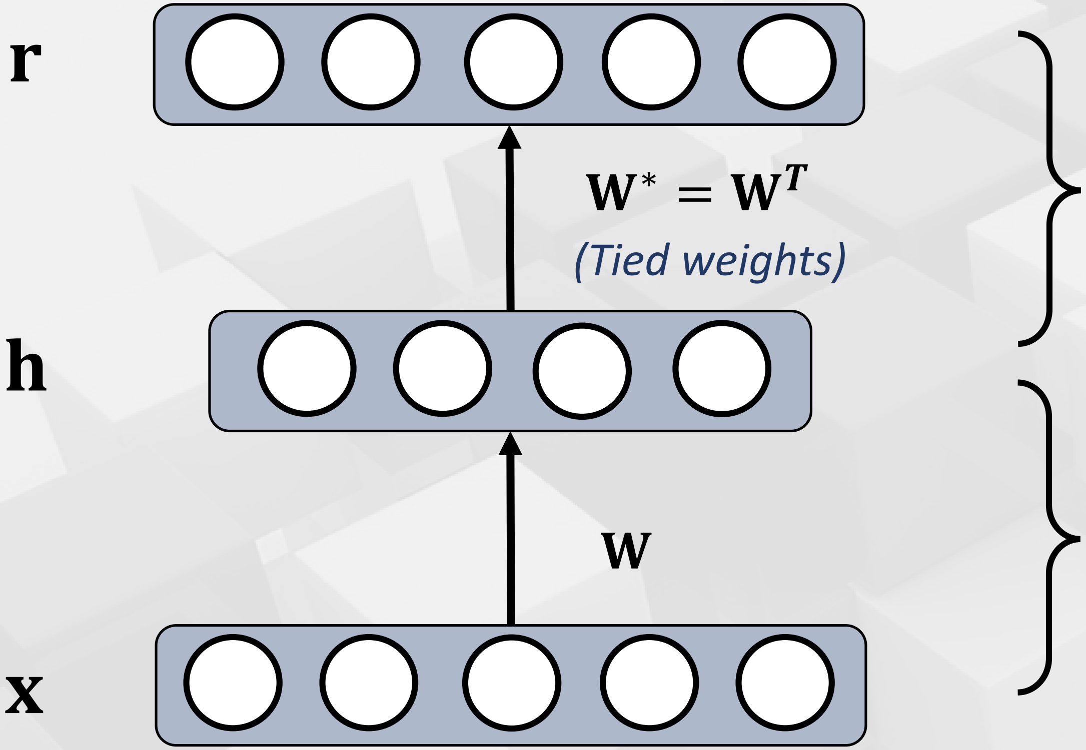
Autoencoders

Shilin HE

Autoencoders

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



Decoder

$$\mathbf{r} = g(\mathbf{x})$$
$$= \sigma(\mathbf{W}^*\mathbf{h} + \mathbf{b})$$

Encoder

$$\mathbf{h} = f(\mathbf{x})$$
$$= \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

Unsupervised learning: only use the input X for learning

Autoencoders

Loss function:

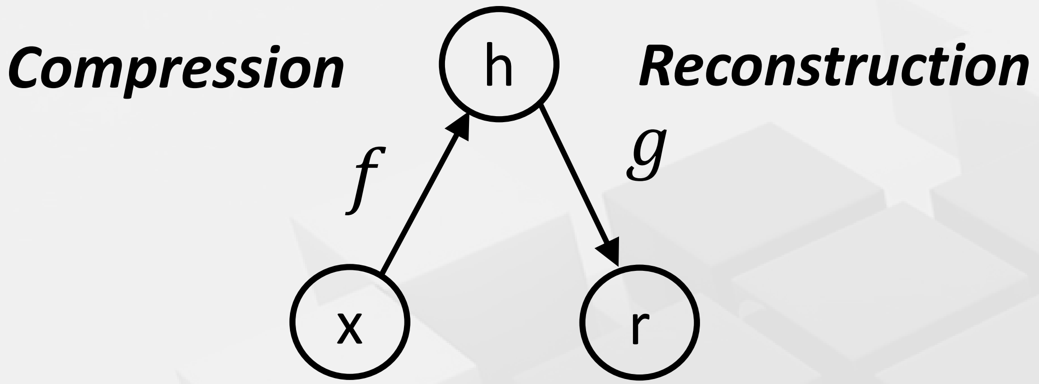
$$\textbf{Min } L(x, g(f(x)))$$

L is a loss function penalizing g(f(x)) for being dissimilar from x, e.g., mean squared error.

Train with backpropagation

When computing gradients with tied weights ($\mathbf{W}^* = \mathbf{W}^{\boldsymbol{T}}$), $\nabla_{\mathrm{W}} L(x, g(f(x)))$ is the sum of two gradients!
-- because **W** is present in the encoder **and** in the decoder

Autoencoders



General structure:

- Encoder f: mapping x to h
- Decoder g: mapping h to r

Autoencoders may learn identity function precisely: $g(f(x)) = x$

$\Rightarrow$ Not useful!

