Neural networks

Autoencoder - definition

UNSUPERVISED LEARNING

Topics: unsupervised learning

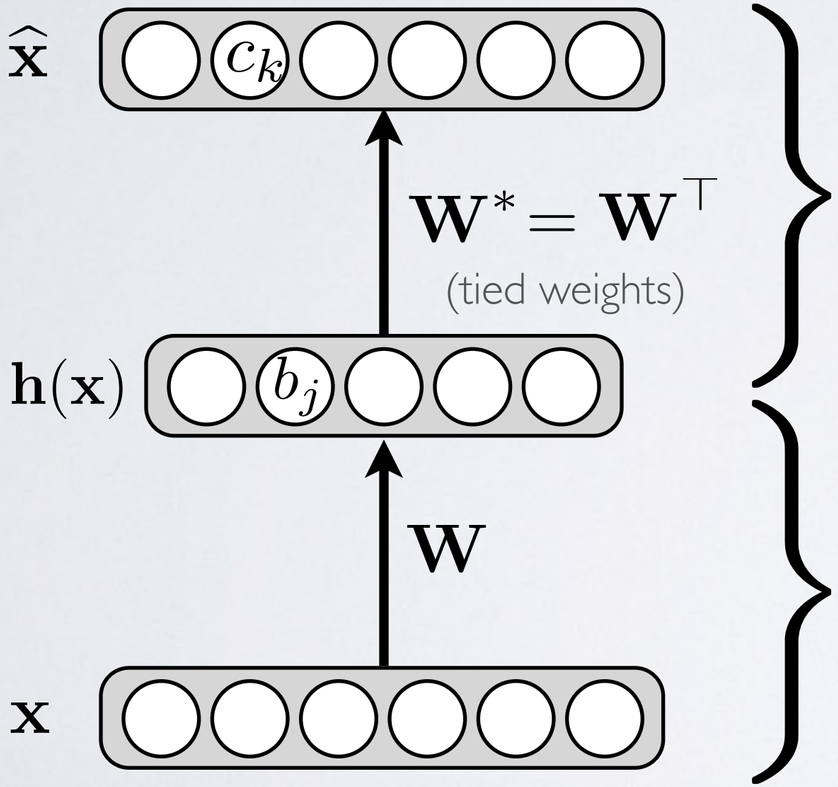
- Unsupervised learning: only use the inputs $\mathbf{x}^{(t)}$ for learning
  - automatically extract meaningful features for your data
  - leverage the availability of unlabeled data
  - add a data-dependent regularizer to trainings

- We will see a number of neural networks for unsupervised learning
  - **autoencoders / sparse coding**
  - variational autoencoders
  - generative adversarial networks
  - autoregressive models
  - restricted Boltzmann machines

AUTOENCODER

Topics: autoencoder, encoder, decoder, tied weights

- Feed-forward neural network trained to reproduce its input at the output layer



Decoder

$$\begin{aligned}\hat{\mathbf{x}} &= o(\hat{\mathbf{a}}(\mathbf{x})) \\ &= \underbrace{\mathrm{sigm}}_{\text{for binary inputs}}(\mathbf{c} + \mathbf{W}^*\mathbf{h}(\mathbf{x}))\end{aligned}$$

Encoder

$$\begin{aligned}\mathbf{h}(\mathbf{x}) &= g(\mathbf{a}(\mathbf{x})) \\ &= \mathrm{sigm}(\mathbf{b} + \mathbf{W}\mathbf{x})\end{aligned}$$

AUTOENCODER

Topics: loss function

$$f(\mathbf{x}) \equiv \widehat{\mathbf{x}}$$

- For binary inputs:

$$l(f(\mathbf{x})) = -\sum_k \left(x_k \log(\widehat{x}_k) + (1 - x_k) \log(1 - \widehat{x}_k)\right)$$

  - cross-entropy (more precisely: sum of Bernoulli cross-entropies)

- For real-valued inputs:

$$l(f(\mathbf{x})) = \frac{1}{2} \sum_k (\widehat{x}_k - x_k)^2$$

  - sum of squared differences (squared euclidean distance)
  - we use a linear activation function at the output

AUTOENCODER

Topics: loss function gradient

- For both cases, the gradient $\nabla_{\widehat{\mathbf{a}}(\mathbf{x}^{(t)})} l(f(\mathbf{x}^{(t)}))$ has a very simple form:

$$f(\mathbf{x}) \equiv \widehat{\mathbf{x}}$$

$$\nabla_{\widehat{\mathbf{a}}(\mathbf{x}^{(t)})} l(f(\mathbf{x}^{(t)})) = \widehat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}$$

- Parameter gradients are obtained by backpropagating the gradient $\nabla_{\widehat{\mathbf{a}}(\mathbf{x}^{(t)})} l(f(\mathbf{x}^{(t)}))$ like in a regular network
  - **important**: when using tied weights ($\mathbf{W}^* = \mathbf{W}^\top$), $\nabla_{\mathbf{W}} l(f(\mathbf{x}^{(t)}))$ is the sum of two gradients !
    - this is because $\mathbf{W}$ is present in the encoder **and** in the decoder

AUTOENCODER

Topics: adaptation to the type of input

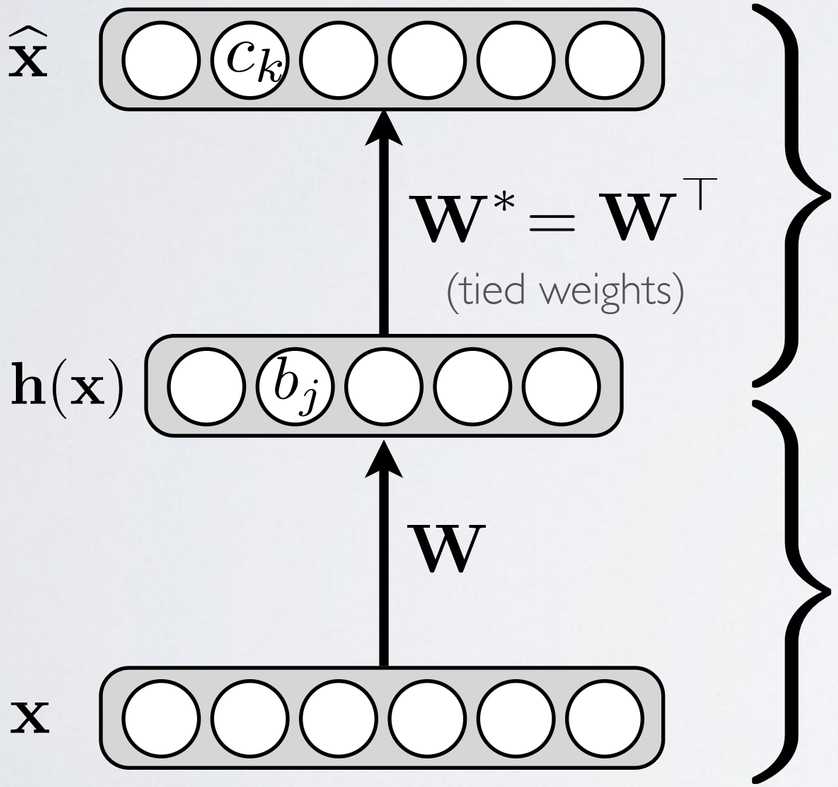
$$f(\mathbf{x}) \equiv \hat{\mathbf{x}}$$

- Recipe to adapt an autoencoder to a new type of input
  - choose a joint distribution $p(\mathbf{x}|\boldsymbol{\mu})$ over the inputs
    - $\boldsymbol{\mu}$ is the vector of parameters of that distribution
  - choose the relationship between $\boldsymbol{\mu}$ and the hidden layer $\mathbf{h}(\mathbf{x})$
  - use $l(f(\mathbf{x})) = -\log p(\mathbf{x}|\boldsymbol{\mu})$ as the loss function
- Example: we get the sum of squared distance by
  - choosing a Gaussian distribution with mean $\boldsymbol{\mu}$ and identity covariance for $p(\mathbf{x}|\boldsymbol{\mu}) = \frac{1}{(2\pi)^{D/2}} \exp(-\frac{1}{2}\sum_k (x_k - \mu_k)^2)$
  - choosing $\boldsymbol{\mu} = \mathbf{c} + \mathbf{W}^* \mathbf{h}(\mathbf{x})$

AUTOENCODER

Topics: autoencoder, encoder, decoder, tied weights

- Feed-forward neural network trained to reproduce its input at the output layer



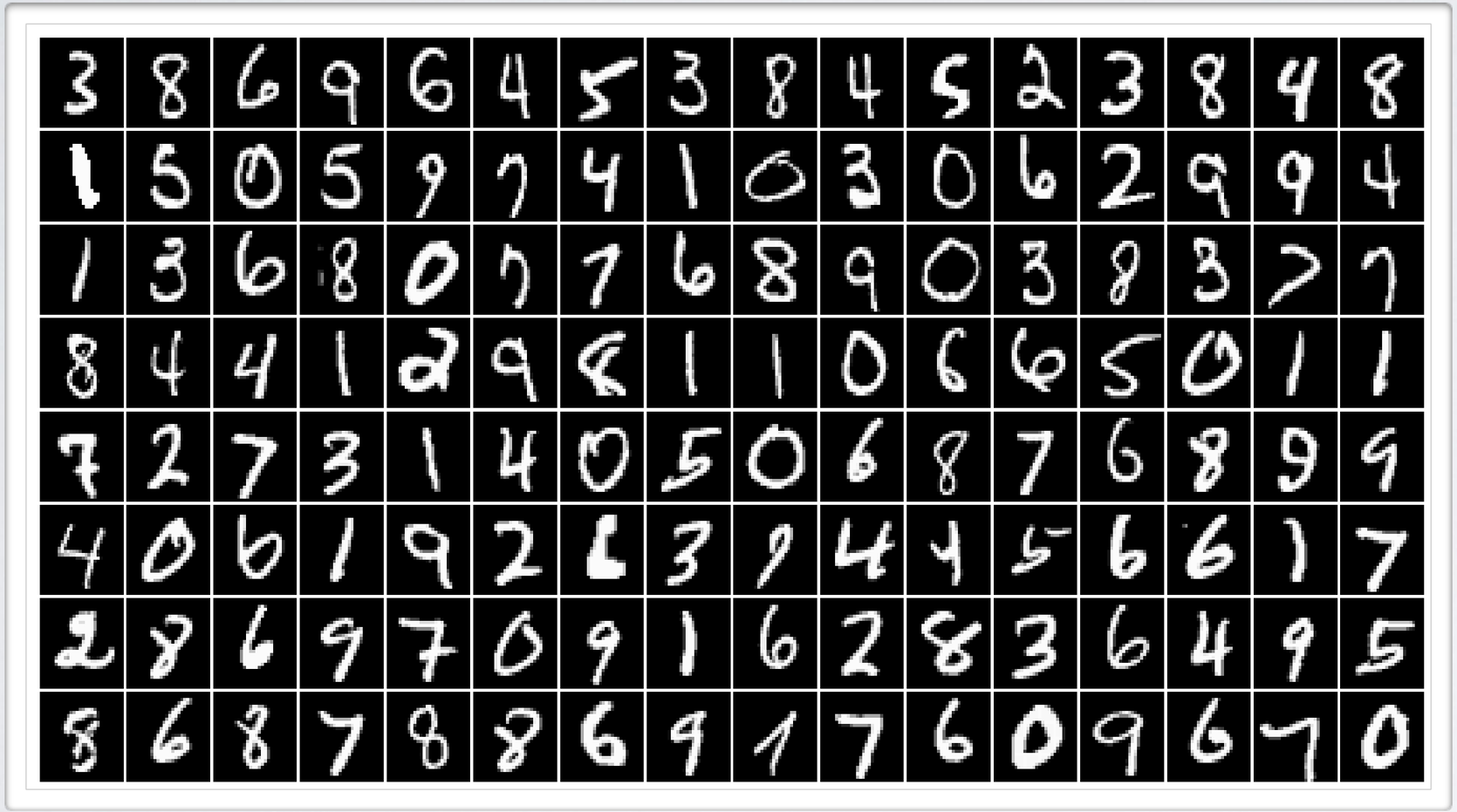
Decoder

$$\begin{aligned}\hat{\mathbf{x}} &= o(\hat{\mathbf{a}}(\mathbf{x})) \\ &= \underbrace{\mathrm{sigm}}_{\text{for binary inputs}}(\mathbf{c} + \mathbf{W}^*\mathbf{h}(\mathbf{x}))\end{aligned}$$

Encoder

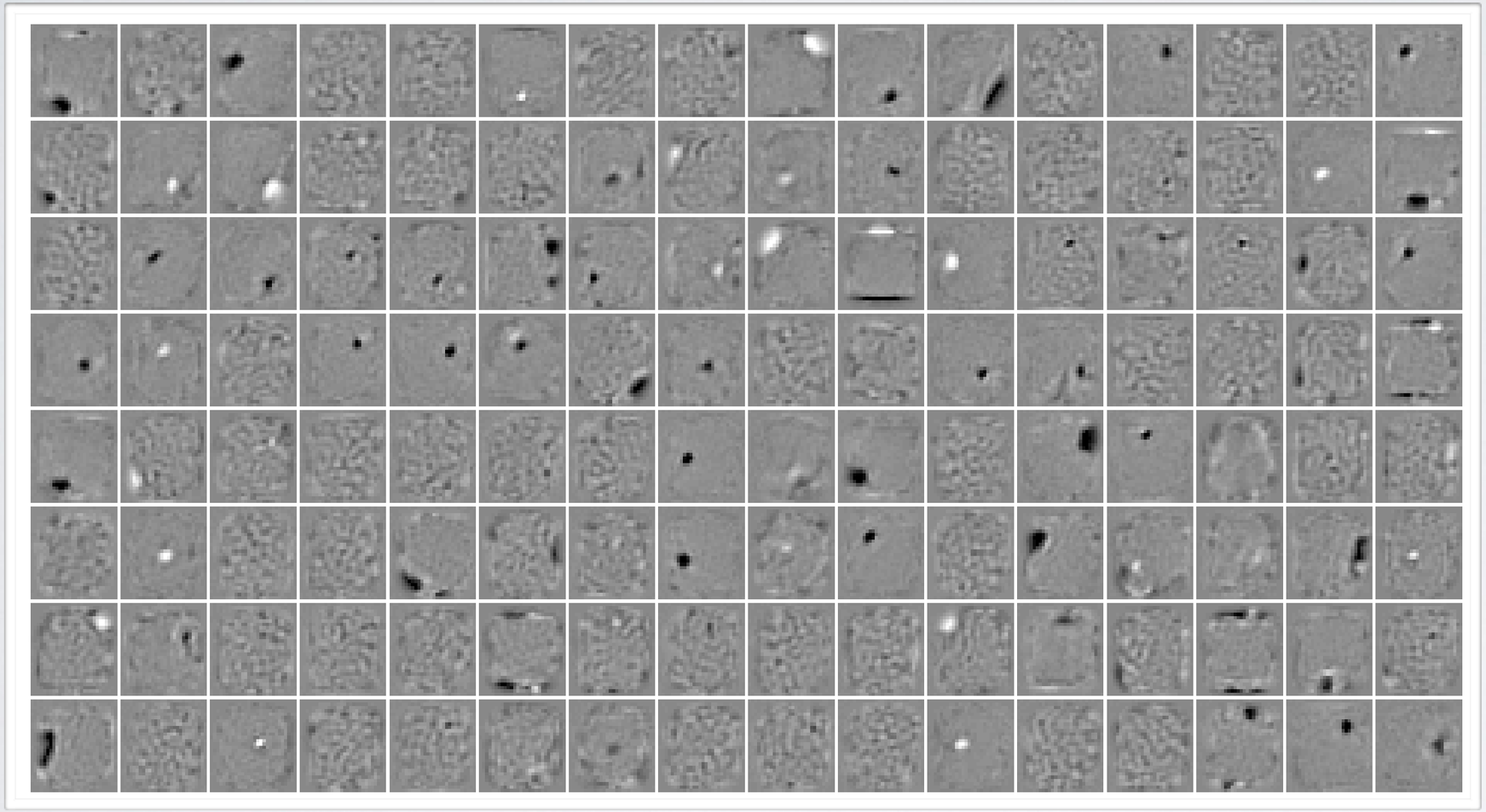
$$\begin{aligned}\mathbf{h}(\mathbf{x}) &= g(\mathbf{a}(\mathbf{x})) \\ &= \mathrm{sigm}(\mathbf{b} + \mathbf{W}\mathbf{x})\end{aligned}$$

EXAMPLE OF DATA SET: MNIST



FILTERS (AUTOENCODER)

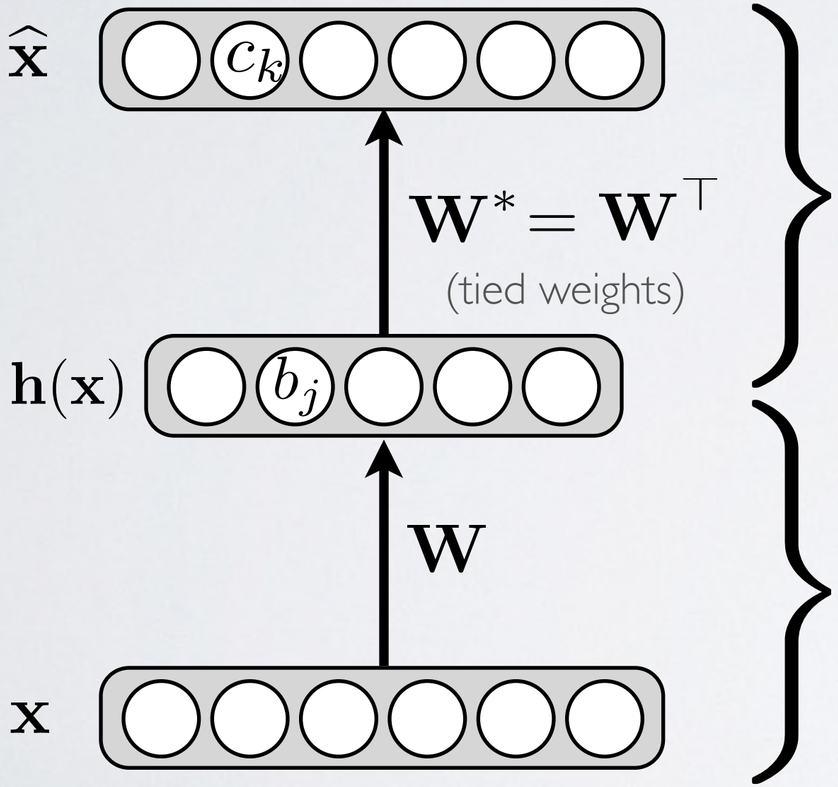
(Larochelle et al., JMLR2009)



AUTOENCODER

Topics: autoencoder, encoder, decoder, tied weights

- Feed-forward neural network trained to reproduce its input at the output layer



Decoder

$$\begin{aligned}\hat{\mathbf{x}} &= o(\hat{\mathbf{a}}(\mathbf{x})) \\ &= \underbrace{\mathrm{sigm}}_{\text{for binary inputs}}(\mathbf{c} + \mathbf{W}^*\mathbf{h}(\mathbf{x}))\end{aligned}$$

Encoder

$$\begin{aligned}\mathbf{h}(\mathbf{x}) &= g(\mathbf{a}(\mathbf{x})) \\ &= \mathrm{sigm}(\mathbf{b} + \mathbf{W}\mathbf{x})\end{aligned}$$

AUTOENCODER

Topics: optimality of a linear autoencoder

- To do the proof, we need the following theorem:
  - let $\mathbf{A}$ be any matrix, with singular value decomposition $\mathbf{A} = \mathbf{U}\,\Sigma\,\mathbf{V}^\top$
    - Σ is a diagonal matrix
    - $\mathbf{U}, \mathbf{V}$ are orthonormal matrices (columns/rows are orthonormal vectors)
  - let $\mathbf{U}_{\cdot,\leq k}\,\Sigma_{\leq k,\leq k}\,\mathbf{V}^\top_{\cdot,\leq k}$ be the decomposition where we keep only the k largest singular values
  - then, the matrix $\mathbf{B}$ of rank k that is closest to $\mathbf{A}$:

$$\mathbf{B}^* = \underset{\mathbf{B}\text{ s.t. }\mathrm{rank}(\mathbf{B})=k}{\arg\min} ||\mathbf{A} - \mathbf{B}||_F$$

  is $\mathbf{B}^* = \mathbf{U}_{\cdot,\leq k}\,\Sigma_{\leq k,\leq k}\,\mathbf{V}^\top_{\cdot,\leq k}$

$$\min_{\theta} \sum_{t} \frac{1}{2} \sum_{i} (x_i^{(t)} - \widehat{x}_i^{(t)})^2$$

