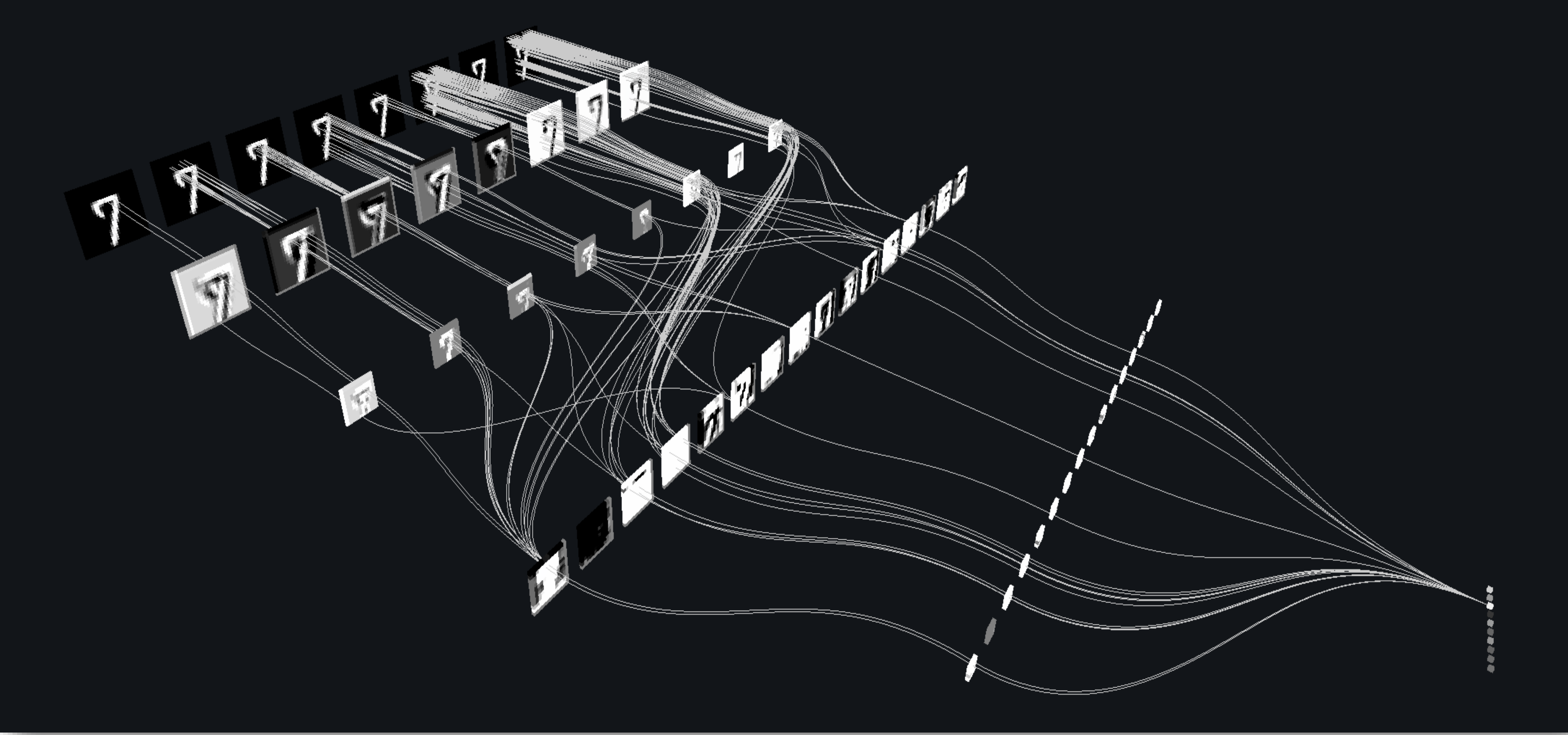
Deep Learning

Md. Jalil Piran, PhD
Asst. Professor
Computer Science and Engineering
Sejong University
Spring, 2021

Outline

- Introduction to Deep Learning (DL)
- The History of DL
- Programming Tools
- Artificial Neural Networks (ANNs)
- Optimization in DL
- Convolutional Neural networks (CNNs)
- **Unsupervised Pre-trained Networks (UPNs)**



ARCHITECTURE OF DEEP LEARNING

DL Architectures

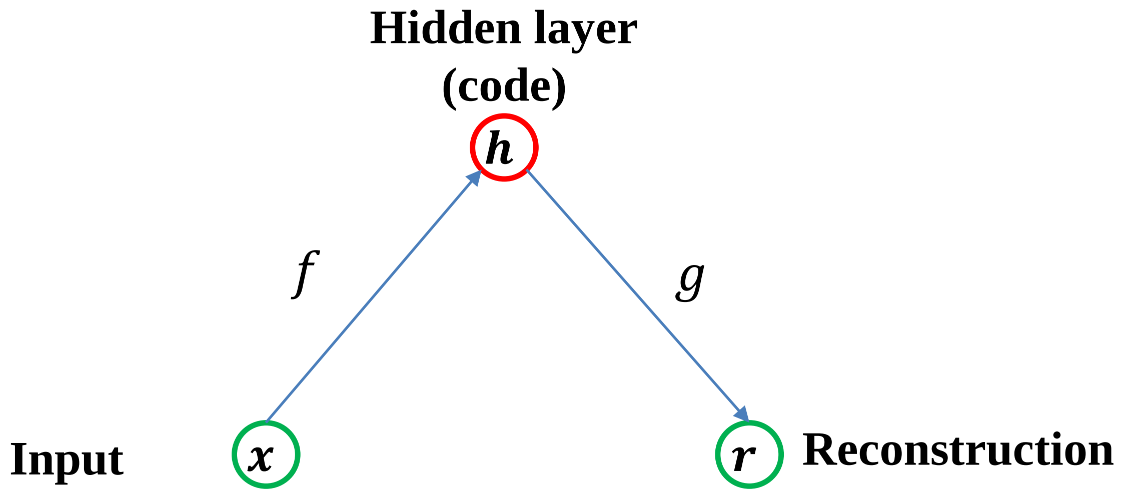
- **Higher-level Architecture**
  - **Convolutional Neural Networks (CNNs)**
  - **Unsupervised Pre-trained Networks (UPNs)**
    - Deep belief networks (DBNs)
    - Autoencoders (AE)
    - Generative adversarial networks (GANs)
  - **Recurrent Neural Networks (RNNs)**
    - Bidirectional recurrent neural networks (BRNN)
    - LSTM
  - **Recursive Neural Networks**

