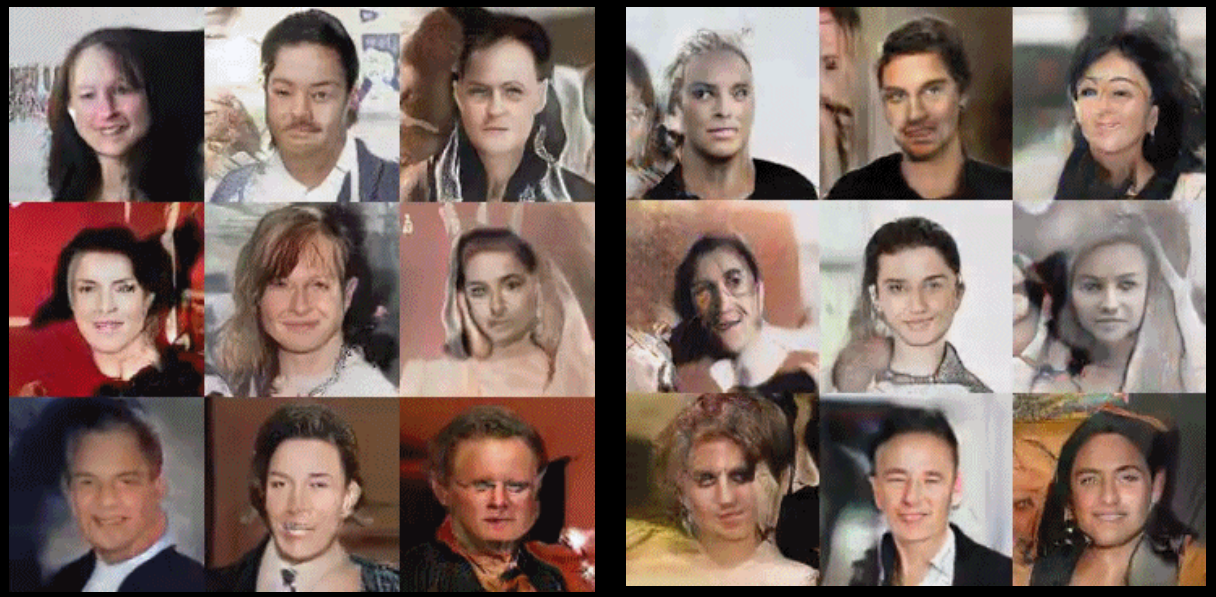
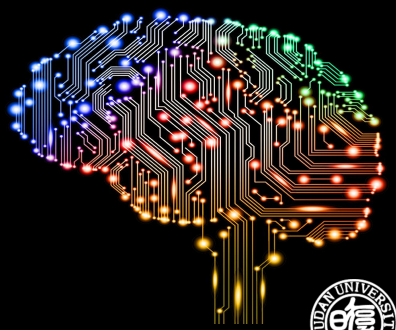
数据驱动的人工智能（6c）基于能量的神经网络