Sketch of proof

$$\min_{\theta} \sum_{t} \frac{1}{2} \sum_{i} (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2$$

Sketch of proof

$$\min_{\theta} \sum_{t} \frac{1}{2} \sum_{i} (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2$$

Sketch of proof

matrix where columns are $\mathbf{x}^{(t)}$

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2$$

Sketch of proof

matrix where columns are $\mathbf{x}^{(t)}$

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

Sketch of proof

matrix where columns are $\mathbf{x}^{(t)}$

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix of all hidden layers (could be any encoder)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2$$

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k} \, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U} \, \Sigma \, \mathbf{V}^\top$ and k is the hidden layer size

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k}\, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U}\, \Sigma\, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k} \, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U} \, \Sigma \, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\mathbf{h}(\mathbf{X}) \quad = \quad \mathbf{V}_{\cdot, \leq k}^\top$$

Sketch of proof

matrix where columns are $\mathbf{x}^{(t)}$

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix of all hidden layers (could be any encoder)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k} \, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U} \, \Sigma \, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned} \mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\ &= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{X}^\top \, \mathbf{X})^{-1} \, (\mathbf{X}^\top \, \mathbf{X}) \end{aligned}$$

← multiplying by identity

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k} \, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U} \, \Sigma \, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned}
\mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{X}^\top \, \mathbf{X})^{-1} \, (\mathbf{X}^\top \, \mathbf{X}) && \leftarrow \text{multiplying by identity} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{U} \, \Sigma \, \mathbf{V}^\top)^{-1} \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X}) && \leftarrow \text{replace with SVD}
\end{aligned}$$

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k} \, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U} \, \Sigma \, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned}
\mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{X}^\top \, \mathbf{X})^{-1} \, (\mathbf{X}^\top \, \mathbf{X}) && \leftarrow \text{multiplying by identity} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{U} \, \Sigma \, \mathbf{V}^\top)^{-1} \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X}) && \leftarrow \text{replace with SVD} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \Sigma \, \mathbf{V}^\top)^{-1} \, \mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{U}^\top \, \mathbf{U} = \mathbf{I} \text{ (orthonormal)}
\end{aligned}$$

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k} \, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U} \, \Sigma \, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned}
\mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{X}^\top \, \mathbf{X})^{-1} \, (\mathbf{X}^\top \, \mathbf{X}) && \leftarrow \text{multiplying by identity} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{U} \, \Sigma \, \mathbf{V}^\top)^{-1} \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X}) && \leftarrow \text{replace with SVD} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \Sigma \, \mathbf{V}^\top)^{-1} \, \mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{U}^\top \, \mathbf{U} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, \mathbf{V} \, (\Sigma^\top \, \Sigma)^{-1} \, \mathbf{V}^\top \, \mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{V}(\Sigma^\top\Sigma)^{-1}\mathbf{V}^\top \, \mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top = \mathbf{I}
\end{aligned}$$

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k}\, \Sigma_{\leq k, \leq k},\ \ \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U}\, \Sigma\, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned}
\mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, (\mathbf{X}^\top\, \mathbf{X})^{-1}\, (\mathbf{X}^\top\, \mathbf{X}) && \leftarrow \text{multiplying by identity} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, (\mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{U}\, \Sigma\, \mathbf{V}^\top)^{-1}\, (\mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X}) && \leftarrow \text{replace with SVD} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, (\mathbf{V}\, \Sigma^\top\, \Sigma\, \mathbf{V}^\top)^{-1}\, \mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \mathbf{U}^\top\, \mathbf{U} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, \mathbf{V}\, (\Sigma^\top\, \Sigma)^{-1}\, \mathbf{V}^\top\, \mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \mathbf{V}(\Sigma^\top\Sigma)^{-1}\mathbf{V}^\top\, \mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top = \mathbf{I} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, \mathbf{V}\, (\Sigma^\top\, \Sigma)^{-1}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \mathbf{V}^\top\, \mathbf{V} = \mathbf{I} \text{ (orthonormal)}
\end{aligned}$$

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k} \, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U} \, \Sigma \, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned}
\mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{X}^\top \, \mathbf{X})^{-1} \, (\mathbf{X}^\top \, \mathbf{X}) && \leftarrow \text{multiplying by identity} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{U} \, \Sigma \, \mathbf{V}^\top)^{-1} \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X}) && \leftarrow \text{replace with SVD} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \Sigma \, \mathbf{V}^\top)^{-1} \, \mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{U}^\top \, \mathbf{U} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, \mathbf{V} \, (\Sigma^\top \, \Sigma)^{-1} \, \mathbf{V}^\top \, \mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{V}(\Sigma^\top\Sigma)^{-1}\mathbf{V}^\top \, \mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top = \mathbf{I} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, \mathbf{V} \, (\Sigma^\top \, \Sigma)^{-1} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{V}^\top \, \mathbf{V} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{I}_{\leq k, \cdot} \, (\Sigma^\top \, \Sigma)^{-1} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \text{idem}
\end{aligned}$$

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k} \, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U} \, \Sigma \, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned}
\mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{X}^\top \, \mathbf{X})^{-1} \, (\mathbf{X}^\top \, \mathbf{X}) && \leftarrow \text{multiplying by identity} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{U} \, \Sigma \, \mathbf{V}^\top)^{-1} \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X}) && \leftarrow \text{replace with SVD} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \Sigma \, \mathbf{V}^\top)^{-1} \, \mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{U}^\top \, \mathbf{U} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, \mathbf{V} \, (\Sigma^\top \, \Sigma)^{-1} \, \mathbf{V}^\top \, \mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{V}(\Sigma^\top\Sigma)^{-1}\mathbf{V}^\top \, \mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top = \mathbf{I} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, \mathbf{V} \, (\Sigma^\top \, \Sigma)^{-1} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{V}^\top \, \mathbf{V} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{I}_{\leq k, \cdot} \, (\Sigma^\top \, \Sigma)^{-1} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \text{idem} \\
&= \mathbf{I}_{\leq k, \cdot} \, \Sigma^{-1} \, (\Sigma^\top)^{-1} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow (\Sigma^\top\Sigma)^{-1} = \Sigma^{-1}(\Sigma^\top)^{-1}
\end{aligned}$$

Sketch of proof

$$\min_\theta \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k}\, \Sigma_{\leq k, \leq k},\ \ \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U}\, \Sigma\, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned}
\mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, (\mathbf{X}^\top\, \mathbf{X})^{-1}\, (\mathbf{X}^\top\, \mathbf{X}) && \leftarrow \text{multiplying by identity} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, (\mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{U}\, \Sigma\, \mathbf{V}^\top)^{-1}\, (\mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X}) && \leftarrow \text{replace with SVD} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, (\mathbf{V}\, \Sigma^\top\, \Sigma\, \mathbf{V}^\top)^{-1}\, \mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \mathbf{U}^\top\, \mathbf{U} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, \mathbf{V}\, (\Sigma^\top\, \Sigma)^{-1}\, \mathbf{V}^\top\, \mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \mathbf{V}(\Sigma^\top\Sigma)^{-1}\mathbf{V}^\top\, \mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top = \mathbf{I} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, \mathbf{V}\, (\Sigma^\top\, \Sigma)^{-1}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \mathbf{V}^\top\, \mathbf{V} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{I}_{\leq k, \cdot}\, (\Sigma^\top\, \Sigma)^{-1}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \text{idem} \\
&= \mathbf{I}_{\leq k, \cdot}\, \Sigma^{-1}\, (\Sigma^\top)^{-1}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow (\Sigma^\top\Sigma)^{-1} = \Sigma^{-1}(\Sigma^\top)^{-1} \\
&= \mathbf{I}_{< k, \cdot}\, \Sigma^{-1}\, \mathbf{U}^\top\, \mathbf{X}
\end{aligned}$$

Sketch of proof

$$\min_{\theta} \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k} \, \Sigma_{\leq k, \leq k}, \;\; \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U} \, \Sigma \, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned}
\mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{X}^\top \, \mathbf{X})^{-1} \, (\mathbf{X}^\top \, \mathbf{X}) && \leftarrow \text{multiplying by identity} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{U} \, \Sigma \, \mathbf{V}^\top)^{-1} \, (\mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X}) && \leftarrow \text{replace with SVD} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, (\mathbf{V} \, \Sigma^\top \, \Sigma \, \mathbf{V}^\top)^{-1} \, \mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{U}^\top \, \mathbf{U} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, \mathbf{V} \, (\Sigma^\top \, \Sigma)^{-1} \, \mathbf{V}^\top \, \mathbf{V} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{V}(\Sigma^\top\Sigma)^{-1}\mathbf{V}^\top \, \mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top = \mathbf{I} \\
&= \mathbf{V}_{\cdot, \leq k}^\top \, \mathbf{V} \, (\Sigma^\top \, \Sigma)^{-1} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \mathbf{V}^\top \, \mathbf{V} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{I}_{\leq k, \cdot} \, (\Sigma^\top \, \Sigma)^{-1} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow \text{idem} \\
&= \mathbf{I}_{\leq k, \cdot} \, \Sigma^{-1} \, (\Sigma^\top)^{-1} \, \Sigma^\top \, \mathbf{U}^\top \, \mathbf{X} && \leftarrow (\Sigma^\top\Sigma)^{-1} = \Sigma^{-1}(\Sigma^\top)^{-1} \\
&= \mathbf{I}_{\leq k, \cdot} \, \Sigma^{-1} \, \mathbf{U}^\top \, \mathbf{X} \\
&= \Sigma_{\leq k, \leq k}^{-1} \, (\mathbf{U}_{\cdot, \leq k})^\top \, \mathbf{X} && \leftarrow \text{multiplying by } \mathbf{I}_{\leq k, \cdot} \text{ selects the } k \text{ first rows}
\end{aligned}$$

Sketch of proof

$$\min_\theta \sum_t \frac{1}{2} \sum_i (x_i^{(t)} - \underbrace{\widehat{x}_i^{(t)}}_{\text{based on linear decoder}})^2 \quad \geq \quad \min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\overbrace{\mathbf{X}}^{\text{matrix where columns are } \mathbf{x}^{(t)}} - \mathbf{W}^* \underbrace{\mathbf{h}(\mathbf{X})}_{\text{matrix of all hidden layers (could be any encoder)}}||_F^2$$

$$\arg\min_{\mathbf{W}^*, \mathbf{h}(\mathbf{X})} \frac{1}{2} ||\mathbf{X} - \mathbf{W}^* \mathbf{h}(\mathbf{X})||_F^2 \quad = \quad \left(\mathbf{W}^* \leftarrow \mathbf{U}_{\cdot, \leq k}\, \Sigma_{\leq k, \leq k},\ \ \mathbf{h}(\mathbf{X}) \leftarrow \mathbf{V}_{\cdot, \leq k}^\top\right)$$

based on previous theorem, where $\mathbf{X} = \mathbf{U}\, \Sigma\, \mathbf{V}^\top$ and k is the hidden layer size

Let's show $\mathbf{h}(\mathbf{X})$ is a linear encoder:

$$\begin{aligned}
\mathbf{h}(\mathbf{X}) &= \mathbf{V}_{\cdot, \leq k}^\top \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, (\mathbf{X}^\top\, \mathbf{X})^{-1}\, (\mathbf{X}^\top\, \mathbf{X}) && \leftarrow \text{multiplying by identity} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, (\mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{U}\, \Sigma\, \mathbf{V}^\top)^{-1}\, (\mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X}) && \leftarrow \text{replace with SVD} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, (\mathbf{V}\, \Sigma^\top\, \Sigma\, \mathbf{V}^\top)^{-1}\, \mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \mathbf{U}^\top\, \mathbf{U} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, \mathbf{V}\, (\Sigma^\top\, \Sigma)^{-1}\, \mathbf{V}^\top\, \mathbf{V}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \mathbf{V}(\Sigma^\top\Sigma)^{-1}\mathbf{V}^\top\, \mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top = \mathbf{I} \\
&= \mathbf{V}_{\cdot, \leq k}^\top\, \mathbf{V}\, (\Sigma^\top\, \Sigma)^{-1}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \mathbf{V}^\top\, \mathbf{V} = \mathbf{I} \text{ (orthonormal)} \\
&= \mathbf{I}_{\leq k, \cdot}\, (\Sigma^\top\, \Sigma)^{-1}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow \text{idem} \\
&= \mathbf{I}_{\leq k, \cdot}\, \Sigma^{-1}\, (\Sigma^\top)^{-1}\, \Sigma^\top\, \mathbf{U}^\top\, \mathbf{X} && \leftarrow (\Sigma^\top\Sigma)^{-1} = \Sigma^{-1}(\Sigma^\top)^{-1} \\
&= \mathbf{I}_{\leq k, \cdot}\, \Sigma^{-1}\, \mathbf{U}^\top\, \mathbf{X} \\
&= \underbrace{\Sigma_{\leq k, \leq k}^{-1}\, (\mathbf{U}_{\cdot, \leq k})^\top}_{\text{this is a linear encoder}}\, \mathbf{X} && \leftarrow \text{multiplying by } \mathbf{I}_{\leq k, \cdot} \text{ selects the } k \text{ first rows}
\end{aligned}$$

AUTOENCODER

Topics: optimality of a linear autoencoder

- So an optimal pair of encoder and decoder is

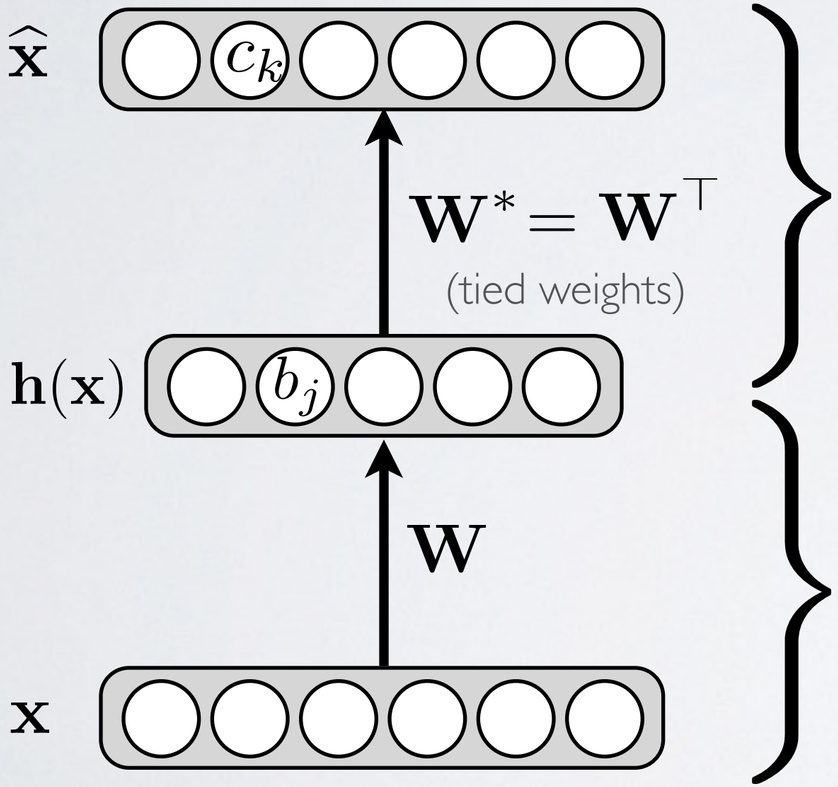
$$\mathbf{h}(\mathbf{x}) = \underbrace{\left(\Sigma^{-1}_{\leq k,\leq k} \left(\mathbf{U}_{\cdot,\leq k}\right)^\top\right)}_{\mathbf{W}} \mathbf{x} \qquad \hat{\mathbf{x}} = \underbrace{\left(\mathbf{U}_{\cdot,\leq k}\, \Sigma_{\leq k,\leq k}\right)}_{\mathbf{W}^*} \mathbf{h}(\mathbf{x})$$

  - for the sum of squared difference error
  - for an autoencoder with a linear decoder
  - where optimality means "has the lowest training reconstruction error"
- If inputs are normalized as follows: $\mathbf{x}^{(t)} \leftarrow \frac{1}{\sqrt{T}} \left(\mathbf{x}^{(t)} - \frac{1}{T} \sum_{t'=1}^{T} \mathbf{x}^{(t')}\right)$
  - encoder corresponds to Principal Component Analysis (PCA)
    - singular values and (left) vectors = the eigenvalues/vectors of covariance matrix

AUTOENCODER

Topics: autoencoder, encoder, decoder, tied weights

- Feed-forward neural network trained to reproduce its input at the output layer



Decoder

$$\begin{aligned}\hat{\mathbf{x}} &= o(\hat{\mathbf{a}}(\mathbf{x})) \\ &= \underbrace{\mathrm{sigm}}_{\text{for binary inputs}}(\mathbf{c} + \mathbf{W}^*\mathbf{h}(\mathbf{x}))\end{aligned}$$

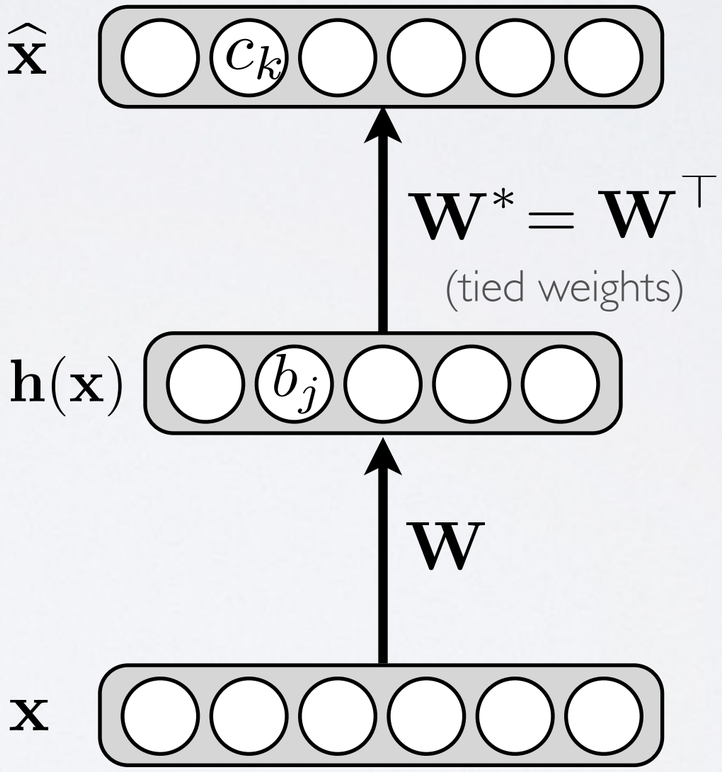
Encoder

$$\begin{aligned}\mathbf{h}(\mathbf{x}) &= g(\mathbf{a}(\mathbf{x})) \\ &= \mathrm{sigm}(\mathbf{b} + \mathbf{W}\mathbf{x})\end{aligned}$$

