0 Energy models

1.
   - definition
   - ICA case, generative vs. energy model

2. Learning
   - exact (hybrid sampling)
   - CD
   - score matching

3. DBN
   - RBM,
   - complementary prior $\Leftrightarrow$ infinite BN w. tied weights

     $\Leftrightarrow$ RBM

   - greedy learning
   - wake-sleep phase

4. PoE, FoE

5. Further reading

Yee Whye Teh et al

④ Learning in energy models

$p^0(\underline{x})$ real distribution of the observed data

$p^\infty(\underline{x};\theta)$ model distribution $= \frac{1}{Z(\theta)} \exp(-E(x;\theta))$

$p^\infty_\theta(\underline{x})$

we want to minimise

$$KL(p^0 \| p^\infty_\theta) = \int p^0(x) \log \frac{p^0(x)}{p^\infty(x)} dx$$

$$= cte + \cancel{\langle} \log Z(\theta) \cancel{\rangle_{p^0}} + \langle E(x;\theta) \rangle_{p^0}$$

$$\frac{\partial KL(p^0 \| p^\infty_\theta)}{\partial \theta} = \cancel{\langle} \frac{\partial}{\partial \theta} \log Z(\theta) \cancel{\rangle_{p^0}} + \left\langle \frac{\partial E(x;\theta)}{\partial \theta} \right\rangle_{p^0}$$

$$\frac{\partial}{\partial \theta} \log Z(\theta) = \frac{1}{Z(\theta)} \frac{\partial}{\partial \theta} \int \exp(-E(x;\theta))\, dx$$

$$= \int \underbrace{\frac{1}{Z(\theta)} \exp(-E(x;\theta))}_{p^\infty(x;\theta)} \left(-\frac{\partial E(x;\theta)}{\partial \theta}\right) dx$$

$$= -\left\langle \frac{\partial E(x;\theta)}{\partial \theta} \right\rangle_{p^\infty_\theta}$$

$$\Rightarrow \frac{\partial KL}{\partial \theta} = \left\langle \frac{\partial E(x;\theta)}{\partial \theta} \right\rangle_{p^0} - \underbrace{\left\langle \frac{\partial E(x;\theta)}{\partial \theta} \right\rangle_{p^\infty_\theta}}_{\text{estimate using MCMC (Hamiltonian)}}$$

- intractable (partition function)

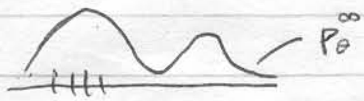
+ consistent
+ easily adaptable to more complex models
- it takes long to reach equilibrium distr.
- variance of MCMC estimator is usually high

⑤ Contrastive Divergence

Two ideas: 1) start the MC at the data distr. p^0
rather than at some vague distr.

data lies on some low-dim manifold

danger



there is no pressure
to remove empty modes

⇒ ~~can ...~~ it's hard to estimate
the volume of this mode (normalizing constant)

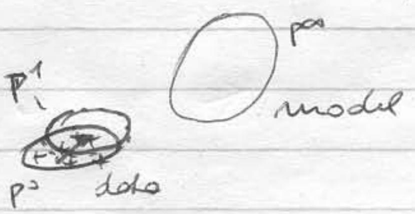
2) run the MC for only a few iterations rather
than until equilibrium

② idea: if the model is correct $p_\theta^\infty = p^0$, then MCMC does not change the distr.

interpretation: remove any consistent tendency

④ to move away from the data distribution

③ (depending of course from MCMC method)



① $$\Delta\theta \propto -\left\langle \frac{\partial E(x)}{\partial\theta} \right\rangle_{p^0} + \left\langle \frac{\partial E(x)}{\partial\theta} \right\rangle_{p^n}$$

<u>n typically very small (1)</u>

CD learning

~~A. compute $\frac{\partial E(x)}{\partial\theta}$, average over data set $\underline{x}_k$~~

1\. 2. run MCMC sampler for n steps starting
at data $\underline{x}_k \rightarrow \underline{s}_k = T^n \underline{x}_k$

