8. Restricted Boltzmann machines

- Restricted Boltzmann machines
- Learning RBMs
- Deep Boltzmann machines
- Learning DBMs

- Unsupervised learning
- Given i.i.d. samples $\{v^{(1)}, \ldots, v^{(n)}\}$, learn the joint distribution $\mu(v)$

Restricted Boltzmann machines

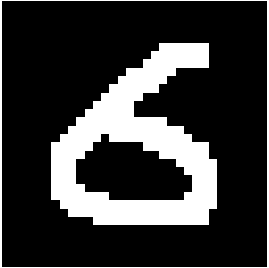
- motivation for studying deep learning
  - concrete example of parameter leaning
  - applications in dimensionality reduction, classification, collaborative filtering, feature learning, topic modeling
  - successful in vision, language, speech
  - unsupervised learning: learn a generative model or a distribution
- running example: learn distribution over hand-written digits (cf. character recognition)
  - 32×32 binary images $v^{(\ell)} \in \{0,1\}^{32\times 32}$

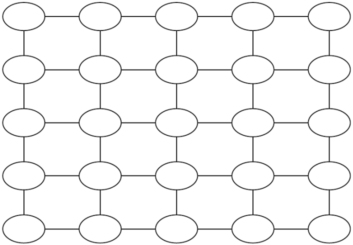
Graphical model

- grid

$$\mu(v) = \frac{1}{Z} \exp\left\{ \sum_i \theta_i v_i + \sum_{(i,j)\in E} \theta_{ij} v_i v_j \right\}$$

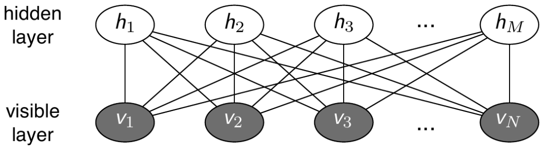
a sample $v^{(\ell)}$





Restricted Boltzmann machine [Smolensky 1986] "harmoniums"

- undirected graphical model
- two layers: visible layer $v \in \{0,1\}^N$ and hidden layer $h \in \{0,1\}^M$
- fully connected bipartite graph

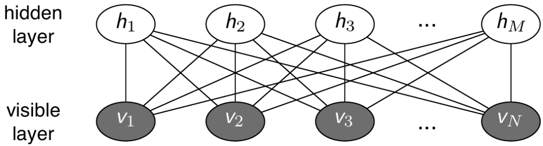


- RBMs represented by parameters $a \in \mathbb{R}^N$, $b \in \mathbb{R}^M$, and $W \in \mathbb{R}^{N \times M}$ such that

$$\mu(v,h) = \frac{1}{Z} \exp\left\{ a^T v + b^T h + v^T W h \right\}$$

- consider the marginal distribution $\mu(v)$ of the visible nodes, then any distribution on v can be modeled arbitrarily well by RBM with $k-1$ hidden nodes, where k is the cardinality of the support of the target distribution

Restricted Boltzmann machine



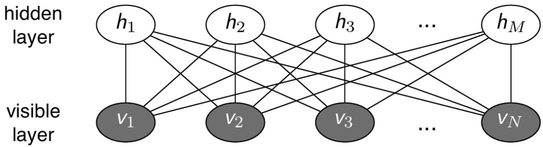
define energy

$$E(v,h) = -a^T v - b^T h - v^T W h$$

$$\begin{aligned} \mu(v,h) &= \frac{1}{Z}\exp\{-E(v,h)\} \\ &= \frac{1}{Z}\prod_{i\in[N]} e^{a_i v_i} \prod_{j\in[M]} e^{b_j h_j} \prod_{(i,j)\in E} e^{v_i W_{ij} h_j} \end{aligned}$$

note that RBM has no intra-layer edges (hence the name **restricted**)
(cf. Boltzmann machine [Hinton,Sejnowski 1985])

