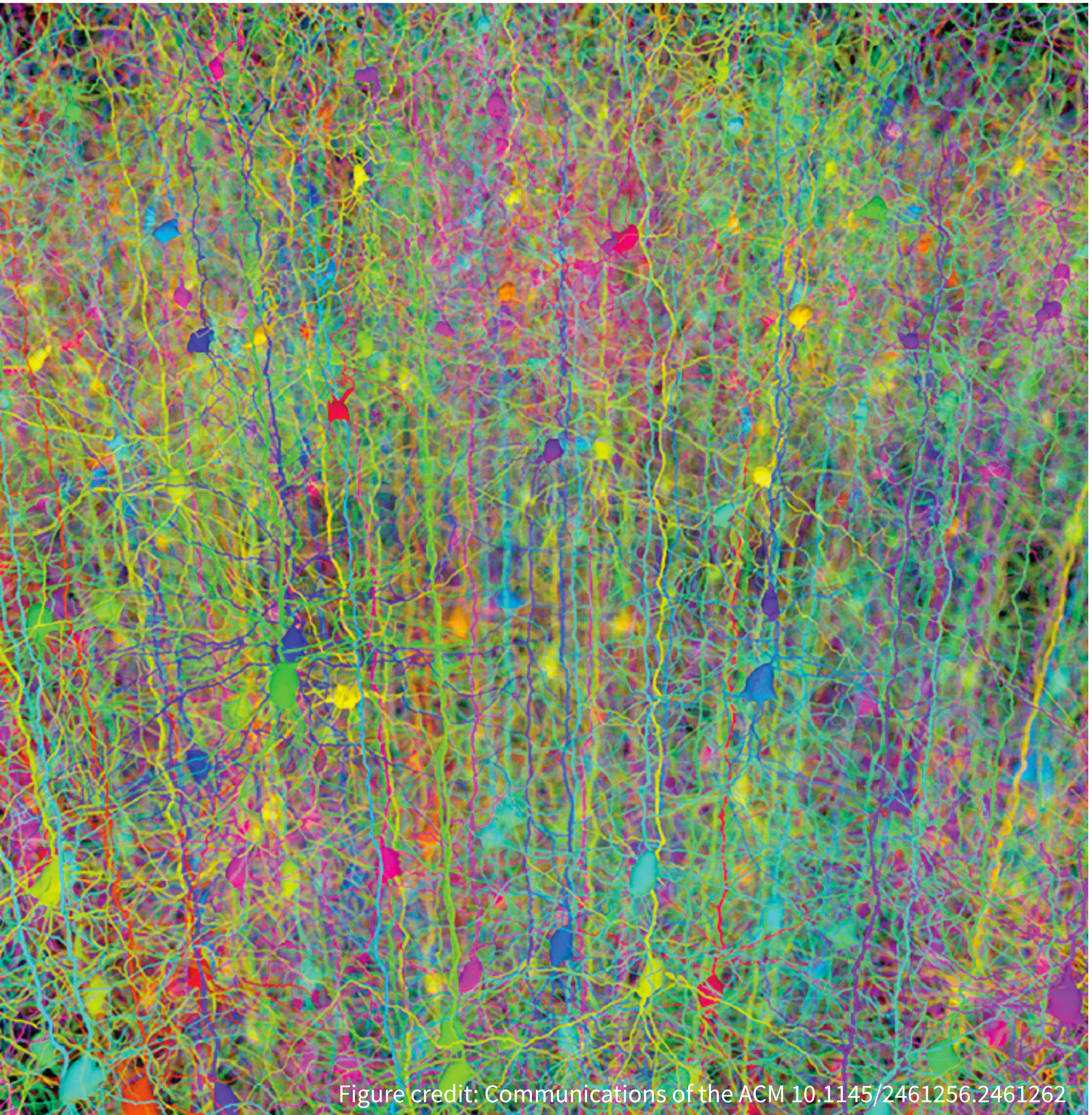
Training Restricted Boltzmann Machines



GRAHAM TAYLOR

VECTOR INSTITUTE

SCHOOL OF ENGINEERING
UNIVERSITY OF GUELPH

CANADIAN INSTITUTE
FOR ADVANCED RESEARCH

UNIVERSITY of GUELPH
CHANGING LIVES
IMPROVING LIFE

CIFAR
CANADIAN INSTITUTE FOR ADVANCED RESEARCH

Training RBMs

- To train an RBM, we would like to minimize the average negative log likelihood:

$$\frac{1}{n}\sum_{i=1}^{n} -\log P(\boldsymbol{v}^{(i)})$$

- We'd then like to proceed by stochastic gradient descent:

$$\frac{\partial -\log P(\boldsymbol{v}^{(i)})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\boldsymbol{h}}\left[\frac{\partial E(\boldsymbol{v}^{(i)},\boldsymbol{h})}{\partial \boldsymbol{\theta}}\,\middle|\,\boldsymbol{v}^{(i)}\right]}_{\text{positive phase}} - \underbrace{\mathbb{E}_{\boldsymbol{v},\boldsymbol{h}}\left[\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}}\right]}_{\text{negative phase}}$$

Reproduced from Hugo Larochelle's slides

Training RBMs

- To train an RBM, we would like to minimize the average negative log likelihood:

$$\frac{1}{n}\sum_{i=1}^{n} -\log P(\boldsymbol{v}^{(i)})$$

- We'd then like to proceed by stochastic gradient descent:

hard to compute!

