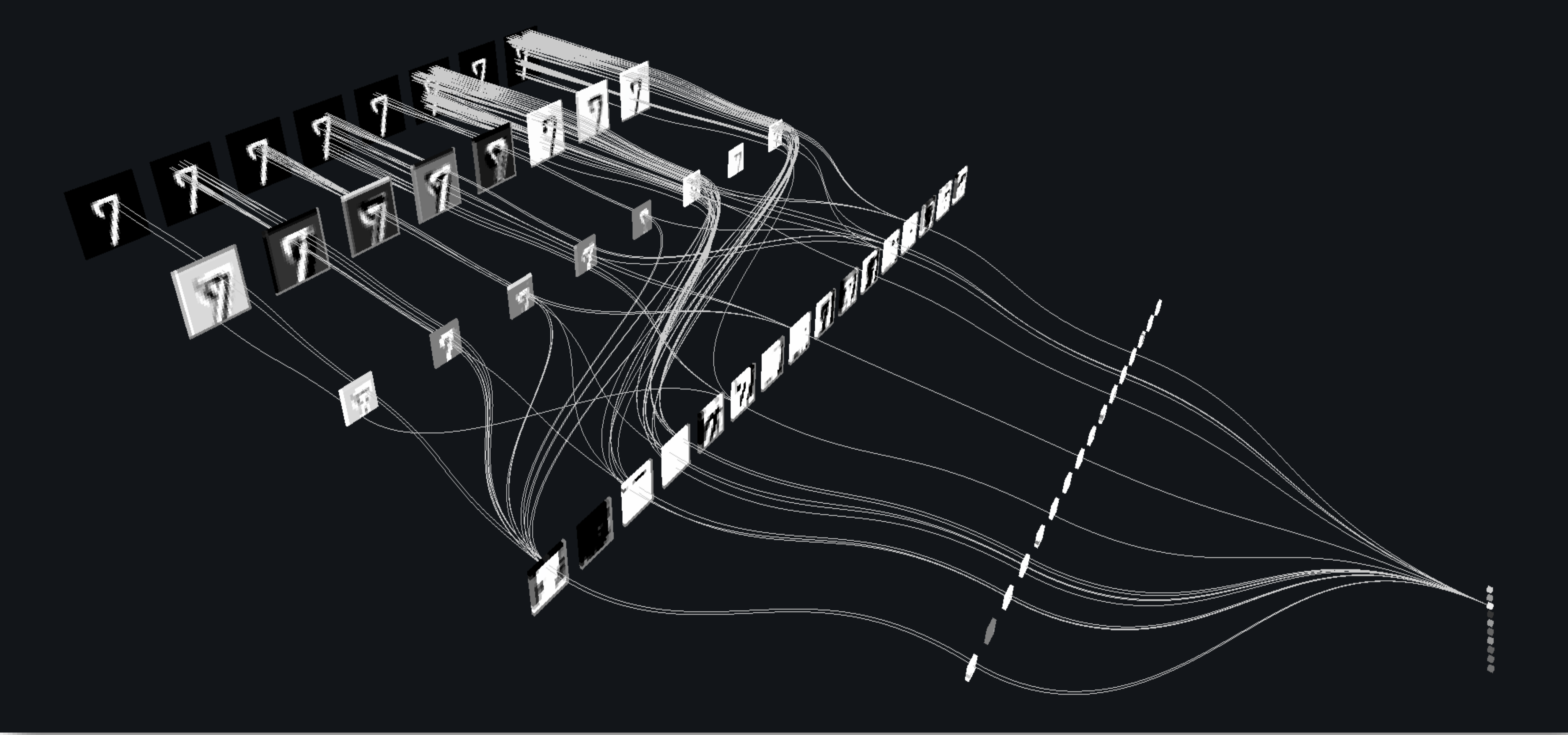
Deep Learning

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
    - Generative adversarial networks (GANs)
  - **Recurrent Neural Networks (RNNs)**
    - Bidirectional recurrent neural networks (BRNN)
    - LSTM
  - **Recursive Neural Networks**

Unsupervised Learning

Motivation and Strengths:

- Unsupervised learning is **not expensive** and **time consuming** like supervised learning.
- Unsupervised learning requires **no human intervention**.
- Unlabeled data is **easy** to **find** with large quantities, unlike labeled data which is scarce.

- **Weaknesses**

- More difficult than supervised learning because there is NO **Single objective** (like test set accuracy)

Unsupervised Feature Learning

- Train representations with unlabeled data.
  - Minimize an *unsupervised* training loss.
    - Often based on generic priors about characteristics of good features
    - Usually train 1 layer of features at a time.