Restricted Boltzmann machine



it follows from the Markov property of MRF that the conditional distribution a product distribution

$$\begin{aligned} \mu(v|h) &\propto \exp\left\{(a + Wh)^T v\right\} = \prod_i \exp\left\{(a_i + \langle W_{i\cdot}, h\rangle)\, v_i\right\} \\ &= \prod_{i\in[N]} \mu(v_i|h) \\ \mu(h|v) &= \prod_{j\in[M]} \mu(h_j|v) \end{aligned}$$

hence, inference in the conditional distribution is efficient
e.g. compute a conditional marginal $\mu(h_i|v)$

since $\mu(h|v) = \prod \mu(h_j|v)$,

$$\begin{aligned}\mu(h_j = 0|v) &= \frac{\mu(h_j = 0|v)}{\mu(h_j = 0|v) + \mu(h_j = 1|v)} \\ &= \frac{1}{1 + e^{b_j + \langle W_{\cdot j}, v\rangle}} \\ \mu(h_j = 1|v) &= \underbrace{\frac{1}{1 + e^{-(b_j + \langle W_{\cdot j}, v\rangle)}}}_{\triangleq \sigma(b_j + \langle W_{\cdot j}, v\rangle) \text{ is a sigmiod}}\end{aligned}$$

similarly,

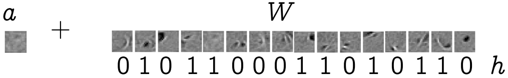
$$\mu(v_i = 1|h) = \sigma(a_i + \langle W_{i\cdot}, h\rangle)$$

where $W_{i\cdot}$ and $W_{\cdot j}$ are i-th row and j-th column of W resepectively, and $\langle \cdot, \cdot \rangle$ denotes the inner product

- one interpretation of RBM is to think of it as a stochastic version of a neural network, where nodes and edges correspond to neurons and synaptic connections, and a single node fires (i.e. $v_i = +1$) stochastically from a sigmoid activation function $\sigma(x) = 1/(1 + e^{-x})$,

$$\mu(v_i = +1|h) = \frac{e^{a_i + \langle W_{i\cdot}, h \rangle}}{1 + e^{a_i + \langle W_{i\cdot}, h \rangle}} = \sigma(a_i + \langle W_{i\cdot}, h \rangle)$$

- another interpretation is to think of the bias a and the components $W_{\cdot j}$ connected to a hidden node h_j encode higher level structure



- if the goal is to sample from this distribution (perhaps to generate images that resemble hand-written digits), the lack of inter-layer connection makes Gibbs sampling particularly easy, since one can apply block Gibbs sampling for each layer all together from $\mu(v|h)$ and $\mu(h|v)$ iteratively

- What about computing the marginal $\mu(v)$?

$$\begin{aligned}
\mu(v) &= \sum_{h\in\{0,1\}^M} \mu(v,h) \\
&= \frac{1}{Z}\sum_h \exp\{-E(v,h)\} \\
&= \frac{1}{Z}\exp\left\{a^T v + \sum_{j=1}^M \log(1+e^{b_j+\langle W_{\cdot j},v\rangle})\right\}
\end{aligned}$$

