Week 3 written assignment

1 背景與動機：為何要用神經網路來近似多項式？

在深入探討這兩個 Lemma 之前，我們需要先理解作者為什麼要這麼做。

神經網路是由許多簡單的神經元所組成，而每個神經元通常會包含一個激活函數。這篇論文的主角是 **tanh**，這是一個平滑的"S" 型曲線。整個神經網路的「表達能力」在很大程度上取決於這個激活函數。

一個根本性的問題是：一個使用 **tanh** 活化函數的神經網路，究竟能用來做什麼？它能準確地模仿或「近似」哪些複雜的數學函數？

根據數學上非常重要的定理:**Weierstrass Approximation Theorem**，任何一個在閉區間上的連續函數，都可以被一個多項式無限精確地近似。

這給了我們一個想法：

1. 如果我們能證明 tanh 神經網路可以精確地近似任何一個單項式 ，例如 x^2, x^3, x^p 等。

2. 那麼我們就能將這些單項式的近似組合起來，去近似任何一個多項式。

3. 這意味著 tanh 神經網路有潛力去近似任何一個連續函數。

因此，Lemma 3.1 和 3.2 正是這個證明中最關鍵的第一步：證明 tanh 神經網路有能力建構出所有多項式的基本組成單位——單項式。

2 Lemma 3.1 ：建構奇數次方的單項式

白話解釋

> 對於任何奇數 s，我們可以建構一個只有一層隱藏層的 'tanh' 神經網路，這個網路可以無限精確地近似所有次方不高於 s 的奇數次單項式（例如 $x, x^3, \ldots, x^s$）。不僅函數值本身可以做到無限接近，就連它的一階、二階 ... 等多階導數也能同時做到無限接近。

想法：利用微積分的「有限差分」

在微積分中，我們知道二階導數可以被「中央有限差分」來近似：

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

這個公式的核心是透過對原始函數進行平移和加權求和，來得到其導數的近似值。更高階的導數也有類似的公式。

Lemma 3.1 的證明正是利用了這個概念，但目標不是求導數，而是反過來建構單項式：

1. 以 'tanh' 函數為基礎，一層隱藏層的神經網路所能做的事，就是對 'tanh' 函數進行平移、縮放和加權求和，例如：$c_1 \cdot \tanh(w_1 x + b_1) + c_2 \cdot \tanh(w_2 x + b_2) + \dots$。
2. 作者發現，如果將 'tanh' 函數本身代入到一個 p 階的有限差分公式中（p 為奇數），並在 $x = 0$ 點進行計算，再經過適當的縮放，所得到的組合函數會非常接近單項式 y^p。
3. 這個近似的誤差與參數 h（有限差分中的步長）的平方 h^2 成正比。意思是只要我們把 h 選得足夠小，就可以讓近似誤差變得任意小。而選擇不同的 h 值，對應到神經網路中就是調整 weight 和 bias。

總結來說，Lemma 3.1 告訴我們，僅僅透過對 'tanh' 函數進行平移和線性組合，就能「拼湊」出任何奇數次方的單項式，且精確度要多高有多高。

3 Lemma 3.2 解析：從奇數次方到偶數次方

白話解釋

> 我們可以建構一個只有一層隱藏層的 'tanh' 神經網路，這個網路可以無限精確地近似所有次方不高於某個奇數 s 的所有單項式（包含奇數和偶數次方）。

想法：遞迴式的代數技巧

直接用 Lemma 3.1 的方法來建構偶數次方的單項式會遇到一些技術困難。所以作者採用了另一個更聰明的策略：利用已知的奇數次方來遞迴地建構偶數次方。

這個策略的基礎是下面這個恆等式（論文中的公式 (25)）：

$$y^{2n} = \frac{1}{2\alpha(2n+1)} \left((y+\alpha)^{2n+1} - (y-\alpha)^{2n+1} - 2\sum_{k=0}^{n-1} \binom{2n+1}{2k} \alpha^{2(n-k)+1} y^{2k} \right)$$

這個公式看起來很複雜，但它的含義非常直觀：

一個「偶數次方」(y^{2n}) 可以被表示成由兩個「奇數次方」($(y+\alpha)^{2n+1}$ 和 $(y-\alpha)^{2n+1}$) 和一些「更低階的偶數次方」($y^{2k}, k < n$) 組合而成。

這就提供了一個遞迴的建構方法：

1. 我們已經從 Lemma 3.1 知道如何用神經網路近似任何奇數次方了。
2. 第一步 **(n=1)**：我們想建構 y^2。根據公式，我們需要 y^3（奇數）和 y^0（常數，很簡單）的近似網路。我們都有，所以可以組合出 y^2 的近似網路。
3. 第二步 **(n=2)**：我們想建構 y^4。根據公式，我們需要 y^5（奇數）和 y^2, y^0（更低階的偶數）的近似網路。我們也都有了，所以可以組合出 y^4 的近似網路。
4. 以此類推：我們可以一步步地建構出所有偶數次方的近似網路。

因為在建構過程中，我們需要用到對中心點在 y、$y+\alpha$ 和 $y-\alpha$ 的函數進行近似，這意味著我們需要三組神經元。這也解釋了為什麼 Lemma 3.2 所需的神經網路寬度（神經元數量）大約是 Lemma 3.1 的三倍。

4 結論

Lemma 3.1 和 Lemma 3.2 證明了：

- **Lemma 3.1**：淺層 'tanh' 神經網路可以透過有限差分的思想，精確構造出奇數次單項式。
- **Lemma 3.2**：基於奇數次方的成果，可以再透過代數遞迴的方式，將構造能力擴展到偶數次單項式。