Unsupervised Learning

Dec 22nd, 2016

Applied Deep Learning

Yun-Nung (Vivian) Chen

WWW.CSIE.NTU.EDU.TW/~YVCHEN/F105-ADL

National Taiwan University

Slide credit from Hung-Yi Lee

Introduction

Big data ≠ Big annotated data

Machine learning techniques include:

- Supervised learning (if we have labelled data)
- Reinforcement learning (if we have an environment for reward)
- Unsupervised learning (if we do not have labelled data)

What can we do if there is no sufficient training data?

Semi-Supervised Learning

Labelled data

cat

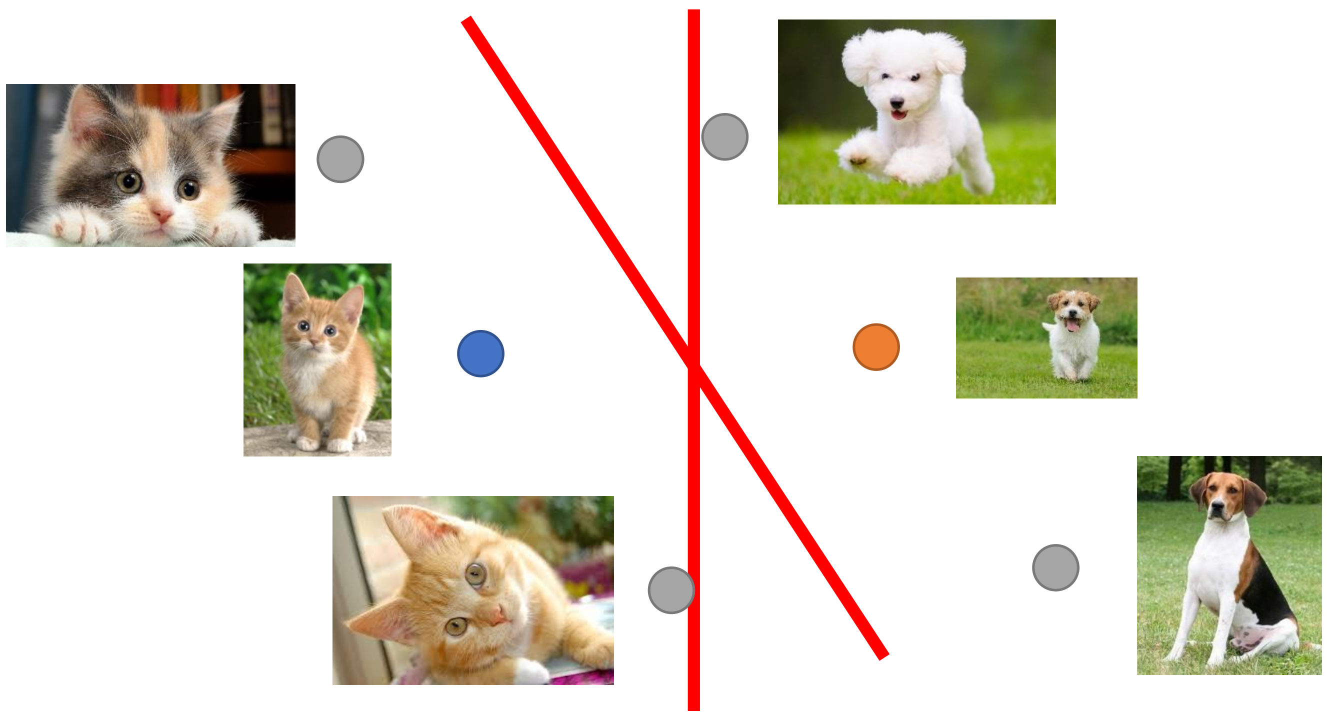
dog

Unlabeled data

(Image of cats and dogs without labeling)

Semi-Supervised Learning

Why semi-supervised learning helps?



The distribution of the unlabeled data provides some cues

Transfer Learning

Labelled data

cat

dog

Labeled data

elephant

elephant

tiger

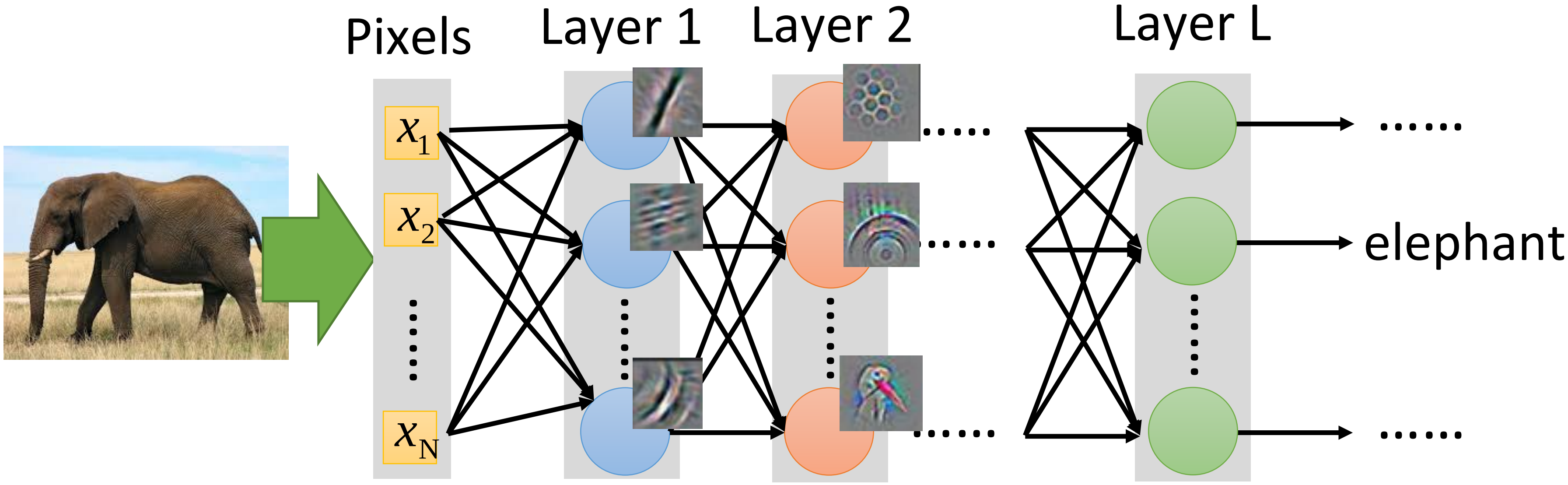
tiger

Not related to the task considered

Transfer Learning

Widely used on image processing

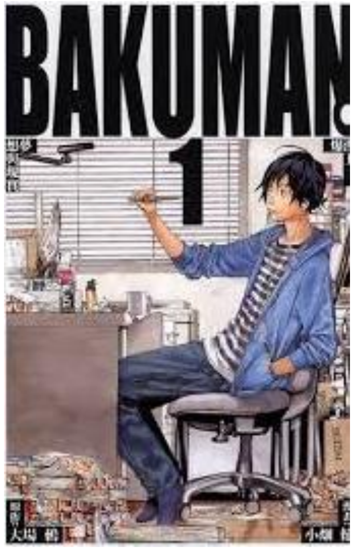
- Using sufficient labeled data to learn a CNN
- Using this CNN as feature extractor



Transfer Learning Example

研究生 online	***漫畫家 online***
研究生	漫畫家
指導教授	責編
跑實驗	畫分鏡
投稿期刊	投稿 jump

研究生
生存守則



爆漫王

Self-Taught Learning

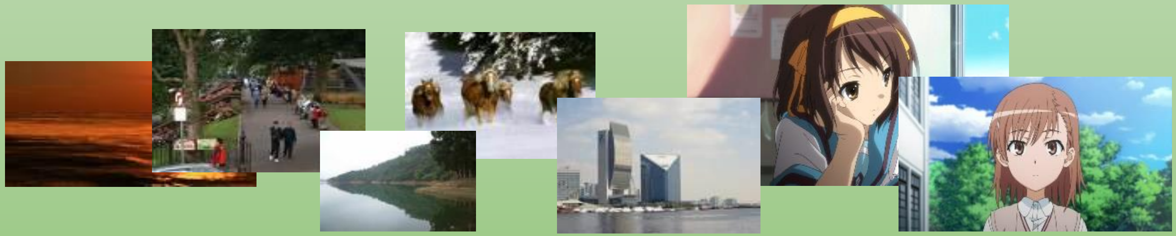
The unlabeled data sometimes is not related to the task

Labelled data

cat

dog

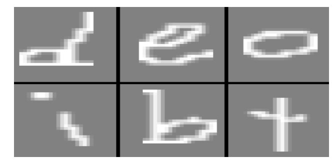
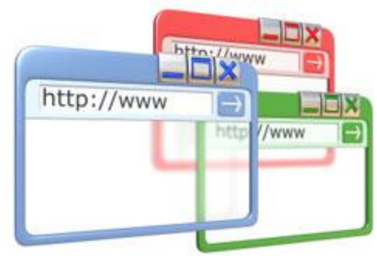
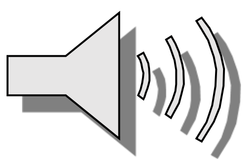
Unlabeled data



(Just crawl millions of images from the Internet)

Self-Taught Learning

The unlabeled data sometimes is not related to the task

	Labelled data	Unlabeled data
Digit Recognition	 Digits	 character
Document Classification	 News	 Webpages
Speech Recognition	 Taiwanese	YouTube English Chinese ……

Why can we use unlabeled and unrelated data to help our tasks?