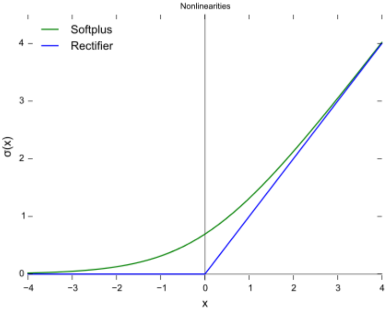
because

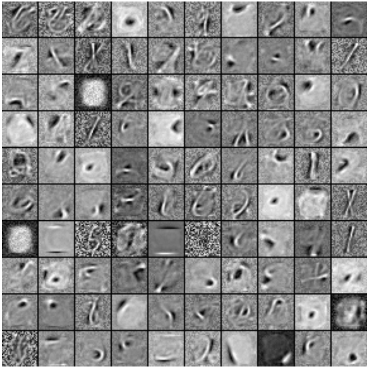
$$\begin{aligned}
\sum_{h\in\{0,1\}^M} e^{a^T v+b^T h+v^T Wh} &= e^{a^T v}\sum_{h\in\{0,1\}^M} e^{b^T h+v^T Wh} \\
&= e^{a^T v}\sum_{h\in\{0,1\}^M}\prod_{j=1}^M e^{b_j h_j+\langle v,W_{\cdot,j}\rangle h_j} \\
&= e^{a^T v}\prod_{j=1}^M\sum_{h_j\in\{0,1\}} e^{b_j h_j+\langle v,W_{\cdot,j}\rangle h_j} \\
&= \cdots
\end{aligned}$$

$$\mu(v) = \frac{1}{Z}\exp\left\{a^T v + \sum_{j=1}^{M}\log(1 + e^{b_j + \langle W_{\cdot j}, v\rangle})\right\}$$

$$= \frac{1}{Z}\exp\left\{a^T v + \sum_{j=1}^{M}\mathrm{softplus}(b_j + \langle W_{\cdot j}, v\rangle)\right\}$$

a_i is the bias in v_i, b_j is the bias in h_j, and $W_{\cdot j}$ is the output "image" v resulting from the hidden node h_j

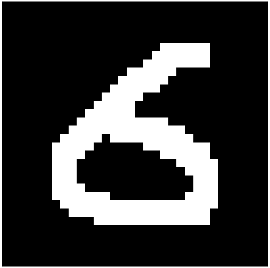




- in the spirit of unsupervised learning, how do we extract features such that we achieve, dimensionality reduction?
  - suppose we have learned the RBM, and have a, b, W at our disposal
  - we want to compute features of a new sample $v^{(\ell)}$
  - we use the conditional distribution of the hidden layers as features

$$\mu(h|v^{(\ell)}) \; = \; [\mu(h_1|v^{(\ell)}), \cdots, \mu(h_M|v^{(\ell)})]$$

a sample $v^{(\ell)}$



[0,.2,0,.3,0,0,0,0,0,.8,0,0,0,.4,1,0]

extracted features

Parameter learning

- given i.i.d. samples $v^{(1)}, \ldots, v^{(n)}$ from the marginal $\mu(v)$, we want to learn the RBM using maximum likelihood estimator

$$\mu(v) = \frac{1}{Z} \sum_{h \in \{0,1\}^M} \exp\left\{ a^T v + b^T h + v^T W h \right\}$$

$$\mathcal{L}(a, b, W) = \frac{1}{n} \sum_{\ell=1}^{n} \log \mu(v^{(\ell)})$$

- Gradient ascent (although not concave, and multi-modal)
  - ★ consider a single sample likelihood

$$\log \mu(v^{(\ell)}) = \log \sum_h \exp(a^T v^{(\ell)} + b^T h + (v^{(\ell)})^T W h) - \log Z$$

$$\frac{\partial \log \mu(v^{(\ell)})}{\partial b_i} = \frac{\sum_h h_i \, e^{a^T v^{(\ell)} + b^T h + (v^{(\ell)})^T W h}}{\sum_h e^{a^T v^{(\ell)} + b^T h + (v^{(\ell)})^T W h}} - \frac{\sum_{v,h} h_i \, e^{a^T v + b^T h + v^T W h}}{Z}$$

$$= \underbrace{\mathbb{E}_{\mu(h|v^{(\ell)})}[h_i]}_{\textbf{data-dependent expectation}} - \underbrace{\mathbb{E}_{\mu(v,h)}[h_i]}_{\textbf{model expectation}}$$

$$\frac{\partial \log \mu(v^{(\ell)})}{\partial a_i} = v_i^{(\ell)} - \mathbb{E}_{\mu(v,h)}[v_i]$$

$$\frac{\partial \log \mu(v^{(\ell)})}{\partial W_{ij}} = \mathbb{E}_{\mu(h|v^{(\ell)})}[v_i^{(\ell)} h_j] - \mathbb{E}_{\mu(v,h)}[v_i h_j]$$