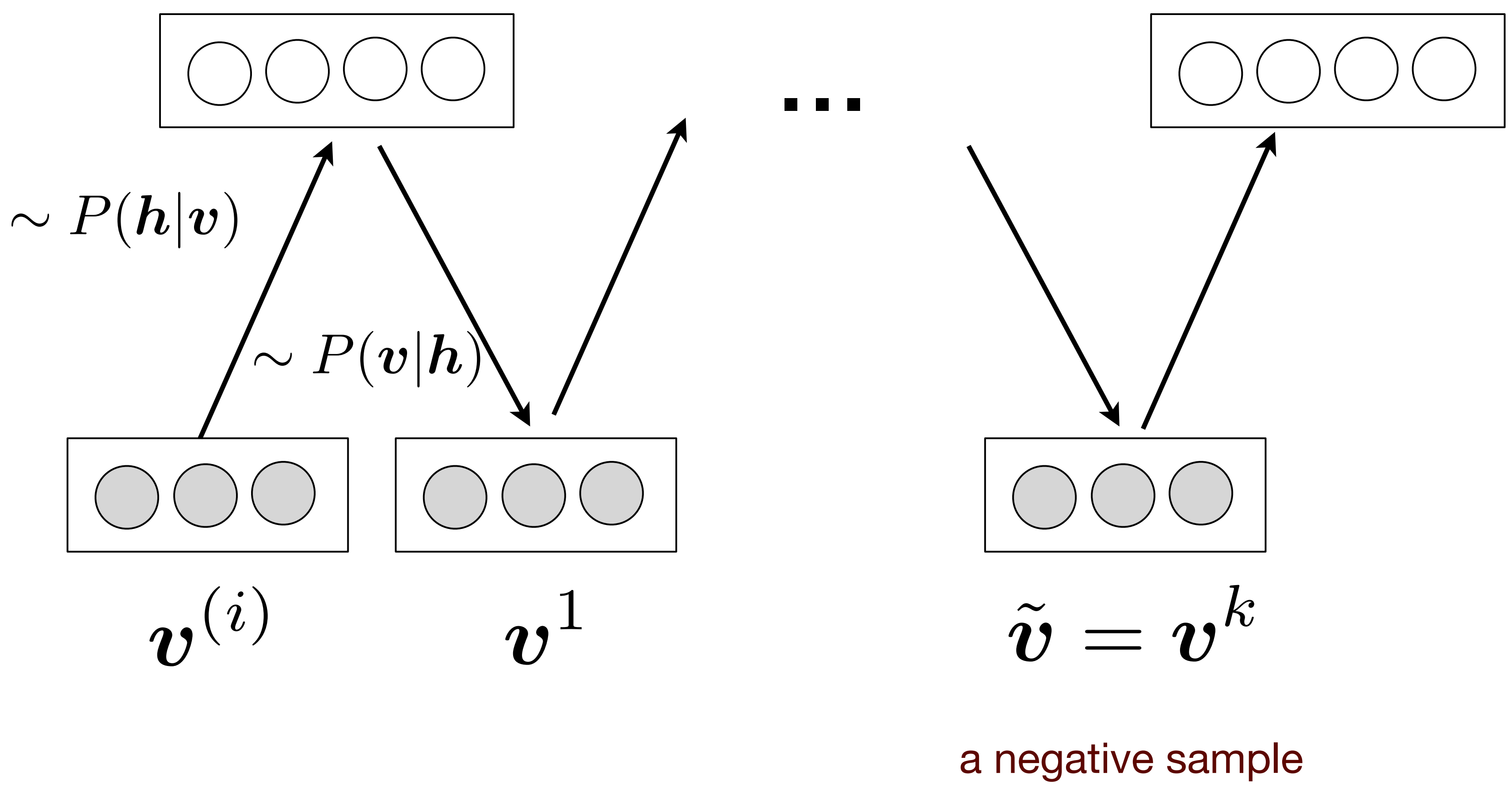
$$\frac{\partial - \log P(\boldsymbol{v}^{(i)})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\boldsymbol{h}}\left[\frac{\partial E(\boldsymbol{v}^{(i)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \middle| \boldsymbol{v}^{(i)}\right]}_{\text{positive phase}} - \underbrace{\mathbb{E}_{\boldsymbol{v},\boldsymbol{h}}\left[\frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}}\right]}_{\text{negative phase}}$$

Reproduced from Hugo Larochelle's slides

Contrastive Divergence (CD)

Idea:

- replace the expectation by a point estimate at $\tilde{\boldsymbol{v}}$
- obtain the point $\tilde{\boldsymbol{v}}$ by Gibbs sampling
- start sampling the chain at $\tilde{\boldsymbol{v}}^{(i)}$