~~3. Compute $\frac{\partial E(x)}{\partial\theta}$, average over $\underline{s}_k$~~

2\. 4. update parameters using

$$\Delta w_{ij} = -\frac{\eta}{N}\left(\sum_{\text{data}} \frac{\partial E(\underline{x}_k)}{\partial\theta} - \sum_{\text{samples}} \frac{\partial E(\underline{s}_k)}{\partial\theta}\right)$$

+∞ amount of data

If p^0 is in the space of p_θ^∞, and the MC mixes good
enough ⇒ fixed point of ML solution

In general, it has been shown that there is a bias,
but in practice it is small.

⑥

The update rule correspond to an approximate gradient descent step on obj. function

$$CD = KL(p^0 \| p^\infty) - KL(p^n \| p^\infty)$$

$$\frac{\partial CD}{\partial \theta} = \underbrace{\left\langle \frac{\partial E(x)}{\partial \theta} \right\rangle_{p^0} - \left\langle \frac{\partial E(x)}{\partial \theta} \right\rangle_{p^n}} - \frac{\partial KL(p^n \| p^\infty)}{\partial p^n} \underbrace{\frac{\partial p^n}{\partial \theta}}$$

effect of change of parameters on the MC
→ empirically found to be small

one can do the same also if there are latent vars!

~~Example: CD learning of RBM params~~

give solution for RBM

<u>Score Matching</u>

An easy way to learn a subset of energy models without performing MCMC

$$p(x;\theta) = \frac{1}{Z(\theta)} q(x;\theta)$$

$$\psi(x;\theta) := \nabla_x \log p(x;\theta) = \begin{pmatrix} \frac{\partial \log p(x;\theta)}{\partial x_1} \\ \vdots \\ \frac{\partial \log p(x;\theta)}{\partial x_n} \end{pmatrix}$$

"score function" improper term

$$= \nabla_x \log q(x;\theta)$$ → does not depend on $Z(\theta)$

$$\psi_0(x) := \nabla_x \log p^0(x)$$

Score matching:

$$\text{minimize } J(\theta) := \frac{1}{2} \int dx\, p^0(x) \overbrace{\| \psi(x;\theta) - \underbrace{\psi_0(x)} \|^2}^{*}$$

$$\hat{\theta} = \arg\min_\theta J(\theta)$$

this term looks difficult to compute

* makes score function of model similar to that of data → not clear what does that mean if ψ_0 is not in $\psi(\cdot,\theta)$ space

(7) Theorem 1

If $\psi(x;\theta)$ differentiable + other reg. conditions

p° diff, $\mathbb{E}_x(\|\psi(x,\theta)\|^2)$ and $\mathbb{E}_x(\|\psi_0(x)\|^2)$ finite $\forall\theta$

$p^\circ(x)\psi(x;\theta) \xrightarrow[\|x\|\to\infty]{} 0 \quad \forall\theta$

$$(2) \quad \Rightarrow J(\theta) = \int p^\circ(x) \sum_{i=1}^{n} \Big(\underbrace{\partial_i \psi_i(x;\theta)}_{\frac{\partial^2 \log q(x;\theta)}{\partial x_i^2}} + \frac{1}{2}\psi_i(x;\theta)^2 \Big) dx + cte$$

$J(\theta)$ depends only on simple expectations of certain functions <u>of the non-normalized pdf</u>

