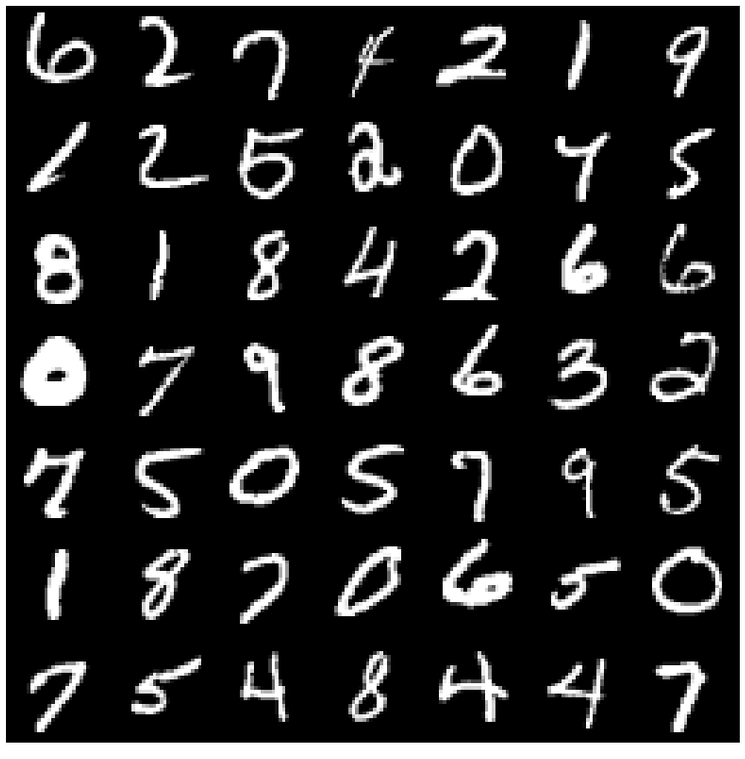
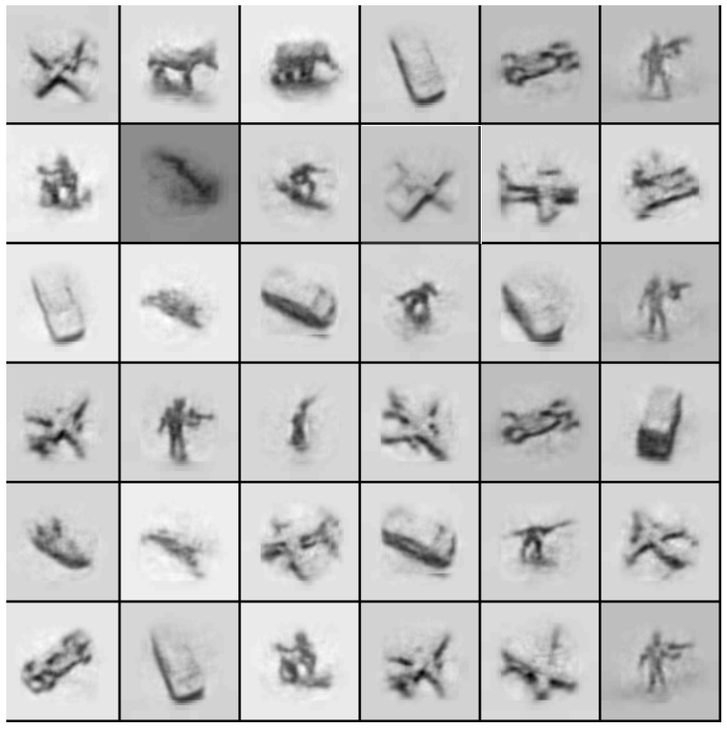
Scaling up natural gradient by factorizing Fisher information

Roger Grosse

Introduction

- Restricted Boltzmann machines (RBMs) and related models have had a lot of success at modeling complex visual datasets



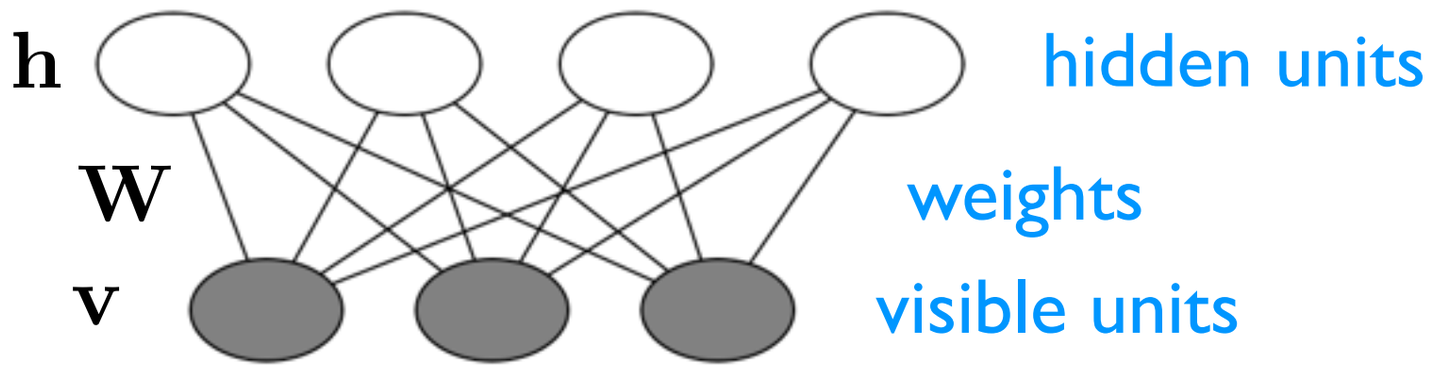


Introduction

- Training RBMs is still a black art for several reasons
  - We can't evaluate the likelihood since this requires the intractable partition function
  - We can't compute the gradient exactly
  - The optimization problem has ill-conditioned curvature since small changes to the model parameters can dramatically change the distribution it represents
- I will focus on the third issue

Introduction

- **RBMs** are an undirected graphical model:



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

Z is the partition function, which is intractable to compute exactly

Introduction

- The log-likelihood gradient is a difference of two expectations:

$$\frac{\partial}{\partial w_{ij}} \log p(\mathbf{x}) = \mathbb{E}_{p(\mathbf{h}|\mathbf{x})}[x_i h_j] - \mathbb{E}_{p(\mathbf{v},\mathbf{h})}[v_i h_j]$$

conditional distribution given data (easy)

model distribution (hard)

- Many clever approximations to the model statistics, e.g.
  - contrastive divergence
  - persistent contrastive divergence
  - tempered transitions

Introduction

- It's expensive to approximate the gradient well, so we'd like to accomplish more per iteration
- Second-order optimization methods find better updates by accounting for curvature
- Usually the practical ones either:
  - use a cheap, tractable approximation (e.g. diagonal or block diagonal)
  - require an expensive iterative procedure (e.g. Hessian-free optimization of neural nets)
- I will propose a second-order optimization method for RBMs which approximates the curvature well, but is only slightly more expensive than SGD

Outline

- Information geometry background
  - exponential families
  - Fisher information matrix
- Factorized Natural Gradient (FANG) for RBM training
  - understanding the curvature of maximum likelihood for RBMs
  - approximating the curvature using a Gaussian graphical model
  - how to efficiently invert the approximate curvature
  - experiments on training RBMs as generative models

Background: (Linear) exponential families

$$p(\mathbf{x}) = \frac{1}{\mathcal{Z}(\boldsymbol{\eta})} h(\mathbf{x}) \exp\left(\boldsymbol{\eta}^T \mathbf{g}(\mathbf{x})\right)$$

sufficient statistics $\mathbf{g}(\mathbf{x})$

natural parameters $\boldsymbol{\eta}$

moments $\mathbf{s} = \mathbb{E}[\mathbf{g}(\mathbf{x})]$

partition function $\mathcal{Z}(\boldsymbol{\eta})$

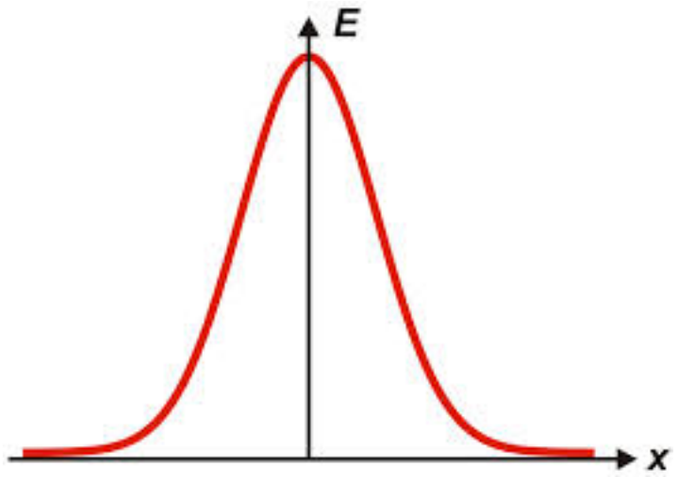
Background: (Linear) exponential families

$$p(\mathbf{x}) = \frac{1}{\mathcal{Z}(\boldsymbol{\eta})} h(\mathbf{x}) \exp\left(\boldsymbol{\eta}^T \mathbf{g}(\mathbf{x})\right)$$

