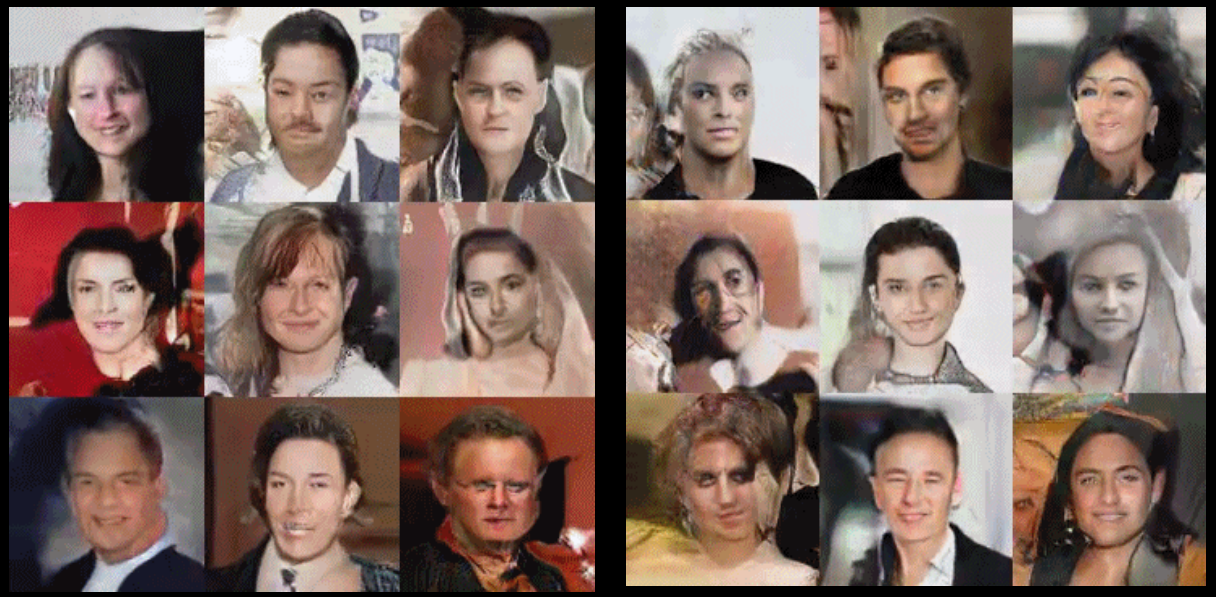
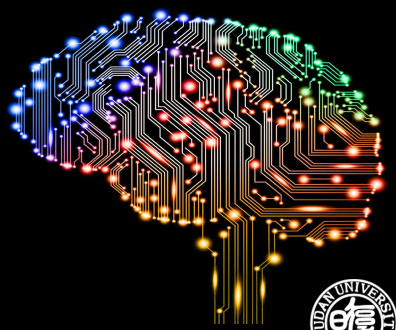
数据驱动的人工智能（6a）基于能量的神经网络

Data Driven Artificial Intelligence

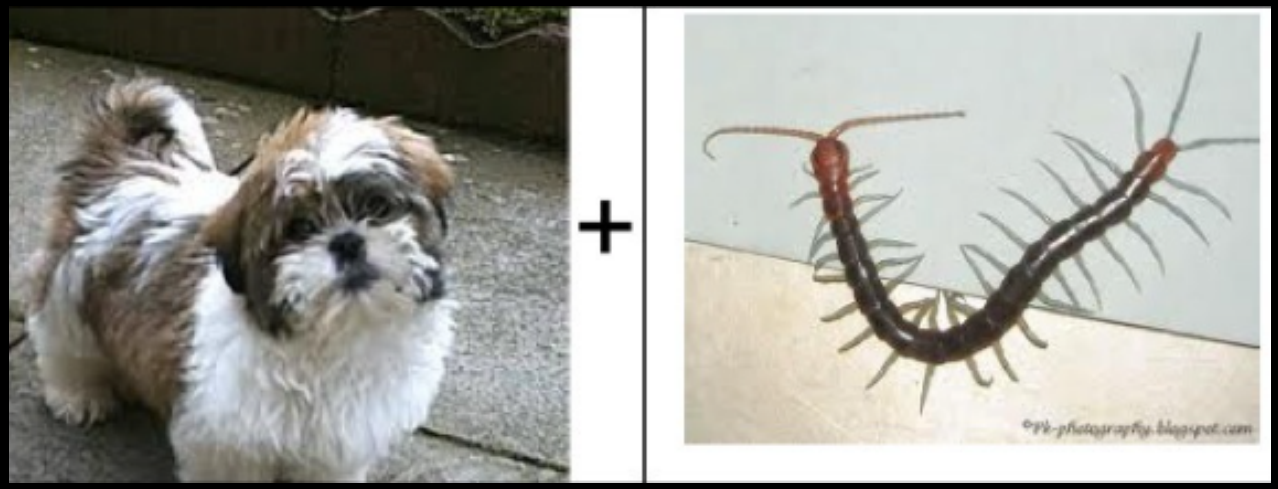
邬学宁 SAP硅谷创新中心

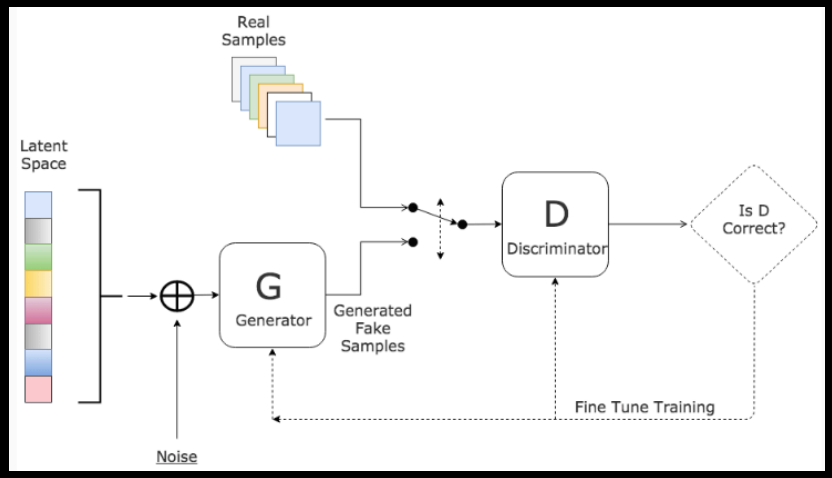
2017 / 03



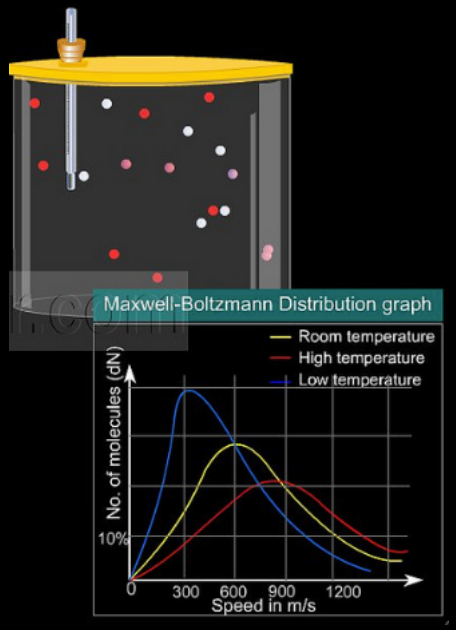


Generative Adversarial Networks





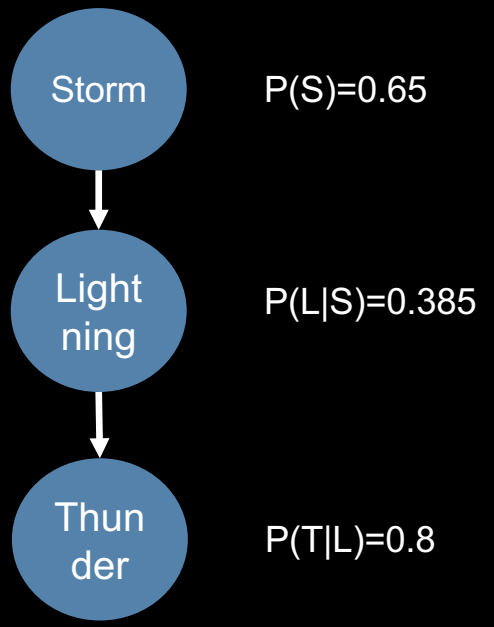
日程: EBN



- Bayesian Belief Network
- Auto-encoder / PCA
- Hopfield Network
- Boltzmann Machine
- Restricted Boltzmann Machine
- Deep Boltzmann Machine
- Deep Belief Network
- Sparse Coding
- Game Theory & Generative Adversarial Network

Bayesian Networks (Belief Net)

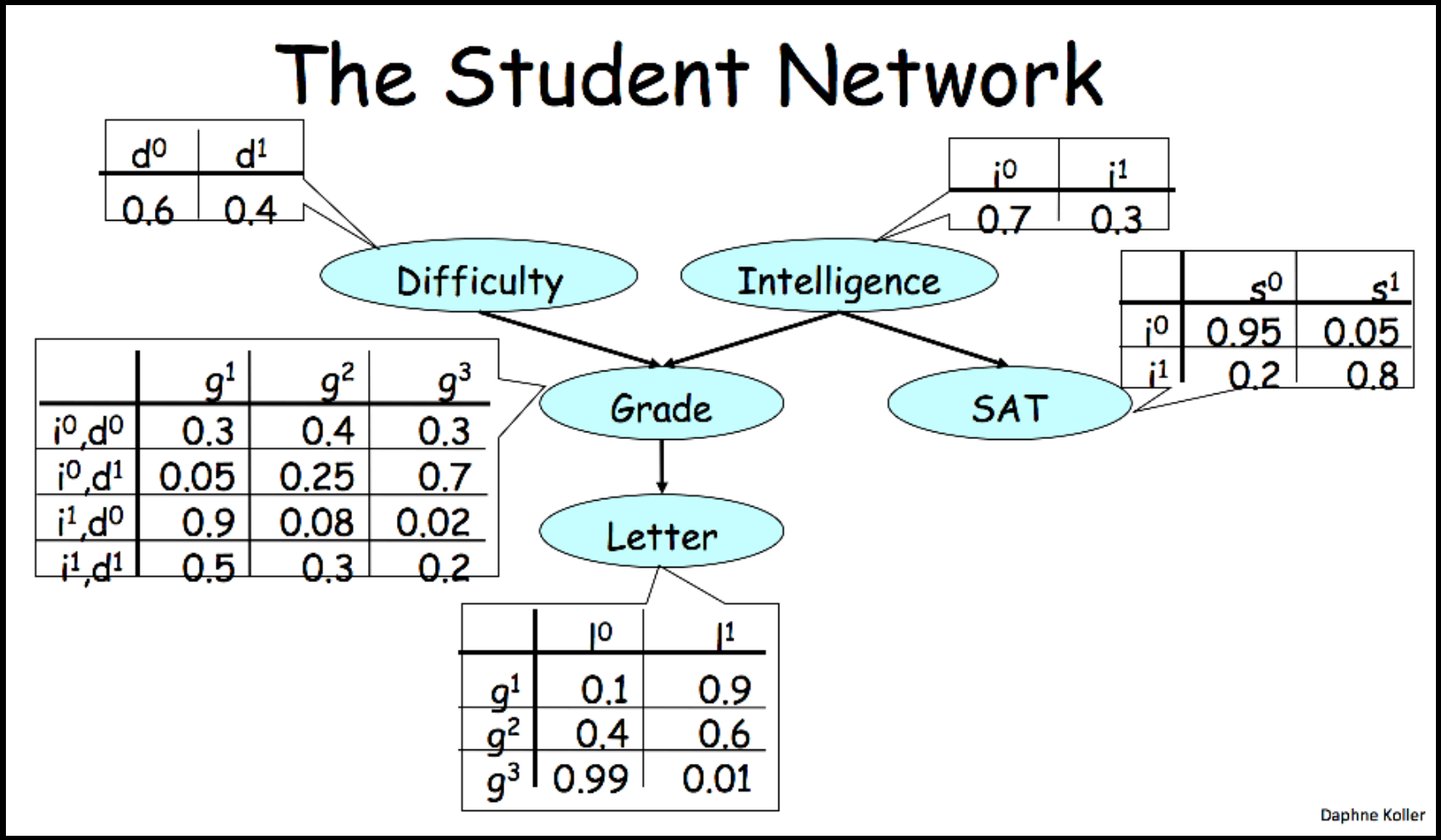
Storm	Lightning	%	Thunder	
T	T	0.25	T	0.20
			F	0.05
T	F	0.40	T	0.04
			F	0.36
F	T	0.05	T	0.04
			F	0.01
F	F	0.30	T	0.03
			F	0.27



A belief net is a Directed Acyclic Graph (DAG) composed of stochastic variables. We get to observe some of the variables and we would like 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 training data.