- once we have the gradient update a, b, and W as

$$W_{ij}^{(t+1)} = W_{ij}^{(t)} + \alpha \underbrace{\left(\mathbb{E}_{\mu(h|v)q(v)}[v_i h_j] - \mathbb{E}_{\mu(v,h)}[v_i h_j]\right)}_{\frac{\partial \log \mu}{\partial W_{ij}}}$$

  $q(v)$ is the empirical distribution of the training samples $v^{(1)}, \ldots, v^{(n)}$
- to compute the gradient, we need to compute the second term $\mathbb{E}_{\mu(v,h)}[v_i h_j]$ requires inference
  - one approach is to approximately compute the expectation using samples from MCMC, but in general can be quite slow for large network
  - custom algorithms designed for RBMs: contrastive divergence, persistent contrastive divergence, parallel tempering

MCMC (block Gibbs sampling)

for computing

$$\mathbb{E}_{\mu(v,h)}[v_i h_j]$$

where μ is defined by given current parameters $a^{(t)}$, $b^{(t)}$, $W^{(t)}$

- MCMC (block Gibbs sampling)
  - start with samples $\mathcal{V} = \{v^{(1)}, \ldots v^{(K)}\} \in \{0,1\}^{N \times K}$ drawn from the data (i.i.d. uniformly at random with replacement)
  - repeat
    - sample $\mathcal{H} = \{h^{(1)} \ldots, h^{(K)}\} \in \{0,1\}^{M \times K}$ each from $\mu(h^{(k)}|v^{(k)})$ for $k \in [K]$
    - sample $\mathcal{V} = \{v^{(1)}, \ldots, v^{(K)}\} \in \{0,1\}^{N \times K}$ each from $\mu(v^{(k)}|h^{(k)})$ for $k \in [K]$
  - compute the estimate $\frac{1}{K}\sum_{k=1}^{K} v_i^{(k)} h_j^{(k)}$
- eventually, if repeated long enough (longer than mixing time), this converges to an unbiased estimate of $\mathbb{E}_{\mu(v,h)}[v_i h_j]$.
- how does it compare to BP?

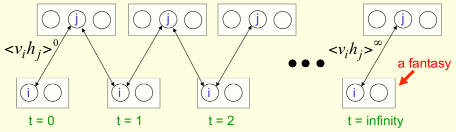
Contrastive Divergence [IPAM tutorial, Hinton 2002]

- a standard approach for learning RBMs
- start sampling chain at $v^{(\ell)}$ from each of the training sample
- run Gibbs sampling for k steps to get $(v(k), h(k))$
- large k reduces bias, but in practice $k = 1$ works well

- k-step contrastive divergence
  - ★ **Input:** Graph G over v, h, training samples $S = \{v^{(1)}, \dots, v^{(n)}\}$
  - ★ **Output:** gradient $\{\Delta w_{ij}\}_{i \in [N], j \in [M]}, \{\Delta a_i\}_{i \in [N]}, \{\Delta b_j\}_{j \in [M]}$

1. initialize $\Delta w_{ij}, \Delta a_i, \Delta b_j = 0$
2. **Repeat**
3. **for all** $v^{(\ell)} \in S$
4. $v(0) \leftarrow v^{(\ell)}$
5.   **for** $t = 0, \dots, k-1$ **do**
6.   **for** $i = 1, \dots, N$ **do** sample $h(t)_i \sim \mu(h_i | v(t))$
7.   **for** $j = 1, \dots, M$ **do** sample $v(t+1)_j \sim \mu(v_j | h(t))$
8.   **for** $i = 1, \dots, N, j = 1, \dots, M$ **do**
9. $\Delta w_{ij} \leftarrow \Delta w_{ij} + \mathbb{E}_{\mu(h_j | v(0))}[h_j v(0)_i] - \mathbb{E}_{\mu(h_j | v(k))}[h_j v(k)_i]$
10. $\Delta a_i \leftarrow \Delta a_i + v(0)_i - v(k)_i$
11. $\Delta b_j \leftarrow \Delta b_j + \mathbb{E}_{\mu(h_j | v(0))}[h_j] - \mathbb{E}_{\mu(h_j | v(k))}[h_j]$
12. **End for.**

