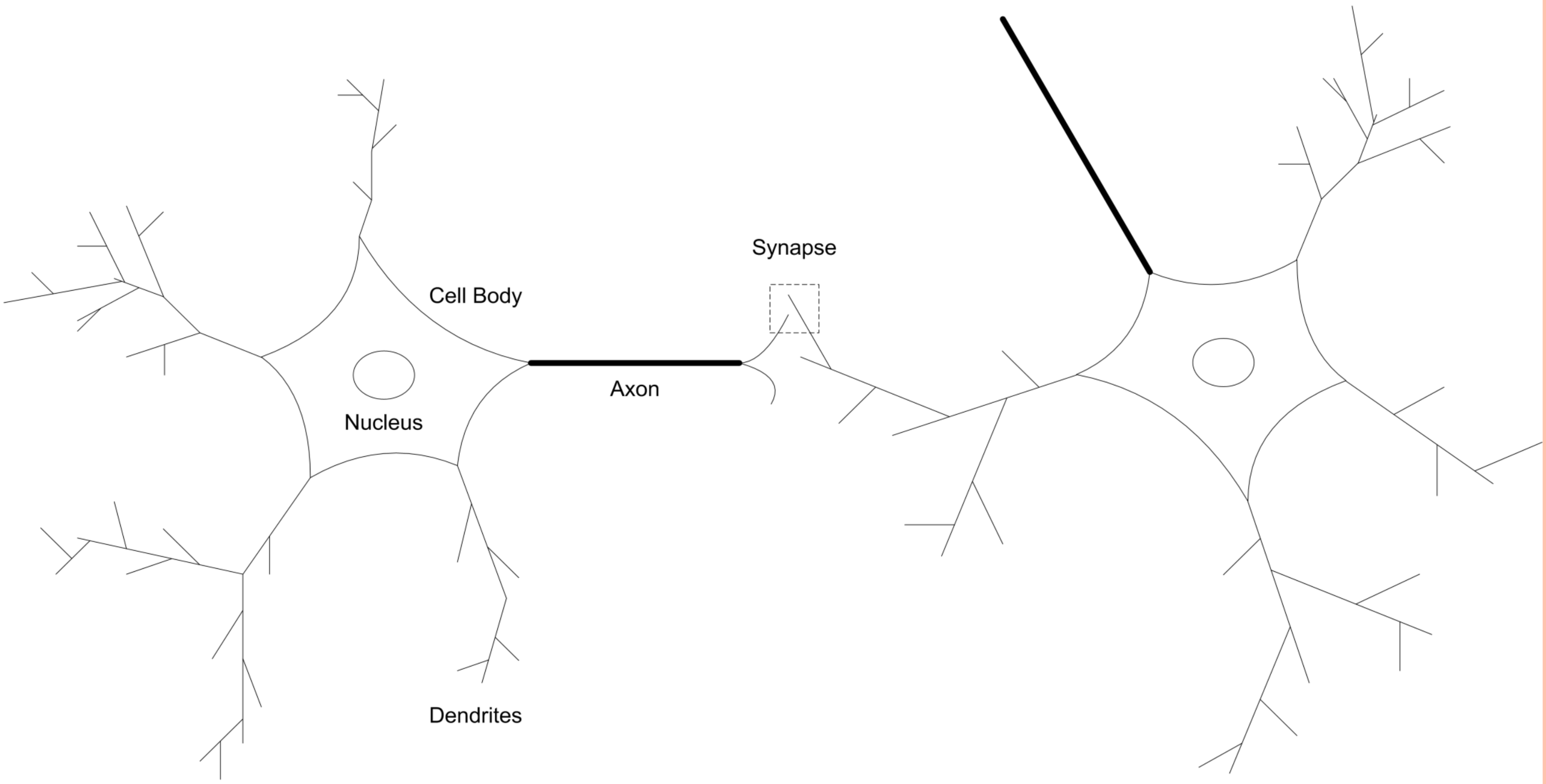
SCC.413 Applied Data Mining
Week 15 Autoencoders

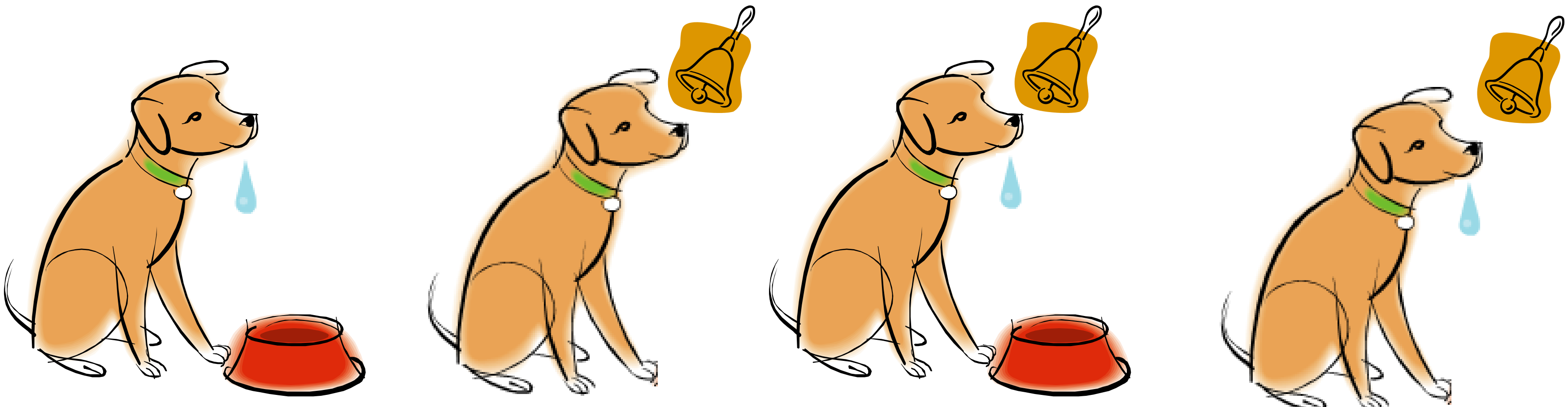
What we expect from AI in general

BIOLOGICAL MOTIVATION

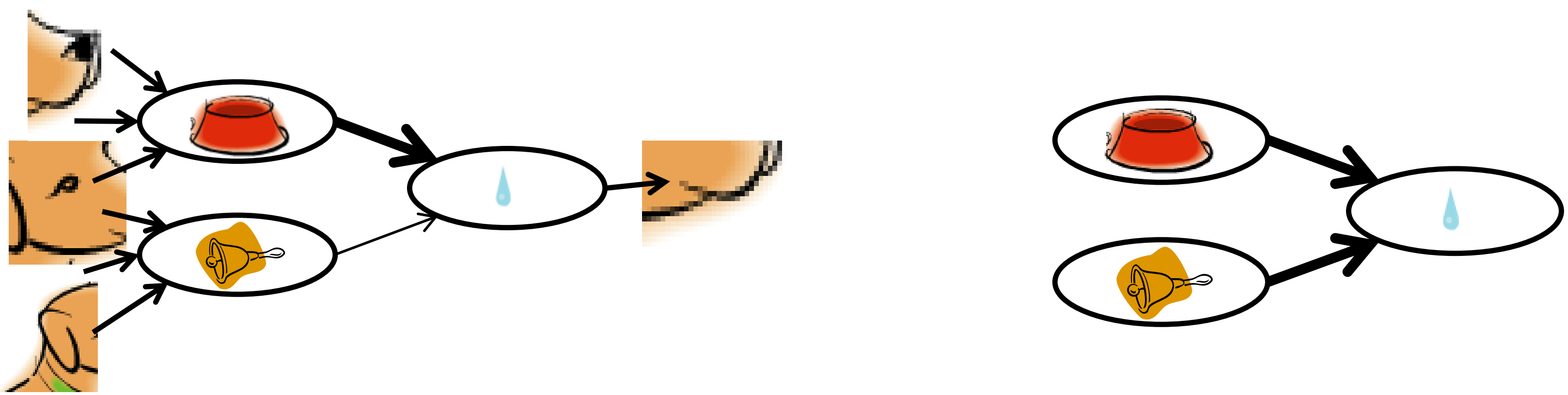
- In 1943, McCulloch & Pitts built a new model for information processing based on their knowledge on neurology.



Hebbian Learning



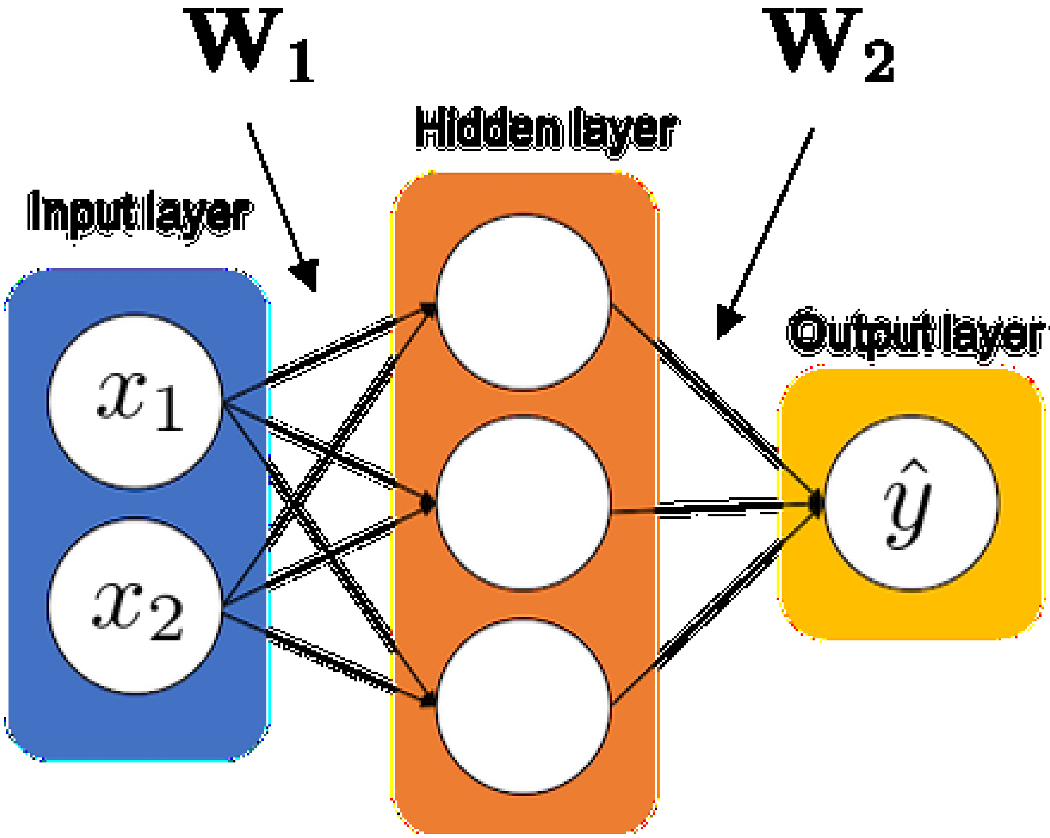
Before conditioning   Conditioning   After Conditioning



"The general idea is that any two cells or systems of cells that are repeatedly active at the same time will tend to become 'associated', so that activity in one facilitates activity in the other." [Hebb 1949]

MULTIPLE LAYER NEURAL NETWORK

- MLP and Backpropagation



Lab Solution

- Task01
- Task02

Outline – Variants of Autoencoders

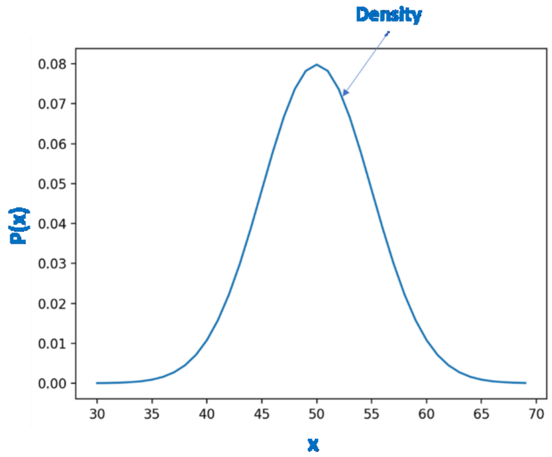
- Restricted Boltzmann Machine
- Autoencoder
- Deep Autoencoder
- Convolutional Autoencoder
- LSTM Autoencoder
- Variational Autoencoder