Topics

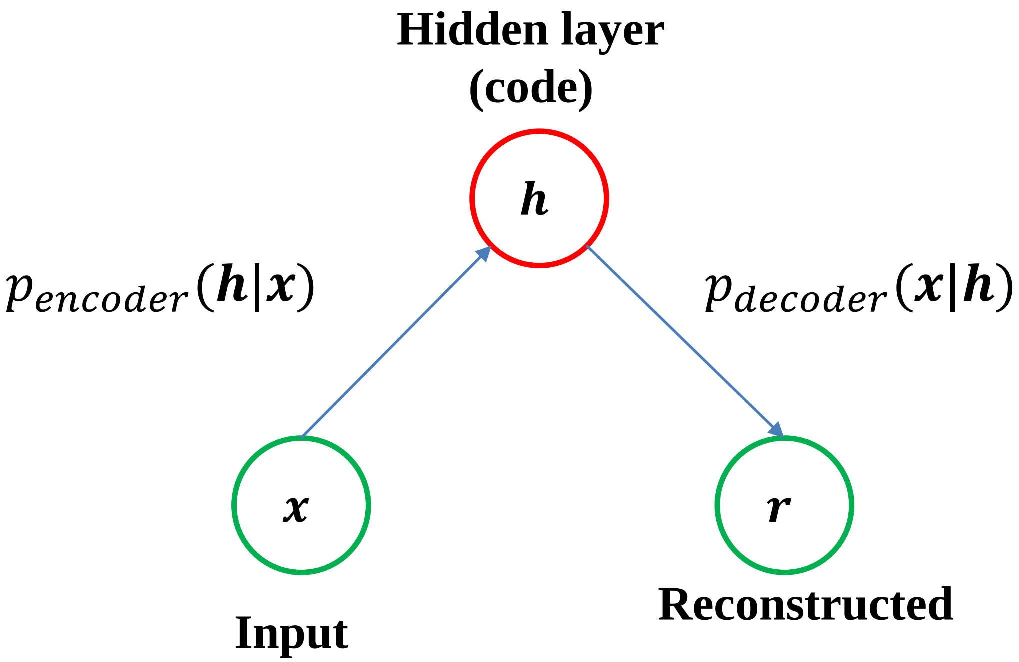
- Introduction
- Sparse AE
- Denoising AE
- Contractive AE
- Applications

Introduction

- **Autoencoder (AE)**
  - A type of artificial neural networks
  - Trained to copy its input to its output
  - Components:
    - **Encoder**: $\boldsymbol{h} = f(\boldsymbol{x})$
    - **Decoder**: $\boldsymbol{r} = g(\boldsymbol{h})$

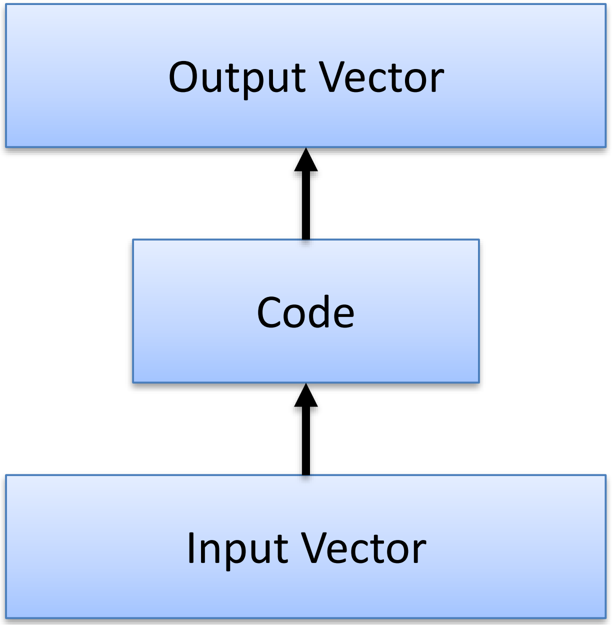


- **Modern AE**
  - Deterministic functions to stochastic mappings



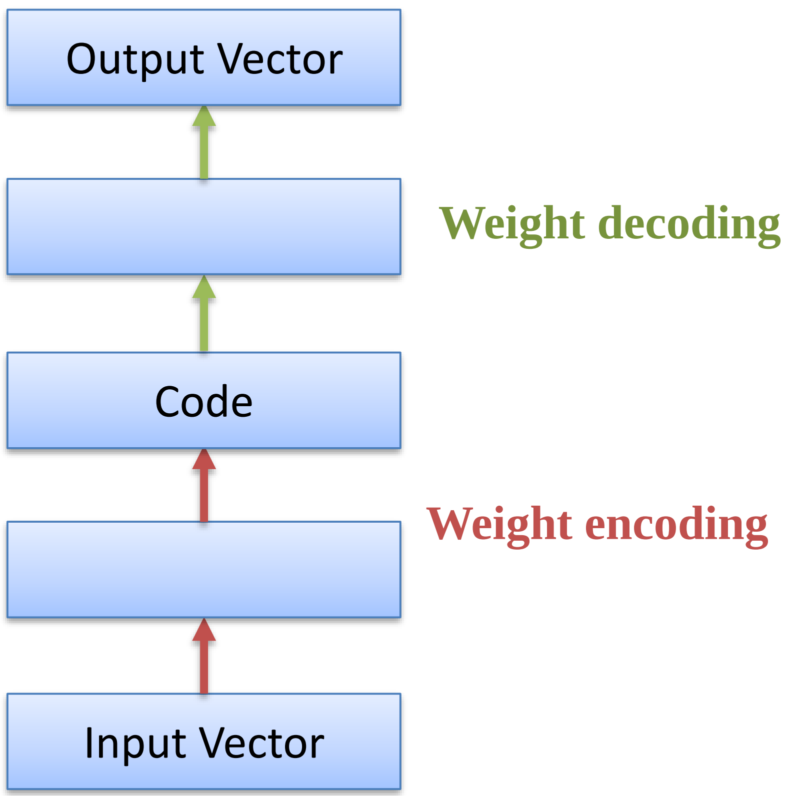
Introduction

- **AE vs. PCA**
  - Try to make the output be the same as the input in a network with a central bottleneck.
  - If the hidden and output layers are linear, it will learn hidden units that are a linear function of the data and minimize the squared reconstruction error.
    - This is exactly the functionality of PCA
    - Their weight vectors may not be orthogonal



