

Neural networks

Sparse coding - definition

UNSUPERVISED LEARNING

Topics: unsupervised learning

- Unsupervised learning: only use the inputs $\mathbf{x}^{(t)}$ for learning
 - ▶ automatically extract meaningful features for your data
 - ▶ leverage the availability of unlabeled data
 - ▶ add a data-dependent regularizer to trainings
- We will see 3 neural networks for unsupervised learning
 - ▶ restricted Boltzmann machines
 - ▶ autoencoders
 - ▶ **sparse coding model**

SPARSE CODING

Topics: sparse coding

- For each $\mathbf{x}^{(t)}$ find a latent representation $\mathbf{h}^{(t)}$ such that:
 - ▶ it is sparse: the vector $\mathbf{h}^{(t)}$ has many zeros
 - ▶ we can reconstruct the original input $\mathbf{x}^{(t)}$ as well as possible
- More formally:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2 + \lambda \|\mathbf{h}^{(t)}\|_1$$

- ▶ we also constrain the columns of \mathbf{D} to be of norm 1
 - otherwise, \mathbf{D} could grow big while $\mathbf{h}^{(t)}$ becomes small to satisfy the prior
- ▶ sometimes the columns are constrained to be no greater than 1

SPARSE CODING

Topics: sparse coding

- For each $\mathbf{x}^{(t)}$ find a latent representation $\mathbf{h}^{(t)}$ such that:
 - ▶ it is sparse: the vector $\mathbf{h}^{(t)}$ has many zeros
 - ▶ we can reconstruct the original input $\mathbf{x}^{(t)}$ as well as possible
- More formally:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} \underbrace{\frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2}_{\text{reconstruction error}} + \lambda \|\mathbf{h}^{(t)}\|_1$$

- ▶ we also constrain the columns of \mathbf{D} to be of norm 1
 - otherwise, \mathbf{D} could grow big while $\mathbf{h}^{(t)}$ becomes small to satisfy the prior
- ▶ sometimes the columns are constrained to be no greater than 1

SPARSE CODING

Topics: sparse coding

- For each $\mathbf{x}^{(t)}$ find a latent representation $\mathbf{h}^{(t)}$ such that:
 - ▶ it is sparse: the vector $\mathbf{h}^{(t)}$ has many zeros
 - ▶ we can reconstruct the original input $\mathbf{x}^{(t)}$ as well as possible
- More formally:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} \underbrace{\frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2}_{\text{reconstruction error}} + \lambda \|\mathbf{h}^{(t)}\|_1$$

reconstruction $\widehat{\mathbf{x}}^{(t)}$

- ▶ we also constrain the columns of \mathbf{D} to be of norm 1
 - otherwise, \mathbf{D} could grow big while $\mathbf{h}^{(t)}$ becomes small to satisfy the prior
- ▶ sometimes the columns are constrained to be no greater than 1

SPARSE CODING

Topics: sparse coding

- For each $\mathbf{x}^{(t)}$ find a latent representation $\mathbf{h}^{(t)}$ such that:

- ▶ it is sparse: the vector $\mathbf{h}^{(t)}$ has many zeros
- ▶ we can reconstruct the original input $\mathbf{x}^{(t)}$ as well as possible

- More formally:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} \underbrace{\frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2}_{\text{reconstruction error}} + \lambda \underbrace{\|\mathbf{h}^{(t)}\|_1}_{\text{sparsity penalty}}$$

reconstruction $\widehat{\mathbf{x}}^{(t)}$

- ▶ we also constrain the columns of \mathbf{D} to be of norm 1
 - otherwise, \mathbf{D} could grow big while $\mathbf{h}^{(t)}$ becomes small to satisfy the prior
- ▶ sometimes the columns are constrained to be no greater than 1

SPARSE CODING

Topics: sparse coding

- For each $\mathbf{x}^{(t)}$ find a latent representation $\mathbf{h}^{(t)}$ such that:

- ▶ it is sparse: the vector $\mathbf{h}^{(t)}$ has many zeros
- ▶ we can reconstruct the original input $\mathbf{x}^{(t)}$ as well as possible

- More formally:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} \underbrace{\frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2}_{\text{reconstruction error}} + \lambda \underbrace{\|\mathbf{h}^{(t)}\|_1}_{\text{sparsity penalty}}$$

reconstruction error
 sparsity penalty
 reconstruction vs.
 sparsity control

- ▶ we also constrain the columns of \mathbf{D} to be of norm 1
 - otherwise, \mathbf{D} could grow big while $\mathbf{h}^{(t)}$ becomes small to satisfy the prior
- ▶ sometimes the columns are constrained to be no greater than 1

SPARSE CODING

Topics: sparse coding

- For each $\mathbf{x}^{(t)}$ find a latent representation $\mathbf{h}^{(t)}$ such that:

- it is sparse: the vector $\mathbf{h}^{(t)}$ has many zeros
- we can reconstruct the original input $\mathbf{x}^{(t)}$ as well as possible

- More formally:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} \underbrace{\frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2}_{\text{reconstruction error}} + \lambda \underbrace{\|\mathbf{h}^{(t)}\|_1}_{\text{sparsity penalty}}$$

reconstruction sparsity penalty

$\widehat{\mathbf{x}}^{(t)}$

reconstruction vs.
sparsity control

- \mathbf{D} is equivalent to the autoencoder output weight matrix
- however, $\mathbf{h}(\mathbf{x}^{(t)})$ is now a complicated function of $\mathbf{x}^{(t)}$
 - encoder is the minimization $\mathbf{h}(\mathbf{x}^{(t)}) = \arg \min_{\mathbf{h}^{(t)}} \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2 + \lambda \|\mathbf{h}^{(t)}\|_1$

SPARSE CODING

Topics: dictionary

- Can also write $\hat{\mathbf{x}}^{(t)} = \mathbf{D} \mathbf{h}(\mathbf{x}^{(t)}) = \sum_{\substack{k \text{ s.t.} \\ h(\mathbf{x}^{(t)})_k \neq 0}} \mathbf{D}_{\cdot, k} h(\mathbf{x}^{(t)})_k$

$$\begin{aligned} \text{7} &= 1 \begin{array}{|c|} \hline \text{8} \\ \hline \end{array} + 1 \begin{array}{|c|} \hline \text{7} \\ \hline \end{array} + 1 \begin{array}{|c|} \hline \text{2} \\ \hline \end{array} + 1 \begin{array}{|c|} \hline \text{4} \\ \hline \end{array} + 1 \begin{array}{|c|} \hline \text{0} \\ \hline \end{array} \\ &\quad + 1 \begin{array}{|c|} \hline \text{1} \\ \hline \end{array} + 1 \begin{array}{|c|} \hline \text{3} \\ \hline \end{array} + 0.8 \begin{array}{|c|} \hline \text{5} \\ \hline \end{array} + 0.8 \begin{array}{|c|} \hline \text{6} \\ \hline \end{array} \end{aligned}$$

- ▶ we also refer to **D** as the dictionary
 - in certain applications, we know what dictionary matrix to use
 - often however, we have to learn it

