Generative Modeling for Machine Learning on the D-Wave

Sunil Thulasidasan
Information Sciences (CCS-3)
Los Alamos National Laboratory
sunil@lanl.gov

LA-UR-16-28813

Generative Models

- Two approaches in machine learning:
  - Discriminative: Learn $P(y|x)$
  - Generative: Learn $P(y,x)$

- Discriminative models are easier to train, but generative models are more powerful because in some sense it "understands" the world better.

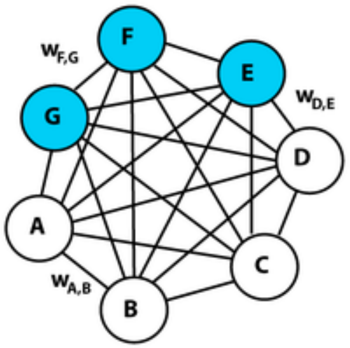
Boltzmann Machines: A Generative Model

- Energy based model. Assign a scalar energy value to configurations of interest
- Associate lower energy with plausible configurations
- Probability given by

$$P(x) = \frac{e^{(-E(x))}}{\mathbf{Z}}$$

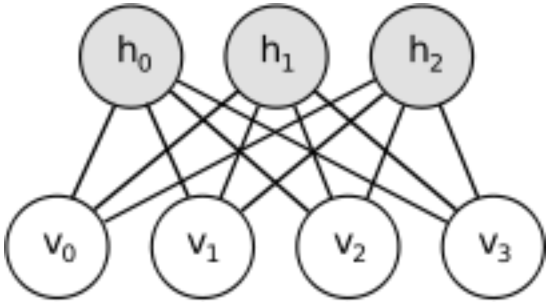
- Consists of **visible units** (data) and **hidden units** (capture dependencies between data)

General Boltzmann machines have arbitrary connectivity. Hard to train.



Restricted Boltzmann Machines

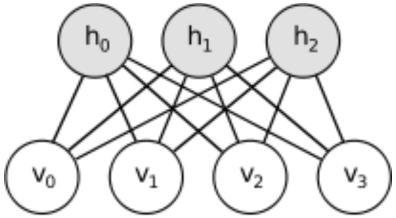
- Restrict connections to occur only between pairs of visible and hidden units. No connections among visible units or hidden units.



- *h’s* are independent given *v* and *v’s* are independent given h (markov property)

Restricted Boltzmann Machines

- Energy given by



$$E(v,h) = -b'v - c'h - h'Wv$$

- Conditional independence implies:

$$p(h|v) = \prod_i p(h_i|v)$$

$$p(v|h) = \prod_j p(v_j|h)$$

- Once we know the parameters (*b*,*c*,*W*) generating data is easy

Learning Parameters: RBM Training

- Learn parameters that maximize log-likelihood of data. Assuming data independence, we have

$$\arg\max_{(w,b,c)} \ell(w,b,c) = \sum_{t=1}^{n} \log P(v^t)$$

- The gradient is given by

$$\nabla_\theta \ell(\theta) = \sum_{t=1}^{n} \mathbb{E}_{p(h|v)} \left[\nabla_\theta(-E(v^t,h)\right] - n\mathbb{E}_{p(v,h)}(\nabla_\theta(-E(v^t,h))$$

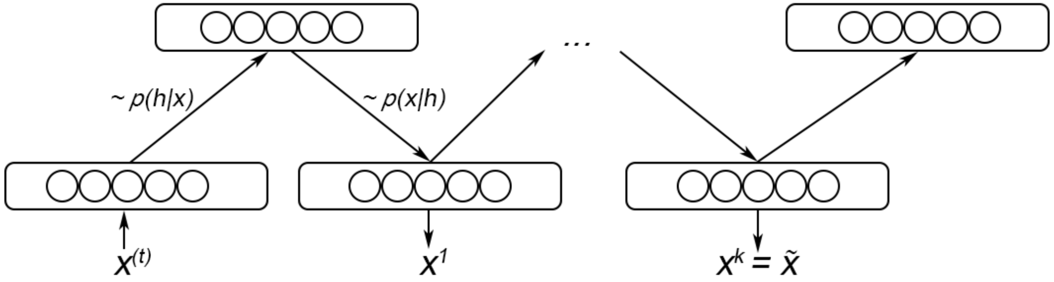
RBM Training

$$\nabla_\theta \ell(\theta) = \sum_{t=1}^{n} \mathbb{E}_{p(h|v)} \left[\nabla_\theta (-E(v^t, h) \right] - n\mathbb{E}_{p(v,h)} (\nabla_\theta (-E(v^t, h))$$

