

Course ID 80250993 Dates: 9/16-12/23/2021 @ I-205
Tencent/VooV Meeting 521 4678 6257 (passwd 1205)
Broadcast and Reply: <https://meeting.tencent.com/j/p11V4drTYoa4> (passwd 1205)



Part IV

Deep Learning and Beyond

Xuegong Zhang
Dec 2, 2021



Xuegong Zhang

80250993 Machine Learning
@Tsinghua University



Chapter 17

Deep Neural Networks

Xuegong Zhang
Dec 2, 2021

Xuegong

2

Let's recall what we have learned

A "simple" example

How can we teach a kid to distinguish man and woman from a portrait picture?

For the green line:
 $w_1 x_1 + w_2 x_2 = 0$
 For each blue dot:
 $w_1 x_1 + w_2 x_2 > 0$
 For each red dot:
 $w_1 x_1 + w_2 x_2 < 0$

Toy data example

Age	Feature	Adult Weight
20	24	54
21	25	55
22	26	56
23	27	57
24	28	58
25	29	59
26	30	60
27	31	61
28	32	62
29	33	63
30	34	64
31	35	65
32	36	66
33	37	67
34	38	68
35	39	69
36	40	70
37	41	71
38	42	72
39	43	73
40	44	74
41	45	75
42	46	76
43	47	77
44	48	78
45	49	79
46	50	80
47	51	81
48	52	82
49	53	83
50	54	84
51	55	85
52	56	86
53	57	87
54	58	88
55	59	89
56	60	90
57	61	91
58	62	92
59	63	93
60	64	94
61	65	95
62	66	96
63	67	97
64	68	98
65	69	99
66	70	100
67	71	101
68	72	102
69	73	103
70	74	104
71	75	105
72	76	106
73	77	107
74	78	108
75	79	109
76	80	110
77	81	111
78	82	112
79	83	113
80	84	114
81	85	115
82	86	116
83	87	117
84	88	118
85	89	119
86	90	120
87	91	121
88	92	122
89	93	123
90	94	124
91	95	125
92	96	126
93	97	127
94	98	128
95	99	129
96	100	130
97	101	131
98	102	132
99	103	133
100	104	134

Simple Linear Regression:
 $y = w_0 + w_1 x$

Let's count the percentage

Proportion (%) of signs of CD according to age group

Age group	Sign	Proportion (%)
1-10	+	10
1-10	-	90
11-20	+	20
11-20	-	80
21-30	+	30
21-30	-	70
31-40	+	40
31-40	-	60
41-50	+	50
41-50	-	50
51-60	+	60
51-60	-	40
61-70	+	70
61-70	-	30
71-80	+	80
71-80	-	20
81-90	+	90
81-90	-	10
91-100	+	100
91-100	-	0

Linear Discriminant Analysis (FLD or LDA)

Linear Discriminant Analysis (FLD or LDA)

Unsupervised Learning

Unsupervised Learning

Learning Condensed Representations of Data

Learning Condensed Representations of Data

Function Estimation Model of the Learning Problem

Function Estimation Model of the Learning Problem

MLP (Multi-layer Perceptron 多层感知机)

MLP (Multi-layer Perceptron 多层感知机)
 — The most popular Artificial Neural Network (NN)

How do we know the posterior $P(w_i|x)?$

How do we know the posterior $P(w_i|x)?$

Restricted Boltzmann Machines (RBM)

Restricted Boltzmann Machines (RBM)

Components of an application system

Components of an application system

Basic Considerations

Basic Considerations

Three Major Types of ANN (in 1980s)

Three Major Types of ANN (in 1980s)

Structural Risk Minimization

Structural Risk Minimization

Bayesian Inference

Bayesian Inference

MLP (Multi-layer Perceptron 多层感知机)

MLP (Multi-layer Perceptron 多层感知机)
 — The most popular Artificial Neural Network (NN)

How do we know the posterior $P(w_i|x)?$

How do we know the posterior $P(w_i|x)?$

Restricted Boltzmann Machines (RBM)

Restricted Boltzmann Machines (RBM)

Components of an application system

Components of an application system

Let's recall what we have learned

A "simple" example

- How can we teach a cat to distinguish man and woman from a portrait picture?

Toy data example

Let's count the percentage

x

features

y

property

SYSTEM

training

prediction

Learning Machine

$f(x, \alpha)$

Basic Strategy: To build a single machine to learn the task.