SPARSE CODING

Topics: dictionary

- Can also write $\hat{\mathbf{x}}^{(t)} = \mathbf{D} \mathbf{h}(\mathbf{x}^{(t)}) = \sum_{\substack{k \text{ s.t.} \\ h(\mathbf{x}^{(t)})_k \neq 0}} \mathbf{D}_{\cdot, k} h(\mathbf{x}^{(t)})_k$

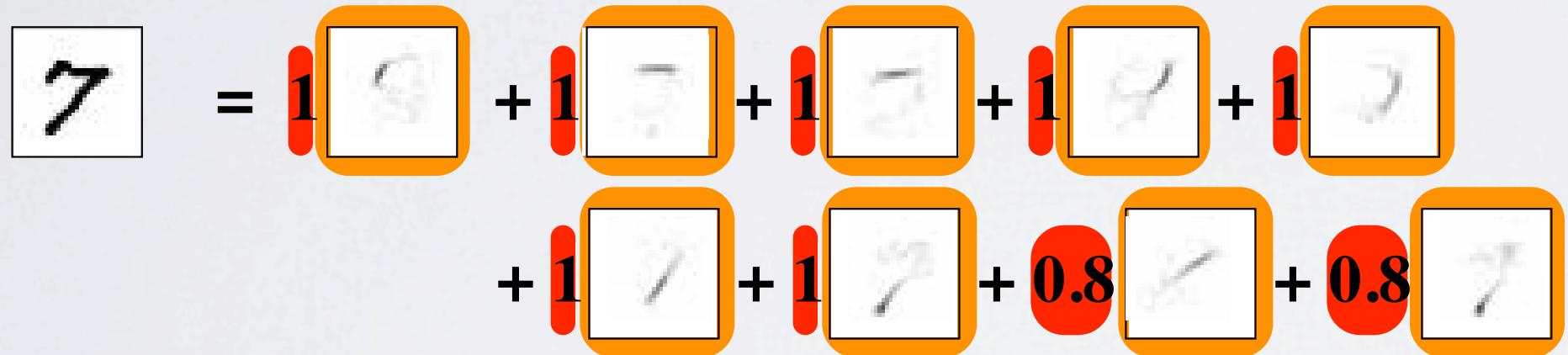
$$\begin{matrix} \text{7} \\ = 1 \end{matrix} \begin{matrix} \text{8} \end{matrix} + 1 \begin{matrix} \text{7} \end{matrix} + 1 \begin{matrix} \text{2} \end{matrix} + 1 \begin{matrix} \text{4} \end{matrix} + 1 \begin{matrix} \text{0} \end{matrix} \\ + 1 \begin{matrix} \text{1} \end{matrix} + 1 \begin{matrix} \text{2} \end{matrix} + 0.8 \begin{matrix} \text{3} \end{matrix} + 0.8 \begin{matrix} \text{7} \end{matrix}$$

- ▶ we also refer to **D** as the dictionary
 - in certain applications, we know what dictionary matrix to use
 - often however, we have to learn it

SPARSE CODING

Topics: dictionary

- Can also write $\hat{\mathbf{x}}^{(t)} = \mathbf{D} \mathbf{h}(\mathbf{x}^{(t)}) = \sum_{\substack{k \text{ s.t.} \\ h(\mathbf{x}^{(t)})_k \neq 0}} \mathbf{D}_{\cdot, k} h(\mathbf{x}^{(t)})_k$



- ▶ we also refer to **D** as the dictionary
 - in certain applications, we know what dictionary matrix to use
 - often however, we have to learn it

Neural networks

Sparse coding - inference (ISTA algorithm)

SPARSE CODING

Topics: sparse coding

- For each $\mathbf{x}^{(t)}$ find a latent representation $\mathbf{h}^{(t)}$ such that:

- it is sparse: the vector $\mathbf{h}^{(t)}$ has many zeros
- we can reconstruct the original input $\mathbf{x}^{(t)}$ as much as possible

- More formally:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} \underbrace{\frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2}_{\text{reconstruction error}} + \lambda \underbrace{\|\mathbf{h}^{(t)}\|_1}_{\text{sparsity penalty}}$$

reconstruction sparsity penalty

$\widehat{\mathbf{x}}^{(t)}$

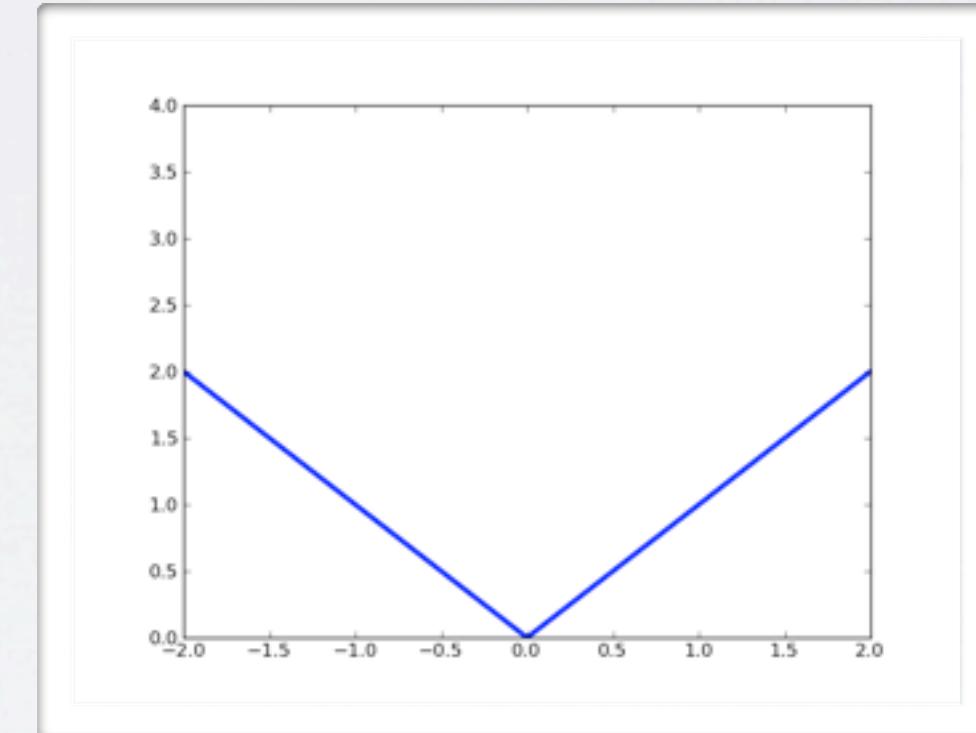
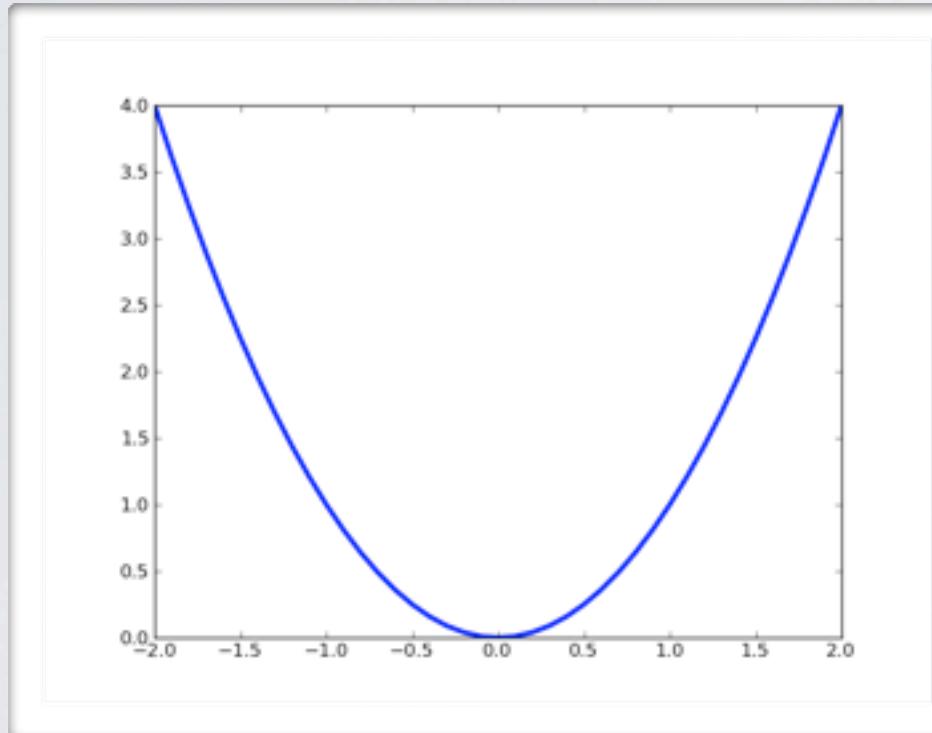
reconstruction vs.
sparsity control