UPNs

- Unsupervised pre-trained networks (UPNs)
  - Motivation: representation **leaning** and **transfer learning**
  - Deep belief networks (DBNs)
  - Autoencoders
  - Generative adversarial networks (GANs)

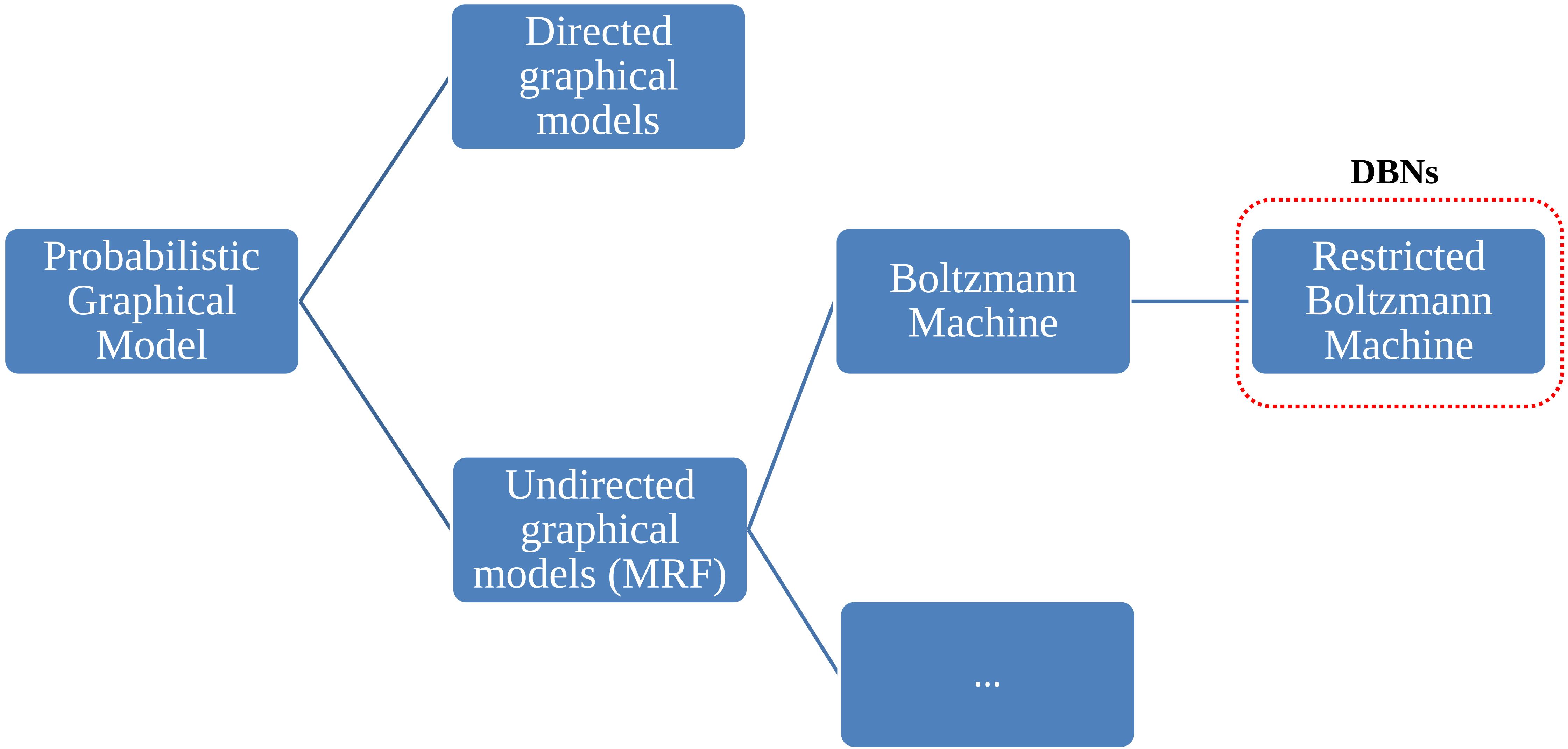
DBN

- DBN's prerequisite
  - MRF
  - Sampling
  - RBMs
  - **DBNs**
    - **Reconstruction**
    - **Classification**

Restricted Boltzmann Machine (RBM)

- RBMs are building blocks for the multi-layer leaning architectures, e.g. DBNs.
- RBMs are a special case of general **Boltzmann Machines** (BMs).
- BMs are a particular form of **Markov Random Field (MRF)**, a.k.a. **Markov networks** or **undirected graph models**.

DBNs


Directed graphical models
DBNs
Probabilistic Graphical Model
Boltzmann Machine
Restricted Boltzmann Machine
Undirected graphical models (MRF)
...