Bernoulli

$$\mathbf{g}(x) = x$$

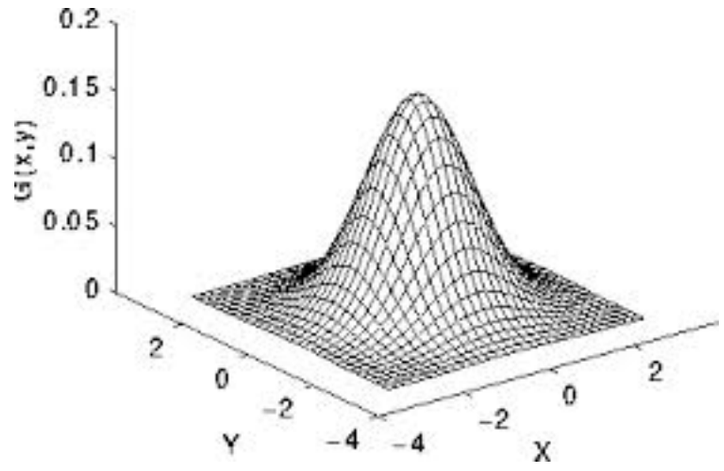
$$\boldsymbol{\eta} = \log \mu - \log(1-\mu)$$



Gaussian

$$\mathbf{g}(x) = \left(x, -\frac{1}{2}x^2\right)$$

$$\boldsymbol{\eta} = (h, \lambda)$$

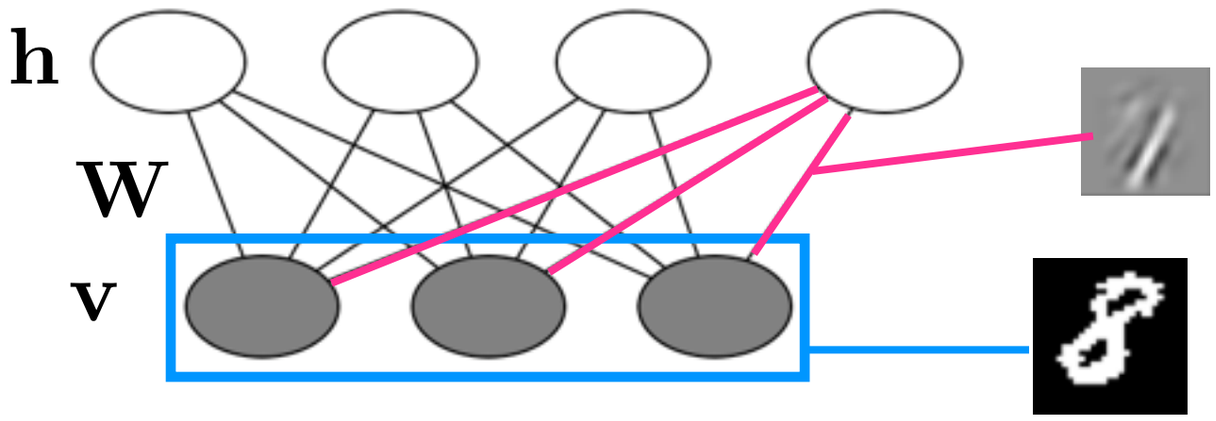


Multivariate Gaussian

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} \mathbf{x} \\ -\frac{1}{2}\,\mathrm{vec}(\mathbf{x}\mathbf{x}^T) \end{pmatrix}$$

$$\boldsymbol{\eta} = \begin{pmatrix} \mathbf{h} \\ \mathrm{vec}(\boldsymbol{\Lambda}) \end{pmatrix}$$

Background: RBMs as exponential families



joint distribution:

$$p(\mathbf{v}, \mathbf{h}) \propto \exp\left(\mathbf{a}^T\mathbf{v} + \mathbf{b}^T\mathbf{h} + \mathbf{v}^T\mathbf{W}\mathbf{h}\right)$$

sufficient statistics:

$$\mathbf{g}(\mathbf{v}, \mathbf{h}) = \begin{pmatrix} \mathbf{v} \\ \mathbf{h} \\ \mathrm{vec}(\mathbf{v}\mathbf{h}^T) \end{pmatrix}$$

Two parameterizations:

$$\eta = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathrm{vec}(\mathbf{W}) \end{pmatrix} \qquad \mathbf{s} = \mathbb{E}[\mathbf{g}(\mathbf{v}, \mathbf{h})] = \begin{pmatrix} \mathbb{E}[\mathbf{v}] \\ \mathbb{E}[\mathbf{h}] \\ \mathrm{vec}\left(\mathbb{E}[\mathbf{v}\mathbf{h}^T]\right) \end{pmatrix}$$

inference

learning

(fully observed)

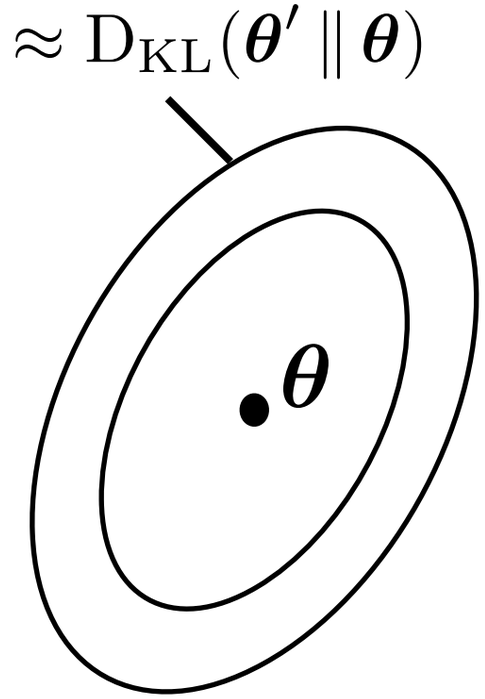
natural parameters

moments

Background: Fisher information matrix

$$\mathbf{G}_{\boldsymbol{\theta}} \triangleq \nabla^2_{\boldsymbol{\theta}'} \mathrm{D}_{\mathrm{KL}}(\boldsymbol{\theta}' \,\|\, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}'=\boldsymbol{\theta}}$$

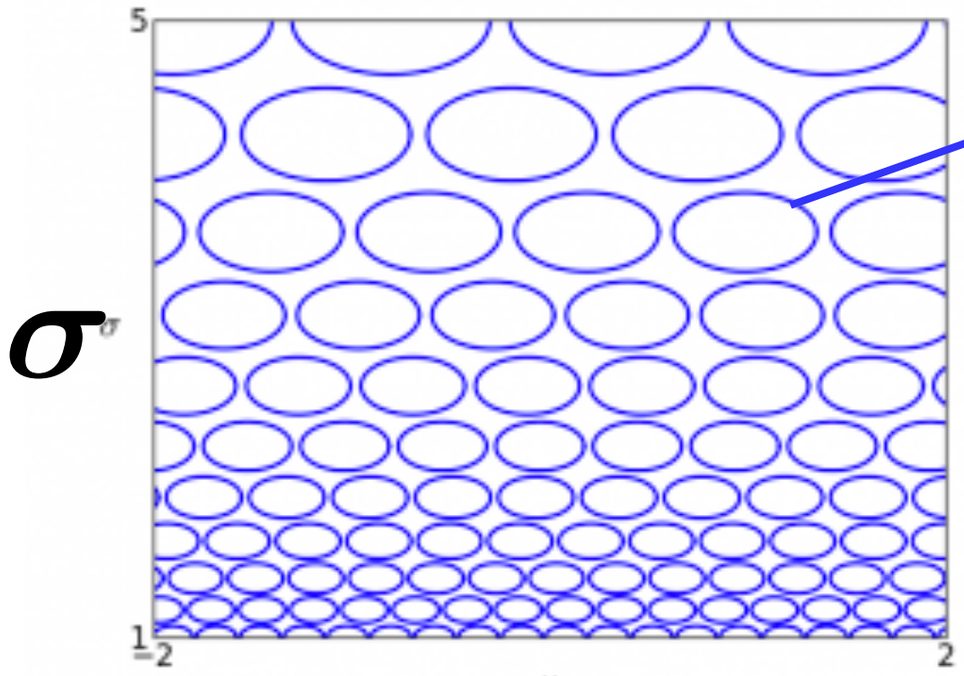
$$\mathrm{D}_{\mathrm{KL}}(\boldsymbol{\theta}' \,\|\, \boldsymbol{\theta}) \approx \frac{1}{2}(\boldsymbol{\theta}' - \boldsymbol{\theta})^T \boldsymbol{G}_{\boldsymbol{\theta}} (\boldsymbol{\theta}' - \boldsymbol{\theta})$$