- \mathbf{D} is equivalent to the autoencoder output weight matrix
- however, $\mathbf{h}(\mathbf{x}^{(t)})$ is now a complicated function of $\mathbf{x}^{(t)}$
 - encoder is the minimization $\mathbf{h}(\mathbf{x}^{(t)}) = \arg \min_{\mathbf{h}^{(t)}} \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2 + \lambda \|\mathbf{h}^{(t)}\|_1$

SPARSE CODING

Topics: inference of sparse codes

- Given \mathbf{D} , how do we compute $\mathbf{h}(\mathbf{x}^{(t)})$
 - we want to optimize $l(\mathbf{x}^{(t)}) = \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2 + \lambda \|\mathbf{h}^{(t)}\|_1$ w.r.t. $\mathbf{h}^{(t)}$



- we could use a gradient descent method:

$$\nabla_{\mathbf{h}^{(t)}} l(\mathbf{x}^{(t)}) = \mathbf{D}^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)}) + \lambda \text{sign}(\mathbf{h}^{(t)})$$

SPARSE CODING

Topics: inference of sparse codes

- For a single hidden unit:

$$\frac{\partial}{\partial h_k^{(t)}} l(\mathbf{x}^{(t)}) = (\mathbf{D}_{\cdot,k})^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)}) + \lambda \operatorname{sign}(h_k^{(t)})$$

- ▶ issue: L1 norm not differentiable at 0
 - very unlikely for gradient descent to “land” on $h_k^{(t)} = 0$ (even if it’s the solution)
- ▶ solution: if $h_k^{(t)}$ changes sign because of L1 norm gradient, clamp to 0
- ▶ each hidden unit update would be performed as follows:
 - $h_k^{(t)} \leftarrow h_k^{(t)} - \alpha (\mathbf{D}_{\cdot,k})^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$
 - if $\operatorname{sign}(h_k^{(t)}) \neq \operatorname{sign}(h_k^{(t)} - \alpha \lambda \operatorname{sign}(h_k^{(t)}))$ then: $h_k^{(t)} \leftarrow 0$
 - else: $h_k^{(t)} \leftarrow h_k^{(t)} - \alpha \lambda \operatorname{sign}(h_k^{(t)})$

SPARSE CODING

Topics: inference of sparse codes

- For a single hidden unit:

$$\frac{\partial}{\partial h_k^{(t)}} l(\mathbf{x}^{(t)}) = (\mathbf{D}_{\cdot,k})^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)}) + \lambda \operatorname{sign}(h_k^{(t)})$$

- ▶ issue: L1 norm not differentiable at 0
 - very unlikely for gradient descent to “land” on $h_k^{(t)} = 0$ (even if it’s the solution)
- ▶ solution: if $h_k^{(t)}$ changes sign because of L1 norm gradient, clamp to 0
- ▶ each hidden unit update would be performed as follows:
 - $h_k^{(t)} \leftarrow h_k^{(t)} - \alpha (\mathbf{D}_{\cdot,k})^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$
 - if $\operatorname{sign}(h_k^{(t)}) \neq \operatorname{sign}(h_k^{(t)} - \alpha \lambda \operatorname{sign}(h_k^{(t)}))$ then: $h_k^{(t)} \leftarrow 0$
 - else: $h_k^{(t)} \leftarrow h_k^{(t)} - \alpha \lambda \operatorname{sign}(h_k^{(t)})$

SPARSE CODING

Topics: inference of sparse codes

- For a single hidden unit:

$$\frac{\partial}{\partial h_k^{(t)}} l(\mathbf{x}^{(t)}) = (\mathbf{D}_{\cdot,k})^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)}) + \lambda \operatorname{sign}(h_k^{(t)})$$

- ▶ issue: L1 norm not differentiable at 0
 - very unlikely for gradient descent to “land” on $h_k^{(t)} = 0$ (even if it’s the solution)
- ▶ solution: if $h_k^{(t)}$ changes sign because of L1 norm gradient, clamp to 0
- ▶ each hidden unit update would be performed as follows:

$$\left. \begin{array}{l} - h_k^{(t)} \leftarrow h_k^{(t)} - \alpha (\mathbf{D}_{\cdot,k})^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)}) \\ - \text{if } \operatorname{sign}(h_k^{(t)}) \neq \operatorname{sign}(h_k^{(t)} - \alpha \lambda \operatorname{sign}(h_k^{(t)})) \text{ then: } h_k^{(t)} \leftarrow 0 \\ - \text{else: } h_k^{(t)} \leftarrow h_k^{(t)} - \alpha \lambda \operatorname{sign}(h_k^{(t)}) \end{array} \right\} \begin{array}{l} \text{update from reconstruction} \\ \text{update from sparsity} \end{array}$$

SPARSE CODING

Topics: ISTA (Iterative Shrinkage and Thresholding Algorithm)

- This process corresponds to the ISTA algorithm:

```
▶ initialize  $\mathbf{h}^{(t)}$  (for instance to 0)  
▶ while  $\mathbf{h}^{(t)}$  has not converged  
    -  $\mathbf{h}^{(t)} \leftarrow \mathbf{h}^{(t)} - \alpha \mathbf{D}^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$   
    -  $\mathbf{h}^{(t)} \leftarrow \text{shrink}(\mathbf{h}^{(t)}, \alpha \lambda)$   
▶ return  $\mathbf{h}^{(t)}$ 
```