UNDERCOMPLETE HIDDEN LAYER

Topics: undercomplete representation

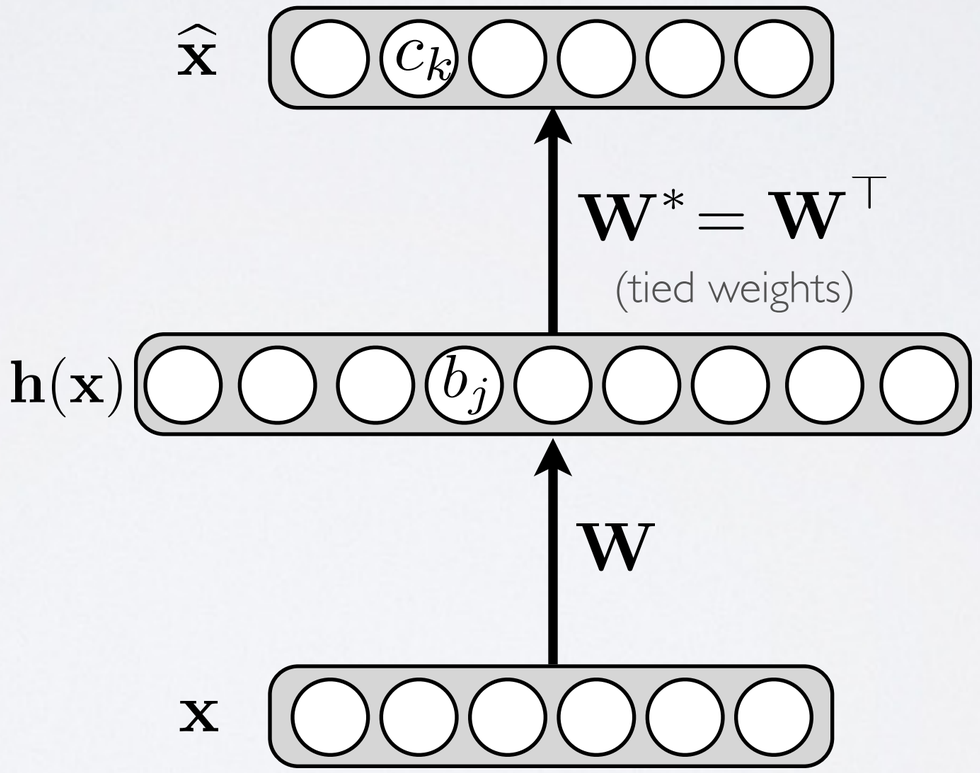
- Hidden layer is undercomplete if smaller than the input layer
  - hidden layer "compresses" the input
  - will compress well only for the training distribution
- Hidden units will be
  - good features for the training distribution
  - but bad for other types of input



OVERCOMPLETE HIDDEN LAYER

Topics: overcomplete representation

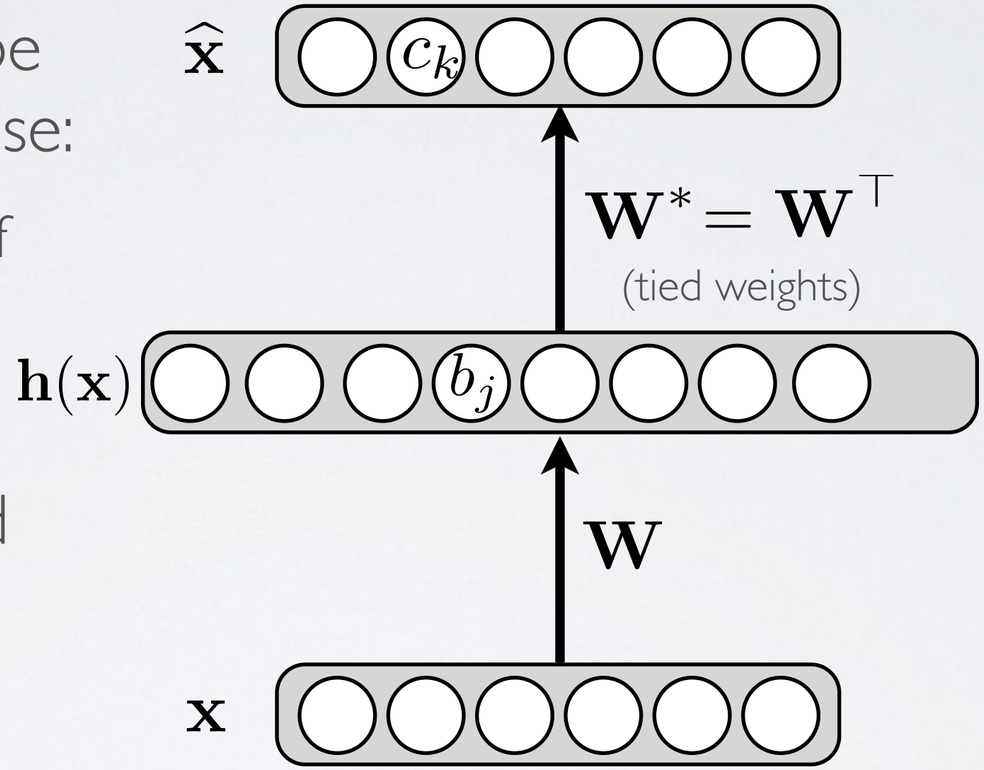
- Hidden layer is overcomplete if greater than the input layer
  - no compression in hidden layer
  - each hidden unit could copy a different input component
- No guarantee that the hidden units will extract meaningful structure



DENOISING AUTOENCODER

Topics: denoising autoencoder

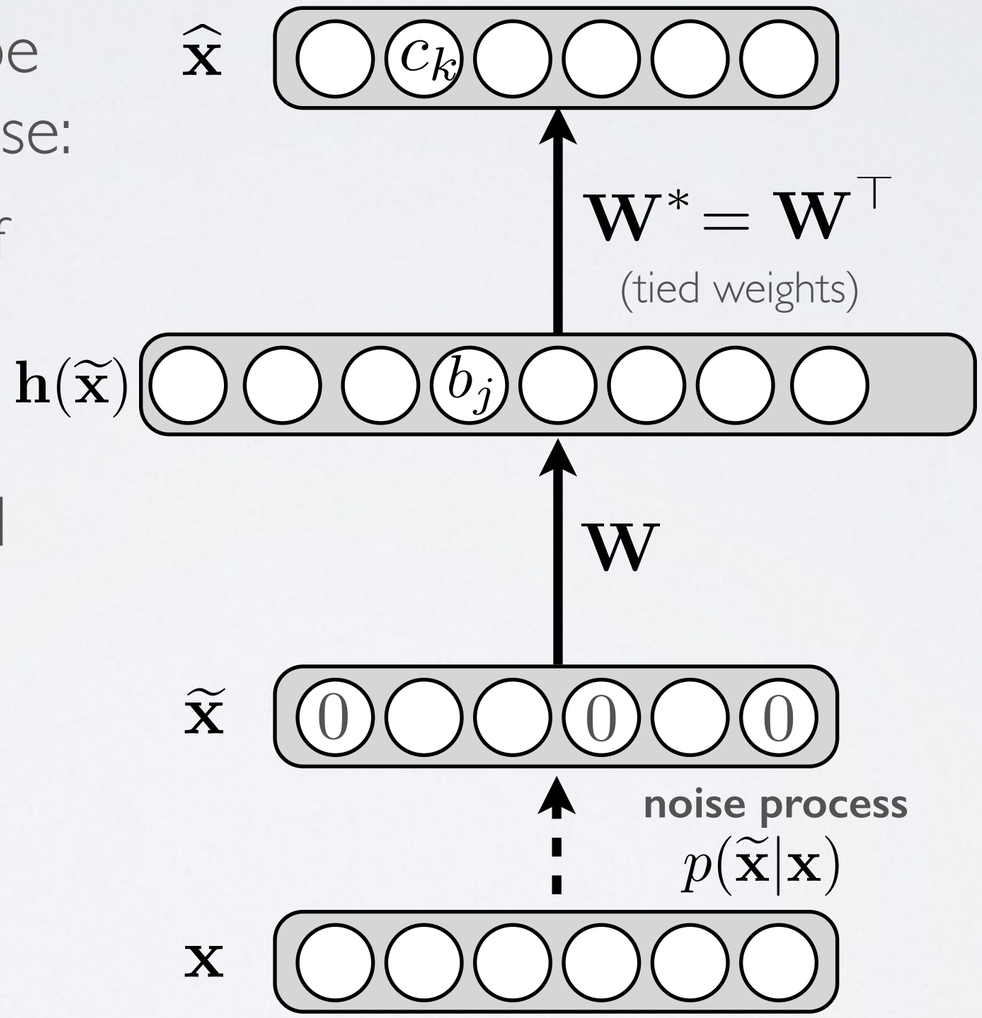
- Idea: representation should be robust to introduction of noise:
  - random assignment of subset of inputs to 0, with probability ν
  - Gaussian additive noise
- Reconstruction $\widehat{\mathbf{x}}$ computed from the corrupted input $\widetilde{\mathbf{x}}$
- Loss function compares $\widehat{\mathbf{x}}$ reconstruction with the **noiseless input** $\mathbf{x}$



DENOISING AUTOENCODER

Topics: denoising autoencoder

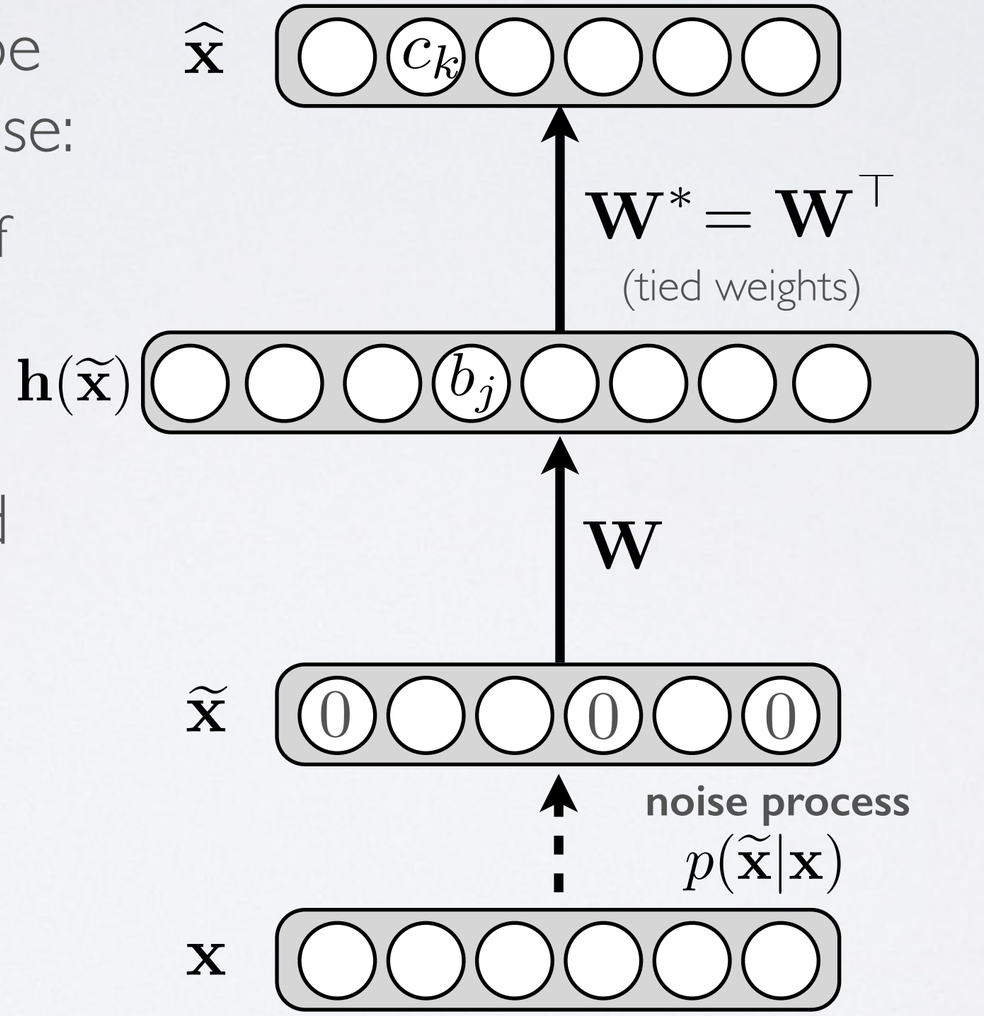
- Idea: representation should be robust to introduction of noise:
  - random assignment of subset of inputs to 0, with probability ν
  - Gaussian additive noise
- Reconstruction $\widehat{\mathbf{x}}$ computed from the corrupted input $\widetilde{\mathbf{x}}$
- Loss function compares $\widehat{\mathbf{x}}$ reconstruction with the **noiseless input** $\mathbf{x}$



DENOISING AUTOENCODER

Topics: denoising autoencoder

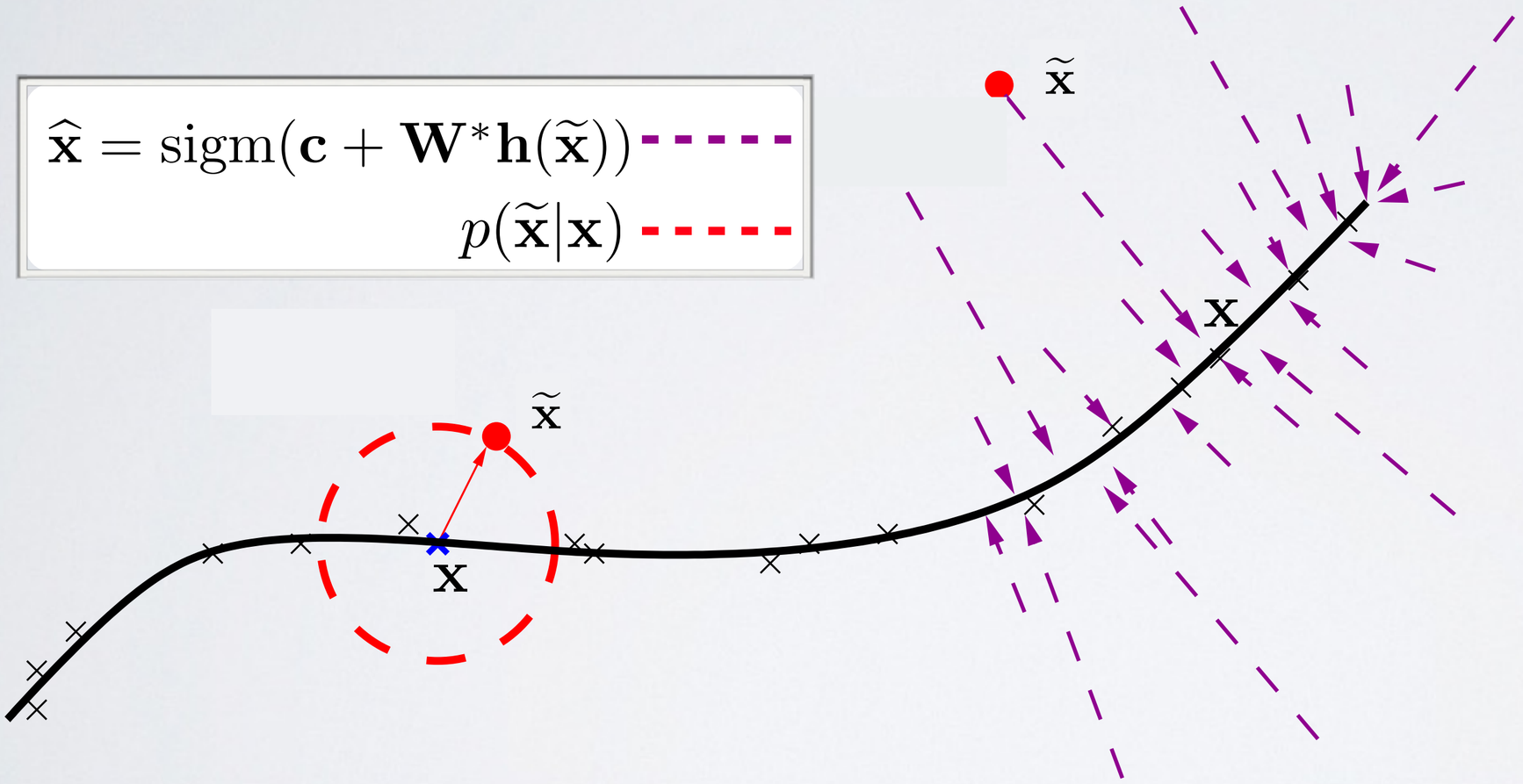
- Idea: representation should be robust to introduction of noise:
  - random assignment of subset of inputs to 0, with probability ν
  - Gaussian additive noise
- Reconstruction $\widehat{\mathbf{x}}$ computed from the corrupted input $\widetilde{\mathbf{x}}$
- Loss function compares $\widehat{\mathbf{x}}$ reconstruction with the **noiseless input** $\mathbf{x}$



DENOISING AUTOENCODER

$$\widehat{\mathbf{x}} = \mathrm{sigm}(\mathbf{c} + \mathbf{W}^{*}\mathbf{h}(\widetilde{\mathbf{x}}))$$

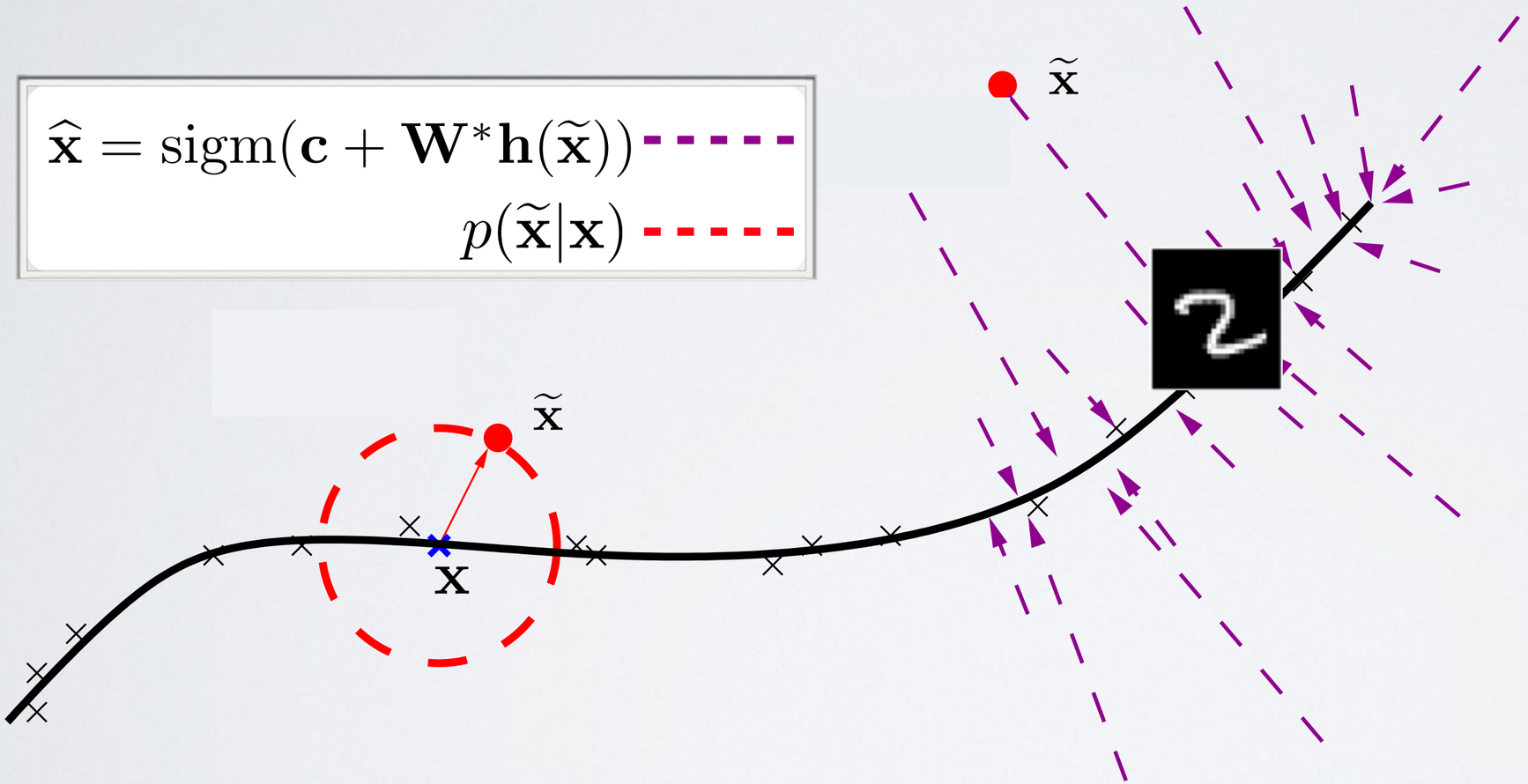
$$p(\widetilde{\mathbf{x}}|\mathbf{x})$$



DENOISING AUTOENCODER

$$\widehat{\mathbf{x}} = \text{sigm}(\mathbf{c} + \mathbf{W}^* \mathbf{h}(\widetilde{\mathbf{x}}))$$