Data Driven Artificial Intelligence

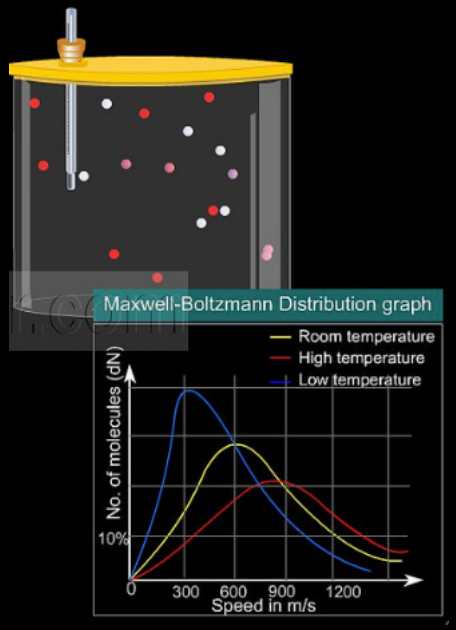
邬学宁 SAP硅谷创新中心

2017 / 03



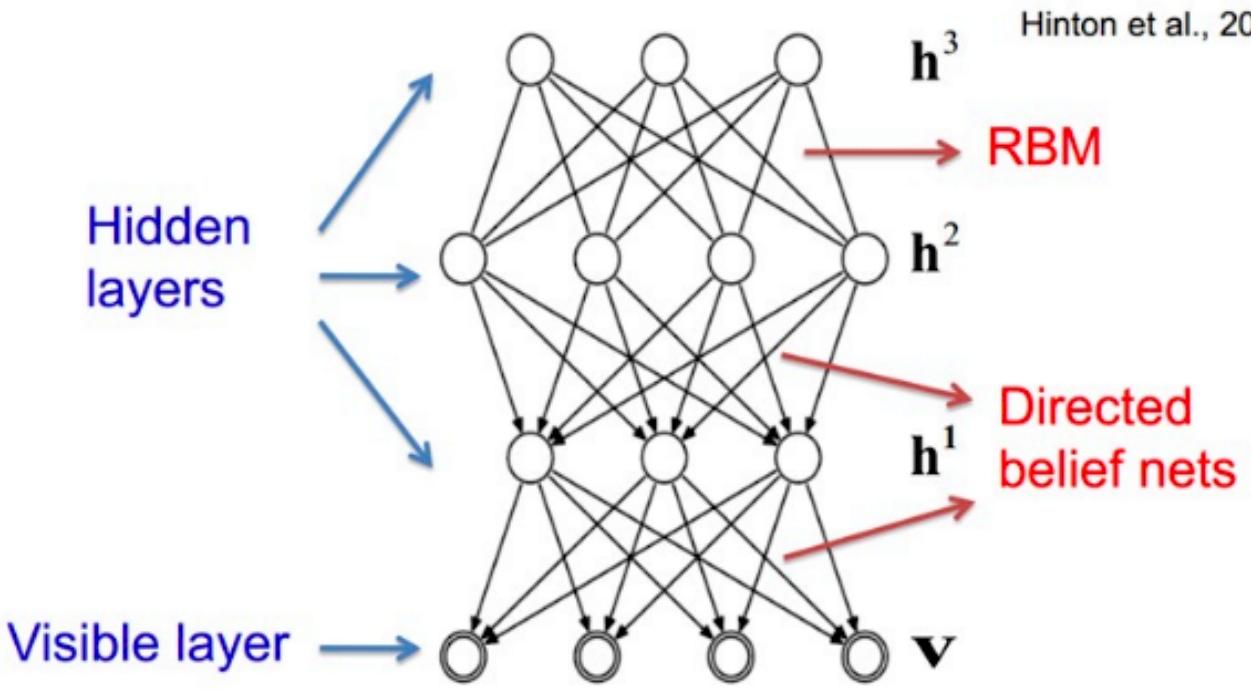


日程: EBN



- 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

DBN structure



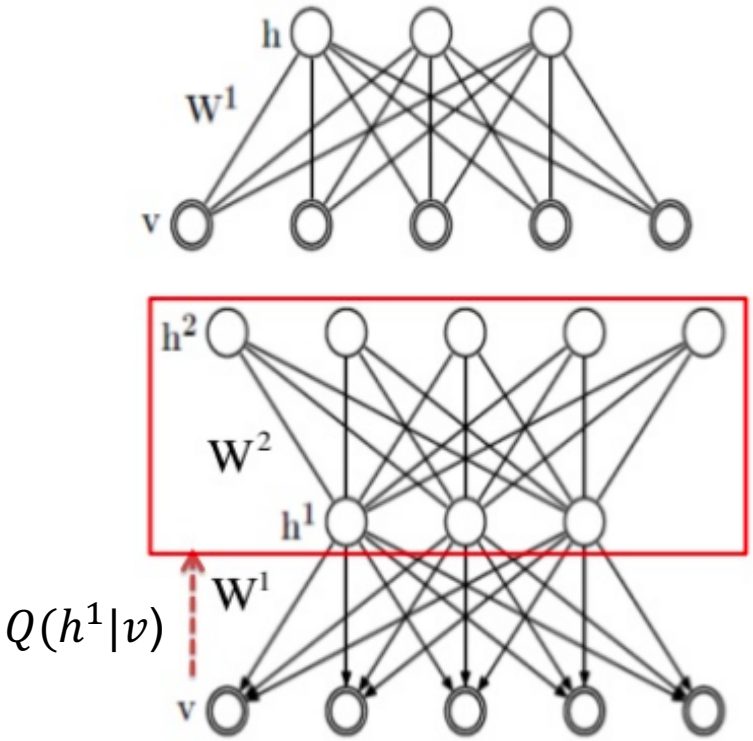
$$P(\mathbf{v},\mathbf{h}^1,\mathbf{h}^2,....,\mathbf{h}^l) = P(\mathbf{v}\,|\,\mathbf{h}^1)P(\mathbf{h}^1\,|\,\mathbf{h}^2)...P(\mathbf{h}^{l-2}\,|\,\mathbf{h}^{l-1})P(\mathbf{h}^{l-1},\mathbf{h}^l)$$

- Probabilistic generative model
- Deep architecture: multiple layer
- Unsupervised pre-learning provides good initialization of the network
- Supervised fine-tuning
- Belief Net is directed graph

DBN Greedy Training

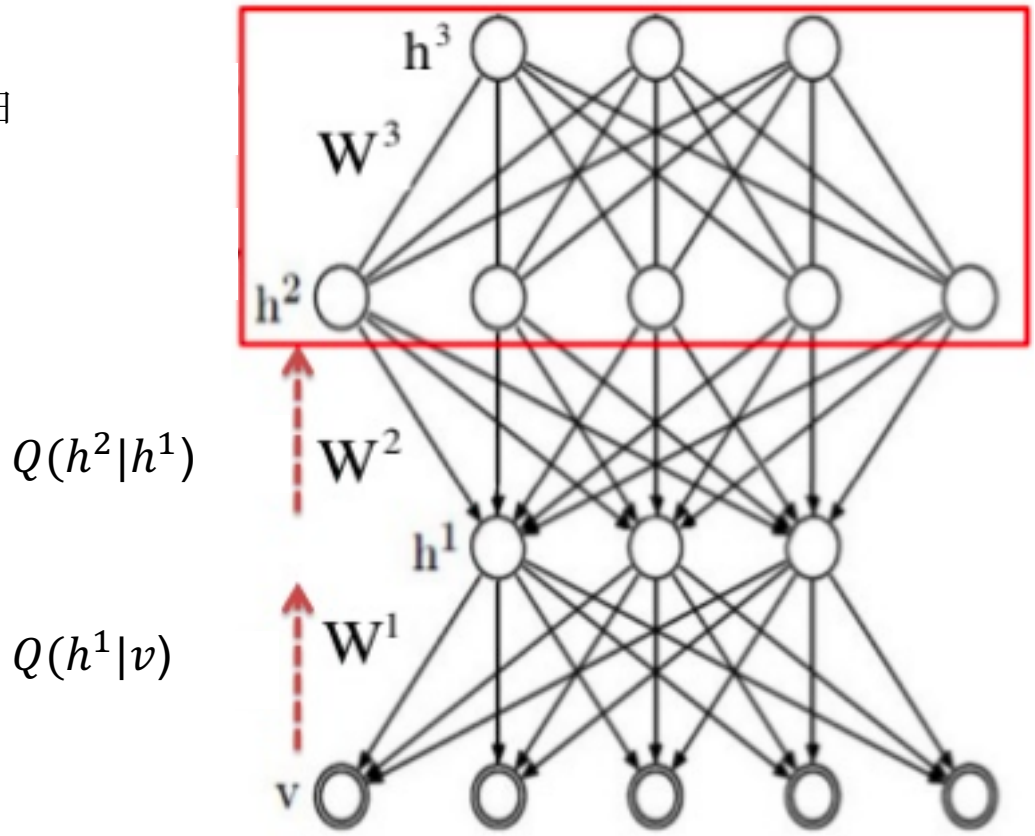
- 1st Step:
  - 以输入层v和隐层h构建RBM
  - 训练RBM