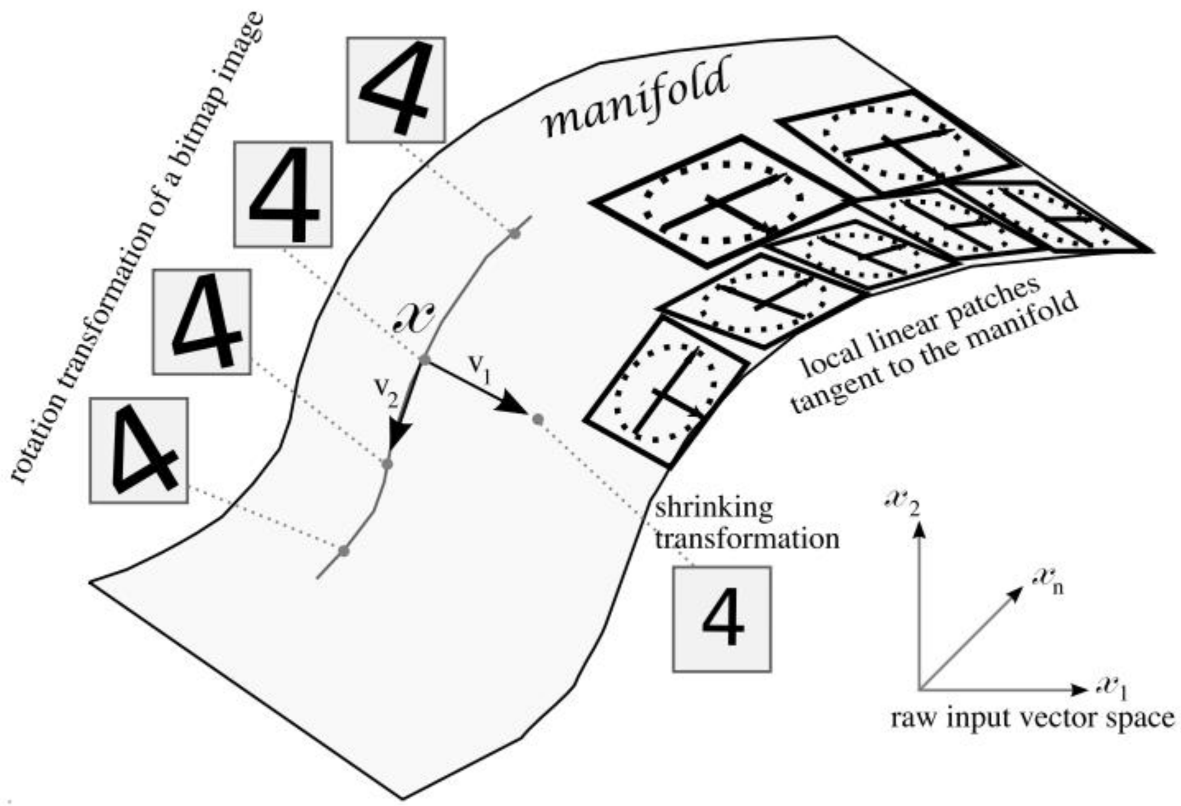
Self-Taught Learning

How does self-taught learning work?

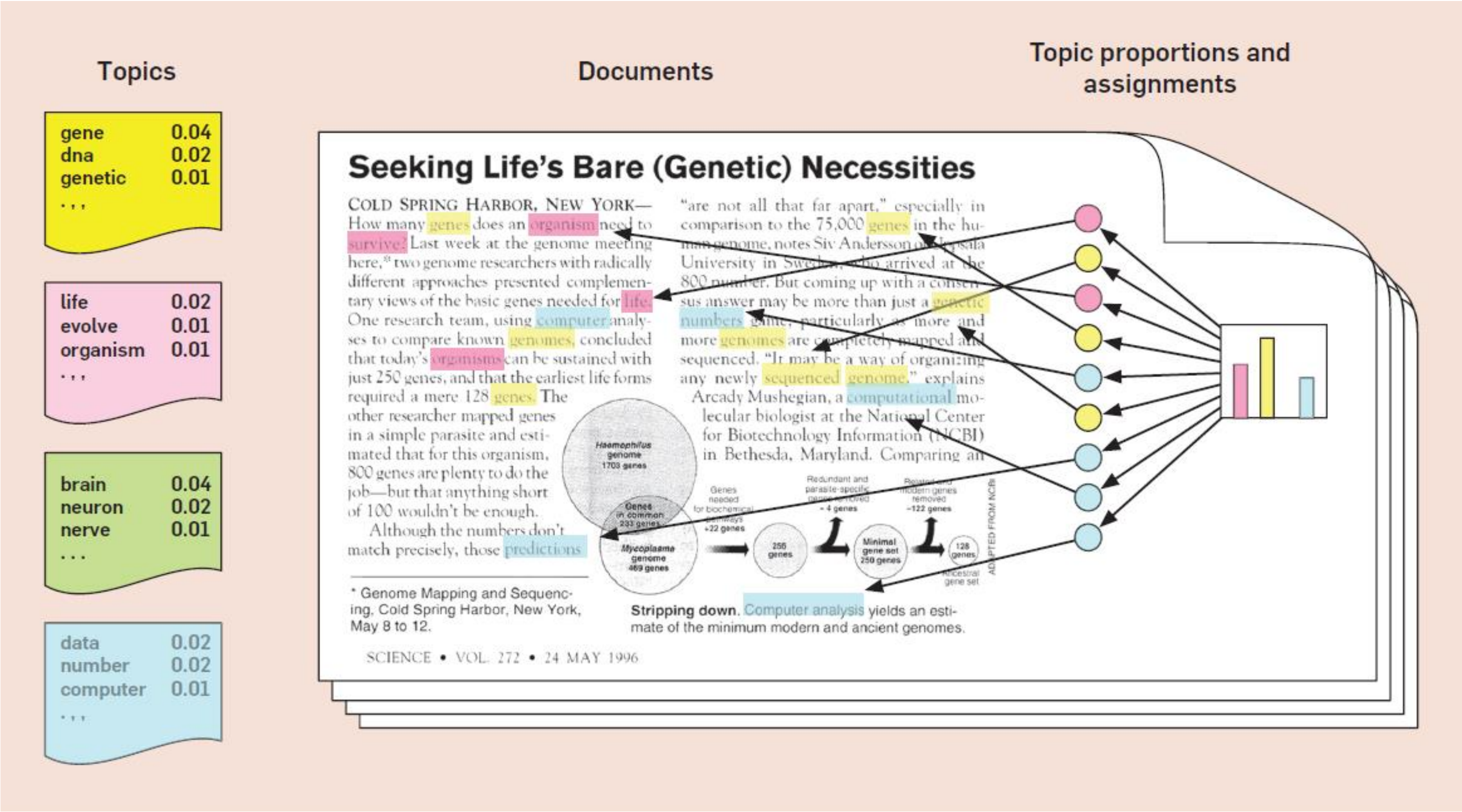
Why does unlabeled and unrelated data help the tasks?

Finding latent factors that control the observations

Latent Factors for Handwritten Digits



Latent Factors for Documents



Latent Factors for Recommendation System




單純呆	A	✓	✓		
傲嬌	B			✓	✓
	C	✓	? ✓		

Latent Factor Exploitation

Handwritten digits

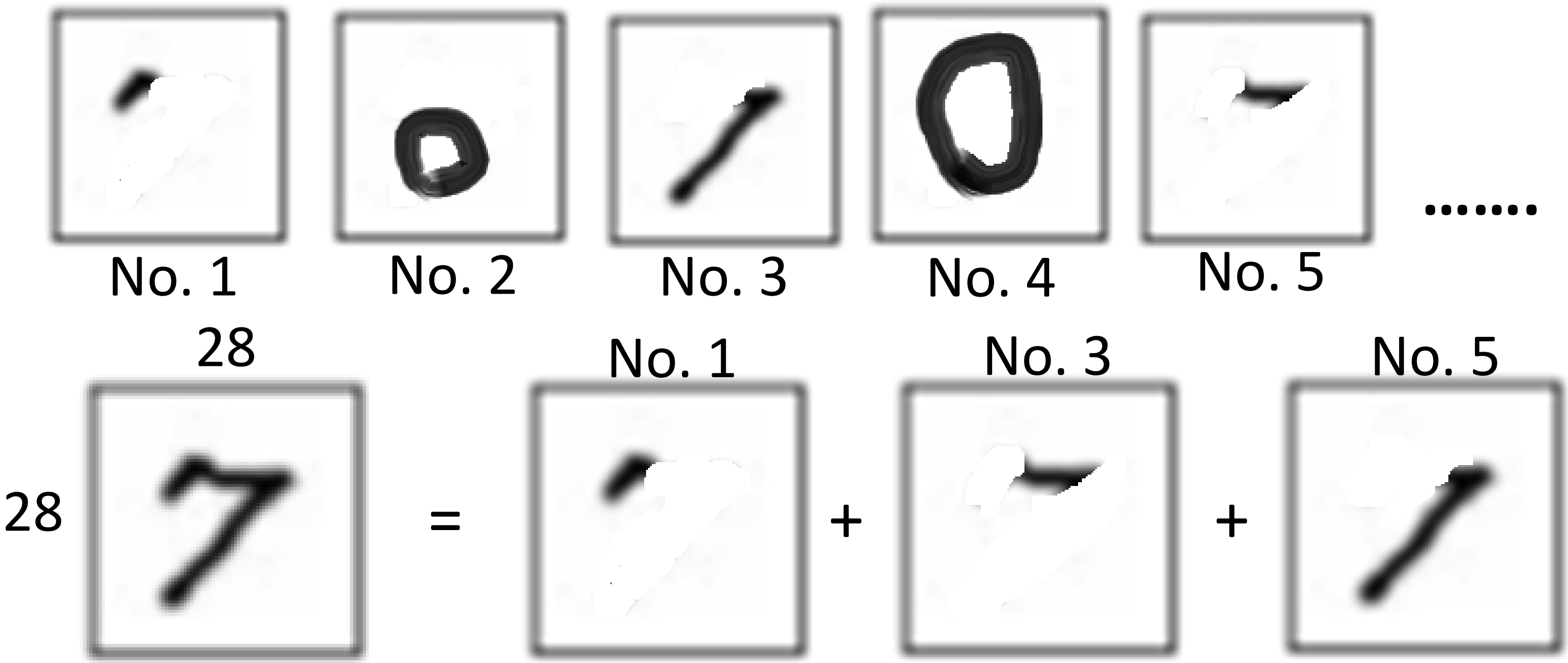
The handwritten images are composed of **strokes**

<u>Strokes (Latent Factors)</u>

No. 1 No. 2 No. 3 No. 4 No. 5 …….