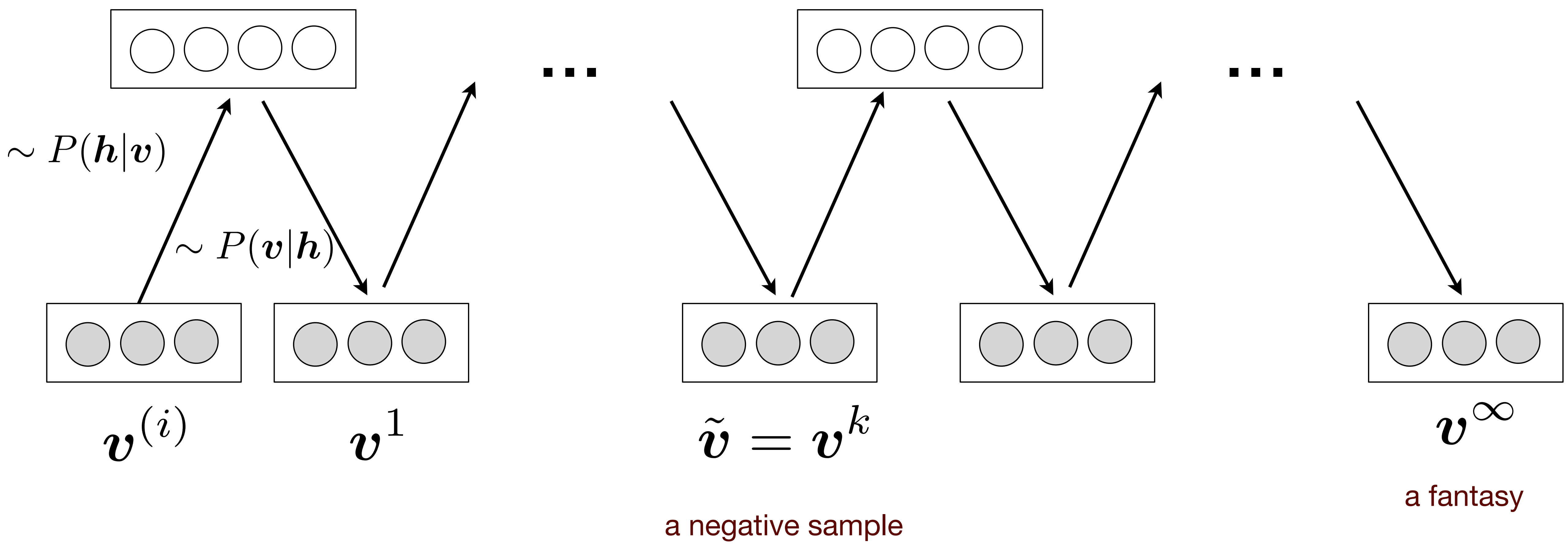


Reproduced from Hugo Larochelle's slides

Contrastive Divergence (CD)

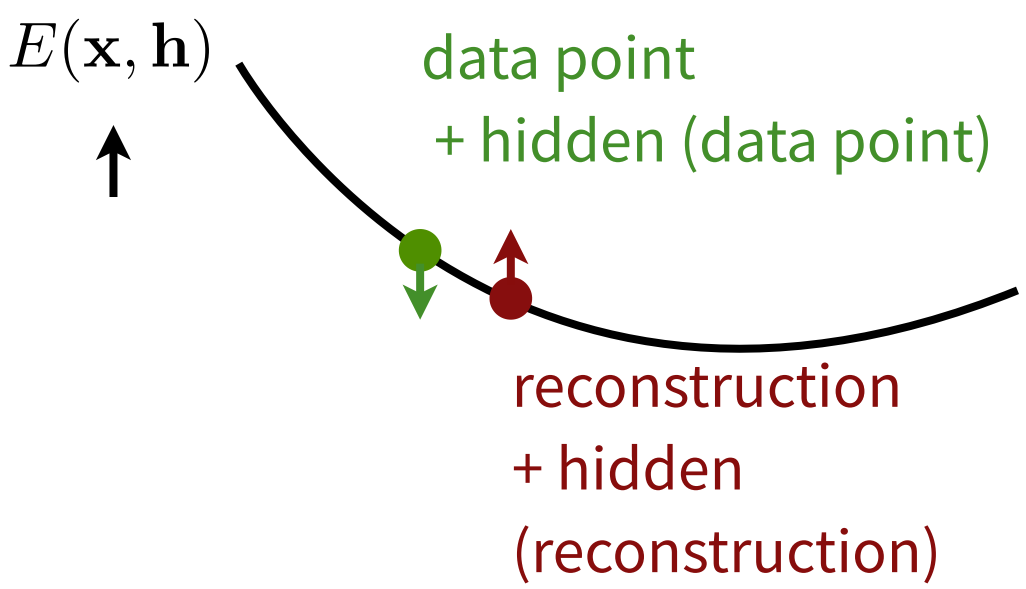
<u>Idea:</u>

- replace the expectation by a point estimate at $\tilde{\boldsymbol{v}}$
- obtain the point $\tilde{\boldsymbol{v}}$ by Gibbs sampling
- start sampling the chain at $\tilde{\boldsymbol{v}}^{(i)}$



Reproduced from Hugo Larochelle's slides

Contrastive Divergence (A picture)



$$\mathbb{E}_{\boldsymbol{h}}\left[\frac{\partial E(\boldsymbol{v}^{(i)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \,\middle|\, \boldsymbol{v}^{(i)}\right] \approx \frac{\partial E(\boldsymbol{v}^{(i)}, \tilde{\boldsymbol{h}}^{(i)})}{\partial \boldsymbol{\theta}}$$

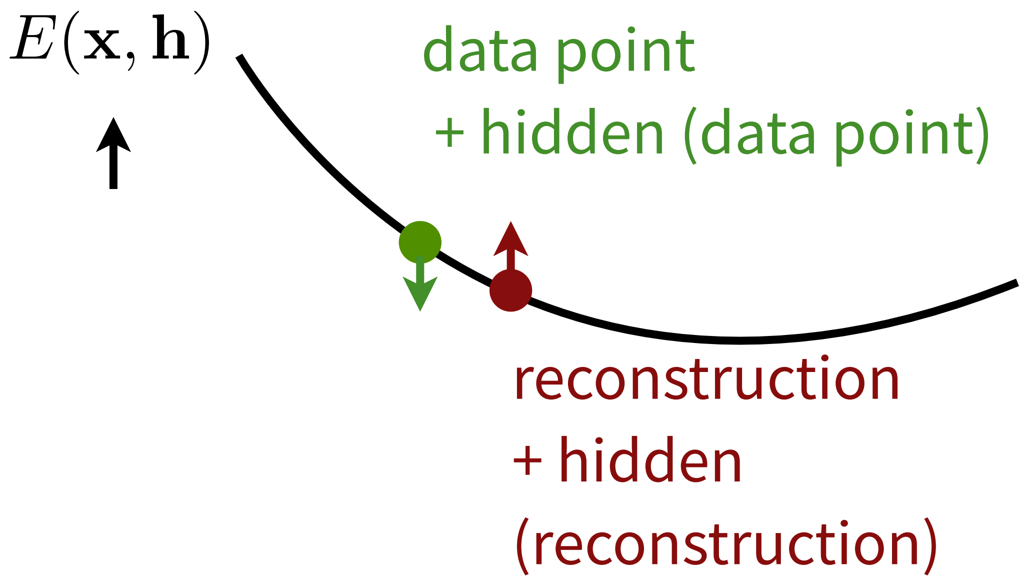
Change the weights to pull the energy down at the data point

