AI and Matrix
人工智能與矩陣

So, Hing Cheung (蘇慶祥)

http://www.ee.cityu.edu.hk/~hcso

Department of Electronic Engineering
City University of Hong Kong
Outline

- What is AI
- Why AI is Important
- How AI Works
- What is Low-Rank Matrix
- Low-Rank Matrix Approximation
- Low-Rank Matrix Recovery
- Application Examples
- References
What is AI (人工智能)

The term artificial intelligence (AI) was coined in 1956 at Dartmouth Artificial Intelligence Conference.

The purpose of AI is to let computers think like humans.

First AI wave boomed during 1950s-1960s but entered its winter in 1970s because only toy problems could be handled.

Second AI wave boomed during 1980s, which emerged from expert systems research. However, knowledge from human experts and their conversion to programmable rules are needed. Its applicability was thus still limited as creation and maintenance of expert systems require demanding works. AI entered its winter again in around 1995.

Now we are at the third AI wave boom, which is mainly driven by big data, high computing power, and deep learning.
Figure extracted from [1]
Checkers (西洋跳棋) → Chess (國際象棋) → Go (圍棋)

Figure extracted from [1]
Machine Learning (機器學習): A branch of AI that specializes in how the computer simulates or realizes human learning behavior, including acquire new knowledge or skills to continually improve its performance.

Deep Learning (深度學習): An important machine learning approach based on neural network and massive training data.

Data Science (數據科學): This subject combines different fields including statistics, applied mathematics, data visualization, pattern recognition, machine learning, high-performance computing, in order to interpret data (massive and complex) and/or extract meaningful information from data for decision making.

Big Data (大數據): Any voluminous amount of structured, semi-structured and unstructured data that has the potential to be mined for information. Traditional data processing application software is usually inadequate to deal with them.
Why AI is Important

We are already surrounded by numerous AI products:

- Intelligent personal assistants such as Apple Siri, Microsoft 小冰 and Baidu DuerOS.
- News summarization, recommendation and writing such as Yahoo Summly, 今日头条 and Automated Insights.
- Computer vision applications such as face recognition, Google Photos and Amazon Go.
- AI powered art apps such as Prisma and 美圖秀秀.
- Search engines such as Facebook FAIR and Google.
- Automatic translation applications such as Google Translate and ili.
- Autonomous cars such as Tesla Autopilot and Waymo self-driving cars.
- Robots such as DHL unmanned aerial vehicles and Wonder Workshop Dash.
According to *Future of Jobs Report* [2] published by World Economic Forum in 2016, The Fourth Industrial Revolution, which includes developments in AI and machine-learning, robotics, nanotechnology, 3-D printing, and genetics and biotechnology, will cause widespread disruption not only to business models but also to labour markets over the next five years, with enormous change predicted in the skill sets needed to thrive in the new landscape.

On March 2017, Premier Li Keqiang (國務院總理李克強) first mentioned in Government Work Report (政府工作報告) that China will accelerate R&D and commercialization of AI. On July 2017, China has laid out a development plan to become the world leader in AI by 2030, aiming to surpass its rivals technologically and build a domestic industry worth almost $150 billion.
On Jan. 2018, Hang Seng Bank has launched two AI virtual assistants with human-like conversations: HARO handles general inquiries about the bank’s products and services, such as calculating mortgage payment; and DORI searches dining discounts and makes recommendations based on customer preferences in credit card services. HSBC has also launched a similar AI Chabot called Amy.

Uniqlo is using AI to help understand what the customer wants and reduce excessive inventory. (Nikkei Asian Review, 26 Aug. 2017)

德勤明言渴求 ABCDR 人才，即 AI（人工智能）、Blockchain（區塊鏈）、Cloud Computing（雲端計算）、Data Analytics（數據分析）及 Robotics（機械人）(香港經濟日報, 16 Mar. 2018)
How AI Works

Among numerous AI technologies, the hottest one is deep learning.

Briefly speaking, deep learning refers to a neural network (NN, 神經網路) with many layers and massive training data.

NN is a computing system inspired by biological NNs that constitute animal brains.

The weights between nodes are adjusted based on training data until the outputs satisfy certain requirements.
Breakthrough in **speech recognition** (around 95% accuracy now which is threshold for human).

---

Figure extracted from [1]
Breakthrough in **image recognition** in ImageNet Large Scale Visual Recognition Challenge [3].