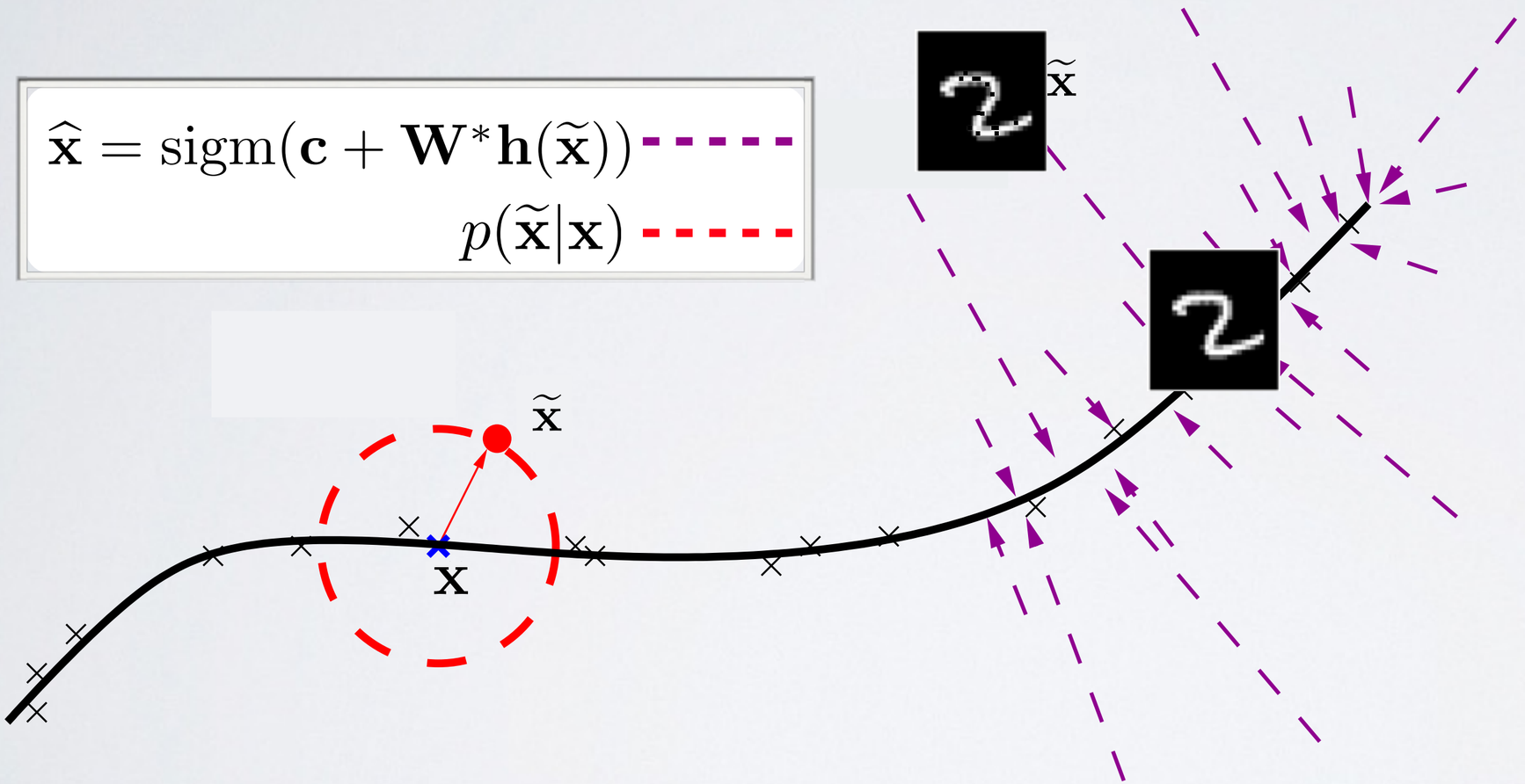
$$p(\widetilde{\mathbf{x}}|\mathbf{x})$$



DENOISING AUTOENCODER

$$\widehat{\mathbf{x}} = \text{sigm}(\mathbf{c} + \mathbf{W}^* \mathbf{h}(\widetilde{\mathbf{x}}))$$

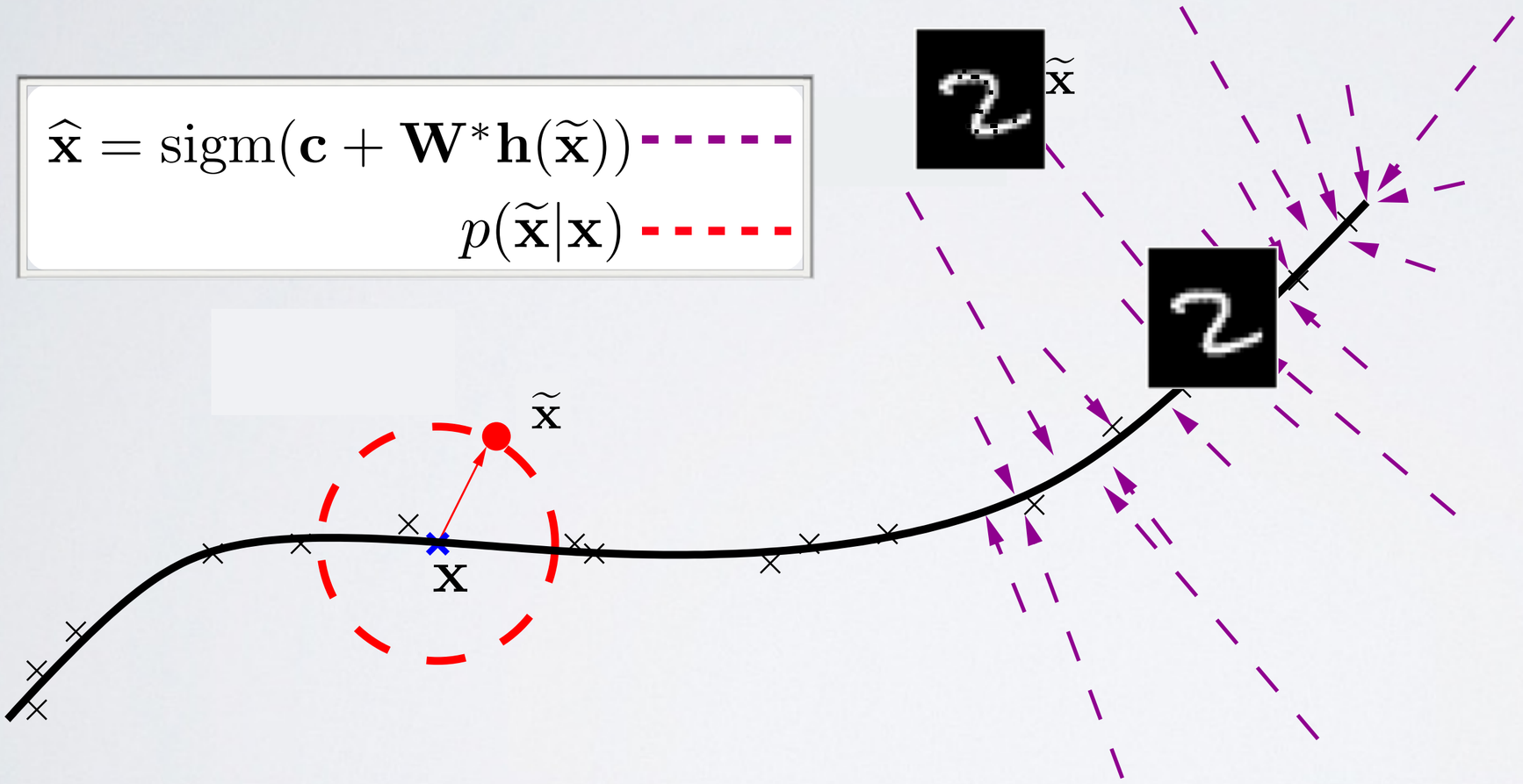
$$p(\widetilde{\mathbf{x}}|\mathbf{x})$$



DENOISING AUTOENCODER

$$\widehat{\mathbf{x}} = \text{sigm}(\mathbf{c} + \mathbf{W}^* \mathbf{h}(\widetilde{\mathbf{x}}))$$

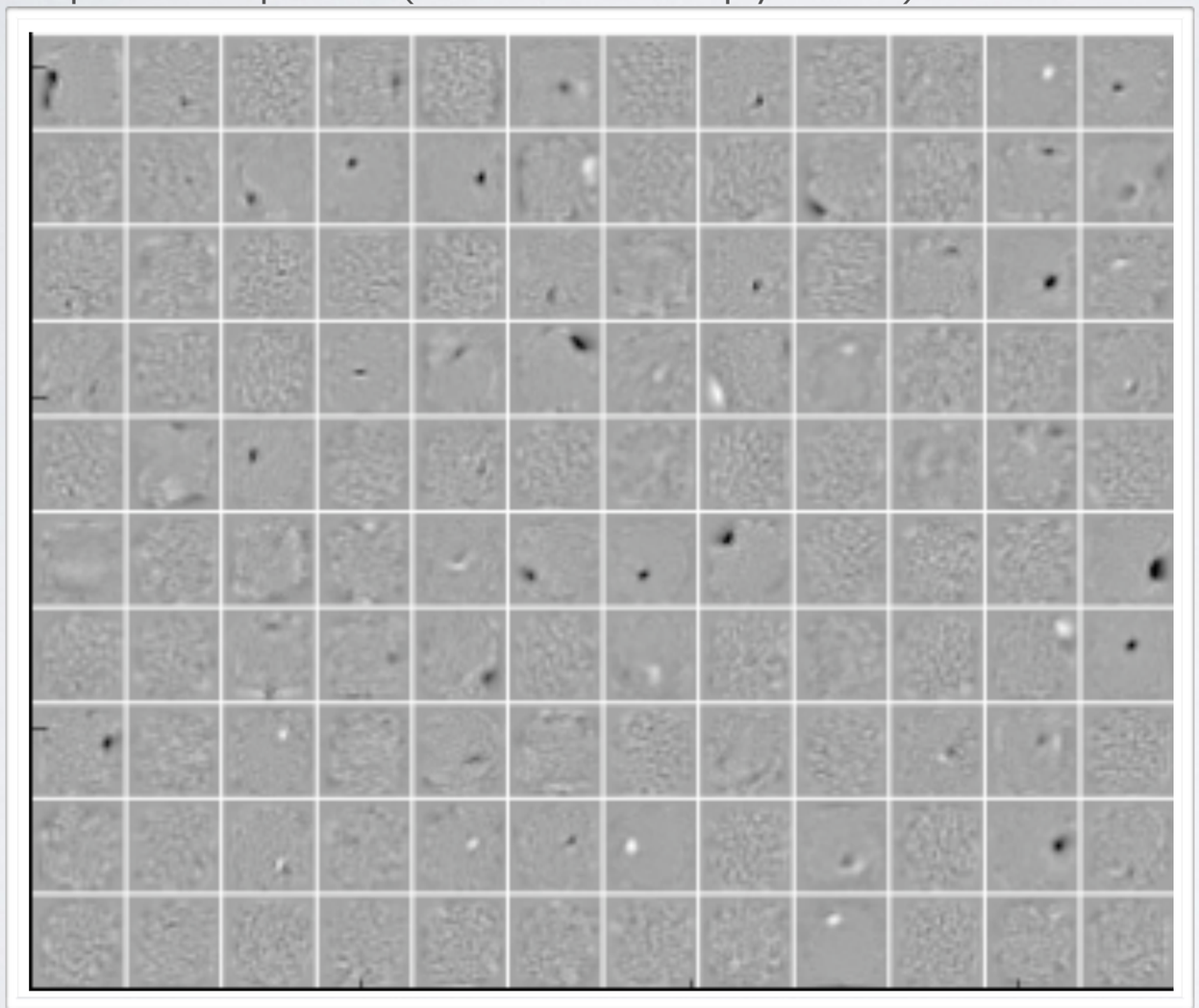
$$p(\widetilde{\mathbf{x}}|\mathbf{x})$$



FILTERS (DENOISING AUTOENCODER)

(Vincent, Larochelle, Bengio and Manzagol, ICML 2008)

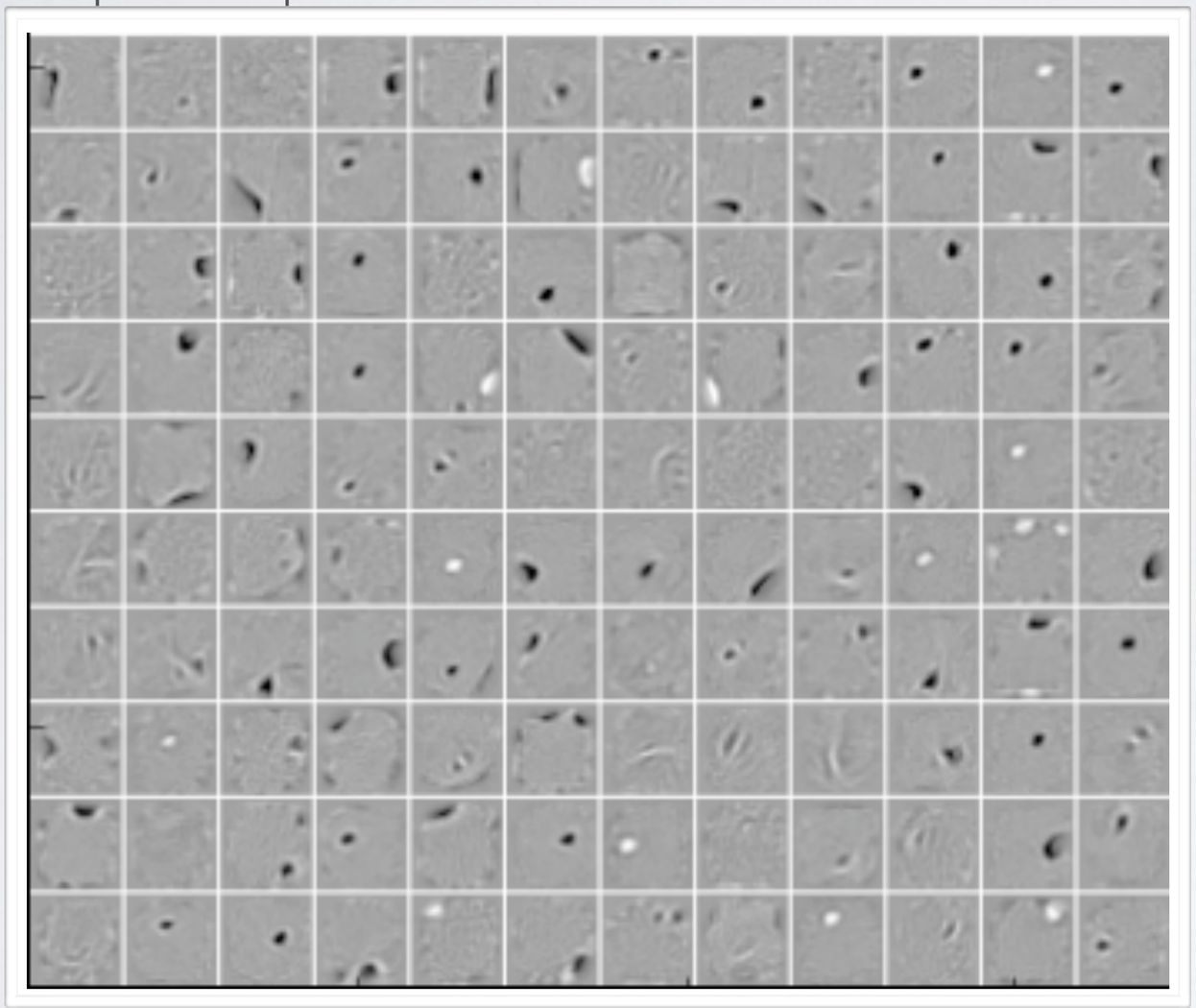
- No corrupted inputs (cross-entropy loss)



FILTERS (DENOISING AUTOENCODER)

(Vincent, Larochelle, Bengio and Manzagol, ICML 2008)

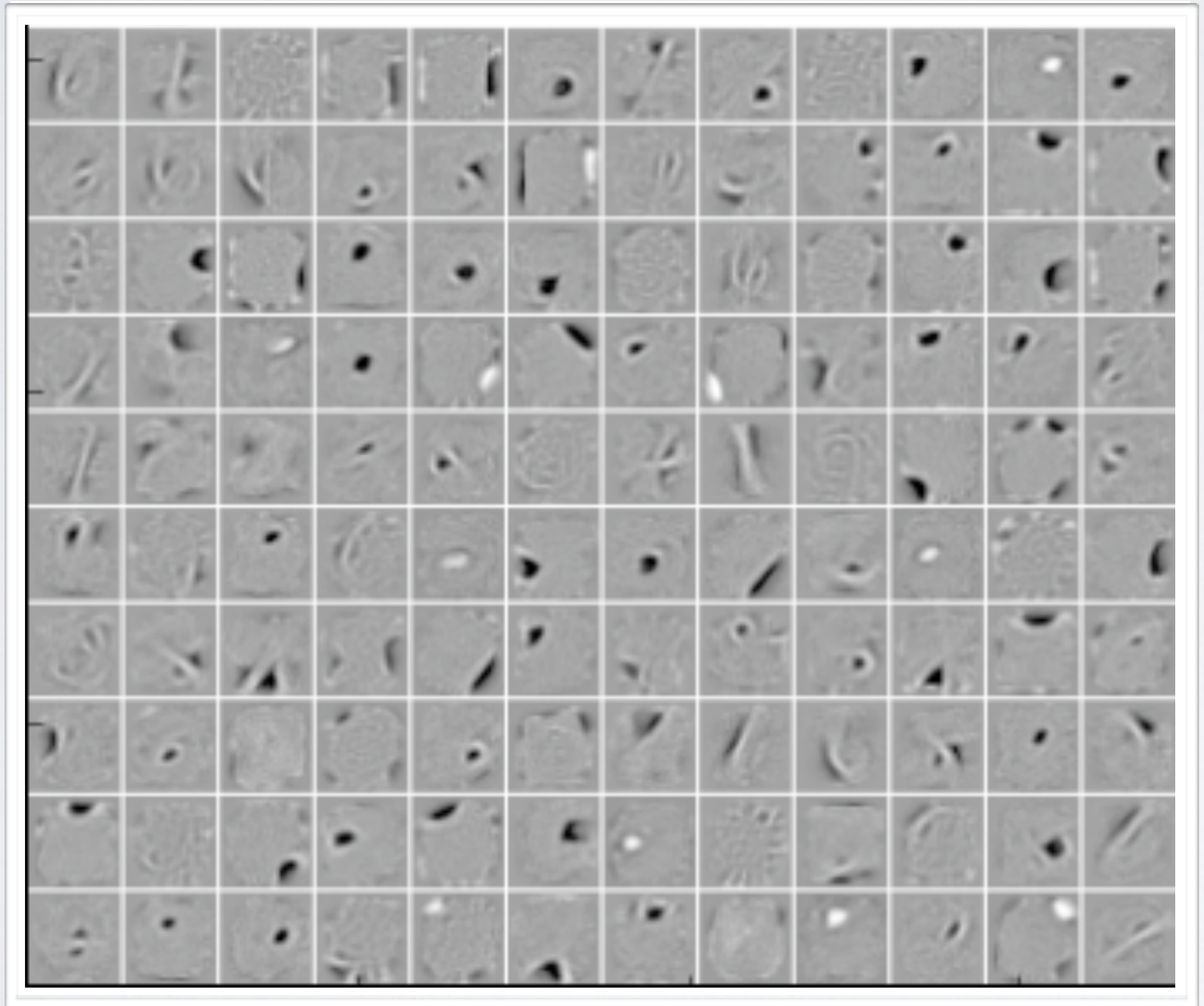
- 25% corrupted inputs



FILTERS (DENOISING AUTOENCODER)