Latent Factor Exploitation

Strokes (Latent Factors)



Represented by
28 X 28 = 784 pixels

[1 0 1 0 1 0 ....]
(simpler representation)

Autoencoder

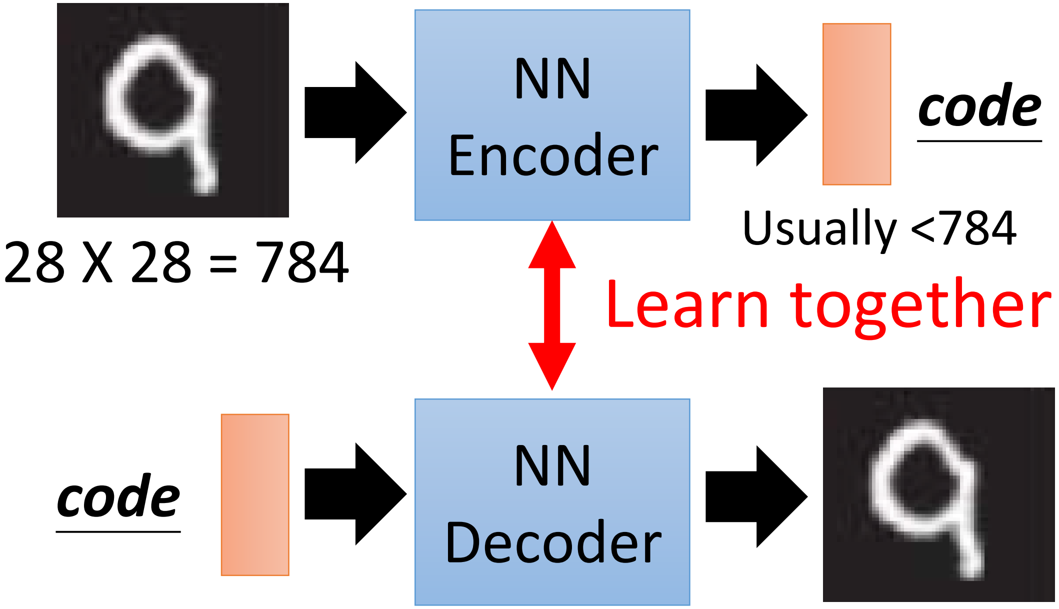
Representation Learning

Autoencoder

Represent a digit using 28 X 28 dimensions

Not all 28 X 28 images are digits

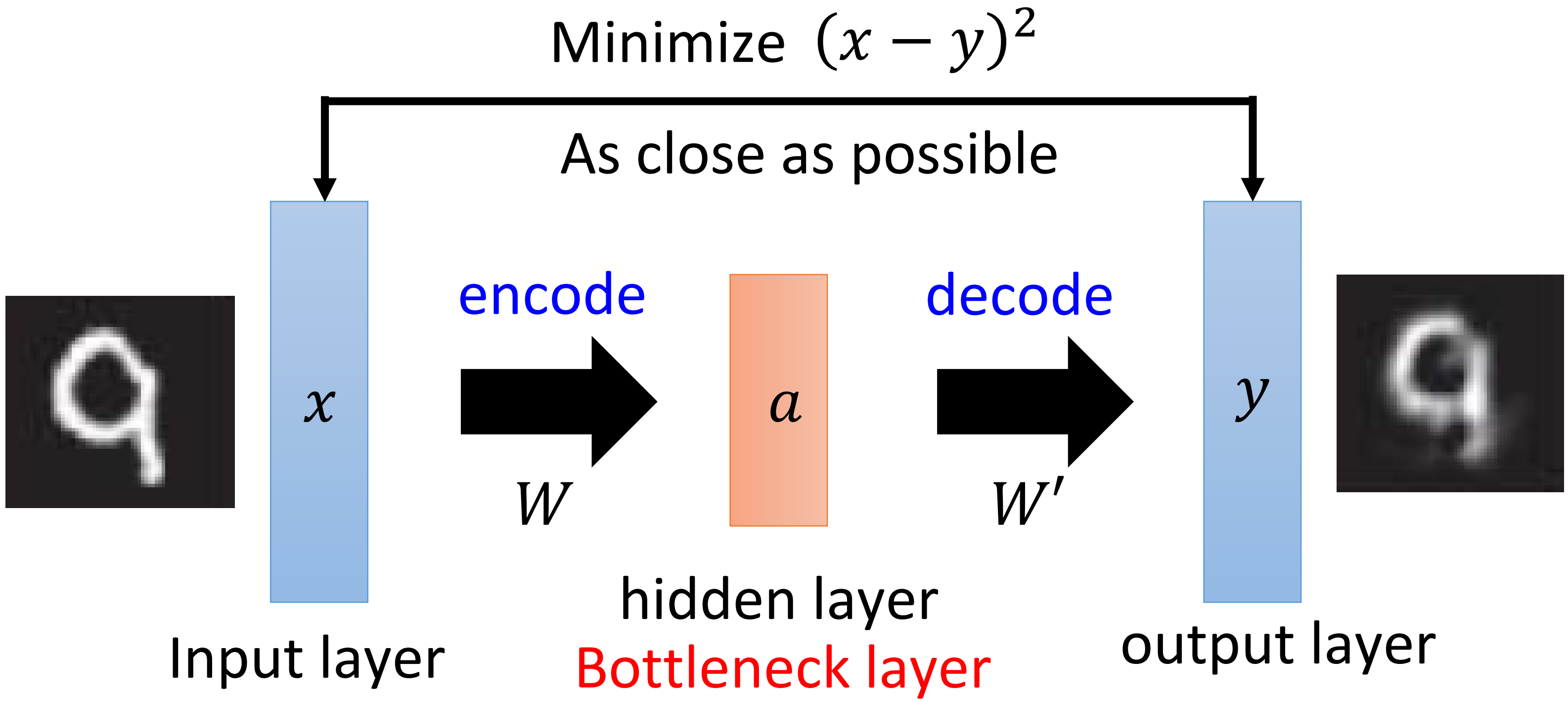
Idea: represent the images of digits in a more compact way