- 2nd Step:
  - 在原有RBM之上叠加一个新的隐层，形成一个新的隐层
  - 固定W^1，从$Q(h^1|v)$中抽样h^1，训练新RBM的W^2



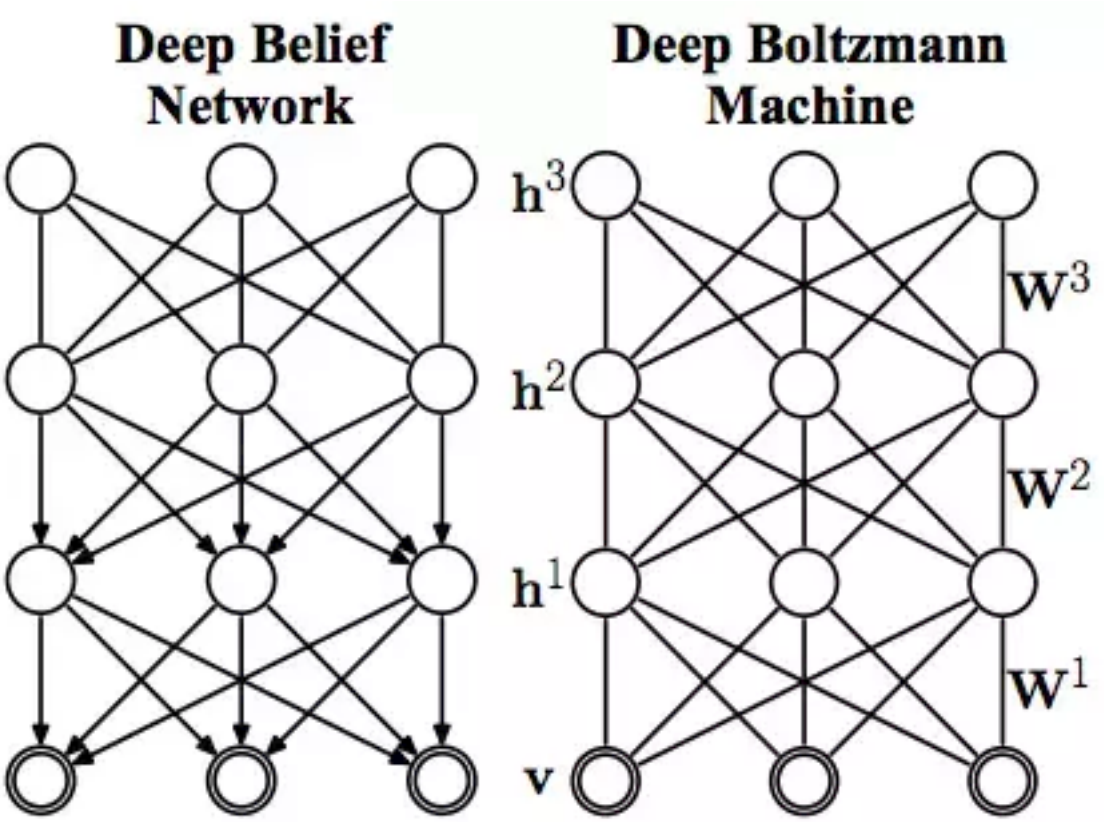
DBN Greedy Training

- 3rd Step:
  - 在已有网络之上，继续叠加新的层，从$Q(h^2|h^1)$中进行抽样

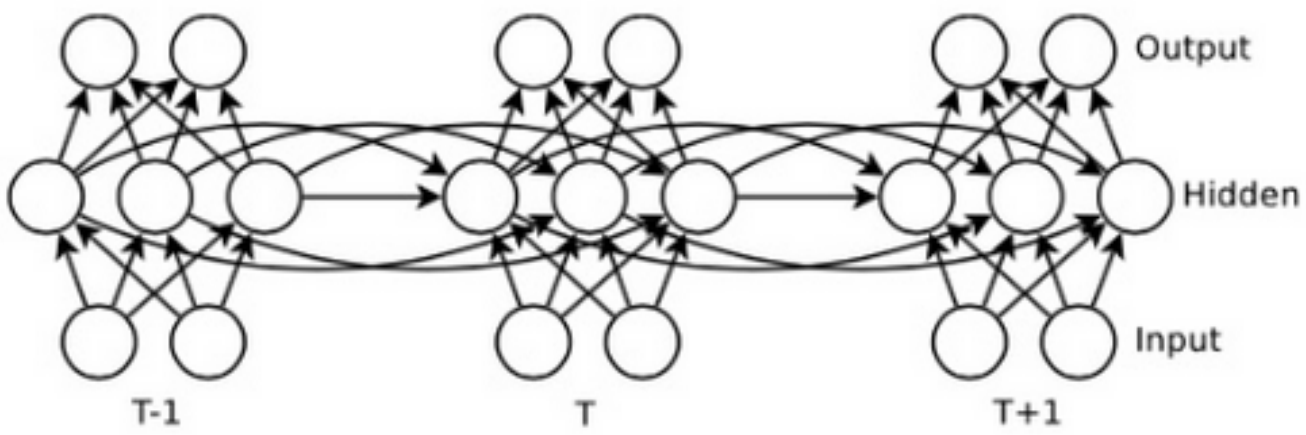


$Q(h^2|h^1)$

$Q(h^1|v)$


Deep Belief Network
Deep Boltzmann Machine
h3
h2
h1
v
W3
W2
W1

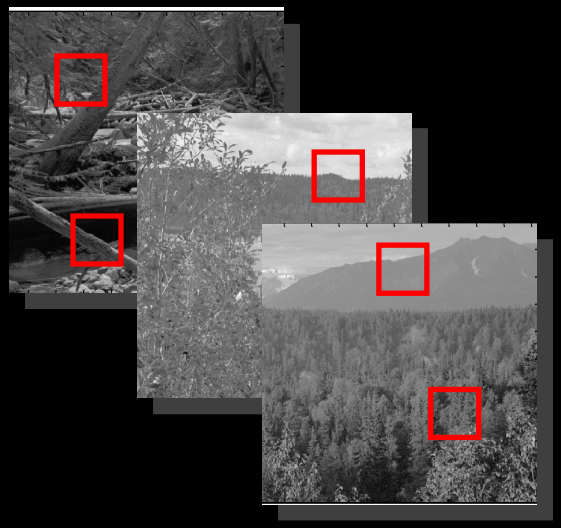
undirected



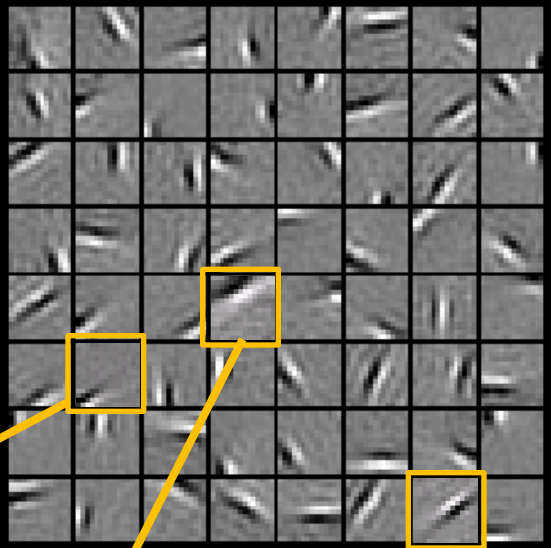
Recurrent Neural Network (RNN)

Sparse coding illustration

Natural Images