Recap: Bayesian Networks (Belief Net)



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

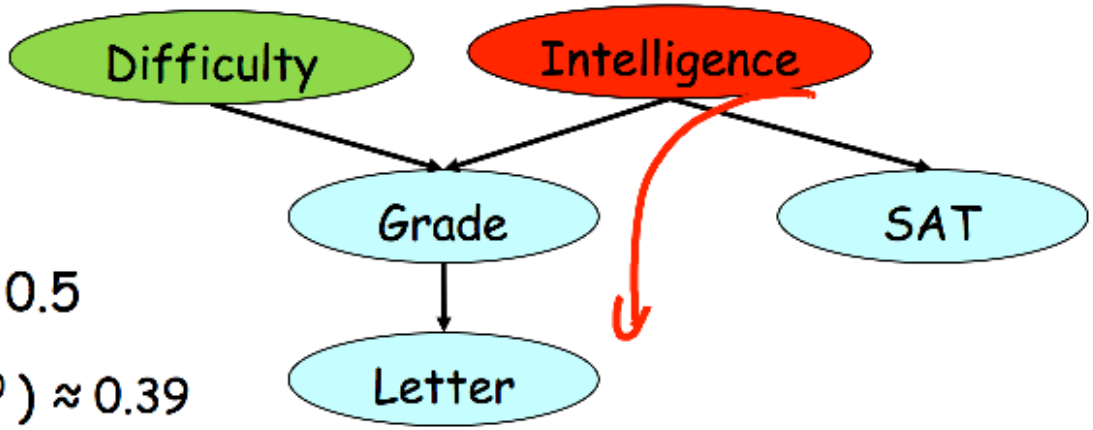
	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Recap: Bayesian Networks (Belief Net)

Causal Reasoning



$P(l^1) \approx 0.5$

$P(l^1 \mid i^0) \approx 0.39$

$P(l^1 \mid i^0, d^0) \approx 0.51$

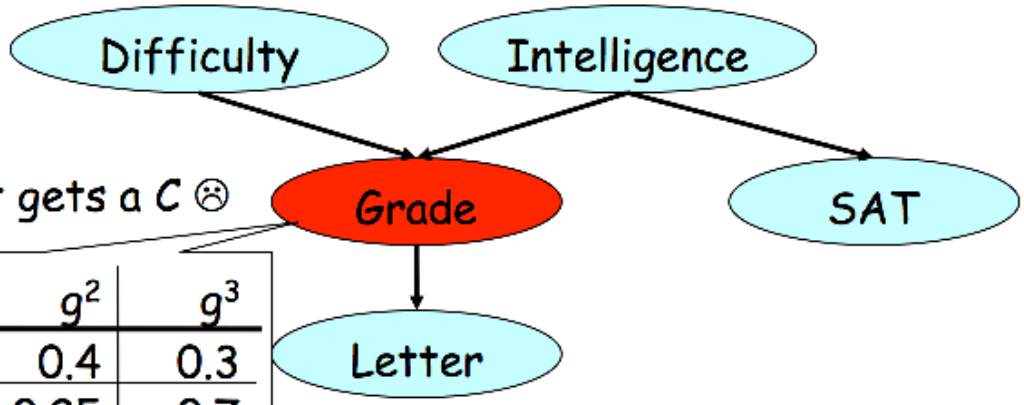
Daphne Koller

Recap: Bayesian Networks (Belief Net)

Evidential Reasoning

$P(d^1) = 0.4$ $\quad$ $P(i^1) = 0.3$

$P(d^1 \mid g^3) \approx 0.63$ $\quad$ $P(i^1 \mid g^3) \approx 0.08$

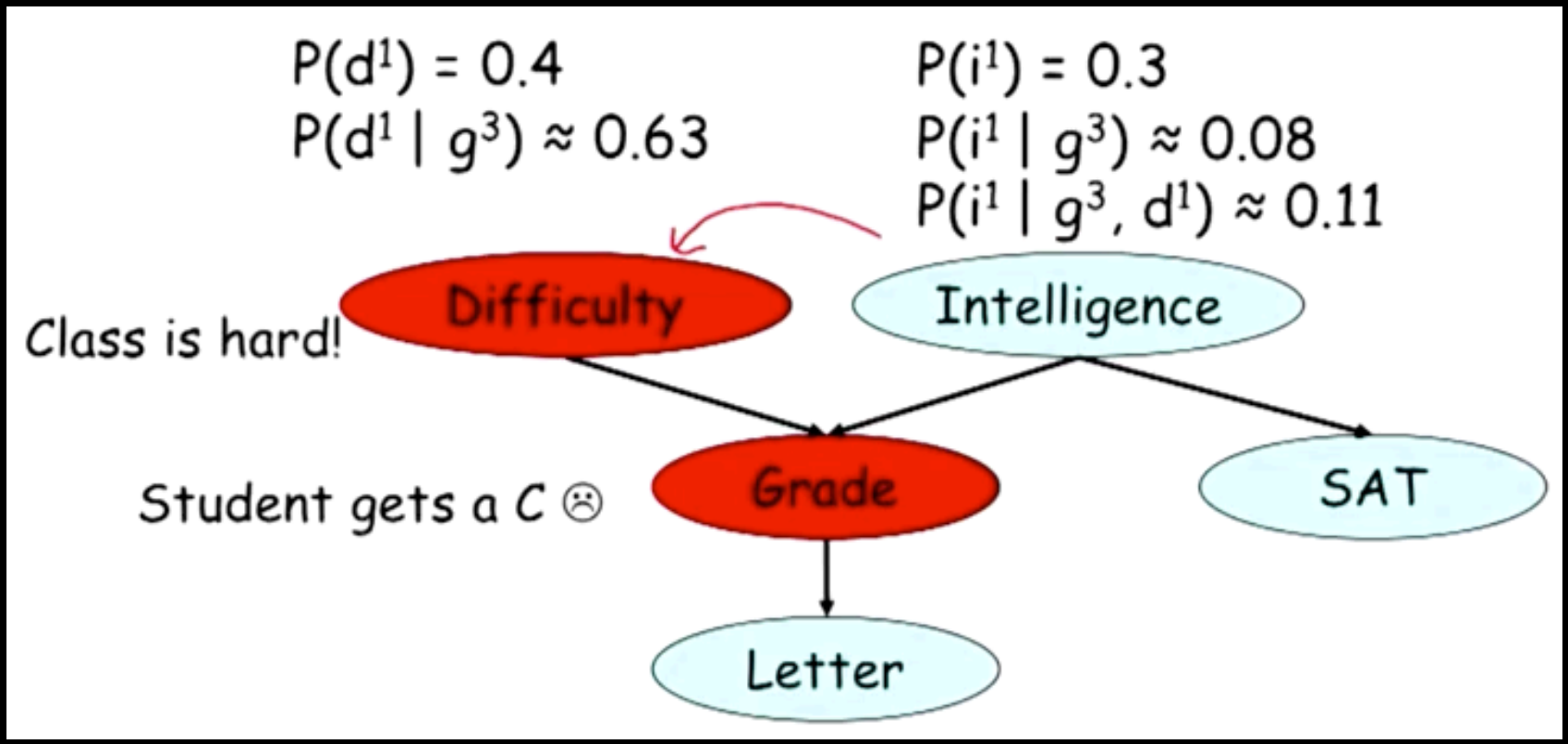


Student gets a C ☹

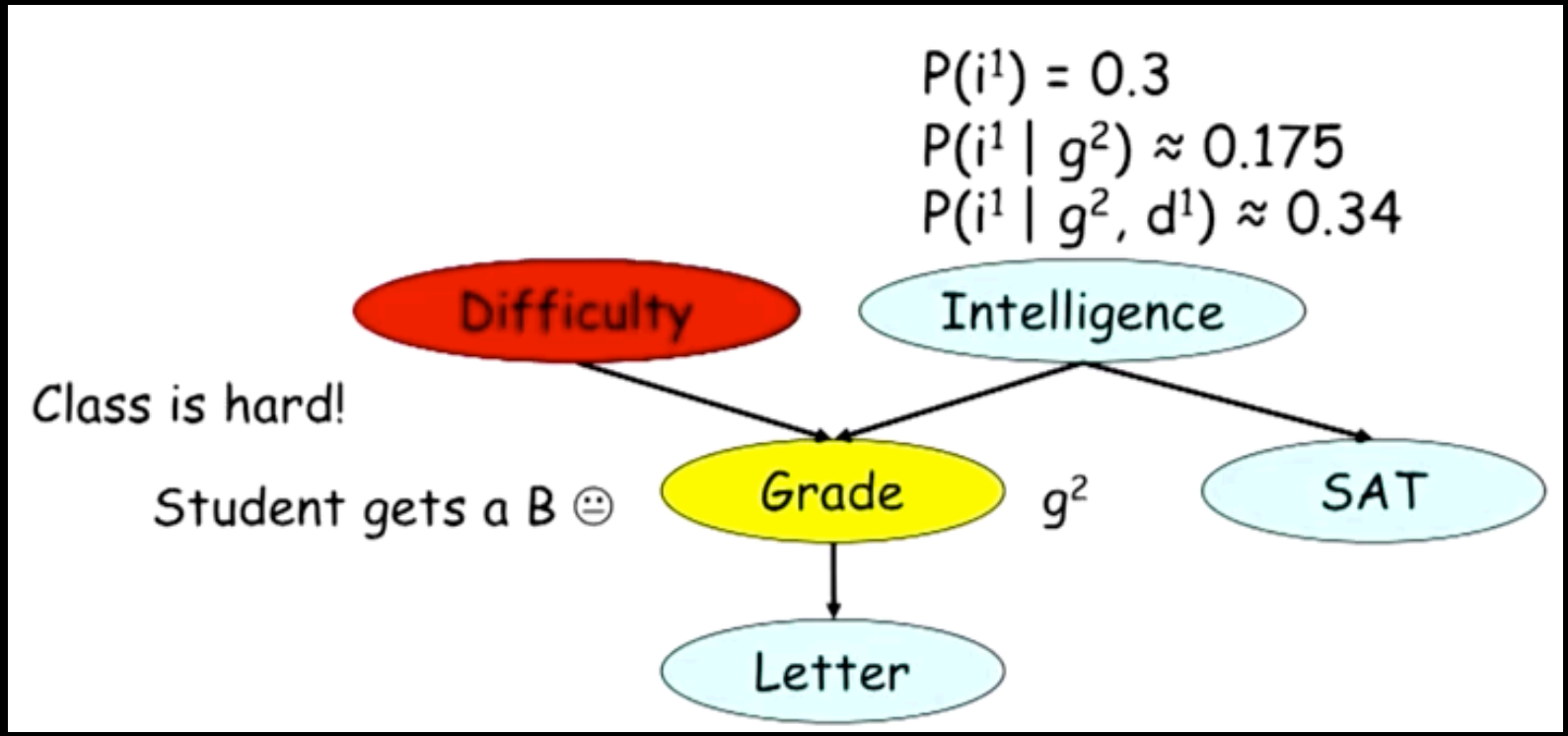
	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

Daphne Koller

Bayesian Belief Networks: Inter-causal Reasoning

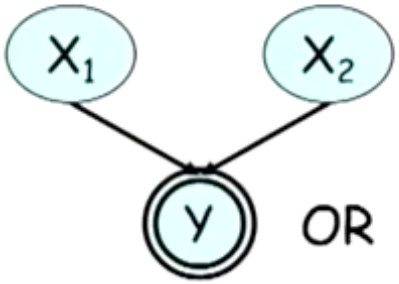


Bayesian Belief Networks: Inter-causal Reasoning



Inter-causal Reasoning Explained

x_1与x_2不再独立！



Y=1

X_1	X_2	Y	Prob
0	0	0	0.25
0	1	1	0.25
1	0	1	0.25
1	1	1	0.25

0.33
0.33
0.33

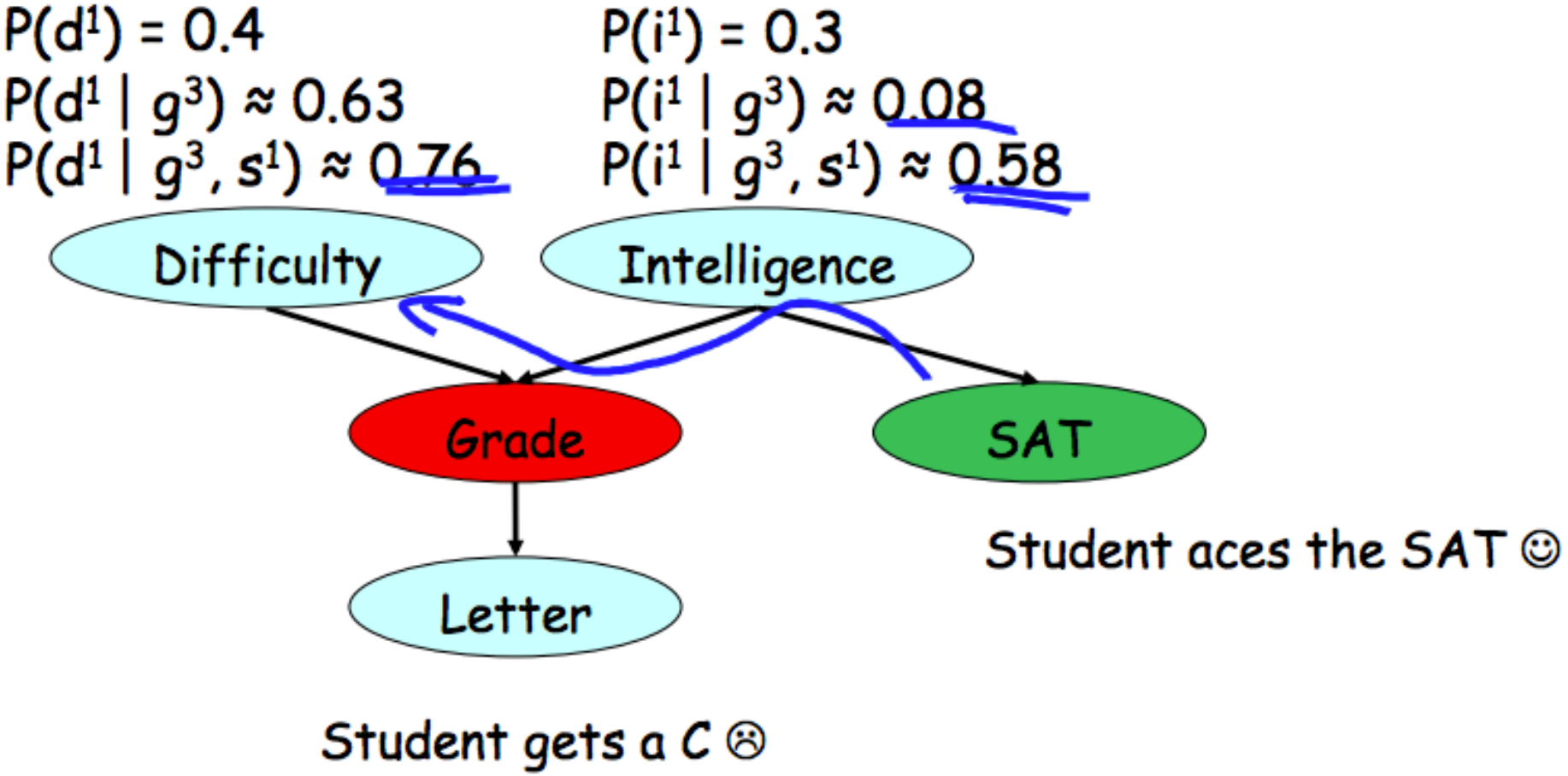
$$p(x_1 = 1) = \frac{2}{3} \quad p(x_2 = 1) = \frac{2}{3}$$

Condition: $x_1 = 1$

$$p(x_2 = 1 | x_1 = 1) = \frac{1}{2}$$

Even if two hidden causes are independent in the prior, they can become dependent when we observe an effect that they can both influence.