Aside: KL divergence is like ***squared*** distance

Background: Fisher information matrix

mean and variance

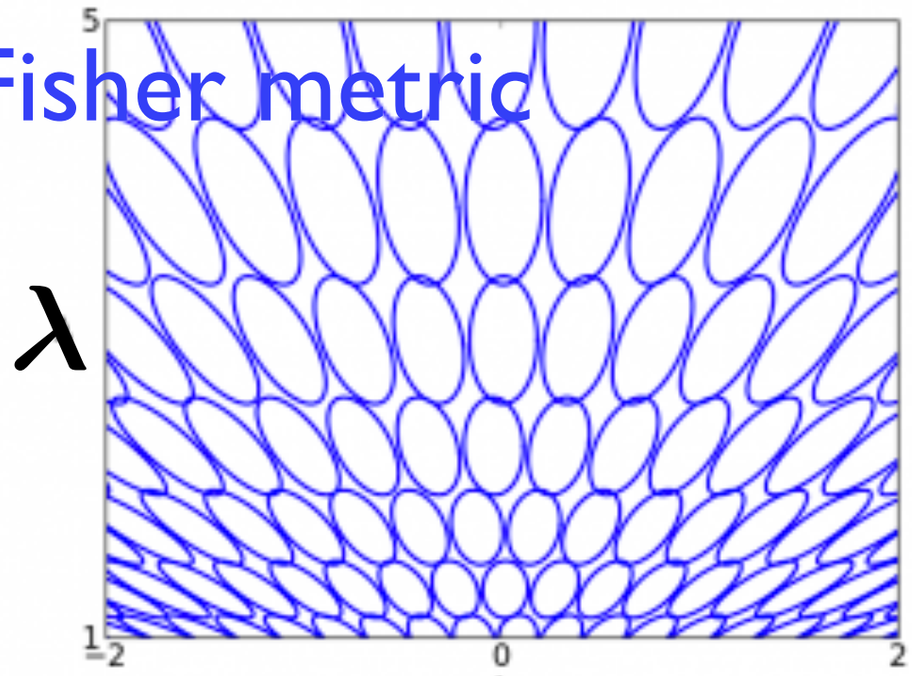


$$p(x) \propto \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

unit of

Fisher metric

information form



$$p(x) \propto \exp\left(hx - \frac{\lambda}{2}x^2\right)$$

Natural gradient (Amari, 1998)

gradient ascent

steepest ascent in Euclidean norm

depends on parameterization

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} \ell$$

Recall:

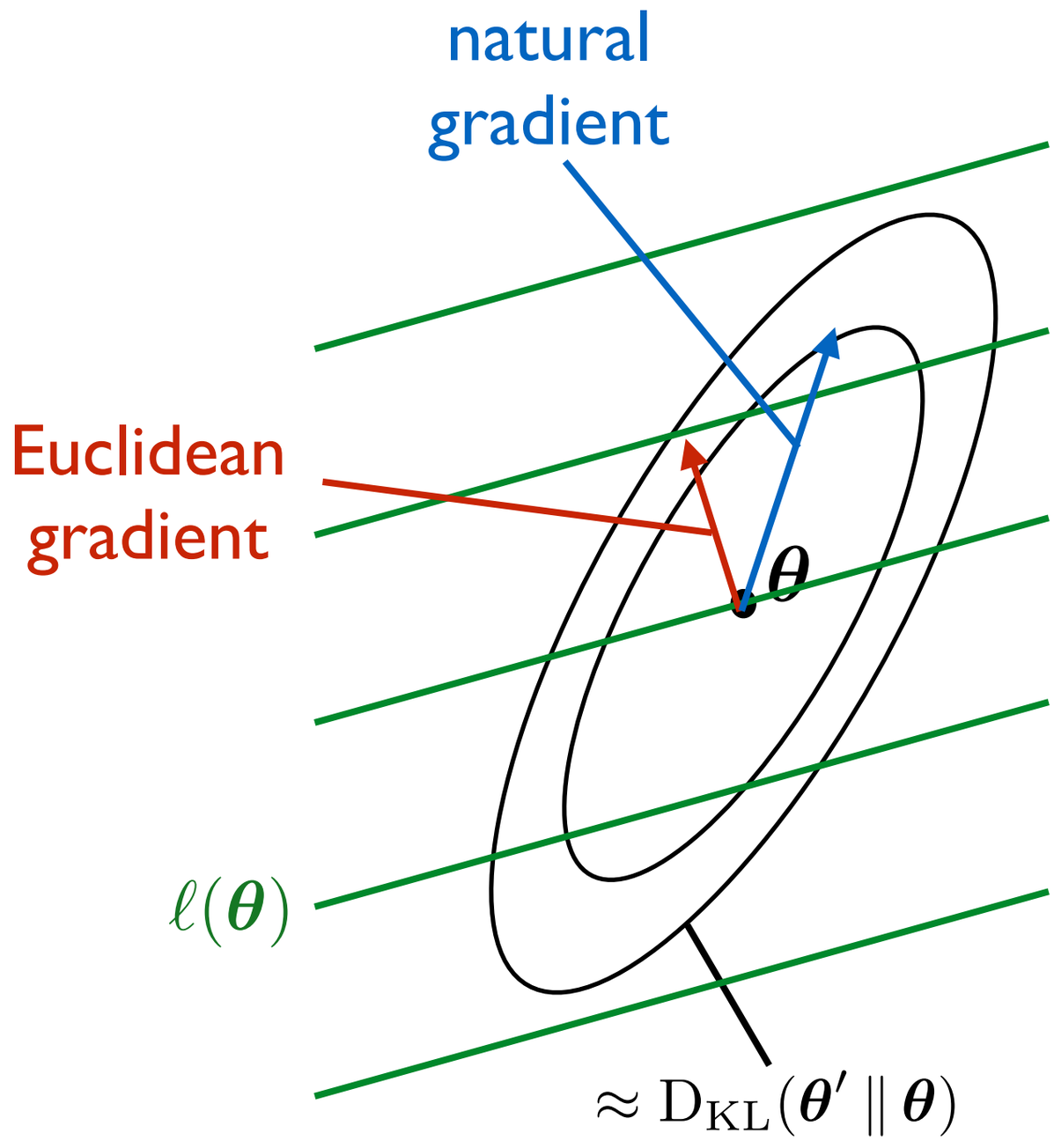
$$\mathrm{D}_{\mathrm{KL}}(\boldsymbol{\theta}' \,\|\, \boldsymbol{\theta}) \approx \frac{1}{2}(\boldsymbol{\theta}' - \boldsymbol{\theta})^T \boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{\theta}' - \boldsymbol{\theta})$$

natural gradient ascent

steepest ascent in Fisher norm

independent of parameterization

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \boldsymbol{G}^{-1} \nabla_{\boldsymbol{\theta}} \ell$$



Natural gradient: RBMs

gradient ascent: $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} + \alpha \left(\mathbf{s}_{\text{data}} - \mathbf{s}_{\text{model}} \right)$

natural gradient ascent: $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} + \alpha \boldsymbol{G}^{-1} \left(\mathbf{s}_{\text{data}} - \mathbf{s}_{\text{model}} \right)$

to first order: $\mathbf{s} \leftarrow \mathbf{s} + \alpha \left(\mathbf{s}_{\text{data}} - \mathbf{s}_{\text{model}} \right) \qquad (\mathrm{d}\mathbf{s} = \boldsymbol{G} \mathrm{d}\boldsymbol{\eta})$

Unfortunately…

G has $(N_v + N_h + N_v N_h)^2$ entries

= 155 billion for an MNIST RBM!

same problem as for other second order methods (Newton, etc.)

