本文衍生自維基百科

十分鐘小論文

最佳化問題的公理化方法

陳鍾誠

2016 年 11 月 20 日

Hi，大家好！

自從十分鐘系列開始以來

- 我們介紹了很多程式和數學方面的知識！
- 甚至有些是科學史或社會領域的。

這些投影片

- 基本上都是介紹已有的知識

而且

- 這些知識幾乎都不是我創造的！

而是

- 全世界的學者、程式人與數學家，一同創造出來的！

所以十分鐘系列

- 到目前為止，都還算是《科普著作》而非《學術研究作品》

但是最近

- 我決定開始做點研究！

把幾年來的想法與研究

- 寫成類似《十分鐘系列》的小論文！

這一系列的投影片

- 就姑且稱為《十分鐘小論文》好了！

這些十分鐘小論文

- 基本上都不會被拿去投稿甚麼期刊或研討會！

因為我

- 反正不想升等，又討厭那些冗長的審稿程序
- 也不希望文章投稿後變成期刊公司的私有財產
- 又不喜歡寫成論文的那種格式
- 所以這些十分鐘小論文就可以寫得很隨興了！

這些小論文

- 我會寫成《像十分鐘系列》一樣的投影片
- 放到像 Slideshare 或 SpeakerDeck 這種投影片網站上，讓大家都可以輕鬆的閱讀！

如果你想要引用這些《像論文的投影片》

- 那就寫上《標題、作者、日期、還有出版網址》就行了。

像是這樣

1. 陳鍾誠，《最佳化問題的公理化方法》，
十分鐘小論文，2016/11/20 日，
網址 :http://www.slideshare.net/ccckmit/ss-69310214

我們今天的小論文主題是

這樣的小論文

- 或許不會有甚麼太高的學術價值
- 但是至少不需要造假，想寫甚麼就寫甚麼！

現在

- 就讓我們開始今天的

  十分鐘小論文吧！

先聲明

- 我們的小論文，沒有一定的法則！
- 有時候有實驗驗證，有時候只是想法而已。

今天這篇

- 只是一種看法，沒有實驗檢證！

我們今天的小論文主題是

十分鐘小論文

最佳化問題的公理化方法

陳鍾誠

2016 年 11 月 20 日

最佳化

- 英文是 Optimization

電腦中的許多問題

- 最後都可被化為一個最佳化問題！

最小擴展樹問題

- 是要找一個樹狀結構，使得所有分枝的權重總和最小！

最短路徑問題

- 則是要找到一條從來起點到終點之間最短的路徑！

最大網路流問題

- 則是要找到符合限制條件下，流量最大的配置方法！

以上這些問題

- 都很明顯的是最佳化問題

但是、很多資訊科學上的程式問題

- 初看並不像是最佳化問題
- 但是仔細想會發現，幾乎都可以化為最佳化問題！

像是排序問題

- 初看並不像最佳化問題！

但是仔細思考後

- 會發現排序問題就是要對一個序列
  - $\{a_1, a_2, \cdots, a_n\}$
- 找到一個置換 Permutation $\{p1, \cdots, pn\}$
  - $\{a_{p1}, a_{p2}, \cdots, a_{pn}\}$
- 使得 $\Sigma\ \mathrm{sign}(a_{pi} - a_{pi+1})$ 最小

所以排序問題

- 也是一個最佳化問題

而大部分的演算法問題

- 則是在尋找一個《速度最快的問題解決算法》
- 只要給定問題（q, input），你能找出一個函數 f，使得 f(input)=>output 且 output 符合 q 的限制要求，這樣就解決了該演算法的問題
- 在所有這些解法中，我們希望找到執行速度最快的那個（這是演算法優劣的衡量方式）