Compact representation of the input object

Can reconstruct the original object

Autoencoder

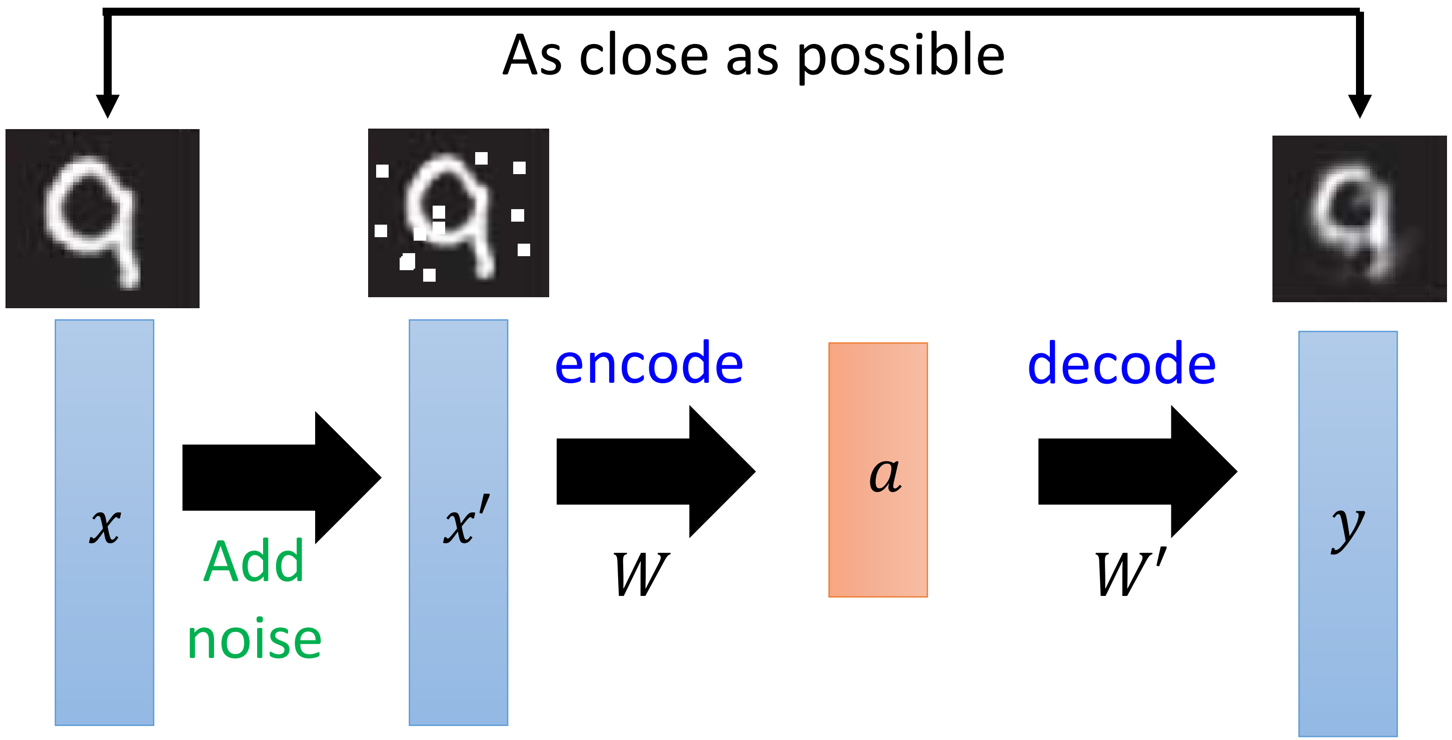


$a=\sigma(Wx+b)$ $y=\sigma(W'a+b')$

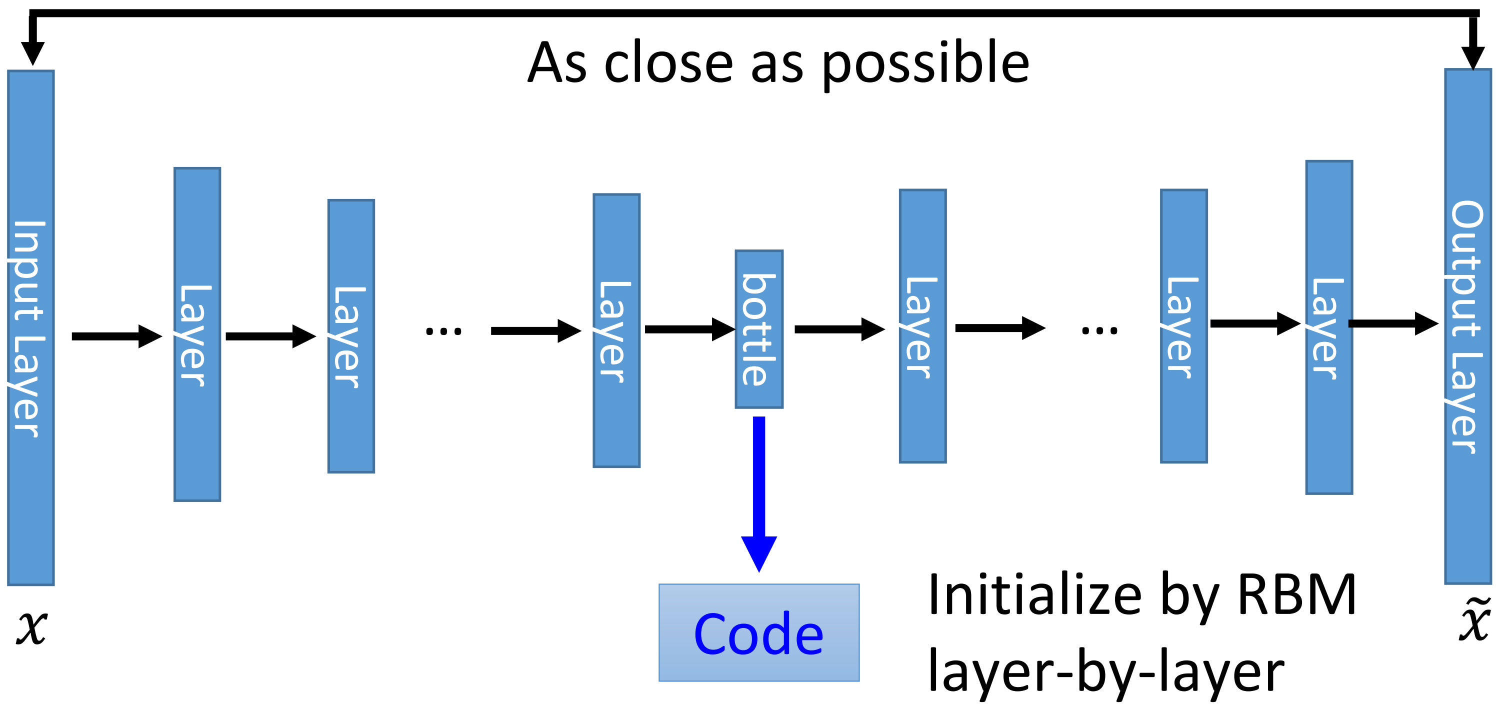
Output of the hidden layer is the code

Autoencoder

De-noising auto-encoder

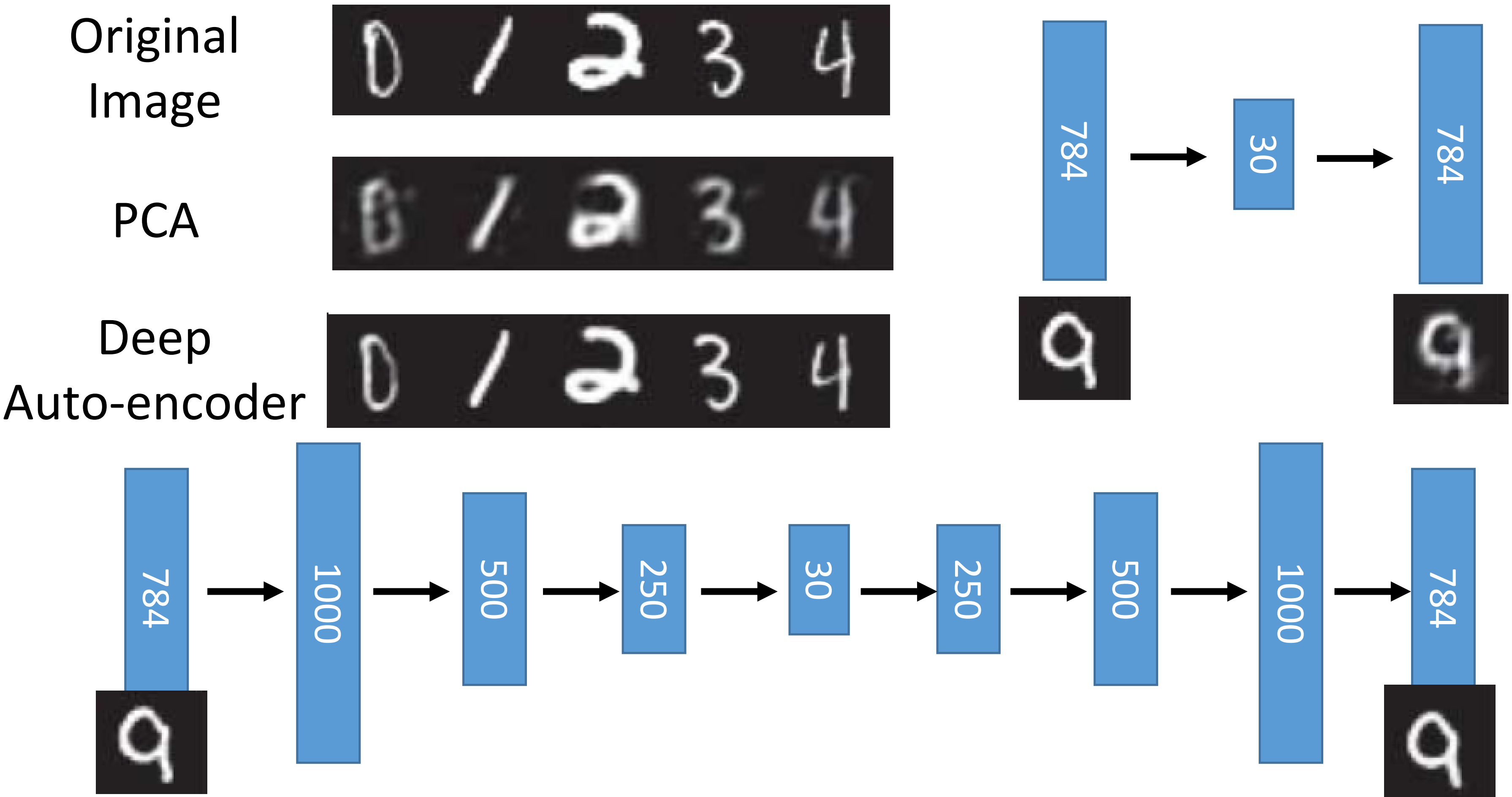


Deep Autoencoder

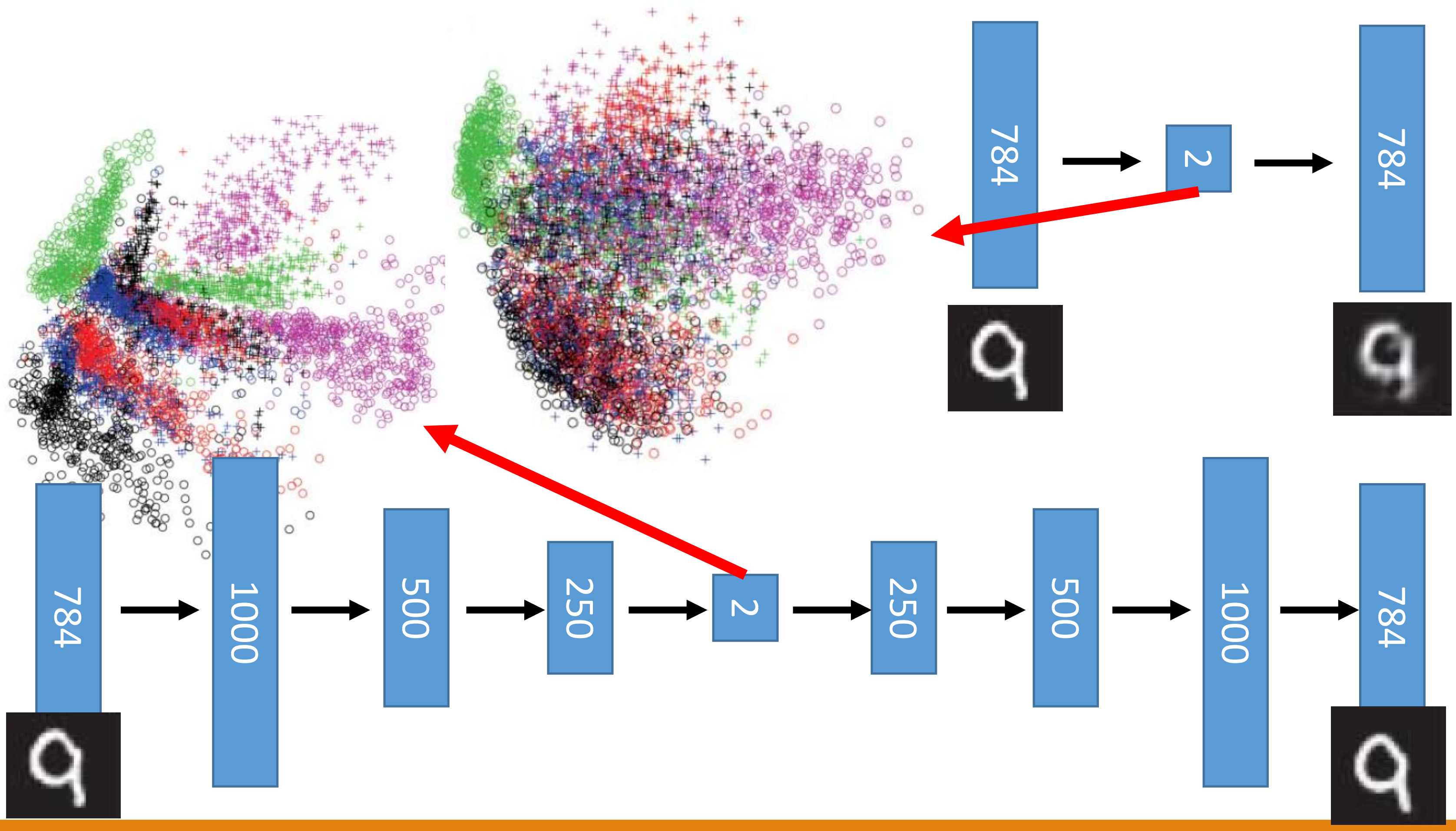


Hinton and Salakhutdinov. "Reducing the dimensionality of data with neural networks," *Science,* 2006.