Natural gradient: RBMs

- Several approximations by analogy with Quasi-Newton
- Approximating G
  - diagonal (e.g. Adagrad; Duchi et al., 2011)
  - block diagonal (e.g. Le Roux et al., 2008)
- Iterative methods
  - Park et al. (2000)
  - metric-free natural gradient (Desjardins et al., 2013)

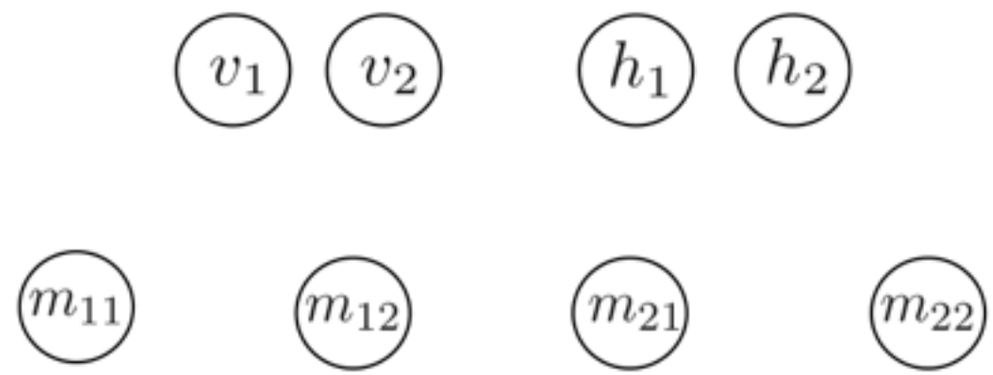
Factorized Natural Gradient (FaNG)

Fisher information is the covariance of the sufficient statistics:

$$\boldsymbol{G} = \mathrm{Cov}_{\mathbf{x}}(\mathbf{g}(\mathbf{x}))$$

We can approximate a covariance matrix with a Gaussian graphical model.

A diagonal approximation corresponds to a fully disconnected graph.



To come up with a better structure, let's analyze the Fisher information of an RBM.

Fisher information of RBMs

computing G for a small RBM (20 hidden units)

$$\mathbb{E}[\mathbf{g}] = \sum_{\mathbf{h}} p(\mathbf{h})\, \mathbb{E}[\mathbf{g} \,|\, \mathbf{h}]$$

$$\mathbb{E}[\mathbf{g}\mathbf{g}^T] = \sum_{\mathbf{h}} p(\mathbf{h}) \left(\mathbb{E}[\mathbf{g} \,|\, \mathbf{h}] \mathbb{E}[\mathbf{g} \,|\, \mathbf{h}]^T + \mathrm{Cov}[\mathbf{g} \,|\, \mathbf{h}] \right).$$

$$\boldsymbol{G} = \mathbb{E}[\mathbf{g}\mathbf{g}^T] - \mathbb{E}[\mathbf{g}] \mathbb{E}[\mathbf{g}]^T$$

Fisher information of RBMs

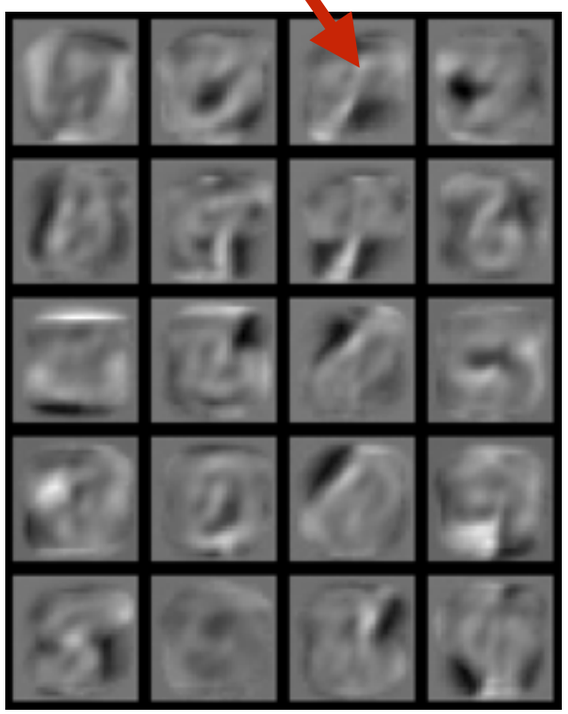
$$\mathrm{d}\mathbf{s} = \boldsymbol{G}\mathrm{d}\boldsymbol{\eta}$$

change in moments

Fisher information

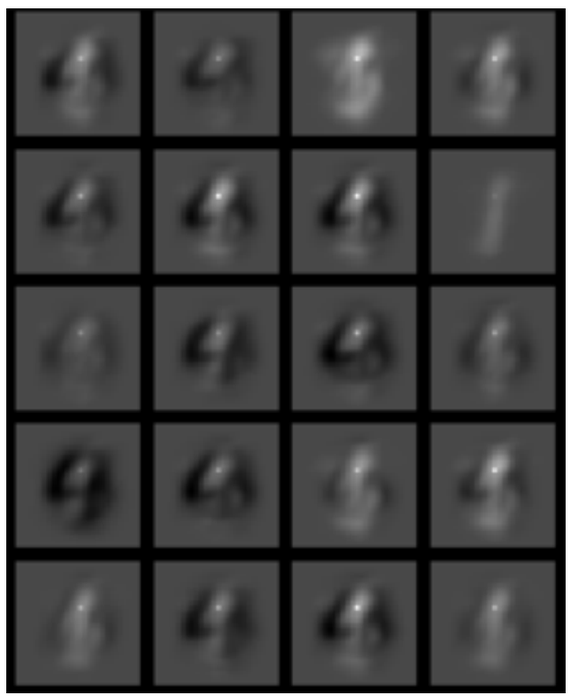
change in natural parameters

Changing this weight…



$\mathbf{W}$

… does this to the moments



$\mathrm{d}\mathbb{E}[\mathbf{v}\mathbf{h}^T]$

Fisher information of RBMs

$$\underbrace{\mathrm{d}\mathbf{s}}_{\text{change in moments}} = \underbrace{\boldsymbol{G}}_{\text{Fisher information}} \underbrace{\mathrm{d}\boldsymbol{\eta}}_{\text{change in natural parameters}}$$

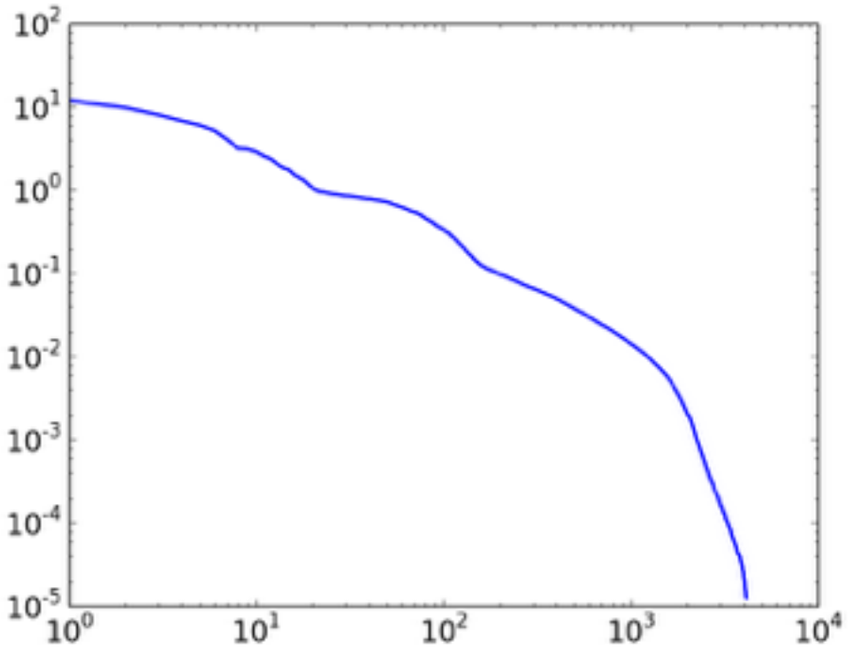
Most of the sufficient statistics are second moments, so let's think about how these vary with the parameters.

$$\mathrm{d}\mathbb{E}[\mathbf{v}\mathbf{h}^T] = \underbrace{\mathrm{d}\mathbb{E}[\mathbf{v}]\mathbb{E}[\mathbf{h}]^T + \mathbb{E}[\mathbf{v}]\mathrm{d}\mathbb{E}[\mathbf{h}]^T}_{\text{change in means (boring)}} + \underbrace{\mathrm{d}\,\mathrm{Cov}(\mathbf{v}, \mathbf{h})}_{\text{change in covariances (interesting)}}.$$