Restricted Boltzmann Machine

- Density Estimation
  - Considering you have a data set $\{x_i\}$
  - In the view of statistician, you would assume it has density distribution P(x),

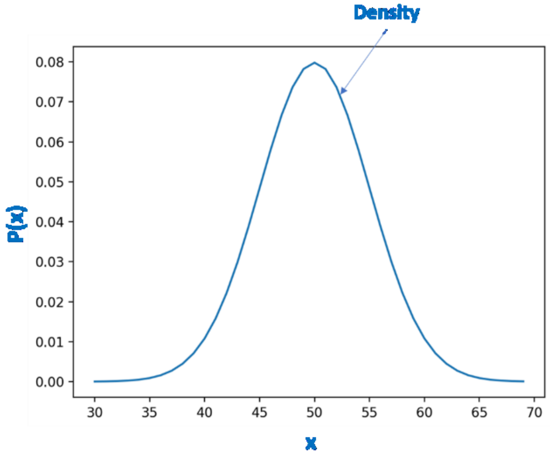
Restricted Boltzmann Machine

- Density Estimation
  - Considering you have a data set $\{x_i\}$
  - In the view of statistician, you would assume it has density distribution P(x),



Restricted Boltzmann Machine

- Density Estimation
  - Considering you have a data set $\{x_i\}$
  - In the view of statistician, you would assume it has density distribution P(x),



- We want to learn about their probability.
  - Explicit Density Estimation: RBM & VAE
  - Implicit Density Estimation: GANs

Restricted Boltzman Machine

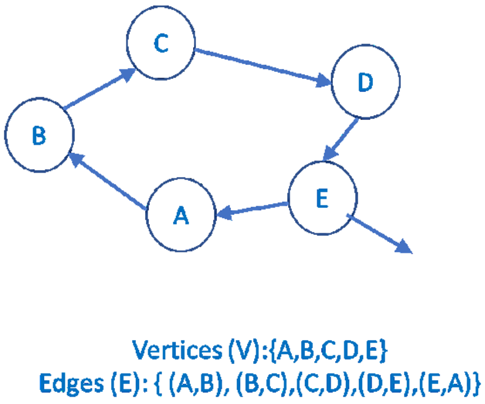
- Graphical Models
  - A graphical probabilistic model is used to express the conditional dependency between random variables.

Restricted Boltzman Machine

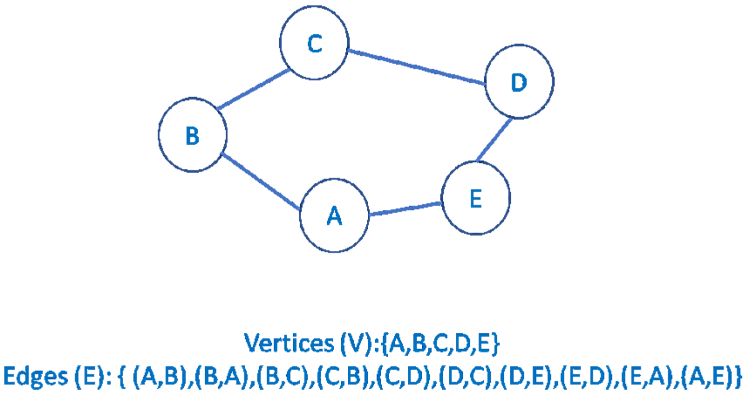
- Graphical Models
  - A graphical probabilistic model is used to express the conditional dependency between random variables.
  - A graphical model has 2 components: vertices & edges.
  - The vertices indicate the state of random variable and the edge indicates direction of transformation

Restricted Boltzman Machine

- Graphical Models
  - A graphical probabilistic model is used to express the conditional dependency between random variables.
  - A graphical model has 2 components: vertices & edges.
  - The vertices indicate the state of random variable and the edge indicates direction of transformation



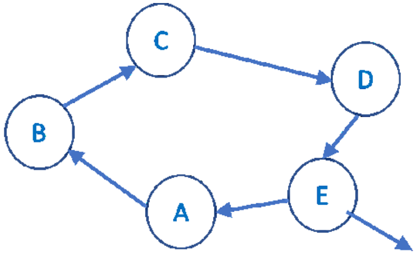
Directed graph



Undirected graph

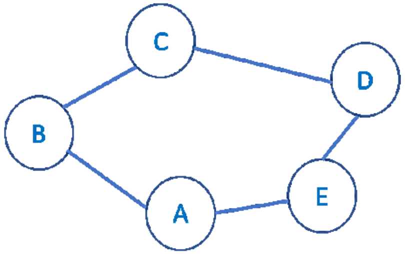
Restricted Boltzman Machine

- Graphical Models
  - A graphical probabilistic model is used to express the conditional dependency between random variables.
  - A graphical model has 2 components: vertices & edges.
  - The vertices indicate the state of random variable and the edge indicates direction of transformation



Vertices (V):{A,B,C,D,E}
Edges (E): { (A,B), (B,C),(C,D),(D,E),(E,A)}

Directed graph



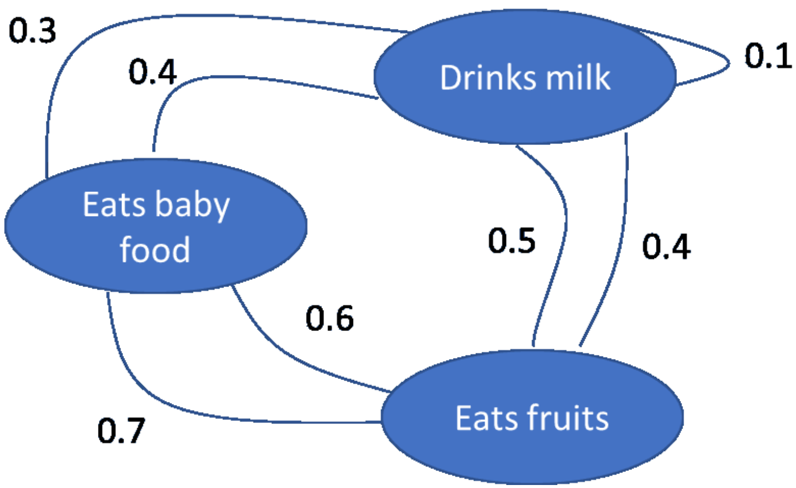
Vertices (V):{A,B,C,D,E}
Edges (E): { (A,B),(B,A),(B,C),(C,B),(C,D),(D,C),(D,E),(E,D),(E,A),(A,E)}

Undirected graph

- Mostly, Bayesian rules could be applied.
  - Causality, Hetu-Phala in Buddism

Restricted Boltzman Machine

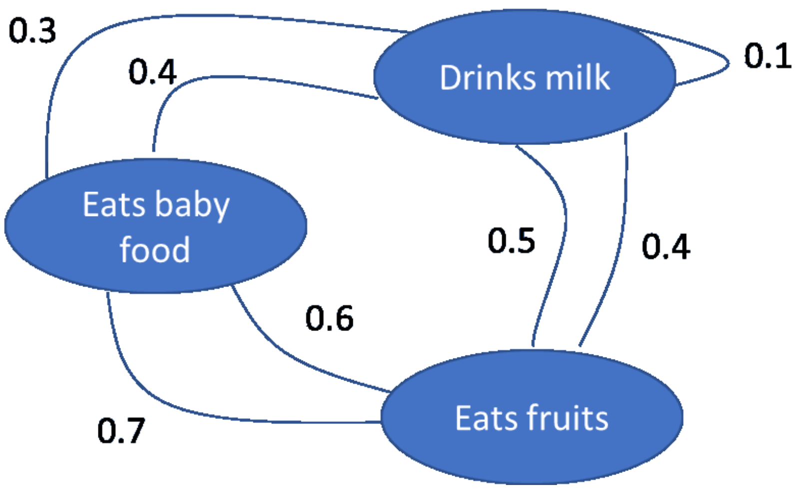
- Graphical Models
  - An undirected graphical model of a Markov process of diet habit of a baby.



  -

Restricted Boltzman Machine

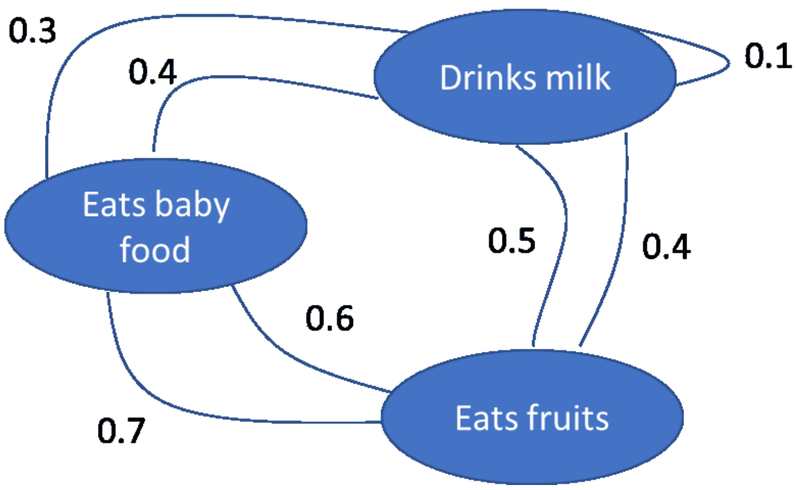
- Graphical Models
  - An undirected graphical model of a Markov process of diet habit of a baby.



  - The graph model is used to indicate a baby's choice for the next meal with the associated probabilities.

Restricted Boltzman Machine

- Graphical Models
  - An undirected graphical model of a Markov process of diet habit of a baby.



  - The graph model is used to indicate a baby's choice for the next meal with the associated probabilities.
  - The baby's choice of next meal depends solely on what it is eating now and not what it ate earlier.

Restricted Boltzmann Machine

- Boltzmann Machine
  - A set of random variables having Markov property and described by an undirected graph is referred to as **Markov Random Field (MRF)** or **Markov network**.

Restricted Boltzmann Machine

- Boltzmann Machine
  - A set of random variables having Markov property and described by an undirected graph is referred to as **Markov Random Field (MRF)** or **Markov network**.
  - **A Boltzmann Machine (BM)** is a probabilistic generative undirected graph model that satisfies Markov property.