- Gradient depends on joint distribution
- Intractable since it involves the partition function **Z**
- To avoid this, use Gibb's sampling to sample from joint (Boltzmann distribution). Involves running a Markov chain to convergence (Markov Chain Monte Carlo or MCMC)

Practical Ways to Train RBM

- Instead of running MCMC to convergence, run it for just a few (*k*) steps. Sample from this distribution (Contrastive Divergence)
- In practice, *k* (number of steps) is < 100. Some times even 1 step works well !



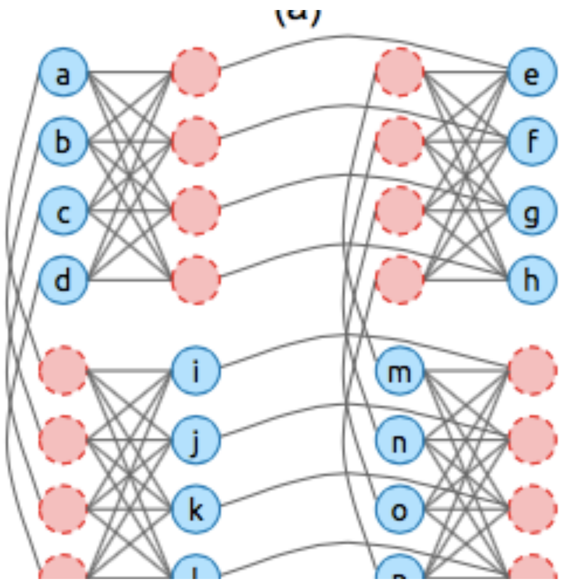
D-Wave as a Boltzmann Sampler

- D-Wave is a physical Boltzmann machine
- In theory, should give samples from a Boltzmann distribution (parameterized by some *effective* temperature) after annealing
- Approach: Instead of Gibbs's sampling, map RBM onto D-Wave and sample from solution states

Mapping RBM onto the D-Wave

- RBM's are full bipartite graphs. D-Wave has sparse connectivity.
- Using logical qubits, can implement up to 48x48 bipartite graph. Lots of qubits lost
- For this work, no qubit chaining. Map each pixel of the training image directly onto a qubit

Chimera Restricted RBM



Pixel blocks embedding

a	b	e	f
c	d	g	h
i	j	m	n
k	l	o	p

Same embedding as in Benedetti et al (2015) and Doulin et al (2014)

Mapping binary RBM to Ising Model

- RBM's are binary {0,1} units.
- To map this to Ising model, where units are in {+1,-1} we use the following transformation described in Domoulin (2014)

$$W' = \frac{W}{4}$$

$$b'_i = \frac{1}{2} b_i + \frac{1}{4} \sum_j W_{ij}$$

$$c'_i = \frac{1}{2} c_i + \frac{1}{4} \sum_j W_{ji}$$

LA-UR-16-28813

Experiments

- Basic Outline (classical side):
  - Initialize visible units and hidden units
  - Clamp visible units to a training sample
  - Run few steps of contrastive divergence for gradient
  - Update parameters
  - Run till convergence

- On the D-Wave, same process except we do not run contrastive divergence, but sample from solution states

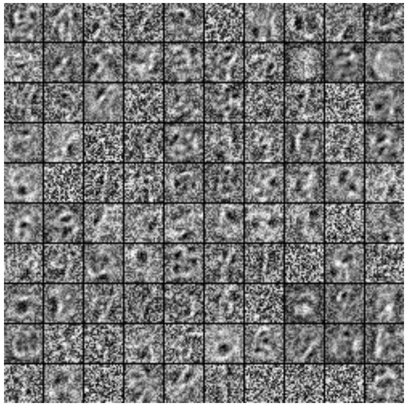
Data

- MNIST (handwritten digits 0-9)
- Train on 1000 digits and learn features.
- And then see if the model can generate its own representations.