What is the student Aces the SAT?



Daphne Koller

Flow of Probabilistic Influence

$$X \rightarrow Y$$

$$Y \rightarrow X$$

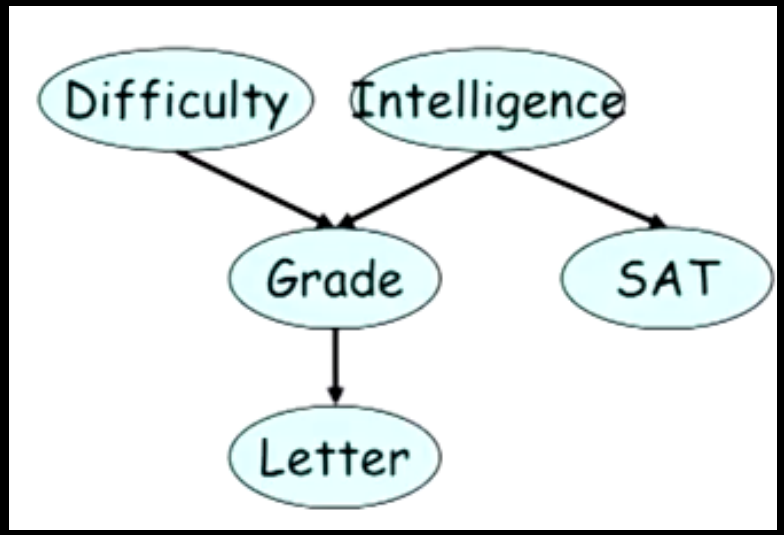
$$X \rightarrow W \rightarrow Y$$

$$X \leftarrow W \leftarrow Y$$

$$X \leftarrow W \rightarrow Y$$

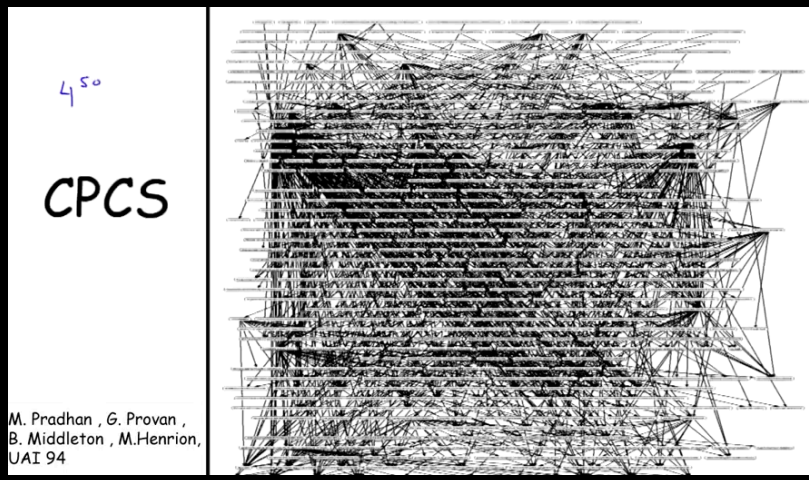
$$X \rightarrow W \leftarrow Y$$

V Structure

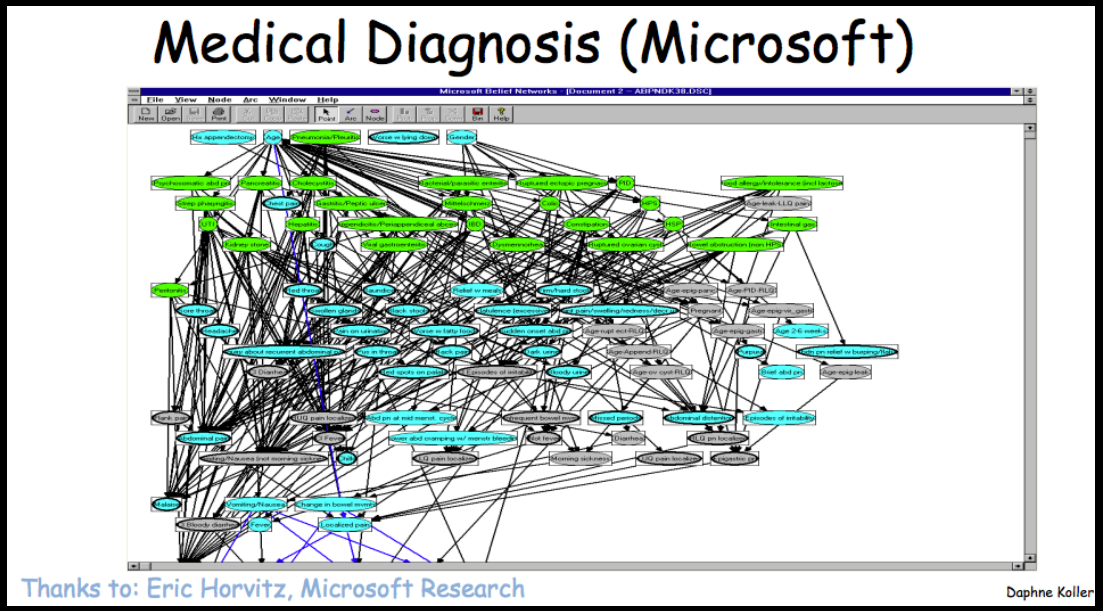


BBN Application: Medical Diagnosis: Pathfinder 1992

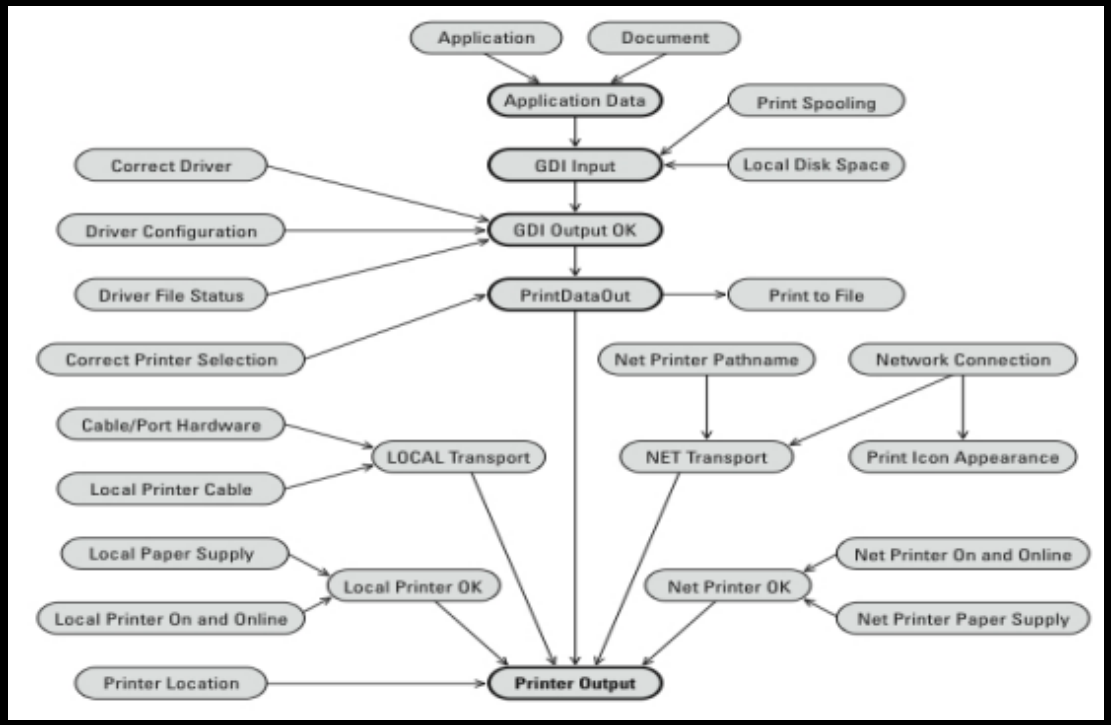
- $Help\ pathologist\ diagnose\ lymph\ node\ pathologies$
- $60\ difference\ diseases$
- Pathfinder I: rule-based system
- Pathfinder II: Naïve Bayes, Superior Performance
- Pathfinder III: Naïve Bayes with knowledge engineering
- Pathfinder IV: Full Bayesian Network
- BN Model agreed with expert panel in 50/53 cases, v.s. 47/53 for naïve Bayes model ,outperform the physician who design the model
- CPCS (# of variables = 500)
- On average 4 values/variable
- $Full\ joined\ distribution$: 4^{500}
- CPD: 133 Million Parameters
- With assumption: 8254 parameters



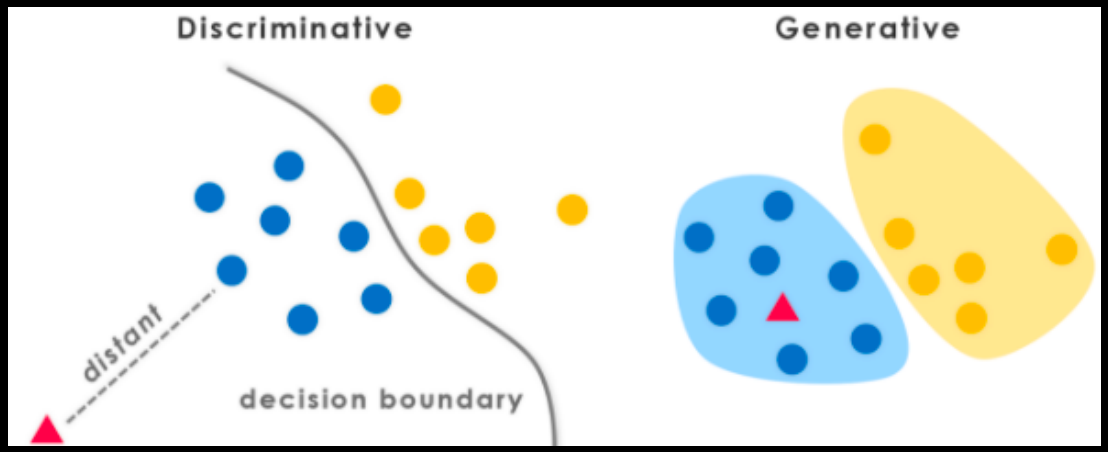
BBN Application: Medical Diagnosis: Microsoft



BBN Application: Microsoft troubleshooters



Recap: Discriminative v.s. Generative Learning Model



Generative

Learn $p(x|y)$& $p(y)$

Learn $p(y|x)$ indirectly

$$p(y|x) = \frac{p(x|y) * p(y)}{p(x)} \propto p(x|y) * p(y)$$

Data X

Label Y

Learn $p(y|x)$ directly

Discriminative

Recap: Deep Learning

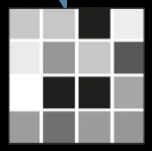
A Deep Learning algorithm is presented with millions of images made up of simple pixels.

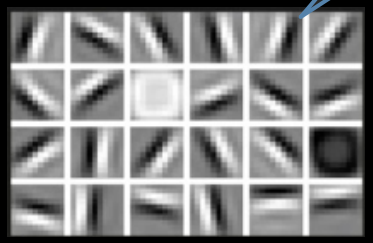
The algorithm discovers simple "regularities" that are present across many/all images, like curves & lines.

The algorithm discovers how these regularities are related to form higher-level concepts.

Ultimately, the system gains a high-level understanding of the original data…

All automatically!





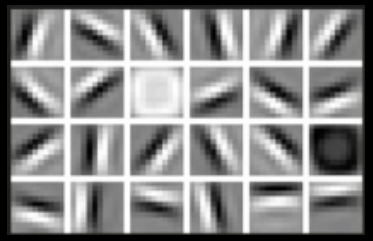
Why is Deep Learning so Exciting?

Because Deep Learning algorithms discover these increasingly abstract concepts embedded in the data automatically!

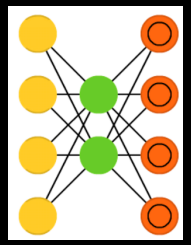
Traditional Machine Learning approaches required a human to identify these and program them by hand – and it never really worked.