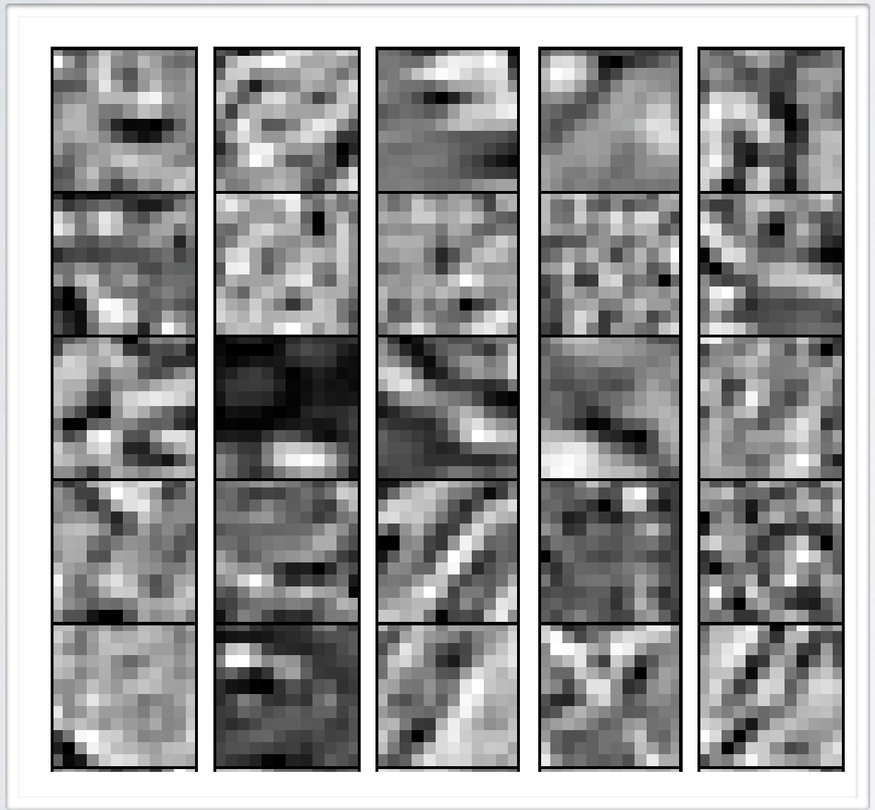
(Vincent, Larochelle, Bengio and Manzagol, ICML 2008)

- 50% corrupted inputs

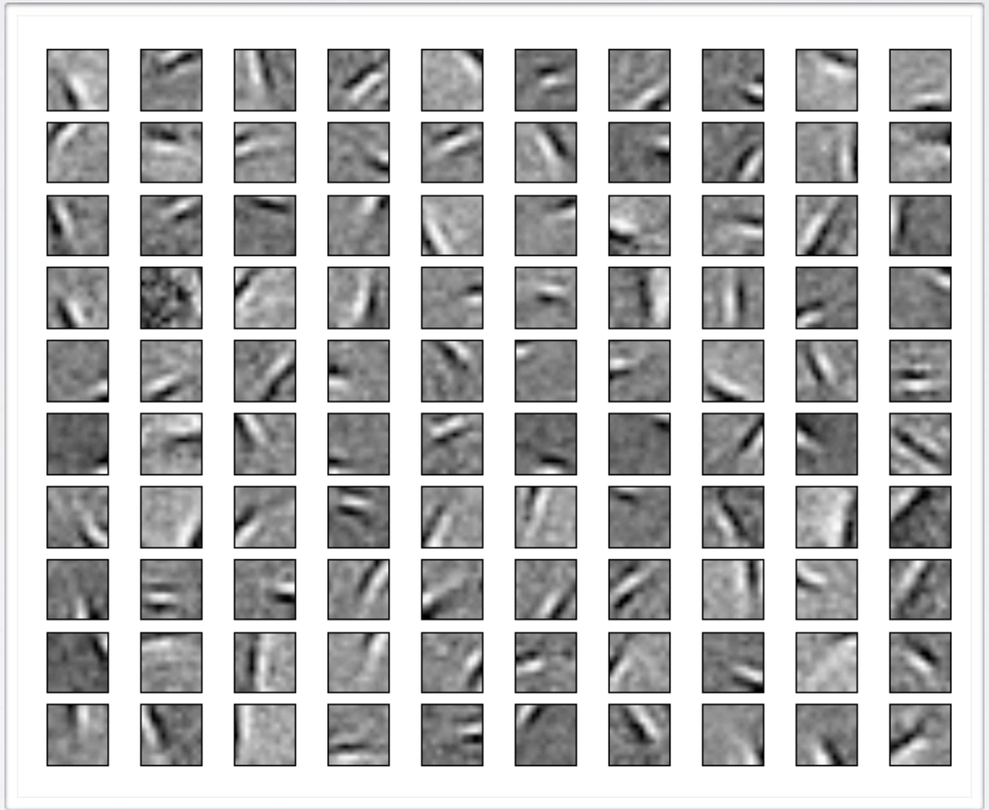


SQUARED ERROR LOSS

- Training on natural image patches, with squared-difference loss
  - PCA is not the best solution



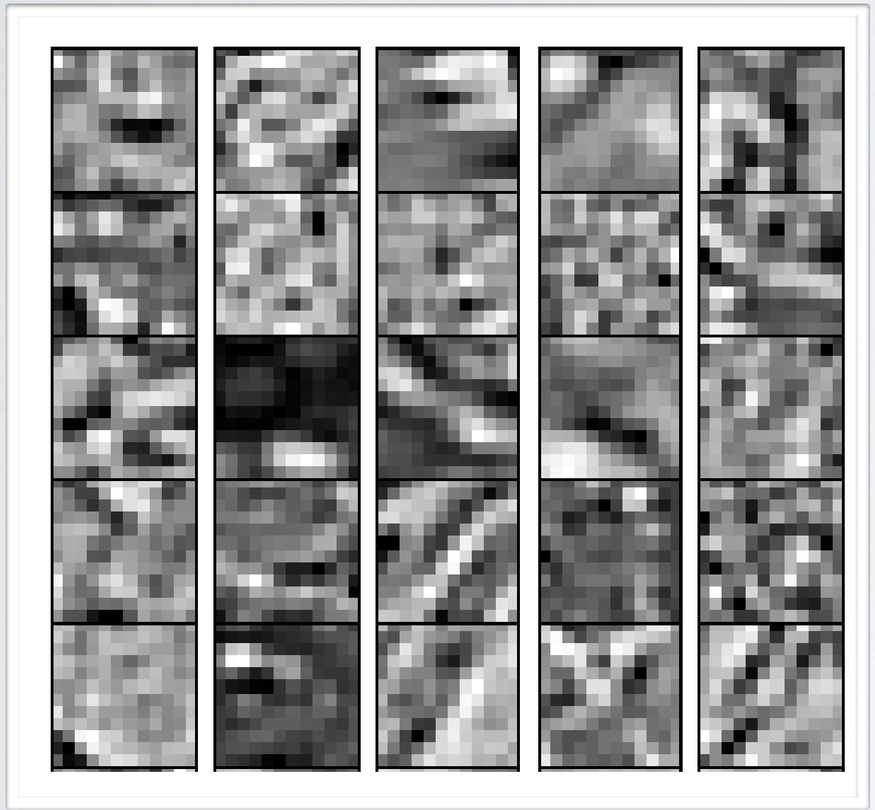
Data



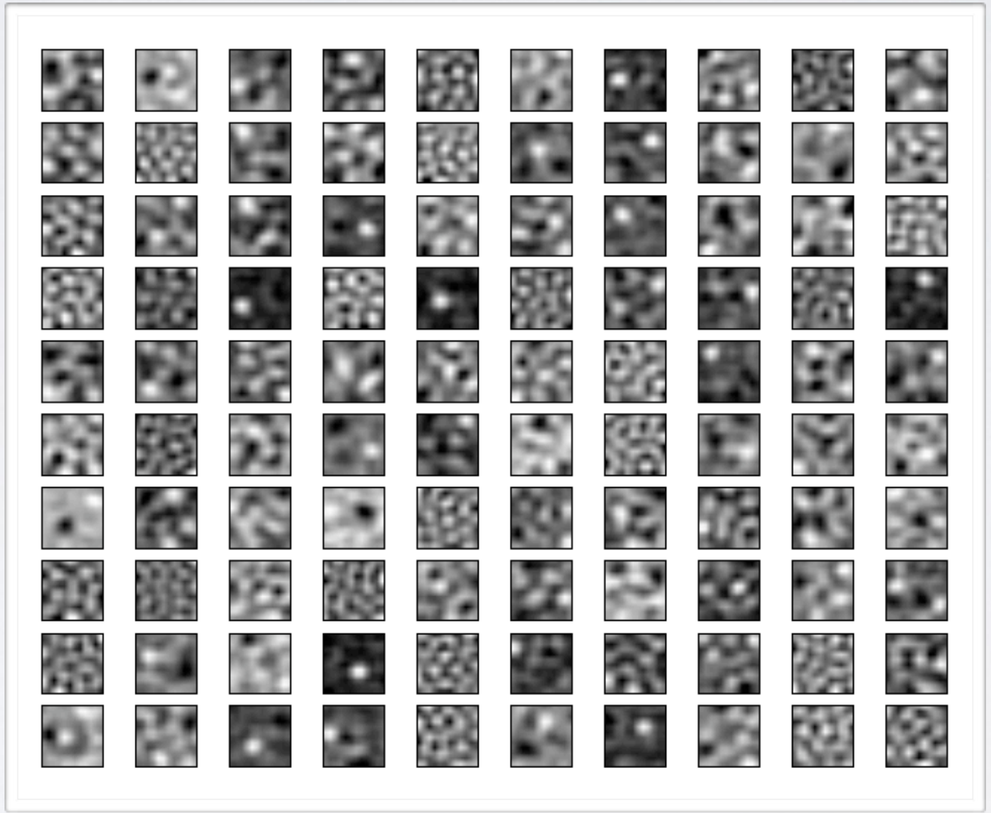
Filters

SQUARED ERROR LOSS

- Training on natural image patches, with squared-difference loss
  - Not equivalent to weight decay



Data

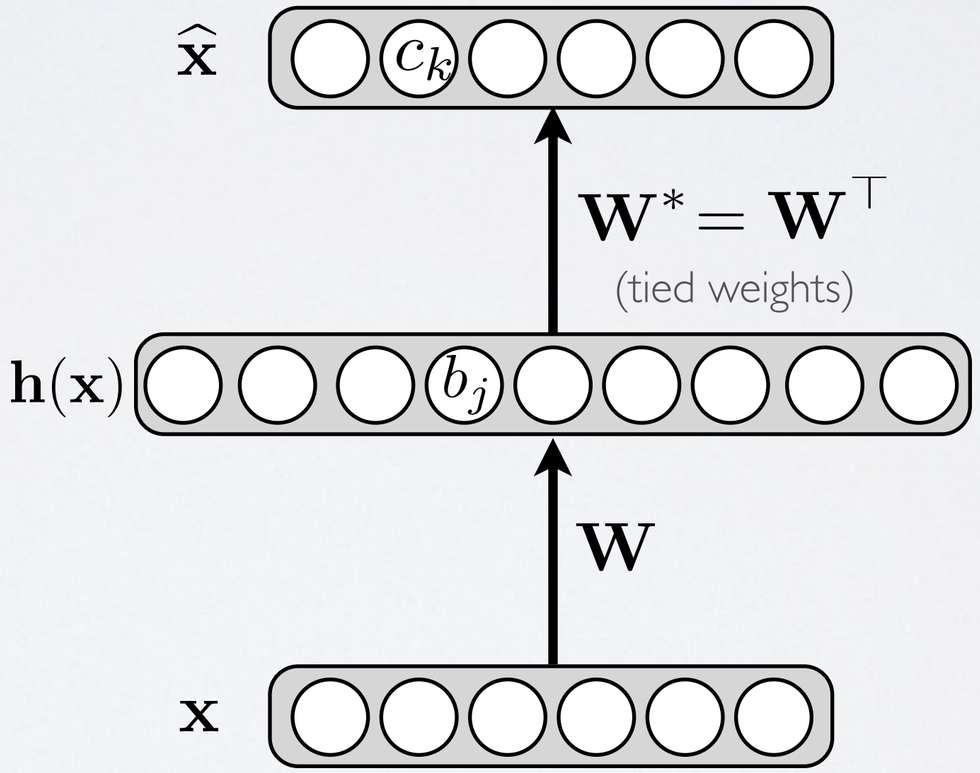


Filters

OVERCOMPLETE HIDDEN LAYER

Topics: overcomplete representation

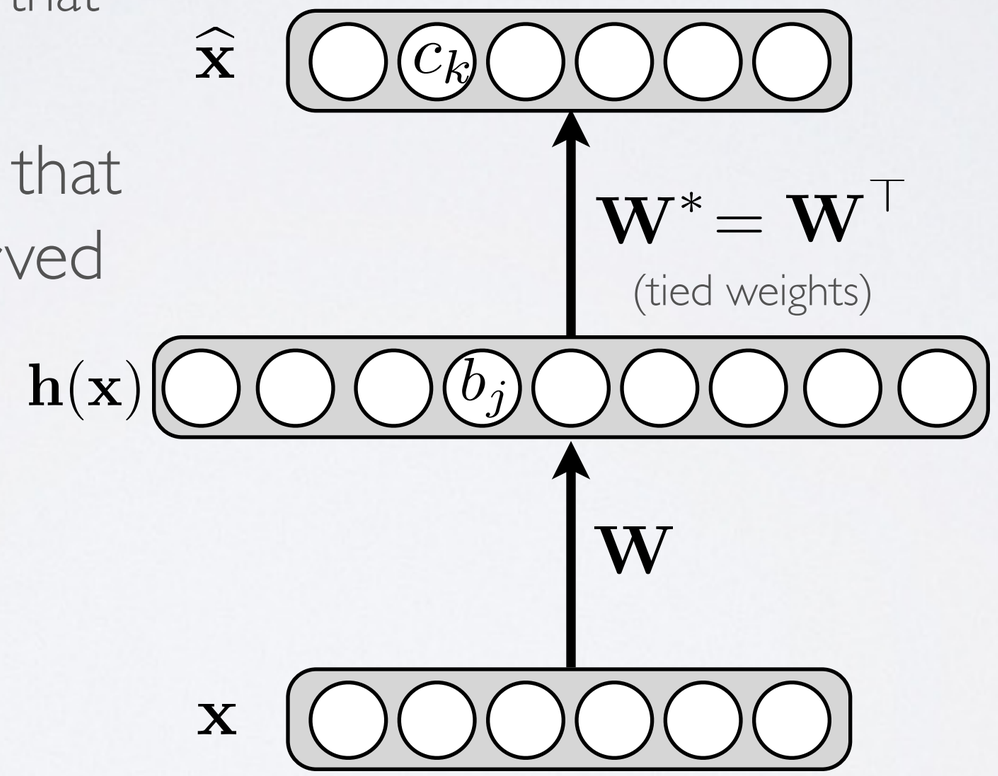
- Hidden layer is overcomplete if greater than the input layer
  - no compression in hidden layer
  - each hidden unit could copy a different input component
- No guarantee that the hidden units will extract meaningful structure



CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

- Alternative approach to avoid uninteresting solutions
  - add an explicit term in the loss that penalizes that solution
- We wish to extract features that **only** reflect variations observed in the training set
  - we'd like to be invariant to the other variations



CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

- New loss function:

$$\underbrace{l(f(\mathbf{x}^{(t)}))}_{\text{autoencoder reconstruction}} + \lambda \underbrace{||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2}_{\text{Jacobian of encoder}}$$

  - where, for binary observations:

$$l(f(\mathbf{x}^{(t)})) = -\sum_k \left(x_k^{(t)} \log(\widehat{x}_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - \widehat{x}_k^{(t)}) \right)$$

$$||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2 = \sum_j \sum_k \left(\frac{\partial h(\mathbf{x}^{(t)})_j}{\partial x_k^{(t)}} \right)^2$$

CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

- New loss function:

$$\underbrace{l(f(\mathbf{x}^{(t)}))}_{\substack{\text{autoencoder} \\ \text{reconstruction}}} + \lambda \underbrace{||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2}_{\substack{\text{Jacobian of} \\ \text{encoder}}}$$

  - where, for binary observations:

$$l(f(\mathbf{x}^{(t)})) = -\sum_k \left(x_k^{(t)} \log(\widehat{x}_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - \widehat{x}_k^{(t)}) \right)$$

encoder keeps good information

$$||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2 = \sum_j \sum_k \left(\frac{\partial h(\mathbf{x}^{(t)})_j}{\partial x_k^{(t)}} \right)^2$$

CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

- New loss function:

$$\underbrace{l(f(\mathbf{x}^{(t)}))}_{\text{autoencoder reconstruction}} + \lambda \underbrace{||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2}_{\text{Jacobian of encoder}}$$

  - where, for binary observations:

$$l(f(\mathbf{x}^{(t)})) = -\sum_k \left(x_k^{(t)} \log(\widehat{x}_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - \widehat{x}_k^{(t)}) \right) \Bigg\} \text{encoder keeps good information}$$

$$||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2 = \sum_j \sum_k \left(\frac{\partial h(\mathbf{x}^{(t)})_j}{\partial x_k^{(t)}} \right)^2 \Bigg\} \text{encoder throws away all information}$$

CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

- New loss function:

$$\underbrace{l(f(\mathbf{x}^{(t)}))}_{\text{autoencoder reconstruction}} + \lambda \underbrace{||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2}_{\text{Jacobian of encoder}}$$

  - where, for binary observations:

$$l(f(\mathbf{x}^{(t)})) = -\sum_k \left(x_k^{(t)} \log(\widehat{x}_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - \widehat{x}_k^{(t)}) \right)$$

encoder keeps good information

$$||\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})||_F^2 = \sum_j \sum_k \left(\frac{\partial h(\mathbf{x}^{(t)})_j}{\partial x_k^{(t)}} \right)^2$$

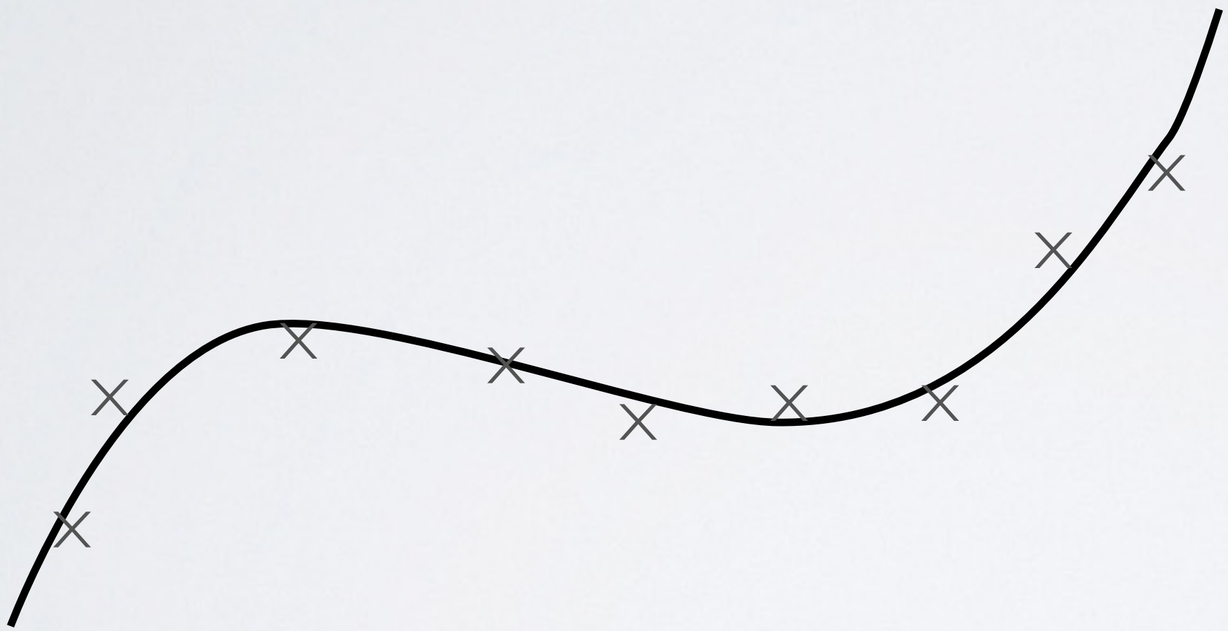
encoder throws away all information

encoder keeps **only** good information

CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

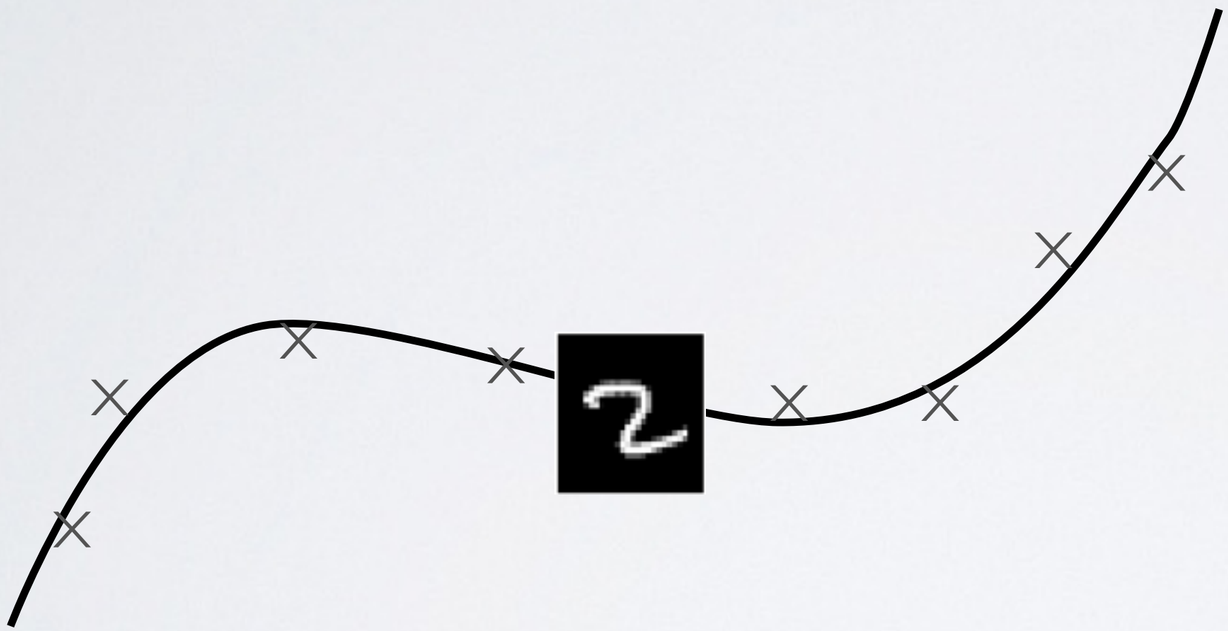
- Illustration:



CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

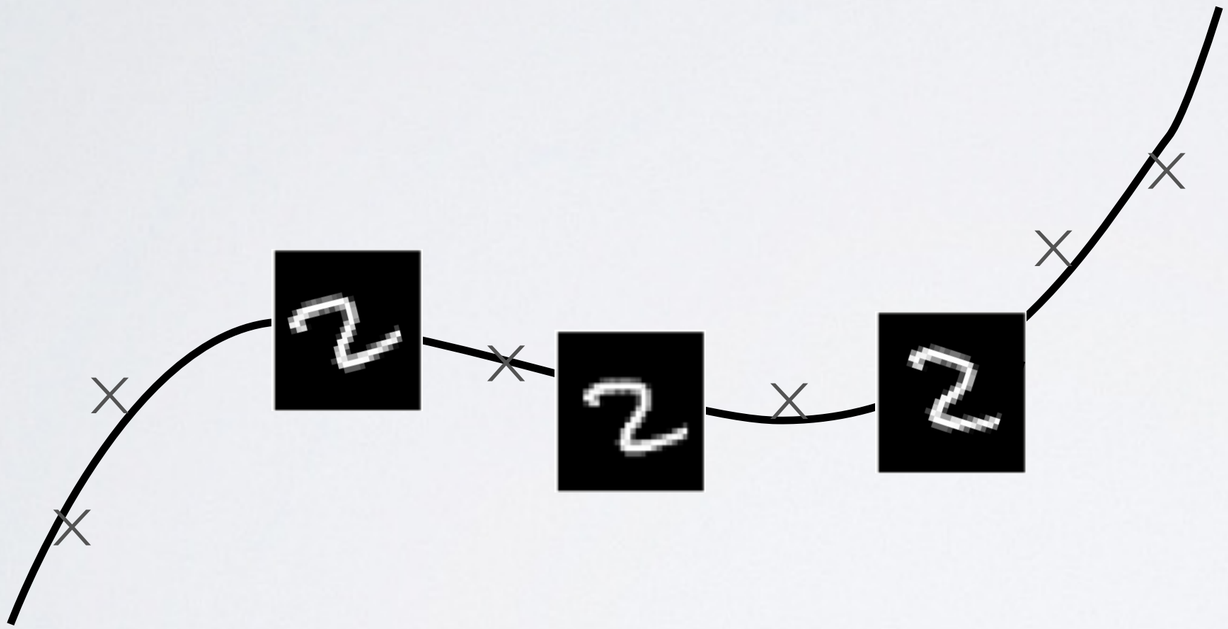
- Illustration:



CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

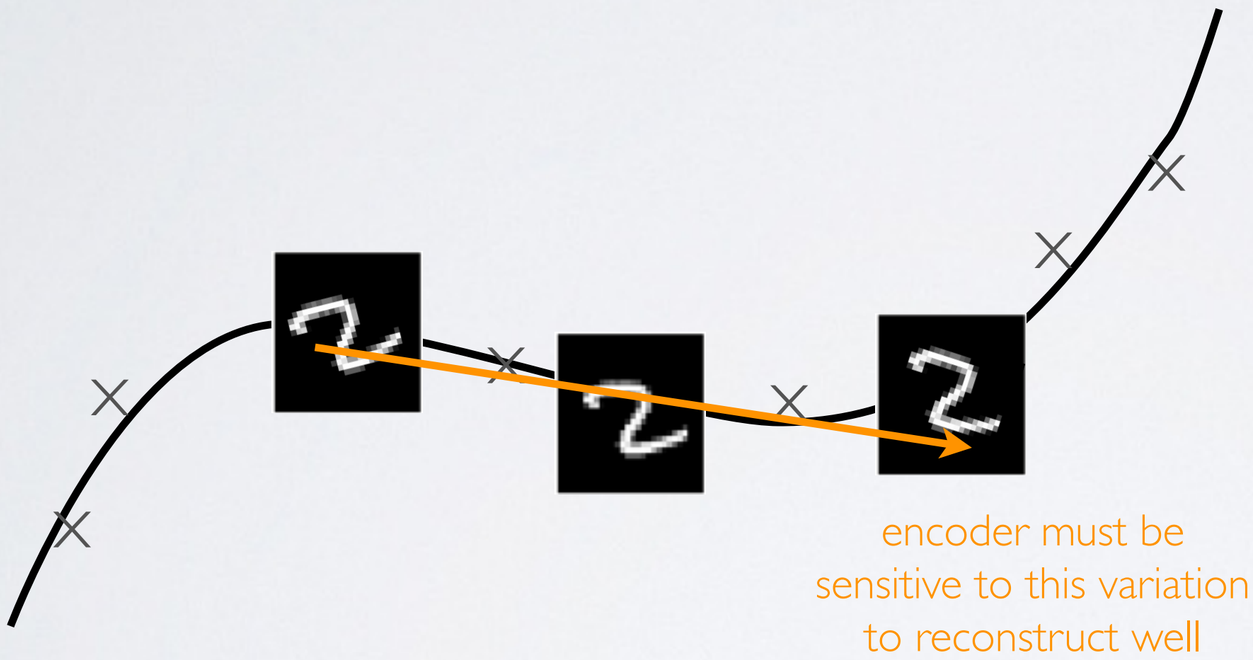
- Illustration:



CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

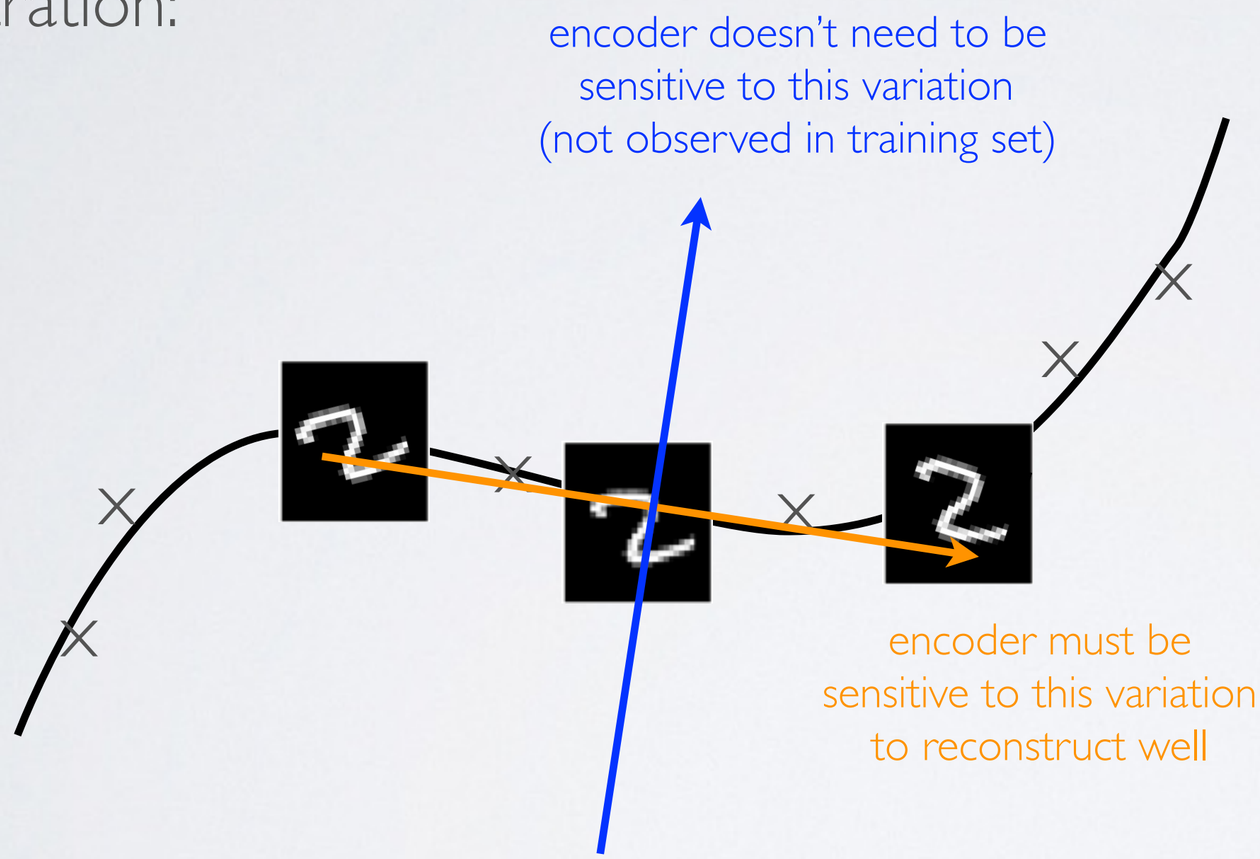
- Illustration:



CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

- Illustration:



WHICH AUTOENCODER ?

Topics: denoising autoencoder, contractive autoencoder

- Both the denoising and contractive autoencoder perform well
  - Advantage of denoising autoencoder: simpler to implement
    - requires adding one or two lines of code to regular autoencoder
    - no need to compute Jacobian of hidden layer
  - Advantage of contractive autoencoder: gradient is deterministic
    - can use second order optimizers (conjugate gradient, LBFGS, etc.)
    - might be more stable than denoising autoencoder, which uses a sampled gradient
- To learn more on contractive autoencoders:
  - Contractive Auto-Encoders: Explicit Invariance During Feature Extraction.
    Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot et Yoshua Bengio, 2011.

Neural networks

Sparse coding - definition

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

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:

reconstruction error

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

  - we also constrain the columns of $\mathbf{D}$ to be of norm 1

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:

reconstruction error

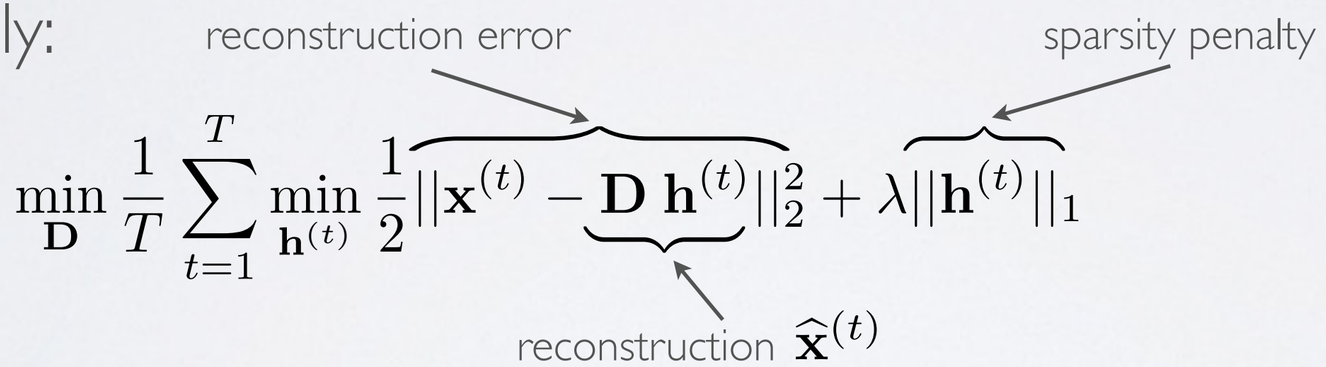
$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^{T} \min_{\mathbf{h}^{(t)}} \overbrace{\frac{1}{2} ||\mathbf{x}^{(t)} - \underbrace{\mathbf{D}\, \mathbf{h}^{(t)}}_{\text{reconstruction } \widehat{\mathbf{x}}^{(t)}}||_2^2}^{} + \lambda ||\mathbf{h}^{(t)}||_1$$

  - we also constrain the columns of $\mathbf{D}$ to be of norm 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:

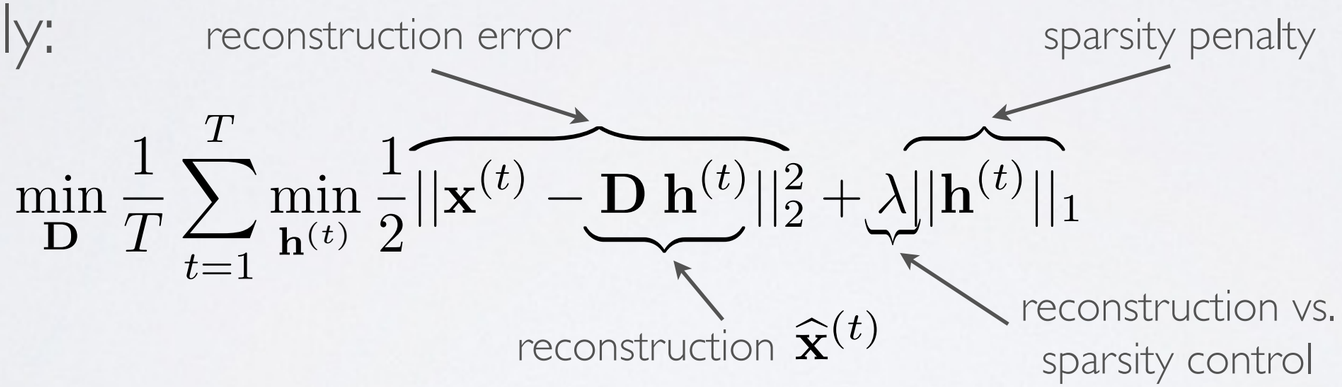


  - we also constrain the columns of $\mathbf{D}$ to be of norm 1

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:

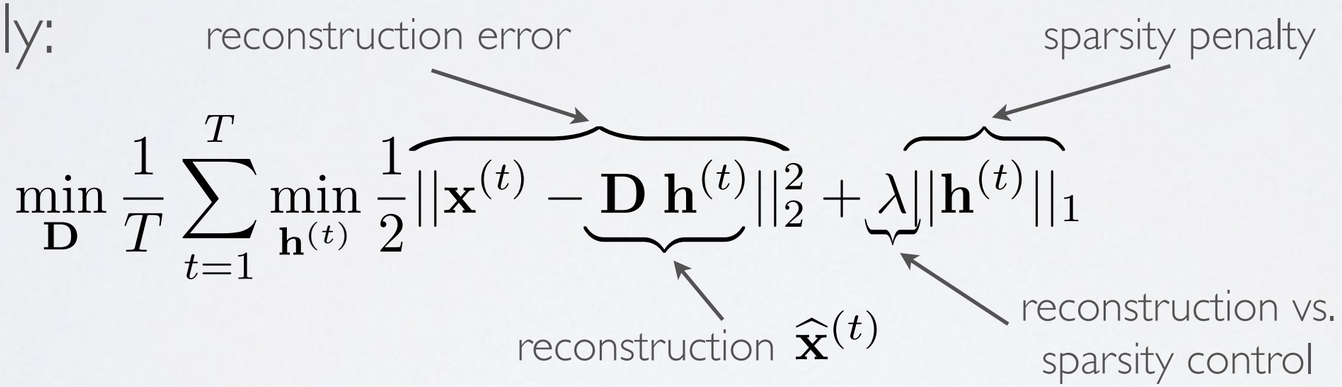


  - we also constrain the columns of $\mathbf{D}$ to be of norm 1

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:

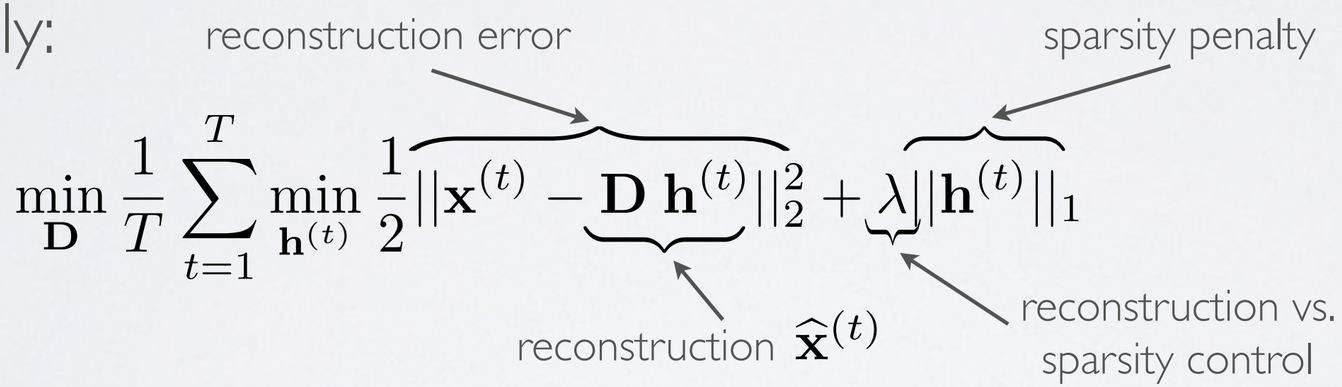


  - we also constrain the columns of $\mathbf{D}$ to be of norm 1
    - otherwise, $\mathbf{D}$ could grow big while $\mathbf{h}^{(t)}$ becomes small to satisfy the prior

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:

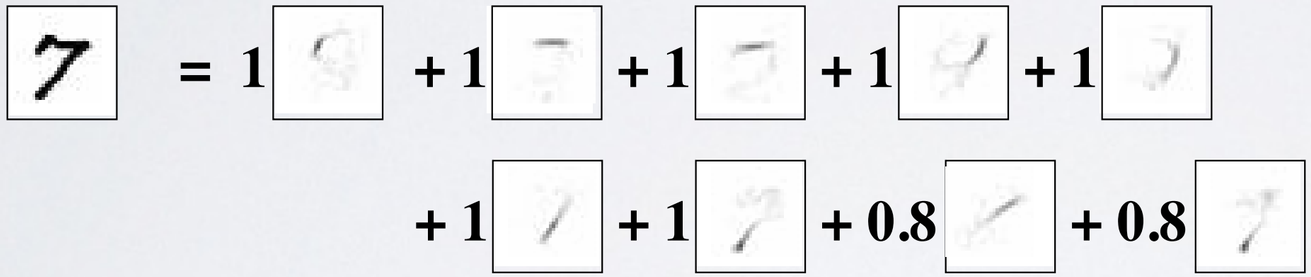


  - $\mathbf{D}$ is equivalent to the autoencoder output weight matrix
  - however, $\mathbf{h}(\mathbf{x}^{(t)})$ is now a complicated function of $\mathbf{x}^{(t)}$
    - encoder is the minimization $\mathbf{h}(\mathbf{x}^{(t)}) = \arg\min_{\mathbf{h}^{(t)}} \frac{1}{2} ||\mathbf{x}^{(t)} - \mathbf{D}\,\mathbf{h}^{(t)}||_2^2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

