Boltzmann Machines

Stochastic generalization of Hopfield net.

Energy: $E = -\sum_{ij} J_{ij} x_i x_j - \sum_i h_i x_i$

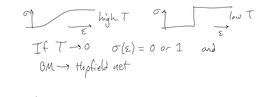
Boltzmann distribution: $Q(x) = \frac{e^{-\frac{1}{T}E(x)}}{Z}$

$Z = \sum_x e^{-\frac{1}{T}E(x)}$ "partition function" $\beta = \frac{1}{T}$ "inverse temperature"

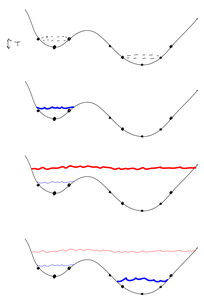
Hopfield update: $x_{i,t+1} = \operatorname{argmin}_{y^i} E(y^i, x_{\neq i}^t)$

Boltzmann update: $Q(x_{i,t+1} = +1) = \sigma\left(\frac{1}{T} E(x_{i=+1}, x_{\neq i}^t)\right)$

"glauber dynamics" or "Gibbs Sampling" $\sigma(\varepsilon) = \frac{1}{1+e^{-\varepsilon}}$

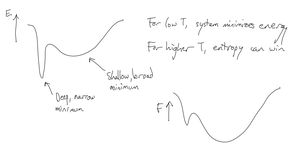


If $T \to 0$ $\sigma(\varepsilon) = 0$ or 1 and BM → Hopfield net



With $T > 0$, Energy is no longer minimized.
But Free Energy is!

$F = U - TS$ ← Entropy
Helmholtz Free Energy ← Total energy $\langle E \rangle$ ← Temperature



Interpretation: a system has a higher probability of exploring many higher energy states than staying in a few lower energy states.

$Q(x) = e^{\Delta E/T}$ so if there are more than $e^{\Delta E/T}$ higher E states, the system will usually live there.

Learning: Try to make $Q(x)$ match a target $P(x)$.

Define a cost function: Kullback-Leibler divergence

$D_{KL}[P \| Q] = \sum_x P(x) \log \frac{P(x)}{Q(x)}$

$D_{KL} \geq 0$ with $=$ iff $P = Q$.

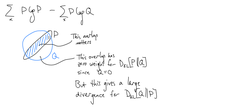
Example:

$D_{KL}[p(x,y) \| p(x)p(y)] = I_{x;y}$

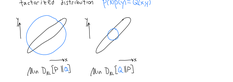
$D_{KL}[\delta_{x_0} \| p(x)] = -\log p(x_0)$

$D_{KL}[Q \| P] \neq D_{KL}[P \| Q] \Rightarrow$ not a distance

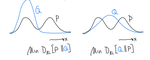
$\sum_x P \log P - \sum_x P \log Q$



Try to approximate $P(x,y)$ with the best factorized distribution $P(x)P(y) = Q(x,y)$



Another example:



Minimize D_{KL} with respect to parameters of $E(x; h, J)$

$$0 = \frac{\partial D_{KL}[P\|Q]}{\partial J_{ij}} = \frac{\partial}{\partial J_{ij}}\left[\sum_x P \log P - \sum_x P \log Q\right]$$

$$= \frac{\partial}{\partial J_{ij}}\left[-\sum_x P\left(-\log Z + \frac{1}{T}E(x)\right)\right]$$

$$= \frac{\partial}{\partial J_{ij}}\left[\log Z + \sum_x P \frac{1}{T}\frac{\partial}{\partial J_{ij}}\left(-\sum_{ij} J_{ij} x_i x_j - \sum_i h_i x_i\right)\right]$$

$$= \frac{1}{Z}\frac{\partial Z}{\partial J_{ij}} - \frac{1}{T}\sum_x P\, x_i x_j$$

$$= \frac{1}{Z}\frac{\partial}{\partial J_{ij}}\sum_x e^{-\frac{1}{T}E(x;J)} - \frac{1}{T}\sum_x P\, x_i x_j$$

$$= \frac{1}{Z}\sum_x e^{-\frac{1}{T}E}\frac{\partial E}{\partial J_{ij}}\left(-\frac{1}{T}\right) - \frac{1}{T}\langle x_i x_j\rangle_P$$

$$= \sum_x Q\left(\frac{-x_i x_j}{-T}\right) - \frac{1}{T}\langle x_i x_j\rangle_P$$

$$\frac{\partial D_{KL}}{\partial J_{ij}} = \frac{1}{T}\left(\langle x_i x_j\rangle_Q - \langle x_i x_j\rangle_P\right)$$

$$\frac{\partial D_{KL}}{\partial h_i} = \frac{1}{T}\left(\langle x_i\rangle_Q - \langle x_i\rangle_P\right)$$

