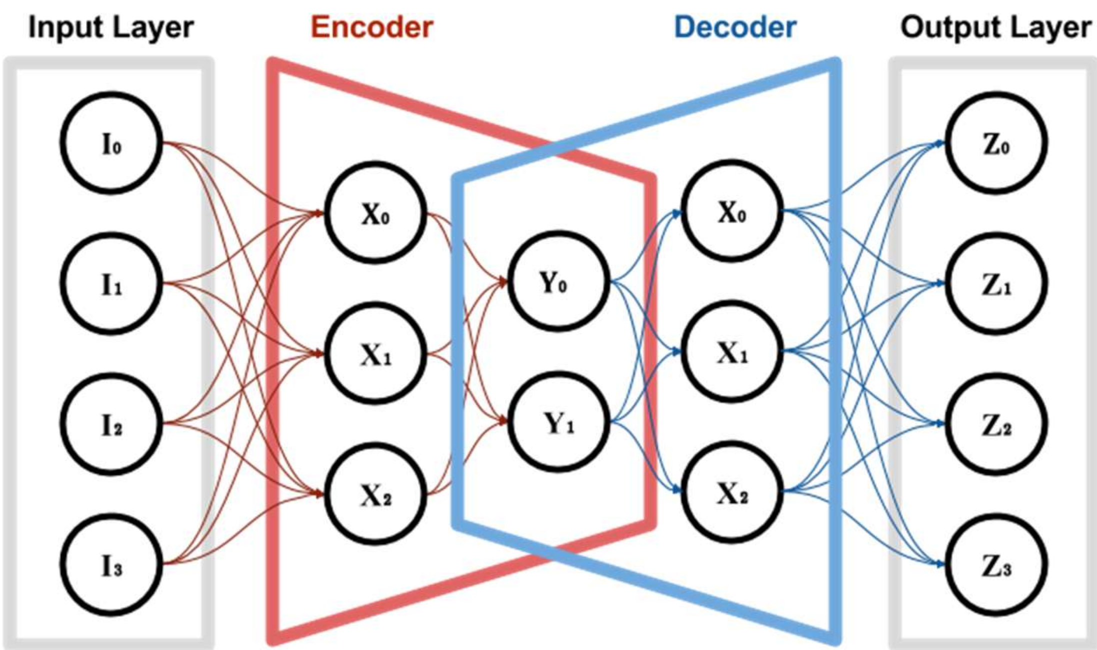
Lecture 21: Autoencoder

Course Teacher: Md. Shariful Islam Bhuyan

Autoencoder

- *Using neural network for self-encoding in latent space (bottleneck)*

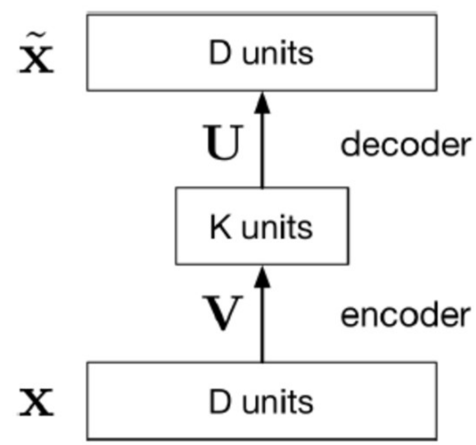


Benefits

- Map high-dimensional data to two dimensions for visualization
- Compression (i.e. reducing the file size)
- Learn abstract features in an unsupervised way
- Unlabeled data can be much more plentiful than labeled data
- Dimensionality reduction, PCA?

