left( e^{-E(x^{(0)})} \sum_{x \in \{h|v_h\}} vh \right)$

is an unbiased estimate of $Z \cdot E[vh]$. Similarly, $e^{-E(x^{(0)})}|T|$ 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

- Relax tour stopping condition: require $x^{(i)} \in S$ instead of $x^{(i)} = x^{(0)}$
- $\uparrow \Pr(S) \Rightarrow \downarrow \text{running time}$
- Increases exact stopping set gradient computation time

Tour Behavior

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

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Training</th>
<th>Testing</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>
<td>-169.8 (2.6)</td>
<td>-169.0 (2.6)</td>
</tr>
<tr>
<td>PCD-1</td>
<td>-153.0 (4.9)</td>
<td>-152.1 (4.7)</td>
<td>-147.8 (0.5)</td>
<td>-147.0 (0.5)</td>
</tr>
<tr>
<td>LVS-1</td>
<td>-134.0 (1.0)</td>
<td>-133.3 (1.0)</td>
<td>-138.3 (1.3)</td>
<td>-137.5 (1.4)</td>
</tr>
<tr>
<td>CD-10</td>
<td>-154.3 (3.3)</td>
<td>-153.4 (3.3)</td>
<td>-156.4 (0.5)</td>
<td>-155.6 (0.5)</td>
</tr>
<tr>
<td>PCD-10</td>
<td>-139.3 (3.2)</td>
<td>-138.5 (3.3)</td>
<td>-147.4 (0.5)</td>
<td>-146.7 (0.5)</td>
</tr>
<tr>
<td>LVS-10</td>
<td>-133.3 (1.0)</td>
<td>-132.6 (1.0)</td>
<td>-138.1 (1.1)</td>
<td>-137.4 (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>MLE</td>
</tr>
<tr>
<td>Running Time Gradients</td>
</tr>
<tr>
<td>States per estimate</td>
</tr>
<tr>
<td>Confidence intervals</td>
</tr>
</tbody>
</table>

Code Repository

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