… topic covered so far

- BMs
- RMBs
- Joint distribution
- Potential functions
- Cliques
- Maximal cliques
- Energy function
- Energy function.
- Conditional independent
- Ascending gradient
- Transformation method
- Rejection sampling
- Gibbs sampling

… topic covered so far

- RBMs Training
- RBMs applications
  - Feature extraction

DBN

- **Deep Belief Network**
  - A generative graphical model
  - A class of deep neural networks
  - Composed of latent variables, e.g. hidden units
  - No intra-layer connects
  - With inter-layer connections
  - Unsupervised learning for feature detector
  - Able to reconstruct its inputs
  - Supervised learning for classification

Geoffrey E. Hinton

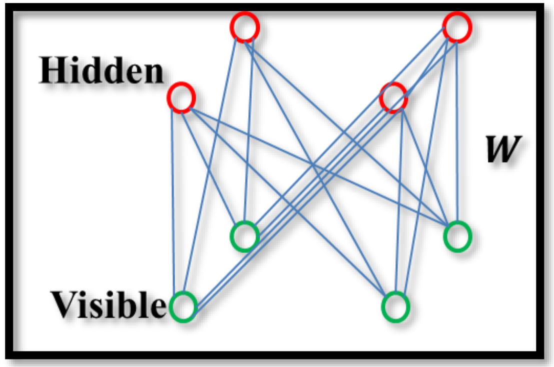
DBN

- DBNs are used to solve two problems
  - **The inference problem**:
    - Infer the states of the unobserved variables.
  - **The learning problem**:
    - Adjust the interactions between variables to make the network more likely to generate the observed data.
    - Greedy layer by layer RBM training (optimize lower bound) and fine tuning
    - Contrastive divergence for RBM training

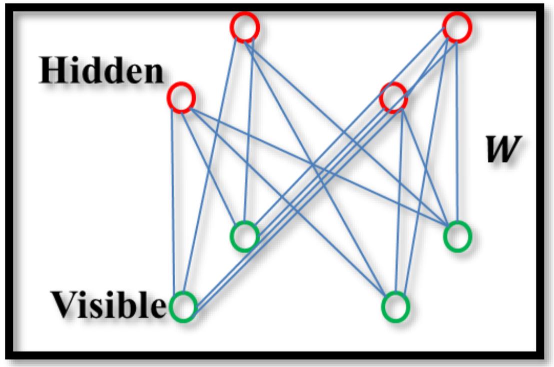
DBN

- **Procedure**
  - First, train a layer of features based on the training data.
  - Then, treat the activations of the trained features as if they were pixels and learn features of features in a second hidden layer.
- Each time we add another layer of features we improve a lower bound on the log probability of the training data.