Proof

$$J(\theta) = \frac{1}{2}\int p^\circ(x)\Big(\|\psi(x;\theta)\|^2 + \underbrace{\cancel{\|\psi_0(x)\|^2}}_{\text{does not depend on } \theta} - 2\psi(x;\theta)^T\psi_0(x)\Big)\,dx$$

$$= \int p^\circ(x)\Big(\underbrace{\frac{1}{2}\sum_i \psi_i(x;\theta)^2}_{\text{second term above}} - \underbrace{\psi(x;\theta)^T\psi_0(x)}_{*}\Big)\,dx$$

$$* = \sum_i \int p^\circ(x)\,\psi_i(x;\theta)\,\psi_i^\circ(x)\,dx$$

$$= \sum_i \int p^\circ(x) \frac{\partial \log p^\circ(x)}{\partial x_i} \psi_i(x;\theta)\,dx$$

$$= \sum_i \int \frac{\cancel{p^\circ(x)}}{\cancel{p^\circ(x)}} \frac{\partial p^\circ(x)}{\partial x_i} \psi_i(x;\theta)\,dx \qquad \int u'v = uv - \int uv'$$

$$= -\int p^\circ(x)\,\partial_i \psi_i(x;\theta)\,dx \qquad \square$$

partial integration, one index at the time

For energy models $\psi_i(x;\theta) = \frac{\partial}{\partial x_i} \log \exp -E(x)$

$$\Rightarrow J(\theta) = \Big\langle \sum_i -\frac{\partial^2 E(x)}{\partial x_i^2} + \frac{1}{2}\Big(\frac{\partial}{\partial x_i}E(x)\Big)^2 \Big\rangle_{p^\circ}$$

(8)

<u>Theorem 2</u> Assume 1) $p^\circ(\cdot) = p(\cdot\,; \theta^*)$ for some θ^* (almost everywhere)

2) the model is non-degenerate, i.e.

$$\nexists\ \theta^{**} : p(\cdot\,;\theta^{**}) = p(\cdot\,;\theta^*)$$

3) $\forall x, \theta \ : \ q(x;\theta) > 0$

$$\Rightarrow \quad J(\theta) = 0 \Leftrightarrow \theta = \theta^*$$

<u>Proof:</u> $J(\theta) = 0 \underset{\substack{q>0 \\ \Rightarrow p^\circ(x)>0}}{\Rightarrow} \psi^\circ(\cdot) = \psi(\cdot\,;\theta)$

$$\Rightarrow \log p^\circ(\cdot) = \log p(\cdot\,;\theta) \underbrace{+c}_{=0\ (\text{pdfs})} \qquad \square$$

$\Rightarrow$ (local) consistence

- In which sense p_θ becomes more similar to p° if p° does not lie in the p_θ's space?

  Eq. 2 $\Rightarrow$ To minimize J, the first term should be negative $\Rightarrow$ maximum of $\log p(x;\theta)$
  { second term $\Rightarrow$ as steep a maximum as possible

- CD is more general since it is applicable to <u>latent variable models</u>

extensions : binary vars ("ratio matching")
non negative data

Example

(1) Multivariate Gaussian density

$$p(\underline{x}; \underline{\underline{M}}, \underline{\mu}) = \frac{1}{Z(\underline{\underline{M}}, \underline{\mu})} \exp -\frac{1}{2}(\underline{x}-\underline{\mu})^T \underline{\underline{M}} (\underline{x}-\underline{\mu})$$

$$\psi(\underline{x}; \underline{\underline{M}}, \underline{\mu}) = \nabla_{\underline{x}} \; -\frac{1}{2}(x-\mu)^T M (x-\mu)$$

$$= -M(x-\mu)$$

$$\partial_i \psi_i(x) = -m_{ii}$$

$$\Rightarrow \tilde{J}(\underline{\underline{M}}, \underline{\mu}) = \frac{1}{T} \sum_{t=1}^{T} \left(\sum_i -m_{ii} + \frac{1}{2}(x_t-\mu)^T M^T M (x_t-\mu) \right)$$

$$\nabla_{\mu} \tilde{J} = MM\mu - MM \frac{1}{T} \sum_t x_t$$

$$= 0 \quad \Leftrightarrow \quad \mu = \sum_t x_t$$

M pos def

$$\nabla_M \tilde{J} = -I + M \frac{1}{2T} \sum_{t=1}^{T} (\underline{x}_t - \underline{\mu})(\underline{x}_t - \underline{\mu})^T$$