Restricted Boltzmann Machine

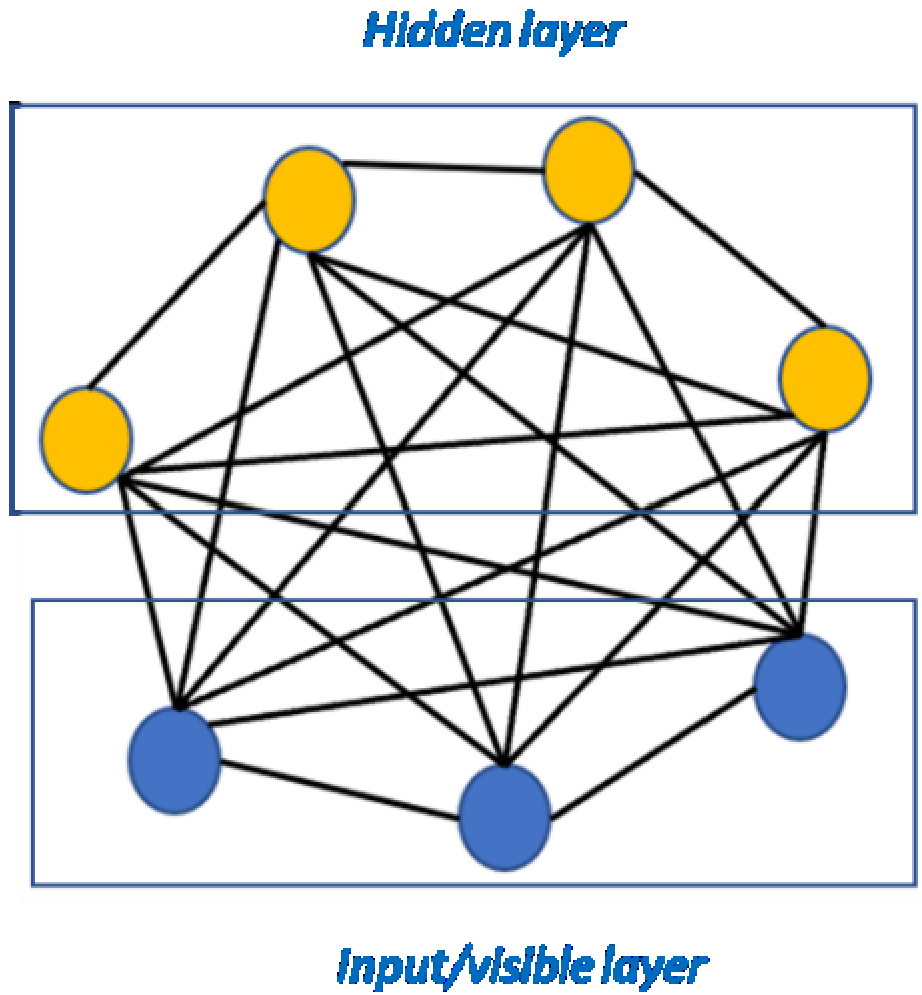
- Boltzmann Machine
  - A set of random variables having Markov property and described by an undirected graph is referred to as **Markov Random Field (MRF)** or **Markov network**.
  - **A Boltzmann Machine (BM)** is a probabilistic generative undirected graph model that satisfies Markov property.
  - BMs learn the probability density from the input data to generating new samples from the same distribution.

Restricted Boltzmann Machine

- Boltzmann Machine
  - A set of random variables having Markov property and described by an undirected graph is referred to as **Markov Random Field (MRF)** or **Markov network**.
  - **A Boltzmann Machine (BM)** is a probabilistic generative undirected graph model that satisfies Markov property.
  - BMs learn the probability density from the input data to generating new samples from the same distribution.
  - A BM has an input or visible layer and one or several hidden layers. There is no output layer.

Restricted Boltzman Machine

- Boltzmann Machine



Restricted Boltzmann Machine

- Analogy between NN and BM
  - The neurons in the network learn to make stochastic decisions about whether to turn on or off based on the data fed to the network during training. This helps the BM discover and model the complex underlying patterns in the data.

Restricted Boltzmann Machine

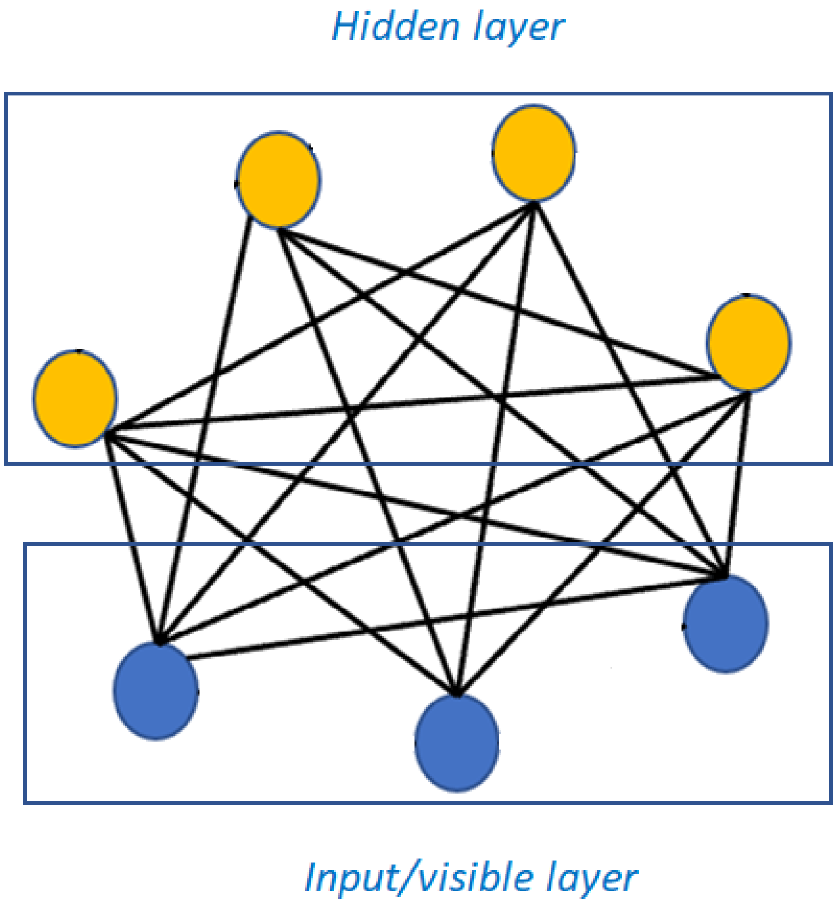
- Analogy between NN and BM
  - The neurons in the network learn to make stochastic decisions about whether to turn on or off based on the data fed to the network during training. This helps the BM discover and model the complex underlying patterns in the data.
  - A vital difference between BM and other popular neural net architectures is that the neurons in BM are connected not only to neurons in other layers but also to neurons within the same layer.

Restricted Boltzmann Machine

- Analogy between NN and BM
  - The neurons in the network learn to make stochastic decisions about whether to turn on or off based on the data fed to the network during training. This helps the BM discover and model the complex underlying patterns in the data.
  - A vital difference between BM and other popular neural net architectures is that the neurons in BM are connected not only to neurons in other layers but also to neurons within the same layer.
  - Essentially, every neuron is connected to every other neuron in the network. This imposes a stiff challenge in training a BM and this version of BM.

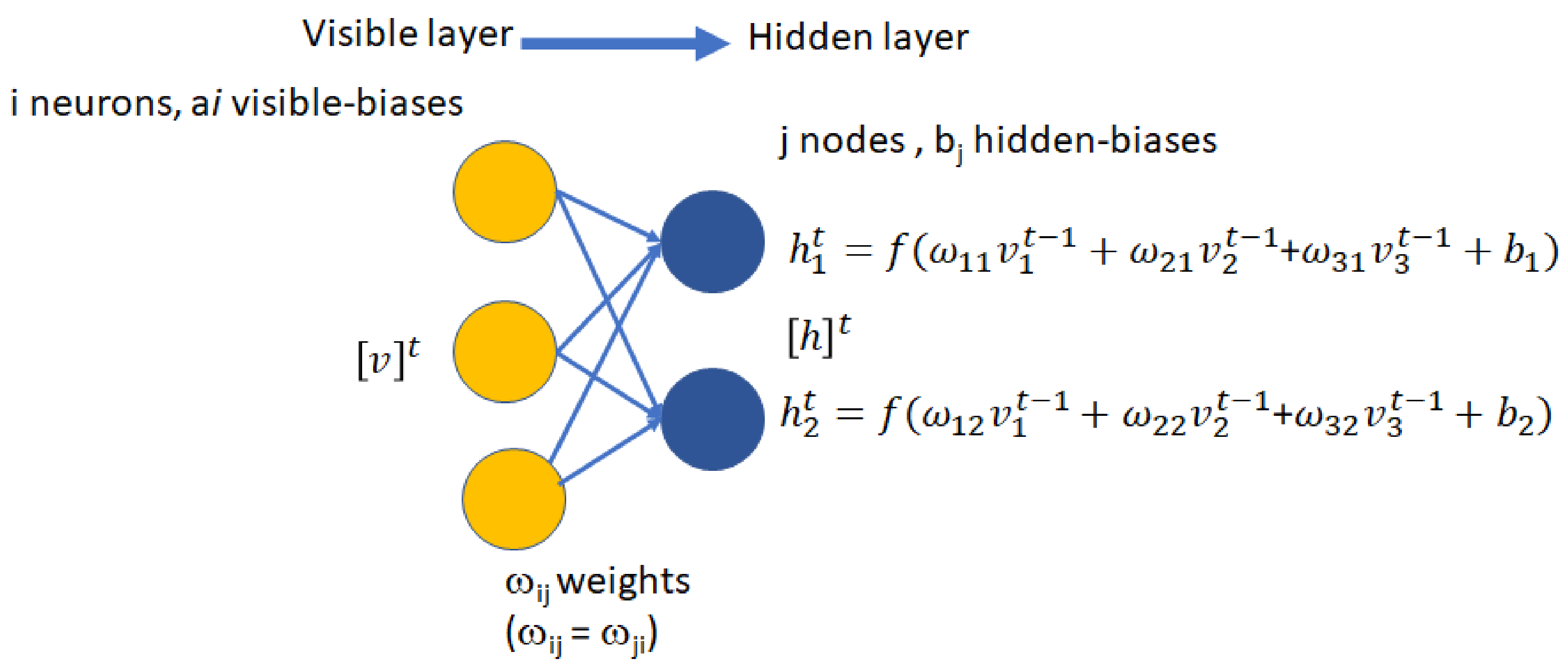
Restricted Boltzmann Machine

- Restricted BM: a two-layer neural network



Restricted Boltzmann Machine

- Training

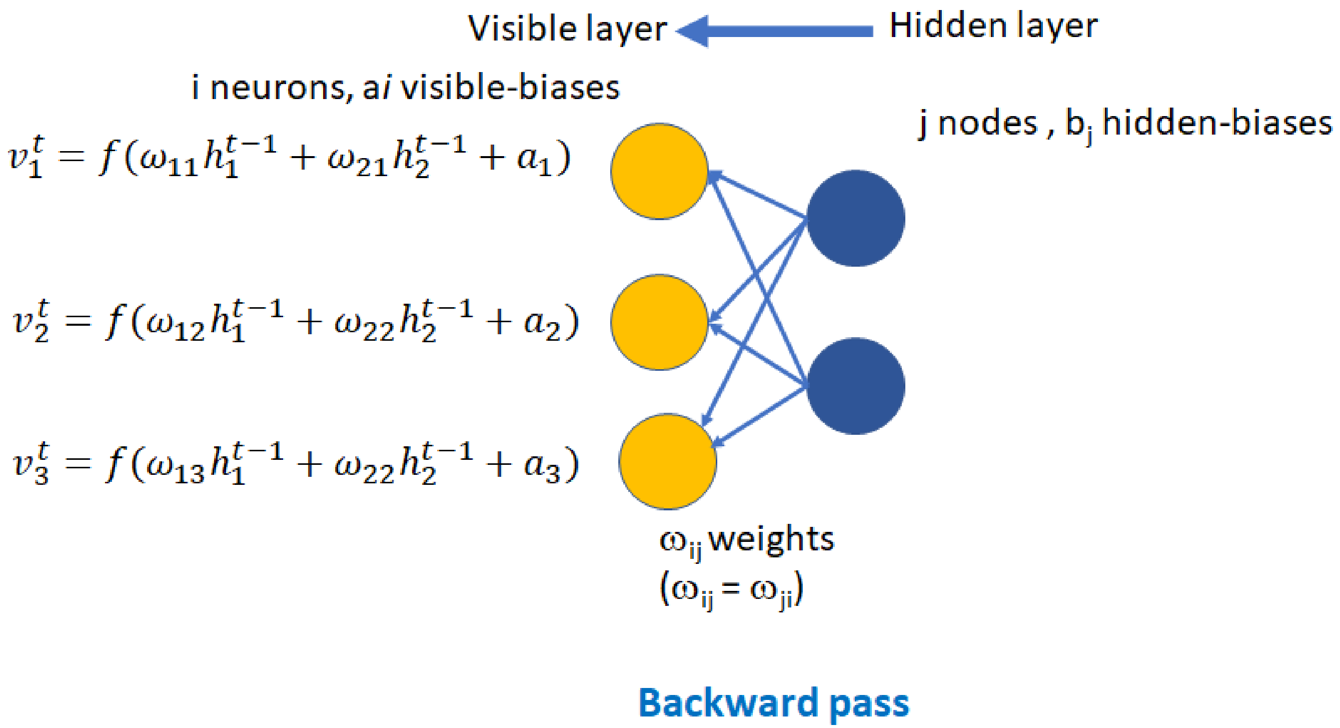


Forward pass

- $P(h|x)$ ~ Gibbs sampling on $WX+b$

Restricted Boltzmann Machine

- Training



- $P(x|h)$ ~ Gibbs sampling on WH+b

Restricted Boltzmann Machine

- Gibbs sampling ➔ Sigmoid

$f(m) = \frac{1}{1+e^{-m}}$ is the sigmoid function

$[v]^t$ is the reconstructed vector at 't' iteration. $[v]^0$ Corresponds to input data [x]