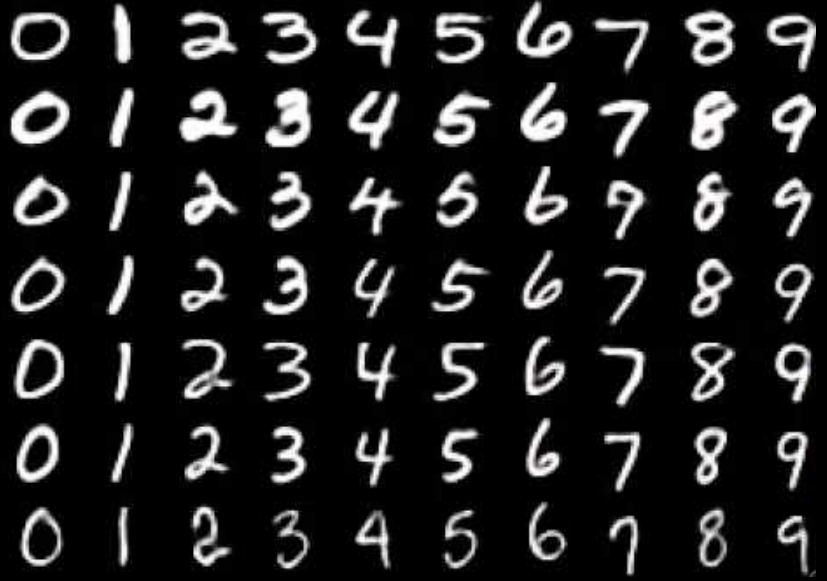
Hastad proof - Problems which can be represented with a polynomial number of nodes with k layers, may require an exponential number of nodes with *k*-1 layers (e.g. parity)

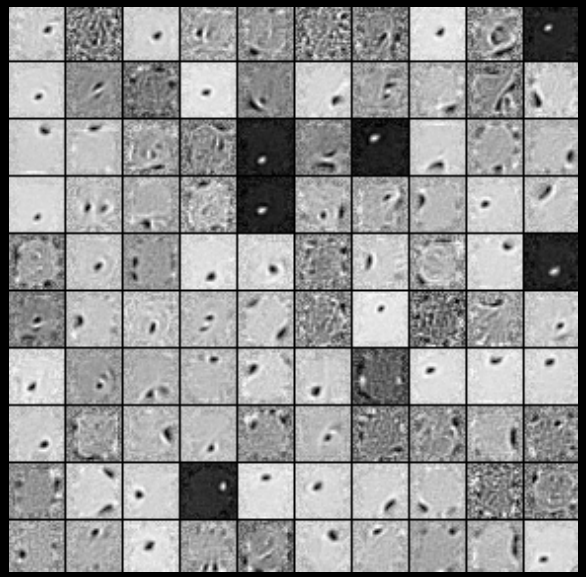
Sub-features created in deep architecture can potentially be shared between multiple tasks： Transfer/Multi-task learning



Auto-encoder MNIST

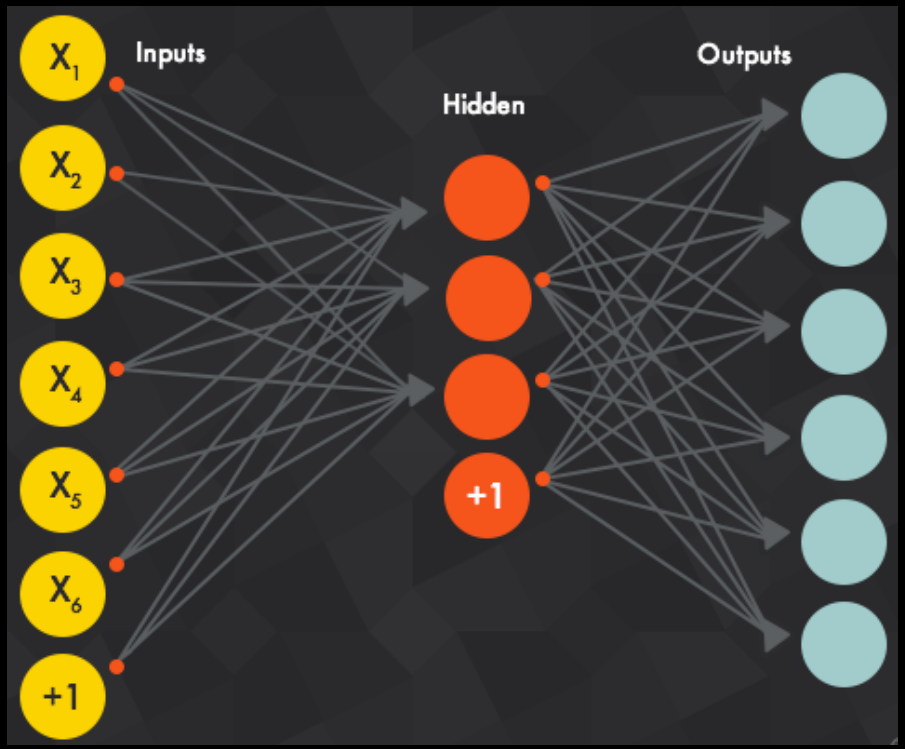






30% noise, 1 hidden layer, 500 neurons
http://deeplearning.net/tutorial/dA.html

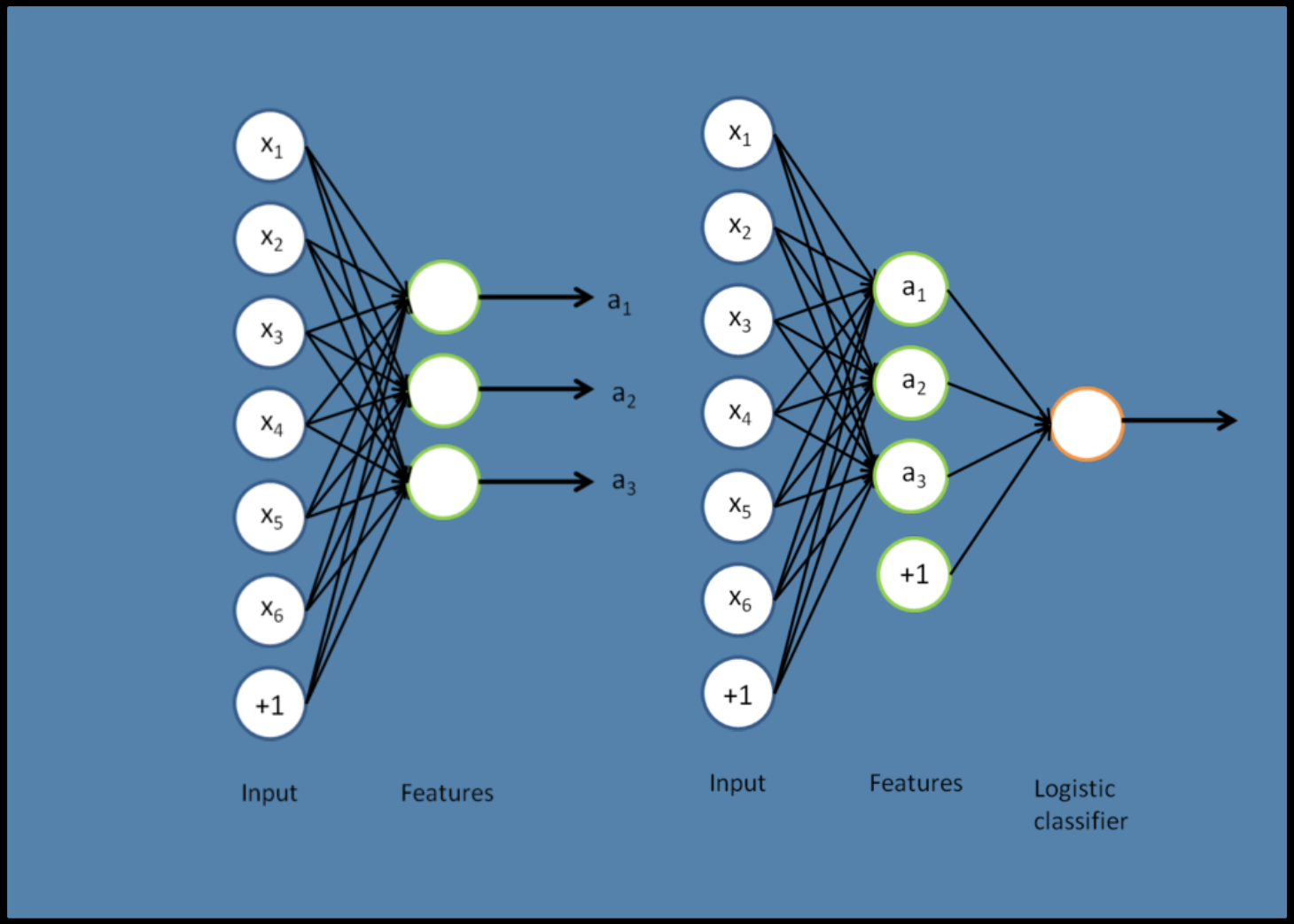
Auto-encoder 学习数据集的压缩表达（编码）



A type of unsupervised learning which tries to discover generic features of the data

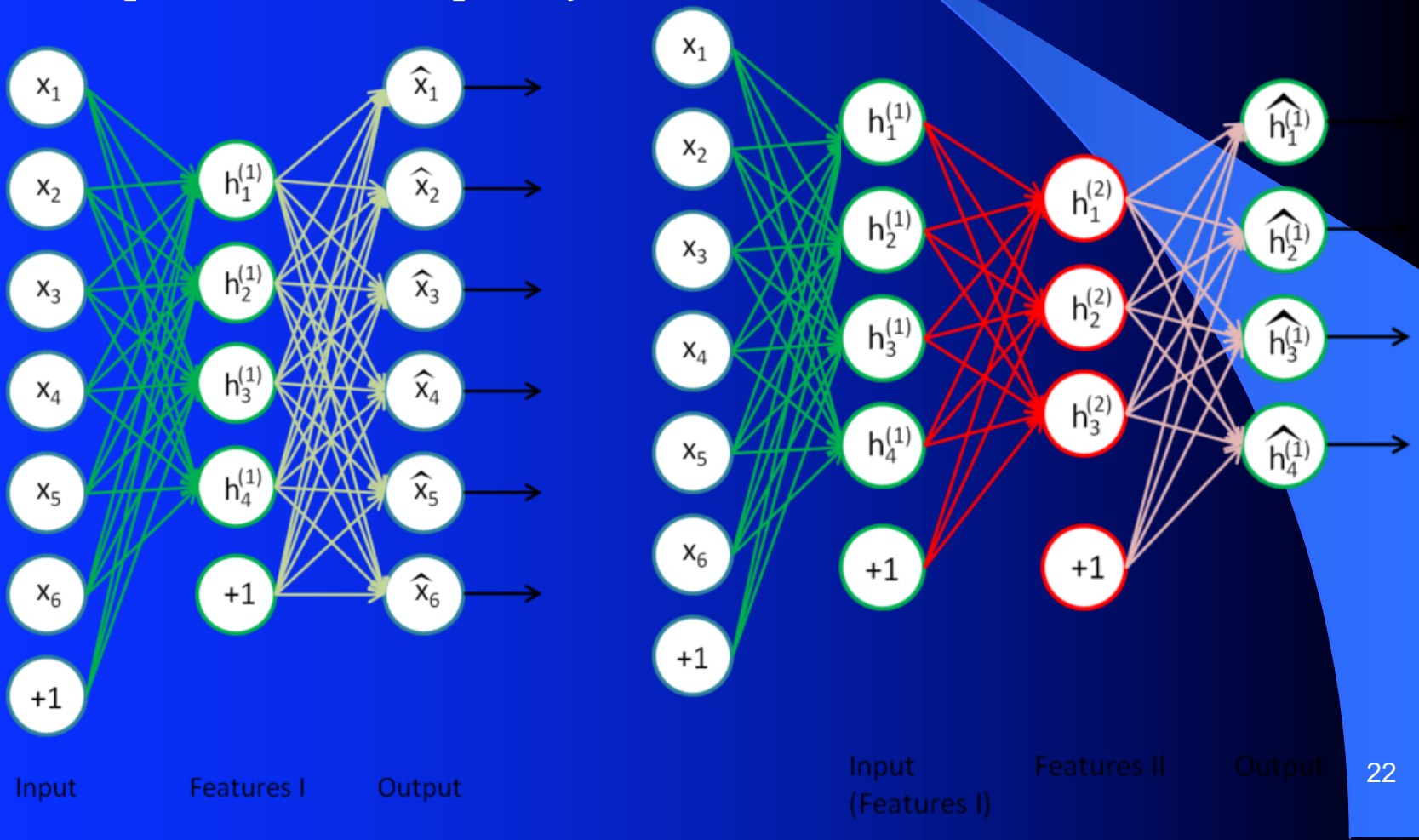
- Learn identity function by learning important sub-features
- Compression, etc.
- Can use just new features in the new training set or concatenate both