Persistent MCMC/Persistent CD [Teleman 2008]



- practical parameter learning algorithms for RBM use heuristics to approximate the log-likelihood gradient to speed up
- use persistent Markov chains to speed up the Markov chain
  - jointly update *fantasy particles* $\mathcal{S}^{(t)} = \{(h, v)^{(k,t)}\}_{k \in [K]}$ and parameters $\theta^{(t)} = (a^{(t)}, b^{(t)}, W^{(t)})$ by repeating
    - ⋆ fix $(a^{(t)}, b^{(t)}, W^{(t)})$ and sample the next fantasy particles $\mathcal{S}^{(t)} = \{(h, v)_{k,(t+1)}\}_{k \in [K]}$ according to a Markov chain
    - ⋆ update $(a^{(t)}, b^{(t)}, W^{(t)})$

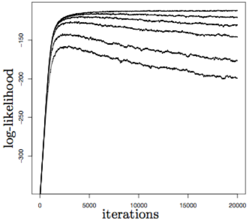
$$W_{ij}^{(t+1)} = W_{ij}^{(t)} + \alpha_t \left(\mathbb{E}_{\mu(h|v)q(v)}[v_i h_j] - \mathbb{E}_{\mu(v,h)}[v_i h_j] \right)$$

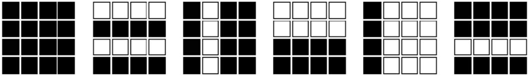
by computing the model expectation (the second term) via $\mathbb{E}_{\mu(v,h)}[v_i h_j] \simeq \frac{1}{K} \sum_{k=1}^{K} (v_i^{(k,t+1)} h_j^{(k,t+1)})$

- since the Markov chain is run for finite k (often k=1 in practice), the resulting approximation of the gradient is biased (is dependent on the training samples $\{v^{(\ell)}\}$)
- **Theorem.**

$$\frac{\log \mu(v(0))}{\partial w_{ij}} = \Delta w_{ij} + \mathbb{E}_{\mu(h|v)P(v(k))}[v(k)_i h_j] - \mathbb{E}_{\mu(v,h)}[v_i h_j]$$

- the gap vanishes as $k \to \infty$ and $p(v(k)) \to \mu(v)$
- in general for finite k, the bias can distort learning procedure and converge to a solution that is not the maximum-likelihood





contrastive divergence with $k = 1, 2, 5, 10, 20, 100$ (bottom to up) and 16 hidden nodes and training data from 4×4 bars-and-stripes [1]

[1][“An Introduction to Restricted Boltzmann Machines”, Fischer, Igel] [“Training

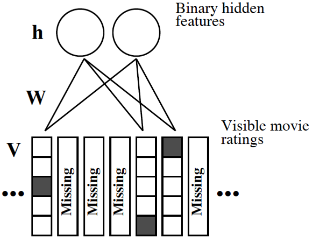
Example: collaborative filtering

- Restricted Boltzmann machines for collaborative filtering [Salakhutdinov, Mnih, Hinton '07]
  - suppose there are M movies and N users
  - each user provides 5-star ratings for a subset of movies

18,000 movies (columns) × 480,000 users (rows)

x	1	1	x	...	x
x	x	x	5	...	x
x	x	3	x	...	x
x	4	3	x	...	2
...	x	x	x	...	x
x	5	x	1	...	x
x	x	3	3	...	x
x	1	x	x	...	2

  - model each user as a restricted Boltzmann machine with different hidden variables
  - equivalently, treat each user as an independent sample from RBM