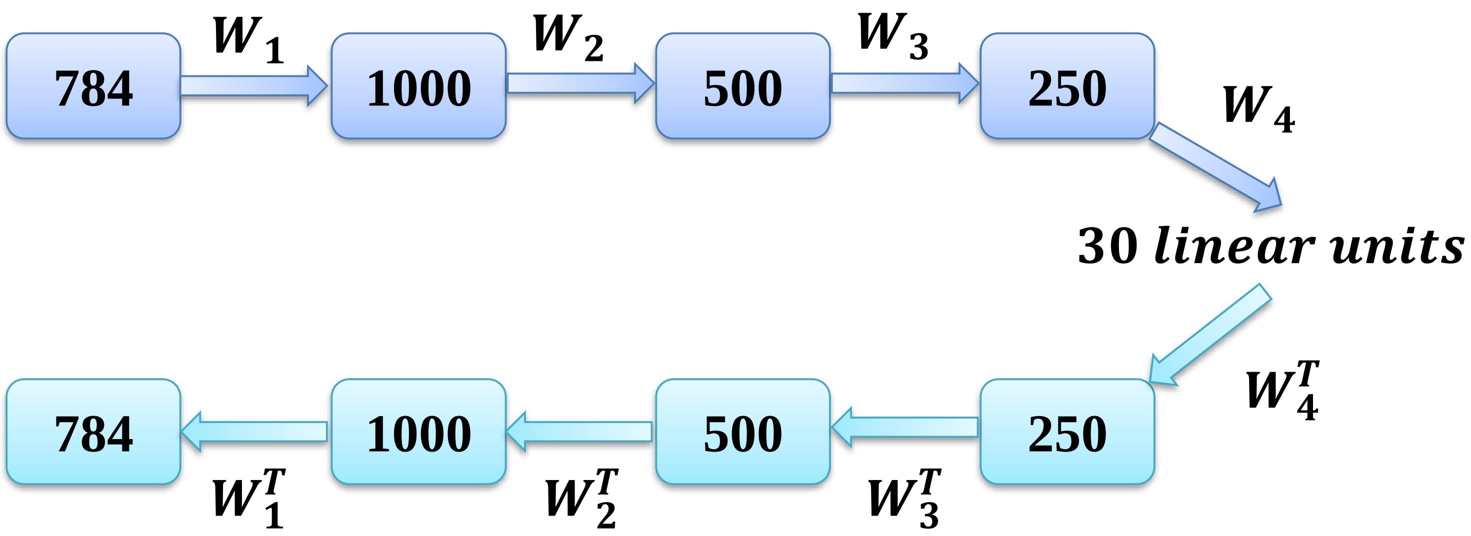
Introduction

- **AE vs. PCA**
  - With non-linear layers before and after the code, it should be possible to efficiently represent data that lies on or near nonlinear **manifold**.
    - The encoder converts coordinates in the input space to coordinates on the manifold.
    - The decoder does the inverse mapping



Deep AE

- Very difficult to optimize deep AE using backpropagation.
  - With small initial weights the backpropagated gradient vanishes.
- 2006: Prof. Hinton applied RBMs for AEs
  - Train a stack of 4 RBMs and then ‘unroll’ them.
  - Then, fine-tune with gentle backpropagation.



Types of AE

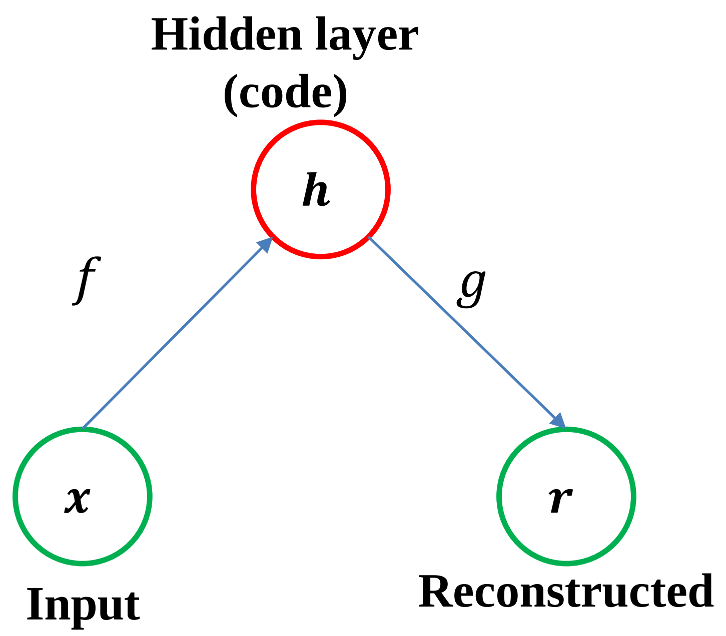
- Types:

1. **Undercomplete AEs:**
    - The dimensions in the code layer is less than the input.
2. **Overcomplete AEs:**
    - The dimensions in the code layer is more than the input.

Types of AE

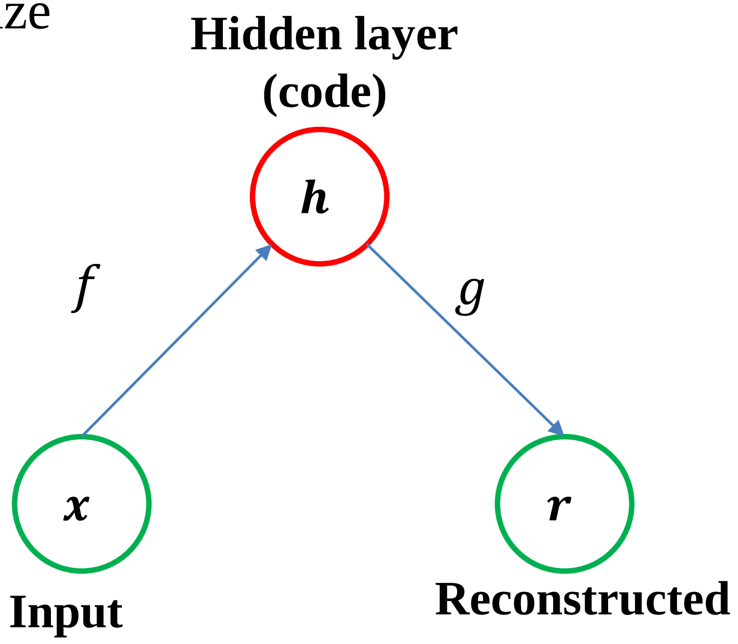
1. Undercomplete AE

- h has lower dimensions than x
- Forces the AE to capture the most salient features
- **Loss function**: $L\left(x, g\left(f(\boldsymbol{x})\right)\right)$
- **Encoder and decoder function**
  - **Linear**: low capacity (learns to span the same subspace as PCA)
  - **Non-linear**: more powerful
    - Problem: copying task with extracting useful information
  - Must discard some information in $\boldsymbol{h}$



2. Overcomplete AE

- $\boldsymbol{h}$ has higher dimensions than $\boldsymbol{x}$
- Problem: Can learn to copy the input to the output without learning anything
- **Solution**:
  - Keeping the encoder and decoder shallow with a small code size
  - **Regularized AE**