$$\mathbb{E}_{\boldsymbol{v},\boldsymbol{h}}\left[\frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}}\right] \approx \frac{\partial E(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{h}})}{\partial \boldsymbol{\theta}}$$

Change the weights to pull the energy up at the reconstruction

Credit: Geoff Hinton

Contrastive Divergence (A picture)

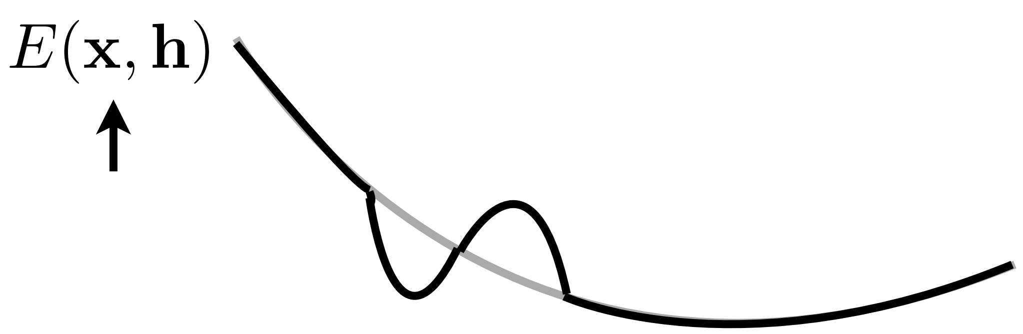


$$\mathbb{E}_{\boldsymbol{h}}\left[\frac{\partial E(\boldsymbol{v}^{(i)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \middle| \boldsymbol{v}^{(i)}\right] \approx \frac{\partial E(\boldsymbol{v}^{(i)}, \tilde{\boldsymbol{h}}^{(i)})}{\partial \boldsymbol{\theta}}$$

Change the weights to pull the energy down at the data point

$$\mathbb{E}_{\boldsymbol{v},\boldsymbol{h}}\left[\frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}}\right] \approx \frac{\partial E(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{h}})}{\partial \boldsymbol{\theta}}$$

Change the weights to pull the energy up at the reconstruction



Credit: Geoff Hinton

Derivation of the Learning Rule

Given $\boldsymbol{v}^{(i)}$ and $\tilde{\boldsymbol{v}}$ the learning rule for $\boldsymbol{\theta} = \boldsymbol{W}$ becomes:

$$
\begin{aligned}
\boldsymbol{W} \Leftarrow \boldsymbol{W} &- \epsilon \left(\nabla_{\boldsymbol{W}} \left\{ -\log P(\boldsymbol{v}^{(i)}) \right\} \right) \\
= \boldsymbol{W} &- \epsilon \left(\mathbb{E}_{\boldsymbol{h}} \left[\nabla_{\boldsymbol{W}} E(\boldsymbol{v}^{(i)}, \boldsymbol{h}) \middle| \boldsymbol{v}^{(i)} \right] - \mathbb{E}_{\boldsymbol{v},\boldsymbol{h}} \left[\nabla_{\boldsymbol{W}} E(\boldsymbol{v}, \boldsymbol{h}) \right] \right) \\
\approx \boldsymbol{W} &- \epsilon \left(\mathbb{E}_{\boldsymbol{h}} \left[\nabla_{\boldsymbol{W}} E(\boldsymbol{v}^{(i)}, \boldsymbol{h}) \middle| \boldsymbol{v}^{(i)} \right] - \mathbb{E}_{\boldsymbol{h}} \left[\nabla_{\boldsymbol{W}} E(\tilde{\boldsymbol{v}}, \boldsymbol{h}) \left| \tilde{\boldsymbol{v}} \right. \right] \right) \\
= \boldsymbol{W} &+ \epsilon \left(\boldsymbol{h}(\boldsymbol{v}^{(i)}) \boldsymbol{v}^{(i)\top} - \boldsymbol{h}(\tilde{\boldsymbol{v}}) \tilde{\boldsymbol{v}}^{\top} \right)
\end{aligned}
$$

CD-k: Algorithm

1. For each training example $\boldsymbol{v}^{(i)}$

   i. generate a negative sample $\tilde{v}$ using k steps of Gibbs sampling, starting at $\boldsymbol{v}^{(i)}$

   ii. update parameters:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} + \epsilon \left(\boldsymbol{h}(\boldsymbol{v}^{(i)}) \boldsymbol{v}^{(i)\top} - \boldsymbol{h}(\tilde{\boldsymbol{v}}) \tilde{\boldsymbol{v}}^{\top} \right)$$

$$\boldsymbol{c} \leftarrow \boldsymbol{c} + \epsilon \left(\boldsymbol{h}(\boldsymbol{v}^{(i)}) - \boldsymbol{h}(\tilde{\boldsymbol{v}}) \right)$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} + \epsilon \left(\boldsymbol{v}^{(i)} - \tilde{\boldsymbol{v}} \right)$$