The MSE criterion

Basic Considerations

- Building a strong classifier with high accuracy and generalization is hard
- That building some weak classifiers can be easier
- Just need to be better than chance
- E.g. can combine the top not into an image as an "expert"
- Can we combine several weak classifiers to build a better classifier?
- Three humble classifiers bootstrapping will make a difference

Learning Condensed Representations of Data

Three Major Types of ANN (in 1980s)

- Feedforward NN
- Multi-Layer Perceptron
- Feedback NN
- Hopfield NN
- Competitive Learning NN
- Self-organizing map

Restricted Boltzmann Machines (RBM)

"Restricted" connections:

- Restricted to connections between visible and hidden nodes
- Bipartite graph
- Visible nodes $v = \{v_1, v_2, \dots, v_n\}$; Hidden nodes $h = \{h_1, h_2, \dots, h_m\}$

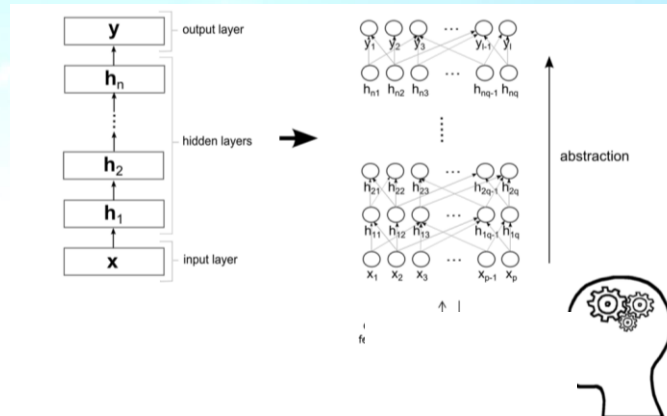
components of an application system

Raw feature → Feature selection → Feature transform → Classifier → Action

Why Deep Learning?

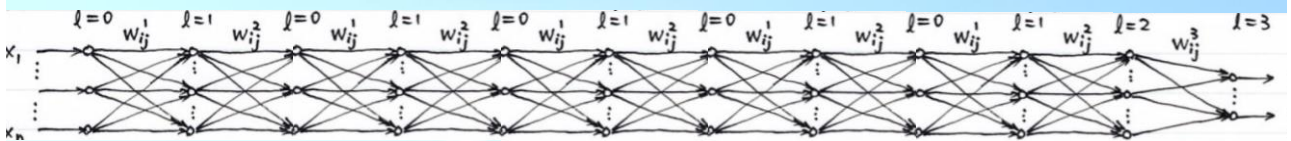


- The world is complicated
 - Human/animal brains do multi-level abstraction to perceive the world
 - Single “shallow” machines may not capture the underlying rules



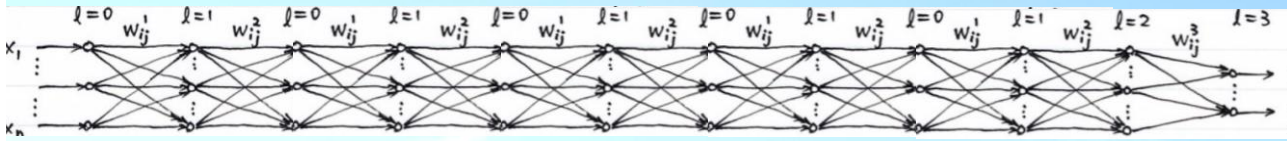
<http://www.simonaullo.com/wordpress/wp-content/uploads/2013/07/deep-learning-1024x768.png>

So, let's go deeper!





Problems with “naïve” Deep Neural Networks



- Perform poorly on visual object recognition
 - Too many layers \rightarrow too many weights to learn
 - Need many labeled data
 - Slow learning
 - Local optima
 - Gradient diffusion
 - ...
 - Spatial/context/relation information not used



17.1 Deep AutoEncoder (DAE)



17.1.1 AutoEncoders

Xuyang Zhang

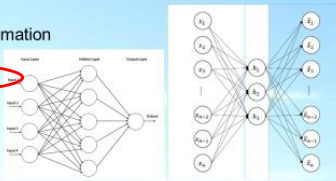
9

Autoencoders (AE)

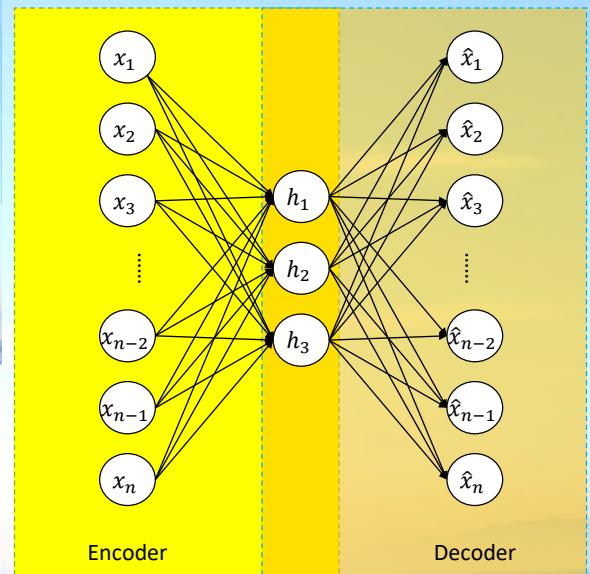


What else can MLP do?

- Other applications
 - Nonlinear regression / function estimation
 - Time series prediction
 - Data compression / representation



$$\hat{x} = f(h(x)) \triangleq x$$



Q: What information the hidden nodes of AE will capture if all activation functions are linear?