SPARSE CODING

Topics: dictionary

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

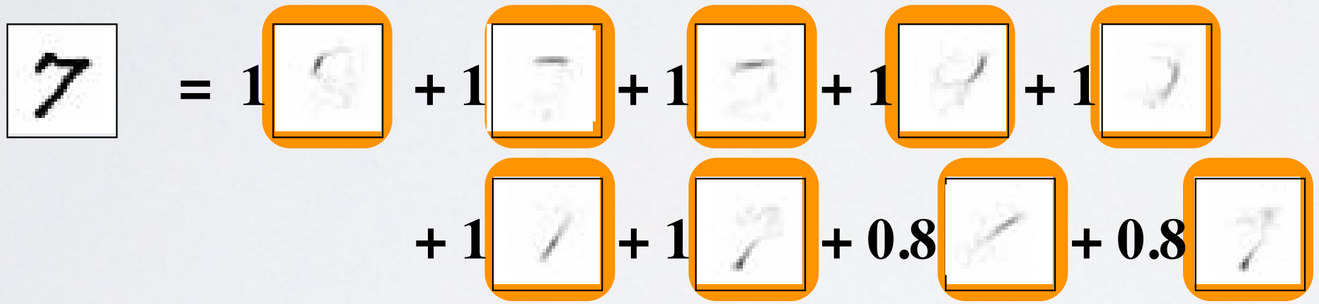


- we also refer to $\mathbf{D}$ as the dictionary
  - in certain applications, we know what dictionary matrix to use
  - often however, we have to learn it

SPARSE CODING

Topics: dictionary

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

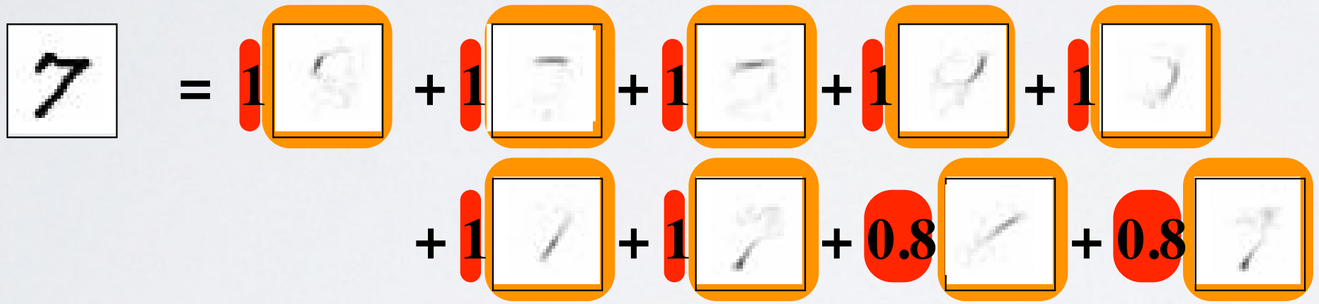


- we also refer to $\mathbf{D}$ as the dictionary
  - in certain applications, we know what dictionary matrix to use
  - often however, we have to learn it

SPARSE CODING

Topics: dictionary

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

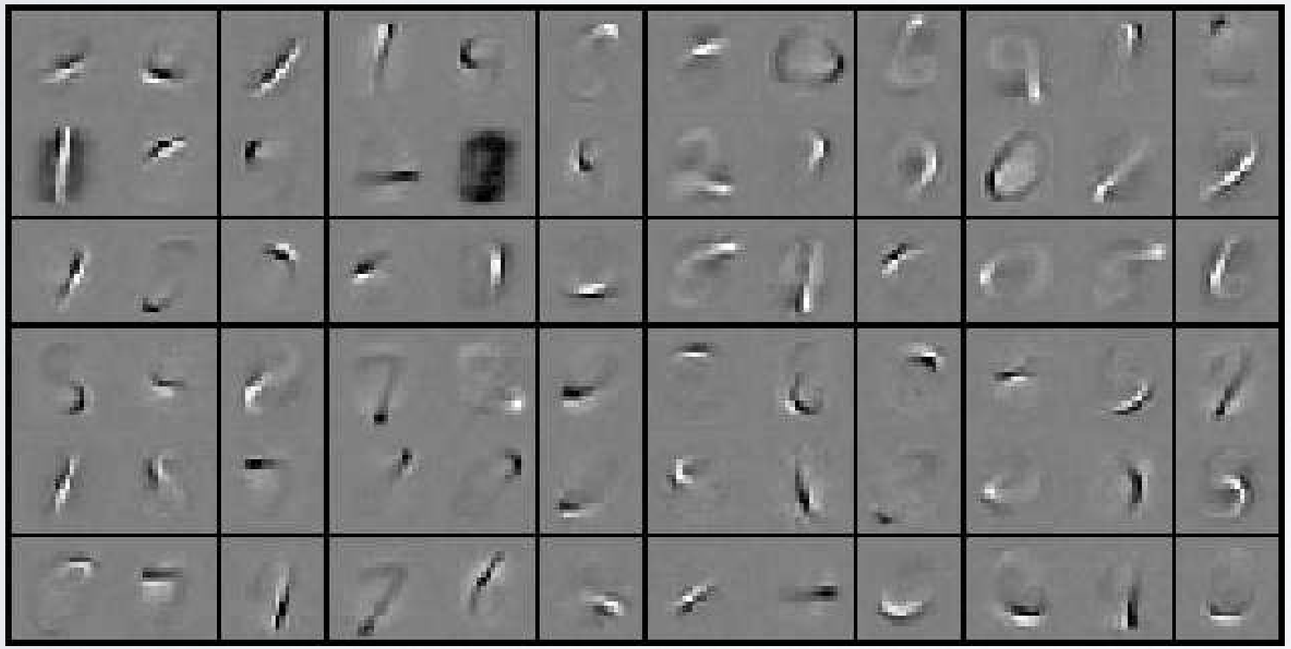


- we also refer to $\mathbf{D}$ as the dictionary
  - in certain applications, we know what dictionary matrix to use
  - often however, we have to learn it

COMPARE FEATURES

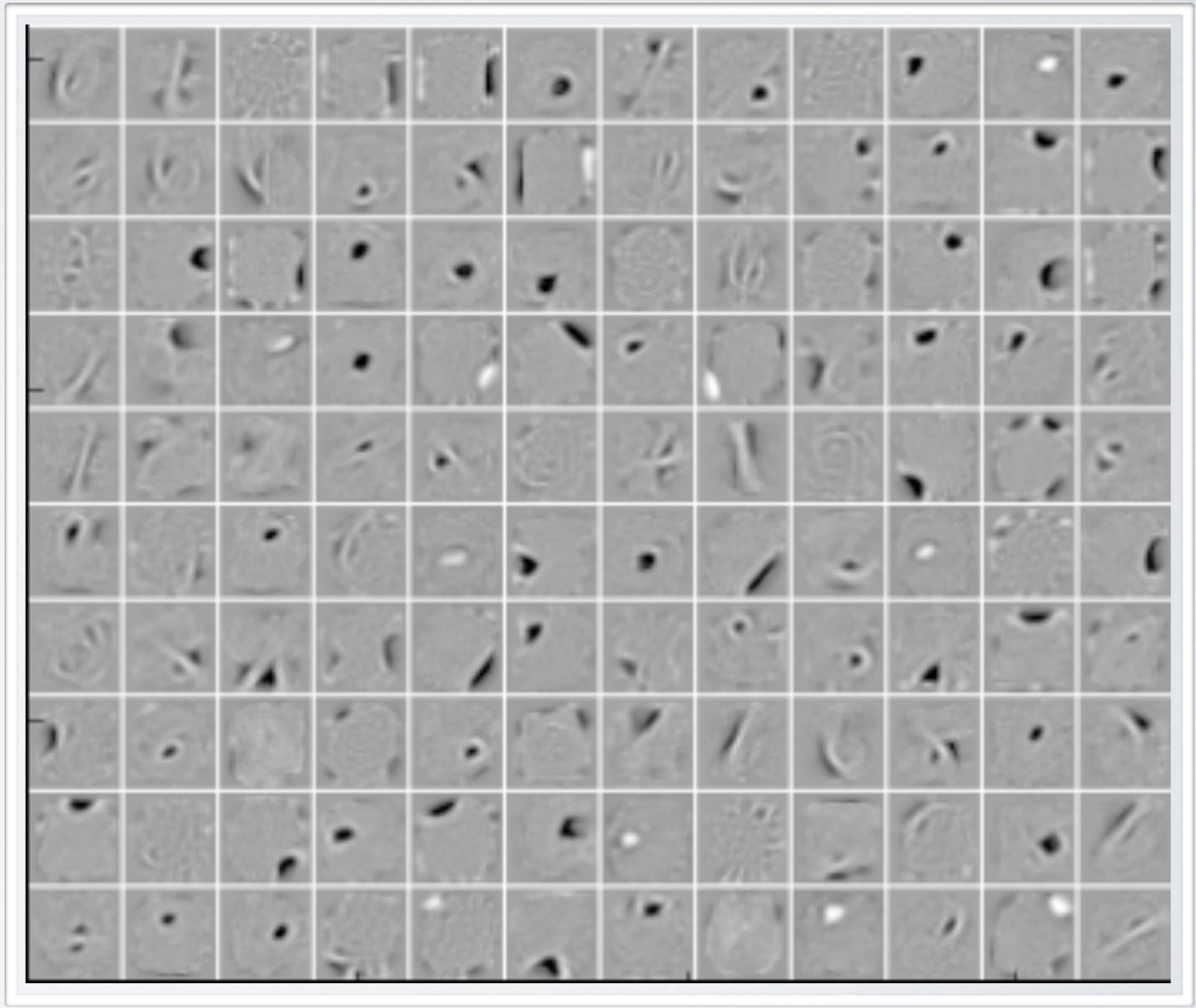
- When trained on handwritten digits (MNIST):

Sparse coding



Self-taught Learning: Transfer Learning from Unlabeled Data
Raina, Battle, Lee, Packer and Ng.

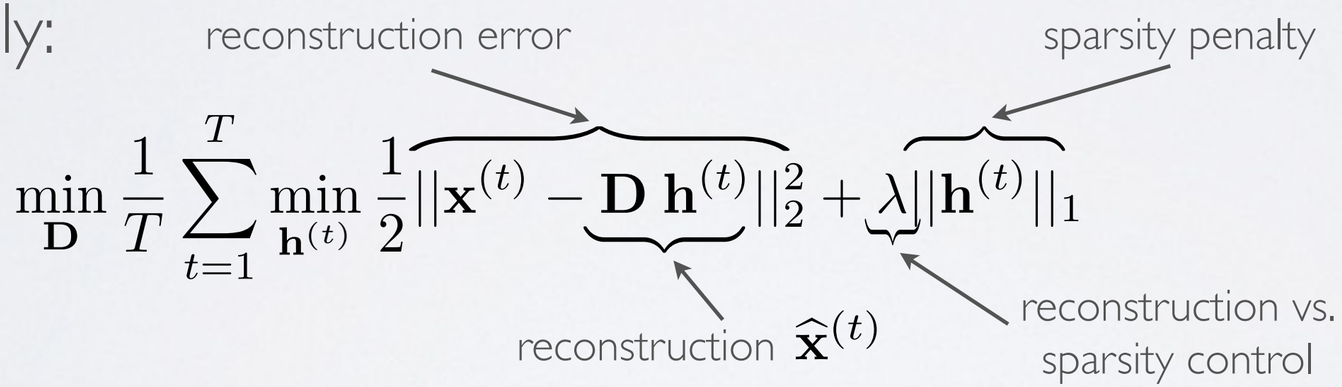
Denoising AE



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:

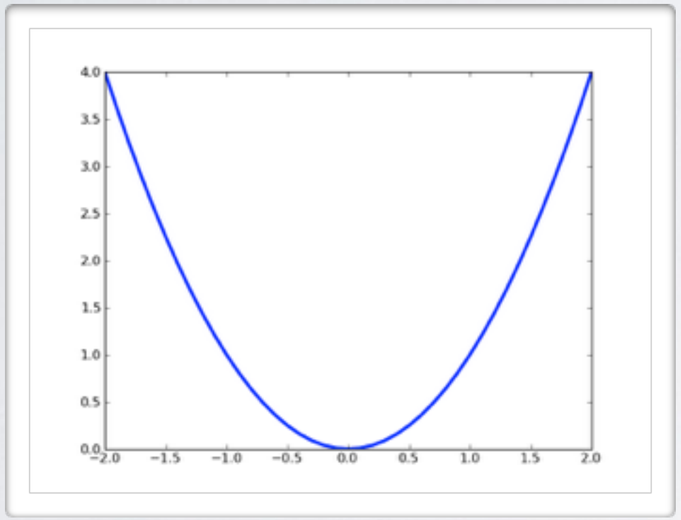


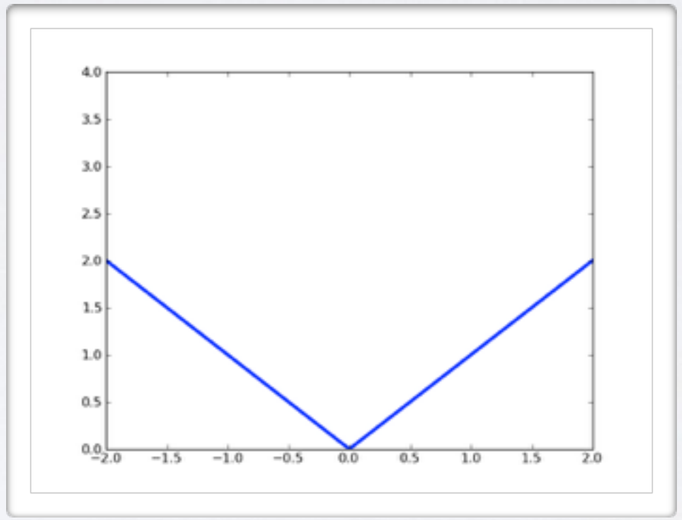
  - $\mathbf{D}$ is equivalent to the autoencoder output weight matrix
  - however, $\mathbf{h}(\mathbf{x}^{(t)})$ is now a complicated function of $\mathbf{x}^{(t)}$
    - encoder is the minimization $\mathbf{h}(\mathbf{x}^{(t)}) = \arg\min_{\mathbf{h}^{(t)}} \frac{1}{2} ||\mathbf{x}^{(t)} - \mathbf{D}\,\mathbf{h}^{(t)}||_2^2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

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:

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

SPARSE CODING

Topics: inference of sparse codes

- For a single hidden unit:

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

  - 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:
    - $h_k^{(t)} \Longleftarrow h_k^{(t)} - \alpha (\mathbf{D}_{.,k})^\top (\mathbf{D}\ \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$
    - if $\text{sign}(h_k^{(t)}) \neq \text{sign}(h_k^{(t)} - \alpha\ \lambda\ \text{sign}(h_k^{(t)}))$ then: $h_k^{(t)} \Longleftarrow 0$
    - else: $h_k^{(t)} \Longleftarrow h_k^{(t)} - \alpha\ \lambda\ \text{sign}(h_k^{(t)})$

SPARSE CODING

Topics: inference of sparse codes

- For a single hidden unit:

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

  - 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:
    - $h_k^{(t)} \Longleftarrow h_k^{(t)} - \alpha (\mathbf{D}_{.,k})^\top (\mathbf{D}\ \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$ } update from reconstruction
    - if $\text{sign}(h_k^{(t)}) \neq \text{sign}(h_k^{(t)} - \alpha\ \lambda\ \text{sign}(h_k^{(t)}))$ then: $h_k^{(t)} \Longleftarrow 0$
    - else: $h_k^{(t)} \Longleftarrow h_k^{(t)} - \alpha\ \lambda\ \text{sign}(h_k^{(t)})$

SPARSE CODING

Topics: inference of sparse codes

- For a single hidden unit:

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

  - 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:
    - $h_k^{(t)} \Longleftarrow h_k^{(t)} - \alpha (\mathbf{D}_{\cdot,k})^\top (\mathbf{D}\ \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$ } update from reconstruction
    - if $\text{sign}(h_k^{(t)}) \neq \text{sign}(h_k^{(t)} - \alpha\ \lambda\ \text{sign}(h_k^{(t)}))$ then: $h_k^{(t)} \Longleftarrow 0$ } update from sparsity
    - else: $h_k^{(t)} \Longleftarrow h_k^{(t)} - \alpha\ \lambda\ \text{sign}(h_k^{(t)})$

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
    - $\mathbf{h}^{(t)} \Longleftarrow \mathbf{h}^{(t)} - \alpha\, \mathbf{D}^\top (\mathbf{D}\, \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$
    - $\mathbf{h}^{(t)} \Longleftarrow \mathrm{shrink}(\mathbf{h}^{(t)}, \alpha\, \lambda)$
  - return $\mathbf{h}^{(t)}$

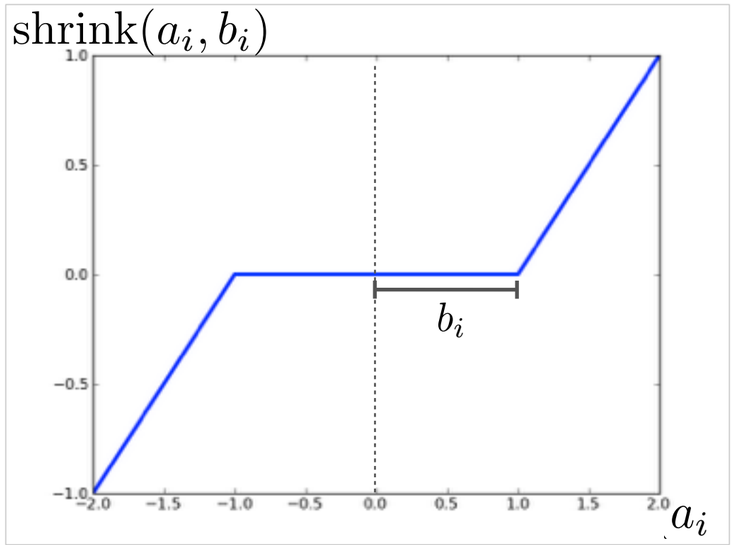
- Here $\mathrm{shrink}(\mathbf{a}, \mathbf{b}) = [\ldots, \mathrm{sign}(a_i)\ \max(|a_i| - b_i, 0), \ldots]$
- Will converge if $\frac{1}{\alpha}$ is bigger than the largest eigenvalue of $\mathbf{D}^\top \mathbf{D}$

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
    - $\mathbf{h}^{(t)} \Longleftarrow \mathbf{h}^{(t)} - \alpha\, \mathbf{D}^\top(\mathbf{D}\, \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$
    - $\mathbf{h}^{(t)} \Longleftarrow \mathrm{shrink}(\mathbf{h}^{(t)}, \alpha\, \lambda)$
  - return $\mathbf{h}^{(t)}$



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

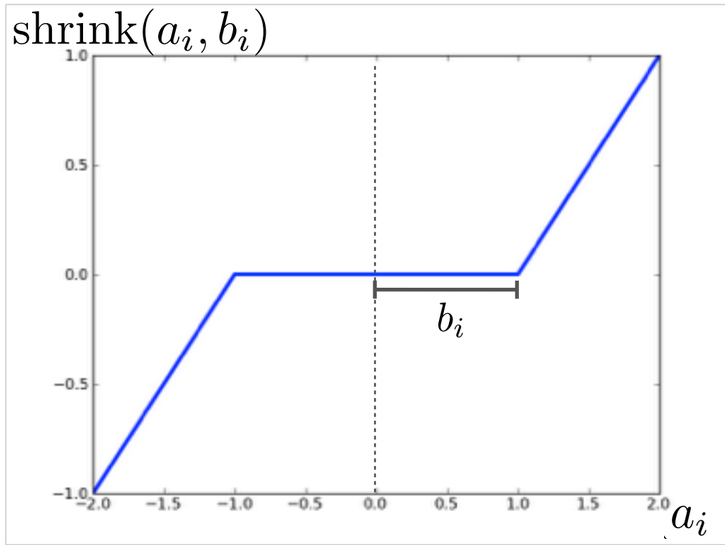
SPARSE CODING

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
    - $\mathbf{h}^{(t)} \Longleftarrow \mathbf{h}^{(t)} - \alpha\, \mathbf{D}^\top(\mathbf{D}\, \mathbf{h}^{(t)} - \mathbf{x}^{(t)})$
    - $\mathbf{h}^{(t)} \Longleftarrow \mathrm{shrink}(\mathbf{h}^{(t)}, \alpha\, \lambda)$
  - return $\mathbf{h}^{(t)}$

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



- Here $\mathrm{shrink}(\mathbf{a}, \mathbf{b}) = [\ldots, \mathrm{sign}(a_i)\ \max(|a_i| - b_i, 0), \ldots]$
- 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