$[h]^t$ is the hidden vector at 't' iteration. For hidden vector t>0

Restricted Boltzmann Machine

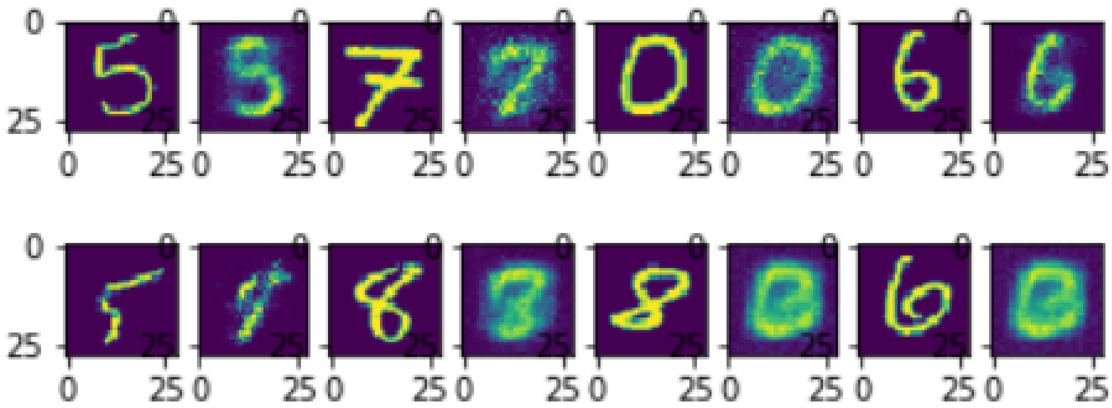
- Gibbs sampling ➔ Sigmoid

$$f(m) = \frac{1}{1+e^{-m}}$$ is the sigmoid function

$[v]^t$ is the reconstructed vector at 't' iteration. $[v]^0$ Corresponds to input data [x]

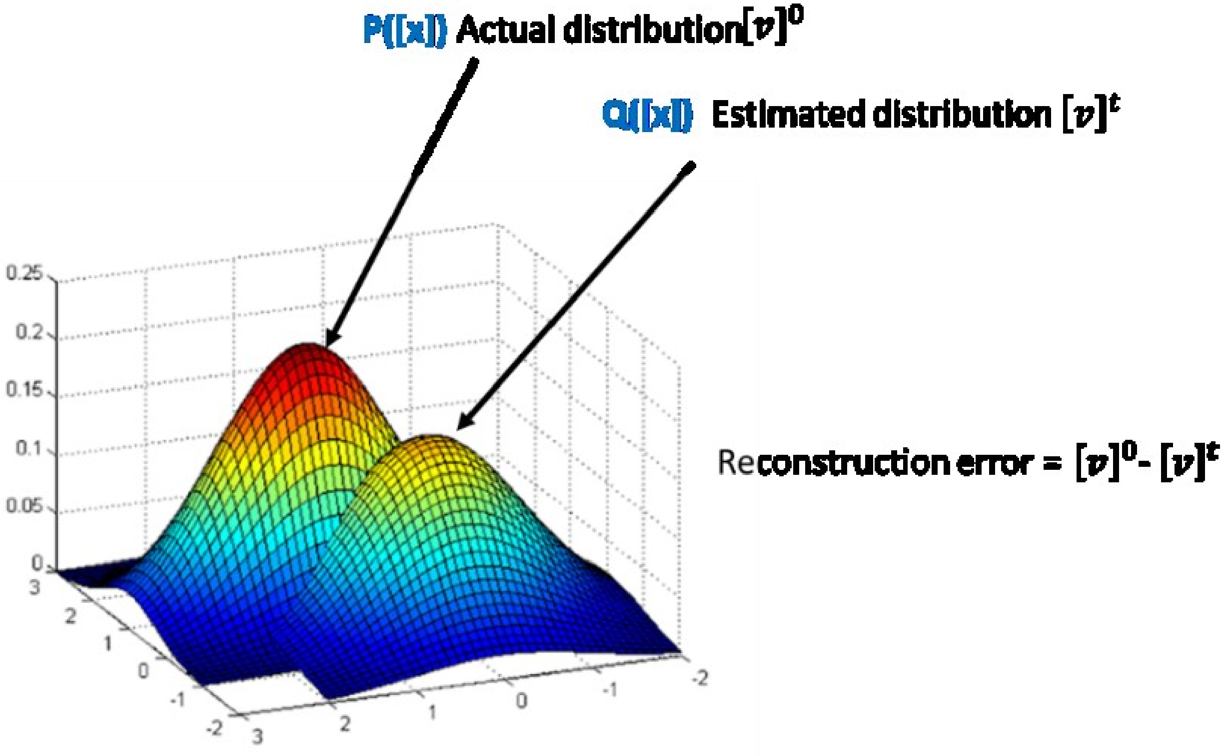
$[h]^t$ is the hidden vector at 't' iteration. For hidden vector t>0

- Example:



Restricted Boltzmann Machine

- Actual + est. distributions & reconstruction error



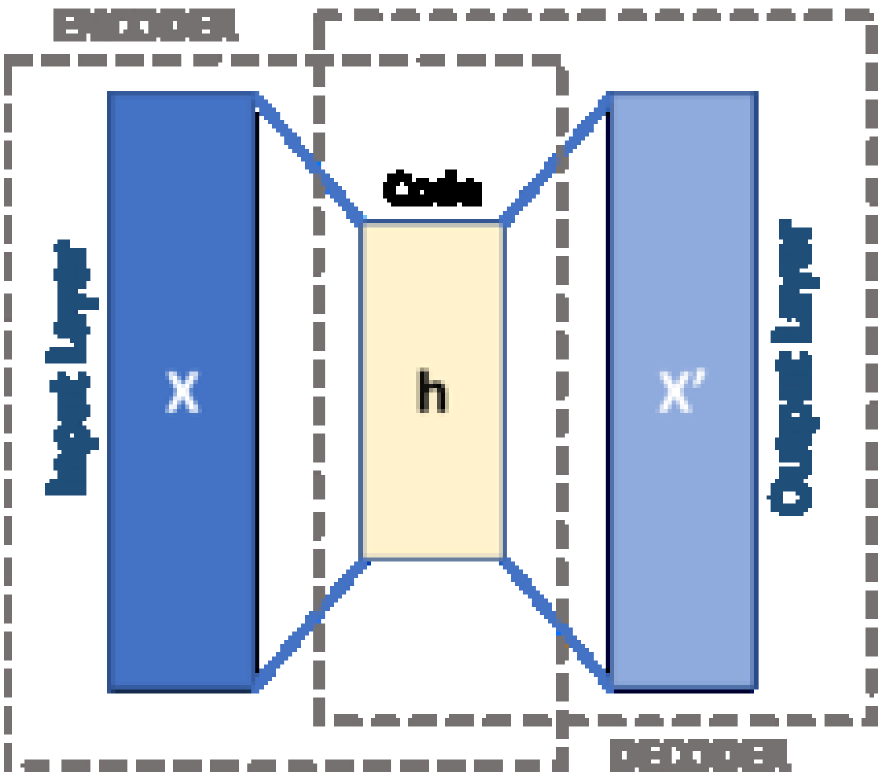
- RBMs follow a **generative learning** approach

Deep Autoencoder

- Basic Autoencoder

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

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

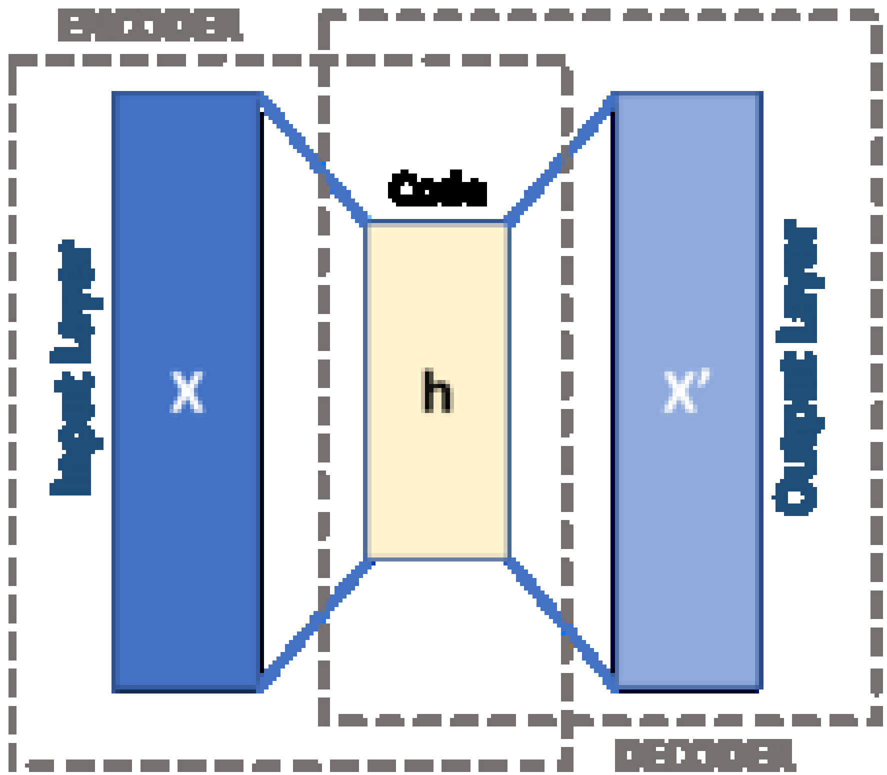


Deep Autoencoder

- Basic Autoencoder

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

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

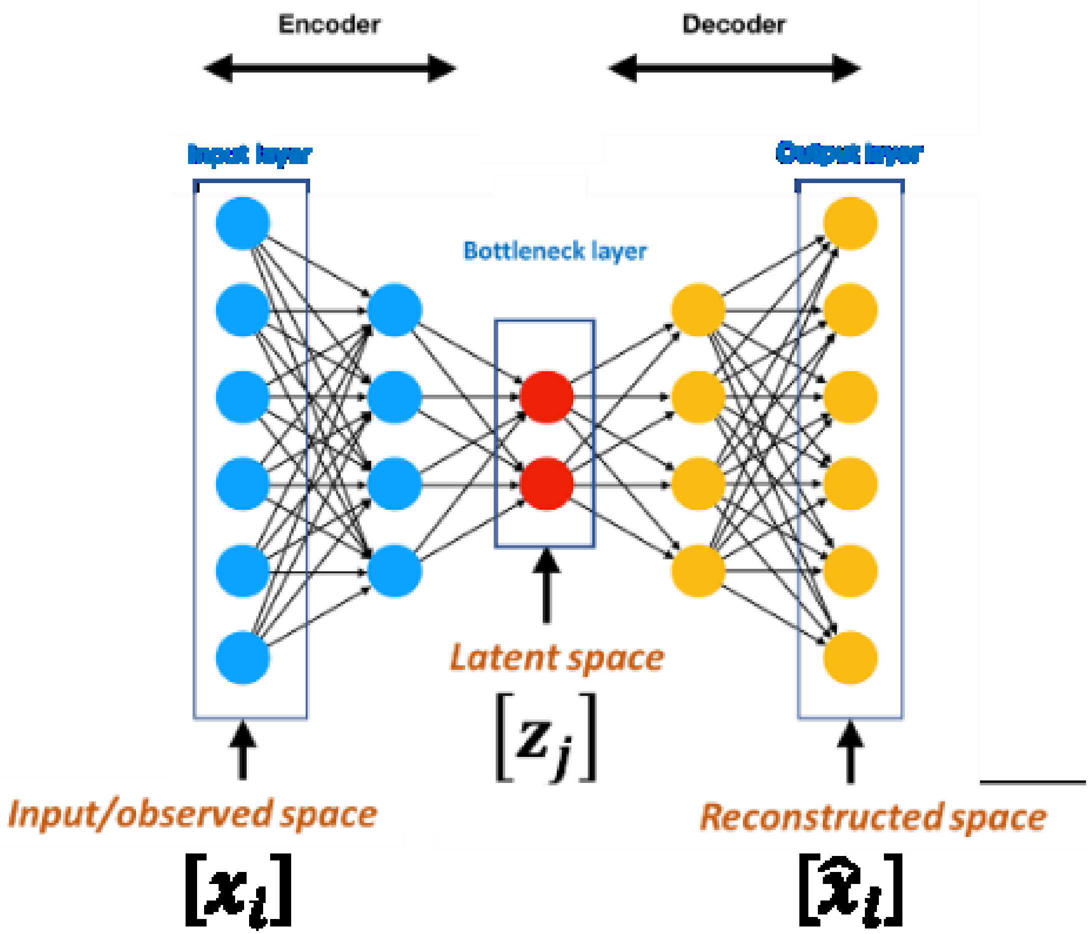


- Cost function:

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')\|^2$$

Deep Autoencoder

- Deep autoencoder: more than 1 hidden layer



Deep Autoencoder

- Training – Hinton's approach
  - Geoffrey Hinton developed a two-step technique for training many-layered deep autoencoders.