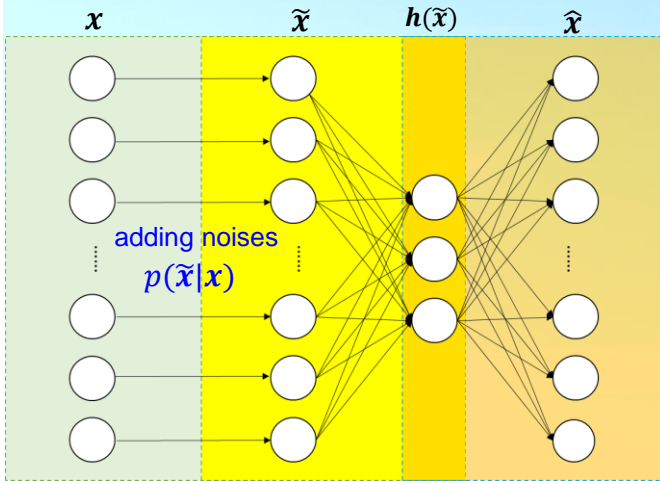


Xuyang Zhang

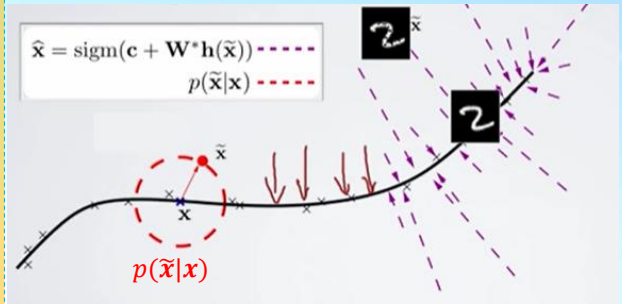
10



Denoising Autoencoders (DAE)



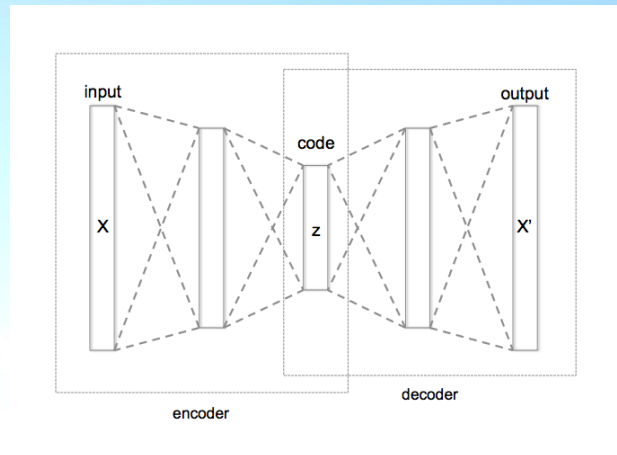
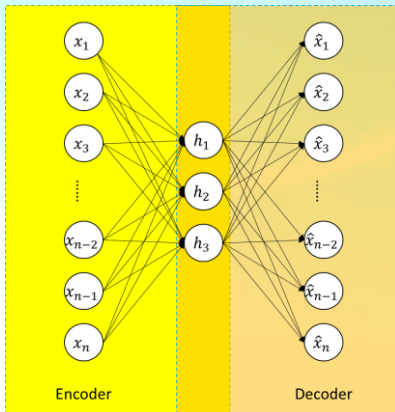
$$\hat{x} = f(h(\tilde{x})) \triangleq x$$



Xu Yang Zhang

11

Autoencoder → Deep Autoencoder

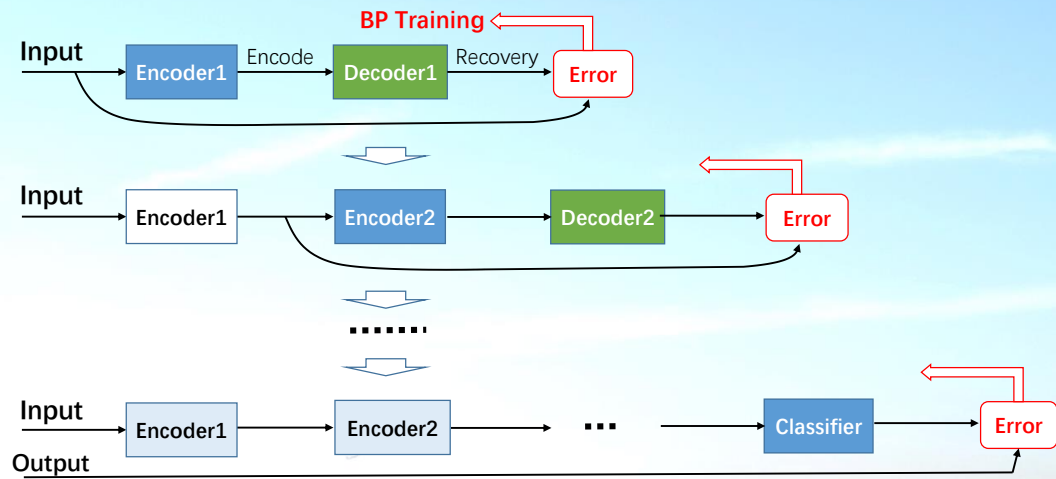


Xu Yang Zhang

12



Cascade Auto-Encoder: Unsupervised pre-training for classification



17.1.2 Deep AutoEncoders (DAE)



Restricted Boltzmann Machines (RBM)



COGNITIVE SCIENCE 9, 147-169 (1985)

A Learning Algorithm for Boltzmann Machines*

DAVID H. ACKLEY
GEOFFREY E. HINTON
*Computer Science Department
Carnegie-Mellon University*
TERRENCE J. SEJNOWSKI
*Biophysics Department
The Johns Hopkins University*

The computational power of massively parallel networks of simple processing elements resides in the communication bandwidth provided by the hardware connections between elements. These connections can allow a significant

red SHG
x in Fig.
we find
he noise
I closely
it power
mission
gle with
ms from
Rs (see
detuning
velength
he SHG
For ex-
vertical
d a small
xcitation
incident
gnificant
gain po-

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.

Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data. A simple and widely used method is principal components analysis (PCA), which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. We describe a nonlinear generalization of PCA that uses an adaptive, multilayer "encoder" network

28 JULY 2006 VOL 313 SCIENCE www.sciencemag.org



Deep Autoencoder



Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal

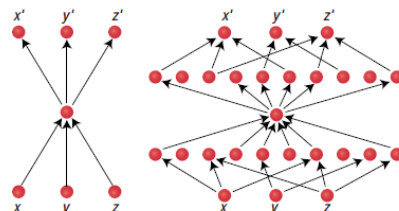
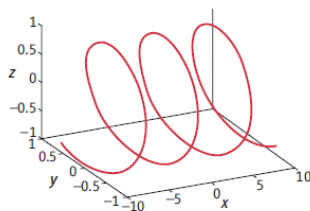
Hinton & Salakhutdinov, *Science*, 2006

New Life for Neural Networks

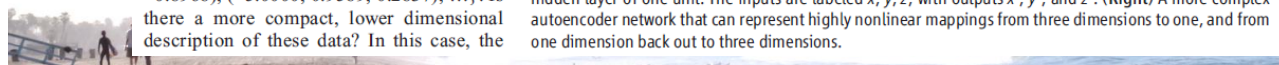
Garrison W. Cottrell

As many researchers have found, the data they have to deal with are often high-dimensional—that is, expressed by many variables—but may contain a great deal of latent structure. Discovering that structure, however, is nontrivial. To illustrate the point, consider a case in the relatively low dimension of three. Suppose you are handed a large number of three-dimensional points in random order (where each point is denoted by its coordinates along the x , y , and z axes): $\{(-7.4000, -0.8987, 0.4385), (3.6000, -0.4425, -0.8968), (-5.0000, 0.9589, 0.2837), \dots\}$. Is there a more compact, lower dimensional description of these data? In this case, the

With the help of neural networks, data sets with many dimensions can be analyzed to find lower dimensional structures within them.



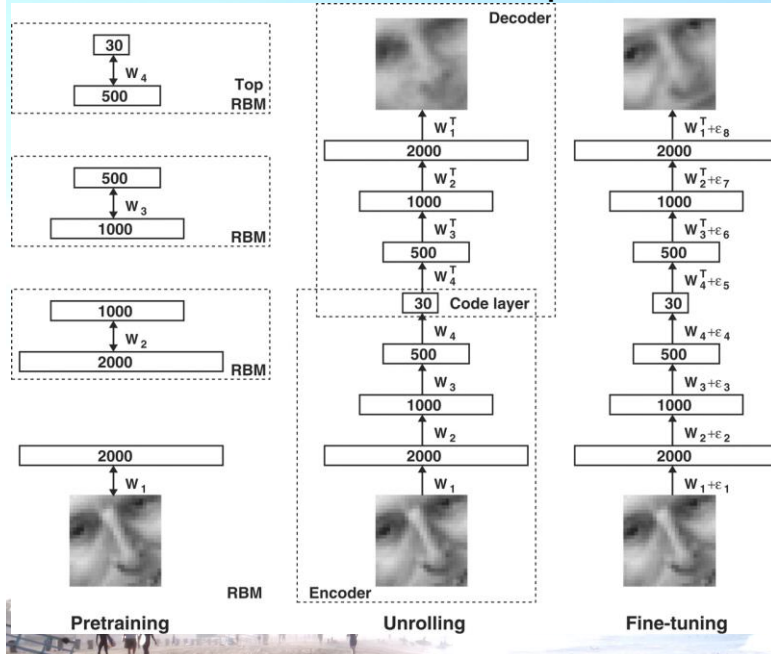
Searching for structure. (Left) Three-dimensional data that are inherently one-dimensional. (Middle) A simple "autoencoder" network that is designed to compress three dimensions to one, through the narrow hidden layer of one unit. The inputs are labeled x , y , z , with outputs x' , y' , and z' . (Right) A more complex autoencoder network that can represent highly nonlinear mappings from three dimensions to one, and from one dimension back out to three dimensions.