Deep Autoencoder

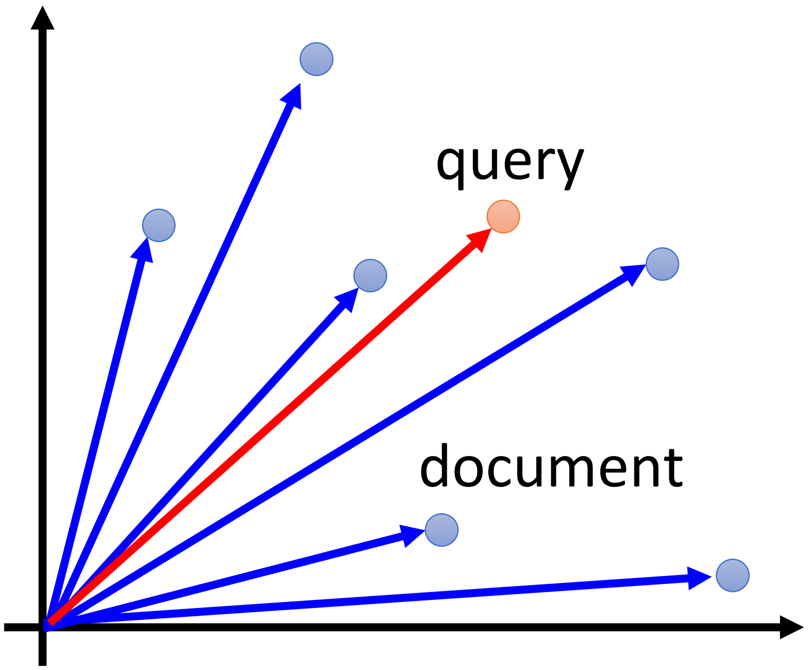


Feature Representation



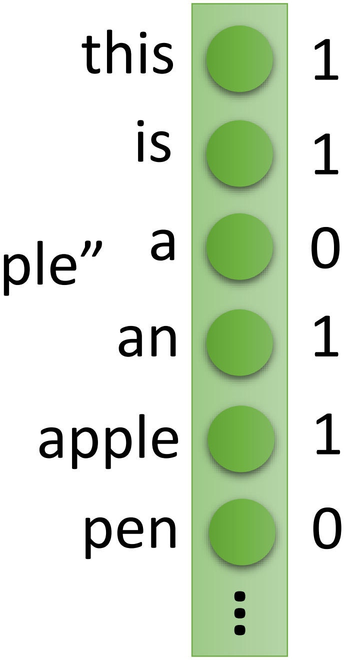
Auto-encoder – Text Retrieval

Vector Space Model



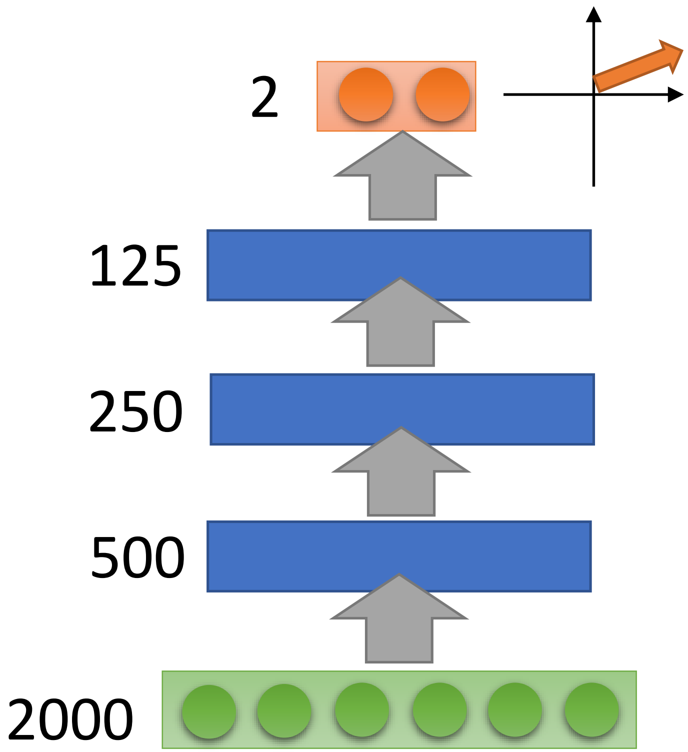
Bag-of-word

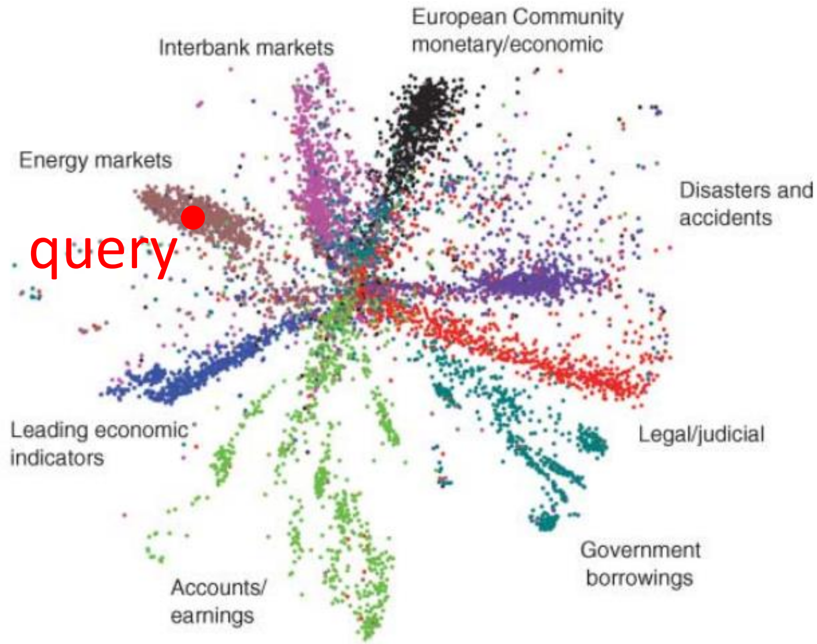
word string:
"This is an apple"



Semantics are not considered

Autoencoder – Text Retrieval





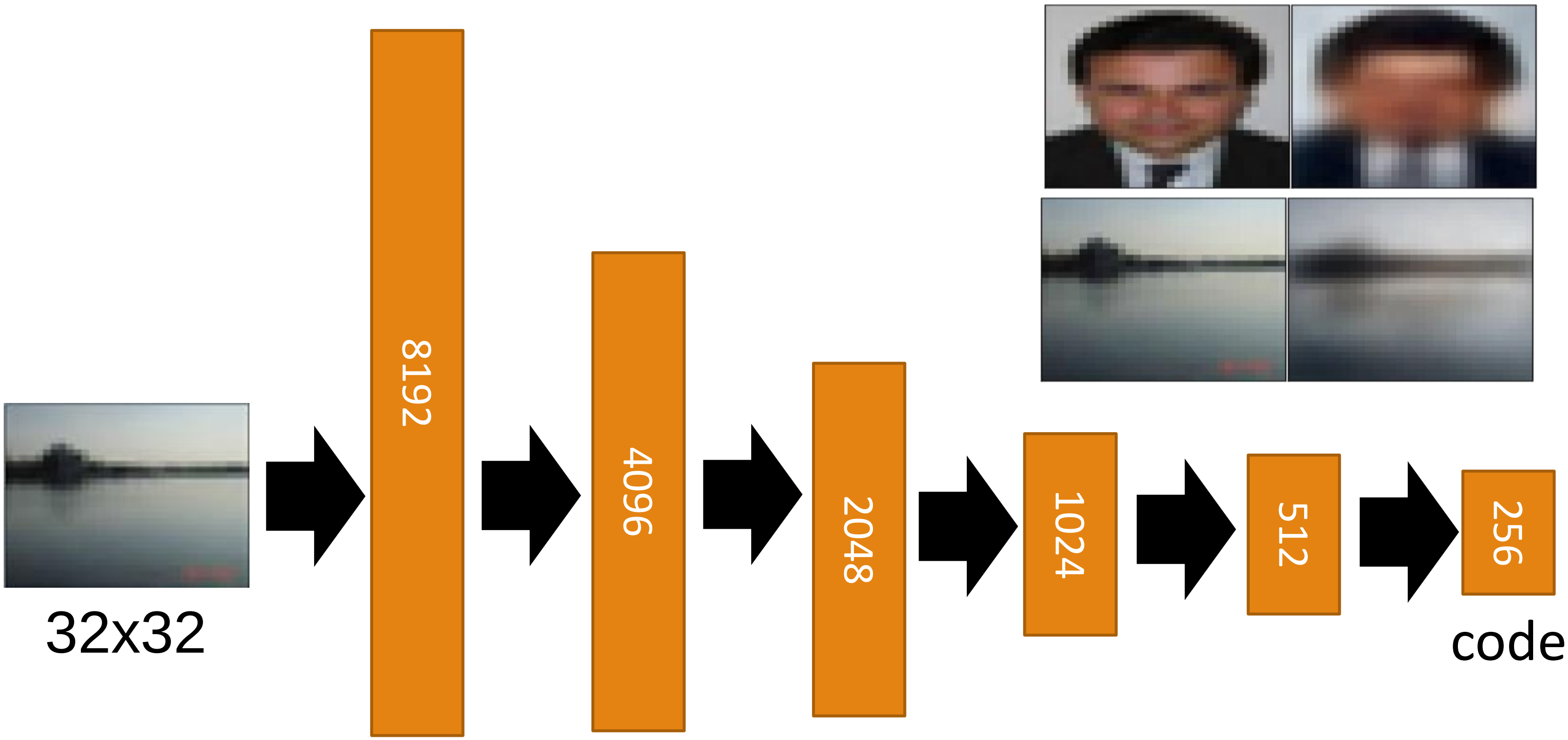
The documents talking about the same thing will have close code

Bag-of-word (document or query)

Autoencoder – Similar Image Retrieval

Retrieved using Euclidean distance in pixel intensity space

Autoencoder – Similar Image Retrieval



(crawl millions of images from the Internet)

Autoencoder – Similar Image Retrieval

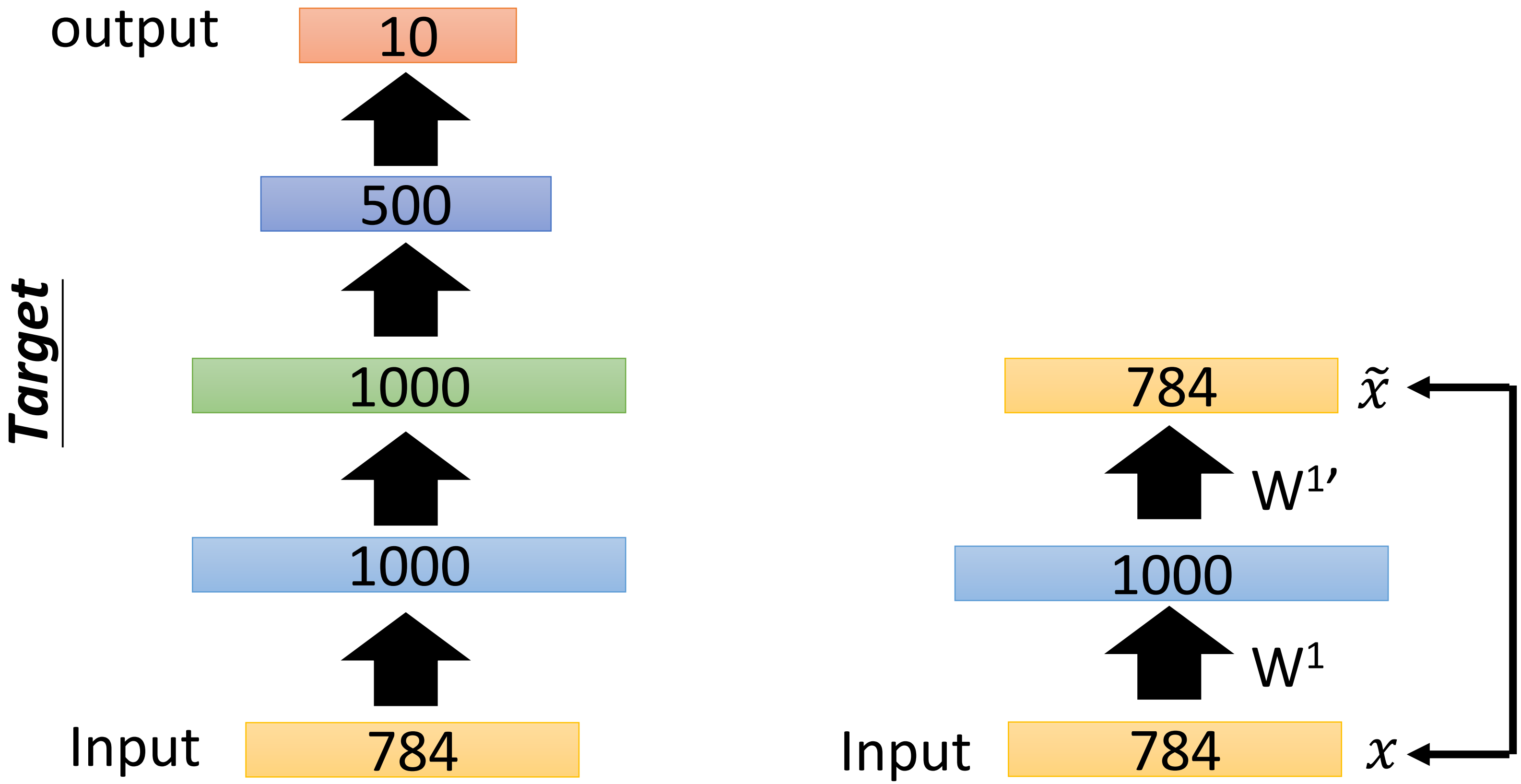
Images retrieved using Euclidean distance in pixel intensity space