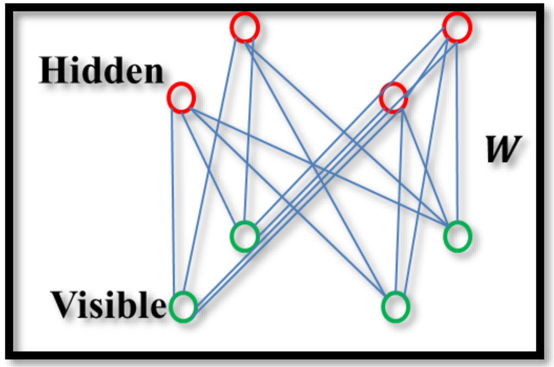
RBM



RBM



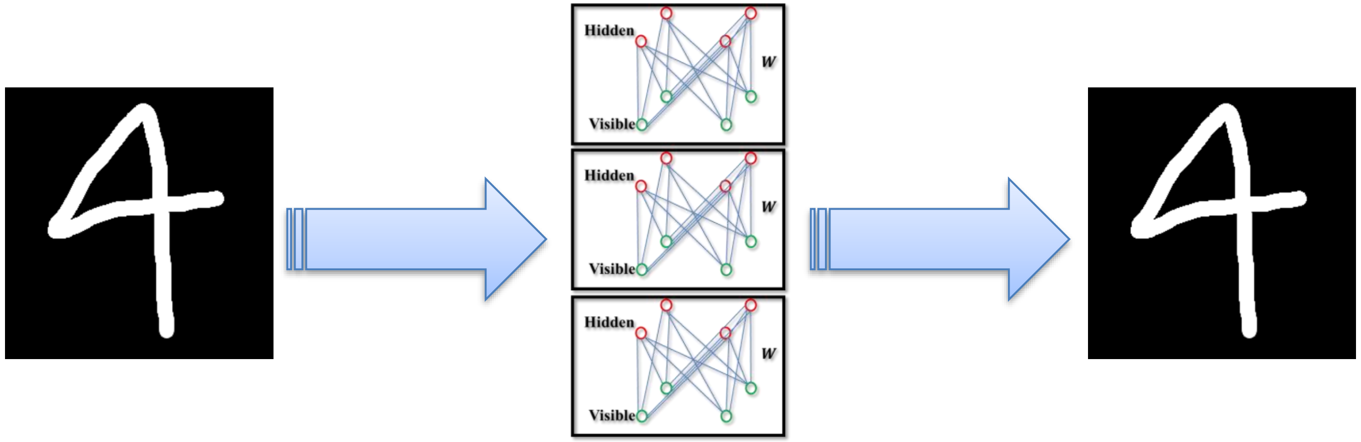
RBM



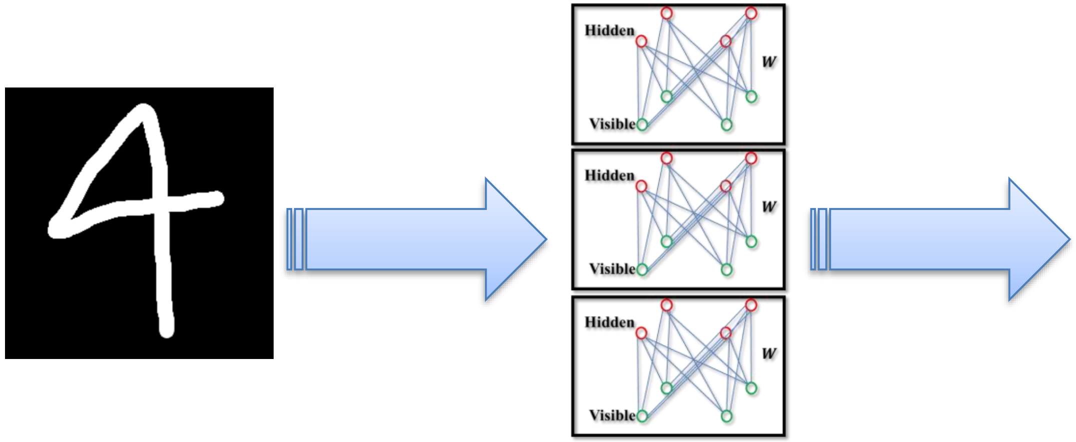
DBN

Applications

- **Reconstruction**



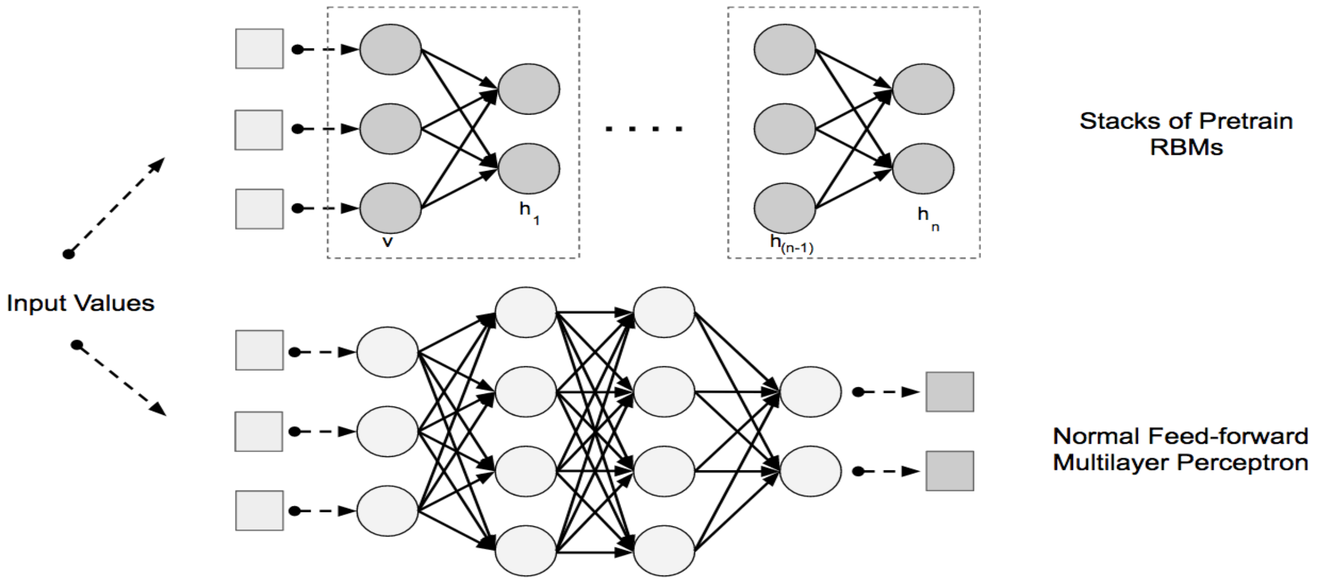
- **Classification**

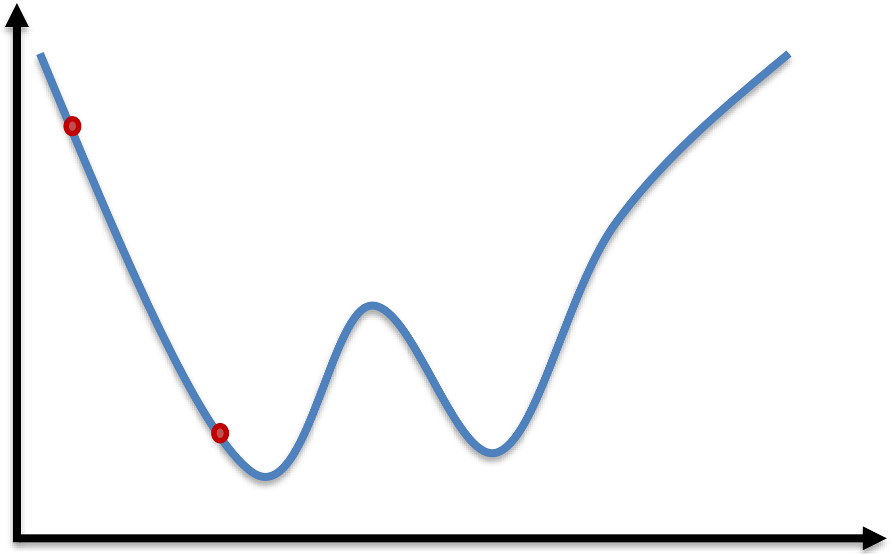


0, 1, 2, 3, 4, 5, 6, 7, 8, 9

DBNs

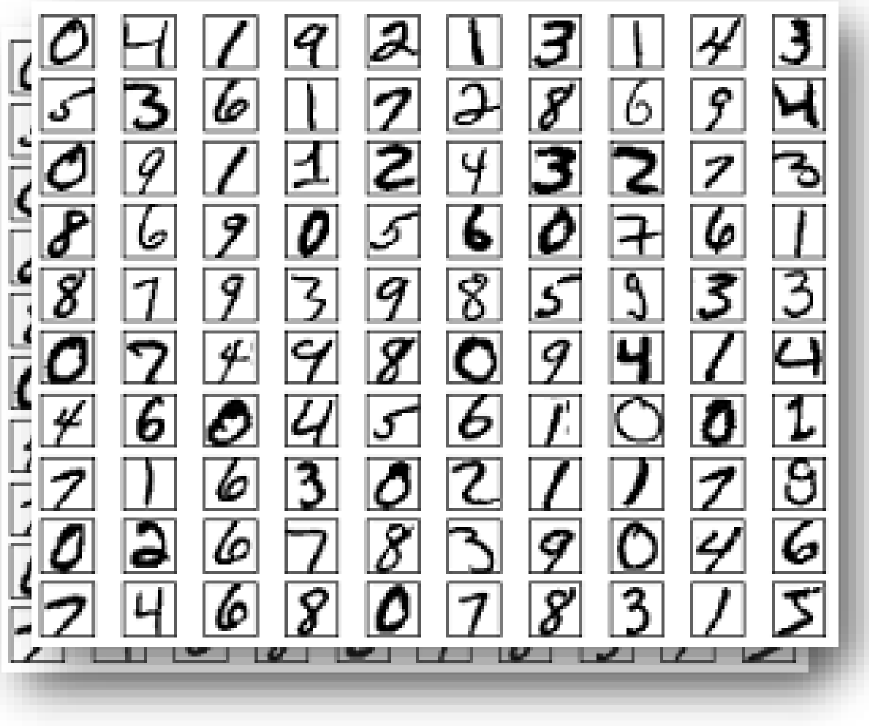
- **Components**:
  - **Pre-train phase**: using **RBM** layers.
    - As a general search of the parameter space in an unsupervised fashion based on training data.
  - **Fine-tune phase**: using **forward Propagation**.
    - Specializing the network and its features for the desired task., e.g. **classification**





Reconstruction

- Data dimensionality reduction
- MNST dataset
  - Handwritten digits
  - Size: 28×28
  - 60,000 training images
  - 10,000 test images