Deep Autoencoder



Hinton & Salakhutdinov, *Science*, 2006



- **Encoding:**
 - a stack of RBMs with cascade pre-training
- **Decoding:**
 - “unrolling” the encoder RBMs
- **Fine-tuning:**
 - BP training for the whole encoder-decoder network

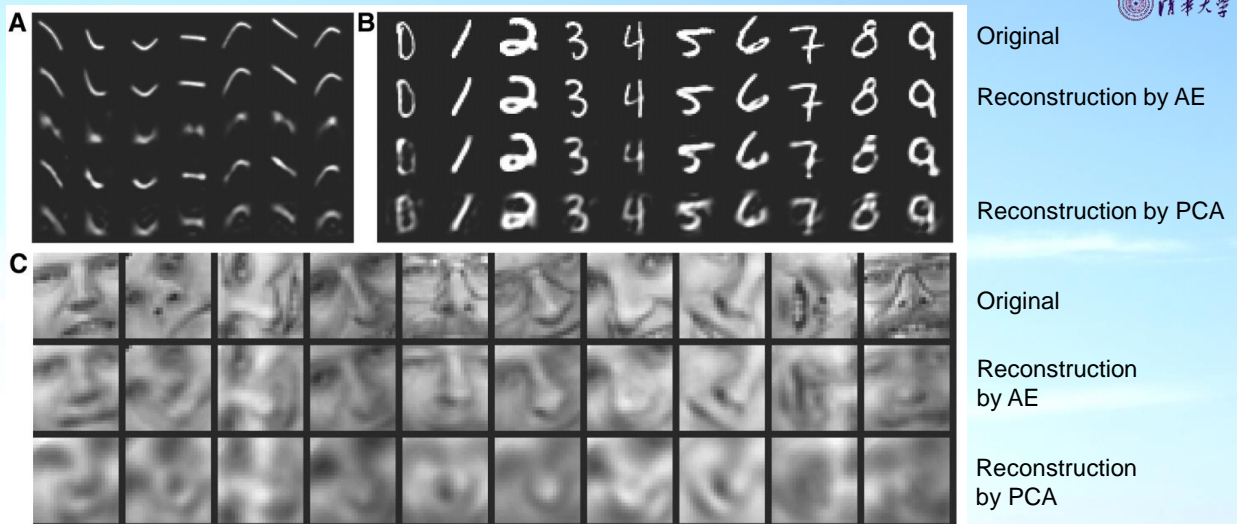


Fig. 2. (A) Top to bottom: Random samples of curves from the test data set; reconstructions produced by the six-dimensional deep autoencoder; reconstructions by “logistic PCA” using six components; reconstructions by logistic PCA and standard PCA using 18 components. The average squared error per image for the last four rows is 1.44, 7.64, 2.45, 5.90. (B) Top to bottom: A random test image from each class; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional logistic PCA and standard PCA. The average squared errors for the last three rows are 3.00, 8.01, and 13.87. (C) Top to bottom: Random samples from the test data set; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional PCA. The average squared errors are 126 and 135.

Hinton & Salakhutdinov, *Science*, 2006

Deep Autoencoder



Hinton & Salakhutdinov, *Science*, 2006

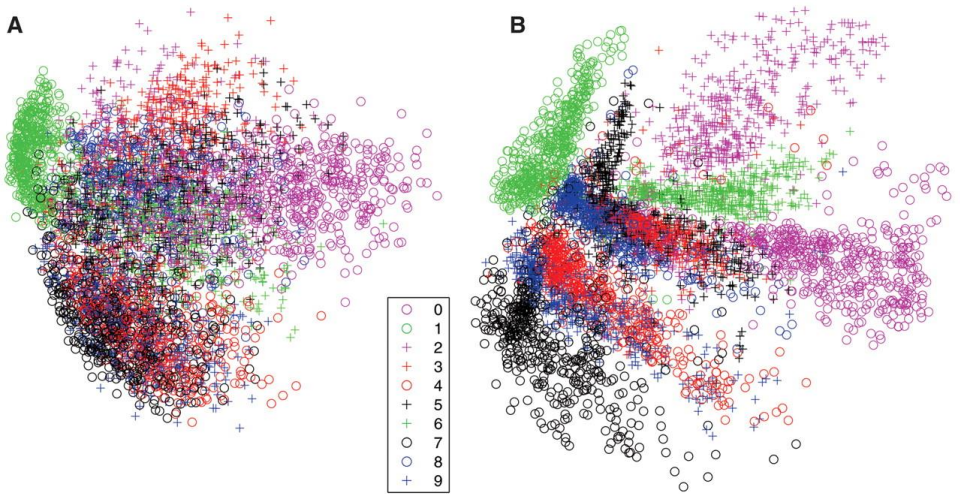


Fig. 3. (A) The two-dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. (B) The two-dimensional codes found by a 784-1000-500-250-2 autoencoder.