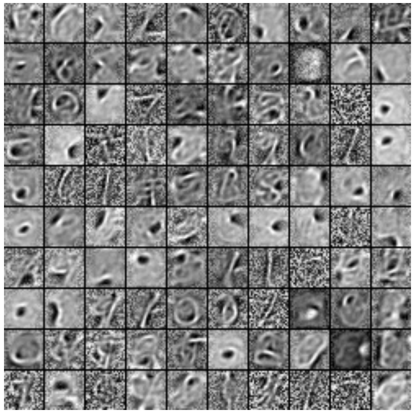
D-Wave Effective Temperature, Parameter Noise etc

- D-Wave *effective temperature* is different from *physical temperature*. Estimate this via sampling and then find a best fit
- Did not do any corrections for weight and bias noise.
- Effective temperature also fluctuates during training (Benedetti et al 2015). Did not correct for this.

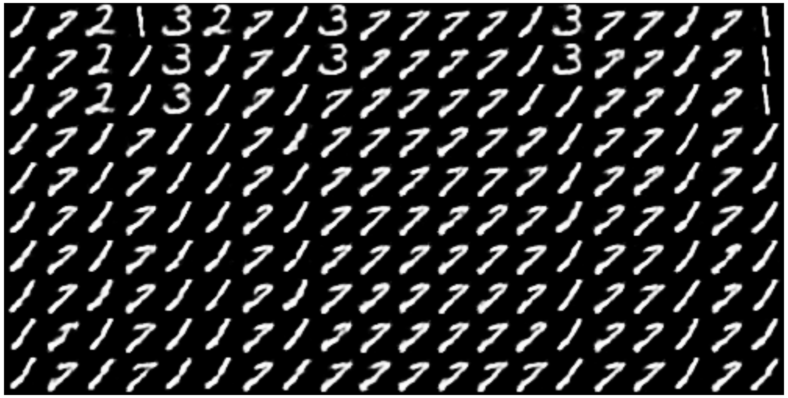
Experiments:
Contrastive Divergence (CD) 1 Step