人工智慧問題

- 也幾乎都是優化問題！
- 而且大部分都在尋找《錯誤率最低的解》

像是影像辨識

- 還有語音辨識等等《模式識別》的問題
- 通常是在尋找《辨識錯誤率最低》的解！

自然語言的機器翻譯

- 則是在尋找《翻譯品質最高的方法》

所以最佳化問題

- 對資訊科學領域是非常重要的！

但問題是

- 資訊科學領域的最佳化問題

並不像數學領域的問題那樣

- 常常有非常系統性的解決方法
- 理論基礎通常不夠穩固！

因此在本文中

- 我們試圖引入《數學的公理化方法》
- 讓電腦的最佳化問題有個穩固的理論基礎！

數學的公理系統

- 是所有數學理論的基礎！
- 透過公理與推論法則，可以推論出《定理》！

只要公理沒有矛盾

- 而且推論法則符合邏輯
- 那麼推論出來的定理就

  和公理具有一樣穩固的基礎！

數學的歷史

- 通常會追溯到希臘時代

希臘數學的集大成者

- 應該非《歐幾里得》莫屬了！

歐幾里得

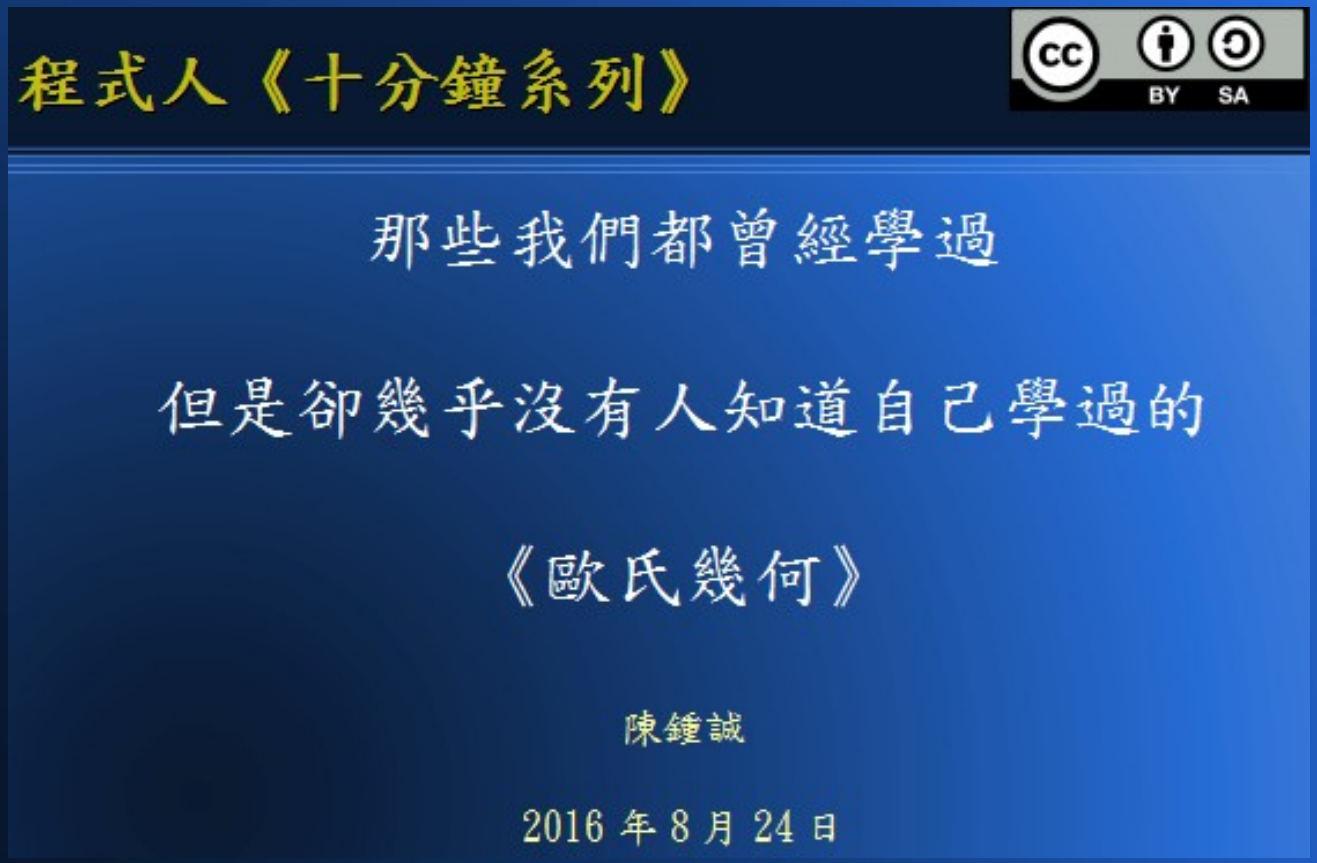
- 在《幾何原本》中給出了用《公理系統》建構數學體系的嚴格推理方法！
- 為數學奠定了一個非常穩固的基礎。

歐幾里得在幾何原本中

- 透過 23 個定義、五條公理與五條公設
- 證明了一條又一條的定理！

關於這個故事

- 我們曾經在下列的十分鐘系列當中探討過！



後來的數學家改變了《平行公理》

- 就發展出了另一類幾何學
- 稱為非歐幾何學

因此只要修改了公理系統

- 就會對理論的適用範圍，有很大的影響
- 很可能會產生非常不同的數學理論

對於最佳化問題

- 如果我們修改了《公理體系》
- 那麼也可能會造成很大的影響！

舉例而言

- 學過《幾何學》的人都知道
- 通過兩點可以決定一直線

但是如果我們把

- 《直線》的條件給放寬，可以容納一元二次多項式《曲線》的話
- 那麼就可以用三點決定一個二次曲線！

如果再把條件放寬到 n 次多項式

- 那麼就可以用 n+1 點決定一個 n 次多項式曲線！

在這個過程中

- 二次曲線涵蓋 1 次的直線
- 三次曲線涵蓋 1, 2 次的曲線
- …
- n+1 次曲線涵蓋 n 次以下的曲線。

所以

- 當我們把《可選取世界放寬之後》
- 就有機會做出更精確的逼近！

不過

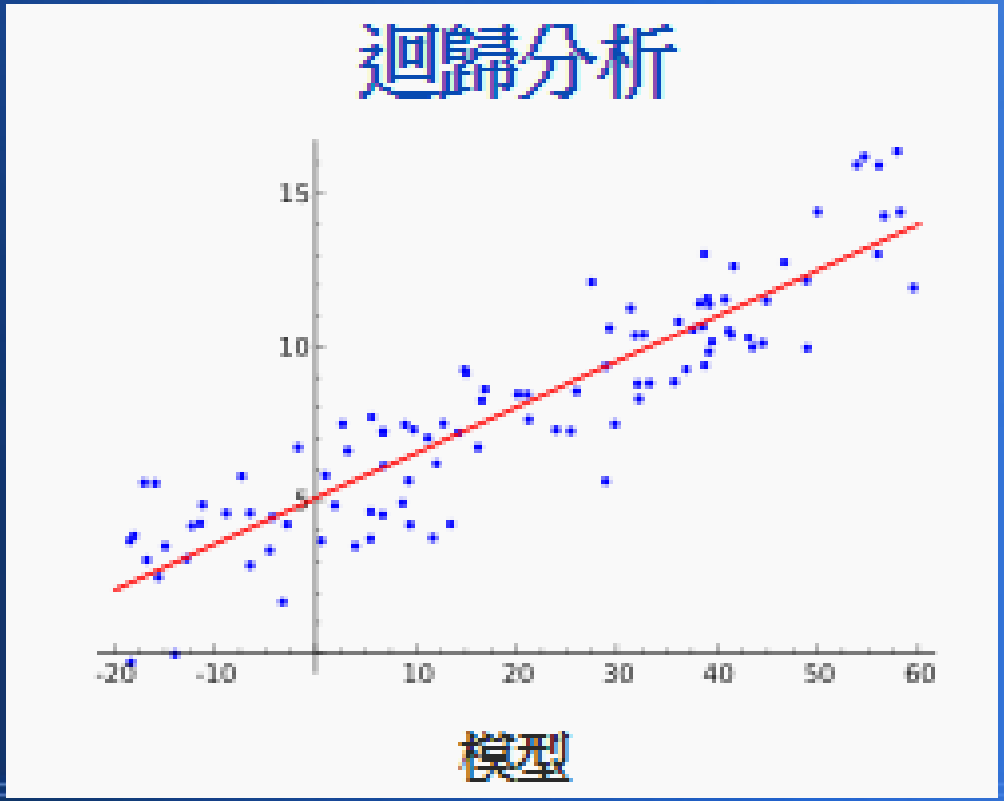
- 上述用 n+1 點決定一個 n 次曲線的問題
- 我們找到的總是精確符合條件的曲線
- 而且沒有考慮《任何點有誤差》情況的容錯機制！