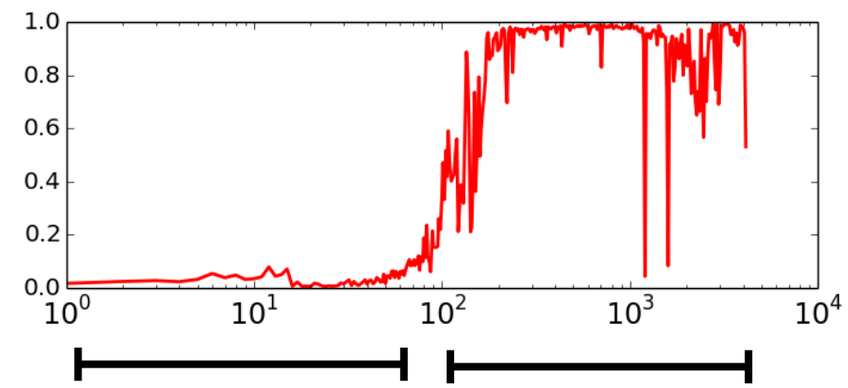
change in second moments = change in means (boring) + change in covariances (interesting)

Fisher information of RBMs

eigenspectrum of G

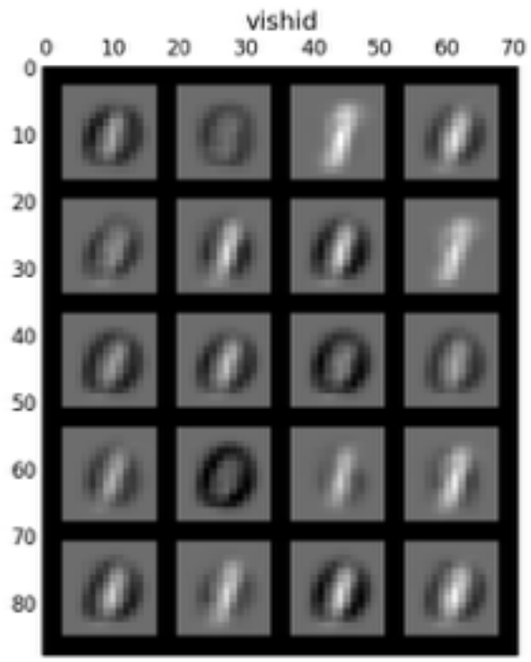


$$\frac{\|\mathrm{d}\,\mathrm{Cov}(\mathbf{v},\mathbf{h})\|^2}{\|\mathrm{d}\mathbf{s}\|^2}$$



mostly 1st order stats

mostly covariances

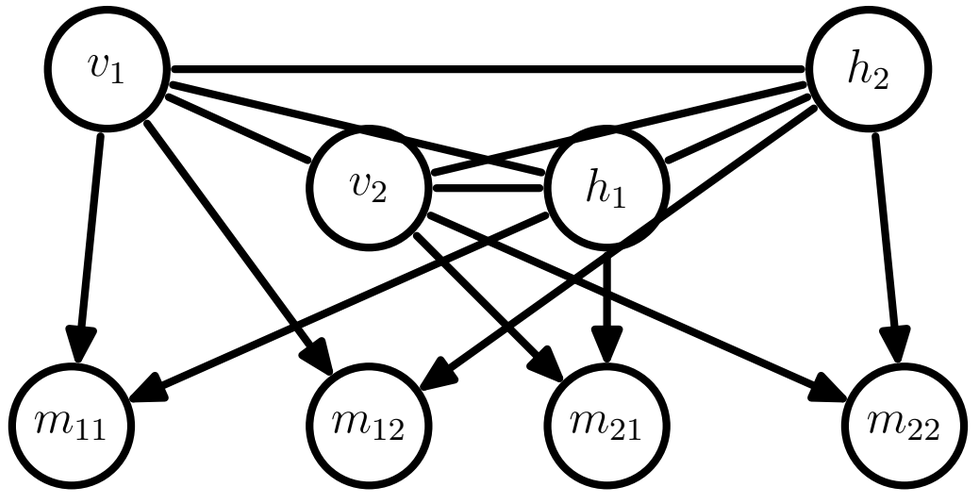


top eigenvector
(direction of highest curvature)

Factorized Natural Gradient (FaNG)

We just saw that the first-order statistics are driving everything.

We propose a graphical model structure which models dependencies between sufficient statistics.



$N_v + N_h$ unary

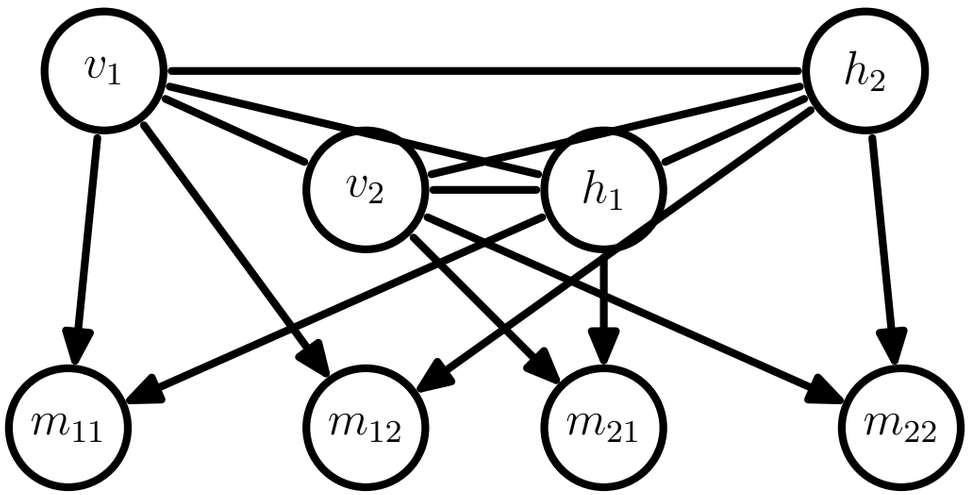
$N_v N_h$ pairwise

If the variables were binary, this would represent the joint distribution exactly.

The approximation is that we're modeling them as jointly Gaussian.

Factorized Natural Gradient (FaNG)

Number of parameters



$$\left[\ \frac{1}{2}(N_v + N_h)^2\right.$$

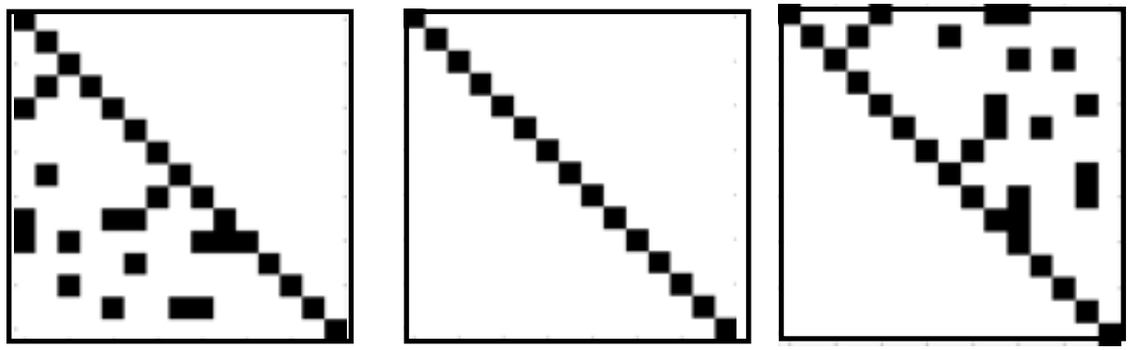
$$\left[\ 3N_vN_h\right.$$

Compared with $(N_v + N_h + N_vN_h)^2$ for full G

Factorized Natural Gradient (FaNG)