Deep Autoencoder

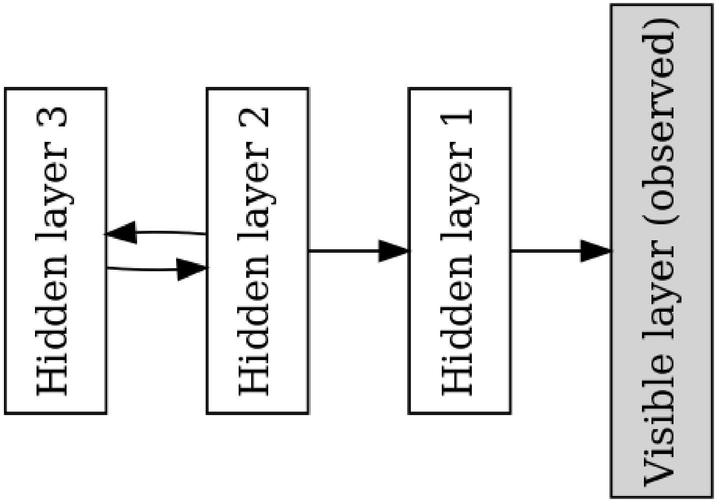
- Training – Hinton’s approach
  - Geoffrey Hinton developed a two-step technique for training many-layered deep autoencoders.
  - Pretraining: treat each neighbouring set of two layers as a RBM, to approximate a good solution

Deep Autoencoder

- Training – Hinton's approach
  - Geoffrey Hinton developed a two-step technique for training many-layered deep autoencoders.
  - Pretraining: treat each neighbouring set of two layers as a RBM, to approximate a good solution
  - Then using backpropagation to fine-tune the results.

Deep Autoencoder

- Training – Hinton's approach
  - Geoffrey Hinton developed a two-step technique for training many-layered deep autoencoders.
  - Pretraining: treat each neighbouring set of two layers as a RBM, to approximate a good solution
  - Then using backpropagation to fine-tune the results.
- This model takes the name of deep belief network

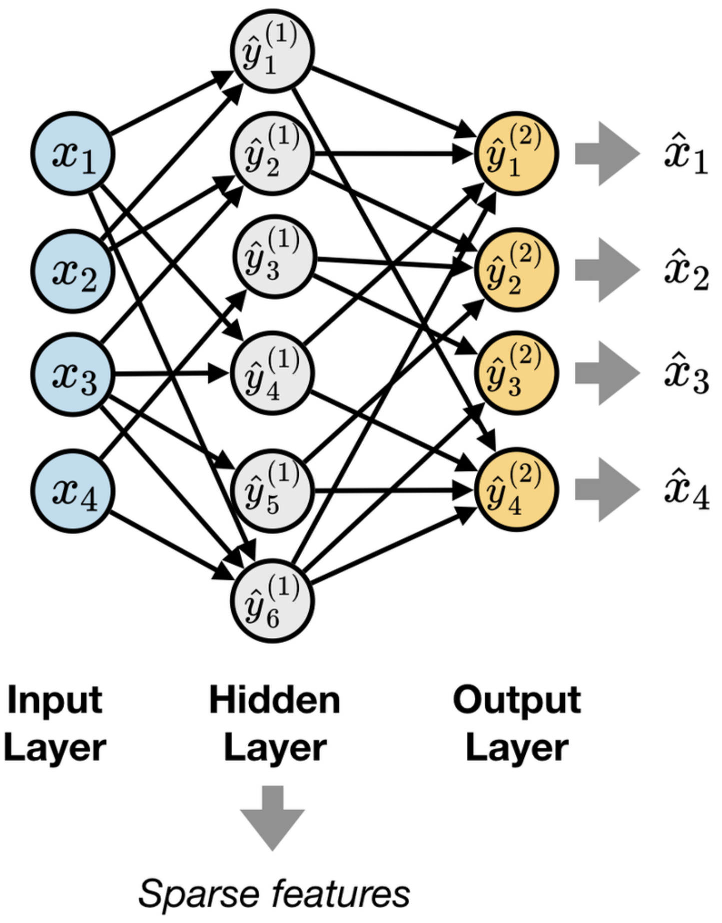


Sparse Autoencoder

- Simple Sparse AE

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') + \Omega(\boldsymbol{h})$$

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



Sparse Autoencoder

- Simple Sparse AE

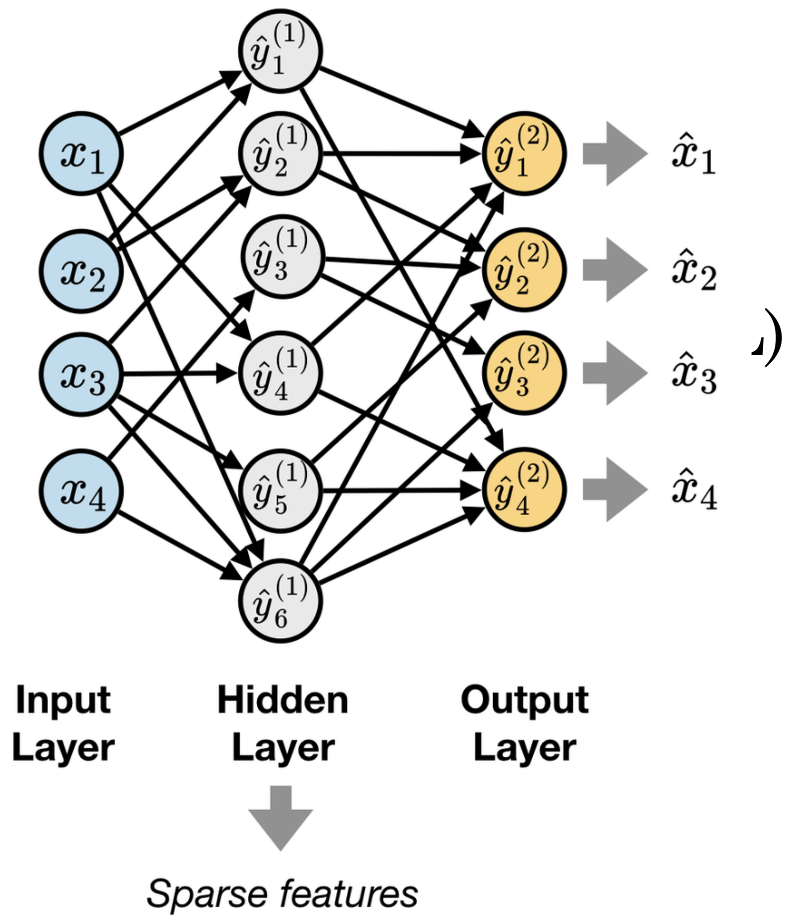
$$\mathcal{L}(\mathbf{x}, \mathbf{x}') + \Omega(\boldsymbol{h})$$

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

- Kullback-Leibler divergence

$$KL(\rho || \hat{\rho}_j)$$

$$\hat{\rho_j} = \frac{1}{m} \sum_{i=1}^{m} [h_j(x_i)]$$



Sparse Autoencoder

- Simple Sparse AE

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') + \Omega(\boldsymbol{h})$$

