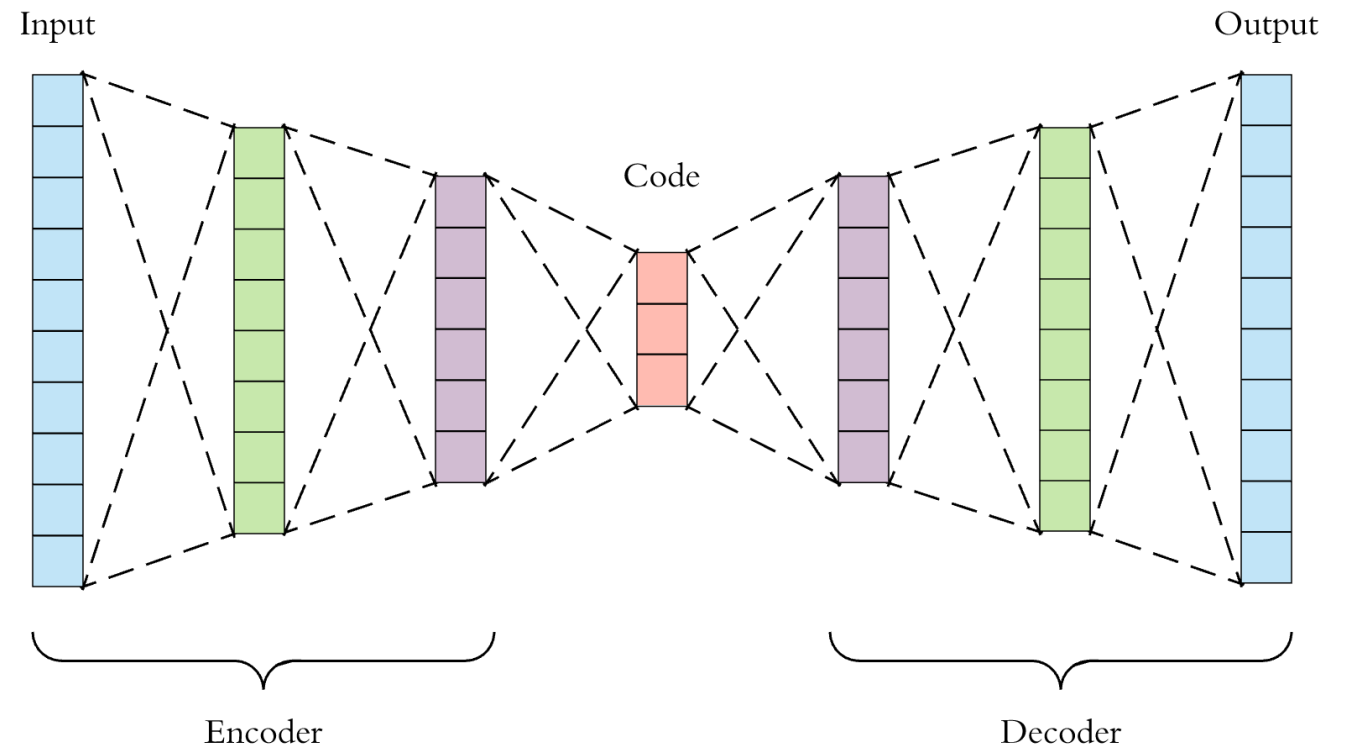
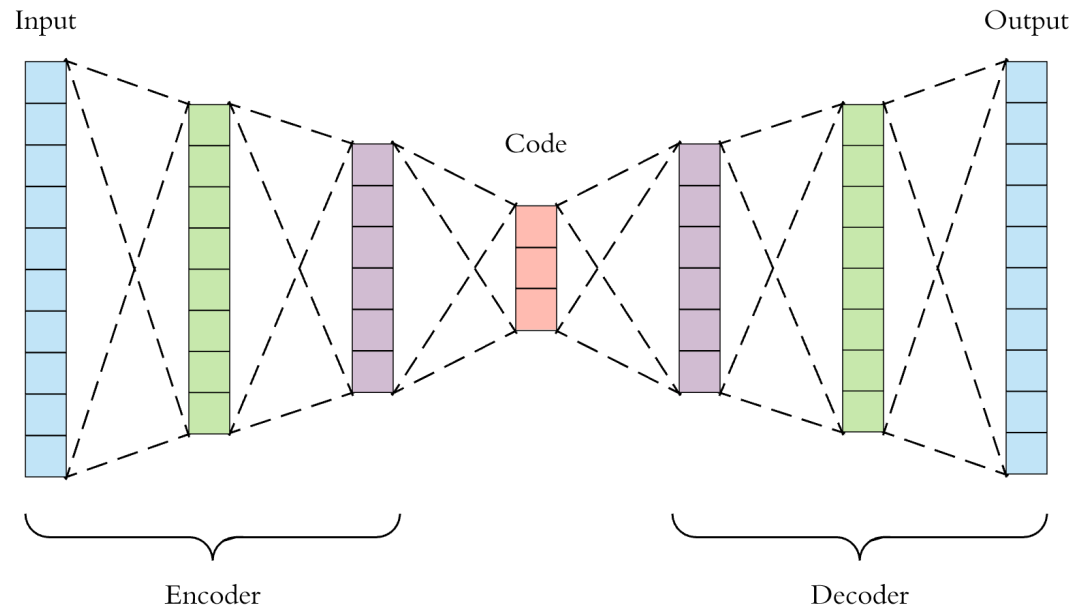