19

Deep Autoencoder



Hinton & Salakhutdinov, *Science*, 2006

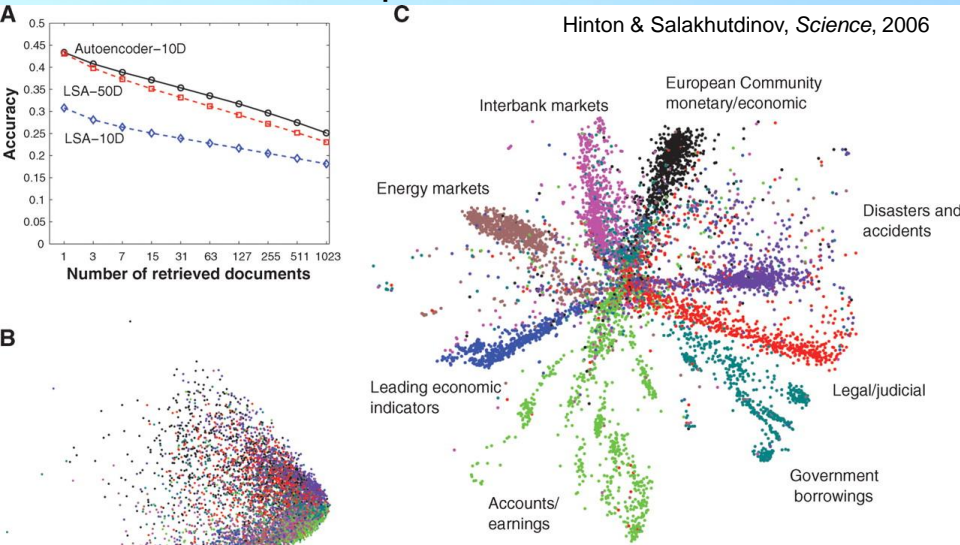


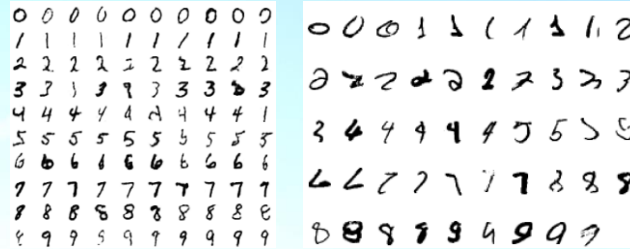
Fig. 4. (A) The fraction of retrieved documents in the same class as the query when a query document from the test set is used to retrieve other test set documents, averaged over all 402,207 possible queries. (B) The codes produced by two-dimensional LSA. (C) The codes produced by a 2000-500-250-125-2 autoencoder.

20



Application on images

- On MNIST



- | | |
|--|-------|
| • Generative model based on RBM's | 1.25% |
| • Support Vector Machine (Decoste et. al.) | 1.4% |
| • Backprop with 1000 hidden (Platt) | ~1.6% |
| • Backprop with 500 --> 300 hidden | ~1.6% |
| • K-Nearest Neighbor | ~3.3% |

Geoffrey Hinton, UCL Tutorial on Deep Belief Nets, 2009

21



10-second break



Xuegong Zhang

22

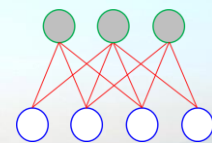
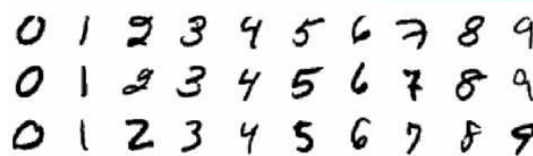
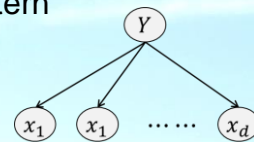
17.2 Deep Belief Networks

Xuegong Zhang

23

Recall: What are “Patterns”?

- Simple patterns
 - Single feature: distribution \rightarrow pattern
 - Multiple independent features: distributions \rightarrow pattern
- Complex patterns
 - Multiple related features:
relations among features \rightarrow pattern



Xuegong Zhang

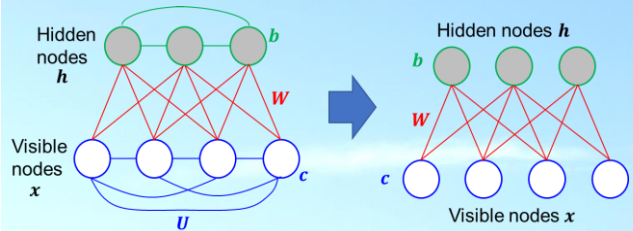
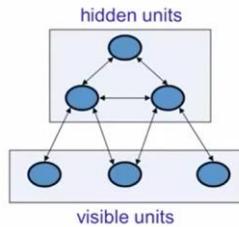
24

What are “Patterns”?



A different computational role for Hopfield nets

- Instead of using the net to store memories, use it to construct interpretations of sensory input.
 - The input is represented by the visible units.
 - The interpretation is represented by the states of the hidden units.
 - The badness of the interpretation is represented by the energy.