Auto-encoder 学习数据集的压缩表达（编码）



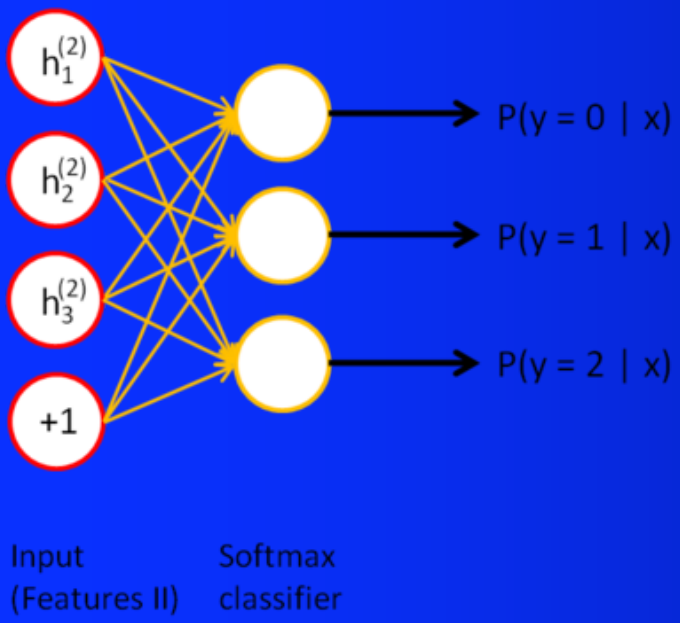
Stacked Auto-Encoders

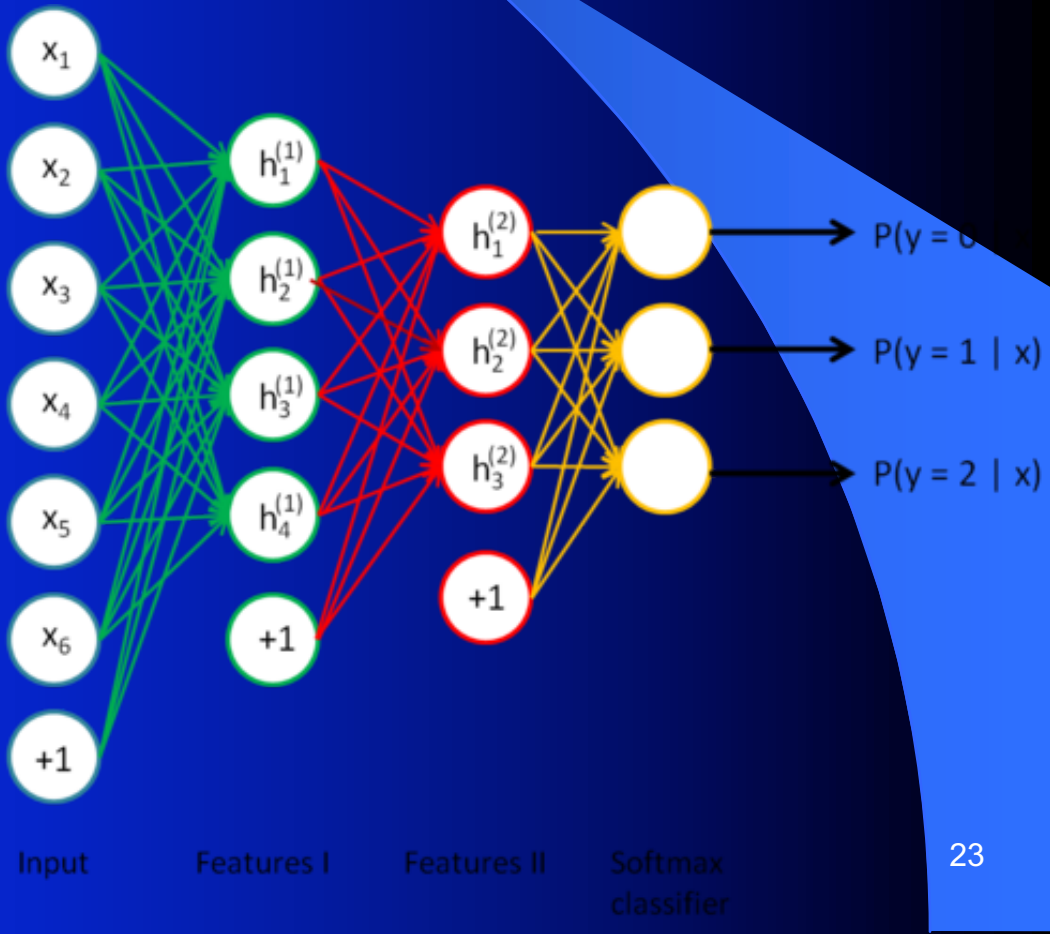
- Bengio (2007) – After Deep Belief Networks (2006)
- Stack many (sparse) auto-encoders in succession and train them using greedy layer-wise training
- Drop the decode output layer each time



Stacked Auto-Encoders

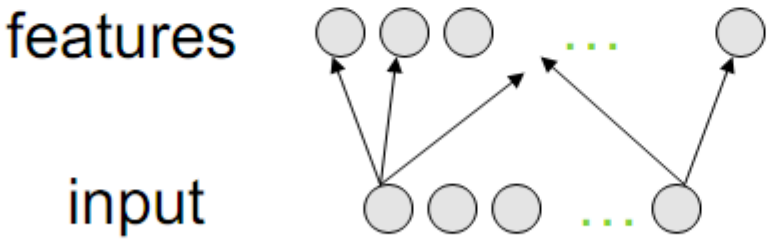
- Do supervised training on the last layer using final features
- Then do supervised training on the entire network to fine-tune all weights