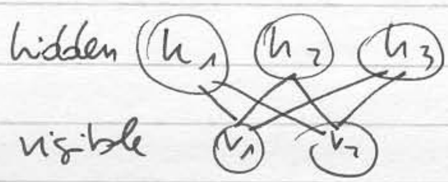
$$+ \frac{1}{2T} \left(\sum_t (\underline{x}_t - \underline{\mu})(\underline{x}_t - \underline{\mu})^T \right) M$$

$$= 0 \quad \Leftrightarrow \quad M^{-1} = \frac{1}{T} \sum_{t=1}^{T} (\underline{x}_t - \mu)(\underline{x}_t - \mu)^T$$

estimators are ~~the~~ the same as ML for any sample, not just asymptotically

(2) ICA (model, results)

(9e) Restricted Boltzmann Machine



bipartite: no hid-hid
or vis-vis connections

⇒ ~~conditional independence~~

$$E(\underline{v},\underline{h}) = -\sum_{ij} v_i h_j w_{ij} - \sum_i b_i^v v_i - \sum_j b_j^h h_j$$

activation rules:

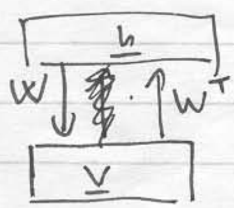
$$p(h_j = 1) = \sigma\left(b_j^h + \sum_i v_i w_{ij}\right), \quad \sigma(a) = \frac{1}{1+e^{-a}} \text{ sigmoid}$$

$$p(v_i = 1) = \sigma\left(h_i^v + \sum_j h_j w_{ij}\right)$$

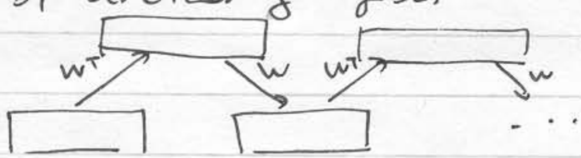
implement Gibbs sampling on $\frac{1}{Z} \exp -E(x)$

<u>Inference</u>: given $\underline{v}$, one can generate h_i independently because of <u>conditional indep</u>

same w. h



sample from equilibrium distribution $p_W(\underline{v},\underline{h})$ by alternating Gibbs



<u>Learning</u> ↗ $\arg\max_{W,b} \log \prod_n p(\underline{v}^{(n)}|W,b) = \arg\max \log \prod_n \sum_{\underline{h}} p(\underline{v}^{(n)},\underline{h}|W,b)$

ML learning: $$\frac{\partial}{\partial w_{ij}} \sum_n \ln p(\underline{v}^{(n)}|W,b) = \langle v_i h_j \rangle_{p^0} - \langle v_i h_j \rangle_{p^\infty}$$

$$\frac{\partial}{\partial b_i^v} \ldots = \langle v_i \rangle_{p^0} - \langle v_i \rangle_{p^\infty}$$

$$\frac{\partial}{\partial b_j^h} \ldots = \langle h_j \rangle_{p^0} - \langle h_j \rangle_{p^\infty}$$

can be efficiently learning using CD

ML learning of RBM

~~Products of experts~~

$$E(\underline{v},\underline{h}) = -\sum_i b_i^v v_i - \sum_j b_j^h h_j - \sum_{ij} v_i h_j w_{ij}$$

$$p(\underline{v},\underline{h}) = \frac{1}{Z} \exp\left(-E(\underline{v},\underline{h})\right)$$

$$p(\underline{v}) = \frac{1}{Z} \sum_{\underline{h}} \exp\left(-E(\underline{v},\underline{h})\right)$$