Regularized AE

- Methods of **regularizations**:
  - **Sparse AE**
  - **Denoising AE**
  - **Contractive AE**

Sparse AE

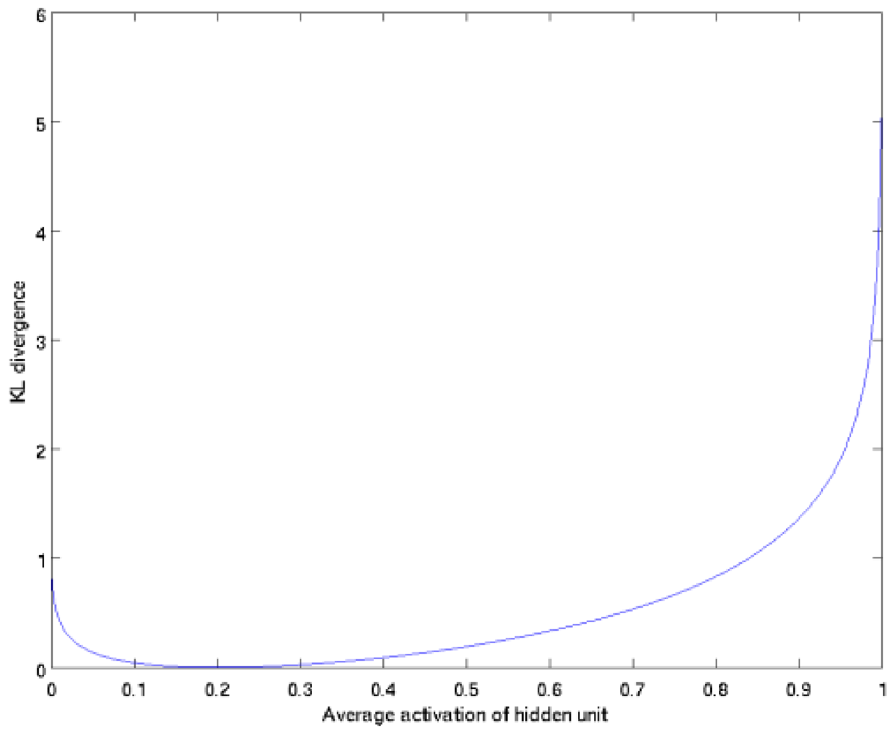
- Limit capacity of AE by adding a term to the cost function:

$$L\left(\boldsymbol{x}, g\left(f(\boldsymbol{x})\right)\right) + \Omega(\boldsymbol{h})$$

  - $\Omega(\boldsymbol{h})$: Kullback-Leibler
  - Constrain the neurons to be active.
- Typically used to learn features for another task such as **classification**

$$J_{sparse}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} KL(\rho \parallel \hat{\rho}_j)$$

- $\hat{\rho}_j$: the average activation of hidden unit j
- ρ: sparsity parameter, e.g. enforce the constraint
  - Normally close to zero ~0.05