對於有誤差的情況

- 數學上有個《最小平方法》可用

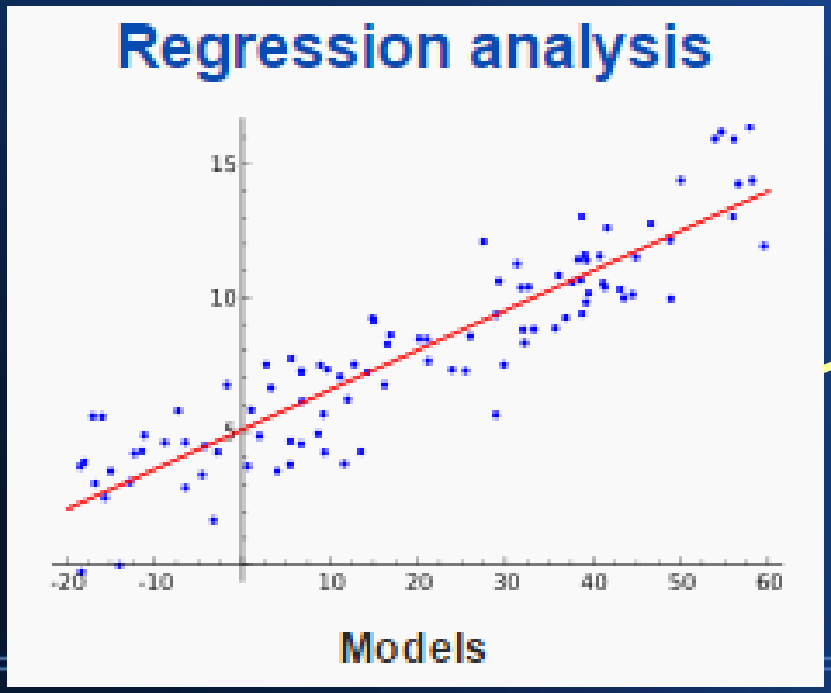
線性版本的最小平方法

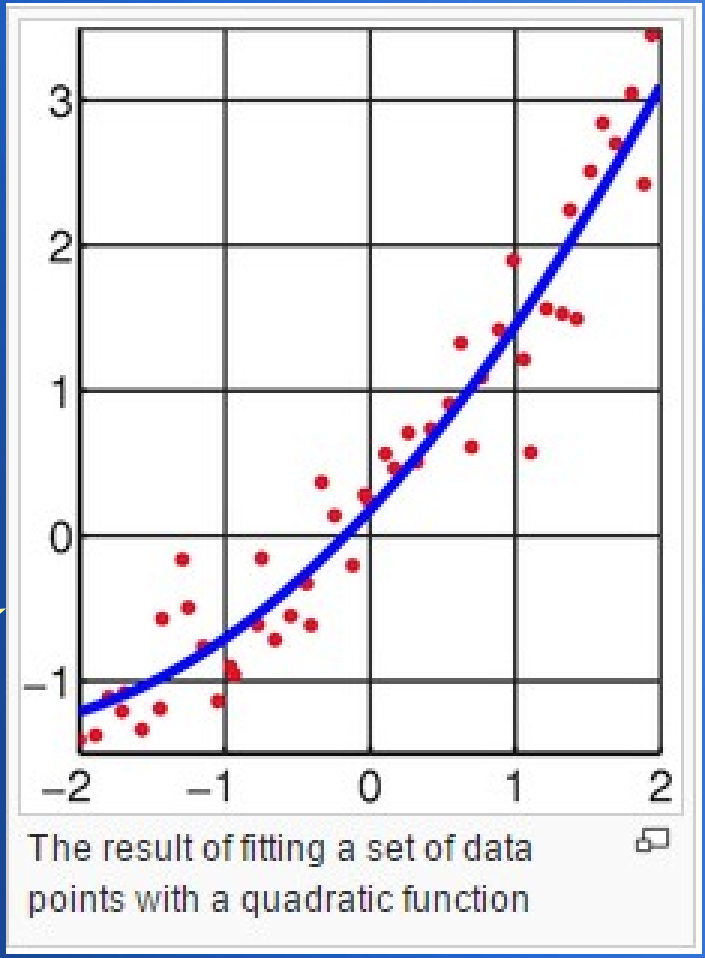
- 可以找出《有誤差資料》中平方差最小的直線。
- 這是一個統計上回歸分析的常用方法。