ML learning

$$\max_{\underline{b},\underline{\underline{W}}} \log \prod_n p(\underline{v}^{(n)} \mid W, b)$$

$$= \sum_n \ln \sum_{\underline{h}} p(\underline{v}^{(n)}, \underline{h} \mid W, b)$$

$$= \sum_n \left(\ln \sum_{\underline{h}} \exp\left(-E(\underline{v},\underline{h})\right) - \ln Z(W) \right)$$

$$\frac{\partial}{\partial w_{ij}} \ln Z(W,b) = \sum_{\underline{v},\underline{h}} \underbrace{\frac{1}{Z(W,b)} \exp\left(-E(\underline{v},\underline{h})\right)}_{= P(\underline{v},\underline{h} \mid W, b)} \cdot \underbrace{\left(-\frac{\partial}{\partial w_{ij}} E(\underline{v},\underline{h})\right)}_{v_i h_j}$$

$$= \sum_{\underline{v},\underline{h}} P(\underline{v},\underline{h} \mid W, b) \cdot v_i h_j$$

$$= \langle v_i h_j \rangle_{model}$$

$$\frac{\partial}{\partial w_{ij}} \ln \sum_{\underline{h}} \exp\left(-E(\underline{v}^{(n)},\underline{h})\right)$$

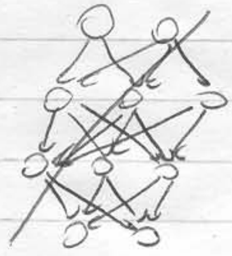
$$= \sum_{\underline{h}} \underbrace{\frac{1}{\sum_{\underline{h}} \exp\left(-E(\underline{v}^{(n)},\underline{h})\right)} \exp\left(-E(\underline{v}^{(n)},\underline{h})\right)}_{} \cdot v_i h_j$$

$$= \sum_{\underline{h}} P(\underline{h} \mid W, b, \underline{v}^{(n)})$$

$$\Rightarrow \boxed{\begin{aligned} \frac{\partial}{\partial w_{ij}} \sum_n \ln p(\underline{v}^{(n)} \mid W, b) &= \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \\ \frac{\partial}{\partial b_i^v} \quad -\text{''}- &= \langle v_i \rangle_{data} - \langle v_i \rangle_{model} \\ \frac{\partial}{\partial b_j^h} \quad -\text{''}- &= \langle h_j \rangle_{data} - \langle h_j \rangle_{model} \end{aligned}}$$

(10) Deep Belief Networks

Energy models and CD can be used to learn efficiently directed belief networks with many layers



draw ⊗

logistic belief network
binary units
generate:

$$\sigma\left(-b_i - \sum_j h_j^{(l+1)} w_{ij}^{l}\right)$$

$$p(h_i^{(l)} = 1 \mid h^{(l+1)}) = \cancel{\frac{1}{1 + \exp\left(-b_i - \sum_j h_j^{(l+1)} w_{ij}^{l}\right)}}$$

Learning is very difficult:

1 - The probability of the observed data is a very complicated fct of the params

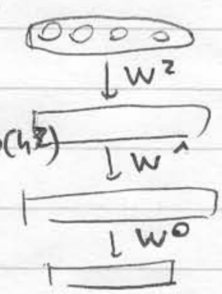
2 - inference is a nightmare because of explaining away → the posterior dist. is typically intractable

(MCMC, variational methods)

It would be nice to learn one layer at the time in a greedy way, but:

① $p(\underline{v}, \underline{h}^0) = p(\underline{v} \mid \underline{h}^0)\, \underbrace{p(\underline{h}^0)}$

$$\sum_{h_2, h_3} p(h^0 \mid h^1)\, p(h^1 \mid h^2)\, p(h^2)$$



need to integrate over all possible configurations of higher layers to get the prior → p. 11

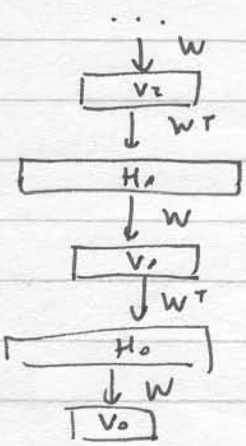
② the weights interact

③ the posterior is going to be intractable, anyway

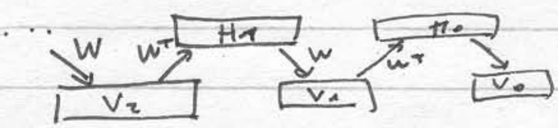
(12)

Let's consider a simpler architecture where this problems are not an issue, and then relax that ~~constraint~~ back to the general case. Maybe that can give us a clever way to initialize ~~these~~ the ~~models~~ parameters.