Filters learned after epoch 1

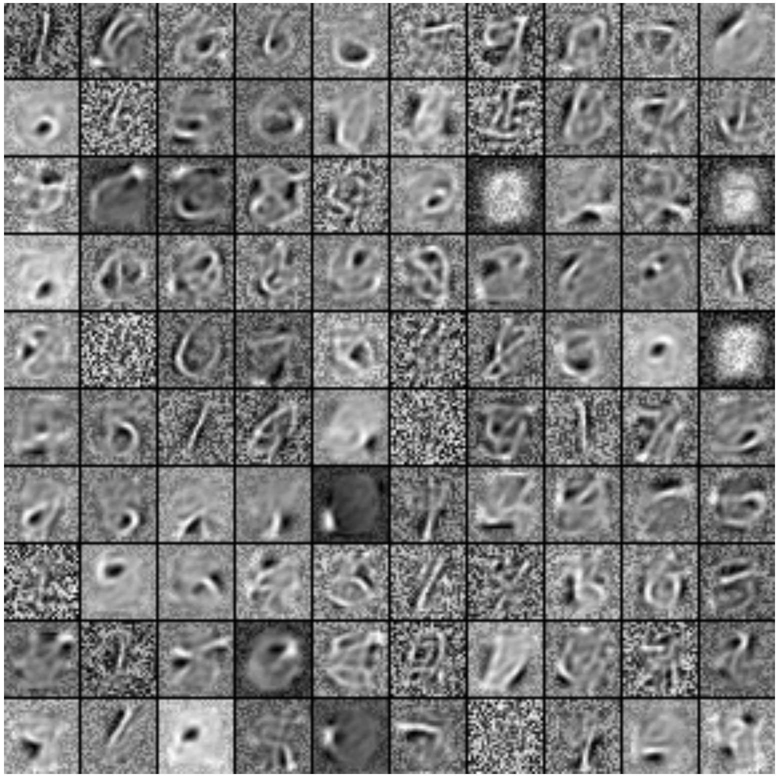


Filters learned after epoch 15

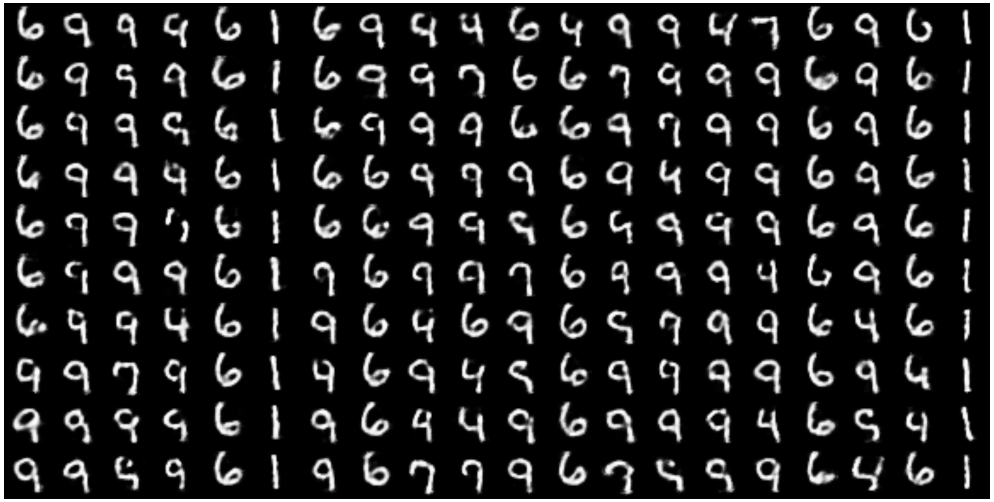


Generated Images

After 50 Steps of CD



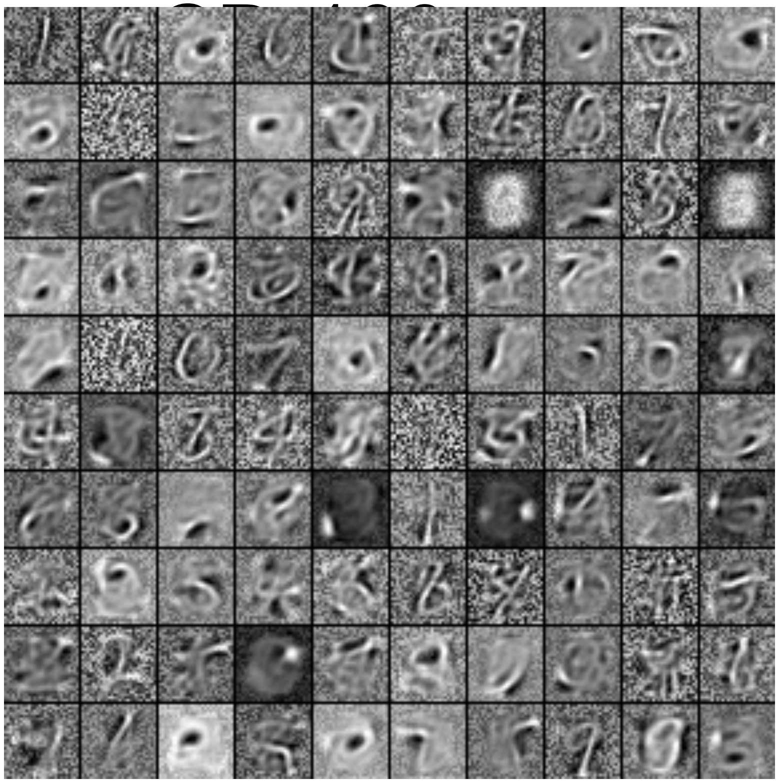
Filters