Learned bases ($\phi_1, \ldots, \phi_{64}$): "Edges"



Test example

$\approx 0.8 *$ $+ 0.3 *$ $+ 0.5 *$

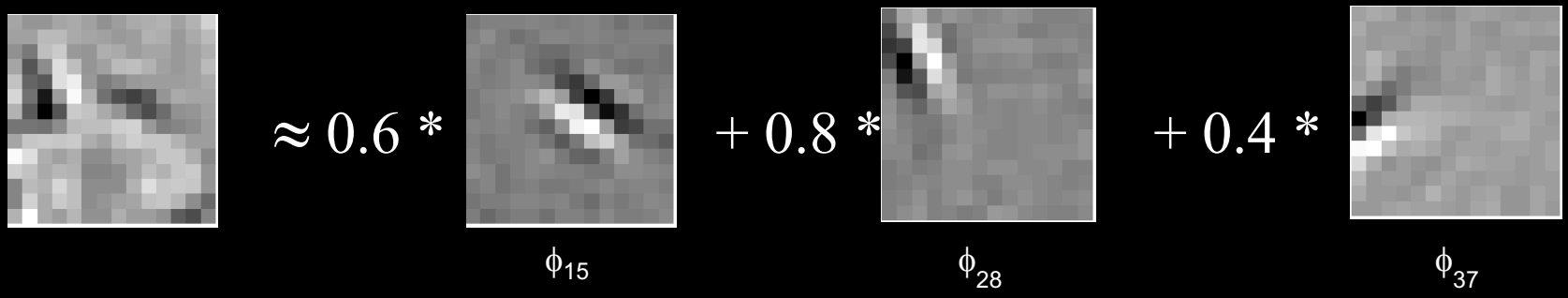
$x \approx 0.8 * \phi_{36} + 0.3 * \phi_{42} + 0.5 * \phi_{63}$

[0, 0, …, 0, **0.8**, 0, …, 0, **0.3**, 0, …, 0, **0.5**, …]
= [a_1, …, a_{64}] (feature representation)

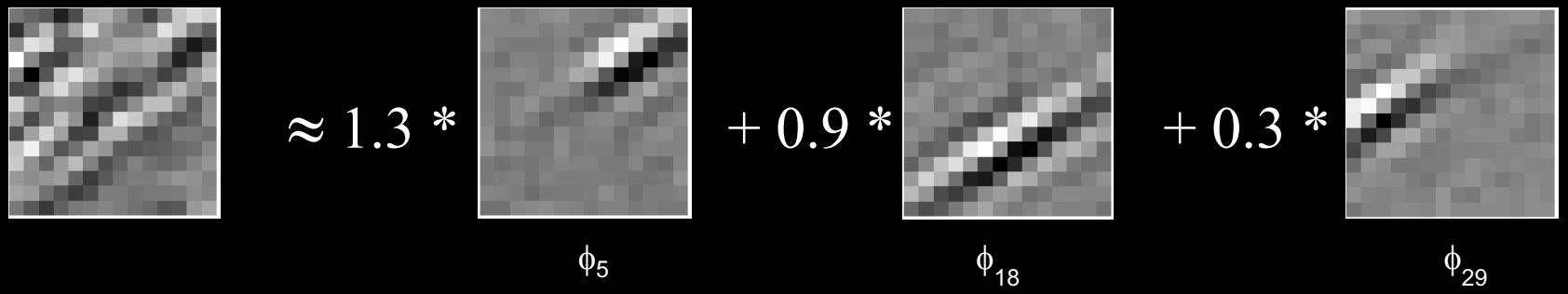
Compact & easily interpretable

Andrew Ng

More examples



Represent as: [0, 0, ..., 0, 0.6, 0, ..., 0, 0.8, 0, ..., 0, 0.4, ...]