Images retrieved using 256 codes

Learning the useful latent factors

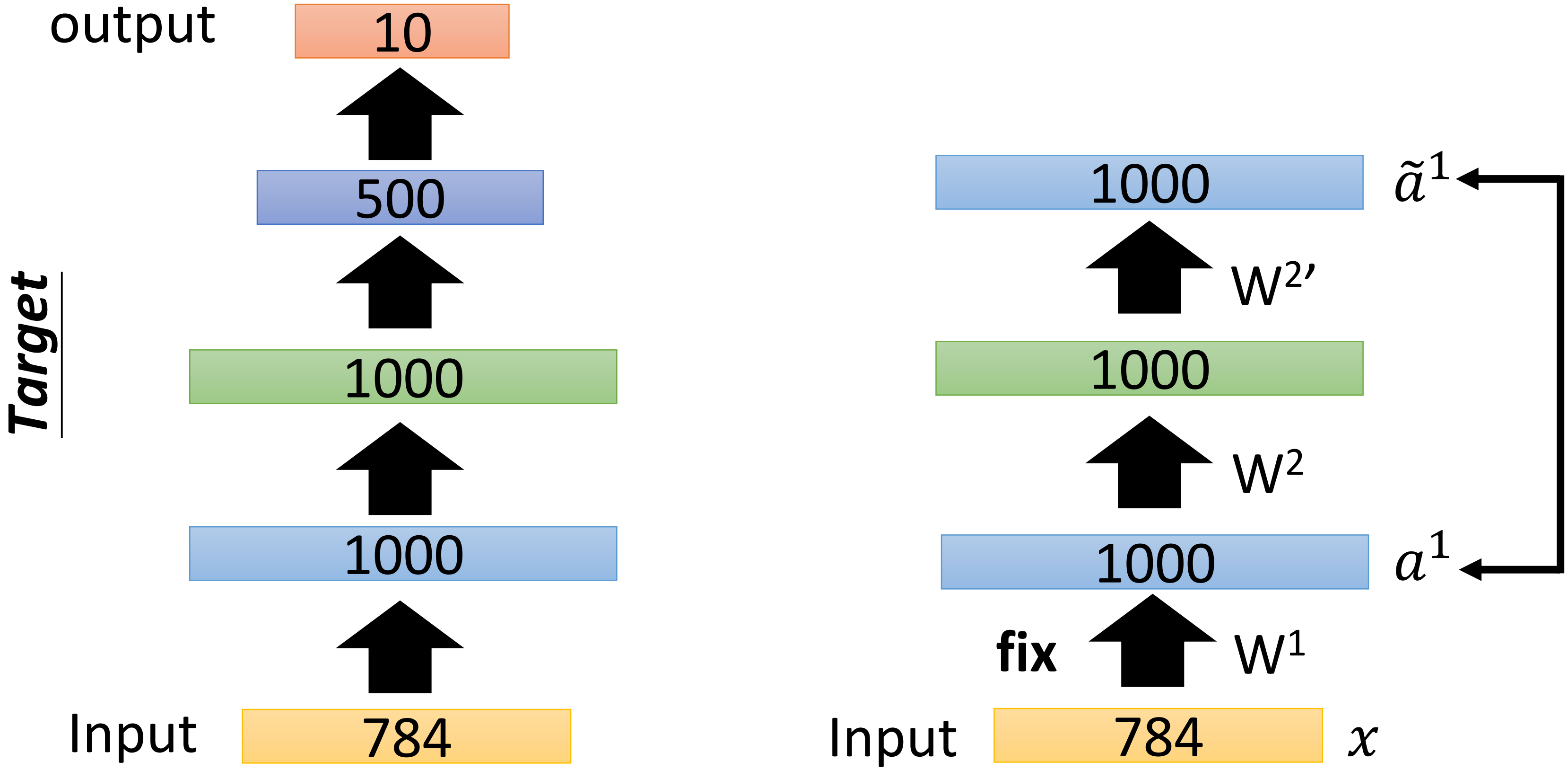
Autoencoder for DNN Pre-Training

Greedy layer-wise pre-training *again*



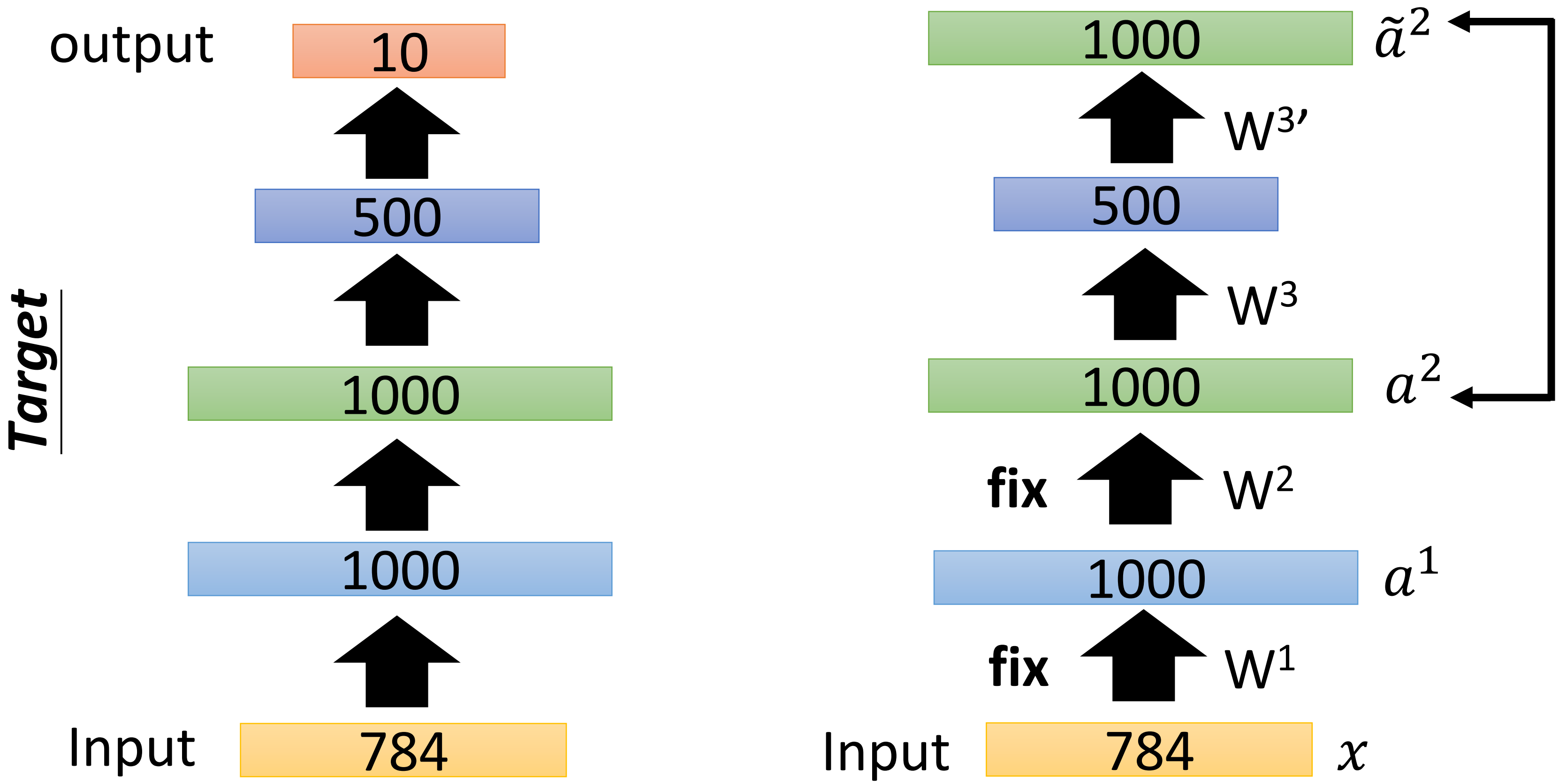
Autoencoder for DNN Pre-Training

Greedy layer-wise pre-training *again*



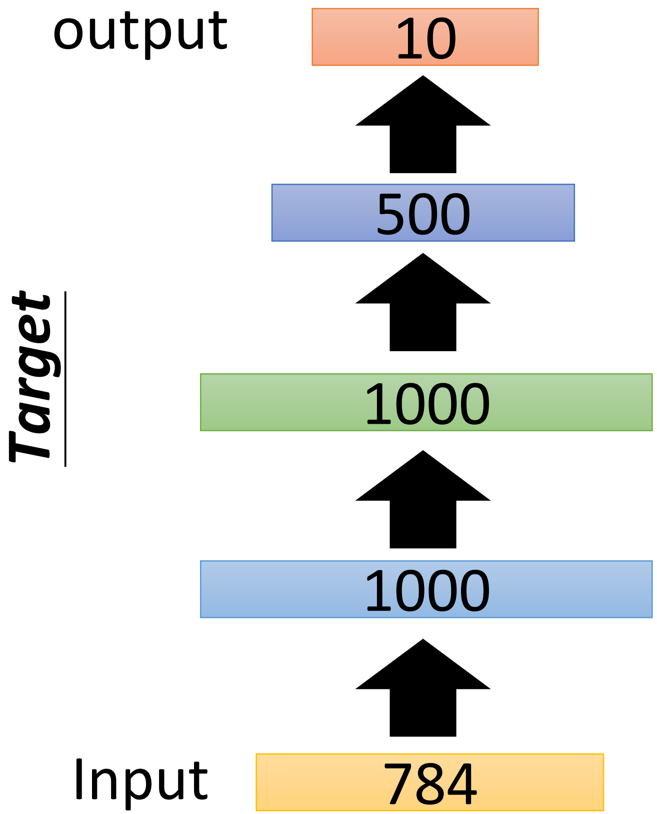
Autoencoder for DNN Pre-Training

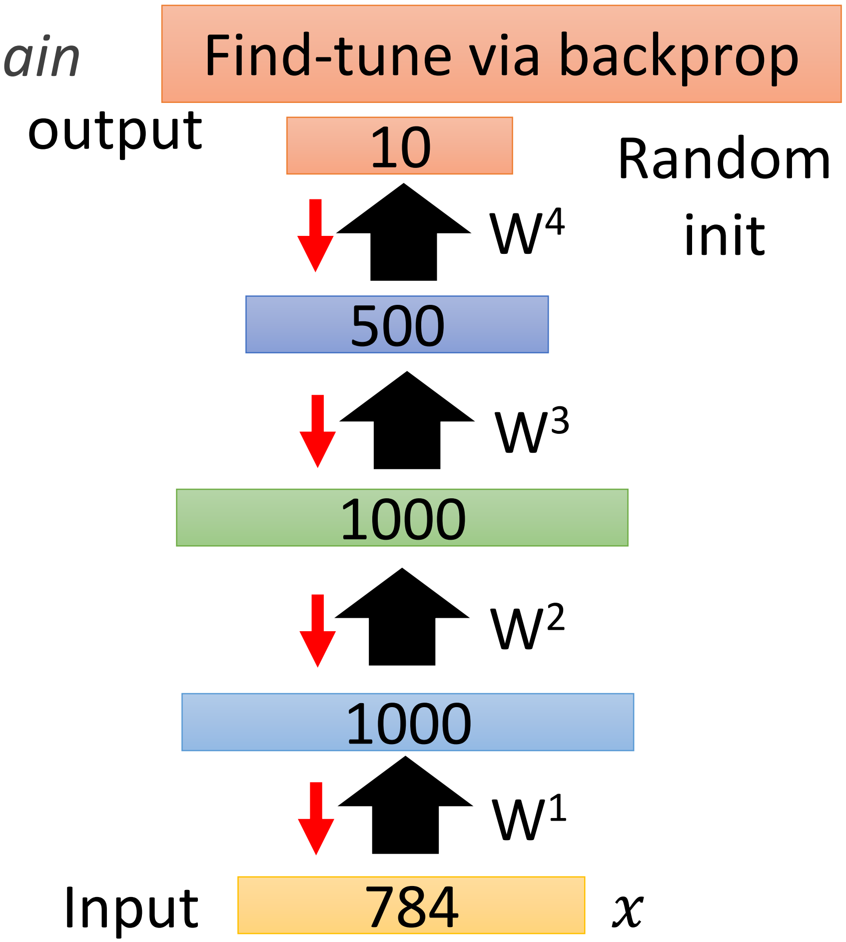
Greedy layer-wise pre-training *again*



Autoencoder for DNN Pre-Training

Greedy layer-wise pre-training *again*

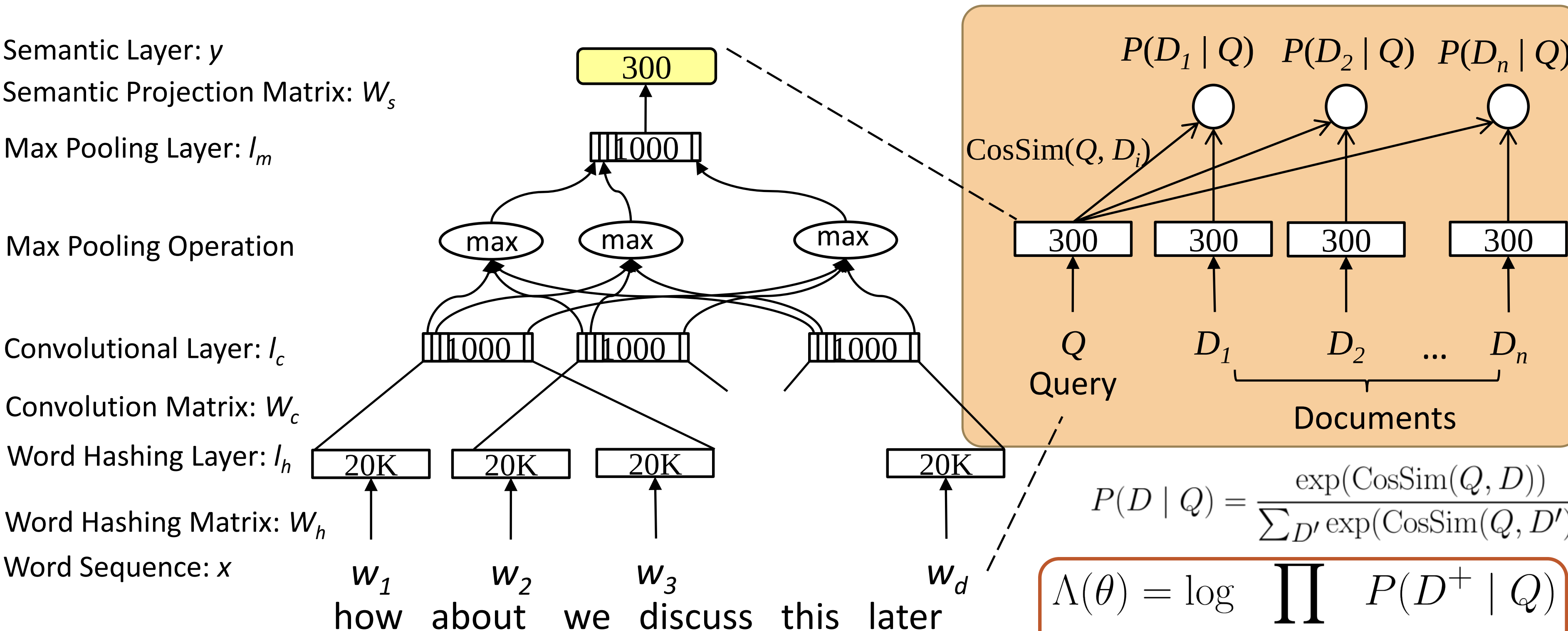




Distant Supervision

Representation Learning by Weak Labels