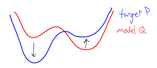
To minimize cost, we could thus set $\langle x_i x_j\rangle_Q = \langle x_i x_j\rangle_P$

It's hard to specify J_{ij} to produce a given $\langle x_i x_j\rangle_Q$.

Instead, we can descent the gradient of D_{KL}:

$J_{ij}^{t+1} = -\alpha\left(\langle x_i x_j\rangle_Q - \langle x_i x_j\rangle_P\right)$

This lowers the energy of x where $Q < P$ and raises it where $Q > P$.

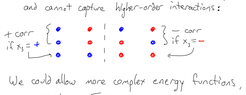


We still need to estimate $\langle x_i x_j\rangle_Q$, which is also hard...

This is accomplished by sampling, or sometimes by approximation (e.g. mean field, assuming neurons are independent)

Why is it hard to just calculate $\langle x_i x_j\rangle$ from $Q(x)$?

The second-order BM is limited in its expressiveness, and cannot capture higher-order interactions:



We could allow more complex energy functions, e.g.

$E(x) = -\sum_{ijk} K_{ijk} x_i x_j x_k$

to capture such effects, but direct triple interactions are not generally neurally plausible.

Instead, we can introduce HIDDEN NODES to capture more complex dependencies.

This architecture can accomodate any distribution over the 2^V possible states of the V visible units

How can we learn this?

Same rule as before, except that the hidden units are not specified by the target distribution:

$\Delta J_{ij} = -\alpha\left[\langle x_i x_j\rangle_{\text{clamped}} - \langle x_i x_j\rangle_{\text{free}}\right]$

clamped: visible units are set to $x \sim P(x)$ and for each sample, hidden units are given by $Q(x_{\text{hidden}} | x_{\text{visible}})$.

free: all $x \sim Q(x)$

This is UNSUPERVISED learning:
no teacher specifies the "correct answers" for the hidden units. The network finds hidden unit connections (latent variables) that best explain the distribution of the input data.

The architecture, however, must be specified, as fully general BMs (fully connected) are intractably slow.

⇒ Restricted Boltzmann Machine

This makes sampling faster, at a loss of generality.
Still need MANY tricks to train a deep BM.

One interesting trick is to stop before equilibrium.
Start at a point from the data, x^0, and only sample for a short time.

- Contrastive divergence
- Minimum probability flow

Another perspective:

What is the distribution $Q(x)$ that matches some specified moments, e.g. $\left(\langle x_i x_j\rangle = A_{ij}\right)$ but is otherwise maximally uncertain $\left(\langle x_i\rangle = B_i\right)$?

Find maximum entropy subject to constraints.

$H = -\sum_x Q \log Q$

$C = H - \sum_{ij} \alpha_{ij}\left(\sum_x Q x_i x_j - A_{ij}\right) - \sum_i \beta_i\left(\sum_x Q x_i - B_i\right) - \gamma\left(\sum_x Q - 1\right)$

$\frac{\partial C}{\partial Q(x)} = 0 = -(1 + \log Q) - \sum_{ij} \alpha_{ij} x_i x_j - \sum_i \beta_i x_i - \gamma$

$Q = e^{-(1+\gamma)} - \sum_{ij} \alpha_{ij} x_i x_j - \sum_i \beta_i x_i$

$= \frac{e^{-\sum_{ij} \alpha_{ij} x_i x_j - \sum_i \beta_i x_i}}{Z}$ $Z = e^{1+\gamma}$

Again the Boltzmann machine!
(if $x \in \pm 1$; if $x \in \mathbb{R}$, then this is gaussian)

Example: http://www.cs.toronto.edu/~hinton/digits.html