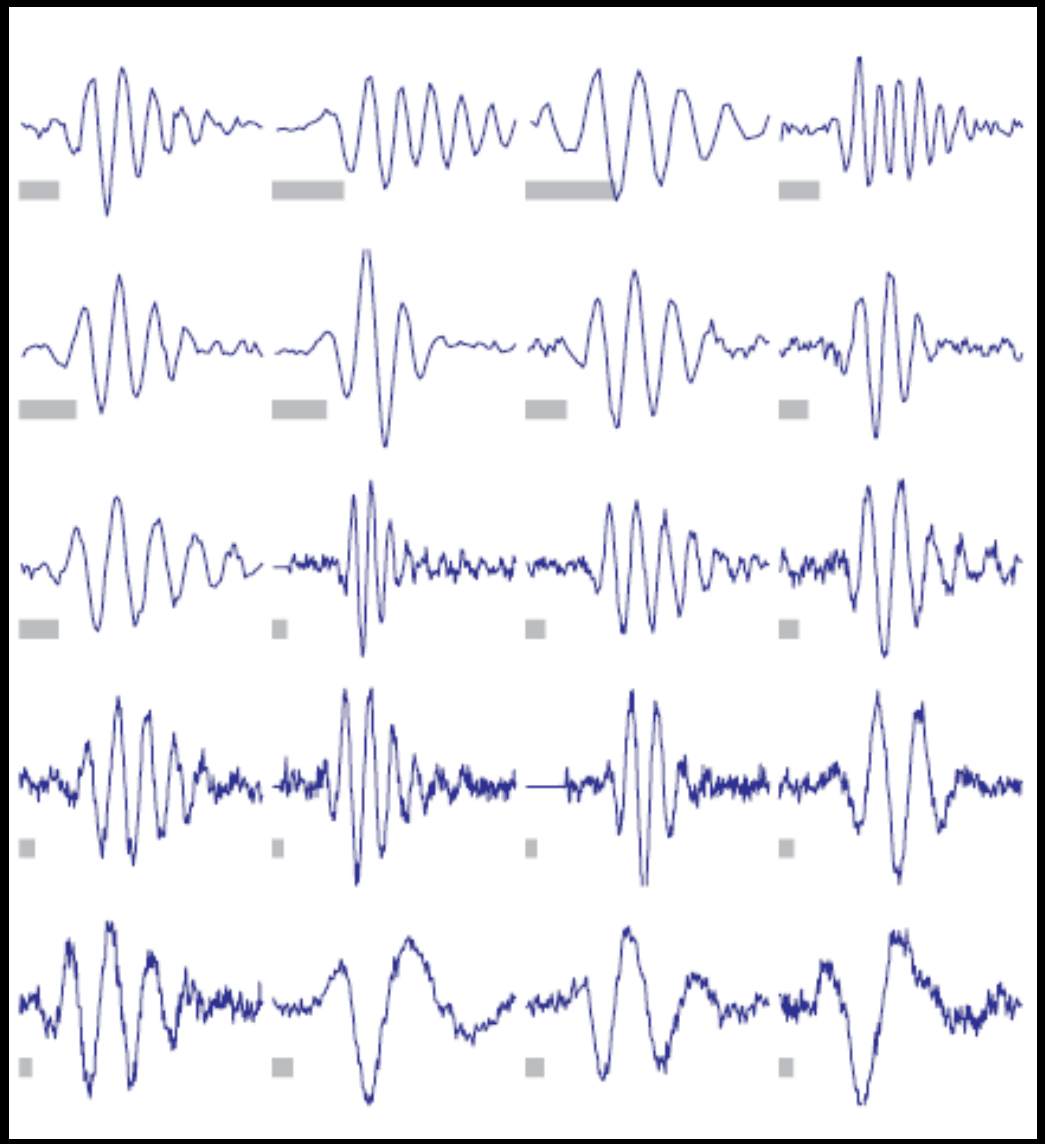


Represent as: [0, 0, ..., 0, 1.3, 0, ..., 0, 0.9, 0, ..., 0, 0.3, ...]

- Method hypothesizes that edge-like patches are the most "basic" elements of a scene, and represents an image in terms of the edges that appear in it.
- Use to obtain a more compact, higher-level representation of the scene than pixels.