但是如果我們

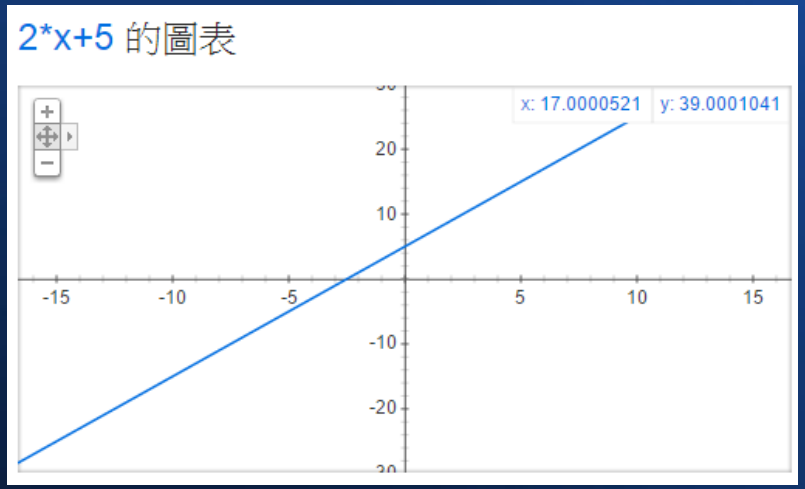
- 把直線放寬到二次曲線
- 那麼就可以涵蓋任何二次多項式

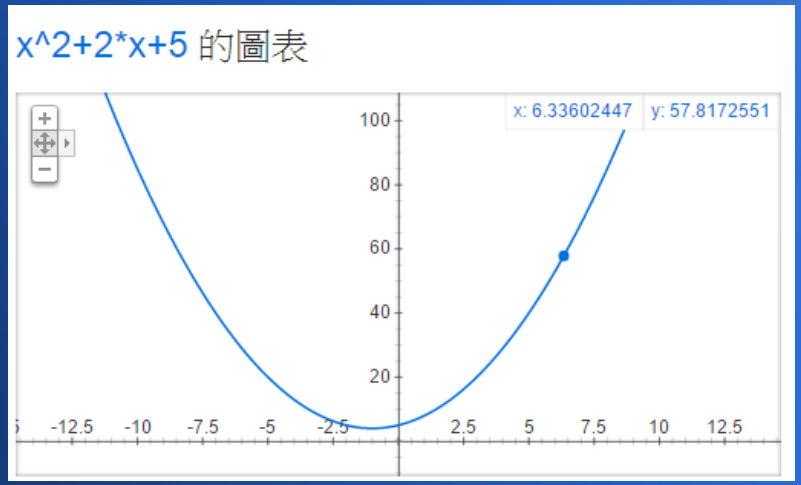


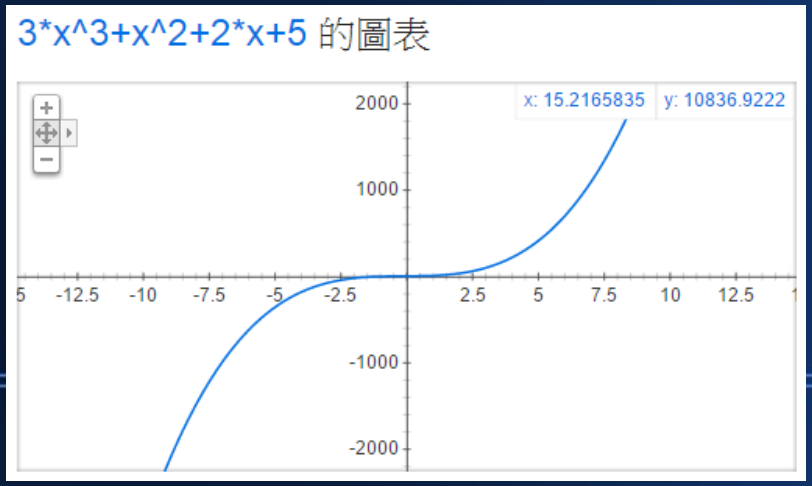


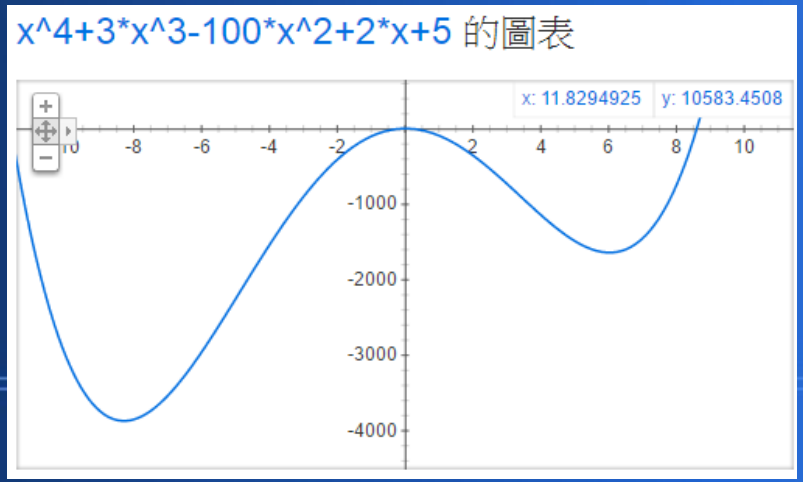
如果再放寬到所有多項式

- 那麼適用的範圍就會更廣！



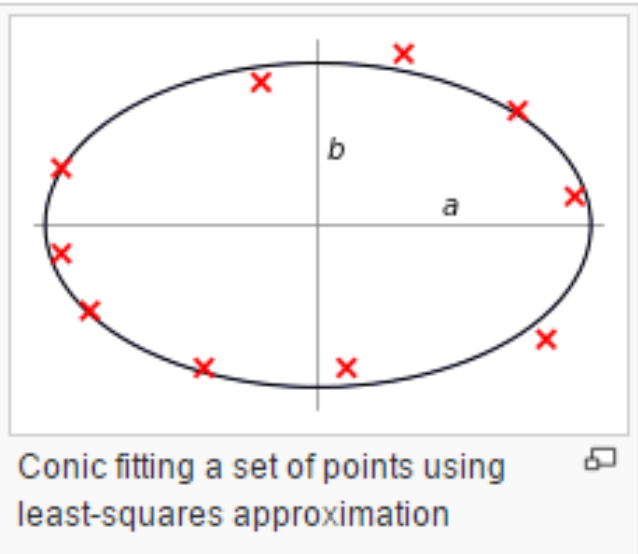






假如再放寬曲線範圍

- 把《圓、橢圓、三角函數》等等圖形加進來



Conic fitting a set of points using least-squares approximation

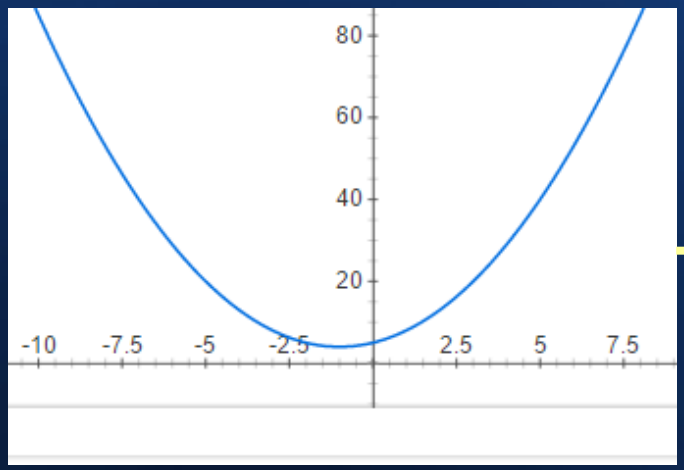
- 那麼就可以做更精確的函數逼近！

如果我們把這些

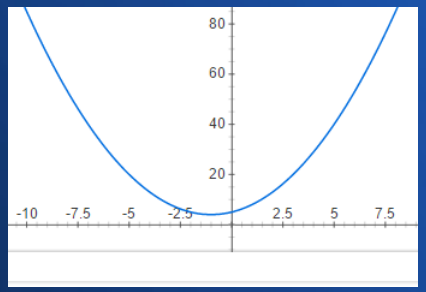
- 《可選擇的曲線集合》，當作是《公理系統的一部分》

然後把《平移、縮放、旋轉》等運算

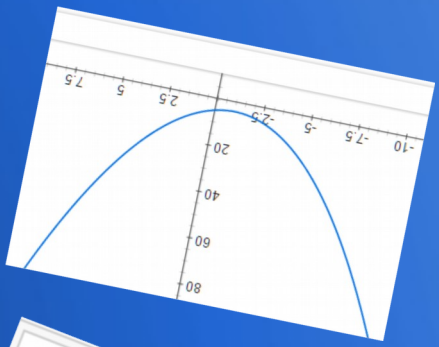
- 當作是《推理法則》，加入到系統當中，
  那麼整個《可用逼近函數》的範圍就會更大！