Infinite directed model with tied weights



generating from this model is equivalent to let an RBM with connections W reach its equilibrium distribution:



$\Rightarrow$ the class of ~~dist~~ distribution that you can model with both is the same

Learning an infinite model w. tied weight $\equiv$ learn a RBM

$\Rightarrow$ prior on h_0 is complementary, inference is easy (use W^T)

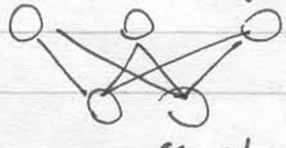
---

alternative derivation: let's say we want to have a complementary prior on h_0

$$\Rightarrow \quad p(\underline{h}_0 | \underline{v}_0) = \prod_j p(h_j | \underline{v}_0)$$

$$p(\underline{v}_0 | \underline{h}_0) = \prod_i p(v_i | \underline{h}_0) \quad \text{by construction}$$

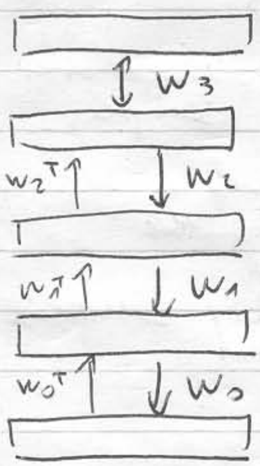
as an undirected graphical model these conditional indep. correspond to an RBM



start from that, unroll it $\rightarrow$ infinite model we've seen: infinite ↔ RBM

Greedy learning for DBN

consider hybrid model



equivalent to assuming ∞-many layers with tied weights

1. Learn W_0 assuming all weight matrices are tied (RBM)

2. Freeze W_0 and commit ourselves to using W_0^T to infer ~~[illegible]~~ the state in 1st hidden layer even if subsequent changes in higher-level weights mean that this inference method is no longer correct

3. Keeping all higher-weight matrices tied to each other, learn a model for the higher-level data that is represented in h^0

"learning by rerepresenting"

Back-fitting with Up-Down algorithm

Greedy learning is efficient but not optimal.

⇒ untie the recognition weights from the generative weights, retain the restriction that the posterior must be approx. by a factorial distr.

Refine the ~~[illegible]~~ weights using a CD variant of the wake-sleep algorithm:

① up-pass: use recognition weights to pick states for hidden variables; adjust generative weights:

$$\Delta w_{ij}^{n+1 \to n} \propto \langle \underbrace{h_j^{n+1}}_{\text{collected in up-pass}} (h_i^n - \underbrace{\hat{h}_i^n}_{\text{reconstructed by sampling → probability}}) \rangle$$ average over sample states ⊛

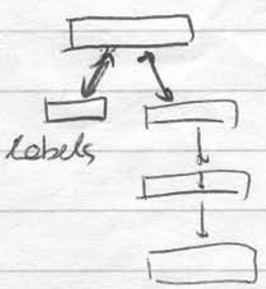
The weight of the top are learned as before using RBM

2 down-pass: start with state of the top-level associative memory, use generative weights to get states for hidden and visible layers; adjust recognition weights

(*)
equivalent to wake-sleep if state taken from equilibrium dist. of top RBM. Here: init top RBM w. up-pass, run only a few iterations of gibbs sampling, then start down step

+ faster
+ eliminate mode averaging

Classification using DBN



softmax labels: exactly one unit is 1

$$p_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$ → probability of picking i

the learning rules are unaffected by the competition

→ website demo!

performance on MNIST database: 1,25%
SVM 1,4% non linear FDA 1,4%

best: 0,5% (I think)