Directed model corresponds to sparse Cholesky factorization

$$\boldsymbol{G}^{-1} \approx \mathbf{L} \quad \mathbf{D} \quad \mathbf{L}^T$$

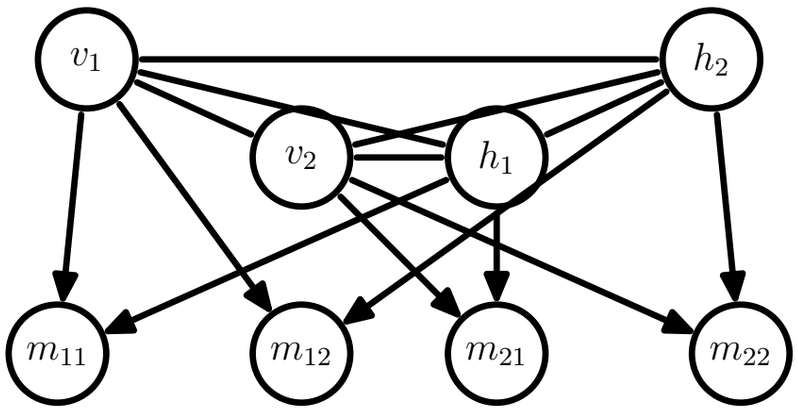


Natural gradient updates are sparse matrix-vector products

Fit maximum likelihood parameters using linear regression

Factorized Natural Gradient (FaNG)

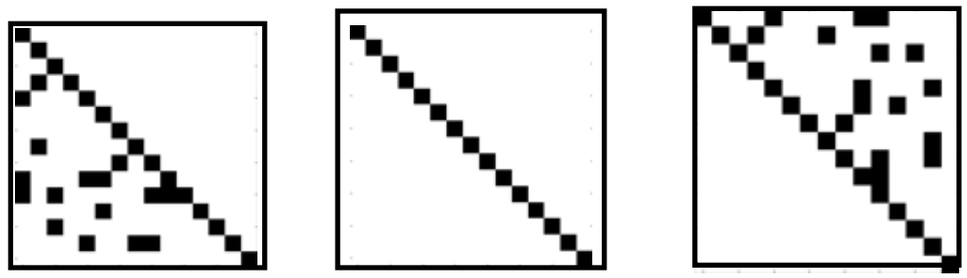
Computational cost of fitting parameters (every ~100 iterations)



$\mathcal{O}((N_v + N_h)^3)$

$\mathcal{O}(N_v N_h)$

Computational cost of updates



$N_v N_h$

Explaining the "centering trick"

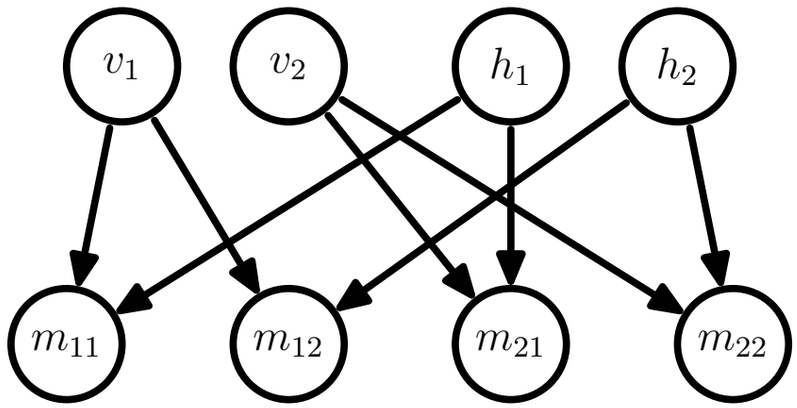
- Enhanced gradient (Cho et al., 2011); centering trick (Montavon and Muller, 2012)
- Reparameterize the RBM:

$$\tilde{\mathbf{v}} = \mathbf{v} - \mathbb{E}[\mathbf{v}]$$
$$\tilde{\mathbf{h}} = \mathbf{h} - \mathbb{E}[\mathbf{h}]$$

- Invariant to flipping on/off
- Train deep Boltzmann machines without pre-training (Montavon and Muller, 2012)
- Need this even in addition to metric-free natural gradient (Desjardins et al., 2013)
- But why should this make such a difference?

Explaining the "centering trick"

- Actually an approximation to natural gradient!

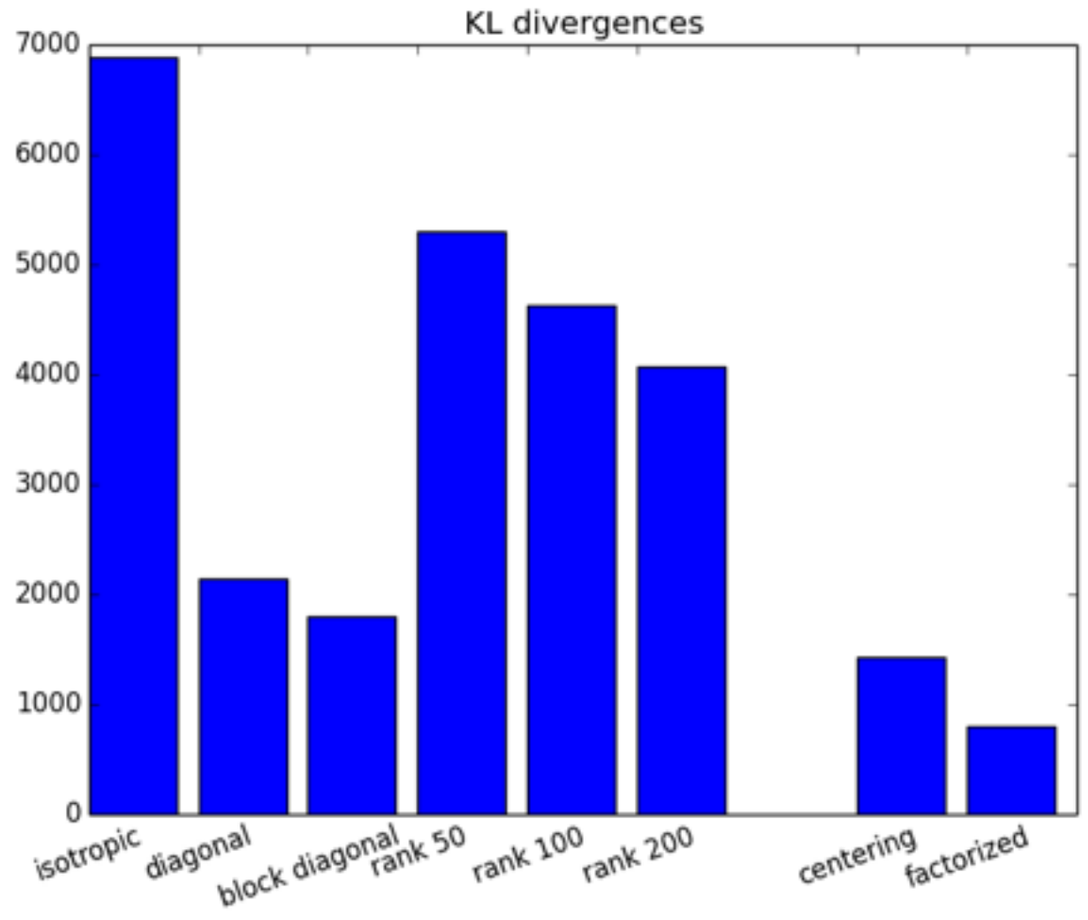


$$m_{ij} \mid v_i, h_j \sim \mathcal{N}\left(\mathbb{E}[m_{ij}] + \mathbb{E}[v_j](h_j - \mathbb{E}[h_j]) + \mathbb{E}[h_j](v_i - \mathbb{E}[v_i]),\ \sigma^2\mathbf{I}\right)$$

- Not the optimal graphical model parameters, but a fairly good guess

Explaining the "centering trick"