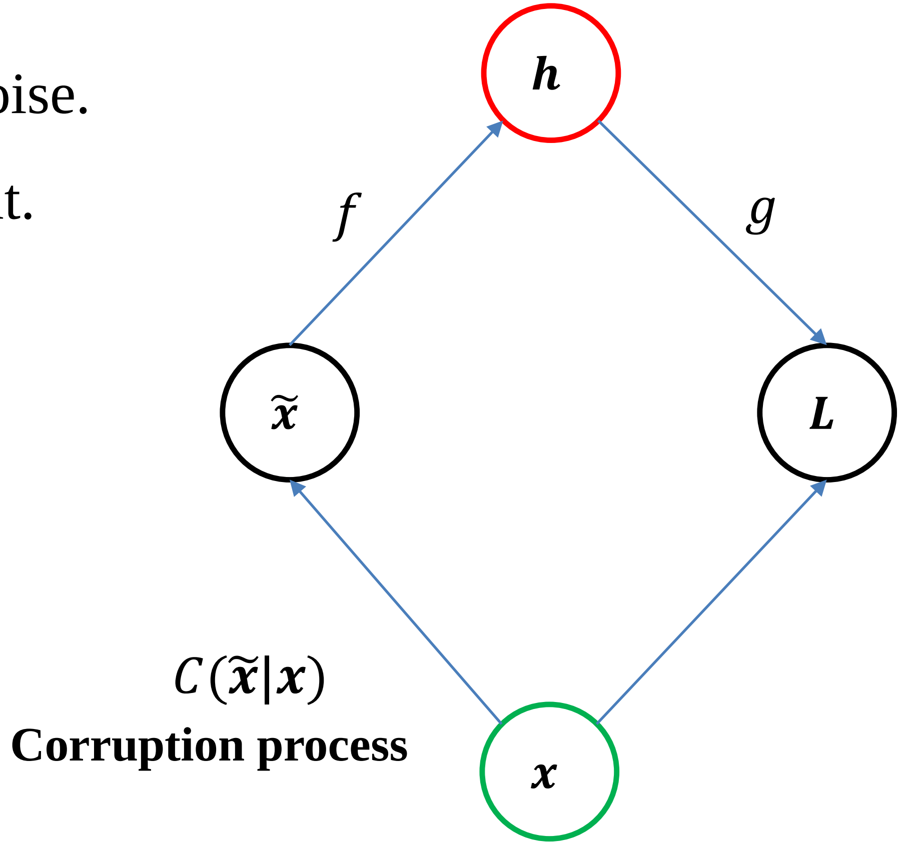


Denoising AE

- **Loss function**

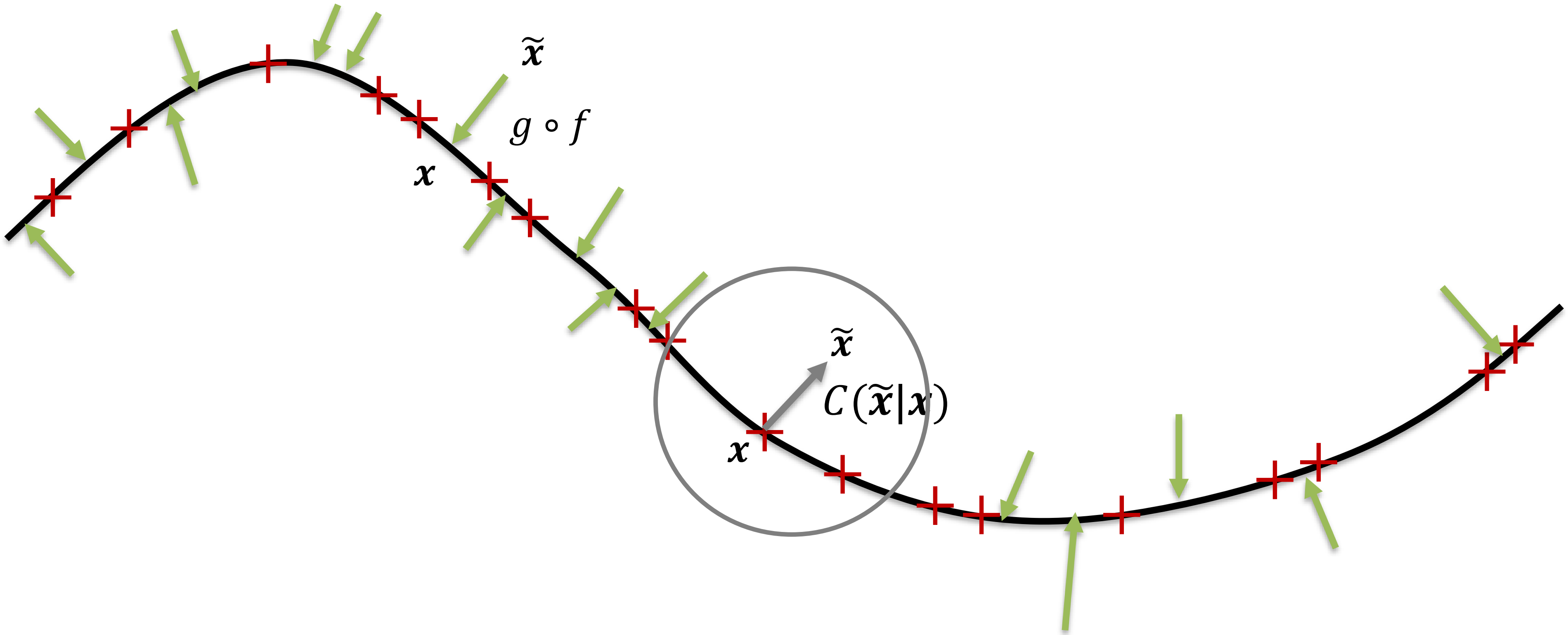
$$L\left(\boldsymbol{x}, g\left(f\left(\widetilde{\boldsymbol{x}}\right)\right)\right)$$

  - $\widetilde{\boldsymbol{x}}$: a copy of $\boldsymbol{x}$ that has been corrupted by some form of noise.
  - Must undo corruption rather than simply copying the input.



Denoising AE

- **Learn a manifold**
  - A denoising AE is trained to map a corrupted data point $\tilde{\boldsymbol{x}}$ back to the original data point $\boldsymbol{x}$



Contractive AE

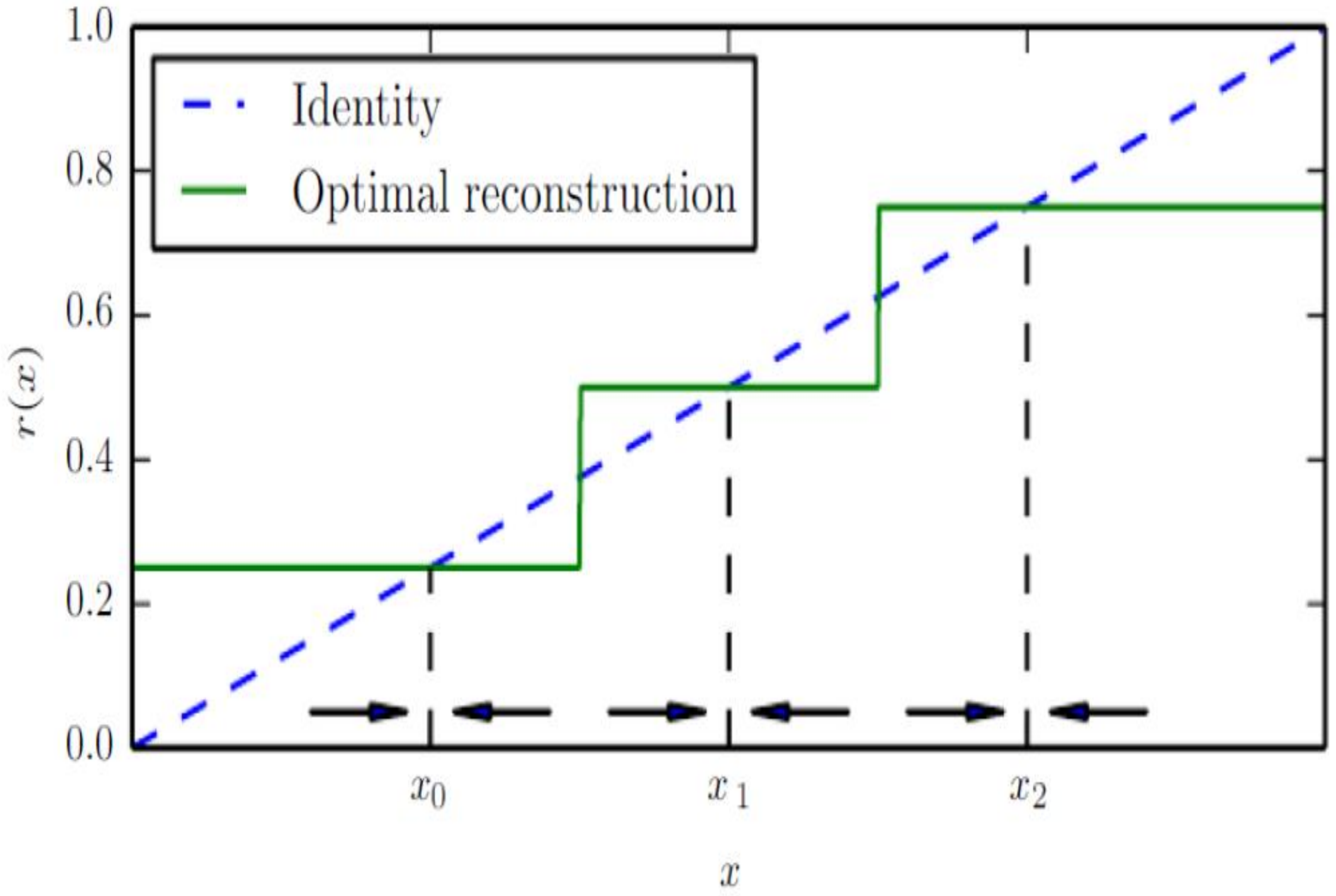
- Encouraging the derivative of f to be as small as possible

$$\Omega(\boldsymbol{h}) = \lambda \left\| \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\|_F^2$$

- Make the feature extraction function resist infinitesimal perturbations of the input.
- Contracting the input neighborhood to smaller output neighborhood.