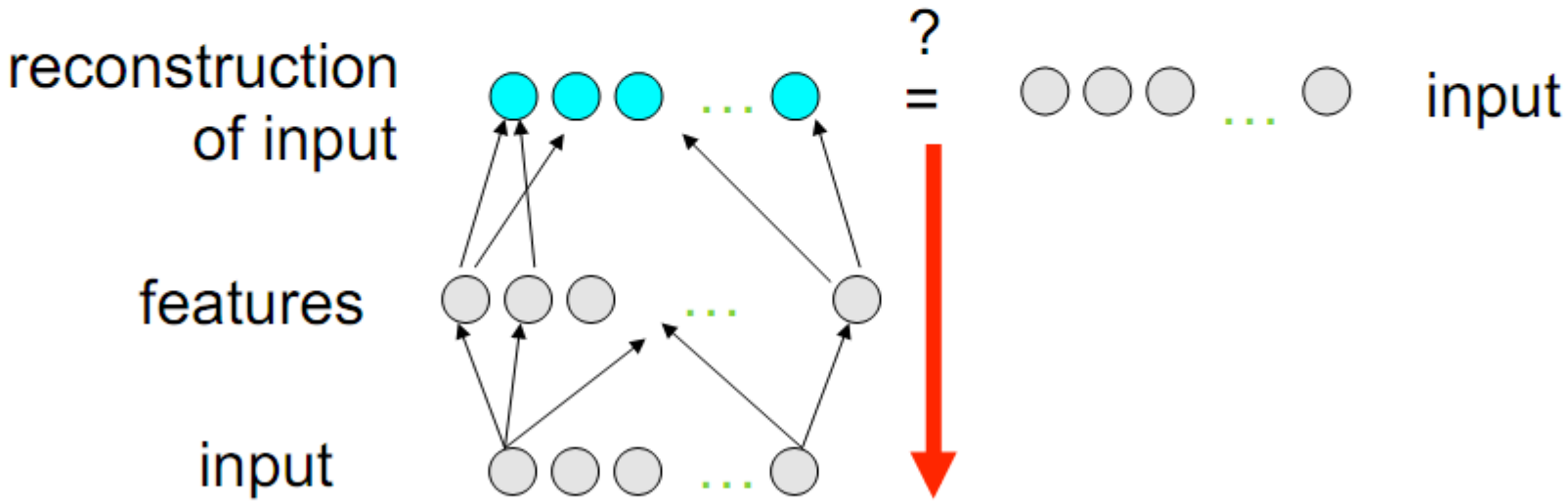


Deep training

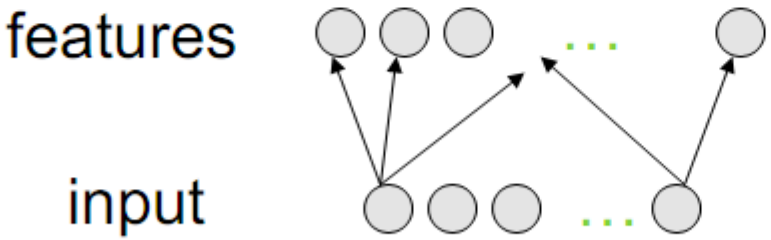
Layer-Wise Unsupervised Pre-training



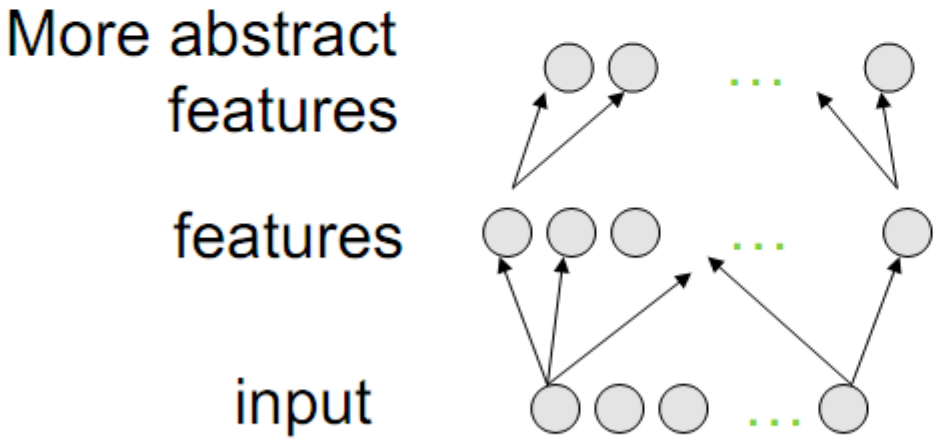
Layer-Wise Unsupervised Pre-training



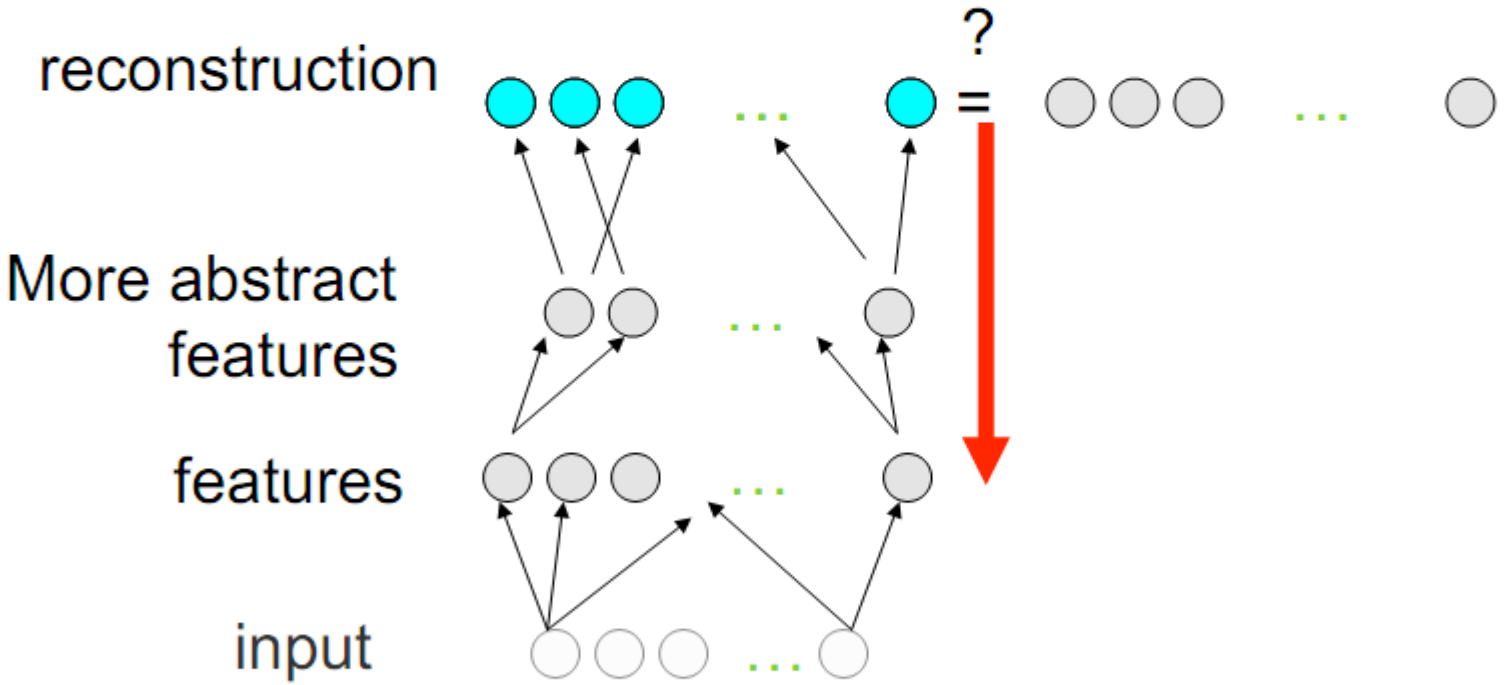
Layer-Wise Unsupervised Pre-training



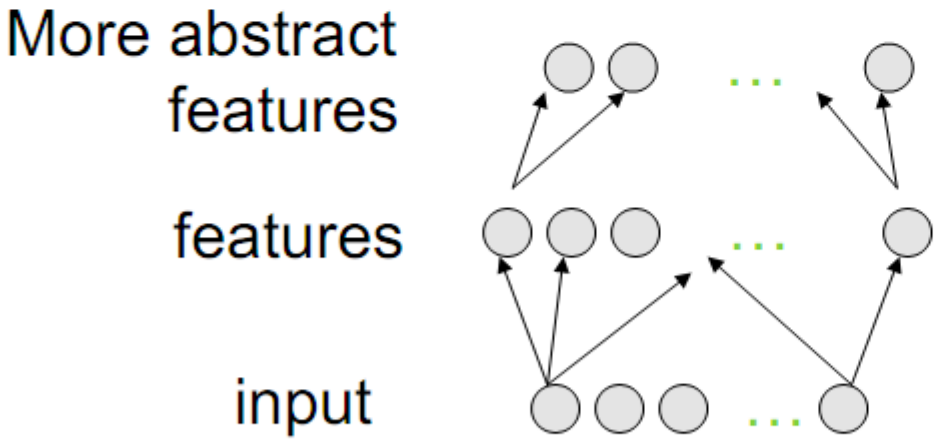
Layer-Wise Unsupervised Pre-training



Layer-Wise Unsupervised Pre-training



Layer-Wise Unsupervised Pre-training



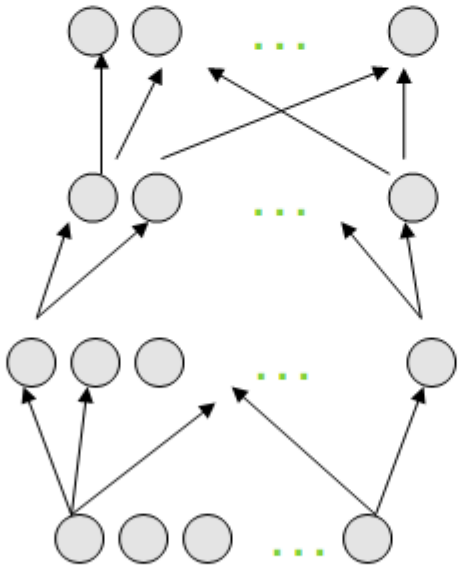
Layer-Wise Unsupervised Pre-training

Even more abstract features

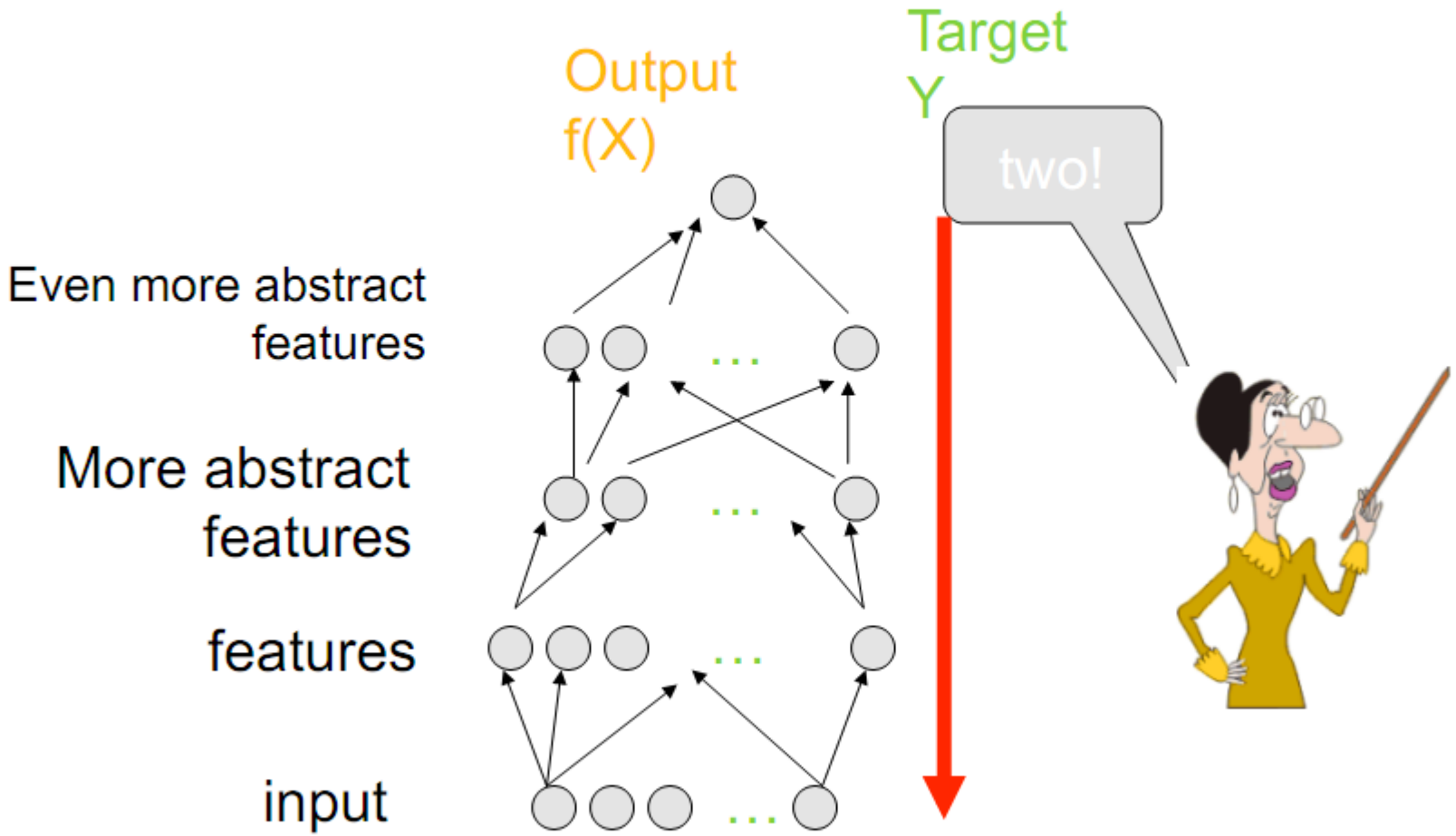
More abstract features

features

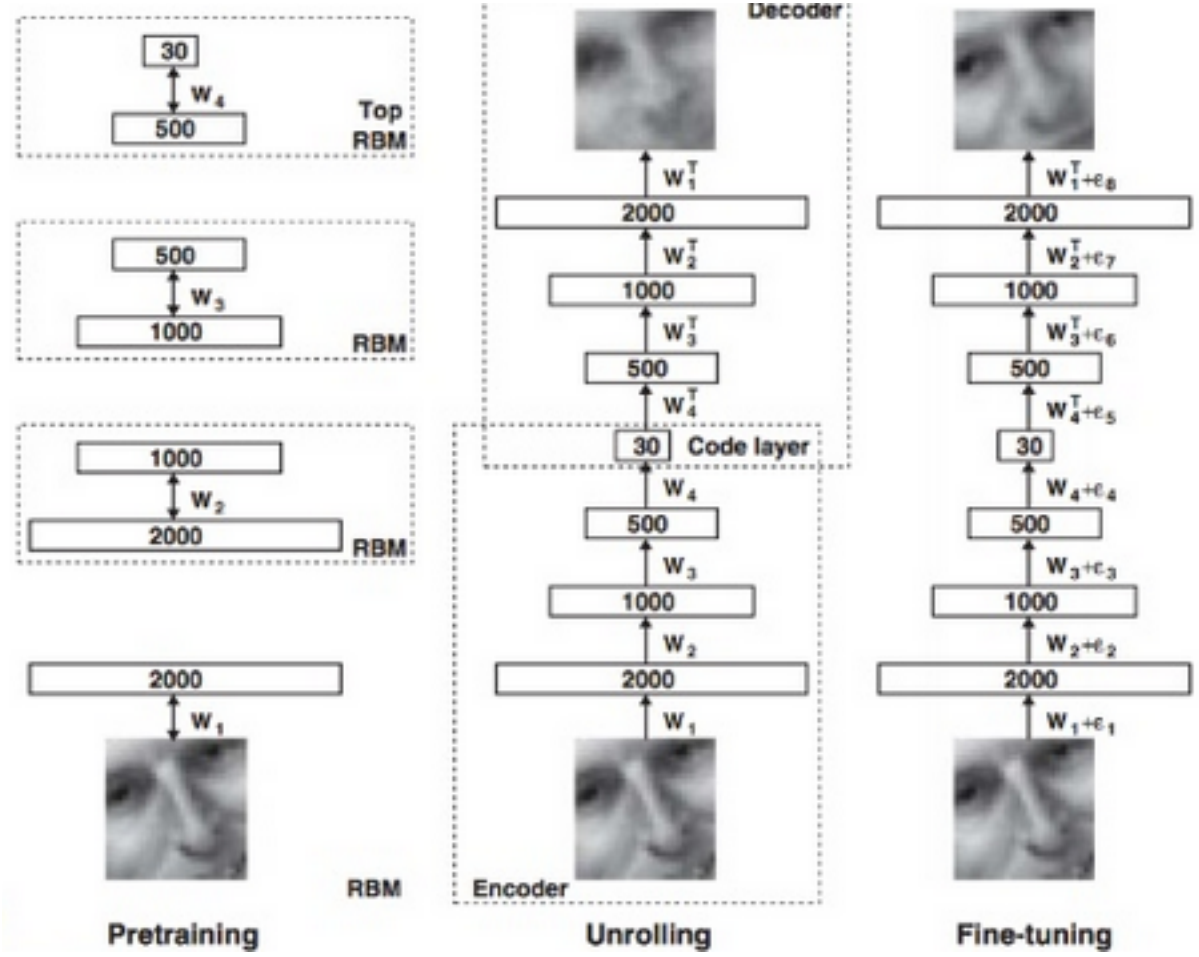
input



Supervised Fine-Tuning

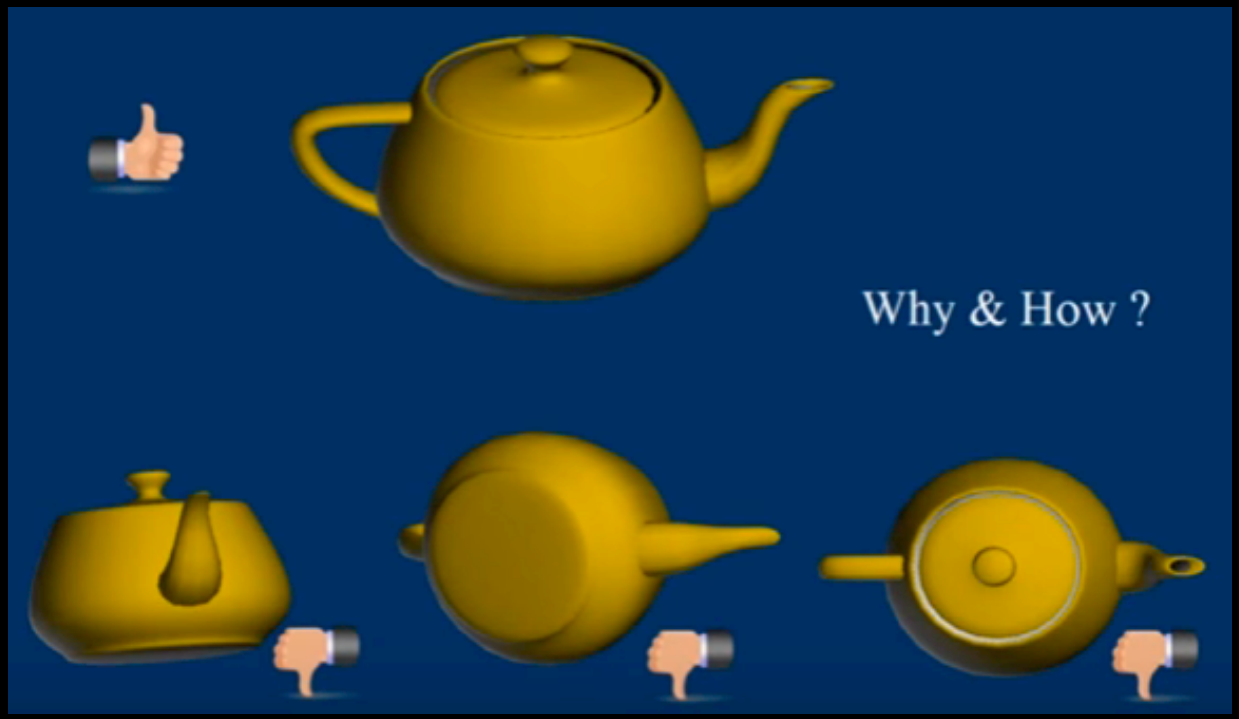


Deep auto-encoder



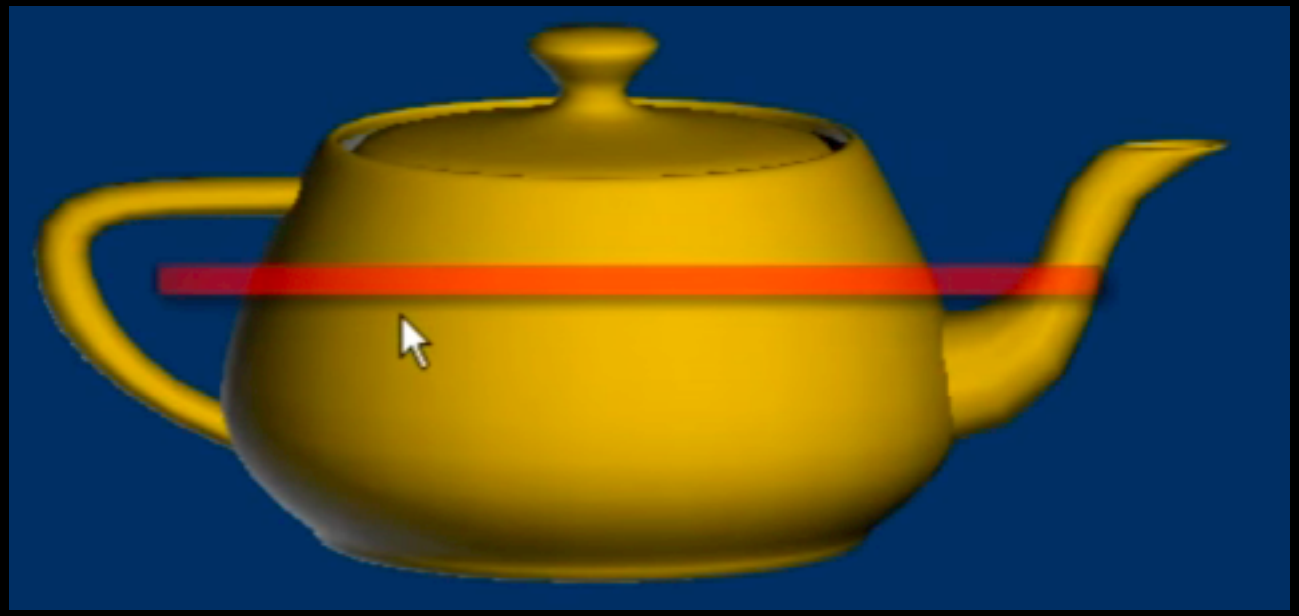
Hinton & Salakhutdinov, Science 2006

Principal Component Analysis

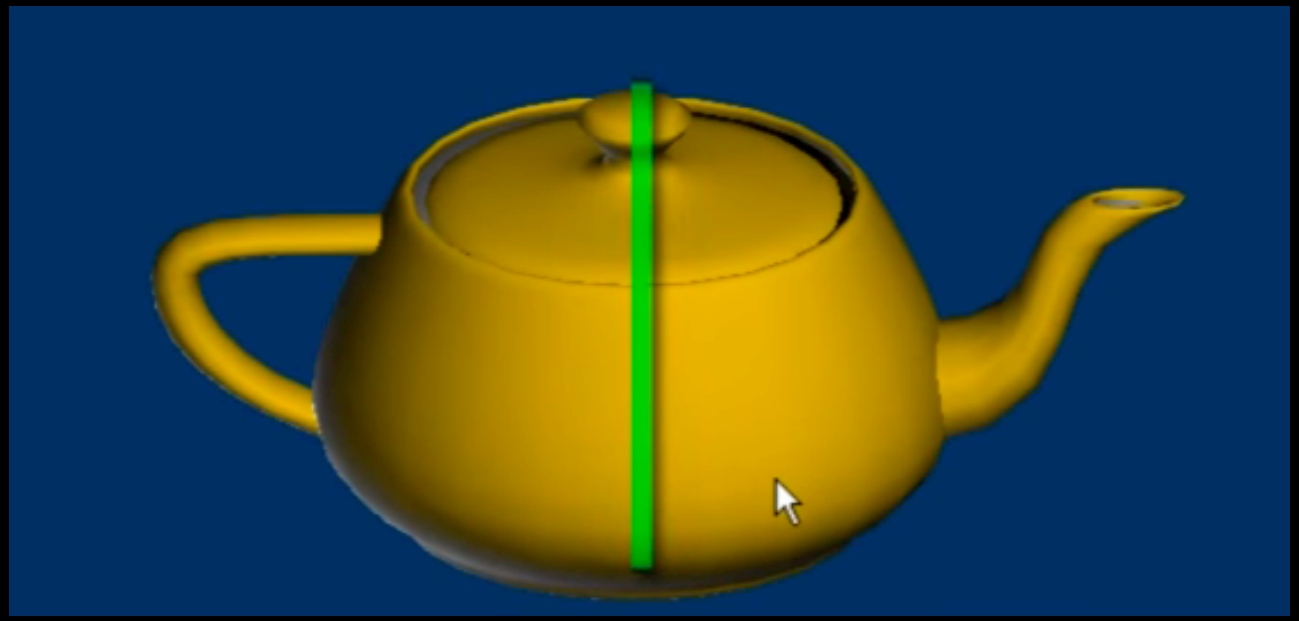


- Why this position?
- Because it provides the most visual information
- How do we find this position?
- Rotate the teapot according to PCA algorithm