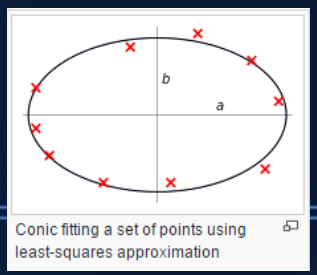


縮小

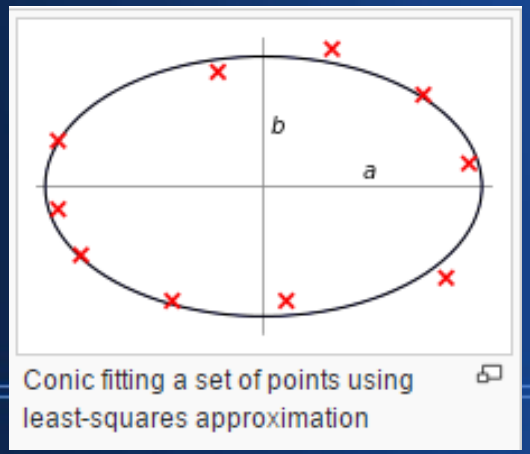


旋轉

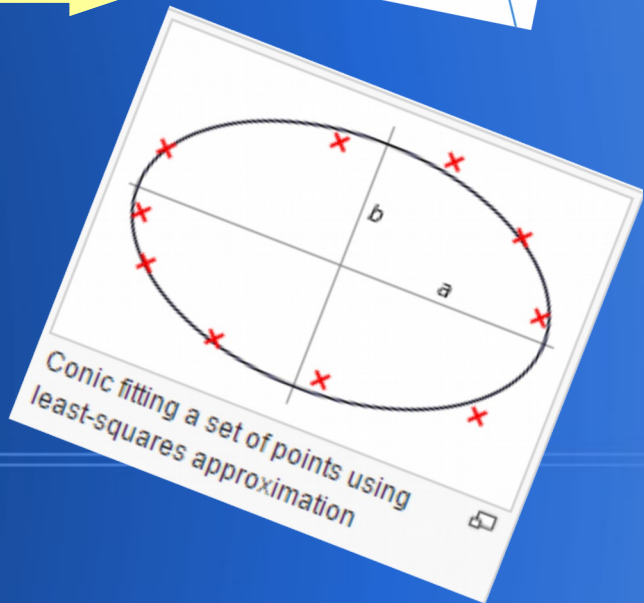




放大



旋轉



如果進一步

- 放寬到所有可微分函數，而且還有《平移、旋轉、縮放》運算可用！
- 那麼涵蓋的函數範圍就會非常廣了！

當然

- 如果繼續放寬到所有函數，不管是連續或不連續，那麼有機會逼近得更好！
- 但是這樣的逼近函數可能會更難找，就算找到了也可能不具普遍代表性！

所以

- 我們通常會要求使用《可微分的函數來逼近》，這樣就會有《微積分》工具可用，也可以利用《泰勒展開式》等數學工具來逼近！

像是傅立葉級數

- 就是利用

  各種頻率與震幅

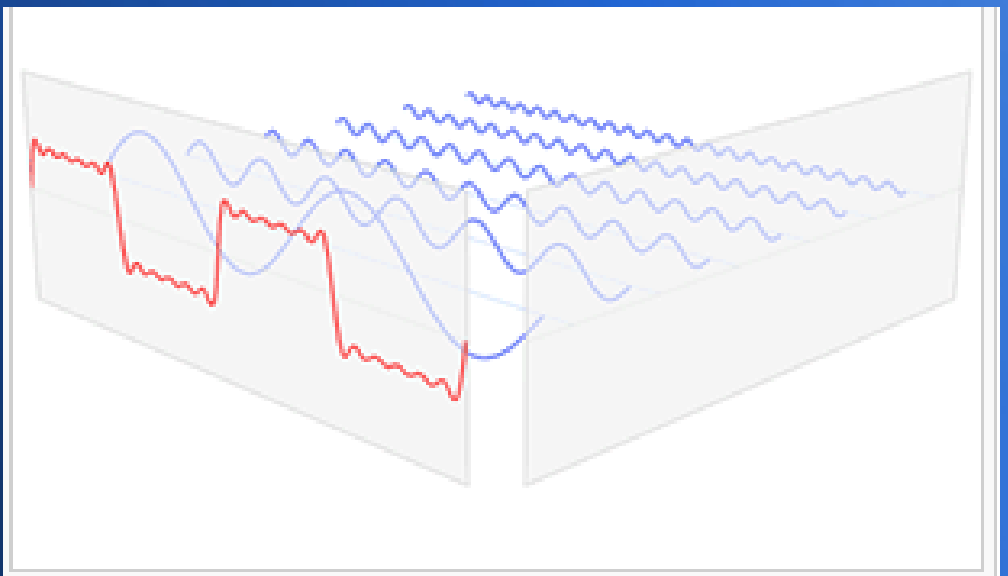
  的三角函數，

  來逼近週期函數 f(x)

  的方法！

$$a_n \cos(nx) + b_n \sin(nx)$$

$$a_n \sin(nx) + b_n \cos(nx)$$



函數 $s(x)$（紅色）是六個不同幅度的諧波關係的正弦函數的和。它們的和叫做傅立葉級數。傅立葉轉換 $S(f)$（藍色），針對幅度與頻率進行描繪，顯示出6種頻率和它們的幅度。