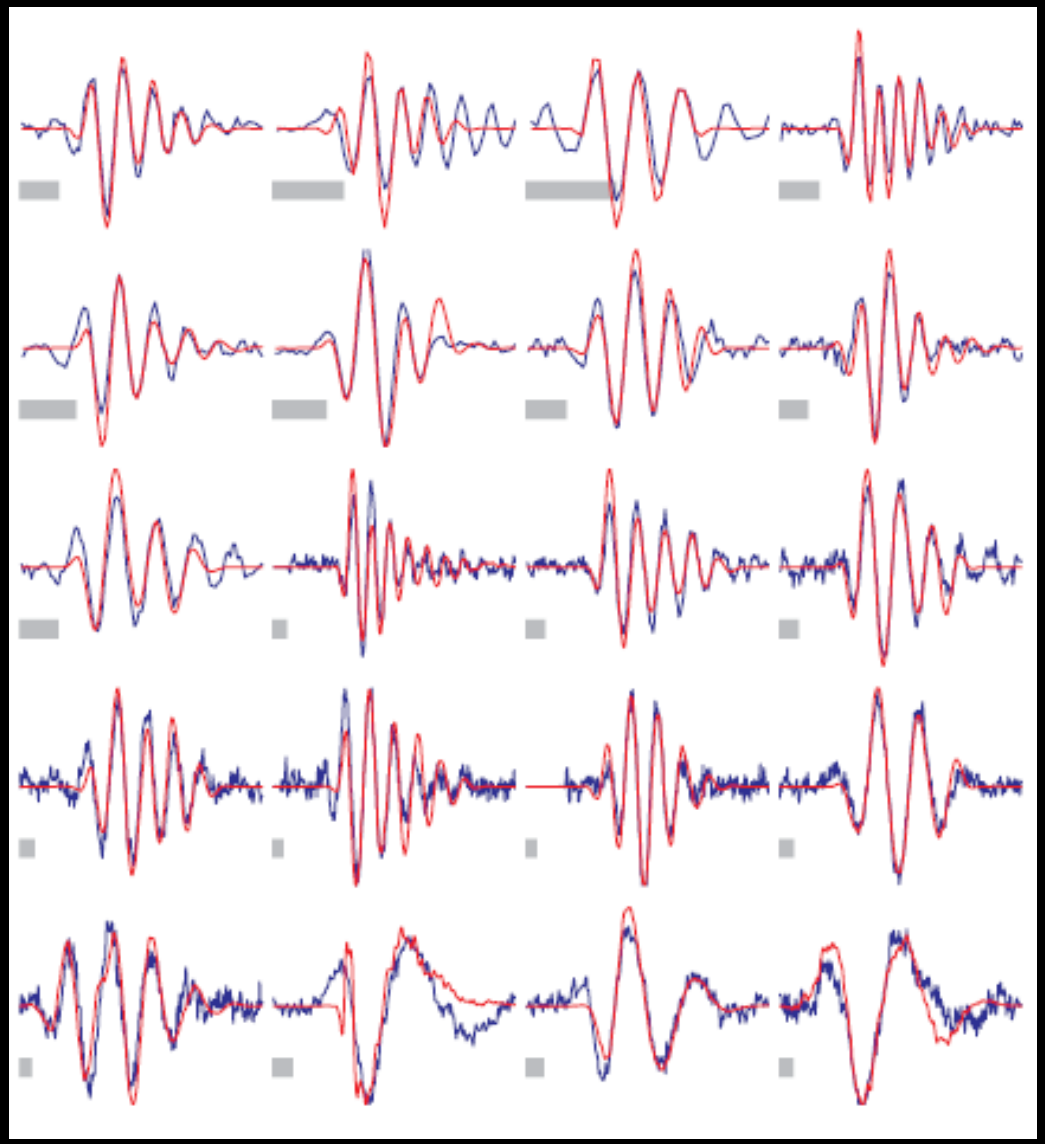
Andrew Ng

Digression: Sparse coding applied to audio



[Evan Smith & Mike Lewicki, 2006]

Digression: Sparse coding applied to audio



[Evan Smith & Mike Lewicki, 2006]

Sparse coding details

Input: Images $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ (each in $\mathbb{R}^{n \times n}$)

$$\min_{a,\phi} \sum_{i=1}^{m} \left(\left\| x^{(i)} - \sum_{j=1}^{k} a_j^{(i)} \phi_j \right\|^2 + \lambda \sum_{j=1}^{k} |a_j^{(i)}| \right)$$

L_1 sparsity term
(causes most $a_j^{(i)}$s to be 0)

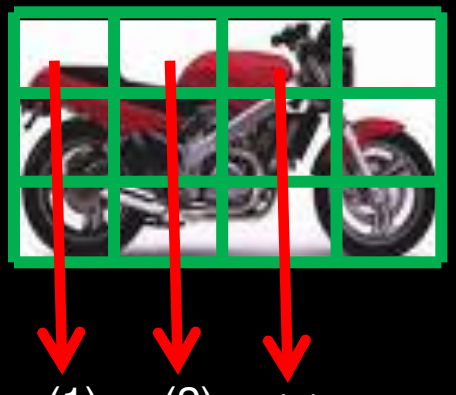
Alternating minimization:
Alternately minimize with respect to ϕ_l's (easy)
and a's (harder).

Putting it together

Feature representation

Learning algorithm

Suppose you've already learned bases $\phi_1, \phi_2, \ldots, \phi_k$. Here's how you represent an image.



$x^{(1)}$ $x^{(2)}$ $x^{(3)}$ …

$a^{(1)}$ $a^{(2)}$ $a^{(3)}$ …

$$\begin{bmatrix} a^{(1)} \\ a^{(2)} \\ \vdots \\ a^{(12)} \end{bmatrix}$$

or $\sum_{i=1}^{12} a^{(i)}$

Learning algorithm

E.g., 73-75% on Caltech 101 (Yang et al., 2009, Boreau et al., 2009)

Sparse coding recap

[0, 0, …, 0, **0.8**, 0, …, 0, **0.3**, 0, …, 0, **0.5**, …]

Much better than pixel representation. But still not competitive with SIFT, etc.

Three ways to make it competitive:

- Combine this with SIFT.
- Advanced versions of sparse coding (LCC).
- Deep learning.