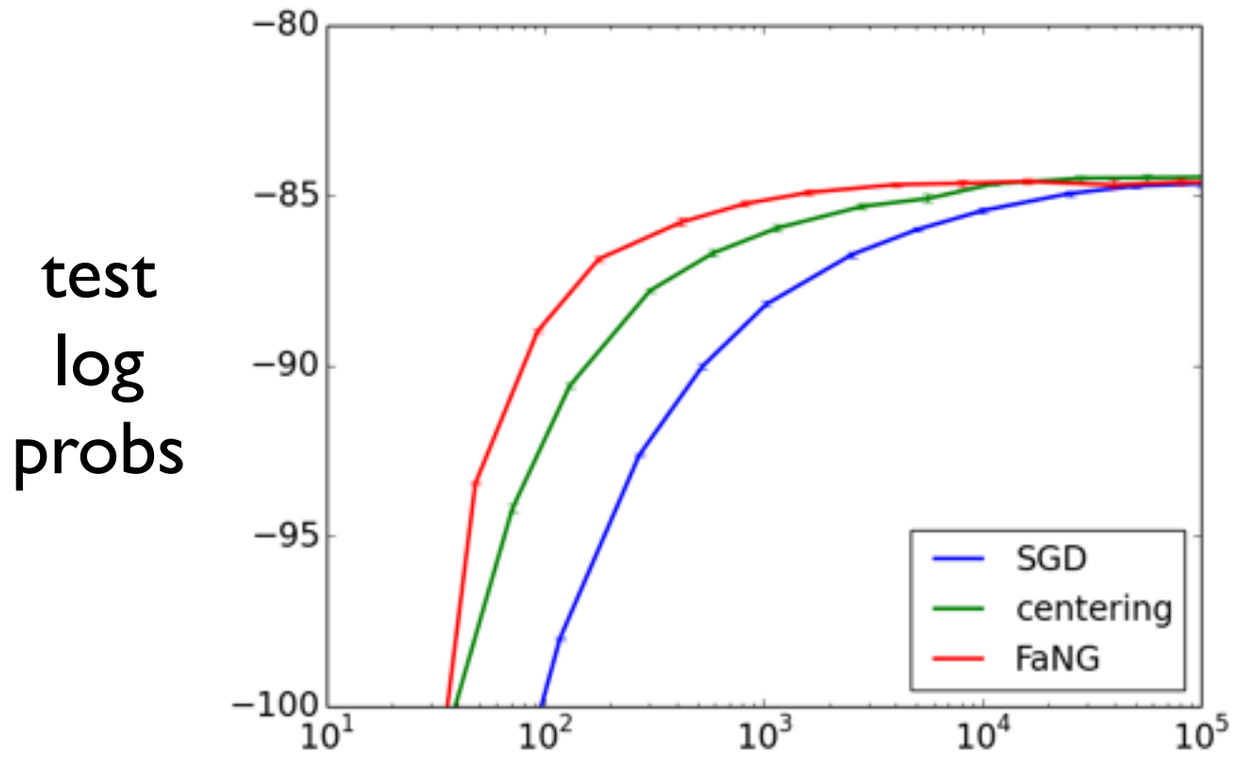
KL divergence to Gaussian with covariance G



RBM training experiments

MNIST

(784 x 500)



Aside: estimating RBM log-likelihoods

- Markov random fields (e.g. RBMs)

$$p(\mathbf{x}) = \frac{f(\mathbf{x})}{\mathcal{Z}}$$

$$\mathcal{Z} = \sum_{\mathbf{x}} f(\mathbf{x})$$

- **Evaluating the likelihood requires estimating the intractable $\mathcal{Z}$**

Aside: estimating RBM log-likelihoods

- Annealed importance sampling (AIS) gives an unbiased estimate of

$$\mathbb{E}[\hat{\mathcal{Z}}_b] = \mathcal{Z}_b$$

- But it gives a biased estimate of

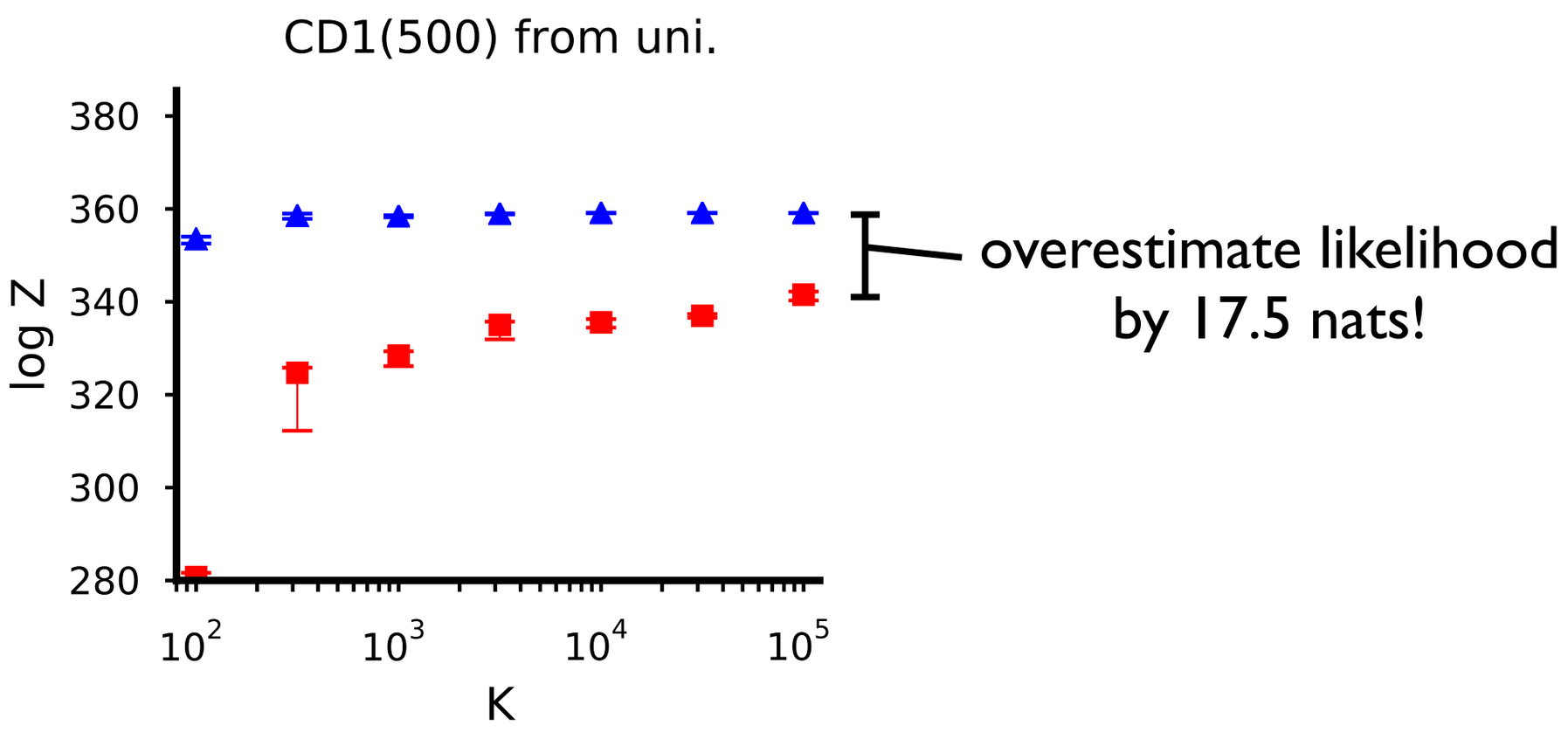
$$\mathbb{E}[\log \mathcal{Z}_b] \leq \log \mathcal{Z}_b$$

- Do you have a good model or a bad partition function estimator?

$$\text{overestimate} \rightarrow p(\mathbf{x}) = f(\mathbf{x})/\mathcal{Z} \leftarrow \text{underestimate}$$

Aside: estimating RBM log-likelihoods

- Is this a problem in practice?