- The world is complicated
 - Human/animal brains do multi-level abstraction to perceive the world
 - Single “shallow” machines may not capture the underlying rules
- **Can we use multiple layers of hidden units to capture complex objects?**



Xuegong Zhang

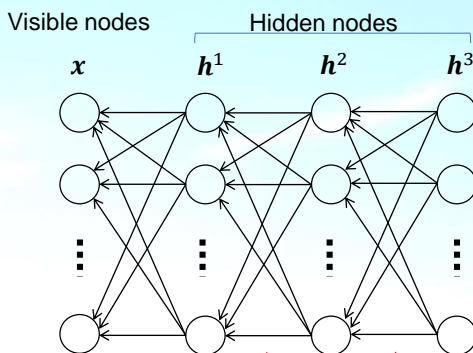
25

Sigmoid Belief Network



(Raiford M. Neal, 1992)

- Neural Network + Bayesian Network



Conditional Probabilities
(not NN connections)

$$P(x_i = 1 | \mathbf{h}^1) = \text{sigm} \left(b_i^0 + \sum_j w_{i,j}^1 h_j^1 \right)$$

$$P(h_i^k = 1 | \mathbf{h}^{k+1}) = \text{sigm} \left(b_i^k + \sum_j w_{i,j}^{k+1} h_j^{k+1} \right),$$

$$k = 1, \dots, l-1$$

$$P(\mathbf{x}, \mathbf{h}^1, \dots, \mathbf{h}^l) = P(\mathbf{h}^l) \left(\prod_{k=1}^{l-1} P(\mathbf{h}^k | \mathbf{h}^{k+1}) \right) P(\mathbf{x} | \mathbf{h}^1)$$

Powerful for modeling complicated data, but **intractable to compute the joint probability of the last layer.**



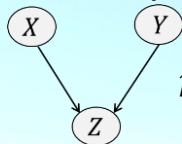
Xuegong Zhang

26

“Explaining Away” in sigmoid belief networks



- Conditional Dependency



$$p(x, y, z) = p(x)p(y)p(z|x, y)$$

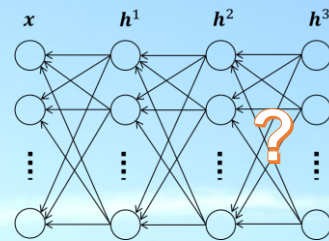
- When Z is unknown, X and Y are independent.

$$p(x, y) = \sum_{z' \in Z} p(x, y, z') = \sum_{z' \in Z} p(x)p(y)p(z'|x, y) = p(x)p(y) \sum_{z' \in Z} p(z'|x, y) = p(x)p(y)$$

- When Z is known, X and Y become dependent.

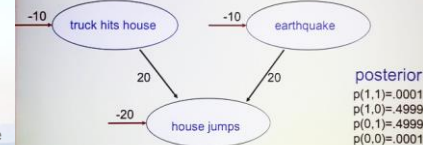
$$p(x, y|z) = p(x)p(y)p(z|x, y)/p(z)$$

→ Intractable to compute the joint probability of the last layer.



Explaining away (Judea Pearl)

- Even if two hidden causes are independent, they can become dependent when we observe an effect that they can both influence.
- If we learn that there was an earthquake it reduces the probability that the house jumped because of a truck.



posterior
 $p(1,1)=0.001$
 $p(1,0)=0.4999$
 $p(0,1)=0.4999$
 $p(0,0)=0.001$

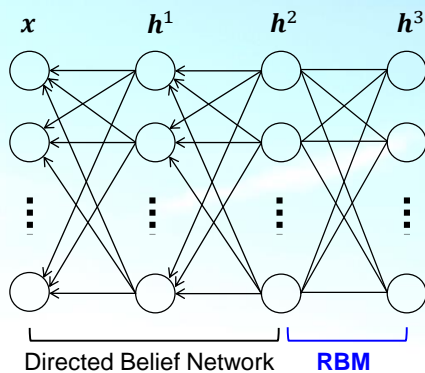
Xuegong Zhang

from Hinton's slide

Deep Belief Network (DBN)



(G.E. Hinton et al, 2006)



$$P(x_i = 1 | \mathbf{h}^1) = \text{sigm} \left(b_i^0 + \sum_j W_{i,j}^1 h_j^1 \right)$$

$$P(h_i^k = 1 | \mathbf{h}^{k+1}) = \text{sigm} \left(b_i^k + \sum_j W_{i,j}^{k+1} h_j^{k+1} \right),$$

$$k = 1, \dots, l-2$$

$$P(\mathbf{h}^{l-1}, \mathbf{h}^l) \propto \exp \left(\mathbf{c}^T \mathbf{h}^{l-1} + \mathbf{b}^T \mathbf{h}^l + \mathbf{h}^{l-1T} \mathbf{W} \mathbf{h}^l \right)$$

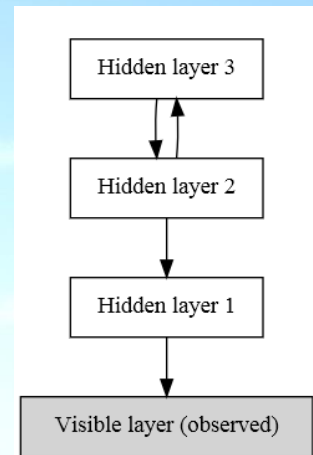
$$P(\mathbf{x}, \mathbf{h}^1, \dots, \mathbf{h}^l) = P(\mathbf{h}^{l-1}, \mathbf{h}^l) \left(\prod_{k=1}^{l-1} P(\mathbf{h}^k | \mathbf{h}^{k+1}) \right) P(\mathbf{x} | \mathbf{h}^1)$$