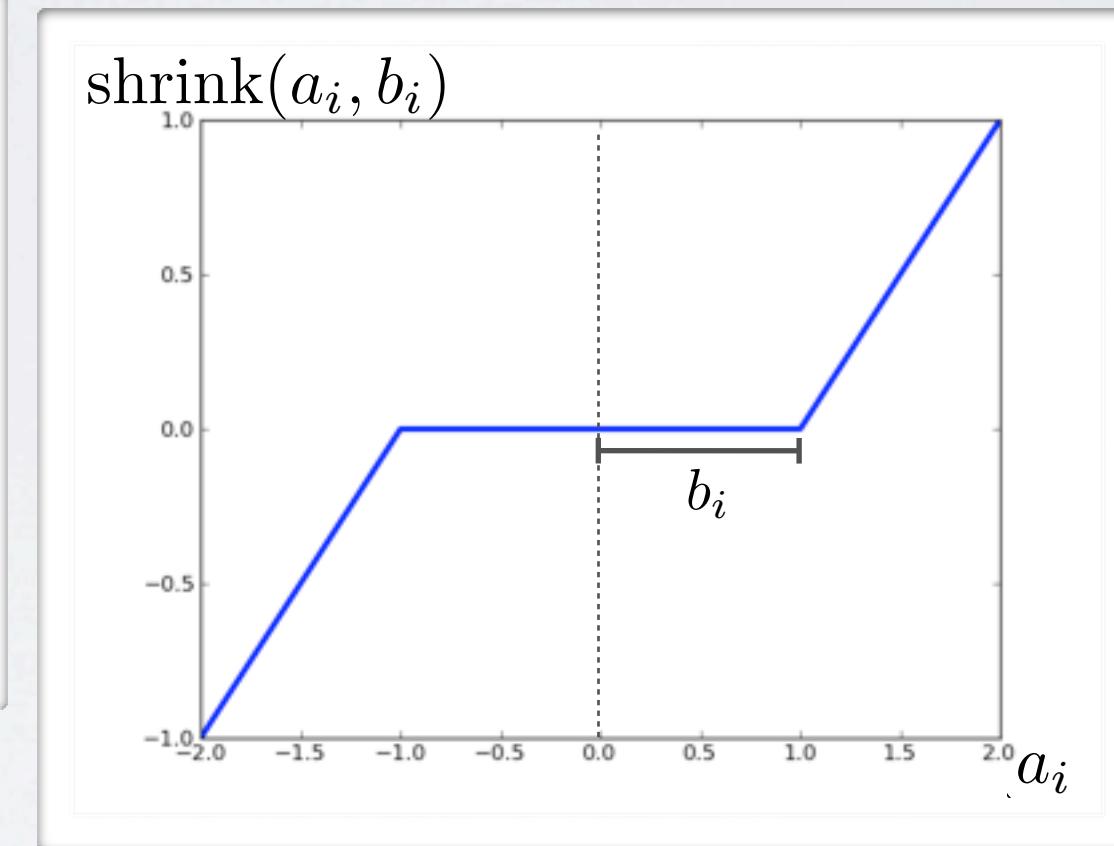
- Here $\text{shrink}(\mathbf{a}, \mathbf{b}) = [\dots, \text{sign}(a_i) \max(|a_i| - b_i, 0), \dots]$
- Will converge if $\frac{1}{\alpha}$ is bigger than the largest eigenvalue of $\mathbf{D}^\top \mathbf{D}$

SPARSE CODING

Topics: ISTA (Iterative Shrinkage and Thresholding Algorithm)

- This process corresponds to the ISTA algorithm:

- initialize $\mathbf{h}^{(t)}$ (for instance to 0)
- while $\mathbf{h}^{(t)}$ has not converged
 - $\mathbf{h}^{(t)} \leftarrow \mathbf{h}^{(t)} - \alpha \mathbf{D}^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$
 - $\mathbf{h}^{(t)} \leftarrow \text{shrink}(\mathbf{h}^{(t)}, \alpha \lambda)$
- return $\mathbf{h}^{(t)}$



- Here $\text{shrink}(\mathbf{a}, \mathbf{b}) = [\dots, \text{sign}(a_i) \max(|a_i| - b_i, 0), \dots]$
- Will converge if $\frac{1}{\alpha}$ is bigger than the largest eigenvalue of $\mathbf{D}^\top \mathbf{D}$

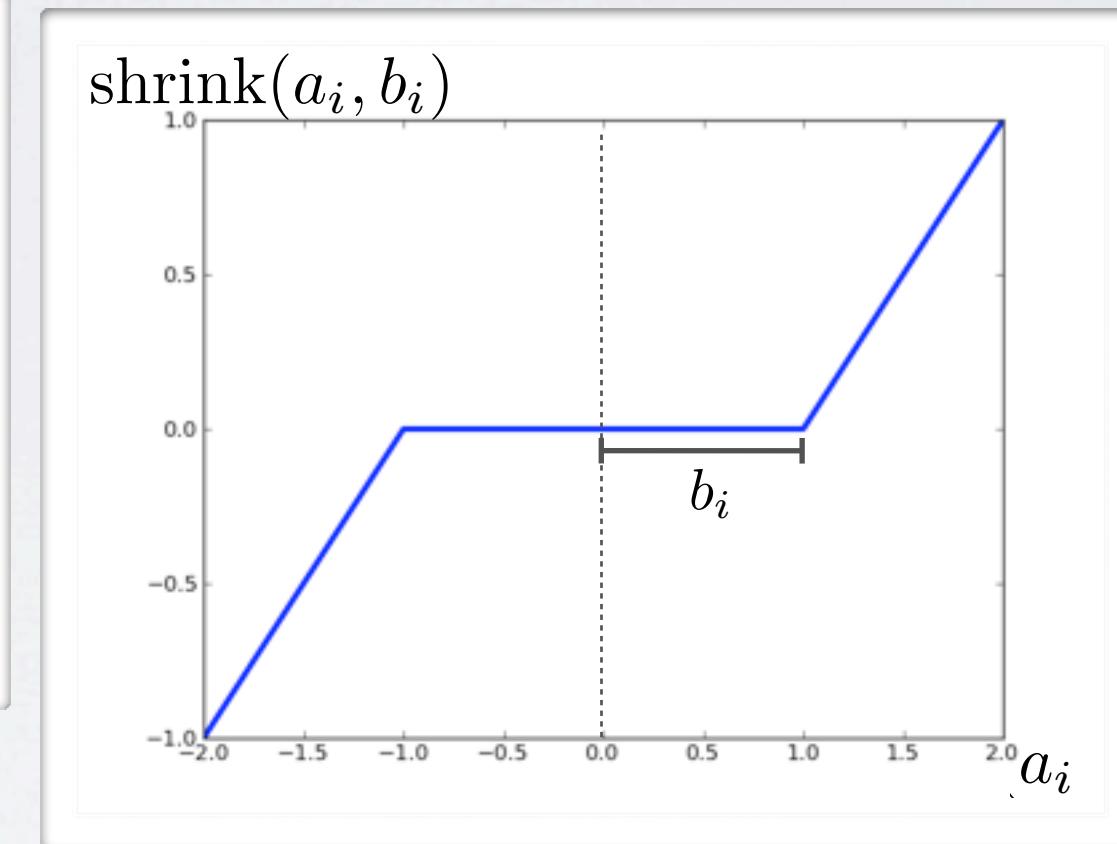
SPARSE CODING

Topics: ISTA (Iterative Shrinkage and Thresholding Algorithm)