Contractive AE

- Derivation of the reconstruction function around the data points.



Applications of AE

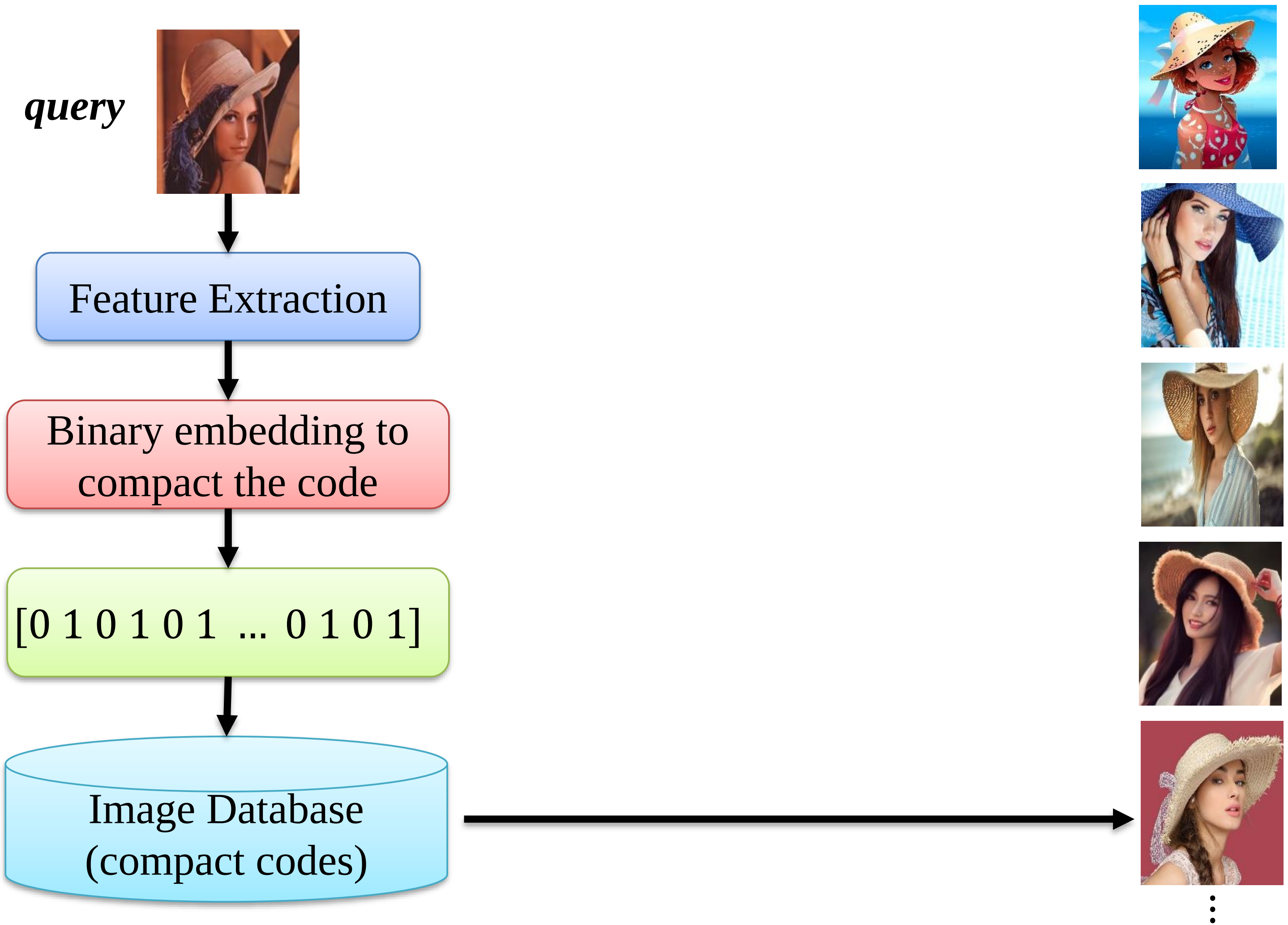
1. **Dimensionality reduction**
   - Lower-dimensional representation can improve performance of many tasks
   - Less memory
   - Cost efficient
   - Time efficient

