在 lemma 3.1 中，

他展示在 $k \in \mathbb{N}_0$，s 為奇數的情況下，保證

存在 shallow tanh neural network $\Psi_{s,\varepsilon} : [-M, M] \to \mathbb{R}^{\frac{s+1}{2}}$

使得 $\max\limits_{p \le s,\ p \text{ odd}} \| f_p - (\Psi_{s,\varepsilon})_{\frac{p+1}{2}} \|_{W^{k,\infty}} \le \varepsilon$.

並且，the weight of $\Psi_{s,\varepsilon}$ 會隨著 ε 變小與 s 變大，以 $O\left(\varepsilon^{-\frac{\varepsilon}{2}} \left(2(s+2)\sqrt{2M}\right)^{s(s+3)}\right)$ 增長。

這個 lemma 保證對於每個奇函數 $f_p(y) = y^p$，$p \in 2\mathbb{N}-1$ and $p \le s$.

我們都可以用一個 shallow（淺層） tanh neural network 在 $[-M, M]$ 的

區間上逼近到任意小 error ε，但隨著 error ε 變小與次方 s 變大，weight 會大幅增加。

※ 在 error 的衡量上，使用 $W^{k,\infty}$. 他保證了函數的逼近不是單純的點對點靠近，而是確保從一階到 k 階導數的 max error $\le \varepsilon$. 所以函數的形狀也被控制而保證逼近。

※ Why we only pick odd p?

因為 tanh 是奇函數，較容易 match.

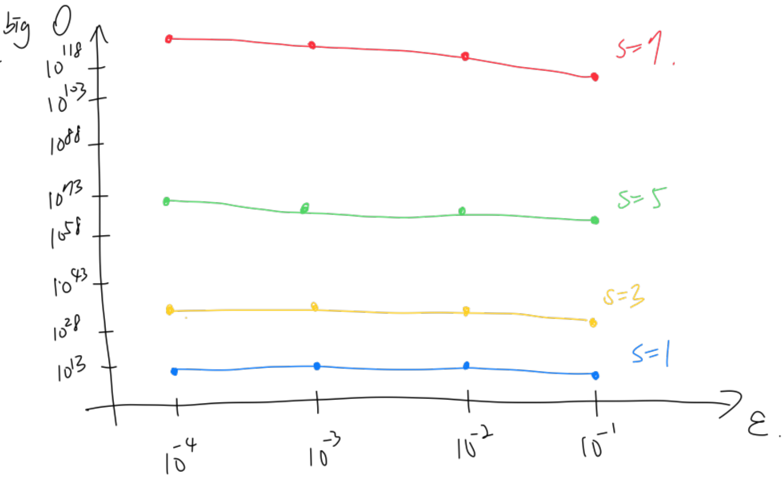
The weight $\Psi_{s,\varepsilon}$ increasing by s and ε. (The following graph getting by the code chatGPT)

Setting. $s = \{1, 3, 5, 7\}$.

$\varepsilon = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$.

$M = 1$

Then



我們可以透過 lemma 3.1 去看 lemma 3.2，

在 lemma 3.2 中，$\Psi_{s,\varepsilon} : [-M, M] \to \mathbb{R}^{s}$ of width $\frac{3(s+1)}{2}$.

且 p 不再侷限於奇數。

並且，the weight scale as $O\left(\varepsilon^{-\frac{2}{2}} \left(\sqrt{M}(s+2)\right)^{\frac{3s(s+3)}{2}}\right)$ for small ε and large s.

在 lemma 3.2 中，逼近不再侷限於奇函數，

而是對於所有的 $f_p(y) = y^p$，$p \le S$ 都能保證存在一個 shallow tanh neural network

足夠逼近在 error ε。

lemma 3.1、3.2 的重要性：

① 建立多項式的逼近

從 lemma 3.1 的逼近奇函數到 lemma 3.2 的逼近所有次數 $\le S$ 的多項式。

多項式可以是後面擴展更加複雜函數的基礎。

② 給予可以量化的 weight

精準度（ε 下降）與多項式次數越高（S 提高）

↓ weight 上升

↓ weight 指數上升

注意：雖然理論上可以逼近任意多項式，但根據 ε、S 的選定，逼近會變得非常困難。