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

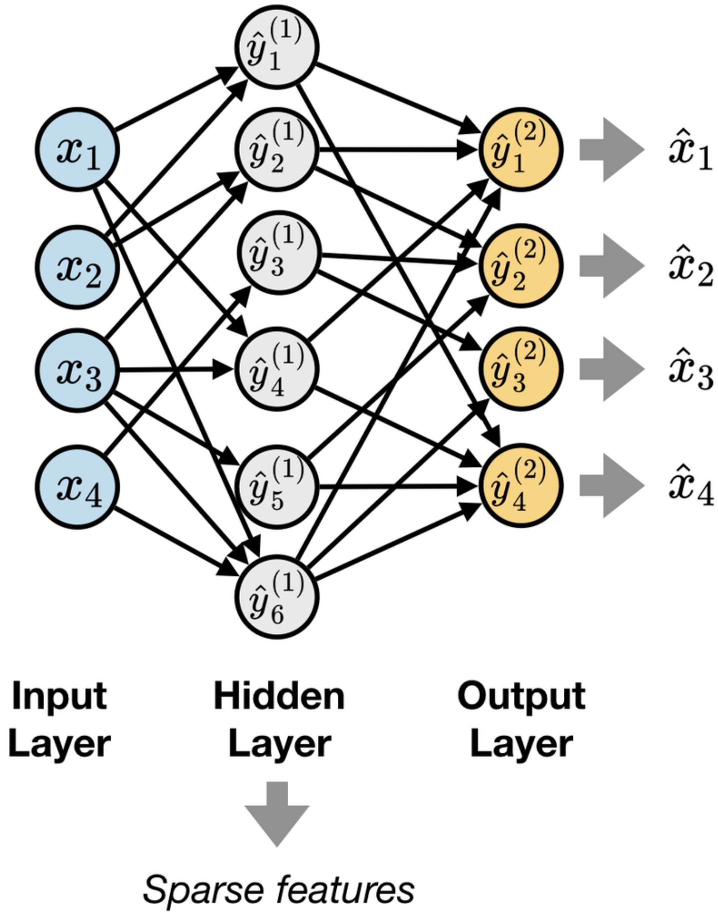
- Kullback-Leibler divergence

$$KL(\rho || \hat{\rho}_j)$$

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} [h_j(x_i)]$$

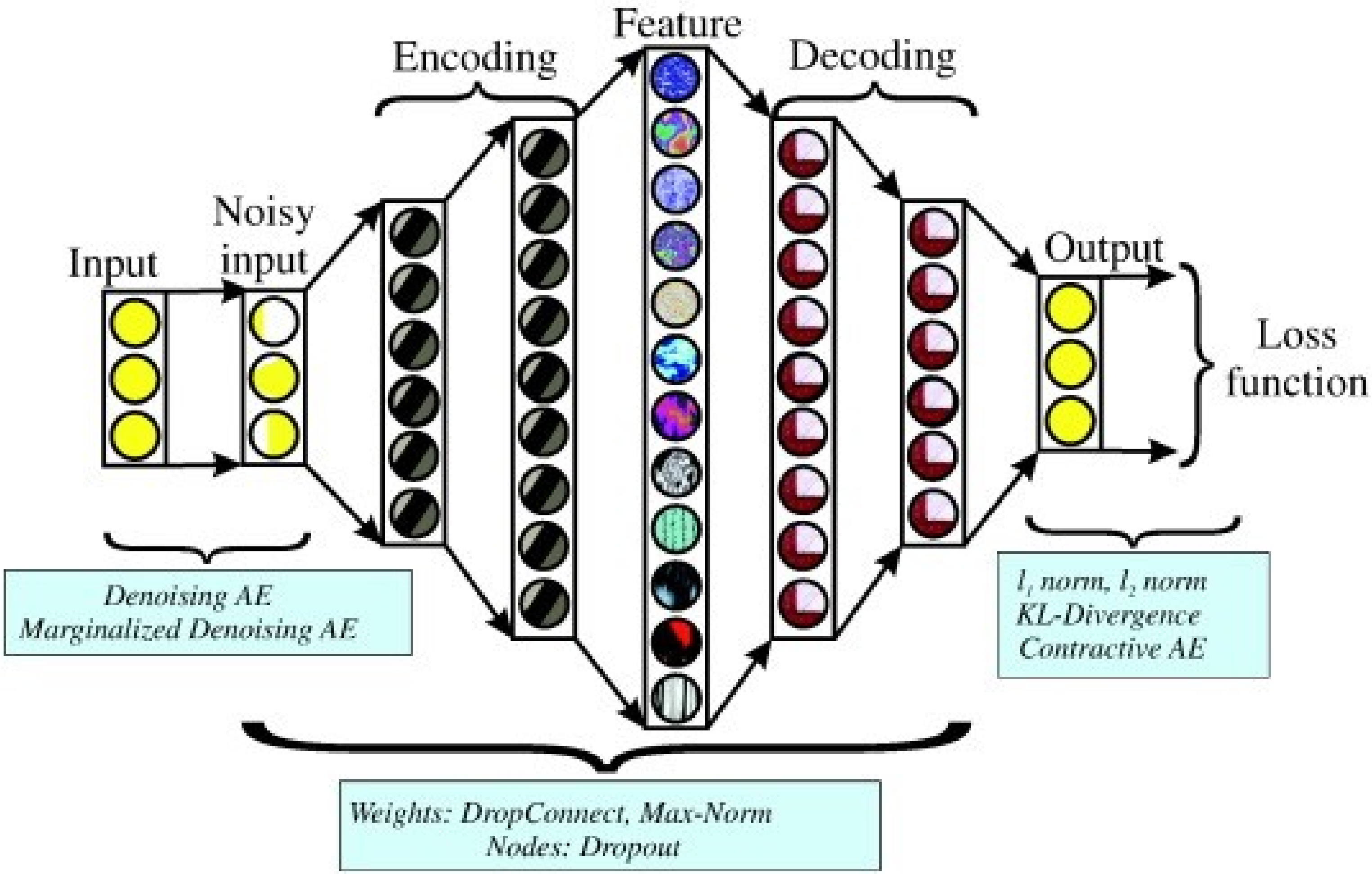
- L1/L2 regularization

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') + \lambda \sum_i |h_i|$$



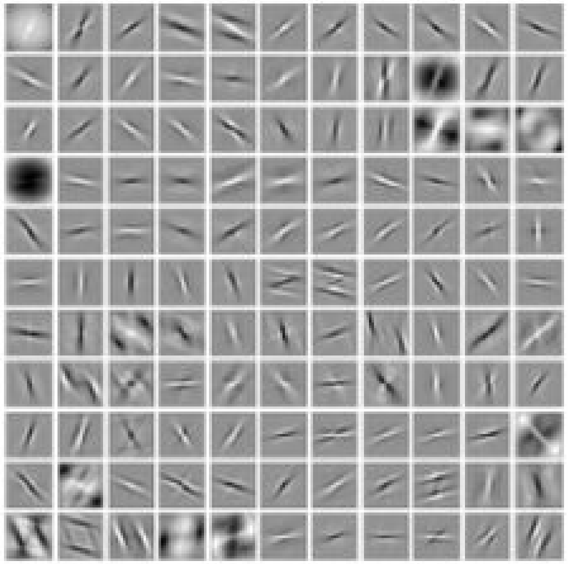
Sparse Autoencoder

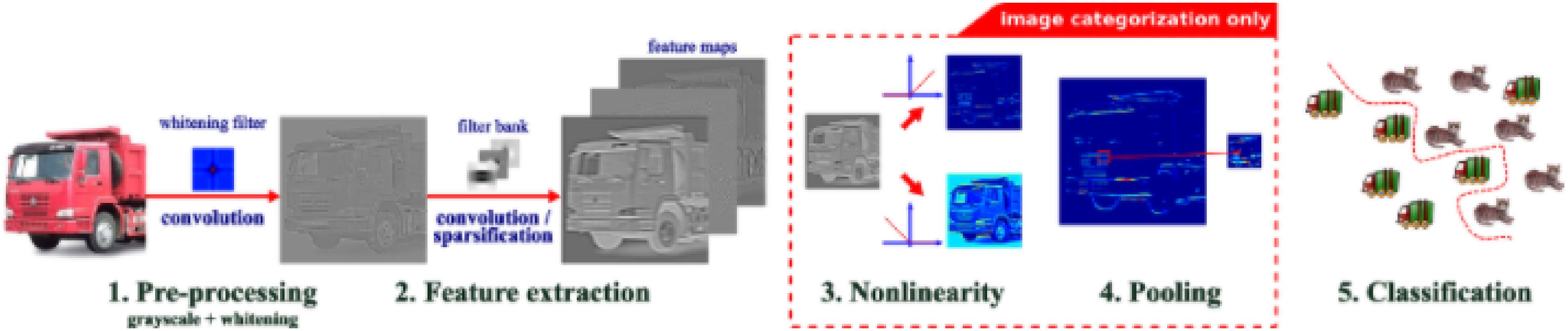
- Deep Sparse AE



Convolutional Autoencoder

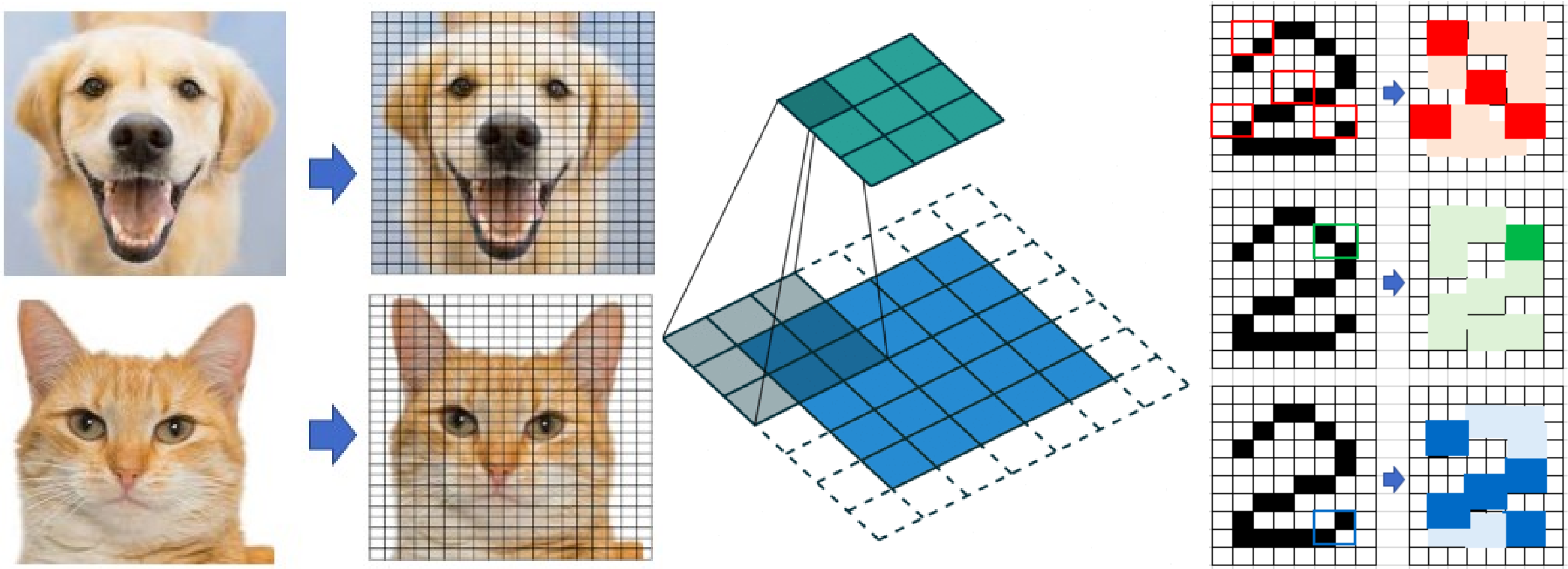
- Convolutional Filter Bank
  - Image feature extraction
  - Derived from Garbor filter
  - CVPR 2011, "Are Sparse Representations Really Relevant for Image Classification?"