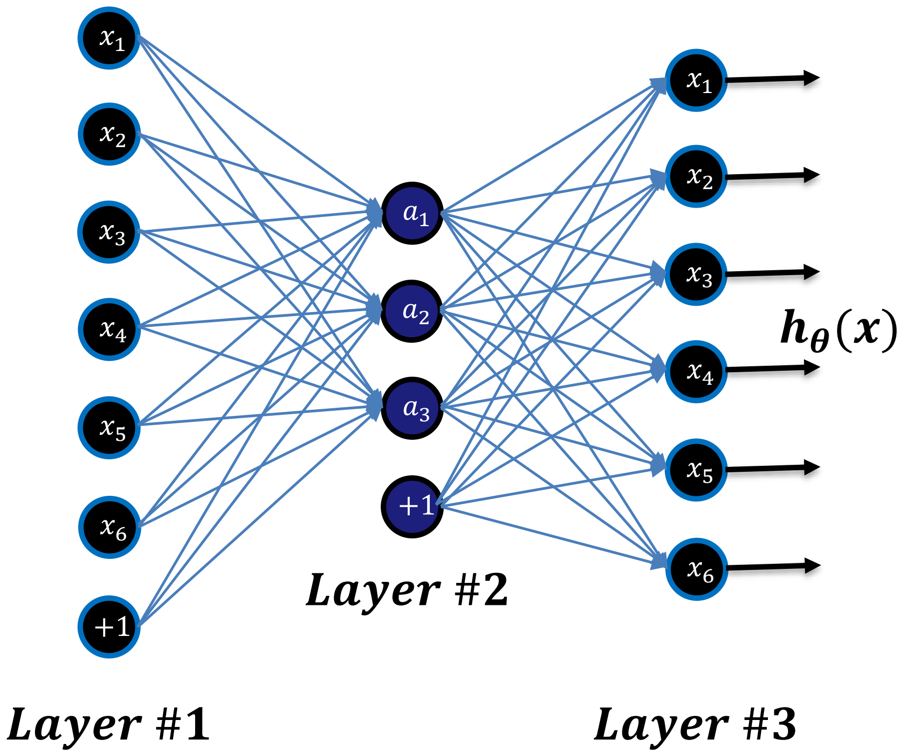


Reconstruction

- Data dimensionality reduction
  - Autoencoder
  - Network is trained to output the input (learn identify function)

$$h_\theta(x) \approx x$$

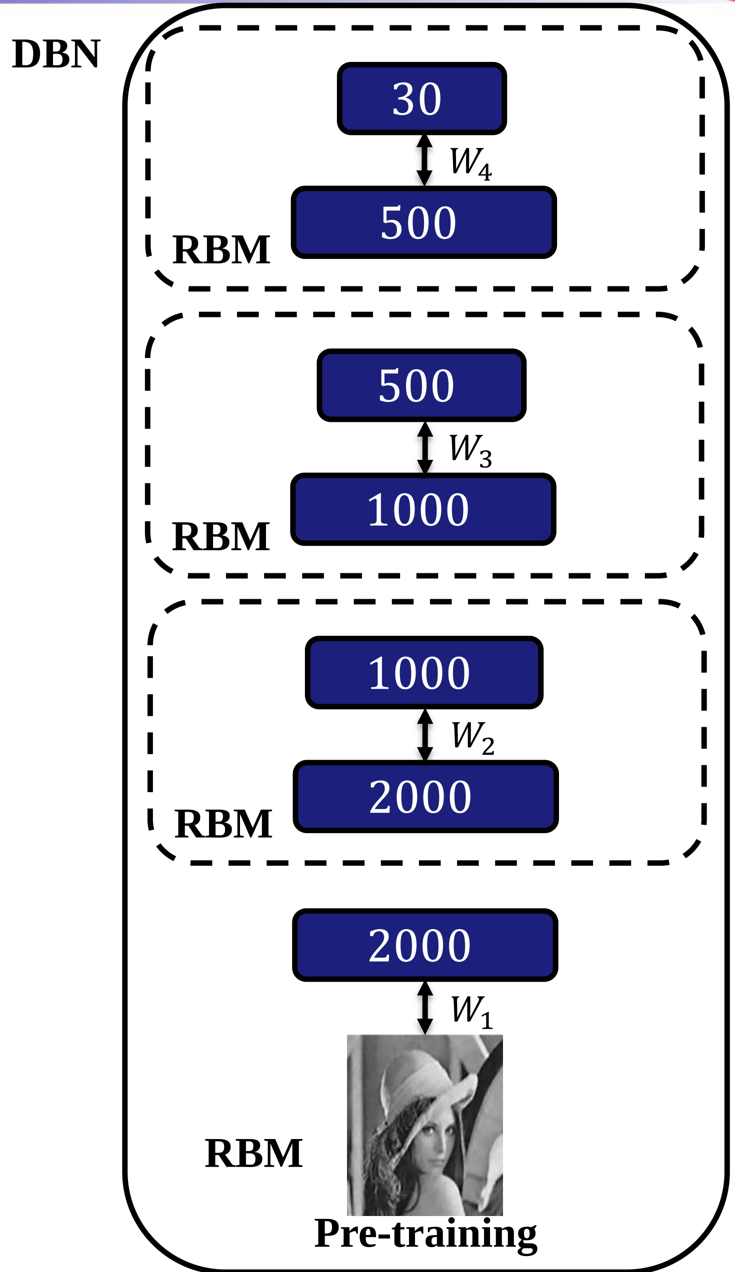
  - It is difficult to optimize the weights in nonlinear autoencoders that have multiple hidden layer (2-4)
  - Autoencoders typically find poor local minima