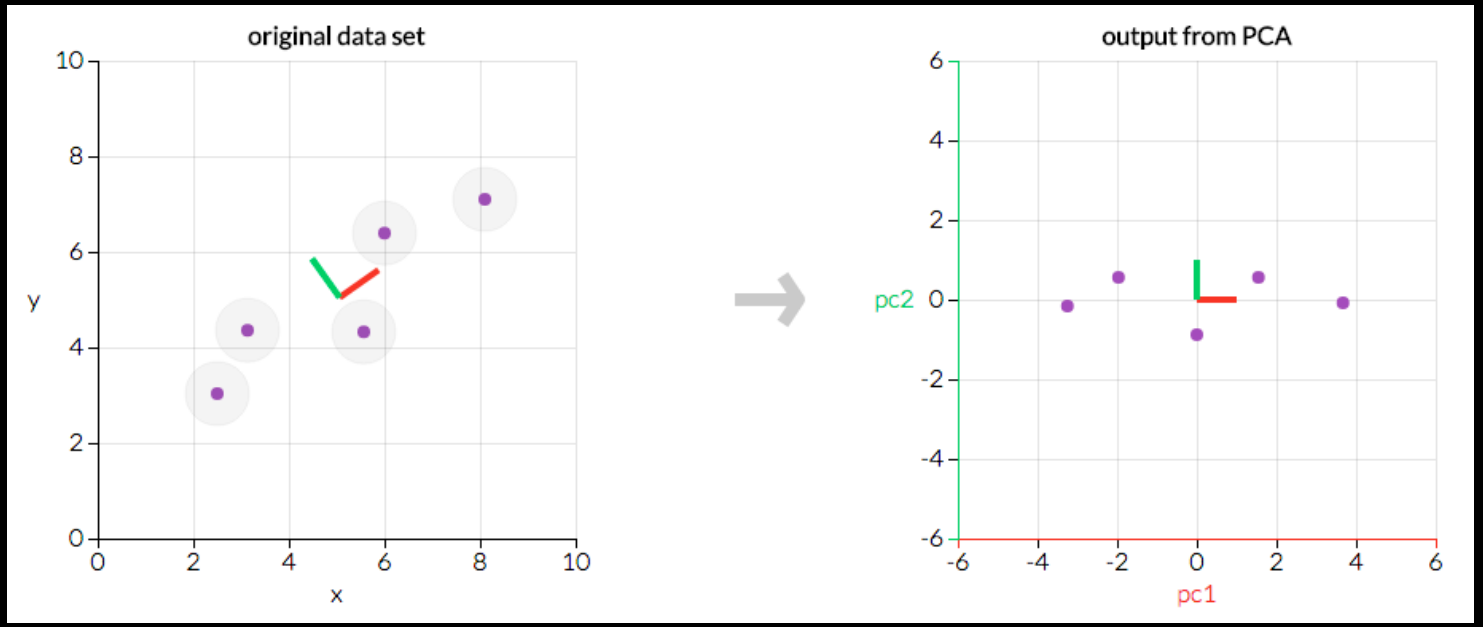
PCA: Find the longest axis



PCA: Find the 2nd longest axis

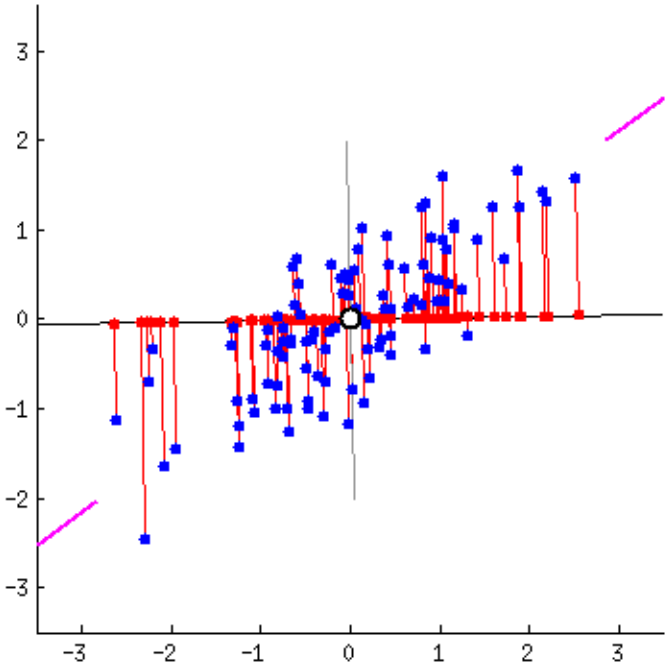


PCA

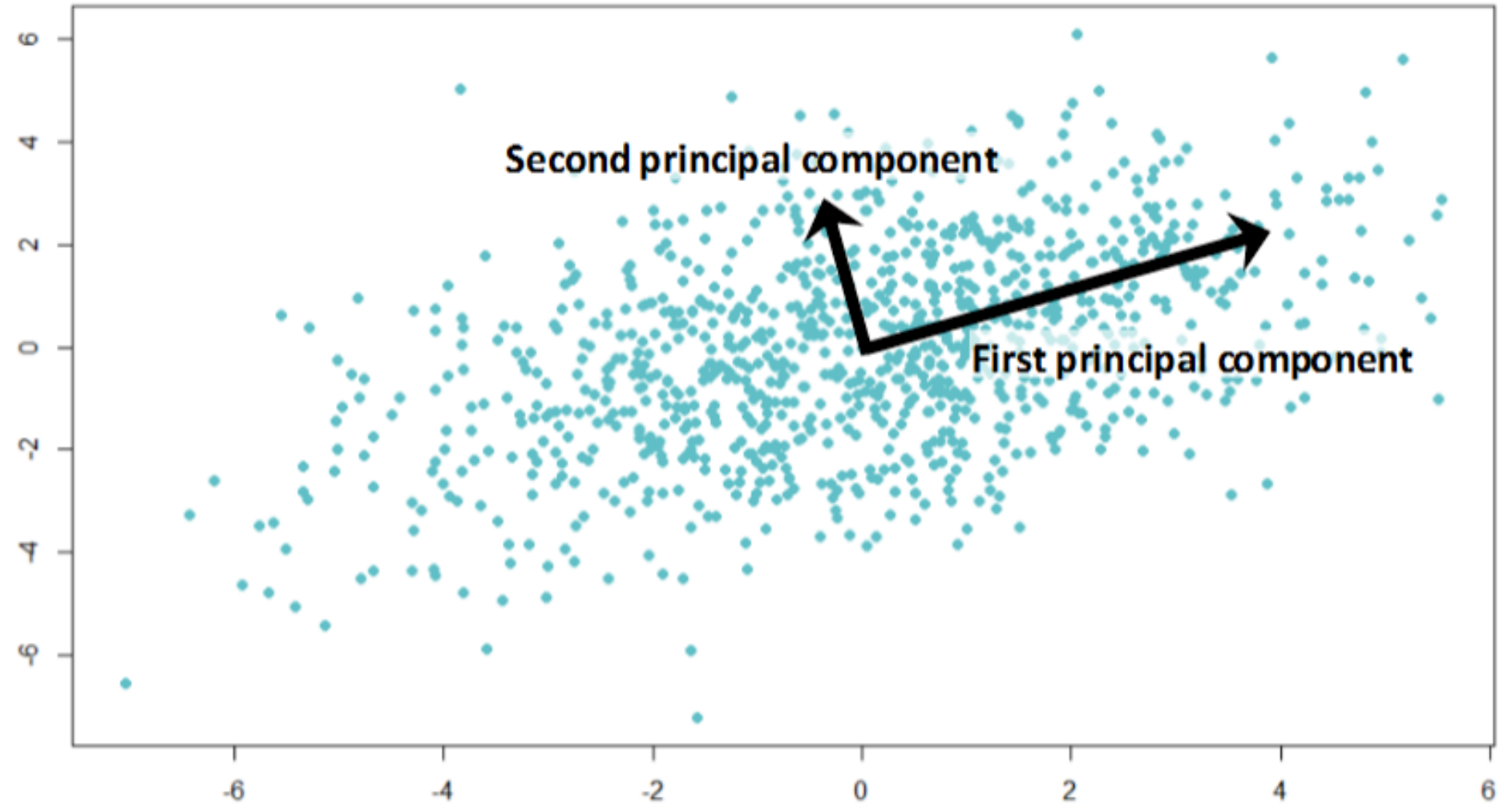


http://setosa.io/ev/principal-component-analysis/

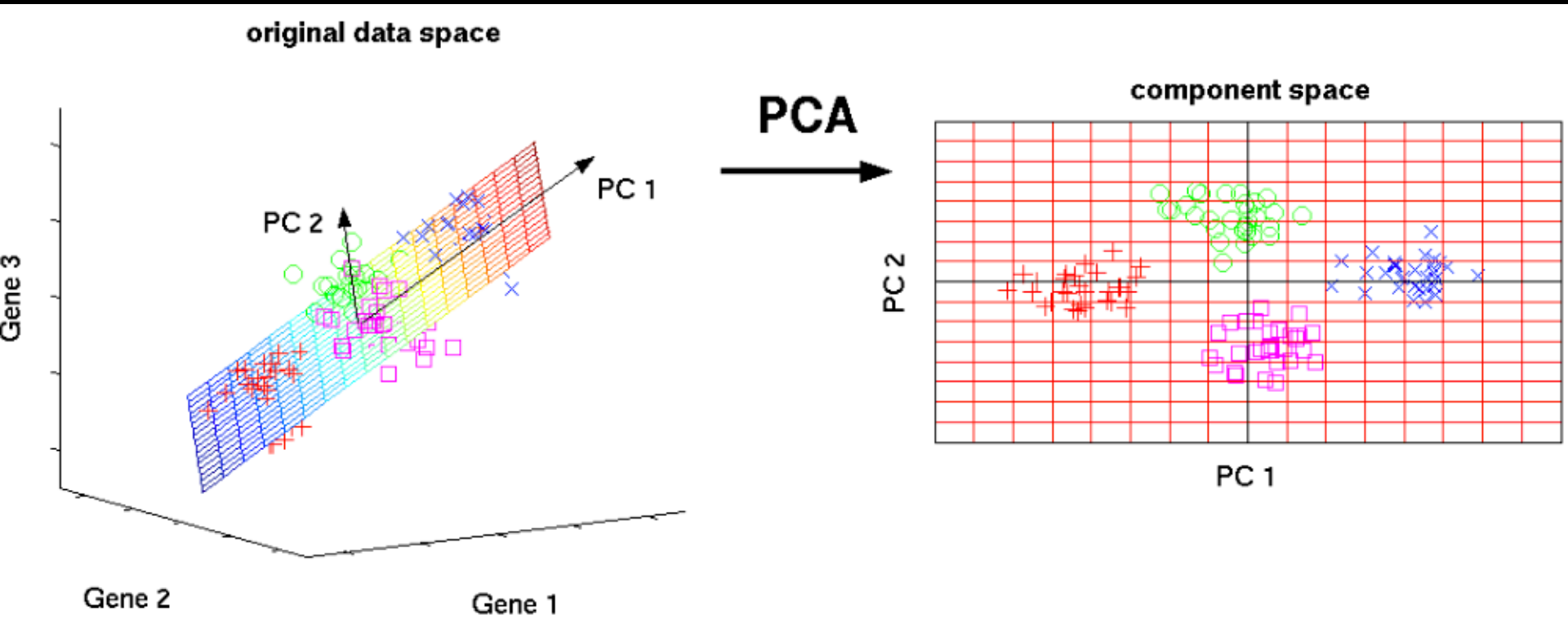
PCA



PCA

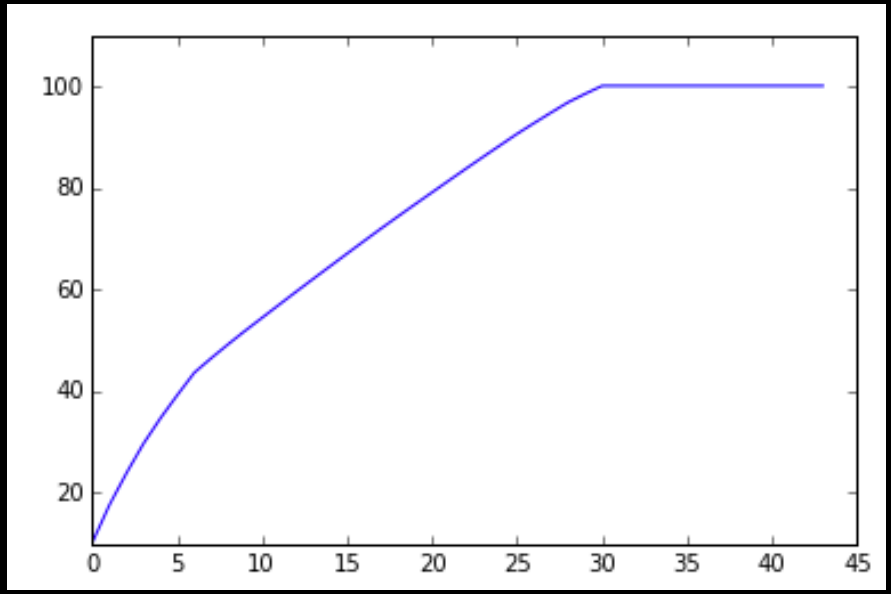

Second principal component
First principal component
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