Xuegong Zhang

28

Deep Belief Network (DBN)

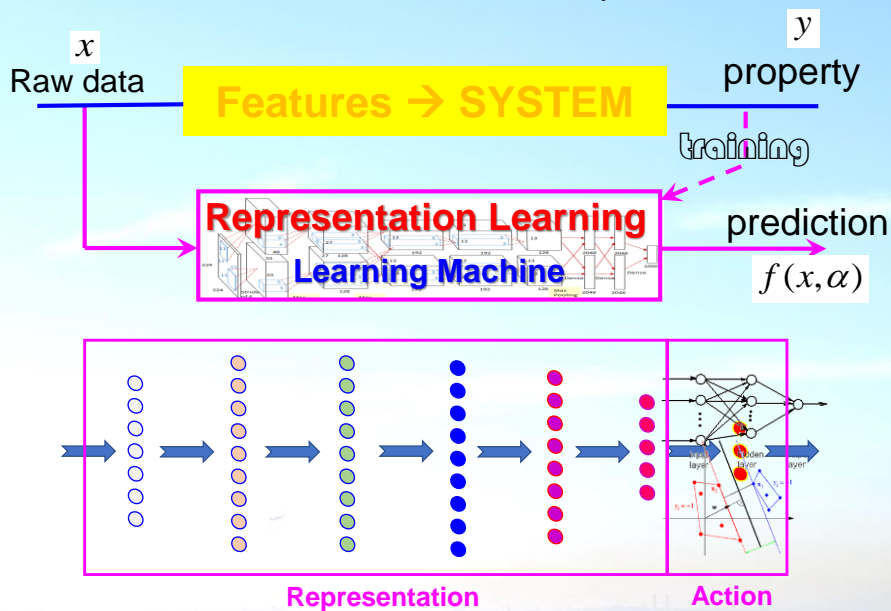
- Let X be a matrix of inputs, regarded as a set of feature vectors.
 1. Train a RBM on X to obtain its weight matrix W . Use this as the weight matrix between the lower two layers of the network.
 2. Transform X by the RBM to produce new data X' , either by sampling or by computing the mean activation of the hidden units.
 3. Repeat this procedure with $X \leftarrow X'$ for the next pair of layers, until the top two layers of the network are reached.



What can DBN be used for?



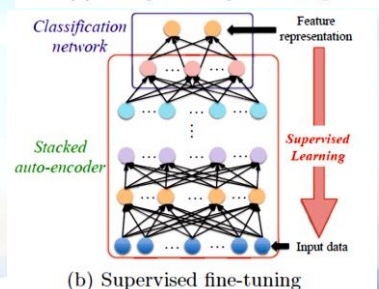
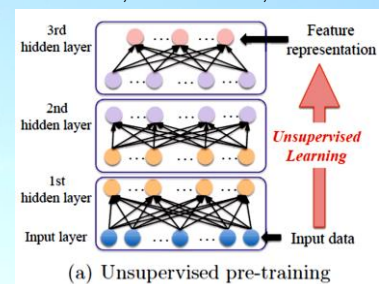
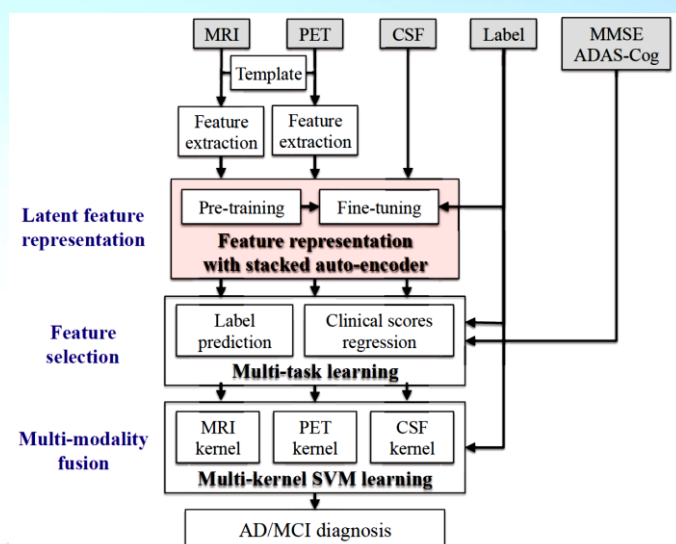
Decision based on Learned Representations



Xueqiang Zhang

31

Example: DL-based feature representation for AD/MCI classification

Suk & Shen, *MICCAI* 2013, LNCS 8150: 583-590

32



Discussion

- Deep neural network is a general name for many methods. There is no particular method named “deep neural network”.
- Major families of deep learning methods
 - Probabilistic models / Generative models
 - DAE, DBN, VAE, ...
 - Unsupervised + supervised
 - Deterministic models
 - CNN, RNN, LSTM, ...



Xuegong Zhang

33



Homework

- No homework for this chapter.
- It's really time to start working on your course project now, if you haven't started it yet.
- **Final exam: Dec.31 Friday, 9:00-11:00am, @ I-205**
- **Open-book exam**



Xuegong Zhang

34

See you next week
for
Convolutional Neural Networks



Xuegong Zhang

35