Figure extracted from [1]
In ImageNet [4], there are 14,197,122 images, indicating massive training data are needed.

Note that giant companies such as Google and Facebook are very advantageous in obtaining massive data via search and social network interactions.

Microsoft ResNet has 152 layers in the NN, indicating huge computations are involved.

To address the complexity issues, parallel/distributed processing, GPUs and cloud computing are employed.

To handle the big data issues, information exchange and storage techniques particularly among networks, are required.
What is Low-Rank Matrix?

A scalar (標量) is a number or symbol, e.g., “3.14”, “x”, “β”.

A vector (向量) is 1-D array of scalars, arranged in a row or column, e.g.,

\[
\begin{bmatrix}
3 & -1 & 4 & -6 & 2
\end{bmatrix} \in \mathbb{R}^{1 \times 5}
\]

\[
\begin{bmatrix}
c \\
a \\
c \\
b
\end{bmatrix} \in \mathbb{R}^{4 \times 1}
\]

A matrix (矩陣) is 2-D array of scalars, arranged in rows and columns, e.g.,

\[
\begin{bmatrix}
1 & 2 & 3 & 4 \\
3 & 5 & 7 & 11 \\
4 & 3 & 2 & 1
\end{bmatrix} \in \mathbb{R}^{3 \times 4}
\]
Many real-world data can be represented as vectors and matrices. For example, the marks and grades of Chinese Language (70, B), English Language (60, C), Mathematics (80, A), and Liberal Studies (70, B) for student Alice may be represented as:

\[
\begin{bmatrix}
70 & 60 & 80 & 70
\end{bmatrix} \in \mathbb{R}^{1 \times 4} \quad \text{and} \quad \begin{bmatrix}
B & C & A & B
\end{bmatrix} \in \mathbb{R}^{1 \times 4}
\]

For a group of students consisting of Alice, Bob and Carol, the matrix representations can be:

\[
\begin{bmatrix}
70 & 60 & 80 & 70 \\
80 & 80 & 55 & 85 \\
68 & 69 & 71 & 72
\end{bmatrix} \in \mathbb{R}^{3 \times 4} \quad \text{and} \quad \begin{bmatrix}
B & C & A & B \\
A & A & C & A \\
B & B & B & B
\end{bmatrix} \in \mathbb{R}^{3 \times 4}
\]

Excel files can also be viewed as matrices.
A matrix $X \in \mathbb{R}^{m \times n}$ has full rank if its rank is $r = \min(m, n)$. This means that all the columns (or rows) are linearly independent.

$X \in \mathbb{R}^{m \times n}$ has low rank if its rank is $r \ll \min(m, n)$. This means that many columns (or rows) are linearly dependent, e.g.,

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 2 & 1 & 2 & 1 & 2 \end{bmatrix} \in \mathbb{R}^{5 \times 5} \Rightarrow r = \text{rank}(X) = 2$$

Row #1: $[1 \ 1 \ 1 \ 1 \ 1] = [0 \ 1 \ 0 \ 1 \ 0] + [1 \ 0 \ 1 \ 0 \ 1]$

Row #2: $[1 \ 2 \ 1 \ 2 \ 1] = 2 \times [0 \ 1 \ 0 \ 1 \ 0] + [1 \ 0 \ 1 \ 0 \ 1]$

Row #5: $[2 \ 1 \ 2 \ 1 \ 2] = [0 \ 1 \ 0 \ 1 \ 0] + 2 \times [1 \ 0 \ 1 \ 0 \ 1]$
When $X$ is of low rank $r$, it can be factorized as a product of two full-rank matrices with rank $r$, e.g.,

$$X = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 2 & 1 & 2 & 1 \\
0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
2 & 1 & 2 & 1 & 2
\end{bmatrix} = UV$$

where

$$U = \begin{bmatrix}
1 & 1 \\
1 & 2 \\
0 & 1 \\
1 & 0 \\
2 & 1
\end{bmatrix} \in \mathbb{R}^{5 \times 2} \quad V = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix} \in \mathbb{R}^{2 \times 5}$$
Real-world matrix data include image, e.g., for gray-scale image using 8-bit representation: 0 (black) to 255 (white):

Figures extracted from [5]

An image will be of low rank if it has specific structure, e.g., regular pattern or texture.
Video can also be expressed as a matrix by converting each frame (matrix) as a vector, e.g.,

Frame #1

\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{bmatrix}
\]