2. Go back to 1 until stopping criteria

Reproduced from Hugo Larochelle's slides

CD-k: Algorithm

1. For each training example $\boldsymbol{v}^{(i)}$

   i. generate a negative sample $\tilde{\boldsymbol{v}}$ using k steps of Gibbs sampling, starting at $\boldsymbol{v}^{(i)}$

   ii. update parameters:

$$\boldsymbol{W} \Leftarrow \boldsymbol{W} + \epsilon \left(\boldsymbol{h}(\boldsymbol{v}^{(i)}) \boldsymbol{v}^{(i)\top} - \boldsymbol{h}(\tilde{\boldsymbol{v}}) \tilde{\boldsymbol{v}}^{\top} \right)$$

$$\boldsymbol{c} \Leftarrow \boldsymbol{c} + \epsilon \left(\boldsymbol{h}(\boldsymbol{v}^{(i)}) - \boldsymbol{h}(\tilde{\boldsymbol{v}}) \right)$$

$$\boldsymbol{b} \Leftarrow \boldsymbol{b} + \epsilon \left(\boldsymbol{v}^{(i)} - \tilde{\boldsymbol{v}} \right)$$

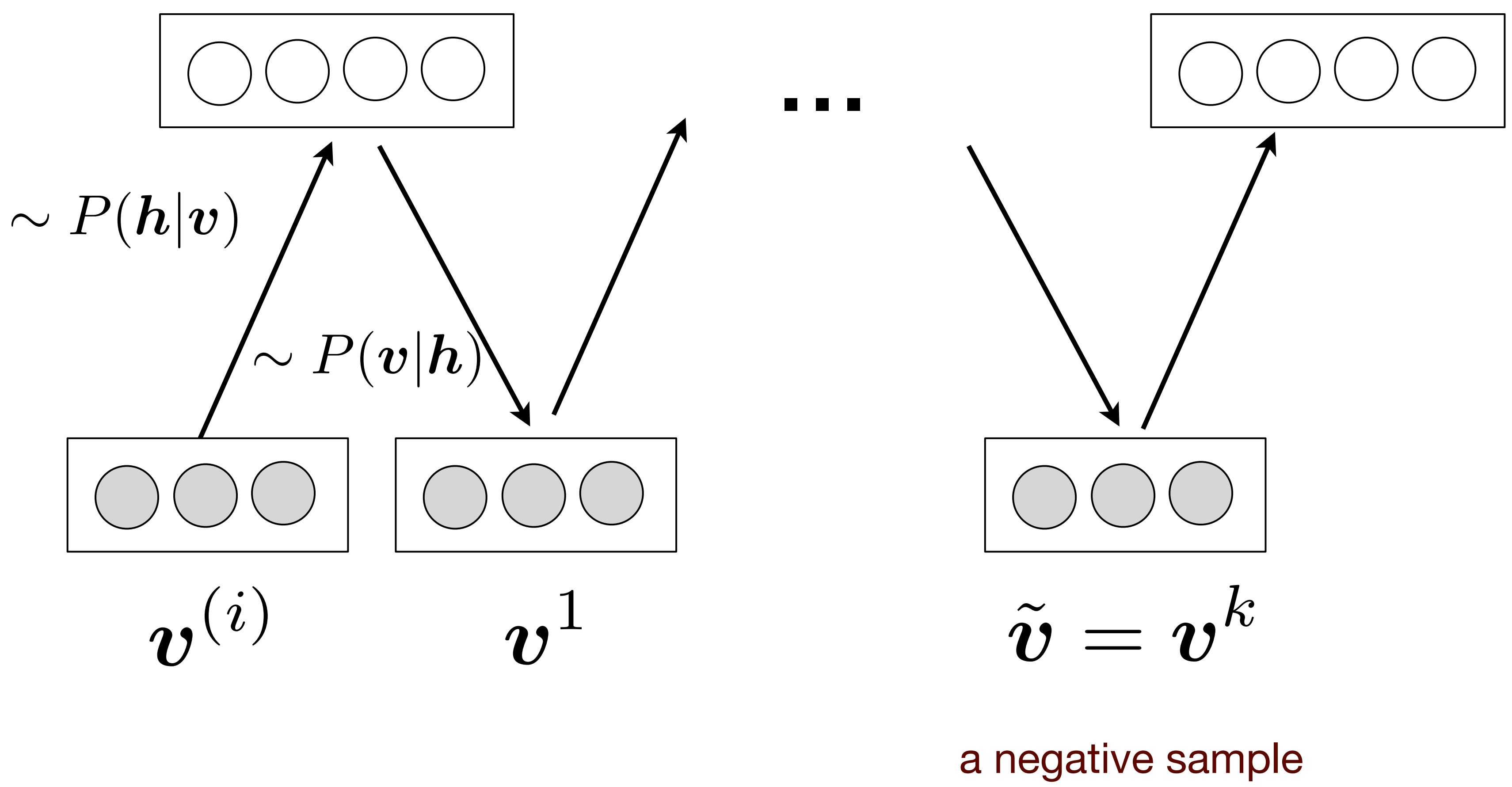
2. Go back to 1 until stopping criteria

In general, the bigger k, the less biased the estimate of the gradient.
In practice, $k = 1$ works well for pre-training.