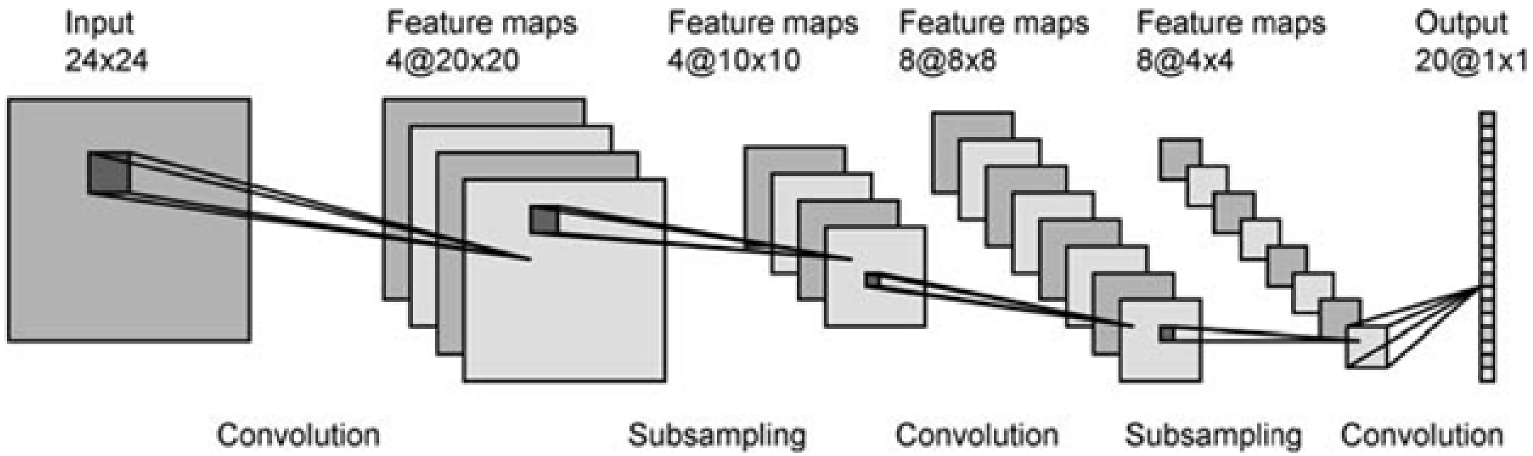
Convolutional Autoencoder

- Convolutional Neural Network
  - Apply conv filter bank to images,



Convolutional Autoencoder

- Convolutional Neural Network
  - Feature maps



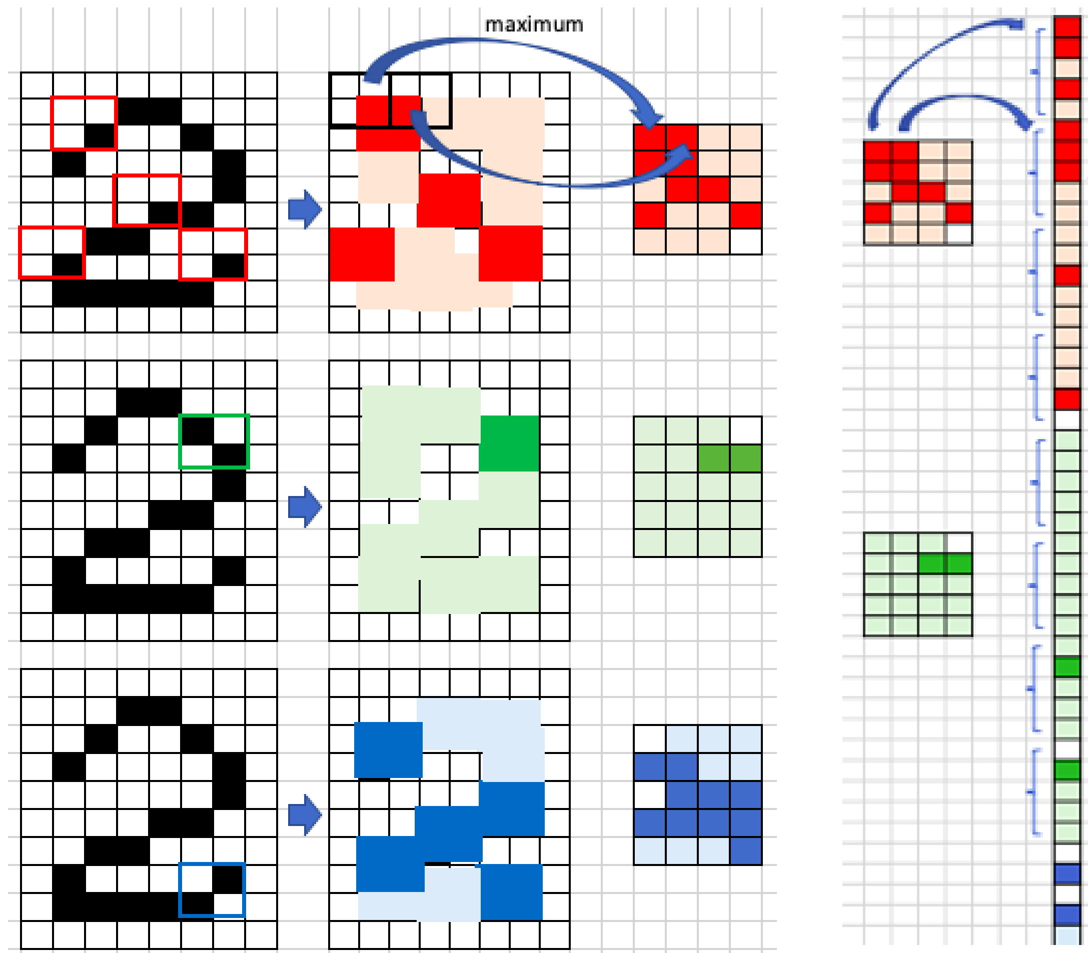
  - Maxpooling
  - Deep CNN: more than 1-layer conv filtering

Convolutional Autoencoder

- CNN based Autoencoder

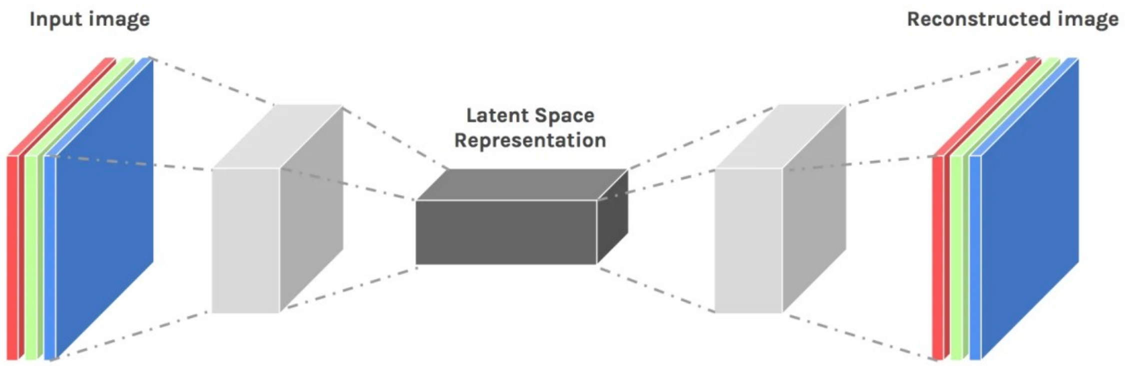
```
x = Conv2D(filters = 16,
          kernel_size = (3, 3),
          strides=(1,1),
          activation='relu',
          padding='same')
        (input_img)

x = MaxPooling2D(
          pool_size = (2, 2),
          padding='same')(x)
```