Autoencoders



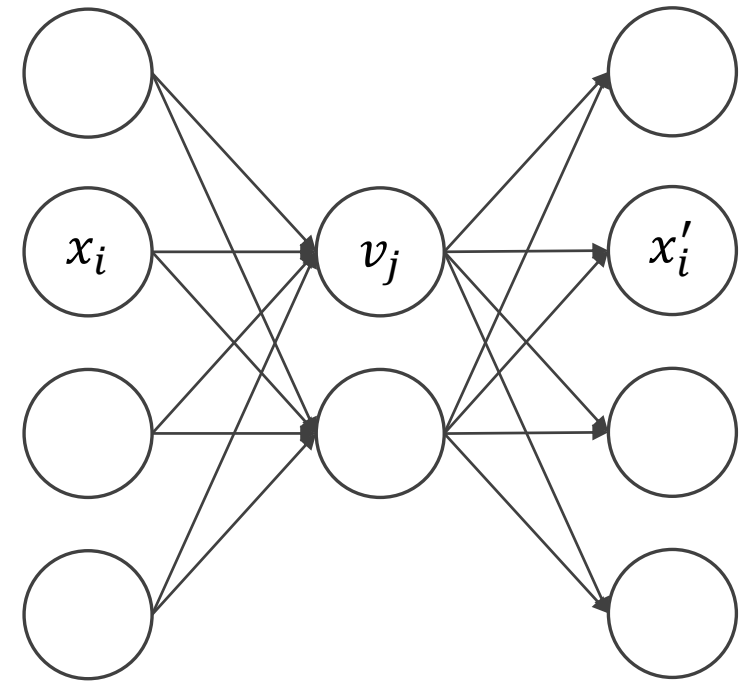
Autoencoders

- Feedforward networks that
 - Have a bottleneck layer with fewer dimensions than the input
 - With an output layer with the same dimensionality as the input
 - And as objective the output to reconstruct the input