- This process corresponds to the ISTA algorithm:

- initialize $\mathbf{h}^{(t)}$ (for instance to 0)
- while $\mathbf{h}^{(t)}$ has not converged
 - $\mathbf{h}^{(t)} \leftarrow \mathbf{h}^{(t)} - \alpha \mathbf{D}^\top (\mathbf{D} \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$
 - $\mathbf{h}^{(t)} \leftarrow \text{shrink}(\mathbf{h}^{(t)}, \alpha \lambda)$
- return $\mathbf{h}^{(t)}$

this is $\mathbf{h}(\mathbf{x}^{(t)})$



- Here $\text{shrink}(\mathbf{a}, \mathbf{b}) = [\dots, \text{sign}(a_i) \max(|a_i| - b_i, 0), \dots]$
- Will converge if $\frac{1}{\alpha}$ is bigger than the largest eigenvalue of $\mathbf{D}^\top \mathbf{D}$

SPARSE CODING

Topics: coordinate descent for sparse coding inference

- ISTA updates all hidden units simultaneously
 - ▶ this is wasteful if many hidden units have already converged
- Idea: update only the “most promising” hidden unit
 - ▶ see coordinate descent algorithm in
 - Learning Fast Approximations of Sparse Coding.
Gregor and Lecun, 2010.
 - ▶ this algorithm has the advantage of not requiring a learning rate α

Neural networks

Sparse coding - dictionary update with projected gradient descent

SPARSE CODING

Topics: sparse coding

- For each $\mathbf{x}^{(t)}$ find a latent representation $\mathbf{h}^{(t)}$ such that:

- it is sparse: the vector $\mathbf{h}^{(t)}$ has many zeros
- we can reconstruct the original input $\mathbf{x}^{(t)}$ as much as possible

- More formally:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} \underbrace{\frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2}_{\text{reconstruction error}} + \lambda \underbrace{\|\mathbf{h}^{(t)}\|_1}_{\text{sparsity penalty}}$$

reconstruction error
 sparsity penalty
 reconstruction $\hat{\mathbf{x}}^{(t)}$ reconstruction vs.
 sparsity control

- \mathbf{D} is equivalent to the autoencoder output weight matrix
- however, $\mathbf{h}(\mathbf{x}^{(t)})$ is now a complicated function of $\mathbf{x}^{(t)}$
 - encoder is the minimization $\mathbf{h}(\mathbf{x}^{(t)}) = \arg \min_{\mathbf{h}^{(t)}} \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2 + \lambda \|\mathbf{h}^{(t)}\|_1$

SPARSE CODING

Topics: dictionary update (algorithm I)

- Going back to our original problem

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} l(\mathbf{x}^{(t)}) = \min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}(\mathbf{x}^{(t)})\|_2^2 + \lambda \|\mathbf{h}(\mathbf{x}^{(t)})\|_1$$

- Let's assume $\mathbf{h}(\mathbf{x}^{(t)})$ doesn't depend on \mathbf{D} (which is false)

- ▶ we must minimize

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}(\mathbf{x}^{(t)})\|_2^2$$

- ▶ we must also constrain the columns of \mathbf{D} to be of unit norm

SPARSE CODING

Topics: dictionary update (algorithm I)

- A gradient descent method could be used here too
 - ▶ specifically, this is a projected gradient descent algorithm
- While \mathbf{D} hasn't converged
 - ▶ perform gradient update of \mathbf{D}

$$\mathbf{D} \leftarrow \mathbf{D} + \alpha \frac{1}{T} \sum_{t=1}^T (\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}(\mathbf{x}^{(t)})) \mathbf{h}(\mathbf{x}^{(t)})^\top$$

- ▶ renormalize the columns of \mathbf{D}
 - for each column $\mathbf{D}_{\cdot,j}$:

$$\mathbf{D}_{\cdot,j} \leftarrow \frac{\mathbf{D}_{\cdot,j}}{\|\mathbf{D}_{\cdot,j}\|_2}$$

Neural networks

Sparse coding - dictionary update with block-coordinate descent

SPARSE CODING

Topics: dictionary update (algorithm 2)

- Going back to our original problem

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} l(\mathbf{x}^{(t)}) = \min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}(\mathbf{x}^{(t)})\|_2^2 + \lambda \|\mathbf{h}(\mathbf{x}^{(t)})\|_1$$

- Let's assume $\mathbf{h}(\mathbf{x}^{(t)})$ doesn't depend on \mathbf{D} (which is false)

- ▶ we must minimize

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}(\mathbf{x}^{(t)})\|_2^2$$

- ▶ we must also constrain the columns of \mathbf{D} to be of unit norm

SPARSE CODING

Topics: dictionary update (algorithm 2)

- An alternative is to solve for each column $\mathbf{D}_{\cdot,j}$ in cycle:

- ▶ setting the gradient for $\mathbf{D}_{\cdot,j}$ to zero, we have

$$0 = \frac{1}{T} \sum_{t=1}^T (\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}(\mathbf{x}^{(t)})) h(\mathbf{x}^{(t)})_j$$

$$0 = \sum_{t=1}^T \left(\mathbf{x}^{(t)} - \left(\sum_{i \neq j} \mathbf{D}_{\cdot,i} h(\mathbf{x}^{(t)})_i \right) - \mathbf{D}_{\cdot,j} h(\mathbf{x}^{(t)})_j \right) h(\mathbf{x}^{(t)})_j$$

$$\sum_{t=1}^T \mathbf{D}_{\cdot,j} h(\mathbf{x}^{(t)})_j^2 = \sum_{t=1}^T \left(\mathbf{x}^{(t)} - \left(\sum_{i \neq j} \mathbf{D}_{\cdot,i} h(\mathbf{x}^{(t)})_i \right) \right) h(\mathbf{x}^{(t)})_j$$

$$\mathbf{D}_{\cdot,j} = \frac{1}{\sum_{t=1}^T h(\mathbf{x}^{(t)})_j^2} \sum_{t=1}^T \left(\mathbf{x}^{(t)} - \left(\sum_{i \neq j} \mathbf{D}_{\cdot,i} h(\mathbf{x}^{(t)})_i \right) \right) h(\mathbf{x}^{(t)})_j$$

- ▶ we don't need to specify a learning rate to update $\mathbf{D}_{\cdot,j}$

SPARSE CODING

Topics: dictionary update (algorithm 2)

- An alternative is to solve for each column $\mathbf{D}_{\cdot,j}$ in cycle:

- ▶ we can rewrite

$$\begin{aligned}
 \mathbf{D}_{\cdot,j} &= \frac{1}{\sum_{t=1}^T h(\mathbf{x}^{(t)})_j^2} \sum_{t=1}^T \left(\mathbf{x}^{(t)} - \left(\sum_{i \neq j} \mathbf{D}_{\cdot,i} h(\mathbf{x}^{(t)})_i \right) \right) h(\mathbf{x}^{(t)})_j \\
 &= \underbrace{\frac{1}{\sum_{t=1}^T h(\mathbf{x}^{(t)})_j^2} \left(\underbrace{\left(\sum_{t=1}^T \mathbf{x}^{(t)} h(\mathbf{x}^{(t)})_j \right)}_{A_{j,j}} - \sum_{i \neq j} \mathbf{D}_{\cdot,i} \underbrace{\left(\sum_{t=1}^T h(\mathbf{x}^{(t)})_i h(\mathbf{x}^{(t)})_j \right)}_{A_{i,j}} \right)}_{\mathbf{B}_{\cdot,j}} \\
 &= \frac{1}{A_{j,j}} (\mathbf{B}_{\cdot,j} - \mathbf{D} \mathbf{A}_{\cdot,j} + \mathbf{D}_{\cdot,j} A_{j,j})
 \end{aligned}$$

- ▶ this way, we only need to store:

- $\mathbf{A} \Leftarrow \sum_{t=1}^T \mathbf{h}(\mathbf{x}^{(t)}) \mathbf{h}(\mathbf{x}^{(t)})^\top$
- $\mathbf{B} \Leftarrow \sum_{t=1}^T \mathbf{x}^{(t)} \mathbf{h}(\mathbf{x}^{(t)})^\top$

SPARSE CODING

Topics: dictionary update (algorithm 2)

- While \mathbf{D} hasn't converged
 - ▶ for each column $\mathbf{D}_{\cdot,j}$ perform updates
 - $\mathbf{D}_{\cdot,j} \leftarrow \frac{1}{A_{j,j}} (\mathbf{B}_{\cdot,j} - \mathbf{D} \mathbf{A}_{\cdot,j} + \mathbf{D}_{\cdot,j} A_{j,j})$
 - $\mathbf{D}_{\cdot,j} \leftarrow \frac{\mathbf{D}_{\cdot,j}}{\|\mathbf{D}_{\cdot,j}\|_2}$
- This is referred to as a block-coordinate descent algorithm
 - ▶ a different block of variables are updated at each step
 - ▶ the “blocks” are the columns $\mathbf{D}_{\cdot,j}$

Neural networks

Sparse coding - dictionary learning algorithm

SPARSE CODING

Topics: sparse coding

- For each $\mathbf{x}^{(t)}$ find a latent representation $\mathbf{h}^{(t)}$ such that:

- it is sparse: the vector $\mathbf{h}^{(t)}$ has many zeros
- we can reconstruct the original input $\mathbf{x}^{(t)}$ as much as possible

- More formally:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^T \min_{\mathbf{h}^{(t)}} \underbrace{\frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2}_{\text{reconstruction error}} + \lambda \underbrace{\|\mathbf{h}^{(t)}\|_1}_{\text{sparsity penalty}}$$

reconstruction error
 sparsity penalty
 reconstruction $\hat{\mathbf{x}}^{(t)}$ reconstruction vs.
 sparsity control

- \mathbf{D} is equivalent to the autoencoder output weight matrix
- however, $\mathbf{h}(\mathbf{x}^{(t)})$ is now a complicated function of $\mathbf{x}^{(t)}$
 - encoder is the minimization $\mathbf{h}(\mathbf{x}^{(t)}) = \arg \min_{\mathbf{h}^{(t)}} \frac{1}{2} \|\mathbf{x}^{(t)} - \mathbf{D} \mathbf{h}^{(t)}\|_2^2 + \lambda \|\mathbf{h}^{(t)}\|_1$

SPARSE CODING

Topics: learning algorithm (putting it all together)

- Learning alternates between inference and dictionary learning

- While \mathbf{D} has not converged
 - ▶ find the sparse codes $\mathbf{h}(\mathbf{x}^{(t)})$ for all $\mathbf{x}^{(t)}$ in my training set with ISTA
 - ▶ update the dictionary:
 - $\mathbf{A} \leftarrow \sum_{t=1}^T \mathbf{x}^{(t)} \mathbf{h}(\mathbf{x}^{(t)})^\top$
 - $\mathbf{B} \leftarrow \sum_{t=1}^T \mathbf{h}(\mathbf{x}^{(t)}) \mathbf{h}(\mathbf{x}^{(t)})^\top$
 - run block-coordinate descent algorithm to update \mathbf{D}

- Similar to the EM algorithm

Neural networks

Sparse coding - ZCA preprocessing

PREPROCESSING

Topics: ZCA

- Before running a sparse coding algorithm, it is beneficial to remove “obvious” structure from the data
 - ▶ normalize such that mean is 0 and covariance is the identity (whitening)
 - ▶ this will remove 1st and 2nd order statistical structure
- ZCA preprocessing
 - ▶ let the empirical mean be $\hat{\mu}$ and the empirical covariance matrix be $\hat{\Sigma} = \mathbf{U}\Lambda\mathbf{U}^\top$ (in its eigenvalue/eigenvector representation)
 - ▶ ZCA transforms each input \mathbf{x} as follows:
 - $\mathbf{x} \leftarrow \mathbf{U}\Lambda^{-\frac{1}{2}}\mathbf{U}^\top(\mathbf{x} - \hat{\mu})$

PREPROCESSING

Topics: ZCA

- After this transformation
 - ▶ the empirical mean is 0

$$\begin{aligned} & \frac{1}{T} \sum_t \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top (\mathbf{x}^{(t)} - \hat{\boldsymbol{\mu}}) \\ &= \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top \left(\left(\frac{1}{T} \sum_t \mathbf{x}^{(t)} \right) - \hat{\boldsymbol{\mu}} \right) \\ &= \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top (\hat{\boldsymbol{\mu}} - \hat{\boldsymbol{\mu}}) \\ &= 0 \end{aligned}$$

PREPROCESSING

Topics: ZCA

- After this transformation
 - ▶ the empirical covariance matrix is the identity

$$\begin{aligned}
 & \frac{1}{T-1} \sum_t \left(\mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top (\mathbf{x}^{(t)} - \hat{\boldsymbol{\mu}}) \right) \left(\mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top (\mathbf{x}^{(t)} - \hat{\boldsymbol{\mu}}) \right)^\top \\
 = & \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top \left(\frac{1}{T-1} \sum_t (\mathbf{x}^{(t)} - \hat{\boldsymbol{\mu}})(\mathbf{x}^{(t)} - \hat{\boldsymbol{\mu}})^\top \right) \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top \\
 = & \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top \hat{\Sigma} \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top \\
 = & \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top \mathbf{U} \Lambda \mathbf{U}^\top \mathbf{U} \Lambda^{-\frac{1}{2}} \mathbf{U}^\top \\
 = & \mathbf{I}
 \end{aligned}$$