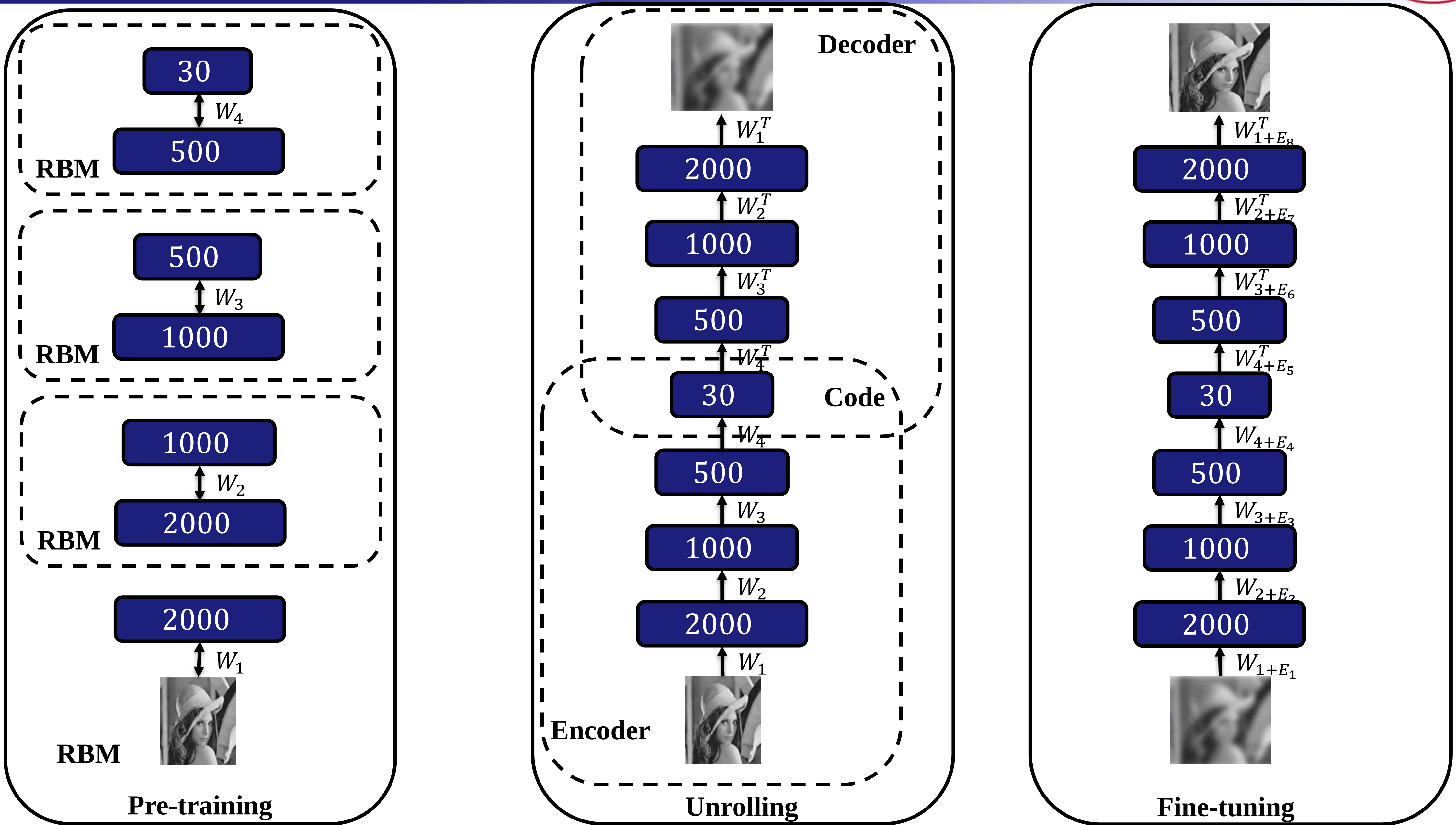
DBN

Reconstruction

- Data dimensionality reduction
- DBN is used to provide suitable initial weights to improve the performance of the gradient.



DBN


30
W_4
500
RBM
500
W_3
1000
RBM
1000
W_2
2000
RBM
2000
W_1
RBM
Pre-training
Decoder
W_1^T
2000
W_2^T
1000
W_3^T
500
W_4^T
30
Code
W_4
500
W_3
1000
W_2
2000
W_1
Encoder
Unrolling
$W_{1+E_8}^T$
2000
$W_{2+E_7}^T$
1000
$W_{3+E_6}^T$
500
$W_{4+E_5}^T$
30
W_{4+E_4}
500
W_{3+E_3}
1000
W_{2+E_2}
2000
W_{1+E_1}
Fine-tuning

