

AI 教育的迷思：需要少兒編程嗎？

By 高煥堂/銘傳大學 教授

我們來深刻掌握 AI 的特質。對於一個中小學生而言，並不需要去編寫程式碼來表達他/她心中的規則。所以，學習 AI 時，並不需要先學習編程邏輯。換句話說，為了學習 AI，而主張先學習編程(如少兒編程)，極可能是教育界的普遍迷思(Myth)。

在學習 AI 時，最常見的迷思是：延續傳統 IT 思維，想把自己心中的規則(Rules)輸入給電腦。例如，如何讓 AI 來計算二進位的 (011)和(011)相加呢？傳統 IT 是利用程式(如 Python)的<編程邏輯>來把心中的規則表達於 Python 程式碼裡，經過編譯(Compile)、連結(Link)之後，載入到電腦裡。

例如，計算二進位的 (011)和(011)相加時：

進行Binary加法

Ex. $(011)_2 + (011)_2 = (110)_2$

學生們要學習二進位加法的基本規則是：個位數 1 和 1 相加，得到 0，且進位 1。下一位則是：1 和 1 和進位 1 相加，得到 1，且進位 1。再下一位則是：0 和 0 和進位 1 相加，得到 1。於是，得到結果是：二進位的 110。然後，寫成 Python 程式碼來表達之：

```
a = int('011', 2)
b = int('011', 2)
c = a + b
print(bin(c))
```

這是傳統 IT 邏輯思維，不是 AI 的邏輯思維。AI 的邏輯思維是：人類只需要給它(電腦)答案，由它自己歸納出規則。例如，

	A	B	C	D	E
1	A	B			T
2	1	1			0
3	1	1			1
4	0	0			1
5					
6					
7					

人們只要給電腦考卷(即輸入值 011 和 011)，並且給予答案(即輸出值 110)就可以了。AI 就能自己歸納出規則，並計算出非常接近正確的答案：

	A	B	C	D	E	F	G	H
1	A	B				T	Z	Error
2	1	1				0	0.06	-0.06
3	1	1				1	0.95	0.05
4	0	0				1	0.99	0.01
5								
6								
7								

AI 計算出來的答案：[0.99, 0.95, 0.06]，只是非常接近正確答案：[1,1,0]。AI 自己以權重來表達它自己歸納出來的規則。

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	A	B			WO[]	-0.18808					T	Z	Error
2	1	1				1.76411					0	0.06	-0.06
3	1	1				-4.1676					1	0.95	0.05
4	0	0				2.957159					1	0.99	0.01
5													
6					WH[]	0.309538	0.8413023	-1.3649	1.074064				
7						0.130715	0.7016351	-0.7311	1.080796				
8						0.742221	1.4802927	-2.27613	2.063437				
9						-0.2973	1.0021918	-1.16479	1.316522				
10													
11					WX[]	0.627594	-0.424885	1.177227	-0.78673				
12						0.427595	-0.424885	0.977224	-0.58673				

傳統的 IT 邏輯編程，是讓人類表達其心中的規則，以程式碼敘述出來，植入到電腦中，讓電腦替人類快速執行(規則)。所以，許多人要學習編程。

如今的 AI，並不需要人類去表達心中的規則，反而 AI 基於大數據而能歸納出比人類更優質、可信的規則。

既然，AI 不需要人類的規則，AI 的學習則當然也不需要先花時間去苦學 Python 編程了，更不需要<少兒編程>了。

~ End ~