Convolutional Deep Structured Semantic Models (CDSSM/DSSM)



$$P(D \mid Q) = \frac{\exp(\mathrm{CosSim}(Q, D))}{\sum_{D'} \exp(\mathrm{CosSim}(Q, D'))}$$

$$\boxed{\Lambda(\theta) = \log \prod_{(Q,D^+)} P(D^+ \mid Q)}$$

maximizes the likelihood of clicked documents given queries

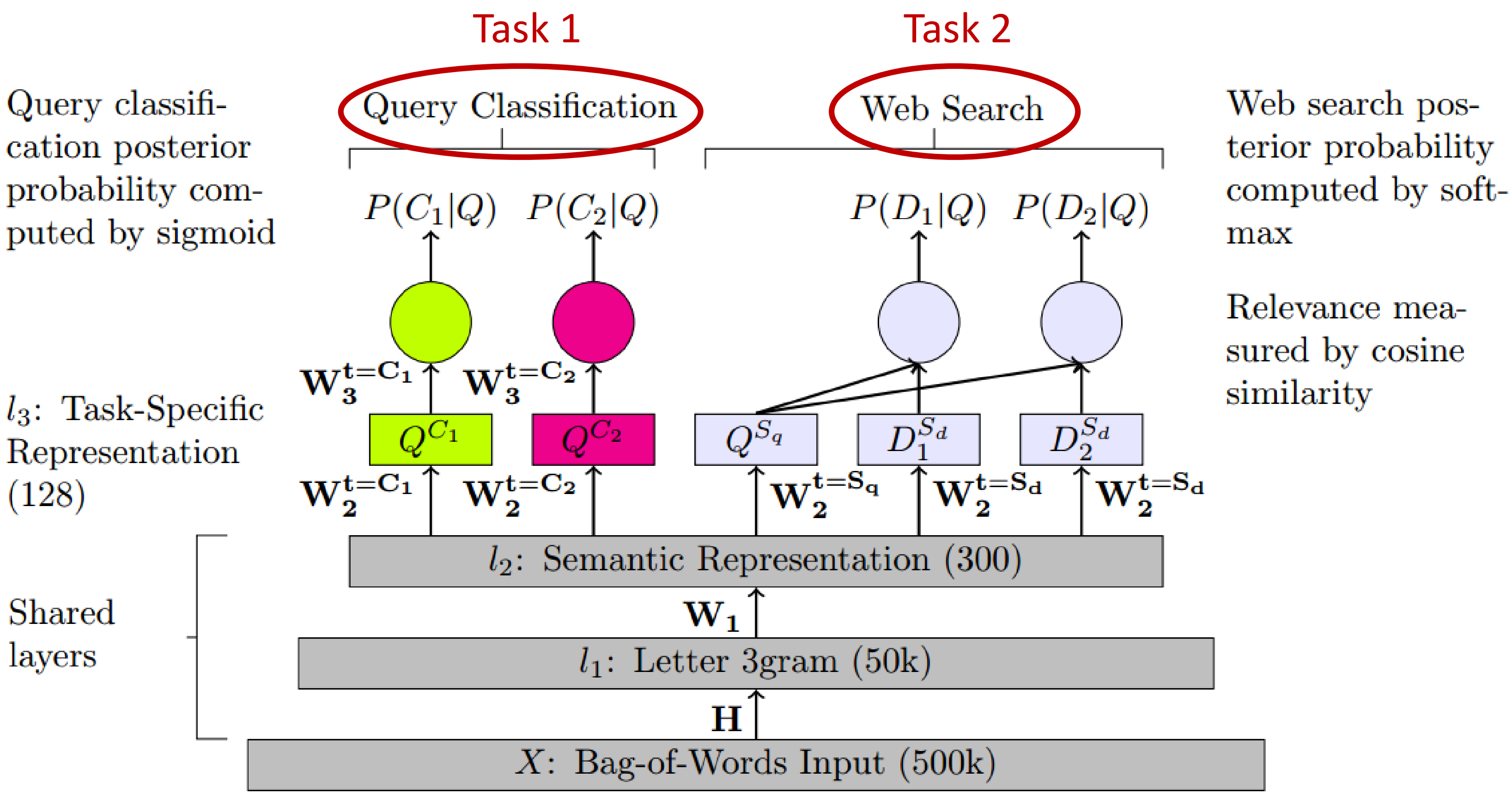
Semantically related documents are close to the query in the encoded space

Huang et al., "Learning deep structured semantic models for web search using clickthrough data," in *Proc. of CIKM*, 2013.
Shen et al., "Learning semantic representations using ´ convolutional neural networks for web search," in *Proc. of WWW*, 2014.