DBN

Reconstruction

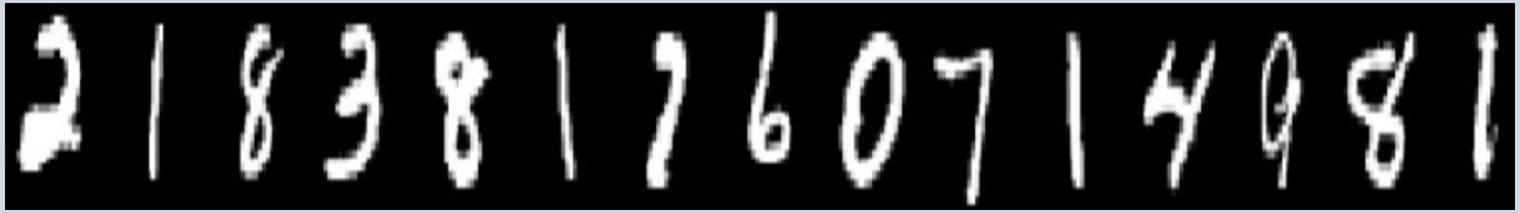
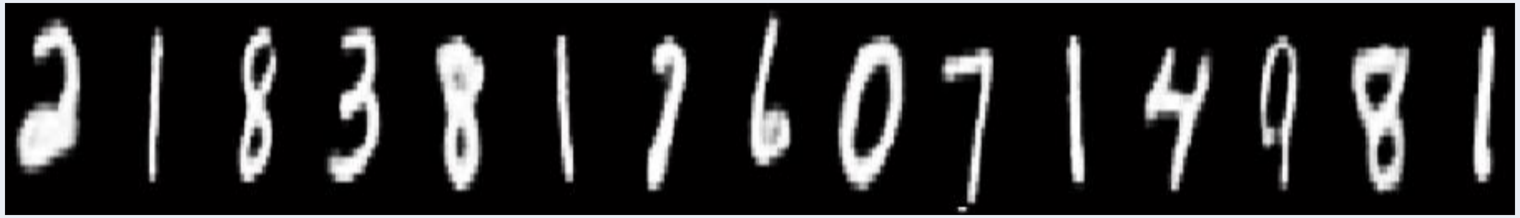
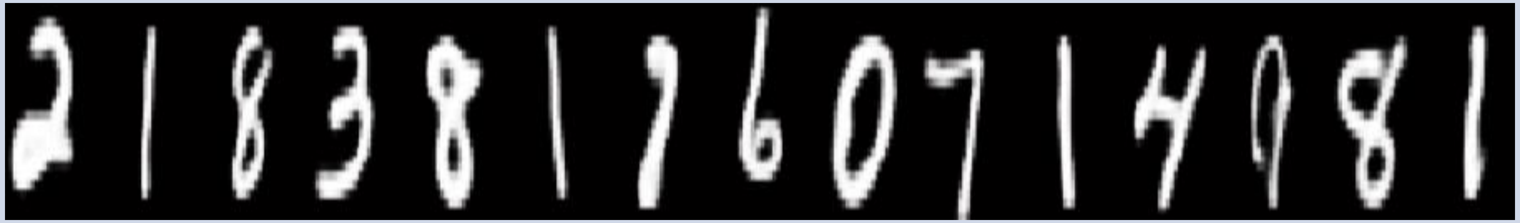
- Data dimensionality reduction, [784-1000-500-250-30]

Operation	Output	Error
A random test image from each class		0
Reconstruction by a 30-dimensional DBN		3.00
Reconstruction by a 30-dimensional logistic PCA		8.01
Reconstruction by a 30-dimensional standard PCA		13.87

DBN

Reconstruction

- Data dimensionality reduction,

Operation	Output
A random test image	
Reconstruction	
Reconstruction after tuning	

DBN

Reconstruction

- Data dimensionality reduction, [625-2000-1000-500-30]

Operation	Output	Error
Random samples from the test data set		0
Reconstruction by a 30-dimensional DBN		126
Reconstruction by a 30-dimensional PCA		135

Classification

- Classification in an RBM
  - **Generative**:

$$\mathcal{L}_{gen} = -\sum_{i=1}^{N_\Gamma} \log P\left(v^{(i)}, y^{(i)}\right)$$

  - **Discriminative**:

$$\mathcal{L}_{disc} = -\sum_{i=1}^{N_\Gamma} \log P\left(y^{(i)} \mid v^{(i)}\right)$$

  - **Hybrid**:

$$\mathcal{L}_{hybrid}(\Gamma) = \mathcal{L}_{disc}(\Gamma) + \alpha \mathcal{L}_{gen}(\Gamma)$$

Resources

- Geoffrey E. Hinton's readings (with source code available for DBN)
  http://www.cs.toronto.edu/~hinton/csc2515/deeprefs.html
- Notes on Deep Belief Networks http://www.quantumg.net/dbns.php
- Hinton's Tutorial, http://videolectures.net/mlss09uk_hinton_dbn/
- Geoffrey E. Hinton's http://www.cs.toronto.edu/~hinton
- Ruslan Salakhutdinov http://www.utstat.toronto.edu/~rsalakhu/
- Yee-Whye Teh http://www.gatsby.ucl.ac.uk/~ywteh/
- Yoshua Bengio www.iro.umontreal.ca/~bengioy
- Marcus Frean http://ecs.victoria.ac.nz/Main/MarcusFrean
- Rob Fergus http://cs.nyu.edu/~fergus/pmwiki/pmwiki.php
- Deng, Li, and Dong Yu. "Deep learning: methods and applications.", 2014
- Bishop, Christopher M. "Pattern recognition and machine learning", 2006.

Office: #432, Dayang AI Center, **Phone**: 010-8999-8586, **email**: piran@sejong.ac.kr