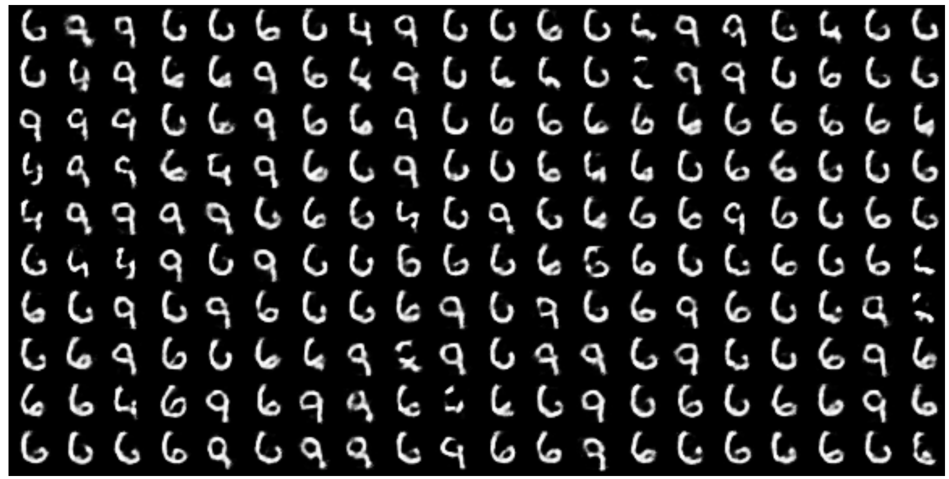
Generated Images

LA-UR-16-28813

After 100 Steps of CD

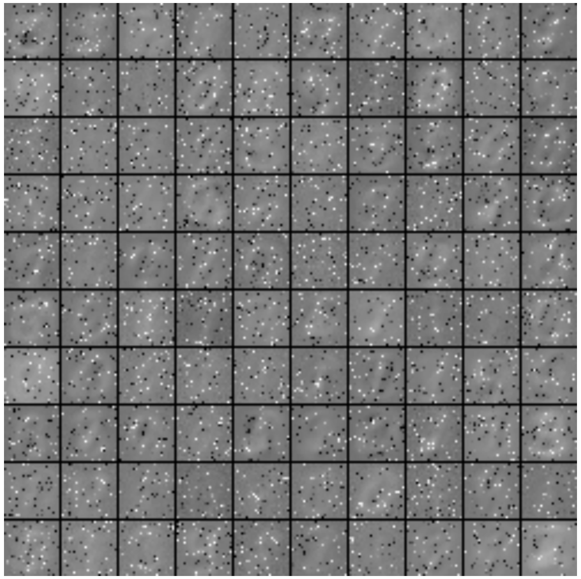


Filters

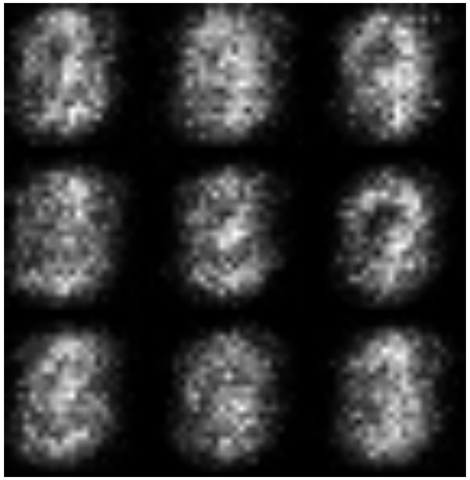


Generated Images

D-Wave (Experiment 1)

