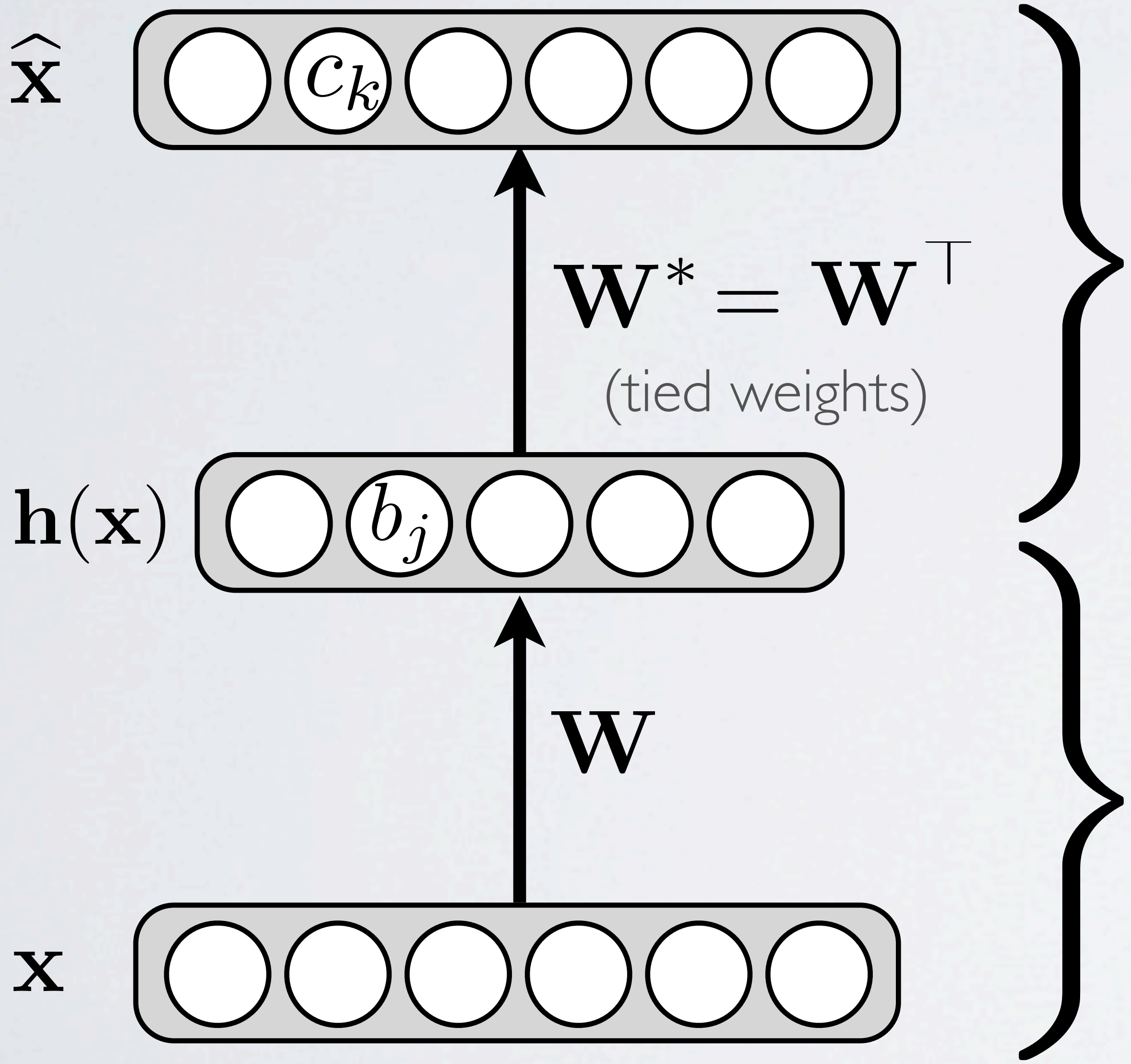
Neural networks

Autoencoder - loss function

AUTOENCODER

Topics: autoencoder, encoder, decoder, tied weights

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



Decoder

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

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

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

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

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

- 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

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

- 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 $\boldsymbol{\mu} = \mathbf{c} + \mathbf{W}^* \mathbf{h}(\mathbf{x})$