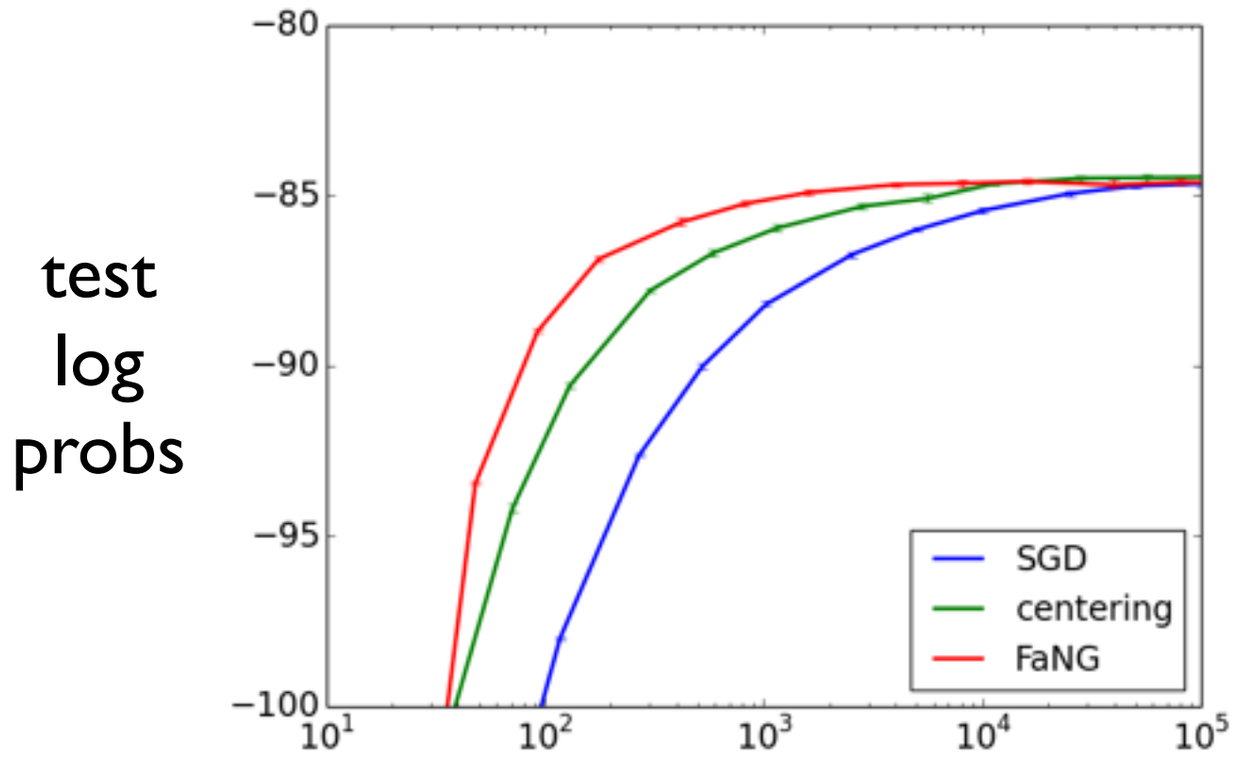
Aside: estimating RBM log-likelihoods

- Used AIS with geometric averages
  - careful choice of initial distribution: set weights to 0, biases to log odds of data conditional distribution
- Computed confidence intervals using the bootstrap (rather than Gaussian approximation, as is typical)
- Checked that final results were consistent with other estimation methods
  - AIS with moment averaging (Grosse et al., 2013)
  - RAISE (Burda et al., 2014)

RBM training experiments

MNIST

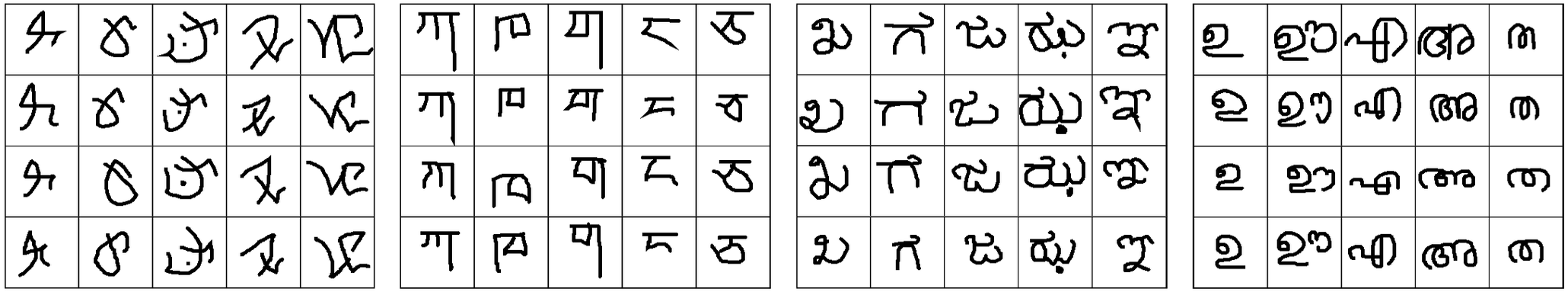
(784 x 500)



RBM training experiments

Omniglot

(784 x 500)

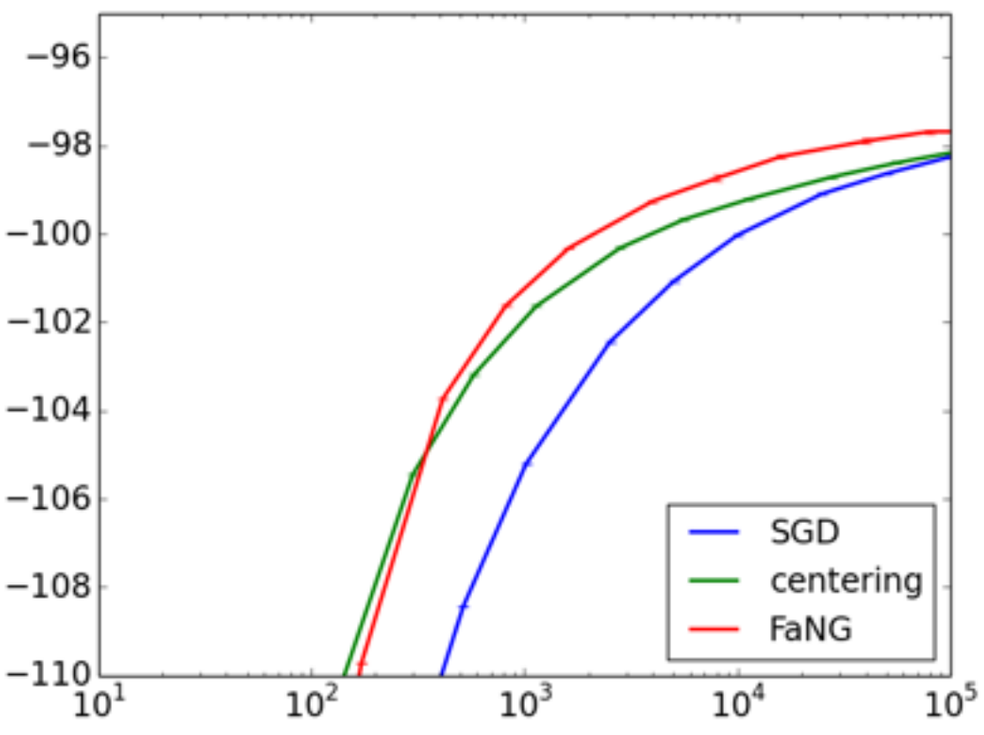


RBM training experiments

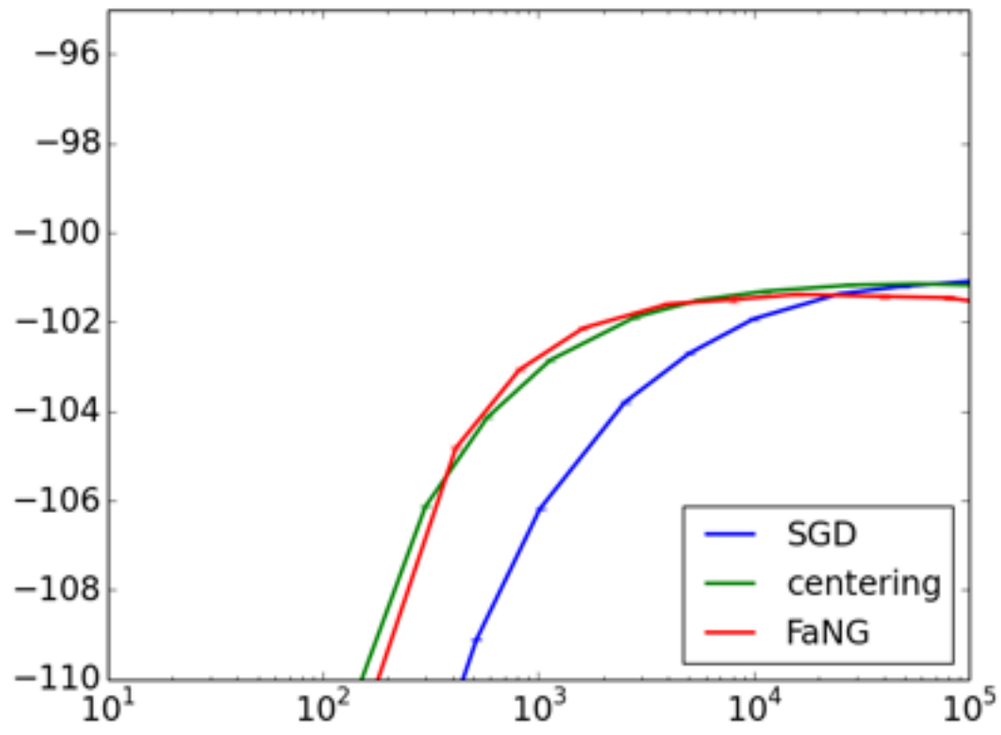
Omniglot

(784 x 500)

training



test



Conclusions

- RBM training suffers from nasty curvature
- important covariance information hidden in smaller eigenvalues
- Factorized Natural Gradient: non-iterative approximation
  - helps explain the impressive performance of the centering trick
  - more generally, looking at the structure of the model can help us improve optimization