Game Theory Inception

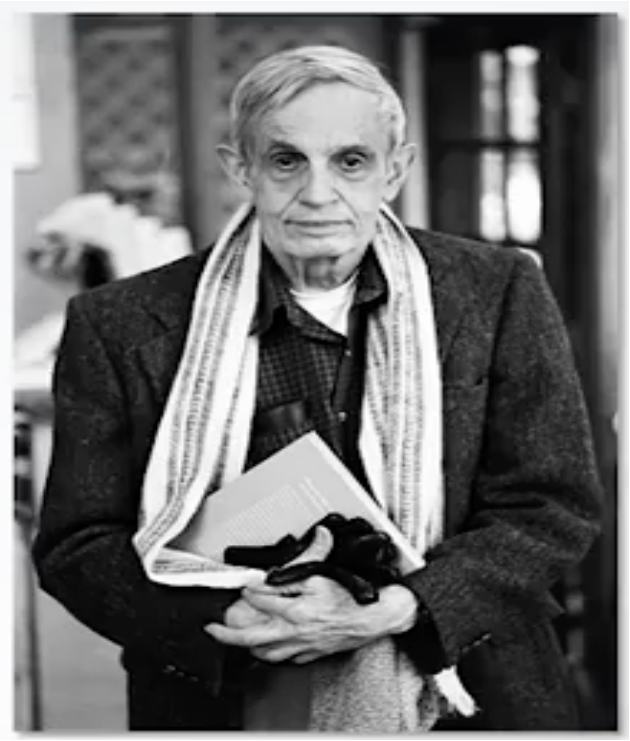
Games Theory was created by Neumann land Morgenstern in 1944, but in 1950, the central problem was still open.

是否存在一个理论(General Solution)能对所有的Social问题进行建模：

Game：

1) Players
2) Strategies
3) Payoffs

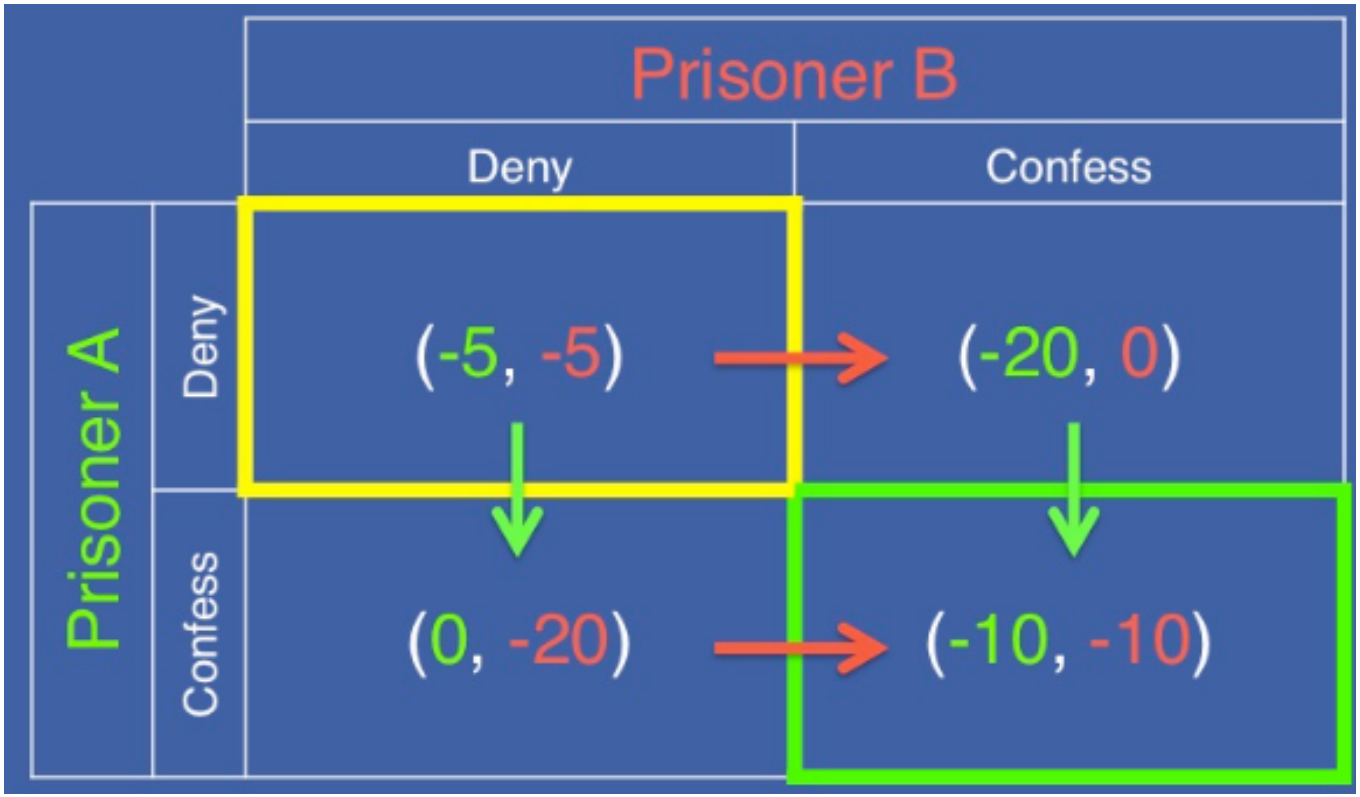
John Nash



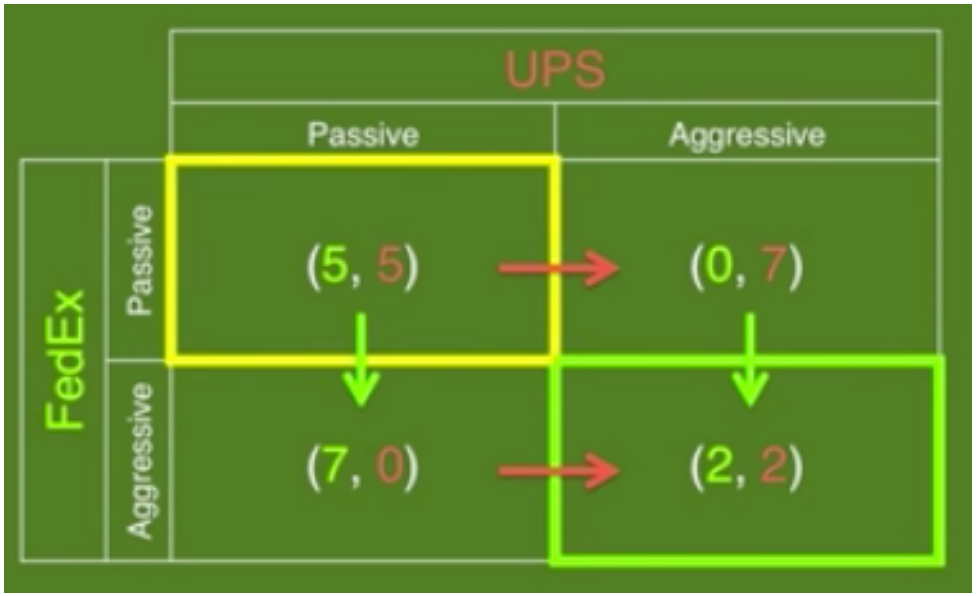
This man is a genius!