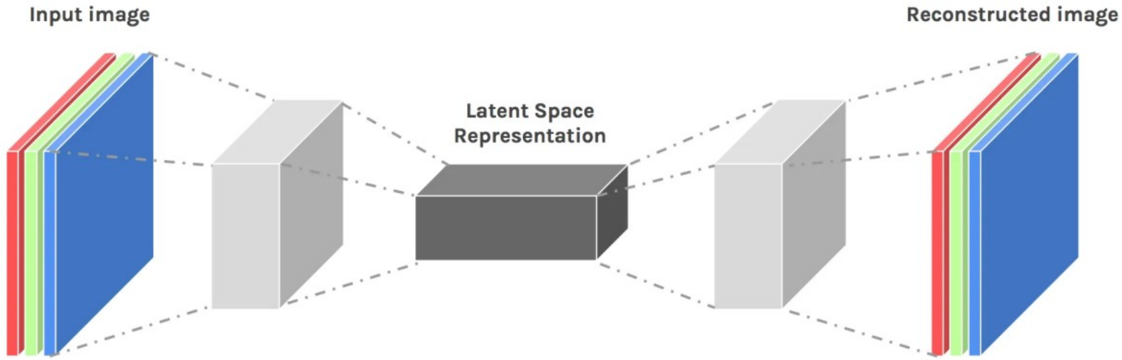
Convolutional Autoencoder

- CNN based Autoencoder
  - Basics: latent space is a 3D vol of feature maps

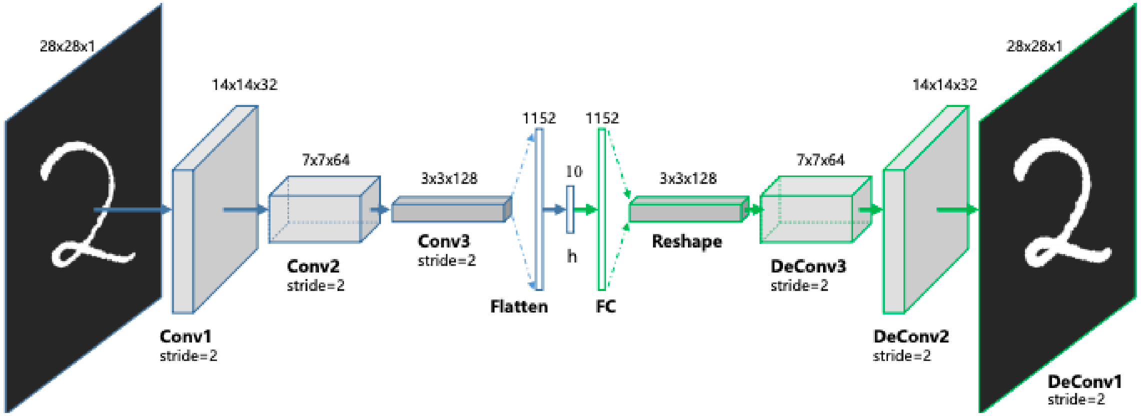


Convolutional Autoencoder

- CNN based Autoencoder
  - Basics: latent space is a 3D vol of feature maps



  - Deep Conv Autoencoder



Convolutional Autoencoder

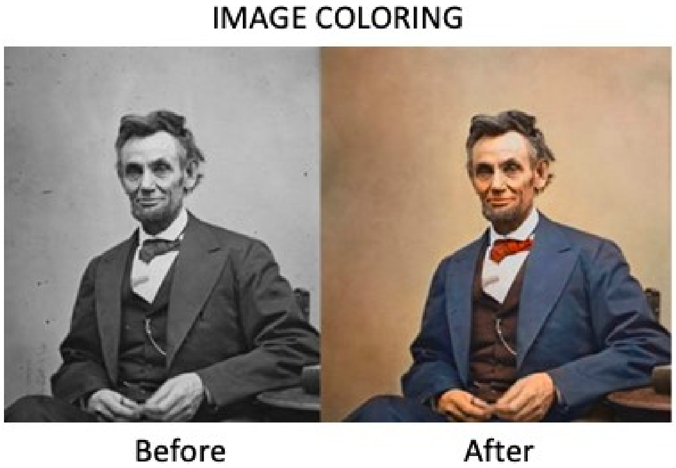
- Image Processing using CAE

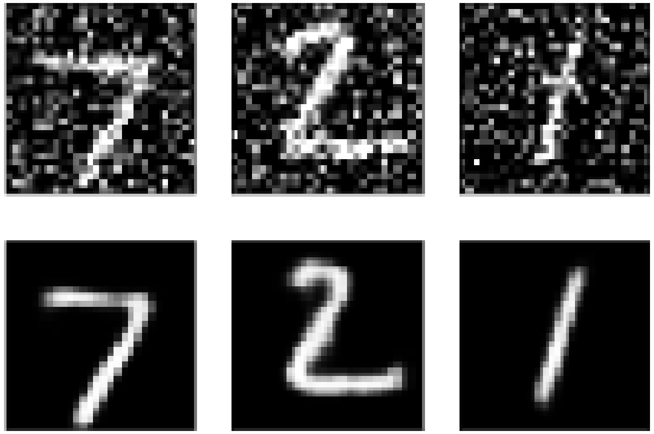
IMAGE COLORING

Before After

Convolutional Autoencoder

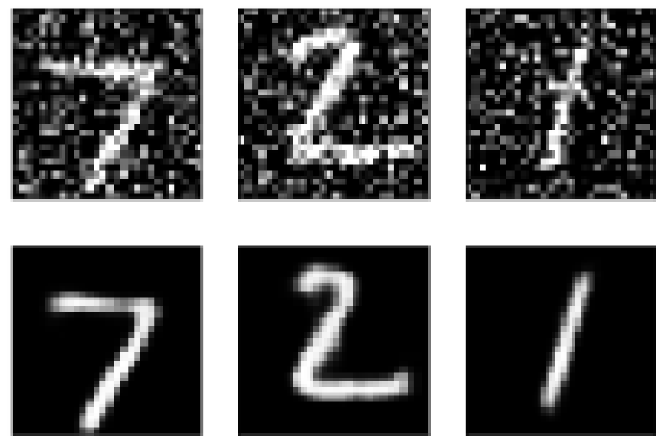
- Image Processing using CAE

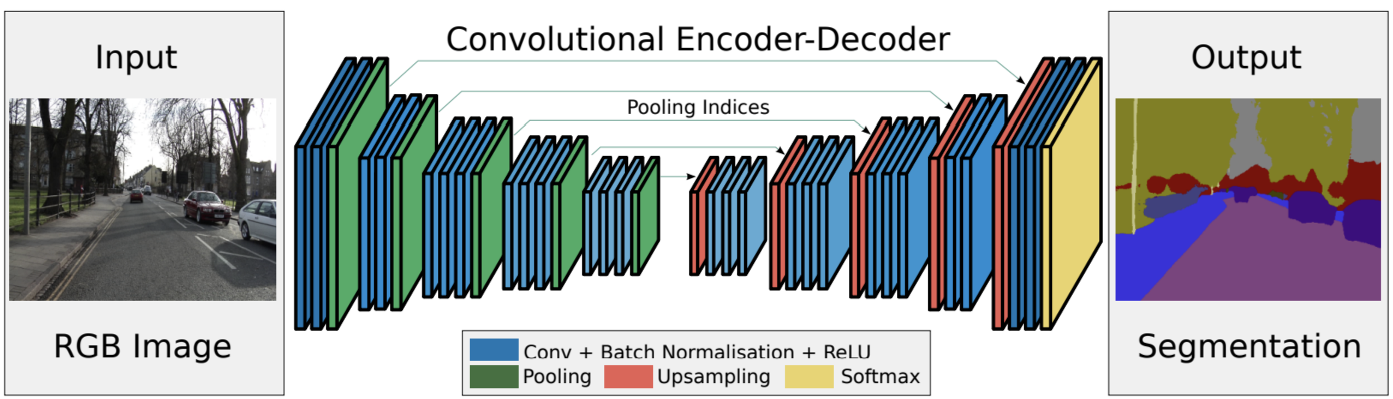




Convolutional Autoencoder

- Image Processing using CAE

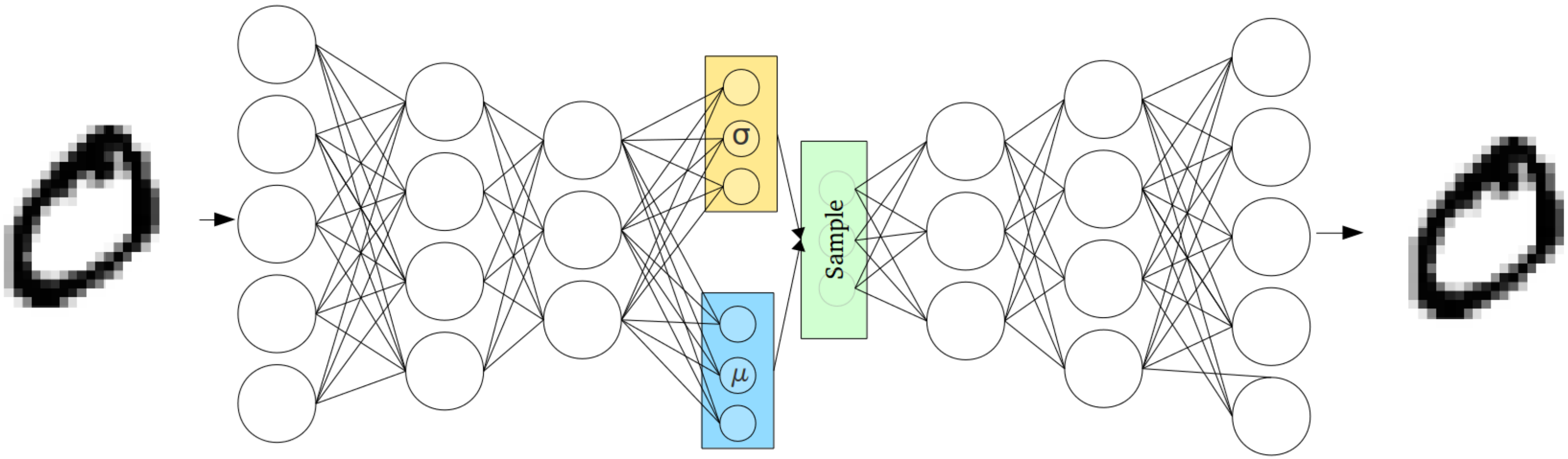




- https://arxiv.org/abs/1511.00561v3

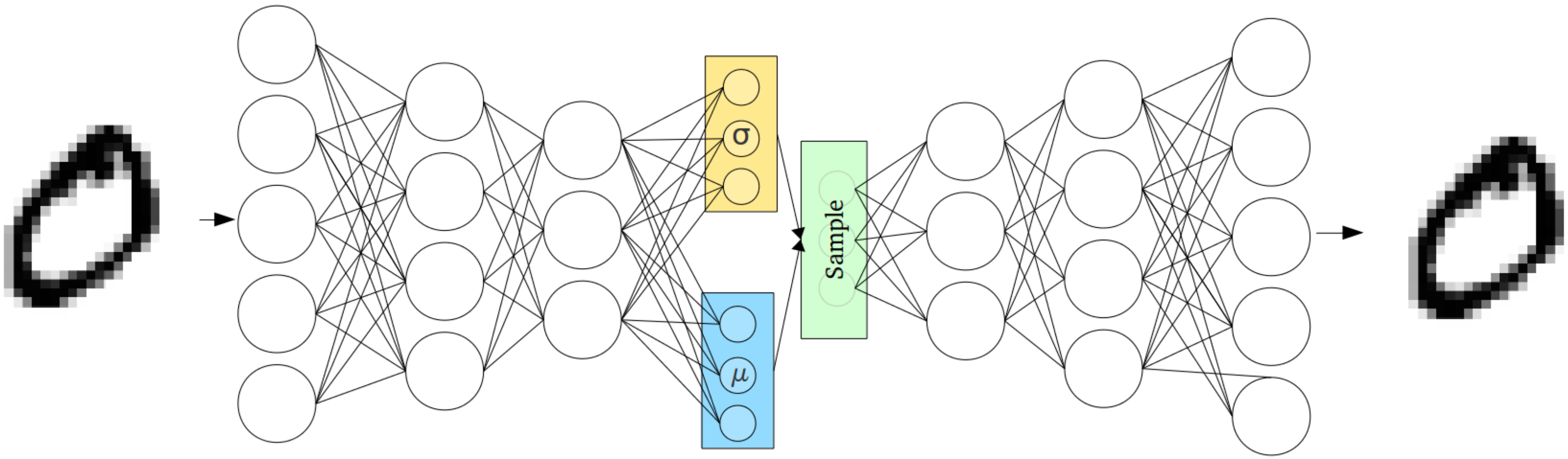
Variational Autoencoder

- Similar to a typical autoencoder



Variational Autoencoder

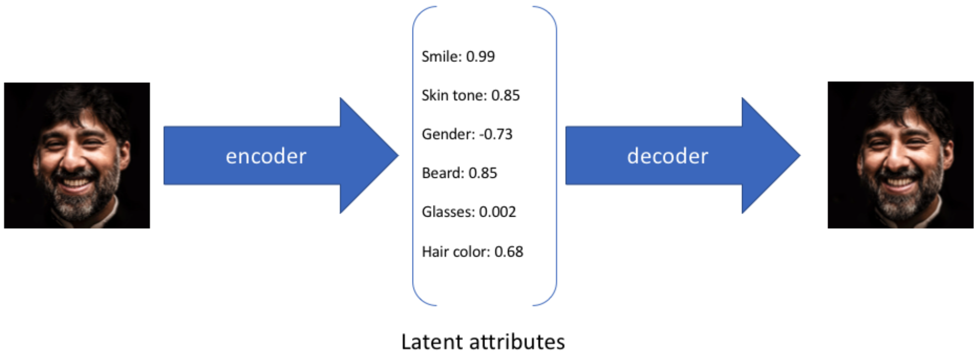
- Similar to a typical autoencoder



- Latent is modelled as a probability distribution
  - Gaussian distribution, $z \sim N(\mu, \varepsilon)$
  - Likelihood formula: $p(z|x) = \dfrac{p(x|z)\,p(z)}{p(x)}$

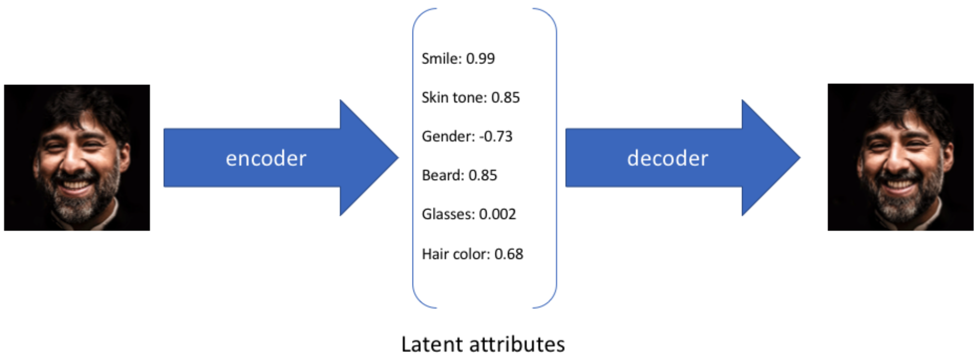
Variational Autoencoder

- A visualized comparison
  - Classical AE

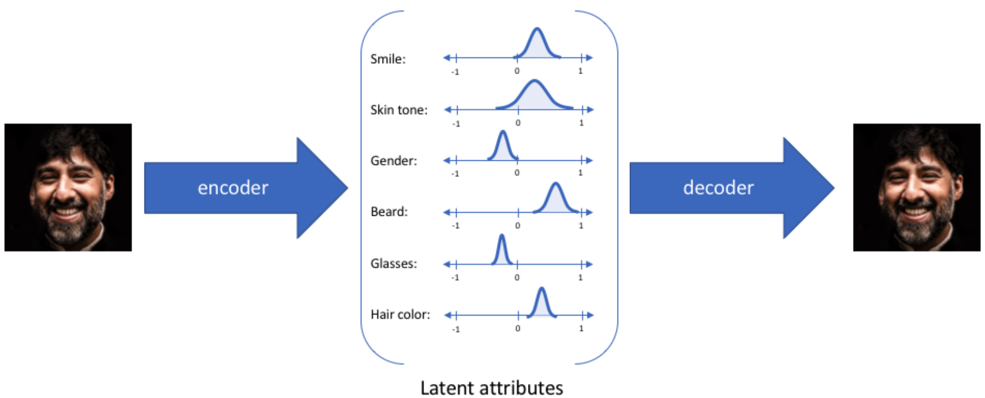


Variational Autoencoder

- A visualized comparison
  - Classical AE



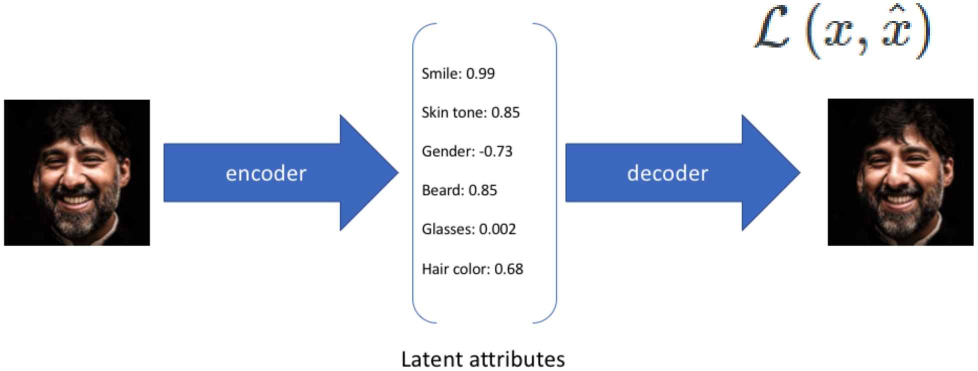
  - Variational AE



VARIATIONAL AUTOENCODER

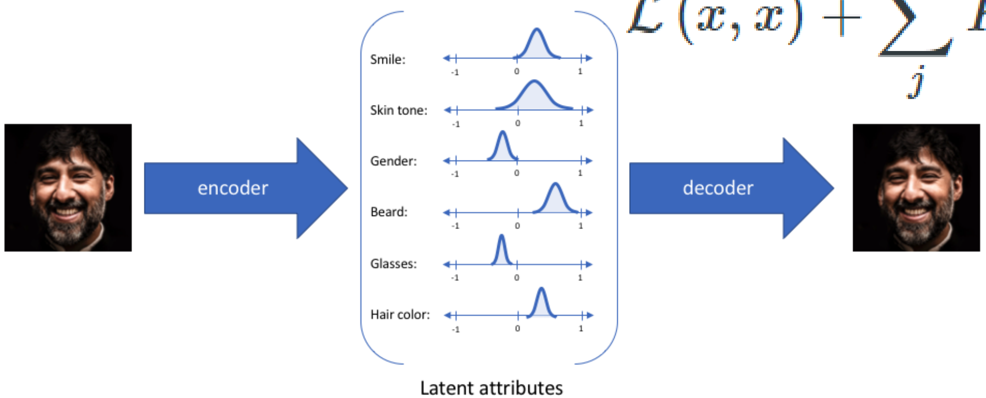
- A visualized comparison
  - Classical AE

$$\mathcal{L}\left(x, \hat{x}\right)$$



  - Variational AE

$$\mathcal{L}\left(x, \hat{x}\right) + \sum_{j} KL\left(q_j\left(z|x\right)||p\left(z\right)\right)$$



Other AEs

- Lab session:
  - AEs in the lecture
  - Denoise Autoencoder
  - LSTM Autoencoder
- Lab task
  - How to use AEs for supervised classification?
  - For example, MNIST digit classification

- Next week:
  - Conditional VAE
  - GANs