Multi-Tasking

Representation Learning by Different Tasks

Task-Shared Representation



The latent factors can be learned by different tasks

Generative Adversarial Network (GAN)

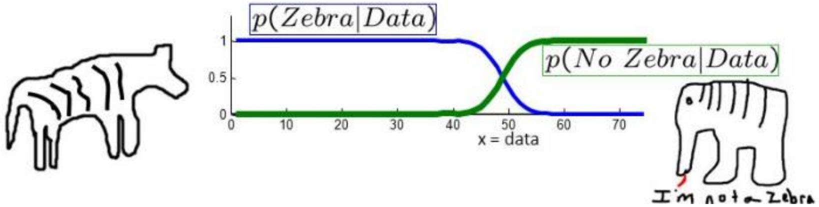
Representation Learning

"There are many interesting recent development in deep learning…The most important one, in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks). This, and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion." – **Yann LeCun**

Discriminative v.s. Generative Models

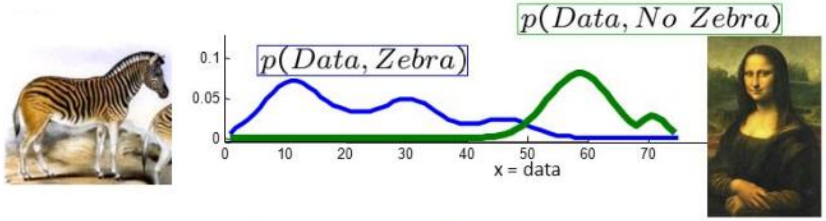
Discriminative

- learns a function that maps the input data (*x*) to some desired output class label (*y*)
  - directly learn the conditional distribution *P(y|x)*



Generative

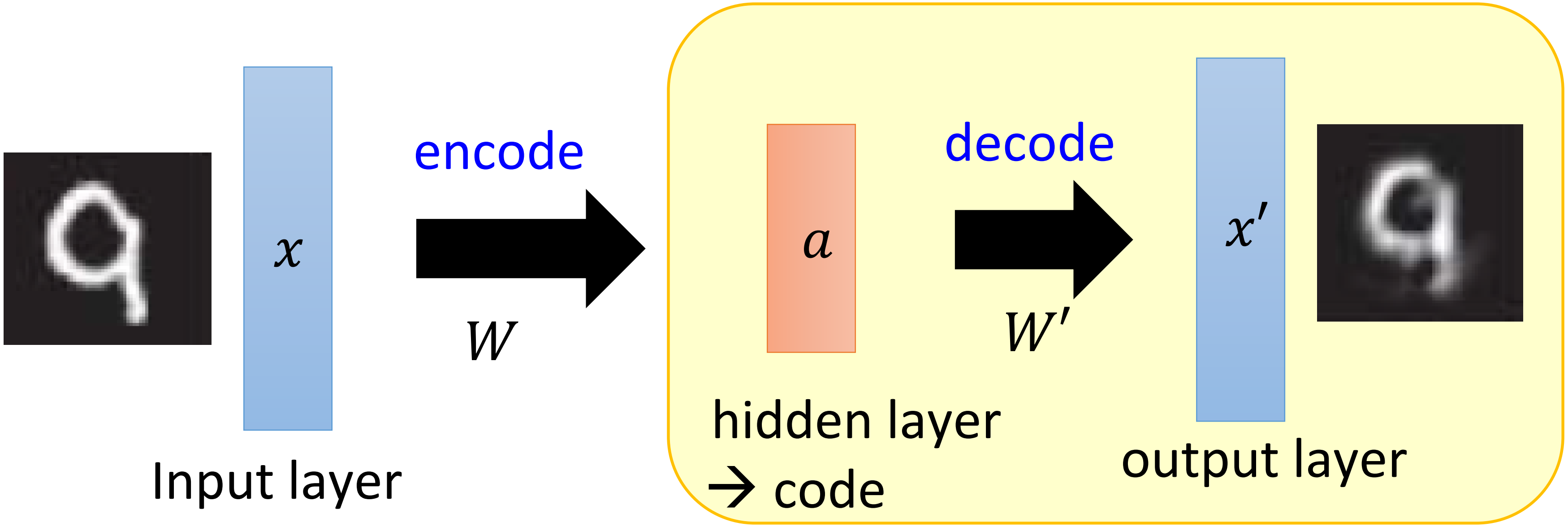
- tries to learn the joint probability of the input data and labels simultaneously, i.e. *P(x,y)*
  - can be converted to *P(y|x)* for classification via Bayes rule



Advantage: generative models have the potential to understand and explain the underlying structure of the input data even when there are no labels

Generator

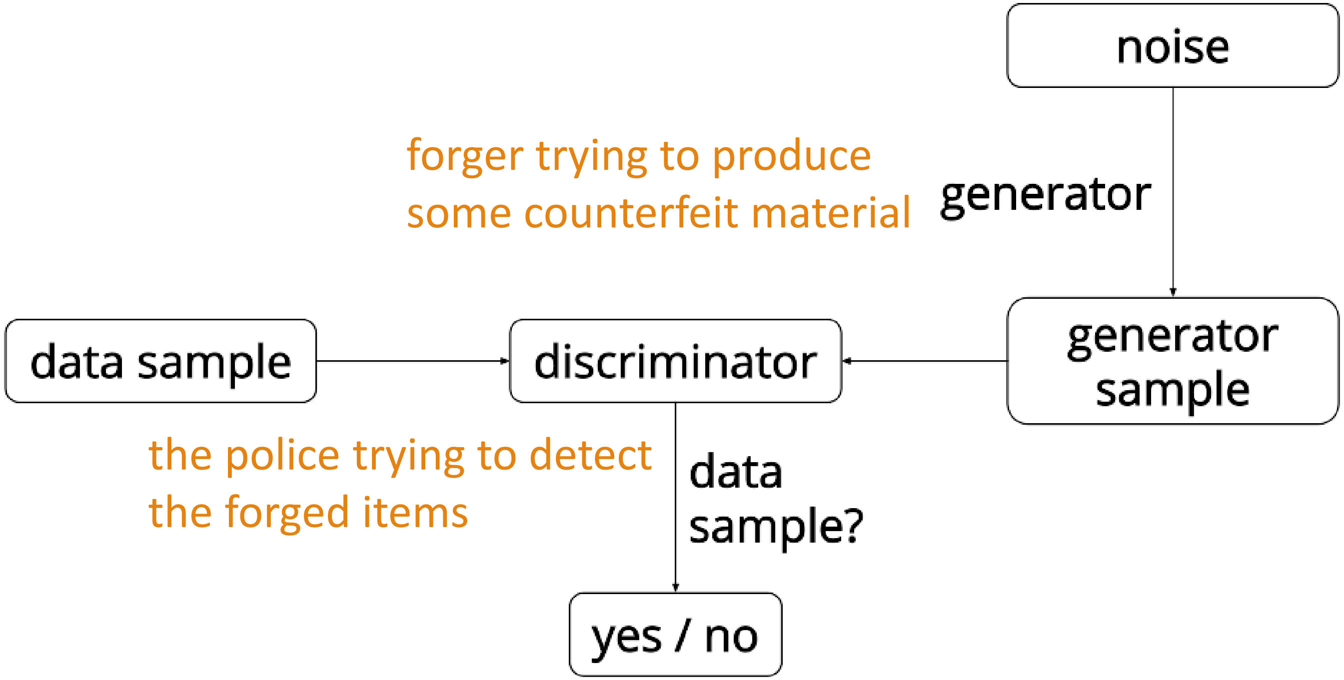
Decoder from autoencoder as generator



The generator is to generate the data from the code

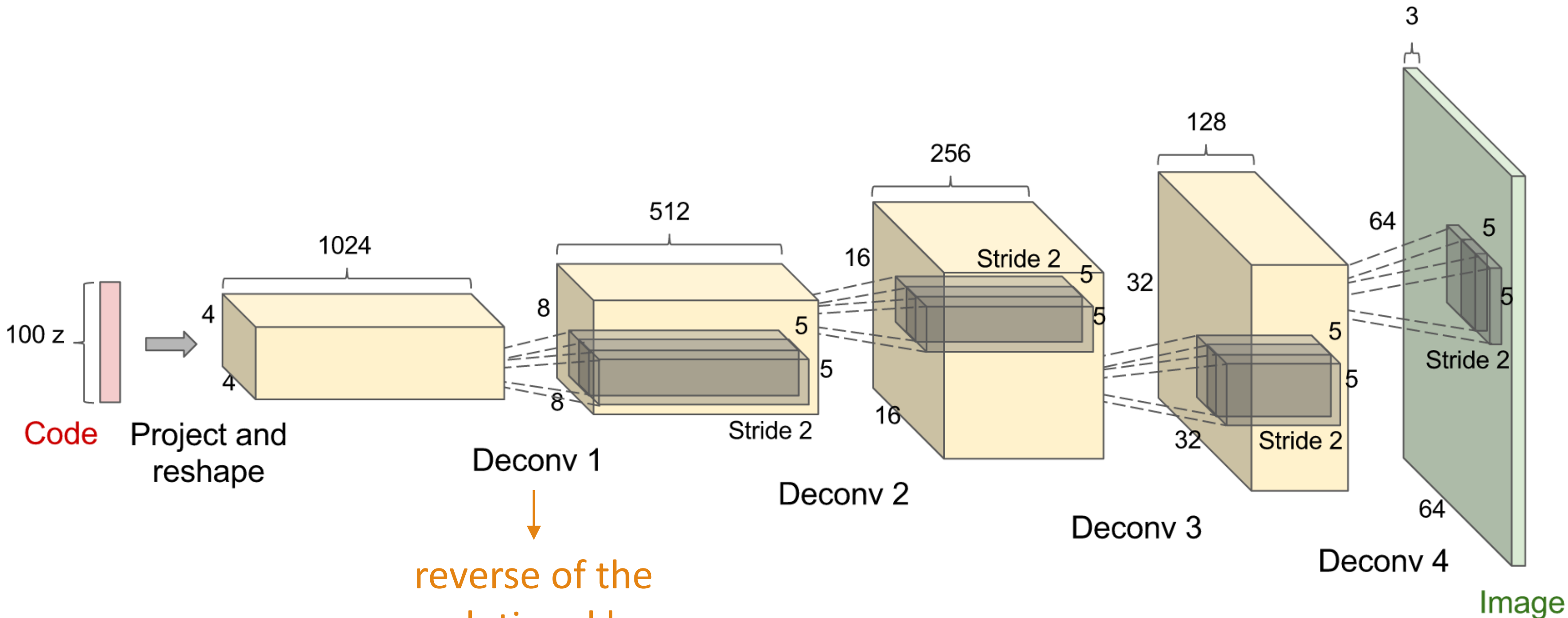
Generative Adversarial Networks (GAN)

Two competing neural networks: generator & discriminator



Training two networks jointly → the generator knows how to adapt its parameters in order to produce output data that can fool the discriminator

Deep Convolutional GAN (DCGAN)



reverse of the convolutional layers

Generated Bedrooms

Concluding Remarks

Labeling data is expensive, but we have large unlabeled data

Autoencoder

- exploits the unlabeled data to learn latent factors as representations
- learned representations can be transfer to other tasks

Generative models

- have the potential to understand and explain the underlying structure of the input data even when there are no labels
- Generative Adversarial Networks (GAN): jointly train two competing networks, **generator** and **discriminator**


Merry
Christmas