Need to constrain complexity:

- By architectural constraint
- Penalty on internal representation

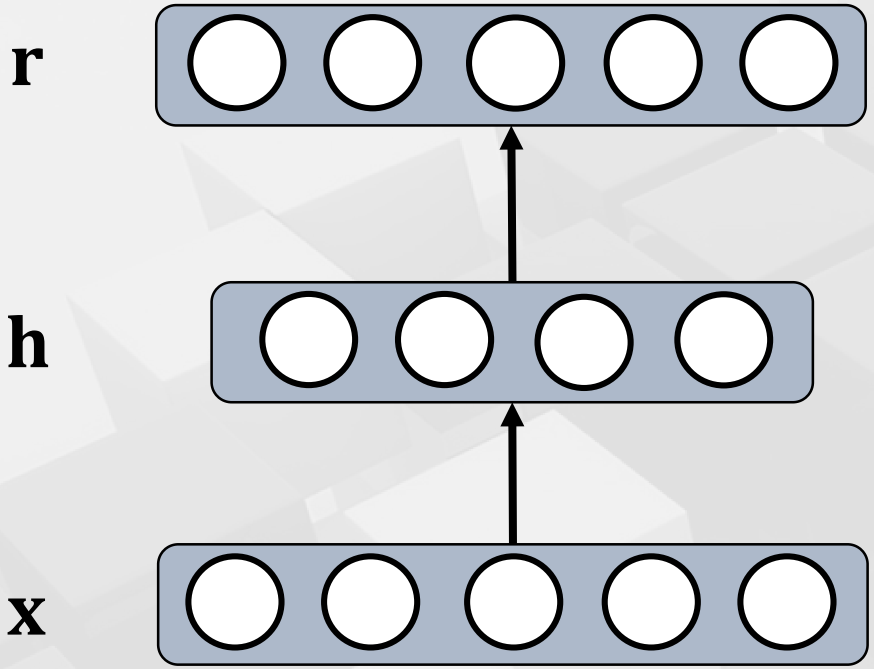
Autoencoders

Autoencoder types:

- Undercomplete Autoencoders → architectural constraint
- Regularized Autoencoders
- Sparse Autoencoders
- Denoising Autoencoders
- Contractive Autoencoders
- …

} Penalty on internal representation (regularized autoencoders)

Undercomplete Autoencoders



Constraint: Dimension of **h** is smaller than **x**

$x \in \mathbb{R}^D, h \in \mathbb{R}^K$

Undercomplete autoencoders if $K < D$

Capture the most salient features

Undercomplete Autoencoders

Undercomplete autoencoders with:

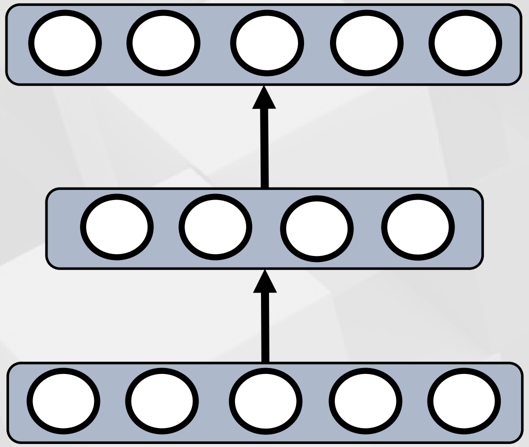
✓ Decoder is linear transformation

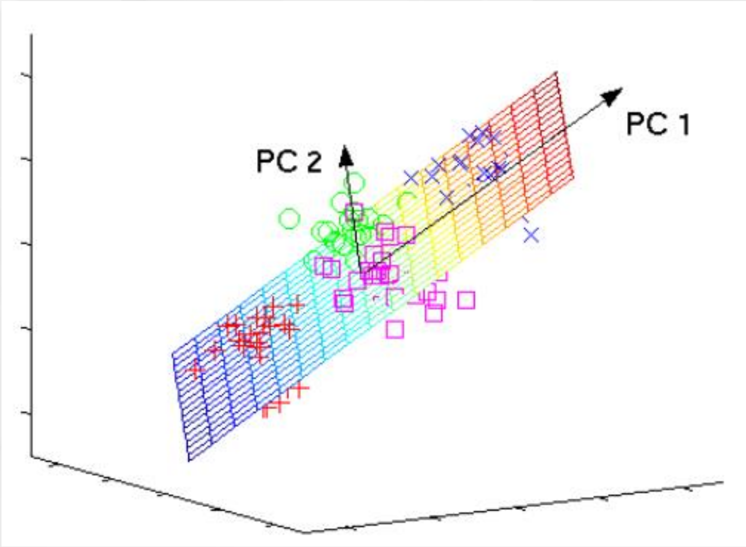
✓ Loss L is mean square error (MSE)

can learn the same subspace as PCA

In this process, two tasks are accomplished:

1. Copy the input to output
2. Learn the principal subspace of training data as a **side-effect**

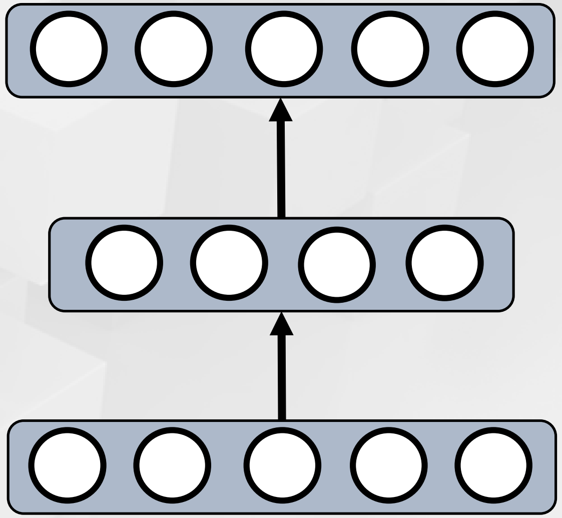




Undercomplete Autoencoders

If the encoder and decoder functions (f, g) are nonlinear,

$\Rightarrow$ A more powerful **nonlinear generalization** of PCA



However,

Too large capacity of encoder and decoder

$\Rightarrow$ can perform the copying task well, but fail to capture useful information on dataset

Regularized Autoencoders

$x \in \mathbb{R}^D, h \in \mathbb{R}^K$

What if $K > D$? => **Overcomplete Autoencoders**

Regularized Autoencoders use a loss function that encourages the model to have some properties besides reproducing inputs:

- Sparsity representation (Sparse Autoencoders)
- Smallness of derivative of representation (Contractive Autoencoders)
- Robustness to noise or to missing inputs (Denoising Autoencoders)

Sparse Autoencoders

$$L\big(x, g\big(f(x)\big)\big) + \Omega(h)$$

Loss for copying inputs

Sparsity penalty

Sparse Autoencoders

In general neural network, we are trying to find the **maximum likelihood:** $p(x|\theta)$

To do the maximum likelihood estimation (MLE), we often use the $\log\big(p(x|\theta)\big)$ for simplification, from which we can get the loss function without regularization.

What about MAP (Maximum a posterior)?

$$p(\theta|x) \propto p(x|\theta) * p(\theta)$$

Posterior Likelihood Prior

$$\max \log\big(p(\theta|x)\big) => \max \{ \log\big(p(x|\theta)\big) + \log(p(\theta)) \}$$

Loss function Regularization penalty

Sparse Autoencoders

$$\max \log\big(p(\theta|x)\big) \Rightarrow \max \{ \log\big(p(x|\theta)\big) + \log(p(\theta)) \}$$

What will happen if $p(\theta)$ follows the **Gaussian Distribution**?

Consider the linear regression model, if

$$p(x) = \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

$$\omega \sim \mathcal{N}(\mathbf{0}, \frac{1}{\lambda}\boldsymbol{I})$$

$$p(w) = \frac{1}{\sqrt{|2\pi\frac{1}{\lambda}I|}} e^{-\frac{1}{2}w^T\lambda I w}$$

$$\Rightarrow \log p(w) = \log\frac{1}{\sqrt{|2\pi\frac{1}{\lambda}I|}} - \frac{\lambda}{2}w^T w$$

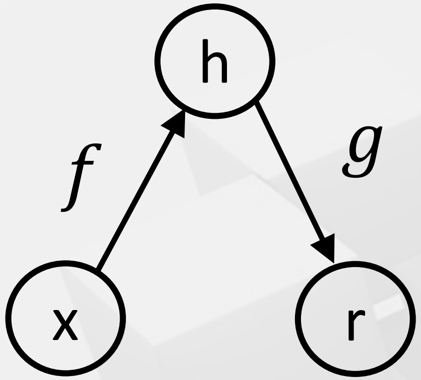
→ L2 Norm

Gaussian Prior => L2 Norm

Similarly, Laplace Prior = > L1 Norm

Sparse Autoencoders

How to get the sparse penalty in sparse autoencoders?



Set the distribution over latent variable h

The joint distribution of h and x is given as:

$$p_{model}(x, h) = p_{model}(h) p_{model}(x|h)$$

$$\log p_{model}(x, h) = \boxed{\log p_{model}(h)} + \log p_{model}(x|h)$$

Sparse penalty

Sparse Autoencoders

$$L(x, g(f(x))) + \Omega(h)$$

Loss for copying inputs — Sparsity penalty

Our target becomes:
Find a distribution of h which can has the characteristic of sparsity

Which distribution?

=> **Laplace distribution!**

$$\log p_{model}(x, h) = \log p_{model}(h) + \log p_{model}(x|h)$$

Sparse penalty

Sparse Autoencoders

Laplace distribution:

$$p(x) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

$$p_{model}(h_i) = \frac{\lambda}{2} e^{-\lambda |h_i|}$$

$$-\log p_{model}(h) = \sum_i (\lambda |h_i| - \log \frac{\lambda}{2})$$

L1 Norm $\Omega(h)$ Constant