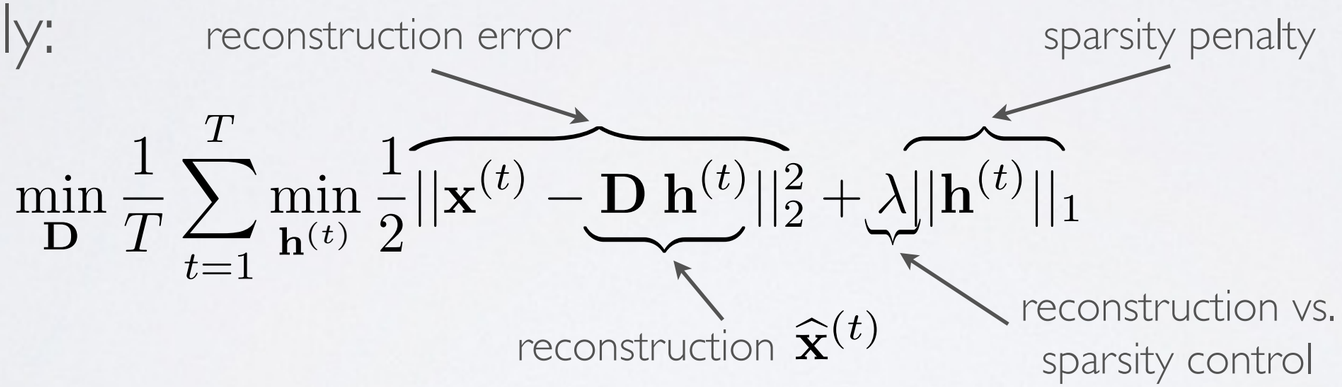
- ISTA updates all hidden units simultaneously
  - this is wasteful if many hidden units have already converged

- 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

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:



  - $\mathbf{D}$ 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 1)

- 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 1)

- 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 $\mathbf{D}$

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

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

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

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

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

  - 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

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

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

$$= \frac{1}{A_{j,j}} (\mathbf{B}_{\cdot,j} - \mathbf{D}\ \mathbf{A}_{\cdot,j} + \mathbf{D}_{\cdot,j} A_{j,j})$$

  - this way, we only need to store:
    - $\mathbf{A} \Longleftarrow \sum_{t=1}^{T} \mathbf{h}(\mathbf{x}^{(t)})\ \mathbf{h}(\mathbf{x}^{(t)})^\top$

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

SPARSE CODING

Topics: dictionary update (algorithm 2)

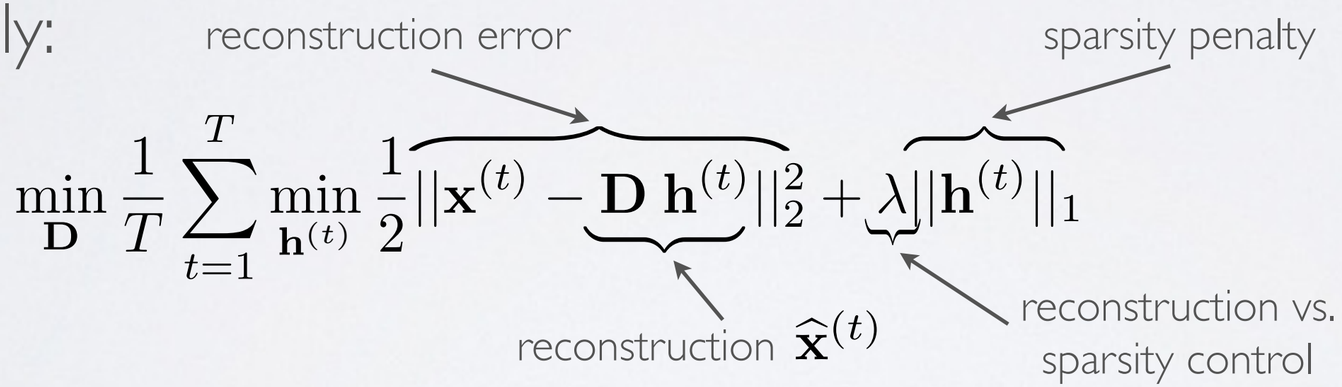
- While $\mathbf{D}$ hasn't converged
  - for each column $\mathbf{D}_{.,j}$ perform updates
    - $\mathbf{D}_{.,j} \Longleftarrow \frac{1}{A_{j,j}} (\mathbf{B}_{.,j} - \mathbf{D}\ \mathbf{A}_{.,j} + \mathbf{D}_{.,j}\ A_{j,j})$
    - $\mathbf{D}_{.,j} \Longleftarrow \frac{\mathbf{D}_{.,j}}{||\mathbf{D}_{.,j}||_2}$

- 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 $\mathbf{D}_{.,j}$

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:



  - $\mathbf{D}$ 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:
    - $\mathbf{A} \Longleftarrow \sum_{t=1}^{T} \mathbf{x}^{(t)}\ \mathbf{h}(\mathbf{x}^{(t)})^\top$
    - $\mathbf{B} \Longleftarrow \sum_{t=1}^{T} \mathbf{h}(\mathbf{x}^{(t)})\ \mathbf{h}(\mathbf{x}^{(t)})^\top$
    - run block-coordinate descent algorithm to update $\mathbf{D}$

- Similar to the EM algorithm

SPARSE CODING

Topics: online learning algorithm

- This algorithm is "batch" (i.e. not online)
  - single update of the dictionary per pass on the training set
  - for large datasets, we'd like to update $\mathbf{D}$ after visiting each $\mathbf{x}^{(t)}$
- Solution: for each $\mathbf{x}^{(t)}$
  - perform inference of $\mathbf{h}(\mathbf{x}^{(t)})$ for the current $\mathbf{x}^{(t)}$
  - update running averages of the quantities required to update $\mathbf{D}$:
    - $\mathbf{B} \Longleftarrow \beta\, \mathbf{B} + (1-\beta)\, \mathbf{x}^{(t)}\, \mathbf{h}(\mathbf{x}^{(t)})^\top$
    - $\mathbf{A} \Longleftarrow \beta\, \mathbf{A} + (1-\beta)\, \mathbf{h}(\mathbf{x}^{(t)})\, \mathbf{h}(\mathbf{x}^{(t)})^\top$
  - use current value of $\mathbf{D}$ as "warm start" to block-coordinate descent

SPARSE CODING

Topics: online learning algorithm

- Initialize $\mathbf{D}$ (not to 0!)
- While $\mathbf{D}$ hasn't converged
  - for each $\mathbf{x}^{(t)}$
    - infer code $\mathbf{h}(\mathbf{x}^{(t)})$
    - update dictionary
      - ✓ $\mathbf{B} \Longleftarrow \beta\, \mathbf{B} + (1-\beta)\, \mathbf{x}^{(t)}\, \mathbf{h}(\mathbf{x}^{(t)})^\top$
      - ✓ $\mathbf{A} \Longleftarrow \beta\, \mathbf{A} + (1-\beta)\, \mathbf{h}(\mathbf{x}^{(T+1)})\, \mathbf{h}(\mathbf{x}^{(T+1)})^\top$
      - ✓ while $\mathbf{D}$ hasn't converged
        - ★ for each column $\mathbf{D}_{\cdot,j}$ perform gradient update

$$\mathbf{D}_{\cdot,j} \Longleftarrow \frac{1}{A_{j,j}} \left(\mathbf{B}_{\cdot,j} - \mathbf{D}\, \mathbf{A}_{\cdot,j} + \mathbf{D}_{\cdot,j}\, A_{j,j}\right)$$

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

Online Dictionary Learning for Sparse Coding. Mairal, Bach, Ponce and Sapiro, 2009.

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 $\widehat{\boldsymbol{\mu}}$ and the empirical covariance matrix be $\widehat{\Sigma} = \mathbf{U}\Lambda\mathbf{U}^\top$ (in its eigenvalue/eigenvector representation)
  - ZCA transforms each input $\mathbf{x}$ as follows:

$$\mathbf{x} \Longleftarrow \mathbf{U}\ \Lambda^{-\frac{1}{2}}\ \mathbf{U}^\top(\mathbf{x} - \widehat{\boldsymbol{\mu}})$$

PREPROCESSING

Topics: ZCA

- After this transformation
  - the empirical mean is 0

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

PREPROCESSING

Topics: ZCA

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

$$\frac{1}{T-1}\sum_t \left(\mathbf{U}\,\Lambda^{-\frac{1}{2}}\,\mathbf{U}^\top(\mathbf{x}^{(t)}-\widehat{\boldsymbol{\mu}})\right)\left(\mathbf{U}\,\Lambda^{-\frac{1}{2}}\,\mathbf{U}^\top(\mathbf{x}^{(t)}-\widehat{\boldsymbol{\mu}})\right)^\top$$

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

FEATURE EXTRACTION

Topics: feature learning

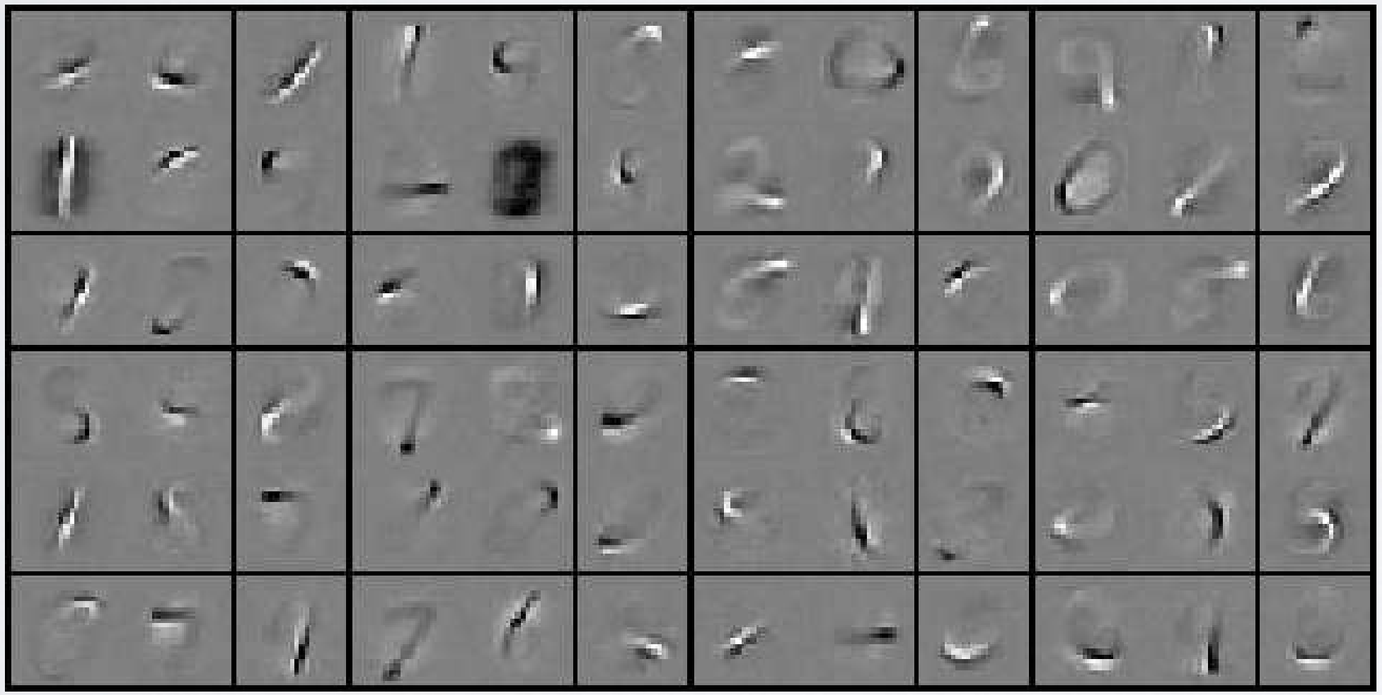
- A sparse coding model can be used to extract features
  - given a labeled training set $\{(\mathbf{x}^{(t)}, y^{(t)})\}$
  - train sparse coding dictionary only on training inputs $\{\mathbf{x}^{(t)}\}$
    - this yields a dictionary $\mathbf{D}$ from which to infer sparse codes $\mathbf{h}(\mathbf{x}^{(t)})$
  - 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:



Self-taught Learning: Transfer Learning from Unlabeled Data
Raina, Battle, Lee, Packer and Ng.

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%**

Self-taught Learning: Transfer Learning from Unlabeled Data
Raina, Battle, Lee, Packer and Ng.

RELATIONSHIP WITH V1

Topics: V1 neurons vs. sparse coding

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

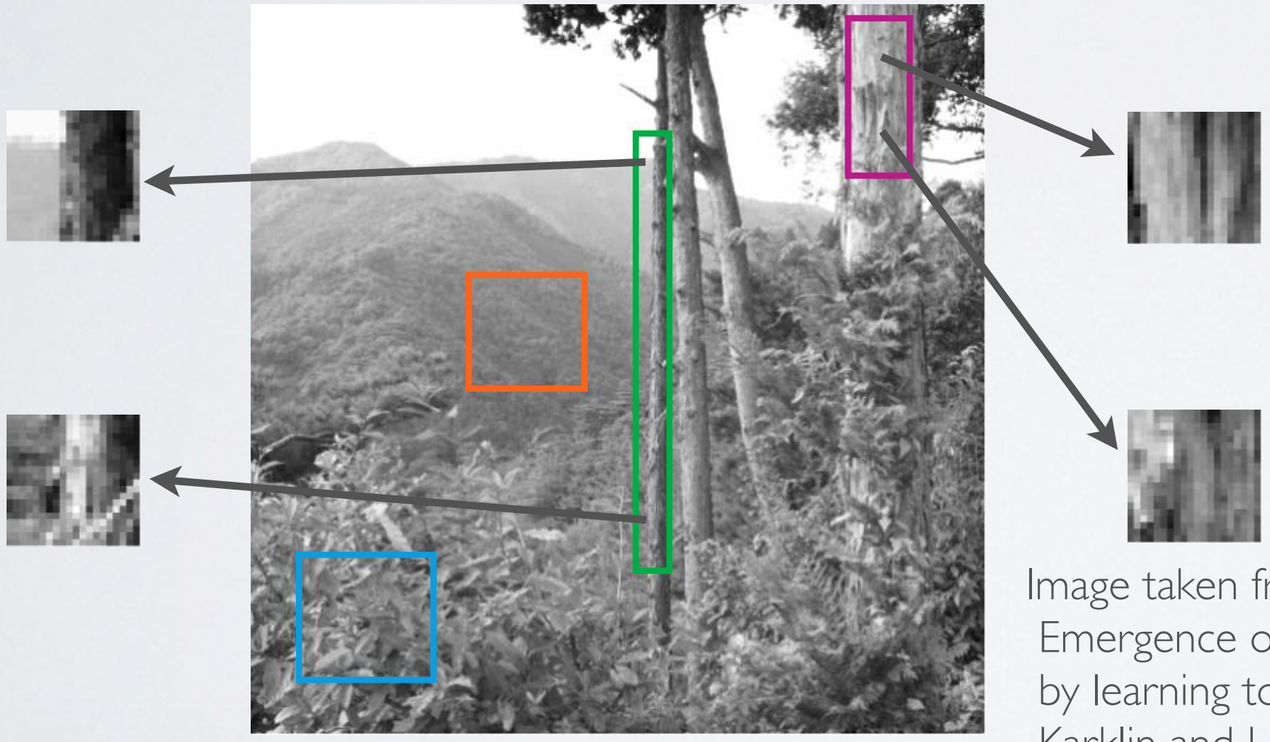
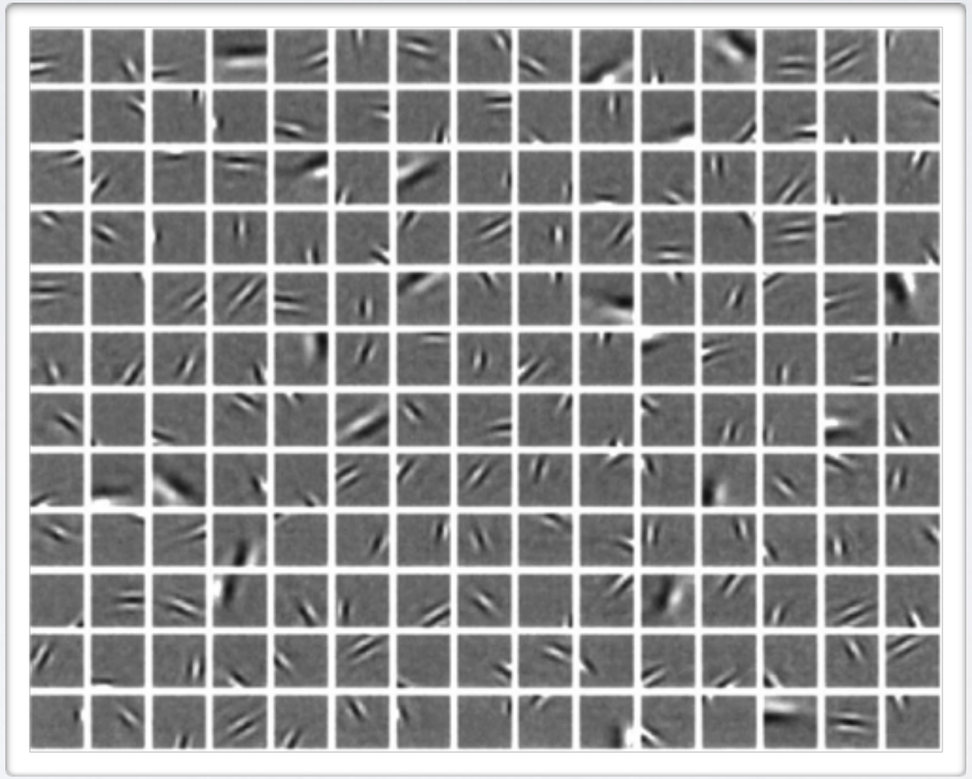


Image taken from:
Emergence of complex cell properties
by learning to generalize in natural scenes.
Karklin and Lewicki, 2009

RELATIONSHIP WITH V1

Topics: V1 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
  - V1 neurons in the mammalian brain have a similar behavior



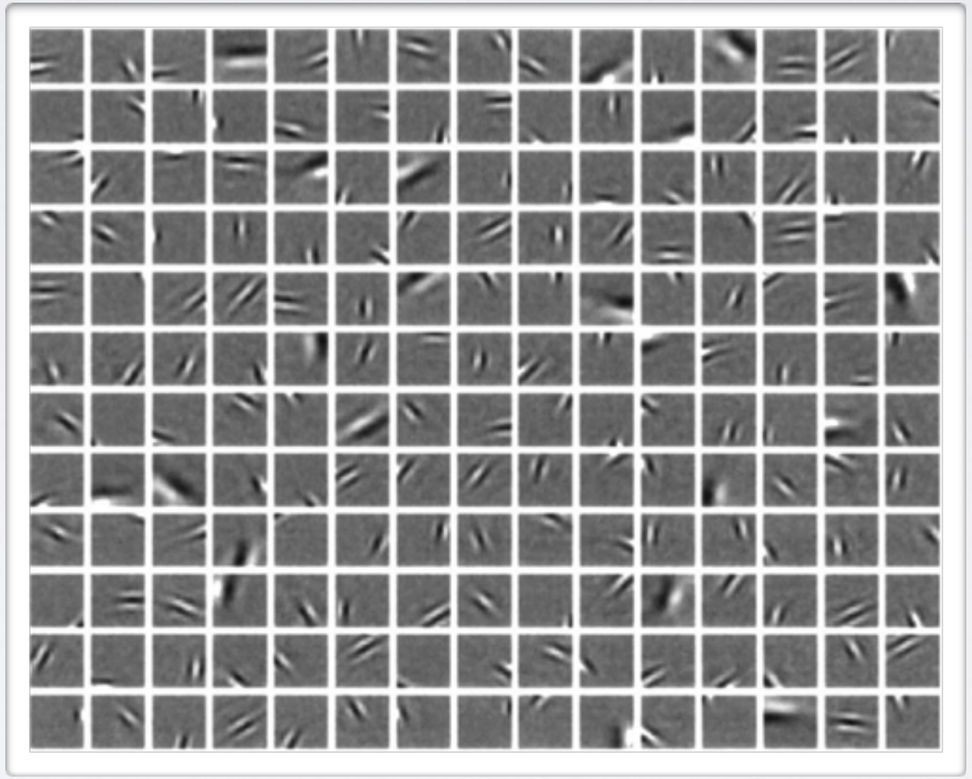
Emergence of simple-cell receptive field properties by learning a sparse code of natural images. Olshausen and Field, 1996.

RELATIONSHIP WITH V1

Topics: V1 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.