Neural networks

Sparse coding - feature extraction

FEATURE EXTRACTION

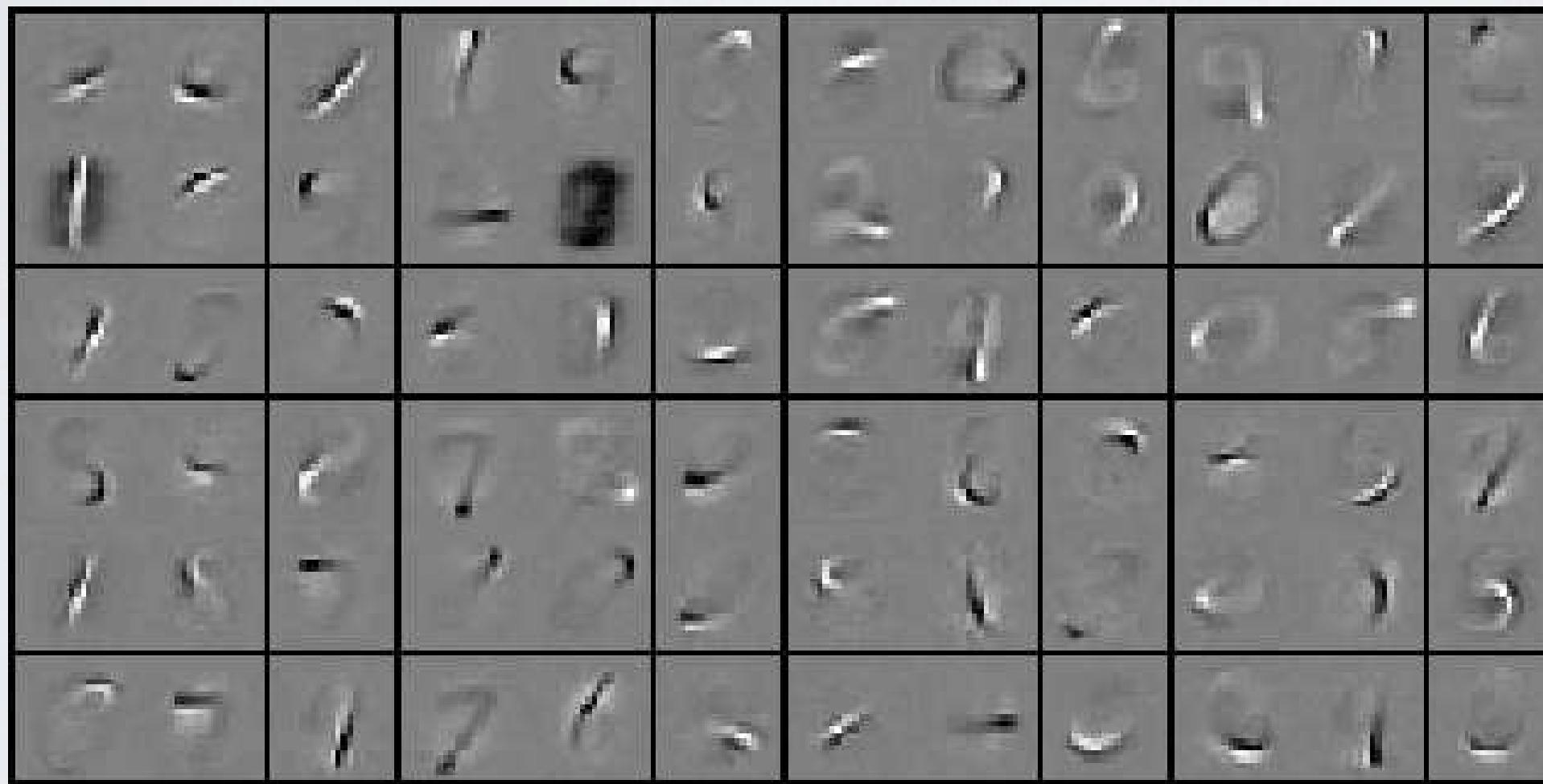
Topics: feature learning

- A sparse coding model can be used to extract features
 - ▶ given a labeled training set $\{(\mathbf{x}^{(t)}, y^{(t)})\}$
 - ▶ train sparse coding dictionary only on training inputs $\{\mathbf{x}^{(t)}\}$
 - this yields a dictionary \mathbf{D} from which to infer sparse codes $\mathbf{h}(\mathbf{x}^{(t)})$
 - ▶ train favorite classifier on transformed training set $\{(\mathbf{h}(\mathbf{x}^{(t)}), y^{(t)})\}$
- When classifying test input \mathbf{x} , must infer its sparse representation $\mathbf{h}(\mathbf{x})$ first, then feed it to the classifier

FEATURE EXTRACTION

Topics: feature learning

- When trained on handwritten digits:



FEATURE EXTRACTION

Topics: self-taught learning

- Self-taught learning:
 - ▶ when features trained on different input distribution
- Example:
 - ▶ train sparse coding dictionary on handwritten digits
 - ▶ use codes (features) to classify handwritten characters

Digits → English handwritten characters			
Training set size	Raw	PCA	Sparse coding
100	39.8%	25.3%	39.7%
500	54.8%	54.8%	58.5%
1000	61.9%	64.5%	65.3%

Neural networks

Sparse coding - relationship with VI

RELATIONSHIP WITH VI

Topics: VI neurons vs. sparse coding

- Natural image patches:
 - ▶ small image regions extracted from an image of nature (forest, grass, ...)