2. **Information retrieval**

Applications

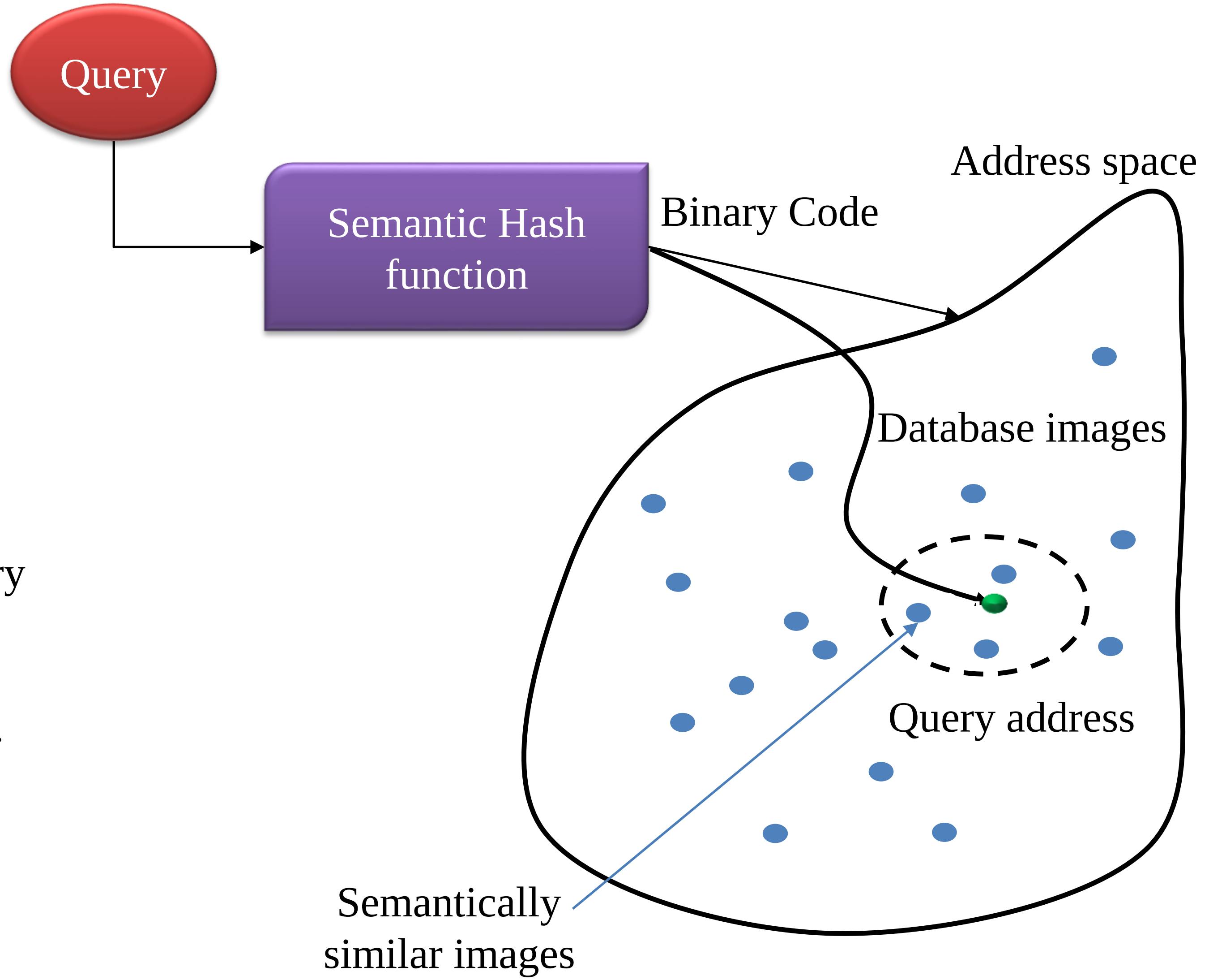
Information retrieval

- Find entries in a database in response to a query.



Applications

- **Information retrieval**
  - **Coding Techniques**
    - **Semantic hashing**
      - Low dimensional and binary codes
      - Store all database entries in a hash table
      - Information retrieval by returning all database entries that have the same binary code as the query
      - Can be used for both textual and images.

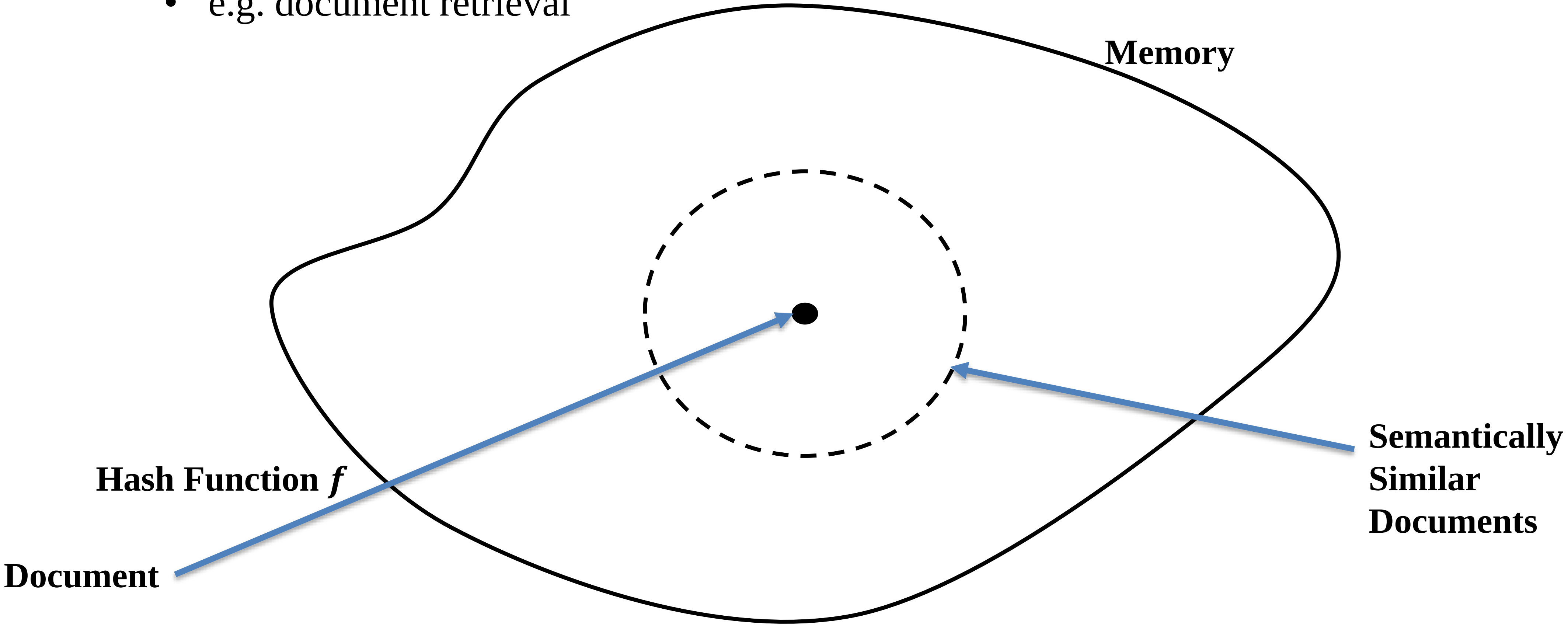


Applications

- **Information retrieval**
  - **Coding Techniques**
    - **Semantic hashing**
      - How to generate those binary codes?
      - Set Sigmoid on the final layer
      - Sigmoid units must be trained to be saturated to nearly 0 or nearly 1 for all input values
        - Inject additive noise just before the sigmoid nonlinearity during the training
        - The magnitude of the noise should increase over time
        - To overcome the noise, the network must increase the magnitude of the inputs to the sigmoid function, until saturation occurs.

Applications

- **Semantic hashing**
  - Deep AE as a hash function to find approximate matches.
    - e.g. document retrieval