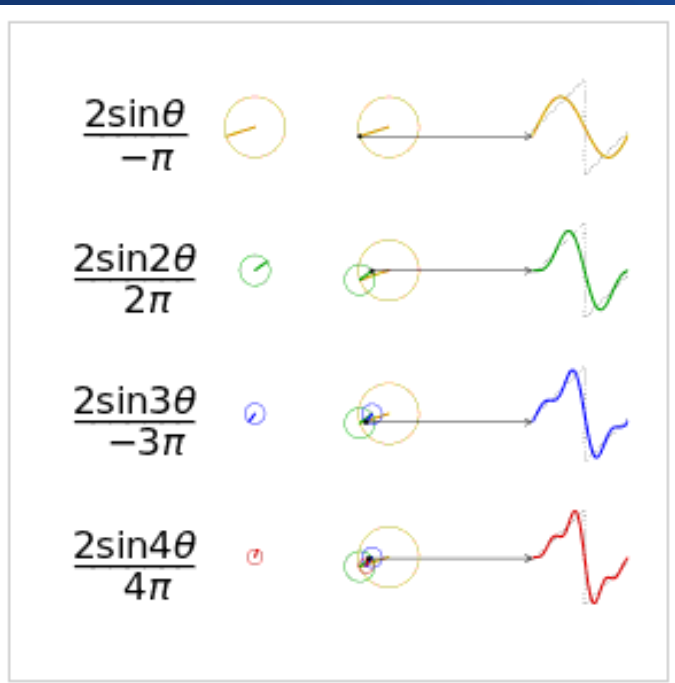
https://en.wikipedia.org/wiki/Fourier_series

傅立葉級數改成積分版

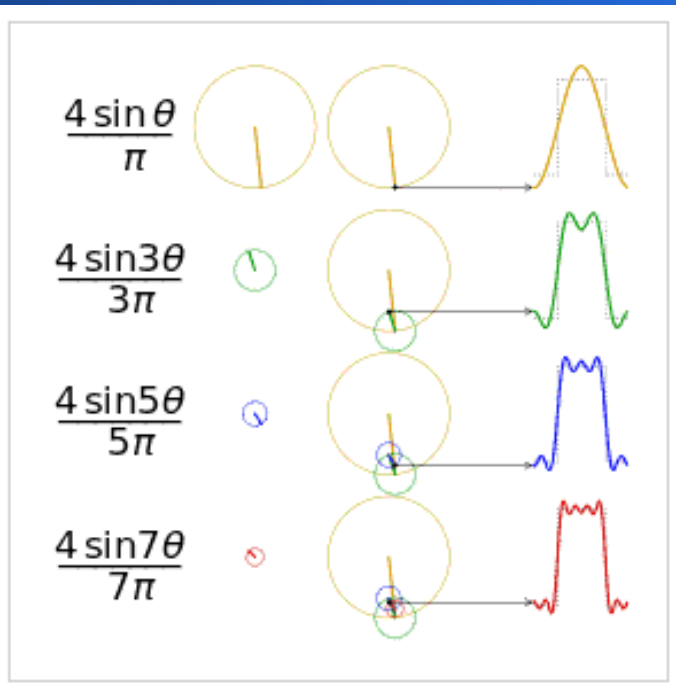
- 就變成了《傅立葉轉換》
- 《快速傅立葉轉換算法》可以很有效的將函數轉換成頻譜領域的 sin, cos 組合

而且數學上已經證明

- 《傅立葉轉換》可以逼近任意週期函數！



一個相同幅度和頻率的鋸齒波的近似的可視化



另一個分別採用傅立葉級數的前 1, 2, 3, 4 項近似方波的可視化。（可以在這裡看到一個交互式的動畫）

https://en.wikipedia.org/wiki/Fourier_series#/media/File:Fourier_series_square_wave_circles_animation.gif

所以若從上述公理系統的角度來看

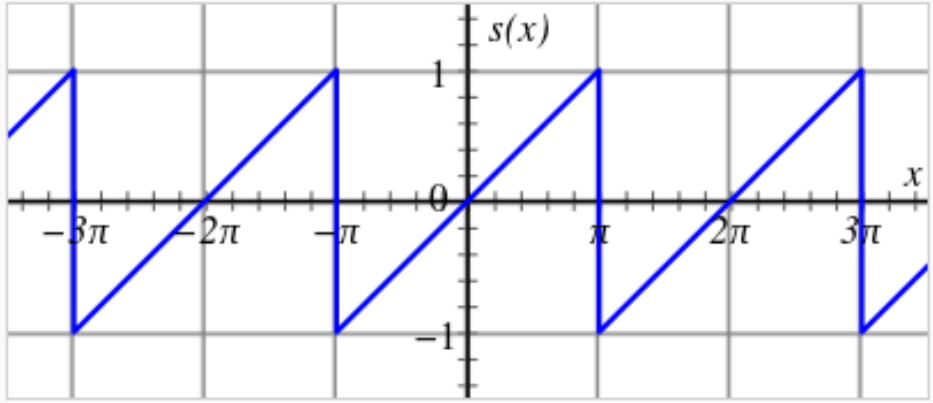
- 傅立葉轉換其實是

  《以三角函數為公理》

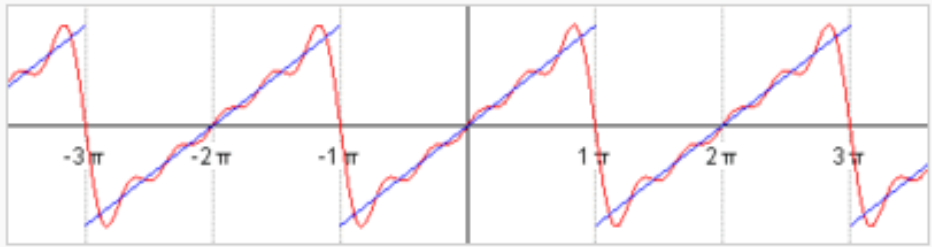
  的函數逼近方法！

即使是不可微分函數

- 傅立葉級數也可以有很好的逼近效果！



Plot of the sawtooth wave, a periodic continuation of the linear function $s(x) = x/\pi$ on the interval $(-\pi, \pi]$