1994:Nobel Prize in economics

Prisoner's Dilemma Payoff Matrix

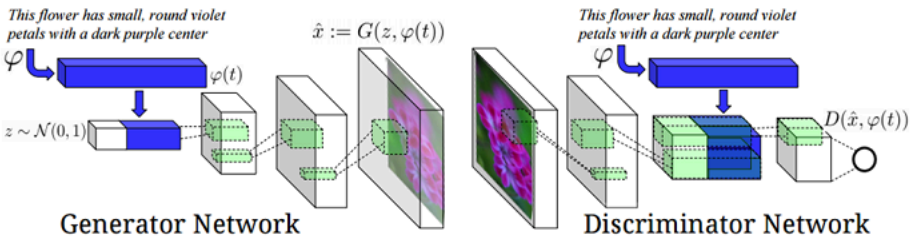


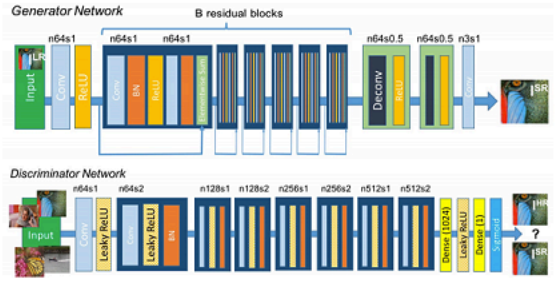
A real world application



- A Nash equilibrium, is a situation in a multiple-player non-cooperative game “where no player has an incentive to deviate from his or her chosen strategy after considering an opponent’s choice.”
- In other words, given another player’s choice, your best choice is obvious.
- If this is true for every player, that combination of choices is a Nash equilibrium: more specifically, a pure-strategy Nash equilibrium.

Generative Adversarial Net





Text-conditional convolutional GAN

Super Resolution

Generative Adversarial Net

```
# Build Generative model ...
nch = 200
g_input = Input(shape=[100])
H = Dense(nch*14*14, init='glorot_normal')(g_input)
H = BatchNormalization(mode=2)(H)
H = Activation('relu')(H)
H = Reshape( [nch, 14, 14] )(H)
H = UpSampling2D(size=(2, 2))(H)
H = Convolution2D(nch/2, 3, 3, border_mode='same', init='glorot_uniform')(H)
H = BatchNormalization(mode=2)(H)
H = Activation('relu')(H)
H = Convolution2D(nch/4, 3, 3, border_mode='same', init='glorot_uniform')(H)
H = BatchNormalization(mode=2)(H)
H = Activation('relu')(H)
H = Convolution2D(1, 1, 1, border_mode='same', init='glorot_uniform')(H)
g_V = Activation('sigmoid')(H)
generator = Model(g_input,g_V)
generator.compile(loss='binary_crossentropy', optimizer=opt)
generator.summary()
```

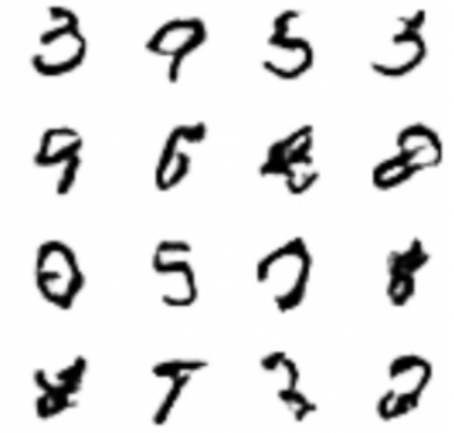
https://oshearesearch.com/index.php/2016/07/01/mnist-generative-adversarial-model-in-keras/

Generative Adversarial Net

```
# Build Discriminative model ...
d_input = Input(shape=shp)
H = Convolution2D(256, 5, 5, subsample=(2, 2), border_mode = 'same', activation='relu')(d_input)
H = LeakyReLU(0.2)(H)
H = Dropout(dropout_rate)(H)
H = Convolution2D(512, 5, 5, subsample=(2, 2), border_mode = 'same', activation='relu')(H)
H = LeakyReLU(0.2)(H)
H = Dropout(dropout_rate)(H)
H = Flatten()(H)
H = Dense(256)(H)
H = LeakyReLU(0.2)(H)
H = Dropout(dropout_rate)(H)
d_V = Dense(2,activation='softmax')(H)
discriminator = Model(d_input,d_V)
```

```
# Freeze weights in the discriminator for stacked training
def make_trainable(net, val):
    net.trainable = val
    for l in net.layers:
        l.trainable = val
make_trainable(discriminator, False)

# Build stacked GAN model
gan_input = Input(shape=[100])
H = generator(gan_input)
gan_V = discriminator(H)
GAN = Model(gan_input, gan_V)
GAN.compile(loss='categorical_crossentropy', optimizer=opt)
GAN.summary()
```



https://oshearesearch.com/index.php/2016/07/01/mnist-generative-adversarial-model-in-keras/

Assignment

- Using Keras to build Auto-encoder & RBM
- https://oshearesearch.com/index.php/2016/07/01/mnist-generative-adversarial-model-in-keras/

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