Applications

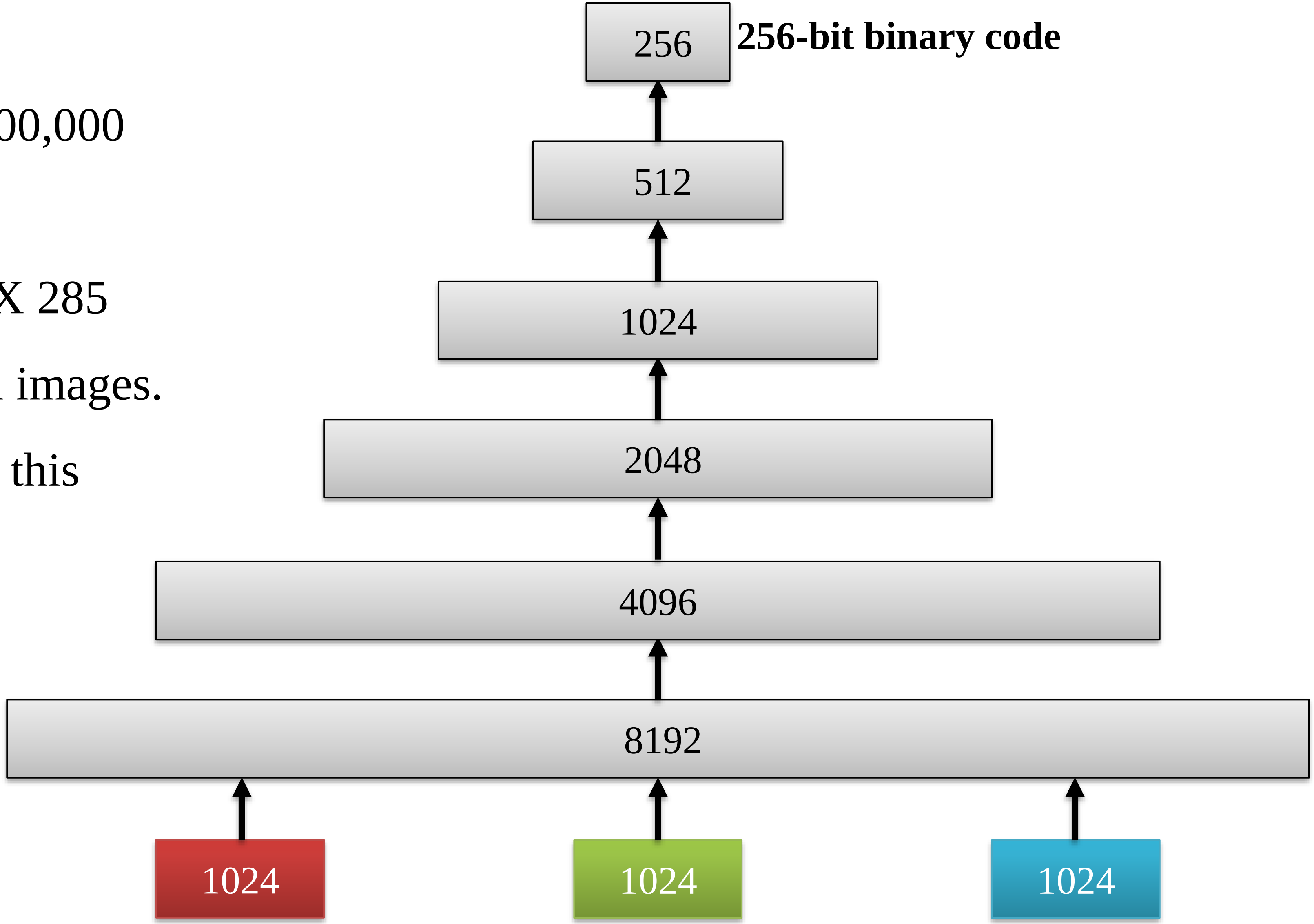
- **Binary codes for image retrieval**
  - Image retrieval is typically done by using the captions and not the images.
    - Unlike words; individual pixels do not tell us much about the content
  - We may extract a real-valued vector that contains information about the content.
    - Matching real-valued vectors in a big database is slow and requires a lot of storage.
  - **Short binary** codes are very easy to store and match.

Applications

- **Binary codes for image retrieval**
  - A **two-stage method**
    - First, generate a semantic hash with 28-bit binary codes to get a long ‘shortlist’ of promising images
    - Then, use 256-bit binary codes to do a serial search for good matches.
      - This only requires a few words of storage per image and the serial search can be done using fast bit-operations.
    - But, how good are the 256-bit binary codes?
      - Do they find the desired images?

Applications

- **Krizhevsky's Deep AE**
  - The encoder has about 67,000,000 parameters.
  - It takes a few days on a GTX 285 GPU to train on two million images.
  - There is no theory to justify this architecture.

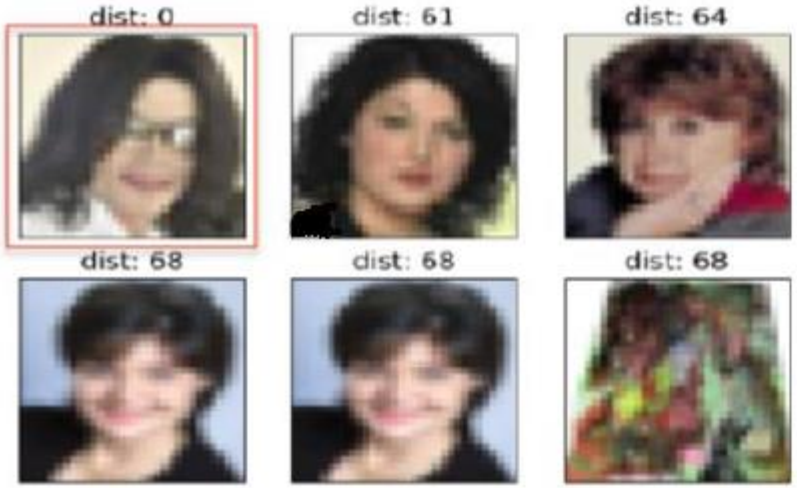


Applications

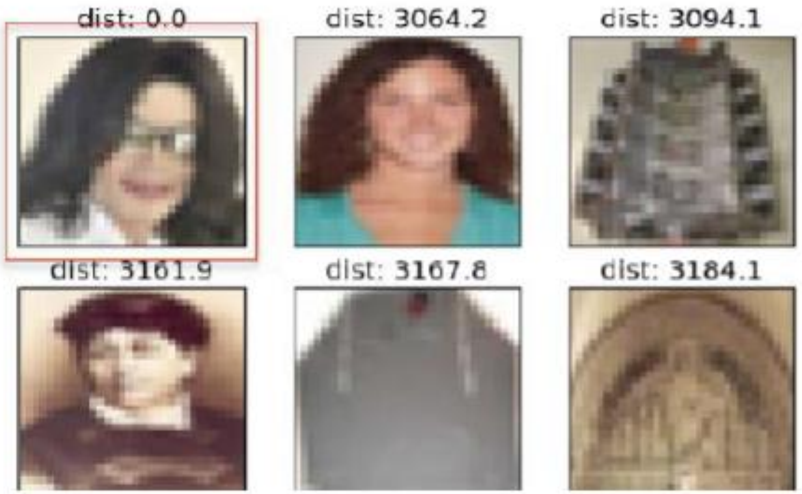
- Reconstruction of 32×32 color image from 256-bit codes.

Applications

- Retrieved images using 256-bit codes.



- Retrieved using Euclidian distance in pixel intensity space



Resources

- https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf
- http://www.iro.umontreal.ca/~lisa/pointeurs/ECML2011_CAE.pdf
- http://www.cs.toronto.edu/~larocheh/publications/icml-2008-denoising-autoencoders.pdf


QUESTIONS?
COMMENTS?
CONCERNS?