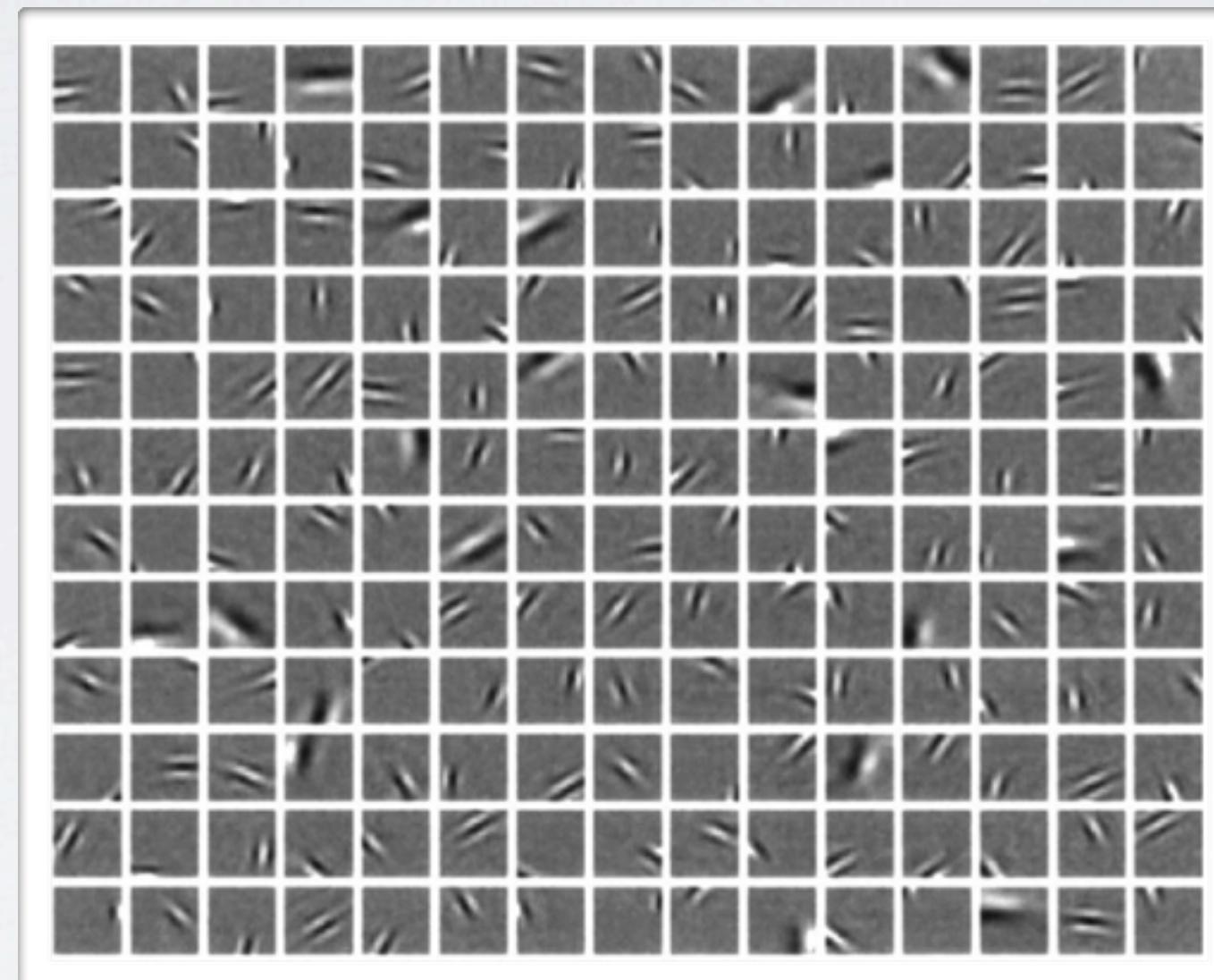


Image taken from:
Emergence of complex cell properties
by learning to generalize in natural scenes.
Karklin and Lewicki, 2009

RELATIONSHIP WITH VI

Topics: VI neurons vs. sparse coding

- When trained on natural image patches
 - ▶ the dictionary columns (“atoms”) look like edge detectors
 - ▶ each atom is tuned to a particular position, orientation and spatial frequency
 - ▶ VI neurons in the mammalian brain have a similar behavior

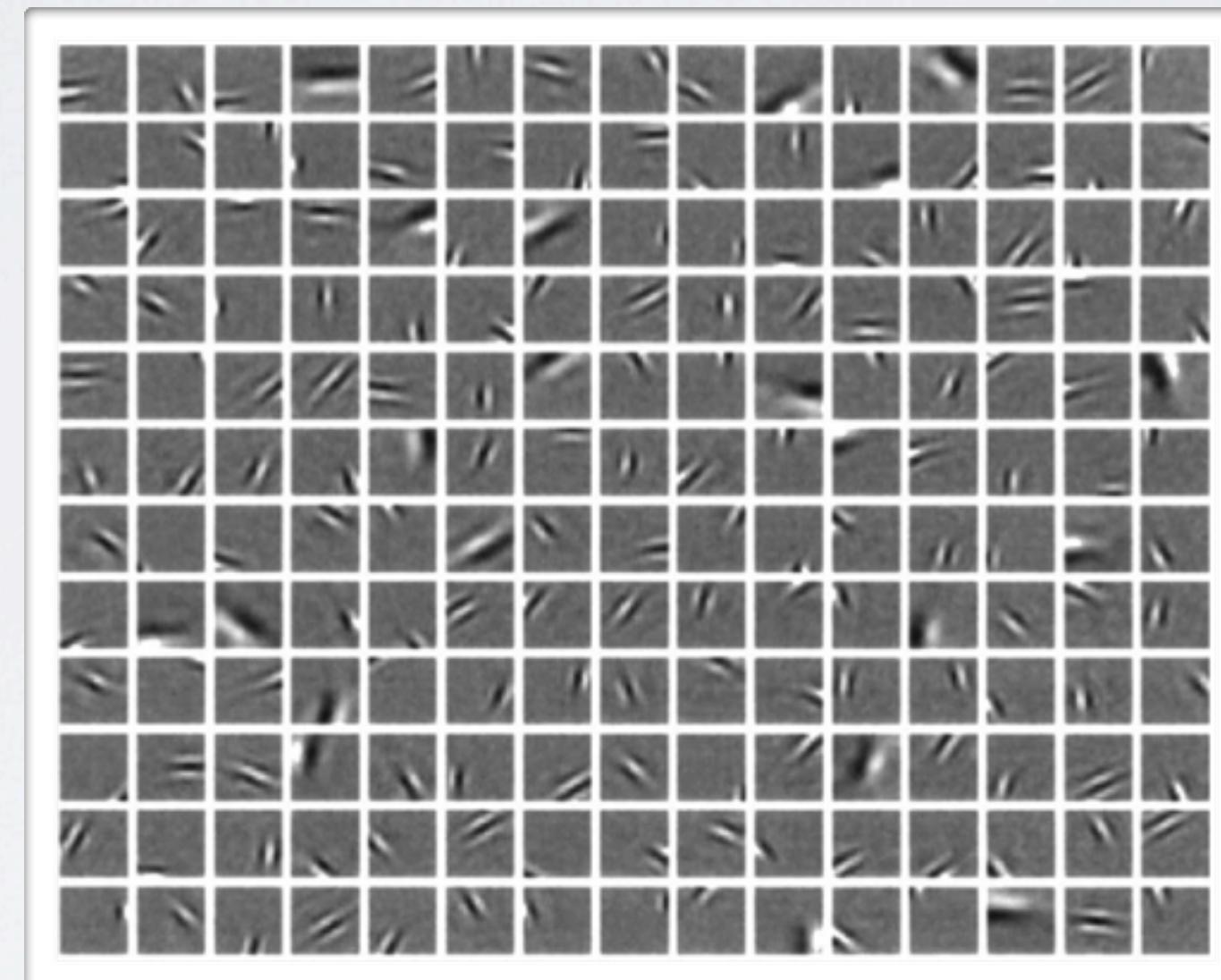


Emergence of simple-cell receptive field properties by learning a sparse code of natural images.
Olshausen and Field, 1996.

RELATIONSHIP WITH VI

Topics: VI neurons vs. sparse coding

- Suggests that the brain might be learning a sparse code of visual stimulus
- Since then, many other models have been shown to learn similar features
 - ▶ they usually all incorporate a notion of sparsity



Emergence of simple-cell receptive field properties by learning a sparse code of natural images.
Olshausen and Field, 1996.