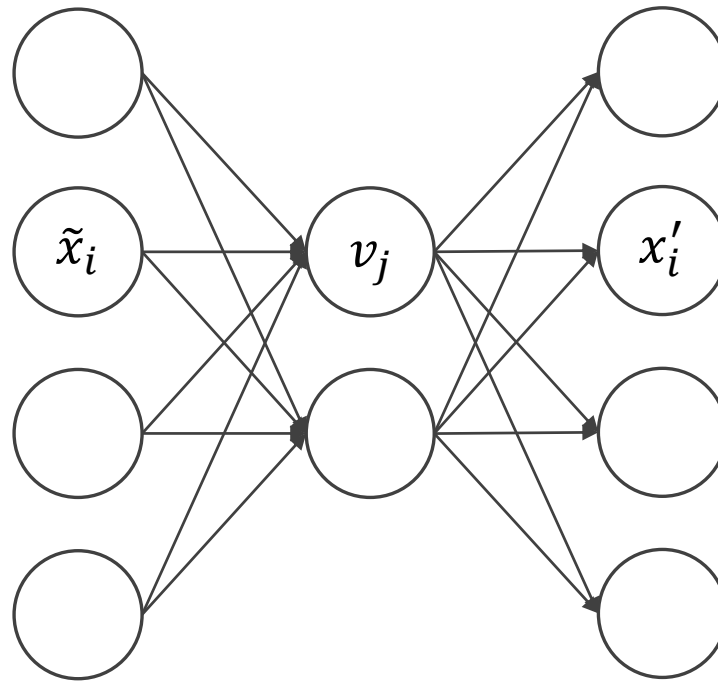
Autoencoders

- There is an encoder network: $\mathbf{v} = h(\mathbf{W}\mathbf{x})$
- And a decoder network: $\mathbf{x}' = h(\mathbf{W}'\mathbf{v})$
- The weights are often tied together: $\mathbf{W}^T = \mathbf{W}'$
- If there is no bottleneck and no regularization
 - \rightarrow no learning
 - The input can be simply copied to the output



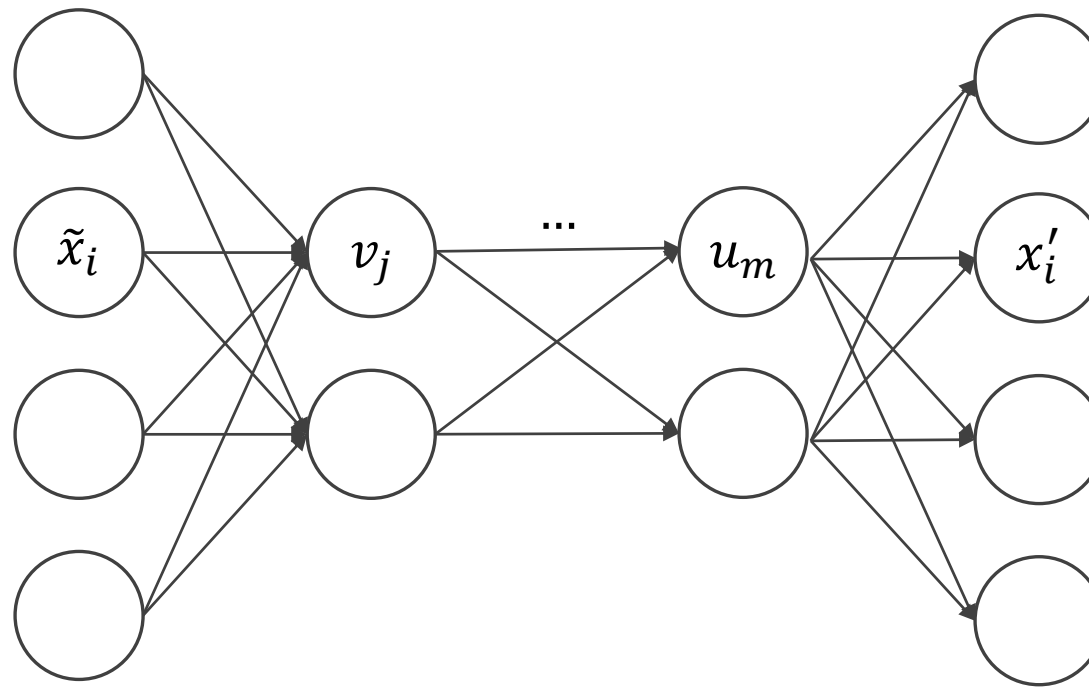
Denoising autoencoders

- Add some noise to the input: $\tilde{x} = x + \varepsilon$
- Then ask the autoencoder to reconstruct the original input x regardless
- Learns more generalizable embeddings



Deep autoencoders

- Neural networks that learn to embed data on some form of nonlinear manifold
- Compared to PCA, they can learn better embeddings
 - Due to the nonlinearities



Applications

- Autoencoders are used for visualization
- Autoencoders are used to learn compression
 - Similar to PCA
 - In fact, linear autoencoders learn the same subspace as PCA
- Autoencoders can be used for pre-training
 - No need for labelled data
- Deep autoencoders are not probabilistic and not generative models
 - They can reconstruct the input
 - They cannot generate new data points

[For more nice insights check R. Grosse's slides](#)