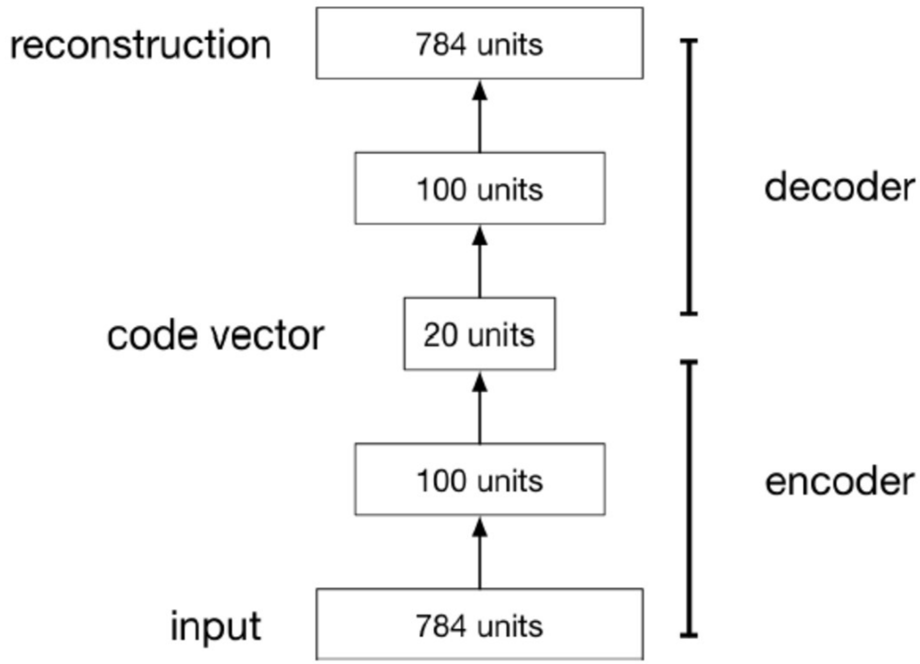
PCA: Simplest Autoencoder

- One hidden layer
- Linear activation
- Minimize squared error loss
- $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2$
- $\hat{\mathbf{x}} = \mathbf{U}\mathbf{V}\mathbf{x}$



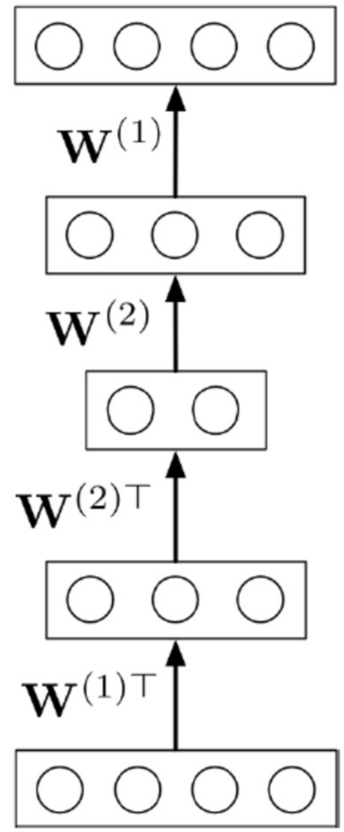
Deep Autoencoders

- Nonlinear activation function
- Nonlinear dimensionality reduction
- Stacked autoencoder
- Denoising autoencoder
- Regularized autoencoder
- Sparse autoencoder
- Convolutional autoencoder



Training

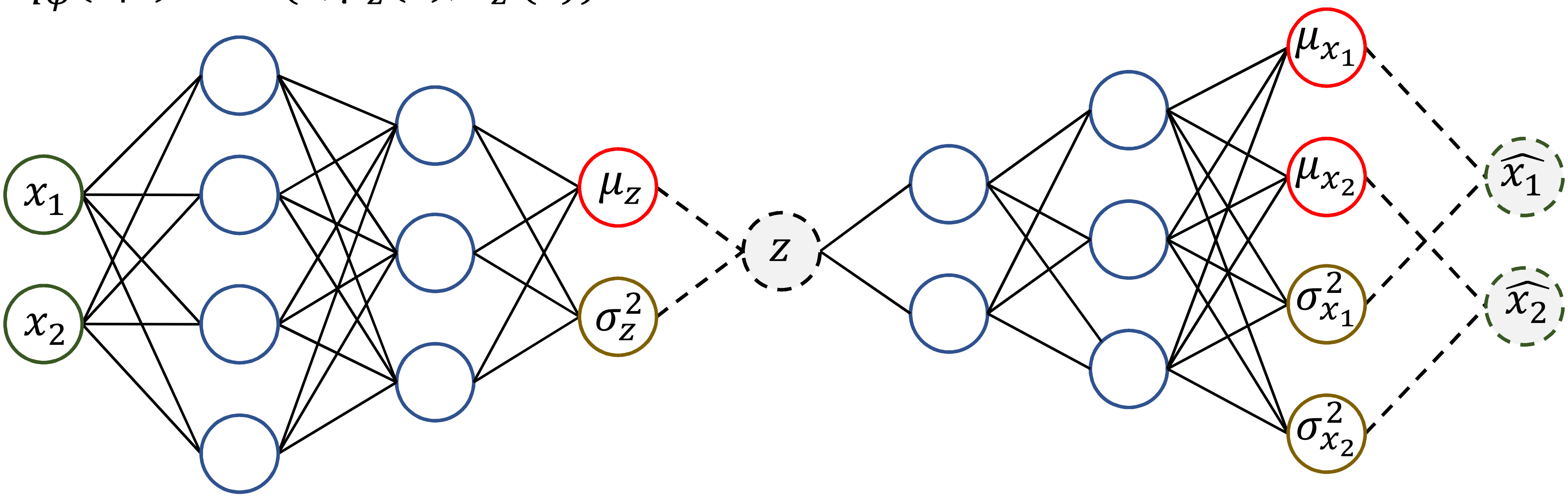
- Greedy layer-wise pretraining
- Pretrained initialization
- Fine-tuning with backpropagation
- Fine-tuning for classification (drop decoder)