Persistent CD (PCD) (Tieleman 2008)

Idea:

- instead of initializing the chain to $\tilde{\boldsymbol{v}}^{(i)}$, initialize the chain to the negative sample of the last iteration

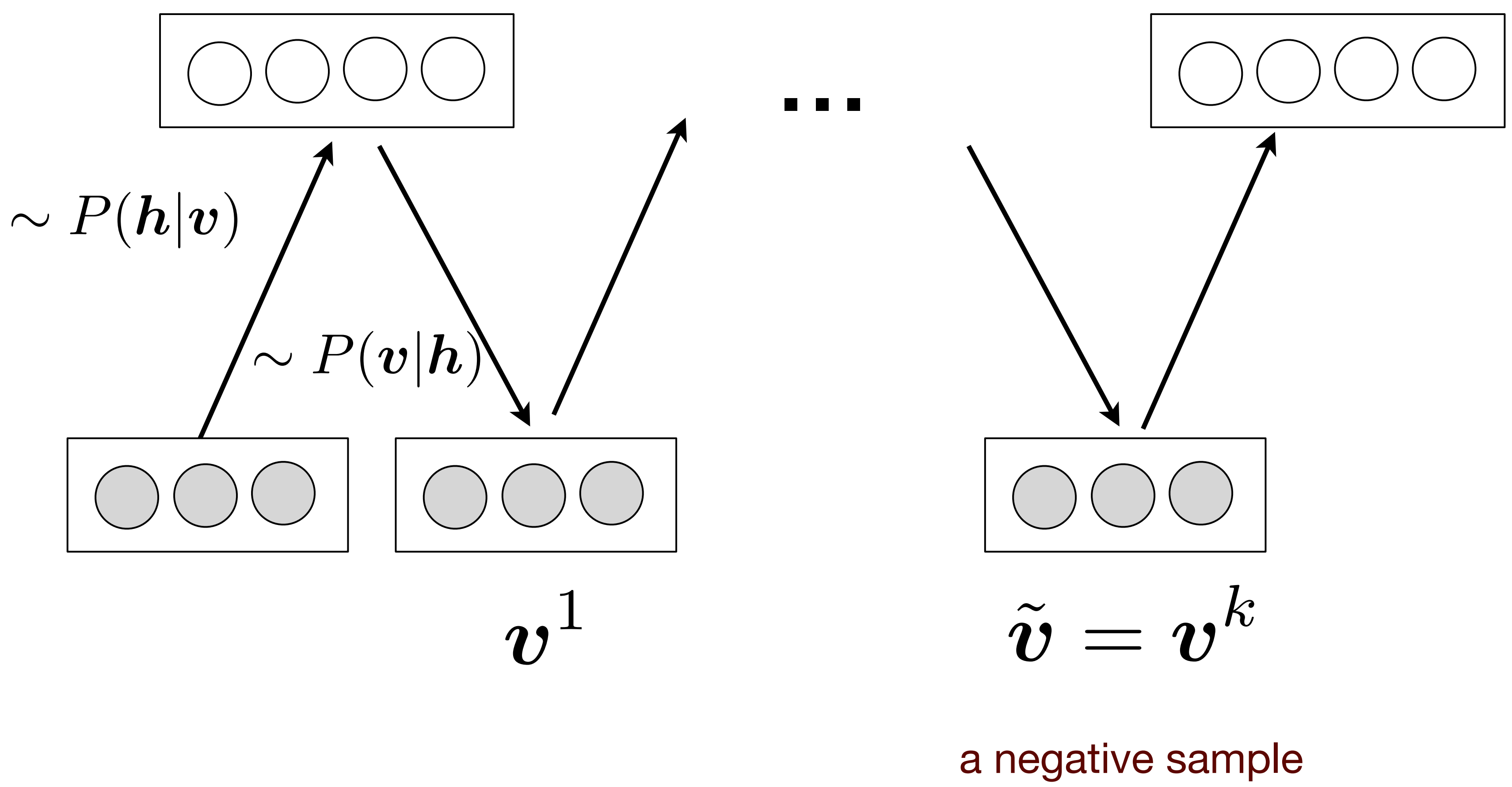


Reproduced from Hugo Larochelle's slides

Persistent CD (PCD) (Tieleman 2008)

Idea:

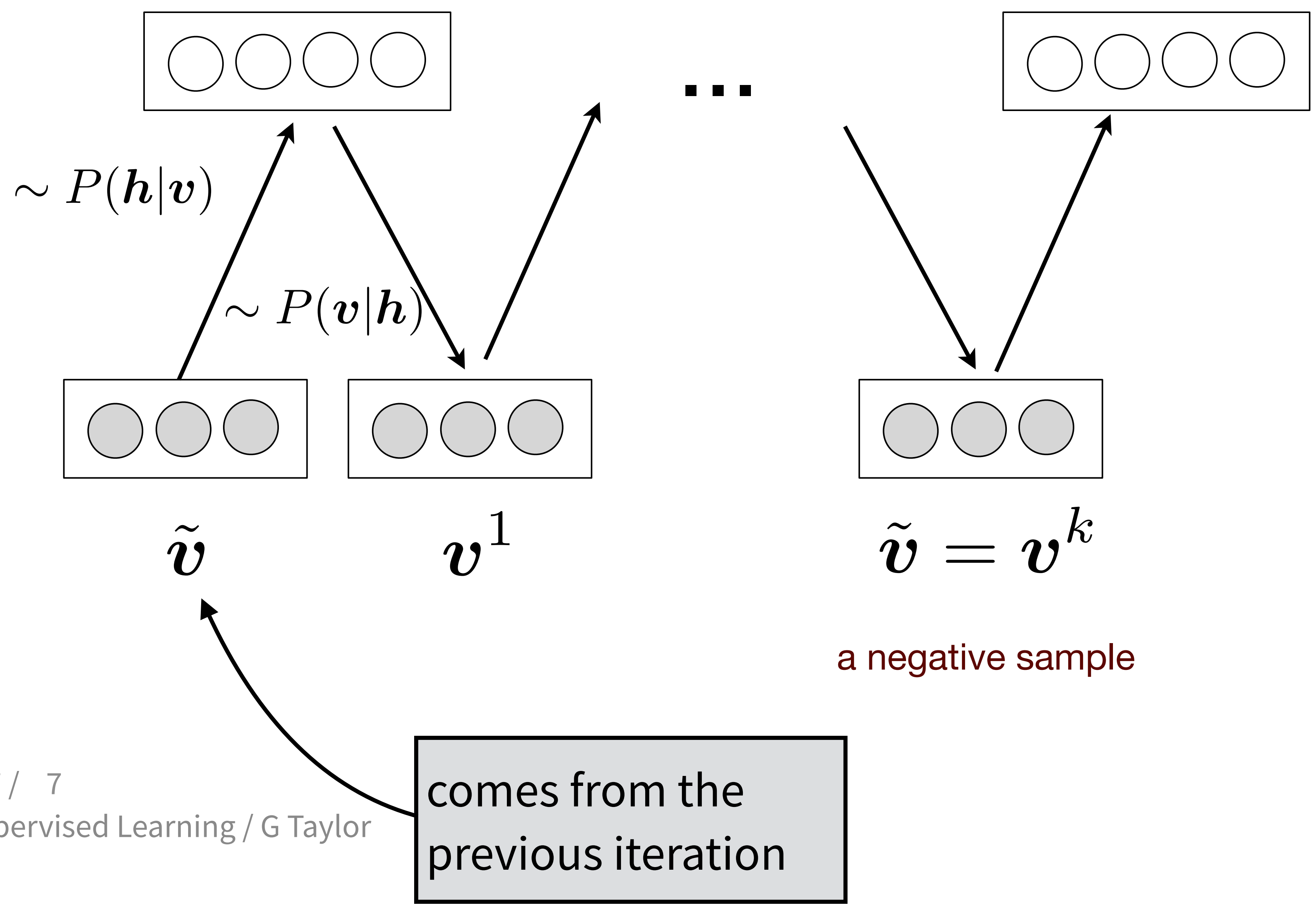
- instead of initializing the chain to $\tilde{\boldsymbol{v}}^{(i)}$, initialize the chain to the negative sample of the last iteration



Persistent CD (PCD) (Tieleman 2008)

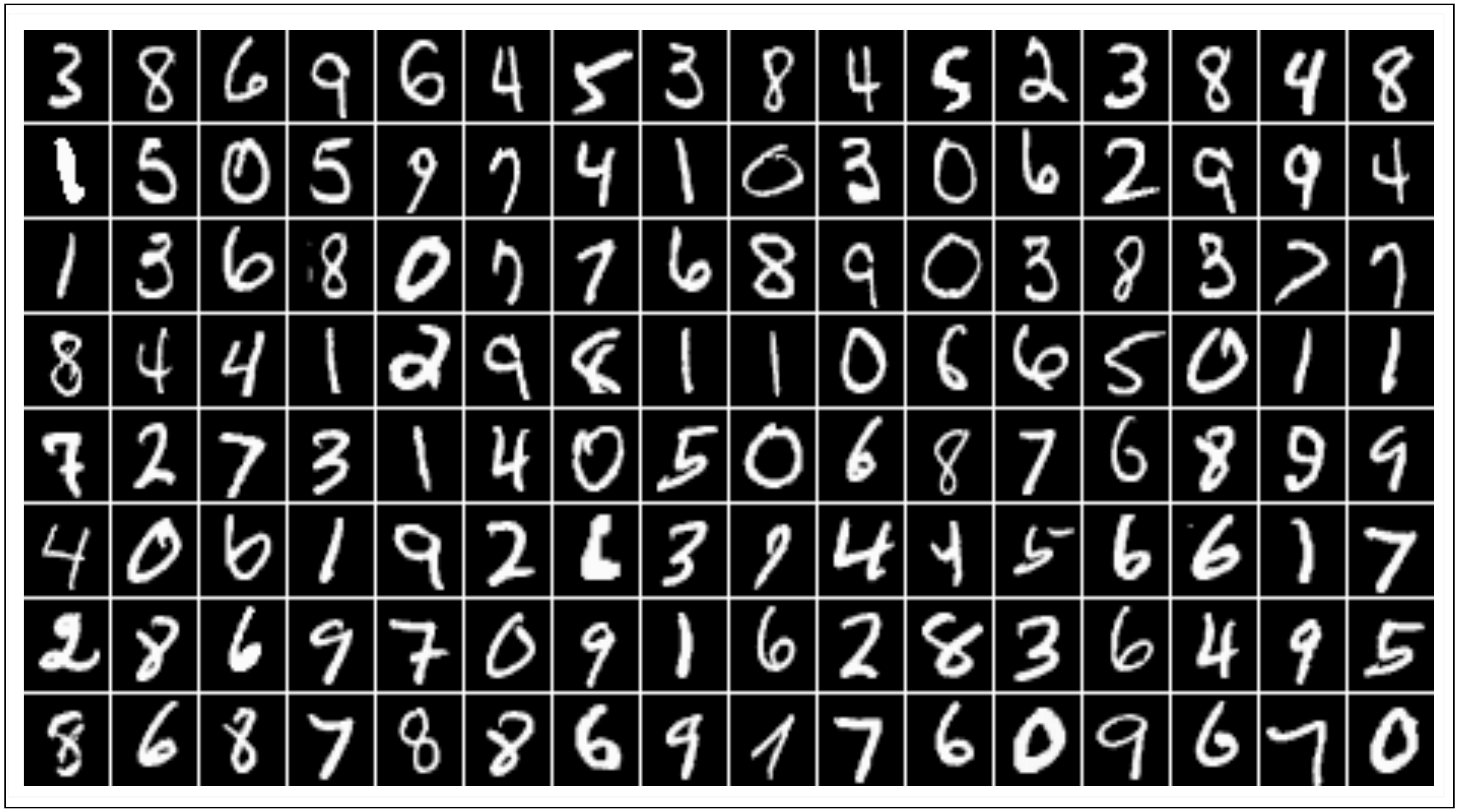
Idea:

- instead of initializing the chain to $\tilde{\boldsymbol{v}}^{(i)}$, initialize the chain to the negative sample of the last iteration



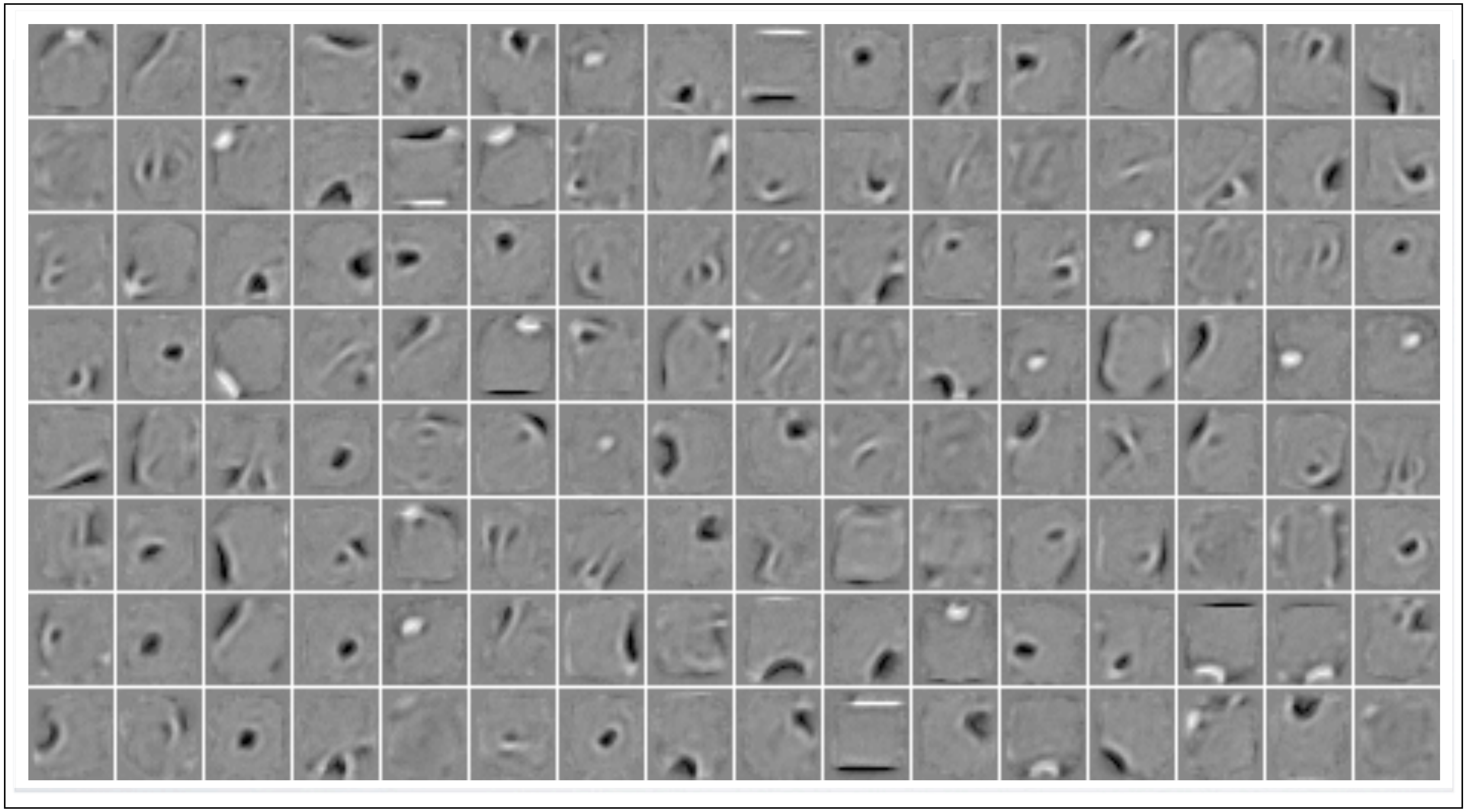
Reproduced from Hugo Larochelle's slides

RBM: Example (MNIST)



Reproduced from Hugo Larochelle's slides

RBM: Example (Filters)



Reproduced from Hugo Larochelle's slides

RBMs: Debugging

- It's very difficult to monitor the training of RBMs because we do not have access to the log likelihood
- It's also not possible to check gradients with finite differences
- Can instead rely on some approximate "tricks":
  - plot average stochastic reconstruction $||\boldsymbol{v}^{(i)} - \tilde{\boldsymbol{v}}||^2$ and see if it tends to decrease
  - for inputs that correspond to images, visualize the connection coming into each hidden unit (the filters)
  - can also try to approximate the partition function Z and see whether the (approximated) NLL decreases

Gaussian-Bernoulli RBM

What if inputs are unbounded real values?

- Add a quadratic term to the energy function

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h} - \frac{1}{2}\boldsymbol{v}^\top \boldsymbol{v}$$

- Only thing that changes is that $P(\boldsymbol{v}|\boldsymbol{h})$ is now a Gaussian distribution with mean $\boldsymbol{\mu} = \boldsymbol{b} + \boldsymbol{W}^\top \boldsymbol{h}$ and identity covariance matrix

- Recommended to normalize the training set by
  - subtracting the mean of each input
  - dividing each input v_k by the training set standard deviation

- Should use a smaller learning rate than in a binary RBM