Animated plot of the first five successive partial Fourier series

所以如果把 sin, cos 納入公理系統

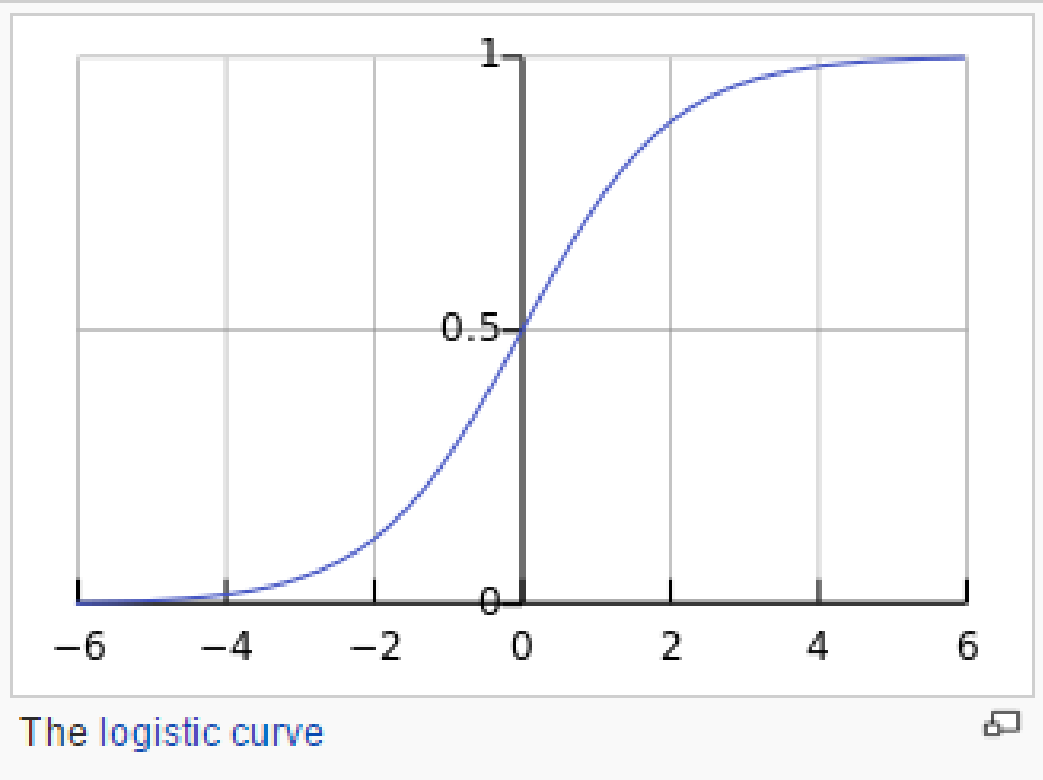
- 就可以用《快速傅立葉轉換》當成推理法則，找出最佳的逼近函數！

在神經網路領域

- 有一種可微分的 sigmoid 函數

$$S(t) = \frac{1}{1 + e^{-t}}$$

- 神經網路就是建構在這類函數上的

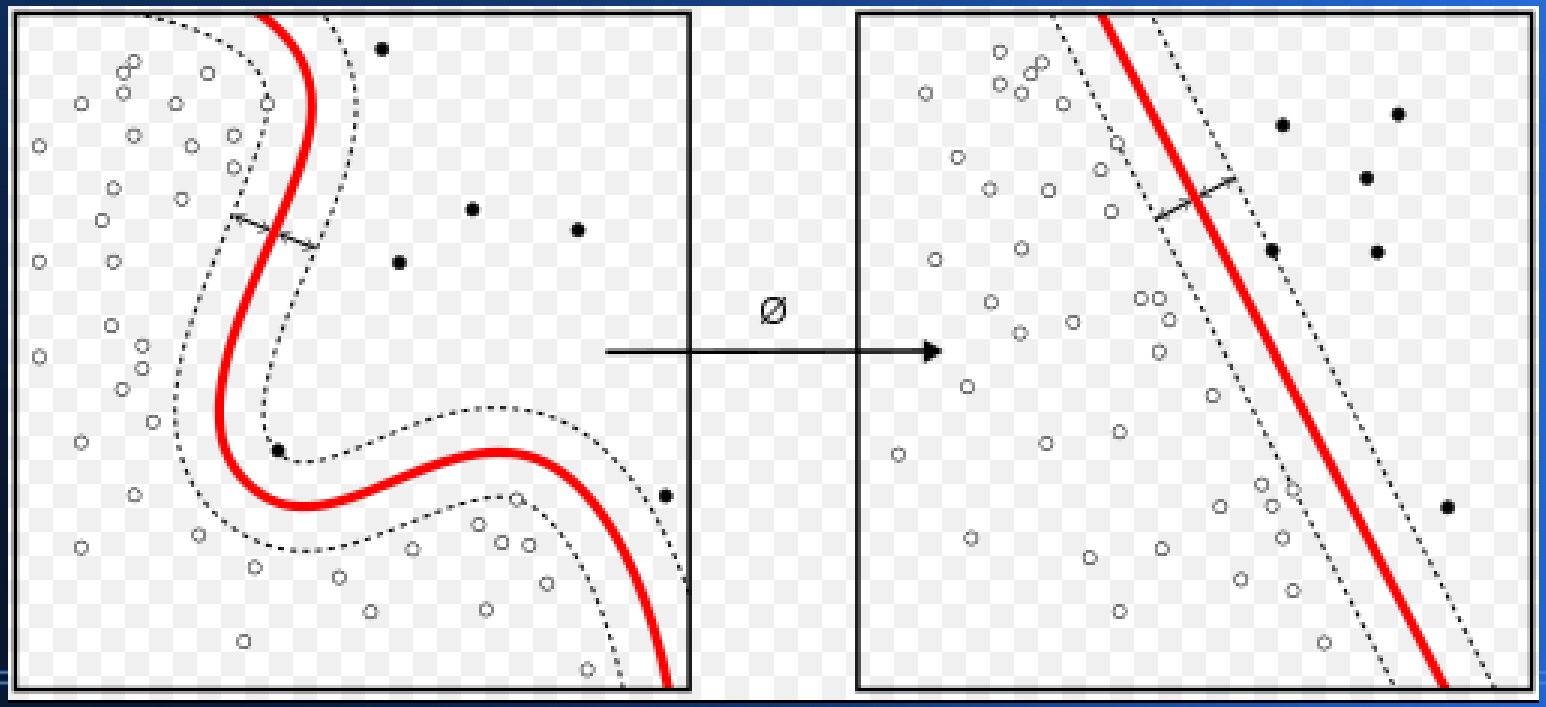


The logistic curve

透過 sigmoid 函數

- 神經網路會找出一分類最佳化函數

  降低整個網路輸出結果的錯誤率！



因此

- 神經網路是以 sigmoid 這類函數為公理的逼近方法！

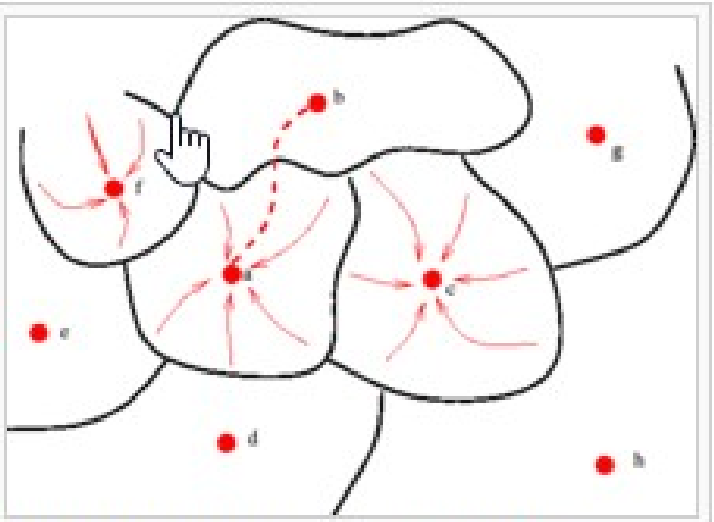
註：也有人用 $\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ 函數取代 sigmoid

tanh 和 sigmoid 兩者的曲線形狀非常相似！

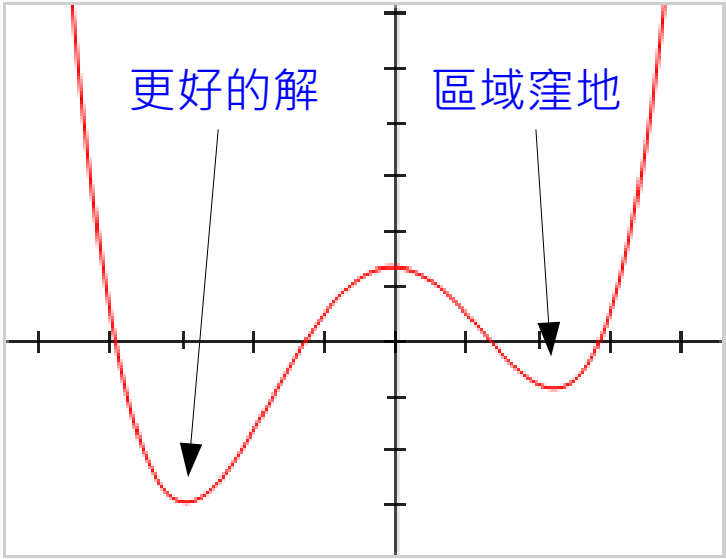
但是神經網路的學習

- 常常很容易困在《區域窪地》，

  而難以找到更好的解！



Attraction basins around locally optimal points



Polynomial of degree 4: the trough on the right is a local minimum and the one on the left is the global minimum. The peak in the center is a local maximum.

雖然在 1986 年

- Hinton 等人就已經發展出了《反傳遞演算法》，透過《偏微分的梯度下降方式》，可以找到神經網路錯誤率低窪點的函數。
- 但還是常常受困在不夠好的區域低點，無法找到很好的逼近函數！

後來在 2002 年

- Hinton 等人又發展出一種快速在《受限波茲曼機 RBM》上迭代尋找低窪點（降低能量）的方法。
- 這個方法的多層套疊表現良好，結果導致《深度學習》領域的大發展！

所以要在人工智慧領域有所突破

- 從上述的《公理系統角度》
- 我們似乎應該關注兩件事！

那就是

- 公理：
  - 可用的《函數群》是那些
- 推論法則：
  - 可用的《運算逼近法則》是那些！

讓我們用公理系統的角度回顧一下