Variational Autoencoder

$$q_\phi(z|\mathbf{x}) = \mathcal{N}(z; \mu_z(\mathbf{x}), \sigma_z^2(\mathbf{x}))$$

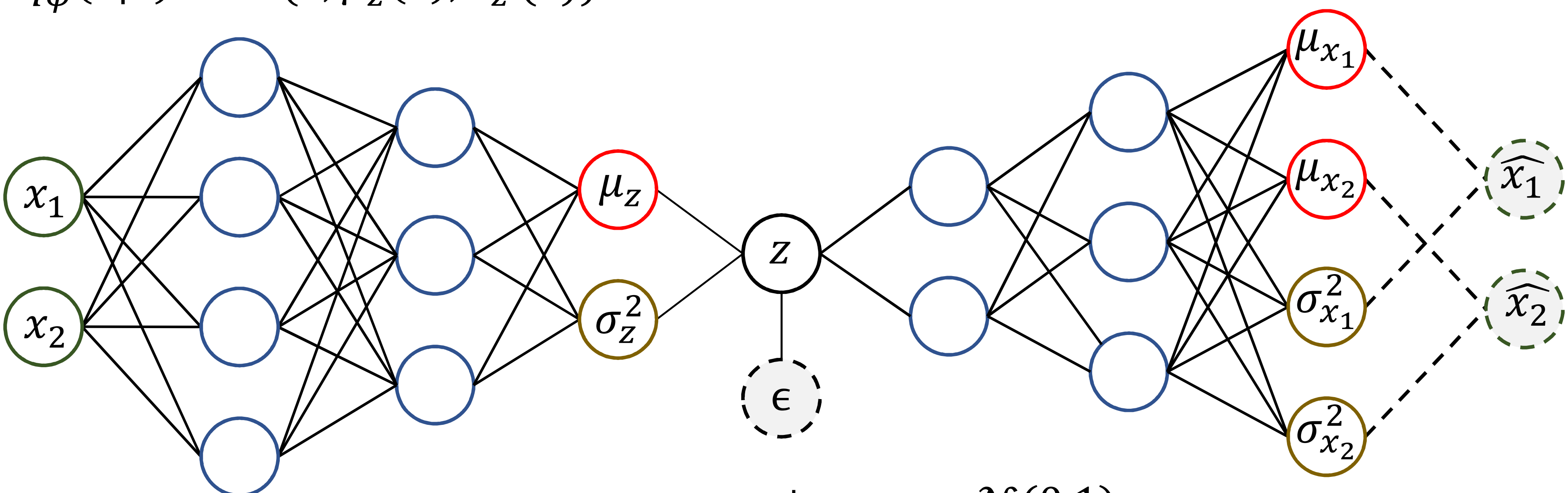
$$p_\theta(\mathbf{x}|z) = \mathcal{N}(\mathbf{x}; \mu_\mathbf{x}(z), \sigma_\mathbf{x}^2(z))$$



Reparameterization Trick for Back-Propagation

$$q_\phi(z|\mathbf{x}) = \mathcal{N}(z; \mu_z(\mathbf{x}), \sigma_z^2(\mathbf{x}))$$

$$p_\theta(\mathbf{x}|z) = \mathcal{N}(\mathbf{x}; \mu_\mathbf{x}(z), \sigma_\mathbf{x}^2(z))$$



$$z = \mu_\mathbf{z} + \epsilon\sigma_z; \ \epsilon \sim \mathcal{N}(0,1)$$