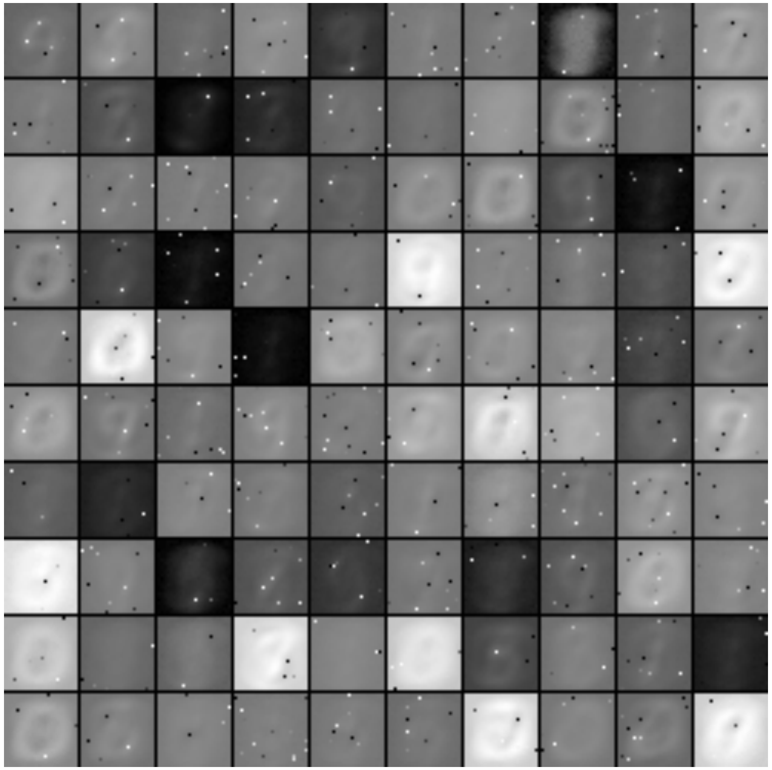


Filters learnt are sparse due to sparse connectivity graph

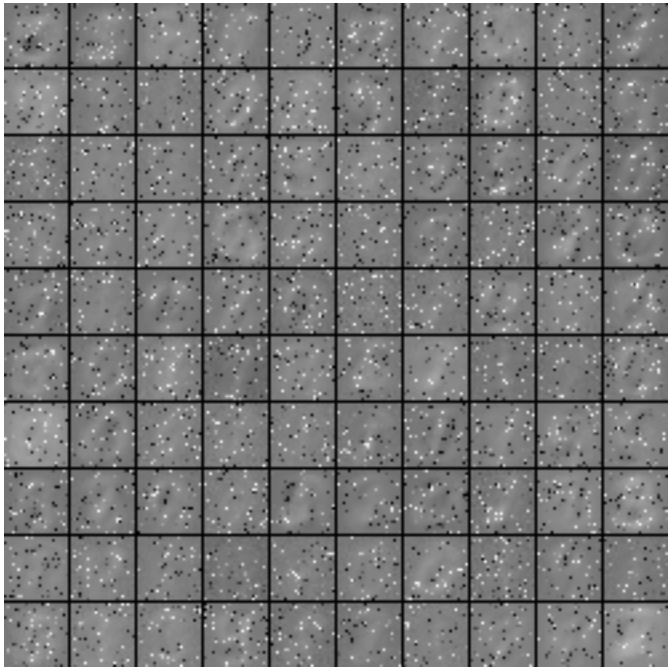


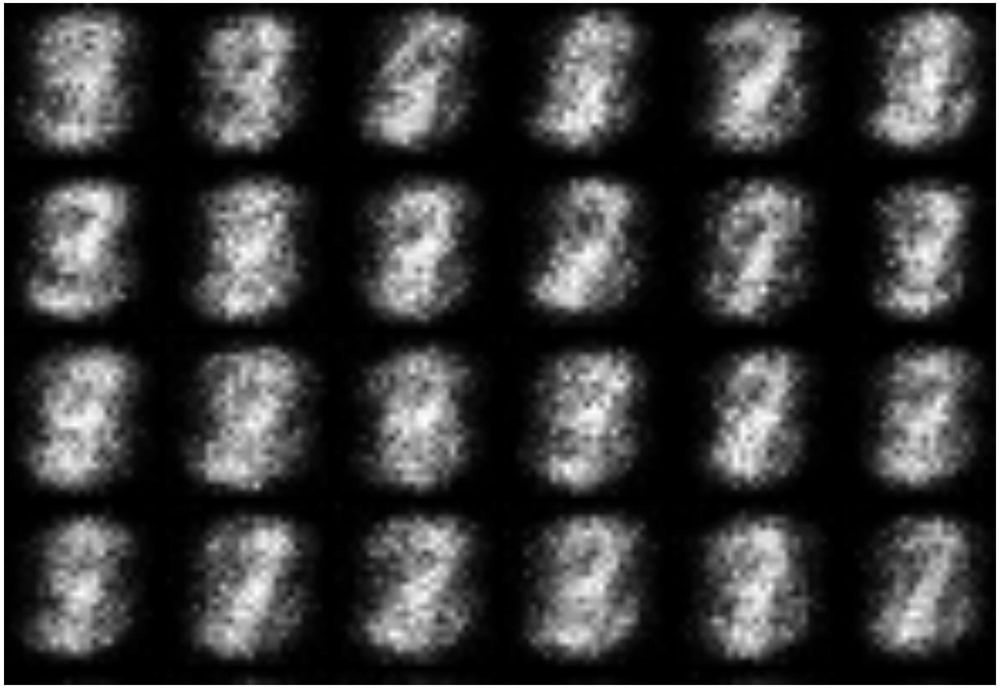
Generated images are noisy and largely indistinguishable from one another

D-Wave (Experiment 2)



D-Wave (Experiment 3)





D-Wave Observations

- Effective temperature and parameter noise affect modeling
- However, limited connectivity is a *much* bigger problem
  - RBM's are robust to limited connections. But the D-Wave has less than 1% of connections of a complete bipartite graph.
  - Qubit chaining can overcomes connectivity issues, but then image has to be significantly down-sampled.

References

- **Dumoulin et al 2014**. On the Challenges of Physical Implementations of RBMs. *In Proceedings of AAAI 2014*
- **S. Adachi, M. Henderson, 2015**. Application of Quantum Annealing to Training of Deep Neural Networks
- **Benedetti et al. 2015**. Estimation of Effective Temperatures in Quantum Annealers for Sampling Applications: A Case Study with Possible Applications in Deep Learning