- for a user who rated m movies let $V \in \{0,1\}^{K \times m}$ be observed ratings where $v_i^k = 1$ if the user gave rating k to movie i
- let h_j be the binary valued hidden variables for $j = \{1, \ldots, J\}$
- users have different RBM's but share the same weights W_{ij}^k
- RBM

$$\mu(V, h) = \frac{1}{Z} \exp\left\{ \sum_{k=1}^{5} (V^k)^T W^k h + \sum_{k=1}^{5} (a^k)^T V^k + b^T h \right\}$$

- learning: compute the gradient for each user and average over all users
  $W_{ij}^k(t+1) = W_{ij}^k(t) + \alpha_t(\mathbb{E}_{\mu(h|V_{\text{sample}})}[V_i^k h_j] - \mathbb{E}_{\mu(V,h)}[V_i^k h_j])$

- RBM

$$\mu(V,h) \;=\; \frac{1}{Z}\exp\Big\{\sum_{k=1}^{5}(V^k)^T W^k h + \sum_{k=1}^{5}(a^k)^T V^k + b^T h\Big\}$$

- prediction
  - predicting a single movie v_{m+1} is easy

$$\mu(v^r_{m+1}=1|V) \propto \sum_{h\in\{0,1\}^J} \mu(v^r_{m+1}=1, v^{\bar{r}}_{m+1}=0, V, h)$$

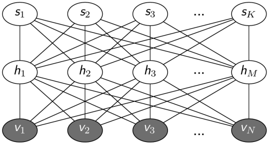
$$\propto \sum_{h\in\{0,1\}^J} \exp\Big\{\sum_{k=1}^{5}(V^k)^T W^k h + a^r_{m+1} v^r_{m+1} + \sum_j W^r_{m+1,j} h_j + b^T h\Big\}$$

$$\propto C_r \prod_{j=1}^{J} \sum_{h_j\in\{0,1\}^J} \exp\Big\{\sum_{i,k} V^k_i W^k_{ij} h_j + W^r_{m+1,j} h_j + b_j h_j\Big\}$$

  - however, predicting L movies require 5^L evaluations
  - instead approximate it by computing $\mu(h|V)$ and using it to evaluate $\mu(v_\ell|h)$
- on $17,770 \times 480,189$ Netflix dataset, $J=100$ works well

Deep Boltzmann machine

- deep Boltzmann machine

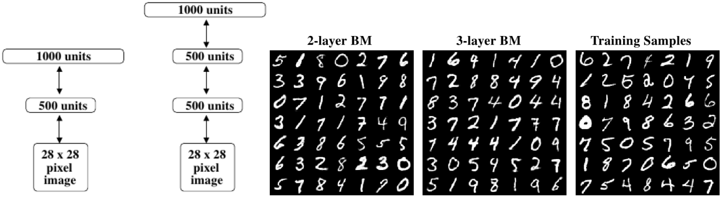


$$\mu(v, h, s) \quad = \quad \frac{1}{Z} \exp\left\{a^T v + b^T h + c^T s + v^T W^1 h + h^T W^2 s\right\}$$

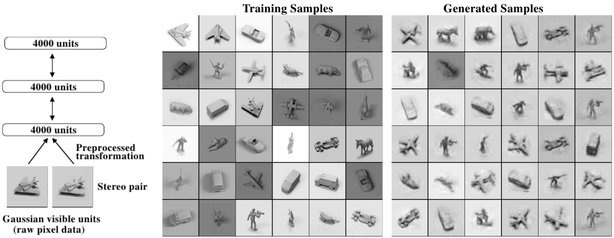
- capable of learning higher level and more complex representations

Deep Boltzmann machine [Salakhutdinov, Hinton '09]

- MNIST dataset with 60,000 training set
  - Samples generated by running Gibbs sampler for 100,000 steps



- NORB dataset with 24,300 stereo image pairs training set
  - 25 toy objects in 5 classes (car, truck, plane, animal, human)



- (stochastic) gradient ascent (drop a, b, c for simplicity)

$$
\begin{aligned}
\mathcal{L}(W^1, W^2) &= \log \mu(v^{(\ell)}) \\
&= \log \sum_{h,s} \exp\left\{ (v^{(\ell)})^T W^1 h + h^T W^2 s \right\} - \log Z
\end{aligned}
$$