Frame #2

\[
\begin{bmatrix}
7 & 8 & 9 \\
10 & 11 & 12
\end{bmatrix}
\]

Frame #3

\[
\begin{bmatrix}
13 & 14 & 15 \\
16 & 17 & 18
\end{bmatrix}
\]

Frame #4

\[
\begin{bmatrix}
19 & 20 & 21 \\
22 & 23 & 24
\end{bmatrix}
\]

The background component in the video is of low rank.
Low-Rank Matrix Approximation

Given an observation matrix $X \in \mathbb{R}^{m \times n}$ which can be expressed as:

$$X = L + Q$$

where $L$ is the low-rank component with $\text{rank}(L) = r \ll \min(m, n)$ and $Q$ is the perturbation matrix.

In image processing, $L$ may be the texture component in the image.

In video processing, $L$ corresponds to the background component in the video frames.

The task of low-rank matrix approximation is to find $L$. 
As $L$ has low rank, we can write [6]-[7]:

$$L = UV, \quad U \in \mathbb{R}^{m \times r}, \quad V \in \mathbb{R}^{r \times n}$$

A conventional technique is to find $U$ and $V$ via:

$$\min_{U,V} \|UV - X\|_2^2, \quad \|X\|_2^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} |x_{i,j}|^2$$

which is based on the least squares (LS) criterion or $\ell_2$-norm. However, it is not robust to outliers such as the non-texture components in image and moving objects in video.

We utilize least absolute deviation (LAD) criterion or $\ell_1$-norm:

$$\min_{U,V} \|UV - X\|_1, \quad \|X\|_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} |x_{i,j}|$$
Comparison between LS and LAD in linear regression
Low-Rank Matrix Recovery

The aim is to recover a low-rank matrix given only a subset of its possibly noisy entries, e.g.,

\[
\begin{pmatrix}
1 & ? & ? & 4 & ? \\
? & 2 & 5 & ? & ? \\
? & ? & 4 & 5 & ? \\
5 & ? & ? & ? & 4
\end{pmatrix}
\]

It is a more challenging problem than matrix approximation because it requires finding the low-rank component \( L \) even when there are missing entries in the observation matrix.
A real-world example is to restore a low-rank image with missing/distorted components:

The distorted pixels are covered by “ICCV 2009 LRTC” [8]
Application Examples

- Texture Extraction

We apply our robust algorithm to an image of a chessboard with $377 \times 370$ pixels using the low-rank information of $r = 2$. 
Video Background Extraction

Considering two open datasets [9], we select the first 200 frames where each frame has dimensions of $240 \times 320$, corresponding to 76,800 pixels. Thus, the observed matrix constructed from each video is $X \in \mathbb{R}^{76800 \times 200}$ where $m = 76,800$ and $n = 200$.

That is, we convert each frame of a video as a column of a matrix, the resultant matrix due to the background is of low-rank due to the correlation between frames.

Foreground objects such as moving cars or walking pedestrians, generally occupy only a fraction of the image pixels and hence can be treated as sparse outliers.

Background is $L$ and foreground is $Q$ and we set the rank as $r = 1$. 
Image Inpainting

We consider the image where the missing data correspond to “ICCV”, “2009”, and “LRTC” [8] and its dimensions are $247 \times 259$.

To increase the research challenge, salt-and-pepper noise is added.

The matrix rank is set to $r = 6$.

The CPU times of the singular value projection (SVP), $\ell_2$-regression, $\ell_1$-regression and $\ell_1$-ADMM are 20.3s, 0.4s, 7.8s and 4.9s, respectively.
Recommender System

Netflix Prize [10], whose goal was to accurately predict user preferences for films based on all user previous ratings.

The ratings are 1, 2, 3, 4 and 5 where 5 means “strongly recommend”.

A database of over 100 million movie ratings made by 480,189 users in 17,770 films is provided, which corresponds to the task of completing a matrix with around 99% missing entries. Note the users or films are not provided.

As the data do not contain outliers, the $\ell_2$-norm approach can produce accurate prediction results. However, due to the large data sizes, cloud computing and distributed processing are needed [11].
<table>
<thead>
<tr>
<th></th>
<th>The Godfather</th>
<th>Beauty and the Beast</th>
<th>Matrix</th>
<th>A Beautiful Mind</th>
<th>Whiplash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carol</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dave</td>
<td>5</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Recommend Alice to watch “Beauty and the Beast”?
References

[1] 李開復, 王詠剛, 人工智慧來了, 天下文化, 2017