Scaling up natural gradient by sparsely factorizing the inverse Fisher matrix

Roger Grosse

Ruslan Salakhutdinov
Introduction

• Most neural nets are trained with stochastic gradient descent
• Would like to use second-order optimization so we can correct for curvature
Introduction

• Flat representations of complicated manifolds distort shape, distance, etc.

• This is the main reason neural net training suffers from badly conditioned curvature
Introduction

• Natural gradient is a second-order method which does steepest descent on an intrinsic distance metric

• Applying it exactly is impractical in high dimensions since it requires inverting a large covariance matrix

• This work is about using a probabilistic graphical model to compactly and tractably approximate the covariance matrix
Introduction

• For simplicity, we focus on Boltzmann machines, which have successfully modeled many complex distributions

(Murray and Salakhutdinov, 2008)  (Salakhutdinov, 2009)  (Lake et al., 2011)
Introduction

• Training RBMs is still a dark art for several reasons
  • Can’t evaluate the likelihood
  • Can’t compute the gradient
  • Badly conditioned curvature

• I will focus on the third issue
Background: Fisher matrix

\[ G_\theta \triangleq \nabla^2_{\theta'} D_{KL}(\theta' \parallel \theta) \bigg|_{\theta' = \theta} \]

\[ D_{KL}(\theta' \parallel \theta) \approx \frac{1}{2} (\theta' - \theta)^T G_\theta (\theta' - \theta) \]
Background: natural gradient

gradient ascent
stepest ascent in Euclidean norm
depends on parameterization
\[ \theta \leftarrow \theta + \alpha \nabla_{\theta} \ell \]

Recall:
\[ D_{KL}(\theta' \| \theta) \approx \frac{1}{2} (\theta' - \theta)^T G_{\theta} (\theta' - \theta) \]

natural gradient ascent
stepest ascent in Fisher norm
independent of parameterization
\[ \theta \leftarrow \theta + \alpha G^{-1} \nabla_{\theta} \ell \]

\[ \approx D_{KL}(\theta' \| \theta) \]
Background: RBMs as exponential families

$$p(v, h) \propto \exp(a^T v + b^T h + v^T Wh)$$

sufficient statistics:

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

Two parameterizations:

$$\eta = \begin{pmatrix} a \\ b \\ \text{vec}(W) \end{pmatrix}$$

inference

natural parameters

s = $E[g(v, h)] = \begin{pmatrix} E[v] \\ E[h] \\ \text{vec}(E[vh^T]) \end{pmatrix}$

learning (fully observed)

moments
Natural gradient: RBMs

gradient ascent: \[ \eta \leftarrow \eta + \alpha (s_{\text{data}} - s_{\text{model}}) \]

natural gradient ascent: \[ \eta \leftarrow \eta + \alpha G^{-1}(s_{\text{data}} - s_{\text{model}}) \]

Unfortunately…

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

= 155 billion for an academic-scale 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)
  • Hessian-free optimization (Martens, 2010)
  • metric-free natural gradient (Desjardins et al., 2013)
The Fisher matrix is the covariance of the sufficient statistics:

\[ G = \text{Cov}_x(g(x)) \]

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)

\[
E[g] = \sum_h p(h) E[g \mid h]
\]

\[
E[gg^T] = \sum_h p(h) \left( E[g \mid h]E[g \mid h]^T + \text{Cov}[g \mid h] \right).
\]

\[
G = E[gg^T] - E[g]E[g]^T
\]
Fisher information of RBMs

\[ ds = G d\eta \]

change in moments

Fisher information

change in natural parameters

Changing this weight...

... does this to the moments

\[ W \]

\[ dE[vh^T] \]
Fisher information of RBMs

\[
ds = Gd\eta
\]

Eigenvectors: \( ds = \lambda d\eta \)

Decomposing the change in moments:

\[
\]

The top 100 eigenvalues of \( G \) contain primarily first order information!
Factorized Natural Gradient (FaNG)

We want to model how the first-order statistics influence the second-order statistics, so that we can correct for this.

Consider this graphical model:

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

\[
\frac{1}{2} (N_v + N_h)^2
\]

\[
3N_v N_h
\]

Compared with

\[
(N_v + N_h + N_v N_h)^2 \quad \text{for full } G
\]
Factorized Natural Gradient (FaNG)

Directed model corresponds to sparse Cholesky factorization

$$G^{-1}_{\text{fac}} = L \quad D \quad L^T$$

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)

\[ O((N_v + N_h)^3) \]

\[ O(N_v N_h) \]

Computational cost of updates

\[ N_v N_h \]
RBM training experiments

MNIST
(784 x 500)

Graph showing test log probs over time (seconds) for different optimization methods.
RBM training experiments

Omniglot
(784 x 500)

| ă ă ă ă ă ă | ă ă ă ă ă ă | ă ă ă ă ă ă | ă ă ă ă ă ă |
| ă ă ă ă ă ă | ă ă ă ă ă ă | ă ă ă ă ă ă | ă ă ă ă ă ă |
| ă ă ă ă ă ă | ă ă ă ă ă ă | ă ă ă ă ă ă | ă ă ă ă ă ă |
| ă ă ă ă ă ă | ă ă ă ă ă ă | ă ă ă ă ă ă | ă ă ă ă ă ă |

In addition to introducing new one-shot learning challenge problems, this paper also introduces Hierarchical Bayesian Program Learning (HBPL), a model that exploits the principles of compositionality and causality to learn a wide range of simple visual concepts from just a single example. We compared the model with people and other competitive computational models for character recognition, including Deep Boltzmann Machines [25] and their Hierarchical Deep extension for learning with very few examples [26]. We find that HBPL classifies new examples with near human-level accuracy, substantially beating the competing models. We also tested the model on generating new exemplars, another natural form of generalization, using a "visual Turing test" to evaluate performance. In this test, both people and the model performed the same task side by side, and then other human participants judged which result was from a person and which was from a machine.
RBM training experiments

Omniglot
(784 x 500)

training

test

log probs

time (seconds)

log probs

time (seconds)
Explaining the “centering trick”

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

- Reparameterize the RBM:
  \[
  \tilde{v} = v - \mathbb{E}[v] \\
  \tilde{h} = h - \mathbb{E}[h]
  \]

- Invariant to flipping on/off

- Can view as an approximation to FANG
  - remove the clique over unaries, solve for CPDs only approximately
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
Thank you