- 看看上述的幾種函數逼近方法，包含：
  - 最小平方迴歸分析
  - 傅立葉轉換
  - 神經網路學習

分別是用甚麼公理和推論法則的！

最小平方迴歸分析

- 最小化： $Y = XA + E$ 當中的 E
- 公理：可用的《函數群》
  - 線性函數
- 推論法則：可用的《運算逼近法則》
  - $A = (X^T X)^{-1} X^T Y$

傅立葉級數逼近

- 最小化：

$$s(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(nx) + b_n \sin(nx)]$$

- 公理：可用的《函數群》
  - $a_n \cos(nx)$, $b_n \sin(nx)$
- 推論法則：可用的《運算逼近法則》
  - 傅立葉轉換

反傳遞神經網路

- 最小化： $E = \frac{1}{2}(t - y)^2$

E 為平方誤差，
t 為訓練樣本的目標輸出，
y 為輸出神經元的實際輸出

- 公理：可用的《函數群》
  - $o_j = \varphi(\mathrm{net}_j) = \varphi\left(\sum_{k=1}^{n} w_{kj} o_k\right)$

$$\varphi(z) = \frac{1}{1 + e^{-z}}$$

- 推論法則：可用的《運算逼近法則》
  - 梯度下降法 $\Delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}}$

受限波茲曼機

- 最小化： $$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j h_j w_{i,j} v_i$$
- 公理：可用的《函數群》
  - $$\varphi(z) = \frac{1}{1+e^{-z}}$$
- 推論法則：可用的《運算逼近法則》
  - $$\Delta w_{i,j} = \epsilon(vh^{\mathsf{T}} - v'h'^{\mathsf{T}})$$

透過這樣的《公理系統》類比

- 如果你想改良某個《函數逼近法》
- 基本上有幾個途徑，包含：
  - 修改最小化函數
  - 修改公理的可用函數群
  - 修改可用的推論法則（運算逼近法則）

以上就是

- 我所看到的：

《最佳化問題與公理化方法的關係》

希望這樣的看法

- 能對您有所幫助！

歡迎多多利用

- 我們《十分鐘小論文》裏的所有資源
- 但是記得要用維基百科的分享方式喔！

因為知識和程式

- 就是要被分享利用才有價值阿！

這就是我們今天的

- 十分鐘小論文！

希望您會喜歡

我們下回見！

Bye Bye!