- gradient

$$
\frac{\partial \log \mu(v^{(\ell)})}{\partial W^1_{ij}} =
$$

$$
\frac{\sum_{h,s} (v_i^{(\ell)} h_j)\, e^{(v^{(\ell)})^T W^1 h + h^T W^2 s}}{\sum_{h,s} e^{(v^{(\ell)})^T W^1 h + h^T W^2 s}} - \frac{\sum_{v,h,s} (v_i^{(\ell)} h_j)\, e^{(v^{(\ell)})^T W^1 h + h^T W^2 s}}{Z}
$$

$$
= \mathbb{E}_{\mu(h|v^{(\ell)})}[v_i^{(\ell)} h_j] - \mathbb{E}_{\mu(v,h)}[v_i h_j]
$$

$$
\frac{\partial \log \mu(v^{(\ell)})}{\partial W^2_{ij}} = \mathbb{E}_{\mu(h,s|v^{(\ell)})}[h_i s_j] - \mathbb{E}_{\mu(h,s)}[h_i s_j]
$$

- problem: now even the data-dependent expectation is difficult to compute

Variational method

- use naive mean-field approximation to get a lower bound on the objective function

$$\begin{aligned}\mathcal{L}(W^1, W^2) &= \log \mu(v^{(\ell)}) \\ &\geq \log \mu(v^{(\ell)}) - D_{\mathrm{KL}}\big(b(h,s) \,\big\|\, \mu(h,s|v^{(\ell)})\big) \\ &= \sum_{h,s} b(h,s) \log \mu(h,s,v^{(\ell)}) + H(b) \\ &\equiv \mathbb{G}(b, W^1, W^2, v^{(\ell)})\end{aligned}$$

- instead of maximizing $\mathcal{L}(W^1, W^2)$, maximize $\mathbb{G}(b, W^1, W^2, v^{(\ell)})$ to get approximately optimal W^1 and W^2
- constrain $b(h,s) = \prod_i p_i(h_i) \prod_j q_j(s_j)$ for simplicity
- and let $p_i = p_i(h_i = 1)$ and $q_j = q_j(s_j = 1)$

$$\mathbb{G} = \sum_{k,i} W^1_{ki} v^{(\ell)}_k p_i + \sum_{i,j} W^2_{ij} p_i q_j - \log Z + \sum_i H(p_i) + \sum_j H(q_j)$$

  - fix (W^1, W^2) and find maximizers $\{p_i^*\}, \{q_j^*\}$
  - fix $\{p_i^*\}, \{q_i^*\}$ and update (W^1, W^2) according to gradient ascent

$$W^2_{ij}(t+1) = W^2_{ij}(t) + \alpha(p_i^* \, q_j^* - \mathbb{E}_{\mu(h,s)}[h_i s_j])$$

$$\mathbb{G} = \sum_{k,i} W^1_{ki} v^{(\ell)}_k p_i + \sum_{i,j} W^2_{ij} p_i q_j - \log Z + \sum_i H(p_i) + \sum_j H(q_j)$$

- how do we find $\{p^*_i\}$ and $\{q^*_j\}$?
- gradient

$$\begin{aligned}
\frac{\partial \mathbb{G}}{\partial p_i} &= \sum_k W^1_{ki} v^{(\ell)}_k + \sum_j W^2_{ij} q_j + \log p_i - \log(1 - p_i) = 0 \\
\frac{\partial \mathbb{G}}{\partial q_j} &= \sum_i W^2_{ij} p_i + \log q_j - \log(1 - q_j) = 0
\end{aligned}$$

$$\begin{aligned}
p^*_i &= \frac{1}{1 + e^{-\sum_k W^1_{ki} v^{(\ell)}_k - \sum_j W^2_{ij} q^*_j}} \\
q^*_j &= \frac{1}{1 + e^{-\sum_i W^2_{ij} p^*_i}}
\end{aligned}$$