PCA

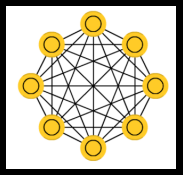

original data space
PC 1
PC 2
Gene 3
Gene 2
Gene 1
PCA
component space
PC 2
PC 1

PCA in python

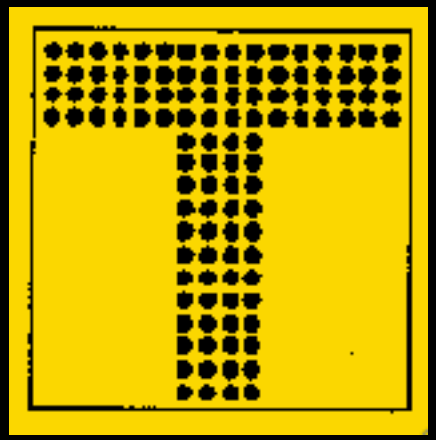
```
import numpy as np
from sklearn.decomposition import PCA
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import scale
%matplotlib inline
#Load data set
data = pd.read_csv('Data.csv')
#convert it to numpy arrays
X=data.values

X = scale(X)
pca = PCA(n_components=44)
pca.fit(X)
#The amount of variance that each PC explains
var= pca.explained_variance_ratio_
#Cumulative Variance explains
var1=np.cumsum(np.round(pca.explained_variance_
ratio_, decimals=4)*100)
```

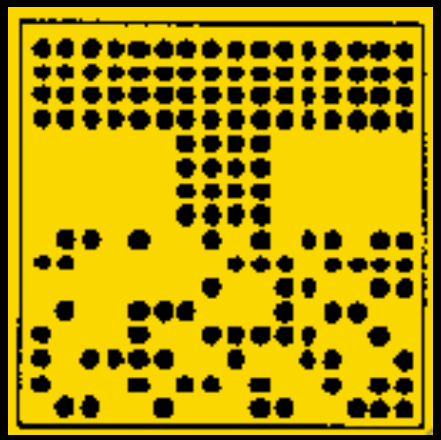


Hopfield Nets



- 1980年代， Hopfield Nets与BP是神经网络再次繁荣的主要原因
- **The purpose of a Hopfield net is to store 1 or more patterns and to recall the full patterns based on partial input.**





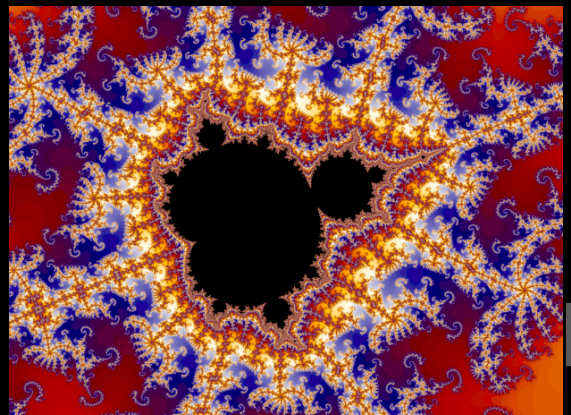
- OCR识别场景，字母大写T的下半部分被污染
- 每个像素对对应Hopfield网络的一个节点
- 利用26个字母的大小写训练网络
- 如果输入右图， Hopfield网络可以输出左图

http://www.tarkvaralabor.ee/doodler/

Hopfield Nets

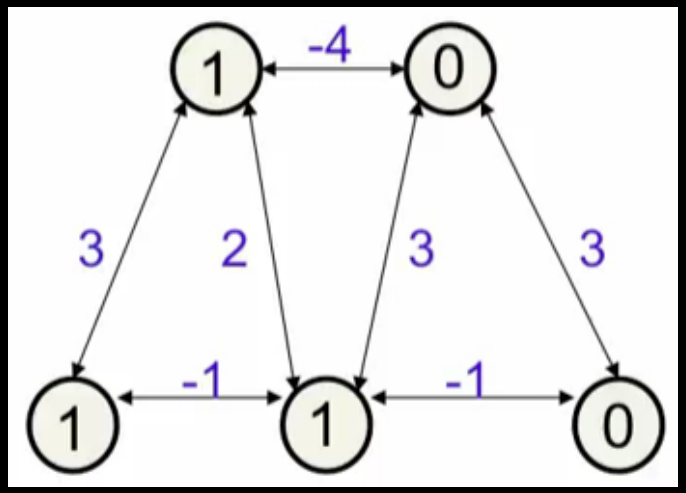
- 简单精巧，可被用来作为存储的分布式活动模式的记忆体
- Hopfield Nets是最简单基于能量的网络（EBN）之一
- EBN的属性是从全局能量函数（Global Energy Function）中衍生出来的
- It is composed by binary threshold units with recurrent connections between them
- 非线性的Recurrent Neural Network (RNN)非常难以分析，因为他们最终可能收敛、震荡或者混沌态（Chaotic）
- John Hopfield意识到，如果连接是对称的，就存在一个全局能量函数：
  - 整个网络的每个Binary Configuration都对应一个能量值
  - The binary threshold rule causing the network to go downhill in energy, if the rule are being kept applying, it will end up in an energy minima.

$$E = -\sum_i s_i b_i - \sum_{i<j} s_i s_j w_{ij}$$



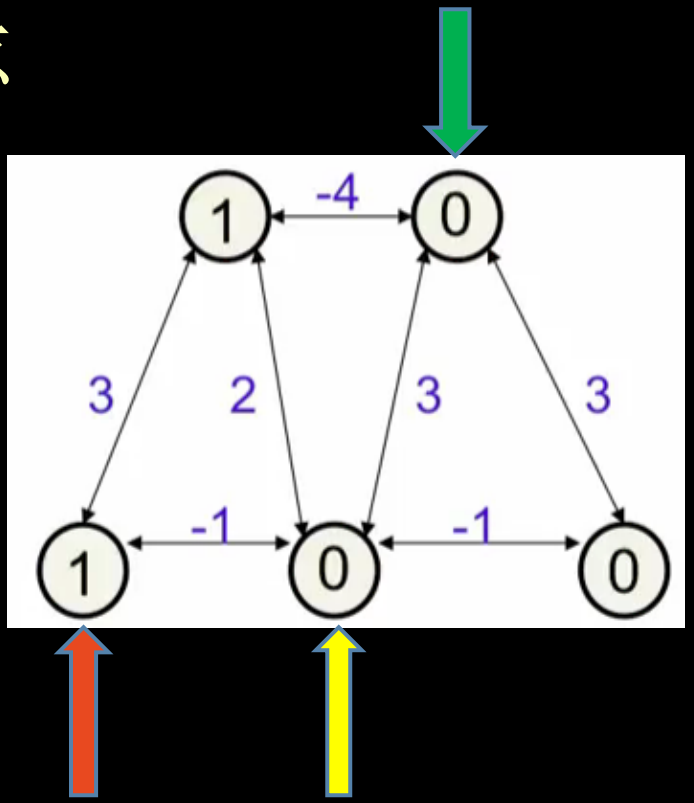
Hopfield Nets 收敛于能量最低点

- 目标：发现网络的能量最低点
- 步骤：从一个随机状态开始，每次随机更新一个Unit，看能否降低网络的整体能量



$-E$ = Goodness = 4

但这并非是该网络的最低能量值，而只是2个最低值之一。



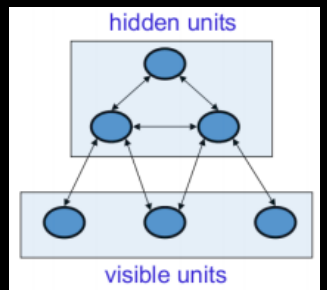
$$E = -\sum_i s_i b_i - \sum_{i<j} s_i s_j w_{ij}$$

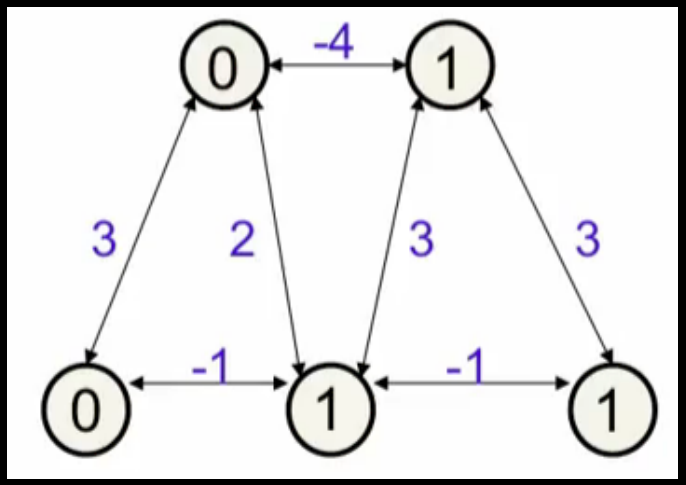
$-E$ = Goodness = 3

where bi is the bias of the i th unit, s is 0 or 1 1 depending on whether the unit is turned off or on respectively. And wij is the weight of the connection between units i and j.

Hopfield Nets 能量最低的特性能帮助实现记忆

- 网络的能量最低点在右三角的节点相互支持的情况下
- 左右两个三角互相“憎恨”对方
- 左三角与右三角不同处在于，有一个Weight=2，导致了右三角存在网络能量最低值
- Hopfield(1982)提出记忆可能是（对称权重的）神经网络的能量最低值
- 记忆可能错误，Hopfield网络可帮助恢复记忆
- 了解部分可以恢复整体
- “Reconstructing a dinosaur from a few bones”
- 增加了隐藏Unit的Hopfield Net



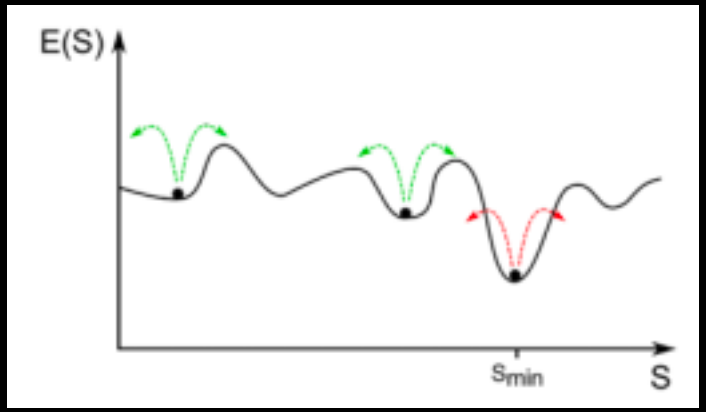


$-E$ = Goodness = 5

- Boltzmann machine 与Hopfield Net都是Binary Units组成的能量网络，但是BM不以记忆作为目标，而是试图学习输入数据的表达，以更紧凑和有意义的方式表达数据

Noisy Network

- The Binary Threshold Decision rule always goes downhill, i.e., reduces energy.
- Hence it is impossible to escape from a local minima.
- Solution : Use random noise to escape from poor, shallow minima.
  - Start with a lot of noise to escape the energy barriers of poor local minima.
  - Slowly reduce the noise so that the system ends up in a deep minima.
  - This process is called “Simulated Annealing”



How to add noise

- Replace the binary threshold units by binary stochastic units that make biased random decisions.
- The “temperature” controls the amount of noise.
- Unit i then turns on with the probability given by the logistic function:

$$P(S_i = 1) = \frac{1}{1 + e^{-\frac{\Delta E_i}{T}}}$$

T=0	Deterministic (Hopfield Net)
T→ ∞	Complete Chaos
T=1	Approaches Boltzmann Distribution

As we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns—the ones we don't know we don't know.

Rumsfeld

Thank you !

Contact information:

Wu Xuening (邬学宁)
SAP硅谷创新中心 首席科学家
A:上海市浦东新区晨辉路1001号
E: x.wu@sap.com
T:021-61085287

Deep Learning Use Cases

General use case	Industry
Sound	
Voice recognition	UX/UI, Automotive, Security, IoT
Voice search	Handset maker, Telecoms
Sentiment analysis	CRM
Flaw detection (engine noise)	Automotive, Aviation
Fraud detection (latent audio artifacts)	Finance, Credit Cards
Time Series	
Log analysis/Risk detection	Data centers, Security, Finance
Enterprise resource planning	Manufacturing, Auto., Supply chain
Predictive analysis using sensor data	IoT, Smart home, Hardware manufact.
Business and Economic analytics	Finance, Accounting, Government
Recommendation engine	E-commerce, Media, Social Networks
Text	
Sentiment Analysis	CRM, Social media, Reputation mgt.
Augmented search, Theme detection	Finance
Threat detection	Social media, Govt.
Fraud detection	Insurance, Finance
Image	
Facial recognition	
Image search	Social media
Machine vision	Automotive, aviation
Photo clustering	Telecom, Handset makers
Video	
Motion detection	Gaming, UX, UI
Real-time threat detection	Security, Airports